FOR PART A

The above graphs and accuracy were shown for part A.

Out of various parameters that were tested the best parameters were

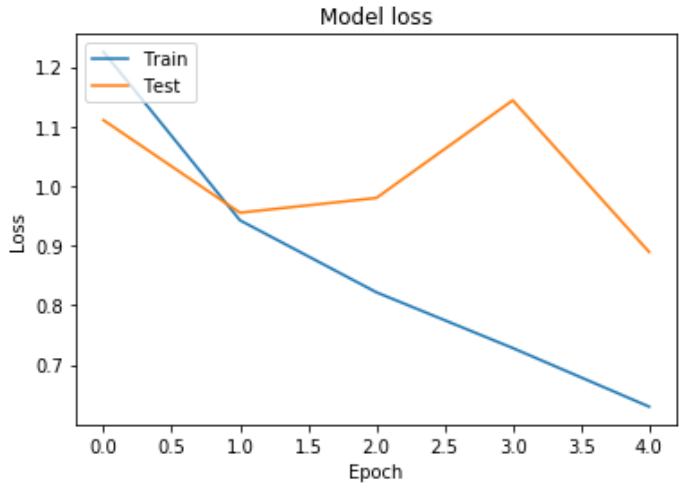
Batch size = 100

Optimizer = RMSprop

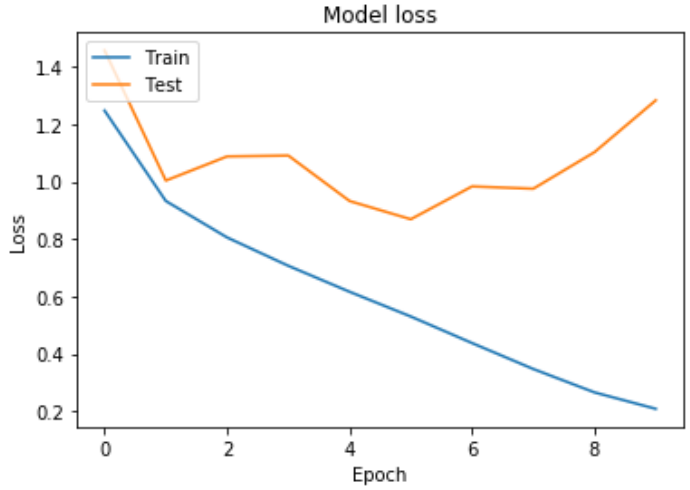
Epochs = -10

Expected accuracy = 71.1%

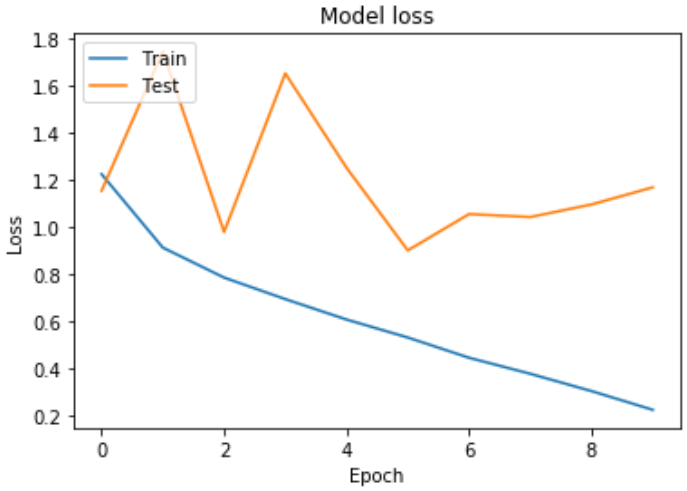
Batch = 50, Epoch = 5, Optimizer = Adam, Accuracy = 70.2



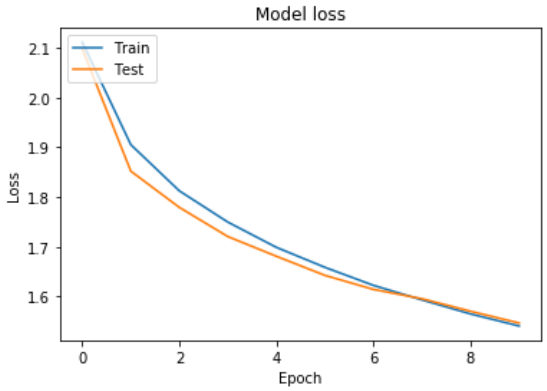
Batch = 50, Epoch = 10, Optimizer = Adam, Accuracy = 70.55



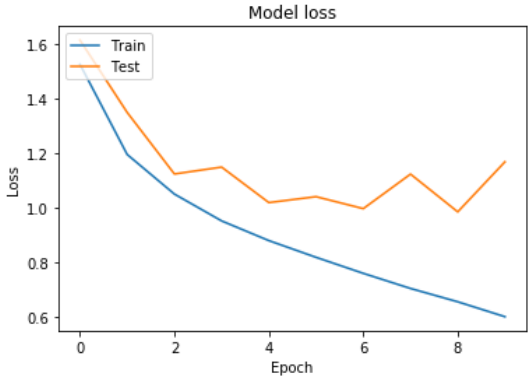
Batch = 100, Epoch = 10, Optimizer = Adam, Accuracy = 70.73



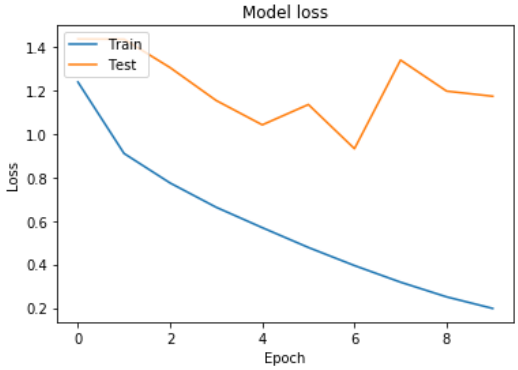
Batch = 100, Epoch = 10, Optimizer = Adadelta, Accuracy = 47.54



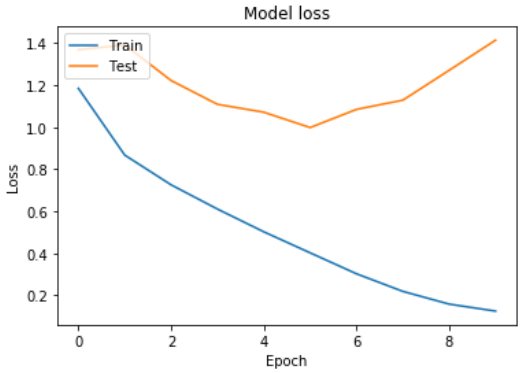
Batch = 100, Epoch = 10, Optimizer =SGD, Accuracy = 61.63



Batch = 100, Epoch = 10, Optimizer =RMSprop, Accuracy = 71.37



Batch = 100, Epoch = 10, Optimizer =Nadam, Accuracy = 71.21



REPORT FOR PART B

NOVELTY and Architecture:

The model involves a simple convolution network with 9 Convolution Layers and One fully connected layer. The Padding and strides value of each layer was one and the kernel size was 3*3. Alternate Batches of layers had a Maxpooling of size 2*2. RELU Nonlinearity was applied at each layer to tackle vanishing gradient descent.

The proposed architecture is very small in terms of depth and yet produces better results. The number of parameters in the architecture is also quite small. The time taken of execution is also not very high and the memory storage is also very low.

HYPER-PARAMETERS & CONFIG:

The Adam optimizer with Cateogorical_crossentropy loss was used for fitting the data. Validation was done on 10% of train dta with a batch size of 100 and a maximum of 50 epochs. Early stopping condition was applied for val_loss with patience of 5. The program usually stopped eary at 23 epochs. For tackling the problem of overfitting Dropout was used after each layer. Relu no linearity was applied everywhere except the last layer which had softmax.

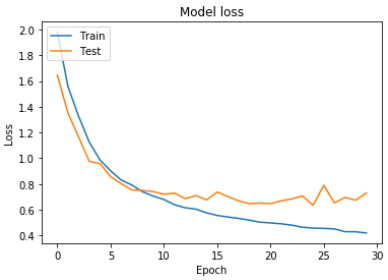
Other parameters were set to the default value of the functions.

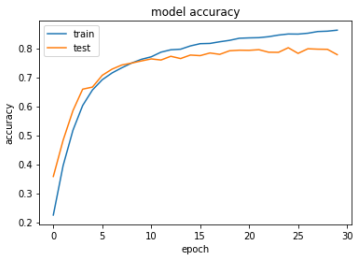
Performance:

No. of parameters = 3.7 Million (approx)

Accuracy = 79.1%(highest)

Time taken = 25 epochs with each epoch taking 70 sec on NVIDIA MX 150





WHY IT WORKS BETTER:

The given architecture is highly homogenous and is not very deep thus reducing its complexity for calculations also lesser number of layers ensure a less memory utilization thus making the architecture simply and effective.

The accuracy obtained is comparatively high this can be attributed to various factors such as homogeneity of the layers, more deeper network, regualrized layers and maximum utilization the given dataset.