Programming Assignment 1

Design and Implementation of Convolution Neural Network to Classify Images in the CIFAR-10 Dataset

Architecture –

Conv2D Layer 1:

Input shape: (32, 32, 3) – The input is a colour image with size 32x32 and 3 channels. Number of filters: 32. Kernel size: (3, 3). Activation: ReLU (Preferred over tanh for inner layers).

Parameters: Number of parameters = (3 x 3 x 3 + 1) x 32 = (27 + 1) x 32 = 28 x 32 = 896

Conv2D Layer 2:

Input Shape: (30, 30, 32) – Output of the first convolution layer. Number of filters: 64 with 3 channels. Kernel size: (3,3). Activation: ReLU (Preferred over tanh for inner layers).

Parameters: Number of parameters = (3 x 3 x 32 + 1) x 64 = 18,496

Conv2D Layer 3:

Input Shape: (28, 28, 64) – Output of the first convolution layer. Number of filters: 64 with 3 channels. Kernel size: (3,3). Activation: ReLU (Preferred over tanh for inner layers).

Parameters: Number of parameters = (3 x 3 x 64 + 1) x 128 = 73,856

MaxPooling 2DLayer:

Pool Size: (2, 2), This layer has no learnable parameters, it is used to prevent overfitting.

Flatten Layer:

Input Shape: (14, 14, 128) – After applying Max Pooling. Output Shape: (14*14*128) = 25,088

Dense Layer:

Number of Parameters: (14, 14, 128). Number of Parameters: (14*14*128 + 1) *128 = 3,211,392. Activation: Tanh (experimented as it has negative value and provides for better convergence)

Dropout Layer 2:

Dropout has no learnable parameters.

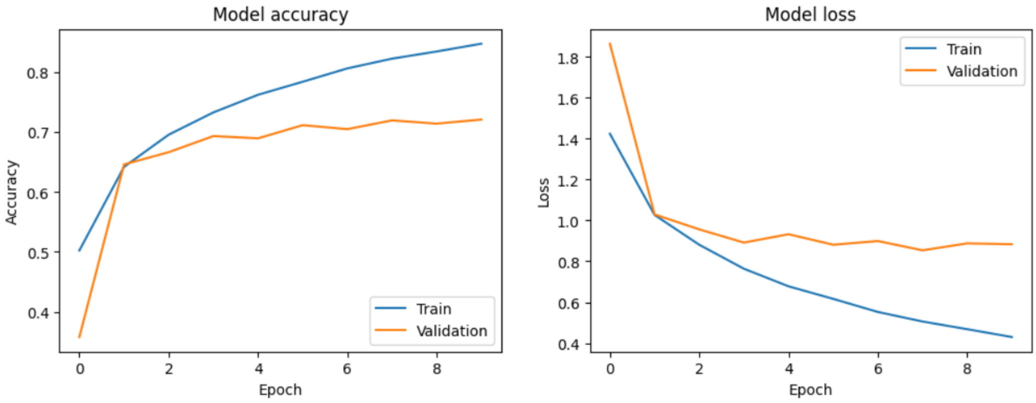
Output Layer (Dense):

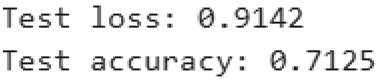
Input Size: 128. Output Size: 10. Activation: Softmax (for multiclass classification). Parameters = (128+1)*10 = 1290

Total Parameters:

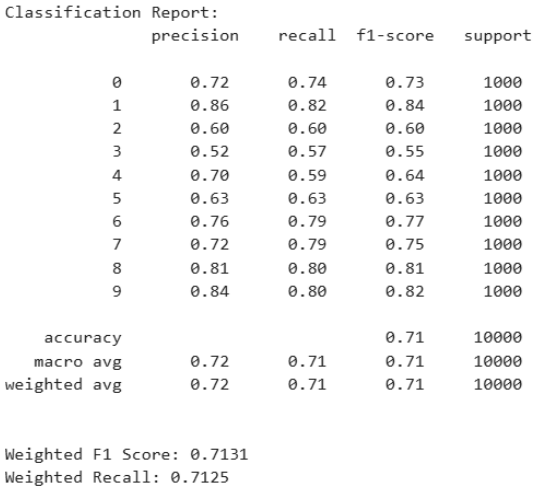
896 (Conv 2D-1) + 18,496(Conv 2D-2) + 73,856(Conv 2D-3) + 25,088(Flatten Layer) + 3,211,392(Dense Layer) + 1290(Output Layer) + 512(Batch Normalization) = 3,306,442.

Model Performance:





```
Test loss: 0.9142
Test accuracy: 0.7125
```



```
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.74      0.73      1000
           1       0.86      0.82      0.84      1000
           2       0.60      0.60      0.60      1000
           3       0.52      0.57      0.55      1000
           4       0.70      0.59      0.64      1000
           5       0.63      0.63      0.63      1000
           6       0.76      0.79      0.77      1000
           7       0.72      0.79      0.75      1000
           8       0.81      0.80      0.81      1000
           9       0.84      0.80      0.82      1000

    accuracy                           0.71     10000
   macro avg       0.72      0.71      0.71     10000
weighted avg       0.72      0.71      0.71     10000


Weighted F1 Score: 0.7131
Weighted Recall: 0.7125
```

Different Trials: I have initially tried with only 2 Convolution Layers, but later on settled with 3 Convolution Layers with 128 filters. I have also added a batch normalization layer at the end of convolution layers. At first I used Dense Layer with 128 output with tanh activation. I have also tried changing the dropout value in the dropout layers to a lesser value but there was high difference in the Validation Error and Training Error. I also tried the RMSProp and Adagrad optimizer but later on settled with ADAM optimizer as it was showing better convergence.

Results: The accuracy of the model that I have designed on the Test Data is 71.25%. It means it can classify roughly 71 images correctly out of 100 images.