Applied Intelligence
https://doi.org/10.1007/s10489-020-02015-5

Research on improved wavelet convolutional wavelet neural networks

Jing-Wei Liu [1,2] · Fang-Ling Zuo [3] · Ying-Xiao Guo [1] · Tian-Yue Li [1] · Jia-Ming Chen [2]

Accepted: 9 October 2020

Abstract

Convolutional neural network (CNN) is recognized as state of the art of deep learning algorithm, which has a good ability on the image classification and recognition. **The problems of CNN are as follows:** the precision, accuracy and efficiency of CNN are expected to be improved to satisfy the requirements of high performance. The **main work is as follows:** Firstly, wavelet convolutional neural network (wCNN) is proposed, where wavelet transform function is added to the convolutional layers of CNN. Secondly, wavelet convolutional wavelet neural network (wCwNN) is proposed, where fully connected neural network (FCNN) of wCNN and CNN are replaced by wavelet neural network (wNN). Thirdly, image classification experiments using CNN, wCNN and wCwNN algorithms, and comparison analysis are implemented with MNIST dataset. **The effect of the improved methods are as follows:** (1) Both precision and accuracy are improved. (2) The mean square error and the rate of error are reduced. (3) The complexitie of the improved algorithms is increased.

Keywords Wavelet convolutional neural network · Convolutional neural network · Wavelet neural network · Deep learning · Image analysis

1 Introduction

Convolutional neural network (CNN) is a typical deep learning method which is based on feature extraction of convolution calculation [9]. It is widely applied to fields of prediction, classification [14] etc. CNN can solve high-dimensional problems which are difficult for traditional machine learning methods [19]. The ability to minimize the system error between the label and the inference [22] of CNN is much more powerful especially in the application of image processing. The neuron weights [12] of CNN are modified by forward propagation and error back propagation [15]. In recent years, the ability of CNN becomes more powerful because the distributed computing power has been greatly improved. Apart from image recognition [3], CNN are also applied in the other fields [11] such as text classification [26], control system [1] and target tracking [21].

The first two authors contributed equally to this work.

✉ Jing-Wei Liu
liujingwei@cueb.edu.cn

✉ Fang-Ling Zuo
zuofangling@cueb.vip

1 Information College, Capital University of Economics and Business, Beijing 100070, China

2 Information Department, Beijing university of technology, Beijing 100124, China

3 School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 100081, China

The development history of CNN is as follows The earliest study about CNN can be traced back to Fukushima, who mimicked the visual cortex of an organism and proposed the Neocognition model [7]. Time-Delay Neural Network (TDNN) was proposed by Alexander Waibel et al. in 1987 [27]. It is proved in TDNN that more hidden layers have greater feature extraction capabilities, which becomes the foundation of further optimization of CNN. After a series of improvements, He-Kaiming et al. released ResNet in 2015 [8]: the network manages to skip some neuron nodes to achieve higher performance. In 2017, Gao Huang et al. proposed DenseNet.

Problems of CNN can be summarized as follows (1) The precision, accuracy and efficiency of CNN are expected to be improved. (2) High-dimensional information contains more details, which is difficult to be learned such as datasets of MINIST and CIFAR. Even human brain also tends to ignore the high-dimensional information. (3) CNN is more complex than classical neural network, but the trained model of CNN cannot be well explained. It is proved that randomly generated

Published online: 27 November 2020

network of CNN can solve difficult problems better than the carefully designed network sometimes. More intelligent module which can identify more detailed information is expected.

Wavelet transform (WT) is often used in deep learning [5, 16, 24]. Many features can be obtained by the discrete wavelet transform which have been improved by researches. The application fields based on WT and deep learning methods are image classification [10, 23], computer vision [4, 17], texture classification [6], etc.

The applications based on wavelet neural network (WNN) in deep learning are as follows In 2019, Pengju Liu et al. proposed a Multi-level Wavelet Convolutional Neural Networks(MWCNN) [16], which is proved to increase the receptive field by reducing the number of map. The Multi-Path Learnable Wavelet Neural Network for Image Classification was introduced by De Silva et al. [5]. This model introduces a multi-path layout with several levels of wavelet decompositions. In the domain of prediction, a convolutional LSTM network using the wavelet decomposition has been proposed in 2018 [28]. It takes the wavelet decomposition as the method of feature extraction rather than the manual feature extraction, which has been also proved by Kiskin et al. in 2017 [13].

The advantages of wavelet analysis are as follows Wavelet analysis has been widely used in signal processing and analysis. Wavelet analysis method is called mathematical microscope [2, 18], which is considered as a powerful tool for zooming details of sound, image, etc. Although the wavelet transformation has some complexity [32], the powerful detail extraction ability of wavelet transformation is helpful and important to solve the above problems of CNN [20].

The motivation of this research is to solve the CNN's problems based on the advantages of the WT. The importance of the research is that the improvements of CNN neurons are focused. Different from the ability of network with deeper layers, it is believed that the improvements of each neuron of CNN can improve the features identification and learning ability of the whole CNN [30]. Wavelet analysis is adopted [29] to improve the CNN network in this study.

The contributions of this study are as follows (1) The wavelet-based Convolutional Neural Network (wCNN) is proposed, where the wavelet transformation is adopted as the activation function in Convolutional Pool Neural Network (CPNN) of CNN. (2) Based on wCNN, the wavelet-ased Convolutional wavelet Neural Network (wCwNN) is proposed, where the Fully connected Neural Network (FCNN) of wCNN is replaced by wavelet Neural Network (wNN). (3) Comparative experiments between CNN, wCNN and wCwNN are implemented on the MNIST dataset.

The following sections are organized as follows The traditional CNN model is described in the second section. The improved wCNN is proposed in the third section. The improved wCwNN is proposed in the fourth section. The performance of CNN, wCNN and wCwNN is verified, analyzed and compared respectively with MNIST dataset in the fifth section. Discussion, conclusions and further research are given in the sixth section.

2 Model of convolutional neural network (CNN)

2.1 Structure of CNN

The structure of classical CNN is shown in Fig. 1. There are two parts in CNN: the first part is CPNN, and the second part is FCNN. In CPNN, the first layer is an input layer, and the following layers of CPNN are several pairs of convolutional layers and pooling layers. In FCNN, the first layer is an input

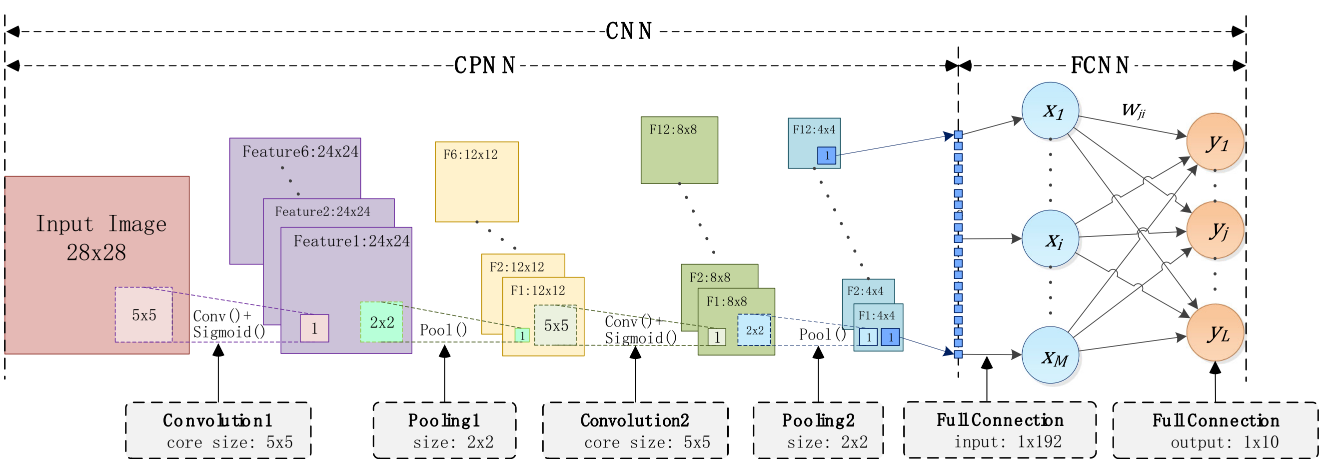

Fig. 1 Structure of CNN

layer, and the second layer of FCNN is an output layer, both layers of FCNN are fully connected.

The relation and features of CPNN and FCNN are as follows. (1) The input layer of CPNN is the input layer of CNN; (2) The last layer of CPNN is the input layer of FCNN; (3) The output layer of FCNN is the output layer of CNN; (4) The activation function of the convolutional layer in CPNN and the output layer in FCNN is sigmoid function; (5) There are not any activation functions in the input layer and pooling layer of CPNN and the input layer of FCNN.

2.2 Algorithm of CNN

The algorithm of CNN can be described as follows: (1) Initializing weights between layers and bias of neurons. (2) Forward propagating. (3) Calculating the mean square error (MSE) of all samples according to the loss function. (4) Calculating the errors of back propagating for each layer, which are the results of derivation by the chain rule. (5) Applying gradient to adjust the weights and bias according to the back-propagated errors. (6) Repeating the step (2) to step (5) until the MSE is small enough. (6) Evaluating the accuracy, precision and efficiency.

2.2.1 Forward propagation of CNN

Forward propagation of CNN is the calculation process from the input layer to the output layer, which can be described as follows: (1) The input layer of CNN is filled by a two-dimensional matrix of pixels of an image. (2) Forward propagation is calculated in convolutional and pooling layers (CPNN). (3) Forward propagation is calculated in fully connected layer (FCNN).

Definition 1: net^l and O^l are the input and the output of neurons in layer l. The output of each neuron can be calculated according to the input and the activation function of each neuron. l is the layer number, e.g. $l = 1$ stands for the first layer, and $l = -1$ stands for the last layer. i and j are the row number and column number respectively.

According to the above definition, net^{-1}, net^{-2} and net^{-3} stand for the input of the last FCNN layer, the input of the first FCNN layer and the input of the layer before FCNN (i.e. the last layer of pooling layers) respectively. The data structures of net^l and O^l of each layer of CPNN are two-dimensional matrix, while the net^l and O^l of each layer in FCNN are one-dimensional vectors.

Definition 2: w_{ij}^l and b_j^l are the weights and bias of layer l. w_{ij}^{-1} and b_j^{-1} are the weights and bias of the last layer of FCNN. If the layer l is a convolutional layer or a pooling layer, the size of the convolutional kernel or the pooling windows can be expressed as $size^l \times size^l$. If layer l is a fully-connected layer, the number of neurons is expressed as $size^l$.

Definition 3: int(x) is the function for getting the integer part of x, e.g., int(5.1) = int(5.7) = 5.

Forward propagation of convolutional layer The input of convolutional layer (net^l) can be calculated according to Eq. (1). The net_{mn}^l stands for each input value of neurons in layer l. The convolution(O^{l-1}, w^l, m, n) is the function for convolution calculations. The O^{l-1} is the output of the previous layer. The w^l is the matrix of weights between the input of layer l (net^l) and the output of the previous layer (O^{l-1}). The b^l is the bias of layer l.

$$net_{mn}^l = \text{convolution}\left(O^{l-1}, w^l, m, n\right) + b^l = \sum_{i=0}^{size^l-1}\sum_{j=0}^{size^l-1}\left(O_{m+i,n+j}^{l-1}\cdot w_{i,j}^l + b^l\right) \tag{1}$$

An example of convolution operation is provided. If $x = \begin{matrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33}\end{matrix}$, $y = \begin{matrix} y_{11} & y_{12} \\ y_{21} & y_{22}\end{matrix}$, the formula of convolution(x, y) can be expressed as Eq. (2):

$$\text{convolution}(x,y) = \begin{matrix} x_{11}y_{11}+x_{12}y_{12}+x_{21}y_{21}+x_{22}y_{22} & x_{12}y_{11}+x_{13}y_{12}+x_{22}y_{21}+x_{23}y_{22} \\ x_{21}y_{11}+x_{22}y_{12}+x_{31}y_{21}+x_{32}y_{22} & x_{22}y_{11}+x_{23}y_{12}+x_{32}y_{21}+x_{33}y_{22}\end{matrix} \tag{2}$$

The output of the convolutional layer l (O_{mn}^l) can be calculated as Eq. (3), where sigmoid() is the activation function.

$$O_{mn}^l = \mathrm{F}\left(net_{mn}^l\right) = \text{sigmoid}\left(net_{mn}^l\right) = \frac{1}{1+e^{-net_{mn}^l}} \tag{3}$$

Forward propagation of pooling layer Definition 4: The function pool(x) represents the average pooling of matrix x. The formula of pool(x) can be expressed as Eq. (4). The $size^l$ stands for the size of the pooling window.

$$y_{ij} = \text{pool}(x,i,j) = \frac{\sum_{m=1}^{size^l}\sum_{n=1}^{size^l} x_{size^l\times(i-1)+m,\, size^l\times(j-1)+n}^{l-1}}{size^l \times size^l} \tag{4}$$

An example of average pooling is provided: If $x = \begin{matrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44}\end{matrix}$, The pooling result is calculated as Eq. (5).