Expert Systems With Applications 63 (2016) 1–7

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Deep learning with adaptive learning rate using laplacian score

B. Chandra*, Rajesh K. Sharma

Indian Institute of Technology Delhi, New Delhi, India

ARTICLE INFO

Article history:
Received 28 September 2015
Revised 12 May 2016
Accepted 13 May 2016
Available online 16 May 2016

Keywords:
Adaptive learning rate
Deep learning
Gradient descent
Laplacian score

ABSTRACT

An attempt has been made to improve the performance of Deep Learning with Multilayer Perceptron (MLP). Tuning the learning rate or finding an optimum learning rate in MLP is a major challenge. Depending on the value of the learning rate, classification accuracy can vary drastically. This issue has been taken as a challenge in this paper. In this paper, a new approach has been proposed to combine adaptive learning rate in conjunction with the concept of Laplacian score for varying the weights. Learning rate is taken as a function of parameter which itself is updated on the basis of error gradient by forming mini-batches. Laplacian score of the neuron is further used for updating the incoming weights. This removes the bottleneck involved in finding the optimum value for the learning rate in Deep Learning by using MLP. It is observed on benchmark datasets that this approach leads to increase in classification accuracy as compared to the existing benchmark levels achieved by the well known methods of deep learning.

1. Introduction

Deep learning has achieved the state-of-the-art results in the field of Machine Learning such as computer vision and speech recognition. The field of Deep learning started with the development of Deep Belief Network (Hinton, Osindero, & Teh, 2006). Deep Belief Network uses unsupervised pre-training and supervised fine tuning. Based on the same concept of unsupervised pre-training, several algorithms have been developed such as Stacked Denoising autoencoder (Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010) and Contractive auto-encoders (Rifai, Vincent, Muller, Glorot, & Bengio, 2011).

With the development of Rectified Linear Network (Glorot, Bordes, & Bengio, 2011), it has been possible to get superior performance in deep learning without unsupervised pre training. Deep learning is prone to over-fitting since it deals with training a large number of parameters. In order to prevent Deep MLP from over-fitting, several regularization conditions have been proposed such as adding noise to the input (Vincent et al., 2010), using probabilistic activation functions (Hinton et al., 2006), dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) and making the activation values sparse (Glorot et al., 2011). In dropout, the input and hidden neurons are masked with certain probability during each iteration. Dropout has been shown to improve the performance of different types of neural networks (Dahl, Sainath & Hinton, 2013; Goodfellow, Warde-Farley, Mirza, Courville, & Bengio, 2013; Hinton et al., 2012).

Max-out Network has been developed (Goodfellow et al., 2013) for Deep learning without unsupervised pre-training. Both Max-out Network and Rectified Linear Network use dropout to prevent overtraining. Though there is a marked improvement in the performance by using Rectified Linear Network and Max-out Network, there is a limitation due to the involvement of hyper-parameters.

When deep learning is used for classifying images for intelligent systems, an appropriate learning rate has to be chosen for each of the layers which is a difficult task. Moreover, intelligent systems require high precision in terms of accuracy.

Multilayer Perceptron (MLP) has been applied successfully for the detection of medical fraud. Neural Network has been used for intrusion detection in which an expert system module has been included which analyses the output of neural network and relates it for intrusion detection. However, there is another important area where credit card frauds have been detected by using neural network (Patidar & Sharma, 2011; Raj & Portia, 2011, March). Neural Network has also been used to control an autonomous surface vehicle where the vehicle dynamics are unknown and suffer from uncertainties (Pan, Lai, Yang, & Wu, 2013).

MLP has been used to predict the utility of online reviews (Lee & Choeh, 2014). Details of the product and characteristics of the tests contained in the reviews are used as input to MLP in order to make prediction about usefulness of the reviews. Another application in expert systems includes forecasting of movie revenues during the pre-production (Ghiassi, Lio, & Moon, 2015). A model has been developed by using MLP to forecast the total movie revenue during the pre-production phase itself. MLP uses pre-release

* Corresponding author at: Room MZ 149, Indian Institute of Technology Delhi, India.
E-mail addresses: bchandra104@yahoo.co.in (B. Chandra), justrks@gmail.com (R.K. Sharma).

http://dx.doi.org/10.1016/j.eswa.2016.05.022
0957-4174/

advertisement expenditure, runtime, seasonality of tentative release date and production budget to make accurate prediction about the movie revenue.

In business domain, Neural Network approach has been successfully applied to predict the outcome of price negotiation (Moosmayer, Chong, Liu, & Schuppar, 2013). The price negotiation process of companies has been analyzed by predicting the final negotiated price between the seller and buyer by using Neural Network where sellers' reservation price, target price and initial offer price are used to make the prediction. Another application of Neural network in business domain is for the prediction of bankruptcy (Iturriaga & Sanz, 2015) where Neural network is used to make three year ahead prediction of bankruptcy risk on the basis of information about bank's loans portfolio, default rate, capital composition, liquidity etc.

Neural Network has also been used to predict the direction of stock price changes on the basis of technical analysis, fundamental analysis and time series analysis (Dase & Pawar, 2010; de Oliveira, Nobre, & Zarate, 2013; Hadavandi, Shavandi, & Ghanbari, 2010; Kara, Boyacioglu, & Baykan, 2011).

In order to train the neural networks without using learning rate, Expectation Propagation (EP) method was proposed (Soudry, Hubara, & Meir, 2014). However, this method has several limitations. EP specifically uses stepwise activation functions and cannot be generalized in case of other types of neural networks. It has been quoted by the authors (Soudry et al., 2014) that EP is slower as compared to the standard back propagation algorithm.

In methods where learning rate is considered as a hyper-parameter, several DNNs are trained with different values of learning rate. The learning rate that has the least validation error is chosen as the optimum learning rate. This leads to high computation cost due to the training of several DNNs. In this paper, a novel method has been proposed to solve the problem of optimizing the learning rate hyper-parameter. Learning rate is taken as a function of another parameter which is updated along with the weights based on the error. In addition, the proposed method also uses the concept of Laplacian score (He, Cai, & Niyogi, 2005). Laplacian Score of the activation values of a particular layer gives a measure of relevance of the activation values of neurons in that layer. Since the activation value directly depends on the incoming weights, the value of incoming weights to the neurons with high Laplacian Scores are more significant than the weights corresponding to the neurons with low Laplacian Score. This follows that learning rate for incoming weight to neuron with higher Laplacian Score should be lower than the incoming weights to other neurons.

It has been shown on benchmark datasets that the proposed method achieves lower misclassification error than Rectified Linear network and Max-out network with dropout.

2. Overview of previous work

This section presents an overview of some of the existing techniques used in deep learning. Since the proposed work uses Deep MLP without pre-training, recent developments in deep learning without pre-training have been discussed.

Dropout technique is a regularization technique which efficiently prevents the network from overtraining. In dropout, the neurons in input and hidden layers are randomly masked with certain probability during each training iteration. Since the network obtained during an iteration can contain any combination of neurons, dropout removes the mutual dependence of neurons on each other, and hence increases the generalization capacity of the network. The Dropout technique was used to train Rectified Linear Network (Dahl, Sainath, & Hinton, 2013), which further improved the performance of Rectified Linear Network.

Maxout Network (Goodfellow et al., 2013) takes full advantage of dropout technique. For a network having I and J as the size of input and hidden layers, the Maxout Unit is given as follows.

$$f_j(x) = \underset{k\in[1,K]}{max}\ Z_{jk}$$

Where f_j denotes the activation of j^{th} hidden neuron and Z_{jk} is given by

$$Z_{jk} = \sum_{i=1}^{I} x_i W_{ij}^{(k)} + b_{jk}$$

Where $x \in R^I$ denotes the input, K is the number of linear units in Max-out unit, $W \in R^{IxJxk}$ and $b \in R^{JxK}$ denotes the weight and bias respectively. The Maxout activation is computed as maximum over K linear functions and hence can approximate any activation function for the large value of K.

Rectified Linear activation function (Glorot et al., 2011) is used in Deep MLP to remove the problem of vanishing gradient which occurs in the case of nonlinear activation functions such as sigmoid and tanh. Thus, it facilitates learning in supervised manner without unsupervised pre-training. The Rectified Linear activation function is given by

$$f(x) = max(0, x)$$

Where x is the net input to the neuron.

ADADELTA (Zeiler, 2012) is an adaptive learning rate algorithm which updates the learning rate of each parameter in a Deep MLP during training. The learning rate is updated by using the moving average of squares of weight updates and squares of gradients. Square root of the ratio between the two moving averages is taken as the learning rate. There are two hyper-parameters in ADADELTA, but the hyper-parameters have been shown to have no significant impact on the performance.

Based on gradient, hyper-parameter optimization method has been proposed (Maclaurin, Duvenaud, & Adams, 2015) to optimize hyper-parameters while training. The method has several limitations in the sense that for removing the exploding gradient problem, the learning rates are initialized to small values and their optimization is stopped when the error gradient begins to grow in magnitude. Hence, manual intervention is necessary for training of each hyper-parameter.

3. Proposed method

In Deep learning algorithms which use gradient descent, learning rate is considered as a hyper-parameter and is optimized on the basis of the least validation error. Magnitude of back propagated error derivative is less for lower layers as compared to the upper layers (Hochreiter, Bengio, Frasconi, & Schmidhuber, 2001). Hence, different layers should have different learning rates for efficient training. With increase in the depth of Neural Network, there is an exponential growth in the possible sets of learning rates. Search for optimal hyper-parameter requires training for many DNNs with different learning rates in each layer, and hence the optimization of learning rate is computationally expensive.

To remove the problem of optimizing learning rate, a new method has been proposed which removes the drawbacks of previous methods such as Expectation Propagation (EP) (Soudry et al., 2014). EP has a drawback that it is derived specifically for stepwise activation function and cannot be generalized in case of other types of activation functions. In addition, training of EP is two to five times slower as compared to the standard back propagation.

An attempt has been made to develop a new hyper parameter free adaptive learning algorithm for Deep MLP. The proposed method tries to avoid the extensive search for finding the optimum learning rate hyper-parameter in MLP. Learning rate is composed of