Computing, Performance and Communication Systems (2026)
Clausius Scientific Press, Canada
DOI: 10.23977/cpcs.2026.100102
ISSN 2371-8889 Vol. 10 Num. 1

A Review of the Evolution of Core Neural Network Models in Deep Learning

Suyang Wu

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China
1615079253@qq.com

Keywords: Deep Learning; Neural Networks; Model Evolution; Core Logic; Development Prospect

Abstract: As a core technical branch in the field of artificial intelligence, the development of deep learning is inseparable from the continuous iteration of neural network models. Starting from early neural network models, this paper systematically sorts out the birth background and key technical breakthroughs of core models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformers. It deeply analyzes the structural characteristics, application scenarios, and inherent limitations of each model. On this basis, it summarizes the core logic of the evolution of deep learning models, such as the transformation from local dependency to global dependency, and from sequential processing to parallel computing. Combined with the current technological development trend, it looks forward to the future development directions of neural network models, such as lightweight and modularization, providing a reference for research and application in related fields.

1. Introduction

Since the concept of artificial neural networks was proposed in the 1940s, deep learning technology has undergone decades of ups and downs and has become the core driving force for promoting artificial intelligence from theory to practical application. As the core carrier of deep learning, each breakthrough in the structural design and learning mechanism of neural network models has greatly expanded the application boundary of artificial intelligence, showing strong technical advantages from early handwritten character recognition to today's complex tasks such as natural language generation and autonomous driving.

The essence of deep learning is to realize the fitting of complex nonlinear relationships through hierarchical feature extraction and abstract modeling of data by multi-layer neural networks. Looking back at its development process, the evolution of neural network models has always centered on three core goals: improving feature expression capabilities, optimizing training efficiency, and expanding application scenarios. Early perceptrons and BP neural networks laid the basic framework of neural networks, solving the problem of from scratch; convolutional neural networks and recurrent neural networks achieved special breakthroughs targeting the characteristics

of spatial data and temporal data respectively; the self-attention mechanism introduced by the Transformer architecture broke the structural constraints of traditional models, realizing the leapfrog development of global dependency modeling and parallel computing.

This paper systematically sorts out the evolution context of core neural network models in deep learning, deeply analyzes the technical characteristics and application limitations of each key model, summarizes the internal logic of model evolution, and looks forward to future development directions. It aims to comprehensively present the overall development of core deep learning models, providing theoretical reference for researchers in related fields to grasp the laws of technological development and carry out innovative research.

2. The Foundational Role of Early Neural Network Models

2.1. Perceptron: The Prototype of Artificial Neural Networks

In the mid-20th century, the development of biological neuroscience provided an important inspiration for the birth of artificial neural networks. In 1943, McCulloch and Pitts proposed the first artificial neuron model (MP model)[1], which simulated the logical operation mechanism of biological neurons through mathematical formulas, pioneering the research direction of imitating the structure and function of the human brain with electronic devices. However, this model lacked autonomous learning capabilities and was difficult to adapt to complex task requirements. In 1957, Rosenblatt proposed the perceptron model on the basis of the MP model[2], which first realized a trainable neural network architecture, marking the transformation of artificial neural network research from pure theoretical discussion to engineering implementation.

The structural characteristic of the perceptron is a single-layer neural network architecture containing only an input layer and an output layer, which is essentially a binary linear classifier. Its core working mechanism is to receive multiple input signals, compare the weighted sum of weights with a threshold, output a binarized result through a step function, and automatically adjust the input weights and thresholds using the gradient descent algorithm to realize the learning of simple linearly separable problems. In the early stage of its birth, the perceptron showed certain effectiveness in tasks such as simple handwritten character recognition, promoting the first upsurge in neural network research.

However, the perceptron has significant limitations. In 1969, Minsky and Papert clearly pointed out in "Perceptrons: An Introduction to Computational Geometry" [3]that perceptrons can only handle linearly separable problems, cannot solve basic nonlinear problems such as "XOR", and cannot realize nonlinear modeling through simple hierarchical superposition. This conclusion directly led to a more than ten-year downturn in neural network research. Nevertheless, the weight learning mechanism and hierarchical signal processing idea proposed by the perceptron laid the foundation for the development of subsequent neural network models.

2.2. BP Neural Network: Breakthroughs in Multi-Layer Architecture and Backpropagation

With the improvement of computer computing power and the development of nonlinear theory, neural network research ushered in a recovery in the 1980s. In 1974, Werbos first proposed the backpropagation algorithm for training multi-layer neural networks in his doctoral thesis, but it did not have a wide impact because it was not published. In 1986, Rumelhart et al. re-proposed and systematically improved the Backpropagation (BP) algorithm[4], solving the gradient transfer problem of multi-layer neural networks, making it possible to train deep structures of "input layer-hidden layer-output layer", and the BP neural network thus came into being.

The core structural characteristic of the BP neural network is the introduction of a hidden layer,

forming a multi-layer perceptron architecture. The addition of the hidden layer enables the model to fit complex functions through multi-layer nonlinear transformations, breaking the linear classification limitation of the perceptron. Theoretically, a multi-layer feedforward neural network with a hidden layer containing enough neurons can approximate any continuous function of any complexity with arbitrary precision. Its working mechanism adopts a cyclic iterative mode of "forward propagation + backpropagation": in the forward propagation stage, input data is transmitted to the output layer after weighted by weights of each layer and processed by activation functions to obtain prediction results; in the backpropagation stage, the gradient of the prediction error with respect to each layer's parameters is calculated through the chain rule, and the weights and biases are adjusted layer by layer from the output layer to the input layer using the gradient descent algorithm to minimize the prediction error.

In terms of application scenarios, relying on its nonlinear fitting ability, BP neural networks are widely used in many fields such as image recognition, speech processing, and sales forecasting, such as nonlinear feature matching in mobile phone fingerprint recognition and speech signal conversion in voice assistants. However, the model still has obvious limitations: first, the training speed is slow, the multi-layer structure leads to a sharp increase in the number of parameters, and the repeated iterative adjustment process is time-consuming; second, it is prone to overfitting problems, where the model performs excellently on training data but has weak generalization ability on new data; third, the gradient vanishing problem initially appears, and as the number of network layers increases, the gradient gradually decays during backpropagation, making it difficult to effectively train deep networks. Despite its shortcomings, the multi-layer architecture and backpropagation training paradigm constructed by BP neural networks provide a core framework for the development of subsequent deep learning models.

3. Iteration and Breakthrough of Core Neural Network Models

3.1. Convolutional Neural Network (CNN): Efficient Extraction of Spatial Features

In the 1980s, with the increasing demand for image recognition tasks, problems such as parameter redundancy and insufficient spatial feature extraction capabilities of traditional neural networks in processing image data became increasingly prominent. In 1980, Japanese scholar Kunihiko Fukushima proposed the Neocognitron neural network[5], introducing a convolution-like structure to achieve translation-invariant processing of images, which became the prototype of convolutional neural networks. In 1989, Yann LeCun et al. proposed the classic LeNet-5 model[6], which first applied convolutional neural networks to handwritten digit recognition with an accuracy rate of 98%, laying the core architecture of CNN. In 2012, AlexNet achieved an accuracy rate of 84.7% in the ImageNet image classification competition, far exceeding traditional algorithms, proving the superiority of deep convolutional neural networks in complex image tasks and promoting deep learning into an explosive period.

The core structural characteristic of CNN is the adoption of a hierarchical architecture of "convolutional layer-pooling layer-full connection layer", realizing efficient extraction of spatial features through three core mechanisms: local receptive field, parameter sharing, and pooling dimensionality reduction. The convolutional layer captures local spatial features (such as edges, textures, and shapes) by sliding convolution kernels on the image; the parameter sharing mechanism greatly reduces the number of model parameters and improves computational efficiency; the pooling layer performs dimensionality reduction on feature maps through downsampling operations, retaining key features while reducing the risk of overfitting; the fully connected layer maps the extracted features to a specific category space to complete classification or regression tasks. This architecture imitates the receptive field mechanism of the biological visual cortex, enabling

automatic learning of hierarchical image features from low-level edge features to high-level semantic features.

In terms of application scenarios, CNN is mainly used to process data with spatial structures, such as image classification, object detection, image segmentation, and video analysis, and is widely applied in industries such as security monitoring, autonomous driving, and medical image diagnosis. Its limitations are mainly manifested in: first, weak ability to process temporal information, making it difficult to effectively capture temporal dependency relationships in data; second, the model's generalization ability is greatly affected by data distribution, and its adaptability to unseen image styles or deformation modes is poor; third, deep CNN still has the gradient vanishing problem, which needs to be improved through technologies such as residual connections.

3.2. Recurrent Neural Network (RNN) and Improved Models: Capability of Modeling Temporal Dependencies

With the rise of time-series related tasks such as natural language processing and speech recognition, there is a need for a model that can capture temporal dependency relationships in sequence data. In 1986, David Rumelhart et al. proposed the Recurrent Neural Network (RNN), whose core innovation is the introduction of a recurrent structure, enabling the network to use historical input information to assist current prediction, thereby possessing the ability to process time-series data. The birth of RNN filled the gap of traditional models in time-series data processing, but the gradient vanishing problem of the original model limited its ability to model long-sequence data.

The structural characteristic of the original RNN is that there are feedback connections between hidden layer neurons. Each node not only receives the current input but also the output of the previous node, forming a dynamic memory mechanism. Theoretically, this structure can propagate the influence of early inputs on subsequent outputs arbitrarily far through hidden states, but in actual training, due to the exponential decay during gradient backpropagation, it is difficult to retain long-range dependency information, that is, the gradient vanishing problem, making the model only able to learn short-term dependency relationships. Its application scenarios mainly include simple language models and short-term time-series prediction, but it performs poorly in complex tasks such as long text processing and speech recognition.

To solve the gradient vanishing problem of the original RNN, researchers have improved its structure and successively proposed Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). In 1997, Hochreiter and Schmidhuber proposed LSTM, which realizes dynamic control of information memory and forgetting through the introduction of gating mechanisms such as input gate, forget gate, and output gate. The forget gate determines how much historical information to retain, the input gate controls the amount of current information received, and the output gate adjusts the contribution of current memory information to the output. This structure enables the model to retain contextual information over a time span of dozens or even hundreds of steps, effectively overcoming the gradient vanishing problem. In 2014, Cho et al. proposed GRU, which simplifies the gating structure of LSTM by merging the input gate and forget gate into an update gate, while retaining the reset gate to adjust the weight of historical information, reducing the number of parameters and improving training efficiency while ensuring modeling effect.

The application scenarios of LSTM and GRU widely cover the field of long-sequence data processing, such as machine translation, speech recognition, long text sentiment analysis, and time-series prediction, becoming the basic models in the field of natural language processing. However, both still have limitations: first, they adopt a sequential processing method, which cannot realize parallel computing and has low training efficiency; second, their ability to model long-range

dependencies of ultra-long sequences is still limited, and the model structure is relatively complex with poor interpretability.

3.3. Transformer: Innovation of Attention Mechanism and Parallel Computing

The core structural characteristic of the Transformer is based on the self-attention mechanism and the encoder-decoder architecture. The self-attention mechanism calculates the correlation weight between any two elements in the sequence, enabling the model to simultaneously focus on all other positions in the sequence related to the current element when encoding it, realizing direct modeling of global dependency relationships and solving the limitation of local dependency modeling of traditional models. To further improve the feature expression ability of the model, the Transformer introduces the multi-head attention mechanism, which captures diverse relational features from different subspaces through multiple parallel attention heads, and then splices and fuses the results to enhance the model's ability to model complex semantic relationships. In addition, the Transformer adopts components such as a fully connected feedforward network and layer normalization, and improves the model's representation ability by stacking multiple layers of encoder-decoder structures.

In terms of application scenarios, relying on its global modeling ability and parallel computing advantages, the Transformer has quickly become the standard architecture in the field of natural language processing, widely used in tasks such as machine translation, text generation, and sentiment analysis. Pre-trained models based on the Transformer, such as the GPT series and BERT, have shown amazing generation and reasoning capabilities. At the same time, the idea of the Transformer has also been extended to the field of computer vision, forming visual Transformer models such as ViT, realizing breakthroughs in cross-modal tasks. Its limitations are mainly manifested in: first, poor adaptability to small-scale data, and the high model complexity leads to easy overfitting; second, the computational complexity of the self-attention mechanism grows with the square of the sequence length, resulting in low efficiency when processing ultra-long sequences; third, the model has a large number of parameters, requiring high computing power and high deployment costs.

4. Core Logic of the Evolution of Deep Learning Models

4.1. Modeling Upgrade from Local Dependency to Global Dependency

From the perspective of the evolution process of models, the transformation from local dependency modeling to global dependency modeling is one of the core logics. Early perceptrons and BP neural networks can only perform local linear or nonlinear transformations on input data, and cannot capture long-distance dependency relationships between data; CNN extracts local spatial features of images through the local receptive field mechanism, and although it can achieve a certain degree of global feature fusion through hierarchical stacking, it essentially still relies on the gradual transmission of local features, with limited global dependency modeling capabilities; RNN series models can capture the front-back dependency relationships of time-series data, but are limited by the gradient vanishing problem and can only effectively model short-term local dependencies; the self-attention mechanism introduced by the Transformer directly realizes the global correlation modeling between any elements in the sequence, breaking the constraint of local dependencies and enabling the model to more comprehensively capture complex relationships in data. This upgrading trend stems from the continuous improvement of task requirements. From simple classification tasks to complex semantic understanding and cross-modal generation tasks, models are required to have stronger global contextual awareness capabilities.

4.2. Efficiency Optimization from Sequential Processing to Parallel Computing

The transformation of the model's computing method from sequential processing to parallel computing is a key logic for improving training efficiency. In early models, RNN series models adopt a recurrent structure and must process elements sequentially in sequence. Each step of computing depends on the result of the previous step, making it impossible to realize parallel computing, resulting in low training efficiency and difficulty in processing large-scale data. Although CNN realizes a certain degree of parallel computing in the convolutional layer, the overall is still limited by the sequence of hierarchical transmission. The Transformer completely abandons the recurrent structure and realizes the synchronous processing of all elements in the sequence through the self-attention mechanism, transforming the computational complexity from linear growth dependent on sequences to parallel matrix operations, which greatly improves training efficiency and makes the training of large-scale pre-trained models possible. Behind this transformation is the dual drive of improved computing power and growing data scale. With the emergence of massive data, the traditional sequential processing method can no longer meet the demand for training efficiency, and parallel computing has become an inevitable trend in model evolution.

4.3. Evolution of Feature Extraction from Manual Design to Automatic Learning

The evolution of feature extraction methods from manual design to automatic learning is another core logic of the evolution of deep learning models. In the stage of traditional machine learning and early neural networks, feature extraction mainly relied on manual design, such as edge detection operators in image recognition and bag-of-words models in text processing. Manually designed features often have limitations and are difficult to adapt to complex data distributions. The emergence of CNN realized the automatic extraction of image features, replacing manually designed feature operators through the automatic learning of convolution kernels; RNN series models realized the automatic capture of temporal features; the Transformer further realized the automatic learning of high-level semantic features of multi-modal data such as text and images. This evolutionary trend enables models to more adaptively adapt to different data types and task requirements, reduces reliance on domain knowledge, and improves the generalization ability and versatility of models.

5. Future Development Directions of Neural Network Models

5.1. Lightweight and High Efficiency: Adapting to Edge Device Deployment

Current mainstream deep learning models mostly have problems such as large parameter scale, high computational complexity, and high energy consumption, making it difficult to adapt to the deployment requirements of edge devices (such as smartphones and Internet of Things devices). Therefore, lightweight and high efficiency have become important directions for future model development. Lightweight models greatly reduce the number of parameters and computational load while ensuring model performance through technologies such as model compression, parameter pruning, quantization, and knowledge distillation. For example, MobileNet series models use depthwise separable convolution instead of standard convolution to effectively reduce computational complexity; ShuffleNet improves feature reuse efficiency through channel shuffling mechanism. In the future, research on lightweight models will pay more attention to the balance between performance and efficiency, design dedicated architectures combined with hardware characteristics, and promote the wide application of deep learning technology in the field of edge

computing.

5.2. Modularization and Composability: Improving Model Generalization and Adaptability

Modularization and composability are important development directions to solve the problems of insufficient generality and poor adaptability of current models. Traditional models are mostly integrated architectures designed for specific tasks, making it difficult to flexibly adapt to different scenario requirements. Modular models decompose the network into multiple functionally independent modules, such as feature extraction modules, attention modules, and prediction modules, and realize rapid adaptation to different tasks through flexible combination and replacement of modules. For example, the encoder-decoder structure of the Transformer is essentially a modular design, which can adapt to different data types such as text and images by replacing different attention modules or feedforward network modules. In the future, modular research will further explore standardized module interfaces and adaptive module selection mechanisms, realize flexible customization and efficient migration of models, and improve the generalization ability and scalability of models.

5.3. Improvement of Interpretability and Reliability: Promoting Safe and Compliant Applications

Current deep learning models are mostly regarded as "black boxes" and lack sufficient interpretability, leading to potential safety risks in the application of models in key fields such as medical care, finance, and autonomous driving. Therefore, improving the interpretability and reliability of models has become one of the core directions of future research. Interpretability research will be carried out from multiple angles such as model structure design, feature visualization, and causal reasoning. For example, by introducing attention weight visualization technology to intuitively show the focus of the model; by constructing a causal relationship network to reveal the internal logic of model decisions. At the same time, reliability research will focus on improving the robustness of the model, enhancing the model's resistance to noisy data and adversarial examples, and ensuring the stable operation of the model in complex real environments. The improvement of interpretability and reliability will promote the safe and compliant application of deep learning technology in key fields and enhance users' trust in the technology.

5.4. Multi-Modal Fusion and Cross-Domain Transfer: Expanding Application Boundaries

The improvement of multi-modal fusion and cross-domain transfer capabilities will further expand the application boundaries of deep learning models. Current models are mostly designed for single-modal data (text, images, speech), while real-world tasks often need to process multi-modal information. In the future, multi-modal models will realize the fusion modeling of multi-modal data such as text, images, and speech through technologies such as unified representation space construction and cross-modal attention mechanisms, improving the model's ability to understand complex scenarios. At the same time, cross-domain transfer learning technology will solve the problem of data distribution differences in different domains through methods such as parameter transfer of pre-trained models and domain adaptive adjustment, reducing the training cost of models in new domains. The development of multi-modal fusion and cross-domain transfer technologies will promote the in-depth application of deep learning in complex scenarios such as intelligent interaction, autonomous driving, and virtual-real fusion.

6. Conclusion

The evolution of core neural network models in deep learning is an innovative process of continuously breaking through technical bottlenecks and adapting to task requirements. From the linear learning framework laid by early perceptrons, to the multi-layer nonlinear modeling realized by BP neural networks, to the special optimization of spatial and temporal data by CNN and RNN series models, and finally to the global modeling and parallel computing breakthroughs realized by the Transformer architecture through attention mechanisms, each model iteration has promoted the leapfrog development of deep learning technology. This evolution process follows the core logic of transforming from local dependency to global dependency, from sequential processing to parallel computing, and from manual design to automatic learning, reflecting the collaborative driving role of task requirements, improved computing power, and algorithmic innovation.

Looking forward to the future, deep learning models will continue to develop in the directions of lightweight, modularization, improved interpretability, and multi-modal fusion. These development directions will not only solve the current problems of models such as efficiency, generality, and reliability, but also further expand the application boundaries of deep learning, promoting the in-depth implementation of technology in edge computing, key industries, and complex scenarios. Faced with increasingly complex application requirements and technical challenges, future research needs to pay more attention to the combination of theoretical innovation and engineering practice, continuously break through the performance boundaries and application limitations of models, and promote deep learning technology to continuously empower the development of the artificial intelligence field.

References

[1] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5.4 (1943): 115-133.

[2] Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.

[3] Minsky, Marvin, and Seymour Papert. "An introduction to computational geometry." Cambridge tiass., HIT 479.480 (1969): 104.

[4] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533-536.

[5] Fukushima, Kunihiko. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." Biological cybernetics 36.4 (1980): 193-202.

[6] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (2002): 2278-2324.