## Programming Test - Learning Activations in Neural Networks

As the dataset was different, the approach to train the model was also different. Initially, the dataset is being divided into test and train dataset for training as well as validating the model. The algorithm is used in the assignment was back-propagation.

The deep learning models can be majorly divided into two networks. i.e.: -

- Feed-forward network.
- Back-propagation network.

The feed forward network usually propagates in a single direction. The weights are usually not updated in this kind of networks and also the accuracy is not good.

Whereas, in back-propagation, the weights and learning rate are being updated after every epoch. This is usually done by the following formula:

$$\frac{\partial E_2}{\partial y2} = \frac{\partial E_2}{\partial y2_{\mathrm{fianl}}} \times \frac{\partial y2_{\mathrm{final}}}{\partial y2}$$

$$= \frac{\partial(\frac{1}{2}(T2 - y2_{\mathrm{final}})^2)}{\partial y2_{\mathrm{final}}} \times \frac{\partial y2_{\mathrm{final}}}{\partial y2}$$

$$= 2 \times \frac{1}{2}(T2 - y2_{\mathrm{final}}) \times (-1) \times \frac{\partial y2_{\mathrm{final}}}{\partial y2} \dots\dots\dots (21)$$

$$y2_{\mathrm{final}} = \frac{1}{1 + e^{-y2}} \dots\dots\dots\dots\dots\dots\ (22)$$

$$\frac{\partial y2_{\mathrm{fianl}}}{\partial y2} = \frac{\partial(\frac{1}{1 + e^{-y2}})}{\partial y2}$$

$$= \frac{e^{-y2}}{(1 + e^{-y2})^2}$$

$$= e^{-y2} \times (y2_{\mathrm{final}})^2 \dots\dots\dots\dots\dots (23)$$

$$y2_{\mathrm{final}} = \frac{1}{1 + e^{-y2}}$$

$$e^{-y2} = \frac{1 - y2_{\mathrm{final}}}{y2_{\mathrm{final}}} \dots\dots\dots\dots\dots (24)$$

The new weight is been calculated by the subtraction of old weight and integration of partial derivation loss function. This is a hyper parameter and is usually tuned during the training of the network. The back propagation is usually preferred over feed forward propagation because of the property of tuning the hyper parameters during the training.

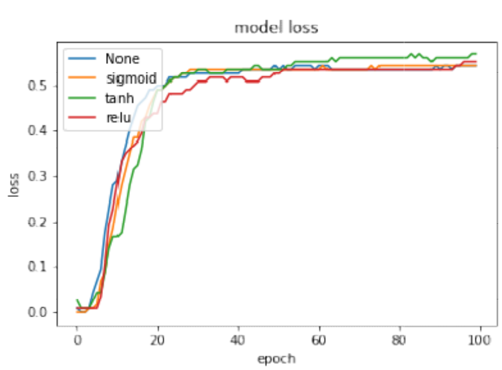
According the problem statement, a program is being designed using keras for identifying the best activation function for the dataset without using brute force and grid search.

Initial parameters used: -

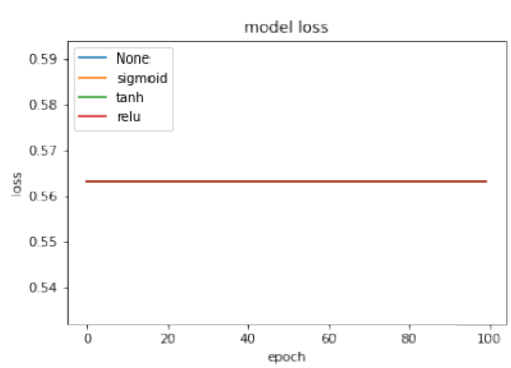
1. K0 = 0.04
2. K1 = 0.02
3. Batch Size = 128
4. Epochs = 20
5. No. of Steps = 8
6. Dataset : Bank-Note Best Performing Activation function : relu having 57.52% accuracy
7. Dataset : Iris Best Performing Activation function : sigmoid having 90.00% accuracy
8. Dataset : Breast Cancer Best Performing Activation function : tanh having 65.50% accuracy
9. Dataset : MNIST Best Performing Activation function : None having 34.03% accuracy
10. f1_val = 0.9186602870813397 (MNIST Dataset)

| Dataset | Test Loss | Train Loss |
|---|---|---|
| Bank Note Dataset | 5.4067e-08 | 5.0635e-08 |
| Iris Dataset | 0.5839 | 0.7339 |
| Breast Cancer Dataset | 6.6708 | 6.3988 |
| MNIST Dataset | 11.7976 | 12.9060 |

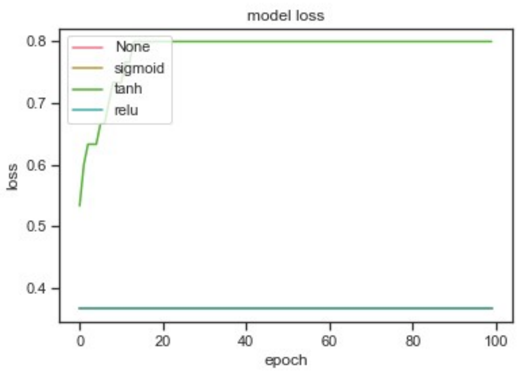
The results of the activation function are given below: -

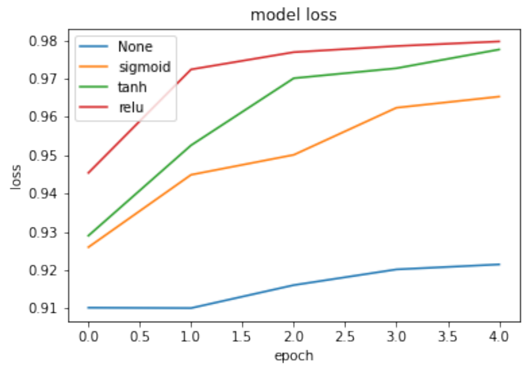



Wisconsin Breast Cancer Dataset




Bank Note Authentication Dataset




Iris Dataset MNIST Dataset

As it can be observed from the figures, different activation functions perform in a different way in the dataset. The activation functions used in the test are as follows: -

- None
- Sigmoid
- ReLU
- TanH

The model also uses none of the activation function to observe the parameters and analysis the performance of the model.

The sigmoid/logistic activation function is the defined as: -