Neurocomputing 000 (2017) 1–7

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Improving deep neural networks with multi-layer maxout networks and a novel initialization method

Weichen Sun[a,*], Fei Su[a,b], Leiquan Wang[c]

[a] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, 100876
[b] Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, Beijing, China, 100876
[c] School of Computer and Communication Engineering, China University of Petroleum (Huadong), Qingdao, China, 266580

ARTICLE INFO

Article history:
Received 14 October 2016
Revised 26 March 2017
Accepted 13 May 2017
Available online xxx

Keywords:
Deep learning
Convolutional neural networks
Activation function
Image classification
Initialization

ABSTRACT

For the purpose of enhancing the discriminability of convolutional neural networks (CNNs) and facilitating the optimization, we investigate the activation function for a neural network and the corresponding initialization method in this paper. Firstly, a trainable activation function with a multi-layer structure (named "Multi-layer Maxout Network", MMN) is proposed. MMN is a multi-layer structured maxout, inheriting advantages of both a non-saturated activation function and a trainable activation function approximator. Secondly, we derive a robust initialization method specifically for the MMN activation with a theoretical proof, which works for the maxout activation as well. Our novel initialization method could reduce internal covariate shift when signals are propagated through layers and solve the so called "exploding/vanishing gradient" problem, which leads a more efficient training procedure of deep neural networks. Experimental results show that our proposed model yields better performance on three image classification benchmark datasets (CIFAR-10, CIFAR-100 and ImageNet) than quite a few state-of-the-art methods and our novel initialization method improves performance further. Furthermore, the influence of MMN in different hidden layers is analyzed, and a trade-off scheme between the accuracy and computing resources is given.

1. Introduction

A resurgence of neural networks (NN), also called deep learning (DL), has drawn much attention since 2006, mainly due to the significant performance gain in visual recognition tasks, such as recognizing objects [1–3], faces [4,5] and hand-written digits [6], especially in the presence of a large amount of training data. DL is a branch of machine learning based on a set of algorithms that attempt to learn representations of data using deep architectures [7–11].

The convolutional neural network (CNN) [12] is one of the most popular DL models applied in computer vision and its representational power increases dramatically with the depth [2,13]. However, when training a deep CNN employing the stochastic gradient decrease (SGD) algorithm, the error gradient weakens as it moves from the back of the network to the front. Consequently, higher layers could be trained well while lower layers often get stuck during training, almost stay the same as what they were initialized. Furthermore, the performance of our deep CNN may be not superior than that of a shallow network. Such arduousness in the training procedure is mainly due to the vanishing or exploding gradients problems [14,15] with the depth increasing. Several techniques, such as pre-training [16], data augmentation [1–3], regularization techniques [6,17,18], novel non-linear activation functions [1,19–25] and sophisticated initialization methods [14,25], have been proposed to solve difficulties of training deep neural networks.

Among recent advances of DL above, various activation functions and sophisticated initialization methods are two most effective ones that solve the vanishing or exploding gradients problems. On the one hand, whether the error gradient could propagate back to the lower layer depends on the activation function. The widely used strategy is to replace the saturated activation function (e.g. sigmoid or tanh) with the rectified linear unit (ReLU) [19], which is a non-saturated activation function and the error gradient could be passed back into the lower network. ReLU is a piecewise linear function which projects negative inputs to zeros, leading to a desirable property that activations of neural nodes are sparse. However, ReLU assigns a zero slope to the negative part. Hence back-propagation will be blocked when the unit is not activated and a vanishing error back flow has almost no effect on weight updates of lower layers. Authors in [21] presented a novel activation

* Corresponding author.
E-mail addresses: weichern.sun@gmail.com, weichen.sun@hotmail.com (W. Sun).

http://dx.doi.org/10.1016/j.neucom.2017.05.103
0925-2312/

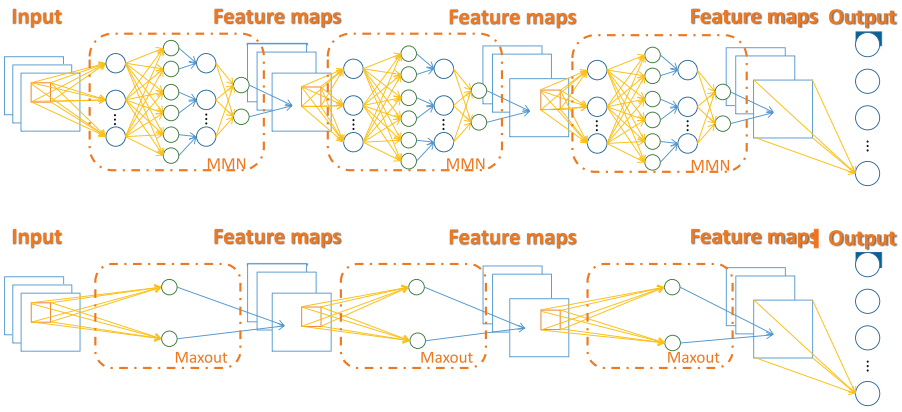

Fig. 1. Comparison of our model and a convolution neural network with maxout units.

function called maxout which assigns a non-zero slope to both the positive part and the negative part. Maxout facilitates the optimization procedure by partly preventing hidden units from transiting to inactive. Although the maxout unit is a trainable activation function, it is not a arbitrary function approximator. On the other hand, a suitable initialization method plays an crucial role on training a deep neural networks. Otherwise, a unfavorable initialization may lead to a long learning stage and a poor solution. As we all know, deep learning is non-convex optimization and the solution to a non-convex optimization algorithm depends on the initial values of parameters. Approaches such as the unsupervised pre-training [26], the transfer learning [27] and various initialization methods [14,25] are proposed for the sake of superior initial values for the optimization procedure. As pointed out in Glorot et al. [14], a properly scaled uniform distribution for initialization could solve the vanishing or exploding gradients problems, which is called "Xavier" initialization method in [28]. Nevertheless, the theoretical derivation is based on the assumption that the activations are linear, which is obviously not valid in a deep neural network. Afterward, He et al. [25] derived a robust initialization method particularly considering the rectifier nonlinearities. By rescaling the distribution of each layer's inputs at the initial state, lower layers in a extremely deep architecture could be efficiently trained.

Inspired by replacing a single conventional convolutional layer by a more complicated micro neural network, called "Network In Network (NIN)" [23], we proposed a joint supervised training framework that a micro deep network called MMN is trained as the nonlinear activation function together with parameters of each filter in convolutional layers. The primary advantage is that a micro deep network with increasingly complex structures could compactly represent a larger set of activation functions more efficiently than a shallow network. In other words, there are activation functions which could be compactly represented by a k-layer network, but could not be represented by a (k-1)-layer network unless it has an exponentially larger number of hidden units. In our proposed model, the activation function is replaced with the MMN, which is a more general nonlinear activation function approximator than the maxout unit. Here, weights of MMNs are shared as those of the maxout unit. By sliding convolutional filters with the MMN activation over the local patch, feature maps are obtained and then fed into higher layers. We stacked several convolutional layers with such MMNs and pooling layers to compose a deep CNN, which is trained by supervised back-propagation algorithm [29]. Since each hidden unit in MMN is a maxout unit, MMN can be treated as a multi-layer generalization of the maxout unit, which preserves the properties of the maxout unit while improving the capability in modelling various distributions of latent concepts. Furthermore, because existing initialization methods [14,25] do not make sense for MMN and maxout, we proposed a novel initialization method specialized for our MMN (valid for the maxout activation function as well) and provided a theoretical proof. In addition, the influence of MMN in different hidden layers is considered to show a trade-off scheme between the accuracy and the computational cost. The comparison of our model and a convolutional neural network with maxout units is shown in Fig. 1. To verify the performance of our proposed model and the novel initialization method, experiments are conducted on CIFAR-10, CIFAR-100 [30] and ImageNet [31] datasets. Experimental results show better performance of our model with dropout than those of current state-of-the-art methods.

2. Related work

2.1. Rectified linear unit

Rectified Linear Unit(ReLU) is first applied in Restricted Boltzmann Machines (RBM) [19]. It is defined as

$$y_i = \begin{cases} x_i, & if\ x_i \geq 0 \\ 0, & if\ x_i < 0 \end{cases} \tag{1}$$

where x_i is the input to a neuron and y_i is the output. ReLU has a desirable property that activations of neural nodes are sparse.

2.2. Maxout

The maxout is a more general version of ReLU, which takes the max operation on $k(k = 2)$ trainable linear functions. Given an input $x \in R^d$ (x is either the raw input vector or the state vector of a hidden layer), the output of a maxout unit is formulized as follows:

$$h_i(x) = \max_{j \in [1,k]} z_{ij} \tag{2}$$

Here, $z_{ij} = x^T W_{\cdots ij} + b_{ij}$, $W \in R^{d \times m \times k}$ and $b \in R^{m \times k}$ are trainable parameters. k is the number of linear sub-hidden units where taking the maximum Fig. 2 across. In a CNN, the activation of a maxout unit equals the maximum over the k feature maps. Though the maxout unit is similar to the commonly used spatial max-pooling in CNNs, it takes the maximum value over a subspace of k trainable linear transformations over the same input, whereas the spatial max-pooling is corresponding to k different input.

2.3. "Xavier" Initialization

During the forward propagation, there are two kinds of unfavorable conditions. On one hand, if the weights in a network start too

Please cite this article as: W. Sun et al., Improving deep neural networks with multi-layer maxout networks and a novel initialization method, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.05.103