Essay

AGI Considered Impractical

Decao Mao *

Posted Date: 29 September 2025

doi: 10.20944/preprints202509.2365.v1

Keywords: Turing machine; Church-Turing Thesis; Turing test; connectionism; symbolism; AGI; Two Minds Hypothesis; System 1; System 2; linear regression

 doi:10.20944/preprints202509.2365.v1

Essay

AGI Considered Impractical

Decao Mao

decaomao@outlook.com

Abstract

In recent years, AI technology has made significant and astonishing progress, leading optimistic researchers who have succeeded in this field to claim that artificial intelligence will be able to do everything we humans can, meaning it can realize a complete form of AGI. However, this article argues that it is unlikely, it is impractical for machines to replicate all cognitive activities of the human brain; at least in the foreseeable future, some activities in the human brain cannot be imitated because no such algorithms are invented, these processes are not "Turing computable". To realize AGI, the "Two Minds Hypothesis" must be implemented.

Keywords: Turing machine; Church-Turing thesis; Turing test; connectionism; symbolism; AGI; Two Minds Hypothesis; System 1; System 2; linear regression

---

In recent years, AI technology has made significant and astonishing progress, leading successful researchers in this field to become overly ambitious and optimistic, they declare that artificial intelligence will be able to accomplish everything we humans can do, achieving a truly complete form of *Artificial General Intelligence*, namely AGI. Some even predict that AGI will be realized before 2030, and a *singularity* is imminent.

However, in the foreseeable future, it is highly unlikely and unrealistic for machines to accomplish everything our human brains can do, because some intelligent activities of our human brain simply cannot yet be performed by machines; these activities of the human brain are not computable by a Turing machine so far, since no algorithm is available to implement that.

Nevertheless, the inability to achieve AGI does not mean that machines cannot *imitate* many of the activities in the human brain; actually, most of them can be imitated. While connectionist neural networks and deep learning essentially perform linear regression on previously occurred samples, symbolic logic reasoning is also a key feature of the human brain; neither one should be neglected. In the old-day, researches in AI technology concentrated on symbolic logic while excluding connectionism, which turned out to be a failure; but now a focus on connectionism while neglecting symbolic logic is not a good strategy either. Future developments in AI technology should balance both. Yet despite this, we are still far from achieving AGI. This article will explain why.

§1 Turing machine and Church-Turing Thesis

In 1936, Turing published his concept of the *Turing Machine* and proposed the *Turing Thesis*, which was later combined with the earlier *Church Thesis* to become the *Church-Turing Thesis*. Their discussions used rigorous and profound formal logic extensively, which is very theoretical; whereas most people are obviously not familiar with formal logic, that gives the *Church-Turing Thesis* a cloth of mystery. Many people find it too difficult to understand and get deterred by it. However, the basic principles they discussed are actually not complex and could be expressed in much simpler terms that everyone could understand (though not as rigorously). Nevertheless, actually they had no choice, and they had to talk in that way, as the audience they were facing is logicians and scholars, the foundation of logic is precise and rigorous, every step in the reasoning must be logically solid, and it is inevitable many peculiar symbols are to be used, which is exactly what makes logic unassailable and robust. However, unfortunately, this also deters many people from reading and understanding, which is regrettable but unavoidable.

 doi:10.20944/preprints202509.2365.v1

At that time, large and complex calculation projects, say the compiling of a common logarithm table, would require a considerable amount of manpower to get involved. These computing workers, the *computors* (not computers), would perform calculations under the guidance and instructions of the chief mathematicians, using hand-cranked calculators which had been available since 1878. However, hand-cranked calculators could only perform addition, subtraction, multiplication, and division; while the computations the computing workers were asked to do were far more complicated. Taking the compilation of the common logarithm table as an example, it required the participation of many workers. These workers were well-educated but were not necessarily mathematicians. If they were each, for example, assigned a task to compute from log(1.0) to log(1.999), from log(2.0) to log(2.999), and so forth, with a step length of 0.001, they simply don't know how to compute. Anyway, who knows how to compute for example log(1.21)? However, if they were given with the Taylor series expansion of the logarithm function, which breaks down the computation into addition, subtraction, multiplication, and division, so that they could compute step by step with the basic arithmetic calculations; and if an instruction on how many terms to take in the series is given, so that a certain precision can be achieved, then they could be happy and perform the calculations easily.

This brings up the question of what kind of functions are *computable*. In common sense, if you want me to do something, you have to tell me how to do it, if I didn't know it beforehand. So, the chief mathematician must provide a concrete method of computation along with a task assignment, and every step in this method must be known to the workers, and the method must terminate in a finite number of steps. Otherwise the assignment is simply not computable. To fulfil this requirement, Church's solution is to provide the worker with an *algorithm* written in a certain kind of expressions which he called a *λ-function*, or *λ-calculus*. A valid λ-function is an expression indicating how the function (the mapping) can be implemented step by step, all in known methods, such as arithmetic calculations; instead of how the function is mathematically defined. The word *algorithm* means a method of computation, whereas the word *computation* means the step-by-step execution of an *algorithm*. So, a valid λ-function represents the algorithm of a certain function, such as the Taylor series expansion of the logarithm function, instead of the mathematical definition of that function, like log(x).

However, not all functions have known algorithms, and thus not all functions can be expressed as valid λ-functions; computations that cannot be expressed in valid λ-functions cannot be computed in practice. For example, the original function definition log(x) cannot be used as a valid λ-function, because we don't know how to compute, which should be: find a value e, so that 10^e equals x. But if we know log(x) can be expanded into a Taylor series, then the Taylor series can be written as a valid λ-function of log(x), with a given length; since every term in a Taylor series can be calculated with arithmetic operations, by the hand-cranked calculators. However, if the λ-function representing its algorithm has been explicitly given in advance, and it is well known by the workers, then the function log(x) can be directly written as a λ-function, since the workers all know that means using the given algorithm, just like what we now called it a subroutine call.

However, there are some special problems which are intrinsically not computable, or "undecidable", that means no algorithm will work for these problems. So, there comes an issue of what kind of problems are "computable", for which an effective algorithm it possible to be found, earlier or later, for which Church asserts that every "recursive function" can be expressed as a valid λ-function, and thus is computable, but this is a profound issue, and we do not need to dive so deep. But note that even if we cannot prove a certain problem is intrinsically not computable, it is practically incomputable before an effective algorithm is found. On the other hand, in the related literatures, the discussion of computability was usually focused on the intrinsically incomputable problems, which makes the discussion very deep and obscure, but what the readers need to know is actually the issue of *practically* computable.

Nevertheless, we will see later that, for those problems we don't have valid algorithms yet, it is also possible to define *end-to-end* mappings by providing key-value pairs, akin to using lookup tables

 doi:10.20944/preprints202509.2365.v1

to obtain function values without specific calculations; in this case, the table lookup operation is also a computation in a broader sense. But which actually relies on the computational results of others; for example, you can refer to a logarithm table, but someone had to compute the input-output mappings for this logarithm table, so this does not solve the problem of computability.

Therefore, Church addressed the issue of how to enable the human workers to carry out computations. While it seems not a big deal, his idea of expressing computable functions using λ calculus as the algorithms, is of significant importance. In fact, the so-called dynamically defined functions we use in today's programming are essentially λ functions.

This is the essence of the *Church Thesis*. It is termed *thesis* rather than *theorem* because it is not a mathematical concept, instead it is a concept for practice, for execution, for implementation. The *computation* did by the workers is the concrete performing of algorithms, which is essentially the application of λ functions.

Turing, on the other hand, took a significant step forward; he didn't pay attention to having humans execute algorithms; instead he considered how to make machines automatically perform computations based on given algorithms. The machine model he envisioned later became known as *Turing Machine*. In Turing's initial imagination, as long as an effective algorithm is given, a machine can always be constructed for that algorithm, it can (and can only) execute the given algorithm. Later, he further developed the idea of the *Universal Turing Machine*; the machine is universal, by providing it with an algorithm represented with symbols on a paper tape, it can execute the given algorithm, which is quite similar to modern general-purpose computers. Based on the Turing machine, Turing proposed his *Turing Thesis*, which can be summarized in the following points:

- Only an effective algorithm can be implemented with a Turing machine, which means every step must be clearly defined and can be effectively performed, and the process will halt after a finite number of steps (for example, take finite terms in a Taylor series). Clearly, the effective algorithm he mentioned is the same as Church's valid λ function.

- As long as there is an effective algorithm, there is always a Turing machine that can be built which can implement the algorithm.

- Without an algorithm, or if the execution of the algorithm does not terminate, the problem is not Turing machine computable, or not Turing computable.

In fact, even before Turing machine was proposed, there was already a machine that was capable of doing computations; that was Babbage's "Difference Engine" (which didn't really get built), for which Ada (the daughter of poet Byron) even wrote programs to be executed on that machine. However, Babbage's machine was only for numeric computations, whereas Turing machines were not limited to numeric, but also for logical and symbolic computations. Actually, Turing machines were based on symbol processing, numerical computations were implemented through symbol processing, just like in modern digital computers. So, Babbage's machine was merely an arithmetic machine, while Turing machines were logical machines, and this distinction is significant and critical.

Apparently, the essence of the *Church thesis* and *Turing thesis* are the same, and thus people combined them to be the *Church-Turing thesis*. Although their logical arguments are obscure and difficult to understand, it essentially comes down to the same idea, namely: computation is possible only if an effective algorithm is given. Therefore, they are logically equivalent, but the *Church thesis* pertains to humans, while the *Turing thesis* raises the issue of replacing humans with machines for computation, which carries greater significance.

Despite the profundity and obscurity of their arguments and narratives, their core idea is simple: a computation is merely the execution of an algorithm. There are some problems which are inherently incomputable, or "undecidable"; for which Wikipedia listed over forty such problems in its webpage "List of undecidable problems". These problems are definitely not computable on machines, and they

 doi:10.20944/preprints202509.2365.v1

are unsolvable in a human's brain either. However, even for a problem which is not inherently incomputable, it is still practically incomputable on any machine if no algorithm is given, it is still not Turing computable.

On the other hand, since any effective algorithm can be implemented by a Turing machine, if other machines can also implement this algorithm, then they are logically and functionally equivalent to the Turing machine, even though their performance may be significantly better. In contrast, if no algorithm is given, then no machine can carry out any computation, so it's again equivalent to the Turing machine. In other words, there is no computation that a Turing machine cannot perform but another machine can. Therefore, in this sense, it can be said that any machine capable of computation is equivalent to a Turing machine.

With the existence of Turing machines, people naturally begin to ask the question: if machines can compute like humans, then can machines *think* like humans? The *Turing thesis* has actually answered this question: if the process of thinking, or the activity in human's brain, can be specified with an algorithm, then yes; otherwise, if there is no algorithm, then no.

Unfortunately, some intelligent activities in human's brain do not have an algorithm yet so far, not even a guessed algorithm; and thus those activities cannot be imitated on machines. For example, dreaming, can you determine the exact steps to take, so that under a certain set of external conditions, you will certainly have a specific type of dream? It seems human's dreaming is probably not a computation. The Indian mathematical prodigy Ramanujan said some of his peculiar mathematical expressions came to him in dreams; can others see these expressions in their dreams as well? Another example, Archimedes, while lying in a bathtub, suddenly discovered the principle of buoyant force. Who can describe what the algorithm, or the process, was in Archimedes' brain at that moment? Therefore, some people say that "greatness cannot be planned", because no one can provide an algorithm that enables a person to achieve greatness. So, it is evident that making machines identical to humans in terms of intelligence, so that a machine can do everything the human brain can, namely achieving AGI, is unrealistic. Of course, some things that cannot be done now do not prevent them from happening in the future, and some algorithms that do not exist now might be invented in the years to come, but that would be a long historical process and not something that can be achieved in the short term. For this reason, it is necessary for us to explore how the activities in the human brain are carried out.

§2 One Brain, Two Systems

The disciplines studying the human brain are mainly in two fields: one is physiology (including research in neurology), and the other is psychology. Physiology concerns the physiological structure of the brain from an anatomical perspective, but this does not resolve questions about the thinking process, as it is now still very hard to capture and decode signals in the brain, as we can do for testing and repairing computer hardware. Although there are researches on brain-computer interfaces, that is only related to the output stage of the brain and does not involve the thinking process; while what we need to explore are algorithms describing the thinking and consciousness processes. Therefore, research on the thinking and consciousness process primarily lies within psychology.

In psychological research, Freud's studies on the subconscious played an important role, but his research (especially that related to the interpretation of dreams) was far from empirical, so later psychological studies were mainly guided by *behaviorism*. Behaviorist researches start with the observed behaviors of humans and animals to infer the activities occurring in the brain, and all experiments must be supported by observed behaviors. Fundamentally, this is right; but in practice most experiments in this area are difficult to carry out (though some are simple), leading to slow progress in researches. People realized that bolder and more active exploration in the psychology field was needed, so starting in the 1950s, research on *cognitive psychology* emerged, encouraging people to explore more actively and boldly, so that more hypotheses and theories can be proposed.

Nobel Prize winner Daniel Kahneman talked about human cognition and the human brain in his book "*Thinking, Fast and Slow*", saying that there are two systems, System 1 and System 2, in the

 doi:10.20944/preprints202509.2365.v1

human brain. System 1 is responsible for fast, instinctive, and intuitive responses and behaviors without reasoning; while System 2 deals with thinking that requires slow logical reasoning. The concepts he cited originate from a theory known as the "*Two Minds Hypothesis*", primarily advocated by British psychology professor Jonathan Evans and American psychology professor Keith Stanovich, and others; they held an academic conference at the University of Cambridge in 2006, resulting in a collection of papers titled "*In Two Minds*". However, perspectives on this topic can actually be traced back far earlier. The *Two Minds Hypothesis* posits that there are two systems in the human brain, System 1 and System 2, which align with our daily observations and personal experiences.

Below is a table for comparison, provided by Professor Evans in the collection "*In Two Minds*". This table is not the only one of its kind, but these tables are generally similar.

System 1	System 2
Evolutionarily old	Evolutionarily recent
Unconscious, Preconscious	Conscious
Shared with animals	Uniquely (distinctively) human
Implicit knowledge	Explicit knowledge
Automatic	Controlled
Fast	Slow
Parallel	Sequential
High capacity	Low capacity
Intuitive	Reflective
Contextualized	Abstract
Pragmatic	Logical
Associative	Rule-based
Independent of general intelligence	Linked to general intelligence

Many of the assertions listed in this table can be observed in our daily lives. For example, when avoiding danger, if some heavy stuff is thrown at you, you will instinctively dodge it without having time to calculate, and your reaction speed is extremely fast. Afterwards, you might want to analyze its "trajectory", which is a much slower computation. This illustrates the difference between Intuitive and Reflective. Similarly, humans have habitual and unthinking (Pragmatic) reactions to many things, and these reactions occur automatically (Automatic). But sometimes it is necessary to calm down (Controlled) in order to engage in logical reasoning (Logical); or as the saying goes, "think twice" before acting. Specifically, cognition and thought arising from System 1 are Associative, which only involves "end-to-end" associations and connections; while cognition and thought from System 2 are "Rule-based", involving a lot of "if-then-else" reasoning, mostly implementing a syllogistic logical process. It is important to note that whether the reasoning is based on rules is a clear distinction between the two systems. For System 2 activities, we can easily figure out related algorithms, but for system 1 we don't even know how the responses are made; you can guess, but it might be wrong.

System 2 and System 1, or intuition and reasoning, are at two different levels, with System 2 in the higher level and System 1 in the lower level. People first perceive stimuli from the external world by intuition, which causes the response of System 1; and then System 1 either submits the result to System 2 for reasoning, by awakening System 2; or takes actions directly without bothering System 2; or it simply gives up. In fact, in most cases, System 1 will not submit its result to System 2; or System 2 will not take it, as it is relatively lazy.

The strength of System 2 activities varies among individuals. This is partly due to biological diversity and partly related to the acquired education and training. For instance, when we see two triangles, we could immediately recognize that they are similar to each other; however, we need to ascend to System 2 to make sure, to prove it; yet we of course will not try to prove the similarity of every pair of triangles we see, if they look similar.

 doi:10.20944/preprints202509.2365.v1

It is often said that human thinking includes “figurative thinking” (or “imaginative thinking”) and “logical thinking”, where “figurative thinking” is based on shape, color, voice, smell, situation, and so forth; it is more akin to System 1, but is not necessarily fully confined to System 1; while “logical thinking” clearly belongs to System 2.

In reality, the interaction between System 2 and System 1 might be very complex. Gary Marcus, an American psychologist and AI scholar, has repeatedly mentioned a phenomenon where children, when using irregular verbs like *eat*, will often say *eated*, following the rule for the regular past tense; then they realize this is an irregular verb and correct themselves to say *ate*. Clearly, when saying *eated*, they are applying the rule, and then, upon realizing it's an irregular verb, they need to use a higher rule to correct themselves to say *ate*. Apparently, this is a complex process; in this process the mouth is just an executor controlled by System 1.

Interestingly, they believe that in the history of human evolution, System 1 existed first, then System 2. Moreover, they think System 2 is unique to humans, and the evolution of System 2 didn’t happen for other animals. This may provoke some controversy, as perhaps animals also possess some level of rule-based reasoning; however, even if animals do have System 2, it is certainly much weaker than that of humans. It is precisely the existence of a robust System 2 that sets humans apart from animals in cognitive capability.

Since System 2 operates in a logical and rule-based manner, it must be symbol-based. In fact, proficiency in symbolic reasoning greatly impacts humans’ cognitive ability. In fact, a person can develop significant intelligence even without attending any school, by various forms of associative learning and training in daily life, and there will also be some reasoning ability due to the existence of human System 2 in their brains. But if they attend schools, the school education is predominantly focused on symbols (the textbook itself is in symbols), which will greatly enhance their System 2, and they will have an even better reasoning capability. Conversely, if an individual's System 2 has been greatly strengthened, but their System 1 is too weak, they may be “clumsy” and sluggish in daily life, which is also quite common.

Based on the observed behaviors, System 1's response to external stimuli should be distributed and massively parallel, because the human body can respond to various external stimuli simultaneously; what you hear and what you see can occur at the same time. In contrast, System 2 should be concentrated, that is because: at first it is difficult to imagine one single step of logical reasoning, the applying of one simple rule of if-then-else can be distributed; and second, in fact, we know the human brain can only focus on one issue at a time, you cannot be pondering on two different issues exactly at the same time.

In addition, there is a special type of thinking known as *Hypothetical Thinking*, as referred to by Evans. When humans engage in innovation or run into a totally new situation, they need to employ hypothetical thinking, as it requires considering potential scenarios that may occur under various environments and conditions, akin to what we commonly call a “thought experiment”, which is clearly a function of System 2.

So far, there is no anatomical evidence regarding the physical existence of the two systems, nor can we specify which part of the human brain implements System 2. Although System 1 seems closer to the cerebellum, which controls human body’s motion and balance, it cannot be determined if it is solely located in the cerebellum. In fact, the anatomy of the human brain suggests that the entire brain functions like a network of neurons, without a specific area identifiable for concentrated thinking (though images of the hippocampus in Alzheimer's patients' brains might indicate that this region could be where System 2 resides). This is why psychologists refer to System 1 and System 2, rather than directly specifying locations such as the parietal lobe or the cerebellum. However, regardless of whether System 1 and System 2 exist in different areas of the brain, it is clearly observable and self-perceptible that there are two distinct types of thinking processes in the human brain. One is rule-based reasoning, while the other is more akin to stress responses, occurring without deliberate thought but involving end-to-end associations. In fact, some psychologists do refer to these as Type 1 and Type 2, rather than System 1 and System 2.

 doi:10.20944/preprints202509.2365.v1

But, if System 2 is based on rule-based reasoning, then where do these rules come from? It seems some rules are inherited, these are human instincts; more are coming from education; others, perhaps more numerous, have arisen from self-learning, summarized and abstracted from the stimuli perceived by System 1 and its responses. This ability itself should be a part of System 2 in the brain, since that should involve rule-based reasoning; while a human's initial raw System 2 is inherited, which is a result of evolution and genetics. Some psychologists argue this belongs to a third system, but that doesn't matter. Conversely, having more rules will strengthen System 2's capacity and enhance its ability to guide and correct System 1's behavior.

In fact, *learning* in true sense is not simply about memorizing a lot of knowledge, which is merely end-to-end correlation (Associative). Instead, it depends on the ability to abstract reasoning rules for System 2 to use. With this ability, System 1 and System 2 can promote and complement each other, and improve together.

The theoretical approach of the two-system hypothesis offers significant insights into both psychology and artificial intelligence. To make machines fully imitate the human brain, we need to simulate both System 1 and System 2; missing either one distances us from achieving AGI, the full brown artificial general intelligence. However, unfortunately, the process of summarizing and abstracting rules from the System 1 responses is precisely what cannot be expressed in an algorithm, and thus cannot be computed by a Turing machine, because so far we don't know what the process is at all.

§3 Connectionism and Symbolism

In the history of artificial intelligence, the rivalry between the connectionist and symbolic logic schools, and their respective technical approaches, is already well-known, so I won't elaborate on that here. Generally speaking, people from both schools intended to achieve artificial intelligence as soon as possible, but they did not believe the theories and technical approaches of the other party can play a significant role in the "Imitation Game".

Clearly, symbolic logic aimed to realize the System 2 in the human brain on machines, even though the term did not exist at that time, and rule-based logical reasoning is precisely the fundamental feature of System 2. They believed that logic encompassed the entire human intelligence, and every intellectual behavior could be reduced to logical reasoning. Therefore, they focused all their efforts on symbolic reasoning; they rejected connectionism, which was still in its infancy at the time. Their greatest achievement was the realization of expert systems, with a rule base as the core of an expert system, possibly supplemented by a knowledge base providing basic facts.

However, without System 1, their rule base was like a water stream without a source; the rules in their rule base were provided and filled in manually by humans. When the number of rules became very large, the rule base maintenance turned into a nightmare. In fact, not to mention the source of the rules, so far we don't even have a good algorithm for rule bass maintenance. In the human brain, when new rules come in, some existing related rules may be negated, or their "weights" may be degraded, or some additional boundary conditions may be applied, but there seems to be no clear and effective algorithm was invented for this. The failure of the so-called "fifth-generation computer" is essentially due to this reason. A System 2 without the support from System 1 is destined to fail. Of course, there was no understanding of this aspect at that time.

Therefore, it is just natural and inevitable that the focus of artificial intelligence research has shifted to connectionism. This is because connectionism, or the neural networks, is essentially implementing a computerized System 1, it realizes the functions of System 1, covering the more fundamental human intelligence activities, which is particularly evident and prominent in several aspects:

- Structurally, neural networks are very similar to the neural system in the brain, achieving large-scale parallel processing. Each node in the network, a tiny computing unit, imitates a neural cell in the human brain.

 doi:10.20944/preprints202509.2365.v1

- A deep learning neural network is like a state machine, a response is given whenever it perceives an input stimulus, at a speed that far exceeds the speed of symbolic processing. But this response is close to a subconscious and instinctive reaction, including response to visual and auditory stimuli, which do not involve logical reasoning. For example, when we hear a familiar person's voice, we immediately recognize who it is, without having to analyze the frequency characteristics of the audio signal, which might be buried in noise.

- The knowledge learned by deep learning is distributed among a large number of neural parameters (weights), and the knowledge memorized in these parameters is all about end-to-end correlations, without involving the causality steps between the two ends.

- This knowledge is "flatly" distributed across a large number of neural parameters, rather than being concentrated in databases and knowledge bases; it does not constitute a knowledge hierarchy.

The learning in neural networks is totally empirical and based on the relevance of things; such an end-to-end learning is essentially no different from Pavlov's conditional reflexing.

Apparently, whether intentionally or unintentionally, what a neural network simulates is a System 1 in the human brain, and System 1 only. So far, neural network systems, including deep learning and large language models, have remained at the System 1 level only, without the existence of a System 2, and do not contain explicit rule-based logical reasoning. However, the samples used for training are all derived from human behaviors, which may contain some outputs of people's System 2 intelligent activities, but they are only memorized as separate frozen end-to-end knowledge and do not provide the ability to perform logical reasoning on new external stimuli.

The old day symbolist school focused on System 2 only and neglected System 1, it lost its foundation and roots, becoming like water without a source, and thus it is destined to fail. However, in the reverse direction, if we now focus on System 1 only, without System 2, we will be lacking the important ability for reasoning and judgment.

In particular, one major function of System 2 is to correct System 1; without the corrections from System 2, malfunctions in System 1 cannot be suppressed, and that might be the source of many "hallucination" issues in large language models. At the very least, without the logical reasoning of System 2, the output results from neural networks can only be accepted without enough confidence. This brings issues to the construction and implementation of many "mission-critical" systems. Apparently, current large language models also produce results that seem to contain certain level of reasoning, but those results are still based on empiricism, merely because "everyone says that" or "someone said that", rather than stemming from rule-based logical reasoning. People can trace and identify errors in rule-based logical reasoning, step by step, maybe trace back to the wrong rules, but with "everyone says that", we cannot distinguish truth from falsehood at all.

Since the human brain contains both Systems, if only System 1 is implemented in machines, then obviously an important part is missing, compared to the human brain; so how can we talk about AGI?

On the other hand, deep learning is based on massive samples, with the so-called "scaling law"; however, once nearly all of accumulated human knowledge has been used for training, how much more room for improvement can we still have? The number of neurons, that is, the number of distributed parameters, although we don't know when it will saturate, and when the scaling law reaches its end, but sooner or later it will. Without a System 2, no matter how strong a System 1 is, it still cannot provide the System 2 functionalities.

As for *reinforcement learning*, it is a little bit closer to System 2 than deep learning, but it is still not a part of System 2. The knowledge and cognition learned through reinforcement learning are essentially purely empirical as well. This is because, if one specific action gets rewarded this time, and gets rewarded again next time, then it is encouraged to do the same afterwards; this actually is

 doi:10.20944/preprints202509.2365.v1

not fundamentally different from Pavlov's conditional reflex, and not fundamentally different from neuron networks.

Of course, reinforcement learning is necessary, as it addresses the motivation issue of learning. But reinforcement learning is insufficient; it cannot replace the existence and the role of System 2.

Whether it is deep learning or reinforcement learning, neither provides the functions and roles of System 2, let alone hypothetical thinking. As for the so-called "chain of thinking (CoT)" and "prompt engineering" in large language models, which are essentially just attempts to guide a mechanism somewhat resembling hypothetical thinking by the inquirer (the user), not the machine. However, firstly, that is provided by the user, not spontaneously by the system; secondly, what can be provided in that way is merely using experienced scenarios as hypotheses; whereas hypothetical thinking in System 2 relates to innovation, to situations that have never happened before. Although it is not impossible for CoT to trigger some innovation in System 1 by chance (old experiences could have new combinations), the probability of its happening will be far lower than the hypothetical thinking in System 2, as rule-based logical reasoning can potentially lead to unprecedented results. Thus, as long as System 1 and System 2 have not been integrated together, AI will remain far from the intelligence of the human brain, and that can not be AGI.

However, there are still many issues to be solved if we intend to implement both System 1 and System 2 together in a machine. While we know our thinking can freely go back and forth between the two systems, we don't know yet how the two systems interact with each other, which means no concrete algorithms can be proposed. Of course, we can speculate and experiment; for example, we can guess in what kind of circumstances System 1 will submit a result to System 2, and how System 2 can guide, suppress, and correct the activities of System 1 in return.

While for this problem we can speculate and experiment, another problem seems to be hopeless at the moment, which is how to abstract new rules from the System 1 activities and incorporate them into the rule base in System 2. There is no algorithm for this yet, and we cannot even see a prospect for the algorithms to emerge. All these issues lead to the conclusion that it is impossible to achieve AGI in the near future, not to mention before 2030.

§4 Turing Test

When Turing proposed his Turing machine and Turing thesis in 1936, the concept of "artificial intelligence" as we know it did not exist yet; however, it is clear that Turing was already contemplating whether we can use machines to imitate the human brain. On the other hand, Turing never claimed that a second brain could be built outside the human body. What he advocated was merely for machines to "*imitate*" the brain. He actually never proposed the requirement of AGI (Artificial General Intelligence); instead, he explored whether a machine's imitation of the human brain could be convincing enough for people to believe that the machine had achieved the intelligence equivalent to that of humans. Here, the word *imitate* refers to external behaviors mimicking, as we cannot ascertain the actual processes occurring in the human brain, and we can only guess and hypothesize, so that algorithms can be proposed for machines to execute, and then observe whether the results can convince people to believe the both are the same. Nevertheless, even though these are just guesses and hypotheses, some of their correctness is evidently observable, such as the existence of System 2.

Whether a machine has achieved the intelligence equivalent to that of humans requires an objective judgment method. Of course, the "judgment" is actually a human's judgment, which naturally carries subjectivity; however, the methods used should be objective enough so that different people can arrive at the same judgment. In practice, such an objective method can only be a comparative method, and factors other than intelligence must be excluded from the comparisons. We know in advance that this involves a machine imitating the human brain, so we must hide all the physical, chemical, and biological characteristics from comparison. The "test" we can conduct can only be behavioral, by observing whether the reactions of machines and human brains are the same when subjected to the same stimuli. This is undoubtedly a behaviorist approach, which judges solely

 doi:10.20944/preprints202509.2365.v1

based on external behavioral observations; the so-called test is merely a comparison and differentiation of behaviors. Of course, we cannot expect exactly identical reactions, as even reactions from two people could still differ when they are experiencing the same stimulus simultaneously.

The *Turing Test* compares the behaviors and responses of a machine and a specific person when subjected to the same stimuli. The method of the test is roughly as follows: a person and the machine to be tested are both hidden in a closed room, or behind a heavy curtain, so that outsiders cannot see them, nor hear them; both can only communicate with the outside world through text, for example, by keyboard and printer. The tester then continuously poses various questions to both and tries to determine which one is human and which one is a machine, based on their answers. If the answers from the machine are flawless, despite many tricky or not tricky questions being posted by the tester, making him unable to distinguish between the human and the machine, then it is said that the machine has attained the same intelligence as the human.

The principle of the Turing test is actually quite simple. There is a saying in the West: "If it looks like a duck, swims like a duck, and quacks like a duck, then it is a duck". Of course, what we are comparing with is not a duck, and so we cannot let the tester see what it looks like and how it swims, or hear it quacking. However, the principle of judging based on their specific characteristics is the same. Moreover, this is not an individual interrogation held separately, but is a direct and immediate comparison against a real human, asking them to answer the same question. The tester can do his best to devise various questions as tricky as possible, trying to identify the machine based on any subtle difference between the answers. Since the only communication is via typing, it eliminates all possible interference from physical and biological characteristics, strictly limiting the comparison to their "intelligence". Under such conditions, if no obvious differences in responses can be detected, and answers cannot be identified as coming from a machine rather than a human being, then the tester has to conclude that the two are the same, and thus we believe that the machine possesses the intelligence equivalent to that of humans. Of course, the tester, or the questioner, should be an intellectually sound person, or it can actually be a group of people.

Alternatively, we can understand the *Turing Test* from a different perspective. Suppose we want to define a set of objects that possess human intelligence, including both real humans and machines. We start with a real person in this set to serve as a comparison template. Now, for each given object, ask you to judge whether it belongs to this set, by comparison with the template; but you cannot see the object, and the only means of communication is text. To judge whether an element belongs to a certain set, it must meet the criteria of that set, but it is hard to provide a full list of the criteria, so a template is provided for comparison. Now, the criterion is whether you can feel that their behaviors are different; if you cannot distinguish who is the real human serving as the template and who is the object under test, then you have to admit this object belongs to the set, and this is the *Turing Test*.

Therefore, the design of the *Turing Test* is impeccable in its principle.

However, Turing did not provide specific details at the time, such as how long each test should last, how many questions could be asked, whether there were requirements for the speed of responses, and so on. But here, the most important point is: what kind of person should be used as a comparison template? There are significant differences in human intelligence; comparing someone to Einstein is completely different from comparing to an illiterate layperson. Each specific test is conducted between a specific person and a specific machine, so strictly speaking, if a machine passes the Turing Test, we can only say that this machine (the artificial intelligence) has the same intelligence as that particular individual, not the whole of humanity. In addition, if we broadly ask whether a machine has the same intelligence as humans, the comparison should be made between the strongest current artificial intelligence and the best human groups.

There are some disputes against the Turing Test, the major one was presented by the philosopher Searle in the so-called "Chinese Room" argument as follows: Suppose there is a person hiding in a sealed room, who can only communicate with the outside world by messages printed on paper. Every time, the external tester requests a translation from English to Chinese. Even if every translation is correct, we cannot conclude that the individual in the room really understands Chinese, because he

 doi:10.20944/preprints202509.2365.v1

is probably just relying on a dictionary to map the words in an end-to-end conversion, akin to looking up a table, while he actually doesn't comprehend Chinese at all. Searle argues that a machine passed the *Turing Test*, like the person in the "Chinese Room", could still lack genuine human intelligence.

Actually, Searle's question is not entirely the same as what Turing proposed. The *Turing Test* views the machine under test as a black box, without questioning its internal structure or processes; what is examined is only its external behavior. As long as its external behavior is indistinguishable from that of a human, it is judged to exhibit a human-equivalent intelligence. However, Searle takes it a step further, proposing a higher standard, which requires not only observing external behavior but also checking its internal process. In fact, this aligns perfectly with the current development of artificial intelligence. Today's large language models, based on neural networks and deep learning, indeed resemble the person in the "Chinese Room". Searle posed this question in the 1980s, as if he were foreseeing the development of modern language models. Searle's requirement, for the person in the room to have a genuine understanding of input messages, necessitates not only System 1 but also System 2, which can provide a logical explanation for the outputs, so this is clearly a higher requirement. On the other hand, merely asking for translation is not a good *Turing Test* at all; it relaxed the testing conditions, because translation often can be achieved through a table lookup method. But it is not easy to identify either, even if a "why?" is added to every question, the answers could still seem valid, as the same questions could have been asked and answered before by real people, and the answers to these questions have appeared in the training materials; the "table" in the "Chinese Room" is not just a dictionary, it keeps record of all the training samples.

In fact, the true distinction between machines and humans lies in creativity, a point that was noted long before Turing by Lady Lovelace, the daughter of the poet Byron, who was hailed as the first programmer in human history. Based on her understanding of Babbage's "Analytical Engine", she wrote:

Computers can't create anything. For creation requires, minimally, originating something. But computers originate nothing; they merely do that which we order them, via programs, to do. But if computers can't create/originate, then they can't be intelligent, thinking beings.

Lovelace's demands are higher than Turing's; she requires machines to be as creative as humans. However, human creativity involves randomness, and thus cannot be asked to demonstrate during testing; you cannot ask the human and the machine behind the curtains each to present a patent application during a Turing test. In fact, if Lovelace's requirements can be met and machines can be creative, then we would have essentially achieved AGI, reaching the goal that artificial intelligence will be able to do whatever we (humans) can. However, we know it is impossible, at least in the near future.

In addition, creativity manifests in different aspects. It is possible for machines to exhibit some creativity in literature and art, for example, you can ask it to write an essay for a given topic in a Turing test, or compose a poem, or write a performing arts script, and so forth. Current large language models like ChatGPT should be quite adept in this aspect. This is because artistic imaginative thinking does not require strict logic, and its acceptance largely depends on the subjective feeling of the readers and viewers. In contrast, creativity in science and technology is totally different. As said above, a Turing test clearly cannot require both parties to each provide a valid patent application. Technical inventions are strictly based on reasoning and logic (and hypothetical thinking), representing what humans can do (but no guarantee), which machines generally cannot do yet. For inventions and creations, so far we don't have any algorithm, and we don't know what was happening in human's brain. If someday invention and creation could have an effective algorithm, then "greatness" *can* be planned.

As the title of the book "*Why Greatness Cannot Be Planned*" indicates, greatness, or great inventions and creations, cannot happen according to a plan, which is impossible even among humans, let alone machines. If inventions could be produced according to a plan by machines, then it would become possible to mass-produce machines that could create all possible inventions. This is philosophically absurd. But why can't inventions be produced according to a plan? We will discuss

 doi:10.20944/preprints202509.2365.v1

that later, which is simply because there is no algorithm for producing inventions, while machines can only execute given algorithms. Only activities having effective algorithms are "Turing machine computable", and only processes that are Turing machine computable can be executed by machines. Inventions and creations are activities that could occur in the human brain, but no algorithm can be provided for it, at least for now. Astonishingly, Lady Lovelace keenly recognized this issue one hundred eighty years ago.

Although we cannot propose a better testing method than the *Turing Test*, it is not spotless. Indeed, the "Chinese Room" problem does exist to some extent; machines trained on massive samples based on deep learning has accumulated a wealth of experience; although this experience is purely end-to-end mapping and does not provide reasoning for conclusions, it is not impossible for it to "fool" the questioner in the test.

Actually, the difference in intelligence between machines and humans is primarily concentrated in a portion of brain activities that cannot be described by algorithms, but whether a human's questioning can elicit and seize this difference is also a question. It is akin to a courtroom situation; if a lawyer's questioning does not hit the mark, it will be a failure. However, there is truly no better method than the *Turing Test*. Therefore, AI that passes the Turing Test is merely "considered" to possess human-equivalent intelligence, but this does not imply that it truly has intelligence exactly the same as that of humans, nor does it indicate that it has achieved "Artificial General Intelligence" or AGI.

From the proposal of the *Turing Machine* and *Turing's thesis* to the introduction of the *Turing Test*, there is a continuous thread that indicates Turing was always contemplating the extent to which machines could achieve human intelligence. The *Turing Machine* and *Turing's thesis* address the question of what kind of brain activity can be simulated by machines, while the *Turing Test* focuses on how to judge the results.

§5 Three Approaches Toward AI

Turing's idea is about how machines can mimic the human brain, and he clearly pointed out that machines can only imitate those activities in the human brain that can be expressed with algorithms, thus what can be realized by machines is only a subset of the human brain's activities. Obviously, the first approach to implement AI is to build Turing machines for these effective algorithms, make each Turing machine as an individual function or module, then integrate them together in application software products, just like other software programs did. Old-day symbolic AI algorithms, including rule-based reasoning, are all implemented in this way.

In the history of the artificial intelligence, many algorithms for symbolic logic have been proposed and implemented; these are all in System 2. Intelligent activities in System 2 are relatively easier to have algorithms, since System 2 is rule-based, and each computable function, or algorithm, provides the rules for implementing the mapping. The applying of the rules is under the control of some higher order rules, but there are not so many higher order rules for logical reasoning, although theoretically there can be an infinite variety of specific lower order rules in the rule base, the higher order rules for applying these rules are limited. For example, the so-called logical "syllogism" works in its specific way, with a few rules, but a syllogism can be applied to countless specific lower order rules.

However, at least for now, there are many brain activities for which we do not have an algorithm yet, and it is hard to predict the future. In fact, activities in System 1 are difficult to have algorithms, because the exact mechanisms are unknown so far. For example, what reaction will be made, and how it is made, when a person feels pain in his skin? Can we write down a concrete algorithm for this phenomenon?

For these brain activities without specific algorithms, although a specific Turing machine cannot be built, as long as we are only caring about their effects or behaviors, some workaround still can be built with some general algorithms, for some of these activities (but not for all activities). Actually, some algorithms are universal and scalable, and a single such algorithm can be used to implement

 doi:10.20944/preprints202509.2365.v1

countless function mappings. The simplest way is to build a database for stimulus-reaction pairs, as key-value pairs, mimicking the way humans are using paper notebooks. Strictly speaking, each time we add or modify a record in the database, the mapping from input to output changes. However, the algorithms for inserting, querying, and so forth on the database remain unchanged, and these algorithms can certainly be implemented by machines. In fact, this is a similar approach used for the rule base, but the rule base is a list of "if-then-else" actions, while this is simply an end-to-end mapping, a knowledge base. By doing this, we are actually using System 2 algorithms to mimic System 1 activities.

Assuming there is a GPA registration form that lists each student's GPA by their student ID:

$$GPA(x) = \begin{cases} 3.3 & x = 20220103 \\ 3{,}7 & x = 20220089 \\ & \dots \\ 3.8 & x = 20220231 \\ 3.2 & x = 20220483 \end{cases}$$

This registration form GPA is a mapping from a student's ID to her academic performance, so this is a function *GPA*(). Now, let's assume that two students with IDs 20220089 and 20220483 have transferred, while three new students have joined. As a result, the particular mapping has changed, and thus the specific definition of the function *GPA*() has changed; in fact, it has become a new function *GPA'*(). However, the algorithms for querying, inserting, updating, and deleting on this table have not changed; they remain the same.

In this way, under the premise of keeping the algorithms unchanged, a function can "*learn*" from the data changes and alter its specific mapping. This is true for databases as well as knowledge bases. Just as instructions are data, knowledge is also data. The specific knowledge in a knowledge base can vary, and the number of knowledge items can be large or small, but the algorithms that maintain the knowledge records do not change.

However, the above mentioned method only applies to enumerative definitions on input-output correspondence, or key-value pairs; it does not involve any specific computation process (such as Taylor series). The process of "learning from data" relies on the changes in the quantity and contents of the key-value pairs included in the mapping relationship, which updates the database. In database, only one key-value pair is retained for the same key name, and the old key-value pair gets overwritten.

So, in a sense, operations like inserting and content updating for a database can also be seen as "supervised" training. Of course, mechanisms providing similar functionalities are not limited to databases; in fact, neural networks are also one of them.

However, such enumerative function definitions can only be used for finite sets, meaning that their domain is a finite set, and practically it would be a small set. If the domain is an infinite set, or even though it is a finite set, it is a very large set, then such mechanisms are not applicable, or not appropriate. But mechanisms and algorithms for such mappings indeed exist, such as regression, particularly linear regression.

Regression has effective algorithms that can be implemented on machines. By providing a large number of input-output pair samples for the regression algorithm, a general mapping relationship from input to output can be formed, that is, the curve of the function can be formed. This curve changes with the arrival of more samples, striving to approximate the functional relationship implied in the given samples, making the curve closely resemble the underlying function curve. So, this serves the purpose of "curve fitting", it alters the specific mapping at the specific points, without changing the regression algorithm, achieving the effect of "learning from data". This is essentially a supervised training process. Once the training is complete, given an input value, it outputs a result calculated according to the fitted function curve. This result may not accurately match the actual existence, but it minimizes the error. Or, in other words, it computes based on the fitted curve to obtain a value with the highest probability to be correct, with the least error range.

 doi:10.20944/preprints202509.2365.v1

In fact, the connectionist approach, specifically the algorithms developed based on the neural network model and deep learning, whether consciously or unconsciously, overlaps significantly with the algorithms of linear regression. Essentially, deep learning is a kind of linear regression (including a part of nonlinear regression). Compared to the human brain, linear regression, or deep learning (including the large language models, LLMs), its functionalities and the role it plays clearly belong to System 1, but the regression algorithms implementing these functionalities are actually from the rule-based System 2, as we don't even know how these functionalities are really implemented in human System 1.

With such a linear regression algorithm, end-to-end mapping becomes straightforward, but of course, there will be errors. In most cases, these errors are minimized and will not exceed the extent perceivable by the average user. However, for some specific mappings, the errors can be significant, which is what we commonly refer to as "hallucinations". Getting that, if we users know it is a hallucination, we might be able to adjust the query prompt and provide different key values, so that it may yield another answer with less hallucination, closer to the expected value. However, to truly address the issue of hallucinations, we need to go up to System 2, allowing System 2 to make judgments and corrections based on its rules. So, here the user's "expected value" essentially comes from the human's System 2, rather than the AI, but the particular user's System 2 may not necessarily have the relevant rules and knowledge either. On the other hand, whether this error is easily perceived also relates to the specific content. Generally, errors in scientific and engineering contents are sensitive and could be harmful or even dangerous; however, for artistic contents such as images, sounds, novels, performing scripts, etc., it is less sensitive because they involve users' subjective feelings. This also explains why deep learning is well-suited for AIGC (AI Generated Contents), as the content generated is primarily artistic, where "messing things up" is not a big problem; having Khrushchev smoking Churchill's pipe wouldn't be viewed as an issue.

§6 Interactions Between the 2 Systems

According to the cognitive psychology hypothesis of "one brain, two systems", System 1 is prone to mistakes and requires corrections from System 2. System 1 also easily gets "excited" and needs suppression from System 2. Moreover, as long as a person is awake, System 1 operates around the clock, constantly receiving and perceiving external stimuli and making responses. However, System 2 is different; it has a high degree of independence and only intervenes when activated by System 1 (or System 2 itself) and it is "willing" to engage. While the "two systems" hypothesis matches people's observations and experiences, there is currently no definitive understanding about the details of how the two systems interact, and, of course, there are no specific algorithms.

Technically, it's not a problem to stack an old expert system with today's LLMs together; for example, using the LLM as the front end of the entire system, playing the role of System 1, while using the expert system as the back end, playing the role of System 2. After all, those old expert systems are now just small modules. Nevertheless, how to make them interact and combine is a problem. For instance, under what circumstances should the front-end results bypass the back end to be output directly, since System 2's reaction would be too slow; in what cases should it go through the back end; and should it come back to the front end after passing through the back end? All these are questions that need to be explored. However, these issues are actually not too difficult, because these are all hypothesis and verification issues, and algorithms for these hypotheses are generally not hard to figure out. If one algorithm doesn't work, just try another.

The real challenge comes from abstraction. How to abstract reasoning rules from the external stimuli perceived by System 1 and its responses, this is the key. Of course, once rules are abstracted, we need to fill them into the rule base of the expert system, and there is also the problem of how to maintain the rule base, but that is easier.

Generally speaking, a reasoning rule is a proposition, which may be a universal proposition or an existential proposition. If a proposition has been put forward, then it becomes a matter of theorem proving, and perhaps mathematical induction can play a significant role in that. However, the key

 doi:10.20944/preprints202509.2365.v1

issue is the formulation and raising of the proposition, namely, how to abstract a proposition from the information provided by System 1. In this sense, the raising of a proposition is more important than the proof of the proposition (and thus becomes a rule).

For example, suppose we have an arbitrary integer, such as 123. If we reverse its order, it becomes 321. Then, if we subtract the two numbers, we will find that the resulting difference must be divisible by 9. For example, 321 - 123 = 198, and 198 is clearly divisible by 9; the same goes for all integers, for example 13579 and 97531, so this holds true for integers of any length. This is a universal proposition. Having observed and realized this proposition, it's relatively easy to prove it using mathematical induction, but discovering and proposing such a proposition is a different matter. Even if an AI software capable of proposing propositions in this pattern is available, it cannot propose propositions in other patterns, and there can be infinitely many such patterns. Only with such propositions, rules can be created and added to the System 2 rule base, so that they can be used in System 2 reasoning. Without new rules, System 2 becomes water without a source.

Unfortunately, while the existence of System 2 is indeed necessary, algorithms in this area are unlikely to emerge in the near future. So we must find a way to address the problem of the rule sources. Actually, in the foreseeable future, it seems that we will have to rely on manual setup and update; that is, using humans' System 2 to find and figure out the rules, then add them to the rule base so that the System 2 implemented in machines can use them. However, rule base maintenance and updating is also a topic for researching, but that is much easier.

This issue is likely to remain unresolved in the foreseeable future, and it remains uncertain even in the distant future. That is, we can build an AI with both System 1 and System 2, but we cannot build the bridge between the two systems, at least right now. However, if this problem cannot be solved, machines will clearly fail to reach the intelligence level of the human brain, and AGI will just be an empty promise.

§7 Linear Regression

As mentioned earlier, machines can only perform calculations that have specific algorithms; a computation is the execution of an algorithm. However, some activities of the human brain cannot be described as algorithms, at least not now. An algorithm essentially describes a process of mapping from input to output, step by step (not just an end-to-end mapping); this also includes what we refer to as "side effects", which means external behaviors. But this process must be concrete, where each step is attainable. For instance, the function sin(x) is mathematically defined as the ratio of the opposite side length to the hypotenuse length in a right triangle, and we know that sin (30°) is 0.5. However, how do we calculate sin(31°)? We do not know how long the opposite side is when the hypotenuse of the right triangle at 31° is kept at unit length. Even if we draw a right triangle on paper using a protractor and then measure it, the measurement's precision cannot be guaranteed. Therefore, the ratio of the lengths of the opposite side and the hypotenuse can only define the sin() *function*, but cannot define its *algorithm*. However, if you tell me the values for sin(31°) and sin(32°) in advance, and then ask me what sin(31.5°) is, I can interpolate using the existing values of sin(31°) and sin(32°) to estimate, which may not be precise but will be approximately correct, or will be correct in a large probability. Further, if many known values of function sin() at various angles can be given, then "*linear regression*" is an even better approach, we can perform the regression algorithms to derive a function curve based on the given sample values, and then make calculations based on that curve. The more the given samples, the more accurate the curve is. Thus, regardless of the concrete values provided (in input-output pairs) or their physical significance, the algorithm for linear regression remains unchanged; but it achieves the effect of "learning from data", and the process of linear regression is actually the process of "training". By using this method, some mappings without an algorithm now become computable on machines.

So, what does the linear regression process look like? Its general formula is as follows:

$$y(x) = w^T x + \varepsilon = \sum_{j=1}^{D} w_j x_j + \varepsilon$$

 doi:10.20944/preprints202509.2365.v1

Here, assuming that the regression has been completed, or in other words the training of the function has been completed, the result of the training is reflected in a weight vector w, and the length of this vector depends on the number of samples provided in the regression process, which is D. Note that here what the function y(x) defined is an algorithm, not how the function is mathematically defined. The real mapping implicitly defined by the function y() is embodied in these weights, which are distributed across that many parameters. It is through this weight vector and the algorithm listed here this function approximates the unknown function hidden in these samples, it achieves a "curve fitting". If a different weight vector w' is used, the function will fit another function. So, the concrete mapping of function y() depends on the weight vector w.

Now, given an input vector x, we want to obtain the value of y(x). In this calculation, x is also a vector of the same length as vector w; the function y(x) is a mapping from a vector to a scalar. In this algorithm expression, the first equality symbol defines the specific algorithm generally, indicating that the output is obtained by multiplying the transpose of vector w by vector x, and the resulting scalar is the output value of the function, with an error ε. The second equality symbol further expands the algorithm, making it more concrete, explaining how to compute it in more detail. Clearly, the number of steps in this algorithm is bounded; the vector length determines how many steps are in the main part of the algorithm, as the summation range is from 1 to D. Then, each input sample x_j's value is multiplied by a weight w_j and then summed. Here, x_j and w_j represent the value and weight of the j'th input sample, respectively. After summation, this produces the function value output, but this function value also carries an error ε, which is decided by the training samples; and the user has no way of knowing this error. In case the given input is a scalar, then make it a vector before computation, with all its elements x_j the same.

Conversely, during the regression process, the training process, what is given is a set of (x_j, y_j) 2-tuples, which is also a vector of length D, where each element is a known 2-tuple (x_j, y_j). By applying algebraic transformations to this formula, we can obtain an algorithm for calculating the weight vector, whose initial value is all 1's or randomly assigned. However, the goal is to minimize the error ε during the computation. That is, for each given tuple (x_j, y_j), take input x_j, compute its output, then compare it with y_j, and adjust the weights accordingly, so that the error can be minimized. That is the algorithm for the regression, or the training process.

People familiar with neural networks would immediately realize this is exactly the same algorithm for calculating the output of a one-layer neural network, or the input strength of a node in the first hidden layer in a multi-layer neural network, if we interpret j as the index of the input layer nodes in the network. The number of nodes in its input layer is D, where x_j represents the input strength of a specific node, and w_j is the weight of that node. The hidden layer node receiving this signal will figure out its own weight as well. Of course, there are other algorithms used in modern neural networks, including some nonlinear functions; but these are merely optimizations to improve its accuracy and performance, the basic algorithm is the same.

It can be seen that the function achieved by neural networks is essentially linear regression (with a certain degree of non-linear regression, but for the sake of simplicity, we will refer to it as linear regression below). Looking back at the history of neural systems, the researchers' intention was not necessarily to implement linear regression; instead, they aimed to mimic the neural system in the brain. However, what they inadvertently achieved is precisely linear regression, or in other words, they ended up at linear regression through different paths, which is certainly not coincidental. Of course, back propagation played a very important role in this process, but it only made the neural network training more efficient and accurate; it didn't change the essence that neural network training is actually a linear regression process.

By applying linear regression algorithms, many mappings whose algorithms are unknown can achieve end-to-end implementation on machines. However, those mappings are, of course, in System 1, something like intuitive responses, and in principle, we cannot expect them to explain why the mapped results were produced. The reason for the mapped results is merely "people said so". In addition, the errors in certain cases cannot be ignored, as these errors not only stem from the

 doi:10.20944/preprints202509.2365.v1

calculations in the regression process but also from the training samples, and is mainly from the training samples.

Especially in large language models, many of the training samples come from the media, but the materials in the media are often significantly biased and distorted. This is mostly not the fault of the media. We know that according to the principles of information theory, the lower the probability of an event occurring, the higher its information content. Thus, "a dog biting a person is not news; a person biting a dog is news". It is understandable that the media tends to report news with higher informational content. But this leads to the issue of proportional distortion. The machine under training does not understand that a large number of repetitive samples do not reflect what is really happening in the real world; it merely "parrots" what others have said, and the more samples that hold the same assertion, the higher the probability that this assertion will be output. However, users of large language models may be totally oblivious, assuming everything the LLM said is reliable. This is a typical scenario that requires correction from System 2, based on its rule base, but System 2 is missing in LLMs. In contrast, when we see news on the media, especially social media, we naturally have some degree of vigilance, because we do have a System 2 in mind.

Moreover, as said above, different genres have varying degrees of tolerance for errors. Artistic genres have a high tolerance for errors, so AIGC can achieve good results in literature and art. However, scientific genres are very sensitive to errors; it is a matter where a slight difference can result in a substantial deviation.

Lastly, but not least importantly, regression is based on the training samples, and it is only effective within the range covered by the training samples. Going beyond this range requires extrapolation, which leads to larger errors. This is particularly significant for inventions and innovations, as the contents of inventions and innovations are novel, didn't happen before, and thus require extrapolation, which carries a higher degree of errors. If we regard a specific invention as a vector in a multidimensional Hilbert space, then an extrapolated vector needs to precisely coincide with that vector to constitute the invention; any slight deviation would not match, and thus it cannot constitute the particular invention. Although the possibility of "hitting" cannot be ruled out, the "hit rate" is clearly very low. Therefore, in general, we should not rely on deep learning or large language models to produce inventions.

§8 Randomness and Innovations

We know that invention and creativity do not have algorithms, so it is impossible to construct a machine that specializes in inventing and creating (otherwise, greatness can be planned). Moreover, inventions cannot generally be achieved through exhaustive methods, because the space is simply too vast. Additionally, the method of linear regression, which is the method of deep learning, as mentioned above, is also almost unlikely to exactly "hit" an invention. Thus, the only possible algorithm left is random collision.

In fact, human inventions have a lot of randomness, but that does not mean the process of innovation is the same as a random collision, not at all. Archimedes was lying in a bathtub when he suddenly thought of the buoyant force acting on an object in water being equal to the weight of the water it displaced. This discovery clearly has an element of randomness. However, on the other hand, that is because it happened to Archimedes, and it happened when he was lying in a bathtub; given that, the randomness decreased significantly, and that gave it an extent of certainty. Yet, no one can draw quantitative conclusions from this example, nor is there an algorithm for such a discovery. True random collisions should not be able to ensure the yielding of such results; at least, the probability of such occurrences would be very low. In this sense, at least for now, we are still far from the notion that "machines can do everything humans can".

 doi:10.20944/preprints202509.2365.v1

§9 Conclusions

Through the above narrative and reasoning, we can conclude that the artificial intelligence achievable on a machine is still far from the human intelligence, and AGI is impractical, at least in the foreseeable future. However, this does not mean that artificial intelligence poses no threat to humanity. In fact, even without AGI, the power of artificial intelligence is already surprising and frightening. Even just as a tool, while AI can greatly help humans enhance their capabilities, it can also bring harm. This is true even for tools without intelligence; the invention of the automobile obviously provided convenience and increased the capability, yet it also led to many accidents. Should one day AI truly become AGI, and machines can possess all human intelligence, it must be a complete surpassing of humans because machines' performance will be much higher than that of humans; and machine intelligence can be easily replicated, whereas human children must learn from scratch for each generation.

However, at least in the foreseeable future, AGI (in its original meaning) is unrealistic, it is impractical (unless the definition of AGI changes, which is a different matter), and the “singularity” is not coming, let alone before the year 2030.

References

B.Jack Copeland, The Essential Turing
B.Jack Copeland et al., Computability - Turing, Gödel, Church and Beyond.
Rod Adams, An Early History of Recursive Function and Computability
Bernardo Gonçalves, The Turing Test Argument.
Jonathan St. B. Evens & Leith Frankish, In Two Minds - Dual Processes and Beyond.
Jonathan St. B. Evens, Thinking Twice - Two Minds in One Brain.
Jonathan St. B. Evens, Hypothetical Thinking
Christopher M.Bishop, Deep Learning - Foundations and Concepts
Kevin Murphy, Machine Learning: A Probabilistic Perspective