Expert Systems With Applications 209 (2022) 118221

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Tracing the evolution of AI in the past decade and forecasting the emerging trends☆

Zhou Shao [a,b,1], Ruoyan Zhao [d,1], Sha Yuan [a,*], Ming Ding [b], Yongli Wang [c]

[a] Beijing Academy of Artificial Intelligence, China
[b] Department of Computer Science and Technology, Tsinghua University, China
[c] School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[d] Polytechnic Institute, Zhejiang University, China

ARTICLE INFO

Keywords:
Artificial Intelligence
Frontier research
Future trend
Data analytics
Science of Science

ABSTRACT

The past decade has witnessed the rapid development of Artificial Intelligence (AI), especially the explosion of deep learning-related connectionist approaches. This study combines traditional literature review, bibliometric methods, and the Science of Science (SciSci) theory to scrutinize the development context of AI in the last decade on AMiner.[2] With the assistance of AMiner tools and datasets, this paper aims to describe a further explicit context and evolution of AI in the past decade from the development of connectionist approaches. Five aspects of the past decade are highlighted: self-learning and self-coding algorithms, Recurrent Neural Networks (RNN) algorithms, reinforcement learning, pre-trained models, and other typical deep learning algorithms, which represent the significant progress of this field. By combining these critical parts, we then summarize the current limitations and corresponding future of AI trends in the next decade and discuss some topics about the next generation of AI. Discoveries in this paper will benefit AI research in promoting understanding of the current critical stage and future trends of AI development and the AI industry in the dramatic ascendant for the academic research results transformation and its industrial layout.

1. Introduction

Artificial Intelligence (AI) is a vital research area in academia and industry, whose remarkable revolutions in theories and applications dramatically changed our daily lives in the last decade. Nevertheless, this is only the beginning of the new era of AI in its theory, method, research, and application, which is experiencing significant bottlenecks in its dramatic expansion process at the present moment. However, AI has been an obscure subject since its establishment in 1956 for a long time and the process of building the concept of AI is long and complex. Since its birth, the concept of AI is becoming gradationally clear and is increasingly well accepted. Gradually, AI has developed into a broad discipline that draws upon computer science, mathematics, artificial psychology (Wang & Xie, 1999), linguistics, and many others in the past decades. Frontier scientists also believe that finding of new goals and tasks is also a key element of artificial intelligence. The rapid development of AI has dramatically changed our way of production and life. People pay attention incrementally to a series of AI-related topics, such as AI talent, AI application, AI technology, AI research, AI education, and so on. Growingly, people of insight realize that AI is an inflection point in global history, and AI is gradually becoming an enterprise strategy and national strategy. As is well-known, AI has become the new arena of the new round of scientific and technological revolution and industrial transformation and is a significant breakthrough to seize the opportunity for future development. The importance of AI is self-evident that AI will have an ever-increasing role in scholarly research and innovative applications shortly.

However, the development of AI is not plain sailing. There have been several ups and downs in the history of AI, which shows that AI development has been full of twists and turns. It is generally accepted that AI is divided into three schools: symbolicism, connectionism, and actionism. Zhang, Zhu, and Su (2020) suggested that the development of AI can be conceptually divided into three stages: (1) Symbol AI, also

☆ AI is a very broad concept. In this paper, we pay particular attention to bibliometric analysis on connectionist approaches. Therefrom, we discuss the present situation of connectionist approaches and the future of AI.

* Corresponding author.
E-mail addresses: shaozhou@tsinghua.edu.cn (Z. Shao), ryanzh@zju.edu.cn (R. Zhao), yuansha@baai.ac.cn (S. Yuan), dm18@mails.tsinghua.edu.cn (M. Ding), yongliwang@njust.edu.cn (Y. Wang).

[1] Zhou Shao and Ruoyan Zhao are co-first authors of the article. They contribute equally to this article.

[2] www.aminer.cn, AMiner is an academic mining system, which will be introduced in the following section.

https://doi.org/10.1016/j.eswa.2022.118221
Received 28 April 2021; Received in revised form 3 March 2022; Accepted 17 July 2022
Available online 28 July 2022

called knowledge-driven approach; (2) data-driven approach, based on deep learning; (3) the Third Generation AI–an interpretable robust theory, combines knowledge-driven and data-driven theory. In reality, the development of AI in the last decade has wholly established people's cognition of AI and has become the so-called Fourth Industrial Revolution. Additionally, this happened when Geoffrey Hinton introduced DBN in 2006 and the layer-wise pre-training technique, opening the current deep learning era (Hinton, Osindero, & Teh, 2006). However, it has only been few years since the modern deep learning era began at the 2012 ImageNet competition (Krizhevsky, Sutskever, & Hinton, 2017). Since then, AI represented by deep learning is advancing by leaps and bounds. At the same time, the application of AI in many fields is exploding. As a remarkable phenomenon, people call it AI Plus (AI+), such as the AI application in industry, finance, medicine, education, agriculture, etc. The application of AI in these fields increases the innovation and productivity of society as a whole.

Over the last few decades, the course of AI development has brought us much enlightenment. Since the early days of AI, the research on the development history and the trend of AI has never stopped. Looking backward to AI history, some intriguing regularities can help us accelerate its development. Empirical evidence over the last several decades shows that it is vital to grasp the development trend of AI, which is conducive to the formulation of AI-related strategies. Recently much work (Haenlein & Kaplan, 2019; López-Robles, Otegi-Olaso, Gómez, & Cobo, 2019) has been done in analyzing the various stages of AI development, significant innovations, primary applications, future trend, etc. These studies (Shao, Shen, et al., 2020; Shao, Yuan, & Wang, 2020; Yuan, Shao, Liang, et al., 2020; Yuan, Shao, Wei, et al., 2020) have instructive significance in AI scholarly research and application. Remarkably, the past decade has witnessed successively critical breakthroughs in AI academic research and application than ever. Comparing with the past decades, the development of AI in the last decade is particularly worthy of study (Shao, Shen, et al., 2020; Yuan, Shao, Wei, et al., 2020). In recent years, many scholars have contributed to the literature review of AI applications in many fields, such as industry (Choubey & Karmakar, 2020), commerce (Cosma & Acampora, 2016; López-Robles, Rodríguez-Salvador, Gamboa-Rosales, Ramirez-Rosales, & Cobo, 2019), medical care (Ávila-Tomás, Mayer-Pujadas, & Quesada-Varela, 2020), education (Tahiru, 2021), literature-based fields analysis (López-Robles et al., 2018), etc. Overall, these studies illustrate the development of AI in the last ten years, which have important research implications. Nevertheless, they have the following shortcomings: (1) Ignoring the integrity of the current development of AI because AI technologies are closely linked; (2) Most of these studies are based on a literature review with a strong personal perspective, lacking quantification analysis, and the connection between techniques may be ignored; (3) Insufficient analysis of the future trend combined with the current situation.

In this paper, a method combined traditional literature review with bibliometric methods and SciSci is employed to study the developments over the last decade and future trends of connectionist approaches, which can offer a glimpse into AI. However, this method is not easy to be conducted for the rapid growth of papers and complex citation relationships between them. The development of scientific and technological big data mining analysis systems has brought new opportunities to this research. On the one hand, this method can provide quantitative analysis and define the research highlights of the study. On the other hand, it can help us understand the evolution and relevance of AI technologies. With the help of these platforms, this joint method makes the analysis additionally objective and even comprehensive.

Firstly, this study states the research background and gives an overview of significant research directions. Secondly, we illustrate the methodology and demonstrate the data and tools used in this paper. And then, a literature review on five aspects has been carried out on the guidance of the proposed methodology to illustrate the development of AI on connectionist approaches. Furthermore, based on previous studies and analysis, this paper emphasis the limitations and future trend of AI. At last, combined with some current advanced views, we discuss the way to the next generation of AI.

2. Background

The development of AI has gone through a long historical process. In the beginning, people's understanding of AI was mostly related to myths. There were legends of using witchcraft or alchemy to give consciousness to inanimate matter in the Middle Ages, such as the Takwin of Jabir, the Golem of Homunculus of Paracelsus, and the golem of Judah Loew (Russell & Norvig, 1995). The origin of AI can be traced back to 1942 when American science fiction writer Isaac Asimov published his science fiction novel "Runaround". In the 1940s and 1950s, neurological studies showed that the brain was a neuronal neural network that emits with or without pulses, triggering discussions among a few scientists from mathematics, psychology, engineering, etc. They began to explore the possibility of an artificial brain. In 1943, neurologist Warren McCulloch and mathematician Walter Pitts co-authored a book that combines mathematics and algorithms, establishes neural networks and mathematical models, and simulates human thinking activities, thus opening the door to artificial neural networks. Claude Shannon proposed computer games in 1950 and published "Programming a Computer for Playing Chess" (Shannon, 1950), expounding the "methods of realizing man-machine games". This is the first article discussing the development of computer programs. This article started the theoretical study of computer chess. The main ideas can still be seen in "Deep Blue" (Hsu, Campbell, & Hoane Jr, 1995) and AlphaGo (Silver et al., 2016) many years later. In 1950, Turing published the article "Computing Machinery and Intelligence" (Turing, 1950), where he described how to create intelligent machines and how to test their intelligence. This test is the famous Turing Test, which is still used now as a criterion for judging whether a machine is intelligent. Turing proposed a hypothesis that a person conducts a series of questions and answers with the other party in a special way without contacting him. After a while, if he can judge whether the other party is a human or a computer based on these questions, then this computer can be considered to be able to think. In 1954, the first transistor computer came out at Bell Labs in the United States. At that time, integrated circuits had not yet been invented, and computer networks were unheard of. In 1956, the concept of AI was formally proposed at a two-month AI seminar in Dartmouth by John McCarthy et al. marking the birth of AI. At the Dartmouth Conference, the problem of using machines to simulate human intelligence was discussed seriously, and formally used the term "AI" for the first time, which marks its birth. Since the Dartmouth Conference, many participants in this conference played a pivotal role in the development of AI in the following decades. This new field has prospered and developed in the next period.

By the 1960s, AI had its first wave of climax, developing symbolic logic (Uhr & Vossler, 1961), solving several common problems, and initially sprouting NLP and human–computer dialogue technologies. In 1964, Daniel Bonrow developed a system STUDENT that can understand natural language input, which is one of the representative events in AI history. The system is an early AI program written in Lisp that can solve algebraic word problems. In 1966, MIT computer scientist Joseph Weizenbaum (Weizenbaum, 1966) developed the world's first computer program ELIZA that can communicate with humans. In the same year, Herbert A. Simon and George Baylor developed a chess mating combinations program MATER (Baylor & Simon, 1966). After that, the Massachusetts Institute of Technology (MIT) began to develop the MACSYMA system, which can solve more than 600 mathematical problems after continuous expansion (Martin & Fateman, 1971). Now, expert systems are an important branch of AI and are listed as the three major research directions of AI alongside NLP and robotics. In 1969, Minsky and Papert proved various theorems about single-layer perceptrons in their book (Marvin & Seymour, 1969). During this period, people were very optimistic about the development of AI, and there were predictions that appeared, such as "In twenty years, machines will be able to do all the work that humans can do" (Simon, 1965). As AI researchers failed to make correct judgments on the difficulty of their

topics, the previous overly optimistic attitude made people have too high expectations for the development of AI.

In the 1970s, AI encountered difficulties that could not be overcome at the time. The limited memory and processing speed of computers were not enough to solve any practical AI problems. No one could make a huge database then, and the research progress came to a standstill. In 1976, the heuristic search was introduced in discovery in mathematics (Lenat, 1976). The development of technology related to visual information processing (Marr, 1976) gradually makes computer vision a vital subfield of AI. Infinite attention is being paid to knowledge representation, and reasoning in this stage leads to new knowledge technology becoming the key to research. Infinite attention is being paid to knowledge representation and reasoning (Minsky, 1974) in this stage, leading to new knowledge technology becoming the key to research (Feigenbaum & McCorduck, 1983). In the same year, Randall Davis pointed out that the use of an integrated object-oriented model is a complete solution for improving the development, maintenance, and use of knowledge bases (Davis, 1976).

In the 1980s, AI entered its second climax. The non-monotonic logic proposed by Drew McDermott and Jon Doyle in 1980 (McDermott & Doyle, 1980) and the robot system was proposed later. In the same year, Hans Berliner's computer victory over the world champion of backgammon became a landmark event (Berliner, 1980). In 1982, David Marr made a profound impact on Cognitive Science (Marr, 1982). Subsequently, in 1986, Brooks proposed an entirely different view and structure about intelligence from symbol-based AI, marking the creation of behavioral robotics, which became a vital development branch of AI (Brooks, 1986). In terms of machine learning algorithms, the multi-layer perceptron proposed by Geoffrey Hinton et al. solved the inability of perceptron to perform non-linear classification. And the probability method and Bayesian network (Pearl, 1985) advocated by Judea Pearl laid the foundation for subsequent causal inference (Pearl, 1990).

In the 1990s, there were two significant developments in AI. On the one hand, Tim Berners-Lee proposed the semantic web in 1998 (Berners-Lee et al., 1998), a semantic-based knowledge network or knowledge representation. Its core is by adding metadata that computers can understand to documents on the World Wide Web (such as HTML). The entire Internet becomes a general information exchange medium. The most essential element of semantic web is the linked node. Later, OWL language and other related knowledge description languages appeared, which provides a possible solution for two core problems of the knowledge base: knowledge expression and open knowledge. However, this idea has not been widely recognized afterward. It was not until 2012 that Google put forward the knowledge graph concept (Singhal, 2012), which gave this direction a clear development idea. A significant development is the statistical machine learning theory, including the support vector machine proposed by Vapnik Vladimir (Cortes & Vapnik, 1995) et al. It has shown unique advantages in solving small samples, nonlinear and high-dimensional pattern recognition, and can be extended to function fitting and other machine learning problems. John Lafferty et al. proposed a conditional random field, a discriminative probability graph model based on the Bayesian theoretical framework (Lafferty, McCallum, & Pereira, 2001), used for text segmentation and labeling. Simultaneously, it performs exceptionally well in many NLP tasks such as word segmentation and named entity recognition. David Blei, Michael Jordan, et al. put forward the model LDA (Blei, Ng, & Jordan, 2003). It has made groundbreaking contributions in the field of knowledge-based systems and human–computer interaction. Edward Feigenbaum proposed the world's first expert system–DENDRAL (Lindsay, Buchanan, Feigenbaum, & Lederberg, 1993), and the system imitates the decision-making process of human experts. Non-experts can get suggestions from the system and give a preliminary definition of the knowledge base. In 1993, DEC company and Carnegie Mellon University developed the XCON-R1 expert system, saving DEC company millions of dollars every year (Moynihan, 1993). Expert systems began to be adopted by companies worldwide, ushering in another boom in AI development. During this period, "knowledge reasoning" became a research hotspot of mainstream AI. However, later, the difficulty of maintaining and upgrading the expert system brought another cold winter in the history of AI development. The AI up to this point is mainly based on computing and storage, called the first generation of AI. They proposed a reasoning model based on knowledge and experience and used it to simulate human intelligence behavior, such as reasoning, planning, and decision-making. One of the most representative results of an AI system based on this principle is the chess program of IBM Deep Blue, which defeated the world champion in 1997. Deep blue could process 200 million possible moves per second and determine the optimal next move looking 20 moves ahead (Campbell, Hoane Jr, & Hsu, 2002). Generally speaking, the primary theme of this period is the steady development of AI, and various fields related to AI have made significant progress.

The third wave of AI begins after the arrival of deep learning (LeCun, Bengio, & Hinton, 2015), which accelerates the development and progress of society tremendously. It is not long since deep learning made a tremendous breakthrough in solving the challenges of ImageNet in 2012. However, the rise of deep learning made AI everywhere these days, which is an enormous victory for connectionist approaches. The prosperity of AI manifest in several significant phenomena: (1) Important new deep learning algorithms and non-deep learning algorithms emerge in endlessly, such as Convolutional Neural Networks (CNN), RNN, Generative Adversarial Networks (GAN) (Goodfellow et al., 2014), VAE, Federated Learning (McMahan, Moore, Ramage, Hampson, & y Arcas, 2017), Transfer Learning (Pan & Yang, 2009), etc.; (2) AI has overcome many problems that we could not solve, especially in computer vision and NLP (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013); (3) AI application and AI-related topics have been ingrained into people's routines, such as self-driving cars, knowledge graphs, Robots of Boston Dynamics, Sophia, IBM Watson, AlphaGo-Master, etc. Besides, the development of ImageNet has also accelerated the emergence of AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) and ResNet (He, Zhang, Ren, & Sun, 2016) models. Simultaneously, many optimization methods for neural networks have appeared, such as Gradient method, Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), Batch Normalization, Layer Normalization, etc. Various AI open platforms like OpenAI and Mila began to appear, as well as some deep learning frameworks are widely used, such as TensorFlow (Abadi et al., 2016), PyTorch, etc. In addition, the pre-training model represented by BERT (Devlin, Chang, Lee, & Toutanova, 2019) has achieved rapid development in recent years.

To summarize, it is of utmost importance that tracing the evolution of AI in the past decade from the development of connectionist approaches is becoming more urgently needed. This study proposes the bibliometric analysis and literature review method for the in-depth exploration of the research topic. The methodology and the specific research priorities are in detail described in the sections below.

3. Methodology & research priorities

This paper initiates the idea of combining traditional literature review, bibliometric methods, and SciSci for the study. Each method has inherent strengths and weaknesses, but in combination, these approaches complemented each other. The proposed joint approach can identify research highlights and order the development context of connectionist approaches technologies. This section mainly elaborate our methodology from the platform and tools, the methodology, and the research priorities of this paper.

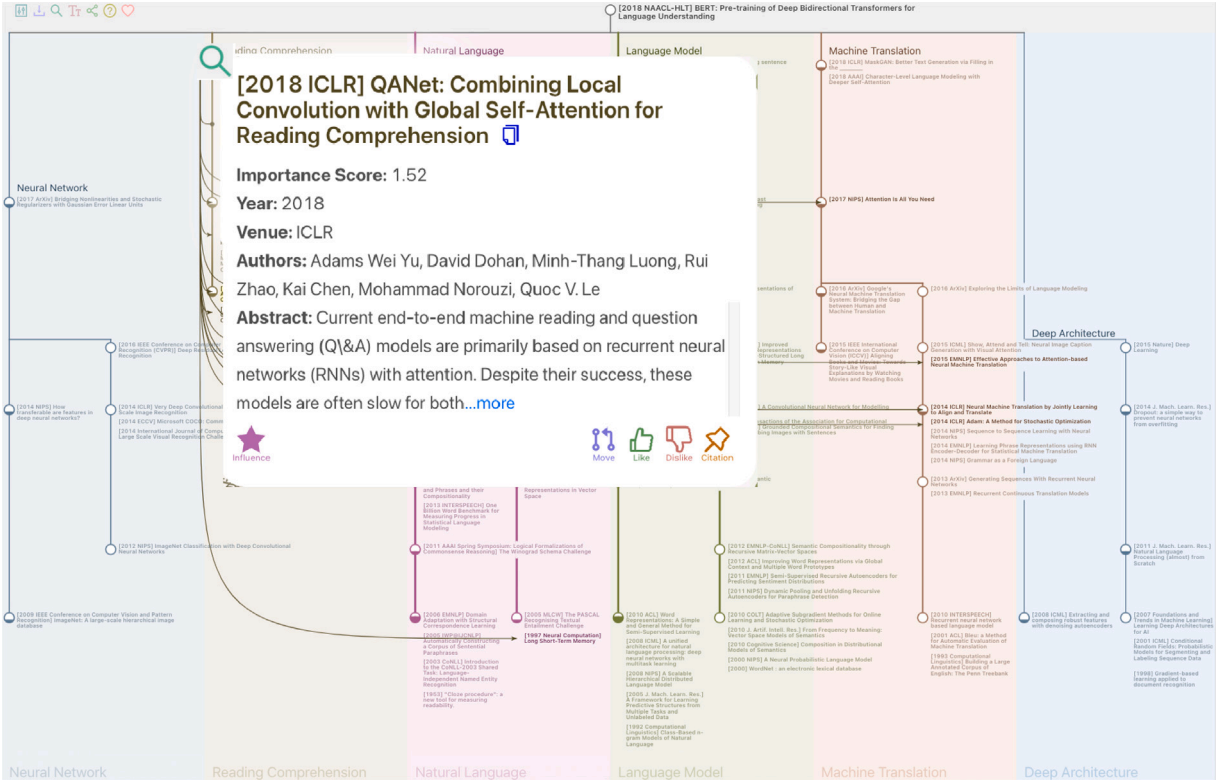

Fig. 1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.[8]

3.1. Master reading tree

AMiner[3] (the second generation of ArnetMiner (Tang et al., 2008)) is designed to search and perform data mining operations against academic publications on the Internet, using social network analysis to identify connections between researchers, conferences, and publications. On the one hand, AMiner provides us with enough academic data for quantitative analysis. On the other hand, AMiner provides us with a series of machine learning-based applications, such as Scholar Trajectory,[4] Topic Trend,[5] Master Reading Tree (MRT),[6] KnowledgeAtlas,[7] etc. Based on AMiner data and these tools, much quantitative analysis has been done from the perspective of SciSci in AMiner. In this paper, MRT gives great importance to our study.

MRT is a tool designed for helping scholars learn the evolution of publications by plotting a functional citation flow map. Using this tool, we can build annotated evolution roadmaps for publications and identify important previous works or evolution tracks by generating expressive embeddings and clustering them into various groups. In MRT, users can see how references are clustered and contribute to the source paper differently. The computation of MRT follows four main steps: Retrieving, Reading, Roadmapping, Reasoning. Yin, Tam, Ding, and Tang (2021) integrated the proposed MRT framework on AMiner to help users generating roadmaps using MRT for free. On the one hand, it can help us analyze the complex history of technology development with reliable bibliometrics analysis. On the other hand, it gives us an excellent perspective to predict the future development of technology from the structured visual graph.

The reasoning function of MRT tries to learn the reason why works correlate to each other. An MRT analysis of BERT (Devlin et al., 2019) in Fig. 1 shows the detail of the process. A dynamic and interactive web page is available free online, which will bring us more details and can help MRT refine the results. What is more, the details of Fig. 1 will be described in the following sections in combination with other pre-trained models for NLP.

3.2. Methodology

To fundamentally promote AI development, it is necessary to thoroughly understand what has been tried in history and why the current technology exists in its current form. This paper tries to summarize fundamental AI development changes from the perspective of connectionist approaches and explain these technologies' main ideas and their relationship with previous studies. Previous studies on the history of a field mainly relied on literature review. The development of academic mining platforms provides us unlimited ways to explore a specific domain from the perspective of SciSci (Fortunato et al., 2018). This study describes the history of AI from the perspective of connectionist approaches by a picture throw bibliometric analysis as well as NLP (Natural Language Processing) and knowledge graph-based keywords method, which shows its application subfields, critical technologies, heat degree, and milestones. In Fig. 2, lines represent the development of application subfields and critical technologies of AI from the perspective of connectionist approaches (denote by SC), the width of a line represents its heat degree in that year, and some remarkable milestones is labeled on the side. The generation process of the underlying data in Fig. 2 is as follows.

(1) find the keywords of SC from academic knowledge graph in AMiner;
(2) find the keywords of each paper by NLP from AMiner AI paper data set;

[3] https://www.aminer.cn/
[4] https://map.aminer.cn/geo/touch_v2/trajectory/
[5] https://trend.aminer.cn/
[6] https://www.aminer.cn/mrt/
[7] http://knowledgeatlas.aminer.cn/
[8] See more clearer online graph at: https://mrt.aminer.cn/5dd3de98e07b013b38cf3399.

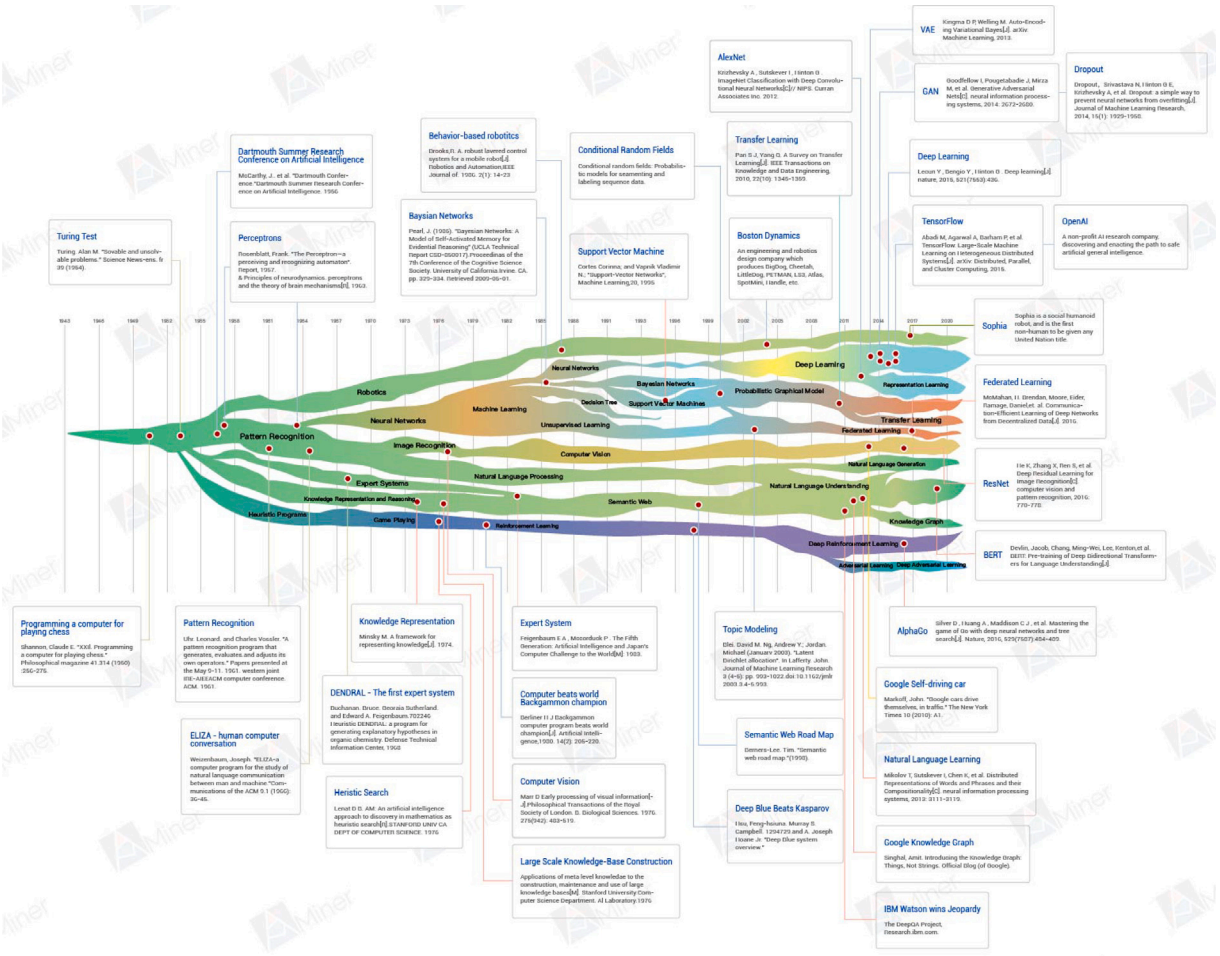

Fig. 2. AI History (mainly from the perspective of connectionist approaches). Some key scholars, papers, events, technologies, fields, points in time, applications, and relationships between them.[10]

(3) find the relationship between papers and *SC* by matching their keywords after disambiguation through MRT;
(4) measure the heat degree of *SC* in a year by its number of papers[9];
(5) find some milestones by expert inquiry referring some highly cited/recommended papers;
(6) add some individual knowledge about AI as a supplement.

In this way, based on the combining and description of the AMiner technology intelligent engine, we use a two-dimensional graph to show the development course of the most important fields and technologies of AI (mainly from the perspective of connectionist approaches) by combining the above knowledge. Fig. 2 is a statistical time-series diagram, whose milestone achievements of AI from the perspective of connectionist approaches in the development history of more than 60 years in the form of a river map, hoping the rising and fall would appear in front of us with a clearer picture.

This paper gives an overview of the history of AI as well as its significant trends shortly, summarizes the vital significance behind these, and on this basis, proposes many directions to guide future AI research. Of particular significance, this figure gives more detailed information about the evolution of AI in the last decade, especially model and algorithm innovation. Going even further, this study lists the main contributions of these models in Table 1 by the above steps, which will be introduced in detail in the following section.

In this way, this study constructs a method to find fine-grained technical details, even their relationships, from a large number of papers. This method brings convenience for tracing the evolution of AI in the past decade from the development of connectionist approaches by literature review. The development of connectionist approaches is multi-faceted. It is meaningful to sort out the focuses, describe the dynamic trends, and predict the future research directions using bibliometric analysis methods.

3.3. Research priorities

According to the idea and experimental design of Fig. 2, the development of AI subfields and the critical technologies of AI are interrelated. Further analysis shows that deep learning is undoubtedly the most critical event in the last decade, the development of which is represented by CNN, RNN, GAN, etc. What is more, this paper uses a network-like graph to show the most critical development context of deep learning by MRT. The detailed information is shown in Fig. 3 as follows.

Based on the proposed method, Fig. 3 sheds new light on the analysis of deep learning over the last decade. The main research idea highlights critical technologies of connectionist approaches in the past decade, namely deep learning algorithms and their logical connections. From this point, it benefits not only to summarize the essential characteristics of the second generation of AI but also to foresee its changes

[9] Due to the delay in the inclusion of papers, the data for the last few years were estimated from the fitting function.
[10] See more clearer online figure at: https://www.aminer.cn/ai-history. The figure omits some contributions for ease of illustration.

Table 1
The development overview and main contributions of these models.

Model	Year	Architecture	Main contribution	References	Model	Year	Architecture	Main contribution	References
LeNet	1998	CNN	First popular CNN architecture	LeCun, Bottou, Bengio, and Haffner (1998)	CGAN	2014	GAN	The conditional version of GAN	Mirza and Osindero (2014)
ALexNet	2012	CNN	Deeper and wider than the LeNet	Krizhevsky et al. (2012)	StyleGAN	2018	GAN	Redesign the architecture of the generator network	Karras, Laine, and Aila (2019)
NIN	2013	CNN	Proposed MLP convolutional layer; Dimensionality reduction or increase of the number of channels	Lin, Chen, and Yan (2014)	DCGAN	2016	GAN	Practiced de-CNN structural design that significantly stabilizes GAN training	Radford, Metz, and Chintala (2016)
ZFNET	2013	CNN	Visualize the intermediate layer	Zeiler and Fergus (2014)	SinGAN	2019	GAN	can learn from a single natural image	Shaham, Dekel, and Michaeli (2019)
VGG	2014	CNN	Homogeneous topology;Uses small size kernels	Simonyan and Zisserman (2015)	RNN	1997	–	A model that can process sequence data	Zaremba et al. (Hopfield, 1982)
GoogleNet	2015	CNN	Introduced block concept;Split transform merge idea	Szegedy et al. (2015)	LSTM	1997	RNN	Solve the problem of gradient disappearance in RNN	Hochreiter and Schmidhuber (1997)
ResNet	2016	CNN	propose identity shortcut connection	He et al. (2016)	GRU	2014	RNN	It performs similarly to LSTM but is computationally cheaper	Kyunghyun Cho et al. (Cho et al., 2014)
DenseNet	2017	CNN	Dense connection;Feature reuse	Huang, Liu, Van Der Maaten, and Weinberger (2017)	Attention	2014	–	Focus on the important points and ignore other unimportant factors	Bahdanau, Cho, and Bengio (2014)
SENet	2018	CNN	Propose the SE module	Hu, Shen, and Sun (2018)	Transformer	2017	Attention	Abandoning CNN and RNN, the entire network structure is completely composed of Attention mechanism	Vaswani et al. (2017)
MobileNet	2017	CNN	build light weight deep neural networks	Howard et al. (2017)	DQN	2013	–	Combine neural network and Q learning	Mnih et al. (2013)
ShuffleNet	2018	CNN	greatly reduce computation cost while maintaining accuracy	Zhang, Zhou, Lin, and Sun (2018)	DPG	2014	–	Introduce an off-policy actor–critic algorithm	Silver et al. (2014)
EfficientNet	2019	CNN	Proposed a new model scaling method	Tan and Le (2019)	DDQN	2015	DQN	Eliminate the problem of overestimation by decoupling the target Q value selection and calculation process	van Hasselt, Guez, and Silver (2016)
RegNet	2019	CNN	Explore the structure aspect of network design and arrive at a low-dimensional design space	He et al. (Radosavovic, Kosaraju, Girshick, He, & Dollár, 2020)	TD3	2018	DQN	Solve the problems of overestimation bias and high variance	Fujimoto, Hoof, and Meger (2018)
MoCo	2019	CNN	Build a dynamic dictionary with a queue and a moving-averaged encoder	He, Fan, Wu, Xie, and Girshick (2020)	DDPG	2015	DPG+DQN	Can handle continuous motion output and large motion space	Lillicrap et al. (2016)
SimCLR	2019	CNN	Opened the door to self-supervised learning and contrastive learning	Chen, Kornblith, Norouzi, and Hinton (2020)	A3C	2016	DPG+DQN	A deep reinforcement learning algorithm that is better and more versatile than DQN	Mnih et al. (2016)
DAE	2008	GM	Add noise to the original sample, and hope to use DAE to restore the noise sample to a pure sample	Vincent el al. (Bengio, Yao, Alain, & Vincent, 2013)	ELMo	2018	LSTM	Introduce a new type of deep contextualized word representation method	Clark et al. (Peters, Neumann, Iyyer, Gardner, & Zettlemoyer, 2018)
RBM	1986	GM	Can be used for dimensionality reduction, classification, regression, collaborative filtering, feature extraction and topic modeling	Hinton et al. (Smolensky, 1986)	GPT	2018	Transformer	Introduced Semi-supervised method.	Radford, Narasimhan, Salimans, and Sutskever (2018)
DBN	2006	GM	It is a probabilistic generative model that establishes a joint distribution between observation data and labels	Hinton et al. (Ranzato, Susskind, Mnih, & Hinton, 2011)	GPT2	2019	Transformer	Can use unsupervised pre-trained models to do supervised tasks	Radford et al. (2019)
AE	1989	GM	The popular type of GM	Bourlard and Kamp (1988)	BERT	2019	Transformer	Achieve the best results in 11 different NLP tests	Devlin et al. (2019)
VAE	2013	GM	Comprises a probabilistic encoder and a probabilistic decoder	Kingma et al. (Kingma & Welling, 2014)	SpanBERT	2020	Transformer	Improving Pre-training by Representing and Predicting Spans	Joshi et al. (2020)
VQ-VAE-2	2019	GM	an improved version of Vector Quantized VAE that generates a large size image	Razavi, van den Oord, and Vinyals (2019)	RoBERTa	2019	Transformer	A better pre-training model training method is proposed based on BERT	Liu, Ott, et al. (2019)
LSGAN	2017	GAN	uses the least-squares loss function instead of the original cross-entropy loss function	Mao et al. (2017)	XLNET	2020	Transformer	Adopted Permutation Language Modeling method	Yang et al. (2019)
WGAN	2017	GAN	improves GAN from the perspective of the loss function	Arjovsky, Chintala, and Bottou (2017)	Mass	2019	Transformer	It is a Seq2Seq model pre-trained using GPT+BERT	Song, Tan, Qin, Lu, and Liu (2019)
GAN	2014	GM+DM	Setting up generator network and discriminator with different objects	Goodfellow et al. (Goodfellow et al., 2014)	GPT3	2020	Transformer	The amount of parameters is ten times larger than Turing NLP	Brown et al. (2020)

in the next generation. Based on the knowledge of deep learning technologies, we decide the main concerns of this paper by involving repeated cycles of technologies categorization, inductive analysis, and refinement of categories. Finally, this study mainly focuses on five parts: self-learning and self-coding algorithms, RNN algorithms, reinforcement learning, pre-trained models, and other typical deep learning algorithms, which will be discussed in more detail in the following section. It is indisputable that other topics in the field of AI over the last decade are also worth exploring and studying, such as ethics of AI, AI privacy, etc. However, the development of deep learning algorithms has come first over the past decade. This paper mainly involves the five significant concerns, covering all the critical technologies and their evolutions in Fig. 2 over the last decade. In this way, we will provide a particular perspective and comprehensive analysis of the past decade of AI mainly from the perspective of connectionist approaches.

4. Development of algorithms for connectionist approaches

Undoubtedly, the development of deep learning algorithms is the most crucial cause why connectionist approaches made an outstanding contribution to AI development in the past decade. According to our methodology, this study mainly focuses on the following five research highlights, including the primary research breakthroughs of the past decade. This section will elaborate on their development history and logical connection by literature review supported by MRT.

4.1. Self-learning and self-coding algorithms

Since the birth of neural networks, supervised neural networks have always been the focus of researchers, and significant progress has also been made. The development and introduction of generative models, such as Restricted Boltzmann Machine (RBM), GAN, and AutoEncoder (AE), accelerate the development of unsupervised neural networks. However, supervised neural networks are getting deeper and more accurate, but the progress of unsupervised neural networks is far worse than supervised neural networks.

Generative models are classic models for unsupervised learning. The development process of the generative model is shown in Fig. 4. The earliest generative models are Boltzmann Machine (BM) and RBM. BM and RBM are probabilistic graphical models that a stochastic neural network can explain. In 1986, Smolensky proposed RBM based on BM (Smolensky, 1986). The so-called stochastic means that the neurons in the network are random neurons and the output state is only activated and inactivated. Especially, the statistical law of probability determines the specific value of the state (Fischer & Igel, 2012).

The idea of AutoEncoder dates back to 1988 (Bourlard & Kamp, 1988). The model at that time was challenging to optimize due to

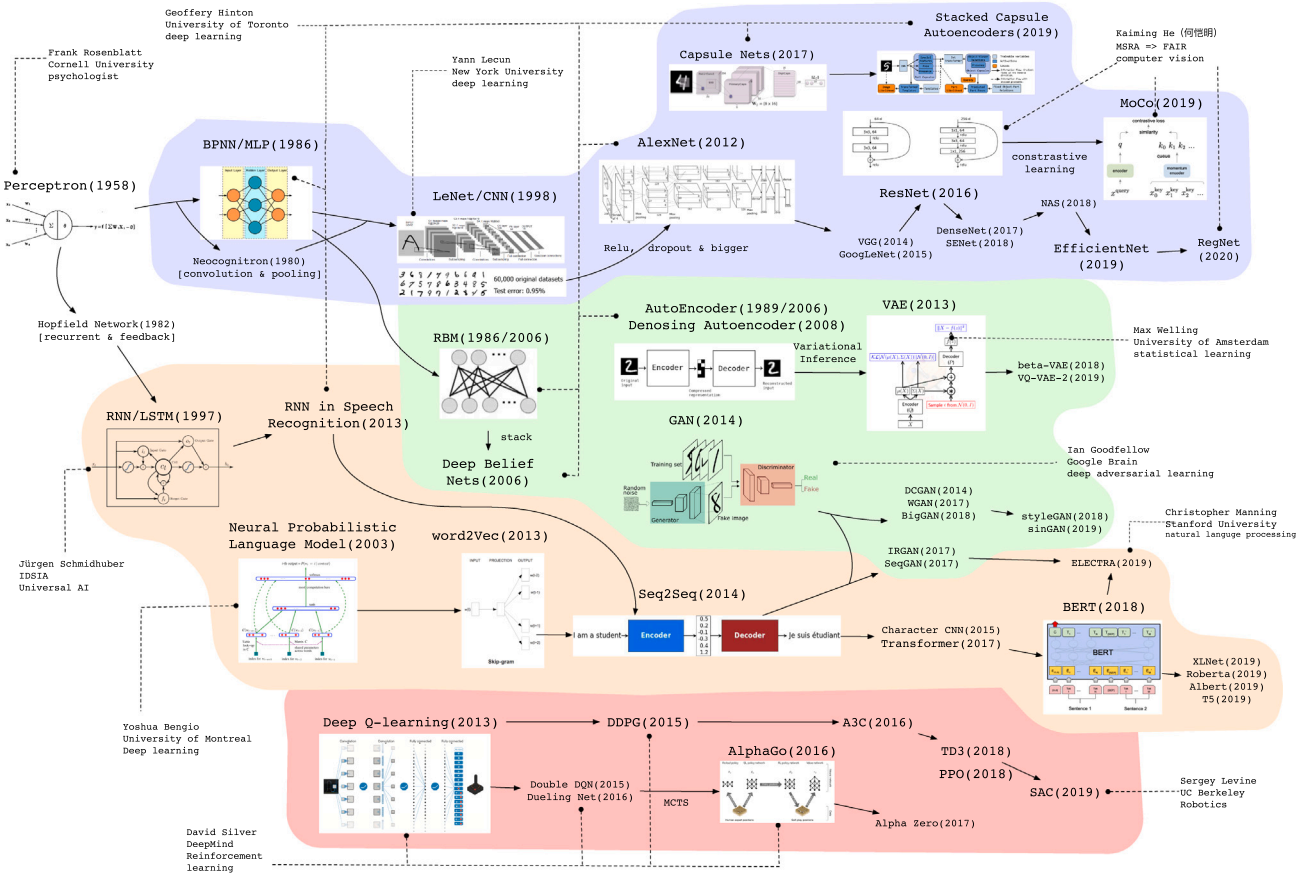

Fig. 3. Most Critical Development Context of Deep Learning.

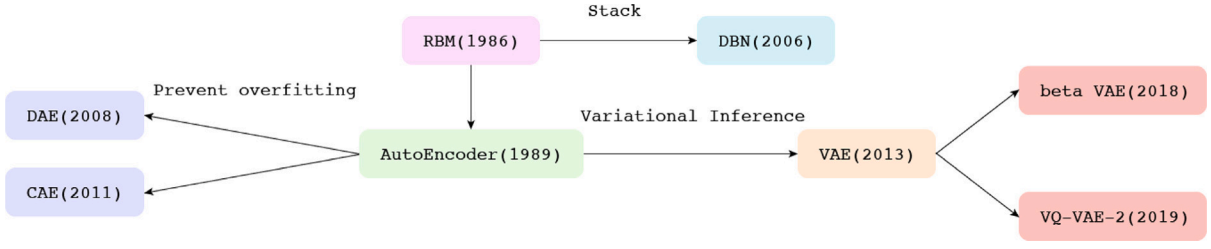

Fig. 4. The development history of AutoEncoder.

sparse data and high computational complexity. Until 2006, Hinton et al. used gradient descent to optimize RBM layer by layer to achieve an abstract representation of the original sample, which achieved significant results in feature dimensionality reduction, which made the method of using neural networks to build AutoEncoder gain widespread attention. An AutoEncoder is a typical unsupervised learning algorithm, which aims to set the target values so that they are equal to the original input (Dong, Liao, Liu, & Kuang, 2018). The AutoEncoder framework includes two modules: the encoding process and the decoding process. The input samples are mapped to the feature space through the encoding process, and then the abstract samples are mapped back to the original space through the decoding process to obtain reconstructed samples. AutoEncoder does not need to use the label of the sample in the optimization process. In essence, the input of the sample is simultaneously used as the input and output of the neural network, and the abstract feature representation of the sample is learned by minimizing the reconstruction error. This unsupervised optimization method significantly improves the versatility of the model. However, because AutoEncoder learns the unique abstract representation of each sample through a neural network, this will lead to overfitting when the neural network parameters are complex to a certain extent. Vincent et al. proposed Denoising AutoEncoder (DAE) (Bengio et al., 2013) to solve this problem, which adds random noise to the input layer of the traditional AutoEncoder to enhance the robustness of the model. Rifai et al. proposed Contractive AutoEncoder (CAE) (Rifai, Vincent, Muller, Glorot, & Bengio, 2011) by adding the Jacobian matrix paradigm of the encoder to the AutoEncoder objective function as a constraint to learn the abstract features of anti-interference.

Because AutoEncoders are trained on specific samples, their applicability is greatly limited to data similar to training samples. VAE (Kingma & Welling, 2014) is a generative model proposed by Kingma and Welling in 2014, which can be used to generate samples that are not in the training samples. VAE uses two neural networks to establish two probability density distribution models. One is used for the variational inference of the original input data to generate the variational probability distribution of the hidden variables, which is called the inference network. The other is based on the generated variational probability distribution of the hidden variables, restoring

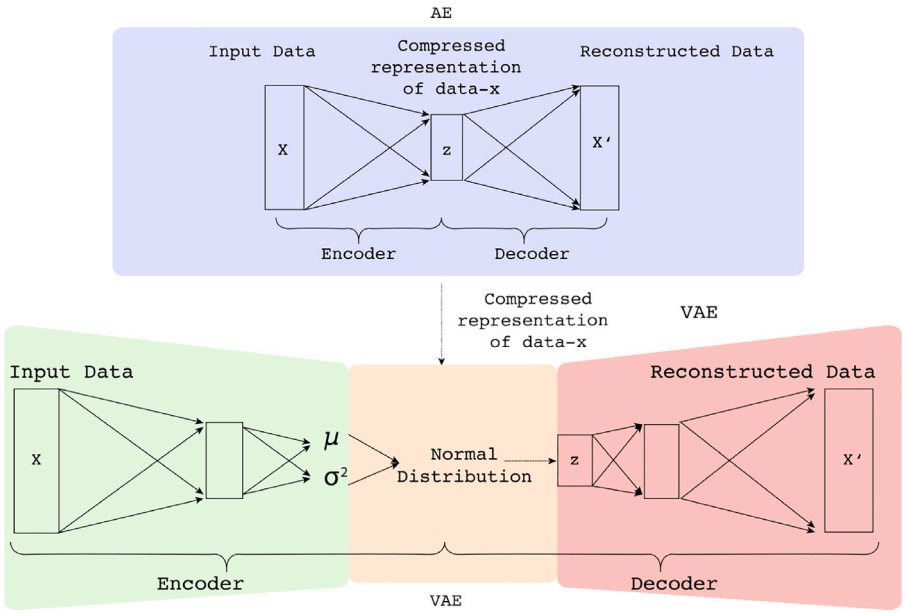

Fig. 5. AE and VAE comparison.

the approximate probability distribution of the original data, called the generation network. The overall structure of Variational AutoEncoder (VAE) is similar to the AutoEncoder. Fig. 5 demonstrates the relationship between VAE and AE (AutoEncoder). However, the working principle of VAE is entirely different from that of AutoEncoder. The output of VAE's "encoder" and "decoder" is the probability density distributions of variables subject to parameter constraints rather than a specific code. VAE no longer maps the input X to a fixed abstract feature Z, but assumes that the abstract feature Z of sample X obeys a normal distribution, generates abstract features through distribution, and finally gets the output decoder based on abstract features. Since the abstract feature Z is generated by sampling from a normal distribution, the encoder part of VAE is a generative model and then combined with the decoder to achieve reconstruction to ensure that the information is not lost. VAE is a milestone research result that combines probability graphs to enhance the robustness of the model. There are many subsequent extensions based on VAE in the past decade, including InfoVAE (Zhao, Song, & Ermon, 2017), betaVAE (Higgins et al., 2017), and factorVAE (Kim & Mnih, 2018). Later, Vector Quantized Variational AutoEncoder (VQ-VAE) (van den Oord, Vinyals, & Kavukcuoglu, 2017) was proposed. Unlike VAE, VQ-VAE discretizes the hidden variable z through vector quantization. Each dimension of z in VQ-VAE is a discrete integer, which can successfully model important features that usually span many dimensions in data space. In the case of using discretized variables, VQ-VAE obtains similar accuracy as using continuous latent variables, but there is a problem of blurred images. In 2019, VQ-VAE-2 (Razavi et al., 2019) was proposed, which layered the encoder and decoder of VQ-VAE. The bottom layer models the local feature, and the top layer models the global feature. At the same time, in order to allow the top layer to effectively extract the global information, adding self-attention to the network, which can generate very clear high-resolution pictures.

The latest development is an AutoEncoder that combines adversarial ideas. In 2014, Goodfellow et al. proposed a GAN (Goodfellow et al., 2014), which set off a revolution in Deep Learning. GAN has two networks, which are the generator network and discriminator network, respectively. The generator network tries to generate real data. In contrast, another network tries to discriminate between real and generated data, repeatedly iterating until the generative and discriminant models cannot improve. That is, the discriminant model cannot judge whether a picture is generated or actually exists. The design of the GAN architecture is shown in Fig. 6.

In the last decade, the emergence of GAN has greatly promoted the research of unsupervised learning and image generation. It has been extended to various fields of computer vision, such as image segmentation, video prediction, and style transfer. CycleGAN (Zhu, Park, Isola, & Efros, 2017) is a crucial application model of GAN in the image field, which can transform a crying face image into a smiling face. StarGAN (Choi et al., 2018) has expanded the ability to turn a smiley face into multiple expressions. In the video field, Mathieu, Couprie, and Lecun (2016) applied GAN training to video prediction for the first time. After that, Vondrick, Pirsiavash, and Torralba (2016) also made significant progress, using GAN to generate 32 real video frames with a resolution of 64*64. In the field of human–computer interaction, Santana and Hotz (2016) have realized assisted autonomous driving using GAN, which is widely studied and concerned because of its power generation ability.

Since 2016, many improved GAN-based structures have appeared. The evolution of GAN is shown in Fig. 7. The original GAN generation process can start training with random noise, no longer need a hypothetical data distribution. However, such an accessible and sloppy way is not very controllable for larger images. Conditional Generative Adversarial Nets (CGAN) proposes a GAN with conditional constraints (Mirza & Osindero, 2014), adding conditions to the model through additional information to guide the data generation process. In this way, CGAN turns a purely unsupervised GAN into a supervised model. The Deep Convolutional Generation Network (DCGAN) (Radford et al., 2016) uses a convolutional layer instead of a fully connected layer. Both the generator and discriminator of DCGAN discard the pooling layer of CNN, the discriminator retains the overall architecture of CNN, and the generator replaces the convolutional layer with deconvolution to better extract image features. Both the generator and discriminator of DCGAN discard the pooling layer of CNN. However, the discriminator retains the overall architecture of CNN. Moreover, the generator replaces the convolutional layer with deconvolution to better extract image features. The discriminator and generator exist symmetrically, which dramatically improves the stability of GAN training and the quality of the generated results. However, DCGAN does not fundamentally solve the problem of instability in GAN training. It is still necessary to carefully balance the training of generator and discriminator, often training one multiple time and training the other once. Moreover, the resolution of generated images is also relatively low, which is also a common problem in various generative confrontation networks. WGAN (Arjovsky et al., 2017) mainly improves GAN

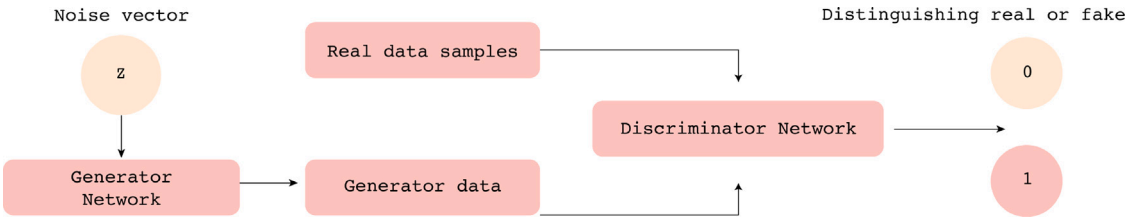

Fig. 6. The design of GAN architecture.

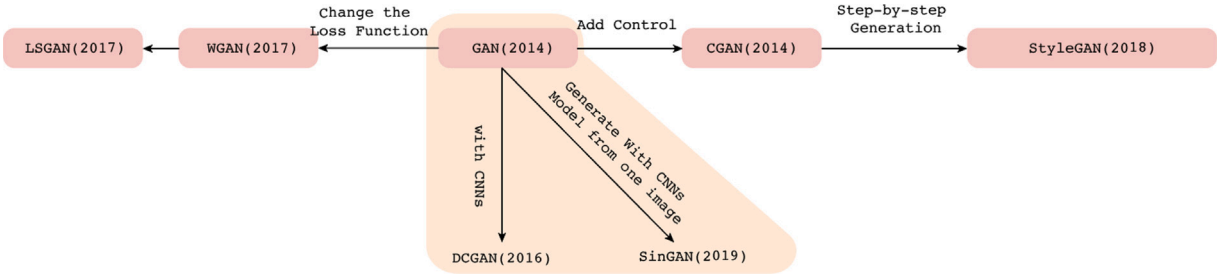

Fig. 7. The development history of GAN.

from the perspective of the loss function, whose improvement makes it possible to achieve good performance even on the fully connected layer. Wasserstein distance is used instead of Jensen–Shannon divergence to measure the distance between real data and generated data. Theoretically, solve unstable training and no longer need to balance the training of generator and discriminator carefully. LSGAN (Mao et al., 2017) uses the least-squares loss function instead of the cross-entropy loss function of the original GAN. Using cross-entropy as the loss function will make the generator no longer optimize the generated pictures recognized as real pictures by the discriminator, even if these generated pictures and real pictures. There is still a big gap between these two loss functions. The least-squares loss function requires the generator to pull the generated pictures far from the decision boundary to the decision boundary under the premise of confusing the discriminator. Summarizing the development history of the previous GAN, with the help of the discriminator, the generator realized the conversion from simple distribution to complex distribution. Nevertheless, how can we generate high-quality large pictures? In 2018, NVIDIA released the StyleGAN model (Karras et al., 2019), which refers to the step-by-step generation of ProGAN from small to large (Karras, Aila, Laine, & Lehtinen, 2018) while adding control to it to produce a high-definition picture with rich details. In 2019, SinGAN was proposed (Shaham et al., 2019), bringing GAN into a new field, learning unconditional generative models from a single natural image. It can capture the internal block distribution information of an image and generate high-quality and changeable samples with the same visual content.

4.2. RNN algorithms

The RNN is a neural network that takes sequence data as input and all nodes are connected in a chain. It can memorize the previous information and apply it to the calculation of the current output. The development process of RNN is shown in Fig. 8. As early as 1982, John Hopfield discovered a single-layer feedback neural network (Hopfield Network (Hopfield, 1982)), that is, the idea of adding a recurrent network to the neural network. The Hopfield network is the earliest prototype of RNN, which structure is shown in Fig. 9 below.

However, there was no suitable application scenario due to the complex implementation of the early Hopfield network, so a forward neural network gradually replaced it. In 1997, Hochreiter and Schmidhuber proposed a long short-term memory (LSTM) network (Hochreiter & Schmidhuber, 1997), which significantly promoted the development of RNN. Its core is to adjust the focus of memory according to the training goal and then perform the entire string encoding, which can effectively alleviate the gradient message and gradient explosion problems. Gated recurrent unit (GRU) is a lightweight variant of LSTM, which has made some simplifications and adjustments based on the LSMT model, saving much time when the training data set is relatively large (Cho et al., 2014).

Although previous research has made researchers pay attention to RNN, it was Hinton et al. who used RNN for speech recognition in 2013 that drew widespread attention to the sequential neural network model. In 2014, the Bengio team (Cho et al., 2014) and Google (Sutskever, Vinyals, & Le, 2014) proposed the Sequence-to-Sequence (Seq2Seq) architecture. Although DNN has achieved outstanding results in image classification and other issues before its proposal, their input and output must be fixed-length vectors, which brings much trouble when the output length cannot be determined. Under this circumstance, the Seq2Seq framework is proposed. The most considerable improvement of this structure is that lengths are variable for input sequences and output sequences, which can be used for translation, chatbots, syntax analysis, text summarization, etc. However, there are also some shortcomings. For example, compressing a whole question sequence into a small vector through the encoder will make it difficult to decode perfectly without missing information through the decoder. Almost at the same time, the Bengio team proposed the Attention mechanism (Bahdanau et al., 2014). Unlike the Seq2Seq structure, the Attention model assigns different weights to different words. The output is no longer a fixed-length intermediate semantics in the encoder process but a sequence composed of vectors of different lengths. When each output is generated, the information of the input sequence can be fully utilized, which solves the problem of information loss. Later, Google proposed Transformer framework (Vaswani et al., 2017), which is improved based on the Seq2Seq model. The biggest problem of Seq2Seq is to compress all the information on the Encoder side into a fixed-length vector, which will lose much information on the Encoder side, and the Decoder side cannot pay attention to the information it interested in. Transformer improves the defects of Seq2Seq by introducing Multi-Head Attention. Transformer also introduced a self-attention module to make the source sequence and the target sequence "self-associate" first, making the embedding representation of the source sequence and the target sequence itself contain more information. At the same time, the parallel computing capabilities of Transformer far exceed the Seq2Seq series model.

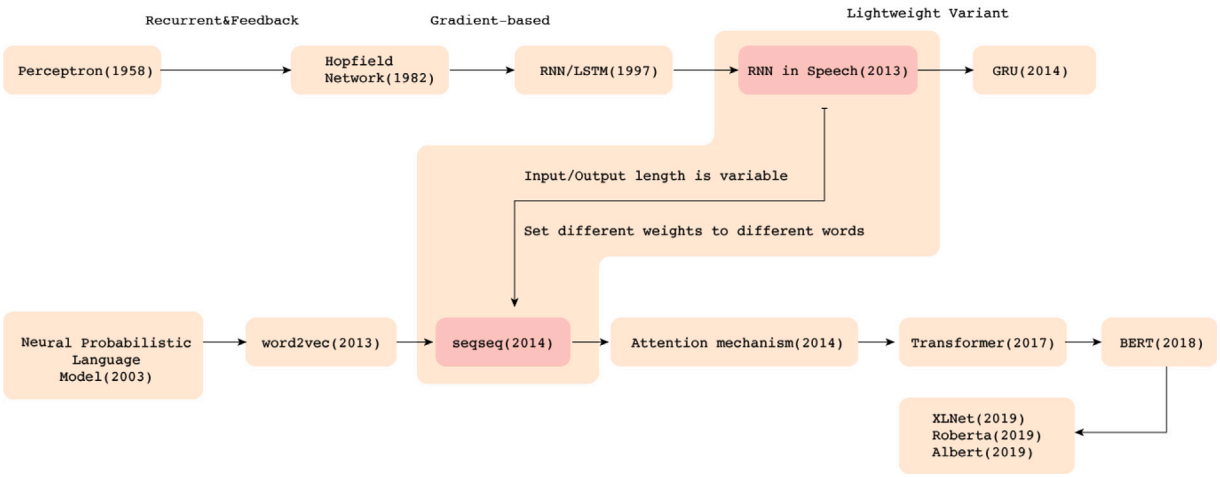

Fig. 8. The development history of sequence model.

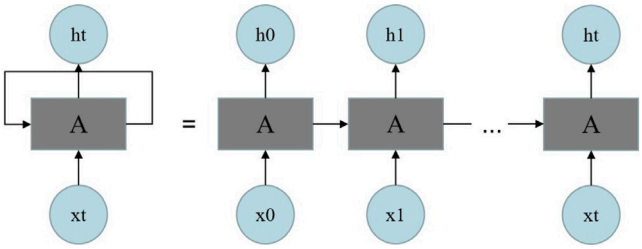

Fig. 9. RNN cell through time (Hopfield, 1982).

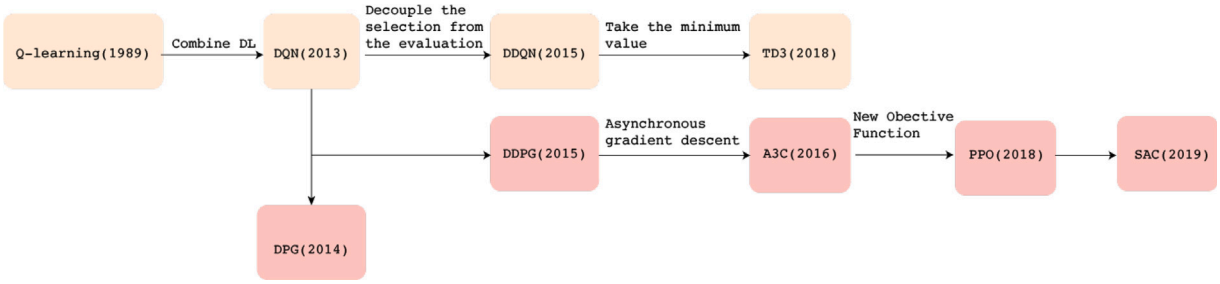

Fig. 10. The development history of Reinforcement Learning.

In the past decade, RNN was widely used in NLP, such as Classification, Question Answering (QA), Named entity recognition (NER), etc. For these different tasks, the earliest approach is to customize different models according to each type of task, enter the pre-trained embedding, and then use the task-specific data set to train the model. However, this approach faces some problems like insufficient data set, complicated calculations, etc. By referring to the idea of pre-training models in the image field, the NLP field also introduces a pre-training model method, using a general model to pre-train on a very large corpus, then fine-tune it according to the specific task, which perfectly solves the problem of insufficient data for some small sample tasks, simultaneously reduces the burden of training and manual annotation. More detailed information about pre-training models in the last decade will be introduced in Section 4.4.

4.3. Reinforcement learning

The development history of Reinforcement Learning (RL) is shown in Fig. 10. Q-learning is the most basic algorithm of RL, proposed by Watkins in 1989 (Watkins & Dayan, 1992). In Q-learning, a table of Q values is maintained. The dimension of the table is $S * A$, where S represents the number of states and A represents the number of actions. Each number in the table represents the discounted sum of the future benefits obtained by taking action A in the current state S. Continuously iterating the Q value table to make it converge, and finally, according to the Q value table, the algorithm can select an optimal strategy in each state. In ordinary Q-learning, when the state and action space are discrete and the dimension is not high, Q-Table can be used to store the Q value of each state–action pair. When the state and action spaces are with continuous high-dimension, using Q-Table becomes unrealistic. The solution for that is to transform the Q-Table update problem into a function fitting problem. Similar output actions can be obtained for similar states. However, there are some problems with the combination of Deep Learning and RL. For example, Deep Learning requires many labeled samples for supervised learning, while RL only has a reward return value. In Deep Learning, the samples are independent of each other, but the former state and the latter state are related in RL. The target distribution of Deep Learning is fixed, but it changes in RL, and the non-linear network representation value function in RL is unstable. Although the idea of combining Deep Learning and RL was tried a few years ago, the real success began with the publication of Mnih's article (Mnih et al., 2013). This

paper proposed the Deep Q-Learning method, in which neural network structure is designed, the Q value is fitted through the function, and the reward is used to construct the label, which solves the missing labels in Deep Learning. The Deep Q-Learning use a variant of the Q-Learning algorithm for training and use stochastic gradient descent to update the weights to solve correlation and non-static distribution. The Deep Q-Learning also use a neural network to generate the current Q value, and use another neural network to generate the target Q value to solve the instability of the non-linear network representation value function. The trained deep reinforcement learning models were tested in seven Atari 2600 games, of which the performance in six games surpassed all previous methods, and in three games, it exceeded the level of human experts.

Since this learning model is a CNN that combines Q-learning for training and learning, it is called Deep Q-Learning, and the corresponding network is called DQN. However, because DQN is an off-policy method, each time it learns, instead of using the actual action of the next interaction, it uses the action currently considered to be the most valuable to update the target value function. DQN will be a problem of overestimating the Q value. So, Hasselt proposed the Double Deep Q-Learning (DDQN) method and applied it to DQN to solve this problem (van Hasselt et al., 2016).

RL has multiplied since the successful application of DQN and has received significant attention from the academia. Before this, RL was mainly based on model-based models. Its characteristics are that the problem is highly pertinent, requires many artificial assumptions, and requires specific models to be embedded for different problems. An algorithm is suitable for all tasks, so it also improves the comparability between models. DQN relies on finding the maximum value of the action-value function in each optimal iteration. There is no way for DQN to output the action-value function of each action for the continuous action space, so it can only handle discrete, low-dimensional action spaces. Subsequently, the Deep Deterministic Policy Gradient (DDPG) was proposed (Lillicrap et al., 2016), combining Actor–Critic and DQN algorithms. Since Actor–Critic can usually improve stability, DDPG has adopted two lines of actor and critic to update the policy network. DDPG can be divided into "Deep" and "Deterministic Policy Gradient", and then "Deterministic Policy Gradient" can be subdivided into "Deterministic" and "Policy Gradient". Deep here means that the experience pool and dual network structure in DQN can promote the effective learning of neural networks. Deterministic means that our actor no longer outputs the probability of each action but a specific action, which is more helpful for learning in the continuous action space. Traditional DQN uses a "hard" mode of target-net network parameter update. That is, the parameters in eval-net are assigned to the past every certain number of steps. However, unlike traditional DQN, a "soft" mode of target-net network parameter update is adopted in DDPG. That is, every step is to update the parameters in the target-net network a little bit. Experience shows that this parameter update method can significantly improve the stability of learning. DDPG successfully introduces DQN into the continuous action space.

In order to alleviate the instability that occurs when the traditional policy gradient method is combined with the neural network, various depth gradient methods use an experience replay mechanism to eliminate the correlation between training data. However, the experience replay mechanism is memory-consuming and computationally expensive, and the agent can only update based on the data generated by the old strategy. Based on the idea of asynchronous reinforcement learning (ARL), Mnih et al. proposed a lightweight DRL framework (Mnih et al., 2016). This framework can use asynchronous gradient descent to optimize the parameters of the network controller. It can also combine a variety of RL algorithms. Among them, the Asynchronous Advantage Actor–Critic (A3C) algorithm (Mnih et al., 2016) performs best in various continuous action space control tasks.

Although the Actor–Critic method can rely on the interplay of the policy-value function to correct overestimation to a certain extent, as the number of steps increases, this error may lead to accumulation. In 2018, the TD3 algorithm was proposed (Fujimoto et al., 2018), mainly for the overestimation of the value function. Based on the DDPG network, the idea of clipped double Q-Learning was added to correct the overestimation. In the same year, the Proximal Policy Optimization (PPO) algorithm was also proposed (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017). The PPO algorithm is a new type of Policy Gradient algorithm. The Policy Gradient algorithm is very sensitive to the step size, but it is not easy to choose a suitable step size. Based on this, PPO proposes a new objective function that can be updated in multiple training steps in small batches, which solves the difficulty in determining the step length of the Policy Gradient algorithm.

Soft Actor–Critic (SAC) is an offline maximum entropy actor–critic algorithm, which surpasses the classic algorithms DDPG and PPO in performance and becomes a new generation benchmark. The current model-free deep reinforcement learning has two main challenges. One is the difficulty of sampling due to the complexity of the samples faced by on-policy. The second is that the off-policy faces many hyperparameters that require careful adjustment, which makes it difficult to obtain stable results. Online strategies such as PPO, A3C, etc., need a new sample every time the gradient is obtained, so the utilization rate is extremely low. On the other hand, offline strategies such as DDPG can use past samples. On the other hand, it requires a neural network to perform high-dimensional nonlinear estimation, leading to instability and convergence difficulties. SAC uses the maximum entropy framework based on the original method of maximizing expected rewards. In this framework, the goal of a participant is to maximize the expected return while also maximizing the entropy value, that is, to act as randomly as possible while completing the task. It has dramatically improved exploratory and stability, considering the advantages of high sample utilization and high stability.

4.4. Pre-trained models

Since the Deep Learning boom, the pre-training process has been a relatively conventional approach in the CV field. For the hierarchical CNN structure, neurons at different levels have learned various image features, and the hierarchical structure is formed from bottom-up features. The features extracted at the bottom layer are more versatile than others for tasks. Therefore, the pre-trained parameters at the bottom layer are generally used to initialize the network parameters of a new task. While the high-level features are more related to the task, fine-tuning technology can fine-tune the model. This approach is practical and can achieve good results. Researchers in the NLP field migrate the ideas of the Pre-trained Language Model (PLM) in the CV field. The task of NLP can be divided into two parts: pre-training to generate word vectors and to operate on word vectors.

The development process of PLM in NLP is shown in Fig. 11. Word embedding (Bengio, Ducharme, Vincent, & Jauvin, 2003) in 2003 was an early pre-trained technology in NLP, but its performance in downstream tasks was limited. In 2013, Google proposed Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013), and pre-trained technology in the NLP field gradually entered people's field of vision. In English, the polysemy of a word is very common, but in Word Embedding, the same word occupies the parameter space of the same line when a word is encoded. That means two different contextual information will be encoded into the same word embedding space. So word embedding cannot distinguish the different semantics of polysemous words and cannot understand the complex context, which is its main disadvantage. To solve this problem, Embedding from Language Models (ELMo) (Peters et al., 2018) was proposed. Before ELMo, Wording Embedding was essentially a static method. After training, the expression of each word is fixed, which means the word embedding of this word will not change with the context. ELMo dynamically adjusts word embedding according to the current context and builds text representation through a deep bidirectional language

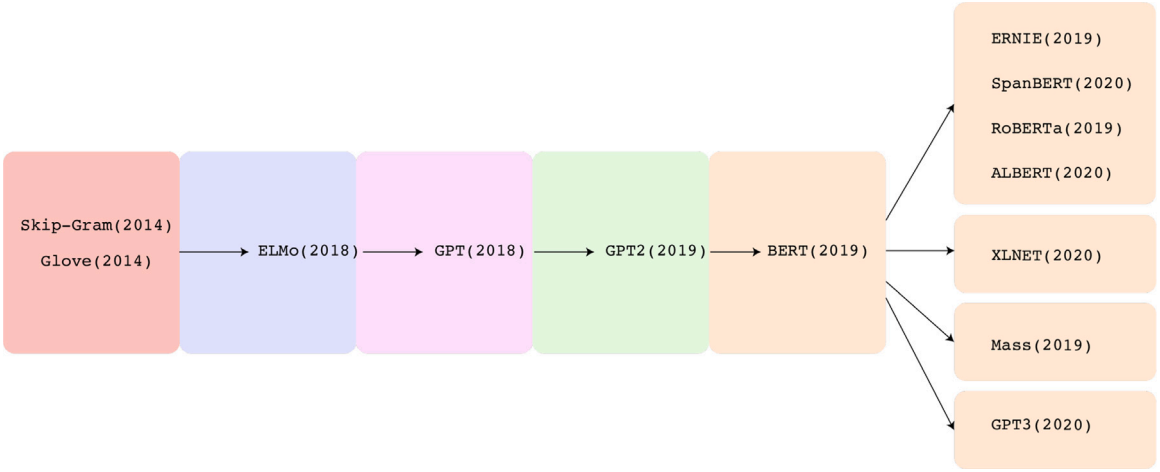

Fig. 11. The development history of PTM in NLP.

model. The problem of polysemous words is almost perfectly solved. ELMo pre-trains a Bi-directional LSTM language model from a large-scale unsupervised corpus. Compared with a one-way model, it can more easily capture contextual information and effectively improve the performance of the model. The process of ELMo is divided into two stages. The first stage uses the language model for pre-trained on a large-scale corpus. The second stage is when doing downstream tasks, the word embeddings of each layer of the network that extract the corresponding words from the pre-training network are added as new features to the downstream tasks. ELMo is a typical pre-training model based on feature fusion. From Word2Vec to ELMo, models gradually transfer the downstream specific vector operation tasks to pre-training to generate word vectors.

The Generative Pre-Training (GPT) (Radford et al., 2018) model follows the practice of using the pre-trained model parameters as the starting point of the downstream supervision model in ELMo. The difference between GPT and ELMo is that GPT uses Transformer as the feature extractor, which has stronger extraction capabilities. GPT adopts a traversal approach to transform structured input into an ordered sequence that can be processed by the pre-trained model, strengthening the migration ability of a language model. Subsequently, GPT-2 (Radford et al., 2019) was proposed. Its language model is universal and does not need to be fine-tuned according to different downstream tasks. However, GPT is still a one-way language model with limited modeling capabilities for semantic information.

Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2019) is also a typical fine-tuning model structure. The idea of BERT is to use the encoding stage of Transformer to generate a language model. The prototype of Transformer is two independent mechanisms, an encoder needs to receive a text as input, and a decoder is responsible for predicting the results of the task. The goal of BERT is to generate a language model, so only the encoding mechanism is needed. When training a language model, one challenge is to define a prediction target. Models such as Generative Pre-Training (GPT) predict the next word in a sequence, but the Bi-directional method has limitations in such tasks. BERT uses Masked Language Model (MLM) and Next Sentence Prediction (NSP) strategies to overcome this problem. When GPT predicts words, it only predicts the next word, so only the information above can be used. In contrast, BERT predicts words deducted from the text, which can make full use of contextual information, making the model more expressive. In some NLP tasks, sentence relationships need to be judged, such as whether two sentences have the same meaning and whether the NSP strategy of BERT can solve this problem well. The invention of BERT is a landmark, ever since, a large number of pre-training models similar to BERT emerged, including the generalized autoregressive model XLNet (Yang et al., 2019). Similar to the GPT model, BERT also builds a basic model by stacking Transformer substructures, which is a bidirectional PLM based on Transformer (see Fig. 12).

First, pre-training the language model, and then use the fine-tuning mode to solve downstream tasks. One of its innovations is to use Masked Language Model (MLM) to perform deep two-way training, randomly select 15 percent of the words in the corpus, replace it with the MASK, and then make the model predict the word at the MASK. Finally, the model parameters are adjusted by the Maximum Likelihood Estimation(MLE). At the same time, Next Sentence Prediction (NSP) is used to capture the dependency between sentences. In other words, when pre-trained the language model, select sentences in two situations: one is to choose the sentences that are connected in real order in the corpus; the other is to select a sentence and spell it after another sentence randomly. Finally, make the relationship prediction between sentences. BERT has achieved significant effect improvement in 11 tasks in the NLP field. Beginning in 2018, researches on the PLM has been surging. PLM can make full use of large-scale unlabeled data to learn a standard language model and then use a small amount of labeled data for downstream tasks to fine-tune the model. Compared with the model that directly trains specific tasks, the model fine-tuned on the PLM has achieved significantly improved performance in various NLP tasks. The structure of the PTM is shown in Fig. 13.

Although BERT works well, its shortcomings are also pronounced. The random selection of 15 percent of the character mask ignores the possibility that there may be semantic associations between the masked characters, thus losing some context information. Moreover, there is no mask mark in the fine-tuning stage, which leads to inconsistent pre-training and fine-tuning settings. In the follow-up, there are many related kinds of research based on improving the BERT model. A large number of pre-training models have emerged, which can be roughly divided into improved models based on BERT, XLNet (Yang et al., 2019), and generative models represented by MASS (Song et al., 2019).

Before ELMo was proposed, the language models are all Auto-Regressive (AR) models. Its disadvantage is that it can only use the above or below information but cannot use them simultaneously. BERT adopts the idea of DAE, which can be more naturally integrated into the Bi-directional language model. The above and below of the predicted word can be observed simultaneously. However, due to the introduction of the MASK mark, the pre-training and fine-tuning stages are inconsistent. The starting point of XLNet is the ability to integrate the advantages of both the AR language model and the DAE language model. XLNet proposes the idea of using an AR language model to simultaneously encode bidirectional semantic information, which can overcome the lack of dependence and the inconsistency of training and fine-tuning stages in BERT. XLNet introduces two-way

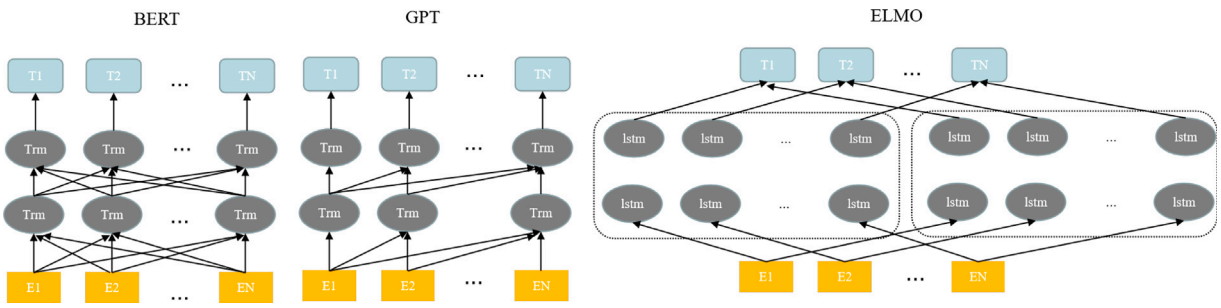

Fig. 12. Structure diagram of BERT, GPT and ELMO (Devlin et al., 2019).

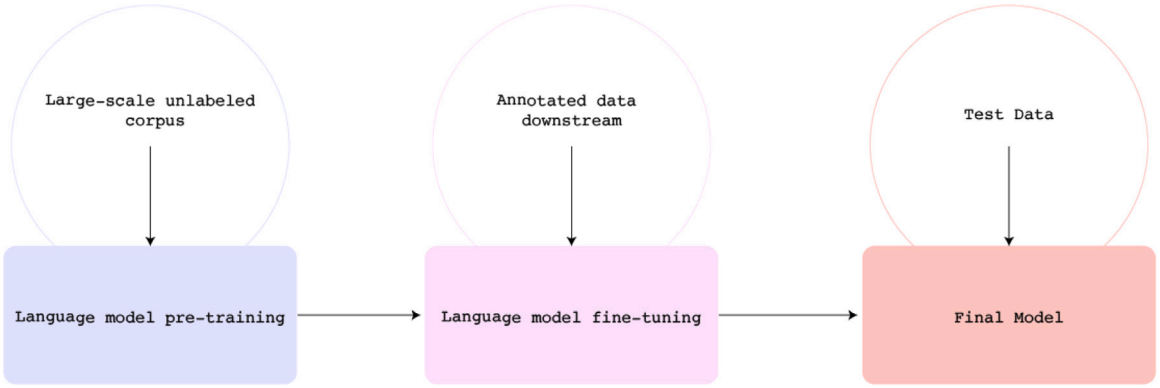

Fig. 13. The Training Process of PLM.

context information in BERT, RoBERTa (Liu, Ott, et al., 2019), and SpanBERT (Joshi et al., 2020) that improve BERT training methods and goals. The combination of multitasking and knowledge distillation enhances BERT MT-DNN (Liu, He, Chen, & Gao, 2019) and so on. The training method of the Permutation Language Model is also proposed to solve the defect that the context cannot be seen at the same time during Auto-Regressive training. The bidirectional self-attention mechanism and Transformer-XL are introduced to implement the model.

The improvement from BERT to RoBERTa and GPT to GPT2 in the early stage can prove that more data can run a more robust and general pre-trained model. PLM has made significant breakthroughs in parameter scale and performance with the dual support of GPU multi-machine multi-card parallel computing power and massive unlabeled text data, pushing the model scale and performance to new heights. The newly released GPT3 has reached a parameter scale of 175 billion, using tens of thousands of GPUs for training. Some PLM conditions in recent years are shown in Table 2. PTMs have not yet reached their upper limit. At present, most PTMs can be further improved through more training steps and a larger corpus. At the same time, they also require a deeper architecture, a larger corpus, and more complex and efficient training techniques (Qiu et al., 2020).

Recently, GShard (Lepikhin et al., 2021) with 1571B parameters was proposed, which is a deep learning module that partitions computation at scale automatically. Dmitry Lepikhin et al. applied GShard to scale up Transformer architecture with Sparsely-Gated Mixture-of-Experts layers (MoETransformer) and demonstrated a 600B parameter multilingual neural machine translation model could efficiently be trained in 4 days achieving superior performance and quality compared to prior art when translating 100 languages to English with a single model. Fedus, Zoph, and Shazeer (2021) simplify the MoE routing algorithm and design intuitive improved models with reduced communication and computational costs. The proposed training techniques help wrangle the instabilities, and large sparse models may be trained with lower precision formats for the first time.

Table 2
Basic situation of PLM development.

Proposal Time	Organization	Model Name	Model Scale	Data Size
2018.6	OpenAI	GPT	110M	4GB
2018.1	Google	BERT	330M	16GB
2019.2	OpenAI	GPT-2	1.5B	40GB
2019.7	Facebook	RoBERTa	330M	160GB
2019.1	Google	T5	11B	800GB
2020.6	OpenAI	GPT-3	175B	2TB
2020.6	Google	GShard	600B	4TB[a]
2021.1	Google	Switch Transformer	1571 B	750 GB

[a]1T tokens, about 4 TB.

4.5. Other typical deep learning algorithms

In the last decade, connectionist approaches have made the greatest achievements in image recognition. The CNN is mainly composed of a convolutional layer and a pooling layer. The convolutional layer can extract the local features of the image. And the pooling layer can reduce the dimensionality of a hidden layer in the middle, which is convenient for operation. The development process of the forward neural network is shown in Fig. 14. To better understand the development of image recognition algorithms, this paper traces its development history firstly. The development of CNN can be traced back to 1962, when Hubel and Wiesel studied the cat's brain's visual system and proposed a hierarchical model of the visual nervous system. In 1980, Fukushima and Miyake (1982) proposed a neural network with a visual pattern recognition mechanism, neocognitron, and proposed convolution and pooling. The model has self-organization capabilities, and the response of the network is hardly affected by the position of the stimulus model. In 1986, Rumelhart, Hinton, and Williams (1986) proposed the Back Propagation (BP) algorithm. The motivation of BP algorithm is to

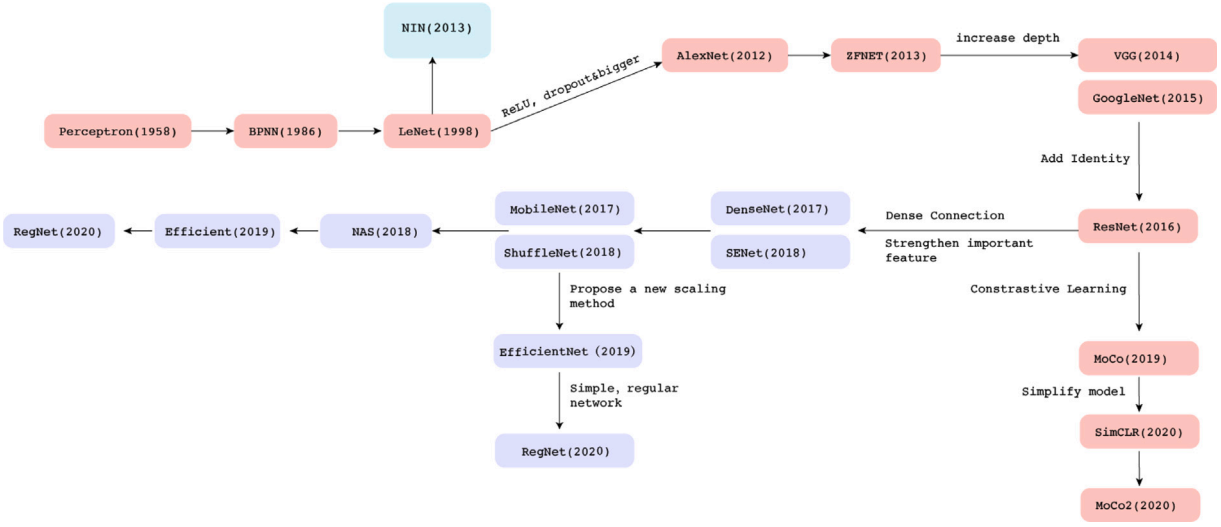

Fig. 14. Image recognition algorithms development history.

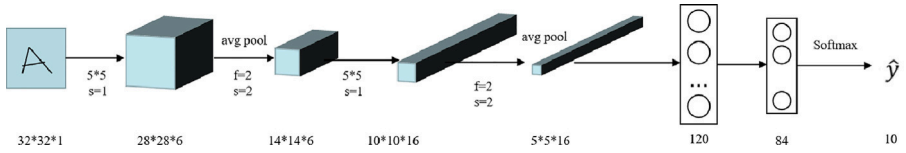

Fig. 15. Structure diagram of LeNet-5 model (LeCun et al., 1998).

obtain the best mapping function between the input and output by adaptively adjusting the connection weight between the neurons during the training process of the neural network. In this way, the objective function or loss function can be minimized, complete tasks such as classification and regression. At present, various network models in deep learning also use the BP algorithm proposed in 1986. On this basis, in 1989, Yann Lecun used the BP algorithm to train a multilayer neural network (LeCun et al., 1989), which was applied to recognize handwritten zip codes and proved the ability of a backpropagation network to process a large amount of low-level information. Then in 1998, Yann Lecun proposed a seven-layer CNN (LeCun et al., 1998) to solve the visual task of handwritten digit recognition and determined the basics of CNN architecture: the convolutional layer, the pooling layer, and the fully functional connected layer, which marks the actual emergence of CNN. It is used for handwritten digit recognition, and its structure diagram is shown in Fig. 15. Today, LeNet used in various deep learning frameworks is a simplified and improved LeNet-5. However, due to the difficulty of CNN training at that time, and other algorithms (SVM, Boosting algorithm) can achieve similar effects or even surpass CNN, CNN failed to become famous for some time. Nevertheless, at that time, there was no GPU to help with training, and even the speed of the CPU was plodding. Therefore, the ability to save parameters and the calculation process was a critical development.

Furthermore, what made CNN famous is the 2012 ImageNet competition, the AlexNet network (Krizhevsky et al., 2012), which introduced a new deep structure and dropout method to improve the recognition error rate and concluded that the depth of the model is essential to improve the performance of the algorithm. The reduction from twenty-five percent to fifteen percent has overturned the field of image recognition. The AlexNet network consists of 5 convoluted layers, max-pooling layers, dropout layers, and three fully connected layers. The output layer has 1000 neurons. After the Softmax function, the probability of 1000 categories can be obtained. AlexNet has many innovations, which introduces dropout to prevent over-fitting, uses translation, flipping, and intercepting part of the picture to increase training data, and uses ReLU instead of Sigmoid as the activation function to accelerate the convergence of SGD. Due to the limited amount of memory available to the GPU, AlexNet used two GPUs to train in parallel for six days. If there are a faster GPU and a larger data set, the experimental results will improve. AlexNet is the first deep neural network, and the structure diagram is shown in Fig. 16.

Following the idea of AlexNet, Zeiler and Fergus (Zeiler & Fergus, 2014) used deconvolutional networks to visualize CNN to understand the role of each layer of CNN, and the proposed ZF-Net, which became the champion of ImageNet classification tasks in 2013. A visualization technique is introduced to understand what the hidden layer does and how to classify it. The author finds that the small-scale filter features are better than the large-scale filter based on this technique. The step size of the first layer is too large, resulting in an aliasing phenomenon. The learned characteristics are not very good. ZF-Net is a fine-tuning of AlexNet to keep the number of network layers at eight. The model uses less training data to train and changes the first layer convolution kernel of AlexNet from 11 to 7. The adjusted network performance is better than AlexNet on many issues. On the other hand, Yan Shuicheng of NUS (Lin et al., 2014) proposed an important Network in Network (NIN) method, which is improved based on the traditional CNN method. To improve the nonlinearity, they added a $1 * 1$ ("1*1" means a two dimensional matrix, and the length of each dimension is 1) convolutional layer, which performs more complex operations on neurons with each local receptive field. They then proposed a global mean pooling method to replace the fully connected layer in traditional CNN, significantly reducing the network and avoiding overfitting. The brand-new model design proposed by NIN has triggered bold innovations in the structural changes of CNN.

The Visual Geometry Group (VGG) of Oxford University added a convolutional layer to AlexNet to study the effect of network depth. The results showed that the deeper the neural network, the better the effect is. In this way, they proposed VGGNets (Simonyan & Zisserman, 2015). By repeatedly stacking $3 * 3$ small convolution kernels and $2 * 2$ maximum pooling layers, VGGNet successfully constructed a 16–19 layer deep CNN. Compared with the ZFNet $7 * 7$ convolution kernel, the size of the VGGNet convolution kernel is only $3 * 3$, which makes the

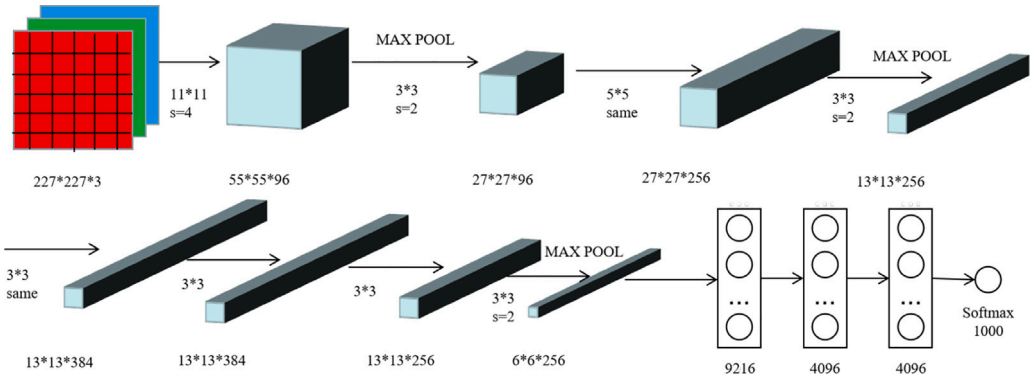

Fig. 16. Structure diagram of AlexNet model (Krizhevsky et al., 2012).

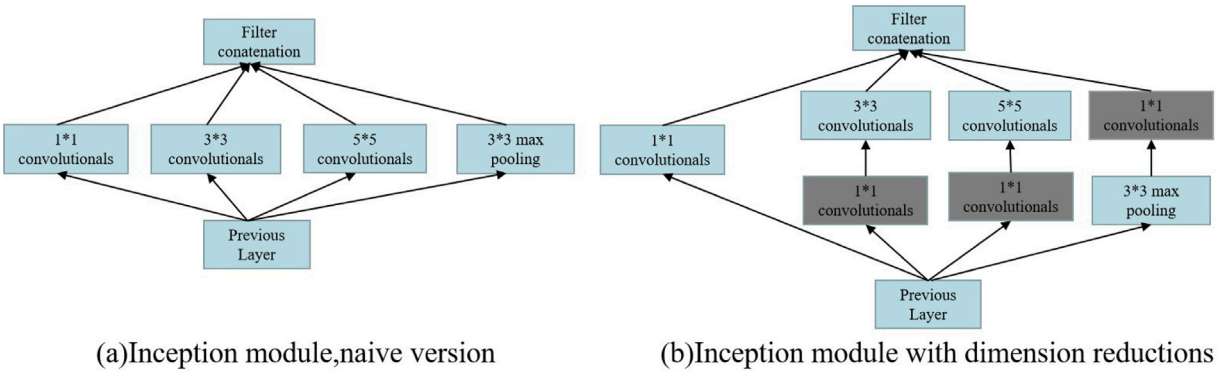

Fig. 17. Inception module (Szegedy et al., 2015).

parameters of the model is lesser than ZFNet. Moreover, two successive convolutional layers give it the effect of a $7 * 7$ convolution kernel. The small convolutional layer has more non-linear transformations than the large one, making CNN more vital to learn features. VGG-Nets came in second in the 2014 ImageNet Large-scale Visual Recognition Challenge (ILSVRC-2014), while GoogleNet (Szegedy et al., 2015) came in first in the same year. Before this, AlexNet, VGG, and other structures have achieved better training effects by increasing the depth of the network, but the increase in the number of layers would bring many side effects, such as overfitting, gradient disappearance, gradient explosion, etc. The inception module of the network structure introduced by GoogleNet improves the training results from another perspective: it can use computing resources more efficiently. Its network layers have 22 layers, and more features can be extracted under the same amount of calculation, which obtained outstanding performance. The inception module is a parallel structure of multiple convolution kernels and pooling, which is connected in series by merging layers to select the network automatically. As shown on the left in Fig. 17, to reduce the number of overall parameters, the team also borrowed the idea of NIN and added 1*1 convolution to the inception block, which effectively reduced the dimension of features and avoided parameter explosion. As shown on the right in Fig. 17, GoogleNet is also known as Inception V1, followed by improved versions called V2, V3, and V4.

In 2015, Shaoqing Ren et al. of MSRA proposed a new network structure named Deep Residual Network (ResNet) (He et al., 2016). ResNet solves the deep level that cannot be trained and dramatically improves the model degradation in Deep Neutral Networks (DNN) by reformulating the layers as learning residual functions with reference to the layer inputs. They tried to add identity to the CNN, deepen the CNN to 152 or even 1000 layers, significantly improve the accuracy of the model, and become the ILSVRC 2015 classification competition champion that year. The image error rate of the classification task was 3.57 percent, which exceeded that of humans by 5.1 percent. The design ideas of residual blocks and shortcuts have also profoundly affected the development of DNN structures in the future. Since the proposal of ResNet, the variant networks of ResNet have emerged in an endless stream, each with its own characteristics. ResNeXt (Xie, Girshick, Dollár, Tu, & He, 2017), Residual-Attention (Wang et al., 2017), SENet (Hu et al., 2018), and others made their contributions, introducing Group Convolution, Attention, channel wise-attention, etc., which ultimately reduced the error rate of ImageNet to 2.2 percent, significantly surpassing the human error rate. Another problem arises as the depth becomes deeper and deeper: some feature gradients may be lost after passing through many layers. Several papers address this problem by creating the shortest possible connection path between the input and output. Based on this, three authors from Conway University, Tsinghua University, and Facebook (Huang et al., 2017) jointly proposed the DenseNet network structure and won the CVPR 2017 Best Paper Award. The input of each layer of the network is the union of all the previous output of layers. The feature map learned by this layer will also be directly passed to all subsequent layers as input, which alleviates the problem of gradient disappearance and strengthens the propagation of features, encourages feature reuse, and reduces the number of parameters. Dense is another excellent network structure after ResNet, which draws on the idea of the shortcut connection in ResNet and refers to the design of the Inception module. Dense is a brand new structure that further improves network performance.

In another challenge of image detection, Shaoqing Ren, Yuming He, Jian Sun, and others have successively proposed Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren, He, Girshick, & Sun, 2015) methods, which significantly improved the speed of model training. Faster R-CNN uses the Region Proposal Network (RPN) for real-time target detection, and the Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017) proposed later adds a semantic segmentation task based on Faster R-CNN. These two innovations directly subverted the entire field of computer vision.

As the number of neural network layers is increasing, its performance is getting better. The efficiency problem comes along, including

the storage problem of the model and the speed of model prediction. Therefore, more attention has been paid to lightweight model design, whose main idea is to design a more efficient network structure to reduce network parameters without loss of network performance. Based on this, Google proposed MobileNet (Howard et al., 2017), and Face++ proposed ShuffleNet (Zhang et al., 2018). The core idea of MobileNet is to use depthwise separable convolution to build a lightweight DNN. In the case of the same input and output dimensions, it can reduce the amount of calculation and parameters several times compared with the standard convolution operation. ShuffleNet mainly uses pointwise group convolution to help reduce computational complexity. On this basis, channel shuffle is also proposed to help information flow. Compared with the existing advanced models, ShuffleNet greatly reduces the amount of calculation under similar accuracy, and ShuffleNet shows superior performance than other advanced models on ImageNet and MS COCO. In practical applications, the development of CNN has solved the problems in the traditional computer vision field and has also triggered changes in other fields. AlphaGo (Silver et al., 2016) uses CNN to judge the situation of the game of Go and successfully defeated Shishi Li. Abdel-Hamid et al. (2014) combined the CNN with a hidden Markov model and applied it to the field of speech recognition; Grefenstette et al. (2014) also successfully applied CNN to NLP. The CNN is brilliant in various fields, and its design ideas are developing towards deeper, wider, more branches, lighter, and more effective convolution methods.

Just as an electric motor gradually replaces the steam engine, the neural network structure and design transform from manual design to automatic design. In 2016, Google using RL for neural network structure search (Zoph & Le, 2017), surpassing the previous hand-designed network in image classification and language modeling tasks. Although they perform well in some tasks, they still cannot do without the cumbersome hyperparameter selection. In 2019, Mingxing Tan et al. proposed EfficientNet (Tan & Le, 2019), a more generalized idea is proposed for optimizing the current classification network. They believe that the commonly used methods of widening the network, deepening the network, and increasing the resolution should not be independent of each other. Thus, they proposed a compound model scaling algorithm. By comprehensively optimizing the network width, network depth, and resolution, index improvement can be achieved. The number of model parameters and calculations can be significantly reduced when the accuracy index is similar to the existing classification network. Subsequently, Ilija Radosavovic et al. proposed a new network design paradigm called RegNet (Radosavovic et al., 2020). Unlike previous studies, they did not focus on designing a single network instance but designed a network design space for parameterized network groups. This new network design paradigm combines the advantages of hand-designed networks and NAS, which surpasses EfficientNet and achieves five times acceleration on the GPU.

In NLP, unsupervised learning has achieved great success, but the CV field is still dominated by supervised learning. The main reason for this gap is that CV and NLP have different signal representation spaces. The signal space in the CV field is continuous and high-dimensional, which is not conducive to the construction of the dictionary. The MoCo (He et al., 2020) method proposed by He Kaiming et al. uses contrast loss to construct a large-capacity and coordinated dictionary to deal with unsupervised learning problems and represents the dictionary as a data sampling queue. The encoding of the current mini-batch indicates that the feature is enqueued, and the encoding of the old mini-batch indicates that the feature is dequeued. The use of the queue makes the dictionary size independent of the mini-batch size, so the dictionary can have a large capacity. At the same time, the critical value of the dictionary comes from the fusion of several previous mini-batches, which is calculated using the momentum-based moving average of the query set encoding features, ensuring the continuity of the dictionary. Dictionary is a significant development, which marks a new level of pre-training. Later, Hinton et al. proposed the SimCLR method on this basis (Chen, Kornblith, et al., 2020). SimCLR is a simple visual representation contrast learning framework that is better than previous work and more straightforward. SimCLR uses a combination of multiple data enhancement methods to ensure that a significant feature representation is generated. A learnable nonlinear transformation is introduced between feature representation and contrast loss, improving the quality of representation. First, learn the general representation of the image on the unlabeled data set and then fine-tune it with a small number of labeled images to perform well for a given classification task. Due to the impact of SimCLR, the author added some of these methods to MoCo, replaced the last fully connected layer of MoCo with MLP, added blur augmentation to the image enhancement method, and proposed MoCo V2 (Chen, Fan, Girshick, & He, 2020), which improved the effect of MoCo.

5. Limitations and future trend analysis

Although AI has developed rapidly in recent years, some aspects still need to be improved for the current stage of AI, especially connectionist approaches. According to the above study, combined with some mainstream opinions of the current society, AI algorithms represented by deep learning and RL still have several significant problems: big data dependence, lack of interpretability, and vulnerability to attack. Therefore, when faced with a dynamically changing environment, incomplete information, interference, and false information, the performance of the AI system will significantly decrease. Researches on AI algorithms will be carried out in four aspects: Understandable, Usable, Universal, and Ubiquitous. In the following part, we will introduce the limitations of AI development at this stage and some of the main trends of future development.

5.1. Strong data dependence & its future trend

As we all know, deep learning has achieved great success in many areas of AI (Devlin et al., 2019; He et al., 2016; Silver et al., 2016). Three major driving factors have contributed to the success of deep learning: data availability, algorithm design, and computing powers. Its success is mainly in the areas where a large amount of data can be collected or simulated, and a large number of computing resources can be used, which has substantial limitations. In the current stage of deep learning, data and computing power limit the development of mainstream deep learning algorithms. Unlike traditional machine learning, deep learning can use massive amounts of data to improve its performance. The continuous improvement of recognition accuracy in AI competitions such as ImageNet is behind the fact that countless human and material resources have been invested in high-quality labeled data. As many scholars quipped, “more artificial makes more intelligent”.

What is more, mainstream deep learning methods that rely excessively on massive amounts of labeled data face more and more challenges. Companies such as Google and Facebook are using big data, their systems have become better and better, and the error rate has become lower and lower. For example, the NLP system of Google has obtained almost all network and natural language data in the world for training. However, only these giants have the opportunity to obtain this big data in some specific fields. For fields such as law, finance, and medical care where labeled data cannot be collected, or the collection is costly, the establishment of AI applications becomes difficult. Similarly, in intelligent medicine, the threshold for high-quality medical image labeled data is very high.

Many machine learning methods only work well under the same assumptions. Training data and test data are extracted from the same feature space and the same distribution. When the distribution changes, most statistical models need to be built from scratch using re-collected data (Pan & Yang, 2009). Because of this, new research fields such as transfer learning, few-shot learning, and meta-learning have gradually developed. If knowledge transfer is successful, learning performance

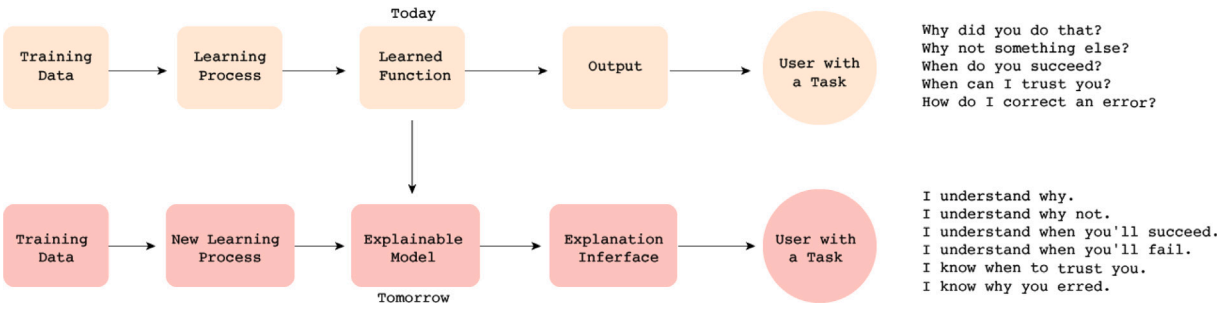

Fig. 18. AI of the future.

will be significantly improved by avoiding expensive data labeling work. In other words, a good general model has been trained in a big data environment, and then the trained model is migrated to a related task. If it is feasible in the new scenario, there is no need to reacquire the big data. This is an advantageous way to solve big data defects.

5.2. Poor interpretability & its solutions

The success of machine learning has led to the explosion of AI applications, and researchers have developed new AI capabilities for various tasks. Continued progress is expected to produce autonomous systems that will perceive, learn, decide and act by themselves. The complexity of AI-driven systems has recently increased to a point where almost no human intervention is required to design and deploy. For example, in the past few years, there have been opaque decision-making systems such as DNNs. Its success comes from the combination of efficient learning algorithms and their huge parameter space. It contains hundreds of layers and millions of parameters, making DNNs considered as complex black-box models (Castelvecchi, 2016). Given an input, the model will make a judgment or prediction. However, it is very hard for connectionist approaches to tell us why it made such a judgment or prediction and whether we should trust its judgment.

As black-box machine learning models are increasingly used to make essential predictions in critical environments, the need for transparency among various stakeholders in AI is increasing. The danger lies in making decisions that are unreasonable, illegal, or do not allow detailed explanations of their actions. For example, humans and machines need to cooperate in many risk-sensitive application fields, such as medical, military, and finance. The critical prerequisite for human–machine collaboration is that humans and machines can understand each other. The output of AI technology is becoming more and more explainable, for example uplift models make them more suited for human comprehension (Devriendt, Moldovan, & Verbeke, 2018; Gutierrez & Gérardy, 2017). However, it is still not enough for some fields, which requires the model to explain its thoughts and actions to human users.

Nevertheless, even humans sometimes have difficulty explaining their thoughts and actions to other humans, albeit they do a far better job than current AI can. The way to solve these problems is to create a set of machine learning techniques to generate more interpretable models while maintaining a high level of learning performance. That is, to conduct related research on the interpretability of AI. Miller defines "interpretability" as "the degree to which people can understand the reasons for a decision" (Miller, 2019). The higher the interpretability of the machine learning model, the easier it is for people to understand why a specific decision or prediction is made. DARPA has initiated the XAI project in 2017. The project comprehensively researches interpretable AI systems from three aspects: interpretable machine learning systems and models, human–computer interaction technology, and interpretable psychology (Gunning et al., 2019). The function description of interpretable AI is shown in Fig. 18. New machine learning systems can explain their basic principles, characterize their strengths and weaknesses, and convey an understanding of how they will perform in the future. Although research on interpretable machine learning has been increasing recently, a practical and widely used interpretable AI technology has not yet emerged. This challenge is daunting.

5.3. Vulnerable to attack and its dilemma

With the substantial increase in computer performance and processing capabilities brought about by the computer industry, AI has been widely used in audio and video recognition, NLP, and game theory. In this context, it is crucial to ensure that the core of AI, the deep learning algorithm, is safe and robust. However, recent studies have found that deep learning models are vulnerable to security risks from adversarial sample attacks. A slight disturbance can make deep learning models make wrong judgments. There is a "blind spot" in every DNN. Some inputs that the human eye cannot detect can completely fool the deep learning model. Small disturbances can be carefully designed to formulate adversarial examples, thereby forcing the network to produce false predictions with high confidence (Wang, Li, Kuang, Tan, & Li, 2019). Therefore, counter-attack technology has gradually been applied to academia and industry. In computer vision, there are confrontation attacks in image classification. In NLP, there are confrontation attacks in machine translation and text generation. In cyberspace security, there are confrontation attacks in cloud services, malware detection, and network intrusion detection. In the physical world, there are also confrontational attacks such as road sign recognition and deception of cameras. It is mainly divided into attacks in the training phase and attacks in the testing phase. In the training phase, the poisoning attack is a typical attack method, trying to modify the statistical characteristics of the training data set. The testing phase is a process of generating inferences based on the training model. Most attack methods are developed based on DNN. The main attack methods in the testing phase include the L-BFGS method, Fast Gradient Sign Method (FGSM), Universal Adversarial Perturbations (UAP), etc. Researchers have made some explanations for the existence of adversarial samples. Some people think that Taga, Kameyama, and Toraichi (2003) over-fitting or under-regularization leads to insufficient generalization ability of the model in the face of unknown data. Moreover, other people think that Papernot et al. (2017) it is the extreme nonlinearity of the DNN.

Nevertheless, the root cause of this incident is still uncertain. In order to improve the robustness of neural network adversarial attacks, researchers have proposed a large number of adversarial defense methods, which can be generally divided into data modification, model modification, and the use of auxiliary tools. The methods of modifying data include adversarial training, gradient hiding (Tramèr et al., 2018), data compression, etc. mainly. However, it is unrealistic to introduce adversarial training for all unknown attacks. A large amount of compression will also cause a decrease in the accuracy of the picture, and a small compression will not be enough to eliminate the influence of interference (Qiu, Liu, Zhou, & Wu, 2019). Modifying the model refers to modifying the target neural network, but there may be a problem of effectively resisting adversarial attacks and reducing the accuracy of real sample classification. Using an auxiliary refers to the

use of additional tools as auxiliary tools for neural network models, such as defense-GAN (Samangouei, Kabkab, & Chellappa, 2018), MagNet (Meng & Chen, 2017), HGD (Liao et al., 2018), etc. Nevertheless, for defense-GAN, the training of GAN is challenging. That is, without proper training, it is difficult for defense-GAN to achieve good performance.

Although there has been much research work related to adversarial defense in recent years, there is no practical and universal adversarial defense technology framework and method in actual security applications. It seems like that the situation remains sobering. The current adversarial training defense technology is not mature enough and the defense methods have certain flaws. To achieve practical defense goals, this urgently requires a theoretical breakthrough in deep learning and also requires a combination of system framework, security testing, environmental adaptation, and other security technologies to promote the leapfrog development of deep learning adversarial security.

6. Discussion

The second generation of AI is connectionism approaches, represented by deep learning. It is generally known that deep learning is an algorithm tool in the era of big data. Compared with traditional machine learning algorithms, deep learning technology can continuously improve performance as data increases, and features can be extracted directly from the data, eliminating the work of designing feature extractors for each problem. Feifei Li began to study the idea of ImageNet since 2006. While most AI research focuses on models and algorithms, Feifei Li hopes to expand and improve the data that can be used to train AI algorithms with the development of computing power and big data. Since then, a new way of thinking has been derived: the focus on computer vision has shifted from model to data. There is a growing trend to gain more intelligence through bigger models and more data. Despite the fact that these big models have made tremendous development, the second generation of AI also has significant limitations, such as inexplicability, insecurity, vulnerability, difficulty to promote, and requires a large number of samples. Moreover, as Guojie Li considered, AI faces three paradoxes: Moravec's Paradox (Agrawal, 2010), New Knowledge Paradox (Li, 2017), and Heuristic Paradox (Minsky, 1956). These paradoxes are closely related to the limitations of the second generation of AI. Reflections on computing and intelligence may provide more ideas for the development of AI.

Although AI is becoming more and more intelligent nowadays, quite a few scientists insist that the artificial brain is the way to AGI (Artificial General Intelligence, also called Strong AI), such as Tiejun Huang from Peking University. Unlike traditional and machine-learning approaches, which attempt to copy brain function, neuromorphic computing mimics the structure of the brain. By building an electronic brain with neuromorphic devices, brain-like computing asserts that parsing the physiological structure of the brain (neurons, synapses, neural circuits, and the functional regions of the cortex) is more straightforward than unveiling the principles of intelligence and that a similar structure could generate similar functions. This field could fundamentally change current computing models and even our understanding of intelligence, whether brain-like, brain-inspired, or brain-mimicking.[11] Brain-like computing is likely to be used primarily for the AI tasks they are good at, rather than replacing the traditional computers of the von Neumann architecture. The complementarity and integration of the two will be the possible trend of AI in the future.

So far, the development of AI has just kicked off, and its application scope is very limited. The next generation of AI must solve the above shortcomings, so we must establish an explainable and robust AI theory and develop safe, credible, reliable, and scalable AI technology. Only in this way can a technological breakthrough be achieved. According to the viewpoint of Bo Zhang et al. the third generation AI is a combination of the knowledge-driven approach and the data-driven approach, which makes the most of the four elements (knowledge, data, algorithm, and computing power) (Zhang et al., 2020). Regardless of predictions about the AI future, we are at the critical point of moving towards AGI. AI in the next decade can be the hybrid application of Narrow AI[12] and the evolving age of AGI, thus enabling the AI to think more effectively than a human.

7. Conclusion

The current stage is the critical period from the second generation of AI to the third generation and from Narrow AI to AGI. Even though the way to realize the third generation of AI or AGI needs further exploration, the AI-enabled future will be "less artificial, more intelligent". It is worth studying AI development in the past decade, primarily the significant breakthrough in connectionist approaches, enlightening for the future. Toward this aim, this review introduces the development over the last decade with the help of the proposed approach and discusses the limitations and future trends of AI in the next decade from several perspectives. In detail, according to the mining analysis results of AMiner, we study the development of AI from five aspects, almost covering all the top hot spots in the past decade. These five critical parts include (1) self-learning and self-coding algorithms, (2) RNN algorithms, (3) reinforcement learning, (4) pre-trained models, and (5) other typical deep learning algorithms, which are selected by the proposed method. In the investigation, this paper proposes a particular perspective to examine the evolution of AI from the development of connectionist approaches, which has significant implications for understanding significant evolutions of AI in the past decade. During the last decade, the breakthrough of connectionist approaches significantly propelled its progress in study and application, which has been systematically teased apart in this paper. However, lack of interpretability, big data dependence, and vulnerability to attack are still the most lethal problem for current AI, hindering the development of AI and inhibiting its utilization. Improvements in these three aspects will be of great significance to the development of AI. This review complements other reviews for throwing light on the law and trend of the most prominent breakthrough of the past decade. Therefore, the proposed joint approach should be of value to practitioners wishing to study the history and trends of a discipline. The findings presented in this paper add to our understanding of AI, instructive for national AI strategies. However, more research is needed to develop a deeper understanding of the relationships between the current progress and future AI, which will be the subsequent focus of future studies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This article has been awarded by the National Natural Science Foundation of China (61941113, 61806111) and Science and Technology on Information System Engineering Laboratory (05202104).

[11] This view comes from Tiejun Huang's talk in 2022 Report of the IEEE Computer Society.

[12] Narrow AI is a term used to describe AI systems that are specified to handle a singular or limited task.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR abs/1603.04467, arXiv:1603.04467.

Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(10), 1533–1545.

Agrawal, K. (2010). To study the phenomenon of the Moravec's paradox. CoRR abs/1012.3148, arXiv:1012.3148.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. CoRR abs/1701.07875, arXiv:1701.07875.

Ávila-Tomás, J. F., Mayer-Pujadas, M. A., & Quesada-Varela, V. J. (2020). Artificial intelligence and its applications in medicine II: Current importance and practical applications. *Atencion Primaria*, *53*(1), 81–88.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Computer Science*.

Baylor, G. W., & Simon, H. A. (1966). A chess mating combinations program. In *Proceedings of the April 26-28, 1966, Spring joint computer conference* (pp. 431–447).

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*(Feb), 1137–1155.

Bengio, Y., Yao, L., Alain, G., & Vincent, P. (2013). Generalized denoising auto-encoders as generative models. In *Advances in neural information processing systems* (pp. 899–907).

Berliner, H. J. (1980). Backgammon computer program beats world champion. *Artificial Intelligence*, *14*(2), 205–220.

Berners-Lee, T., et al. (1998). Semantic web road map.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, *59*(4–5), 291–294.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, *2*(1), 14–23.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. CoRR abs/2005.14165, arXiv:2005.14165.

Campbell, M., Hoane Jr, A. J., & Hsu, F.-h. (2002). Deep blue. *Artificial Intelligence*, *134*(1–2), 57–83.

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature News*, *538*(7623), 20.

Chen, X., Fan, H., Girshick, R. B., & He, K. (2020). Improved baselines with momentum contrastive learning. CoRR abs/2003.04297, arXiv:2003.04297.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607). PMLR.

Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Computer Science*.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8789–8797).

Choubey, S., & Karmakar, G. (2020). Artificial intelligence techniques and their application in oil and gas industry. *Artificial Intelligence Review*, 1–19.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Cosma, G., & Acampora, G. (2016). A computational intelligence approach to efficiently predicting review ratings in e-commerce. *Applied Soft Computing*, *44*, 153–162.

Davis, R. (1976). Applications of meta level knowledge to the construction, maintenance and use of large knowledge bases. Artificial Intelligence Laboratory, Stanford University, Stanford, CA.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 Conference of the north American chapter of the association for computational linguistics: human language technologies, vol. 1 (long and short papers)* (pp. 4171–4186). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/n19-1423.

Devriendt, F., Moldovan, D., & Verbeke, W. (2018). A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. *Big Data*, *6*(1), 13–41.

Dong, G., Liao, G., Liu, H., & Kuang, G. (2018). A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images. *IEEE Geoscience and Remote Sensing Magazine*, *6*(3), 44–68.

Fedus, W., Zoph, B., & Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. CoRR abs/2101.03961, arXiv:2101.03961.

Feigenbaum, E. A., & McCorduck, P. (1983). *The fifth generation*. Addison-Wesley Pub.

Fischer, A., & Igel, C. (2012). An introduction to restricted Boltzmann machines. In *Iberoamerican congress on pattern recognition* (pp. 14–36). Springer.

Fortunato, S., Bergstrom, C. T., Börner, K., Evans, J. A., Helbing, D., Milojević, S., et al. (2018). Science of science. *Science*, *359*(6379), eaao0185.

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (pp. 1587–1596). PMLR.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267–285). Springer.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Grefenstette, E., Blunsom, P., et al. (2014). A convolutional neural network for modelling sentences. In *The 52nd annual meeting of the association for computational linguistics*.

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G. (2019). XAI - Explainable artificial intelligence. *Science Robotics*, *4*(37), http://dx.doi.org/10.1126/scirobotics.aay7120.

Gutierrez, P., & Gérardy, J.-Y. (2017). Causal inference and uplift modelling: A review of the literature. In *International conference on predictive applications and APIs* (pp. 1–13). PMLR.

Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, *61*(4), 5–14.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., et al. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *5th international conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, conference track proceedings*. OpenReview.net, URL: https://openreview.net/forum?id=Sy2fzU9gl.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861, arXiv:1704.04861.

Hsu, F.-h., Campbell, M. S., & Hoane Jr, A. J. (1995). Deep blue system overview. In *Proceedings of the 9th international conference on supercomputing* (pp. 240–244).

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, *8*, 64–77.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, conference track proceedings*. OpenReview.net, URL: https://openreview.net/forum?id=Hk99zCeAb.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4401–4410).

Kim, H., & Mnih, A. (2018). Disentangling by factorising. In *International conference on machine learning* (pp. 2649–2658). PMLR.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *ICLR*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the 18th intl. conf. on machine learning*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lenat, D. B. (1976). *AM: An artificial intelligence approach to discovery in mathematics as heuristic search: Technical report*, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., et al. (2021). Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th international conference on learning representations*. OpenReview.net, URL: https://openreview.net/forum?id=qrwe7XHTmYb.

Li, G. (2017). Three paradoxes of artificial intelligence(in chinese). *CCCF, 11*(13), 1.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1778–1787).

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2016). Continuous control with deep reinforcement learning. In Y. Bengio, Y. LeCun (Eds.), *4th international conference on learning representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, conference track proceedings*.

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. In *ICLR*.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., & Lederberg, J. (1993). DENDRAL: a case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, *61*(2), 209–261.

Liu, X., He, P., Chen, W., & Gao, J. (2019). Multi-task deep neural networks for natural language understanding. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, vol. 1: long papers* ACL 2019, Florence, Italy, July 28- August 2, 2019, (pp. 4487–4496). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/p19-1441.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692, arXiv:1907.11692.

López-Robles, J. R., Otegi-Olaso, J. R., Gómez, I. P., & Cobo, M. J. (2019). 30 Years of intelligence models in management and business: A bibliometric review. *International Journal of Information Management*, *48*, 22–38.

López-Robles, J. R., Otegi-Olaso, J. R., Gómez, I. P., Gamboa-Rosales, N. K., Gamboa-Rosales, H., & Robles-Berumen, H. (2018). Bibliometric network analysis to identify the intellectual structure and evolution of the big data research field. In *International conference on intelligent data engineering and automated learning* (pp. 113–120). Springer.

López-Robles, J. R., Rodríguez-Salvador, M., Gamboa-Rosales, N. K., Ramirez-Rosales, S., & Cobo, M. J. (2019). The last five years of big data research in economics, econometrics and finance: Identification and conceptual analysis. *Procedia Computer Science*, *162*, 729–736.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2794–2802).

Marr, D. (1976). Early processing of visual information. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *275*(942), 483–519.

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information, henry holt and co Inc., New York, NY. 2, 4.2.

Martin, W. A., & Fateman, R. J. (1971). The MACSYMA system. In *Proceedings of the second ACM symposium on symbolic and algebraic manipulation* (pp. 59–75). New York, New York, USA: ACM Press, ACM.

Marvin, M., & Seymour, P. (1969). *Perceptrons : An introduction to computational geometry*. The MIT Press.

Mathieu, M., Couprie, C., & Lecun, Y. (2016). Deep multi-scale video prediction beyond mean square error. In *ICLR*.

McDermott, D., & Doyle, J. (1980). Non-monotonic logic I. *Artificial Intelligence*, *13*(1–2), 41–72.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282). PMLR.

Meng, D., & Chen, H. (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 135–147).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the international conference on learning representations*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38.

Minsky, M. (1956). *Heuristic aspects of the artificial intelligence problem*. Ed. Services Technical Information agency.

Minsky, M. (1974). *A framework for representing knowledge*. MIT-AI Laboratory Memo.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *Computer Science*, 2672–2680.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928–1937).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing atari with deep reinforcement learning. *Computer Science*.

Moynihan, G. P. (1993). Application of expert systems to engineering design. In *Concurrent engineering: contemporary issues and modern design tools* (pp. 375–385). Boston, MA: Springer US, ISBN: 978-1-4615-3062-6.

van den Oord, A., Vinyals, O., & Kavukcuoglu, K. (2017). Neural discrete representation learning. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30: annual conference on neural information processing systems* (pp. 6306–6315). URL: https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security* (pp. 506–519).

Pearl, J. (1985). BayesIan netwcrks: A model cf self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the cognitive science society* (pp. 15–17). Irvine, CA, USA: University of California.

Pearl, J. (1990). Probabilistic reasoning in intelligent systems: Networks of plausible inference (Judea Pearl). *Artificial Intelligence*, *48*(8), 117–124.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*.

Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, *9*(5), 909.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. CoRR abs/2003.08271, arXiv:2003.08271.

Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Y. Bengio, & Y. LeCun (Eds.), *4th international conference on learning representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, conference track proceedings*.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollár, P. (2020). Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10428–10436).

Ranzato, M., Susskind, J., Mnih, V., & Hinton, G. (2011). On deep generative models with applications to recognition. In *CVPR 2011* (pp. 2857–2864). IEEE.

Razavi, A., van den Oord, A., & Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems* (pp. 14866–14876).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In L. Getoor, & T. Scheffer (Eds.), *Proceedings of the 28th international conference on machine learning* (pp. 833–840). Omni Press, URL: https://icml.cc/2011/papers/455_icmlpaper.pdf.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Russell, S., & Norvig, P. (1995). Artificial intelligence: A modern approach, 3rd edition. *Applied Mechanics & Materials*, *263*(5), 2829–2833.

Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, conference track proceedings*. OpenReview.net, URL: https://openreview.net/forum?id=BkJ3ibb0-.

Santana, E., & Hotz, G. (2016). Learning a driving simulator. CoRR abs/1608.01230, arXiv:1608.01230.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. CoRR abs/1707.06347, arXiv:1707.06347.

Shaham, T. R., Dekel, T., & Michaeli, T. (2019). Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE international conference on computer vision* (pp. 4570–4580).

Shannon, C. E. (1950). XXII. Programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *41*(314), 256–275.

Shao, Z., Shen, Z., Yuan, S., Tang, J., Wang, Y., Wu, L., et al. (2020). AI 2000: A decade of artificial intelligence. In *12th ACM conference on web science* (pp. 345–354).

Shao, Z., Yuan, S., & Wang, Y. (2020). Institutional collaboration and competition in artificial intelligence. *IEEE Access*, *8*, 69734–69741.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International conference on machine learning* (pp. 387–395). PMLR.

Simon, H. A. (1965). *The shape of automation for men and management, vol. 13* (p. 96). Harper & Row New York.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio, & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings*.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official Google Blog*, *5*, 16.

Smolensky, P. (1986). *Information processing in dynamical systems: Foundations of harmony theory*: *Technical report*, Colorado Univ at Boulder Dept of Computer Science.

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. (2019). MASS: masked sequence to sequence pre-training for language generation. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of machine learning research*: *vol. 97*, *Proceedings of the 36th international conference on machine learning* (pp. 5926–5936). PMLR, URL: http://proceedings.mlr.press/v97/song19d.html.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Taga, K., Kameyama, K., & Toraichi, K. (2003). Regularization of hidden layer unit response for neural networks. In *2003 IEEE Pacific rim conference on communications computers and signal processing (PACRIM 2003)(Cat. No. 03CH37490), vol. 1* (pp. 348–351). IEEE.

Tahiru, F. (2021). AI in education: A systematic literature review. *Journal of Cases on Information Technology (JCIT)*, *23*(1), 1–20.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 990–998). ACM.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I. J., Boneh, D., & McDaniel, P. D. (2018). Ensemble adversarial training: Attacks and defenses. In *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, conference track proceedings*. OpenReview.net, URL: https://openreview.net/forum?id=rkZvSe-RZ.

Turing, I. B. A. (1950). Computing machinery and intelligence-AM turing. *Mind*, *59*(236), 433.

Uhr, L., & Vossler, C. (1961). A pattern recognition program that generates, evaluates, and adjusts its own operators. In *Papers presented At the May 9-11, 1961, Western Joint IRE-AIEE-ACM computer conference* (pp. 555–569).

van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. In D. Schuurmans, & M. P. Wellman (Eds.), *Proceedings of the thirtieth AAAI conference on artificial intelligence* (pp. 2094–2100). AAAI Press, URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics. In *Advances in neural information processing systems* (pp. 613–621).

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Wang, X., Li, J., Kuang, X., Tan, Y.-a., & Li, J. (2019). The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing*, *130*, 12–23.

Wang, Z., & Xie, L. (1999). Artificial psychology: an attainable scientific research on the human brain. In *Proceedings of the second international conference on intelligent processing and manufacturing of materials. IPMM'99 (Cat. No. 99EX296), vol. 2* (pp. 1067–1072). IEEE.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3–4), 279–292.

Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5753–5763).

Yin, D., Tam, W. L., Ding, M., & Tang, J. (2021). MRT: Tracing the evolution of scientific publications. *IEEE Transactions on Knowledge and Data Engineering*.

Yuan, S., Shao, Z., Liang, Y., Tang, J., Hall, W., Liu, G., et al. (2020). International scientific collaboration in artificial intelligence an analysis based on web data. In *12th ACM conference on web science* (pp. 69–75).

Yuan, S., Shao, Z., Wei, X., Tang, J., Hall, W., Wang, Y., et al. (2020). Science behind AI: the evolution of trend, mobility, and collaboration. *Scientometrics*, *124*, 1–21.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6848–6856).

Zhang, B., Zhu, J., & Su, H. (2020). Towards the third generation of artificial intelligence(in Chinese). *Scientia Sinica Informationis*, *50*(09), 7–28.

Zhao, S., Song, J., & Ermon, S. (2017). InfoVAE: Information maximizing variational autoencoders. CoRR abs/1706.02262, arXiv:1706.02262.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).

Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. In *5th international conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, conference track proceedings*. OpenReview.net, URL: https://openreview.net/forum?id=r1Ue8Hcxg.