Feature Extraction and Image Recognition with Convolutional Neural Networks

IOP Conf. Series: Journal of Physics: Conf. Series **1087** (2018) 062032 doi:10.1088/1742-6596/1087/6/062032

Feature Extraction and Image Recognition with Convolutional Neural Networks

Yu Han LIU

Glasgow College, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

liuyhnnn@126.com

Abstract. The human has a very complex perception system, including vision, auditory, olfactory, touch, and gustation. This paper will introduce the recent studies about providing a technical solution for image recognition, by applying a algorithm called Convolutional Neural Network (CNN) which is inspired by animal visual system. Convolution serves as a perfect realization of an optic nerve cell which merely responds to its receptive field and it performs well in image feature extraction. Being highly-hierarchical networks, CNN is structured with a series of different functional layers. The function blocks are separated and described clearly by each layer in this paper. Additionally, the recognition process and result of a pioneering CNN on MNIST database are presented.

1. Introduction

Along with the development of Machine Learning, especially Deep Learning, people is caring more and more about the practical uses of this technology. Image recognition is one of the most common study fields in the recent years. The basic problem of it is to determine whether in an image does an object exist, to describe the object ‘s location and to identify a specific object category. Image recognition started in the late 1960s [1], as part of pattern recognition, it began to be an independent field in the next decade and most of its basic concepts were set up with early algorithms in 1970s including the ideas of feature extraction [2]. The coming years has witnessed the great development and progress of image recognition, among which Convolutional Neural Networks (CNNs) show up as a modern pioneer.

A Convolutional Neural Network (CNN) is a feed-forward artificial neural network inspired by animal visual cortexes, it is designed for visual imagery. CNNs have been applied in many practical fields, such as pattern recognition, vocal recognition, natural language processing, and video analysis [3]. In CNNs, the most significance features are weight sharing and hierarchical connections with automatic self-training. Semi-connected layers and full-connected layers play different roles and provide reasonable environment for training process feed-forward as well as backward propagation of errors. The backward propagation process is usually called Back-Propagation [4].

This paper is going to talk about the basic understanding of CNN, including its structure, different layers, and working process. Also, an example of handwriting recognition with CNN is illustrated in the following part. Finally, there is a short conclusion.

2. Convolutional Neural Networks

2.1. Convolution

Convolution is an integral transformation operation on a function through a specific operator. It reforms the original function into a new representation of the similarity of two functions in a certain window. It is widely used in image processing for feature detecting without cancelling the spatial associations between pixels. It searches for a certain feature with the help of corresponding operator in a much larger pixel set.

Mathematically, a single convolution operation is constructed by first reversing the operator, namely the convolution kernel, both by row and column. Occasionally, this step is skipped due to kernels of central symmetry. Secondly, the convolution kernel is shifted by a fixed stride over the whole original image (the kernel should not exceed the image range). And at each placement, the resulting value would be a combination of the original elements weighted by the reversed kernel respectively.

Detailed in image processing, convolution is an efficient way of feature extraction, skilled in reducing data dimension and producing a less redundant data set, also called as a feature map. Each kernel works as a feature identifier, filtering out where the feature exists in the original image. Eventually it produces a map whose altitude reveals how these features distribute.

Since directly analyzed original data ask for a great amount of preprocessing such as image segmentation, and can hardly deconstruct the symbolic information of the images, it is necessary that images are processed with some certain general feature describing methods like convolution. It aims to extract target features automatically and to make them more 'visible' for computers. Operating convolutions not only fully describe the characteristics of the input, but also does it lower the demand for manual operations.

While lower-order convolution kernels are usually in smaller size comparing to the input image, extracted features focus more on local perception. However, higher-order convolutions enable expansions in the overall receptive field which gradually converts local features to global features which is also the case how human gazes at an object and recognizes it.

Additionally, kernel parameters are kept constant over the whole recognition process. This leads to the weight sharing property of each receptive neuron, resulting in a great cut in calculations. Moreover, pooling further samples the features and reduces calculating operations for computers.

2.2. Neural Networks

Neural Networks, or Artificial Neural Networks, are computing models derived from Biological Neural Networks which consists of a large number of dense neurons. While Biological Neural Networks bring consciousness to living beings, Neural Networks manage to learn from given examples rather than operate under specific commands. Neural Networks' ultimate goal is marked as reproducing the achievements of biological neural systems, learning from the way in which human recognize the world, and thus making good and proper use of this network structure to help solve complex problems like object classification or image recognition.

Neural Networks are composed of massively connected artificial neurons within hierarchical network structures. Artificial neurons here take the chair of input signal handling and processed signal propagating. Each of the artificial neuron has a weight that modifies the input signal. This weight of the neuron is variable during the training process and the change of weight can be taken as a gradually-learning procedure of the artificial neuron to fit the training set. A bias is introduced to the weighted sum of the inputs to further adjust the signal. At the end, the signal would pass through an activation function to produce the final output of a neuron.

What a single neuron does is to convert all its connected input into a scalar which best describes the inputs' state after a weighted sum and an activation. Neural Networks do the same work in a paralleled and complex structure of neurons to transform the inputs into a class representing vector.

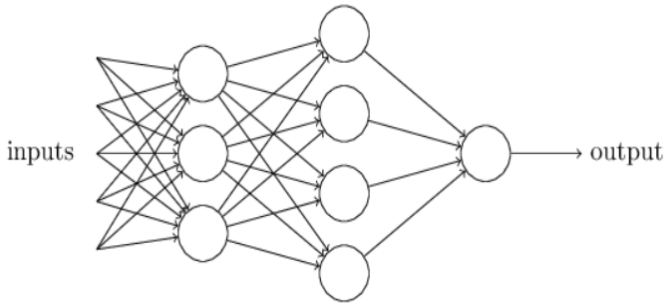

Figure 1 A simple multi-layer neural network structure

Figure 1 is a neural network structure where each circle represents a neuron. This structure consists of three different type of layers, an input layer, hidden layers and an output layer. The input layer receives the input signals and feeds them to the hidden layers that lie between the input and the output layer. Through the hidden layers, signals are processed and activated under certain parameters and passed on. Finally, output signals are generalized in the output layer.

Convolutional Neural Network is a feed-forward artificial neural network. Conceptually inspired by animal visual cortexes, Convolutional Neural Network is born to be an image processing network, or rather, a low-dimension data processing network (usually one or two-dimension data). A basic neuron of a Convolutional Neural Network does not response to the input in whole. Instead, it collects the activations of the non-overlapping segmentations of the input and produces a feature map.

CNNs produce satisfactory results of recognition for practical use, however, they also give birth to a black box of the feature extraction process as the layers pile up. High-order features tend to be somewhat ambiguous.

3. Recognition Process

CNNs are highly layered structural neural networks, most of which have the same basic function layers including convolution layers, pooling layers and a classification layer. LeNet-5 was proposed as the first modern Convolutional Neural Network of practical use [5]. Basically, CNNs differ from each other by how these fundamental layers are installed and packaged and also the method of training the network.

Here I give a thorough recognition process of a Convolutional Neural Network with the help of the structure in the Figure 2.

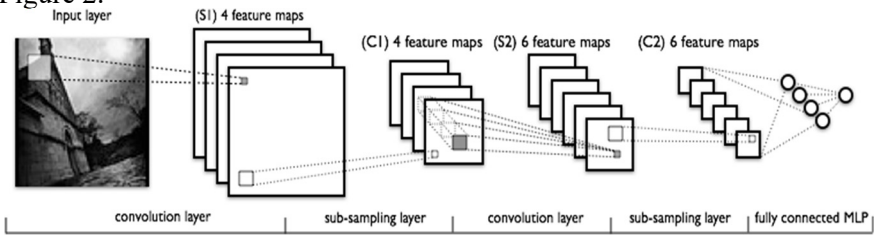

Figure. 2 A Convolutional Neural Network structure [6]

First, the input images are preprocessed to a standard normalization. Then, the data flow into couples of convolutional layers with pooling layers where feature extraction and redundancy reduction occur. The simple features gradually gather in an efficient way. Afterwards, all the features are combined partially and the resulting features each count for a part of configuration of the labelled class. Eventually, these top features are sent into the fully-connected layer and an estimate of the classification is provided by this layer.

3.1. Preprocessing Layer

Beginning with the input layer, a set of input images are to be imported. Before then, several preprocessing operations are required including sizing and normalization. However, CNNs demand

much less preprocessing operations than other neural networks. Otherwise, a simple preprocessing layer is necessary for eliminating unconcerned differences.

3.2. Convolutional Layer

A convolutional layer condenses the input by extracting features of interest from it and produces feature maps in response to different feature detectors as indicated above. In the first convolutional layer, simple feature as edges are filtered by the neurons. The neurons learn to gather the information to gain a bigger picture of the image in the following convolutional layers, hence a high-order feature detecting.

While each convolution kernel is capable of feature extracting all over the input plane, neurons are allocated to handle different parts of the input image plane to produce the feature maps with the same size of receptive field. This scheme of convolutional weight sharing originates from Time-Delay Neural Networks (TDNNs) [7] and denotes to a stack of feature map. The number of the neuron type leads to the depth of the stack. Each type of neurons shares the same weight & bias vector and produces a slice of the stack.

3.2.1. Convolution

Each convolutional layer is defined with several parameters including the input size, kernel size, depth of the map stack, zero-padding and stride.

The output size can be calculated by:

$$\mathrm{M}_x = \frac{I_x - K_x}{S_x} \quad , \quad \mathrm{M}_y = \frac{I_y - K_y}{S_y}$$

Where (M_x, M_y), (I_x, I_y), (K_x, K_y) indicate the map size, input size, kernel size separately, S_x, S_y indicate the stride in row & column.

3.2.2. Activation

After the weighted sum and a bias, there should still be an activation. Besides the pure perceptrons, a non-linear activation function is needed to break the simplex linear combination of the input and make it possible for a neural network to become a universal approximator of continuous functions in a Euclidean space. LeCun mentioned a sigmoid function to squash the output of a pooling layer [5]. Yet later Jarrett et al. brings Rectified Linear Units(ReLUs) into CNNs to improve the performance [8]. Soon after, Xavier Glorot et al. states that the outstanding performance of ReLU should be attributed to its hard-non-linearity, non-differentiability at zero and its sparse feature [9]. Eventually, ReLUs are widely accepted for activating convolutional outputs. ReLU is given by:

$$\mathrm{f(x)} = \max(0, \mathrm{x})$$

3.3 Pooling Layer

Pooling, or subsampling, includes several types of operation as general-pooling, overlapping-pooling, etc. It usually mediates between several convolutional layers. Max-pooling and average-pooling are the most frequently used pooling strategies in CNNs. Pooling brings much benefits to CNNs. In particular, pooling aims at preventing overfitting by concentrating local data with a pooling window thus reducing data dimensionality. Data dimensionality reduction also help cut down calculations. Apart from that, pooling brings about invariance including translation, rotation and scale because a few dislocations or scalings make no differences after proper pooling.

Non-overlapping max-pooling simply excites the maximum outcome of each pooling region $(\mathrm{P}_x, \mathrm{P}_y)$over the input image and compresses it by P_x and P_y respectively on width and height.

3.4. Classification Layer

The classification layer is the top layer of a network that collects the final convoluted feature and returns a column vector where each row points towards a class. More precisely, elements of the output vector each represent the probability estimation of each class and the sum of the elements is one.

While convolutions combined with pooling map raw image data into a feature space, classification layer takes an effect on sample space projection which provides an obvious exhibition of classification. Original CNNs end up with a fully-connected layer in most cases. Actually, the fully-connected classification layer is a legacy that extends from the idea of 'feature extraction & classification' in the Artificial Intelligence. Full connections of inputs with neurons are conducted to combine and reweight all the high-order features to achieve the spatial transformation.

4. Handwriting Recognition

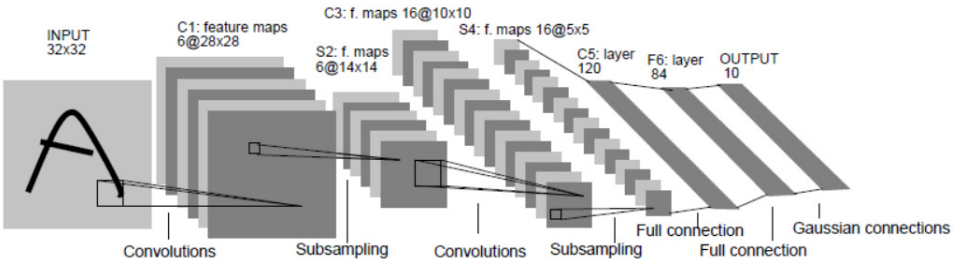

Figure 3 Network Architecture of classic CNN, LeNet-5 [5]

Figure 3 shows one of the most classical structures of CNN, LeNet-5, by LeCun et al. It has been widely used for handwritten digits. It was then limited by the computing power of the time. However, current application of Graphics Processing Unit helps make better use of this neural network structure.

LeNet-5 consists of seven layers apart from the input layer, layers are labelled by Cx, Sx, Fx which respectively stands for convolutional layer, subsampling layer and fully-connected layer and x indicates the layer sequence number. And in this network, convolutions all share a 5x5 kernel size and a stride of 1. So, each convolution kernel should be parameterized with 25xD (kernel depth) trainable kernel weights. Besides a trainable bias, each convoluted feature map demands for (25D+1) trainable parameters. Subsampling layer average its input in an 2x2 area with a stride of 2. The average is multiplied by a trainable parameter and added with a bias. A total of two trainable parameters are required for a subsampling kernel.

First, the input layer is provided with normalized 32x32 pixel images whose digits are centered and kept inside of an area of 20x20 pixels in the central part of the whole input plane. This characteristic guarantees that potential features can be presented in the center of the highest-order feature receptive-field.

C1 consists of 6 plane convolution kernels with, leading to 6x (25+1) =156 trainable parameters, results in 6 feature maps sized 28x28. The total connections in this layer is given by (25x1+1) x(28x28) x6=122304.

S2 consists of 6 subsampling kernels according to its input. This leads to 6x2=12 trainable coefficients and a total connection of (2x2+1) x(14x14) x6=5880.

C3 consists of 16 mixed-depth convolution kernels instead of uniform kernel size which is shown in the Figure 4.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

Figure 4 Layer-C3 convolution kernel depth arrangement of LeNet-5 [5]

Thus, in total, there are 6x(25x3+1) +9x(25x4+1) +1x(25x6+1) =1,516 trainable parameters and 1,516x10x10=151,600 connections.

S4 consists of 16 subsampling kernels, ergo 16x2=32 trainable parameters are required and (2x2+1) x(5x5) x16=2,000 connections are built.

C5 consists of 120 convolution kernels, it happens to fully connect itself with the 5x5 outputs of S4, but it still should be labeled as a convolutional layer rather than a fully-connected layer because if the input image is of a size larger than 32x32, C5 would not give birth to 1x1 feature map. And there are (25x16+1) x120=48,120 trainable parameters and a same number of connections.

F6 connects the output features fully with its input and hence giving (120+1) x84=10,164 trainable parameters and connections.

Output layer is made up of Euclidean Radial Basis Function (RBF). Each unit is connected to all its 84 inputs. The output comes as:

$$y_i = \sum_j (x_j - w_{ij})^2$$

where i & j indicates the unit number and input number respectively.

The RBF unit is responsible for similarity measurements of its inputs to the digit model by computing the Euclidean distance between them. The better the input fit with the standard model, the closer the distance approaches to 0.

Eventually, a 10x1 classification vector in the sample space is produced whose elements 0 indicates its estimated class.

Trained with pure MNIST [10], the test error rate converges to 0. 95% and stabilizes after about 10 passes of the training set. For the training set, LeNet-5 achieves an error rate of 0. 35% after 19 passes [5].

5. Conclusion

The thorough recognition process of the CNN is presented in this paper. With its local connection and weight sharing characteristics, CNN manages to scan images and extract objects' features with much lower compute cost. Furthermore, pooling enhances the robustness of the network upon spatial variances. However, features extracted tend to be abstract which are hard to explain, CNN finishes its job as an end-to-end model during an image recognition process. For the future understanding of CNNs, the most coming challenge is to provide a more stable calculating environment and increase the computing speed. Both hardware and software improvement will help with this issue.

Although, it is still difficult for people to fully understand what has been achieved inside the 'black box' of the neural network, it cannot be denied that CNN gives an excellent performance on image recognition.

References

[1] Jain, A. K., & Li, S. Z. (2011). Handbook of face recognition. New York: springer. preface.
[2] R.Szeliski. (2010). Computer Vision: Algorithms and Applications. Springer Science & Business Media. pp. 11. ISBN 978-1-84882-935-0.
[3] Koushik, J. (2016). Understanding Convolutional Neural Networks. arXiv preprint arXiv:1605.09081.
[4] R. Hecht-Nielsen. (1989). "Theory of the backpropagation neural network," in International Joint Conference on Neural Networks, pp. 593– 605.
[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. (1998). "Gradient based learning applied to document recognition". Proceedings of the IEEE, 86(11):2278–2324.
[6] Theano Development Team, Deep Learning Tutorials, Convolutional Neural Networks (LeNet), The Full Model: LeNet.
[7] A. Waibel, T. hanazawa, G. Hinton, K. Shikano and K. Lang. (1989). "Phoneme Recognition Using Time-Delay Neural Networks" IEEE Transactions on Acoustics, Speech and Signal Processing,vol. 37:328-339.

[8] K Jarrett, K Kavukcuoglu, M Ranzato, Y Lecun. (2010). “What is the best multi-stage architecture for object recognition?”. IEEE International Conference on Computer Vision , 30 (2) :2146 – 2153.
[9] X Glorot, A Bordes, Y Bengio. (2011). “Deep Sparse Rectifier Neural Networks”, Proceedings of the Fourteenth International Conference on Artificial Intelligencs & Statistics, (AISTATS). 130, 297.
[10] Y. LeCun, Corinna Cortes, J.C. Burges. (2013). "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges".