**ASSIGNMENT 1**

USING KERAS

**Problem 1:**

Model 1:

Using Keras to build a CNN network for CIFAR-10 dataset. Each record is of size 1\*3072. Building a CNN network to classify the data into the 10 classes.

DATASET:

CIFAR-10 dataset The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

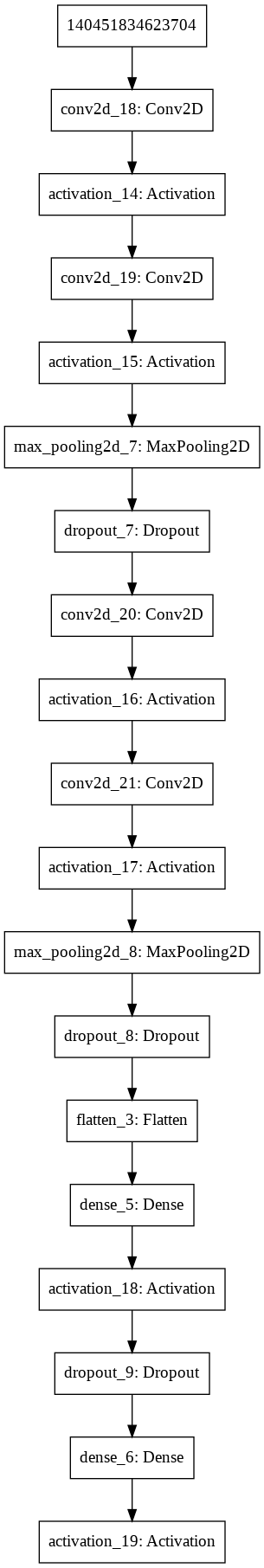
<http://www.cs.utoronto.ca/~kriz/cifar.html>

Modify the following parameters and discuss the effect of changing parameters on loss and

Accuracy:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | No of epochs | Batch size | Number of neurons | Number of layers | Learning rate | Activation functions | Dropout rates |
| Model 1 | 50 | 32 | 3,690 | 19 | 0.0001 | relu, softmax | 0.25 |
| Model 2 | 60 | 32 | 3,306 | 19 | 0.0001 | relu, softmax | 0.5 |

* Based on the above results, CNN model 1 performed well and gave an accuracy of 78.28%
* CNN using relu & softmax activation functions improved the results.



Model Fitting:

Epoch 35/50

50000/50000 [==============================] - 25s 496us/step - loss: 0.6220 - acc: 0.7935 - val\_loss: 0.6758 - val\_acc: 0.7861

Epoch 36/50

50000/50000 [==============================] - 44s 872us/step - loss: 0.6191 - acc: 0.7953 - val\_loss: 0.7021 - val\_acc: 0.7699

Epoch 37/50

50000/50000 [==============================] - 43s 852us/step - loss: 0.6217 - acc: 0.7929 - val\_loss: 0.6554 - val\_acc: 0.7899

Epoch 38/50

50000/50000 [==============================] - 42s 847us/step - loss: 0.6223 - acc: 0.7930 - val\_loss: 0.7450 - val\_acc: 0.7587

Epoch 39/50

50000/50000 [==============================] - 42s 841us/step - loss: 0.6211 - acc: 0.7944 - val\_loss: 0.7343 - val\_acc: 0.7573

Epoch 40/50

50000/50000 [==============================] - 44s 884us/step - loss: 0.6177 - acc: 0.7966 - val\_loss: 0.6375 - val\_acc: 0.7911

Epoch 41/50

50000/50000 [==============================] - 43s 850us/step - loss: 0.6190 - acc: 0.7944 - val\_loss: 0.6561 - val\_acc: 0.7835

Epoch 42/50

50000/50000 [==============================] - 43s 861us/step - loss: 0.6202 - acc: 0.7930 - val\_loss: 0.7403 - val\_acc: 0.7601

Epoch 43/50

50000/50000 [==============================] - 43s 861us/step - loss: 0.6131 - acc: 0.7971 - val\_loss: 0.6778 - val\_acc: 0.7757

Epoch 44/50

50000/50000 [==============================] - 43s 863us/step - loss: 0.6213 - acc: 0.7947 - val\_loss: 0.6707 - val\_acc: 0.7816

Epoch 45/50

50000/50000 [==============================] - 43s 851us/step - loss: 0.6175 - acc: 0.7966 - val\_loss: 0.7192 - val\_acc: 0.7697

Epoch 46/50

50000/50000 [==============================] - 42s 845us/step - loss: 0.6186 - acc: 0.7954 - val\_loss: 0.6585 - val\_acc: 0.7884

Epoch 47/50

50000/50000 [==============================] - 42s 846us/step - loss: 0.6218 - acc: 0.7945 - val\_loss: 0.6239 - val\_acc: 0.7989

Epoch 48/50

50000/50000 [==============================] - 43s 862us/step - loss: 0.6260 - acc: 0.7945 - val\_loss: 0.6996 - val\_acc: 0.7685

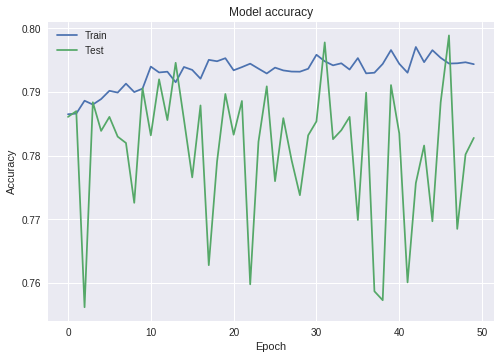
Epoch 49/50

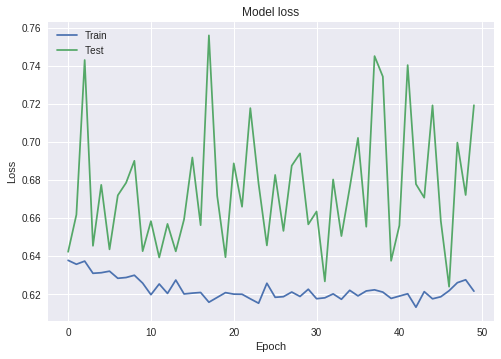
50000/50000 [==============================] - 42s 844us/step - loss: 0.6275 - acc: 0.7947 - val\_loss: 0.6721 - val\_acc: 0.7802

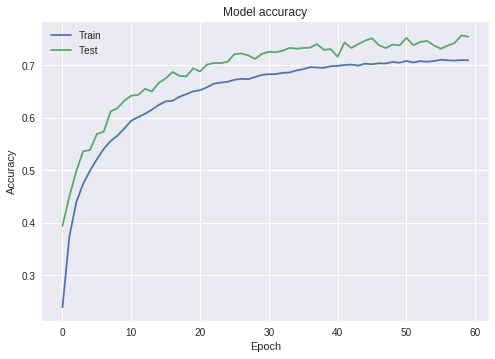
Epoch 50/50

50000/50000 [==============================] - 42s 847us/step - loss: 0.6215 - acc: 0.7944 - val\_loss: 0.7192 - val\_acc: 0.7828

Plotting:









Results:

1. Provide a recommendation for the best model you would recommend for classification. Which model (with parameter values) would you choose and why?

According to me, the best model for classification is cnn model 1. The best parameters that helped improve the model are:

epochs = 50

batch\_size = 32

keep\_probability = 0.7

learning\_rate = 0.001

Optimizer

1. Comment on how good your model is ? Does it overfit/underfit data ? What could you do to improve the model?

I ran it for 50 epochs and got almost 78% accuracy. It can surely go much further since it was still undertrained! To improve the model the learning rate should be improved. Also, number of neurons can be more complicated for a better fit.