# Learning Activations in Neural Networks

Test Assignment for Data Science Intern
Happymonk Technology Pvt. Ltd.

-GOKULAVASAN K R

**PROBLEM STATEMENT :**

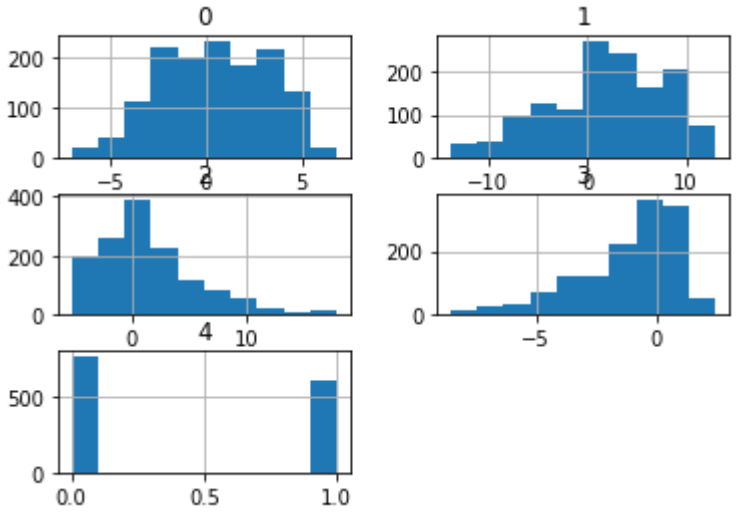
The purpose is to build a classification model using neural networks that classifies the target variable accurately. The neural network must consist of a 1-hidden layer that adapts to the most suitable activation function according to the bank note dataset.

**DATA SET:**

The dataset used for the problem is data_banknote dataset. The number of instances in the data set is 1372, and the number of variables is 5. The objective here is to detect the fraudulent notes accurately based on the independent features.

**VISUALIZATION:**

1) **Histogram of the features that shows the distribution**




2) **Information about data**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1372 entries, 0 to 1371
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   0       1372 non-null   float64
 1   1       1372 non-null   float64
 2   2       1372 non-null   float64
 3   3       1372 non-null   float64
 4   4       1372 non-null   int64  
dtypes: float64(4), int64(1)
memory usage: 53.7 KB
```

### 3) Dataset

```
(1372, 5)
```

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | 3.62160 | 8.6661 | -2.8073 | -0.44699 | 0 |
| **1** | 4.54590 | 8.1674 | -2.4586 | -1.46210 | 0 |
| **2** | 3.86600 | -2.6383 | 1.9242 | 0.10645 | 0 |
| **3** | 3.45660 | 9.5228 | -4.0112 | -3.59440 | 0 |
| **4** | 0.32924 | -4.4552 | 4.5718 | -0.98880 | 0 |

## ANN

### 1) IMPLEMENTING FOR 15 EPOCHS

```
35/35 - 0s - loss: 1.1369 - accuracy: 0.6208 - val_loss: 1.2108 - val_accuracy: 0.6145 - 148ms/epoch - 4ms/step
Epoch 10/15
35/35 - 0s - loss: 1.0064 - accuracy: 0.6545 - val_loss: 1.0730 - val_accuracy: 0.6436 - 265ms/epoch - 8ms/step
Epoch 11/15
35/35 - 0s - loss: 0.8947 - accuracy: 0.6764 - val_loss: 0.9471 - val_accuracy: 0.6836 - 198ms/epoch - 6ms/step
Epoch 12/15
35/35 - 0s - loss: 0.7989 - accuracy: 0.7019 - val_loss: 0.8495 - val_accuracy: 0.7055 - 137ms/epoch - 4ms/step
Epoch 13/15
35/35 - 0s - loss: 0.7170 - accuracy: 0.7338 - val_loss: 0.7571 - val_accuracy: 0.7345 - 141ms/epoch - 4ms/step
Epoch 14/15
35/35 - 0s - loss: 0.6455 - accuracy: 0.7603 - val_loss: 0.6842 - val_accuracy: 0.7491 - 270ms/epoch - 8ms/step
Epoch 15/15
35/35 - 0s - loss: 0.5833 - accuracy: 0.7794 - val_loss: 0.6155 - val_accuracy: 0.7673 - 278ms/epoch - 8ms/step
9/9 [==============================] - 0s 6ms/step
Accuracy: 0.767
```

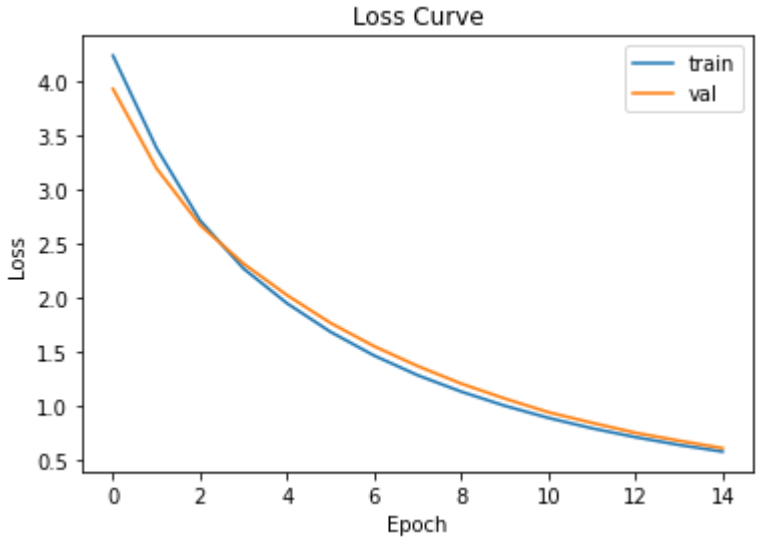
### REPORT FOR 15 EPOCHS

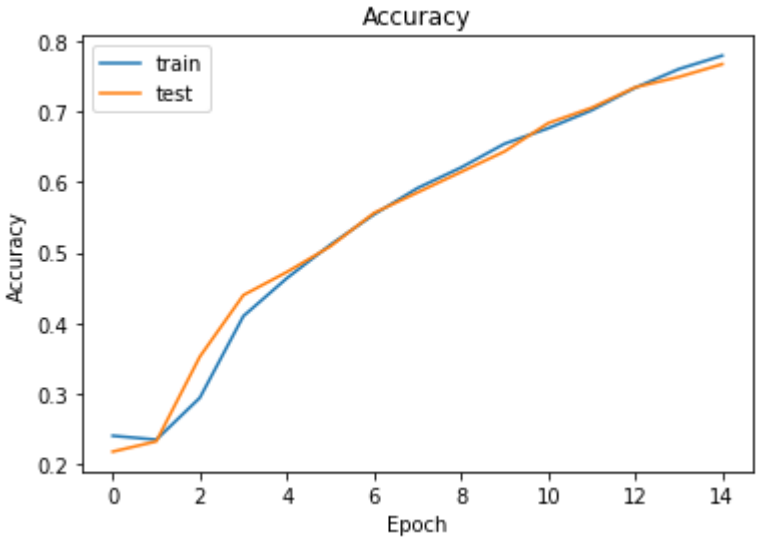
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 0.77 | 0.79 | 152 |
| 1 | 0.73 | 0.76 | 0.75 | 123 |
| accuracy | | | 0.77 | 275 |
| macro avg | 0.77 | 0.77 | 0.77 | 275 |
| weighted avg | 0.77 | 0.77 | 0.77 | 275 |

## TRAIN AND TEST ACCURACY FOR 15 EPOCHS

```
Train: 0.789, Test: 0.767
```

## LOSS AND ACCURACY CURVE WITH RESPECT TO THE EPOCHS




## SUMMARY FOR 15 EPOCHS

```
9/9 - 0s - 44ms/epoch - 5ms/step
Accuracy: 0.767273
Precision: 0.728682
Recall: 0.764228
F1 score: 0.746032
Cohens kappa: 0.531491
ROC AUC: 0.766982
[[117  35]
 [ 29  94]]
```

## 2) IMPLEMENTING FOR 25 EPOCHS

```
Epoch 19/25
35/35 - 0s - loss: 0.1337 - accuracy: 0.9881 - val_loss: 0.1249 - val_accuracy: 0.9818 - 141ms/epoch - 4ms/step
Epoch 20/25
35/35 - 0s - loss: 0.1241 - accuracy: 0.9891 - val_loss: 0.1170 - val_accuracy: 0.9818 - 129ms/epoch - 4ms/step
Epoch 21/25
35/35 - 0s - loss: 0.1157 - accuracy: 0.9936 - val_loss: 0.1105 - val_accuracy: 0.9818 - 211ms/epoch - 6ms/step
Epoch 22/25
35/35 - 0s - loss: 0.1084 - accuracy: 0.9936 - val_loss: 0.1046 - val_accuracy: 0.9818 - 122ms/epoch - 3ms/step
Epoch 23/25
35/35 - 0s - loss: 0.1020 - accuracy: 0.9954 - val_loss: 0.0993 - val_accuracy: 0.9818 - 153ms/epoch - 4ms/step
Epoch 24/25
35/35 - 0s - loss: 0.0964 - accuracy: 0.9973 - val_loss: 0.0949 - val_accuracy: 0.9818 - 165ms/epoch - 5ms/step
Epoch 25/25
35/35 - 0s - loss: 0.0914 - accuracy: 0.9973 - val_loss: 0.0903 - val_accuracy: 0.9818 - 131ms/epoch - 4ms/step
9/9 [==============================] - 0s 2ms/step
Accuracy: 0.982
```

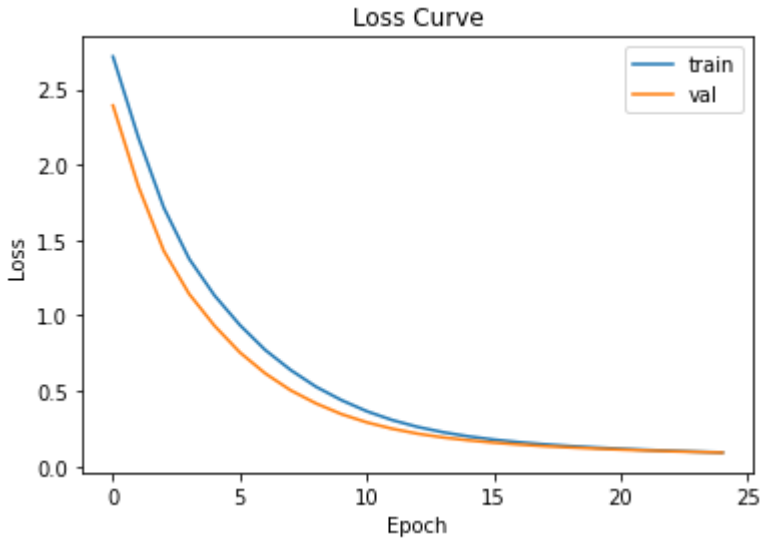
## REPORT FOR 15 EPOCHS

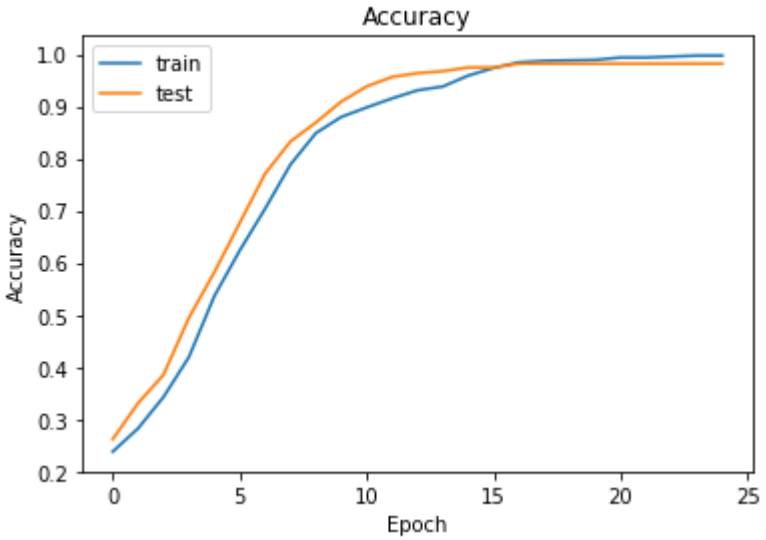
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.98 | 0.98 | 152 |
| 1 | 0.98 | 0.98 | 0.98 | 123 |
| accuracy | | | 0.98 | 275 |
| macro avg | 0.98 | 0.98 | 0.98 | 275 |
| weighted avg | 0.98 | 0.98 | 0.98 | 275 |

## TRAIN AND TEST ACCURACY FOR 25 EPOCHS

```
Train: 0.997, Test: 0.982
```

## LOSS AND ACCURACY CURVE WITH RESPECT TO THE EPOCHS




## SUMMARY FOR 25 EPOCHS

```
9/9 - 0s - 40ms/epoch - 4ms/step
Accuracy: 0.981818
Precision: 0.975806
Recall: 0.983740
F1 score: 0.979757
Cohens kappa: 0.963256
ROC AUC: 0.982001
[[149   3]
 [  2 121]]
```

**CONCLUSION:**

After increasing the number of epochs we can see a significant improvement in the training and test data accuracy.

Increasing the number of epochs above 25 makes the model overfitting since all evaluation indicators will be 1.

Hence all required solutions are proposed

**GITHUB LINK:**

https://github.com/GOKU-GO/Happy-monk-assignment