CISC 452 assignment 2 report

### ANN structure

The model structure with 784-input unit, a hidden layer1 with 300 nodes, hidden layer 2 output layer with 10 nodes.

10

nodes

300

nodes

784

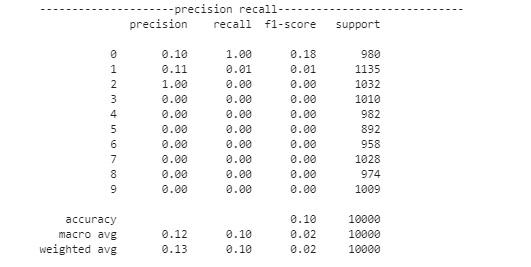
nodes

Input layer hidden layer 1 output layer

The initial weight values are set to random number between [-1/sqrt(node number),1/sqrt(node number)], and all the bias are set to 0 initially.

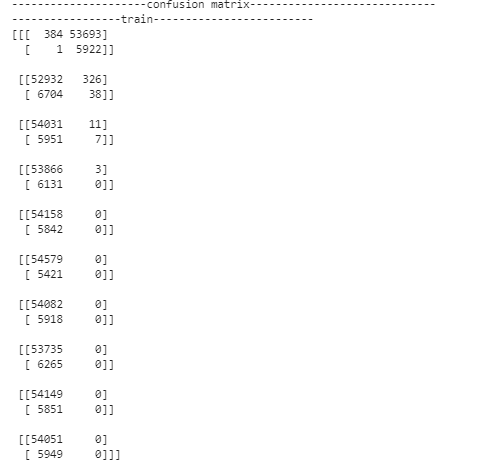
The training iterations is set to 200 epoch with all the training data per epoch with learning rate at 0.05, momentum\_factor at 0.01. The termination condition is either all the epoch ends or the loss during the training step is less than 0.01 for over 300 times on one epoch.

The precision-recall matrix are list as follows:

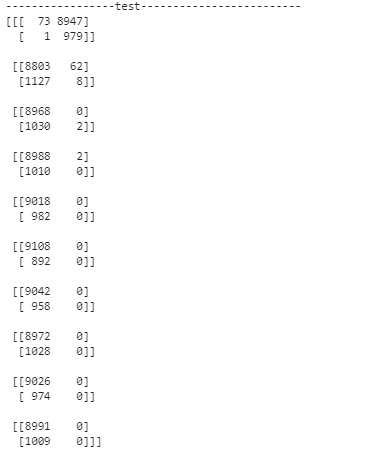


The confusion matrix are list as follows:

For training set:



For testing set:



The result is very bad, since based of the huge amount of calculation, the performance of the model is ~0.2 accuracy after several trials, and the final result is ~0.1 accuracy.

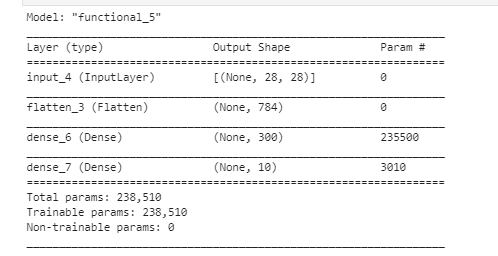
Maybe is because the dimension of the original data is too large.

The BP works fine, but the network tend to classify all the input into one category. Maybe the bias update have some problem.

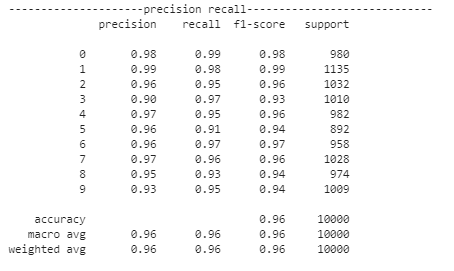
### The library Parts

I used the keras library for part2. while keep the structure and activation same as part1.

For model 2A the structure is like this, while keeping all the activation same with the model\_1(sigmoid)



The precision-recall matrix are list as follows:

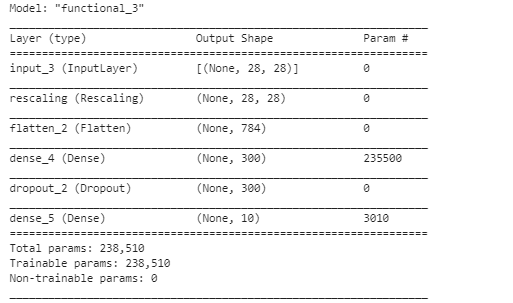


For model\_2B. I have changed the activation function for hidden layer to “relu” and “softmax” for the output layer.

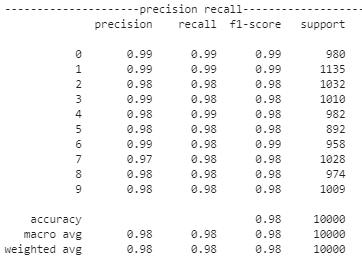
Added rescaling layer to convert all the input stay between 0~1 which is better for ANN.

Also added a dropout layer to prevent overfitting.

The structure of Model\_2B is like this.

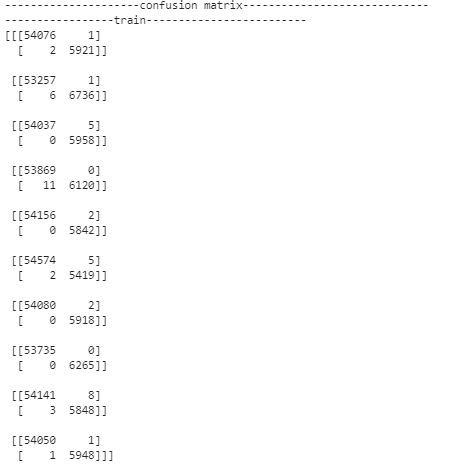


The precision-recall matrix are list as follows:

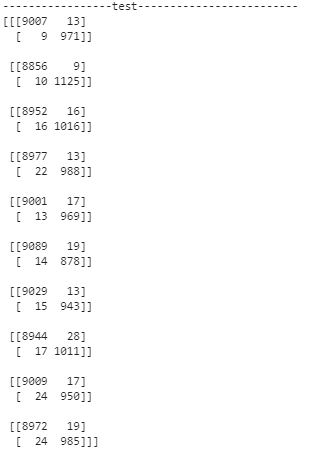


The confusion matrix are list as follows:

For training set:



For testing set:



**Conclusion**

Compare to the library model, the programming model have a really bad performance even after several adjustment.

This might because of following reason:

1. The bias update have some problems which result in a highly biased result.
2. Something might went wrong in the training process which would result in a functional model but with poor performance.
3. The amount of categories need to be classified is relatively large(10)

For the library model.

Adding rescaling layer and change activation function boost the model performance obviously. Even if the original 2A model already achieve a very high accuracy rating.

Which means rescaling the input and use the softmax activation function at output layer have positive effect on imagine classification.