CNN and RNN mixed model for image classification

Qiwei Yin[1], *Ruixun* Zhang[2] , and *XiuLi* Shao[1,*]

[1]College of Computer and Control Engineering, Nankai University, Tianjin, China
[2]Laboratory for Financial Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract. In this paper, we propose a CNN(Convolutional neural networks) and RNN(recurrent neural networks) mixed model for image classification, the proposed network, called CNN-RNN model. Image data can be viewed as two-dimensional wave data, and convolution calculation is a filtering process. It can filter non-critical band information in an image, leaving behind important features of image information. The CNN-RNN model can use the RNN to Calculate the Dependency and Continuity Features of the Intermediate Layer Output of the CNN Model, connect the characteristics of these middle tiers to the final full-connection network for classification prediction, which will result in better classification accuracy. At the same time, in order to satisfy the restriction of the length of the input sequence by the RNN model and prevent the gradient explosion or gradient disappearing in the network, this paper combines the wavelet transform (WT) method in the Fourier transform to filter the input data. We will test the proposed CNN-RNN model on a widely-used datasets CIFAR-10. The results prove the proposed method has a better classification effect than the original CNN network, and that further investigation is needed.

1 Introduction

Current CNN [1-2] networks have become a standard method of machine learning for 'mesh data' (pictures, videos, etc.). Researchers have created many different structural network models on the basis of the CNN model, and have proved to be successful at a variety of image correlation problems including handwritten digital recognition [3], natural picture recognition, etc. Among them, [4] is one of the most classic networks currently used for image recognition. AlexNet [5] is also a neural network based on Convolution + Pool structure, It is one of the origins of deep learning. Starting from AlexNet. After that, there are a lot of CNN models appearing, ZF-Net [6] began to use the de-convolution to output the feature of the CNN network middle layer, and this feature enables people to more intuitively understand the working principle and network architecture within CNN. They discover the characteristics of edges, colors, etc. in shallow CNN detection images, and deeper CNN layer will detect shape features of objects to be identified. While the RNN has become the standard method for machine learning of sequence data (audio, natural

* Corresponding author: shaoxl@nankai.edu.cn

language, etc.). NLP [7] and machine translation [8-9] are the most current applications. Combining RNN processing sequence and CNN to process image data, the main research fields include image tagging [10], target detection [11], video screen behavior detection, etc. [12] innovatively proposed a network model ReNet for image classification. This network replaces the Convolution + Pooling layer in CNN using four common UNI-Dimensional RNNs layers. In this paper, we use a similar structure in the ReNet model, and propose our model CNN-RNN model for image classification. The model has some similarities with ResNet [13], this article is a reference to the jump connection method of ResNet model, using RNN Layer to Calculate Continuity Features in CNN Layer, and then connect these features to the final fully connected network for classification prediction. In this work, we will test the proposed CNN-RNN model on a widely used object recognition datasets CIFAR-10[14]. Experiments have shown that the CNN-RNN model has better accuracy and call more research in the future.

2 Model description

In this paper, we make some changes to the RNN layer in [12], and no longer calculate sequence results in four directions. We only calculate the sequence results in two directions, as shown in Figure 1 (example picture is for reference only, non-experiment dataset picture), it is a one layer structure of RNN calculations. It can be seen as a bidirectional RNN model. The RNN model can be LSTM (Long Short Term Memory) [15], GRU (Gated Recurrent Unit) [16], IRNN [17] or ConvLSTM (Convolutional LSTM Network) [18].

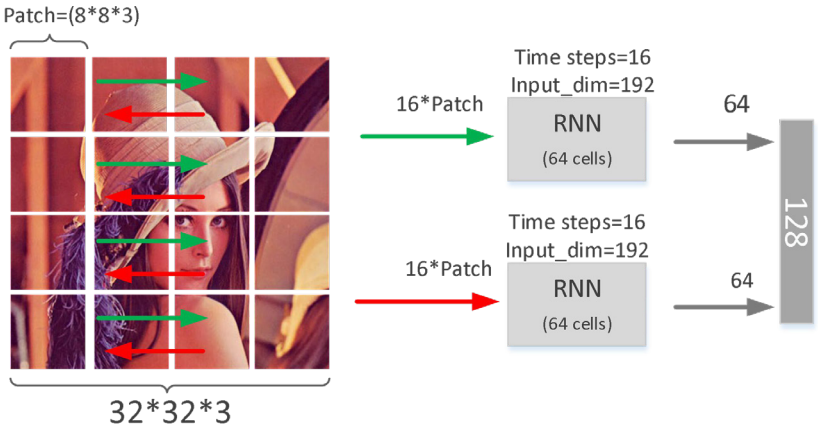

Fig. 1. One-Layer RNN.

Denote the input data of the RNN layer is $x \in R^{w \times h \times c}$, w and h denote the width, height of input x, and if x is the original picture, the c represents the number of color channels, other c represents the number of filters (or the number of convolution kernels). According to [6], the output of the convolutional layer after deconvolution operation is the result of the edge, texture, and shape of the image. Therefore, the output of the convolution layer is the original image after the filter processing, which has the original image part of the feature attributes. Define the identification area (Patch in the figure 1) $w_p \times h_p$, and we can split the input into a $M \times N$ patches $x_{m,n} \in R^{w_p \times h_p \times c}$, where $M = \frac{w}{w_p}, N = \frac{h}{h_p}$, Inputting $x_{m,n}$ into a bidirectional RNN network in a sequence:

$$S_{i,j}^{F} = f_{FWD}(S_{m,n-1}^{F}, x_{m,n}),\ for n = 1,...N \quad (1)$$

$$S_{i,j}^{R} = f_{REV}(S_{m,n+1}^{R}, x_{m,n}),\ for n = N,...,1 \quad (2)$$

where f_{FWD} and f_{REV} return forward sequence and reverse sequence recursively hide the results. If you need to add multiple layers of RNN, then return the entire sequence of $S_{i,j}^{F}$ and $S_{i,j}^{R}$, otherwise return the final state result of $S_{i,j}^{F}$ and $S_{i,j}^{R}$. The complete model diagram is shown in Figure 2. Both the CNN layer and the RNN layer can be added to multiple layers, and the RNN model can extract continuous dependency features from the output of the intermediate CNN layer.

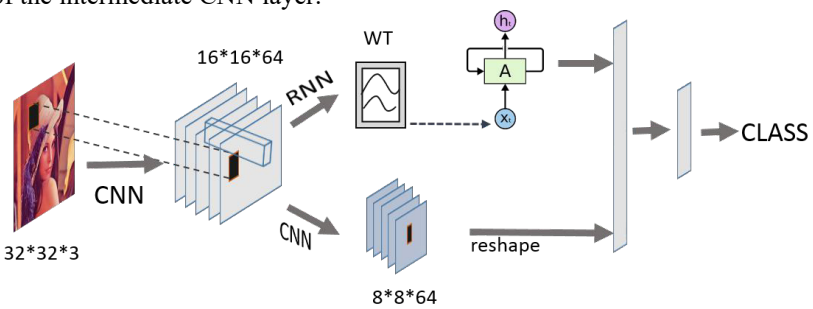

Fig. 2. CNN-RNN Model.

3 Model analysis

The filter is an operation in image processing. The specific operation of filtering is the sum of the product of the image pixels and the filter. Convolution and filtering are largely similar, but in fact it can be seen as a filtering process. The convolution operation can also be implemented using wave processing. The wave of the image is a non-periodic discrete signal in the time domain. Therefore, the Fourier transform can be used to operate the image wave. Convolution calculation is the most important in CNN. If the images in CNN are regarded as non-periodic discrete signals in time domain, the convolution of images is the process of taking waves in different frequency bands. The image can be viewed as a discrete two-dimensional function f(x,y), and the one-dimensional Fourier transform can be extended to two dimensions. The transformation from the spatial domain to the frequency domain is the Fourier transform:

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)e^{-i(ux+uy)}dudv$$

The essence of a convolution kernel is a two-dimensional function, which has a corresponding spectral function. $F(u,v)$ in the formula is a certain kind of spectral function. The convolution of the image is actually taking the characteristics of different frequency bands in the image. The convolution operation in the time domain segment is actually equal to the multiplication operation in the frequency domain segment. However, the Fourier transform has limitations because it cannot characterize the local characteristics of the signal in the time domain, and does not work well for sudden and non-stationary signals. So using a wavelet transform is a better choice:

$$WT(a,\tau)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}f(t)*\psi(\frac{t-\tau}{a})dt$$

where ψ represents a wavelet basis, the difference from e^{ix} in the ordinary Fourier transform is that it is not a simple positive or cosine wave basis function but a wavelet basis function that satisfies certain conditions. Two-dimensional waves have not only local features but also sequence features. Therefore, this paper uses the RNN network to extract

the sequence features of the two-dimensional wave, and also uses the sequence features as an important feature to judge the image classification.

ResNet's skip connection allows it to stack layers of the network deep without vanishing gradient problem [20], because of this hopping connection method, the depth of the ResNet model is not deep. The feature of the middle layer output can be directly connected to the final full-connection network. ResNet is actually a neural network structure composed of multiple shallow CNNs. The CNN-RNN method in this paper cannot achieve the same deep network structure as ResNet, but it can also use the different feature output of the middle layer to connect to the final full-connection layer.

4 Experiments

4.1 DataSets and Model Architectures

The CIFAR-10 dataset contains 60,000 images, and each of which belongs to one of ten categories; airplane, automobile, bird, truck, ship, horse, frog, dog, deer, and cat. The shape of each image is (32,32,3).

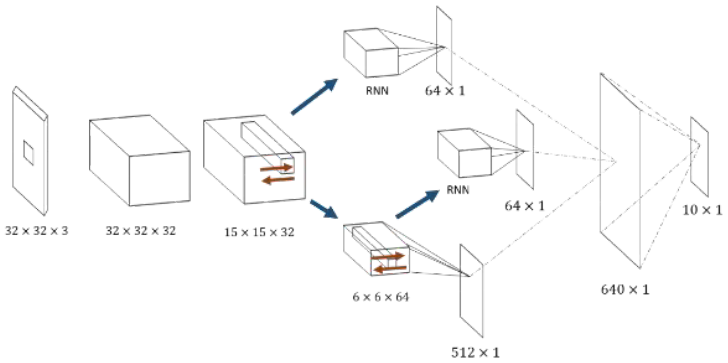

Fig. 3. CNN-RNN model used for CIFAR-10 classification.

ConvLSTM (Convolutional LSTM Network) was proposed by [18] to solve the problem of precipitation nowcasting. ConvLSTM is an improvement over full-length long-term memory networks (FC-LSTM). It is similar to an LSTM layer, but the input transformations and recurrent transformations are both convolutional. ConvLSTM not only has the timing modeling capabilities of LSTM, but also features local features like CNN. All ConvLSTM inputs $x_1,...,x_t$, Cell outputs $C_1,...,C_t$, hidden states $H_1,...,H_t$, and gates i_t, f_t, o_t are all 3D tensors:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * x_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
C_t &= f_t * C_{t-1} + i_t \circ \tanh(W_{xC} * x_t + W_{hC} * H_{t-1} + b_C) \\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
$$

where '*' indicates convolution operation and '°' indicates Hadamard Product. For more details on ConvLSTM reference [18]. In this paper, we use various types of RNN models to compare and find that different types of RNNs have little effect on the results. Figure 3 shows the model structure of the CNN-RNN model trained on the CIFAR-10 data set proposed in this paper.

4.2 Training

The CNN-RNN model is also an End-to-End learning process that does not require additional preprocessing. To train the networks, we choose a wildly used adaptive learning rate algorithm, called Adam [21]. The CNN-RNN model will bring the problem of model overfitting due to the addition of an additional network structure. To reduce over-fitting problem. We will apply the normalization term imposed on the BatchNormalization [22] and RNN weights at the output of the RNN layer.

5 Results and Analysis

In the model training, we found that the standard VGG model will occur vanishing gradient problem, due to the network hierarchy being too deep, and the model hardly converged. Therefore, in this paper, we use a simplify VGG-9 model. We properly remove the two Pool layers to get this article's streamlined VGG-9. The learning rate in the model training process uses a fixed size of 0.001. The ResNet [13] used the variable learning rate in the original paper, so the results in this paper will be different. The training results of the model are shown in Table 1, including the accuracy of each model and the accuracy of the test.

Table 1. Model Test Results.

MODEL	TRAINING ACC	TEST ACC
CNN	0.932	0.772
CNN + RNN	**0.891**	**0.80**
VGGNET-9 [SIMONYAN K, ZISSERMAN A.]	0.943	0.769
VGGNET-9 + RNN	**0.929**	**0.782**
RESNET-20 [HE K, ZHANG X, REN S, ET AL.]	0.965	0.789
RESNET-20+ RNN	**0.961**	**0.812**

Figure 4 and Figure 5 show the accuracy of the original CNN model and CNN-RNN models modified by RNN (including ConvLSTM and RNN), respectively. It can be seen that the improved accuracy of the CNN-RNN model is generally higher than that of the original CNN model. The combination of CNN+ConvLSTM and CNN+IRNN does not differ greatly in accuracy. This shows that the feature of CNN output are not long-term dependencies. And there are only simple dependencies, so using an ordinary IRNN will satisfy the requirement. Therefore, the general RNN can meet the requirements.

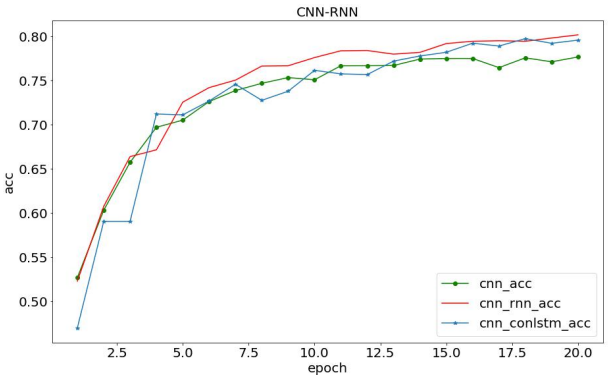

Fig. 4. CNN and CNN-RNN classification.

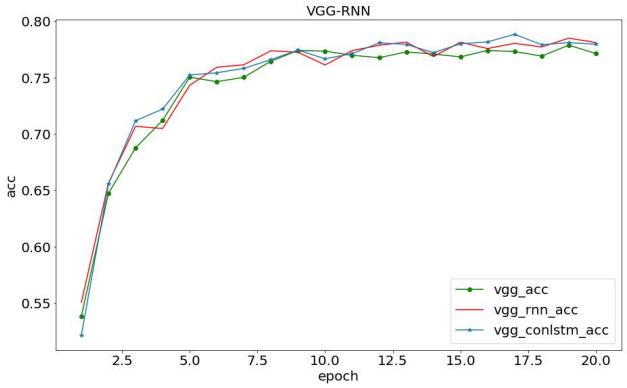

Fig. 5. VGG and CNN-RNN classification.

6 Conclusion and Future Work

In this paper, we propose an improved model CNN-RNN model, which has a certain improvement in the accuracy of image recognition problems. But it will increase the complexity of the model. It will reduce the training speed, and increase risk of overfitting. It need to add regularization items to limit the complexity of the model. In this paper, we simply extract feature from the CNN layer through the RNN network. There are many other extraction methods and combinations that need to be studied.

References

1. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position[J]. Biological Cybernetics, 1980, 36(4):193-202.
2. Lecun Y, Boser B, Denker J S, et al. Backpropagation Applied to Handwritten Zip Code Recognition[J]. Neural Computation, 1989, 1(4):541-551.
3. Cireşan, Dan, Meier U, Schmidhuber, Juergen. Multi-column Deep Neural Networks for Image Classification[J]. Eprint Arxiv, 2012, 157(10):3642-3649.
4. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
5. Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.
6. Zeiler M D, Fergus R. Visualizing and Understanding Convolutional Networks[C]// European Conference on Computer Vision. Springer, Cham, 2014:818-833.
7. Mikolov T A. Statistical Language Models Based on Neural Networks[J]. 2012.
8. Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[J]. 2014, 4:3104-3112.
9. Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014
10. Mao J, Xu W, Yang Y, et al. Explain Images with Multimodal Recurrent Neural Networks[J]. Computer Science, 2014.

11. Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
12. Visin F, Kastner K, Cho K, et al. ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks[J]. Computer Science, 2015, 25(7):2983-2996.
13. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.
14. Krizhevsky A. Learning Multiple Layers of Features from Tiny Images[J]. 2009.
15. Surhone L M, Tennoe M T, Henssonow S F. Long Short Term Memory[J]. Betascript Publishing, 2010.
16. Cho K, Merrienboer B V, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.
17. Le Q V, Jaitly N, Hinton G E. A Simple Way to Initialize Recurrent Networks of Rectified Linear Units[J]. Computer Science, 2015.
18. Shi X, Chen Z, Wang H, et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting[J]. 2015, 9199:802-810.
19. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
20. Hochreiter S. Recurrent Neural Net Learning and Vanishing Gradient[J]. 1998.
21. Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
22. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015:448-456.