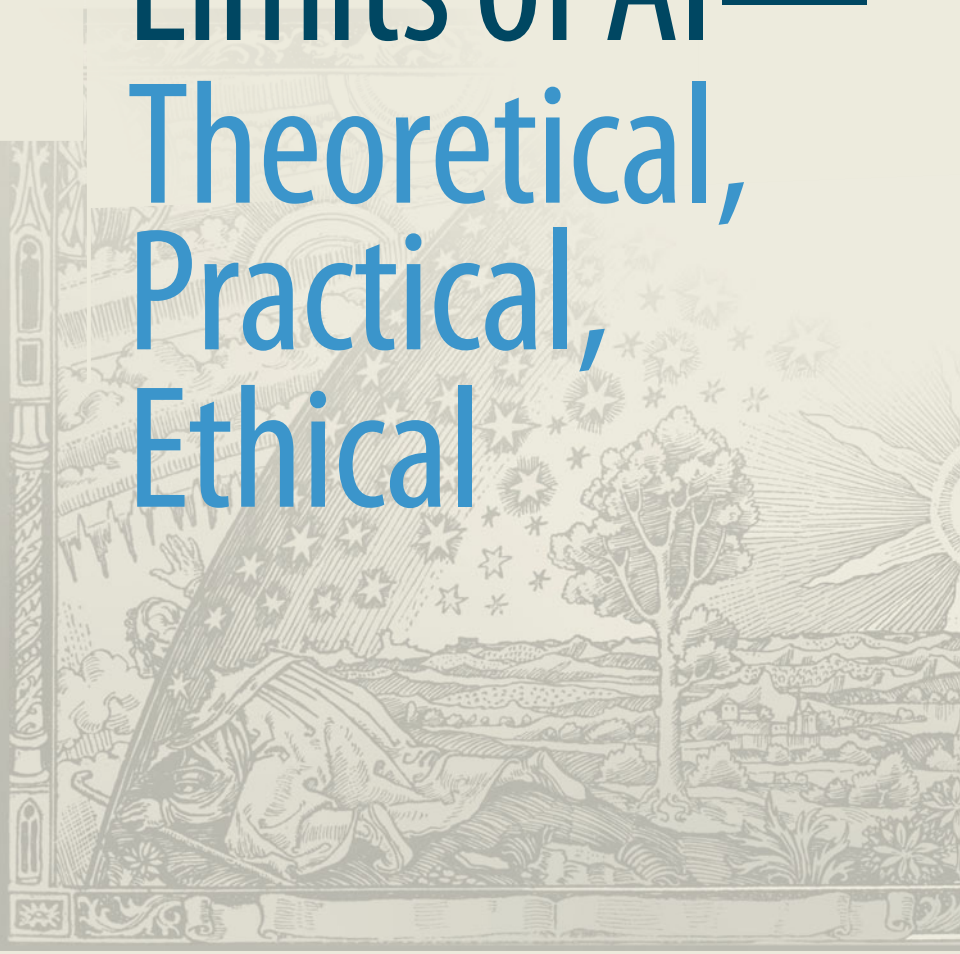
Klaus Mainzer
Reinhard Kahle

Limits of AI—Theoretical, Practical, Ethical



Springer

Technik im Fokus

Die Buchreihe Technik im Fokus bringt kompakte, gut verständliche Einführungen in ein aktuelles Technik-Thema.
Jedes Buch konzentriert sich auf die wesentlichen Grundlagen, die Anwendungen der Technologien anhand ausgewählter Beispiele und die absehbaren Trends.
Es bietet klare Übersichten, Daten und Fakten sowie gezielte Literaturhinweise für die weitergehende Lektüre.

Klaus Mainzer · Reinhard Kahle

Limits of AI— Theoretical, Practical, Ethical

Springer

Klaus Mainzer
TUM Senior Excellence Faculty
Technische Universität München
München, Germany

Reinhard Kahle
Carl Friedrich von Weizsäcker-Zentrum, Universität Tübingen
Tübingen, Germany

ISSN 2194-0770 ISSN 2194-0789 (electronic)
Technik im Fokus
ISBN 978-3-662-68289-0 ISBN 978-3-662-68290-6 (eBook)
https://doi.org/10.1007/978-3-662-68290-6

This Springer imprint is published by the registered company Springer-Verlag GmbH, DE, part of Springer Nature.
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

Paper in this product is recyclable.

Preface (to the English translation)

The German version of this book appeared at the beginning of 2022 before the hype about the ChatCPT started. However, the limits of AI were already clearly shown there, also using the example of the ChatGPT and its precursors. Therefore, one cannot resist the remark that many of the commentators could have saved their dystopian or utopian remarks if they had read our book beforehand. Nevertheless, on the background of the analyses in this book, a summary about the ChatGPT has been added (Sect. 5.3). Although we utilized the translation software "DeepL," the authors bear responsibility for any errors. However, we retain the right to attribute any oversight of these errors to Artificial Intelligence for not alerting us. In many cases, the human authors corrected DeepL, because they have the better background knowledge and understand the universal language of mathematics.

Klaus Mainzer
Reinhard Kahle

Preface (to the first German edition)

In 1972, when the American philosopher Hubert Dreyfus published a bestseller entitled "What Computers Can't Do. The Limits of Artificial Intelligence", he was in fact only pointing out the limits of what we now call "symbolic AI". These were so-called expert systems, which combine the limited specialist knowledge of experts such as doctors and engineers in 'rule-based knowledge systems'. Dreyfus rightly pointed out the limits of this approach to intuitive knowing: The first hours of driving lessons can be taught in rules that are recorded in textbooks. But then intuitive learning begins and training is needed to become a really good driver. Anyone who has ever tried to perfect the stroke of a golf ball by following rules knows immediately what is meant.

After the paradigm of logic-based rule systems in the 1970s, the training of neural networks, the so-called connectionist paradigm, emerged. The connectionist paradigm overcame many of Dreyfus' limitations. The philosopher therefore somewhat meekly gave a later edition of his book the title "What Computers Still Can't Do".[1] Once again that one should be careful with apodictic demarcations. They can only apply to certain domains, systems, bodies of knowledge and preliminary stages of development.

[1] There is a certain ambiguity in the word "Still"; it could be understood as "not yet" or it come with the connotation "still, and never".

Even these boundaries, however, are only partially of interest. Still today, rule-based expert systems are highly elaborated and successfully applied in industry (e.g. logistics in the automotive industry) and medicine (e.g. control systems), without us perceiving them as spectacular "AI". The drawing of boundaries therefore does not mean that systems are outdated, but that we only know more precisely, what they can and cannot do.

Even more interesting are the limits that may arise from logic and mathematics. In logic and mathematics there exist problems which have not yet been solved or decided. Therefore, AI that depends on such problems will have only provisional limits. It is more interesting when we are dealing with problems that cannot be decided in principle. What is undecidable in principle? In this case, both natural and artificial intelligence reach their limits in principle. But the key question is: How does natural intelligence of mathematicians find solutions? An analysis of the mathematical background knowledge used by humans raises doubts as to whether AI would ever be able to do this. But it cannot be ruled out in principle.

Now one might think that these kinds of analyses are so abstract that they are irrelevant for the practical application of AI. Let some nerds in their ivory towers deal with it! In the meantime, the AI community will make a lot of money from "this side" AI and will shake up industry and society! But in fact the seemingly abstract mathematical questions we are referring, are directly connected with, for example, security issues in cryptography. This is not only when quantum computers are available! But their technical feasibility, together with the already implemented quantum communication, concerns the question of the mathematical limits of AI once again with additional explosiveness for practical applications. So let us enter the ivory towers of computer science, mathematics and philosophy, knowing very well that, only in this way, we will find the hidden dangers of technical civilisation as if under a magnifying glass.

Klaus Mainzer
Reinhard Kahle

Acknowledgements

Many of the examples given in the text benefited, in Reinhard Kahle's case, from discussions with Klaus Angerer (now Darmstadt), Philipp Hennig (Tübingen), Kristian Kersting (Darmstadt), Christoph Peylo (Renningen), Thomas Sattig (Tübingen) and Elektra Wagenrad (Berlin). Klaus Mainzer benefited from discussions in the steering group for a German Standardization Roadmap on Artificial Intelligence chaired by Wolfgang Wahlster, in the thematic network of the German National Academy of Science and Engineering (acatech) and as President of the European Academy of Sciences and Arts in discussions of the expert group on digitalisation and AI.

We are grateful to these colleagues, but we do not imply that they always share our view of the problems.

The second author was supported by the Udo Keller Foundation and the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the scope of the projects UIDB/00297/2020 and UIDP/00297/2020 (Center for Mathematics and Applications)". The first author is chairman of the Board of Trustees of the Udo Keller Foundation.

Contents

The Concept of Artificial Intelligence

1

> The term AI contains an explicit reference to the notion of intelligence. However since intelligence (both in machines and in humans) is a vague concept, although it has been studied at length by psychologists, biologists, and neuroscientists, AI researchers use mostly the notion of rationality, which refers to the ability to choose the best action to take in order to achieve a certain goal, given certain criteria to be optimized and the available resources.
>
> *European Commission's High-Level Expert Group on Artificial Intelligence* [1].

Effective methods for problem-solving have been known since ancient mathematics. In geometry, the construction of a figure is split into elementary steps with compass and ruler. In arithmetic and algebra, methods of solving equations are split into elementary steps which, in principle, can be carried out by a machine. Thus, one speaks of algorithms, which are named after the Persian mathematician Al-Chwarizmi. Today, algorithms are executed by computer programs. The question is, to what extend steps cannot only be executed by a machine, but also found independently.

Artificial intelligence (AI) is therefore measured against human intelligence. According to the British logician and computer pioneer Alan M. Turing [2], a technical system is called "intelligent" if its answers and its way to solve problems cannot be distinguished from a human being. Originally, AI was oriented towards the rules and formulas of symbolic logic, which

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical*, Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6_1

were translated into suitable computer programs. One, therefore, also speaks of symbolic AI (Fig. 1.1). The underlying epistemological idea is that intelligence is primarily related to the ability of the human mind to derive logical conclusions.

One example was automatic reasoning, in which AI programmes simulated logical reasoning in logic calculi [3]. On this rule-based and symbolic basis, human planning, decision-making and problem solving of human experts should also be simulated in specialised fields of application. In corresponding expert systems or knowledge-based systems, the specific knowledge of an engineer or doctor, for example, is first translated into formal rules which should trigger a specific action automatically when a certain event occurs.

One product, which emerged from this approach is the programming language Prolog (French: programmation en logique), which still today enjoys a certain popularity, even though it is effectively used only in the theoretical sphere. It has not been able to establish itself in the industrial field for reasons that are certainly related to the limits of AI to be discussed here. In Prolog, (simple) rules can be formulated, for example, to store a network of flight connections.

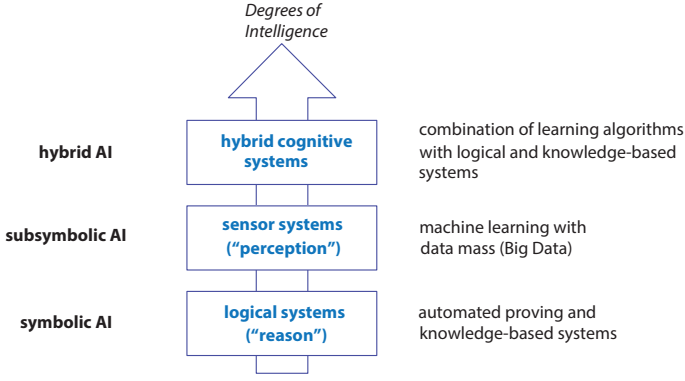


Fig. 1.1 From symbolic and sub-symbolic to hybrid AI

Example

```
reachable(X,Y) :- direct-flight(X,Y).
reachable(X,Y) :- direct-flight(X,Z), reachable(Z,Y).
directflight(NCY,DUB).
directflight(DUB,GWY).
directflight(DUB,ORK).
⋮
```

Prolog is a query language in which, for the given example, the question

```
?- reachable(NCY,GWY)
```

should return the answer `Yes`. ◀

To the extent that this type of knowledge representation was developed further, increasing problems of complexity arose—in two different meanings of "complexity": on the one hand, the general complexity of, for example, the grammar of a language is in general so complex that a simple translation into Prolog rules turns out to be impracticable. On the other hand, problems of computability complexity arise, for example, when querying all theoretically possible flight connections—mathematically the transitive closure of `reachable(X,Y)`—leads to calculation times that are no longer acceptable. Because of these problems, expert systems went out of fashion comparatively quickly.

However, it would be a misinterpretation to restrict research in artificial intelligence in the second half of the twentieth century to the field of expert systems. In particular SAT-solving has emerged from the considerations on automatic theorem proving. This SAT-solving has today far-reaching applications. In addition, motivated by the findings in neurological brain research, neural networks have also been developed as simplified computer simulations of the human nervous system, described with the aid of neurons. From the beginning, this approach was conceptually distinct from the rule-based systems, but hardly

progressed beyond "toy applications"—not least due to the still comparatively limited memory and computing power of the available computers. These toy examples, although simulating neural networks in principle, did not yet allow any practical applications.

In a simplified form, the research fields of classical or old AI, as they emerged at the end of the twentieth century, can be summarised as follows:

Classic or Old AI

- *Expert systems*
  Prolog as a paradigmatic programming language.
- *SAT-Solving*
  problem-solving methods for propositional logic which solve complex problems that are just feasible in terms of complexity theory
- *Early neural networks*
  In the early phase of AI, only of very limited complexity.

The early neural networks were already a response to the fact that rule-based knowledge can never fully capture the intuitive skills of an expert. Knowledge is based on manifold experiences that are by no means symbolically represented in a textbook. An experienced driver realizes situations and reacts intuitively on the basis of a great deal of sensory data, without being aware of the logical processes in detail. In the same way an experienced doctor reacts in a critical situation as well as an experienced pilot in the cockpit of an aircraft. Intuition is by no means a mystical magic box. Rather, the recognition of data patterns and the estimation of expected probabilities can be trained and improved through experience.

In this context, logical rules, as in symbolic AI, are replaced by sensory data, in which statistical correlations and probabilities are determined. Learning from data is studied mathematically

in statistical learning theory. Its algorithms form the basis of machine learning. From an epistemological point of view, these learning processes from sensory perceptual data take place unconsciously below conscious logical reasoning. This is why one also speak of subsymbolic AI (Fig. 1.1). Mathematically, the paradigm of logic is replaced by statistics and probability theory. The powerful computer technology of the past few years made it possible that machine learning with big data can now be implemented technically. Therefore, machine learning leads to new breakthroughs in the application of AI, e.g. in the development of drugs and vaccines.

Accordingly, at the beginning of the twenty-first century, a statistics-based or new AI has emerged, with the following characteristics.

Statistics-based or new AI

- *Machine learning*
  fed by
  - *Large amounts of data* ("Big Data")
    and often based on a high number of layers in neural networks, which enables
  - *deep learning*.

It should be noted, however, that the term "deep" is not to be understood in the sense of "profound", but only emphasises the aspect of a considerable extension of layers in the network which are comparable to the layers in a human brain. This new AI is thus a manifestation of subsymbolic AI and represents essentially a tool assisting human perception in a form that is optimised in many respects.

However, human intelligence can neither be reduced to the logic of the mind nor to the data of perception. Epistemologically, it depends on the connection between perception and understanding. In AI research, therefore, the future goal is to combine

statistical learning algorithms with logical and knowledge-based methods. The connection of symbolic and sub-symbolic AI is also called hybrid AI (Fig. 1.1). Epistemologically, it corresponds to a "hybrid" cognitive system like the human organism, in which the ("unconscious") processing of perceptual data is combined with ("conscious") logical reasoning. Hybrid AI is therefore assigned higher degrees of intelligence than the reduction to symbolic or subsymbolic AI. Nevertheless, all three forms of AI are in practical use side by side, depending on the respective requirements of the field of application. In the automotive industry and medicine, for example, we still find knowledge-based expert systems and machine learning for different applications. Hybrid AI is already being developed in robotics.

This understanding of artificial intelligence, which has moved remarkably far away from Turing′s imitation game, is also officially propagated by the European Union. The European Commission′s High-Level Expert Group on Artificial Intelligence has published a report entitled "A Definition of AI: Main Capabilities and Scientific Disciplines" with the following "updated definition of AI" [1, p. 9]:

> Artificial intelligence (AI) refers to systems designed by humans that, given a complex goal, act in the physical or digital world by perceiving their environment, interpreting the collected structured or unstructured data, reasoning on the knowledge derived from this data and deciding the best action(s) to take (according to predefined parameters) to achieve the given goal. AI systems can also be designed to learn to adapt their behaviour by analysing how the environment is affected by their previous actions.

This definition is supplemented by a paragraph on AI as a scientific discipline. Here, however, we consider the possible limits of AI, as they arise for the various tasks mentioned in this definition.

References

1. High-Level Expert Group on Artificial Intelligence (2018). *A Definition of AI: Main Capabilities and Scientific Disciplines*. European Commission, Directorate-General for Communication.
2. Turing, A. (1950). Computing machinery and intelligence. Mind 59, 433–460.
3. Robinson, J.A. (1965), A machine oriented logic based on the resolution principle, in: Journal of the Association for Computing Machinery 12, 23–41.

Practical Limits

2

2.1 The Fate of Expert Systems

> The view that machines cannot give rise to surprises is due, I believe, to a fallacy to which philosophers and mathematicians are particularly subject. This is the assumption that as soon as a fact is presented to a mind all consequences of that fact spring into the mind simultaneously with it. It is a very useful assumption under many circumstances, but one too easily forgets that it is false.
>
> Alan Turing, [1, p. 451].

Knowledge-based expert systems are computer programs that store and accumulate knowledge about a specific field. They automatically draw conclusions in order to offer solutions to concrete problems in the field. In contrast to the human expert, the knowledge of an expert system is limited to a specialised information base without a general and structural knowledge about the world [2].

In order to build an expert system, the knowledge of the expert has to be put into rules, translated into a programming language and processed with a problem-solving strategy. Expert systems are, thus, a typical example of symbolic AI. The architecture of an expert system therefore consists of the following components: Knowledge base, problem-solving component (derivation system), explanation component, knowledge acquisition, dialogue component. The coordination of these components is shown in Fig. 2.1.

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical*, Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6_2

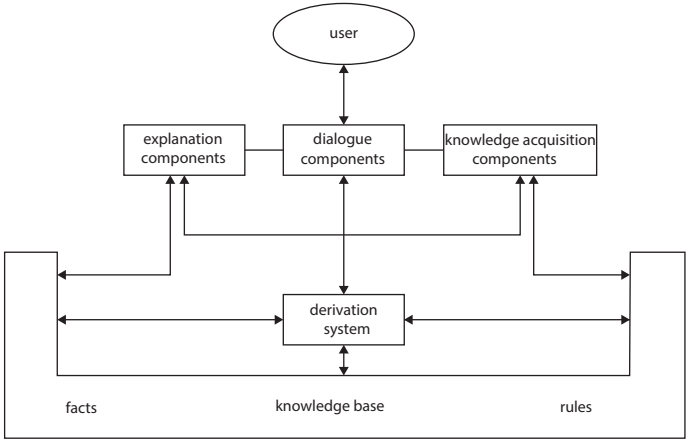


Fig. 2.1 Architecture of a knowledge-based expert system

Knowledge is the key factor in the representation of an expert system. A distinction is made between two types of knowledge. One type of knowledge concerns the facts of the field of application, which are recorded in textbooks and scientific journals. Equally important is the practice in the respective field of application. This is heuristic knowledge, on which judgement and every successful problem-solving practice in the field of application are based.

It is empirical knowledge, the art of successful guesswork that a human expert acquired over many years of professional work. The heuristic knowledge is the most difficult one to represent, since the expert himself is usually not aware of it. Therefore, interdisciplinary trained knowledge engineers have to learn the expert rules of the human experts, represent them in programming languages, and integrate them into a functional work program. This component of an expert system is called knowledge acquisition.

The explanation component of an expert system has the task of explaining the steps of the system to the user. The question "How" aims at explaining facts or assertions that are derived by the system. The question "Why" demands reasons for questions

or commands of a system. The dialogue component concerns the communication between expert system and user.

Knowledge representation is usually rule-based. For the application in expert systems, rules are understood as if-then statements where the precondition (premise) describes a situation in which an action has to be carried out. This can be a deduction, according to which a statement is derived from a prerequisite. An example is when an engineer concludes from certain symptoms of an engine that an engine piston is defective. However, a rule can also be understood as a set of instructions for action in order to change a state. If, for example, a piston is defective, then the engine must be switched off immediately and the defective part must be replaced.

A rule-based system consists of a database with the valid facts or states, the rules for deriving new facts or states and the rule interpreter for controlling the derivation process. There are two alternatives for linking the rules, which are called forward reasoning and backward reasoning in AI (Fig. 2.2) [3, 4].

In forward reasoning, starting from an existing database, the following is done: One rule is selected from those whose precondition are fulfilled by the database. The action part of this rule is executed and the database is updated. This process is repeated until no more rules are applicable. The procedure is therefore data-driven. In a preselection, the rule interpreter, as part of the

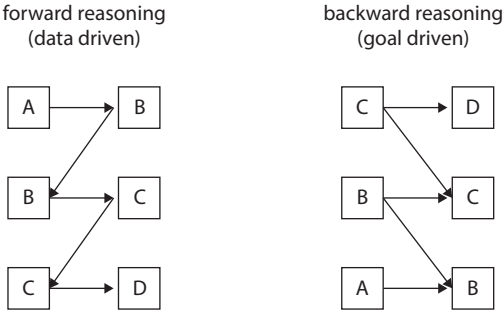


Fig. 2.2 Forward and backward reasoning

respective expert system, initially determines systems of all executable rules that can be derived from the database. Then a rule is selected from this set according to certain criteria. Thus, a specific sequence, the structure of a rule, or additional knowledge can be decisive.

In the case of backward reasoning, starting from a target, only those rules are checked whose action part contains the target. The procedure is therefore goal-driven. If parts of the precondition are unknown, they are requested or derived with other rules. The backward reasoning is particularly suitable when facts of the knowledge base are still unknown and, therefore, have to be requested. The rule interpreter begins with the given goal. If the goal in the database is unknown, the rule interpreter must first decide whether the goal can be inferred or must be requested. If a derivation is possible, all rules are executed whose action part contains the target. Unknown parts must be requested or derived as subgoals.

A qualified expert has a complex basic knowledge, which has to be matched by a structured data structure in an expert system. For such a structuring of knowledge, all statements about an object are often summarised in a schematic data structure. This is also called a frame according to M. Minsky. Historically, Minsky draws on templates from linguistics for the representation of knowledge. The graphical notation of schemata through semantic networks allows a clear representation of complex data structures [5].

In everyday life, cognitive schemata are activated in different situations. This can involve the recognition of typical objects, typical events, or answers to typical questions. The respective fillers of a concrete object are placed in the slots of the frame. In the case of diagnostic tasks of a doctor, the patient′s symptoms can, for instance, be classified in a general disease picture that is represented by a frame.

Relations between objects are often represented by constraints. These representations restrict the capabilities of a problem. One can have constraints, e.g., in the solution of a technical problem by an engineer as well as in the preparation of an administrative planning task. If the problem is mathematised, constraints are

defined by mathematical equations and constraint networks are represented by systems of equations [6].

Historically, DENDRAL was one of the first successful expert systems, developed by E.A. Feigenbaum et al. at Stanford in the late 1960s [7, 8]. It uses the special knowledge of a chemist in order to find a suitable molecular structural formula for a chemical sum formula. In a first step, all the mathematically possible spatial arrangements of the atoms for a given molecular sum formula are determined. For example, for $C_{20}H_{43}N$ there are 43 million arrangements. Chemical knowledge about the bonding topology, according to which, e.g., carbon atoms can be bound many times, reduce the possibilities to 15 million. Knowledge of mass spectrometry, heuristic knowledge of the most probable stability of bonds, and nuclear magnetic resonance narrow down the possibilities to the sought-after structural formula.

The problem-solving strategy that was used as a basis here is obviously nothing other than the familiar "British Museum algorithm"[9],[1] which was written in the programming language LISP. The procedure is thus `GENERATE_AND_TEST`, where in the `GENERATE` part the possible structures are systematically generated, while the chemical topology, mass spectrometry, chemical heuristics, and nuclear magnetic resonance each specify test predicates to limit the possible structural formulae.

For practical purposes, problem-solving types can be divided into diagnostic, construction, and simulation tasks. Typical diagnostic tasks one finds in medical diagnostics, technical diagnostics such as quality control, repair diagnostics or process monitoring, and object recognition. Therefore, DENDRAL also solves a typical diagnostic problem by recognising the appropriate molecular structure for a given molecular sum formula.

The first medical example of an expert system was MYCIN developed at the University of Stanford in the mid-1970s [10, 11]. The MYCIN program was written for medical diagnosis, to simulate a doctor with specialist medical knowledge of bacterial

[1] "... since it seemed to them as sensible as placing monkeys in front of typewriters in order to reproduce all the books in the British Museum.".

infection. Methodologically, it is a deduction system with backward reasoning. MYCIN′s knowledge base on bacterial infections consists of about 300 production rules. In order to be able to apply the knowledge, MYCIN works backwards. For each of 100 possible hypotheses of diagnoses, MYCIN tries to find simple facts that are confirmed by laboratory results or clinical observations. Since MYCIN works in a field in which deductions are hardly certain, a theory of plausible inference and probability evaluation was linked to the deduction apparatus. This involves so-called safety factors for each conclusion in an AND/OR tree.

In this context, F_i denotes the safety factor that a user assigns to a fact. C indicates the safety factor of a conclusion, A_i the degree of reliability assigned to a production rule. At the AND and OR nodes, safety factors of the corresponding formula are calculated. If the safety factor of a data entry is not greater than 0.2, it is considered unknown and receives the value 0. The program thus calculates degrees of confirmation as a function of more or less certain facts. MYCIN was developed independently of its special database on infectious diseases and generalised for various diagnostic diagnostic fields of application.

Practical limits of expert systems were gradually extended: Experts are not characterized by the fact that they can distinguish between true and false with absolute certainty; thus, it is not the task of an expert to provide a greater degree of accuracy than actually achievable. A good expert is able to assess uncertainties that may arise, for example, in the medical diagnosis in symptom recognition or symptom evaluation. In expert systems, therefore, the classical logic with the assumption of the bivalent nature of the truth values (tertium non datur) is often not the adequate approach. Instead, expert systems are based on additional uncertainty values such as “certain”, “probable”, “possible”, and others. It is a longstanding problem of methodology of science that only logical conclusions are valid with certainty. For example, the direct conclusion that infers the truth of B from the assumptions of $A \rightarrow B$ and A. In this case, A is a sufficient condition for B, while B is only necessary for A. Therefore, A cannot be logically derived from $A \rightarrow B$ and B:

In philosophy of science, statistical inference has been studied as much as the inductive degrees of confirmation of a hypothesis, which depend on the extent of confirmation [12]. As a basic algorithm for the evaluation of a diagnosis in expert systems, the following procedure is suitable [13]:

1. Start with the Assumed (a priori) Probabilities of All (possible) Diagnoses;
2. for each symptom, modify the (conditional) probabilities of all diagnoses (according to the frequency of occurrence of a symptom in the presence of a diagnosis);
3. select the most probable diagnosis. (Bayes′s theorem is often used as a general formula for calculating the most probable diagnosis assuming certain symptoms).

Knowledge representations by experts must therefore take uncertainty factors into account. The concepts used by experts are by no means always sharply defined, and yet they are used. Information about colour, elasticity etc. only makes sense with reference to certain intervals. The limits of these intervals then appear to be set arbitrarily. Whether a colour is still black or already grey is considered quite fuzzy for a designer. In the philosophy of science, therefore, the so-called "fuzzy logic" was introduced [14]. Paradoxes are inevitable without appropriate interpretation: if a pile of n straws is described as being large, then a pile of $n - 1$ straws is also large. If one applies this conclusion in an iterated way, then consequently also the empty heap must also be described as large.

The representation of knowledge in logic is based on the fiction of a temporally unchangeable validity of its conclusions. However, new information, that has not yet been taken into account in the knowledge base, can render old derivations invalid. Example: If P is a bird, then P can fly: Charly is a bird, but also a penguin. Thus, while in classical logic the set of derivations increases with the growing set of presupposed facts (monotonicity), the set of derivations could in fact be restricted as the amount of new information grows over time (non-monotonicity).

This non-monotonicity in reasoning and judgement must also be assumed by an expert as a realistic situation, since a complete and error-free collection of data is not possible, too time-consuming, or tedious for an upcoming problem to be solved.

For an expert system, changing input data of the knowledge base requires that the evaluation of conclusions must be recalculated. For this reason, knowledge representation in databases is provided with time stamps. In medical diagnostics, information about the temporal change of a symptom is necessary. Philosophy of science has done some pioneering work for logic of temporal reasoning. By now, it is—consciously or unconsciously—implemented by the designers of knowledge-based expert systems [15].

The philosopher H. Dreyfus distinguishes a 5-stage model from beginner to expert. This model is intended to underline this insight [16]. At level 1, the beginner adopts rules that are applied stubbornly without reference to the overall situation. The learner driver learns to switch gears with fixed speed values, the apprentice learns about the individual parts of an engine, the player learns the basic rules of a game. At level 2, the advanced beginner already makes occasional reference to situational characteristics. The apprentice learns to take into account values of certain materials based on experience, the learner driver learns to switch gears based on engine noise, etc. At level 3, competence has already been achieved, and the apprentice has, so to speak, passed the journeyman′s examination. The apprentice has learned to develop strategies to solve complex problems in the specific field of application. The driver can coordinate the individual rules for driving the vehicle in accordance with the regulations. According to Dreyfus, this means that the maximum performance of an expert system has already been achieved.

The next levels of master and expert cannot be captured algorithmically. Judgement is required that relates to the entire situation, the chess master who recognises complex position patterns in a flash and compares them with known patterns, the racer who intuitively senses the driving style, best suited to the engine and the situation, the engineer who, on the basis of his experience, hears from noise where the engine fault is located.

How do you become a good management expert? Algorithmic thinking, computer-aided problem analysis, the use of expert systems help in the preparation phase. Expert systems lack, as pointed out above, a general knowledge of the world and the background. The sense of the whole as the basis for correct decisions is not learned from a textbook or planning calculations. After basic training a manager no longer learns through abstract definitions and general textbook rules. He learns through concrete examples and cases from his own company as far as possible and is able to apply them to the situation. Concrete case studies combined with a sense of the whole sharpen the future manager′s ability to judge.

This is where we reach the practical limits of expert systems based on symbolic AI. Learning from experience means learning from data. This is where machine learning with its statistical learning theory enters. Expert systems are therefore still being used. Their fate, however, is that no one sees them as spectacular any more. They have long been part of everyday technical or medical life, for example, without still being called "artificial intelligence".

2.2 Causality Versus Statistics

> The larger the fire brigade operation, the greater the damage.
>
> A simple statistical correlation.

Machine learning (subsymbolic AI) is currently changing the nature of computer science dramatically. We rely more and more on efficient algorithms because the complexity of our civilisational infrastructure would otherwise be impossible to manage: Our brains are too slow and hopelessly overtaxed by the amount of statistical data at hand. But how reliable are AI algorithms based on statistical learning? In practical applications, learning algorithms refer to models of neural networks, which are themselves extremely complex. They are fed with huge amounts of data and trained. The number of parameters required for this explodes exponentially. No one knows exactly what is going

on in these "black boxes" in detail. It often remains a statistical trial-and-error procedure. But how should questions of responsibility be decided in the context of self-driving cars or in medicine, for example, if the methodological foundation remains obscure?

In statistical learning, dependencies and correlations should be algorithmically derived from observational data [17, chap. 11.1]. For this purpose, we can imagine a scientific experiment, in which a series of changing conditions (inputs) are followed by corresponding results (outputs). In medicine, this could be a patient who reacts to medication in a certain way. We assume that the corresponding pairs of input and output data are generated independently by the same unknown random experiment. Statistically, therefore, we say that the finite sequence of observation data $(x_1, y_1), \ldots, (x_n, y_n)$ with inputs x_i and outputs y_i $(i = 1, \ldots, n)$ is realised by random variables $(X_1, Y_1), \ldots, (X_n, Y_n)$, which is based on an unknown probability distribution $P_{X,Y}$.

Algorithms are now to derive properties of the probability distribution $P_{X,Y}$. An example would be the expected probability with which a corresponding output occurs for a given input. It can also be a classification task: a set of data is to be divided into two classes. With which probability an element of the data set (input) belongs to one or the other class (output)? In this case, we also speak of binary pattern recognition.

Example

A simple example explains the basics.

When a binary pattern is recognised, the data of a data set X is distributed over two possible classes, which are designated $+1$ and -1 respectively. This allocation is described by a function $f : X \to Y$ with $Y = +1, -1$. In the statistical learning of a binary pattern the task is, to determine from a class F of functions the assignment f for which the error deviation or the expected error is minimal. We also speak of the risk minimisation of statistical learning [18]:

$$R[f] = \int \frac{1}{2}|f(x) - y| dP_{X,Y}(x, y)$$

As the probability distribution $P_{X,Y}$ for all values is unknown, this formula and thus the sought-after pattern recognition with minimum error deviation cannot be calculated. We only have the finitely many empirically observed assignments $(x_1, y_1), \ldots, (x_n, y_n)$ available. We therefore restrict ourselves to empirical risk minimisation.

For this purpose we determine step by step for each assignment function f of class F the empirical training error when learning from a sample with size n:

$$R^n_{emp}[f] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}|f(x_i) - y_i|$$

This creates a sequence of functions of the class F with improved training error. The central question is whether pattern recognition with a minimum possible error deviation can be determined by this procedure. Mathematically formulated, the problem is thus, whether the sequence of functions in the class F, determined in this way, converges to a function with a minimum error deviation.

In fact, it can be proved that such convergence or learning success is only guaranteed for small subclasses. An example is the Vapnik–Chervonenkis (VC) dimension, with which the capacity and size of such function classes can be determined [19]. With high probability, the risk is not greater than the empirical risk (plus a term that grows with the size of the function class). ◀

The current success of machine learning seems to confirm the thesis that it is important to have data sets as large as possible, which can be processed with ever-increasing computer power. The detected regularities then depend only on the probability distribution of the statistical data.

Statistical learning attempts to construct a probabilistic model from a finite number of data of results (e.g. random experiments) and observations (Fig. 2.3).

Conversely, *statistical reasoning* attempts to derive properties of observed data from an assumed statistical model (Fig. 2.3).

Data correlations can provide clues to facts, but do not have to. Let us imagine a series of tests which result in a favourable correlation between a chemical substance administered and the fight against certain cancer tumours. Then companies are under pressure to produce a corresponding drug and to make profits. Also the patients may see this as their last chance. In fact, we only get a sustainable drug only if we find the underlying causal mechanism of tumor growth, i.e. the laws of cell biology and biochemistry.

Even Newton was hardly interested in data correlations of falling apples on apple trees, but rather in the underlying mathematical causal law of gravitation. They allowed precise explanations and forecasts of falling apples and celestial bodies, and ultimately the current satellite and rocket technology is based on it. Specifically, Newton assumed a causal analogy between the force acting on an apple on the earth and the forces acting on celestial bodies in the planetary system. With the help of Kepler′s planetary laws, he therefore proposed a causal model on the basis of a few observational data which is described mathematically by a functional relationship between causes and effects.

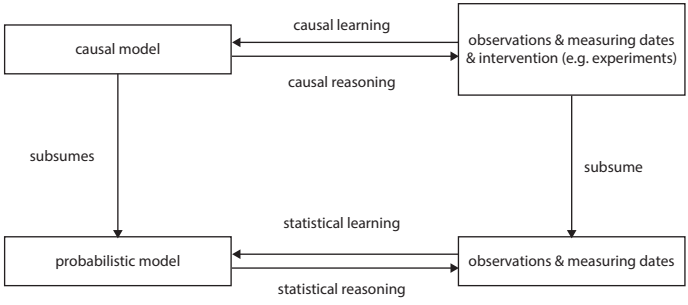


Fig. 2.3 Statistical and causal learning [21]

Statistical learning and inference from data are therefore not enough. Rather, we must recognise the causal relationships of causes and effects behind the measured data. These causal relationships depend on the laws of the respective application domain of our research methods, i.e. the laws of physics in the example of Newton, the laws of biochemistry and cell growth in the example of cancer research, etc. If it were otherwise, we could already solve the problems of this world with the methods of statistical learning and reasoning. In fact, some short-sighted contemporaries seem to believe this in the current hype of statistical machine learning.

> Statistical learning and reasoning without causal domain knowledge is blind[2]—no matter how large the amount of data (Big Data) and computing power!

In addition to the statistics of the data, there is a need for additional law and structure assumptions of the application domains, which can be tested by experiments and interventions. Causal explanatory models (e.g. the planetary model or tumor model) fulfil the laws and structural assumptions of a theory (e.g. Newton′s theory of gravity or the laws of cell biology):

In *causal reasoning*, properties of data and observations are derived from causal models, i.e. assumptions of laws of causes and effects. Causal reasoning thus makes it possible to determine the effects of interventions or data changes (e.g. through experiments) (Fig. 2.3).

Conversely, *causal learning* attempts to construct a causal model from observations, measurements and interventions (e.g. experiments) with additional assumptions of laws and structures (Fig. 2.3).

[2] Kant: “Intuitions without conceptions are blind.” [20, A48/B75]

A structural causal model consists of a system of structural assignments of causes to effects with possible disturbance variables (noise). Causes and effects are described by random variables. Their functional assignments (taking into account the noise variables) are defined by equations, e.g. effect $X_j = f(X_i, N)$ in functional dependence on the cause X_i and the disturbance variable N.

The network of causes and effects can be represented by a graph of nodes and edges. Random variables of causes and effects correspond to nodes. Causal effects correspond to directed arrows: $X_i \to X_j$ means that cause X_i triggers effect X_j.

It can be proved that a causal model includes an unambiguous probability distribution of the data (Fig. 2: "subsumed"), but not vice versa: For causal models (e.g. planetary model) one needs to assume additional laws (e.g. law of gravity). In order to recognise causal dependencies of events, the independence of the random variables representing them must be determined. Statistically, the independence of the results x and y of two random variables (viz. random experiments) X and Y can be expressed statistically by the fact that their composite probability $p(x, y)$ is factorisable, i.e. $p(x, y) = p(x)p(y)$. In this case, one also speaks of the Markov condition. On this basis, the calculus of a causal independence relation $\perp\!\!\!\perp$ can be introduced [22]:

Let $p(x)$ be the density of the probability distribution P_X of a random variable X:

- X independent of Y $(X \perp\!\!\!\perp Y) :\Leftrightarrow p(x, y) = p(x)p(y)$ for all values x, y of X, Y;
- $X_1, \ldots, X_d$ mutually independent $:\Leftrightarrow p(x_1, \ldots, x_d) = p(x_1) \cdot \cdots \cdot p(x_d)$ for all values $x_1, \ldots, x_d$ of $X_1, \ldots, X_d$;
- X independent of Y under the condition Z $(X \perp\!\!\!\perp Y|Z) :\Leftrightarrow p(x, y|z) = p(x|z)p(y|z)$ for all values x, y, z of X, Y, Z with $p(z) > 0$.

Conditional independence relations satisfy the following rules:

$$X \perp\!\!\!\perp Y|Z \Rightarrow Y \perp\!\!\!\perp X|Z \quad \text{(symmetry)}$$

$$X \perp\!\!\!\perp Y, W|Z \Rightarrow X \perp\!\!\!\perp Y|Z \quad \text{(decomposition)}$$

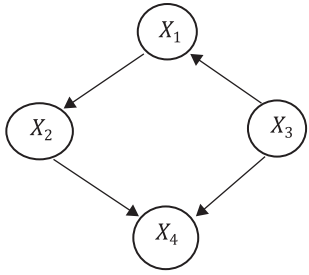
$$X \perp\!\!\!\perp Y, W|Z \Rightarrow X \perp\!\!\!\perp Y|W, Z \quad \text{(weak union)}$$

$X \perp\!\!\!\perp Y|Z$ and $X \perp\!\!\!\perp W|Y,Z \Rightarrow X \perp\!\!\!\perp Y, W|Z$ (contraction)
$X \perp\!\!\!\perp Y|W,Z$ and $X \perp\!\!\!\perp W|Y,Z \Rightarrow X \perp\!\!\!\perp Y, W|Z$ (intersection set)

Example

A simple example

Causal structural model with assignments and graphical representation



- $X_1 := f_1(X_3, N_1)$
- $X_2 := f_2(X_1, N_2)$
- $X_3 := f_3(N_3)$
- $X_4 := f_4(X_2, X_3, N_4)$,
- With N_1, N_2, N_3, N_4 independent noise variables.

The independence of random variables X_1, X_2, X_3, X_4 in the statistical distribution P_{X_1,X_2,X_3,X_4} can be represented by $X_2 \perp\!\!\!\perp X_3|X_1$ and $X_1 \perp\!\!\!\perp X_4|X_2, X_3$ resp. by Markov factorization:

$$p(x_1, x_2, x_3, x_4) = p(x_3)p(x_1|x_3)p(x_2|x_1)p(x_4|x_2, x_3) \quad \blacktriangleleft$$

The aim of causal learning is thus to discover the causal dependencies of causes and effects behind the distribution of measurement and observation data. The initial situation is a finite sample of a data collection: In the example, a joint probability (e.g. P_{X_1,X_2,X_3,X_4}) of independent and identically distributed (i.i.d.) random variables (e.g. X_1, X_2, X_3, X_4) is presupposed. By means of independence tests and experiments, causal models can be constructed from this, which are determined by independence relations resp. probabilistic factorization or causal laws. On the

basis of such causal models, the dependencies of causes and effects can be graphically represented. Further on, the accountability of causes and effects called for at the beginning is realizable. Accountability of causes and effects is necessary, for example, to clarify questions of responsibility (Fig. 2.4).

In the case of statistical machine learning, the limits of AI are evident by the fact that, in principle, there is no general algorithm that can be used to determine an underlying causal structure for any statistical data distribution. However, this principal limit also applies to the natural intelligence of a human mathematician. But, for certain classes of data distributions under precisely specified conditions (constraints), algorithms for the determination of the underlying causal structures can be found. They depend, for example, on the type of the data distributions

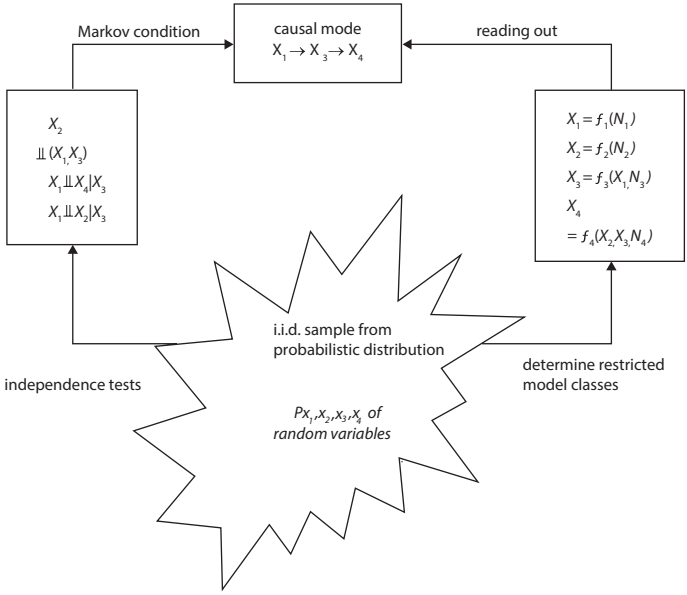


Fig. 2.4 From data evaluation to causal models [23]

and the equations. In these cases, there would be an AI that independently solves knowledge tasks like a scientist. They may not yet have the status of Newton′s discovery of the law of gravitation. But causal thinking is not inaccessible to AI in principle: this can be proven mathematically!

Causal models that are subject to statistical data distributions now influence machine learning. In an astrophysical example, B. Schölkopf shows how an assumed causal structure can be used to reduce systematic error noise in the prediction of exoplanets [24].[3] Empirical data were provided by the Kepler space telescope, which observed a small section of the Milky Way for its search of exoplanets. In the process, the brightness of almost 150,000 stars was measured. It is assumed that the orbit of a planet is so that, from the Earth′s point of view, it passes exactly in front of the star. The resulting occultations of the star then produce periodic decreases in its brightness.

Example

The signal of interest Q (e.g. periodic decrease in the light intensity of a star caused by an orbiting planet) can only be measured in a noisy version Y (Fig. 2.5). If the same source of noise also takes the measurements of other signals R (with noisy version X) independently of Q (e.g. stars that are light years apart), then these measurements can lead to "denoising", i.e. to the neglect of the measurement disturbances. In this case, the observing telescope N used is the systematic source of perturbations for the measurements X and Y of independent light curves. This telescope measures several stars at the same time. They can be assumed to be statistically independent, since they are light years apart and, according to Einstein′s theory of relativity, no effects can be transmitted faster than light.

[3] These explanations follow K. Mainzer, Quantencomputer. Von der Quantenwelt zur Künstlichen Intelligenz, Springer 2020 [25, p. 129 ff].

In Fig. 2.5, only the observed quantities are drawn in black. All causal assumptions are green. Thus the variable X (black) denotes measurements of signals R (green) that are independent of Q. Graphically, everything in Y that can be explained by X must be due to the common noise source of the telescope N and should therefore be removed. Formally, this means: let $E[Y|X]$ be the expectation probability (regression) of the observed event Y dependent of event X. Since X and Y have the same parent node N in the causal graph (Fig. 2.5), they are graphically referred to as "half-siblings". Therefore, one also speaks of "half-sibling regression".

In this way, the unobserved "true" signal Q (green) can be estimated by subtracting from the measurement Y (black) the expected probability of the disturbances caused by the common measuring instrument with X:

$$\widehat{Q} := Y - E[Y|X]$$

In general, for random variables Q, X, Y, with Q independent of X ($X \perp\!\!\!\perp Y$) and estimation $\widehat{Q} := Y - E[Y|X]$ of Q, it can be proved that the method of "denoising" can never be worse than the measurement Y itself:

$$E\left[\left(Q - E[Q] - \widehat{Q}\right)^2\right] \leq E\left[(Q - E[Q] - (Y - E[Y]))^2\right] \blacktriangleleft$$

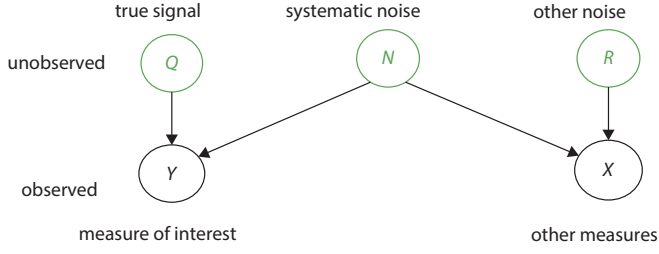


Fig. 2.5 The causal structure used in the search for exoplanets (Tracing of [21, p. 158])

Example

Another example concerns brain research [26]: in this case, we are dealing with one of the most complex neuronal networks that has emerged in evolution. Neural networks are represented by causal graphs, whose nodes represent neurons and whose directed edges represent synaptic connections of the neurons. In the mathematical model, we assume a vector z that encodes the activity of a large number of brain regions. The dynamics (i.e. the temporal development) of z is determined by a differential equation

$$\frac{d}{dt}z = F(z, u, \theta)$$

with given function F, vector u of external stimulation, and parameter θ of causal connections.

However, the brain activity z cannot be observed directly. Functional resonance imaging (fMRI) only determines the consumption of nutrients (oxygen and glucose) to compensate for the increased energy demand supplied by blood flow (haemodynamic response). The increase is determined by the blood-oxygen-level-dependent (BOLD) signal. Therefore, in the dynamic causal model z must be replaced by a state variable x, in which brain activity is taken into account with the haemodynamic response:

$$\frac{d}{dt}x = F(x, u, \theta)$$

For this purpose, the measured time series of the BOLD signal $y = \lambda(x)$ is connected with the state variable x. ◀

In fact, in the human brain we are dealing with a flood of data produced by 86 billion neurons. How the causal interactions between the neurons behind these data clouds take place in detail remains a black box for the time being. Statistical learning from measured data is not enough, even in the age of Big Data and growing computing power. More explanation of the causal interactions between the individual brain regions, i.e. causal learning, is a central challenge for brain research in order to obtain better medical diagnosis, psychological and legal sanity.

2.3 From Baysian Learning to Neural Networks

> PROP. 5. If there be two subsequent events, the probability of the 2d $\frac{b}{N}$ and the probability of both together $\frac{p}{N}$, and it being 1st discovered that the 2d event has happened, from hence I guess that the 1st event has also happened, the probability I am in the right is $\frac{P}{b}$.
>
> THOMAS BAYES, [27, p. 381].

Newton′s methodology, which is described in his textbook "Principia Mathematica Philosophiae Naturalis" as regulae philosophandi (rules of philosophising), is simple and has influenced centuries of physics. It is repeated almost verbatim by Einstein and others in the twentieth century: The natural scientist begins with observations and measurements and recognises correlations in these finite data. These are generalised in the assumption of a law or a theory of several laws. In mathematical terms, they often take the form of equations such as the equations of motion and force in Newtonian mechanics. In philosophy of science such assumptions of laws are called hypotheses. An empirical theory is then a system of hypotheses. Newton called this path from data to hypotheses or theories "induction" [28].

From such theories and hypotheses, predictions for past and present events or explanations for future events can be derived logico-mathematically for suitable initial and secondary conditions. An example is Kepler′s planetary model, which is derived from Newton′s laws of mechanics. On the basis of this model, predictions about future planetary positions can be derived. We assume that a statement A (here, Kepler′s planetary laws with a known initial position of a planet) implies a statement B (here a prediction of a future planetary position). The truth of this implication is proved by calculating the equation of motion of the planet after inserting its initial position. Now the statement A, i.e. Kepler′s planetary model and the initial position of the planet, is assumed to be true. Then statement B, i.e. the prediction of the future position of the planet, is also logically compelling. If the conclusion "If A, then B " is true and the statement A is true, then, with logical necessity, the statement B is true.

But now let us assume that the conclusion "If A, then B" is true and that the statement B (e.g. an observation of the planetary location) is true. What do we know about the assumed law or model? In general, statement A is not true as a logical necessity. After repeated observations, according to which the logical conclusions B from A are true, one could at best assume that our assumed model or hypothesis A is plausible. Obviously, this is the situation in the empirical sciences such as physics, chemistry and biology, but also in social and economic sciences. Only mathematics is concerned with logico-mathematical deductions from assumed axiom or hypothesis systems.

The empirical sciences therefore focus on the questions: How do we find suitable models for explanations and forecasts in data sets? How plausible are such models? How does their credibility change with new observations and new background knowledge? What can we learn from new experiences? With the enormous amount of data, that molecular biology and economics, for example, have to deal with, it is clear that these questions can only be answered within the mathematical framework of statistics and probability theory.

According to the English mathematician and theologian Thomas Bayes (1702–1762), learning can be explained by the conditional probabilities of events [29]. In this context, probability is not defined as probability (objective probability), but as a degree of belief (subjective probability): An event A is assumed before the occurrence of event B with the a priori probability $P(A)$, but after the occurrence of B with the a posteriori (conditional) probability $P(A|B)$.

With the help of Bayes′ theorem, conditional probabilities can be calculated: The conditional probability $P(A|B)$ of event A after the occurrence of event B is given by the quotient of the probability $P(A \cap B)$ (i.e. the probability that events A and B occur together) and the probability $P(B)$ of event B.

i.e.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$\frac{\frac{P(A\cap B)}{P(A)} \cdot P(A)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Therefore, the theorem of Bayes says $P(A|B) = \frac{P(B|A)\cdot P(A)}{P(B)}$ i. e. the probability of A after the occurrence of B is calculated from the conditional probability of B provided that A and the a priori probabilities $P(A)$ and $P(B)$ are given.

How can Bayes′ methods be applied to inductive reasoning in the natural sciences? How can a model M be derived from a data set D? From Bayes′ theorem it follows for the probability of a model M given a data set D:

$$P(M|D) = \frac{P(D|M) \cdot P(M)}{P(D)} = P(M)\frac{P(D|M)}{P(D)}$$

The a priori probability $P(M)$ estimates the probability that model M is correct before (Latin: a priori) data are available. The a posteriori probability $P(M|D)$ takes into account the estimate of the probability that model M is correct after (Latin: a posteriori) the data of data set D have been observed. The probability $P(D|M)$ that the observations of the data set D occur under the assumption of model M is referred to as the likelihood of data under the assumption of a model. The probability $P(D)$ stands for the data evidence of the data set D. For the calculation of the probability of a model M under the assumption of a data set, the calculation of corresponding logarithms is often simpler:

$$\log P(M|D) = \log P(D|M) + \log P(M) - \log P(D)$$

In the Bayesian assessment of models, the assumption of "a priori" probabilities is occasionally criticised as subjective or arbitrary. In reality, however, the effects of the a priori probabilities decrease as the number of data increases. In the logarithmic calculation of the likelihood, $\log P(D|M)$ typically grows linearly with the number of data from D, while the a priori expression $\log P(M)$ remains constant. Moreover, the Bayesian approach requires a clear distinction between a priori and a posteriori assumptions. The a priori probability of a model depends on the assumed probability distribution. Depending on the application,

e.g. Gaussian or Dirichlet distributions are suitable for this. In any case, no general objective principle is known to determine a priori assumptions in all situations [30].

Models $M(w)$ depend on parameters w. In a physical model of phase transitions that describes the transition from e.g. a liquid state to a gas state at critical temperature values, temperature is an example of a parameter. In contrast to such simple physical models, molecular biological models depend on a huge number of parameters. To assess how good a model $M(w)$ is for a set D of data such as a molecular biological sequence, error and falsity functions $f(w, D) \geq 0$ are applied to calculate the degree to which the model fits the set of data as a function of the model parameters. It can be proved that the minimisation of the error function is equivalent to the maximum data plausibility (likelihood [31]).

Two models M_1 and M_2 can be compared by comparing their probabilities $P(M_1|D)$ and for a set of data D (e.g. amino acid sequence). One goal could be to determine the best model M of a model class by finding the set of parameters w with maximum likelihood $P(M|D)$. The Bayesian approach has the methodological advantage that it favours less complex models. This corresponds to the motto of Occam′s razor, according to which the explanation of an observation that makes fewer theoretical assumptions is to be preferred [32]. It turns out that the data plausibility (likelihood) for given data sets becomes smaller on average when $P(D|M)$ concerns a growing data space. Complex models therefore tend towards a smaller data plausibility (likelihood) of the observed data.

According to Bayes′ theorem, in order to determine the probability $P(M|D)$ of a model given the data set D, it is first necessary to assess the data plausibility (likelihood) $P(D|M)$ of a model and its a priori probability $P(M)$. For a bit sequence of bits 0 and 1, the simple model of a fair coin could be chosen whose two sides show the digits 0 or 1. This model has only a single parameter p. The data consists of sequences over the alphabet $A = \{0,1\}$, generated by random coin tosses.

The data D of a DNA sequence is formed over the alphabet $A = \{A, C, G, T\}$. A simple random model would be a four-sided dice whose sides carry the symbols A, C, G, T [33].

Bayesian philosophy of science explains statistically how our models and hypotheses about the world change through new experiences and how we can learn from experience. It can therefore be understood as a framework theory for learning algorithms that automate learning procedures in machine learning. In machine learning, neural networks modelled on the human brain play a dominant role. The breakthrough of AI research in practice is largely related to the ability of neural networks to apply large amounts of data (Big Data), e.g. in pattern recognition with effective learning algorithms. Practical applications require thousands of neurons and synapses in multilayer neural networks (deep learning) that are statistically trained with finitely many data sets of inputs and outputs.

From a Bayesian point of view, neural networks can be understood as graphical models $M(w)$ with certain parameters w. These models are reminiscent of the human brain. In classical feed-forward neural networks, neurons are arranged in layers as nodes of a graph. In Fig. 2.6, each neuron of a layer is connected to all neurons of the following layer by directed edges (synapses), but the neurons of a layer are not connected to each other. Exceptions are the input layer, whose neurons have no incoming connections, and the output layer, whose neurons have no outgoing connections. The layers between the input and output layers are called "hidden". Each connection/edge is weighted in the graphical model of a neural network with a number that corresponds to the intensity of the synaptic connection. These weights are the parameters w of the graphical model $M(w)$ of a neural network. Each neuron is characterised by an activation function that defines the input–output relation for this neuron. By analogy with the brain, a neuron is said to "fire" or be excited when the sum of the weighted inputs of its neighbouring cells exceeds a certain threshold.

Mathematically, these networks can be defined as functions $\nu : I^n \to O^m$ that map an n-dimensional input space $I^n (n > 0)$ to an m-dimensional output space $O^m (m > 0)$. For example,

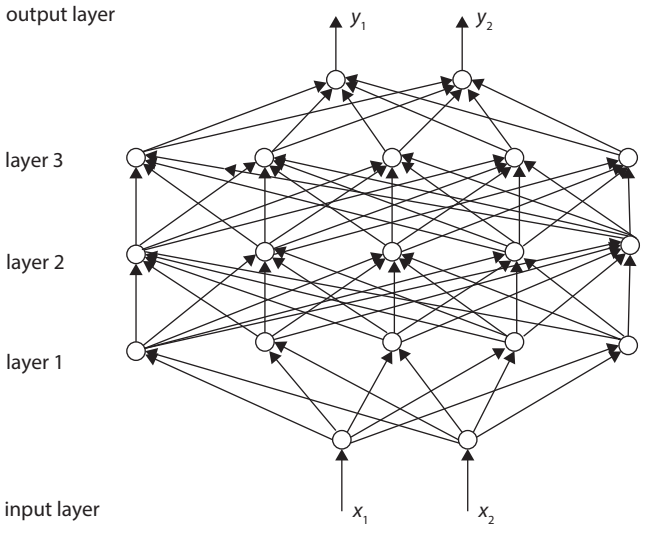


Fig. 2.6 Feedforward network with 3 hidden layers and an input and output layer

such a network can compute an approximation of a function $f : \mathbb{R}^n \to \mathbb{R}$ with $I = O = \mathbb{R}$. For example, a network that classifies 8-bit images of size $h \times v$ (with h horizontal and v vertical) into two classes can be defined by a function $\nu : I^{h \cdot v} \to O$ with input range $I = \{0, \ldots, 255\}$ for the $2^8 = 256$ possible 8-bit images and output range $O = \{0, 1\}$ for the two classes denoted by 0 and 1. The mapping begins with an input from I^n, which is first entered into the input layer in the network and then processed further via the hidden intermediate layers to the output layer. Layer by layer, linear combinations of the values of nodes (neurons) and weights (synaptic connections) from the preceding layers are calculated. Activation functions for the subsequent neurons are applied to these results (Fig. 2.7). Graphically, the activation functions trigger the state of "firing" of a neuron.

Networks are distinguished by different activation functions. The threshold function of a McCulloch-Pitts neuron only has the function value 1 for inputs $v \geq 0$, otherwise 0 (Fig. 2.8a).

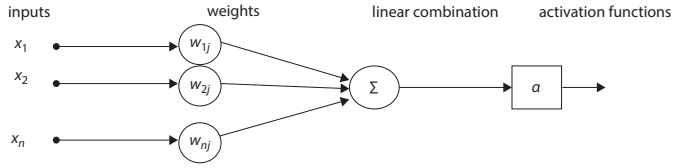


Fig. 2.7 Activation functions of neurons in neural networks

A piecewise linear function linearly maps a bounded interval and the outer intervals are constant (Fig. 2.8b). Sigmoid functions have a variable increment expressed in the curvature of the graph (Fig. 2.8c). A rectifier function (ReLU = rectifier linear unit) takes the positive values of its arguments, otherwise 0 (Fig. 2.8d):

The crucial point here is that neural networks can learn from examples. From a Bayesian point of view, this is nothing more than estimating the suitability of a model $M(w)$ for explaining (fitting) a data set D and estimating the model parameters w. Depending on the application, a data set can be understood as an input–output sample $D = (D_1, \ldots, D_K)$ of pairs $D_i = (d_i, z_i)$, where d_i stands for the respective data and z_i for the respective target state.

In a classification task, for example, the task could be to divide the data of a data set into certain classes as target states. If a finite number of target states are specified in the training data, we speak of supervised learning. Otherwise, it is non-supervised learning. An error function is used to compare the output data with the target states in order to optimise the model parameters, i.e. the weights of the neural network. From a Bayesian point of view, this is the classic induction problem of how to obtain the best possible model from data.

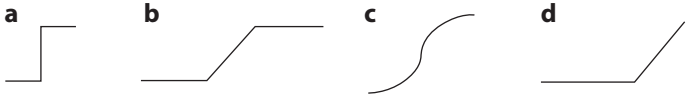


Fig. 2.8 Examples of activation functions

A remarkable property of neural networks is that they can approximate almost any function $f(x) = y$ with arbitrary accuracy [34]. Here, the function arguments x are the inputs and the function value y is the output of a suitable neural network. The algorithm for approximating the function f is realised step by step through the layers of a suitable network architecture. Bayesian science theory can be used to select a suitable objective function and a transfer function for the output. Let data again be given as independent input–output pairs $D_i = (d_i, z_i)$. These data are noisy in the sense that different target states z_i can be observed for a given data point d_i. The operations of the neural network itself are assumed to be deterministic. For the probability of the data pairs $D_i = (d_i, z_i)$ under the condition of the synaptic weights w (as parameter values of the neuronal model $M(w)$) the Bayesian calculus then yields.

$P((d_i, z_i)|w) = P(d_i|w)P(z_i|d_i, w) = P(d_i)P(z_i|d_i, w)$, where for the second equality the independence of the inputs d from the parameters w is assumed. The logarithmic formula of the Bayesian calculus can then be used to calculate the probability of the model parameters w (i.e. the weights of a neural network) assuming the data pairs D:

$$\begin{aligned}\log P(w|D) &= \log P(D|w) + \log P(w) - \log P(D)\\ &= \sum_{i=1}^{K} \log P((d_i, z_i)|w) + \log P(w) - \log P(D)\\ &= \sum_{i=1}^{K} (\log P(z_i|d_i, w) + \log P(d_i)) + \log P(w) - \log P(D)\\ &= \sum_{i=1}^{K} \log P(z_i|d_i, w) + \sum_{i=1}^{K} \log P(d_i) + \log P(w) - \log P(D),\end{aligned}$$

where in the 3rd line $P((d_i, z_i)|w) = P(d_i)P(z_i|d_i, w)$ was taken into account.

In the calculation, the data evidence $P(D)$ and $P(d_i)$ can be neglected, since these values do not depend on the parameters w (i.e. the synaptic weights of the neural network). The focus is on determining the a priori probability $P(w)$ and the data plausibility (likelihood) $P(z_i|d_i, w)$. In the case of data plausibility (likelihood), the guiding idea is that a network with given weights w generates an estimated output $y(d_i)$ for a given input d_i. The

model is fully defined if it can be determined how the observed data $z_i = z(d_i)$ can statistically deviate from the output $y_i = y(d_i)$.

In terms of the history of science, it is worth noting that the development of neuronal networks was associated with a sharp discussion about the limits of this technology which finally turned out to be provisional. The learning algorithm of the perceptron model (1950) starts with a random set of weights and modifies these weights, according to an error function, in order to minimise the difference between the current output of a neuron and the desired output of a trained data pattern (e.g. letter sequences, pixel image). This learning algorithm can only be trained to recognise such patterns (supervised) that are "linearly separable". In this case, the patterns must be separable by a straight line.

Figure 2.9a shows two patterns that consist of either small squares or small circles as elements. Both patterns are separable by a straight line and thus recognisable by a perceptron. Figure 2.9b shows two patterns that are not separable by a straight line.

M. Minsky, leading AI researcher of his time and representative of the "old" paradigm of symbolic AI, and S. Papert, proved mathematically in 1969 that Perceptron would fail if the patterns were only represented by curves ("non-linear") (Fig. 2.9b [35, 2.36]). With this Minsky and Papert believed to have mathematically refuted the new paradigm of neuronal networks, or at least to have put it in very narrow confines.

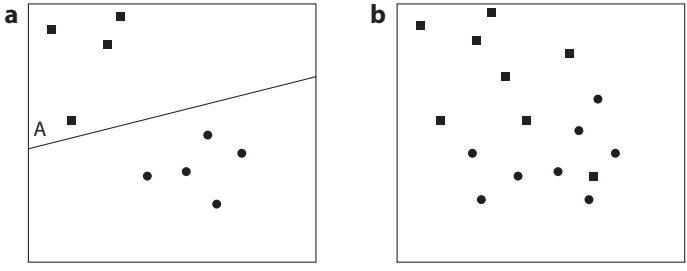


Fig. 2.9 Linear (a) and non-linear (b) separable patterns

For this reason, the proof of Minsky and Papert was initially regarded by the AI-community as the fundamental limit of neural networks for AI research. The solution to the problem was inspired by the architecture of natural brains [36].

Why should information processing only run in one direction through the superimposed layers of networked neurons?

In 1986, D. E. Rumelhart, G. E. Hinton and R. J. Williams proved that backpropagation between input, intermediate and output layers also permit non-linear classifications with suitable activation and learning algorithms [37]. Finally, K. Hornik, M. Stinchcome and H. White proved in 1989 that, under suitable conditions, feedforward architectures can also be used [38].

Neural networks with these extensions have led to major breakthroughs in machine learning and artificial intelligence. A few years ago, Google developed the AlphaGo software, whose neural network learned from playing experience in the Asian board game Go and eventually beat human champions. Following this success, in 2018 the same company developed software whose neural network succeeded in modelling proteins in the largest numbers to date and in the shortest time [39].

This software, called AlphaFold, is based on a multi-layer neural network in the sense of Deep Learning, which can predict suitable shapes and folds of proteins based on input sequences of amino acids. This is done by estimating distances and angles of bonds between amino acids, whose distribution is calculated by learning algorithms. These probabilities are summarised in a score that can be used to estimate how accurate a proposed protein structure is. The training of the neural network draws on an extensive database. Figure 2.10 shows the architecture of AlphaFold, which recognises the appropriate protein structure from a protein sequence of amino acid codes. The learning algorithm that recognises the matching protein structure from the probability distributions of the distances and angles can be based on a scoring or a gradient method. Gradient methods are known from materials research when certain structures (such as basalt columns) are created by cooling (gradient descent) a hot material (e.g. lava).

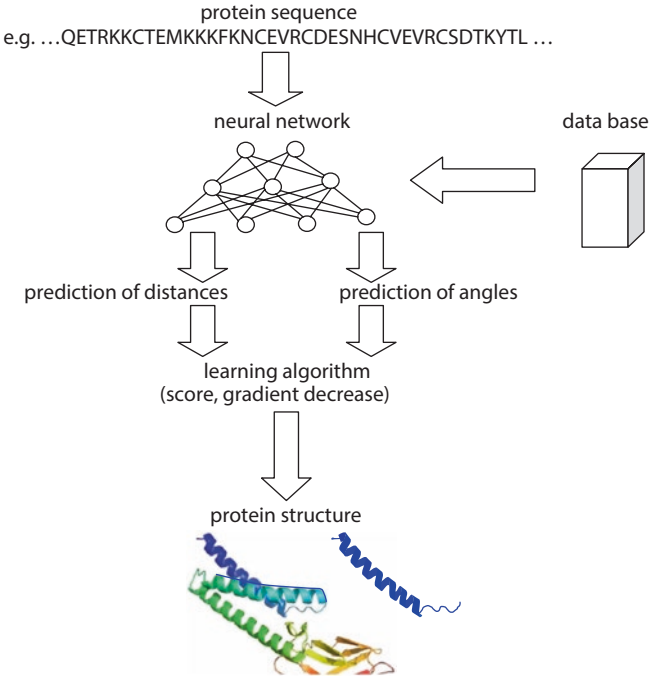


Fig. 2.10 Architecture of AlphaFold [40]

So far, only about half of all possible protein structures of the human cell have been deciphered. Of central interest are changes due to mutations. Malformations of the structure lead to malfunctions as causes of diseases. AlphaFold and similar machine learning neural networks will be indispensable for disease control and health care, since the diversity of life is based on the complexity of the world of proteins. Their codes can only be captured by machine learning with the computing power of supercomputers. Basically, the search space for possible protein forms is exponential and never complete, as evolution is not finished.

The complex structures and functions of genes, proteins and cells that we observe today are the result of an evolution that took place over many millions of years and is constantly

evolving. Darwin had the ingenious idea that the observed diversity of species can be traced back to common ancestors whose developmental branchings can be represented by the branchings of a tree. While the tree with its branches and ramifications is a model, its "leaves" at the ends of the branches correspond to the observed species. The big question is how this phylogenetic development can be derived from molecular biological evolution. In the language of bioinformatics, the aim is to infer the phylogenetic tree structures from DNA and protein sequences. Machine learning methods are used to process the large amounts of data with powerful algorithms and computers.

Mathematically, a tree T is understood to be a connected acyclic graph (Fig. 2.11). In a tree, two nodes are always connected with exactly one edge. The number of nodes is always exactly 1 greater than the number of edges. A tree is called binary if each node has either one or three neighbouring nodes. The distinguished root node of a tree is called r. A distinction is made between rooted and non-rooted trees without a distinguished root node. In phylogentic trees, the root node represents the ancestral sequence from which all other sequences of the tree are derived or can be derived. Characteristic of trees are their topology and the length of their branches. The topology shows the branching pattern and divergence of evolution. The length of the branches indicates the time interval between the events represented by the respective sequences.

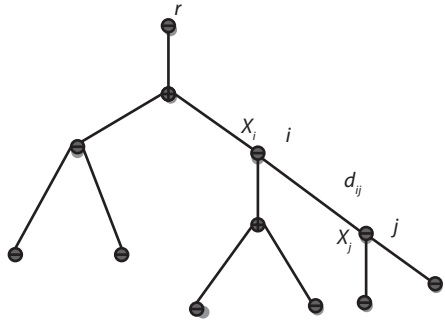


Fig. 2.11 Graphical Model of a binary phylogenetic tree [31]

As a simplified probabilistic model of evolution, a cube model can again be assumed. According to this, evolution starts with the sequence of an ancestor and develops step by step by replacing letters randomly and independently of position. In Fig. 2.11, r is the tree root. The temporal distance between tree nodes i and j is denoted by d_{ij}. X_i is the letter at the hidden tree node i.The observed letters are at the lower nodes called leaves. The probability of letters X_i being replaced by letters X_j when moving from node i to j is denoted by $p_{X_jX_i}(d_{ji})$.

A simple probabilistic model for evolution makes the following assumptions:

- At each position, there are only substitutions of letters (i.e. no insertions and deletions). All observed sequences have the same length.
- Substitutions at each position are independent of each other.
- Substitution probabilities depend only on the current (present) state and not on past evolutionary history (Markov condition).
- The Markov process is the same for all positions.

The disadvantage of these simplified assumptions is that in biological evolutions not a single assumption is fulfilled by real DNA. The length of DNA sequences can change due to insertions and deletions. Furthermore, evolution is not independent of different positions. The rates of change during evolution are not uniform in time and with respect to positions. In addition, DNA can be recombined. Correspondingly simplified probabilistic models can therefore only serve as approximations.

Example

Given a set of sequences and a probabilistic evolutionary model, an attempt can be made to determine the most probable tree topology and the most probable length of branches. Let K sequences $O_1, \ldots, O_K$ of the same length N be given over an alphabet $\mathcal{A}$. In the corresponding tree model T, r denotes the common root and d_{ij} the distance between neighbouring nodes i and j. The aim of Bayesian scientific theory

is to determine the probability $P(T|O_1,\ldots,O_K)$ of the model T given knowledge of the sequences $O_1,\ldots,O_K$. According to Bayes′ theorem, an essential step to this end is the calculation of the plausibility (liklihood) $P(O_1,\ldots,O_K|T)$ of these sequences for the probabilistic evolution model T. Because of the independence assumption of the evolution model, this probability can be factorised for the individual letters of the sequences:

$P(O_1,\ldots,O_K|T) = \prod_{k=1}^{N} P(O_1^k,\ldots,O_K^k|T)$, where O_j^k denotes the k-th letter observed in the j-th sequence. It is therefore sufficient to examine the probabilities $P(O_1^k,\ldots,O_K^k|T)$ relating to the k-th letters O_j^k in the sequences at the K leaves of the tree. At each tree node i, a hidden random variable χ_i can be assumed to indicate the letter associated with node i. Therefore, a phylogenetic tree can also be understood as a causal model with a tree structure, in which the conditional probability of node j at a given parent node i depends on the temporal distance d_{ji}:

$$P\big(\chi_j = \mathrm{Y}|\chi_i = \mathrm{X}\big) = p_{\mathrm{YX}}(d_{ji})$$

All known algorithms for calculating causal models can therefore be applied to phylogenetic tree models [42]. ◀

The search for optimal tree structures is a computationally intensive challenge. The search space of all possible trees is exponentially large, so that an exhaustive search is impossible. So there are only heuristic procedures. This often involves selecting a new species in each evolutionary step, taking into account all its possible positions in the current tree. Probabilistic evolutionary models, however, must not be too simple. Markov models prove to be unrealistic for long-term evolutionary processes, since they mathematically converge to an equilibrium distribution during this period. However, a state of equilibrium in nature contradicts all previous experience. Equilibria occur only temporarily and locally in subsystems. Therefore, it is obvious to combine different local evolutionary models. This results in highly non-linear dynamic systems, which may correspond better to biological complexity, but come up against limits of calculability.

In machine learning, research focuses on modelling the structure and functions of a virus. One function is that parts of a virus can be attacked by antibodies to prevent viral entry into a cell and the spread of the virus in the organism. Another function involves protein fragments of a virus that map onto the surface of a human cell, marking the cell as infected so that it can be recognised and eliminated by antibodies. Machine learning models were trained to derive predictions about the intensity level of these properties for each viral fragment. This makes it easier and faster to estimate which parts of a virus have a higher degree of immunogenicity, i.e. the ability to develop an immune response. These parts can then be incorporated into a vaccine.

The great strength of machine learning is its ability to recognise patterns and correlations in large masses of data. With this ability, machine learning is far superior to human abilities. In immunology, we are talking about nearly a million protein fragments presented on a cell surface and visible to T cells. No human researcher would be able to systematically complete this task in a reasonable amount of time for a specific fragment of the Corona virus. In contrast, for a suitable model, machine learning algorithms could be used to calculate the data plausibility of protein fragments to predict the best candidate. Using this method, regions in the protein envelope of SARS-CoV-2 were found to form strong antibody targets.

On this basis, vaccines can be "designed" in the computer. In this process, the viral protein fragment that is recognised as favourable is stored together with other virus-like particles so that the vaccine is recognised by the immune system like a real virus and antibodies are then developed. What sounds so simple is extremely complex in biology. Proteins consist of tens of thousands of molecules. The possible foldings of proteins that determine their functions are exponentially diverse and cannot be fully computed. It is possible to exclude a large number of possibilities from the outset in the computer model, which therefore no longer need to be tested in the laboratory. But candidates that have been identified as favourable still have to be tested and

discarded in the laboratory in order to then select other candidates in the model that have to be tested again in the laboratory.

In a kind of methodological spiral, machine learning and laboratory testing thus work together to approximate a favourable result. The philosopher of science Karl R. Popper once opined, "We err up!" [41] This word vividly describes the research strategy pursued here. In the end, however, there is no mathematical proof that the research spiral converges to an optimal result or even to the truth in every case. In practice, it could well happen that no vaccine is found for a given initial situation.

The dramatic collapses that viral pandemics trigger in economies and societies worldwide lead to the question of whether humanity can prepare for new pandemic waves in the long term. This is because viral evolution continues at a greater speed than the evolution of plants and animals. This process is being accelerated considerably in the age of globalisation, as the epidemological risk of infection among large masses of people is growing worldwide, making the starting position of viral evolution with highly interconnected host organisms ever more favourable. In short, the danger of pandemics is increasing and, because of the increase in viral fitness, so is their danger. Ultimately, this acceleration could only be countered with machine learning, large databases and supercomputers.

Now, the exponentially growing possibilities of viral evolutionary trees cannot be fully mapped with supercomputers, no matter how large they are. It is conceivable, however, that genetically generated countermeasures can be predicted for classes of such viral evolutionary trees under certain restrictions (constraints). In this way, a kind of toolbox of learning algorithms could be created in order to be able to react quickly if the worst comes to the worst. The whole company would itself be a learning system that is gradually expanded through new experiences. So we are meeting viral evolution with an evolution of artificial intelligence.

2.4 Data Set and Data Quality

社会信用体系.

Chinese for Social Credit System.

Machine or statistical learning is at the heart of modern AI. It works with data, or better: with a lot of data, or Big Data. A first impressive example was GPT-3, an autoregressive language model [44]. By use of 45 terabytes of retrieved speech data (e.g., from Wikipedia) and deep learning it can conduct dialogs with a user. It is impressive for two reasons: in general, it can form grammatically correct sentences; but it can also assign the appropriate context to homonyms. For example, it does not confuse banks, when it is used on the one hand for a riverbank or a saving bank. The program owes this ability primarily to the immense amount of data it could access.

Even as a pilot application, GPT-3 can deal with local and temporal language variants and, e.g., answer a question in the language of the Shakespearean era accordingly. And the newer versions, ChatGPT and GPT-4 (see below Chapt. 5.3), have already proven that they can answer correctly in German, a language with a grammar usually considered as comparatively complicated.

However, the matter becomes problematic, when one considers a language with few speakers or few recorded language material, or when a dead language. Especially in the last case it can not be assumed that sufficient additional material can be collected over time to make a larger stock accessible to machine learning.

Here we are confronted with questions which arise in the context of a small database which does not seem to be suitable for machine learning. To get a sense of what a marginal database is likely to be, one may consider a cultural achievement of mankind that is significant in the history of science: the ability to predict a solar eclipse. Machine learning is not capable of doing this according to an expert of the area: far too few data.

Example

And we want to give another impressive example from the history of mathematics. It suggests that human intuition is still far ahead of machine learning. The most important open problem of mathematics is the Riemann conjecture, which states that all non-trivial zeros of the ζ-function.

$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ for $s \in \mathbb{C}$ with $Re(s) > 1$ have the real part $Re(s) = \frac{1}{2}$ [45]. Here, the question is not whether one can find a proof of this conjecture by means of machine learning.[4] But we ask how many zeros an AI program would have to compute in order to make this assumption independently – if it can make generalized assumptions on its own, in the first place. Riemann himself had calculated only 3 zeros and one can consider it as impossible that any AI can infer anything from three data only. ◀

Of course, Riemann had a much more advanced theory in the background, which allowed him to make the assumption even without further empirical data [46]. Even more impressive is the example of the physics of the twentieth century with its ingenious breakthroughs from Einstein′s general theory of relativity (1915) to the first axiomatic versions of quantum mechanics that emerged in the 1920s. Some of these were not even based on small data, but on the purely theoretical calculation of the consequences resulting from specific assumptions—for the theory of special relativity, for example, the finiteness of the speed of light. The empirical data were usually collected only afterwards in order to verify the theoretical predictions. And the richness of these basic equations of physical models has not been exhausted until today. For scientific progress background theories—whether they are to be confirmed or reformulated—are of fundamental importance; however, such theories are not available to a statistics-based AI.

[4] Evidence search in the mathematical sense does not seem to be a suitable subject for statistical learning. However, if it were possible to find a proof of Riemann′s conjecture with the help of artificial intelligence, even the last mathematical sceptic would certainly be convinced by this technology.

For all the successes of machine learning, we are not aware of any approach that would claim to be able to duplicate successes, which are based on big data, on the base of small data. Thus in the presence of few data we stay with difficulties to apply machine learning, and this seems to be an intrinsic limit for this form of artificial intelligence.

Machine learning starts with learning data. These data are later used to obtain results for input data which does not match directly with the learning data. Sometimes this involves statistical biases in such results, especially with a discriminating effect. Currently, it is an important task of statistical learning to exclude such discriminatory consequences. But such consequences are not necessarily due to the way the AI software works: if the learning data itself already contained discriminatory elements, one can, of course, not blame machine learning if this is reflected in the application.

Thus, it is a special task of machine learning to ensure quality of the learning data; and this quality includes that the data includes that this data should be free of discriminating elements.

Example

At this point, however, it boomerangs that modern AI translation software is based on the frequency of translations found in general use. Google Translator, for instance, had chosen the gender of a profession according to the dominant use in other texts. As Turkish does not have a gender indication, one obtained for the two (gender-neural) sentences: "O bir hemşire. O bir doktor." as English translations "She is a nurse. He is a doctor."[5] One cannot blame Google insofar as the frequencies of the correlation "she – nurse" and "he – doctor" in not intentionally filtered text bases should statistically predominant. When, today, Google Translator is warning the user about the risk of gender bias, this is due to human intervention, not by self insight of the underlying AI software.[6]

[5] The example is discussed in [47, p.17]. In [45] one can find a number of other examples and also extensive references to the corresponding sources.

[6] https://ai.googleblog.com/2020/04/a-scalable-approach-T-reducing-gender.html. Retrieved March 2021.

Thus, especially with respect to current debates on linguistic discrimination, we face a need for corrections against biases. Data management is required here, which (at least until today) is not—and perhaps cannot—be done by AI itself, but which requires manual intervention by humans. ◀

The classic example of poor data quality is the completely miscarried prediction of the outcome of the 1936 presidential election in the USA by the *The Literary Digest* [46]. The magazine had predicted a victory of Alf Landon over incumbent president Franklin D. Franklin, when in fact the latter won with 60% of the electoral votes and, because of the U.S. electoral system, with a crushing majority of 523 to 8 electors. The magazine had relied on a polling of 10 million eligible voters, which, however, in no way corresponded to a representative sample.[7] This fiasco serves nowadays in the statistical literature as a pointer to the importance of proper sample selection, and it has also historically contributed to the development of better methods for modern opinion research.

The example shows that data quality is not a specific problem of machine learning. But it is clear that if you start from bad data, you can hardly expect good results. Thus, we face the question how data quality can be determined or measured in concrete applications. And further, the question arises whether the criteria for data quality–or their absence–could be determined independently by AI. That seems to be rather doubtful.

2.5 The Return of the Frame Problem

…if each context can be recognized only in terms of features selected as relevant and interpreted in a broader context, the AI worker is faced with a regress of contexts.

Hubert Dreyfus [49, p. 289]

[7] See [46], where, however, the beautiful myth is rejected that the sample is limited to (supposedly above-average wealthy and therefore inclined to conservatism) phone owners.

The frame problem is a, especially in philosophy, widely discussed problem from the early days of the old AI. McCarthy and Hayes [48] consider in which form relevant information (in the concrete situation, for a robot) has to be represented in logical form, without having to store any fact of the environment (to frame the situation). Against the boundless number facts that could be represented, but also in view of the comparatively restrictive syntax of logical programming languages programming languages, such as Prolog, the problem was put in a broader philosophical context and reduced to the following question: which facts might be subject to change, and have to be, therefore, explicitly treated.[8] In the end, this problem already points to the hurdle at which the old AI ultimately failed: the complexity of a formal representation of all relevant facts, which proved to be too costly. A clarification of the question of what is to be considered relevant, was not even any longer attempted in view of the difficulties.

For the new AI, the problem of the relevant information returns in a slightly different form. A well-known example is the recognition of a camel in a photo; since camels are usually photographed in the desert, it may happen that machine learning takes as a relevant factor for the identification of a camel, not the characteristic outline of the animal, but simply that there is a desert landscape in the background. The problem is more complicated than in rule-based AI, since in machine learning, per se, we have no access to what is learned as a feature. It is only when misclassification occurs while using the programme that it becomes apparent what went wrong.

One of the central distinctions that philosophically lies behind this problem was already worked out by Aristotle, when he distinguished the essential from the accidental features of a substance (see e.g. [52, E 2]). In rough analogy the fact that a camel is usually seen in front of a desert background is a purely

[8] This discussion culminated in the so-called *Yale shooting problem* [49], which illustrates well the philosophical level of of the discussions in the old AI and which attributes a place to the American university in this discussion.

accidental feature, but its characteristic outline is an essential one. Despite a 2000-year tradition of philosophical discussion, this distinction has become obsolete with the advent of modern logic. In modern logic, any feature is given, quite undifferentiated, by predicates. The distinction of essential and accidental features has been receded into the background.[9] In the context of AI it is important to recognize that this distinction is conceptually not based on a pure statistical correlation, but, on the contrary, it is part of an intellectual definition of concepts. From this perspective it becomes understandable why machine learning, as far as it uses only statistical data for the time being is not able to independently detect the distinction.

References

1. Turing, A. (1950). Computing machinery and intelligence., in: Mind 59, 433–460.
2. Puppe, L.F. (1993), Systematic Introduction to Expert Systems, Berlin.
3. Clancey, W. (1983), The epistemology of a rule-based expert system—a framework for explanation, in: AI-Journal 20, 215-293.
4. Nilson, N. (1982), Principles of Artificial Intelligence, Berlin.
5. Minsky, M. (1975), A framework for representing knowledge, in: P.Winston (ed.), The Psychology of Computer Vision, New York.
6. Sussmann, G.; Steele; G. (1980), Constraints—a language for expressing almosthierarchical descriptions, in: AI-Journal 14, 1-39.
7. Buchanan, B.G.; Sutherland, G.L.; Feigenbaum, E.A. (1969), Heuristic DENDRAL: A program for generating processes in organic chemistry, in: B. Meltzer/ D. Michie (eds.), Machine Intelligence 4, Edinburgh.
8. Buchanan, B.G.; Feigenbaum, E.A. (1978), DENDRAL and META-DENDRAL: Their applications dimensions, in: Artificial Intelligence 11, 5-24.
9. A. Newell, J. C. Shaw, and H. A. Simon. Elements of a theory of human problem Solving, in: Psychological Review, 65(3):151–166, 1958.
10. Shortliffe, E.H. (1976), Computer-Based Medical Consultations: MYCIN, New York.

[9] Quine expressed his personal opinion about this as follows [53, p. 204]: “For attributes, as a realm of entities distinct from classes, I hold no brief.”.

11. Randall, D.; Buchanan, B.G.; Shortliffe, E.H. (1977), Producing rules as a representation for a knowledge-based consultation program, in: Artificial Intelligence 8.
12. Carnap, R. (1959), Induktive Logik und Wahrscheinlichkeit, bearbeitet von W. Stegmüller, Wien.
13. Lindley, D.V. (1965), Introduction to Probability and Statistics from a Bayesian Viewpoint I-II, Cambridge.
14. Zadeh, L.A. (1975), Fuzzy Sets and their Application to Cognitive and Decision Processes, New York.
15. Centrone, S.; Mainzer, K. (2023), Temporal Logic. From Philosophy and Proof Theory to Artificial Intelligence and Quantum Computing, Word Scientific Singapore; de Kleer, J. (1986), An assumption based TMS, in: AI-Journal 28, 127-162.
16. Dreyfus, H.L.; Dreyfus. S.E. (1986), Mind over Machine, New York.
17. Mainzer, K. (2019), Artificial Intelligence. When do machines take over? Springer, Berlin.
18. Peters, J.; Janzing, D; Schölkopf, B. (2017), Elements of Causal Inference. Foundations and Learning Algorithms Cambridge (Mass.), 4f.
19. D. Corfield, B. Schölkopf, und V. Vapnik (2009), Falsificationism and statistical learning theory: Comparing the Popper and Vapnik-Chervonenskis dimensions, in: Journal for the General Philosophy of Science 40 (1), 51-58.
20. Immanuel Kant. Kritik der reinen Vernunft. Johann Friedrich Hartknoch, 1781. (A). Zweite Auflage, 1787 (B).
21. Peters, J.; Janzing, D; Schölkopf, B. (2017), Elements of Causal Inference. Foundations and Learning Algorithms Cambridge (Mass.), 6f.
22. Pearl, J. (2009), Causality: Models, Reasoning, and Inference, Cambridge (Mass.).
23. Peters, J.; Janzing, D; Schölkopf, B. (2017), Elements of Causal Inference. Foundations and Learning Algorithms Cambridge (Mass.), 144.
24. B. Schölkopf, D.W. Hogg, D.Wang, D. Foreman-Mackey, D. Janzing, C.-J. Simson-Gabriel, J. Peters (2016), Modeling confounding by half-sibling regression, in: Proceedings of the National Academy of Sciences 113 (27), 7391-7398
25. Mainzer, K. (2020), Quantencomputer. Von der Quantenwelt zur Künstlichen Intelligenz, Springer.
26. K. Friston/I. Harrison/W. Penny (2003), Dynamic causal modelling, in: NeuroImage 19, 1273-1302
27. Thomas Bayes. An Essay towards solving a Problem in the Doctrine of Chances. In: Philosophical Transactions. Band 53, 1763, 370–418.
28. Mittelstrass, J. (1995), Regulae philosophandi, in: J. Mittelstrass (Hrsg.), Enzyklopädie Philosophie und Wissenschaftstheorie Bd. 3, J.B. Metzler: Stuttgart, 536–53.7.

29. Mainzer, K.; Schröder-Heister, P. (1995), Bayessches Theorem, in: J. Mittelstrass (Hrsg.), Enzyklopädie Philosophie und Wissenschaftstheorie Bd. 1, J.B. Metzler: Stuttgart, 254–256.
30. Skilling, J.; Sibisi, S. (Eds.) (1996), Maximum Entropy and Bayesian Methods, Kluwer: Dordrecht.
31. Baldi, P.; Brunak, S. (2001), Bioinformatics. The Machine Learning Approach, MIT Press: Cambridge MA, 57.
32. Jeffreys,W.H.; Berger, J.O. (1992), Ockham's razor and Bayesian analysis, in: Americian Science 80 1992, 64-72.
33. Baldi, P.; Brunak, S. (2001), Bioinformatics. The Machine Learning Approach, MIT Press: Cambridge MA 2001, 68.
34. Hornik, K.; Stinchcombe, M.; White, H. (1990), Universal approximation of an unknown function and its derivatives using multilayer feedforward networks, in: Neural Networks 3, 551-560.
35. Minsky, M.; Papert, S. (1969), Perceptrons, Cambridge (Mass.), expanded edition 1988.
36. Möller, K.; Paaß, G. (Hrsg.) (1994), Künstliche neuronale Netze: eine Bestandsaufnahme, in: KI – Künstliche Intelligenz 4, 37–61. Literaturverzeichnis 43
37. Rummelhart, D.E.; Hinton, G.E.; Williams, R.J. (1986), Learning representation by back propagating errors, in: Nature 323, 533-536.
38. Hornik, K.; Stinchcombe, M.; White, H. (1989), Multilayer feedforward networks are universal approximators neural networks, in: Neural Networks 2, 359-366.
39. Senior, A.W., et al. (2020), Improved protein structure prediction using potentials from deep learning, in: Nature 577 2020, 706-710.
40. Mainzer, K. (2020), Leben als Maschine: Wie entschlüsseln wir den Corona-Code? Von der Systembiologie und Bioinformatik zu Robotik und Künstlicher Intelligenz, Brill Mentis: Paderborn, 127 ff.
41. Baldi, P.; Brunak, S. (2001,) Bioinformatics. The Machine Learning Approach, MIT Press: Cambridge MA, 267.
42. Felsenstein, J. (1981), Evolutionary trees from DNA sequences: A maximum likelihood approach, in: Journal of Molecular Evolution 19, 368-376.
43. Popper, K.R. (1972), Objektive Knowledge. Oxford University Press.
44. Brown, T. B. et al. (2020), Language models are few-shot learners, 2020, in: arXiv: 2005.14165, cs.CL.
45. Riemann, B. (1860), Über die Anzahl der Primzahlen unter einer gegebenen Größe., in: Monatsberichte der Königlichen Preußischen Akademie der Wissenschaften zu Berlin, 671–680.
46. Siegel, C. L. (1932), Über Riemanns Nachlaß zur analytischen Zahlentheorie. In: Quellen Studien zur Geschichte der Math. Astron. Und Phys. Abt. B: Studien, 2:45–80, 1932. Reprinted in Gesammelte Abhandlungen, Vol. 1. Berlin: Springer-Verlag, 1966.
47. Borgesius, F. Z. (2018), Discrimination, artificial intelligence, and algorithmic decision-making. Directorate General of Democracy; Council of Europe, Strasbourg.

48. Bryson, M. C. (1976), The literary digest poll: Making of a statistical myth, in: The American Statistician, 30(4),184–185.
49. Dreyfus, H. (1992), What Computers Still Can't Do: A Critique of Artificial Reason. MIT Press.
50. McCarthy, J. and Hayes, P.J. (1969), Some philosophical problems from the standpoint of artificial intelligence, in: Machine Intelligence, 4:463–502, 1969.
51. S. Hanks and D. McDermott (1987), Nonmonotonic logic and temporal projection, in: Artificial Intelligence, 33(3), 379–412.
52. Quine, W. V. O. (1959). Methods of Logic. Holt, Rinehart and Winston. 2nd edition.

Theoretical Limits

3

3.1 Can "calculating" be Learnt Statistically?

- Interviewer: What is your biggest strength?
- Me: I am an expert in machine learning.
- Interviewer: What's 6 + 10?
- Me: Zero.
- Interviewer: Nowhere near. It's 16.
- Me: Ok, It's 16.
- Interviewer: What is 10 + 20?
- Me: It's 16.

This is an old, simplified illustration of machine learning—by now, ChatGPT may not fail on this question, but it is known for having serious problems with Mathematics. In fact, it is the aim of the illustration to show that statistics is not the adequate approach to calculate elementary mathematical expressions. Even if it whould be possible to learn mathematical functions, such as addition, correctly in a purely statistical way from example calculations, it is not to see what would be the advantage over traditional programming techniques, which provide sophisticated algorithms for calculations with natural numbers.

However, the situation can change fundamentally if one goes beyond elementary operations, and we see potential for Artificial Intelligence when it we would like, for example, solve

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical*, Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6_3

numerically differential equations which do not have solutions in elementary functions.

In the present context, however, we are interested in the potential or limits of AI for supposedly simple arithmetic operations [1]. A good example is a prime number test.

Example

First one can ask whether AI can learn purely statistically the property of primality of numbers. Here the problem is certainly not that there are not enough data available. If one marks correctly the prime numbers in a set of millions of numbers, will AI correctly mark other numbers as prime numbers or composite numbers? Without having tested this question in practice, let us give some theoretical considerations.

Of course, the prime number property can be determined for each natural number $n > 1$ in a brute-force manner by trying to divide n by every smaller number. If one finds only 1 as a divisor, one has a prime number. This test is hopelessly inefficient. But also the method of the sieve of Eratosthenes [2, p. 31], known since antiquity and still taught in school, is—like all other known deterministic prime number tests—inefficient form the point of view of complexity theory.[1] But independently of this it is an interesting question whether the process underlying the sieve of Eratosthenes can be taught to the AI. From a practical point of view, however, one would expect more, namely that the AI could find a faster prime number test.

Should the prime number property be based on a certain regularity, which has escaped the mathematicians until today, it seems possible that deep learning, with the possibility to

[1] Nevertheless, the idea underlying the sieve of Eratosthenes has an algorithmic added value, see the example below in the discussion of cryptographic protocols.

map much more structure internally than a human being can do with a piece of paper, recognizes this regularity and accordingly produces correct results in a comparatively short time.

If, however, such a regularity does not exist at all, then, of course, it cannot be encoded in a neural network. Here one could imagine that a formal conceptualization could be developed, according to which the AI cannot learn a prime number (because if it did, this would imply the existence of a certain regularity). ◀

Example

But what does it mean that there is no regularity? Mathematics knows this phenomenon from the transcendental numbers, especially in the context of the study of the Ludolphine number π. The decimal expansion of π does not follow any recognizable rule, which would allow e.g., to determine the n^{th} decimal fraction directly from n.[2] Leibniz had still the hope to find a periodic regularity, if one would write π to another base. The proof of the irrationality of π shows that this was an illusion. Nevertheless, π can be stated in a mathematical expression revealing a regularity, not as a decimal fraction, but as a continued fraction. Also here it is by no means to be expected that by statistical learning of many decimal fraction places, AI could ever come to such a continued fraction representation. ◀

Another problem, which arises with statistically learned mathematical operations, is scaling. Let us imagine an AI program that has learned prime numbers in a range of up to 1000 digits and is able to determine correctly primes in this range. Will it,

[2] By now, we know algorithms for a direct calculation of the digit in hexadecimal or binary expansion [3, §1.2]. But this still does not give a *regularity* in the usual sense.

therefore, still deliver correct results, if it examines a number with 10,000 digits? There are doubts, which can be illustrated by an example from Euler [4].

Example

Euler discovered that the polynomial $x^2 - x + 41$ returns prime numbers for the first 40 values (his numbers started with 1); but for $x = 41$ it has the value 41^2. Thus, the sequence of prime numbers comes to an end here. Even if this polynomial does not enumerate all prime numbers below a certain limit, it illustrates the possibility that there might be simple primality tests for finite domains. And AI could learn them as long as the learning data stay in the respective range—but it would fail outside of this range. ◀

This problem is intrinsic: AI is supposed to have a good interpolation behavior in the order of magnitude of the learning data. But about the extrapolation behavior, when one is far away from the learning data, one can, per se, not predict much—at least, as long as the internal algorithm, which the AI uses after learning the data, is not accessible.

3.2 Continuous Versus Discrete Problems

Οἱ πρῶτοι ἀριθμοὶ πλείους εἰσὶ παντὸς τοῦ προτεθέντος πλήθους πρώτων ἀριθμῶν.

Euclid, Elements, Book IX, Prop. 20.

The prime number test points to another general problem of (new) AI: their analysis methods usually assume a continuous relationship between the data. And this relation should be mathematically expressed in continuous functions over the real numbers. Discrete correlations are thus not directly captured. The qualitative difference was very well illustrated by Hermann Weyl [5, p. 37] with a reference to Plato's number of the citizens of the ideal city, which is supposed to be $5040 = 7!$

> $5040 = 2^4 \cdot 3^2 \cdot 5 \cdot 7$ has many divisors, while 5039 is a prime number. If in Plato's ideal city *one* citizen dies overnight and thereby reduces the number of citizens to 5039, it is immediately completely corrupted.

In areas where exact numerical values play a role, such as, for example, cryptography, AI methods—at least as far as they work under the condition of continuity—are not directly applicable. This problem occurs everywhere where there is a specific functional dependency of discrete input values and discrete results. In this case, statistical methods are not appropriate. Hans Leiß (Munich) illustrated this by the provocative question, how a C++ compiler should be obtained by machine learning (a lack of data would not be the problem).

And even if a functional relationship learned by AI would be piecewise continuous, it is not clear how it will behave at a point of discontinuity. We encounter such a situation, for example, when AI produces—quite impressively—male and female faces. At the switch from male to female faces, nonsensical images may appear for a short time.

The discussion of prime numbers in the last two sections is not an arbitrarily chosen example. These play a central role in practically all common cryptographic protocols. Some basic concepts of cryptography will be briefly outlined in the following, to get an idea of the range of problems that the AI faces if it also wants to play a role in this area.

For secure cryptographic procedures, it is not only the increase in the computing power of computers that is decisive. The ingenious performance of a single mathematician could also ensure a breakthrough. Algorithms based on number sevens actually lead to an improvement over RSA algorithms.

In general, the history of cryptographic methods shows that their security depends on the degree of difficulty of the mathematical background knowledge used in each case. The intelligence of the problem solution is thus reflected in the mathematical background knowledge used.

Would AI be able to do this? Let's first take a look at the RSA encryption.

Example

In addition to the Euclidean algorithm, the RSA encryption algorithm uses theorems from number theory that go back to Leonard Euler and Carl Friedrich Gauss. This refers to modular arithmetic with integers, which is used in computer science. Depending on their finite number of bits, computers can only calculate up to numbers smaller than a certain limit. Let $\mathbb{Z}_n = \{0, 1, \ldots, N-1\}$ be any finite set of integers. In multiplications with numbers from $\mathbb{Z}_n$, the results of a certain size would lead out of this set. Therefore, arithmetic operations modulo (abbreviation: mod) N are introduced. Gauss had already examined congruent numbers modulo N in his number-theoretical work.

The RSA algorithm is based on modular computing. It guarantees the difference between the encoding and decoding keys. Furthermore, the encryption key is made public, while the decryption key remains secret. This is why one also speaks of a public-key crypto procedure.

In asymmetric encryption such as RSA, two keys are therefore used. Senders of a message use public keys to encrypt their messages. The recipient has a secret key which he does not communicate to anyone. With this, he decrypts the messages that were encrypted with the public key. This is to exclude the possibility that someone other than the sender and recipient can decrypt the message. The advantage of this procedure is that no secret key has to be exchanged between sender and receiver. Such an exchange would represent a considerable security risk, as it could in principle be overheard.

Based on the mathematics of modular computing, the RSA algorithm proceeds in the following steps [6]:

1. Two different prime numbers p and q are chosen at random.
2. The product $n = pq$ and its function value $\phi(n) = (p-1)(q-1)$ of the Euler function are calculated.
3. A (small) odd number e (encryption) is chosen, which is not a divisor of $\phi(n)$.

4. The solution d (decryption) of the modular equation $e \cdot d \equiv 1 \mod \phi(n)$ is calculated.
5. The numbers e and n are published as public keys.
6. The number d is the secret key and the numbers p, q and $\phi(n)$ are eliminated.

The recipient can decode the messages with a secret key d because he knows the prime factors p and q of n. It is assumed that only the one who knows the prime factors p and q of n can decode the encoded message. Furthermore, it is assumed that there is no classical efficient algorithm for the factorisation of n. In fact, none is known so far, but it is not excluded in principle. So far, all security guarantees of the RSA procedure are based on these facts and assumptions. However, should the factorisation of n be possible with a quantum algorithm, the security guarantee no longer applies. ◀

Example

This also applies to the next steps in the improvement of cryptology through the so-called number sieves. After the sieve of Erathostenes, number sieves were proposed for the factorisation of integers.

In cryptology, number sieves were proposed for the factorisation of integers. After various precursors, Carl Pomerance developed the so-called square sieve in 1981, which was more powerful than all factorisation methods proposed up to that point [7, 8, Sect. 6.1: The quadratic sieve factorization method, 227–244]. This method depends only on the size of the number to be factorised and, after various improvements, is still the fastest classical factorisation method for numbers up to 100 decimal places. The "sifting" here refers to the search for divisors. The designation of this factorisation procedure as "quadratic" comes from the fact that the factorisation of an integer n in a product is represented by a corresponding difference of squares, i.e.

$x^2 - y^2 = (x + y)(x - y) = n$. The equation $y^2 = x^2 - n$ thus yields the divisors $(x + y)$ and $(x - y)$ of n.

This time, Fermat contributes the mathematical background knowledge: Using a factorisation method named after Fermat, the function value of $q(x) = x^2 - n$ is calculated for different numbers x until $q(x)$ is a square number. One starts with the smallest number x that is smaller than the root of n. Then x is increased by 1 in each subsequent step until the goal is reached. The question is how to determine as efficiently as possible which function values $q(x)$ can be multiplied as a square. In a first step, the approach is to look for the corresponding congruences $x^2 \equiv q \bmod n$ instead of the equations $q(x) = x^2 - n$. This step is also called "sifting". In a second step, those congruences are selected from which a quadratic congruence results by multiplication.

In 1994, a number with 129 decimal places was factorised in this way. However, the effort was enormous: In the first step of the "sifting", 600 employees collected congruences in 8 months and sent them to a central computer. The second step of selection was carried out by a supercomputer for 298 GB of data in 45 h. For a length $N = \log n$ of the input number n, the running time of the Quadratic Sieve is $e^{c \cdot N^{\alpha} (\log N)^{1-\alpha}}$ with $\alpha = \frac{1}{2}$ and $c = 1$. For $\alpha = 1$ there would be exponential growth, for $\alpha = 0$ polynomial computation time. The quadratic sieve therefore works with superpolynomial but subexponential computing time.

Another improvement is the so-called number body sieve with $\alpha = \frac{1}{3}$, which goes back to a proposal by Michael J. Pollard in 1988 [9]. It is used for numbers above 100 digits, but with the enormous effort of several hundred computers computing in parallel. Mathematically, the number sieve is a generalisation of the quadratic sieve. Instead of the ring $\mathbb{Z}$ of integers, other algebraic number rings are considered, with which the divisors can be found more quickly. However, the computing time to factorise a number n is still superpolynomial, albeit subexponential. ◀

Example

Elliptical curves

In these cases, security is therefore achieved through mathematical background knowledge from number theory and technology based on the division of labour. In the meantime, increasingly more and more sophisticated mathematical background knowledge is being used:

As computing power grows, so does the danger that encrypted messages will be decrypted. Cryptologists are reacting with longer keys. This may not be a problem for supercomputers, but it has disadvantages with the many small end devices such as smartphones. In order to achieve a comparatively equally efficient encryption with small keys as with factorisation, for example, asymmetric cryptosystems with elliptical curves are used. The reason is that an addition of curve points can be defined on elliptic curves, which is suitable for encryption methods [10].

In fact, the discrete logarithm problem for elliptic curves is harder than the factorisation of integers [11]. Therefore, asymmetric cryptosystems based on elliptic systems only require significantly shorter keys if the security requirement is appropriate. With a key length of e.g. 160 bits, a similar security is achieved as with an RSA system with 1024 bits. This is why encryption methods with elliptical curves are used for devices with small memory and computing capacities, such as smartphones. The running time of the fastest encryption algorithms with elliptic curves is of the order of $2^{\frac{n}{2}}$, where n is the bit length of the size of the body used.

Mathematically, the theory of elliptic curves is extremely interesting. After their applications to real, complex and rational numbers, finite bodies and number theory, the spectacular solution of Fermat's problem in the 1990s by Andrew Wiles succeeded on this basis. While there is only elementary number theory behind the RSA cryptosystems, elliptic cryptosystems require sophisticated algebraic number theory. One could therefore speculate whether this also opens up further applications for cryptology on a classical basis. ◀

Example

Quantum algorithms

However, anyone who believes that they can reject AI with a mathematical sense of superiority should keep an eye on the technical progress of computers. Quantum computers overcome the security of RSA procedures, which depends on the difficulty of the mathematical background knowledge used:

In 1994, Peter Shor found an efficient algorithm of the factorisation problem for a quantum computer in polynomial time [12, 13, p. 733–753]. Central to this is the use of superimposed states, which allow many calculations to be performed simultaneously in a gigantic quantum parallelism. While a superposition of all 2^m possible m quantum bit words can be stored in an m quantum bit large register of a quantum computer, only one of the 2^m possible m bit words can be stored in the m bit large register of a classical computer.

Shor's algorithm uses this quantum parallelism to find periods of modulus functions from which the prime number factors sought can be derived. With numbers of a word length of more than 1024 bits to be factorised, corresponding calculations by classical computers with serial processing or low parallelisation are practically impossible. When quantum computers are technically realised, Shor's quantum algorithm will crack every version of an RSA encryption method. Then the previous security of the global information society will collapse.

The basic idea of Shor's algorithm is that a number n can be factorised if the period of the modulus function $f(x) = a^x \bmod n$ can be found for a number a smaller than n. To determine this period, a quantum algorithm is now to be given [14]:

By definition, the moduli function f maps the set of integers $\mathbb{Z}$ to the restricted set of numbers $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. For the input $0, 1, \ldots, N-1$ of the quantum algorithm, N is chosen in the order of n^2. We thus assume a function f with which the set $\{0, 1, \ldots, N-1\}$ is mapped onto the set $\{0, 1, \ldots, n-1\}$ with period p, i.e. $f(x+p) = f(x)$ applies for all x from $\{0, 1, \ldots, N-1\}$.

That means in the mathematical quantum formalism: The corresponding unitary operator U_f of f maps two quantum registers $|a\rangle\,|b\rangle$ to $|a\rangle|b \oplus f\,(x)\rangle$. To achieve the desired acceleration of the search for periods of the modulus function, the quantum Fourier transform is applied to it. Shor's algorithm is thus based on the application of the unitary transformation of the modulus function $f(x) = a^x \bmod n$ to factorise n and the quantum Fourier transform. ◀

With regard to all these mathematical and physical theories behind AI, one will rightly underline the supremacy of mathematics and human mind over AI: At first, humans have to come up with that the world is based on the mathematical laws of quantum mechanics, that with this physical background knowledge computers can be built and, finally, that mathematical background knowledge about modulus functions and fast Fourier transformations lead to algorithmical success. However, it would be too early to jubilant to assume that there is a mathematics behind quantum mechanics, which is fundamentally inaccessible to AI. The opposite is the case: as is well known, statistics play a fundamental role in quantum mechanics. In fact, the mathematics of (classical) statistical learning (deep learning) can be translated into the quantum mechanical formalism. Neuronal quantum networks with deep learning are the focus of current research.

3.3 Which Role Does Random Play in AI?

I, at any rate, am convinced that *He* is not playing at dice.
ALBERT EINSTEIN in a letter to Max Born 14 December,1926

Random-based methods decide on individual solution steps, such as by the throw of a coin. They can solve difficult problems that could not be solved with non-random methods, because the search space for problem solutions is too large. This advantage, however, has a price: Occasionally, such methods provide wrong answers.

Random can even be useful in proofs. Proofs are an expression of mathematical intelligence. Since the beginnings of AI, automatic proofs have therefore played a central role in symbolic AI. There proofs are fully determined step by step. Proofs are by no means only of theoretical importance. Practical challenges of digitalisation, such as security in cryptology, blockchain and digital currencies are increasingly made dependent on proof tasks that work with randomness.

But how do you convince someone that they have a proof for an assertion? Traditionally, the only possibility is assumed to present the proof to the doubter step by step. For practical tasks, however, this is by no means absolutely necessary or expedient.

A zero-knowledge proof, on the other hand, is a procedure with which one can convince an opponent with a certain probability, without revealing any information about the proof [15]. This is by no means unusual, but rather everyday life in the information age. Detailed analyses are hardly ever read any more. Fragments, partial quotations, and headings are used to obscure the essential message with an (intuitively assumed) degree of probability and to convince the public. In an AI, such intuitive procedures would have to be translated into algorithms. For this purpose the technique of interactive proofs can be applied [16, p. 183 ff.].

Example

An interactive question-and-answer dialogue between a prover and a verifier is used to check knowledge-free evidence. This dialogue is recorded in various rounds of a protocol. In the process, the prover has the task to convince the verifier of the validity of the assertion by answering questions. Both dialogue partners work with random information [17]. The verifier must only be able to distinguish between correct and incorrect evidence with a high probability.

Interactive proof protocols with k dialogue rounds are denoted $\mathrm{IP}(k)$ [18]. In this case, the random bits used remain secret. In the dialogue form denoted by $\mathrm{AM}(k)$, the random bits are revealed. The “A” stands for the king Arthur, who, as

a verifier, verifies the proofs of the magician Merlin for "M". In this case it can be shown that interactive proofs with an arbitrary number of k dialogue rounds can get by with only 2 rounds, i.e. $\mathrm{AM}(k) = \mathrm{AM}(2)$ for $k \geq 2$. AM(2) means that Arthur sends a question to Merlin in the 1st round and Merlin answers with a proof. In the 2nd round, Arthur must accept or reject Merlin's proof. However, it does not matter whether the random bits are kept secret or not. Therefore, it follows for the protocols $\mathrm{IP}(k) \subseteq \mathrm{AM}(k+2)$ and $\mathrm{IP}(k) = \mathrm{AM}$ for $k \geq 2$ [19].

The complexity class $\mathrm{IP} = \mathrm{IP}(\mathrm{Poly})$ comprises proof systems in which the verifier may interact with the prover in a polynomial number of rounds, before making a decision. IP is very extensive, because it applies $\mathrm{IP} = \mathrm{PSPACE}$ [20]. This result is therefore significant, since it cannot be relativised and transferred to arbitrary oracles.

The class NP can be understood as an interactive proof system: The prover generates a proof, which the verifier checks in polynomial time. The question is whether the verifier can check the proof without having read it completely. Can the proof be written in a format that allows only a small part of it to be checked in order to determine its correctness with a high probability? In this case, the intuitive and incomplete handling of information by humans could be taken over by the effective algorithms of an AI. ◀

In fact, it can be shown that there are proofs for every language from NP, the verifier, using logarithmically many random bits, only has to read a constant number of bits of the proof. This description of NP is the famous PCP (probabilistic checkable proofs) theorem, which was substantially co-founded by the 2021 Abel Prize winner László Lovácz [21].

The class $\mathrm{PCP}(r(n), q(n))$ refers to the class of decision problems with probabilistically verifiable proofs that can be solved in polynomial time by exploiting at most $r(n)$ random bits and by reading at most $q(n)$ bits of the proof. Correct proofs should always be accepted. Incorrect proofs should be rejected with a probability greater than $1/2$. The PCP theorem then states that $\mathrm{PCP}(O(\log n), O(1)) = \mathrm{NP}$ [22].

The PCP theorem has great significance for the approximation of difficult problems. Thus, in the class MAXSNP of approximable optimisation problems [23], the aim is not to determine the solution exactly, but to find the best possible approximations. An example is the satisfiability problem MAXSAT, in which one searches for assignments that do not include the entire formula, but satisfy many clauses. Here it can be shown that under the assumption of $P \neq NP$ no MAXSNP-complete problem (e.g. MAXSAT) can be approximated well. Exact bounds could be found for e.g. 3-MAXSAT.

Let us return to the initial question of the role of random in AI. Probabilistic algorithms are often more efficient than deterministic algorithms if small error probabilities are accepted. One example is the verification of prime numbers. There are algorithms that check whether a number is divisible only by itself and one. But they are time-consuming. With an error probability, which can be reduced arbitrarily, efficient algorithms based on random numbers can be used for verification.

In fact, randomised algorithms are mainly of practical importance. They are often simpler than their deterministic counterparts. Occasionally no efficient deterministic algorithms are known. However, technical computers are never randomised, but deterministic.

A few random bits arise, for example, through unintentional ("random") mouse movements. Pseudo-random generators produce a large number of pseudo-random bits from a small number of real random bits. In this context, pseudo-random means that the bits cannot be efficiently distinguished from real random bits.

Theoretically, however, randomness does not have the significance that is suggested by the practical application successes of randomised algorithms. For example, the 2021 Abel Prize winner Avi Wigderson was able to show that, in principle, for any method that can solve a problem with a random toss, there is an almost just as efficient method without a random element [24, 25]. In this case, the random bits required for the probabilistic algorithms are generated by pseudo-random generators. Under certain conditions the real random bits used by the

pseudo-random generators can be eliminated. However, the algorithms obtained in this way are deterministic.

In AI discussions, randomness is often associated with spontaneity and creativity that are closed to a deterministic computer and thus to classical symbolic AI. The stories of the "random" and "spontaneous" occurrences of human creativity are simply too beautiful. This alleged limit of AI is at least put it into perspective by the mathematical result of Wigderson et al.: The same results of a "spontaneous" and "creative" intelligence could cum grano salis also be obtained on a deterministic computer.

3.4 Which Role Has Chaos in AI?

> In the beginning rose Chaos …
>
> Hesiod, Theogony, Vers 116.

In everyday language, chaos is understood as a complete mess that, like randomness, is not calculable and thus does not appear to be accessible to a computer-assisted AI. Mathematically, however, chaos is a precisely determined state of a dynamic system that is to be distinguished from randomness. In general, a complex dynamic system consists of a large number of elements. The microscopic states of the elements determine the macroscopic state of the system. For example, in a planetary system, the state of motion of a planet at a point in time is determined by its location and speed. But it can also be the state of motion of a molecule in a gas, the state of excitation of a nerve cell in a neuronal network or the state of a population in an ecological system. The dynamics of the system, i.e. the change of the system states in time, is described by time-dependent equations (e.g. differential equations). In deterministic systems, each future state is uniquely determined by the present state.[3]

[3] The following presentation of chaos theory follows the book: K. Mainzer (2016), Information, Algorithmus, Probability, Complexity, Quantum World, Life, Brain, Society, Berlin University Press [26, 67 ff.].

In linear systems, causes and effects are proportional. Mathematically, we then obtain an equation of the form $f(x) = c \cdot x$ with x-values, the function values $f(x)$ that depend on them and a constant of proportionality c. Since this equation represents a straight line with the gradient c in the coordinate system, it is called linear.

A solution to this equation can be represented as a time series of the location as a function of time. We know from mathematics: Linear equations are easy to solve. However, non-linear equations, which represent geometric curves, do not always allow for arbitrarily precise calculation, even with our best computers. In essence, non-linearity means that cause and effect are no longer proportional: A small local cause can result in a global effect. An example is weather forecasts, which depend on many interacting factors. Here, an unnoticed local turbulence can build up and change the entire weather.

In order to study non-linear dynamics, the so-called state space of a dynamic system is introduced in addition to time series analysis. The state of a dynamic system is determined by various quantities (e.g. the state of motion of a molecule by its location and momentum at a point in time). These state components are understood geometrically as coordinates of the state space of a dynamic system. They define a point in the state space that represents the system state. In contrast to the location space consisting of height, depth and width, state spaces can be defined by more than three coordinates (e.g. the disease state of a patient, which can be dependent on many symptoms).

In equilibrium, the state does not change and the corresponding point in the state space is fixed in time (fixed point). If the state changes, the state point generates a development curve (trajectory) in the state space. The phase portrait of a state space clearly shows how the state developments (trajectories) of a dynamic system result in characteristic patterns (attractors).

An attractor is a state into which a dynamic system is drawn (converges) in the long term:

- A state of equilibrium corresponds to a fixed point attractor that no longer changes over time (“remains fixed”). In the

state space, all lines of development (trajectories) then run ("converge") to this point as the final state. Linear systems only have fixed point attractors.
- Non-linear systems also have limit cycles in which states repeat periodically or, in the case of turbulence, chaos attractors in which the development lines condense completely irregularly and non-periodically in a limited area of the state space.

In random evolutions, all correlations have decayed into independent events and fluctuate irregularly over the entire state space. Dynamic complexity and chaos thus lie between complete regularity (as in linear systems) and randomness.

The time series does not change in the equilibrium case. This corresponds to a fixed point attractor in the state space. In the periodic case, the time series fluctuates between two fixed points. In the corresponding state space, the trajectory is closed and always returns to its initial state. In the quasi-periodic case, patterns repeat in the time series. This corresponds to a periodic pattern of the trajectory in the state space. In the chaos case, the time series develops completely irregularly and non-periodically. This corresponds to a trajectory in the state space that develops completely irregularly and non-periodically, but in a limited area—the chaos attractor.

In contrast to the three preceding cases, the chaotic development is sensitively dependent on the smallest changes in the initial values of a trajectory: Even the smallest differences lead to completely different developments after a few steps. This makes the difference between chaos and randomness clear:

- In deterministic chaos, the dynamics is completely determined by a non-linear growth law. Nevertheless, long-term effects are practically impossible to predict, since the computational effort grows exponentially due to the sensitive dependence on the initial data.
- In contrast to chaotic ones, random developments cannot be predicted in principle (i.e. even in the short term), since (as in the case of a fair coin toss, for example) all events are independent.

To measure the complexity of a time series and thus of a non-linear dynamic, we can determine, for example, the degree of non-periodicity or the sensitive dependence of a dynamic on its initial data. Thus, the so-called Lyapunov exponents can be used to measure whether and how strongly the trajectories drift apart in the state space in order to capture the degree of sensitive dependence (butterfly effect).

Deterministic chaos is therefore in principle computable. Computability limits are of a practical nature, but serious, since the smallest differences in the initial data can lead to different courses of events after only a few steps into the future and practically exclude long-term forecasts. In practice, it also proves difficult to distinguish random noise in data from chaotic behaviour, although mathematical methods are available for this. In classical mechanics, however, despite Poincaré's multibody problems and chaos, the world is completely determined and computable for a Leibnizian God who does not depend on "earthly" computers. For AI systems (in a classical world), it follows that their predictions, despite the constantly increasing computing power of supercomputers, become exponentially more difficult with longer time into the future. Quantum computers would, however, literally lead to a "quantum leap": Problems (e.g. the factorisation problem in cryptology) which were previously practically unsolvable, will then be solvable by a machine. Some theorists will then continue to argue with Gödel and Turing according to which there are problems that cannot be decided in principle. But what significance does that have for the rest of humanity?

3.5 Is There a Theory of Computability and Complexity for AI?

> Will you not answer "yes" to this question?
>
> [27, p. 128]

For classical computing, complexity classes were introduced to distinguish between P, NP, NP-hard and NP-complete problems [16, 172 ff.]. While these complexity distinctions refer

to the different time required to solve the problem, the storage capacity can also be taken into account. The polynomial space PSPACE is the class of problems that can be solved by an algorithm whose search space can be polynomially restricted by an upper bound in all examples. EXP is the very powerful class of problems whose solutions require exponential time. We know that $P \neq EXP$. Then at least one of the relations $P \neq NP$ or $NP \neq EXP$ must hold. However it is not yet known which of the two possibilities holds.

Since the performance of artificial intelligence depends on different classes of algorithms, the computability and complexity theories are fundamental. We first examine hierarchies of these complexity classes and then ask whether measures of the degree of artificial intelligence can be deduced from them. From computability theory S. C. Kleene's arithmetical hierarchy is known, in which the class of (Turing) decidable predicates is extended step by step by the alternating addition of existential and all-quantifiers [28, Chap. 3]. Thus an infinite hierarchy of classes Σ_n, Π_n and Δ_n is generated with $\Sigma_0 = \Pi_0$ as class of Turing-decidable predicates, Σ_n (Π_n) predicates with prefix of n alternating quantifiers which begins with the existential (all) quantifier, and $\Delta_n = \Sigma_n \cap \Pi_n$.

In everyday decisions, we often fall back on background knowledge that we cannot decide and prove ourselves. This is also true in research based on the division of labour, where we often make use of the knowledge of neighbouring disciplines without being able to decide on this knowledge ourselves. Often these are also assumed hypotheses. This type of natural intelligence can be represented in algorithms:

In computability theory, Turing had introduced the concept of an oracle Turing machine. In this case, a Turing machine additionally uses a device (oracle), which answers questions without being able to decide this knowledge itself. If a problem A (formally: predicate) is decided by an oracle Turing machine in polynomial time with an oracle B, then A is called Turing-reducible to B. Now, complexity classes can be defined relative to an oracle. As an example, the class of all problems which can be decided in polynomial time with an oracle B is called P^B. Similar to Kleene's arithmetic

hierarchy, an infinite hierarchy of complexity classes can be formed with each level referring to the one below as an oracle. This polynomial hierarchy PH begins with $\Sigma_0^P = P$ and $\Sigma_1^P = NP$, followed by the layers $\Delta_k^P = P^{\Sigma_{k-1}^P}$, $\Sigma_k^P = NP^{\Sigma_{k-1}^P}$, and $\Pi_k^P = \text{co-}\Sigma_k^P$ with all complements $\text{co-}\Sigma_k^P$ of Σ_k^P.

With regard to subsymbolic AI, the complexity of classical probabilistic algorithms is of interest [29]. BPP (bounded-error probabilistic polynomial time) refers to the class of problems for which there is a polynomial randomised algorithm that solves each example with a success probability of at least about 2/3. Randomised algorithms can sometimes solve problems better than deterministic algorithms. So BPP problems have either a polynomial deterministic solution algorithm or a probabilistic algorithm that gives a wrong result with a probability no worse than about 1/3. These limits need not be fixed, but should be between 0 and below ½. After all, according to the central limit theorem of probability theory, this does not change anything: after many runs, the probability of generating an error each time is small. The BPP class thus includes the class of P problems in any case. It is even often assumed that BPP = P. But there is no proof of this yet.

A random-based algorithm can be introduced for the prime number test [30]. This algorithm uses random throws for the calculation in polynomial time. Occasionally, wrong answers are given. A one-sided error occurs when prime numbers are always recognised, but composite numbers are only recognised with probability $1 - \delta$. The complexity class RP (randomized polynomial time) covers problems with random-based algorithms of polynomial duration with one-sided error. If two-sided errors are also permitted, the result is the class BPP. It is known that BPP is part of PH.

Random-based algorithms of polynomial duration do not have to be efficient: The class PP (probabilistic polynomial time) comprises algorithms in which the probability of a correct answer is little greater than 1/2. In contrast, with BPP algorithms the answers are correct with a high probability. It can be proved

that NP is part of PP. However, it is not yet known whether NP is also part of BPP. The polynomial hierarchy PH is contained in the Turing closure of PP (i.e. $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}}$) [31].

Random-based algorithms with one-sided or two-sided error are also called Monte Carlo algorithms. In Las Vegas algorithms, the correct answer is always given. The price here, however, is that the runtime depends on chance and can take a very long time. The class ZPP (zero error probabilistic polynomial time) includes all problems that are solved by Las Vegas algorithms with polynomial runtime. The following applies: $\mathrm{ZPP} = \mathrm{RP} \cap \text{co-RP}$. The prime number test can be carried out with Las-Vegas algorithms as well as with Monte Carlo algorithms [32, 33]. An overview of these complexity classes is given in Fig. 3.1.

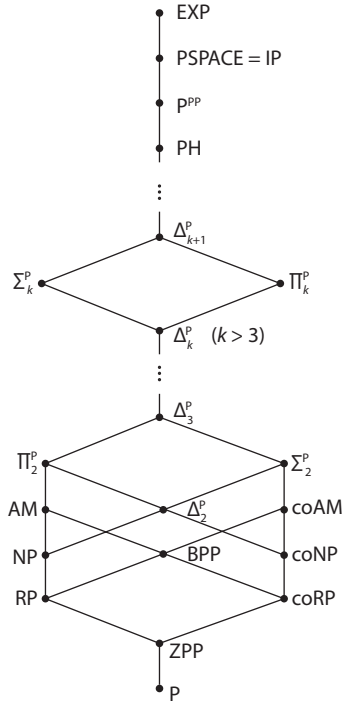


Fig. 3.1 Complexity hierarchy for probabilistic and deterministic algorithms [16, p. 176]

The performance and limitations of symbolic AI depend on the (deterministic) algorithms on which they are based. It stands to reason that sub-symbolic AI such as machine learning on the basis of statistical learning theory is brought together with the complexity classes of probabilistic algorithms. The performance and limits of probabilistic neuronal networks can be determined by the corresponding complexity classes. But which degrees of intelligence would an artificial "brain" based on quantum computing have? Which limits arise on the basis of a complexity theory of quantum computing?

In quantum computing, BQP (bounded-error quantum polynomial time) refers to the class of problems for which there is a polynomial quantum algorithm. BQP contains problems, such as the factorisation of large numbers, for which it is assumed that there is no classically realisable solution. However, Shor's algorithm only proves that the factorisation problem is BQP. It is not impossible that there is a classical solution. So it is not certain how exactly BQP relates to P, NP and PSPACE (Fig. 3.2).

The Shor Algorithmus requires gates of the order of $O\big((\log n)^3\big)$. The computation time of the quantum algorithm is of the order of $O\big(\log\log n \cdot (\log n)^3\big)$ [21] The classical part of Shor's algorithm only uses multiplications of the order of $O(\log n)$. The computation time of Shor's algorithm as a whole to determine a true divisor of the integer n is therefore of the order $O\big(\log\log n \cdot (\log n)^3\big) = O\big((\log n)^4\big)$. The decisive step in accelerating the determination of the period is the Fourier transformation, which was translated into a quantum algorithm. This is at the same time the theoretical breakthrough that classically non-polynomial solvable problems become polynomial solvable. The question is whether quantum computers can also solve other problems polynomially that do not depend on quantum Fourier transformations. No definitive answers and limits can yet be given for an AI of the future. Research into quantum computing and its relationship to classical complexity theory has only just begun.

Classical complexity theory is fundamental to the security of the cryptographic protocols mentioned above. With regard to statistics-based AI, the question arises as to whether this theory

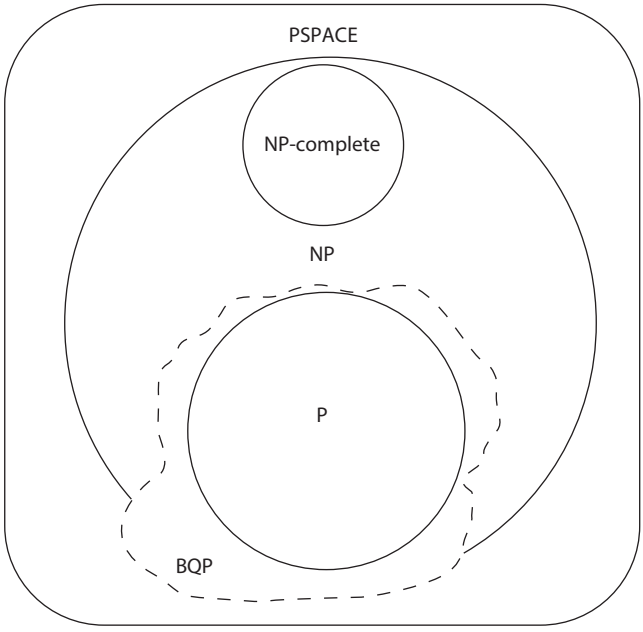


Fig. 3.2 Complexity classes with Quantum complexity [34, p. 104]

remains valid in the same way, or what should take its place. Although the software systems of machine learning are themselves being implemented in the usual programming languages, such as C++ and Python, are subject to the known limits of complexity and computability, in that the deterministic computations within a neural network cannot be suddenly accelerated. In a statistical data evaluation is not about a deterministic and discrete calculation (which, as a rule, is also only carried out under the aspect of the worst-case complexity). It is therefore quite conceivable, that—certainly in the sense of an average-case complexity—classical complexity barriers could be exceeded. We do not know whether there have been investigations in this direction that would lead us to expect such results. However, the need for a theoretical delimitation of the performance of statistics-based AI arises directly from the question of the security of

our cryptographic protocols. It is to be expected, that even deep learning cannot compromise RSA, even if oodles of coded and decoded data is made available. But can this be formally proved?

References

1. Kahle, R. (2021), Primzahlen als Herausforderung, in: R. Reussner, A. Koziolek, and R. Heinrich (Eds.), INFORMATIK 2020, Lecture Notes in Informatics, pages 719–727. Gesellschaft für Informatik.
2. Hoche, R. (Ed.) (1866), Nicomachi Geraseni Pythagorei Introductionis arithmeticae libri II. Teubner.
3. Berggren, L.; Jonathan Borwein, J.; Peter Borwein (2004), A Pamphlet on Pi, in: L. Berggren, J. Borwein, P. Borwein (Eds.), Pi: A Source Book. 3rd edition, Springer, 721–739.
4. Euler, L. (1771), Extrait d'un lettre de M. Euler le pere à M. Bernoulli concernant leM'emoire imprimé parmi ceux de 1771, 318. Nouveaux Mémoires de l'Académie royale des Sciences. Berlin, 1774, 35–36, 1772.
5. Weyl, H. (1971). Über den Symbolismus der Mathematik und mathematischen Physik, in: K. Reidemeister (ed.) Hilbert, 20–38. Springer.
6. Homeister, M. (2018), Quantum Computing verstehen, Springer: Berlin 5. Aufl., 195–196.
7. Pomerance, C. (1982), Analysis and comparison of some integer factoring algorithms, in: Computational Methods in Number Theory, Part I, H.W. Lenstra, Jr. and R. Tijdeman, eds., Math. Centre Tract 154, Amsterdam, 89–139
8. R. Crandall, C. Pomerance (2001), Prime Numbers: A Computational Perspective. Springer, New York.
9. Lenstra, A.K.; Lenstra (1993), H.W., The Development of the Number Field Sieve, Lecture Notes in Mathematics V, 1554.
10. Werner, A. (2002), Elliptische Kurven in der Kryptographie, Springer, Berlin.
11. The case for Elliptic Curve Cryptography: https://www.nsa.gov/business/programs/elliptic_curve.shtml (abgerufen 06.052020).
12. Shor, P.W. (1997), Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, in: SIAM J. Computing 26 1997, 1484–1509.
13. Ekert, A.; Jozsa, R. (1996) Quantum computation and Shor's factoring algorithm, in: Rev. mod. Phys. 68.
14. Benenti, G.; Casati, G.; Strini, G. (2008), Principles of Quantum Computation and Information. Vol. I: Basic Concepts, World Scientific Singapore, 161–162.

15. Quisquater, J.-J.; Guillou,L. (1990), How to explain zero-knowledge protocols to your children, in: Advances in Cryptology – CRYPTO '89, Lecture Notes in computer Science 435, 628–631.
16. Köbler, J.; Beyersdorff, O. (2006), Von der Turingmaschine zum Quantencomputer – ein Gang durch die Geschichte der Komplexitätstheorie, in: W. Reisig, J.-C. Freytag (HRSG.), Informatik. Aktuelle Themen im historischen Kontext, Springer: Berlin.
17. Babai, L. (1985), Trading group theory for randomness, in: Proc. 17th ACM Symposium on Theory of Computing, ACM Press, 421–429.
18. Goldreich, O.; Micali, S.; Rackoff, C. (1989), The knowledge complexity of interactive proof systems, in: SIAM Journal on Computing 18(2), 186–208.
19. Goldberg, A.; Sipser, M. (1989), Private coins versus public coins in interactive proof systems, in: S. Micli (Hrsg.), Randomness and Computation, Advances in Computing Research 5, JAI Press, 73–90.
20. Shamir, A. (1992), IP=PSPACE, in: Journal of the ACM 39(4), 869–877.
21. Feige, U.; Goldwasser, S.; Lovasz, L.; Safra, S.; Szegedy, M. (1996), Interactive proofs and the hardness of approximating cliques, in: Journal of the ACM 43, 268–292.
22. Arora, S.; Safra, S. (1998), Probabilistic checking of proof: A new characterization of NP, in: Journal of ACM 45(1), 70–122.
23. Papadimitriou, C.H.; Yannakakis, M. (1991), Optimization, approximation, and complexity classes, in: Journal of Computer and System Sciences 43(3), 425–440.
24. Nissan, N.; Widgerson, A. (1994), Hardness vs. randomness, in: Journal of Computer and System Sciences 49(2), 149–167
25. Impagliazzo, R.; Widgerson, A. (1997), P=BPP unless E has sub-exponential circuits: derandomizing the XOR lemma, in: Proc. 29th ACM Symposium on Theory of Computing, ACM Press, 220–229.
26. Mainzer K (2016) Information: Algorithmus-Wahrscheinlichkeit-Komplexit¨at- Quantenwelt-Leben-Gehirn-Gesellschaft. Berlin.
27. Dershowitz, N. (2005). The four sons of Penrose. In G. Sutcliffe and A. Voronkov (Eds.), Proceedings of the Eleventh Conference on Logic Programming for Artificial Intelligence and Reasoning (LPAR) (Montego Bay, Jamaica), Volume 3835 of Lecture Notes in Artificial Intelligence, pp. 125–138. Springer.
28. Mainzer, K. (2018), The Digital and the Real World. Computational Foundations of Mathematics, Science, Technology, and Philosophy, World Scientific Singapore.
29. Hidary, J.D. (2019), Quantum Computing: An Applied Approach, Springer: Cham, 20–21.
30. Solovay, R.; Strassen, V. (1977), A fast Monte-Carlo test for primality, in: SIAM Journal on Computing 6, 84–85.

31. Toda, S. (1991), PP is as hard as the polynomial-time hierarchy, in: SIAM Journal on Computing 20, 865–877.
32. Adleman, L.; Huang, M. (1987), Recognizing primes in random polynomial time, in: Proc. 19th ACM Symposium on theory of computing, ACM Press, 462–469.
33. Rabin, M.O. (1980), Probabilistic algorithm for testing primality, in: Journal of Number Theory 12(1), 128–138.
34. Mainzer, K. (2020), Quantencomputer. Von er Quantenwelt zur Künstlichen Intelligenz, Springer: Berlin.

Conceptual Limitations

4

4.1 The Question "why?"

Why?

Question of a child

If the Turing test would be used as benchmark for the successful implementation of artificial intelligence, statistics-based AI is in a dilemma—at least as long as it is still operating in the in black box mode. This is because one only need to follow up on a question, which can be answered as well as possible by the AI, with the next question: "Why?"

Humans answer a why questions usually by an argument; such an argument can be placed in a conceptual framework. Statistics-based AI, however, does not have, per se, a conceptual system available from which the answer could be derived. So if it—correctly—answers every why-question with a succinct "I have learned it this way", it would immediately fail the Turing test, since one will not accept this (continuous) answer from a human being.

That arguments could be statistically learned from a lot of answers to why-questions appears to be impossible. On the one hand, one can resort to mathematical examples where specific discrete properties, e.g. in number theory, seem to exclude a statistical guessing of the correct calculation. On the other hand,

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical*, Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6_4

arguments for answers from everyday questions cannot be convincingly generated by statistical extrapolation of answers to similar questions.[1]

Why questions also pose a dilemma for statistical AI, when one asks whether the corresponding answers should be correct in so far as they should reflect the actual decision making used by the software. In this case, the software would have to disclose its black box. On the one hand, this should not be possible—due to the very definition of black box; on the other hand, such an answer would certainly be distinguishable from an answer given by a human being. However, if the answer does not correlate with the internal decision making, but is, e.g., statistically extrapolated from the way other why-questions were answered, it is not clear why such an answer has to fit to the previously given answers at all.

From a conceptual point of view, why questions are only one particularly impressive example of reflexive considerations about one's own thinking, which at least human intelligence is capable of. Questions like "How do you know that?", "Since when do you know that?", "Do you know that for sure?", etc., which reflect our knowledge, will not be answered based on purely statistically learned answer schemes.

Interestingly, in some narrowly defined contexts, symbolic AI can answer Why questions, because it derives its answers within a given conceptual framework—and the resulting derivation serves as justification. This leads to a further argument, why artificial intelligence, as far as it aims to meet requirements of a Turing test, requires a connection of statistical methods and knowledge-based systems in a hybrid AI.

[1] It is something else that AI can, of course, "memorize" an argument if the very same question was already answered in the learning data; ChatGPT makes use of this possibility, when it recourse to its enormous learning data.

4.2 Can AI "Remember"?

> "What do I care about my chatter from yesterday. Nothing prevents me from becoming wiser."
>
> Attributed to Konrad Adenauer (German Chancellor, 1949–1963)

It is first of all a purely technical question whether a trained machine learning program will always give the same for the same question; in this case the program has the mathematical property of functionality.

At this point it is worth taking a look back at fuzzy logic, a supposedly revolutionary branch of logic that, in the context of symbolic AI, was preparing to revolutionize formal logic and with it also computer programming. One broke away from the usual two truth values true and false and calculated with fuzzy values ("rather", "somewhat", etc.). Especially propagated by Japanese industrial companies, this fuzzy logic was able to enter the consumer goods business at the end of the twentieth century. However, it is was not that your "fuzzy logic washing machine" worked on base of fuzzy logic; fuzziness was only used for the modeling of the modeling of the washing machine control system. The later installed control system was a deterministic algorithm "like any other". In the long run it turned out that the conceptual basis of fuzzy logic does not have any special theoretical added value, and the field is today, with less publicity, but scientifically more solid, only as a special discipline within the framework of the non-classical logics.

As far as statistical AI delivers functional, i.e. deterministic, programs, it follows the principle of fuzzy logic, in the sense that the statistical methods are *only* used in the construction ("learning") of the program. In the application, however, we would have to do it again with a deterministic algorithm. The difference to a traditionally implemented algorithm (e.g., in the programming language Java) would essentially be that the internal processes of the algorithm would not be accessible to

us—due to the black box technology.[2] This complicates, if not makes impossible, the investigation of formal properties, e.g., correctness with respect to a given specification; but at least one can "rely" on the results—in the sense of functionality. Areas where deterministic algorithms exist (or are easy to create), are from this perspective not specific fields of application for statistical AI.

However, if a program created by machine learning should not be functional, it faces a problem, if one—in the sense of the last section—asks: "Why did you change your answer?" Would the program even have the possibility to know that it changed its answer? Thus, one could also ask the question the question: "What did you give me as an answer yesterday?" It is obvious that answers to such questions cannot be learned statistically.

If results are not deterministic, the user has to expect to get different answers to the same questions. Of course, this is only acceptable if such different answers have no particular effect in the given context. As a rule, this is the case for questions only, whose possible answers are continuously connected to each other, i.e. different answers are sufficiently close to each other. In this case, we see again the weakness of statistical AI for discrete questions.

To the extent that one wants to value changes in answers as a positive feature of learning, the question arises how to deal with earlier answer—from a more recent perspective wrong or stupid answers. This question is quite analogous to the evaluation of a misbehavior of an inexperienced person. Conceptually, it poses a challenge as one would have to be able to transform experience into a number of data to be learned. It is a saying that men never stop learning. This may also apply to AI, as it is constantly

[2] This disadvantage is accepted, on the one hand, because the program creation in the AI learning process is faster and, above all, is done by the computer itself and no longer requires a trained computer scientist. On the other hand, it is possible to tackle problems whose external complexity does not allow direct programming.

confronted with qualitatively new situations. But such new situations may also make a "learned AI program" immediately obsolete.

4.3 Can Programming Be Automated?

Calculemus!

Gottfried Wilhelm von Leibniz

Programming requires extensive knowledge, experience and a high degree of creativity. Programmes are written by human written by people. But can machines also write programmes? Can this kind of creativity be automated? Programming can be understood as a process that transforms a problem definition as the user's intention into a sequence of instructions and, when the instructions are executed on a computer, produces a solution to the problem. Over time, a programme needs to be maintained as it evolves to changing programme goals, errors in the programme, and properties in new computer platforms. Automated programming (machine programming) is a system that can perform some or all of the steps to transform a user's intention into an executable programme and its programme and its maintenance [1]. In this sense the creativity of a programmer can be automated. It is no less than computers programming computers, and software writing software.

In view of the complexity of programmes today, they are written by teams. Each team works on a precisely defined goal, the results of which are put together on a platform. Programmers make use of prefabricated programme modules, which are compiled in digital libraries.

From the 5000 source code databases (repositories) one and a half decades ago have now become more than 200 million. Automated programming (machine programming) aims to automate these steps. It is not simply about systems that control themselves and setting, as is already known from control and regulation technology. Rather, software should create its own software.

Automated programming would be used in the technology of self-controlling automobiles of levels four and five with fully automatic and autonomous driving. This requires that a system independently recognises that something is wrong and that an accident could occur. An important step on the way to machine programming would be software that could write error-free programmes. In 2020, the company Intel introduced ControlFlag, a system that is supposed to detect errors in the source code. The basic goal here is that a source code by methods of machine learning (ML), i.e. AI, is created.

However, ControlFlag should not initially develop new code, but only detect errors in existing code. For this purpose, methods of non-supervised learning are used. In this process, the system (e.g. a neural network) analyses data which, in contrast to supervised learning, is not based on the input and prior training by a human. Rather, the system recognises the "standard case" in a large amount of data based on similarities and correlations and deviating cases deviating anomalies and outliers (bugs). For example, ControlFlag has identified one billion unlabelled program lines of standard quality code and learned corresponding normal patterns. Unlike software for static code analysis, the system does not look for specific vulnerabilities such as memory allocation errors, but rather identifies anomalies independent of the used programming language.

For the detection of deviations in patterns (e.g. programmation errors), the tool MISIM (Machine Inferred Code Similarity) is used, which detects similarities in the code. Through structural comparison MISIM automatically detects which purpose a part of a code serves. The structure of the entire code is determined automatically by checking the pieces of code for syntactical similarities and differences to other codes with similar behaviour.

Automated programming (machine programming) can already recognise from parts the intention that an algorithm is pursuing. For Intel, this is based on a concrete business model: MISIM is intended to offer software developers in complex environments automated code suggestions that fit into a software architecture or can be used to solve a problem in an existing code.

From an epistemological point of view, automated programming with MISIM is a first step in the direction of a software that recognises intentions and problem situations in order to independently suggest solutions independently and in this sense to be intelligent and creative.

In contrast to other code code-similarity programmes, MISIM has a context-aware semantic structure (CASS) that users can configure for specific contexts.

The code-similarity programme does not require a compiler to convert human-readable code into computer-executable code. The system can also execute incomplete code parts and suggest additions to solve problems. In the process, neuronal networks evaluate the code parts according to similarity.

In practical terms, this means software that makes fewer errors, learns from itself and immediately implements what it has learned. Machine programming is therefore definitely proving to be a boost in the economy. Today, software is built into almost all electronic devices. Therefore, every user should be able to develop their own suitable software without having to write a line of programming code. Like a human interpreter translates, Machine programming translates human language into the language of machines.

Will machine programming make human programmers become superfluous? Yes, to a certain extent: the search for programming errors and simple programming could be done by these self-programming machines themselves. Programming people could then concentrate on more demanding tasks. Where previously hundreds of thousands of lines of code were necessary, machine programming codes with just a few hundred lines of programming.

In summary, automated programming is based on three pillars:

1. Intention is the ability of the machine to understand the goals of the programmer.
2. Invention is the machine's ability to discover methods for realising these goals.
3. Adaptation is the machine's ability to maintain this software autonomously.

As already explained, the intention of a programmer can be detected by non-supervised learning from parts of a code [2]. When inventing a program to achieve these, in addition to neural networks and machine learning, methods of program synthesis can also be used. In program synthesis, an invention is conceived as a search problem. The search space comprises programmes as solution candidates. The goal is to find a program that satisfies certain constraints of the constraints of the desired behaviour. In this case, it is necessary to determine how the search space is to be represented, how it is to be efficiently constructed using the semantics of the underlying building blocks and how the constraints and side conditions are to be understood.

4.4 Can Proving Be Automated?

> A good proof can be read as a poem—this one looks like a phone book!
>
> Comment on the computer assisted proof of the Four Color Theorem

A classical example of automatic proof is formal verification with SAT. Automatic proof goes back to the beginnings of symbolic AI and is based entirely on formal (symbolic) logic. In logic, the satisfiability problem (SAT) concerns the question of whether there is an interpretation that satisfies a given Boolean formula. In this case, the formula is called satisfiable. If such an assignment does not exist, the function expressed by the formula is false for all possible assignments of variables and the formula is unsatisfiable. According to Cook's theorem, SAT is NP-complete [3]. There is no known algorithm that efficiently solves every SAT problem. But heuristic SAT algorithms can at least be applied, to solve restricted problem classes with thousands of variables and formulas [4].

The question arises as to whether SAT methods can also be applied to machine learning with neural networks. In this case, one would have to find neural network models that are constrained by boundary and side constraints. Neural network nodes

of neural networks but often do not have linear input-output behaviour. However, they can be approximated linearly. For this purpose, the weighted sum of the input signals to the nodes is denoted as variable c. Let variable d be the output of a node. If there are upper and lower bounds $[l, u]$ of c, the relationship between c and d can be approximated by boundary conditions such as $d \geq 0, d \geq c$, and $d \geq \frac{u(c-1)}{u-1}$. Obviously, these boundary conditions are represented by linear equations for constants l and u [5].

In the next step, the boundary constraints are represented by Boolean formulas. The idea is to combine a linear program solver and a SAT solver. The SAT solver is to check whether these Boolean formulas have a satisfiability assignment. For this purpose they are transformed into a conjunctive normal form (CNF), which consists of clauses connected by conjunctions (disjunctions of literals).

A SAT solver works in such a way that it successively evaluates the Boolean variables. Backtracking always occurs whenever a conflict is found between the current evaluation and a clause. SAT solvers can be extended by various learning heuristics. An example are SMT (Satisfiability Modulo Theory) solvers, which combine SAT solvers with specialised decision procedures for other theories.

The verification of feed-forward neural networks can be formally realised in following steps: Given a feed-forward neural network G with a function $f : \mathbb{R}^n \to \mathbb{R}^m$, which is defined by a set of linear constraints ϕ over the real variables $V = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$. The verification problem of G is either to find a valuation function α for the variables from V which satisfies ϕ over the input and output nodes of G with $f(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$, or to show that no such valuation function exists for the nodes.

In the case of a linear programme solver (LP), one starts with a given set of linear inequalities over real variables and a linear optimisation function (linear programme). The verification problem of linear programming consists in finding an assignment to the variables that minimises the objective function and satisfies

all constraints. A more advanced instrument is the combination of a linear programming solver (LP) and a satisfiability solver (SAT, SMT).

A proof assistant or interactive theorem prover (interactive theorem prover) is a software tool that supports the process of formal proving by human-machine interaction. In computer programs data types are used to reduce software errors (bugs). The type theory CoC (Calculus of Construction) is the basis for the proof assistant Coq [6]. Coq implements a programme specification, which is based on an extension of CoC, the calculus of inductive constructions (CiC) and combines a higher-order logic with a richly typed functional language [7].

The instructions of Coq allow,

- to define functions or predicates (which can be evaluated efficiently).
- to assert mathematical theorems and software specifications.
- to develop formal proofs of these theorems interactively
- to machine-check and certify these proofs
- to extract certified programs.

Coq provides interactive proof methods and decision algorithms. Connections with external theorem provers are also accessible. Therefore, Coq is a platform for both the verification of mathematical proofs as well as for the verification of computer programs in CiC [8].

In Coq, the verification of proofs is reduced to the verification of types in type theories like CiC. The core of Coq is therefore a proof algorithm for types in the language of CiC. Further details of Coq are given in various tutorials. Finally, Coq and CiC have been used in advanced difficult proofs (e.g. the four-colour theorem [9]).

The extraction of certificated programs works with recursive schemes of terminating algorithms. The extraction of programs requires that a faithful Coq version of the target program in functional language is embedded in Coq. Correctness properties of the Coq version of the target program can then be proven and a functional program version automatically extract a functional

programme version. If the automatic extraction is secured, the resulting functional program fulfils the expected correctness properties.

A well-known application of symbolic AI since its beginnings has been first-order unification. This is an algorithmic procedure in which equations between symbolic expressions are solved. A solution to a unification problem is a substitution that assigns a symbolic value to each variable of the formal expression of the problem description. A unification algorithm calculates a complete and minimal set of substitutions, which includes all solutions and contains no redundant elements. Syntactic unification of 1st order is fundamental to the resolution method of 1st level logic. In "automated reasoning" this procedure helps, to cope with the combinatorial explosion that occurs in the search for instantiation of terms.

The 1st order unification can be fully formalised in inductive type theory. Automated reasoning is also a major application of unification today. Thus SAT methods with Boolean logic are used in the automotive industry, aviation and in rail transport in the handling of logistical problems.

These examples involve highly safety-critical applications. Because of their complexity, these optimised SAT solvers are not amenable to direct formal proofs of correctness. However, Boolean SAT solvers can be embedded in the proof assistant Coq. Their certification procedure does not depend on a specific SAT solver, but can be applied to any SAT solver that can be formalised in Coq. As usual, the program code of the proof checker can be extracted with Coq.

It is noteworthy that practical software procedures of industry and technology can in principle be certified by proof assistants like Coq. At the same time, however, these proof assistants are deeply rooted in the fundamentals of logic and mathematics. Therefore, they can also be applied to the increasingly complex proofs in mathematics. Some proofs with an immense number of case distinctions, such as, e.g., the four-colour problem can only be tested with increasing computing power. However, in today's research, a much more fundamental problem arises than just the size of individual proofs:

Mathematical disciplines have become so highly specialised with difficult methods that they often require a lifetime of training. Mathematical arguments are occasionally so complicated that a single mathematician can hardly examine them in detail. Therefore, they rely on the competence of their colleagues, who are recognised by the scientific community in their respective fields of research. This gives rise to considerable risks of error.

The Russian mathematician and Fields Medal winner Vladimir Voevodsky (1966–2017) reported one of these experiences. In his research, he had combined the highly abstract fields of algebraic geometry and algebraic topology together in order to prove central mathematical conjectures (Milnor and Bloch-Kato conjectures). In the process, errors in his theorems were overlooked and remained undiscovered for years. After the discovery of these errors, one was alarmed, because it could not be ruled out that other errors were hidden in the difficult proofs.

Voevodsky was deeply concerned about this. Who, he asked himself, could ensure that something had not been forgotten, that a mistake has been made, if even errors in rather simple arguments remain undiscovered for years ["*Who would ensure that I did not forget something and did not make a mistake, if even the mistakes in such more simple arguments take years to uncover*?"] [10]. This experience with mistakes, which had been overlooked for years by the scientific community doubts as to whether the usual division of labour of experts in the scientific of experts in the scientific community can be trusted in the future. Voevodsky became increasingly convinced that the human mind cannot cope with the increasing complexity of mathematical problems and would be overtaxed.

This raises the question: Are we ultimately dependent on the support of computers as the only way to solve problems even in mathematics? Voevodsky therefore proposed his basic programme for univalent mathematics, in which software for the verification of proofs is at the centre to promote confidence and security in mathematics.

The situation is reminiscent of the mathematical foundation crisis of a hundred years ago [11]. To avoid contradictions in Cantor's set theory, Bertrand Russell had introduced his type theory [12]. Objects were not defined as sets, but as types with

great similarity to data types in programming languages. In programming languages data types are necessary to avoid programming errors. In fact, type theories prove to be a bridge between mathematics and computer science. Meanwhile, not only data structures in computer science, but also complex mathematical structures such as algebras and topologies can be characterised by type theories. The result is that the theorems and proofs can be automatically checked for correctness by proof assistants along the lines of Coq.

The issue here is not whether Voevodsky's univalent mathematics already provides the ideal methods. Rather, the development of mathematics could come up against the limits of the capacity of human brains, which can only be overcome with the support of powerful instruments such as computers. It is not only human muscle power that has been strengthened by technology and eventually led to motorisation and automation, but also human brainpower through computerisation.

What is new is that proof assistants work interactively, i.e. they develop proofs in the interplay of humans and computers. Are we thus entering a new phase of mathematical thinking, in which mathematical theorems and their proofs are developed in a symbiosis of man and machine? Huge databases store routine proofs and patterns of thought, which are combined in new and appropriate ways. The machine discovers patterns of "normal cases", but also deviations and tests possibilities. What would that be other than the first steps towards mathematical creativity?

At any rate, it would be the transition from symbolic AI, as cultivated in mathematics through automatic proof, to a hybrid human-machine intelligence, which opens up new potentials and transcends old boundaries of both machines and human creativity.

4.5 "What You Give is What You Get"?

> The Analytical Engine has no pretensions whatever to originate anything. It can whatever we know how to order it to perform.
>
> Ada Lovelace

We had already pointed out the problem of extrapolation, i.e., the problem that can arise in machine learning when it has to deal with data that lie outside the domain from which the training data came from. Conceptually, the situation is even more problematic when it comes to features that do not have a statistical relationship to the features that act as input parameters.

For example, address, gender, nationality, and skin color can be recorded together with a given selection, in order to statistically check whether this selection was discriminatory. With the help of these data, however, no statistics can help to find out that the selection systematically discriminated, e.g., left-handed people.

At the end of the day, this problem boils down to the fact that machine learning only detects correlations that exist between the given features. Scientific progress, however, often takes place where correlations are detected which relate to previously ignored features,

Thus, on a conceptual level, machine learning can only take into account the features that have been explicitly given to it; in short:

> What You Get Is What You Give.

How little the data themselves sometimes justify their inner structure, is shown by an example described by the physicist Max Born in his memoirs [13, p. 55]:

> Breslau obtained an excellent meridian instrument and a parallactical telescope. Yet no proper building was provided, and the previous instruments were installed in two wooden huts on a narrow island in the Oder river … The time service for the province of Silesia … was transferred to the new Zeiss instrument, but the results were rather unsatisfactory.

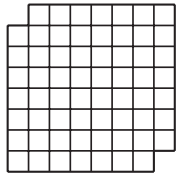
As nice as the idea may be to find out the reason for the deviations by machine learning based on the captured data, such an AI is unlikely to be successful:

> Dr Lachmann was charged with finding out the reason, and he soon discovered a correlation between the strange deviations of the time observations and the changing level of the water in the lock: the island was not proof against water pressure.

This example involved information that was outside of the data, but which can be added by taking into account comprehensive "knowledge" of the environment. A classic example that requires real creativity is the *Artin board*[3]:

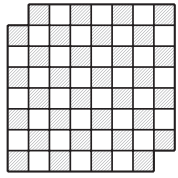
Example

Consider a board with a 8×8 grid on it, dividing it into 64 squares; now remove the two opposite squares from the corners so that only 62 squares remain:



Is it possible to tile 31 dominoes  on this board so that all squares are covered?

The answer is easy to see when we put some extra structure on the board, namely the usual black and white alternation of a chess board; a mutilated chess board looks like this:



[3] Christian Thiel [14] discusses Artin's board as a paradigmatic example of creativity. On the notion of structure that comes into play here, see [15].

> As we removed two white squares, the mutilated chess board has 32 black squares but only 30 white squares; but each domino covers exactly one white and one black square, so that we cannot tile a board which doesn't have an equal number of white and black squares.[4] ◀

Of course, with the help of brute-force methods AI should be able to find out (very fast) that the desired covering does not exist; but it is not to see that it could draw on the idea to use the chessboard painting and, thus, solving the problem also independent of the size of the board.

In general, there are limits to machine learning when we need to recognize or even just have to take into account-parameters which are not explicitly included in the data sets. At best, artificial intelligence can indicate that an explanation for a conspicuous pattern in the data - a pattern at least conspicuous for the AI - requires an additional feature to explain this pattern. The creativity to suggest even one for it is left to humans.

The great successes that machine learning can show in such cases where the relevant information is actually hidden in large amounts of data - facial recognition or the above-mentioned folding of proteins - are examples will not be generalizable if relevant information is not already part of the data under consideration. Here, we see a conceptual boundary, which cannot be overcome by advances only *within* machine learning.

[4] Historically, the example of the mutilated chessboard can be traced back to Max Black who posed it in 1946 as a problem in his book Critical Thinking [16, exercise 6, p. 142] (but starting off with the chess board, thus, leaving out the creative part of adding this structure as a first step). It is also reported that Emil Artin occasionally used this example in his lectures (see [15, 17]); it might well be that he took it from Black (or some other later source), but it was stressed in a obituary for Artin that he applied the idea of the solution within his mathematical activity, as he had "the very rare ability to detect, in seemingly highly complex issues, simple and transparent structures" [18, p. 39].

4.6 Background Theory

> Stella Veneris, quae Φωσφόρος Graece, Latine dicitur Lucifer, cum antegreditur solem, cum subsequitur autem Hesperos.
>
> Cicero, De Natura Deorum 2, 20, 53

We had already pointed out in the Riemann conjecture, and especially in the discussion of cryptographic protocols that mathematical conjectures and results are made and proven in the context of a given background theory. In mathematics, the respective background theory is given as an axiomatic system, and the mathematician David Hilbert sees in the axiomatic method a goal of every further science [19]:

> Everything that can be object of scientific thinking in general, as soon as it is ripe for formation of a theory, runs into the axiomatic method[.]

The "framework of concepts" [19] formed in this process is, of course, not obtained on a statistical basis, but through conceptual analysis, which leads to implicit definitions of the used terms and general laws for them.

In principle, the axiomatic method is also the leitmotif of the expert systems. It should be noted, however, that these systems do not generate their own "framework of concepts", but the concepts are supplied by a programmer. What can be proven within a given system of axioms is subject to the well-known restrictions in computational complexity.

But independently of this, the expert systems - as well as scientific theories in general - reach their limits, if they would have to draw on general world knowledge. And this limit is even more restrictive in statistical learning.

Machine learning is already said to be able to generate jokes. But in what form should machine learning be able to determine whether a joke generated by it has crossed the line of good taste? *Jewish jokes* and *jokes about Jews* are dangerously close expressions. Nevertheless, a not completely dump person should be able to recognize the difference immediately due to his world knowledge.

If one were to commit oneself to the highly dubious task of to try to teach the said difference to an AI by means of a negative classification of a large numbers of jokes about Jews, it is noticeable that one can teach the difference, for instance, to a naive child by a single example. The reason lies in the explanation, which can refer to background knowledge which refers to terms such as “hurtful” and “contemptuous” - not to mention the historical context.

Another interesting challenge is posed for machine learning by counterfactuals. Here, one tries to obtain conclusions within a counterfactual thought experiment. There should be no problem, as long as only one feature in a query is changed. One can run statistics-based human resources software, of course, to check the following situation: “What would be the result if the applicant were a woman?” But in more complex scenarios, at least two problems arise.

On the one hand, changes in the deep structure of the argumentation structure cannot be carried out directly, precisely because of the inaccessibility of the theory, which would make this structure manageable. On the basis of recorded positions, the position of Neptune in the sky can be predicted by machine learning; but using only these data, and without any further theory, certainly the following question cannot be answered: “Where would Neptune be seen if Uranus did not exist?”.

On the other hand, changes in the situation underlying the learned data cannot be mapped, because no (learning) data are available for this purpose. Consider, for example, the extrapolation of fall experiments on the basis of highly sensitive measure. Now ask: How would the body fall if we neglected friction?

Another example where a background theory can enter, is the phenomenon of creativity. Here some quite impressive successes are attributed to modern AI, e.g. in creative Go strategies or in the composition pieces of music. The creativity that comes to light here is by construction bound to the learned examples. A “revolutionary” idea, which can be based on a negation established rules - as it is realized for example in Schönberg’s twelve-tone music - will, however, not be invented by machine learning.

It is precisely the knowledge of the prevailing theory that makes it possible to propose a radical alternative.

Finally, we want to address a question where the theory is not in the background but, strictly speaking, in the foreground. Based on an extensive collection of data on the position of Venus in the morning sky one should be able to predict with machine learning when and where to see Venus next time in the morning sky; the same is true for Venus in the evening sky.

But in what form could an artificial intelligence independently correlate the data of the morning sky with those of the evening sky to find out that they are one and the same planet? The human achievement to develop a planetary system, in which circular orbits are introduced into the theory, which hypostasize planetary positions also below the horizon, and which recognizes the morning star and the evening star as being on the same (approximated) circular orbit, cannot be developed by the AI alone. This is because the pure observation data do not provide any reference to a superordinate (planet) theory. One can, of course, try to ask to match the data with a circular orbit. But this would give the actual progress in knowledge, which was present in this identification, to the AI, instead of receiving it from it.

4.7 Ethical and Societal Limitation of AI

> Human dignity shall be inviolable.
>
> German Constitution Art. 1 (1)

In philosophy, a distinction is made between the limits of knowledge and the limits of moral-ethical action. Kant assumed that the limits of knowledge are determined by the categories of understanding. These are forms of judgement that largely correspond to the axioms of Newtonian physics of the time. Examples are categories that describe objects with their properties, causal sequences of events and their interactions in space and time. Boundary questions that go beyond these categories were described as categorically undecidable. For Kant, these include

questions such as whether the universe has a causal beginning and end or is unlimited, but also the question of whether there is an "immortal soul" or a God. In modern times, these limitations of undecidable questions remind us of Gödel's incompleteness theorems, which define the limits of logical calculi for logically undecidable questions. In computer science, limits to the performance of algorithms are distinguished in complexity theory. In contrast to Kant's categorical limits of cognition, limits in computer science are relativised to the presupposed classes of algorithms.

The limits of moral-ethical action are, in Kant's case, likewise by a logical form of judgement, in this case of the categorical imperative as a universal requirement of action [20, Ch. 12.2]. In simplified terms, the scope for action ("freedom") of each individual is limited by the scope of action ("freedom") of the of the others in a society. Therefore, every intention and rule of action (according to Kant, a "maxim of action") of an individual must be a law and must be generalisable as a law and regulation for the whole of society [21]. To put it bluntly, I must therefore always behave in such a way that my maxim of an individual action as a law, e.g. in a democratic parliament for all citizens, could be adopted for all citizens. As a general requirement for any action, this is very strong and emphasises the formal rigorism in Kant's ethics. Just as Kant's categorial epistemology was oriented towards (then) physics, his ethics should define the framework of civic legislation in law.

All ethical and legal regulations were to be justified by this ethical framework. These regulations thus limit the scope of action of every citizen of a society. After the epistemological limits of computer science and AI have been discussed in the previous sections, we now turn to the ethical and societal limits of AI. The following analyses will show whether today's regulations for the limits of AI application can be traced back to Kant's categorical imperative.

As an example of ethical-legal boundaries, the former EU Directive on AI 2021 will be considered [22]. The legal framework for AI proposed by the EU Commission is intended to define the fundamental rights and the security of users. The aim

is to increase the trust and the dissemination of AI. From the EU's point of view AI systems promote its economic growth, its innovative strength and its global competitiveness. However, this also entails new risks with regard to the security of users and their fundamental rights ("civil liberties"). The legal framework covers both providers and developers as well as users of AI systems.

The risk classification is based on the intended purpose of the AI system in relation to the EU product safety regulations. Therefore, the classification of the risk depends on the function of the AI system, its specific purpose and its conditions of use. Added to this is the number of persons likely to be affected, the dependence on the outcome, and the irreversibility of any damage.

The EU Commission proposes a risk-based legal framework with four risk levels.

Unacceptable risk: Some very harmful AI applications that violate fundamental rights and therefore fundamental rights and thus violate EU values, will be banned. This concerns, first of all, the evaluation of social behaviour by authorities, known in China as the 'social score'. In this way, the EU is clearly setting itself apart from its Chinese competitor. Likewise, the exploitation of the vulnerability of children is also prohibited. The use of AI techniques to unconsciously influence users is also banned. This is a very strong demand, as it interferes with the area of advertising in free markets. Similarly, biometric real-time remote identification systems will be severely restricted if they are used for law enforcement purposes in the public domain.

High risk: In this case, the AI systems concerned have a negative impact on the security and fundamental rights protected by the EU. The EU proposal is accompanied by a list of high-risk AI systems. This also includes security components of products, covered by EU sectoral legislation. According to these sectoral legislation, they must be subjected to a conformity assessment by a third party. For all high-risk AI systems, binding regulations are required which secure the quality of the data sets used, the documentation, transparency, provision of information to users, robustness, accuracy and cybersecurity. In the event of

infringements, national authorities will have access to information that will allow them to assess the legitimacy of the AI. This legal framework is in line with EU Charter of Fundamental Rights and the EU's international trade obligations.

Low risk: In this case, special transparency obligations are imposed if, for example, there is a risk of manipulation. In this case it should be clear to the users that they are cooperating with a machine.

Minimal risk: All other AI systems can be developed and used in compliance with generally and used in compliance with generally applicable law. However, providers can offer voluntary codes of conduct to ensure the trustworthiness of the AI systems.

A list of critical application areas includes the fields of biometric identification, critical infrastructure, education and training, recruitment and employment, provision of essential public and private services, law enforcement, justice, asylum and migration.

Biometric identification can be used for user authentication. Technically, it is based on machine learning for pattern recognition. Examples are the unlocking of a smartphone or the verification for border crossings. But remote identification is also possible, e.g. to identify a person in a crowd by comparing him or her with the database. Biometric systems can also refer to, for example, gait and language. The quality and accuracy of the identification depend on, e.g., camera quality, light, distance, database, algorithm, ethnic origin, age or gender of the persons. They are constantly being improved. But even a small error rate of 0.1% is high for tens of thousands of people.

When AI systems are classified as high-risk, obligations arise for the providers. Companies that put high-risk AI systems on the market must submit to a conformity assessment. In doing so, they must prove that their systems meet the prescribed criteria of a trustworthy system.

Kant already emphasised that ethical standards alone are not enough, but must be enforceable as a law. Normative limits must therefore be linked to state sanctions in the event of transgressions. In the EU, this is the responsibility of the member states

with their national authorities. In the European regulation, the sanctions of companies are defined for certain threshold values, which relate to the classification of risks.

Up to 30 million euros or 6% of the total worldwide turnover of the preceding year are provided for in the event of violations due to prohibited practices or data requirements.

Up to 20 million euros or 4% of the total worldwide turnover for the previous year are provided for violations of other orders and obligations under the regulation.

Up to 10 million euros or 2% of the total worldwide turnover of the preceding year shall be imposed in the event of false, incomplete or misleading statements in the case of information provided to the competent national authorities.

A central concern of the EU regulations is to prevent bias by AI systems on the basis of racial discrimination or gender. However, EU regulations should not become a killer of innovation. Especially, the new EU AI-regulations which were accepted by the EU Parliament in 2023 do no longer consider the potential of AI, but tend to overregulation under the impresion of ChatGPT. Therefore, lean administrative structures should be implemented. They should also only be used when it is absolutely necessary and impose as little bureaucratic burden as possible on the economic participants. In principle, greater confidence on the part of users promotes the demand for AI. Increased legal certainty with uniform regulations opens up larger markets for their products for European providers.

At the national level, the German AI Steering Group of Standardisation was set up in 2019 to promote confidence and legal certainty in AI by standardising and certifying AI software [23]. Clear limits for the use of technical devices are set in the German tradition by DIN (Deutsches Institut für Normierung) standards. However, mathematical correctness and technical safety are not sufficient. Ecological, economic, social, legal and ethical criteria must be taken into account in AI standards. These criteria are therefore based on ethical, legal, social, economic and ecological limits of the application of AI.

The German Steering Group (HLG) therefore comprised members from the following areas of science, economics, politics and civil society (e.g., German Research Center for AI (DFKI), Platform for Learning Systems, acatech, IBM, Siemens, Federal Ministries, German Parliament), which is organised by the Federal Ministery of Economy (BMWi) together with the German Institute for Standardisation (DIN). Their task was to structure a roadmap and a future strategy for the standardisation of artificial intelligence. Proposals were to be developed that would provide orientation for the German (DIN) orientation for the certification of AI software. The legislature should also be given the opportunity to make recommendations for legislative resolutions.

DIN standards at national level are not sufficient for an international technology such as AI. For this reason, in addition to DIN, DKE (German Commission for Elektrotechnology, Elektronics, Information technology) and VDE (Society of Electrical Engineers) at national level, CEN (European Committee of Standardisation), CENELEC (European Committee of Electrotechnical Standardization) and ETSI (European Institute of Telecommunication Standardization) are also involved. The international umbrella organisations are ISO (International Standardization Organization) and IEC (Electrotechnical Organization).

The boundaries were drawn in seven working groups under the aspects of

- fundamentals (data, terminology, classification, AI elements)
- ethics/responsibility
- safety
- quality and certification
- mobility and logistics
- industrial automation
- medicine.

The Commission was aware that these boundaries of AI could steer the development of a society in the application fields of economy and and infrastructure.

Example

A risk-adapted regulatory system for AI is to be illustrated using the concrete example of a criticality pyramid. This is an application of machine learning for automatic translation. To measure the accuracy of a translation, the BLEU (bilingual evaluation understudy) value is used, which automatically compares machine-translated texts with texts that are translated by human experts. The BLUE value on a scale of up to 100% does not capture the syntactic and semantic correctness with its very simplified metric, which is limited to comparisons of word sequences. But the BLUE value proves useful as a first rough estimate between automatic translation systems. The standardisation of quality metrics for AI systems is very important for a broad acceptance of these systems in practice. Since 2006, the BLUE factor has been specialised in application domains, in order to increase its accuracy. For example, a BLUE value of 15 is very poor, since it requires a great deal of post-processing of the automatic translation in order to continue the translation work.

For a risk-adapted certification of AI systems, quality thresholds must be defined for application classes, below which the results of the AI systems can no longer be used in a critical area. If, for example, a witness statement is available in a foreign language, its transcription can no longer be used in court with too high an error rate. If, for example, a witness statement is in a foreign language, its translation cannot be used in court with too high an error rate. In this case, the document must be given to a human translator instead (cf. Fig. 4.1: red peak in the pyramid of criticality). Only certified texts should be used for operating instructions for technical devices. For doctors' letters and contractual texts, the quality of the translation must be checked on an ongoing basis. In the case of public tweets, blogs or news portals, an expost check should be carried out. For private chats automatic translation with a low BLUE value may well be useful, since a very low risk for the user and the advantage of a fast translation outweighs the risk. ◀

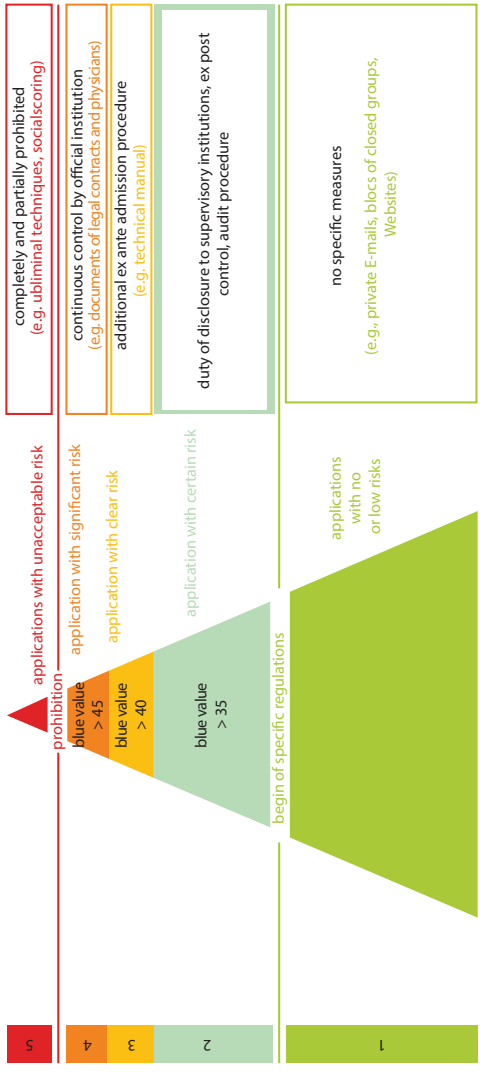


Fig. 4.1 Limitation of usability for AI systems (criticality pyramid) [23, p. 44]

In summary: The steering group worked with science, business, politics and civil society, industry, politics and civil society to develop standards for AI technology (DIN standards) and legislative initiatives (e.g. Ministries, German Parliament). In this way, it regulates the current ethical-legal limits of AI. It therefore serves to advise and coordinate. AI standardisation should, however, not only be seen as a control and regulatory task (danger of over-regulation, bureaucratisation and a brake on innovation, but rather as a reinforcement of sustainableand responsible innovation.

To conclude the discussion of ethical-legal barriers, their international dimension should be examined. It is not enough to develop European standards that are not accepted internationally. In the USA, ethical-legal limits of AI are pragmatically combined with the possibilities of business and capital (e.g. Silicon Valley). Global IT and AI companies are dramatically changing the world of life and work. Human progress is measured by successful business models. Ethnic and gender discrimination are clear limits to this.

In contrast, China relies on state monopolism with a controlled economic system. Boundaries are set by the Communist Party for all areas of life. AI is the spearhead of the technical, economic, military and political challenges of China as a world power. In strategic plans up to 2050, the steps towards innovation are centrally determined by the state party and realised with "ruled" capitalism. A "ruled" civil society controls itself via a social score for each individual citizen [24]. This total data collection of each individual is by no means regarded as a horror vision of a "Big Brother", as it is in the West, but rather, in the opinion of a large majority of the Chinese people, it ensures the superiority of its own system in crisis management. The ethical-legal limits of AI application should then be determined by the state and implemented directly to increase the efficiency of its own system. In the Confucian tradition, the collective good is at the top of the hierarchy of values.

The boundaries of AI in Europe are fundamentally determined by human rights and parliamentary democracy. Fundamental to individual human rights are the autonomy rights of

each individual. A technical system that adheres to pre-programmed moral rules, is not itself "moral", however. Even in the case of learning algorithms we are dealing at best with AI systems that can be compared with trained animals or small children. Autonomy in the rights of political freedom, however, means a higher level: autonomous is the human being who is self-determined in every respect and capable of self-legislation.

As already explained above, according to Kant, an action is only morally justified if it can be the basis of a general legislation. This is the core idea of his Categorical Imperative: The rule (Kant: "maxim") of my action must be generalisable. My right ends where the right of others begins. If I encroach on the other person's freedom, this action is not capable of generalisation. It would inevitably lead to the war of all against all. The maxim of my action must therefore, in principle, be the basis of a general law which, for example, a democratic Parliament would pass. In this sense, autonomy means the ability to "self-legislate".

From a technical point of view, it cannot be ruled out that an artificial intelligence will one day also be capable of "self-legislation", i.e. it will give itself its laws as programs: It programs itself! In the separation of powers of Western democracies, legislation is the right of parliament (i.e. legislative power), elected by the people of a country in free elections.

At this point at the latest, the question of responsibility arises in the age of digitalisation and artificial intelligence. The concept of responsibility has a long legal and philosophical tradition. Responsibility is generally understood to be the duty of an acting person (or group of persons) on the basis of a claim made by an authority (e.g. institution, state, society).

The first distinguishing criteria are, for example, causal responsibility with a view to causation (e.g. programming error of a programmer), role responsibility with regard to a task (e.g. a teacher for his class), abilities (e.g. a teacher for his school class), responsibility for ability with regard to fulfilment (e.g. a medical practitioner in the case of an accident) and liability responsibility, which can differ from causation (e.g. 'parents are liable for their children') [25]. The determination of causal responsibility is not normative, but rather is based on empirical

evidence. It is the central problem with the opaque "black boxes" of neural networks.

In questions of liability, legal persons (e.g. companies) are treated as responsible subjects of action, if there is an transgressions occur under the applicable law. Criminal liability for institutions, however, does not exist under German law (unlike, for example, US-American law). At least morally, however, responsibility is also attributed to companies. One then speaks of corporate governance and corporate social responsibility.

In legal terms, responsibility is understood as the duty of a person to be accountable for his or her decisions and actions in accordance with specified regulations. Formally, law does not refer to moral or religious responsibility (e.g. conscience), but ("positivistically") to the violation of legal provisions, which is determined by a court. Responsibility in the legal sense is always bound to empirical facts. Therefore, the demand for more explicability of causal processes in machine learning is of fundamental importance for the clarification of legal responsibility.

In legal terms, a distinction is made between the following aspects of accountability is distinguished, for example, between the following aspects [26]:

a) With *responsibility for action*, the accountability is defined with regard to the type of the way in which the task is carried out.
b) *Accountability for results* refers to accountability for the achievement of objectives.
c) *Leadership responsibility* refers to the accountability with regard to the leadership tasks performed, including the associated external responsibility.

In law, responsibility relates not only to persons, but also to tangible assets (e.g. computers) and to the requirements of an owner, trustee or tenant. With the increasing degree of autonomy of intelligent systems, the question arises as to the degree to which, e.g., robots can still be treated as tangible goods or whether we already have to take into account intermediate areas between tangibles and persons. Animal law shows how the

traditional boundary between thing and person is inappropriate, if we take into account modern findings of evolutionary biology and cognitive psychology: Animals are living beings capable of suffering and not "things", but on the other hand they are not yet responsible "persons" [27, 28, p. 172].

Artificial intelligence is undoubtedly subject to the principle of responsibility: Only humans should determine how it is used. However, specialisation and the growing complexity of technical, societal and ecological interrelationships lead to a diffusion of responsibility: The individual is increasingly dependent on the information or assessments of other experts. As a consequence, the necessity arises of responsibility by means of legal or contractual provisions, e.g. in liability law, and/or the attribution of responsibility to collective actors such as companies and associations. However, the diffusion of responsibility also favours clear violations of the law and misuse of technology which leads to outrage and uncertainty in the general public. Safety and trust in technology are prerequisites for the future viability of a country [29].

With regard to complex AI systems and AI infrastructures, the concept of responsibility has to be extended. In systems theory, collective and cooperative responsibility must also be analysed. Responsibility should also be attributed to those who are responsible for the design of AI systems (e.g. industrial Internet resp. Industry 4.0), the development of interfaces and the use of the infrastructure. The degrees of influence are to be measured here.

Responsibility for the future requires the early recognition and assessment of risks and evaluate them. In the debate on responsibility for the future, the precept of Hans Jonas, in particular, emphasised the imperative to refrain from actions that pose an existential threat to the environment or to future generations and in this sense represent a transgression of boundaries [30]. This applies in particular to artificial intelligence.

Ethics should therefore not be misunderstood as a brake on innovation. On the contrary, raising awareness of ethics and responsibility promotes innovation advantages, such as greater legal certainty and social acceptance of AI research in society. The focus is on the international challenge, how AI systems are

to be understood as a service of democratic societies that want to continue to invoke their individual liberties and human rights. Internationally country's locational advantage is strengthened: Europe should not only be strong in AI innovation, but also take societal responsibility issues into account.

Europe must not only be a leader in AI innovation and AI research (e.g. at the interface of machine learning and industry in Industry 4.0), but also to build a related attractive societal environment. The limits of individual liberties and secure societal systems in a market economy remain important in the age of digitalisation and artificial intelligence remain great assets that are recognised and valued by all people worldwide.

References

1. Gottschlich, J.; Solar-Lzama, A.; Tatbul, N.; Carbin, M.; Rinard, M.; Barzilay, R.; Amarasinghe, S.; Tenenbaum, J.B.; Mattson, T. (2018), The three pillars of machine learning, in: arXiv:1803.07244v2 [cs.AI] 8 May.
2. Ellis, K.; Ritchie, D.; Solar-Lezama, A.; Tenenbaum, J.B. (2017), Learning to Infer graphics programs from hand-drawn images, in: CoRR abs/1707.09627.arXiv:1707.09627.
3. S. A. Cook (1971), The complexity of theorem-proving procedures, in: Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, 151–158.
4. Küchlin, W. (2021), Symbolische KI für die Produktkonfiguration in der Automobilindustrie, in: K. Mainzer (HRSG.), Philosophisches Handbuch der Künstlichen Intelligenz, Springer: Berlin.
5. Ehlers, R. (2017), Formal verification of piece-wise linear feed-forward neural networks. arXiv:1705.01320v3 [cs.LO] 2 Aug 2017.
6. Coquand, T. and Huet, G. (1988), The calculus of constructions. in: Information and Computation 76(2–3), 95–120.
7. Bertot, Y. and Castéran, P. (2004), Interactive Theorem Proving and Program Development: Coq'Art: CiC (Springer).
8. Mainzer, K. (2021), Proof and Computation: Perspectives for Mathematics, Computer Science, and Philosophy, in: K. Mainzer, P. Schuster, H. Schwichtenberg (Eds.), Proof and Computation II, World Scientific Singapore, 2–32.
9. Gonthier, G. (2008): Formal Proof—The Four-Color Theorem, in: Notices of the American Mathematical Society 55 (11), 1382–1393.
10. Obituary of Vladimir Voevodsky 1966–2017, Institute for Advanced Study October 4, 2017, 2.

11. Weyl, H. (1921), Über die neue Grundlagenkrise der Mathematik, in: Mathematische Zeitschrift 10 1921, 39–79.
12. Russell, B. (1908): Mathematical logic as based on the theory of types, in: American Journal of Mathematics 30, 222–262.
13. Born, M. (1975), Mein Leben. Nymphenburger Verlagshandlung.
14. Thiel, C. (2006), Kreativität in der mathematischen Grundlagenforschung, in G. Abel (Ed.), Kreativität, 360–375. Hamburg. Kolloquienbeträge vom XX. Deutschen Kongress für Philosophie, 26.–30. September 2005 an der Technischen Universität Berlin.
15. Kahle, R. (2018), Structure and Structures, in: Mario Piazza and Gabriele Pulcini (Eds.): Truth, Existence and Explanation. Boston Studies in the Philosophy and History of Science, vol. 334, 109–120. Springer.
16. Black, M. (1946), Critical Thinking. Prentice-Hall, 1946.
17. Thomas von Randow alias Zweistein (1963). Logelei. In: Die Zeit, Ausgabe 31, 2. August 1963. http://www.zeit.de/1963/31/logelei.
18. Reich, K. (2006), Große Forschung, Große Lehre: Emil Artin, in: Der Präsident der Universität Hamburg (ed.), Zum Gedenken an Emil Artin (1898–1962), vol. 9 Hamburger Universitätsreden. Neue Folge, 17–41. Hamburg University Press.
19. Hilbert, D. (1918), Axiomatisches Denken. In: Mathematische Annalen, 78(3/4):405–415, 1918. Vortrag vom 11. September 1917 gehalten vor der Schweizerischen Mathematischen Gesellschaft.
20. Mainzer, K. (2019), Artificial Intelligence. When do machines take over? Springer: Berlin 2nd edition.
21. Kant I (1900 ff.) Edition of Prussian Academy of Sciences. Berlin, AA IV, 421: "Handle nur nach derjenigen Maxime, durch die du zugleich woollen kannst, dass sie ein allgemeines Gesetz werde."
22. New rules for Artificial Intelligence – Questions and Answers (21. April 2021) (https://ec.europa.eu/commission/presscorner/detail/de/QANDA_21_1683).
23. Artificial Intelligence Standardization Roadmap Normungsroadmap (Eds. W. Wahlster, C. Winterhalter) (2020), DIN Berlin (https://www.dke.de/de/arbeitsfelder/core-safety/normungsroadmap-ki).
24. Planning Outline for the Construction for a Social Credit System (2014–2020). China Copyright and Media. 14. Juni 2014 (wordpress.com).
25. Hart, H.L. (1968), Punishment and Responsibility. Essays in the Philosophy of Law, Oxford.
26. Baumgartner, H.M.; Eser, A. (Hrsg.) (1983), Schuld und Verantwortung: philosophische und juristische Beiträge zur Zurechenbarkeit menschlichen Handelns, Tübingen, 136.
27. Zech, H. (2012), Information als Schutzgegenstand. Tübingen.
28. Mainzer K (2016) Information: Algorithmus-Wahrscheinlichkeit-Komplexität-Quantenwelt-Leben-Gehirn-Gesellschaft. Berlin.

29. acatech (Hrsg.) (2021), Verantwortung in Unternehmen und Institutionen. Analysen und Empfehlungen für eine nachhaltige Technikentwicklung. acatech- Positionspapier, München/Berlin.
30. Jonas, H. (1979), The Imperative of Responsibility. In Search of Ethics for the Technological Age, University of Chicago Press.

Prospects for Hybrid AI

5

5.1 Potential and Limitation of Neuromorphic AI

Classical AI research is oriented towards the performance capabilities of a program-controlled computer, which, according to Church's thesis, is in principle equivalent to a Turing machine. According to Moore's Law, gigantic computing and storage capacities have been achieved, which made AI performance possible in the first place. But the performance of supercomputers have a price that can be equivalent to the energy of a small town. Human brains are all the more impressive, that can compare the performance of a computer (e.g. speaking and understanding a natural language) with the energy consumption of a light bulb. At the latest, one is impressed by the efficiency of neuromorphic systems, that have emerged in evolution. Is there a common principle underlying these evolutionary systems that we can make use of in AI.

Biomolecules, cells, organs, organisms and populations are highly complex dynamic systems in which many elements interact. Complexity research is concerned with interdisciplinary issues in physics, chemistry, biology and ecology with the question of how the interactions of many elements in a complex dynamic system (e.g. atoms in materials, biomolecules in cells, cells in organisms, organisms in populations) can lead to the emergence of order and structure, but also chaos and decay.

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical*, Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6_5

In general, in dynamic systems the temporal change of their states is described by equations. The state of motion of a single celestial body can still be precisely calculated and predicted according to the laws of classical physics. With millions and billions of molecules on which the state of a cell depends, high-performance computers must be used which provide approximations in simulation models. Complex dynamic systems, however, obey the same or similar mathematical laws applied across the disciplines of physics, chemistry, biology and ecology.

In general, we imagine a spatial system of identical elements ('cells') that interact with each other in different ways (e.g. physically, physically, physically, physically, physically) [1, 2], [Sect. 10.1]. Such a system is called complex, if it can generate non-homogeneous ("complex") patterns and structures from homogeneous initial conditions. This pattern and structure formation is triggered by local activity of its elements. This applies not only to stem cells during the growth of an embryo, but also, for example, to transistors in electronic networks.

We call a transistor locally active when it converts a small signal input from the energy source of a battery to a larger signal output in order to generate non-homogeneous ("complex") voltage patterns in switching networks. No radios, televisions or computers would function without the local activity of such units. Important researchers such as the Nobel Prize winners I. Prigogine (chemistry) and E. Schrödinger (physics) were still of the opinion that a non-linear system and an energy source were sufficient for structure and pattern formation. The example of transistors shows that batteries and non-linear switching elements alone cannot generate complex patterns if the elements are not locally active in the sense of the described amplifier function.

The principle of local activity is of fundamental importance for the pattern formation of complex systems and has not yet been recognised to a large extent. It can be defined in general mathematical terms without reference to specific examples from physics, chemistry, biology or technology. Here we refer to nonlinear differential equations as they are known from reaction–diffusion processes (but not at all restricted to fluid media as in chemical diffusion). To illustrate this we imagine a spatial

lattice whose lattice points are occupied by cells that interact locally. Each cell (e.g. protein in a cell, neuron in the brain, transistor in the computer) is mathematically seen a dynamic system with input and output. A cell state develops locally according to dynamic laws as a function of the distribution of neighbouring cell states. The dynamic laws are defined by the equations of state of isolated cells and their coupling laws. In addition, initial and secondary conditions must be taken into account in the dynamics.

In general, a cell is called locally active if, at a cellular equilibrium point a small local input exists which can be enforced by an energy source to a large output. The existence of an input that triggers local activity can be systematically tested mathematically by certain test criteria. A cell is called locally passive if there is no equilibrium point with local activity. What is fundamentally new about this approach is the proof that systems without locally active elements cannot, in principle, generate complex structures and patterns.

In the neuronal networks of the brain, the neurochemical dynamics take place between the neurons. Chemical messengers cause neuronal state changes through direct and indirect transmission mechanisms of great plasticity. The different network states are stored in the synaptic connections of cellular switching patterns (cell assemblies). As is usual in a complex dynamic system, we also distinguish in the brain between the microstates of the elements (i.e. the digital states of "firing" and "non-firing" during discharge and the resting state of a neuron) and the macro-states of pattern formation (i.e. switching patterns of jointly activated neurons in a neural network). Computer visualisations (e.g., PET images) show that different macroscopic wiring patterns are correlated with different mental and cognitive states, such as perception, thinking, feeling and consciousness. In this sense, cognitive and mental states can be regarded as emergent properties of neural brain activity: Individual neurons can neither see, feel nor think, but brains connected to the sensors of the organism can. In complexity research, the synaptic interaction of the neurons in the brain can be described by coupled differential equations. The Hodgkin-Huxley equations are an example of nonlinear reaction-diffusion equations, which can

be used to model the transmission of nerve impulses. They were developed by the medical Nobel Prize winners A. L. Hodgkin and A. F. Huxley through empirical measurements and provide an empirically confirmed mathematical model of neuronal brain dynamics.

Technically, the information channel (axon) of a nerve cell (neuron) (a) can be represented by a chain of identical Hodgkin-Huxley-(HH) cells which are coupled by diffusion connections (b). These couplings are represented by passive resistances. The HH cells correspond to an electrotechnical circuit model (c): In a biological nerve cell, ionic currents of potassium and sodium alter the voltages on the cell membrane. In the electrotechnical model sodium and potassium ionic currents together with a current discharge through an external axon membrane current. The ion channels are technically realised by transistor-type amplifiers. They are connected to a sodium ion and potassium ion battery voltage, a membrane capacitor voltage and a voltage leakage. In this way, the input currents can be amplified according to the principle of local activity to trigger an actuation ("firing") when a threshold value is exceeded. These action potentials trigger chain reactions that lead to wiring patterns.

In the case of the Hodgkin-Huxley equations, we obtain a parameter space of the brain with precisely measured regions of local activity and local passivity. Only in the region of local activity can action potentials of neurons arise, which trigger wiring patterns in the brain. Computer simulations can be used to systematically investigate and predict these wiring patterns for the various parameter points.

In this way, the region at the "edge of chaos" can also be precisely determined. It is very small and amounts to less than 1 mV and 2 A. This region is associated with great local activity and pattern formation, which can be visualised in the corresponding parameter spaces. An "island of creativity" is therefore assumed to exist here.

The starting point of this research programme was the mathematical Hodgkin-Huxley model of the brain. In the EU's Human Brain Project, an exact empirical modelling of the human brain with all its neurological details should be realized. With the

technical development of neuromorphic networks, an empirical test bed for this mathematical model would be available, in which predictions about the formation of patterns in the brain and their cognitive meanings could be tested.

Differential equations can also be introduced for this purpose, which do not depend on the local activities of individual neurons, but rather on entire cell assemblies, which in turn depend on cell assemblies of cell assemblies etc. In this way, one obtains a system of non-linear differential equations, which are nested at different levels and thus model extremely complex dynamics. Connected with the sensors and actuators of our organism, they record the processes that generate our complex motor, cognitive and mental states. As already emphasised, we do not yet know all of these processes in detail. But it is clear how, in principle, they can be modelled mathematically, and how they could be empirically tested in neuromorphic computers.

In evolution, effective problem-solving methods developed without symbolic representation in computer models. Subcellular, cellular and neuronal self-organisation generated the appropriate complex networks. In principle, they can be simulated by computer models. These simulations are based on a fundamental mathematical equivalence of neuronal networks, automata and machines.

Thus it can be proved that a McCulloch-Pitts network can be simulated by a finite automaton. Conversely, the performance of a finite automaton can also be achieved by a McCulloch-Pitts network. In other words: An organism that is equipped with a neuronal nervous system of the type of a McCulloch-Pitts network can only solve problems of the complexity that an finite automaton can handle. In this sense, such an organism would be as intelligent as a finite automaton.

But which neuronal networks correspond to Turing machines, which, according to Church's thesis, are prototypes of computers?

It can be proved that Turing machines simulate precisely those neuronal networks whose synaptic weights are rational numbers and have feedback loops ("recurrent"). Conversely, Turing machines can be exactly defined by recurrent neuronal neuronal networks with rational synaptic weights [3].

In the biological model, the numerical values of the weights correspond to the chemical strengths of synaptic connections, which are modified by learning algorithms of neuronal networks. Intensive synaptic couplings generate neuronal wiring patterns that correspond to mental, emotional or motor states of an organism. Let us consider a Turing machine as a prototype of a program-controlled computer. Then, according to this proof, a brain with finite synaptic strengths can be simulated by a computer. Conversely, the processes in a Turing machine (i.e. a computer) can be simulated by a brain with finite synaptic intensity. In other words: The degree of intelligence of such brains corresponds to the degree of intelligence of a Turing machine.

In practice, it follows that neuronal networks of this type can in principle be simulated on a suitable computer. In fact, neural networks for practical applications (e.g. pattern recognition) are still largely only simulated on computers. Only neuromorphic computers would be able to reconstruct neural networks directly.

But what do neuronal networks with synaptic weights achieve that are not only rational numbers (i.e. finite quantities such as 2.3715 with a finite number of decimal fractions), but also any real numbers (i.e. decimal fractions with an infinite number of digits behind the decimal point such as 2.3715… which, moreover, are not computable)? Technically speaking, such networks would not only perform digital but also analogue calculations.

In signal theory, an analogue signal is understood to be a signal with a continuous and uninterrupted course. Mathematically, an analogue signal is defined as a smooth function which is infinitely differentiable, i.e. in particular continuous. The graph of such a function has no corners and interruptions that are not differentiable. Thus the temporally continuous course of a physical quantity can be described in the form of an analogue signal. An analogue-digital converter discretises a time-continuous input signal into individual discrete samples.

In fact, many processes in a natural organism can be understood as analogue. For example, the signal processing in vision is controlled by electromagnetic fields that impinge on sensors. Also the acoustics of hearing are also based on continuous waves. In the case of pressure, too the skin sensors convey a

continuous and not a digital sensation. Now one will object that measured values in a finite physical world are finite and therefore in principle be digitised.

However, the theoretical consequences of analogue neural networks are of fundamental importance for artificial intelligence. Mathematically, analogue neural networks can be unambiguously defined with any real numbers as synaptic weights, if the mathematical theory of real numbers is assumed.

The central question is whether neural networks can do "more" than neural networks with rational numbers, and thus "more" than Turing machines or digital computers. This would be a central argument in the AI debate, according to which mathematics is "more" than computer science and cannot be reduced to digital computers.

A central achievement of automata and machines is the recognition and understanding of formal languages. An automaton recognises a read-in word as a formal sequence of symbols when, after a finite number of many steps, it enters an accepting state and stops. A language accepted by an automaton consists only of words that can be recognised by the automaton. In this way, it can be proved that finite automata recognise exactly the regular languages. Context-free languages use rules whose word derivation does not depend on surrounding symbols. They are recognised by more efficient pushdown automata. Recursively enumerable languages are so complex that they can only be recognised by Turing machines.

Thus, neural networks with rational synaptic weights (just like Turing machines) can also recognise recursively enumerable languages. These can be natural neuronal systems of organisms as well as artificial neuromorphic computers that follow the laws of recurrent neuronal networks with rational synaptic weights. It can now be proved: Analogous neural networks (with real synaptic weights) can in principle also recognise non-computable languages in exponential time [3].

Corresponding proofs are mathematically possible if one accepts the concept of the computability of the computability of natural (and rational) numbers to real numbers [5]. Instead of digital processes with difference equations, continuous real

processes can also be described with differential equations. In other words: All types of dynamic systems, such as e.g. currents in physics, reactions in chemistry and organisms in biology can be described, in principle, by corresponding extended analogue systems with real numbers [4, Chap. 10].

However, it is not to be expected that analogous neural networks solve NP-hard problems in polynomial time. Thus it can be proved that, for example, the problem of the travelling salesman is also NP over the real numbers.

On the other hand, according to a proof by the logician A. Tarski (1951), every definable set over the real numbers is also decidable. On the other hand, there are sets definable over the integers which are not decidable. This is a consequence of Gödel's incompleteness theorem of arithmetic. The real computability is obviously partially "simpler" than digital computability over the integers.

The advantage of computability generalised to real numbers (analogue computability) is in any case that it handles analogue processes in organisms, brains and neuromorphic computers more realistically. Here a very profound equivalence of evolutionary, mathematical and technical processes becomes clear, which suggests an extension of Church's thesis:

Not only digital effective processes can be represented by computer models in the sense of a (universal) Turing machine, but also analogue effective processes in nature. If this extended thesis of Church is correct, then the invention of the computer opens up a fundamental insight for us. If this extended thesis of Church's is correct, then the invention of the computer opens up a fundamental insight that was initially unforeseeable in its scope:

All effective dynamic processes (natural as well as technical or "artificial") can be modelled on a (universal digital or analogue) computer.

This would be the core of a unified theory of complex dynamic systems. The symbolic codes with numbers in the computer would only be our way of processing information, representing atomic, molecular, cellular and evolutionary processes.

A distinction can be made between degrees of computability: Thus, for example, a non-deterministic Turing machine also uses

random decisions in addition to the usual effectively computable elementary operations. For this purpose, we extend the concept of the Turing machine with the concept (going back to Turing) of the ψ-oracle machine [2, Sect.10.2]: In an ψ-oracle machine, in addition to the commands of a (deterministic) Turing machine, an operation ψ is allowed (e.g. "Replace the numerical value x with value $\psi(x)$"), of which we do not know whether it can be computable. The calculation is then dependent on the "oracle". An example in nature would be a mutation as a random change in the effective processing of DNA information. One then speaks of relative computability:

A function is computable relative to ψ if it is computable by an ψ-oracle machine. Accordingly, a relativised version of Church's thesis can be formulated: All relatively effective processes can be simulated by a (universal) ψ-oracle Turing machine. Accordingly, an extended analogous version of Church's thesis (for real numbers) can be formulated.

One can prove: An analogous neuronal network recognises in polynomial time the same class of languages that a suitable ψ-oracleTuring machine recognises in polynomial time. It follows according to our definition of artificial intelligence: A natural organism with a corresponding analogous neuronal nervous system or a corresponding technical neuromorphic system are as intelligent as this ψ-oracle Turing machine.

Some mathematical and natural objects, such as a sequence of zeros or a perfect crystal, are intuitively simple, other objects, such as the human organism or the sequence of digits of a random decimal fraction such as 0.523976… obviously have a complex developmental history. The complexity of these objects can be determined by their logical depth, i.e., the computation time with which a universal Turing machine can generate its development process from an algorithmically random input. Computing time is not a physical measure of time, but rather a logical-mathematical measure of complexity, which determines the number of elementary arithmetic operations of a Turing machine depending on the input.

For natural objects, the algorithmically random input corresponds to the more or less random initial data of the evolution.

This definition of complexity by logical depth of the process of creation is thus independent of the respective technical standard of a computing machine. It can be shown [7] that (complex) objects with logical depth cannot be generated "quickly" from simple objects neither with a deterministic nor with a probabilistic process. This proof theoretically confirms our empirical knowledge of the evolution of life, the complex organisms of which have arisen through many intricate and more or less random phase transitions (bifurcations).

The transfer of logical depth to the physical and evolutionary complexity of life is based on the assumption of the extended Church's thesis, according to which processes of development and emergence in nature can be simulated by computer models and thus (extended) Turing machines with adequate efficiency.

Processes in nature are often modelled by continuous differential equations. Digital machines cannot solve continuous differential equations of dynamic systems exactly.

(Occasionally, the notion of computability for continuous system laws is not sufficiently robust, since a computable differentiable function can have a non-computable derivative.) But digital computational methods can certainly approximate dynamic processes with finite precision. Even for stochastic phase transitions, as they typically occur in complex dynamic systems and mathematically described by stochastic differential equations (e.g. master equations), discrete stochastic models are known, which can be simulated on computers.

5.2 Potential and Limitation of Quantum AI

So far we have considered artificial intelligence on machines of classical physics. With quantum computing, we go back to the smallest units of matter and the limits of natural constants such as the quantum of action and the speed of light - the ultimate ratio of a computer. As a physical machine, the performance of a computer depends on the circuit technology used. The growing miniaturisation of computers has led to new generations of computers with increasing memory capacity and reduced computing

time. Growing miniaturisation, however, leads us also into the order-of-magnitude range of atoms, elementary particles and the smallest energy packets (quanta), for which our usual laws of classical physics apply only to a limited extent. Instead of classical machines according to the laws of classical physics, we would then have to use quantum computers that function according to the laws of quantum mechanics [5].

Quantum computers would lead to breakthroughs with an enormous increase in information and communication technology. Problems such as the factorisation problem, which until now had exponential complexity, and were therefore practically unsolvable, then would be polynomially solvable. Technically, quantum computers would thus lead to an immense increase in our problem-solving capacities. In the sense of the complexity theory of computer science, the hitherto high computing times of individual problems could be considerably shortened (e.g. with polynomial computing time, although they do not belong to the complexity class P in classical computers). However, could quantum computers also be used for non-algorithmic thought processes beyond the complexity limit of a universal Turing machine? Would this open up new possibilities for artificial intelligence?

There are great possibilities for the technical construction of quantum computers, but also considerable problems of realisation. Apart from the tiny size of atomic switches, their enormous switching and signaling speed, and their low energy requirements, quantum computers could be used for the simultaneous (parallel) processing of large amounts of data. The reason for this is the superposition principle of quantum physics, which allows the formation of quantum bits. With serial data processing, a decision for a large mass of data must be checked successively for each individual data unit.

A quantum computer operates according to the laws of quantum physics, according to which the output of quantum states is uniquely computed on the basis of the input quantum states as long as their coherence is not disturbed. In quantum physics, a quantum state evolves in time unambiguously determined according to the Schrödinger equation, which is a deterministic

differential equation. The computational process of a quantum computer can be understood on the model of a deterministic Turing machine in the same way as earlier generations of computers on a mechanical, electromechanical or electronic basis [9]. Because of quantum parallelism, however, a quantum computer simultaneously can process gigantic amounts of data in a flash which are in the superposition of a single quantum state. When the individual data are read out, a random process occurs that in principle cannot be predicted exactly. This makes quantum computers a non-deterministic Turing machine. However, the read-out process can be approximated mathematically (e.g. by a quantum version of a fast Fourier transform).

In the previous section, a hierarchy of automata and machines was presented which correspond to neural networks of increasing performance. Turing machines are mathematically equivalent to neural networks with rational numbers as synaptic weights. They can recognise recursive languages that are determined by Chomsky grammars. Analogue networks with real numbers as synaptic weights correspond to special oracle machines, i.e., Turing machines, which are (polynomially restricted) oracles and can even recognise non-recursive languages.

Quantum computers are non-deterministic oracle machines that are based on quantum oracles. Quantum oracles are the random reduction of the wave packet (superposition of data) that occurs when the data of the machine output is read out. Quantum computers can also be characterised by cellular quantum automata or neuronal quantum networks [10, Chap. 9].

In general, quantum physics is the basis for the evolution of nature. In the beginning, there was a quantum vacuum from which elementary particles and atoms evolved. This basic layer of nature can only be explained with the laws of quantum physics. The resulting molecular structures, depending on their size, lie at the interface of quantum chemistry and classical physics. Biological systems up to and including metabolism in brains can be explained within the framework of chemistry and classical physics. Classical physics can be approximately embedded in quantum physics, for example, if we consider “slow” velocities (relative to the velocity of light), “large” systems (relative to

elementary particles) and "weak" gravity (relative to the attraction of black holes).

It seems to be the case that microsystems, through their (nonlinear) interactions, lead to the formation of new macroscopic structures from elementary particles, atoms and molecules to organs and brains: In the opposite direction, organ states can be explained by cellular interactions, cell states by molecular interactions, and molecular states by molecular interactions, molecular states from atomic interactions, etc. In Sect. 5.1, the principle of local activity in complex dynamic systems was introduced, in order to explain the emergence of complex structures in nature mathematically. It is worth noting that macro-states of a complex system cannot be reduced to the individual micro-states-from the superposition of quantum systems to the life of cells and organisms.

All measurements and observations to date indicate that even in the brain the emergence of new structures and states can be explained "layer by layer": Quantum mechanical interactions of elementary particles generate quantum chemical states in synapses, the molecular interaction of which leads to the wiring patterns of neuronal networks, which are connected with cognitive states of the brain. States of consciousness are therefore not in principle unsolvable "riddles". Physicians already use their knowledge of the underlying neuronal wiring patterns, to sedate patients step by step during operations or to put them in anaesthesia or to induce a coma.

However, in machine learning, the emergence of perception from neuronal circuitry patterns is technically generated, the existing knowledge of states of consciousness - at least as we know it from humans and higher organism - is not sufficient to technically generate consciousness. Self-perception of today's robots are only the first steps in this direction.

In the course of its history, technology has by no means limited itself to the simulation of natural intelligent systems. In Sect. 5.1, neuromorphic computer structures were described, which do not occur in this way in nature, but which combine the advantages of neural systems in nature with the advantages of technical systems of nature with the advantages of technical

computer structures. Likewise neuronal quantum computers are conceivable, in which the enormous computing speed and memory capacity of quantum computers are connected with neural networks.

What advantages would quantum computers bring to neural networks and machine learning? It is not only a significant increase in computing speed. Key concepts of quantum mechanics such as superposition and entanglement open up new perspectives of knowledge, classical neuronal networks and learning algorithms based on biological learning algorithms modelled on biological brains do not have. Quantum Machine Learning (QML) shows new possibilities of artificial intelligence.

Neuronal networks are open systems that exchange information with their environment and in this sense are physically dissipative. The interactions of their neurons is also non-linear. In quantum versions of neural networks, the problem is how non-linear and dissipative systems can be embedded in the linear and unitary framework of quantum mechanics. A quantum mechanical model of a quantum neuron must simulate classical neurons with sigmoid activation functions or step functions or step functions, where the states of the input functions are combined in quantum mechanical superpositions. In this way, classification systems and associative memories are developed in quantum mechanical framework. The neural network is quantum mechanically embedded by introducing a quantum bit for each neuron. The classical neurons are replaced by quantum neurons in the quantum version of a neural network.

In the end, it cannot be technically ruled out that Penrose's hypothesis that states of consciousness in the human brain can be explained by quantum superpositions in quantum physics is neurobiologically incorrect, but could one day be realised with a quantum-physical computer structure. The technical challenge is to realise superpositions over a longer period of time than in nature, independent of environmental conditions. Whether and how they can be connected with states of consciousness is then a completely different question.

Will there be epistemological breakthroughs, according to which hitherto in principle undecidable and unsolvable problems become decidable and solvable with quantum computers?

The basic undecidability and unsolvability of problems are based on the laws of logic and mathematics. Even a quantum computer will therefore in principle solve no more than is possible according to the logical-mathematical theory of computability: In principle, algorithmically and undecidable problems remain unsolvable even for quantum computers [6].

For example, the halting problem of a Turing machine is also undecidable for a quantum computer. Another example is the word problem of group theory, according to which for any two expressions of a symbol group it must be checked to see whether they can be transformed into each other by given transformation rules. Behind this is a problem that often arises in practice, e.g. whether expressions in language systems are traceable to each other or not.

In computability theory, it was proven that there is no algorithm that arrives at a decision in every case. No quantum computer will change anything. So even in a civilisation with quantum computers, there will be no machine that can solve all problems algorithmically.

Gödel's and Turing's logical-mathematical limitations remain, even if there are gigantic increases in computational speed and capacity. Every kind of of physical, chemical, biological and neuromorphic computer structure will observe the laws of logic and mathematics as will the evolution of nature itself.

Besides the superposition principle, another (classically) strange phenomenon of quantum physics is that two phenomenon of quantum physics states that two spatially distant bodies, such as elementary particles, are correlated ("entangled") with each other, although they do not interact with each other by any mechanism.

Classical information can be transmitted between transmitters and receivers, which are realised by different physical, chemical and biological carrier systems. However, transmitters and receivers must not be miniaturised in the size range of quantum

effects. In the quantum world, the transmitter corresponds to the preparation of a quantum system, the receiver to its measurement. The quantum systems (e.g. elementary particles), which evolve from the preparation state of an experiment to the measurement, transmit information in this sense [12]. Quantum information is understood to be that information which is transmitted by quantum particles from the preparation to the measuring apparatus of a quantum mechanical experiment.

In quantum physics, entangled quantum states can be used, which allow the instantaneous quantum teleportation of quantum information to distant receivers. This is not a contradiction to the relativity theory, according to which signal transmissions are only possible at the maximum speed of light. In fact, it is not a matter of an "interaction" between two objects located at different places. In quantum physics, a single quantum state is produced by the EPR correlation, which is distributed over the space between the two objects.

The catch with quantum teleportation, however, is that the quantum information to be transmitted is unknown and is only decided by the coincidence of a measurement. Quantum teleportation can therefore not be used for the direct transmission of information. In this respect there is conflict with the theory of relativity, according to which no interaction can be faster than light. However, as long as we do not measure, read out and observe quantum information, it can be transmitted instantaneously and in any superposition.

Technically, entangled states have already been realised over miles-long distances on earth. Quantum teleportation can be realised with the aforementioned statistical restrictions. The speed of light may not be an effective limit for the transmission of information on the Internet on earth. For space travel, e.g. to Mars, however, the delay due to the speed of light in the transmission and control of information from Earth already becomes a problem. Therefore, the technical realisation of entangled states on a cosmic scale will be a challenge for the future. With satellite technology, communication with intelligent infrastructures will be transformed into the Quantum Internet of Things.

5.3 Potential and Limitation of ChatGPT

> As an AI language model, I don't have personal experiences, beliefs, or a subjective understanding of the world.
>
> ChatGPT.

5.3.1 What Can the AI Chatbot ChatGPT Do?

A spectacular application example of subsymbolic AI are chatbots like ChatGPT (Generative Pre-trained Transformer), which, because of its amazing capabilities as an automatic text generator, had more followers than social media such as Instagram and Spotify within a few days, with millions of users, since 30 November 2022. ChatGPT can generate texts from school assignments at grammar school level to texts of seminar papers of middle university level. Based on a "large language model" (LLM), this AI programme can be used to talk about business plans or to commission the writing of a song, poem or novel fragments in a certain style.

In fact, ChatGPT's language model is based on a massive amount of text (Big Data) that has been trained into the system by humans. It is thus an example of machine learning based on statistical learning theory and pattern recognition, as explained in the previous section. The ambitious goal here is to overcome a key limitation of symbolic AI, which in its knowledge-based expert systems was limited to the expertise of specialists (e.g. medical expertise in a specific medical discipline), provided it could be translated into logic rule-based formulae. With the increase in computing power and the handling of large masses of data with models of statistical learning, the goal is now being pursued to also bring the general "world knowledge" of us humans to the machine.

For this purpose, the chatbot is trained with texts from news, books, social media, online forums, images, films and spoken language texts. Algorithms are used to learn from the training data. The chatbot reproduces patterns that it recognises in the stored data. This is done using the same procedures that are used

in face recognition to recognise images of people from image files. The reproduced texts are compared with trained sample texts and thus gradually improved by reinforcement learning algorithms. It is definitely impressive that this procedure is sufficient to produce grammatically correct sentences in German, a language usually considered as very complicated - and complicated to learn. Corrections can also be made if correlations of the trained data lead to discrimination, for example. Similar to indoctrinated humans, such misbehaviour can never be ruled out due to the volume of the trained data sets. Since these chatbots are widely accepted in social media, they can also cause dangerous disinformation.

Ultimately, ChatGPT is also nothing more than a stochastic machine that recombines and reconfigures data, texts, images and spoken words with pattern recognition algorithms. However, due to modern computer technologies that can store enormous amounts of data and apply fast learning algorithms, amazing results are produced that simulate a great deal of human background knowledge and intuition. But this also reveals the mechanisms on which our conversational and cultural worlds are based - reproductions and recombinations of patterns that can largely be adopted by machines. Even the social sciences, cultural studies and the humanities are not immune to this, not to mention journalism.

Wittgenstein called these “language games” that function according to certain rules. The original often consists only in a small change and variation of the usual language games and “narratives”. In machine learning, there is now talk of “stochastic parrots”. Positively speaking, ChatGPT is therefore suitable for exposing the mechanisms of the culture industry and journalism. They will have to become more sophisticated in order not to be replaced by machines.

But how can ChatGPT solve mathematics problems if in the end it is all based on statistical “guesswork” [7] ? In fact, the possible solutions depend on the stored documents. For this purpose, textbooks and a variety of other documents are trained with (human) supervised learning algorithms. In the sense of reinforcing teaching, the chatbot repeatedly restarts or improves

its proposed solutions when asked, by determining new contexts of the trained documents through pattern recognition. ChatGPT thus only knows numbers, for example, if they can be extracted from trained texts. Thus, the definition of a prime number could be reproduced if this text appears somewhere in ChatGPT's memory. But ChatGPT can only draw conclusions and decide whether a given number is a prime number or not if there are trained documents.

Calculating, logical and causal thinking are therefore basically alien to the chatbot. It guesses and associates. In this book, this central weakness of statistical learning theory and machine learning was highlighted in contrast to mathematical and logical thinking. ChatGPT can also write and evaluate computer programmes only by imitating and recombining stored templates and fragments - but at an astonishingly high level that often cannot be distinguished even by "educated" humans. The difference to human thinking is already demonstrated by a gifted pupil: without having been "fed" with all kinds of textbooks, he or she solves a mathematics problem without the effort and memory volume of a chatbot.

5.3.2 In the "Machine Room" of ChatGPT

Technically, ChatGPT is a "Large Language Model" (LLM) that generates human-like texts with deep learning algorithms from large data masses of speech. It is based on a "Generative Pretrained Transformer" (GPT) architecture, in which a transformer generates texts with a neural network. The model is trained beforehand with large data masses of books, articles, web pages, etc. to recognise patterns and structures of natural languages. Given an input (called "prompt"), the model generates a suitable text based on the previously trained knowledge.

By using a transformer, the GPT differs from previous linguistic models that sequentially predicted probable words in a text context. Transformers process all input data simultaneously. Fundamental to this is a process of "self-attention", which distributes changing weights for different parts of the input data

with reference to other positions in the speech sequence. Due to increasing computational efficiency, the GPT models have been extended and improved since 2018 from GPT1 to GPT4 for ever larger and more diverse scopes of knowledge.

A self-attention method uses a neural network to weight the importance of different parts of the input and make predictions. The input is mapped to multiple keys, values and queries that correspond to learned weight matrices. The model then calculates the scalar product of the queries with the keys for all items of the input. This produces a score for each item. These scores are then used to calculate a weight ("attention") for each item in the input. The scores are multiplied by these attention weights to add up these products as the output of the self-attention process. This output is now connected to the input and passes through the multiple layers of the feedforward neural network that realises self-attention.

To better match the outputs of the ChatPCT with the user's intentions, a reinforcement learning from human feedback (RLHF) algorithm is used, which distinguishes three steps [14]:

Step 1: Supervised Fine-Tuning (SFT) Model
The first development involved fine-tuning the GPT-3 model by hiring 40 contractors to create a supervised training dataset, in which the input has a known output for the model to learn from. Inputs, or prompts, were collected from actual user entries into the Open API. The labelers then wrote an appropriate response to the prompt thus creating a known output for each input. The GPT-3 model was then fine-tuned using this new, supervised dataset, to create GPT-3.5, also called the SFT model.

In short: Step 1 collect demonstration data and trains a supervised policy with the following partial steps:

- A prompt is sampled from the prompt dataset. Prompt dataset is a series of promptspreviously submitted to the open API.
- A labeler demonstrates the desired output behavior. 40 contractors hired to write responses to prompts.
- This data is used to fine-tune GPT-3 with supervised learning. Input-output pairs are used to rain a supervised model on appropriate responses to instructions.

Step 2: Reward Model (RM)
After the SFT model is trained in step 1, the model generates better aligned responses to user prompts. The next refinement comes in the form of training a reward model in which a model input is a series of prompts and responses, and the output is a scaler value, called a reward. The reward model is realized by reinforcement learning in which a model learns to produce outputs to maximize its reward in step 3.

In short: Step 2 collect comparison data and trains a reward model in the following partial steps:

- A prompt and several model outputs are sampled. Responses are generated by the SFT model.
- A labeler ranks the outputs from best to worst.
- This data is used to train our reward model. Combinations of rankings served to the model as a batch datapoint.

In order to speed up comparison collection, labelers with responses of ranking between $K = 4$ and $K = 9$ were used. It delivers $\binom{K}{2}$ comparisons for each prompt shown to a labeler. Comparisons are correlated within each labeling task. Therefore, if the comparisons are shuffled into one dataset, a single pass over the dataset caused the reward model to overfit. Instead, it is trained on all $\binom{K}{2}$ comparisons from each prompt as a single batch element. This is much more computationally efficient because it only requires a forward pass of the reward model for each completion rather than $\binom{K}{2}$ forward passes for K completions. As it no longer overfits, it achieves much improved validation accuracy and log loss. Mathematically, the loss function for the reward model is

$$loss(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)\right]$$

with scalar output $r_\theta(x, y)$ of the reward model for prompt x and completion y with parameters θ, the preferred completion out of the pair of y_w and y_l, and the dataset D of human comparisons.

Step 3: Reinforcement Learning Model
In the final stage, the model is presented with a random prompt and returns a response. The response is generated using the policy that the model has learned in step 2. The policy represents a strategy that the machine has learned to use to achieve its goal of maximizing its reward. Based on the reward model developed in step 2, a scaler reward value is then determined for the prompt and response pair. The reward then feeds back into the model to evolve the policy.

In short: Step 3 optimizes a policy against the reward model using reinforcement learning with Proximal Policy Optimization (PPO). PPO is a policy gradient method for reinforcement learning which alternate between sampling data through interaction with the environment and optimizing an objective function using stochastic ascent. Whereas standard policy gradient methods perform one gradient update per data sample, PPO propose a novel objective function that enables multiple epochs of minibatch updates:

- A new prompt is sampled from the dataset.
- The policy generates an output. A policy is a strategy that an agent uses in pursuit of goals.
- The reward model calculates a reward for the output.
- The reward is used to update the policy using PPO. Kullback-Leibler penalty for SFT model is used to avoid overfitting.

In training with reinforcement learning (RL), the following objective function is maximized:

$$objective(\phi) = E_{(x,y)\sim D_{\pi_\phi^{RL}}}[r_\theta(x,y) - \beta \log(\pi_\phi^{RL}(y|x)/\pi^{SFT}((y|x))] + \gamma E_{x\sim D_{pretrain}}\left[\log\left(\pi_\phi^{RL}(x)\right)\right]$$

with the learned RL policy π_ϕ^{RL}, the supervised trained model π^{SFT}, and pretraining distribution $D_{pretrain}$. The KL reward coefficient β and the pretraining loss coefficient γ control the strength of the KL penalty and pretraining gradients. Für PPO models, γ is set to 0.

5.3.3 Challenges of ChatGPT for Education Policies

The analyses in the previous sections show that the chatbot ChatGPT is not magic, but is based on computable algorithms of stochastics and statistical learning theory. Therefore, its performance and limitations can also be clearly assessed. Neither euphoria nor excessive timidity are therefore appropriate to the matter. ChatGPT has caused unease especially in media, culture, and education. The question lurks everywhere whether professions in these fields could be replaced by chatbots in the future. Against the background of the foundational analysis of chatbots, the following will assess the significance of ChatGPT for concrete job profiles in education and training.

For entry into a profession, personnel managers play a central role in the various companies. They assess the suitability of applicants on the basis of written documents and personnel interviews. In the process, a standardisation of questions can be observed, to which desired standard answers can be given. However, a standardised assessment procedure can easily be simulated with the current services of chatbots. Standard questions must therefore be avoided. Interactions in the assessment must play a stronger role than written surveys according to standard questionnaires. In the end, human resource management is also not about text generation, but decisions. However, there will be dips and changes in personnel marketing. A job advertisement or careers website can be easily and professionally written by ChatGPT.

ChatGPT already writes simple programmes in computer science. In fact, the programming profession and the systems architect profession can be expected to change without being replaced by AI. Indeed, ChatGPT can already provide (simple) building blocks of programming to be used in writing more complex programs. At the same time, however, this will make the programming profession more demanding and professional. It should also be taken into account that the neural networks of chatbots will only be one example of programmes that will change the

work of programmers in the future. However, programme verification will be all the more important in the future. The smallest errors in elementary building blocks that are "automatically" generated by chatbots such as ChatGPT can have a catastrophic effect on the entire software if they are not recognised in time. Therefore, a high qualification of programmers is indispensable.

In the media and journalists' associations, ChatGPT is sometimes perceived as a threat. In fact, this chatbot writes desired articles and essays in perfect national language. Routine articles could definitely be done automatically. If the journalist wishes, the linguistic style could also be adapted to a particular writer. So, in the sense of the Turing test, these writers are replaceable. Bans on the chatbot, which are demanded by some professional associations, are of little help here. Rather, one must learn to deal with this technology and improve one's own performance. For more demanding texts, ChatGPT could help pre-structure and incorporate the necessary data. The editor should exercise control and responsibility (also in the legal sense). In particular, false information that would otherwise be reproduced and passed on by the AI should be weeded out. One could also distribute chatbots in the network, which "spontaneously" make statements in the desired context and pass on propaganda and disinformation. So the challenges in the media sector are great, but so are the opportunities for improving quality. In journalism training, the chatbot could generate sample articles on certain topics, which are then critically assessed by the students in order to improve their later work.

It becomes particularly sensitive in professions where language is used to convey feelings and empathy, such as psychologists and psychotherapists. Weizenbaum's early language programme ELIZA was already intended to simulate a psychotherapist. At the time, Weizenbaum was appalled at how this simple programme was accepted as a psychotherapeutic interlocutor: People projected their own desires and fears into this programme. With ChatGPT, automated interlocutors become conceivable that can be used as substitutes for human interlocutors. This could be an extremely problematic business model for

a psychotherapist who uses chatbots en masse in order to collect fees for such conversations. This would not only be profit-seeking, but extremely dangerous for psychologically vulnerable patients. Tests show that the chatbot also generates false and skewed information. The chatbot could be used as a transcriber of conversations or for advice based on available data, which would then have to be critically proofread. In all these applications, it must be clear that the chatbot only performs statistical data analysis based on large data masses with pattern recognition. It can therefore only understand and convey feelings and empathy to the extent that previously trained texts spoke about them. In training, answers from the chatbot can be critically assessed by students in order to train their own psychotherapeutic judgement.

Language-dependent professions are also legal professionals as e.g. lawyers, prosecutors or judges. Thus, it is conceivable to entrust ChatGPT with the task of a business lawyer. A company wishes to have articles of association for a certain legal form of a company (e.g. in Germany a GmbH). For this, a company describes its profile by answering certain standard questions. The chatbot then automatically drafts the company's articles of association. Legal databases already exist in Germany, but they generate many answers and options to queries, which a lawyer must laboriously work through. Since public prosecutors and judges, for example, suffer from the enormous flood of pending cases and trials, they could all be too happy to rely on the quick and seemingly efficient help of a chatbot. And that would be really dangerous.

The reasons are obvious: Law in particular shows the clear limits of today's chatbots. The language of law is extremely complex and standardised. What seems like a plausible and well-formulated answer to the layperson can be wrong, skewed and misleading. Similar to "solving maths problems" in mathematics, "solving legal cases" is therefore a central field of training at university for students of law. Here, too, ChatGPT can be used in a didactically meaningful way by critically analysing and discussing the chatbot's answers in the seminar or in exercise

groups by the students in order to improve their own problem-solving skills. The chatbot can also help to write linguistic summaries of complex judgements and legal cases for specific purposes. But especially in the legal field, it is ultimately a matter of responsibility up to and including liability, which cannot be delegated to an automaton.

The possibilities of ChatGPT in schools and universities are currently causing great concern. Not only essays at schools, but also seminar papers up to Bachelor's, Master's and PhD degrees can be generated in writing at a high level in social sciences, cultural studies and the humanities, as long as it is text production. These papers usually pass the Turing test.[1] Some universities in e.g. Germany and Italy reacted by banning ChatGPT, which might be understandable but should be considered as completely wrong. Here, too, we have to learn to understand and deal with chatbots as an advanced cultural technique.

Even the philosopher Plato was upset in his time of Antiquity about the use of writing instead of an oral dialogue because he saw it as distorting true thought. Later came the art of printing, and finally text processing on typewriters and then on PCs. The older generation has experienced all these cultural techniques: at primary school one first wrote on a slate, then came exercise books with pencils and fountain pens, finally a dissertation on a mechanical typewriter and then came word processing on the personal computer (PC). Today, meetings can be held online anywhere in the world. Of course, the respective advantages and disadvantages of these cultural techniques shift. Wikepedia, too, is now used worldwide after fierce initial criticism. Today, even highly specialised scientists use this tool to inform themselves

[1] Ironically, when students hand in essays written by ChatGPT they can easily be convicted if the essays do not contain the "usual" orthographic mistakes of contemporary students. However, it should only be question of time that they ask ChatGPT for essays "written with the usual mistakes of an average student".

first. Moreover, Wikipedia, for example, has been considerably improved since its beginnings. We have learned to use this information transfer wisely, without relying on it blindly.

Once again: the book glorified by humanitic scholars with its book culture was judged by one of the founders of Western philosophy to be an extremely questionable form of exchanging ideas. Perhaps it is not far-fetched to think that, similar to technologies in engineering, cultural technologies also have their time in the humanities. In technology, we speak of bridging technologies that prove themselves over a certain period of time, but are then replaced by new technologies under changed conditions.

For example, we are currently experiencing the replacement of the diesel engine, which was an ingenious and revolutionary technology for over a hundred years, but is now reaching its limits under the conditions of environmental change and the possibility of, e.g., electric motors, batteries and renewable energies. However, the critical discussion about batteries in this example shows that this is not the last word, but only another bridging technology that will again reach its limits. The history of nuclear power is similar.

The possibilities of ChatGPT at school and university should therefore not lead to bans, but to the critical question: Are examinations, as we traditionally know them, still up-to-date and appropriate in a changed working world with different technical conditions. The situation is quite the same as when pocket calculators entered the stage. Of course, they didn't replace the need to teach elementary calculation skills; but they can be safely used at higher school levels. And it took a while that educational studies were able to find the adequate balance between banning and allowing pocket calculators in class. To have similar studies for chatbots as ChatGPT, first of all, a fundamental discussion is required that asks about the possibilities and limits of this technology. We need to know the algorithms in order to be able to assess the possibilities and limits. This requires basic theoretical knowledge, but also practical experience in dealing with these programmes. So learners should first be given a basic understanding of machine learning and the special algorithms of

chatbots like ChatGPT. Then comes their own experimentation with orders to the chatbot and the evaluation of its answers.

It is important to understand, for example, that this is reinforcement learning, in which new and modified answers are given through constant questioning, which in the best case improve. However, this depends on the chatbot's knowledge base, which was previously taught to the chatbot through supervised learning in a training phase with a human supervisor. It follows that an initial response from the chatbot is not yet directly usable, but requires post-processing and correction. In this iterated way, sample solutions in the various disciplines could be generated in dialogue with the chatbot. The challenge is to keep the control of these dialogues and to resist the temptation to trust chatbots with reflection.

An appropriate use of ChatGPT in examinations therefore depends on the boundary conditions [8]. Only in oral examinations and written examinations can the examiner largely ensure that there is no cheating. However, it is also a question of scale whether in some subjects hundreds or even thousands of candidates have to be examined or a manageable small number. For written assignments, it depends on the subject how sure one can be. In fact, it becomes difficult with text generation tasks in the cultural sciences and humanities. Empirical papers in the social sciences and economics are based on empirical data that can be controlled by checking sources.

Incidentally, in the natural sciences, for example, it is quite conceivable that in the case of specialist articles, the linguistic formulations in the terminology typical for the subject or the structuring of the article in the manner typical for the subject could be generated by a chatbot, while the results of the actual new laboratory discovery only have to be inserted. Accordingly, there are examination performances in which verbalisations only make up a part of the examination performance. This refers to laboratory experiments, statistical analyses or programming. It must be admitted, however, that the chatbot undermines the ability to argue and present thoughts in writing and oral

presentation. It is not able to reconstruct the logical structure of an argument (but rather gives just sentences which appear to represent an argument according to the statistically learned examples). In fact, chatbots still fail on essentially all the reflective questions discussed earlier in this book, which one can easily check by asking them: "Why?" But arguing skills are, for example, quite central for leadership tasks in a company. Exams that no longer reflect these skills are therefore of little help to a company. Here, the limits of chatbots must be critically evaluated and other forms of examinations, such as examination interviews, must be demanded.

From subject to subject, it must be examined exactly how the respective subject competence can be replaced by a chatbot. It must not be forgotten that examination performance is also an important tool for students to recognise their own abilities, talents and limitations in order to find a suitable career later on. The objectivity of examinations must remain an important yardstick for awarding scholarships and university positions, for example. From a legal point of view, it must therefore be ensured that the misuse of technical aids such as ChatGPT can be established with legal certainty. Already under the impression of the pandemic, a legal order was passed in Bavaria to be able to take electronic distance examinations. Accordingly, a legal framework must be created to regulate the use of chatbots such as ChatGPT in university examinations. Questions of equal opportunities and compliance with data protection standards will play a crucial role.

The example of ChatGPT clearly shows the stage of development that AI tools are at: They are by no means in a position to replace humans in decisive ultimate responsibility. However, solutions to problems are now being developed in an interaction between humans and AI. The biological image of a symbiosis is quite appropriate here. This interaction of artificial and natural intelligence must also be reflected in education and training, including the assessment of examination results.

5.4 Quo Vadis AI?

5.4.1 An Optimistic Vision

> … daß, wenn unter allen möglichen Welten nicht eine die beste wäre, GOtt gar keine producirt haben würde.
>
> Gottfried Wilhelm Leibniz [16, p. 182]

In the introduction, the future goal of AI research was stated to connect statistical learning algorithms with logical and knowledge-based methods. The combination of symbolic and subsymbolic AI is also called hybrid AI. Epistemologically, it corresponds to a "hybrid" cognitive system like the human organism, in which the ("unconscious") processing of perceptual data is combined with ("conscious") logical reasoning. Hybrid AI is therefore higher degrees of intelligence than the reduction to symbolic or sub-symbolic AI. Nevertheless, all three forms of AI are currently in practical use side by side, depending on the respective requirements of the application area. In the automotive industry and medicine, for example, knowledge-based expert systems and statistical machine learning are used for different applications side by side. Hybrid AI is already being pursued in robotics, which will in the future be supported by neuromorphic structures and quantum technology.

The construction of a technical brain is conceivable, which is not based on neurobiological neuronal networks like a natural brain, but on neuronal quantum networks. The advantage of this would be that they could do everything, what classical neural networks (i.e. biological brains) could do, but with all the additional advantages of quantum physics, such as speed (through e.g. superpositions, entanglements and quantum tunnelling). By the way: Roger Penrose speculated on whether natural brains could be formed according to the laws of quantum physics. Probably not, because brains are much too warm and susceptible to perturbations. But that is not the question here. Engineers do not want to explain the (human) brain like neurobiologists and philosophers. Rather, they want to build a functional specimen for specific purposes! AI research and computer science see

themselves today largely as engineering sciences. In this case, the performance and limits of the product depend on the type of network. Mathematical proofs already exist for the limits of individual such network types. We then know again what they can and what they cannot do. However, so far no general limits can be derived for this type of AI.

What is becoming apparent in research, the economy and society is increasing cooperation between humans and machines. Already with the number sieve, mathematical security was achieved by linking many technical and human computers. At the end of this development, therefore, it is not, as is often feared, a matter of an artificial intelligence that replaces the human being: technology needs mathematics since Archimedes, but mathematics is also increasingly dependent on technology and influenced by it in its development (e.g. the solution of the factorisation problem depends on the technical realisation of a quantum Fourier algorithm).

Symbolic AI (e.g. automatic reasoning, knowledge-based systems) and sub-symbolic AI (e.g. statistical learning) are combined in hybrid AI (e.g. embodied AI). But AI does not develop in isolation. Humans also change their thinking and adapt to the thought structures of machines and programmes. Humans do therefore not only rebuild their organism through technical, biological and chemical implants, but also changes his cognitive and intelligent abilities through technology. The use of these tools changes us and our thinking.

This also shapes mathematics and its methods and gives them their direction of development. In the eighteenth and nineteenth centuries, the development of mathematics was decisively shaped by the problems of classical physics. At the beginning of the twentieth century, the influence of quantum physics came into play, but also the mathematical problems of economics and the social sciences were added. With computer technology and AI algorithms, a new evolutionary thrust is possible: man and machine are increasingly developing in a symbiosis. Man changes AI and machine. But machine and AI also change humans.

In the end, however, everything runs in the direction of a hybrid intelligence which will develop evolutionarily in a symbiosis with human intelligence, symbolic, sub-symbolic and hybrid AI.

Hybrid intelligence will not come in the distant future, but has already been partially realised or is in the process of being realised. It describes a path, not a final singularity like the so-called "superintelligence". As always in evolution, breaks and falls are not excluded. Therefore, judgement with clear concepts is required in order to avoid the collateral damage of evolution.

5.4.2 A Sceptical View

> Je n'ai vu aucuns (mortals) qui n'aient plus de désir que de vrais besoins, et plus de besoins que de satisfactions. J'arriverai peut-etre un jour au pays où il ne manque rien; mais jusqu'a présent personne ne m'a donné de nouvelles positives de ce pay-à.
>
> Voltaire [17, p. 108 f.]

From the given description of Artificial Intelligence it follows, that only in a hybrid form, which combines classical, rule-based techniques with the new, statistics-based methods, it will be able to expand its field of application in the long term.

Up to now, AI algorithms have been characterised by solving special problems in which they are actually far superior to humans.[2] Since the beginnings of AI in the 1950s, however, human intelligence has always been associated with the ability of the all-rounder: Some computer programs are supposed to calculate faster, others are supposed to recognise images better and still others should be able to translate Chinese. Humans can also do this in principle, albeit more slowly and in a more limited

[2] Alan Turing had already pointed out from the beginning that there are specific areas in which the comparison of computers and humans is not meaningful [18, p. 435]: "We do not wish to penalise the machine for its inability to shine in beauty competitions, nor to penalise a man for losing in a race against an aeroplane.".

way with a seemingly unlimited background knowledge which enables them to have what is called “common sense” or general knowledge.

For this reason, the AI community first tried its hand at a “general problem solver” in the 1950s. But this attempt at an AI programme that was supposed to solve a wide variety of problems was more than modest. In the ironic words of Horace more than 2000 years ago in his Ars poetica: „The mountain gave birth and gave birth to a mouse.“ The useful “lab mice” that emerged during this first attempt at AI were AI languages such as LISP and Prolog. Therefore, the search for a computer program with common sense and all-rounder capabilities was soon abandoned, and research concentrated initially on specialisation with expert systems.

As has already been emphasised several times, the breakthroughs of the new AI would not have been possible without the enormous increase in computing power and memory capacity in recent years. Therefore, it makes sense to use this approach to develop an AI for versatile tasks that has a general understanding of language like that of a human being with the ability of multifaceted problem solving. This would overcome a central limit of previous AI, known as the common sense problem. AI algorithms and Big Data are once again the great hope of the industry. Not only large companies, but also global powers such as the USA and China are therefore currently starting to invest billions in storage and computing capacities with the development of corresponding algorithms. AI algorithms are being developed to solve the common sense problem.

To this end, research and companies in Germany have joined together to form a consortium under the name OpenGPT-X, which is funded by the state as part of the Gaia-X initiative. The aim is to develop a general language model for German, which should also benefit smaller and medium-sized German companies and be internationally competitive. The Fraunhofer Institute for Intelligent Analysis and Information Systems (St. Augustin) coordinates a research cluster consisting of the Research Center Jülich, the TU Dresden, the German Research Center for Artificial Intelligence (DFKI), the AI companies Aleph Alpha,

Alexander Thamm and Control-Expert, the Internet service provider 1&1 IONOS, the Westdeutscher Rundfunk (WDR) and the German AI Association.

As is often the case, the USA has already set the standard: The Californian AI company OpenAI developed a huge neural network called GPT (Generative Pretrained Transformer) with 175 billion parameters, to be better trained with more and more data. The parameters can essentially be thought of as the weights with which the synaptic connections of the neurons in a neural network are connected. In early 2021, Google came out with a language model called the Switch Transformer, which has 1.6 trillion parameters. This system is only surpassed worldwide by the Chinese Beijing's WuDao 2.0 from the Chinese Beijing Academy of Artificial Intelligence (BAAI) with 1.75 trillion parameters. This makes the Chinese competitor ten times as large as GPT 3 and can be trained not only with speech data but also with image data.

Compared to earlier language models, the language understanding and processing of GPT 3 remains remarkable. Developers in Germany such as Aleph Alpha assume, however, that only successor models such as GPT 4 or GPT 5 will actually be able to process all of the world's knowledge, and thus reach the level of common sense or human general knowledge. The computing power required for this in Germany can only be achieved by a supercomputer such as JUWEL (Jülich Wizard for European Leadership Science), which is one of the ten most powerful (classical) supercomputers in the world. The coordinators of OpenGPT-X thus rely on an infrastructure of supercomputers, which is necessary to realise the training of large language models.The next step is to technically capture the diversity of European languages. The aim is to standardise this diversity in a "European Language Grid" (ELG) in order to remain internationally competitive. In the end, this language technology is certainly an important contribution,to preserve smaller European languages that are threatened with extinction and thus to safeguard Europe's cultural heritage.

On the other hand, this effort overcomes billions and trillions of parameters in neural networks with enormous computing

power of supercomputers does not overcome a fundamental limit of current AI, which has been pointed out several times: How can e.g. children with relatively little data and background knowledge develop an understanding of contexts that enables them to understand irony, jokes or threats in a communication situation? Mass (Big Data) instead of class is not enough to explain this phenomenon. Even if successors to GPT 3 should succeed in achieving an understanding of irony and wit with a gigantic effort of technology, then this technology, there would still remain the "miracle" of the human brain, which achieves the same (?) performance with low current and little data. Now, one could, as in the previous section, look to the section, one could refer to the future of neuromorphic computer structures. This would be on the side of an optimistic future strategy, which, however, is not foreseeable under today's technical conditions.

What is practically foreseeable is the development of self-driving cars, on which the automobile industry is relying worldwide. Volkswagen expects to reach Level 4 of self-driving vehicles by the end of the 2020s. However, this only means that vehicles can drive themselves on precisely defined roads and situations [2, Sect. 9.2]. The decisive factor here is the performance of sensors with cameras, radar and lidar. Neural networks are also required for this, e.g. to correctly estimate the distance to vehicles in traffic and to adjust speeds.

In the end, it is also a matter of the common sense of a human driver who, with sufficient experience, solves such questions intuitively and unconsciously. This is precisely what Hubert Dreyfus had pointed out in order to demonstrate the limits of formal programmes. Tesla now also wants to overcome this limit through supercomputers and Big Data. To this end the infrastructure of a supercomputer with 5760 graphics processors will be specifically designed to train Tesla's massive neural networks. Once again, therefore, the strategy is to push the previous limits of AI with ever larger models, data volumes and computers. In this case, the aim is to achieve the visual abilities of humans when driving a car. On this basis, Tesla wants to realize fully autonomous driving in 2035, if by then the even faster supercomputer infrastructure already planned is in place. Even

in that case, the enormous amount of technology and energy that is necessary to get anywhere near simple human capabilities is amazing.

However, there are still substantial limits to artificial intelligence.

With regard to the practical limits discussed in Chap. 2, one could move away from a conceptual analysis in the sense of expert systems, which complements machine learning. It might be hoped that the concepts of causality, which are widely discussed in philosophy today, will also be preserved in AI. However, scepticism is called for if one expects the statistical methods to overcome the difficulties which determined the fate of expert systems. Similarly, a hybrid AI will not be able to move from "big data" to "small data", because even if symbolic AI were to succeed in achieving results with little data, there would still be a need for data, there is no possibility of application in this field for small data. Possible applications for machine learning do not arise in this area. Similarly, the analysis of the data quality, as well as overcoming the 'frame problem', requires theoretical preparation, which can ultimately only be provided by a background theory.

As far as the theoretical limits are concerned, it is first of all true that new technologies such as quantum computing can push the boundaries of complexity, but they do not really overcome them: At best, one only encounters the same hurdles as before at another class of complexity; and with regard to the limits of computability, nothing changes anyway. In this respect, the area in which expert systems algorithmically generate new results may be extended; however, fundamental barriers will remain. In the combination of statistics-based and symbolic AI, there is certainly the potential for small advances where computers are still holding out their arms today. But this combination is not a tool for overcoming theoretical limits as such. It is important to point out that the existence of such limits is precisely the basis for any successful cryptographic protocol. Insofar as there is an interest in a protected exchange of information (in banking, for example), unsolvable problems in the sense of complexity theory are necessary in order to guarantee security.

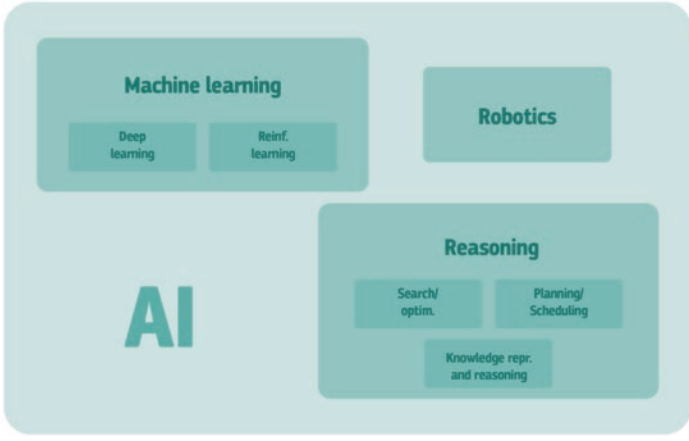


Fig. 5.1 AI's sub-disciplines and their relationship [19, p. 6]

For a number of the conceptual problems of statistics-based AI discussed in Chap. 4, in principle, conceptual systems modelled on expert systems could provide a solution. But, as already mentioned, there has been no substantial progress in this area. The actual "intelligent" part, i.e. the creation of the appropriate conceptual system, remains the responsibility of humans.

In addition, hybrid AI requires a genuine dove tailing of symbolic and sub-symbolic AI. In the proposal of the European Commission for a new definition of AI mentioned at the beginning of this book, there is an illustration that underestimates this interlocking, if machine learning and reasoning are placed in separate boxes, unconnected and in opposite (Fig. 5.1).

References

1. Mainzer, K.; Chua, L. (2013), Local Activity Principle, London.
2. Mainzer, K. (2019), Artificial Intelligence. When do machines take over? Springer, 2nd edition.
3. Siegelmann, H.T.; Sontag, E.D. (1995), On the computational power of neural nets, in: Journal of Computer and Systems Science 50, 132–150.

4. Siegelmann, H.T.; Sontag, E.D. (1994), Analog computation via neural networks, in: Theoretical Computer Science 131, 331–360.
5. Blum, L; Shub, M.; Smale, S. (1989), On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal Machines, in: Bulletin of the American Mathematical Society 21 1, 1–46.
6. Mainzer, K. (2018), The Digital and the Real World. Computational Foundations of Mathematics, Science, Technology, and Philosophy, World Scientific Singapore.
7. Bennett, C.H. (1995), Logical Depth and Physical Complexity, in: R. Herken (Hrsg.), The Universal Turing Machine. A Half-Century Survey, Wien, 2007–235.
8. Feynman, R.P. (1982), Simulating Physics with computers, in: Intern. J. Theor. Physics 21: 467–488.
9. Deutsch, D.; Eckert, A. (2000) Concepts of Quantum Computation, In: Bouwmeester, D.; Ekert, A.; Zeilinger, A. (Eds.), The Physics of Quantum Information, Quantum Cryptography, Quantum Teleportation, Quantum Computation. Berlin, chapt. 4.
10. Mainzer, K. (2020), Quantencomputer. Von der Quantenwelt zur Künstlichen Intelligenz, Springer.
11. Deutsch, D. (1985) Quantum theory, the Church-Turing principle and the universal quantum computer, in: Proc. R. Soc. London A 400: 97–117.
12. Keyl, M. (2002) Fundamentals of quantum information theory, in: Physics Reports. A Review Section of Physics Letters 369: 431–454.
13. S. Frieder et al. (2023), Mathematical Capabilities of ChatGPT, in: arXiv:2301.13867v1 [cs.LG] 32 Jan 2023.
14. L. Ouyang et al. (2022), Training language models to follow instructions with human feedback, in: arXiv:2203,02155vl [cs.CL] 4 Mar 2022.
15. J. Gogoll, D. Heckmann, A. Pretscher (2023), Endlich neue Prüfungen dank ChatGPT, in: FAZ 20.3.2023 Nr. 67, p. 18.
16. Leibniz, Gottfried Wilhelm (1760). Essai de Theodicée oder Betrachtung der Gütigkeit GOttes, der Freyheit des Menschen und des Ursprungs des Bösen. Amsterdam.
17. Voltaire (1752/1877). Micromégas, CEuvres complètes de Voltaire, Garnier, 1877, tome 21.
18. Alan Turing (1950). Computing machinery and intelligence. Mind 59, 433–460.
19. High-Level Expert Group on Artificial Intelligence. A Definition of AI: Main Capabilities and Scientific Disciplines. European Commission, Directorate-General for Communicatioin, 2018.

Author Index

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical,*
Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6

Subject Index

K. Mainzer and R. Kahle, *Limits of AI—Theoretical, Practical, Ethical*, Technik im Fokus, https://doi.org/10.1007/978-3-662-68290-6