Evaluation and Implementation of Convolutional Neural Networks in Image Recognition

Zheng Yi
College of information science and technology, Nanjing Agricultural University, Nanjing, Jiangsu, 210095, China
zhengyi_mark@163.com

Abstract. With the development of computer technology, the applications of machine learning are more and more extensive. One of those applications is image recognition by using Convolutional Neural Networks (CNNs). The significance of this study is to demonstrate the basic structure of CNNs, to understand the further developments of its efficiency and accuracy. Generally, this paper summarizes the composition, evaluation and implementations of Convolutional Neural Networks and gives a brief introduction of the development of this technique.

1. Introduction
The human has a complex visual recognition system. We distinguish and classify objects independently. Actually, our brain is a deep and complex neural network. However, the computational models of brain information processing that have dominated computational neuroscience are largely shallow architectures performing simple computation [1]. Unsurprisingly, complex tasks such as visual object recognition have remained beyond the reach of computational neuroscience. The convolution layer in the Convolutional Neural Networks (CNNs) is trying to mimic the effect of creature’s brain to calculate the input information from vision [2]. From computer perspective, the source or input is the data or image. This technique has already been one of the most common research fields in computer science.

CNNs are used in variety of areas, including image and pattern recognition, speech recognition, natural language processing, and video analysis [3]. There are couple of reasons that why Convolutional Neural Networks are becoming so important, and better than traditional neural networks. In CNNs, the weights of the convolutional layer are used for feature extraction and the weights of full-connected layers are used for classification, which are determined in the training process [4]. Compared with the traditional networks, the improvement of CNNs leads to both savings in memory requirements and computation complexity requirements.

This paper is going to talk about CNNs, including the steps, layers, main advantages and implementations. In section two, we talk about the structure of CNNs in details, for example, the function of different specific layers in the network. In the next section, we propose some evaluation process about CNNs, such as the main advantage and the methods to reduce overfitting. In the following section, we give three common examples which we can find it easily in our daily but are not familiar with their functions to show CNNs are very common and can be used in many different areas. In the last section, we make some conclusions and briefly summarize the development of CNNs.
2. Convolutional Neural Networks
A neural network is a system of interconnected artificial neurons that exchange messages between each other. Convolutional Neural Network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural network with backpropagation, which has already successfully been applied to solving problems of computer vision and machine learning [4]. Besides, CNN is different from regular neural network in terms of transmission mechanism of signals between neurons and layers, and weights sharing. Traditional neural networks transmit signals along the input-output channel in a single direction, without allowing signals to loop back into the previous layers, which is called feed-forward. Meanwhile, CNN also utilizes backpropagation to transmit signals backward for training. On the other hand, in the conventional networks, it always requires all neurons to be fully connected, resulting in overly-complex network structures. Therefore, the cost of complexity grows when the network is trained with large datasets, because of the limitation of storage to save all parameters and of speed to process calculations among all neurons. However, besides the full-connection, CNN has a special structure of partial connections to reduce the computation. To be specific, each neuron gets connected according to the dependency with adjacent layer. And these are what make Convolutional Neural Network show up. More details will be discussed in the later sections.

According to a research conducted by Hubel and Wiesel in 1986[5], they discovered that the receptive field of cat comprised sub-regions which were layered over each other to cover the entire visual field. These layers act as filters that process input images, which are then passed on to subsequent layers, which is proved to be a simpler and more efficient way to carry signals. Then, scientists tried to capture the organization of neurons in the cat’s visual cortex as a form of artificial neural net, establishing the basis of the first CNN.

A typical CNN usually includes three main steps: convolutions, subsampling and full-connection.

2.1. Convolutional layers
The convolution operation aims to extract different features of the input, and there are usually lower-layers and higher-layers for convolution operation. The lower-layers extract low-level features like edges, lines and corners. And the higher-layers extract high-level features, something more abstract. For example, for an input image with the size of N*N, the convolutional layer has H kernels, then the size of each kernel is K*K. H kernels will operate convolution with the input separately, and each kernel produces one output feature, with H kernels. The output will independently produce H features. In calculation, once the top-right corner is reached, the kernel is moved one element in a downward direction, and again the kernel is moved from left to right, one element at a time. This process is repeated until the kernel reaches the bottom-right corner. For instance, when N=32, K=5 and H=6, there are 28 unique positions from left to right and 28 unique positions from top to bottom that the kernel can take. Therefore, each feature in the output will contain 28*28 elements. In this case, after convolutional operation layer, the input image will be extracted 6 features and the size of each feature is 28*28.

2.2. Subsampling/pooling layers
The subsampling/pooling layer aims to reduce the resolution of the features to make the extracted features robust against noise and distortion. In pooling layer, there are two common ways: max pooling and average pooling. In both cases, the input is divided into non-overlapping two-dimensional spaces. For instance, the size of input feature is 28*28 and the size of pooling is 2*2, which means that the output of pooling layer is 14*14. The difference between max pooling and average pooling is how they calculate the value when they do the pooling. Max pooling chooses the max value among 2*2 matrix, while average pooling calculates the average value within these four elements.

2.3. Fully-connected layers
In the final layers of CNNs, the neurons are usually connected fully, while, to avoid too many parameters, the parameters in the convolutional layers are shared within one kernel. Fully-connected
layers aim to sum the weights of the previous layers with features, indicating the precise mix of ingredients to determine a specific target output result. After the process of convolution layer and pooling layer, the network has already successfully extracted the features of the input image. The function of the fully-connected layers is to project the features to classifier layer to mark the label to the input or feedback to improve the parameters.

3. Evaluation methods
Compared with the traditional neural network, there has been some significant developments in the area of Convolutional Neural Networks in the recent decades. This section covers the benefits of using CNNs for image recognition. In this section, we are also going to introducing some methods which can improve the network [6].

3.1. Fewer memory requirements
The main advantage of CNNs is that it decreases the computing amount of parameters through sharing the parameters within one kernel. For example, if the size of an input image is N*N and the number of neurons in hidden layer is H. In the case of a other networks, the number of parameters that need to be stored and calculated is N*N*H. However, for CNN, in the convolution layer to extract feature, each neuron does not need to connect the input fully. They just connect to part of the input, through extracting the features of a part area to sense the features of the input. Hence, the number of the weight can be decreased. However, the number of parameters is still huge so that the computer cannot deal with it because of the limitation of speed and storage. Therefore, CNNs suggest that all neurons in one kernel should share the same parameters, so the number of parameters could only be related with the size of the kernel. However, according to the same parameters in one kernel, each kernel could only extract only one feature from the input. Therefore, in convolution layer, usually more than one kernels are needed.

3.2. Reducing overfitting
Current approaches to improve the performance and reduce overfitting include collecting more training data, learning more powerful models and using advanced mechanisms. In augmentation of data, we can transform images to be produced from the original images with very little computation, so not only the transformed images do not need to be stored on disk, but the amount of training data can be increased. Besides, combining with the predictions of several different models, the test errors can be reduced. However, the cost of training these several models are too expensive to operate. Therefore, to evaluate the output of the model and to avoid too much cost, a technique called “drop out” was introduced. This technique consists of setting the output of each hidden neuron to zero with probability 0.5. The neurons which are set zero in this way do not contribute to the forward pass and do not participate in back-propagation. So for each input, the network samples a different architecture, while all architectures share weights. With the transformation of images to augment dataset and “drop out” technique to change the architecture, the overfitting can be reduced and the output can be evaluated.

4. Implementation
Convolutional Neural Networks are applied in many image recognition problems. We present three examples here: autonomous vehicle control systems, vehicle license plate recognition and natural language processing.

4.1. Autonomous vehicle control systems
When Convolutional Neural Network is used in a car, the network analyzes the input image and finds areas that have a specific feature like another car or pedestrian. Image identification in autonomous vehicles is not as simple as facial or handwriting recognition because vehicles need to process a full 360-degree dynamic environment. Convolutional Neural Networks are the most promising method for
classifying complex scenes because they closely mimic the structure and classification abilities of the human brain. Besides, CNN could also be used for estimating the distance between an obstacle and the vehicle which is an important consideration in autonomous driving as it ensures safety of the passengers and of other vehicles. Scientists point out that the trainable, multi-layered structure of CNNs is what sets them apart from other neural networks. The capabilities of CNNs to distinguish both the presence and depth of obstacles makes them promising backbones for autonomous transportation. As the more research we relate autonomous vehicles with CNNs, the closer we are to introducing these vehicles as a main form of transportation [7].

4.2. Vehicle license plate recognition
Vehicle license plate recognition is usually done with a sliding window approach. It can have limited performance on datasets with characters that are of variable width. Besides, a sliding window approach also requires training data in the form of pre-segmented characters, which can be more difficult to obtain. With the use of convolutional neural network, we can address the problem of sliding window approaches being unable to perform end-to-end training of the entire model on labelled, full license plate images.

There are experiments showing that the proposed CNN framework for vehicle license plate recognition combines the advantages of feature learning and joint image/label embedding by implementing CNN to model the feature and label sequence. However, vehicle license plate recognition is still challenging due to the majority of our dataset being reasonably centered [8] [9].

4.3. Natural language processing
Convolutional Neural Networks have achieved remarkable results in computer vision and speech recognition. In recent years, within natural language processing, much of the work with deep learning methods has involved learning word vector representations through neural language models and performing composition over the learned word vectors for classification. CNNs utilize layers with convolving filters that are applied to local features. Nowadays, CNNs have subsequently been shown to be effective for language processing [10] [11] [12].

5. Conclusion
This paper briefly introduces one of the most popular algorithms in image recognition. We introduced convolutional neural networks by discussing the main steps and layers within the network. Then, we proposed some evaluations of Convolutional Neural Networks, like the saving of computer storage by sharing parameters, reducing overfitting by augmenting the dataset, and using “drop out” technique to transform the architecture of the network. In the following part, we introduce three main implementation of convolutional neural networks, including image recognition and text processing. Though CNNs were used at image area at the very beginning, nowadays, there are many areas that can take CNNs into consideration. Furthermore, Convolutional Neural Networks and Deep Learning in general are still active areas of research [13]. Nonetheless, even we know that CNN imitates the neurons’ work in our brain, the reasons for the good performance of deep neural network are still not answered fully.

References
[1] Matthew D. Zeiler, Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. 8689, 818-833.
[2] Keonhee Lee, Dong-Chul Park. 2015. Image Classification using Fast Learning Convolutional Neural Networks. Vol.113(Art, Culture, Graphics, Broadcasting and Digital Contents 2015), pp.50-55.
[3] Koushik, J. 2016. Understanding convolutional neural networks.
[4] Samer Hijazi, Rishi Kumar, Chris Rowen. Using convolutional neural networks for image recognition. http://ip.cadence.com/Hubel-Wiesel.
[5] Hubel, D. H., & Wiesel, T. N. (1968). RECEPTIVE FIELDS AND FUNCTIONAL ARCHITECTURE OF MONKEY STRIATE CORTEX. Journal of Physiology-London, 195(1), 215-.

[6] Hoyer, P. O., & Hyvärinen, A. 2000. Independent component analysis applied to feature extraction from colour and stereo images. Network: computation in neural systems, 11(3), 191-210.

[7] Erin Welling, Maddie Oppelt. Convolutional neural networks in autonomous vehicle control systems.

[8] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, Yann LeCun. 2010. What is the Best Multi-Stage Architecture for Object Recognition. IEEE, International Conference on Computer Vision (Vol.30, pp.2146 - 2153). IEEE.

[9] Dan Claudiu Ciresan, Ueli Meier, Luca Maria Gambardella, Jurgen Schmidhuber. 2012. Deep Big Simple Neural Nets Excel on Hand-written Digit Recognition. Corr, 22(12), 3207-3320.

[10] Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[11] Rie Johnson, Tong Zhang. 2015. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. Eprint Arxiv.

[12] Xiang Zhang, Yann LeCun. 2016. Text Understanding from Scratch. arXiv:1502.01710 [cs.LG].

[13] Alex Krizhevsky, Ilya Sutskever, Geoffrey E.Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012).