

Programming Test :Learning Activation in Neural Network

Name – wajoud H Noorani

email – Wajoudnoorani59@gmail.com

This Technical Report is based on three data sets which are classified and analyzed with different activation functions: datasets (iris, mnist , bank notes)

My Dataset :

iris - <https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data>

mnist – fom keras data set (from keras.datasets import mnist)

bank-note-

https://raw.githubusercontent.com/jbrownlee/Datasets/master/banknote_authentication.csv

Common Setup for the Datasets :

Since Few steps are common for the initial stage of this algorithm ill explain the common step below

1 – To upload any dataset we can use Pandas library :

Eg – dataset = pd.read_csv(url , or direct data set name)

2 – To select the paramentrs for k0 , k1 by dividing the data set with respect to its contains

in my code i have paramentrs k0 ,k1 as x , y

3 – Since we have words or string in our data set we need to convert them into arrays or number to classify it

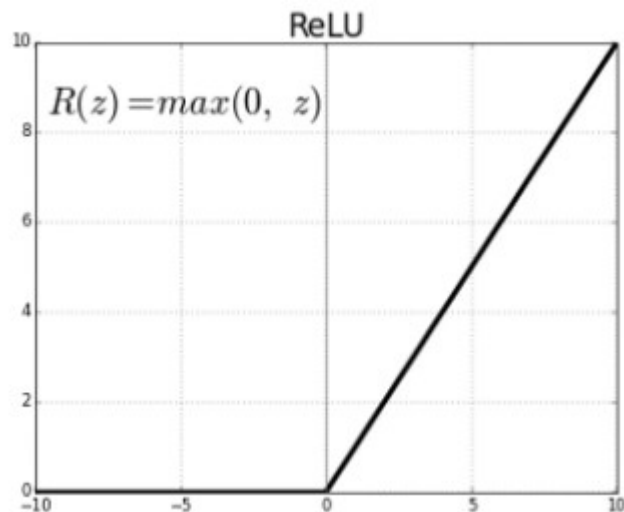
i have used Labeled encoder function to transform the parametres

4 – To classify any model we need to split the data set into training , testing .

In my code i have used from sklearn , train test-split model with test size as 20 % on any random sample

Iris Dataset

About the dataset, the iris dataset consists of 3 different types of iris plants including the parameters of the plants. We consider 1 parameter which includes all the different parameters of the plants and 2 parameters the names of the plant. To classify the data set I used Rectified Linear unit(ReLu) in the input activation function and softmax as the output layer activation function. The Below plot represents the Relu function.



Sampling of parameters :

```
#Parameter k0,k1
```

```
X= dataset.iloc[:,0:4].values
```

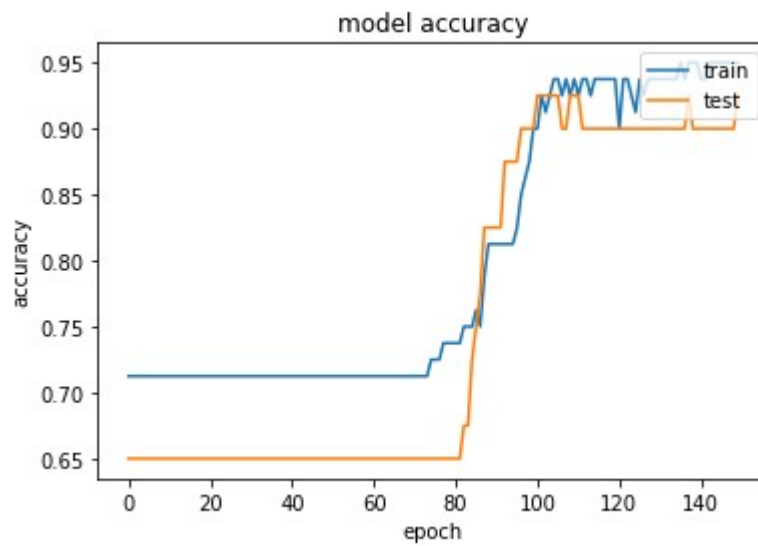
```
Y = dataset.iloc[:,4].values
```

F1 score -confussion matrix

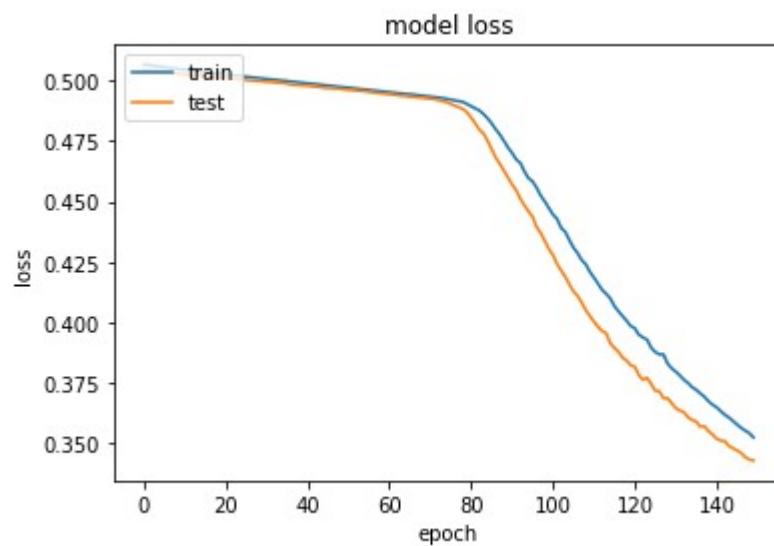
	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	0.00	0.00	0.00	13
2	0.32	1.00	0.48	6
accuracy			0.57	30
macro avg	0.44	0.67	0.49	30
weighted avg	0.43	0.57	0.46	30

```
[[11  0  0]
 [ 0  0 13]
 [ 0  0  6]]
```

Model Accuracy



Model loss

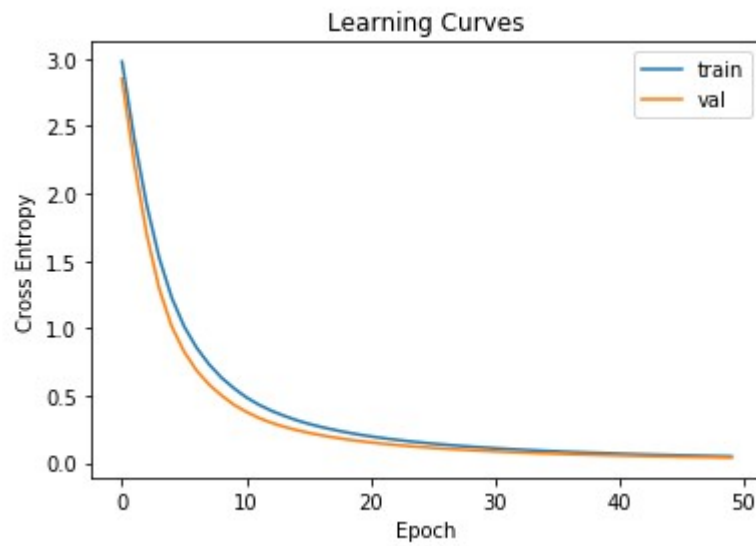


Bank Note Data set

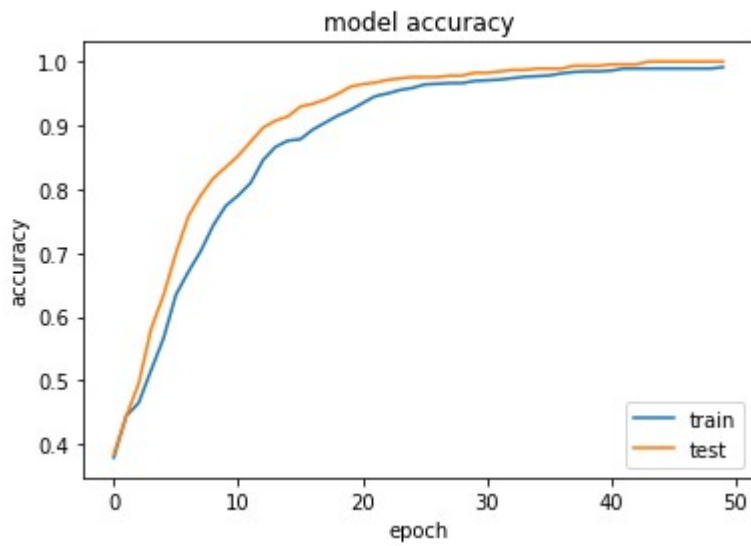
Sampling parametrs

```
X = df.values[:, :-1]  
y = df.values[:, -1]
```

Learning curve



Model accuracy



Mnist Dataset

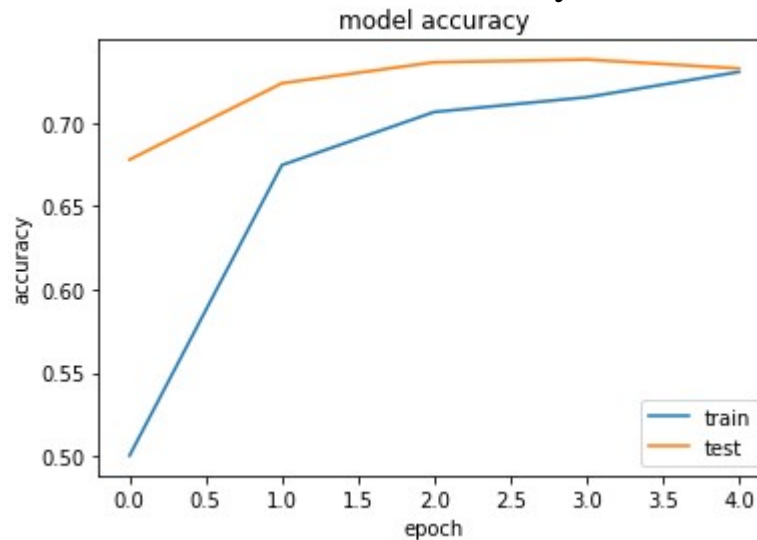
This data set consists of an image of a different handwritten number. Which we classified. To classify this dataset I used the sigmoid activation function for input hidden layer and softmax for output activation function with input size (28,28)

F1 Accuracy :

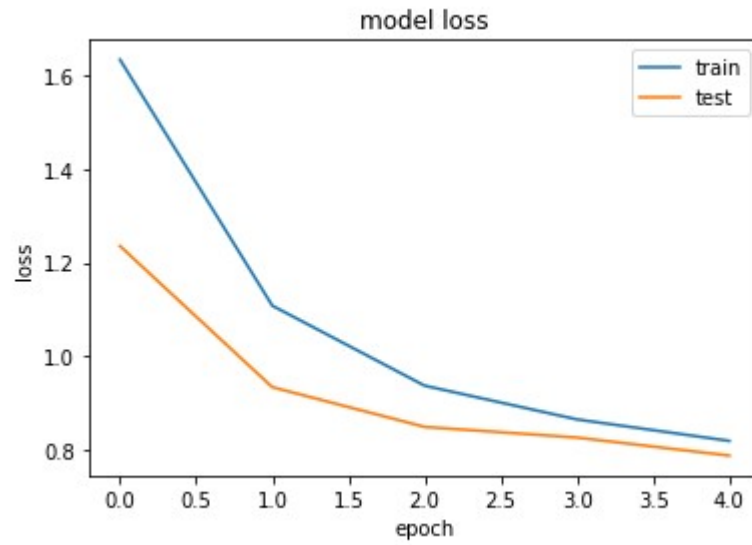
precision recall f1-score support

0	0.87	0.93	0.90	980
1	0.92	0.91	0.92	1135
2	0.80	0.81	0.80	1032
3	0.67	0.83	0.74	1010
4	0.46	0.83	0.59	982
5	0.71	0.61	0.66	892
6	0.72	0.95	0.82	958
7	0.86	0.88	0.87	1028
8	0.80	0.54	0.64	974
9	0.50	0.00	0.01	1009
accuracy			0.73	10000
macro avg	0.73	0.73	0.69	10000
weighted avg	0.73	0.73	0.70	10000

Model accuracy



Model lose :



All the 3 Code are uploaded in github

GitHub Link : <https://github.com/wajoud/Monk-AI-Programming-Test->