Programming Assignment 4

Name: Mayank Kumar
UCF-ID: 5529148
Email: mayank_kumar@ucf.edu

1. Introduction:

Image classification stands at the forefront of advancements in computer vision, playing a pivotal role in various applications ranging from medical diagnostics to autonomous vehicles. At its essence, image classification involves the utilization of machine learning techniques, particularly Convolutional Neural Networks (CNNs), to categorize images into predefined classes. This report delves into the fundamental challenge of designing and implementing CNN architectures for the purpose of recognizing RGB color images. We will also be observing the effect of adding additional Convolutional Layers on the performance of the network.

2. Dataset:

The CIFAR-10 dataset is a widely used benchmark in computer vision, comprising 60,000 32x32 color images distributed across ten distinct classes, each representing various objects and scenes. With a balanced class distribution and a fixed resolution of 32x32 pixels, CIFAR-10 serves as a standard testbed for evaluating the performance of image classification algorithms. The dataset's RGB color images and relatively low resolution provide a challenging yet realistic environment for testing the robustness of classification models, a natural next step from MNIST.

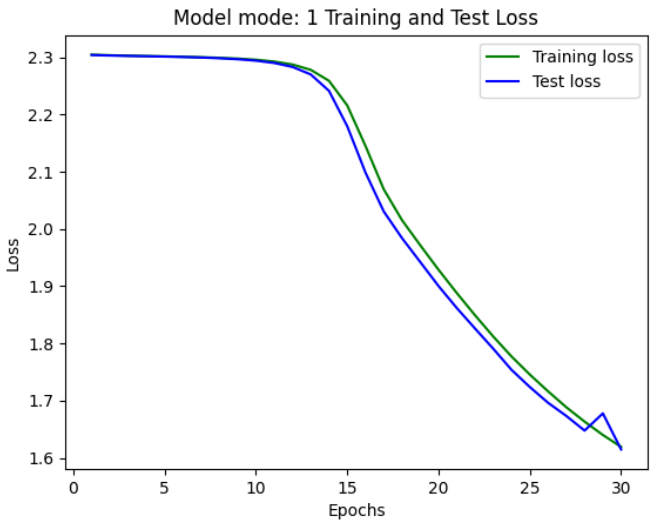
3. Model Architectures:

Both the models were trained for 30 epochs with learning rate of 0.001, batch size of 64 and SGD optimizer. The first Model has 3 Convolutional layers while the second one has 4. Both have 3 fully connected layers, although one is of different size to accommodate the new convolutional layer. The detailed list of parameters can be found in the table below.

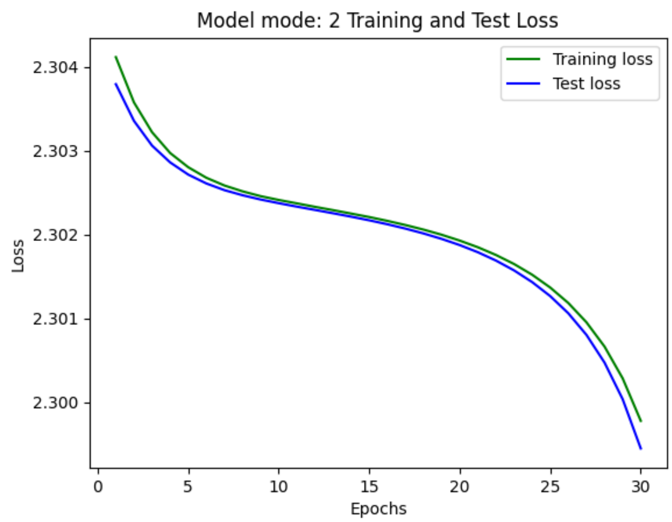
Parametres	Model 1	Model 2
Input	3x32x32	3x32x32
Conv 1	6 filters, 3x3, padding=1	6 filters, 3x3, padding=1
Activation 1	ReLU	ReLU
Pooling 1	2x2, max pooling	2x2, max pooling
Conv 2	16 filters, 3x3, padding=1	16 filters, 3x3, padding=1
Activation2	ReLU	ReLU
Pooling2	2x2, max pooling	2x2, max pooling
Conv 3	32 filters, 3x3, padding=1	32 filters, 3x3, padding=1
Activation 3	ReLU	ReLU
Pooling 3	2x2, max pooling	2x2, max pooling
Conv 4	-	64 filters, 3x3, padding=1
Activation 4	-	ReLU
Pooling 4	-	2x2, max pooling
Flattening	To 32x4x4	To 64x2x2
Fully Connected 1	32x4x4 to 120	64x2x2 to 120
Activation 5	ReLU	ReLU
Fully Connected 2	120 to 84	120 to 84
Activation 6	ReLU	ReLU
Fully Connected 3	84 to 10	84 to 10

Table 1: Model Architectures

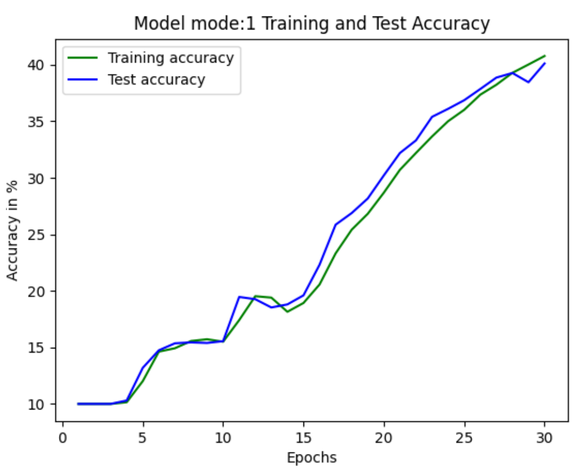
4. Results and Discussion:



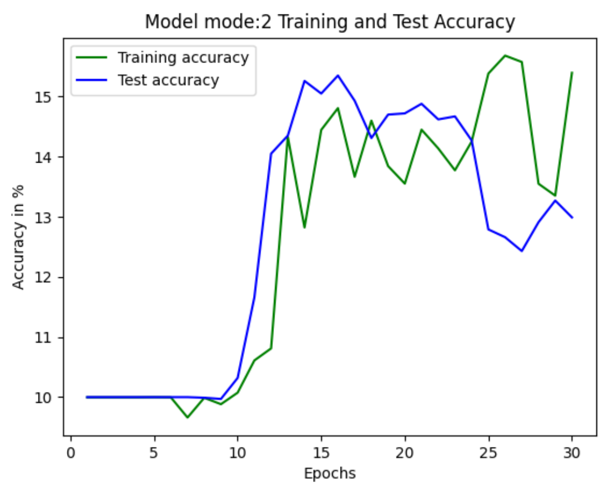
Loss Plot for Model 1



Loss Plot for Model 2



Accuracy Plot for Model 1



Accuracy Plot for Model 2

We observe that adding a new convolutional layer does not increase the performance rather it decreases it. The accuracy in the final epoch are 41% and 13% respectively, whereas the losses are 1.6149 and 2.2995. This shows us that making the network deeper is not always the best way to go.