HIDDEN LAYER USING THE SIMULATION TENSORFLOW PLAYGROUND

Muhammad Haziq Iskandar[1] ,Nursyaza Nisa[2] ,Nurul Dalila[3]

Faculty of Information and Communication Technology, Technical University of Malaysia Melaka, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Malacca, Malaysia.

B032010443@student.utem.edu.my
B032010244@student.utem.edu.my
B032010467@student.utem.edu.my

Abstract — TensorFlow Playground is an interactive web application that allows users to test artificial intelligence algorithms using Google's TensorFlow machine learning frameworks. TensorFlow Playground allows users who aren't familiar with high-level mathematics or coding to experiment with neural networks for deep learning and other machine learning applications. Neural network processes are interactive and visually depicted in the Playground. The TensorFlow Playground is a ds3.js-based Typescript web app. The user may utilise the web to run tiny neural networks, perform tests, and get results. It comes with an open source library that is targeted to educational purposes. The library is speedy, adaptive, expandable, and excellent for learning, research, and production despite its small size.

Keywords — Hidden layer, Neural network,

1. Introduction

Tensorflow playground is an interactive web programme to create and run a neural network model. It is an educational visualization platform to help learners in understanding the core concepts of deep learning. [4] In this case study, the tensorflow playground platform is used to study the hidden layer and its effects on the neural network model in detail.[1] A Hidden layer in an artificial neural network is a layer in between input layers and output layers, where artificial neurons take in a set of weighted inputs and produce an output through an activation function. Hidden layers are defined in several different ways. The weighted inputs can be assigned randomly or fine-tuned and calibrated through a process called backpropagation. The neurons in hidden layer represent the biological neuron of human brain - where the biological neurons takes in probabilistic input signals, process the information and produce signals as output corresponding to the biological neuron's axon. [3] Hidden layers are required if the data are to be separated non-linearly and this is not achievable with single layer network. Therefore, it is important to understand and explore the impacts of hidden layer on non-linear problem.

2. Step by step learning

In order to have a good comprehension about the effects of the hidden layer on the neural network model, hidden layer using the stimulation tensorflow playground. The first step is identifying the problem type and the preferred dataset. Then, fix the properties that need to feed in and the number of hidden layers. Next, set other parameters of neuron network model such as the activation, regularisation and regularisation rate. The following step is to choose the hidden layer and run the neural network model. Stop the model when the desired model performance for both testing and training data is obtained. Observe the number of epochs to reach the desired output. Following, the model is tested with different values of hidden layer while the number of epochs is taken into account and the output is observed for each hidden layer.

3. Analysis

It has been analyzed that hidden layers play a crucial role in the performance of neural network especially in the case of complex problems in which the time complexity and and accuracy become the main constraint. [3] However, the process on how to decide the number of hidden layer is depending on the problem and what the dataset used. For circle dataset, the optimal hidden layer that can be used is 2. As for exclusive or dataset, the optimal hidden layer is 3 with 3 neuron in every hidden layer. For each hidden neuron added will increase the number of weights, thus it is recommended to use the least number of hidden neurons to accomplish the task. Using more hidden neurons than required will add more complexity.

4. How Analysis Conducted

The first step of how we conduct this analysis is by identifying the importance of hidden layers and the effect of hidden layers on neural network model. The hidden layer is tested out by utilizing the simulation tensorflow playground in order to get a better understanding of the implications of hidden layer on model behaviour. Next, we must set a clear measurement priorities [4]. That means that we need to decide what to measure and how to measure it. To make a decision on what to measure, values of all parameters need to be determined for demonstration. As we are only focusing on hidden layers, the only parameter that changes is the value of hidden layers and hidden neurons for each hidden layer. The next step will be on to decide whether hidden layers are required in the data, because hidden layers are only used if and only if the data must be separated non-linearly. Experiments have shown us that the optimum number of neurons in a hidden layer can be determined by:

No. of Neurons =

$$\frac{Trading\ Data\ Samples}{Factor * (Input\ Neurons + \ Output\ Neurons)}$$

The factor is used to prevent over-fitting and it is a number between 1–10. It is also crucial that we need to know the number of hidden neurons in each hidden layer, especially before we start collecting data, because the quantity of hidden neurons will either validate or invalidate our analysis later on. The final step will be on testing and evaluation. Multiple choices of of number of hidden layers and hidden neurons are tested for two data set, exclusive OR and gaussian in order to find an optimal value of hidden layer and hidden neurons needed to reach desired neural network model performances.

5. Detail description

Hidden layer is important for artificial neural network model where the data for the problem needs to be separated non-linearly. The number of hidden layers used for a problem affects the accuracy and time complexity. Time complexity here refers to time taken for the network to learn the whole problem and produce required results in a required time span. The hidden layers are often represented as using the notation of the lowercase h for example h1, h2, etc. [3] A network having large number of hidden layers result in very large time complexity except in case of overfitting. Whereas a network having smaller number of hidden layers show a very satisfactory time constraints except in case of underfitting.

In terms of accuracy, a network having large number of hidden layers normally result in high accuracy even for large and complex problems. High accuracy can only be achieved if the network are able to understood the whole problem without having overfitting and underfitting conditions.

During training, the hidden layers transform the inputs into something that the output layer can use. Then, the output layer transforms the hidden layer activations into desired output. Each hidden layer in the artificial neural network model extract the implicit information of features available. Hidden layers capture more and more complexity with every layer by discovering the relationship between features in the input.

Since we are focusing on the hidden layer only, we will be using all default values for other parameters in this demonstration. The only parameter to be changed is the hidden layer. To compare the result, we will be using 1000 epoch for each run for different number of hidden layers and hidden neurons. The parameters set as constant are activation function (tanh) , regularization (none), regularization rate (0), problem type (classification). Next, the data used (Exclusive OR), ratio of training to test (50%), noise (5), batch size (10), and the learning rate (0.1).

The diagrams below show the example of the constant variables with different hidden layer, 1 and 2 respectively.

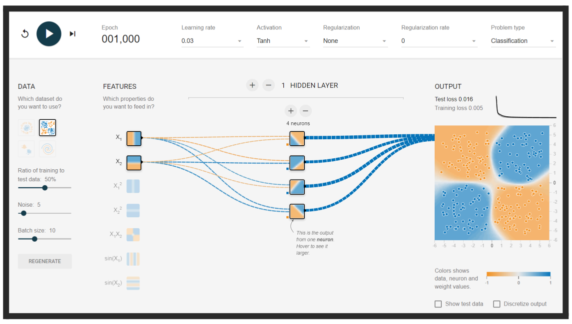

Figure 1 shows 1 hidden layer with other parameters at constant

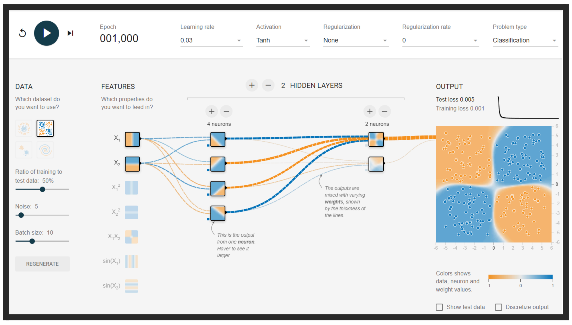

Figure 2 shows 2 hidden layer with other parameters at constant

It can be seen from the figures above that a network with a higher number of hidden layers (Figure 2) resulted in larger time complexity. However, the network with higher number of hidden layers produces higher accuracy which can be proved by the value of test loss and training loss. The performance of the neural network model will perform very well when the value of training loss and testing loss are very close.

6. Output Explanation

When testing the model with different number of hidden layers, we set the epoch to a 1000 and record the output as shown below;

First output is for circle data set with 1 hidden layer and 3 neurons. The test loss here is 0.135 and training loss is 0.094.

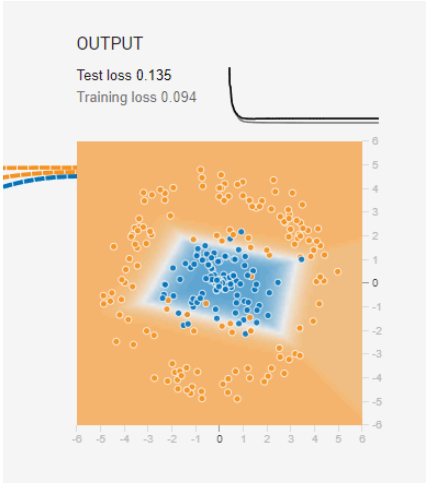

Figure 3 shows output for circle data set with 1 hidden layer.

Second output is for circle data set with 2 hidden layers containing 3 neurons and 2 neurons respectively. The test loss here is 0.121 and training loss is 0.079.

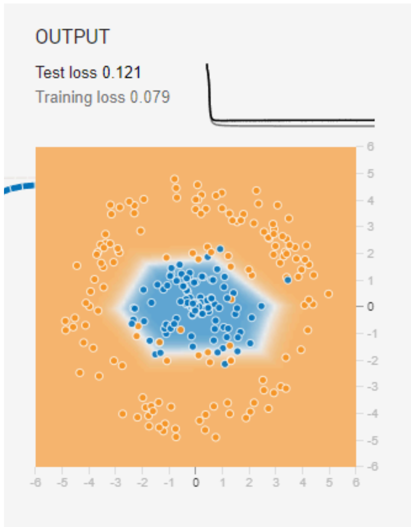

Figure 4 shows output for circle data set with 2 hidden layer.

Third output is for circle data set with 3 hidden layers containing 3 neurons, 2 neurons and 2 neurons respectively. The test loss here is 0.107 and training loss is 0.073.

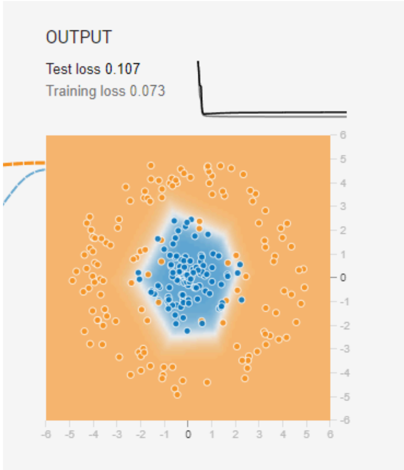

Figure 5 shows output for circle data set with 3 hidden layers.

Here, we can see that for circle data set, the most performed is with 3 hidden layers since the test loss and training loss have the closest value to each other which is 0.034.

Fourth output is for exclusive or data set with 1 hidden layer containing 3 neurons. The test loss here is 0.141 and training loss is 0.135.

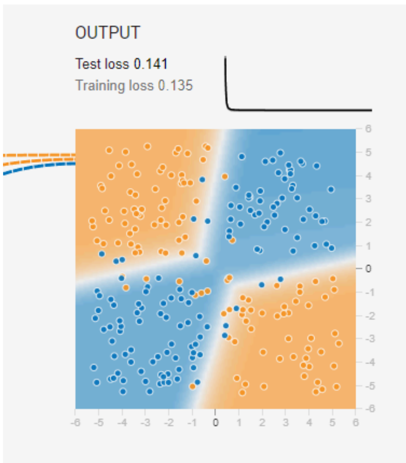

Figure 6 shows output for exclusive or data set with 1 hidden layer.

Fifth output is for exclusive or data set with 3 hidden layers containing 3 neurons and 2 neurons respectively. The test loss here is 0.151 and training loss is 0.115.

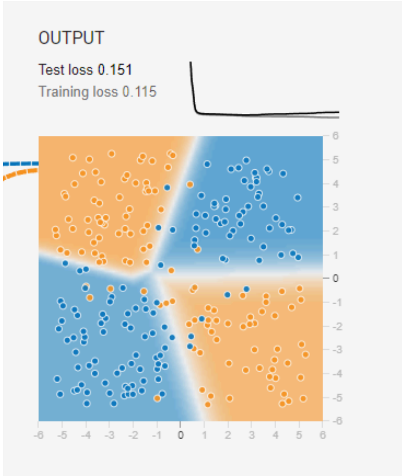

Figure 7 shows output for exclusive or data set with 2 hidden layers.

Fifth output is for exclusive or data set with 3 hidden layers containing 3 neurons ,2 neurons and 2 neurons respectively. The test loss here is 0.390 and training loss is 0.387.

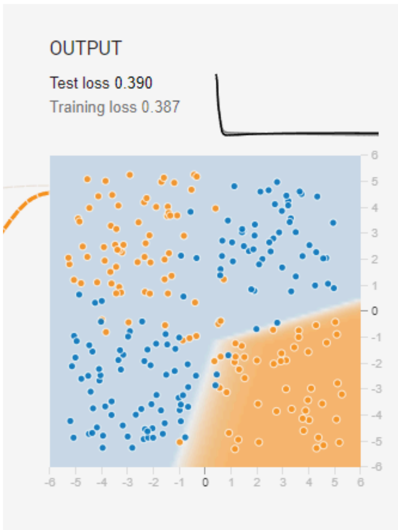

Figure 8 shows output for exclusive or data set with 3 hidden layer.

Here, we can see that for exclusive or set, the most performed is with 3 hidden layers since the test loss and training loss have the closest value to each other which is only 0.003.

7. Conclusion

After applying different number of hidden layer on the model, it has been concluded that different datasets have different optimal number of hidden layer. The circle dataset has an optimal hidden layers of 2 whereas the optimal hidden layer for exclusive or dataset is 3. An optimum number of hidden layers can reduce time complexity while having a high accuracy. Hence, it is important to standardize the number of hidden layers that are required, but the approximaton will be depending the type of database used.

References

[1] *What is a hidden layer?*. (Sep 5, 2018). Retrieved from https://www.techopedia.com/definition/33264/hidden-layer-neural-networks#:~:text=A%20hidden%20layer%20in%20an,output%20through%20an%20activation%20function.

[2] Ahmed, G. (June 27, 2018). *How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?.* Retrieved from https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e#:~:text=In%20artificial%20neural%20networks%2C%20hidden,get%20the%20best%20decision%20boundary.

[3] M. Uzair and N. Jamil, "Effects of Hidden Layers on the Efficiency of Neural networks," *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318195.

[4] Bharath, K. (Sep 30, 2020). *Neural Networks Made Fun With TensorFlow Playground!*. Retrieved from https://towardsdatascience.com/neural-networks-made-fun-with-tensorflow-playground-4e681a0c4529.

[5] Sato, K. (July 26, 2016). *Understanding Neural Networks With Tensorflow Playground*. Retrieved from https://cloud.google.com/blog/products/ai-machine-learning/understanding-neural-networks-with-tensorflow-playground.

APPENDIX

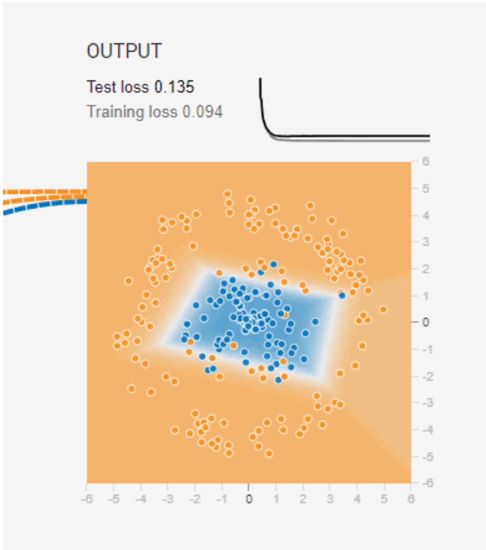

Figure 3 shows output for circle data set with 1 hidden layer

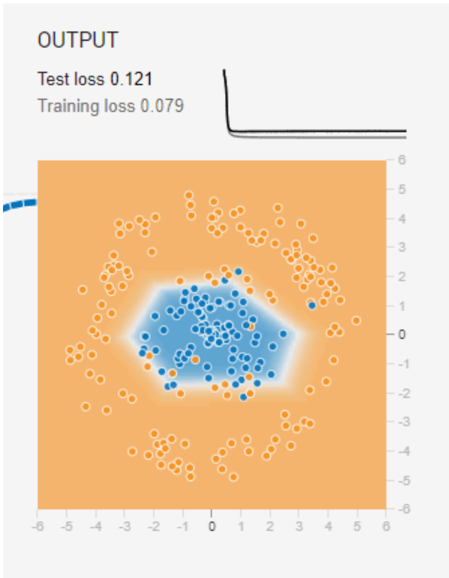

Figure 4 shows output for circle data set with 2 hidden layer.

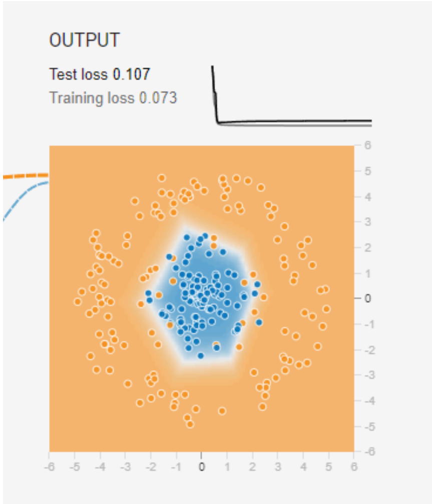

Figure 5 shows output for circle data set with 3 hidden layers.

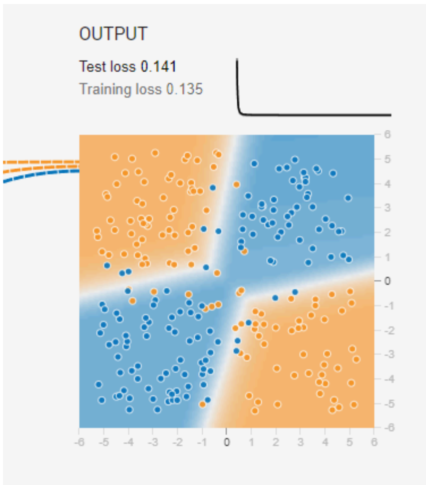

Figure 6 shows output for exclusive or data set with 1 hidden layer.

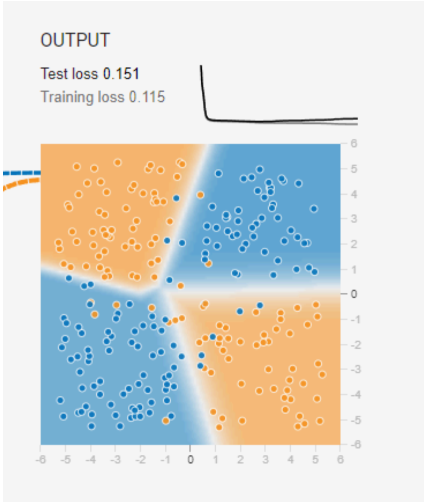

Figure 7 shows output for exclusive or data set with 2 hidden layers.

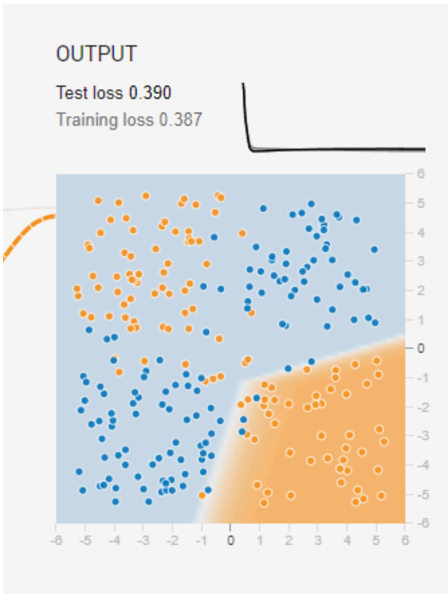

Figure 8 shows output for exclusive or data set with 3 hidden layer.