International Journal of Conceptions on Computing and Information Technology
Vol. 5, Issue. 1, July' 2017; ISSN: 2345 - 9808

An Analysis of Early Stopping and Dropout Regularization in Deep Learning

Chiranjibi Sitaula
Department of Computer Science and IT
Ambition College
Kathmandu, Nepal
candsbro@gmail.com

Nabin Ghimire
Department of Computer Science and Engineering
Kathmandu University
Dhulikhel, Nepal
nabin.ghimire@ku.edu.np

Abstract— **In deep learning, the problem of overfitting is one of the headaches for the convergence, especially in big data analytics. The problem arises due to imbalance in adjusting the parameters while learning and training the networks. This is one the demerits of deep architecture of neural network and their complexity. In order to diminish the problem of such overfitting, there is a novel approach called regularization. In this paper, the Early Stopping criteria and Dropout algorithm are compared and analyzed. In early stopping, the numbers of iteration of epoch times are analyzed and regulate the value from going outside beyond the certain level. Similarly, dropout thins the layer of neural network and thinning helps to prevent overfitting. This research helps to create a network with better regularization technique for big data.**

Keywords- Machine Learning; Deep Learning; Big data; Data Mining; Regularization; Dropout; Early Stopping

I. INTRODUCTION

Deep learning has been a great topic these days, not just in artificial intelligence, but also in big data analytics as it has the capability to process and analyze data with the help of numerous neurons, forward propagation, back propagation and different parameters for performing those tasks in machine learning. Although the concept of deep learning is quite popular in image processing and medical sector, its application in other sector of big data [3] like video can't be underestimated. With the rise of cloud computing and big data these days, there is much demand of deep learning algorithms for analyzing complex data in the world today.

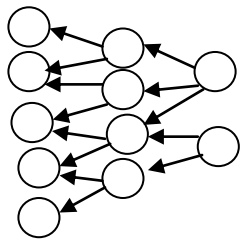

Figure 1. Backward Propagation

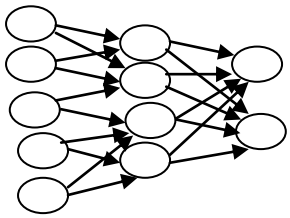

Figure 2. Forward Propagation

The deep neural network is an artificial neural network having more complexity in terms of data manipulation and data extraction for the analysis purpose. It comprises three layers i.e. Input, Hidden and Output [1]. Input layer is the place where we give input the data in terms of pixel or other numeric value with the help of initial weight and bias. Similarly, the hidden layer takes input from input layer and calculates the weight with the help of sigmoid function. The activation is performed with the help of function towards hidden layer. This process as mentioned in Figure 2 is called forward propagation. Furthermore, the backward propagation (as mentioned in Figure 1) is performed in order to update the prior weight. For performing backward propagation, the objective function is calculated as an error and their gradient descent are calculated. The weights are updated with the help of learning rate and gradient descent parameters.

The operation in deep neural network is a bit challenging since it has the problem of overfitting and underfitting. Such demerits can be suppressed with the help of different methods [4, 5]. Regularization [4] helps to regulate the value of weight and objective function from going beyond the limitation. It can be done by different methods available. The popular methods are L1 and L2 regularization [4, 5], also called weight decay approach; and early stopping, the method which uses the concept of epoch time and with the help of time the iterations. Similarly, another method is Dropout [4, 5] which tells the neural network to prune or thin, thereby preventing neural network from going beyond the level of optimality. These sorts of activities help to give the fast performance and convergence towards the level of our requirements.

II. REGULARIZATION TECHNIQUES

In order to perform regularization approach, different researches have been performed so as to optimize the neural network. Some works are for underfitting and some works are for overfitting. For comparing and analyzing the regularization algorithm, [5] performed in-depth analysis with single hidden layer. It performed the analysis between weight decay (L2 norm or ridge regression or Tikonov regularization) and Dropout algorithm. L1 norm was not used in their research for the comparison with Dropout. The result indicated that Dropout outperformed L2 regularization.

Similarly, [6] explained about regularization method for recurrent neural network. Unlike other regularization method for classical neural network, they designed and proposed suitable methods for performing such task. They discussed about the demerits of Dropout, which is one of the popular algorithms, and combined Long Short-Term Memory approach as to fit for the deep architecture of recurrent network and claimed that it reduced overfitting substantially.

[7] Performed their operation under convolutional neural network, which outperformed explicit traditional regularization with re-generalization technique. Their work included theoretical formulation of hypothesis and testing them with the help of state-of-art deep networks and stochastic gradient descent methods.

The research [8, 9, and 10] was concerned with thinning the network using different techniques and their result. Dropout is the method of preventing networking from going overfitting. Thinning is done by randomly dropping the units of neural network during training. Dropout applies its prune condition to the activation function. DropConnect is also the method of thinning where the masking is applied the weight for trimming of neural network. Furthermore, there is another concept called DropAll, which comprises the both concept of Dropout and DropConnect algorithm. There is also another strategy [11] which exhibits the concept of generalization of both algorithms except some difference for performing such task. [12] used deep belief network for deep learning and the regularization method used in their research was the combination of top-down and bottom-up approach by sampling.

III. DROPOUT AND EARLY STOPPING

Dropout is widely used approach in regularization [13, 14, 15, 16] since it calculates probability and thins the network, thereby balancing the weight and network.

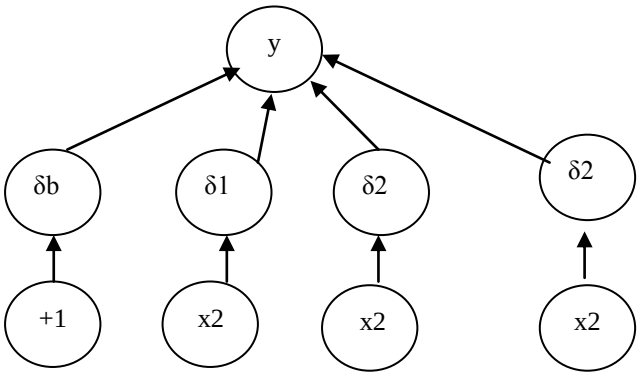

Figure 3. Neural Networks with Dropout [2].

Consider a neural network with L hidden layers. Let $l \in \{1,\ldots\ldots L\}$ index the hidden layers of the network. Let $z^{(l)}$ denote the vector of inputs into layer l, $y^{(l)}$ denote the vector of outputs from layer l ($y^{(0)} = x$ is the input). $W^{(l)}$ and $b^{(l)}$ are the weights and biases at layer l. The feed-forward operation of a standard neural network (Figure 1 and Figure 2) can be described as (for $l \in \{0; : : : ;L - 1\}$ and any hidden unit i)

$$\mathbf{z}^{(l+1)} = \mathbf{W}^{(l+1)}\mathbf{y}^{(l)} + \mathbf{b}^{(l+1)} \quad (1)$$

$$\mathbf{y}^{(l+1)} = \mathbf{a}(\mathbf{z}^{(l+1)}) \quad (2)$$

Where f is any activation function, for example, $f(x) = \frac{1}{1+e^{-x}}$

With dropout, the feed-forward operation becomes (Figure 3)

$$r_j^{(l)} \sim \text{Bernoulli (p)} \quad (3)$$

$$\tilde{y}^{(l)} = r^{(l)} \times y^{(l)} \quad (4)$$

$$z_i^{(l+1)} = W_j^{(l+1)}\tilde{y}^{(l)} + b_i^{(l+1)} \quad (5)$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \quad (6)$$

Early stopping [17] is a simple method, yet superior to many methods available in neural network architecture. Although there are other approaches to fight overfitting, but this method is easy as it controls with the help of number of epochs while back propagation and forward propagation operation.

IV. EXPERIMENT

For the experiment of regularization using dropout and early stopping in deep learning, the MNIST dataset [18] is used. This dataset is a quite popular dataset that has been to benchmark classification performance. It comprises of 60,000 training images and 10,000 test images, for which each is a standardized 28^2 pixel grayscale image of a single handwritten digit. The programming language R is used for the implementation with required packages and frameworks. In order to implement Multi Layer Perceptron for deep learning, input layer of 717 nodes, and two hidden layers of 200 nodes each are used with single output layer.

Firstly the analysis is made using Dropout regularization algorithm for three different activation functions- Rectifier linear, Tanh and Maxout. The RMSE is calculated for each algorithm are filled in the below Table 1.

TABLE 1 RMSE FOR THREE ACTIVATION FUNCTION USING DROPOUT

Input Dropout Ratio	**Root Mean Square Error(RMSE)**					
	TanhT	***MaxoutT***	***LinearT***	***TanhV***	***MaxoutV***	***LinearV***
0.2	1.41	1.04	0.96	1.40	1.07	0.98
0.3	1.53	1.07	0.92	1.50	1.09	0.96
0.4	1.58	1.15	0.97	1.52	1.18	1.0
0.5	1.7	1.55	1.04	1.66	1.58	1.07
0.6	1.76	2.34	1.16	1.71	2.35	1.15
0.7	2.13	2.77	1.29	2.09	2.83	1.29
0.8	2.22	3.57	2.05	2.18	3.51	2.07

In Table1, the root mean square errors obtained by giving different input dropout ratios, starting from 0.2 through 0.8, are listed for training and testing data. The capital letter after

each activation function represents if they are training or validation set. Training sets are represented by 'T', whereas validation set are represented by 'V'.

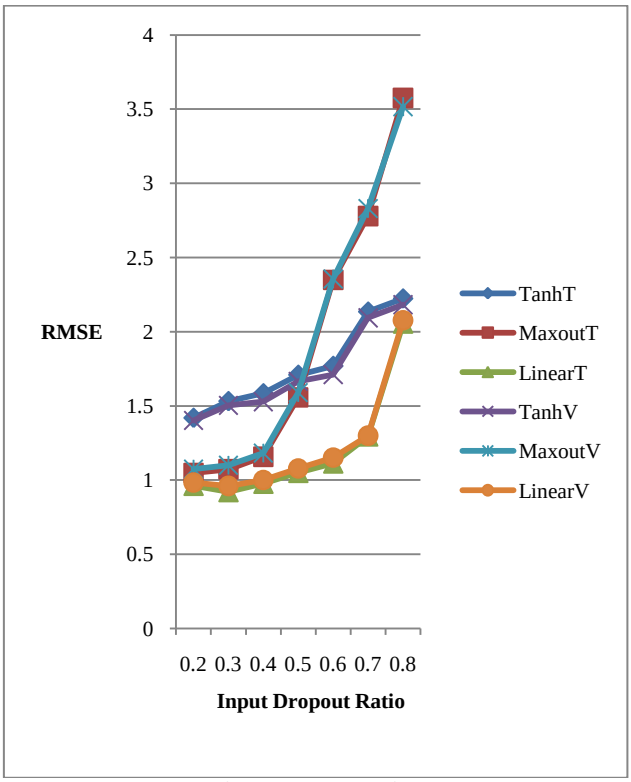

Figure 4. RMSE for three activation functions using Dropout

Similarly, the result of Table 1 was plotted as a line graph. In total, six lines are shown on the graph, three for testing set and remaining three for validation set. Some of the lines are seen as overlapped with others while plotting input dropout ratio with their root mean square error values. It is observed that linear rectifier remained at the bottom having the lowest error rate among others for every input dropout ratio.

Similarly, in Early Stopping regularization, for training and validation datasets, the root mean square errors are listed against their correspond epoch's time in Table 2.

TABLE 2 RMSE FOR THREE ACTIVATION FUNCTION USING EARLY STOPPING

Epoch time for training set	**Root Mean Square Error(RMSE)**					
	TanhT	***MaxoutT***	***LinearT***	***TanhV***	***MaxoutV***	***LinearV***
10	0.63	1.99	0.41	0.81	2.13	0.69
20	0.46	1.68	0.32	0.74	1.71	0.69
30	0.55	1.37	0.35	0.77	1.43	0.70
40	0.48	1.83	0.40	0.75	1.79	0.69
50	0.59	1.40	0.36	0.78	1.47	0.69
60	0.41	1.67	0.41	0.75	1.76	0.6
70	0.41	1.40	0.44	0.74	1.44	0.71

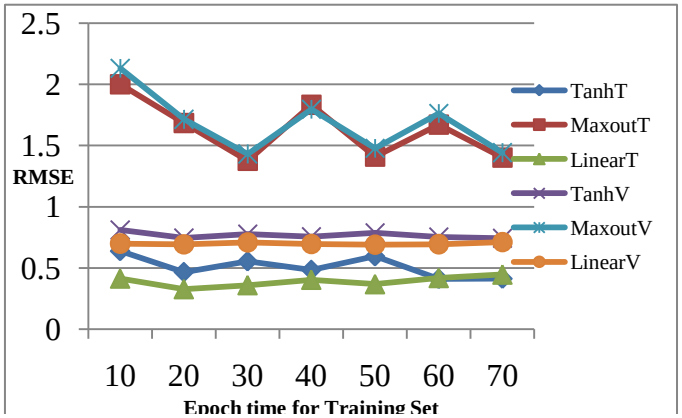

Figure 5. RMSE for three activation functions using Dropout

In Figure 4, it can be seen that the six lines for three functions in different mode (training and testing) are plotted against their errors. It can easily be justified that linear rectifier has the least error rate while training. Hyperbolic tangent function remained at last having the highest error for almost every epoch time.

V. CONCLUSION AND LIMITATION

While observing the experiment on deep learning in addition to two learning algorithms-Early Stopping and Dropout- among others, it can be seen empirically that Early Stopping has the potential to converge with less error rate than Dropout algorithm. It can also easily be proved that, whatever the types of regularizations have been used, linear rectifier outperformed other activation functions available. Furthermore, it is also claimed that for big size of data which was used in the experiment, the regularization seemed work faster than small size of data that has been used in testing.

The main restriction of the experiment is the processors. It would be better if there was recent processor with necessary RAM. Similarly, other datasets could be used in the experiment to make it more trustworthy and reliable. The parallel processing concept could be used for fast execution of big data. Likewise, the analysis of other algorithms like L1 and L2 regularization could also be used in addition to those algorithms.

REFERENCES

[1] L. Yann, B. Yoshua, H. Geoffrey, Deep Learning, Macmillan Publishers Limited, 2015J.

[2] D. Li, Y. Dong, Deep Learning Methods and Applications, Foundations and Trends in Signal Processing, Volume 7, Issues 3-4, 2014.

[3] W. Li, W. Gang, S. Dennis. Deep Learning Algorithms with Applications to Video Analyics for A Smart City: A Survey, arXiv preprint arXiv:1512.03131, 2015

[4] S. Sargur N. Regularization for Deep Learning, University of Buffalo,tutorial.

[5] P. Ekachai. An Analysis of the Regularization between L2 and Dropout in Single Hidden Layer Neural Network, 7th International Conference on Intelligent Systems, Modeling and Simulation, 2016.

[6] Z. Wojciech, S. Ilya Sutskever, V. Oriol. Recurrent Neural Network Regularization, Neural and Evolutionary Computing, arXiv: 1409.2329, 2015.

[7] H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian. Z. Chiuan, B. Samy, H. Moritz, R. Benjamin, V. Oriol. Understanding Deep Learning Requires Re-thinking Generalization, arXiv: 1611.03530v1, 10 Nov 2016.

[8] S. Nitish, H. Geoffrey, K. Alex, S. Ilya, S. Ruslan, Dropout: A Simple Way to Prevent Neural Network from Overfitting, Journal of Machine Learning Research, page 1929-1958, 2014.

[9] F. Xavier, A. Luis A., DropAll: Generalization of Two Convolutional Neural Network Regularization Methods, Springer, 2014.

[10] W. Li, Z. Matthew, Z. Sixin, L.Yann, F. Rob, Regularization of Neural Networks using DropConnect, Proceeding of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.

[11] I. Alexandros, T. Anastasios, P. Ioannis, DropELM:Fast Neural Network Regularization with Dropout and DropConnect, Neurocomputing, pages: 57-66, 25 August,2015

[12] G. Hanlin, T. Nicolas, C. Mattieu, L. Joo-Hwee, Top-Down Regularization of Deep Belief Networks, Advances in Neural Information Processing 26, 2013.

[13] Z. Jingwei, Z. Jun, Z. Bo, Adaptive Dropout Rates for Learning with Corrupted Features, *Proceeding of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI, 2015.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving Neural Network by Preventing Co-Adaptation of Feature Detectors, arXiv:1207.0580v1, 3 Jul 2012.

[15] W. Stefan, W. Sida, L. Percy, Dropout Training as Adaptive Regularization, arXiv:1307.1493v2, 1 Nov 2013.

[16] H. David P., L. Philip M., On the Inductive Bias of Dropout, Tutorial.

[17] P. Lutz, Early Stopping-but when? , Springer-Verlag Berlin Heidelberg, 2012.

[18] Q. Yu , "THE MNIST DATABASE of handwritten digits". Retrieved 18 August 2013.

[19] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science.