# Assignment 1 Report

Yanjun Lyu

Problem1:

Problem set:

Using Keras, build a MLP to classify the CIFAR-10 dataset, build a MLP to classify the data into the 10 classes.

Modify the following parameters : (1) No. of epochs, (2) batch size, (3) No. of nodes in a layer, (4) No. of the layers, (5) learning rate (6) activation function, (7) dropout rate.

## 1.1 Basic Model

First of all, all the parameter test should base on a basic model with default parameter setting. The basic model is fully connected neural-network('Dense' layer in Keras) with 'softmax' being output layer activation function.

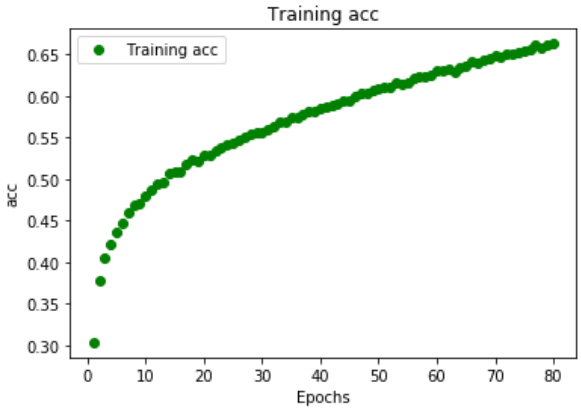
The other para values of the basic model is as follows:

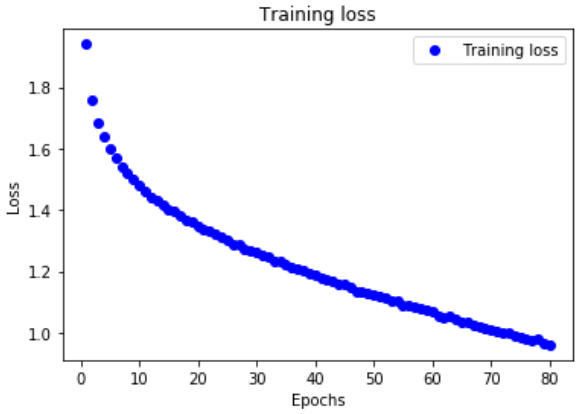
| batch size | # nodes | # layers | learning rate | Active Fun | dropout | epochs |
|---|---|---|---|---|---|---|
| 128 | 512 | 1 | 0.05 | relu | 0.25 | 80 |

However the training accuracy achieved is above 0.6, the test accuracy is 0.53 which indicated the potential overfitting.

test_acc: 0.6592

train acc: 0.5374




## 1.2 Parameters Test

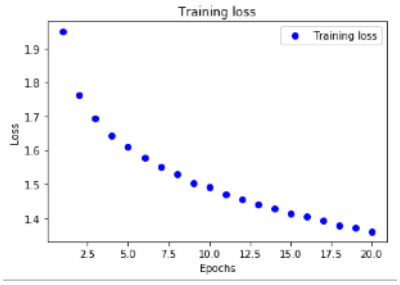
The experiment was designed to test the effects of the change of parameter

Consider every parameter type to be a factor, in every single experiment, other than the objective factor, all the factors are fix and at the same values corresponding to the basic model. There are two level for each factor.

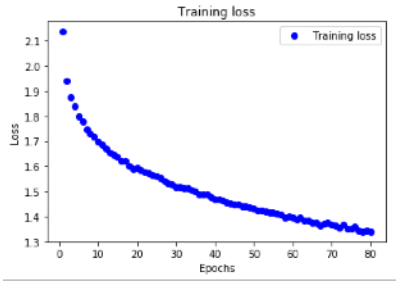
The results of accuracy is listed below:

| **Factor** | batch size | | # nodes | | # layers | | learning rate | | Active Fun | | dropout | | epochs | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Value** | 128 | 1000 | 512 | 64 | 2 | 3 | 0.05 | 0.5 | Relu | Sigm oid | 0.25 | 0.75 | 80 | 20 |
| **Test Acc** | 0.53 | 0.5 | 0.53 | 0.48 | 0.53 | 0.56 | 0.53 | 0.1 | 0.53 | 0.52 | 0.53 | 0.51 | 0.53 | 0.49 |
| **Train Acc** | 0.66 | 0.54 | 0.66 | 0.49 | 0.66 | 0.67 | 0.66 | 0.1 | 0.66 | 0.52 | 0.66 | 0.46 | 0.66 | 0.52 |

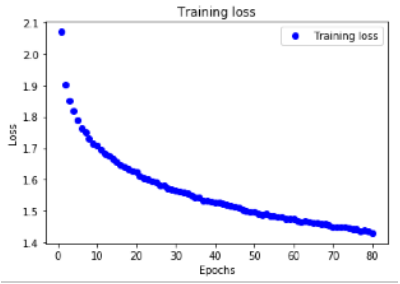
The training loss change with epoch numbers during training:



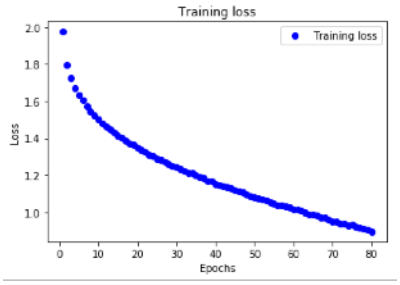

Epoch=20



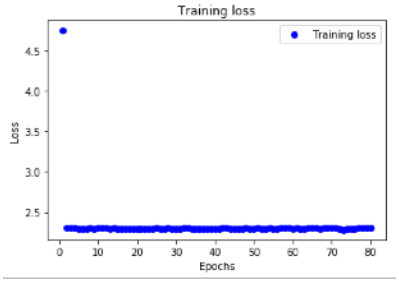

batch size=1000



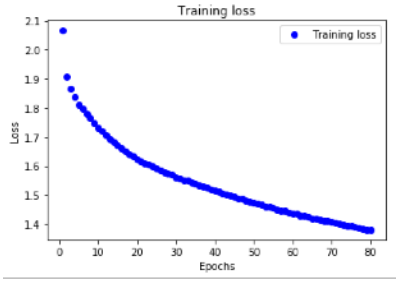

node=64




layer=3




lr=0.5




dropout=0.75

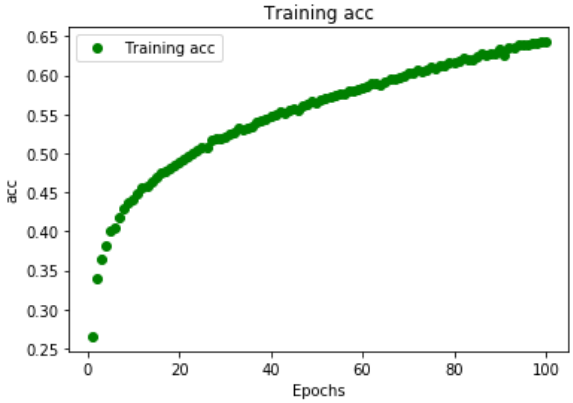
Combined with the losses and accuracy information, it could be concluded that for this data set, MLP may not powerful enough to make better classification. However, a more complex model brings a better result. Obviously, a large learning rate and big batch size are improper.

1.3 Presumably Optimal Model

Base on the result above, making some trade-off consideration to choose the optimal parameters for the case with the relatively higher accuracy, less computing time, and less loss.

The parameters are set to be epoch=100, batch size=256, layers to be 3, node number to be 400, lr=0.05 and the results is showed below:

test_acc=0.56, train_acc=0.64




As for improvement, an available way is to deeper the network as well as involve high regulation penalty.