A PRELIMINARY STUDY ON THE CAPABILITY BOUNDARY OF LLM AND A NEW IMPLEMENTATION APPROACH FOR AGI

Yongcong Chen
New Millennium Future Technology Co.
YongcongChen@sina.com

Ting Zeng
Tsinghua University
ceng-t@mail.tsinghua.edu.cn

Xiaoyi Qian
galaxyeye Technology Co.,Ltd

Jun Zhang
Shenyang Normal University

Xinyue Chen
Tsinghua University High School

ABSTRACT

With a preliminary exploration of the capability boundaries of LLM (Language Large Model), we believe that the current mainstream artificial intelligence generally adopts the technical of "attention mechanism + deep learning" + "reinforcement learning", which cannot be applied to those fields that are difficult to a lot of "trial and error". So, to achieve AGI (Artificial General Intelligence) that works in any field, it's better to change the way we do it. Therefore, we propose a set of machine learning solution different from "deep learning + reinforcement learning". It adopts small samples and cumulative learning, and also realizes the attention mechanism similar to transformer, and also creates a fully connected knowledge network. In addition, it can realize interactive decision making with the environment without using lots of "trial and error" style learning. In addition, humans can preset different innate needs to it to achieve multi-objective balance, thus achieving far higher security than the current artificial intelligence. In this paper, we propose a set of new machine learning techniques which maybe guide humans realizes AGI.

Keywords LLM · ChatGPT · GPT-4 · AGI

1 Introduction

At present, the most artificial intelligence generally adopts the technical road of "attention mechanism + deep learning" + "reinforcement learning". "Attention mechanism + deep learning" is mainly used to build knowledge network; And "reinforcement learning" is mainly used to improve the continuous interactive decision-making ability of machine and environment. For example, GaTo model, launched in June 2022 by Google, can perform more than 600 different tasks in a single model. Another example is GPT-4, which has made amazing progress in knowledge understanding and reasoning. But what is the upper limit of the current LLM? Can "attention mechanism + deep learning" + "reinforcement learning" achieve true "artificial general intelligence"? In this article, we explore the capability boundaries of LLMS and propose a new path towards AGI.
2 The upper Limits of LLM

First, we note that current "reinforcement learning" uses external feedback to tell the machine whether the outcome is "good" or "bad" in different decision paths. This external feedback can come from preset reward functions, such as in alpha go, the feedback comes from the reward function that determines whether the outcome is winning or losing. It can also be the Feedback of Human beings, such as the RLHF (Reinforcement Learning from Human Feedback) technology widely used in LLM [3]. Therefore, the essence of the "reinforcement learning" is the way of "try" first and then "feedback", so that the machine can obtain different "reward" information through different decision paths. Then, the machine can obtain the pros and cons relationship between "state" and "policy", so as to obtain the interactive decision-making ability with the environment. [4] We believe that the current approach of "reinforcement learning " is more like an "evolutionary" learning approach when machines need to interact with the environment to make decisions. Its essence is "trial and error" and elimination is carried out through external feedback. This is very similar to the evolution of biology. Therefore, this type of learning only works in virtual environments, such as games, meta-universes, and content generation tasks. For tasks that require interactive learning in a real environment, such as caring for the elderly or driving a vehicle, reinforcement learning may not work well.

Second, "deep learning" creates knowledge that is difficult for humans to understand. As a result, machines and humans have two sets of knowledge, incomprehensible to each other! For example, humans have difficulty understanding the decision-making processes of LLM. Similarly, LLM have trouble truly understanding the knowledge that human language represents. For example, there is no AI agent that can read the instructions for "toaster" and then operate the toaster to bake bread buns in different bakeries [5] [6] [7]. Humans, on the other hand, can acquire the knowledge about the use of the toaster that others have accumulated by reading the instructions for the toaster. Thus, when facing the new task of "operating the bread machine", humans can make decisions under the guidance of the existing decision knowledge and interact with the environment. For example, when opening the top of the bread machine, humans do not need to try various schemes first (such as smashing the top of the bread machine), but directly acquire the accumulated experience of humans through language.

Therefore, we believe that real machine learning should be like human beings. In the face of a new task, we can predict the "good or bad" of different decision paths according to our past experience. By choosing a limited number of cases to try, we can obtain the decision knowledge to deal with the new task. Further, we believe that true learning should be done in the same way that children learn, by directly acquiring the accumulated experience of humans through language. When facing a new task, don’t even try once, just succeed once! For example, in the laboratory, when teachers teach children to do experiments, they pass on the decision-making experience of human beings directly to the children through language teaching. After acquiring the knowledge passed on by the teacher, children can directly complete the experiment in different environments by making decisions step by step and interacting with the environment. Although it may be the first time for the children to do these experiments!

Therefore, we believe that because of the wrong knowledge structure and learning methods, the current LLM of AI cannot solve the following serious defects:

2.1 Cannot solve problems independently.

For example, the current artificial intelligence does not take the initiative to help when it sees the owner fall down [7]. This is because a machine cannot generate its own goals without its own needs. Since a machine has no goals of its own, it cannot actively create a task. So, in the face of the unexpected situations that arise in real social life, machines have to deal with them through preset processes. Its preset process can come from a prompt process written by an external human [8]. Or the machine can follow the preset instructions and use the input information to find the similar process in the knowledge base to imitate the whole process.
Some people are now trying to write recursive calling functions. For example, under one task, call a function that says "look for processes that have done similar tasks", and then for each process you get, call a function that says "look for processes that have done similar tasks in the past" to get more processes. This is a recursive call formula. The current representative project is AUTO-GPT, which aggregates knowledge obtained from search engines with knowledge obtained from GPT-4 through storage, and then acts as prompt to find solutions from GPT-4. Through the recursive call of this process, a relatively macro task can be accomplished. If all the processes and parameters needed for the task already exist in the knowledge library of the AI (Such as LLM), it can complete the task. But what AUTO-GPT can do does not go beyond the boundaries of what LLM can do. Moreover, in real life, an endless number of unexpected situations may cause that at some point, there is no appropriate process and parameter matching the current environment, which will lead to the failure of recursive calls [9] [10] [11].

As a result, LLM do not spontaneously create new processes! A large model (Such as LLM) is essentially a high-level programming language. The syntax used is natural language. If we understand this, we can see where the boundaries of the large model are. No matter how many high-level functions we add to the large model, no matter how many tools and apps we integrate into the large model, the large model will not spontaneously create new processes. All of its processes are modeled after past processes (recursive calls), or processes those humans preset for them (main program calls). In both cases, it is essentially "using preset processes to deal with all problems". [12] No matter in this process how many “if... else ... ”, considering how many possibilities, it’s all preset, preexisting. It’s not something that's created on the fly for a specific task!

So, the essence of the large model is to provide a programming platform for human beings, and the programming language is the human language. The previous programming languages were C++, Java, Python and other computer languages. At the same time, the large model also provides a set of functions that can be called in human languages. For example, the "Create PPT" function is a human language command plus the parameters required for the function. The large model in the command + parameter mode, complete the task. In this way, everyone can be a programmer in the future. Those with a better understanding of the larger model, who could write better programs, would be called "Prompt engineers" —the programmers of the future.

Now, the large models continue to enrich their functions. For example, in the future, its function input parameter may also include image, video, action, or any sensor input, and its output may also include image, video, action, or any other multi modal sequence [13]. Therefore, in the future, all processes can be programmed in large mode programming languages (natural languages). The large models are machine language platforms, not fundamentally different from Python.

For example, in Python programs, human minds are transformed into Python program flows, and the python platform is responsible for translating functions into machine programs understood by various underlying drivers to accomplish tasks, thus implementing human preconceived problem-solving flows.

So, the large model is the new Python. Python is better suited to human interfaces than C++, so it’s more popular. GPT-4 is better for human interfaces than Python, so it will be more popular in the future. But C++ is not going away, because it has advantages in certain areas. So, the future of all sorts of dedicated large models, or Python, isn’t going away, there’s still room for a few of them. Therefore, in the future, all software can be rewritten in large model languages, and Prompt engineers will be the next generation of "white collar people" who are skills to write large model programs [14].

Therefore, the essence of large model is a more user-friendly "programming language". Its interface is a natural language, its functions are various and multi modal, and it is expected to soon establish its own ecosystem in the future. But it’s still very much a programming language! It is a programming language with very low barriers to entry and greatly expanded capabilities. That is its essence. So, the future is based on a large model that can do any task that can predict the flow. Those complex tasks are nothing more than writing more if ... else ... branch.
But in real life, there are plenty of tasks whose flow cannot be predicted! Such as caring for the elderly, driving to cook, spending time with children, farming, and so on. Humans can consider all kinds of possibilities for helping, but in the process of interacting with real environments, there are always surprises. How machines will handle these contingencies is impossible for humans to predict. So, this can be very dangerous, especially when the functions of large models are involved in all aspects of human life, and can lead to unacceptable losses for humans.

2.2 Knowledge cannot be updated in real time.

At present, with the adoption of big data training in artificial intelligence, knowledge cannot be updated in real time. But knowledge is updated in real time, which is crucial for machines that interact with the environment. Because the interaction between the machine and the environment is the process by which the machine acquires new knowledge. If the knowledge acquired by the machine cannot be updated in real time, the machine will not be able to update its decision-making knowledge based on the feedback of the environment. Therefore, such a machine, faced with the same environment, will keep making the same mistakes [3].

2.3 It cannot be applied to areas that require interaction with real environments.

In fields where interaction with the real environment is required, such as autonomous driving, doing housework, caring for patients, etc., the machine needs to establish knowledge of interactive decision-making between its own behavior and the external environment. These domains are not subject to a lot of trial and error, so machines cannot build these domains by interacting in real environments through reinforcement learning decision making knowledge. Therefore, the current technical solutions of artificial intelligence cannot be applied to these fields [2].

3 Preliminary study on the capability of current large models

3.1 How to describe the information contained in a matrix?

Although a matrix may contain many vectors, our first concern is: how many independent vectors are there? That is, what is the rank of the matrix? We can then, through matrix denationalization, find the corresponding set of eigenvectors, which are numerically the rank of the matrix, and which are also a set of coordinate basis clusters for this matrix. This set of coordinate basis clusters is complete, orthogonal, and is the simplest description of the matrix. Any vector in the matrix can be expressed by this set of coordinate basis clusters. If we set up a coordinate basis cluster is not orthogonal, but if it is complete, then we can also use this coordinate basis cluster to express any information in the matrix. If the coordinate base cluster is incomplete, then there are some vectors in the matrix, which cannot be expressed by the coordinate base cluster. In this case, it is necessary to increase the dimension of the coordinate base cluster.

So, how do we identify what information a vector in the matrix contains? Obviously, if it is a complete basis cluster, each basis is a dimension, and any vector projected onto the coordinate basis can accurately obtain all the information contained in a vector. If the basis coordinate cluster is an orthogonality basis, then we have achieved the simplest coefficients to express all the information of the vector. If the base coordinate family is not completely orthogonal, then we want it to be as close to the orthogonal base family as possible, because then the coefficients obtained are sparse. By combining the sparse coefficient matrix with a set of base coordinate clusters, we can understand the information contained by any vector in the matrix, and the information components are relatively independent. Therefore, the relationship between two vectors is reflected in the relationship between their coordinate base components.

Therefore, if two vectors in a matrix have non-zero components in the same dimension, they are considered to have local similarity. We can assume that there is some kind of connection between the two vectors. If there are multiple local similarities between two vectors, we can assume that there is a stronger connection between the two vectors. If
a similar connection occurs repeatedly in the matrix, we can assume that this connection is a general rule. This is knowledge.

So, in a matrix, to look for knowledge is to look for all the universal rules in it. The rules themselves are expressed by the partial basis components of the matrix, including dimensions as well as dimensions. It is the common connection relation obtained from a large number of vectors. So, it has a lower dimension and a wider range of proxies. So, knowledge is generalized from a large number of vector relations. Each knowledge is itself a vector in the matrix. They can be expressed as a matrix of coefficients. And a large amount of this knowledge constitutes a knowledge network.

Therefore, if we need to discover all the knowledge of a matrix, then, the most important thing is to find a set of base coordinate clusters, and then use it to decompose any vector in the matrix; And then through the coefficient matrix relations, to obtain the connection between different vectors; And then find those connections that can be repeated; They are the knowledge extracted from the matrix information.

But a matrix can have multiple sets of base coordinate clusters. As long as the dimension does not fall below the rank of the matrix, any base coordinate cluster can be adopted essentially. In order to obtain the most concise body of knowledge, we can just use the orthogonal coordinate cluster. However, if the matrix is very large, it is difficult to find a set of orthogonal base clusters. In this case, we can follow the most repetitive knowledge and use them as the base coordinate clusters. In this way, at least most of the knowledge in the matrix can be represented by sparse matrix. That is to say, we obtain a set of coordinate base clusters that can express the common knowledge in the matrix concisely. With the base coordinate cluster, any vector can be decomposed into the base coordinate cluster and represented by the coefficient matrix.

The similarity between any vector can be expressed by their spatial distance. And the space distance can be expressed by similar Euclidean space distance. The mapping relationship between any vector can be realized by the mapping relationship of coefficient matrix.

3.2 How does Deep learning create knowledge?

Suppose a group of people on an alien world build a 4-dimensional information space containing a large number of images, sounds, and actions. The humans there, they use pixels, syllables, and movement patterns as a baseline for their perceptual abilities. As a result, they are unable to see the connections between the complex combinations of pixels, syllables and motion patterns.

So, they began to use trial and error, trying different base coordinate clusters in the hope of finding a specific base coordinate cluster where the combinations of pixels, syllables and motion patterns they were interested in could form separate clusters. This, then, was the base cluster they needed.

Since the number of dimensions of the original data is very high, for example, 64 by 64 images, their dimensions are 64 by 64, they are 64 by 64 two dimensional impulse functions as the original base coordinate cluster. Their target dimension is likely to be much smaller, perhaps much smaller than 64 *64, because their goal is to find common feature combinations, rather than having to express all the information, some of which will inevitably be lost, and they can’t fit all the original information into the final dimension. Therefore, one possible trial-and-error method is to discard or compress some of the transformed dimension information after each attempt of a coordinate base cluster. Then compare it with the target to see if the error is increased or decreased, and decide the next time to change the direction of the coordinate base cluster. Obviously, with each attempt, a portion of the information is lost. After many attempts, the loss of information is too much, and may lead to the loss of useful information in it, so that the task cannot be completed. Therefore, the proportion of useful information in the overall information, the information loss rate after each change, determines the maximum number of changes. However, it may be difficult for the machine to find the optimal coordinate base cluster in the limited transformation trials. Therefore, a feasible solution is to add back some of
the lost information after each transformation, so that the number of transformation layers can be increased, so as to improve the probability of finding the optimal coordinate base cluster. This is known as the residual network. Of course, we can also consider inserting weak nonlinear functions in the mapping process of multiple neural networks to increase the number of times that can be transformed.

In the search for the optimal coordinate base cluster by trial-and-error method, the direction to look for is the reduction of error, and the tool to achieve it is BP algorithm. The data appearing in the neuron is actually the coefficient matrix under the base coordinate cluster, while the base coordinate cluster itself is implicit and does not appear in the multi-layer neural network. The inter-layer transformation coefficient is the coordinate coefficient transformation matrix in the transformation from one implicit basis to another implicit basis. BP algorithm is by adjusting the coordinate coefficient transformation matrix, from an implicit coordinate base, to another implicit coordinate base waiting to be tried.

Of course, if the attempt is made, the selected base coordinate cluster itself is not orthogonal coordinate system. It may occur that modifying the coefficients in one dimension will affect the coefficients in another dimension. This may result in a situation where the overall error is no longer reduced by any adjustment of the coefficients. The core reason is the non-orthogonal system among the base coordinate clusters. If they were orthogonal, this would not be the case. Therefore, it is necessary to go as far as possible towards the orthogonal coordinate cluster in the attempted path. And a sign of approaching orthogonal coordinate cluster is the sparse coefficient matrix. Therefore, the direction of the whole attempt needs to increase the sparse constraint of the coefficient matrix, which is a variety of regularization methods.

In addition, it should be pointed out that the nature of convolution, pooling, or other variants of deep learning has not changed. Convolution, for example, is essentially a neural network map, with an artificially large number of zero coefficients in the coordinate coefficient transformation matrix. Pooling, on the other hand, is nothing more than a layer of neural network mapping in which some of the dimensions are removed and certain nonlinear functions are adopted. This is the essence of deep learning, and a way to find a coordinate basis in a matrix. If there are labels in the data in space, then the number of labels is the dimension of the base site-cluster that is ultimately required. Their goal in finding the final base coordinate cluster is to use the combination of the minimum resolution features (in this case, pixels, syllables, and action patterns) common to each type of labeled data as a representative of each type of label, as well as a base coordinate cluster. The resulting matrix of coefficients is sparse. So, we can see that the essence of deep learning is also to find a suitable set of base coordinate clusters in the information matrix. If the required information is looked at an information subspace, then the base coordinate cluster obtained by deep learning can express any vector in this subspace, which is supervised learning. If the subspace is large and directly contains all the information, then the base coordinate cluster obtained by deep learning can express any vector in the whole information matrix, which is unsupervised learning. If the main purpose of learning is to cluster, and discard the information that cannot be clustered, then this is also unsupervised learning.

### 3.3 What is the nature of the attention mechanism?

The core of the attention mechanism is to discover the common arrangement among the elements of the information matrix. And the common arrangement can be chosen as the “frame work” of the information space. The so-called "frame" is that they generally exist in the information matrix, using them as the coordinate base cluster, can be concise description of the matrix common vectors.

More generally, we can think of each character as a dimension in the linguistic information space. If we adopt such a coordinate base cluster, we can use it to describe any vector in the linguistic information space. However, such a coordinate base cluster may not be optimal. If we take the linguistic information space as a matrix, then the optimal coordinate base cluster is obviously the base cluster composed of the eigenvectors of this matrix. The base cluster composed of eigenvectors has the smallest rank and the most concise description information.
For example, we can take the sentence "I am going to attend a friend’s wedding today" and decompose it according to each character as a dimension, and the obtained coefficient matrix is 9 dimensions. They would be "I, am, going, to, attend, a, friend’s wedding, today" But we can also put "subject...predicate...object" as a coordinate base, the coefficient of this coordinate base is "I...attend...wedding", and then the "auxiliary word + predicate" as a coordinate base, the coefficient of this coordinate base is "to attend today", but after the "attributive + object" as a coordinate base, the coefficient of this coordinate base is "friend’s wedding", obviously, the latter uses a more concise base coordinate cluster, express the same information. Moreover, the base of the latter coordinate cluster is framed. These frame coordinate clusters, under different coordinate components, can constitute a large number of similar information.

And the core of attention mechanism is to establish "frame" coordinate base clusters [16]. The method adopted is to extract the common underlying framework between the elements of the information matrix. This process is very similar to human learning. When we learn the information in a book, it’s the same way: "Books need to be thinned first and then thickened". To "be thinned first" is to summarize the framework information in a book. This is a process of information compression; And then "then thickened" is to add different details on the basis of the framework information to form new knowledge created by us, which is a knowledge generalization process [17] [18] [19]. Therefore, the core of transformer class model is the attention mechanism [15]. The core purpose of the attention mechanism is to get the common arrangement of elements in the information matrix and weight them according to how common they are. The more common the arrangement, the higher the weight. Those high-weight permutations are the main framework for how all the information is organized in the information matrix.

This process is similar to signal processing in communications. In the time domain, the seemingly chaotic and complex signals are converted to frequencies, and their low-frequency components determine the general trend of the signal, which is also the main component of the signal. These low-frequency components are the common form of organization found in this type of signal. If you think of each of these low-frequency components as a base coordinate cluster component, then they’re analogous to attention mechanisms. The low-frequency components express common connective relationships between information, and they are the "frame" of information. So, the attention mechanism, positive is by looking for the weight of the way information is connected to each other, to get the "framework" of the way information is organized. These "frames" are the basis of generalization. The mapping relationship between "frame" and "frame" represents the algorithm of "vector" to the next "vector". Input "frame" + different details is the specific input vector, and through the algorithm of "vector" to the next "vector", the output vector can be obtained, which is the knowledge generalization process.

In fact, human beings adopt the same way in the learning process. The common combinations of features, the specific "concepts," are the ones with the higher weight. They’re just common combinations in space and time. Common combinations of features further summarized from concrete "concepts" are "abstractions." This process can be iterative. So, there are a lot of hierarchical "concepts" in human society, which are frames. Therefore, abstract frame "cat" is a common arrangement of matrix elements in space and time, and this arrangement may contain the multi-modal matrix information elements of "cat", such as language, text, sound, image, action, touch and so on. In this arrangement, some of the matrix elements may have higher weights because they are more common, and they may all be the concept of “animal”. With fewer elements in "animal", its scope is smaller and its scope of application is larger, so between "cat" and "dog", the knowledge related to the combination of features they share (such as the concept of "animal") can be directly generalized.

The central ability of the attention mechanism, then, is to instantiate the statistical associations between languages into embodied input. The statistical association between languages is the statistical association obtained through pre-training, and this statistical association is a kind of incomplete statistical as association. It does not count the correlations between the elements of any combination of languages in all their arrangements, because it is an impossible task. Therefore, the actual correlations between languages in specific permutations need to be further optimized based on statistical correlations. This step is accomplished by the attention mechanism.
The core purpose of attention mechanism is to find out the correlation between input information, and input and output information by using the method of trial and error and using the relationship between human language as self-supervision from statistical correlation, and express this correlation through weight. This correlation, however, is very similar to the human process of learning. So, the machine uses deep learning — essentially trial and error — to find an optimal set of coordinates. This set of coordinate bases is likely to be very close to common human concepts. So, the core of deep learning is using trial and error to find a base coordinate cluster, and the core of attention is using trial and error to bring a base coordinate cluster closer to a humanoid concept.

The core task of LLM is to form a large number of frames, which are fused with each other to form a network. They are hierarchical. After data entry, the most important thing is to find the best matching frame, this is the basis for generalization.

### 3.4 Why do large models have the ability to emerge?

Why do large models "emerge"? It’s simple. For example, if an American comes to China, he can complete the correct translation process based on a large amount of common background information between us humans (such as personal needs, social structure, etc.) and a moderate amount of comparison between Chinese and English. The large model is like an alien. It has no common background information with humans. What it sees is only the way human information is connected. So, it needs to extract the connections between human information in order to predict how that information will develop. In the beginning, when it doesn’t have enough samples, the "information frame" it extracts is very different from the human "information frame", so it keeps making mistakes, groping in the dark, always hitting walls. As the number of samples increased, its "information frame" and the human "information frame" had a higher probability of aligning. But it’s not a linear pass. Like an anthropologist deciphers ancient languages, it gropes in the dark with little progress until it reaches a certain threshold. At a certain point, if the number of correct answers reaches a threshold, the decryption process speeds up dramatically. This is the phenomenon of "emergence".

Machines emerge not from intelligence but from finding "common association of the right pieces of information." This common association of the right information is similar to the way humans use it. And because all the criteria for evaluating it are human criteria, when it has enough bases and it’s close enough to human bases, its capabilities emerge when it has enough bases.

The ability of the large model to "emerge" is, at its core, to bring concepts generated by the large model closer to human concepts through the mechanism of attention. So, the training data must be sufficient for emergence to occur. The ability to generalize occurs because concepts are close to human concepts. Because human concepts, there are a lot of abstract concepts, they are the framework of information mapping. For example, "cat" is an abstract concept because it doesn’t represent a specific cat. So, for example, given a lot of input information and output information, the machine can build up frame information in those inputs and frame information in those outputs. And establish a mapping process from the input frame to the output frame. So, input frame + details, you can get output frame + details through the same mapping process. This is the knowledge generalization process.

When the training data is large enough, the machine is likely to find common complex patterns in it. The spatial association of common combinations of patterns is a thing, and the temporal association of common combinations is a process. The association of common combination patterns in space and time is knowledge. So, the large model seems to have knowledge about things, about flows.

This framed knowledge is called the "world model." It is on the basis of their own framework of knowledge that human beings come to understand and interact with all things. We can think of the large model as analogous to looking at things in the frequency domain. Similar to a picture, we can use a small number of low frequency components to get the main content of the picture. This is the core of image compression technology. And the attention mechanism, at its core, is a similar way of getting at the main content of our world’s information, using a small amount of weight. A
picture, in its low frequency component, can get different stylistic adjustments by configuring different high frequency components. Therefore, the core of generalization is to configure different details through the "frame".

At the core of today’s large models is the ability to create a matrix of transitions from an "input vector" to the next "vector" through deep learning. With this transition matrix, and with frame information, the machine is able to generalize knowledge. So, as long as the human gives it similar "input vector" to the next "vector" of knowledge, it can do the volume generation by imitating the "input frame" to the next "frame" of the transformation, embedding different details. For example, by establishing the attention mechanism of "company and founder", it can generalize "Steve Jobs and Apple" to "Lei Jun and Xiaoping". So large models perform tasks through imitation and creation, which is very similar to human beings. So, it’s not surprising that large models can generalize with small samples or zero samples. And it’s not surprising about magic of "Let’s think step by step", because "Let’s think step by step" propels machines to choose frameworks similar to Step by Step thinking to imitate.

So deep learning is a neat, elegant path, and the attention mechanism is one of the markers on that path, pointing us in the right direction. And "world models" are the fruits of this journey to artificial intelligence.

3.5 Could RLHF finally solve the problem of the larger model?

There are two serious problems with large models:

(1)The problem of toxic content. [20]

The knowledge of machines, the meaning of which is difficult for humans to understand, but which machines can use, seems not to be a problem. But the problem is actually quite serious. At the heart of the problem is this: humans cannot imitate the form of knowledge networks that machines have built, by presenting them with some innate knowledge! This is the nature of the problem. Because a machine cannot be preset with a priori knowledge, it can be preset with some basic requirements in the form of a knowledge network that is impossible to mimic. Without a machine’s own needs, it would be impossible for a machine to have self-perceived rewards and punishments. Without self-perceived rewards and punishments, it is impossible for a machine to spontaneously create projections of various things (i.e., combinations of various base coordinate clusters) into self-rewards or punishments. That is to say, the base coordinate cluster created by the machine lacks the basic dimensions such as reward, punishment, happiness and sadness which humans have and must have! This is because these dimensions are missing from the base cluster, so it is impossible for the machine to project input information to these dimensions and identify the information contained in the input. It is also not possible to predict potential rewards or penalties for these outputs by projecting onto them using different combinations (that is, different decision paths for the machine) when preparing the combined base coordinate clusters as outputs.

The current remedy for large models is RLHF. This is equivalent to adding a reward dimension to some training vectors following the fact. That is to say, a reward dimension is added to the base coordinate cluster of the machine. If the component value of the reward dimension is added to a large number of different types and sufficient number of vectors in the training data, it is equivalent to establishing the common component combination in these training vectors to the projection of the reward dimension. This is the reward function of the machine. So, the machine can also predict the reward component of the output vector produced by different decisions, that is, by different combinations. So, the machine will choose the output with the highest reward component. This is the amazing effect of RLHF learning. Because the knowledge learned through RLHF can actually be generalized. When a machine has its own dimensions of reward and punishment, it has its own preliminary "consciousness of seeking benefits and avoiding harm", which is why we can see the shadowy shadow of "consciousness" from the current large model, because it has the decision-making tendency to seek benefits and avoid disadvantages, it may have such a behavioral tendency.
But this is a patching after the fact approach, meaning that machines try and humans score and feedback, and it can only be used in fields where there is a lot of trial and error. This is similar to a young graduating with a PhD, but has no idea of "right and wrong" at all. The parents can only follow behind, shouting "No", "My God", "Yes" to give him a "right and wrong" idea, and he and his parents can’t communicate directly, only through "yes" and "no" to communicate. Therefore, this kind of learning effect is inefficient, and you may encounter those unexpected corner case forever!

(2) Nonsense in a serious manner [20].

The mechanism of attention is weighted by finding connections between information. The machine gets the "framework" of how information is organized through attention mechanisms (weights) + deep learning (trial and error). These "frames" are the basis for generalization. The mapping relationship between "frame" and "frame" represents the algorithm of "vector" to the next "vector", input "frame" + different details, is the specific input vector, through the algorithm of "vector quantity" to the next "vector", you can get the output vector, this is the knowledge generalization process.

But it needs to be noted that the machine through the "frame" to "frame" mapping, may produce does not exist "facts"! For example, the machine found that many of the journalist’s profiles were followed by links to other web pages of the journalist’s articles, or by awards the journalist had won in the past. If the machine sees a lot of this pattern of information organization, it becomes a mapping of "frame" to "frame." So, if the input information contains a similar frame, but only the name of the reporter is not the same, then the machine can be mapped to "frame + detail" through "frame + section", so as to produce a lot of web links in the output, or awards. But these web links and awards are also built through other "frame + detail" mapping to "frame + detail", they probably do not exist!

These are problems that large models struggle to solve. One solution is that RLHF can be used to break up this "frame" to "frame" link mapping so that the machine does not produce links to articles or awards, but this also reduces the power of the larger model. Another option would be to use a search engine to add information related to the input to the user’s input question, so that the input obtained by the machine contains more details, thus generating more personalized knowledge in the process of mapping "frame + detail" to "frame + detail". Again, this is a palliative solution, because the knowledge gained by search engines is not necessarily correct, and there is a limit to how much knowledge they can acquire for a particular problem.

So, we think of RLHF as a solution, but it’s not the final solution.

4 Attention mechanism + deep learning + reinforcement learning is the right path towards AGI?

Is the LLM right path towards AGI? We think the answer is no.

Deep learning is getting an optimized set of coordinate bases from a large number of samples. And use such coordinate base clusters to express vectors. So, by combining deep learning and attention mechanisms, you can produce optimized coordinate clusters that are similar to how humans express them. This is the real reason why Transformer can generate intellectual "surge".

In field of NLP, human beings from the early word bag model, word vector to EMLO[21], until Transformer, really realize the attention mechanism, and deep learning seamlessly combined to create an incredible miracle. We have noticed that the path adopted by these technologies is "vectorization first, establishing initial relationship; Then through trial and error, to adjust the coordinate base cluster; And then vectorize again under the preferred coordinate base cluster to get the correct relationship". Such a mechanism results in a huge amount of data required, and the knowledge is formed in one time through the training process, which is difficult to update in real time [2] [3].
First of all, we have noticed that deep learning is different from human learning in two aspects: (1) it adopts different minimum information elements. Human beings use the minimum local features they can perceive as the basic elements in the information space. Deep learning, on the other hand, uses its easy-to-use pixels, syllables or action patterns as the basic elements. And these basic elements are actually represented as strings of data arranged in space and time. So, the smallest unit of information (element) in the matrix that deep learning builds, although it may be similar to the human element, may not be the same. Perhaps, it could find a more concise and efficient set of elements.

In the same way, deep learning, based on these minimal units of information, can create a base of coordinate clusters that are just as inconsistent with the concept of humanness as it is difficult for humans to understand. But it may also create a more concise and efficient set of concepts. So deep learning is really an elegant solution! But it’s much better suited to the world of machines. When humans judge machines by human standards, we assume that machines are sometimes mentally retarded.

Secondly, the above problem does solve part of the problem when attention mechanisms are introduced. Through the attention mechanism, deep learning first focuses on the relationships among the smallest information units, and then creates base coordinate clusters based on these relationships. However, since the machine is confronted with data, the smallest information unit obtained from the data may still be inconsistent with the smallest information unit of human beings, so the "concept " created by the machine may still be very different from that of human beings. This is the core problem that causes the machine to not really understand language!

Machine knowledge, which is difficult for humans to understand, but which machines can use, seems to be a problem. But the problem is actually quite serious. At the core of the problem is this: humans cannot imitate the form of knowledge networks built by machines, presetting them with some innate knowledge! Since there is no way to preset a machine with innate knowledge, it is impossible to preset a machine with some knowledge of its basic needs, mimicking the form of knowledge networks the machine has built.

At present, the biggest flaw in artificial intelligence is that machines have no demands of their own. Without its own demands, the machine will not generate its own goals. Without spontaneous goals, there can be no spontaneous action. And a machine that has a spontaneous behavior is a machine that programs itself. A machine that can program itself is a truly intelligent machine. A machine that needs to be programmed externally will always be a machine driven by human intelligence.

How do you establish the demands of the machine? First, the requirements of the machine must themselves be part of the knowledge. Because only then can the machine make decisions based on its knowledge and interaction with the environment to meet its demands. So, demands are knowledge.

To realize that needs are knowledge, we need to mimic the ultimate network form in the memory library by presetting the machine with an innate "least pros and cons" kernel. Through "pros and cons kernel + small sample learning + continuous accumulation", the final form of "fully connected knowledge network with pros and cons information ". With the "fully connected knowledge network with pros and cons information ", the machine can independently predict the possible rewards and penalties under various decision paths according to its own knowledge. So, the machine can make its own decisions in accordance with the principle of pursuing advantages and avoiding disadvantages.

So, all the goals of the machine are created by the machine itself! Only in this way can the machine be in complex in the environment, from the top down, according to the specific situation, autonomous in the field to create sub-goals, autonomous decision-making, autonomous completion of the task! Any AI whose process preset by program, and then executed them, is still program-driven, whether the program interface is natural language or not. They may hit a brick wall in the real world with a lot of surprises in future!
With a machine capable of self-programming, it must also be matched with the corresponding knowledge, in order to truly realize the process of interactive decision-making and real environment. At present, the knowledge of interactive decision making between machine and environment is mainly accomplished through reinforcement learning. And reinforcement learning only works in fields where there’s a lot of trial and error. And what about areas where trial and error is difficult, like patient care, like farming?

More than a decade ago, some of the inventors of our patents discussed the need to build artificial intelligence based on human learning patterns, based on sample learning. Therefore, at the beginning, they tried to go from "symbol expression" to "common sense + cause-effect logic" to "knowledge network". After trying for a few years, I found that the first step on this road was impassable. Because "symbol expression" → "dog" how to express? All the characteristics of "dog" need to be picked out. But a "dog" can be an animal or a person! It can be "a character to be sung about" and it can be "a character to be despised"....So the essence of "dog" is the sum total of "dog " in relation to everything else. This is a definition of "dog" modeled after Marx’s definition of man. So "dog" must be put into the whole network of knowledge, defined by its relationship to all other knowledge. So, Symbolism faces a huge challenge! Because “dog" cannot be separated from another knowledge! "Fully connected knowledge networks " similar to deep learning must be built, which is our first conclusion.

Because the "dog" must be placed within the whole knowledge network, defined by its relationship to all other knowledge. So, you have to have enough knowledge to be able to make this whole thing clear. So, "the amount of knowledge has to be sufficient” so that you have enough background knowledge to understand what a dog is. That’s our second conclusion. When we look back, isn’t that what large models do? "Attention mechanism + deep learning" is to do the fully connected network, the large model is to do the "use a lot of knowledge, to build a fully connected knowledge network".

So why don’t we see robots walking around the streets? Because knowledge networks alone won’t do! It must also be able to "interact with the environment and make continuous decisions"! The current AI, on the other hand, relies on reinforcement learning to train its decision knowledge to interact with the environment. The AIXI algorithm, which is essentially the most idealized reinforcement learning algorithm, requires more computing and more predictions than there are atoms in the universe, which is impossible to achieve. "Alpha go" uses AIXI algorithm and trims the amount of computation through "Monte Carlo Tree" to reduce the amount of computation needed to play Go. So, one possible path to general artificial intelligence is: large model + AIXI algorithm (the strong reinforcement learning algorithm).

So why haven’t we seen roll out robots that walk the streets? The central obstacle to this path is that AIXI algorithm requires two preconditions [24]:

1. The machine needs to know the reward information it can obtain under different decision paths.
2. The machine needs to search all decision possibilities through traversal.

These two conditions can be perfectly satisfied in a game. Because in the end, winning or losing is the reward function. By playing the game ten million times, the machine can sum up the pros and cons of each decision. And the decision knowledge search space of the game is limited in the game, so the computer power required by the machine is capped. However, in real life, one only lives once, and it is impossible to acquire interactive decision-making experience by replicating endlessly. And unlike games, in the face of a task, there is no clearly defined scope of information search! So, training a machine to play a game, a meta-universe, and generate words and images can be trial-and-error, but what about driving a car, cooking a meal, caring for a child, caring for a sick person in a real environment? These areas cannot be trial-and-error, the current artificial intelligence technology road cannot solve!
5 What is the right road forward to AGI?

We believe that true "general artificial intelligence" through true "machine learning" requires 3 prerequisites:

Prerequisite 1: Sufficient knowledge + full Internet connection, no plugins! Any plug- in, and knowledge network cannot be integrated, prone to accidental mental retardation!

Prerequisite 2: Let the machine predict reward and punishment information in various decision paths! So, the machine has to be like a human: it can predict the rewards and penalties for the various decision paths by itself, with only a few attempts; Don’t have to try everything a million times!

Prerequisite 3: Machines can learn directly from the accumulated experience of humans! The nature of current "reinforcement learning" technology is trial and error. It should be called "reinforcement evolution"! It took hundreds of millions of years for human beings to evolve from the intelligence of a single-celled organism to today! Therefore, machines must be able to directly learn the accumulated experience of human civilization history, and cannot go on the old road of "evolution"!

It took us ten years to put forward a set of technical solutions under the guidance of three preconditions required for a truly "general artificial intelligence". It realizes true AGI through true machine learning, which mainly includes:

(a) To build a set of fully connected knowledge network that human beings can understand. And the way to build it is through the mechanism of attention and the mechanism of memory and forgetting. Memory and forgetting mechanisms mainly achieve statistical correlation, while the attention mechanism of information is based on statistical correlation, through the chain association activation process, through the multi-path accumulation of active values, through the extinction of active values over time to achieve.

(b) Because our fully connected knowledge network is an intelligible form of network organization (it is, in fact, a database). So, we can mimic the ultimate form of network in a memory bank by presetting machines with an innate "least pros and cons" kernel. Through "pros and cons kernel + small sample learning + continuous accumulation", the final form of "fully connected knowledge network with pros and cons information". With the "fully connected knowledge network with pros and cons information", the machine can independently predict the possible rewards and penalties under various decision paths according to its own knowledge. So, the machine can create its own goals and make its own decisions in accordance with the principle of pursuing advantages and avoiding disadvantages. Only in this way, the machine may face a complex environment, from the top down, according to the specific situation, independent creation of goals, independent decision-making, independent completion of tasks!

(c) The environment of the machine is very different, the task of the machine is also very different, so, the machine cannot be in any environment, through training to get any task related interactive decision-making knowledge! This is an impossible task!

So, we have to find a new way. Take inspiration from human learning. In fact, when humans are faced with a task, all decisions are made around the core of the advantage and avoid the disadvantage. That’s where avoidance, rejection, seeking more help come in. These behaviors are essentially new behaviors created by humans. Instead of tackling tasks directly, humans turn any task into one task of "how to meet their needs."

Machines must go the same way. The process of making machines learn "how to satisfy their own demands" as they go about their daily lives. Therefore, for any task, the machine is according to its own needs, it converts it into the "how to meet its needs" task. And on this task, it has a great deal of experience to generalize, because all of its learning is centered around this task. So, the solution we propose is that the learning process of the machine should not be task-oriented, but should be oriented to the needs of the machine itself. If the machine has its own needs and
the knowledge of how to meet those needs, then the machine can create new behaviors to meet those needs. In other words, the machine can "program" itself, and the task of programming is only "how to meet its own needs", and all the knowledge of the machine is built around "how to meet its own needs". Therefore, our machines can complete any task. It is in the process of "how to meet its own needs", but also to complete the specific tasks given by human beings. The result of completion may be "done", "rejected", or "further seek more information to evaluate". A machine is safe if it is preset with multiple innate needs for positive feedback from humans. And it will actively take steps to break down tasks and complete them.

because the machine is always learning "how to satisfy its own needs" to complete a task, and it is also always dealing with "how to satisfy its own needs" to complete a task. And completing specific tasks is a by- product of doing the "how to satisfy your needs" task.

The way machines organize their knowledge, presetting their needs. Then, the knowledge created by the machine needs to include knowledge related to the requirements.

What is knowledge? It is the way common features are arranged in space and time! What is requirement-related knowledge? It is the arrangement of common features in time and space, which contains information about the requirements of the machine. The arrangement of common features in time and space, if it includes the needs of the machine, is subjective common sense. That is, the relationship between "the world" and " me". The arrangement of common features in time and space is objective common sense if it does not include the needs of the machine. That is, the relationship "between outer everything". So, the way common features are arranged in space and time is common sense. Common sense is at the heart of an AGI. With common sense, machines will take the initiative to solve tasks and create processes according to their own needs. In other words, machines program themselves! Now that’s true intelligence! And the current large model + everything APP road, still not divorced from the way of human programming!

(d) The machine needs to integrate the knowledge network + machine demand + value evaluation into the same network. In daily life, machines are constantly learning "how to meet their own needs". So, it has a lot of experience that it can generalize to this question. If we preset the machine with a variety of preconceived requirements for positive feedback from humans, then the machine will be safe and will take the initiative to break down and complete the task step by step.

Innate needs need to include the machine’s own operational needs, so that the machine will maintain its own operations. It also needs to include preset human needs for the machine, such as the machine’s desire for positive feedback from humans, just like a human child. In this way, we can interact with machines and train them from an early age to align their values with those of humans. To establish the machine’s instinctive needs, we also establish the machine’s higher order demand, such as "morality", "obeying the law" and so on. At the same time, we can also preset a small amount of innate knowledge, which is mainly used in trial-and-error domains, such as minimal "cliff avoidance" knowledge. In this way, we create a child who has needs, who is selfish, and who has a small amount of instinctive knowledge related to survival. But it craves human approval. It has an innate language for communicating with humans (such as an innate knowledge of nodding or shaking one’s head). Then, based on an innate language for communicating with humans, humans can slowly develop more complex ways of communicating with machines, such as language. Then, by learning in the real world, machines can acquire both knowledge through self-summary and direct human knowledge through language learning. They may also use their unmatched ability to discover common features arranged in ways that humans do not, even though the patterns of arrangement in time and space are not obvious to humans. But a machine can find out by statistics, and it can imitate the way humans use symbols to express common permutations, and symbols to express newly discovered common permutations. This is the new knowledge that machines create.

It’s an iterative process. The machine programs itself to discover more knowledge, and it will develop into super intelligence!
Machines need to learn through small samples and accumulations. Only in this way can knowledge be updated in real time.

In real life, a large number of tasks need to be completed step by step by machine and environment interaction. Therefore, any feedback from the environment must immediately become the basis for the machine’s next decision. And this decision knowledge needs to be updated immediately into the machine’s knowledge base. Otherwise, the machine will make the same mistake in next time. At present, human intelligence uses big data samples, and knowledge is mainly accomplished through a single training. Even fine-tuning for a task cannot be updated in real time. Therefore, the real learning path should be small sample, cumulative learning. This way of learning, which is more similar to human beings, can realize real-time update.

6 The evolution direction of artificial intelligence

We believe that the development of artificial intelligence can be roughly divided into different stages:

(1) The stage before the realization of true consciousness can be considered as the "feature exploration" stage. Before deep learning, it is mainly concentrated in the "human worker exploration" stage. Artificial exploration can be "expert system", "knowledge encyclopedia", "probability statistics" and so on. After deep learning, focus on the "machine exploration" phase, allowing the machine to "discover characteristics" from a large sample.

(2) After the implementation of true attention (Transformer), the machine can be considered to have achieved "knowledge generalization" as its "knowledge" and human "knowledge" are initially aligned. In tasks from human, the machine can exhibit some intelligence through "knowledge generalization".

(3) In the future, we think AI needs to evolve to the next stage: the stage of "autonomous interaction." "Autonomous" means that the machine is no longer a silent "machine", it is capable of generating behaviors spontaneously (which is equivalent to programming itself), and the machine will seek out knowledge on its own (such as actively interacting with the environment to acquire knowledge). "Interactive" means that a machine can interact with its environment in real time, updating its knowledge in real time, making continuous decisions, and performing complex tasks in unfamiliar environments.

The core of achieving "autonomous interaction" is that machines should have their own demands. The needs of the machine must be part of the machine’s knowledge, so that the machine can use its knowledge to create behavior, so as to satisfy its own demands.

And the core of the realization of machine requirements is to first create a knowledge network that human beings can understand. Only in this way can human beings imitate the form of knowledge network and preset the requirements of machines. Then let the machine learn around its own needs, thus establishing all the connections between the information and the needs. So, the machine’s knowledge is all about its demands. In this way, the machine can transform any task into a single task: how to satisfy itself. And all the machine’s exploration and learning process revolves around the task of "how to satisfy its own needs." Therefore, when a machine is faced with "how to meet its own needs", it has a great deal of experience that can be generalized. Only in this way can machines handle tasks that are difficult to try and error. In fact, we believe that humans use similar methods to acquire knowledge and deal with problems.

It is a paradigm shift for AI to be oriented towards "its own demands" rather than towards "Tasks from human". We believe this paradigm shift is necessary. Because external tasks are so diverse, a lot of them have to interact with the real world. They are hard to trial and error, they are hard to get big data samples, and they are hard to gain decision-making knowledge through reinforcement learning. In addition, training machines for each type of task in real environment is an impossible task.
Artificial general intelligence is the beginning of AI, and also the crown of AI. We have put forward a set of technical solutions to realize AGI a few years ago. In the references [25][26][27][28], we present detailed technical details for achieving this path, which may be a viable path to lead mankind towards AGI.

In this scheme, the requirements of the machine are preset by human beings and can have multiple requirements, so the goals generated by the machine are also multi-objectives. At present, artificial intelligence is a single goal. From the point of character, it can be regarded as the artificial intelligence of "All means necessary for the goal". Because of this kind of artificial intelligence, it only pursues a single goal, and does not think about anything beyond the goal. So, this is very dangerous artificial intelligence! In our scheme, the multi-objective also includes the alignment of human values, including the needs of "morality", "rules and laws" and "recognition". Therefore, in our scheme, the machine will comprehensively consider the needs of "morality", "rules and laws" and "recognition". Therefore, our scheme is a feasible way to solve the security of artificial intelligence.

7 A new approach to AGI

We believe that from the point of view of information, all information in the world is a multi-dimensional information matrix. Knowledge is another way of describing this information matrix. This is essentially the same way that we can describe the same matrix in the time domain, or in the frequency domain.

Common knowledge, or "common sense", is a description of the common tokens arrangement in the information matrix. By adopting these coordinate basis clusters, we can conveniently describe common vectors in matrices. Therefore, using "common sense" to describe the information in the matrix is a concise description of the matrix information, which is essentially a form of information compression. These concise descriptions are the "principal components" of the knowledge matrix, which is a kind of frame information. They are similar to the low-frequency principal components in image information.

The nature of the attention mechanism is to find common tokens arrangements in the information matrix and assign weights to these common tokens arrangements. If these Tokens information is one-dimensional, such as language information, then the attention mechanism will find common language combinations. With it, we can generalize language; If these Tokens information is two-dimensional, such as image information, then the attention mechanism will find common image Tokens combinations; With it, we can achieve image generalization; If these Tokens information is three-dimensional, such as stereoscopic spatial information, then the attention mechanism will find common image Tokens combinations; With it, we can achieve the generalization of spatial structure; If these Tokens information is 3D + time dimension, such as process information, then the attention mechanism will find common process Token combinations; With it, we can implement process generalization; And the generalization of processes will bring unmatched new capabilities to machines.

So, attention mechanism solves the problem of "objective common sense". "Objective common sense" is the relationship between the external information created by the machine. But compared to human common sense, large models are currently unable to create "subjective common sense." "Subjective common sense" is the relationship between "external information" and "machine's own requirements". Human beings understand the world, more from the "subjective common sense", to gradually understand the world. And objective common sense, more for the "subjective common sense" service.

At present, large models cannot create "subjective common sense". The core problem is that they do not have "machine’s own needs". At present, humans can only add a "reward and penalty" dimension to the information space through RLHF as an ex post remedy. The robots created in this way act like the "one track thinking" way: they think only about a single goal, not about the right means. It would be very, very dangerous if they took over our lives!
We propose a new machine learning approach. It takes the "connectionism" road, uses small samples, cumulative learning, and achieves the same attention mechanism as the Transformer. But because it uses cumulative learning, its knowledge can be updated in real time. This technical solution has two main features:

(A) Small samples, cumulative learning, real-time update of knowledge.

(B) Machines have their own requirements.

So, we accomplished 3 things:

1. small samples, cumulative learning, achieve the same goal, the attention mechanism and fully connected network, knowledge can be updated in real time.

2. Knowledge network + machine demand + value assessment are integrated into the same network.

3. the machine transforms all tasks into a single task of "how to satisfy itself requirements".

So, we solved the following problems of current AI:

(1) Lack of common sense: especially subjective common sense. So when faced with a new task, you need to learn by trial and error. There is no way to predict pros and cons, so only trial and error, which limits the application area.

(2) In the face of complex tasks, it cannot autonomously find additional help to help it solve the problem: because the machine has no own needs, it will not have autonomous goals; Without autonomous goals, there will be no autonomous decision-making; Without autonomy, you don’t actively seek out extra help to help you make decisions. A robot that does not create tasks autonomously will not be an AGI.

(3) The problem of lack of real understanding of language: because our knowledge is an arrangement of tokens in space and time. The temporal and spatial arrangement of multi-modal tokens activated by input tokens (e.g., linguistic tokens) can be imitated. So machines can learn from human experience through language, and when faced with unfamiliar tasks, they may succeed in the first attempt.

(4) Problem with limited application domains: The application domain of our machine is not just "content generation". The machine converts any external task into a process of "how to satisfy its needs". The machine in daily life, is constantly learning "how to meet its own needs" process. Thus, it has a large amount of experience that can be generalized. Machines perform external tasks in the process of "satisfying their own needs", similar to humans.

(5) Self-learning: Currently, when the AI sees its owner fall, it doesn’t come to help. But our machines have their own needs, and they have their own initiatives. It will arrange tasks for itself, which is equivalent to programming itself, realizing autonomous learning and self-iteration.

(6) Safety problem: Our machine, "needs" are given by humans, humans can train the machine through a variety of "needs", so that its values and human values are aligned, the machine is multi-objective balance, so its safety is far better than the current artificial intelligence.

(7) Here, we briefly introduce the implementation process of our scheme in following. More detailed implementation steps will be disclosed in subsequent documents.

The implementation process of our scheme:

1. The integration of "knowledge network + machine demand + value evaluation" is our core technology. The implementation scheme is to imitate the final network form in the memory library, and preset an innate "minimum pros
and cons" kernel to the machine. Why can we preset a priori knowledge? Because of our knowledge network, humans can understand! Large models can’t do that now!

2. Through the "pros and cons kernel + small sample learning + continuous accumulation", the "fully connected knowledge network with pros and cons information" is finally formed.

3. With the "fully connected knowledge network with pros and cons information", the machine can autonomously predict the possible rewards and punishments under various decision paths based on its knowledge. So the machine can make its own decisions according to the principle of seeking benefits and avoiding disadvantages. So all the targets of the machine are created by the machine itself! Only in this way, it is possible for the machine to autonomously create sub-goals on the spot, autonomously find new information to assist decision-making, autonomously find relevant tools, autonomously make decisions, and autonomously complete tasks in a complex environment, from top to bottom, and according to specific circumstances! So our machines can program themselves, self-iterate, self-evolve.

4. When the information input (external information or machine self-monitoring information) are input, some reward symbols and punishment symbols are activated. Each activation path from the input to the reward symbol and the punishment symbol is a logical link that potentially generates a reward or punishment. If on this logical link, every underlying feature is truthfully realized, then the reward or penalty propagated by this logical link is also realized. So the response of the machine to any input information is the same: increase the probability of the reward logic chain and decrease the probability of the punishment logic chain, so as to achieve the purpose of seeking advantages and avoiding disadvantages.

5. How to increase the reward link and reduce the probability of the penalty link? Is to increase, or decrease, the realization probability of the underlying feature with a high activation value on the link. Tokens with high activation values on a bridge, are the tokens with high weight for this link. When they are true, then the activations propagated along this link are true, so the reward, or penalty, that is eventually activated, is also true. How does it work? From the activation value transfer path of input information → reward and punishment symbol, the N bottom features with the highest activation values are selected, which are the top realization path that leads to reward or punishment. The goal of the machine is to: 1) implement the tokens on the reward path (that is, mimic past experience and make them appear in the input). Make the tokens on the penalty path unattainable (that is, avoiding them from the input by mimicking past experience).

6. The policy space search of the machine, bounded in the set of all tokens that are activated. The machine estimates the potential reward value by passing the activation value of each input → reward/penalty symbol. So, our machine, even during training, pre-estimates the reward value. However, current artificial intelligence, in the training process, gets the reward value after the fact. This determines that the policy-states knowledge of AI and environment interaction decision-making, must come from a lot of trial and error process.

8 conclusion

After analyzing the capability boundaries of the LLM, we propose a new machine learning scheme to break through the current LLM capability boundaries. We believe it may be a feasible path to create a machine who can create tasks autonomously, iterate and evolve itself. Moreover, the machine learns during its life and updates its knowledge in real time. It can directly access all the experience accumulated in the history of human civilization through human language. In reference [25][26][27][28], we propose a specific implementation plan, and we will provide more details in the future.

References


[24] Sunehag P, Hutter M. Principles of Solomonoff Induction and AIXI[C]// Ray Solomonoff memorial conference. Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia; Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia, Department of Computer Science, ETH Zurich, Switzerland, 2011.

