ISSN 2959-6157

Review of Convolutional Neural Network

Zhenyuan Du

Abstract:

Over the past decade, the domain of deep learning has emerged as a swiftly expanding area of interest among researchers globally. It offers substantial benefits over traditional shallow networks, particularly in the realms of feature extraction and model fitting. Deep learning excels at uncovering intricate, distributed features from raw data, which exhibit robust generalization capabilities. It has triumphantly addressed challenges that were once deemed intractable within the field of artificial intelligence. With the exponential growth in the volume of training data and a marked increase in computational power, deep learning has achieved remarkable strides and found applications across a spectrum of domains, including but not limited to object detection, computer vision, natural language processing, speech recognition, and semantic parsing, thereby accelerating the evolution of artificial intelligence.

Deep learning encompasses a series of non-linear transformations organized hierarchically, with deep neural networks representing the predominant form of contemporary deep learning methodologies. Inspired by the neural connections found in the animal visual cortex, these networks are characterized by local connectivity, shared weights, and pooling operations. Such attributes not only simplify the network model and curtail the number of trainable parameters but also engender invariance to shifts, distortions, and scaling, thereby endowing the model with enhanced robustness and fault tolerance. This makes the training and optimization of the network structure more manageable.

This paper commences with a retrospective on the evolution of convolutional neural networks (CNNs). Subsequently, it delineates the architectures of neuron models and multilayer perceptron. It proceeds to dissect the CNN architecture, which typically encompasses a sequence of convolutional and pooling layers, culminating in fully connected layers, each serving distinct functions. The paper also elaborates on various enhanced algorithms for CNNs, such as the "Network in Network" and "Spatial Transformer Networks," while also introducing supervised and unsupervised learning methodologies pertinent to CNNs and highlighting several prevalent open-source tools.

Further, the paper scrutinizes the application of CNNs in various fields, including image classification, facial recognition, audio retrieval, electrocardiogram analysis, and object detection. It posits the amalgamation of CNNs with recurrent neural networks as a potential alternative for training datasets. The research culminates in the design of CNN structures with varying parameters and depths, supported by experimental analysis that elucidates the interplay among these parameters and the ramifications of their configuration. Ultimately, the paper synthesizes the merits and lingering challenges associated with CNNs and their practical applications.

Keywords: convolutional neural network; deep learning; network structure; training method; domain data

1 Introduction

Artificial neural networks serve as an emulation and approximation of their biological counterparts, forming an adaptive and nonlinear dynamic system interconnected by an expansive array of neurons. In 1943, the pioneering MP model—the first mathematical representation of neurons—was introduced by Psychologist McCulloch and mathematical logician Pitts, laying the groundwork for subsequent research endeavors. Advancing into the late 1950s and early 1960s, Rosenblatt enhanced the MP models by integrating learning mechanisms, thereby introducing the single-layer perceptron model and marking the initial practical application of neural network studies. Nonetheless, this model was limited in its inability to address linear inseparability issues. The breakthrough came in 1986 when Rumelhart and Hinton introduced the backpropagation network, a multi-layer feedforward architecture trained via an error backpropagation algorithm, which resolved many of the challenges that single-layer perceptrons could not. Despite this, as the number of layers in neural networks increased, traditional backpropagation networks faced issues such as local optimization, overfitting, and gradient dispersion, which temporarily sidetracked the exploration of deeper models.

In 2006, Hinton and colleagues published a seminal paper in Science outlining key points: (1) artificial neural networks with multiple hidden layers possess superior feature

learning capabilities; (2) the training challenges of deep neural networks could be effectively mitigated through "layer-wise pre-training," sparking a resurgence in the study of deep learning and reigniting interest in artificial neural networks. This pre-training approach in deep learning applies unsupervised learning to each layer sequentially, using the output of one layer's training as the input for the next, followed by fine-tuning the entire network with supervised learning. This method has been particularly effective in improving recognition or detection performance in scenarios with a limited number of labeled samples, such as in handwritten digit recognition or pedestrian detection.

Bengio has provided a systematic exposition of the network structures and learning methods within deep learning. Among the commonly utilized deep learning models are the Deep Belief Network (DBN), Stacked Denoising Autoencoders (SDA), and Convolutional Neural Networks (CNN). On January 28, 2016, AlphaGo, developed by Google DeepMind, made history by defeating a reigning European champion in Go, as highlighted in Nature magazine. This achievement, attributed to deep learning, demonstrated the profound potential of artificial intelligence. AlphaGo utilized value networks to assess board positions and policy networks to select moves, with both types of networks being deep neural networks, marking another significant milestone in the advancement of AI through deep learning.

2 Overview of Convolutional Neural Networks

2.1 Neurons

Neurons act as the fundamental computational elements within artificial neural networks, typically designed to accept multiple inputs and produce a single output. The structural model of a neuron is depicted in the accompanying Figure 1:

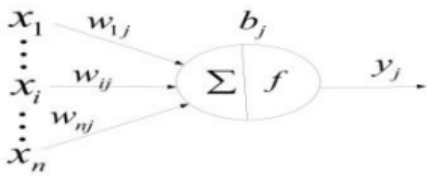

Figure 2-1 neuron model

In this model, *xi* signifies each of the input signals, with 'n' being the total number of inputs that are fed into neuron 'j' at the same time. The term *wij* denotes the weight associated with the connection from input signal *xi* to neuron 'j'. The bias of the neuron, represented by *bj*, adjusts the activation threshold. The output of the neuron is denoted by *yj*. The relationship between the inputs and the output is articulated through the following equation, which involves a weighted sum of inputs followed by a bias adjustment and passed through an activation function:

$$y_j = f[b_j + \sum_{i=1}^{n}(x_i * w_{ij})]$$

This formulation encapsulates how the neuron integrates its inputs to generate an output, where the activation function *f* can introduce non-linearity into the network, enabling it to model complex patterns.(1)(2)

2.2 Multi-layer perceptron

The Multilayer Perceptron (MLP) is an advanced neural network architecture that integrates an input layer, one or more hidden layers, and an output layer, offering a sophisticated approach to address the limitations inherent in single-layer perceptron. Unlike their simpler counterparts, MLPs possess the capability to resolve issues of linear inseparability, where data cannot be distinctly partitioned by a single decision boundary. This enhanced functionality is achieved through the added complexity of the hidden layers, which enable the network to learn and model intricate patterns and relationships within the data. By leveraging the compounded processing power of these layers, the MLP can effectively tackle a broader spectrum of problems, making it a versatile tool in the realm of machine learning and artificial intelligence.(2)

2.3 Convolutional Neural Networks

The foundational architecture of a Convolutional Neural Network (CNN) is composed of several key components, including an input layer, multiple convolutional layers, pooling (or sampling) layers, a fully connected layer, and an output layer. These elements are typically arranged in an alternating sequence: a convolutional layer is followed by a pooling layer, which in turn is succeeded by another convolutional layer, and this pattern continues throughout the network.

Each neuron within the convolutional layer is connected only to a local region of the input, rather than the entire input space. This local connectivity is a defining feature of CNNs, as it allows the network to focus on specific features within the input data. The neuron's response is computed by applying a set of filters (or kernels) across the input, which are shared across the entire input space, a concept known as weight sharing. This results in a weighted sum of the inputs, which is then typically passed through an activation function to introduce non-linearity into the model.

The pooling layer, which follows the convolutional layer, performs a down sampling operation that reduces the spatial dimensions of the feature maps, thus reducing the

computational load and helping to prevent overfitting. The pooling operation also provides a form of translation invariance, as the network becomes less sensitive to the exact location of features within the input.

After several stages of convolution and pooling, the high-level reasoning in the CNN is performed by the fully connected layer, which processes the features extracted by the convolutional and pooling layers to make final decisions or classifications. This comprehensive structure endows CNNs with their powerful capabilities for feature extraction and pattern recognition, making them particularly adept at handling image, video, and speech data.(3)

3 Advanced CNN Algorithms

3.1 Network In Network (NIN) Structure

The traditional CNN relies on a generalized linear model (GLM) for its convolution filters, which, while effective, operates at a lower level of abstraction. To capture more complex features, Lin et al. introduced the "Network In Network" (NIN) model. This innovative approach replaces the standard convolution process with a micro neural network, enhancing the network's ability to represent and abstract features.

The NIN structure employs a global average pooling layer to supplant the conventional fully connected layer, thereby significantly reducing the model's parameters and mitigating the risk of overfitting. This pooling layer functions as a structured regularizer, promoting alignment between feature representations and their respective categories without the need for parameter optimization.

The NIN's nonlinear convolution layer leverages an MLP to slide over the input, extracting features in a manner akin to traditional convolution but with the added depth and complexity management of an MLP. This design not only integrates the BP algorithm for training, aligning with CNN structures, but also capitalizes on the feature reuse inherent in deep models.

Lin et al. demonstrated the algorithm's efficacy on datasets such as MNIST and SVHN, validating its performance. Furthermore, Xu et al. proposed an ML-DNN model that integrates the NIN structure, showcasing its superiority over other methods like sparse coding when applied to the same databases.(6)

3.2 Spatial Transformer Networks (STNs)

Despite the robustness of CNNs as classification models, they can be sensitive to the spatial variations present in data. To counter this, Jaderberg et al. introduced the Spatial Transformer Network (STN), a learnable module designed to adapt to such variations.

The STN module is composed of three main components: a localization network for detecting key features, a grid generator for creating a transformation grid, and a sampler for applying the transformation. This module can be seamlessly integrated into the CNN architecture, either at the input layer or subsequent to the convolutional layers, without altering the underlying CNN structure.

STNs enable the model to dynamically adjust to translations, scaling, rotations, and other spatial transformations, thereby increasing the model's robustness to input variations. Moreover, the STN's efficiency ensures that it has a minimal impact on the training speed of the CNN.(5)

By incorporating these advanced algorithms, CNNs can further refine their feature extraction capabilities, leading to improved performance across a diverse range of tasks and datasets.

4 Training Techniques and Open-Source Tools

4.1 Training Techniques

Convolutional Neural Networks (CNNs) predominantly employ supervised learning methodologies and typically forgo the need for unsupervised pre-training. In practical scenarios, the abundance of unlabeled data compared to the scarcity of annotated samples presents a challenge, given the labor-intensive process of manual data tagging. Despite these challenges, acquiring a substantial volume of labeled training instances is essential for the thorough training of a supervised CNN to enhance its generalization capabilities. This necessity, to some extent, limits the practical applicability of CNNs. Nonetheless, CNNs are also capable of undergoing unsupervised training. However, existing unsupervised learning algorithms often necessitate the tuning of numerous hyperparameters, which can complicate their application.(4)

4.2 Open-Source Tools

The widespread adoption of deep learning across various research domains is significantly supported by numerous high-quality open-source deep learning frameworks. Among the most popular are Caffe, Torch, and Theano.

Caffe is an architecture grounded in the C language and tailored for CNN algorithms, offering an exemplary implementation of CNNs. It operates on both CPU and GPU platforms and offers interfaces for MATLAB and Python. Caffe provides a comprehensive suite for model training, testing, fine-tuning, and deployment. It allows users to introduce new data formats, network layers, and loss functions. Known for its speed, Caffe can handle the training of over 40 million images on a single K40 or Titan GPU within a day. The Caffe community also fosters user participation in development and discussions. However,

Caffe's extensibility is somewhat limited by its legacy architecture, particularly when it comes to handling Recurrent Neural Networks (RNNs), which points to a need for enhanced flexibility.

Torch is a scientific computing framework written in Lua and C, designed to support machine learning algorithms. It offers a pliable environment for the conception and training of machine learning models and is compatible with embedded platforms like iOS and Android. The latest iteration, Torch7, has significantly enhanced the training velocity of CNNs. Torch's time-domain convolution benefits from variable input lengths, particularly advantageous for natural language tasks. However, Torch lacks a Python interface.

Theano is a Python library that facilitates the definition, optimization, and evaluation of mathematical expressions, with the majority of NumPy functions being executable on GPUs. Theano's auto-differentiation capabilities make it well-suited for gradient-based methods. It also enables the straightforward and efficient implementation of RNN models. Nonetheless, Theano's compilation process can be slow, and initializing the library requires time.

5 Practical Applications

5.1 Image Classification

In the realm of image processing, CNNs have seen extensive application. Krizhevsky and colleagues initially utilized CNNs in the LSVRC-12 competition, achieving state-of-the-art classification results with their deeper CNN model, known as AlexNet, which incorporated ReLU and dropout techniques. AlexNet's architecture comprises 5 convolutional layers and 2 fully connected layers. The use of ReLU simplified the model's computations and accelerated training, while dropout techniques bolstered the model's robustness and mitigated overfitting. Additional strategies, such as image translation and transformation, further curtailed overfitting.

Szegedy et al. later introduced GoogLeNet, a 20+ layer CNN that employs 3 types of convolution operations, enhancing computational resource utilization and reducing parameters significantly compared to prior models. GoogLeNet achieved superior accuracy, securing the top position in the LSVRC-14 competition's "specified data" category.

Simonyan et al. emphasized the significance of network depth, demonstrating that increasing it through additional convolutional layers with 3x3 kernels effectively improves model performance, as evidenced by the VGG model. The VGG model also reduces parameter count by substituting large convolution kernels with multiple smaller ones, enhancing the discriminative power of the decision function. It secured the second position in the LSVRC-14 competition, underscoring the importance of depth in visual representation.

5.2 Face Recognition

In face recognition, DeepFace refines the traditional four-step process—detection, alignment, face representation, and classification—by employing 3D face alignment and a deep neural network. DeepFace's initial layers capture low-level features, and its sampling layer enhances robustness to small offsets. The subsequent use of non-shared convolutional layers caters to the varying local statistical features in aligned face images. With two fully connected layers, DeepFace captures feature correlations across different regions of the face image. When applied to the Labeled Faces in the Wild (LFW) database, DeepFace achieved a recognition accuracy close to human levels, overcoming previous methodological limitations.

5.3 Audio Retrieval

Hamid and colleagues combined CNNs with Hidden Markov Models for speech recognition, achieving a 10% reduction in error rates compared to conventional neural network models on the TIMIT database. This indicates the potential of CNNs in enhancing speech recognition accuracy. Subsequent research has explored limited weight sharing in CNNs for better handling of speech features, although this technique has been primarily applied to single convolution layers.

5.4 ECG Analysis

ECG analysis serves as a crucial clinical diagnostic tool for cardiovascular diseases. With the rise of telemedicine, greater access to expert diagnostic services has been facilitated. A review of literature from 2000 to 2015 highlights the increasing application of data mining techniques in cardiovascular disease analysis, with neural networks and support vector machines demonstrating higher accuracy. However, the complex and variable nature of ECG data in big data analysis presents challenges for traditional neural network applications.(1)

5.5 CNN Advantages

CNNs are characterized by local connectivity, weight sharing, pooling operations, and multi-layer architectures. They can automatically learn features from large datasets through multi-layer nonlinear transformations, reducing reliance on handcrafted features. The depth of CNNs is vital, as increasing it enhances the network's ability to fit objective functions and extract distributed features effectively.

Despite their widespread use in fields like pattern recognition and artificial intelligence, CNNs present several

areas for further research, including deepening the understanding of the Hubel-Wiesel model, determining optimal network structures and hyperparameters, addressing dataset distribution discrepancies, and potentially integrating primate visual systems to improve performance.(8)

In summary, CNNs, with their multifaceted capabilities, continue to be a focal point of research, with ongoing advancements expected to expand their application into new domains.

References

(1) Latif, Ghazanfar ; Alghazo, Jaafar ; Khan, Majid Ali ; Ben Brahim, Ghassen ; Fawagreh, Khaled ; Mohammad, Nazeeruddin AIMS mathematics, 2024-01, Vol.9 (8), Deep convolutional neural network (CNN) model optimization techniques—Review for medical imaging.[site on 23/8/2024]

(2) Kattenborn, Teja ; Leitloff, Jens ; Schiefer, Felix ; Hinz, Stefan ISPRS journal of photogrammetry and remote sensing, 2021-03, Vol.173, p.24-49, Review on Convolutional Neural Networks (CNN) in vegetation remote sensing

(3) Renjith, V. R. ; Judith, J. E. Balas, Valentina E ; Suresh, L. Padma ; Panda, Ganapati AIP Conference Proceedings, 2023, Vol.2904 (1), Explainable artificial intelligence for gastrointestinal cancer using CNN-a review.

(4) Alzubaidi, Laith ; Zhang, Jinglan ; Humaidi, Amjad J. ; Al-Dujaili, Ayad ; Duan, Ye ; Al-Shamma, Omran ; Santamaría, J. ; Fadhel, Mohammed A. ; Al-Amidie, Muthana ; Farhan, Laith

Journal of big data, 2021-03, Vol.8 (1), p.53-53, Article 53, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions.

(5) Hara, Kensho ; Kataoka, Hirokatsu ; Satoh, Yutaka 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, p.6546-6555, Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?

(6) Kim, Yoonsik ; Hwang, Insung ; Cho, Nam IkarXiv.org, 2017-01, A New Convolutional Network-in-Network Structure and Its Applications in Skin Detection, Semantic Segmentation, and Artifact Reduction

(7) Adrian J. Shepherd 1961- author. 1997, Second-order methods for neural networks : fast and reliable training methods for multi-layer perceptrons.

(8) Salehi, Ahmad Waleed ; Khan, Shakir ; Gupta, Gaurav ; Alabduallah, Bayan Ibrahimm ; Almjally, Abrar ; Alsolai, Hadeel ; Siddiqui, Tamanna ; Mellit, Adel Sustainability, 2023-03, Vol.15 (7), p.5930, A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope.