**DS 630 – Project 2**

**Submitted to: Dr. Robert Finn**

**Submitted by:**

***Lakshmi Vaidiyanathan***

***Shruti Arora***

**Objective:**

In this project we will employ a neural network of our choice to tackle

the classification of handwritten digits from the mnist.csv data set. We will use k-fold cross validation for evaluating the model. We will perform the modeling task twice: first on the raw mnist data and then a second time using PCA to preprocess the inputs. We will compare the results from these two methodologies. We will also information on the time required to train the networks using the system.time function in R.

**Assumptions :**

* **NN Model with PCA:**
  + **Only the grey pixel area is selected for the analysis from mnist data, hence resulting in 530 components for further analysis**
  + **261 components reporting 98% of the variance are taken for neural network after pca.**
  + **K cross validation is performed with K = 3**
* **NN Model without PCA:**
  + **All 784 components are used for neural net analysis.**
  + **K cross validation is performed with K = 3**

**Pre-requisites :**

1. **R pacakges to be installed are :**

library(readr)

library(caret)

library(nnet)

library(doParallel)

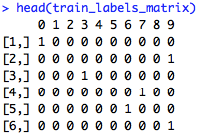
library(reshape)

library(devtools)

1. **Neural Network - raw mnist data:**

First, we split the data into 75% training and 25% testing using random selection. As next step, we normalize the data and apply normalization to both the training and testing datasets.

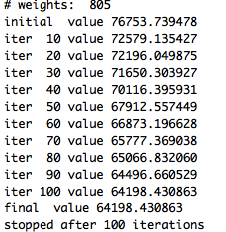
We created a train label matrix as follows:



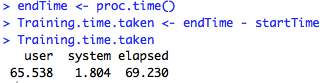
We used the “nnet” package from R to train the model. softmax = True tells the model that this is a classification problem. Maximum number of iterations has been set to 1000 by default.

First we train the model without k-cross validation. The model comprises 784 components, 1 hidden layer, 10 labels and has 805 weights.





The model ran for 50 seconds



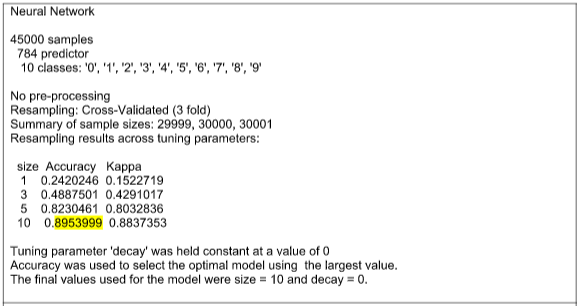
The accuracy is really low (20%) given that there is only one hidden node



**K-fold cross validation:**

As we used k-fold cross validation with k=3, we did not need to split the data into training and testing.

As the number of hidden nodes increases, the model accuracy also increases (89.5% for 10 layers). The k-fold cross validation model ran for 471 seconds.



**R code for all components:**

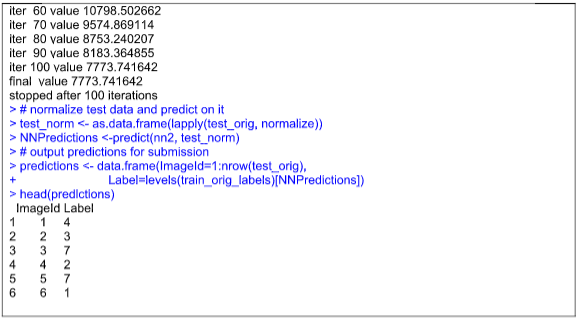


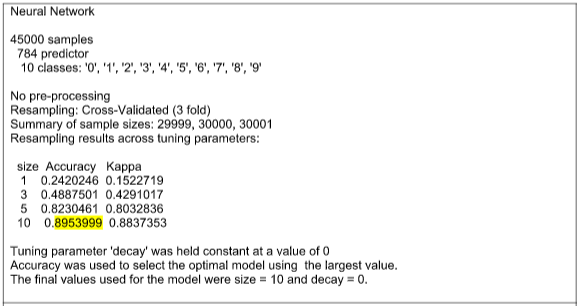


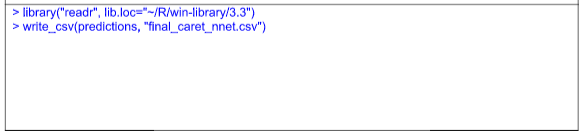


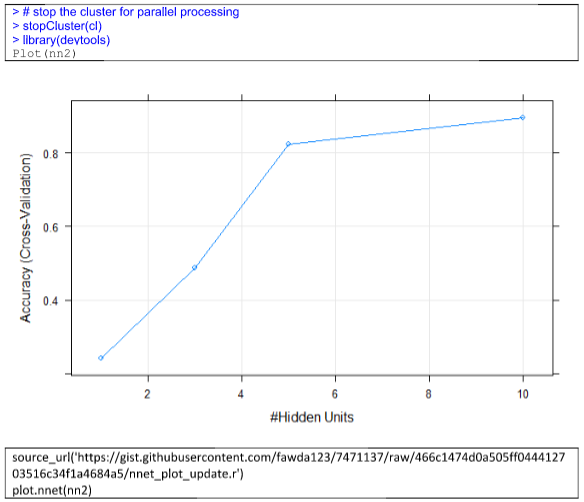


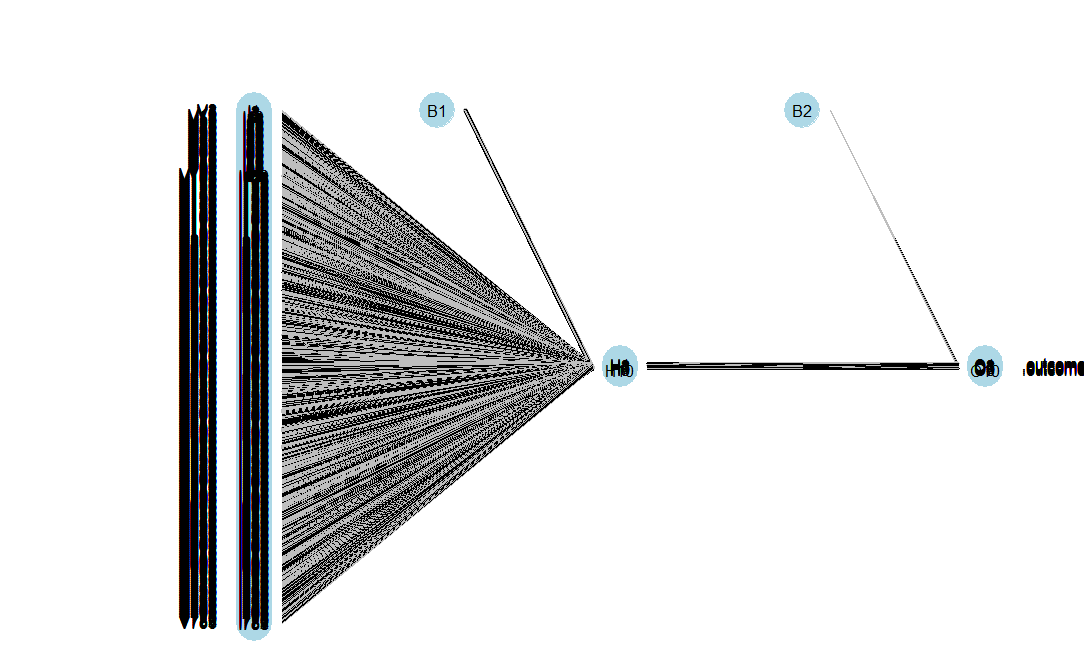












**As the number of hidden layers increases, accuracy of the model increases (89.5% for 10 hidden layers)**

**Predictions are in the .csv file below. Kappa being 0 indicates the prediction is random.**

Accuracy Kappa

0.9402685 0.0000000

****

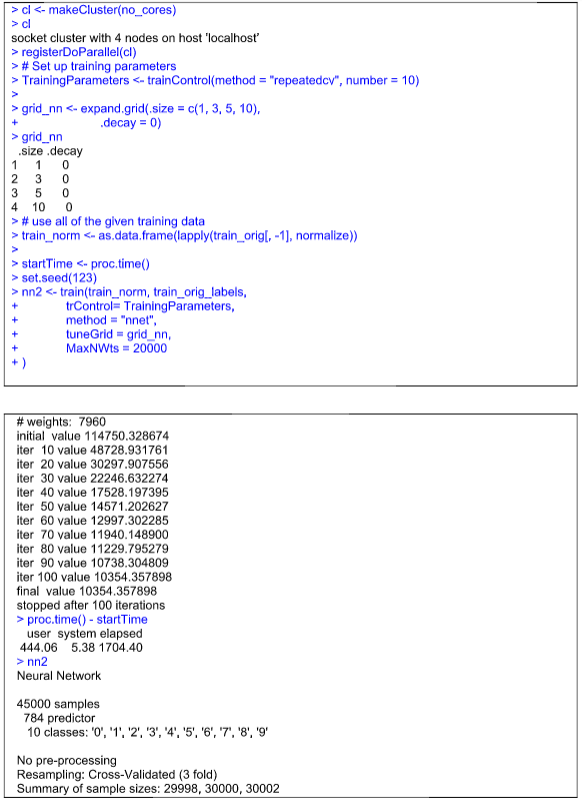
1. **NEURAL NETWORKS FOR FIRST 261 COMPONENTS**

Mnist data is ignored for the border, thus reducing the components to 530

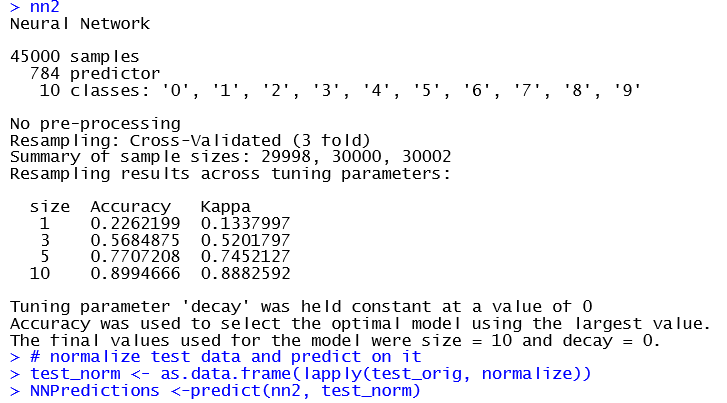
|  |
| --- |
| #ignoring the borders  mnist=mnist[,-(which((1:784)%%28<=2|(1:784)%%28>=26|1:784%/%28<=2|1:784%/%28>=26)+1)] |

From the 530 components, we ran the same model using the first 261 components. The model ran for 444 seconds.





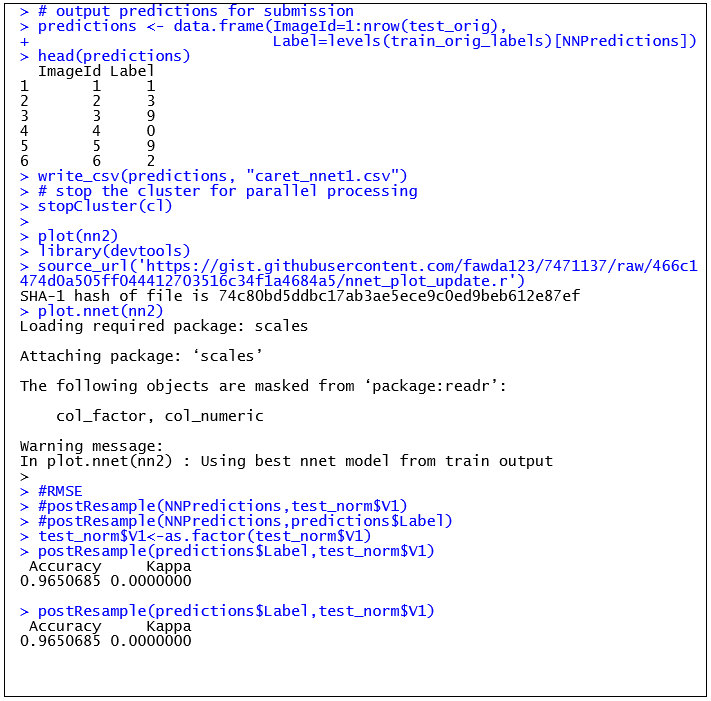
**As the number of layers increase, the accuracy increases to 89.9%**

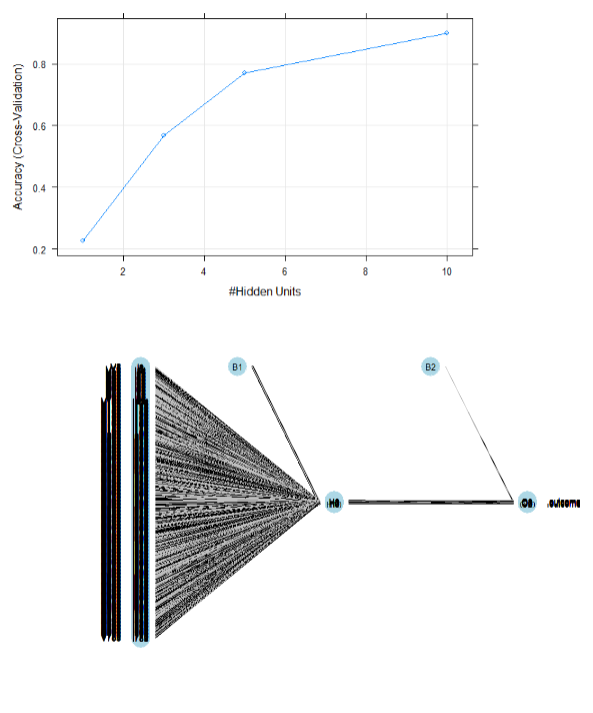


***Model Evaluation:***

*As the number of layers increase, the accuracy increases to 89.9%*

*Kappa is ‘0’ for this model too, indicating the chance being random. where RMSE, looks good.*





**CONCLUSION:**

Comparison between NN model without PCA and with PCA is as below.

Although the PCA model is better in time, it takes only the first necessary components, there are chances of missing important features needed for classification.

Accuracy looks similar with decimal difference.

For model without PCA, increasing the cross validation might help in improving the accuracy and have a better kappa .

|  |  |  |
| --- | --- | --- |
| **Factor** | **Mnistdata \_ 3 fold  cross validation without PCA** | **Mnistdata \_ 3 fold  cross validation with PCA (first 261 components)** |
| Difference | Border data for images are not removed | Border data removed |
| RMSE postResample(predictions$Label,test\_norm$V1) | Accuracy Kappa  0.9402685 0.0000000 | Accuracy Kappa  0.9650685 0.0000000 |
| Time  proc.time() - startTime | user system elapsed   471.98 14.50 2469.80 | user system elapsed   444.06 5.38 1704.40 |
| Summary  (nnet\_model) | size Accuracy Kappa   1 0.2420246 0.1522719  3 0.4887501 0.4291017  5 0.8230461 0.8032836  10 0.8953999 0.8837353 | size Accuracy Kappa   1 0.2262199 0.1337997  3 0.5684875 0.5201797  5 0.7707208 0.7452127  10 0.8994666 0.8882592 |