Programming Assignment-IV

Computer Vision-CAP 5415

Question 1: Image Classification

To develop a CNN classification network to recognize RGB color images. CNN architecture having more than 2 convolutional layers and more than 1 fully connected layers.

Dataset

The CFAR10 dataset consists of colored images that are labeled by one of ten possible classes (airplane, automobile, bird, cat, deer, dog, from, horse, ship, and truck). The size of the images are 32x32. The total number of samples in the dataset is 6000 out of which 5000 training samples and 1000 testing samples. The CFAR dataset is available from the torchvision library.

Model Architecture

CNNs are trained over 30 epoch using cross-entropy loss and Stochastic Gradient Descent (SGD) optimizer with learning rate 0.01. Furthermore, the last FC layer is followed by a SoftMax layer. The batch size is 20.

Model 1 comprises of 3 conv layers with ReLU activation function and a maxpool layer and 3 fully connected(FC) layers.

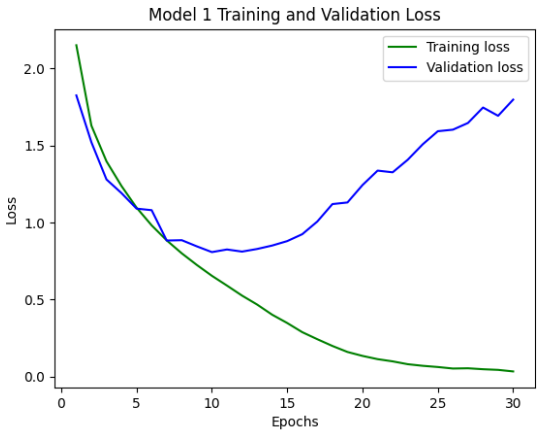
```
# Model 1: 3 conv & 3 FC layers

# ConV Layers
self.conv1 = nn.Conv2d(3, 16, 3, padding=1)
self.conv2 = nn.Conv2d(16, 32, 3, padding=1)
self.conv3 = nn.Conv2d(32, 64, 3, padding=1)
# max pooling layer
self.pool = nn.MaxPool2d(2, 2)
# fully connected layers
self.fc1 = nn.Linear(1024, 512)
self.fc2 = nn.Linear(512, 64)
self.fc3 = nn.Linear(64, 10)
```

The second model is the same as the first expect that it has an additional conv layer thereby making it have 4 conv layers.

```
# Model 2:4 conv& 3 FC layers
# ConV Layers
self.conv_1 = nn.Conv2d(3, 4, 3, padding=1)
self.conv_2 = nn.Conv2d(4, 16, 3, padding=1)
self.conv_3 = nn.Conv2d(16, 32, 3, padding=1)
self.conv_4 = nn.Conv2d(32, 64, 3, padding=1)
# max pooling layer
self.pool = nn.MaxPool2d(2, 2)
# fully connected layers
self.fc_1 = nn.Linear(64* 2*2, 128)
self.fc_2 = nn.Linear(128, 64)
self.fc_3 = nn.Linear(64, 10)
```

Results

The plots show the training and validation loss in fig 1 and training and validation accuracy in fig2 for model 1.



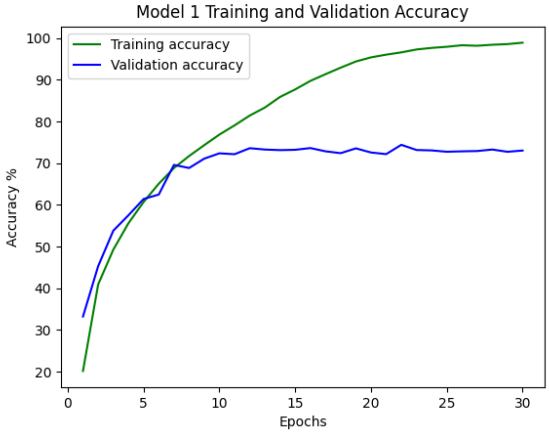


Figure 1

Figure 2

From the plots we can see that after 9 epochs the model converges at an accuracy of 71%.

And similarly fig 3 and fig 4 show the loss and accuracy of model 2, this model converges after 13 epochs and has an accuracy of 62%.

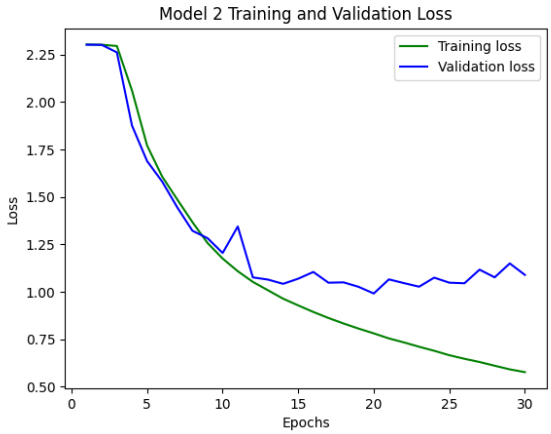


Figure 3

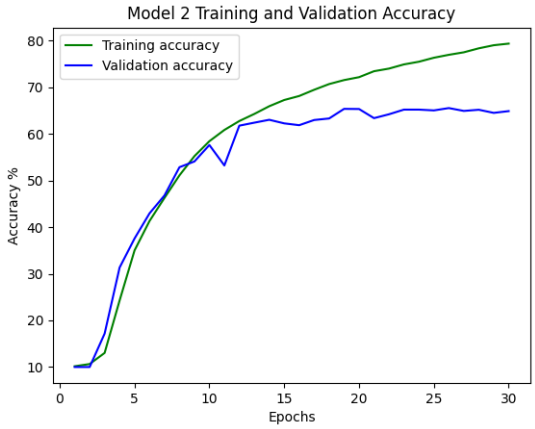


Figure 4

Conclusion

From these results, we can conclude that just adding another conv layer to the neural network did not improve the accuracy. It varies from dataset to dataset and the accuracy can be improved by changing some of the parameters of the model such as batch size, learning rate ,optimizer ,etc. Also, deep neural networks such as these would benefit from using a dropout layer.