Bank note authentication

Includes following algorithm:

Developing ANN on Banknote authantication includes following steps

1) Acquire the dataset

2)importing required libraries

3)Importing the data

4)Identifying and handling the missing values

5) Extract the independent variables from data set and naming it as y

6)Extract the dependent variable from data set and naming it as x

7)Encoding the categorical variables

8) splitting the dataset into train and test set

9)Feature scaling by standardization or normalization

10)selecting the model

11)creating the no.of hidden layers based on input and setting the output neuron to because the data

12)compiling the model based on optimizer

13)Training the model using train data set using model.

14) Predicting the values based on the test set

15)Finding the accuracy by using confusion matrix, f1_score,precesion,accuracy,recall

14)plot of the loss function vs. epochs

The description about the work

The data set Bank note is a csv file has downloaded from the kaggle site. It is imported using the pandas library and all the preprocessing technique such as, finding missing values using (df.info()) , but there is absence of null value and there is no need to do any dropping or adding values in that place.

The data frame containing 4 independent features and one dependent feature are split into separate as X and y sets. Next step is dividing the X and y using train and test sets using train_test_split() module.we will import “StandardScaler” class of the sci-kit-learn library in order to standardize the independent variables of a dataset within a specific range.

We have selected the “sequential model”. In this case, we will use one hidden layer with 8 nodes and one output layer (chosen arbitrarily). We will use the “ReLU” activation function in the hidden layer and the “he_normal” weight initialization, as together, they are a good practice. The output of the model is a sigmoid activation for binary classification and we will minimize binary cross-entropy loss.

The multilayer ANN Using a 1-hidden layer neural network model that adapts to the most suitable activation function("ReLU") according to the data-set because by using the “ReLU” function over other activation functions is that it does not activate all the neurons at the same time.For the negative input values, the result is zero, that means the neuron does not get activated. Since only a certain number of neurons are activated, the ReLU function is far more computationally efficient when compared to the sigmoid and tanh function.

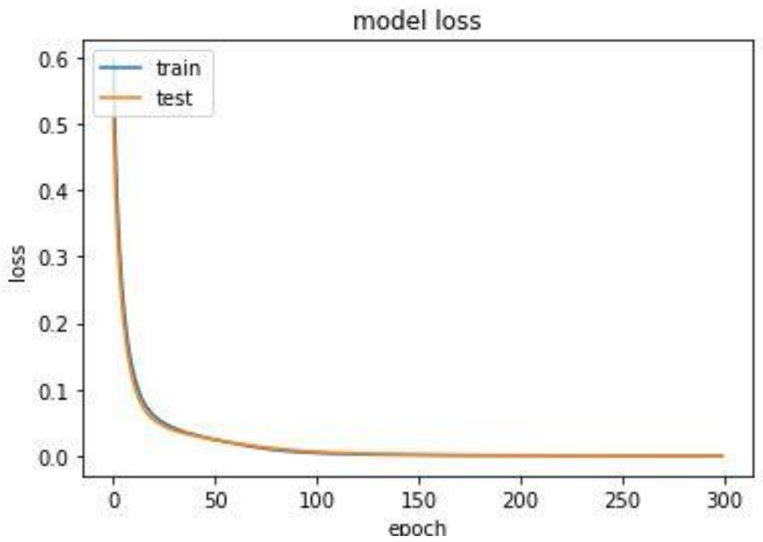
To fit the model for 300 training epochs (chosen arbitrarily) with a batch size of 10.

At the end of training, i have evaluated the model’s performance on the test dataset and report performance as the classification

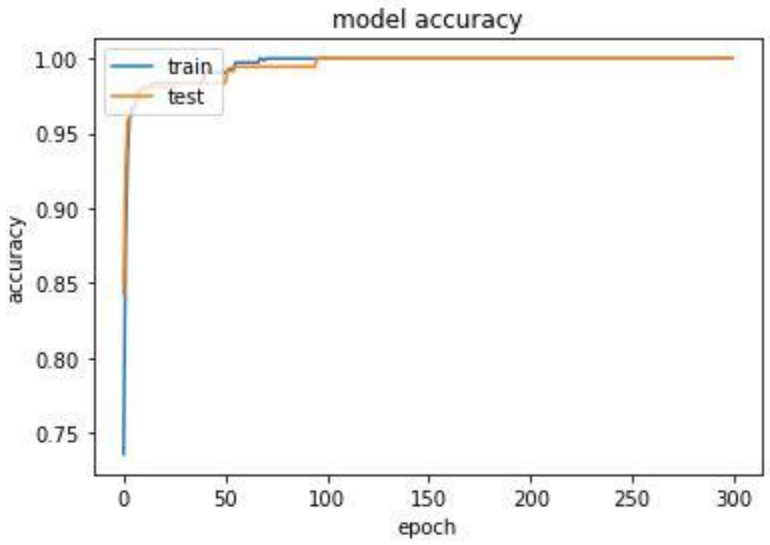
confusion matrix-

```
      array([[158,0],
             [0,117]],dtype=int64)
with accuracy--1.000000
     precesion-1.000000
     recall------1.000000
    F1-score-----1.000000
```

Plot of the loss function vs. epochs



Plot of accuracy vs epochs


model accuracy
train
test
accuracy
epoch