IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-3 E-ISSN: 2347-2693

Comparison of various Activation Functions: A Deep Learning Approach

Mohammed Ibrahim Khan[1*], Akansha Singh[2], Anand Handa[3]

[1*] Computer Science and Engineering, PSIT College of Engineering, Kanpur, India
[2]Computer Science and Engineering, PSIT College of Engineering, Kanpur, India
[3]Computer Science and Engineering, PSIT College of Engineering, Kanpur, India

[*]*Corresponding Author: ibrahimkhan7777@gmail.com, Tel.: 8400666777*

Available online at: www.ijcseonline.org

Received: 20/Feb//2018, Revised: 26/Feb2018, Accepted: 19/Mar/2018, Published: 30/Mar/2018

***Abstract*—**A branch of machine learning that attempts to model high-level abstractions in data through algorithms by the use of multiple processing layers with complex structures and nonlinear transformations is known as Deep Learning. In this paper, we present the results of testing neural networks architectures through tensorflow for various activation functions of machine learning algorithms. It was demonstrated on MNIST database of handwritten digits in single-threaded mode that blind selection of these parameters can hugely increase the runtime without the significant increase of precision. Here, we try out different activation functions in a Convolutional Neural Network on the MNIST database and provide as results the change in loss values during training and the final prediction accuracy for all of the functions used. These results create an impactful analysis for optimization and training loss reduction strategy in image recognition problems and provide useful conclusions regarding the use of these activation functions.

Keywords*—CNN (Convolution Neural Network), activation functions and MNIST(Modified National Institute of Standards and Technology) dataset*

I. INTRODUCTION

Machine learning is a sub branch of Artificial Intelligence discipline. It is a branch of AI through which computer applications and systems are designed that receive data inputs, train few outputs, builds a conclusion. Real life example of ML can be seen in fields like object recognition, visual-semantic embedding, language identification, cloud computing [1], speech recognition [2], video classification, generation of alphabet of symbols for multimodal human-computer interfaces etc. [3]

MNIST dataset contains60,000 training images and 10,000 test images of the digits0 to 9. The images have grayscale values in the range 0:255.Figure 1 gives an example images of handwritten digits that were used in testing. We have trained the network by using the host with Intel Core i5-7200u CPU insight on a subset of 10000 images of the training set.

The four mathematical areas that have achieved advancement in fundamental mathematics of Machine Learning are, namely: network architecture, optimization method and Batch Normalization, activations functions, and objective functions. Zhang & Ma, 2012 have compared the advantages of using multi modal system rather than single model [4]. Neural networks gained attention of the researchers in the 90'sand early 2000's, (Zhou et al., 2002) [5]. Likely other techniques will add up to the benefits of CNN. For example, Dropout layer analysis (Srivastava et al., 2014) in many different architectures [6].

Our work comprises of trying different activation function other than creating a highly complex neural network for the generation of accurate results from the dataset. In this paper we have tried to test different activation function on MNIST to check the most appropriate one for the optimization.

As data passes through a deep neural network, each layer transforms the data to better interpret and gather features. Therefore, the best possible function at the top of a network may not be optimal in the middle or bottom of a network. The advantage of our architecture is that rather than choosing activations at specified layers or over an entire network, one can give the network the option to choose the best possible activation function of each neuron at each layer.

The most typical task in this training module is to vary the constraints and parameters to get accurate performance without losing their efficiency.

The paper is organised as follows: Section I contains the introduction of how a deep neural network trains itself. Section II describes the related work that has been accomplished in this field. Section III describes the methodology and the architecture of CNN used to test various activation functions. Section IV describes the result

of our evaluation and Section V presents the conclusion of our work followed by the references used.

II. RELATED WORK

Today's most important task is to develop the most effective activation function to be used. Sigmoid and hyperbolic tangent functions were used in early age of convolution network because of their simplicity but lately ReLU (Rectifier Linear Unit) have gained much emphasis in the training of CNN to get accuracy as they increase the optimization accuracy and training speed.

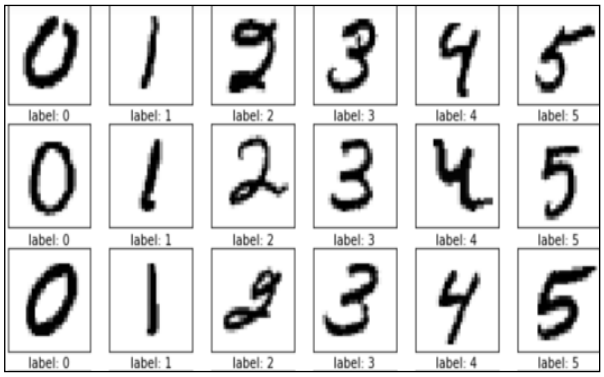

Fig. 1. Example Images of Handwritten Digits.

Gulcehre et al. (2016a) has worked a lot in the field of improving activation functions by introducing a stochastic variable to the sigmoid and hyperbolic tangent functions. Since sigmoid and hyperbolic tangent function both contain areas of high saturation for values in large magnitude, the stochastic variable can aid in pushing the activation functions out of high saturation areas [7].In our work, rather than introducing stochasticity, we introduce several activations at each neuron, from which the network can chose a combination. Thus, it can reap the benefits of the sigmoid and hyperbolic tangent function without being limited to these functions at each layer.

A different approach was taken by Li et al. (2016) from common work that analyses different activation functions. Rather than combining inputs, they use multiple biases to find features hidden within the magnitudes of activation functions [5]. In this way, they can threshold various outputs to find hidden features and help filter out noise from the data. We restrict the range of our activation functions, which helps the neural network find features that may be hidden within the magnitude of another activation function. Thus, we are able to find hidden features via known activation functions without introducing multiple biases.

Scardapane et al. (2016) create an activation function during the training phase of the model. However, they usea cubic spline interpolation rather than using the basis of the rectifier unit. Their work differs from ours in that we use the many different available activation functions rather than creating an entirely new function via interpolation [5]. It is important to note that we restrict our activation function to a particular set and then allow the network to choose the best one or some combination of those available rather than have activation functions with open.

Chen (2016) uses multiple activation functions in a neural network for each neuron in the field of stochastic control. Similar to our work, he combines functions such as theReLU, hyperbolic tangent, and sigmoid. To train the network,Chen uses Neuroevolution of Augmenting Topologies(NEAT) to train his neural network for control purposes. However, he simply adds together the activation functions without capturing magnitude and does not allow the network to choose an optimal set of activations for each neuron.

Activation functions also known as transfer functions are used to map input nodes to output nodes in certain fashion[5] (see the conceptual scheme of an activation function in Figure 2). We are considering here most common activation functions that are widely using for deep learning.

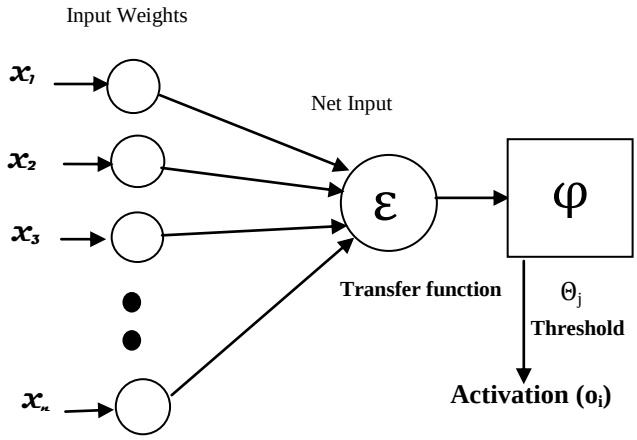

Fig. 2. The Role of Activation Function in the Process of Learning Neural Network

III. METHODOLOGY

The newer versions of tensor flow have provided a large number of activation functions. Some of which are fairly new and less commonly used. We wanted to do a comparative study of these new functions along with the existing popular functions.

In Figure 3. the diagram the architectural construct is shown

carefully to show the simplicity and non-complex details. It contains four convolution layers and two pooling layers with one fully connected and output layer that helps in analyses of stimuli and weights of the MNIST dataset and help achieve the correct analysis of activation functions.

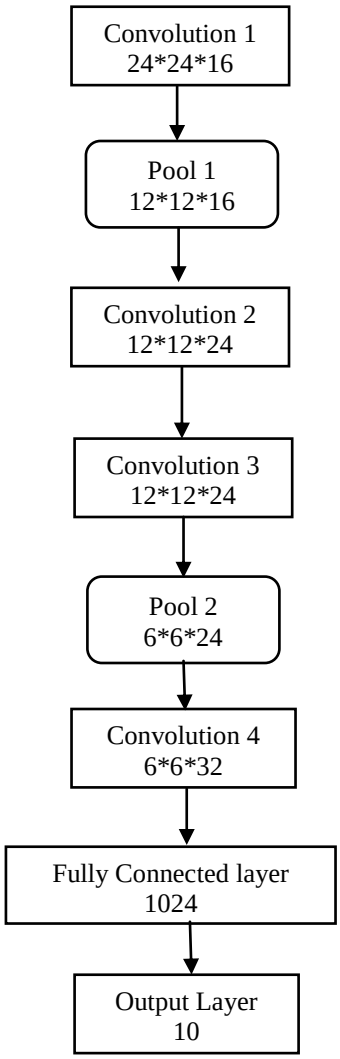

Fig. 3. The architecture of CNN that is used to test the accuracy of activation functions.

Each filter size is experimented to get the correct value of loss generated from the output layer that will help in analysis of correct function. Table 1 shows the exact details about each filter size, strides used and combination of layer developed.

Table 1: Showing different layers of CNN with its complete details of layers and filters.

Layer	Filter Size	Strides	No. of filters
Conv1	5x5	1	16
Pool 1	2x2	2	-
Conv2	3x1	1	24
Conv3	1x3	1	24
Pool2	2x2	2	-
Conv4	3x3	1	32
Fully connected	-	-	768
Output	-	-	10

To analyse the strength and weakness of the neural network in the activation function we took six different activation functions to test the images of MNIST dataset. Each function is experimented with different pairs to analyse its effect on the increase in optimization and on reducing the training loss. The details are shown in Table.1 for different functions trained upon in the CNN on MNIST dataset. The various activation functions that have been used are as follows-

- Sigmoid Function- The function used is as follows-
$$s(x) = 1/(1 + e^{z})$$
- Tan Hyperbolic Function- The function used is as follows-
$$f(x) = \tanh(x) = \frac{(e^{z} - e^{-z})}{(e^{z} + e^{-z})}$$
- Rectified Linear Units- The function used is as follows-
$$f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$
- Leaky Rectified Linear Units- The function used is as follows-
$$f(x) = \begin{cases} 0.2x \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$
- Exponential Rectified Linear Units- The function used is as follows-
$$f(x) = \begin{cases} e^{x} - 1 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$
- Scaled Exponential Units- The function uses a value of $\lambda = 1.0507$

IV. RESULTS AND DISCUSSION

The various activation functions are applied on the subset of MNIST dataset of size 10,000 images. The x-axis represents the number of samples and the y-axis denotes the loss value. The experimental results are as follows:

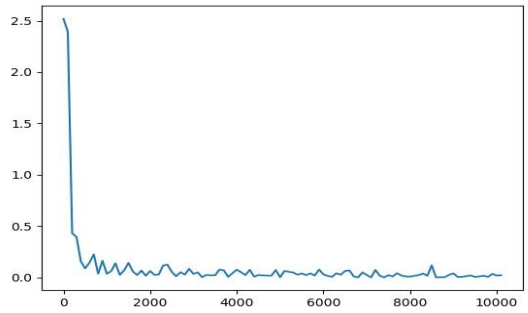

Fig 3.1. Sigmoidal loss graph

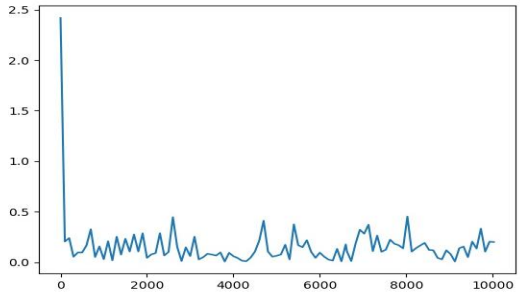

Fig 3.2. Tanh loss graph

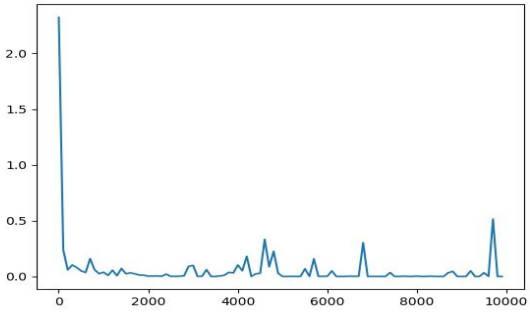

Fig 3.3. ReLU loss graph

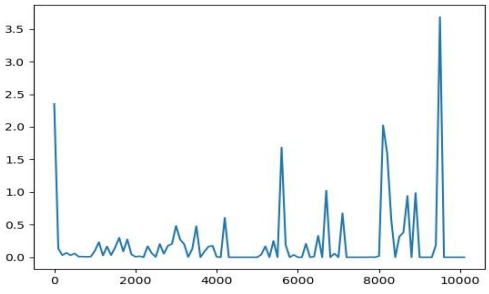

Fig 3.4. Leaky ReLU loss graph

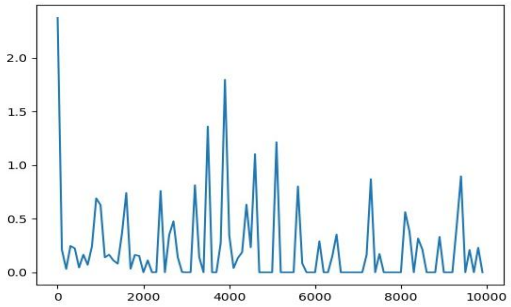

Fig 3.5. eLU loss graph

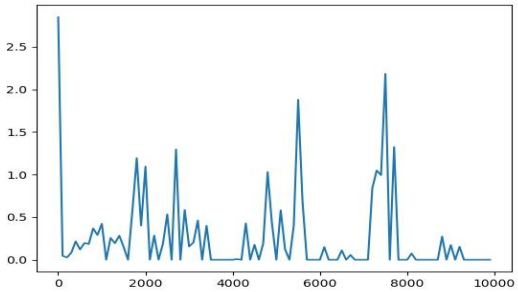

Fig 3.6. seLU loss graph

Table 2: Accuracy and loss details for various activation functions

Activation Function	Accuracy	Loss
Leaky ReLU(Rectified Linear Units)	97.91	4.504
elu (Exponential Linear Units)	98.00	1.912
selu(Scaled Exponential Units)	97.67	2.256
Sigmoid	98.85	0.041
Tanh	96.44	0.314
ReLU	97.79	0.528

The graphs of loss for different activation functions illustrate the pattern of learning in case of each of these functions. For same CNN architecture, different activation functions largely influence the training pattern.

Table 2. provides the accuracy and loss values on the test dataset of 10,000 images for each of the activation functions, which again shows slight variation for different functions.

V. CONCLUSION

From the given results, it is clear that different activation functions start to behave drastically differently as the number of input samples increases. The above comparison thus indicates that when it comes to deep learning tasks, the choice of activation function is crucial and provide improvements is results. This result in no way indicates the superiority of one activation function over the others since on

different dataset the result might be different. It just gives an incentive for researchers to try out different functions.

REFERENCES

[1] Srinivas Jagirdar, K. Venkata Subba Reddy, Dr. Ahmed Abdul Moiz Qyser, "*Cloud Powered Deep Learning-Emerging Trends*", International Journal of Computer Sciences and Engineering (IJCSE), Vol-4, Issue-6, 2016

[2] N. Gordienko, S. Stirenko, Yu. Kochura, O. Alienin, M. Novotarskiy, Yu. Gordienko, A. Rojbi (2017), "*Deep Learning for Fatigue Estimation on the Basis of Multimodal Human-Machine Interactions*", XXIX IUPAP Conference on Computational Physics (CCP2017) (Paris, France).

[3] S. Hamotskyi, A. Rojbi, S. Stirenko, and Yu. Gordienk(2017), "*Automatize Generation of Alphabets of Symbols for Multimodal HumanComputer Interfaces*", Proc. of Federated Conference on Computer Science and Information Systems, Prague (FedCSIS-2017) (Prague,Czech Republic).

[4] Zhang, Cha and Ma, Yunqian. "*EnsembleMachine Learning*" volume 1. Springer, 2012.

[5] Zhou, Zhi-Hua, Wu, Jianxin, and Tang,Wei. "*Ensembling neural networks: many could be better than all*." Artificial Intelligence, 137(1-2):239–263, 2002.

[6] Srivastava, Rupesh K, Greff, Klaus, and Schmidhuber, J¨urgen. "*Training very deep networks*". In Advances in Neural Information Processing Systems, pp. 2377–2385, 2015.

[7] Gulcehre, Caglar, Moczulski, Marcin, Denil, Misha, and Bengio, Yoshua. "*Noisy activation functions*". In International Conference on Machine Learning, pp. 3059–3068,2016a.

Authors Profile

Mohammed Ibrahim Khan is pursing Bachelor of Technology in Computer Science and Engineering from PSIT College of Engineering, Kanpur.

Akansha Singh is pursing Bachelor of Technology in Computer Science and Engineering from PSIT College of Engineering, Kanpur.

Anand Handa pursued Bachelor of Technology from University Institute of Engineering Technology, Kanpur and Masters from RGPV, Bhopal in Computer Science and Engineering. He is currently pusuing Ph.D. and working as an Assistant Professor in Department of Computer Science and Engineering, PSIT College of Engineering, Kanpur.