**ASSIGNMENT 1**

**Problem 1**

Using Keras, build a MLP to classify the CIFAR-10 dataset. Note that each record is of size 1\*3072. Starting with the MNIST example code, build a MLP to classify the data into the 10 classes.

We started with MLP base code for MNIST dataset.

<https://github.com/keras-team/keras/blob/master/examples/mnist_mlp.py>

**Modifying parameters**

1. Batch size

Base model - Batch size was retained as 128, we increased epochs to 100 to get an overview of how it is getting trained. The input shape for cifar 10 is 32\*32\*3 = 3072. We set number of filters and dropout same as mnist values. The initial accuracy value is 49%. We plotted graphs to see train and test accuracies and losses. The first model’s accuracies and losses were close to each other. The model seems to overfit the train dataset.

Model 1 – modified batch\_size to 32 and maintained epochs = 100 and dropout = 0.2. The accuracy dropped to 31%. The model train slower than before and test accuracy was fluctuating in between. Decreasing batch size din’t help to improve accuracy

Model 2 – Increasing batch\_size to 256 and maintaining epochs = 100 and dropout = 0.2. The test accuracy was 48% and train accuracy was 55%. The model is overfitting the training dataset and hence the test accuracy is lower than train accuracy.

1. a. Network Configuration - Number of layers

Model 3 – Added another dense and dropout layer with same number of neurons and other parameters. The accuracy was 47%. The test and train accuracies were on par with each other until last epoch. Model seems to fit well with the train set.

1. b. Network Configuration - Number of neurons in a layer

Model 4 – Increased number of neurons to 1024 in the second layer. Test accuracy dropped down a little to 46%. The model is overfitting the train dataset.

Model 5 – Reset number of neurons to 1024 in both layers to get better details. Accuracy increased a little to 47%.

Model 6 – Increased number of neurons to 2048 in both layers. The accuracy dropped down to 0.1. Model takes longer time to learn due to increase in neurons.

Model 7 – Changing number of neurons to 256 and accuracy improves to 47% and the model seems to fit very well.

Model 8 – Two layers are added and number of neurons is increased to 512 in each layer. Accuracy is 45% and test accuracy is fluctuating in-between.

Model 9 – Changing to three layers and with first layer of 1024 neurons (to capture more details) and increasing epochs to 200. Accuracy increases to 48%.

1. Learning rate (& epochs)

Model 10 – Decreasing learning rate to 0.0001 from the default value of 0.001 increases accuracy to 56%. Train accuracy is 88% which is much higher than test accuracy which shows that model is overfitting the train dataset.

Model 11 – Increasing learning rate to default value of 0.001 accuracy decreases to 48% and no major difference in train and test accuracies which shows that model fits well.

Model 12 – Decreasing learning rate to 0.0008 accuracy increases to 49% and train and test accuracies are on par showing that model fits well.

Model 13 – Decreasing learning rate to 0.0001 and decreasing epochs to 25 to see of model learns faster, accuracy increases to 54% and train-test accuracies are on par. By far, the best model.

Model 14 – Increasing epochs to 70, decreases accuracy to 50%

1. Activation functions

Model 15 – Modifying activation function to tanh and setting epochs = 70 and learning rate = 0.0008. Accuracy is 48%.

Model 16 – Modifying activation function to sigmoid and retaining epochs = 70. Accuracy is increased to 55% but model tries to overfit.

Model 17 – Modifying activation function to soft sign and decreasing dropout to 0.1. accuracy is 53% and there is a difference in train and test accuracies.

1. Dropouts

Model 18 – Decreasing dropout to 0.1. Accuracy is maintained at 52% and both accuracies are on par. (one of the best models). Accuracy increasing by dropping less number of neurons.

Model 19 – Increasing dropout to 0.5. Accuracy drops down to 36%. It is evident that most of the important features are lost and hence setting lesser dropout rates gives better accuracy

**Preferred model**

Setting epochs = 70, batch size = 128, number of layers = 3, activation = relu, learning rate= 0.00008 dropout rate=0.2, the test accuracy is 50% and train accuracy is 52%.

Setting epochs = 25, batch size = 128, number of layers = 3, activation = relu, learning rate= 0.00009 and dropout rate=0.2, the test accuracy is 53% and train accuracy is 57%.

**Comment:** We conclude that cifar 10 dataset works best with slower learning rates, less dropout features and using 3 dense and dropout layers.