Effective usage of activation functions for data classification with TensorFlow in deep neural networks

Buddhika Priyabhashana[1], KPN Jayasena[2]
Department of Computing and Information Systems, Faculty of Applied Sciences, Sabaragamuwa University of Sri Lanka
[1]bpriyabhashana@gmail.com, [2]pubudu@appsc.sab.ac.lk

Abstract— *Artificial neural networks can be known as a computer system modeled on the human brain and neural system. In data classification, neural network provides fast and efficient results. Neural Network models are trained by using sets of labeled data. Neural networks have the ability to work with data, based on the training. There are thousands of interconnected nodes that belong to interconnected hidden layers inside the neural network. Activation function that have included in the neural network provides the output based on given an input or set of inputs. This research work focused on the comparison of the effects of using several activation functions on multiple hidden layers for classification using MNIST (Mixed National Institute of Standards and Technology) data set. Data classification was made using TensorFlow library. Tensorflow library with the help of keras used to build the neural network model. The experiment results of Rectified Linear Unit (ReLu), Leaky ReLU, Hyperbolic Tangent (tanH), Exponential Linear Unit (eLu), sigmoid, softplus, softmax and softsign activation functions. Data have been collected for the experiment in two different methodologies. There is a hidden layer with one activation function and multiple hidden layers with multiple activation function. The result of the study shows that the higher accurate rate than 88% for training and testing when it uses multiple hidden layers with multiple activation functions.*

Keywords— **Neural network, TensorFlow, Deep Learning, Activation Function, MNIST dataset**

I. INTRODUCTION

Artificial Intelligent (AI) can be known as a discipline and machine learning (ML) is specific way of solving AI problems. Instead of programming a computer, able to teach a computer to learn something and it does what user want. When considering the ML, deep learning also a relevant section that supports in ML domain. Deep learning is a subset of machine learning methods based on artificial neural networks (ANN). In deep learning, a computer model trains to perform classification tasks directly from any kind of data sets. Deep learning methods use neural network (NN) architectures, therefore deep learning models are known as deep neural networks. To get expected results from neural networks, have to train models. Deep learning models are trained by using sets of labeled data. Then neural networks have ability to work with data, based on the training.

There are thousands of interconnected nodes that belongs to interconnected hidden layers inside the neural network. Before use the deep learning approaches such as classification and regression, need to train the models by using a suitable way. There are three most common ways: (1) Training from Scratch (2) Transfer Learning (3) Feature Extraction. To train deep learning model with scratch, need to gather large labeled data set and - also required to design a network architecture. From this architecture will learn the features. Transfer learning approach basically fine-tuning a pre-trained model. In the feature extraction use the neural network as feature extractor. All the layers are assigned for learning special feature related to the model.

When training deep learning model, there is a factor that need to consider. It is the variation of the learning rate against loss and epoch. Epoch can be known as time that the learning algorithm work through the entire training dataset. Loss can be known as scalar value that

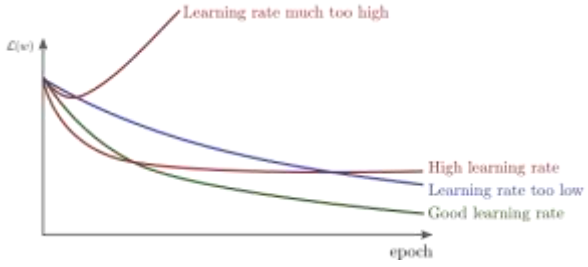


Figure 1. Variation of learning rate

attempt to minimize during the training of deep learning model. We can identify different kind of learning rates when train a model.

When train a deep learning model need to keep good learning rate for accurate results. And the next thing is that the loss should reduce continuously. Other than the

learning rate we consider training loss and validation loss with this research work. In this study, classification was

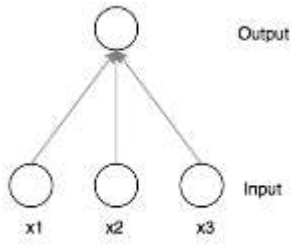


Figure 2. Linear Model

made using TensorFlow and evaluate the training performance against different kind of activation functions and the relationship between hidden layers and neurons. When considering about the deep learning, there are two types of models. One is Linear Model and other one is Non Linear Model. A linear model can be known as nodes and edges.

There are three inputs (x_1, x_2, x_3) and those are combined with some weights given on each edge to produce an output. Then we can define the output (y) as,

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3$$

it represents the linear combination equation. Figure 3 represents that adding a hidden layer in-between node and edges. The hidden nodes also combined with some weights and those are affecting to the output of the model.

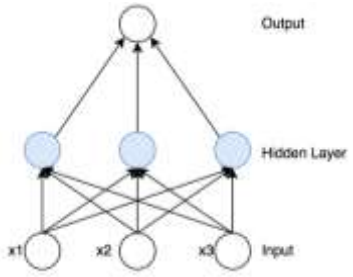


Figure 3. Neural network with hidden layer

When considering about inputs to the hidden nodes, the inputs are liner combinations as well as outputs of hidden nodes. And finally total output also represented as linear combination. Inputs to the hidden nodes can be represented as h_1, h_2, h_3

$h_1 = w_1x_1 + w_4x_2 + w_7x_3$ / $h_2 = w_2x_1 + w_5x_2 + w_8x_3$ / $h_3 = w_3x_1 + w_6x_2 + w_9x_3$

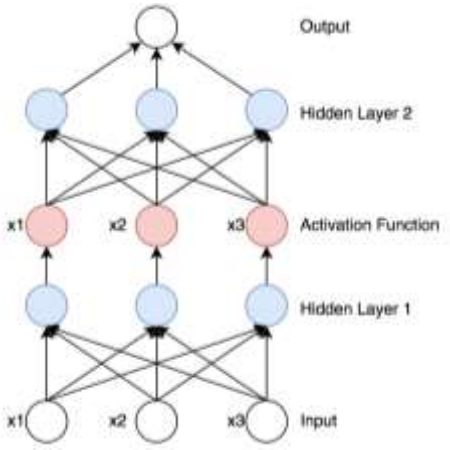
Then output (O) can be represented as,

$O = w_{10}h_1 + w_{11}h_2 + w_{12}h_3$

$O = (w_{10}w_1 + w_{11}w_2 + w_{12}w_3)\ x_1 + (w_{10}w_4 + w_{11}w_5 + w_{12}w_6)\ x_2 + (w_{10}w_7 + w_{11}w_8 + w_{12}w_9)\ x_3$

$O = W_1x_1 + W_2x_2 + W_3x_3$

This model also the same linear model as before despite adding a hidden layer of neurons. According to that, it is



possible to simplify to a linear combination. The thing is adding a hidden layer to the model is the combination collapsing back into just a linear model. As an alternative solution we use activation functions. In computational networks, activation function of a node helps to define the output of the node given an input or set of inputs. Activation functions are most important for an Artificial Neural Network (ANN) to make sense of something complicated and Non-linear complex functional mappings between the inputs and output. Activation functions introduce non-linear properties to neural Network.

There are different kind of activation functions. With this research work what we are going to do is find the effective way to use activation functions by evaluating the functionality and the performance with different kind of models. We can categorize activation functions in to main types. One is linear activation function and other one is nonlinear activation functions. When consider about the linear activation function, output of the functions will not be confined between any range. But in nonlinear activation functions, it makes the model to

Figure 4. Neural network with hidden layers and activation function

generalize or adapt with variety of data and to differentiate between the output.

With this research work we consider about eight types of activation functions. Those are Rectified Linear Unit (ReLu), Leaky ReLU, Hyperbolic Tangent (tanH), Exponential Linear Unit (eLu), sigmoid, softplus, softmax and softsign.

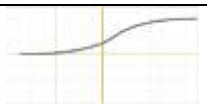
Activation Function	Plot	Equation
Linear		$f(x) = x$
Sigmoid or Logistic		$f(x) = \frac{1}{1+e^{-x}}$
Tanh or hyperbolic tangent		$f(x) = \frac{2}{1+e^{-2x}} - 1$
ReLU (Rectified Linear Unit)		$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases}$
ELU (Exponential Linear Unit)		$f(x) = \begin{cases} x, x > 0 \\ \alpha e^{x}, x \leq 0 \end{cases}$
Leaky ReLU		$f(x) = \begin{cases} x, x > 0 \\ \alpha x, x \leq 0 \end{cases}$

Table 1. Activation Functions

The other important part of the neural network is neurons. Neurons are the basic unit of neural network. As discussed before, neurons could have several inputs and for each input there is a weight. When neuron activate it multiplies all the inputs with connection weight. That's what we did with previous section.

II. RELATED WORK

There are considerable number of research work related to the neural network field. And neural network and deep learning is the one of famous research area among researchers.

Ertam, F. and Aydin, G presented a research work related to data classification in neural network by focusing one activation function with a hidden layer. They have collected testing accuracy and compare them for selecting a best application from them.

III. METHODOLOGY

Neural network can be arbitrarily complex. To increase hidden dimensions, can add more neurons. To increase function composition, can add more layers. And next point is if there are multiple labels per example, can add more outputs. When build a neural network to learn data there are two ways to achieve feasibility and accuracy with the model. First thing is training models using manual feature engineering. Feature engineering can be known as the process of using knowledge of the data to create features that make machine learning algorithms work. The second way is adding more layers and neurons using simple set of input features.

To evaluate the performance, we used MNIST dataset (fashion MNIST). For this study, the MNIST dataset used to measure and evaluate the performance of the TensorFlow library. MNIST is an abbreviation of "Modified National Institute of Standards and Technology" and it is a large data set consisting of handwritten numbers widely used for the training of image processing systems.

In this fashion MNIST dataset, there are 60,000 training examples for training and 10,000 testing examples for the test. Each picture consists of 28×28 grayscale/single channel images.

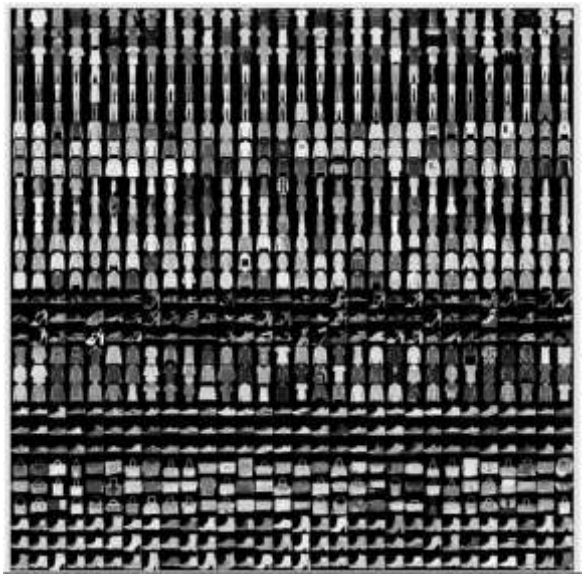


Figure 5. Fashion MNIST example dataset

What we are going to do is, the suitable classification performance of the data set was measured by applying different activation functions in the program prepared with the TensorFlow library. TensorFlow library with the

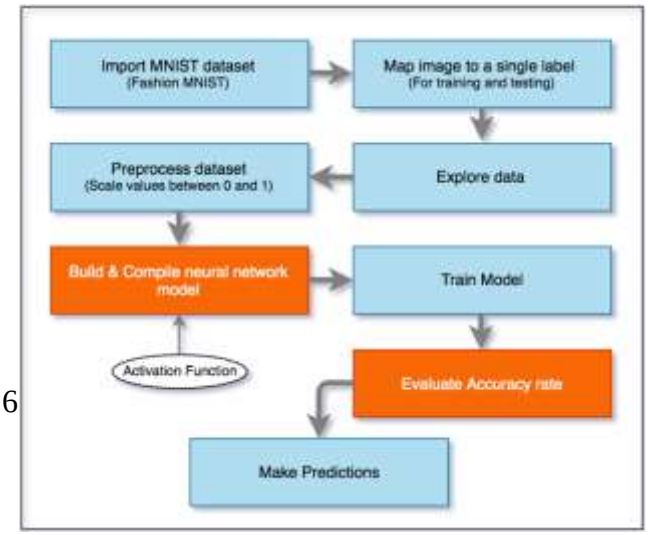


Figure 6. System functions

help of keras used to build the neural network model.

Data have been collected for the experiment in two different methodologies. There is a hidden layer with one activation function and multiple hidden layers with multiple activation function.

As a first methodology, create a neural network with one layer that including 128 neurons and used one activation function with the layer. Used 60,000 training images and 10,000 testing images for each iteration. We take as applicable combination that any iteration shows beyond 0.88 success rate from training and testing accuracy.

Figure 6 shows the overview of system functions. After importing MNIST data set those are mapping into single label. Then explore data and preprocess dataset for scaling the value between 0 and 1. Then build and compile neural network model. In this phase we assign activation functions to relevant layers and change parameters related to the model such as number of hidden layers and neurons in each layer. Then train the model by using labeled data set (60000 images). Then evaluate the accuracy rate using testing dataset (10000 images). Then collect experiment data regarding accuracy in both training and testing phases.

III. RESULTS

Regarding the experiment with one activation function with single layer, combination could not achieve an average accuracy rate for training and testing than 0.88.

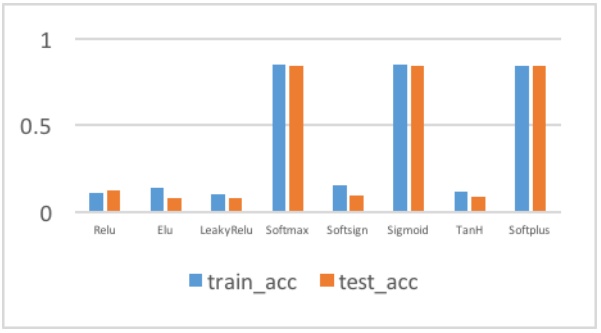


Figure 7. Single layer with single activation function – training and testing accuracy

Now as experiment we collected data by adding two activation functions with two hidden layers by executing all possible iterations. 128 neurons were added to first layer and 10 neurons were added to second layer. When use two activation functions we added common function to the second layer.

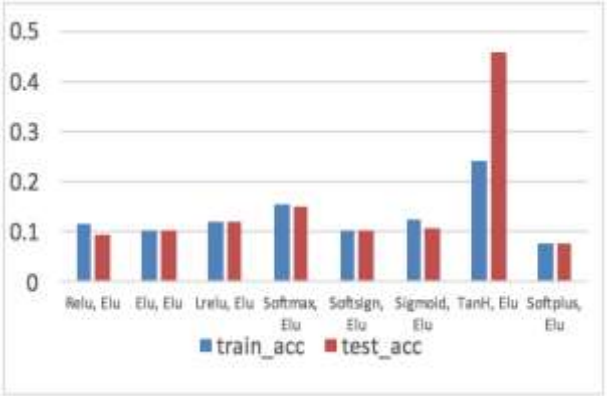


Figure 9. Training and testing accuracy – ELU as common function

After gathering data from experiment generated graphs based on the data for analysing. Further details are

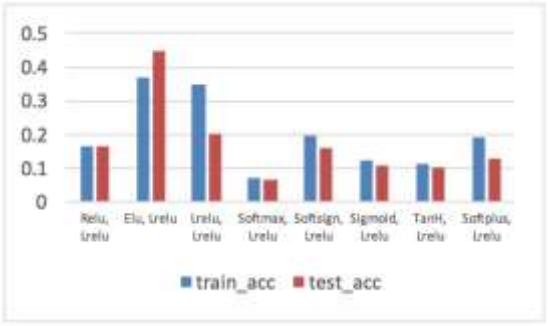


Figure 10. Training and testing accuracy – LeakyReLU

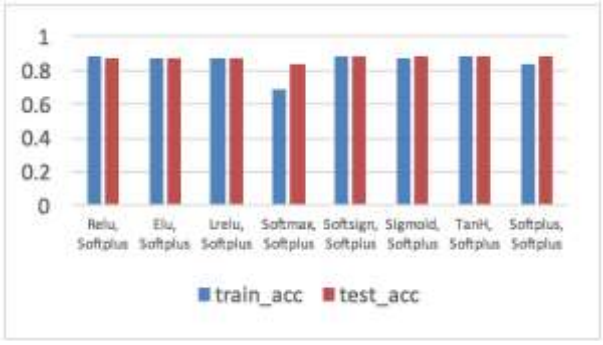


Figure15. Training and testing accuracy – Softplus as common function

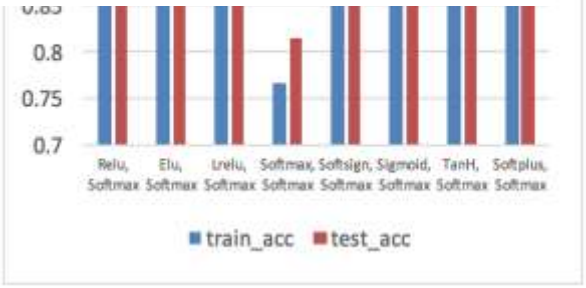


Figure 11. Training and testing accuracy – Softmax as common function

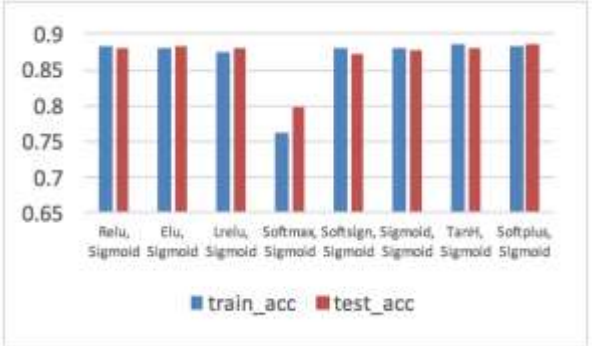


Figure 13. Training and testing accuracy – Sigmoid as common function

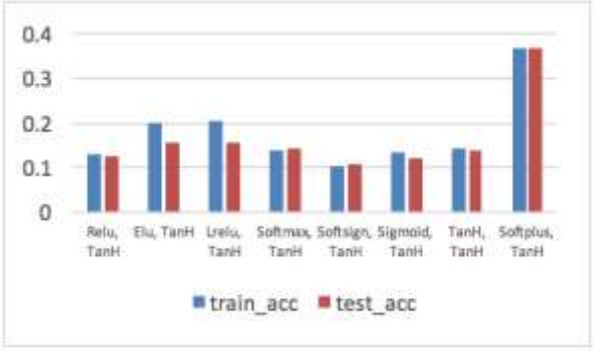


Figure 14. Training and testing accuracy – TanH as common function

descried in below Result and discussion section

IV. DISCUSSION AND CONCLUSIONS

By using the methodology that described above, collected every possible values by changing activation functions. For validate the accuracy of the collected testing and training data, got the mean value from ten iterations in each execution. By using graphs, filtered executions that keep the average accuracy rate of training and testing value above 0.88 (88%) for the further analysis process.

When filter the values, the experiment regarding single hidden layer and single activation function was rejected. Because iterations couldn't pass the minimum accurate levels. So with the result can predict single activation function is not enough powerful for complex models.

When consider about different activation functions with two hidden layers had the ability to pass the minimum expected accurate level in some iterations. The condition for filter data is both testing accuracy and training average accuracy rate should be above 0.88 (88%).

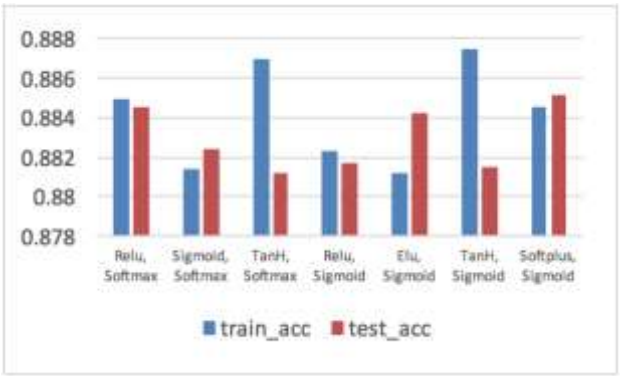


Figure 16. Filtered Training and Testing accuracy rates

You can see here selected iterations of the neural network. And as the result from the graph you can see there is softmax or sigmoid functions involve with the selected processes. Sigmoid or softmax or both functions affected to the training and testing processes for achieve higher rate of accuracy and also those handles the huge part as activation functions in the neural network when consider about fashion MNIST data set. The increase in the number of iterations with different activation functions for multiple hidden layers showed an increase in the accuracy values, but the thing is classification time is also increased. Will take more time for more complex models. with subsequent studies we target to increase accuracy by variations of hidden layers, neurons and activation functions.

REFERENCES

API - Activations — TensorLayer 2.0.1 documentation. [online] Available at: https://tensorlayer.readthedocs.io/en/latest/modules/activation.html

Ertam, F. and Aydin, G. (2017). Data classification with deep learning using Tensorflow. 2017 International Conference on Computer Science and Engineering (UBMK).

Colab.research.google.com. (2019). *Google Colaboratory*. [online] Available at: https://colab.research.google.com/github/tensorflow/ [Accessed 26 May 2019].

TensorFlow. (2019). *Train your first neural network: basic classification | TensorFlow Core | TensorFlow*. [online] Available at: https://www.tensorflow.org/tutorials/keras/basic_classification

Mathworks.com. (2019). *What Is Deep Learning? | How It Works, Techniques & Applications*. [online] Available at: https://www.mathworks.com/discovery/deep-learning.html

GitHub. (2019). *zalandoresearch/fashion-mnist*. [online] Available at: https://github.com/zalandoresearch/fashion-mnist

.