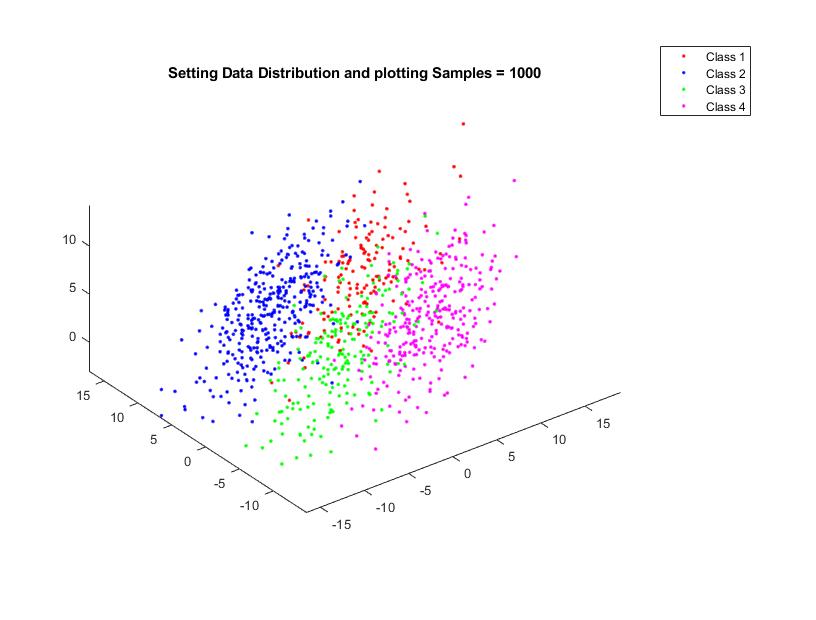
1. Question



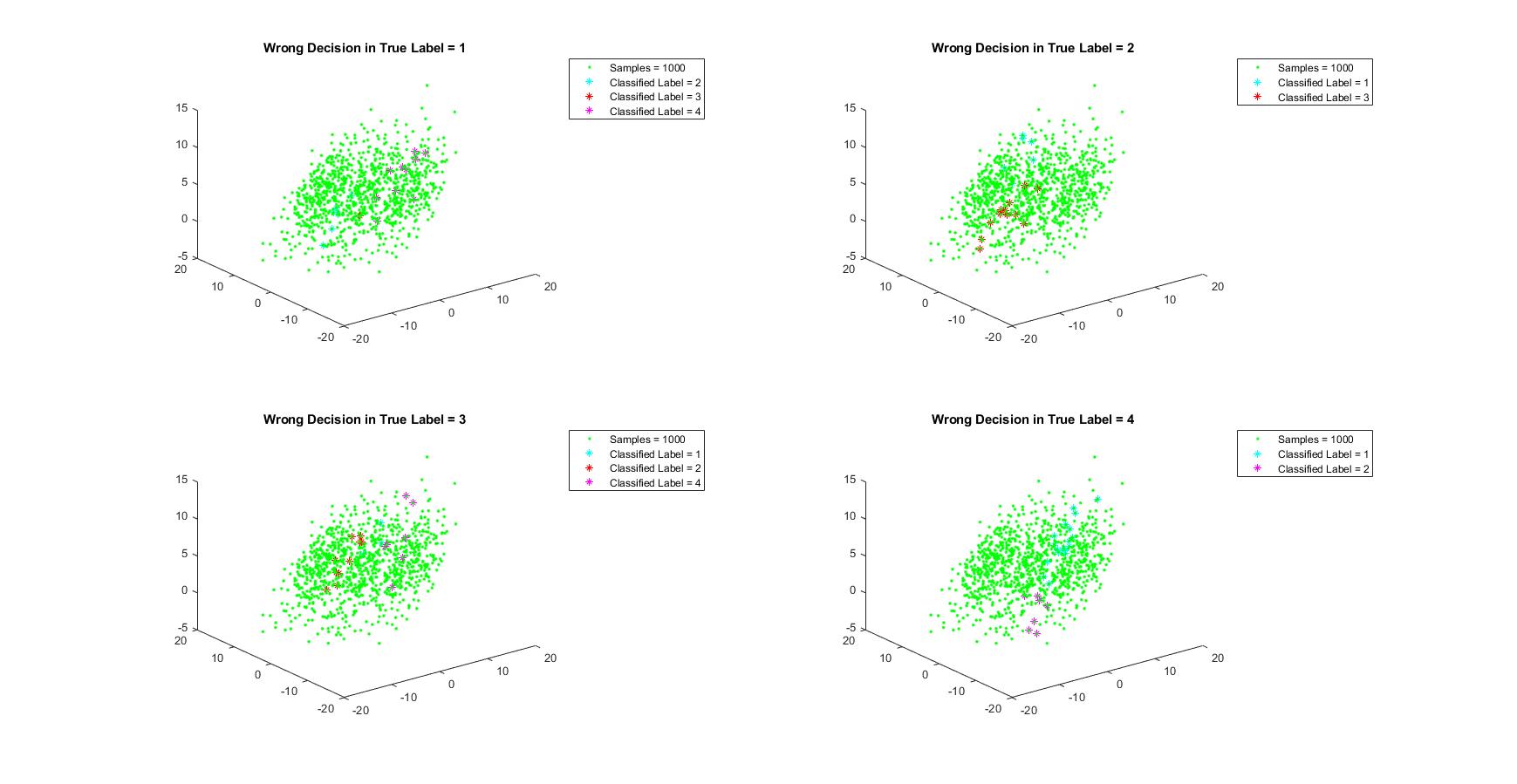
The distribution data is generated with

Prior = [0.15,0.35,0.2,0.3] where true sample number = [154,351,197,298]

Sigma = 0.8\*[5 1 2;1 5 0;2 0 3] /sqrt(2)\* covarianceVector for each class

covarianceVector = [1.3^2 0 0;0 1.2^2 0;0 0 1.4^2]

Apply this data distribution to the MAP classifier:



These plots show the MAP classification result

We get the confusion matrix = [137,6,1,10;6,333,12,0;3,10,177,7;20,0,7,271]

Where represent the right classification of

Class 1 = 137

Class 2 = 333

Class 3 = 177

Class 4 = 271 in 1000 true samples

Thus, we can calculate the whole MAP classification accuracy rate = (137+333+177+271) /1000 = 91.8%

And we get the decision error of each class is

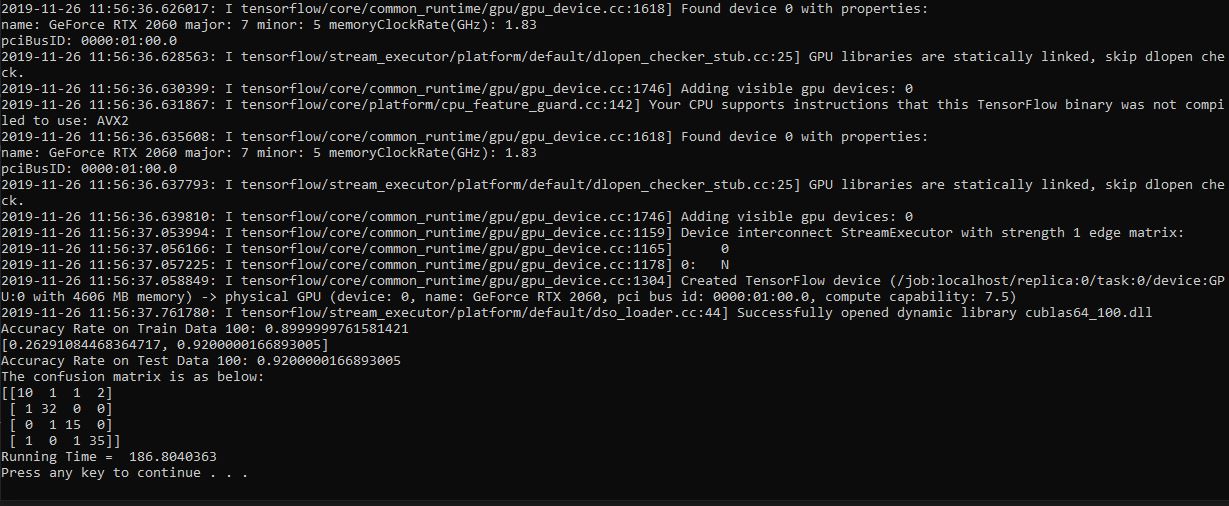
Class 1 error = 0.11

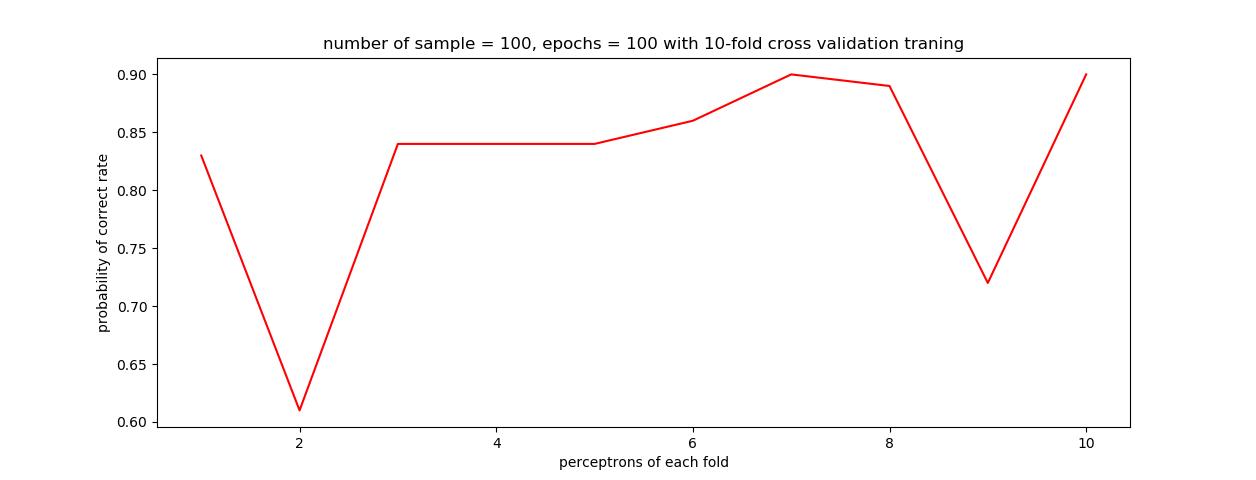
Class 2 error = 0.05

Class 3 error = 0.10

Class 4 error = 0.09

* 1. Training 100 data for getting the model in Keras library in python





Training the 100 samples data with neural network with 2 layers, with 1st layer setting the ‘tanh’ activation function, and the 2nd layer setting the ‘softplus’ activation function.

After the training, the best node, bias and weight will pass to the model of Keras object.

Let this object test the test data of 100 samples.

The result shows the confusion matrix, which allow us to find the accurate estimation

In 100 testing sample and 100 10-fold training.

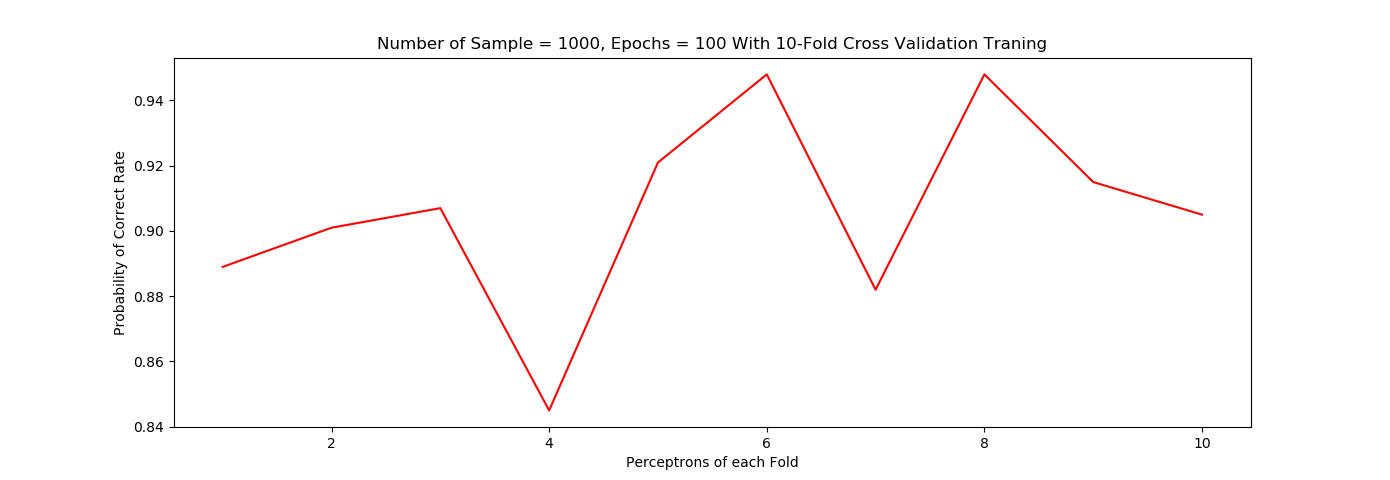
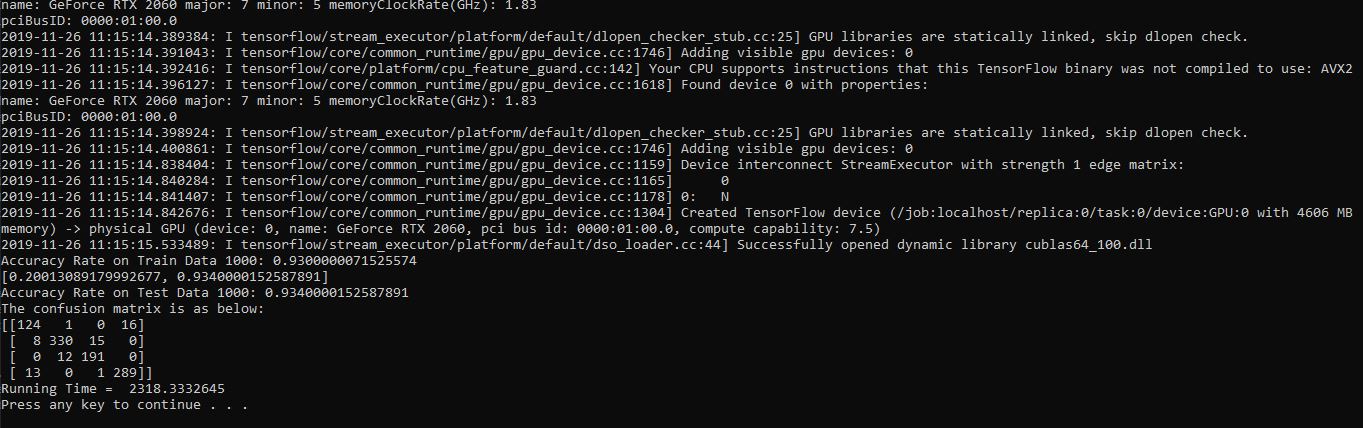
Correct in label 1 = 10

Correct in label 2 = 32

Correct in label 3 = 15

Correct in label 4 = 35 Accuracy = (10+32+15+35)/100 = 0.92

* 1. Training 1000 data for getting the model in Keras library in python



Same as the former training in 100 samples. We take the first layer with activation function ‘tanh’, and the second layer with activation function ‘softplus’.

After getting the best model in the training, fit the model to test samples.

The result shows the confusion matrix, which allow us to find the accurate estimation

In 1000 testing sample and 1000 10-fold training.

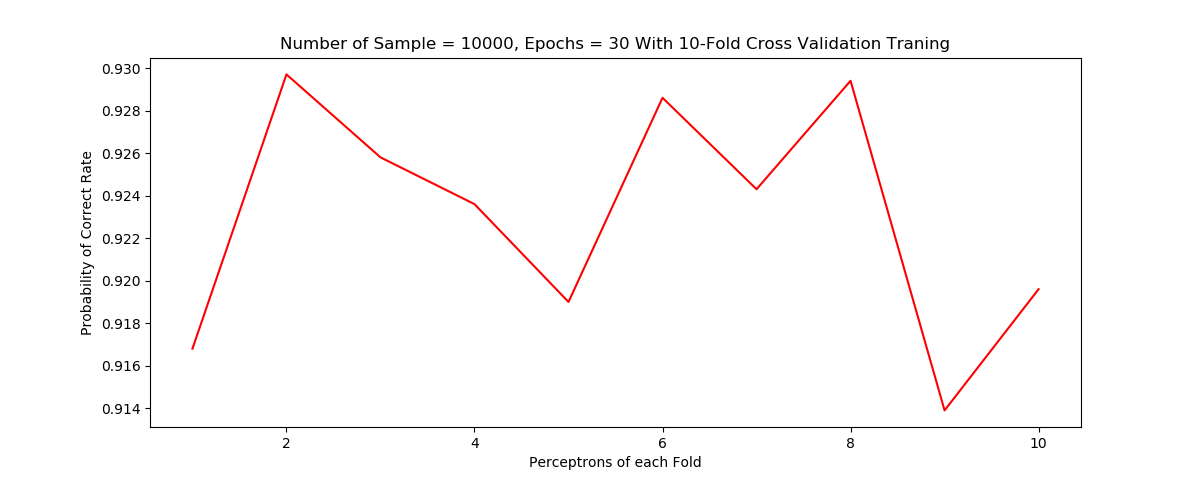
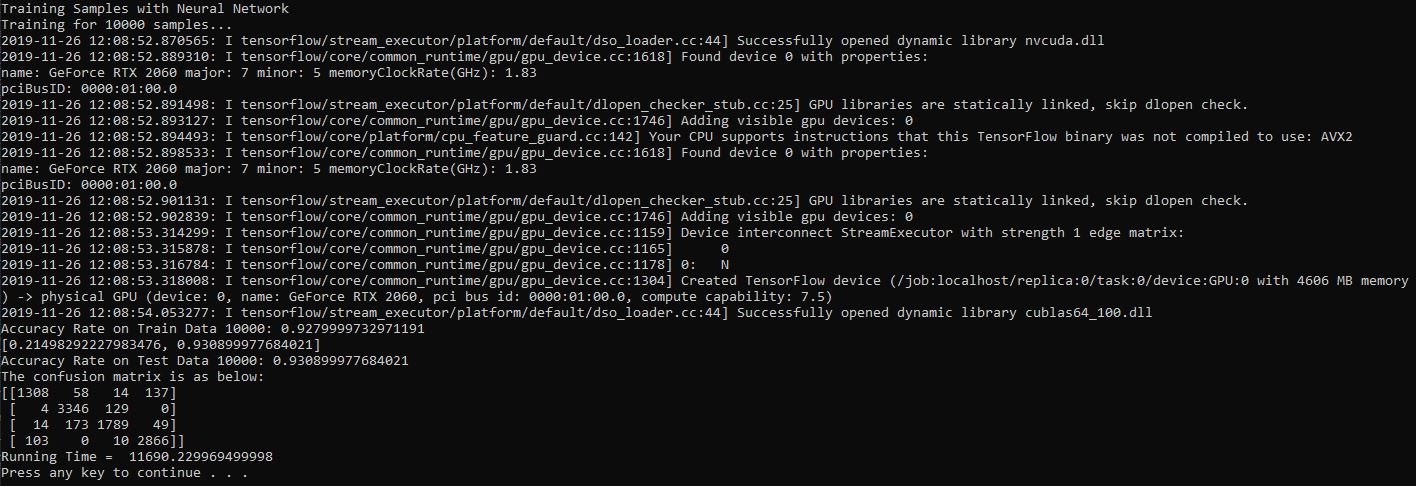
Correct in label 1 = 124

Correct in label 2 = 330

Correct in label 3 = 191

Correct in label 4 = 289 Accuracy = (124+330+191+289)/1000 = 0.934

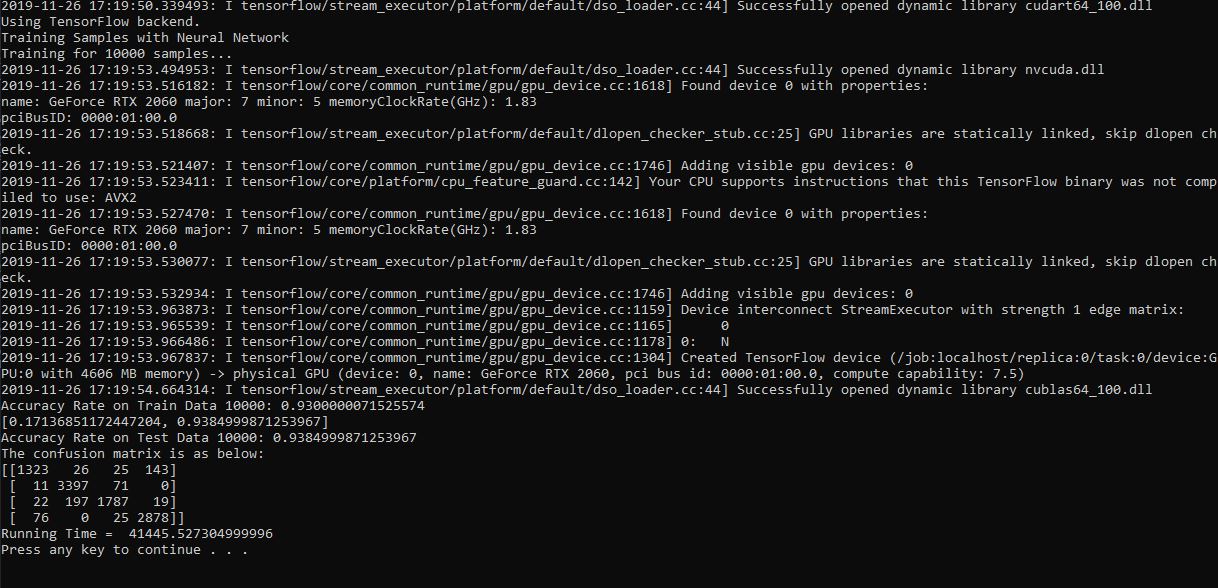
* 1. Training 1000 data with epoch 30 for getting the model in Keras library in python

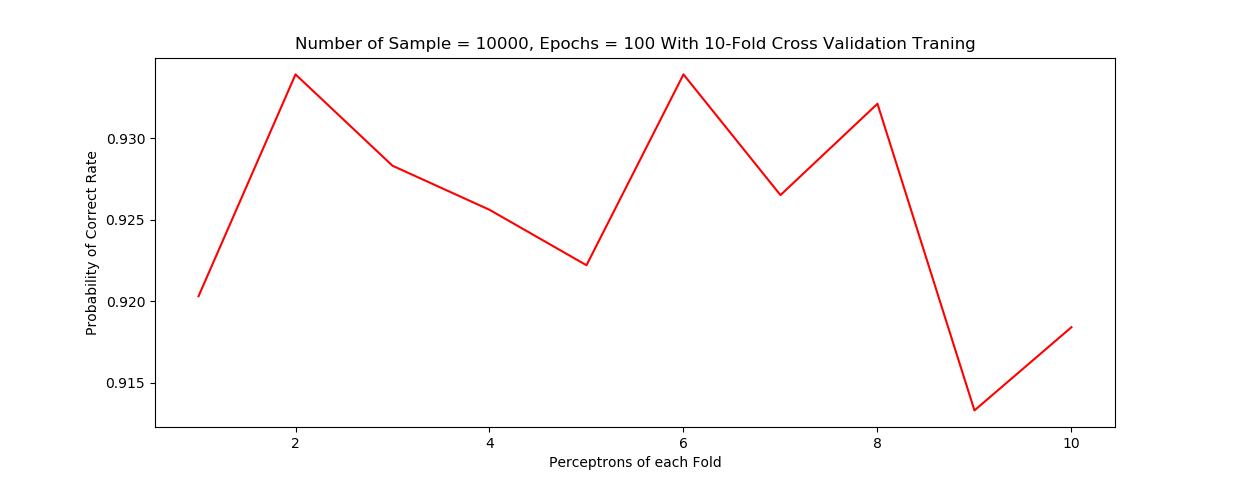


Same method as former two experiment

We get the confusion matrix that we could get the accurate rate and the wrong estimation number

* 1. Training 1000 data with epoch 100 for getting the model in Keras library in python





Here I tried the different if it can get a better result for accurate rate

Increase the epoch from 30 to 100.

The accurate rate: 0.9384

Conclusion of Question 1: