Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33832

Ten Years of Deep Learning: The Evolution and Challenges from Convolutional Networks to Generative Models

Qiyuan Gui

Beijing University of Chemical Technology, International Education College, Beijing, China
gqy13837163779@outlook.com

Abstract. In the last ten years, deep learning has transformed artificial intelligence, leading to significant advancements in computer vision, natural language processing, and speech recognition. Deep learning technologies have consistently advanced the boundaries of AI, exemplified by the exceptional efficacy of Convolutional Neural Networks (CNNs) in image recognition, the robust functionality of Recurrent Neural Networks (RNNs) in processing sequential data, and the remarkable ingenuity of generative models in producing images and text. However, alongside rapid technological advancements, deep learning faces numerous challenges, such as poor model interpretability, low data upload efficiency, weak generalization capabilities, and high computational resource consumption. This article reviews the development of deep learning over the past decade, focusing on the technological evolution of CNNs, RNNs, and generative models, and explores feasible solutions to these challenges. The research methodology combines literature review and case analysis: systematically process key literature in the deep learning field and analyze specific cases to better reveal the internal logic and future trends of deep learning technology. This article aims to help readers process the development skeleton of deep learning, identify research gaps in certain areas, and use this information to better predict future trends. The study is based on a comprehensive review of seminal papers and practical example, utilizing tools such as bibliometric analysis and case study frameworks. The data is sourced from prominent academic databases and real-world applications. The findings highlight the significant advancements and ongoing challenges in deep learning, providing insights into potential future directions and areas for further research.

Keywords: deep learning, CNNS, RNNS, models

1. Introduction

Over the past decade, deep learning has transformed artificial intelligence, resulting in substantial progress in fields such as computer vision, natural language processing, and speech recognition. The outstanding efficacy of convolutional neural networks (CNNs) in image recognition and the strong proficiency of recurrent neural networks (RNNs) in managing sequential data, and the impressive creativity of generative models in producing images and text have all contributed to expanding the boundaries of AI. Nevertheless, as this technology advances rapidly, deep learning also encounters various challenges, including limited model interpretability, inefficient data processing, weak

DOI: 10.54254/2755-2721/2026.GU33832

generalization capabilities, and high demands for computational resources. This paper intends to analyse the evolution of deep learning during the last 10 years, concentrating on the technological developments of CNNs, RNNs, and generative models, while also suggesting possible methods to tackle these challenges. Importantly, the research approach adopted in this paper integrates a literature review with case studies: by comprehensively analyzing key publications in the field of deep learning and examining specific examples, the internal mechanisms and future directions of deep learning's evolution can be more clearly understood. This work will assist readers in grasping the historical context of deep learning's development and identifying existing research gaps, thereby enabling more accurate predictions about its future trajectory.

2. Basic theoretical and technical framework

Deep learning technology is now extensively employed across various domains, including autonomous driving, natural language processing, and image identification. The rapid advancement of deep learning can be attributed to the constant introduction of new technologies during its development. It is crucial to define deep learning before exploring its history: Multiple processing layers make up the computational model referred to as "deep learning," which is geared toward learning data representations with multiple levels of abstraction [1]. The main concept is that complicated data can be efficiently described and processed by using multi-layer neural network architectures to automatically learn how to carry out multi-level feature representations from data. This technology exhibits exceptional efficiency in processing photos, videos, speech, and audio, particularly in managing sequential data like text and speech [2].

2.1. The basic structure of neural networks

Neural networks constitute the foundational models of deep learning. A standard neural network comprises several layers: the input layer, hidden layer, and output layer. Each layer consists of several neurones (or nodes), which are interconnected by weights. The input layer acquires the raw data, the hidden layer derives features via nonlinear transformations, and the output layer produces the final predictions.

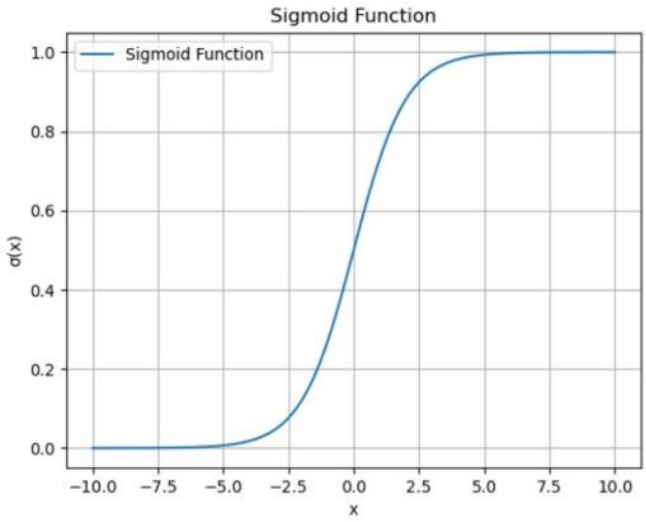

Figure 1. Sigmoid function image

DOI: 10.54254/2755-2721/2026.GU33832

2.2. Backpropagation algorithm

The primary algorithm employed for training neural networks is backpropagation. To minimise the loss function and continue training the multilayer network, the fundamental concept involves calculating the gradient of the loss function concerning the network parameters via the chain rule and subsequently updating the parameters by the gradient descent method. Forward propagation and reverse propagation are the two stages of this process. First, the input data is sent across each network layer via forward propagation, producing an output result and calculating the loss function.The gradient of the parameters for each layer was subsequently calculated when the gradient of the loss function was propagated forward from the output layer, layer by layer. The neural network is effectively trained when the loss function value consistently declines and the network parameters are adjusted in the direction opposite to the gradient through the gradient descent method. The chain rule can efficiently determine the gradient of each layer's parameters, and it is evident that the neural network is a composite function made up of several nonlinear modules. Combining the two significantly speeds up calculations and increases the back propagation algorithm's effectiveness.

2.3. Activation functions

The activation function is a nonlinear transformation unit in a neural network that enables the network to learn intricate functions. For instance, the Sigmoid function:

$$S(x) = \frac{1}{1+e^{-x}}$$

It is a common sigmoid function, also known as sigmoid growth curve, which has the properties of monotonically increasing and monotonically increasing inverse function. It can map variables to between [0,1], and is often used in the study of binary classification problems. And its derivative can also be expressed in terms of itself:

$$S'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = S(x)(1 - S(x))$$

The activation function is a nonlinear transformation unit in a neural network that enables the network to learn intricate functions. For instance, the Sigmoid function:

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

When the input approaches 0, the tanh function approximates a linear transformation. The function's structure resembles that of the sigmoid function, and tanh is "zero-centered," making tanh more advantageous than sigmoid in reality. In contrast to the sigmoid function, it possesses an additional zero-centered output. This enhances the stability and convergence of the data, thereby mitigating the offset in the learning process. Although the gradient of the Tanh function is substantial when the input is near 0, its derivative can still approach 0 for extreme input values, resulting in the vanishing gradient problem. Ultimately, the ReLU function:

$$f(x) = \max(0, x)$$

The ReLU function is a basic nonlinear transformation defined for an element x as the maximum of x and 0. Nonetheless, the output is not centred at zero, resulting in the Dead ReLU phenomenon: when the input is negative, the gradient becomes zero, causing the gradient of this neurone and

DOI: 10.54254/2755-2721/2026.GU33832

subsequent neurones to remain zero and unresponsive to any data, thereby preventing the corresponding parameters from being updated.

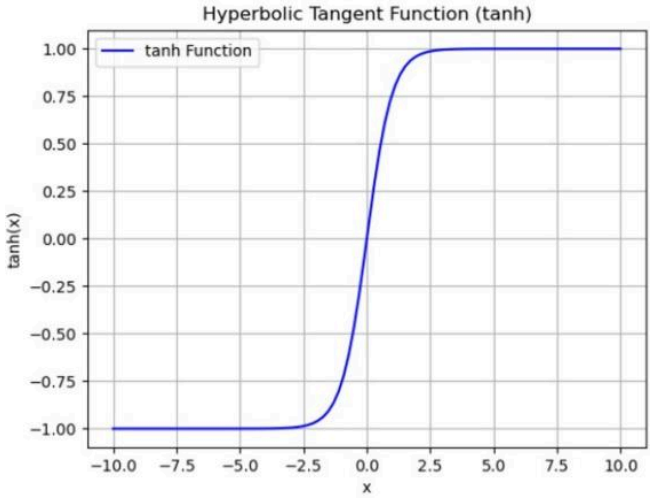

Figure 2. Tanh function image

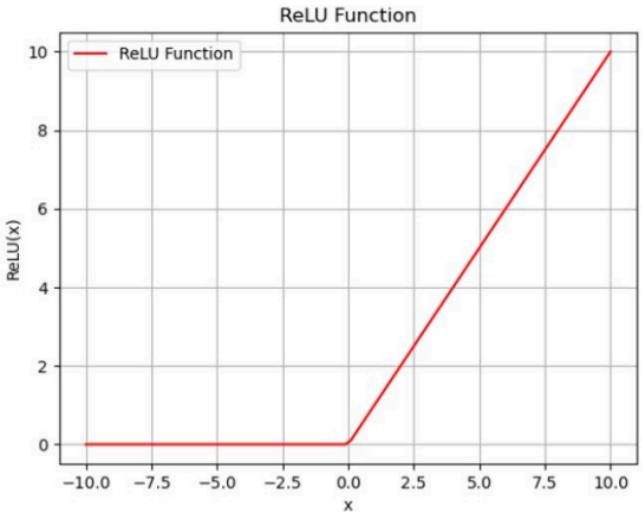

Figure 3. ReLU function image

2.4. Optimization methods

Large-scale non-convex optimization problems are typically used in deep learning training, which raises the bar for choosing the best optimization techniques to boost the model's performance. Stochastic gradient descent (SGD) is the most often used optimization technique: Stochastic gradient Descent (SGD) uses a single sample to update the parameter θ in each iteration [3]. This method adds randomization because each update is based on a single sample. Because of this randomness, the gradient calculated by SGD is not the precise global gradient because the gradient of the entire dataset is approximated using a single sample. The general trend will nevertheless converge toward the global optimal solution for convex optimization problems, even though the loss function at each iteration might not always decrease toward the global optimum. Eventually, the SGD result typically fluctuates about the global ideal solution.

DOI: 10.54254/2755-2721/2026.GU33832

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta\left(x^{(i)}\right))x_j^{(i)} \quad \text{(for every j)}$$

3. Development of CNN

3.1. Early exploration

The inception of CNNs may be dated to the 1980s, when researchers commenced investigations into the application of neural networks for image processing. The Neocognitron, an antecedent to Convolutional Neural Networks (CNN), was introduced by Japanese scientist Kunihiko Fukushima in 1980. The Neocognitron is a hierarchical neural network that processes visual patterns using local receptive fields and weight sharing, establishing the groundwork for subsequent convolutional neural networks (CNNs) [4].

3.2. Revival

The first completely convolutional neural network (CNN) was created in 1989 when Yann LeCun and his group developed the LeNet network design. The network was initially created to address the issue of handwritten digit recognition. In 1998, after over a decade of development, it was successfully implemented in the USPS's handwritten zip code recognition system and found practical use. LeNet is groundbreaking as it use convolutional layers to extract local features from images and incorporates pooling layers to diminish data dimensionality, establishing a foundation for subsequent advancements in deep learning [5].

3.3. Development and application

In 2006, Geoffrey Hinton and his team introduced Deep Belief Networks (DBN), which revitalised the advancement of deep learning [6]. By employing the method of integrating unsupervised pre-training with supervised fine-tuning, DBN effectively alleviates the gradient disappearance problem in deep neural networks, which rekindled the fervour of the academic community for deep learning, particularly convolutional neural networks. In 2009, Yann LeCun's team showcased the efficacy of CNNs for image categorisation through tests conducted on the CIFAR-10 dataset. At the same time, the wide application of GPU technology has injected strong computing power into deep learning, making it possible to train large-scale neural networks. In 2012, AlexNet, introduced by Alex Krizhevsky and colleagues, marked a significant milestone in the evolution of convolutional neural networks (CNNs). In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), AlexNet significantly decreased the top-5 the rate of error from 26% to 15.3%. Its success was mainly due to the adoption of ReLU activation function to significantly improve training efficiency.The Dropout technique was implemented to effectively mitigate overfitting, while the computational capabilities of GPUs were harnessed to significantly expedite the training process. AlexNet's exceptional performance not only reaffirmed the fundamental role of CNNs in computer vision but also ignited a surge in deep learning research. In 2014, the Visual Geometry Group (VGG) at Oxford University proposed VGGNet, which significantly improved the accuracy of image classification by using smaller convolution kernels (3x3) and deeper network architecture (16-19 layers). The core design idea of VGGNet is that a "deeper and wider" network can extract more complex features, which has a profound influence on the development of subsequent convolutional neural network architectures. In the same year, Google launched GoogLeNet (also known as Inception network), which introduced the innovative Inception module, which captured multi- scale features by using

DOI: 10.54254/2755-2721/2026.GU33832

convolution kernels of different sizes in parallel, and used 1x1 convolution to reduce computational complexity.Ultimately, it attains remarkable outcomes with a 6.7% top-5 mistake rate in ILSVRC 2014.

CNN was originally introduced to the field of object detection in 2014 with the proposal of Region-based CNN (R-CNN). Since then, the velocity and precision of detection have markedly improved due to advanced iterations such as Fast R-CNN and Faster R-CNN. To attain real-time detection capabilities with high accuracy, single-stage detectors such as You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD) were introduced in 2016. In the field of semantic segmentation, FCN (Fully Convolutional Network) proposed in 2015 is the pioneering application of CNN to pixel-level classification tasks [7]. Substituting the fully connected layer with a convolutional layer enables the network to process image inputs of arbitrary dimensions and produce segment maps of equivalent dimensions. Subsequent networks such as U-Net and DeepLab further improve the segmentation accuracy. In the field of image generation, the Generative The Generative Adversarial Network (GAN), introduced in 2014, facilitates the production of high-quality images via the adversarial training process between the generator and the discriminator, thereby enhancing the extensive use of Convolutional Neural Networks (CNN) in image generation, style transfer, and related applications [8].

4. Development of RNN

4.1. Early exploration

The Hopfield network, which John Hopfield proposed in 1982, used cyclic connections to store and retrieve particular patterns, laying the groundwork for the later creation of RNN. Then, in 1986, Michael Jordan proposed the Jordan network, the first real recurrent neural network that retains time step information while processing sequence input through context units. The Elman network, which improved the capacity to capture temporal dependency in sequence data through recurrent connections in hidden layers, was further proposed by Jeffrey Elman in 1990.

4.2. The proposal of LSTM and GRU

The inception of Long Short-Term Memory (LSTM) networks by Sepp Hochreiter and Jurgen Schmidhuber in 1997 represented a pivotal moment in the evolution of recurrent neural networks (RNNs). By adding memory units and gating mechanisms, LSTM effectively resolves the gradient vanishing issue in conventional RNN. The input, forget, and output gates in the core architecture are capable of precisely controlling the information flow and selectively retaining or discarding the time step information. This advancement allowed RNNs to handle extended sequence data and produced exceptional outcomes in fields such as speech recognition, machine translation, and time series forecasting, establishing it as one of the most effective designs inside the RNN category [9]. The Gated Recurrent Unit (GRU), a streamlined variant of LSTM, was subsequently introduced by Kyunghyun Cho and colleagues in 2014. In certain situations, GRU is recommended because it minimizes the number of parameters while keeping equivalent performance to LSTM in many tasks and is faster to train by combining input and forget gates. The advent of LSTM and GRU has led to the extensive application of RNN in Natural Language Processing (NLP) [10]. The neural language model put forth by Yoshua Bengio et al. in 2003 was one of the first uses of RNNS in NLP. By using RNNS to model the probability distribution of word sequences, it greatly enhanced the performance

DOI: 10.54254/2755-2721/2026.GU33832

of language modeling tasks. In 2011, RNNS was first used for machine translation jobs. In 2014, Kyunghyun Cho et al.'s Sequences-to-sequence (Seq2Seq) model catapulted RNNS to new heights.

5. Techniques and breakthroughs in generative models

Since their proposal in 2015, Diffusion Models—which gradually introduce noise and learn the inverse process to generate data—have shown promising results in the realm of image generation. Transformer-based generative models, such GPT-3 and DALL-E, are frequently used in text and image production. They were initially introduced in 2017 and are founded on the self-attention mechanism. To improve generation quality and diversity, hybrid models have recently integrated the advantages of many generative models, such VAE-GAN and the amalgamation of diffusion models and GAN.

6. Conclusion

This article summarises the advancements and challenges of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative models, encapsulating the evolution of deep learning during the past decade. Deep learning, utilising multi-layer neural networks, has effectively modelled intricate data; nonetheless, it continues to face challenges related to gradient vanishing and substantial resource usage. Although CNN has made great strides in image identification and other areas, it still has drawbacks such a high processing resource requirement and a heavy reliance on data. The emergence of Transformer presents a challenge to RNN's strong performance in sequence data processing, particularly when it comes to solving the vanishing gradient problem with LSTM and GRU. Despite advances in picture generation and other domains, generative models (e.g., GAN, VAE) continue to struggle with issues like computational complexity and generation quality. This paper's importance lies in its methodical classification of deep learning's technical development, as well as its internal logic and potential future developments. The absence of thorough examination of real-world application scenarios, particularly the application specifics in the financial and medical domains, is the weakness. Model lightweight, enhancing model interpretability, multimodal learning, and investigating continuous learning skills to handle difficulties in dynamic contexts and challenging tasks are some of the future research objectives.

References

[1] Lecun, Y., Bengio, Y., Hinton, G. (2015) Deep Learning. Nature., 521: 436–444.
[2] Collobert, R., et al. (2011) Natural language processing (almost) from scratch. J. Mach. Learn. Res., 12: 2493–2537
[3] Sutskever, I., Martens, J., Dahl, G., Hiten, G. (2013) On the importance of initialization and momentum in deep learning. PMLR., 28: 1139-1147.
[4] Kunihiko, F., (2007) Neocognitron. Scholarpedia., 2(1): 1717.
[5] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., (1998) Gradient-based learning applied to document recognition. Proceedings of IEEE., 86: 2278-2324.
[6] Hiten, G., (2009) Deep belief networks. Scholarpedia., 4: 5947.
[7] Long, J., Shelhamer, E., Darrel, T., (2015) Fully Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE CVPR., pp. 3431-3440.
[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., (2014) Generative Adversarial Nets. Advances in Neural Information Processing Systems 27., pp. 2672-2680.
[9] Hochreiter, S., Schmidhuber, J., (1997) Long Short-Term Memory. Neural Computation., 9: 8.
[10] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Empirical Methods in Natural Language Processing 2014., pp. 1724-1734.