CIFAR-10 Neural Network Report

Task 1

Two transformations were created for the data: one for the train set and another for the test set.

Data augmentation and normalization were applied to the training set, only normalization was applied to the test set.

Augmentations [Train only]:

transforms.RandomHorizontalFlip() -> Default p=0.5

transforms.RandomCrop(32, padding=4)

transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2)

transforms.RandomRotation(10)

Normalization [Train and Test]:

transforms.Normalize((0.4914, 0.4822, 0.4465), (0.247, 0.243, 0.261))

Normalization values specifically for the CIFAR-10 dataset were obtained from: GitHub Thread, Code used to generate values

Task 2

The model consists of two blocks followed by a classifier MLP. Regularization techniques were applied such as batch normalization and drop out were also used in between steps.

Block 1 [3 convolutional layers]:

Input X [batch_size,channels,height,width] -> [32,3,32,32]

Take Average across dim=[2, 3]

Pass through linear layer to obtain [a]

Compute $Conv_k(X)$ *for each convolutional layer*

Compute $O_1 = a_1 * Conv_1(X) + ... + a_k * Conv_k(X)$

Pass O_1 *through sequential*

Block 2 [3 convolutional layers]:

Input O_1 *from Block 1 output [batch_size,channels,height/4,width/4] -> [32,256,8,8]*

Take Average across dim=[2, 3]

Pass through MLP to obtain [a]

Compute $Conv_k(O_1)$ *for each convolutional layer*

Compute $O_2 = a_1 * Conv_1(O_1) + ... + a_k * Conv_k(O_1)$

Pass O_2 *through sequential*

Classifier [MLP]:

Input O_2 from Block 2 output [batch_size,channels,height/16,width/16] -> [32,512,2,2]

Take Average across dim=[2, 3]

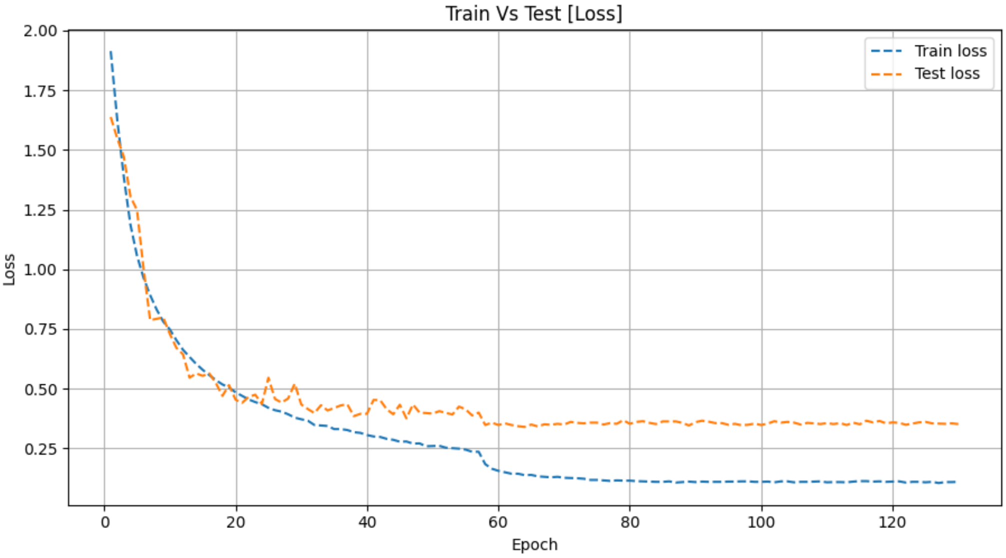
Pass through MLP to obtain 10 output features for classification

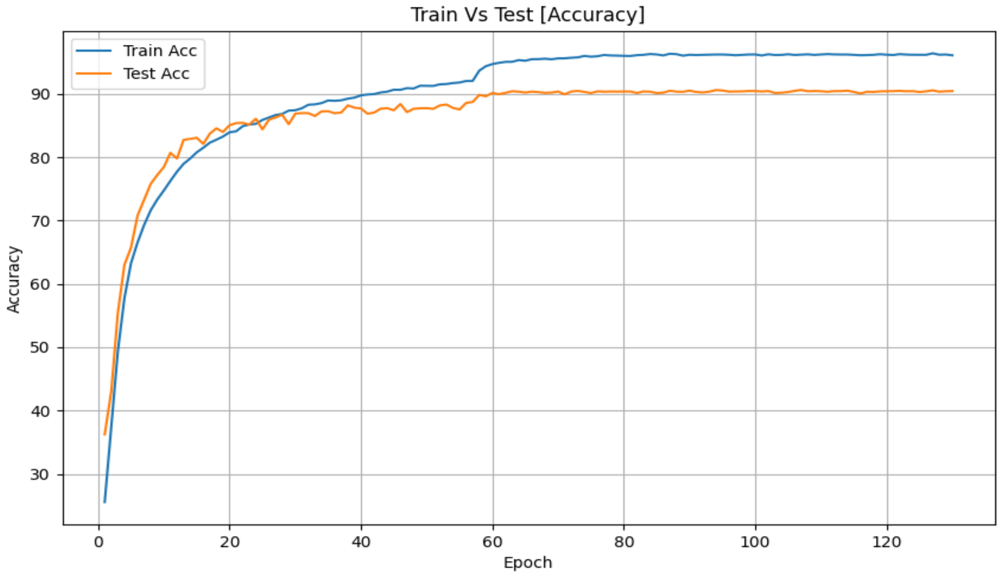
Task 3

optimizer = torch.optim.Adam(net.parameters(), lr=0.01)

loss function = nn.CrossEntropyLoss()

Task 4





Hyperparameters:

Batch Size -> 32

Epochs -> 130

Optimizer -> Adam

Learning Rate -> 0.01 (Added scheduler to reduce lr if test loss did not improve for 10 epochs)

Dropout Probability -> Convolutional layers (0.3), Linear layers (0.5)

Activation Function -> ReLU

Task 5

Model test accuracy peaked at epoch 107 of 130 with an accuracy of 90.63%.