20171097_assignment4

April 13, 2020

[0]:
```
import torch
import torch.nn as nn
import torchvision.datasets as dsets
import torchvision.transforms as transforms
from torch.autograd import Variable
import torchvision
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

1 Assignment 4 - Object classification

1.1 Ajay Shrihari, 20171097

1.1.1 Method

- In this assignment, we build a classifier with 3-4 convolutional layers and observe the effects of different optimizers, batch normalization, weight initialization, learning rate, and adding fully connected layers on the model's performance.
- We plot the training vs testing loss curves, training and testing accuracy curves and the training loss curve with the number of epochs to observe these effects.
- The following models are used: - Model 1: 1. Convolutional layer (Features in = 3, Features out = 6, Kernel size = 5) 2. Relu activation 3. Pooling layer (2*2) 4. Convolutional layer (Features in = 6, Features out = 16, Kernel size = 5) 5. Pooling layer (2*2) 6. Fully connected layer (Input size = 16*5*5 , Output size = 120) 7. Relu activation layer 8. Fully connected layer (Input size = 120 , Output size = 84) 9. Fully connected layer (Input size = 84 , Output size = 10)

```
    - Model 2: Weights initialized using Kaiming initialization
  1. Convolutional layer (Features in = 3, Features out = 12, Kernel size = 5, Zero Padding = 2
    2. Relu activation
```

3. Convolutional layer (Features in = 12, Features out = 8, Kernel size = 3, Zero Padding = 1
4. Relu activation
5. Fully connected layer (Input size = 84 , Output size = 10)

- The rest of the experiments done are on variations of these two models, including adding batch normalization between the layers as well as adding and remving more fully connected layers.

[0]:
```
def get_dataloader(name = 'cifar10'):
    if name == 'cifar10':
        transform = transforms.Compose([transforms.ToTensor()])
        train = torchvision.datasets.CIFAR10(root='./data', train=True,␣
↪download=True,transform=transform)
        trainloader = torch.utils.data.DataLoader(train, batch_size=64,␣
↪shuffle=True, num_workers=2)
        test = torchvision.datasets.CIFAR10(root='./data', train=False,␣
↪download=True,transform=transform)
        testloader = torch.utils.data.DataLoader(test, batch_size=64,␣
↪shuffle=False, num_workers=2)
    if name == 'cifar100':
        transform = transforms.Compose([transforms.ToTensor()])
        train = torchvision.datasets.CIFAR100(root='./data', train=True,␣
↪download=True,transform=transform)
        trainloader = torch.utils.data.DataLoader(train, batch_size=64,␣
↪shuffle=True, num_workers=2)
        test = torchvision.datasets.CIFAR100(root='./data', train=False,␣
↪download=True,transform=transform)
        testloader = torch.utils.data.DataLoader(test, batch_size=64,␣
↪shuffle=False, num_workers=2)
    return trainloader, testloader
```

[26]:
```
cifar10_trainloader, cifar10_testloader = get_dataloader()
```

```
Files already downloaded and verified
Files already downloaded and verified
```

[0]:
```
print (type(cifar10_testloader))
print (len(cifar10_testloader.dataset))
```

```
<class 'torch.utils.data.dataloader.DataLoader'>
10000
```

[0]:
```
class Net(nn.Module):
    def __init__(self,name = 'cifar10'):
        if name == 'cifar10':
            super(Net, self).__init__()
            self.name = name
            self.c1 = nn.Conv2d(3, 6, 5)
```

```
            self.pool = nn.AvgPool2d(2, 2)
            self.c2 = nn.Conv2d(6, 16, 5)
            self.c3 = nn.Conv2d(16, 24, 5)
            self.fc1 = nn.Linear(16*5*5, 120)
            self.fc2 = nn.Linear(120,84)
            self.fc3 = nn.Linear(84,10)

    def forward(self, value):

        value = self.pool(F.relu(self.c1(value)))
        value = self.pool(F.relu(self.c2(value)))
#         value = self.pool(F.relu(self.c3(value)))
        value = value.flatten(1)
        value = F.relu(self.fc1(value))
        value = F.relu(self.fc2(value))
        value = self.fc3(value)
        return value



class Net1(nn.Module):
    def __init__(self,name = 'cifar10', method = 1):

        if name == 'cifar10':
          super(Net1, self).__init__()

          self.c1 = nn.Conv2d(3, 12, 5, padding = 2)
          self.c2 = nn.Conv2d(12, 8, 3, padding = 1)
          self.fc = nn.Linear(8 * 32 * 32 , 10)
          nn.init.kaiming_normal_(self.c1.weight)
          nn.init.kaiming_normal_(self.c2.weight)
          nn.init.kaiming_normal_(self.fc.weight)
    def forward(self, value):

        value = F.relu(self.c1(value))
        value = F.relu(self.c2(value))
        value = value.flatten(1)
        value = self.fc(value)
        return value
class Net2(nn.Module):
    def __init__(self,name = 'cifar10'):
        if name == 'cifar10':
            super(Net2, self).__init__()
            self.name = name
            self.c1 = nn.Conv2d(3, 6, 5)
            self.pool = nn.AvgPool2d(2, 2)
            self.c2 = nn.Conv2d(6, 16, 5)
```

```
            self.c3 = nn.Conv2d(16, 24, 5)
            self.fc1 = nn.Linear(16*5*5, 120)
            self.fc2 = nn.Linear(120,96)
            self.fc3 = nn.Linear(96, 84)
            self.fc4 = nn.Linear(84,42)
            self.fc5 = nn.Linear(42,34)
            self.fc6 = nn.Linear(34,24)
            self.fc7 = nn.Linear(24,10)

    def forward(self, value):

        value = self.pool(F.relu(self.c1(value)))
        value = self.pool(F.relu(self.c2(value)))
#         value = self.pool(F.relu(self.c3(value)))
        value = value.flatten(1)
        value = F.relu(self.fc1(value))
        value = F.relu(self.fc2(value))
        value = F.relu(self.fc3(value))
        value = F.relu(self.fc4(value))
        value = F.relu(self.fc5(value))
        value = F.relu(self.fc6(value))
        value = self.fc7(value)
        return value

class Net3(nn.Module):
    def __init__(self,name = 'cifar10'):
        if name == 'cifar10':
            super(Net3, self).__init__()
            self.name = name
            self.c1 = nn.Conv2d(3, 6, 5)
            self.bn1 = nn.BatchNorm2d(6)
            self.pool = nn.MaxPool2d(2, 2)
            self.c2 = nn.Conv2d(6, 16, 5)
            self.bn2 = nn.BatchNorm2d(16)
            self.c3 = nn.Conv2d(16, 24, 5)
            self.fc1 = nn.Linear(16*5*5, 120)
            self.fc2 = nn.Linear(120,84)
            self.fc3 = nn.Linear(84,10)

    def forward(self, value):

        value = self.pool(F.relu(self.bn1(self.c1(value))))
        value = self.pool(F.relu(self.bn2(self.c2(value))))
#         value = self.pool(F.relu(self.c3(value)))
        value = value.flatten(1)
        value = F.relu(self.fc1(value))
        value = F.relu(self.fc2(value))
```

```
        value = self.fc3(value)
        return value

class Net4(nn.Module):
    def __init__(self,name = 'cifar10'):
        if name == 'cifar10':
            super(Net4, self).__init__()
            self.name = name
            self.c1 = nn.Conv2d(3, 6, 5)
            self.pool = nn.AvgPool2d(2, 2)
            self.c2 = nn.Conv2d(6, 16, 5)
            self.c3 = nn.Conv2d(16, 24, 5)
            self.fc3 = nn.Linear(16*5*5,10)

    def forward(self, value):

        value = self.pool(F.relu(self.c1(value)))
        value = self.pool(F.relu(self.c2(value)))
#         value = self.pool(F.relu(self.c3(value)))
        value = value.flatten(1)
        # value = F.relu(self.fc1(value))
        # value = F.relu(self.fc2(value))
        value = self.fc3(value)
        return value
```

[0]:
```
class losses(nn.Module):
    def __init__(self,loss_type = 'cross_entropy'):
        super(losses,self).__init__()
        self.loss_type = loss_type
        if(self.loss_type == 'l1'):
            self.p = 1
        elif(self.loss_type == 'l2'):
            self.p = 2
        elif(self.loss_type == 'cross_entropy'):
            self.criterion = nn.CrossEntropyLoss()


    def forward(self,outputs,labels):
        if(self.loss_type == 'l1' or self.loss_type == 'l2'):
            onh = torch.FloatTensor(len(labels),10).to(device)
            onh.zero_()
            onh.scatter_(1,labels.view(len(labels),1),1)
            arr = onh - F.softmax(outputs,dim=1)
            return torch.sum(torch.abs(arr).pow(self.p)/len(labels))
        elif(self.loss_type == 'cross_entropy'):
            return self.criterion(outputs,labels)
```

[0]:
```
net = Net()
```

[0]:
```
for name, param in net.named_parameters():
    if param.requires_grad:
        print(name)
```

```
c1.weight
c1.bias
c2.weight
c2.bias
c3.weight
c3.bias
fc1.weight
fc1.bias
fc2.weight
fc2.bias
fc3.weight
fc3.bias
```

[0]:
```
def correct(groundTruth,output):
  assert groundTruth.size(0) == output.size(0)
  _, indices = torch.max(output, 1)
  ch=torch.eq(groundTruth,indices)
  return torch.sum(ch).item()
```

[0]:
```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
def train_test(trainloader,testloader, lr = 1e-4,  momentum = 0.09, num_epochs␣
 ↪= 10, optimi = 'Adam', loss_type = 'cross_entropy', method = 1):
    trainAcc = []
    trainLoss = []
    testAcc = []
    testLoss = []
    if method == 1:

      net = Net().cuda()
    if method == 2:
      net = Net1().cuda()
    if method == 3:
      net = Net2().cuda()
    if method == 4:
      net = Net3().cuda()
    if method == 5:
      net = Net4().cuda()

    # if loss == 'CrossEntropy':
    #     criterion = nn.CrossEntropyLoss()
    criterion = losses(loss_type=loss_type)
```

```
    if optimi == 'Adam':
        optimizer = optim.Adam(net.parameters(), lr)
    if optimi == 'SGD':
      optimizer = optim.SGD(net.parameters(), lr = 1e-2, momentum = momentum)
    if optimi == 'Adagrad':
      optimizer = optim.Adagrad(net.parameters(), lr, lr_decay = 0)
    for epoch in range(num_epochs):
        per_epoch_loss = 0.0
        correct_train = 0
        net.train()
        inputs = None
        labels = None


        for i, data in enumerate(trainloader, 0):
            inputs, labels = data[0].to(device), data[1].to(device)
            optimizer.zero_grad()
            outputs = net(inputs)
            correct_train+=correct(labels, outputs)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()
            per_epoch_loss += loss.item()
        if epoch%2 == 0:
            print("Epoch", epoch, "Train loss",per_epoch_loss/len(trainloader.
↪dataset))
            print("Train Accuracy: ", round((correct_train/len(trainloader.
↪dataset))*100,2),"%" )
        trainAcc.append((correct_train/len(trainloader.dataset))*100)
        trainLoss.append(per_epoch_loss/len(trainloader.dataset))
        net.eval()
        per_epoch_loss = 0.0
        correct_test = 0
        inputs = None
        labels = None
        with torch.no_grad():
            for i, data in enumerate(testloader,0):
                inputs, labels = data[0].to(device), data[1].to(device)
                outputs = net(inputs)
                correct_test+= correct(labels, outputs)
                loss = criterion(outputs, labels)
                per_epoch_loss+= loss.item()
            if epoch % 2 == 0:
                print("Epoch", epoch, "Test loss",per_epoch_loss/len(testloader.
↪dataset))
                print("Test Accuracy: ", round((correct_test/len(testloader.
↪dataset))*100,2),"%" )
```

```
            x = (correct_test/len(testloader.dataset))*100
            testAcc.append(x)
            y = per_epoch_loss/len(testloader.dataset)
            testLoss.append(y)
    return trainAcc, trainLoss, testAcc, testLoss
```

1.2 Testing method 1:

- conv ->relu -> pool -> conv -> relu -> pool -> fc -> relu -> fc

[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, num_epochs = 30)
```

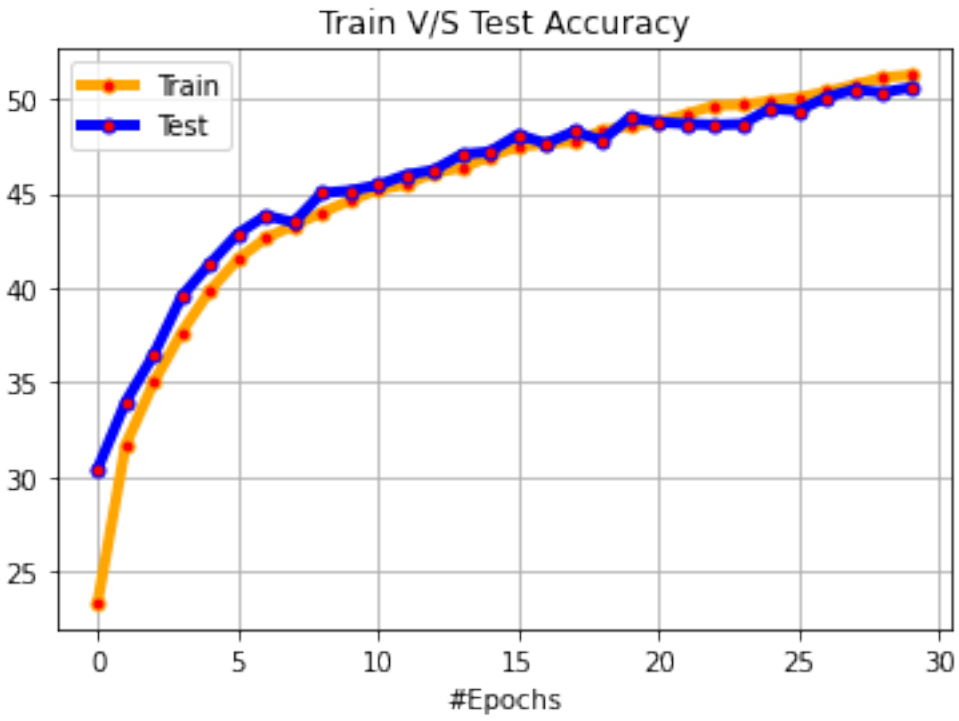
```
Epoch 0 Train loss 0.03235015404462814
Train Accuracy:  23.36 %
Epoch 0 Test loss 0.030265636432170867
Test Accuracy:  30.44 %
Epoch 2 Train loss 0.02821624587059021
Train Accuracy:  35.01 %
Epoch 2 Test loss 0.027574745738506316
Test Accuracy:  36.48 %
Epoch 4 Train loss 0.026082277474403382
Train Accuracy:  39.86 %
Epoch 4 Test loss 0.02558667792081833
Test Accuracy:  41.28 %
Epoch 6 Train loss 0.024904317450523378
Train Accuracy:  42.65 %
Epoch 6 Test loss 0.024611674332618714
Test Accuracy:  43.84 %
Epoch 8 Train loss 0.02427663903713226
Train Accuracy:  43.99 %
Epoch 8 Test loss 0.024036106383800507
Test Accuracy:  45.03 %
Epoch 10 Train loss 0.02379373528242111
Train Accuracy:  45.2 %
Epoch 10 Test loss 0.023694535052776338
Test Accuracy:  45.45 %
Epoch 12 Train loss 0.023414224724769592
Train Accuracy:  46.1 %
Epoch 12 Test loss 0.023550479054450988
Test Accuracy:  46.25 %
Epoch 14 Train loss 0.023074048223495482
Train Accuracy:  46.97 %
Epoch 14 Test loss 0.023021878898143767
Test Accuracy:  47.22 %
```

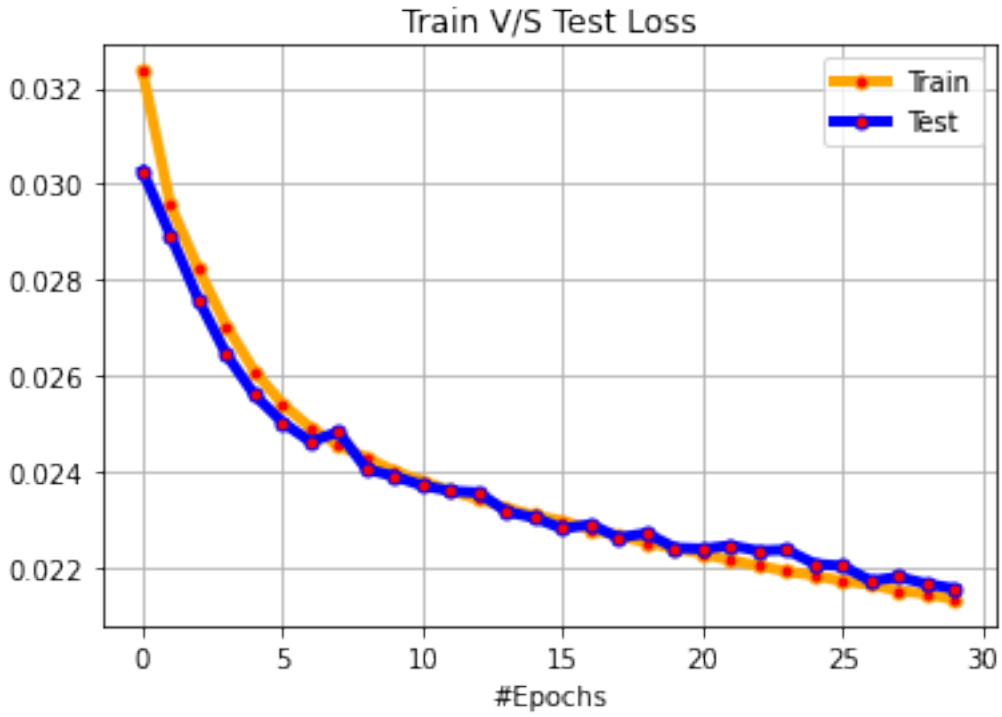
```
Epoch 16 Train loss 0.022785332221984863
Train Accuracy:  47.62 %
Epoch 16 Test loss 0.02288267995119095
Test Accuracy:  47.66 %
Epoch 18 Train loss 0.022509805448055267
Train Accuracy:  48.33 %
Epoch 18 Test loss 0.02269941408634186
Test Accuracy:  47.81 %
Epoch 20 Train loss 0.022271397910118104
Train Accuracy:  48.8 %
Epoch 20 Test loss 0.02237480527162552
Test Accuracy:  48.78 %
Epoch 22 Train loss 0.02201909184217453
Train Accuracy:  49.66 %
Epoch 22 Test loss 0.022334136962890626
Test Accuracy:  48.63 %
Epoch 24 Train loss 0.021816377363204957
Train Accuracy:  49.9 %
Epoch 24 Test loss 0.02206064977645874
Test Accuracy:  49.5 %
Epoch 26 Train loss 0.021626410913467408
Train Accuracy:  50.39 %
Epoch 26 Test loss 0.02170805959701538
Test Accuracy:  50.1 %
Epoch 28 Train loss 0.02143239373445511
Train Accuracy:  51.14 %
Epoch 28 Test loss 0.021643418538570405
Test Accuracy:  50.35 %
```

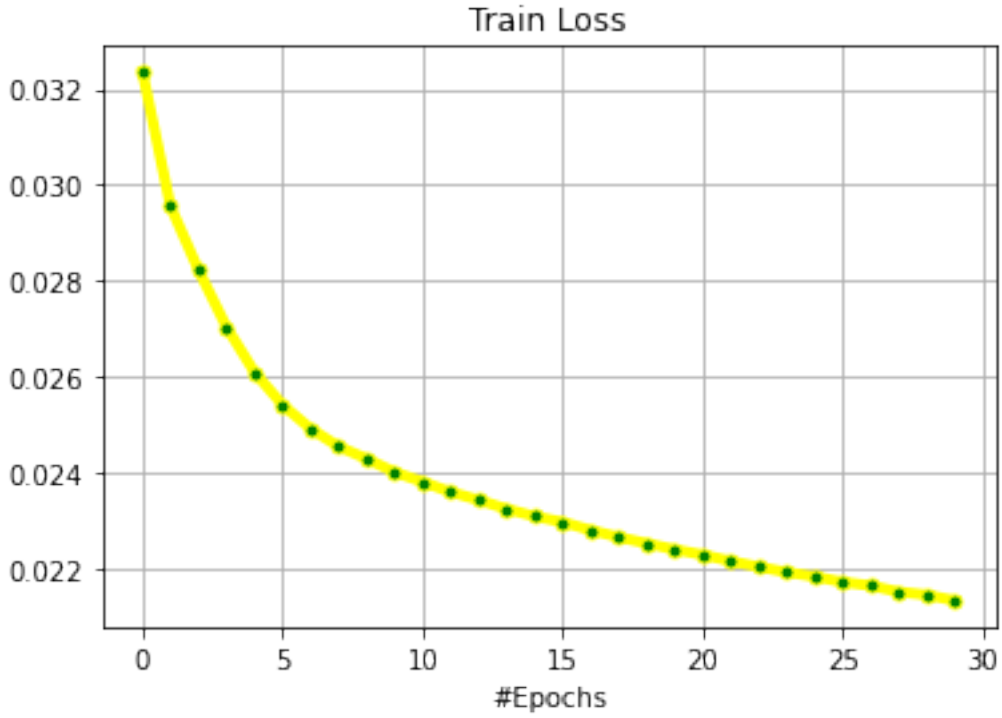
[0]:
```
def plot(trainAcc, trainLoss, testAcc, testLoss):
    fig1, ax1=  plt.subplots()
    plt.grid()
    ax1.plot(trainAcc,marker='o', markerfacecolor='red', markersize=5,␣
 ↪color='orange', linewidth=4)
    ax1.plot(testAcc,marker='o', markerfacecolor='red', markersize=5,␣
 ↪color='blue', linewidth=4)
    plt.gca().legend(('Train','Test'))
    ax1.set_title("Train V/S Test Accuracy")
    ax1.set_xlabel("#Epochs")

    fig2, ax2=  plt.subplots()
    plt.grid()
    ax2.plot(trainLoss,marker='o', markerfacecolor='red', markersize=5,␣
 ↪color='orange', linewidth=4,label="Train Loss")
    ax2.plot(testLoss,marker='o', markerfacecolor='red', markersize=5,␣
 ↪color='blue', linewidth=4,label="Test Loss")
    plt.gca().legend(('Train','Test'))
```

```
    ax2.set_title("Train V/S Test Loss")
    ax2.set_xlabel("#Epochs")

    fig3, ax3=  plt.subplots()
    plt.grid()
    ax3.plot(trainLoss,marker='o', markerfacecolor='green', markersize=5,␣
↪color='yellow', linewidth=4,label="Train Loss")
    ax3.set_title("Train Loss")
    ax3.set_xlabel("#Epochs")
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.032
0.030
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs


Train Loss
0.032
0.030
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs

1.3 Method 2

- conv -> relu -> conv -> relu -> fc
- kaiming initialization used
- This model has better accuracy due to the better initialization of weights.
- But the model overfits after 25 epochs since it learns the training data over the large number of epochs.

[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 2, num_epochs = 70)
```

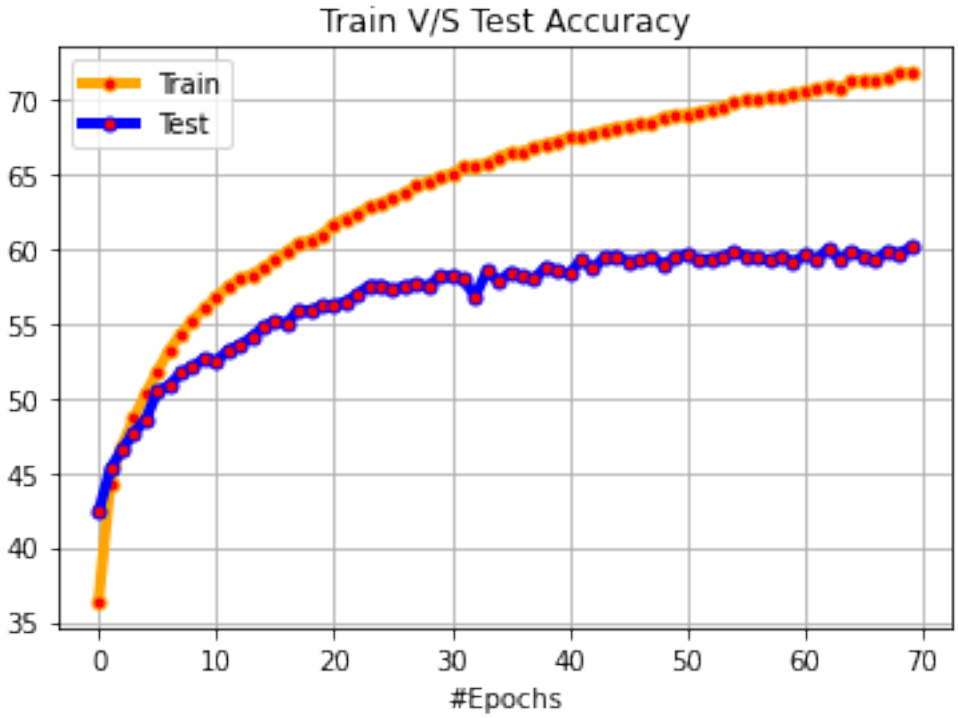
```
Epoch 0 Train loss 0.02831036660671234
Train Accuracy:  36.37 %
Epoch 0 Test loss 0.025647568798065186
Test Accuracy:  42.37 %
Epoch 2 Train loss 0.023714544568061827
Train Accuracy:  46.83 %
Epoch 2 Test loss 0.023781033051013947
Test Accuracy:  46.56 %
Epoch 4 Train loss 0.022228165736198426
Train Accuracy:  50.29 %
Epoch 4 Test loss 0.022537470316886903
Test Accuracy:  48.54 %
Epoch 6 Train loss 0.02100671712875366
Train Accuracy:  53.24 %
Epoch 6 Test loss 0.021666427397727967
Test Accuracy:  50.85 %
Epoch 8 Train loss 0.020136422839164735
Train Accuracy:  55.26 %
Epoch 8 Test loss 0.02112112160921097
Test Accuracy:  52.09 %
Epoch 10 Train loss 0.01948282604575157
Train Accuracy:  56.79 %
Epoch 10 Test loss 0.020740206122398377
Test Accuracy:  52.47 %
Epoch 12 Train loss 0.01900964033842087
Train Accuracy:  57.99 %
Epoch 12 Test loss 0.02040280134677887
Test Accuracy:  53.53 %
Epoch 14 Train loss 0.018570324103832244
Train Accuracy:  58.74 %
Epoch 14 Test loss 0.020119557869434358
Test Accuracy:  54.76 %
Epoch 16 Train loss 0.018143667109012603
Train Accuracy:  59.84 %
Epoch 16 Test loss 0.019909435099363326
Test Accuracy:  55.04 %
```

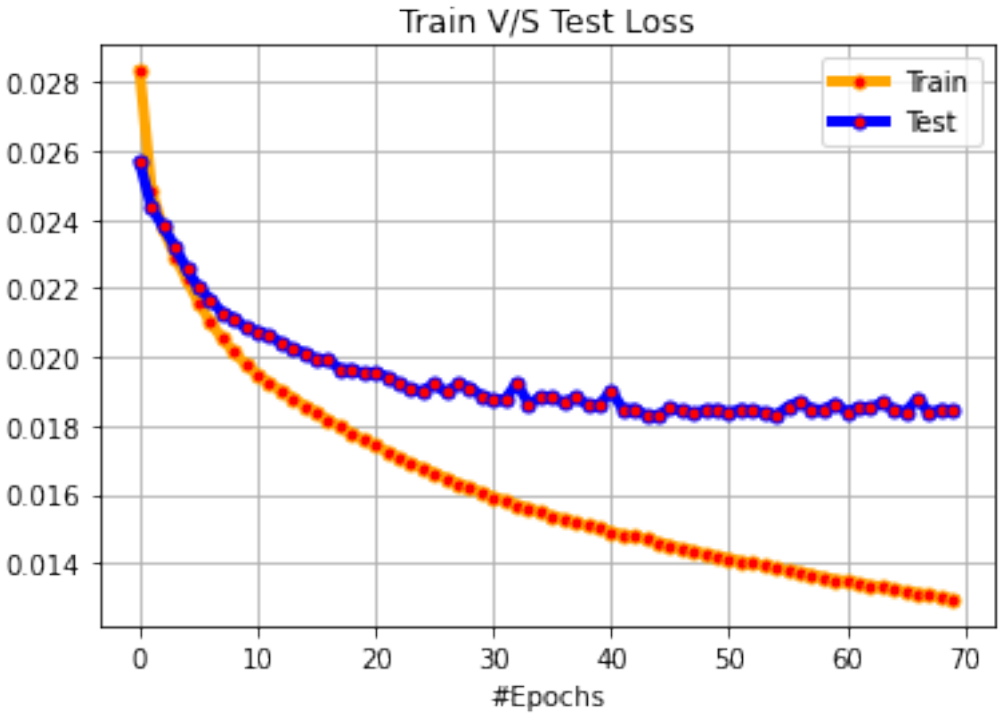
```
Epoch 18 Train loss 0.01776634763240814
Train Accuracy:  60.58 %
Epoch 18 Test loss 0.019645626240968703
Test Accuracy:  55.85 %
Epoch 20 Train loss 0.017420401935577392
Train Accuracy:  61.7 %
Epoch 20 Test loss 0.019551951122283934
Test Accuracy:  56.22 %
Epoch 22 Train loss 0.017056869279146195
Train Accuracy:  62.35 %
Epoch 22 Test loss 0.019214443892240526
Test Accuracy:  56.96 %
Epoch 24 Train loss 0.016769322106838225
Train Accuracy:  63.05 %
Epoch 24 Test loss 0.019010075390338898
Test Accuracy:  57.59 %
Epoch 26 Train loss 0.016471438282728196
Train Accuracy:  63.71 %
Epoch 26 Test loss 0.018962374037504195
Test Accuracy:  57.53 %
Epoch 28 Train loss 0.016189812077283858
Train Accuracy:  64.51 %
Epoch 28 Test loss 0.019100997692346574
Test Accuracy:  57.52 %
Epoch 30 Train loss 0.01592112161040306
Train Accuracy:  65.07 %
Epoch 30 Test loss 0.01874926416873932
Test Accuracy:  58.16 %
Epoch 32 Train loss 0.015681524640321732
Train Accuracy:  65.64 %
Epoch 32 Test loss 0.01923450807929039
Test Accuracy:  56.83 %
Epoch 34 Train loss 0.01547845838546753
Train Accuracy:  66.17 %
Epoch 34 Test loss 0.018825976794958114
Test Accuracy:  57.81 %
Epoch 36 Train loss 0.015287419283390046
Train Accuracy:  66.52 %
Epoch 36 Test loss 0.018663056468963622
Test Accuracy:  58.16 %
Epoch 38 Train loss 0.015080638095140457
Train Accuracy:  67.05 %
Epoch 38 Test loss 0.018607309967279433
Test Accuracy:  58.81 %
Epoch 40 Train loss 0.014916133044958115
Train Accuracy:  67.5 %
Epoch 40 Test loss 0.019009538757801057
Test Accuracy:  58.33 %
```

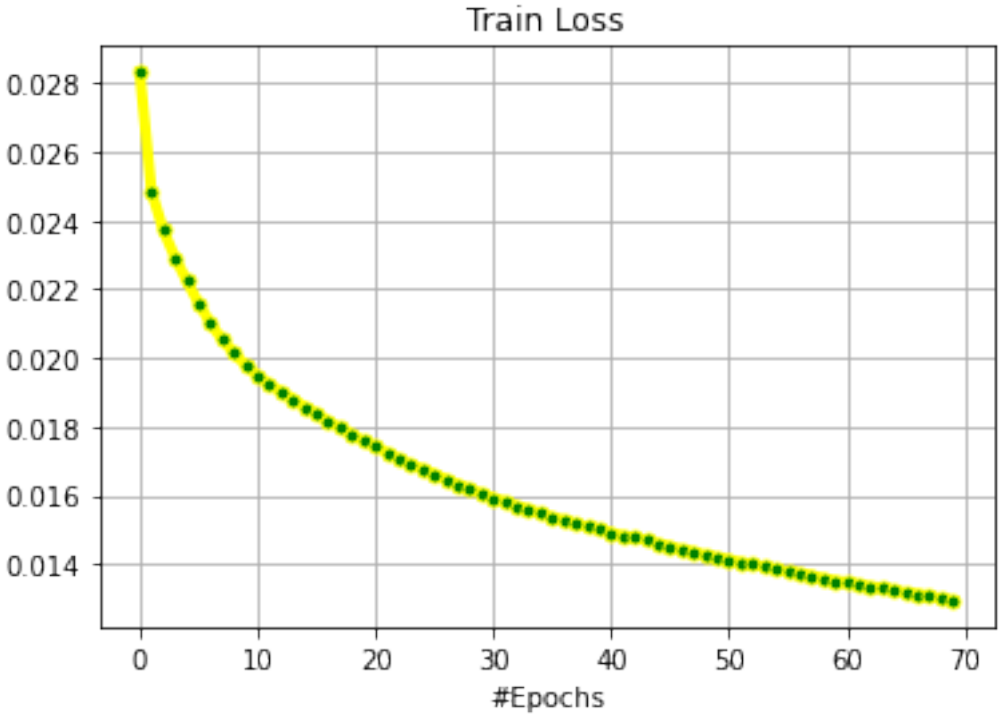
```
Epoch 42 Train loss 0.014777967046499252
Train Accuracy:  67.72 %
Epoch 42 Test loss 0.01849119753241539
Test Accuracy:  58.7 %
Epoch 44 Train loss 0.01459641054391861
Train Accuracy:  68.16 %
Epoch 44 Test loss 0.018297438108921052
Test Accuracy:  59.54 %
Epoch 46 Train loss 0.014423508150577545
Train Accuracy:  68.37 %
Epoch 46 Test loss 0.01845574235916138
Test Accuracy:  59.21 %
Epoch 48 Train loss 0.014261150629520417
Train Accuracy:  68.85 %
Epoch 48 Test loss 0.018462719863653183
Test Accuracy:  58.96 %
Epoch 50 Train loss 0.014139362872838974
Train Accuracy:  69.06 %
Epoch 50 Test loss 0.01835908655524254
Test Accuracy:  59.67 %
Epoch 52 Train loss 0.014003142036199569
Train Accuracy:  69.39 %
Epoch 52 Test loss 0.018432982337474823
Test Accuracy:  59.33 %
Epoch 54 Train loss 0.013855498174428939
Train Accuracy:  69.78 %
Epoch 54 Test loss 0.01829025276899338
Test Accuracy:  59.78 %
Epoch 56 Train loss 0.013738523424863815
Train Accuracy:  69.96 %
Epoch 56 Test loss 0.01869979835152626
Test Accuracy:  59.47 %
Epoch 58 Train loss 0.013589891167879104
Train Accuracy:  70.23 %
Epoch 58 Test loss 0.01844962556362152
Test Accuracy:  59.55 %
Epoch 60 Train loss 0.013484458093643188
Train Accuracy:  70.52 %
Epoch 60 Test loss 0.01840228548645973
Test Accuracy:  59.58 %
Epoch 62 Train loss 0.013348370950222016
Train Accuracy:  70.92 %
Epoch 62 Test loss 0.018502323937416076
Test Accuracy:  60.01 %
Epoch 64 Train loss 0.013239867659211159
Train Accuracy:  71.25 %
Epoch 64 Test loss 0.01845896544456482
Test Accuracy:  59.84 %
```

```
Epoch 66 Train loss 0.013127960882186889
Train Accuracy:  71.36 %
Epoch 66 Test loss 0.018786196702718735
Test Accuracy:  59.22 %
Epoch 68 Train loss 0.012994797644019127
Train Accuracy:  71.78 %
Epoch 68 Test loss 0.01848335019350052
Test Accuracy:  59.69 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0
10
20
30
40
50
60
70
#Epochs


Train Loss
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0
10
20
30
40
50
60
70
#Epochs

1.4 Varying Optimizers

- Adam
- Stochastic Gradient Descent with Nesterov Momentum = 0.9
- Adagrad

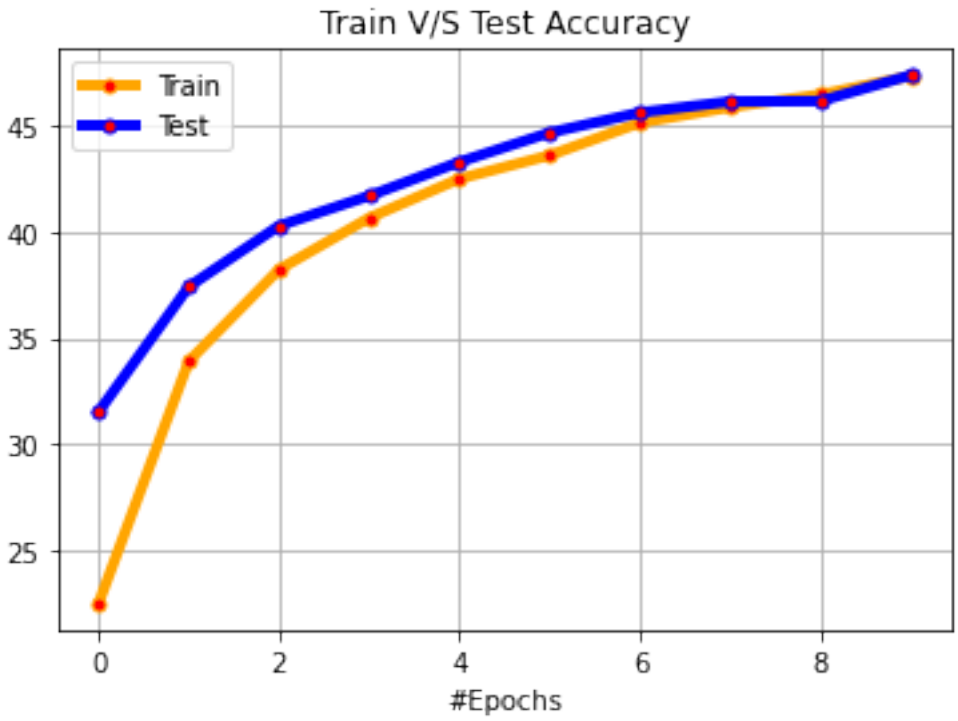
1.5 Adam

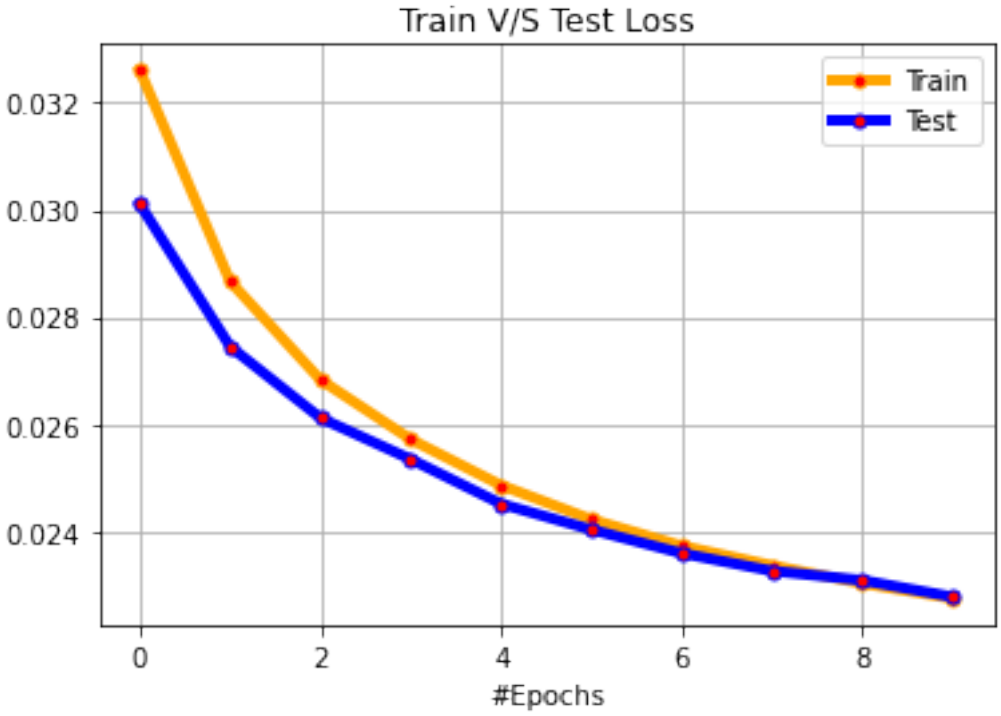
Works as expected, accuracy ~ 45% for 10 epochs

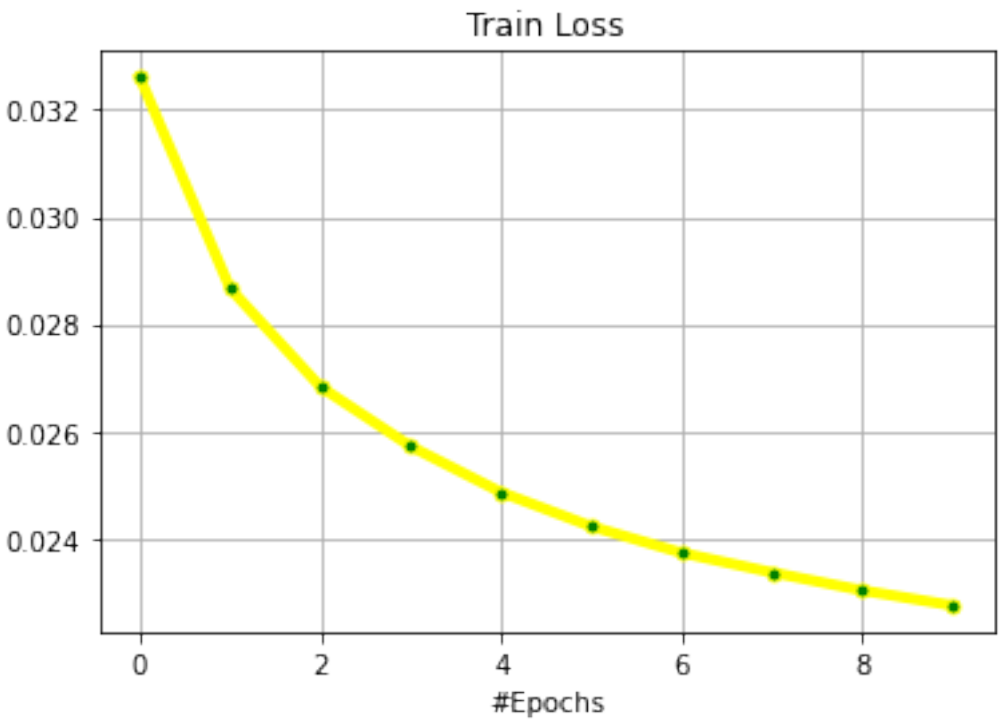
[0]:
```
# Cross Entropy Loss
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 1)
```

```
Epoch 0 Train loss 0.032455288591384884
Train Accuracy:  23.1 %
Epoch 0 Test loss 0.03083781954050064
Test Accuracy:  28.06 %
Epoch 2 Train loss 0.028523230712413788
Train Accuracy:  34.24 %
Epoch 2 Test loss 0.0278296905875206
Test Accuracy:  36.37 %
Epoch 4 Train loss 0.02663347945690155
Train Accuracy:  38.66 %
Epoch 4 Test loss 0.026236345112323763
Test Accuracy:  39.69 %
Epoch 6 Train loss 0.025413034670352937
Train Accuracy:  41.34 %
Epoch 6 Test loss 0.025081539833545684
Test Accuracy:  42.47 %
Epoch 8 Train loss 0.024536538660526276
Train Accuracy:  43.19 %
Epoch 8 Test loss 0.0244231835603714
Test Accuracy:  43.76 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```


Train V/S Test Accuracy
Train
Test
45
40
35
30
25
0
2
4
6
8
#Epochs


Train V/S Test Loss
Train
Test
0.032
0.030
0.028
0.026
0.024
0
2
4
6
8
#Epochs



1.6 Stochastic Gradient Descent

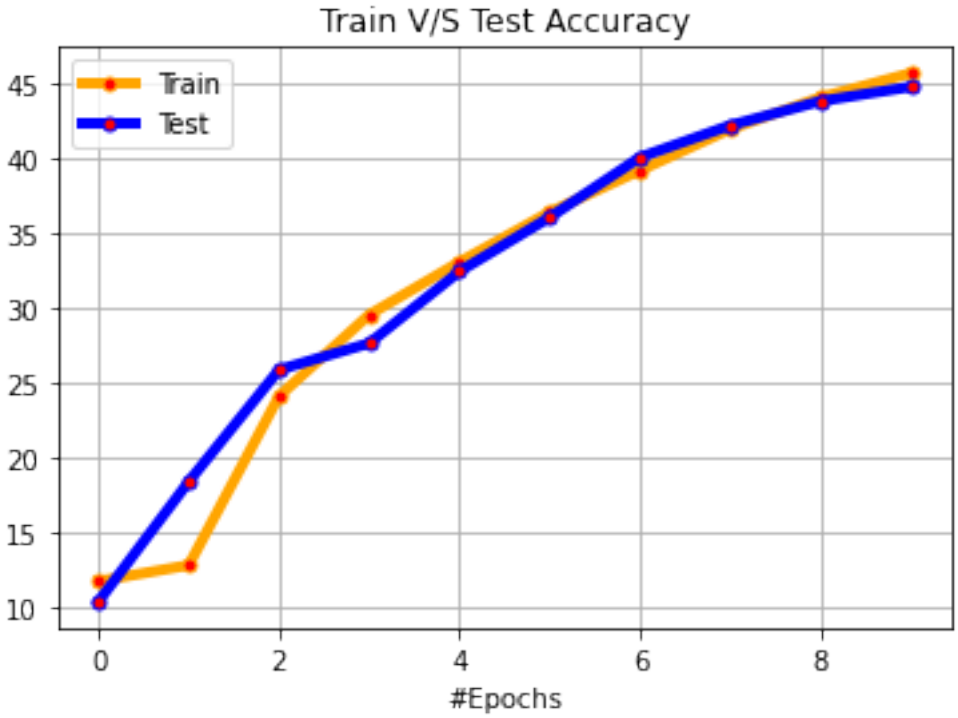
- When using learning rate 1e-4, no learning takes place, and it remains at 10%, which is random classification
- Hence, the learning rate was changed to 1e-2, and a good accuracy was received for 10 epochs.
- Not as good as Adam for this particular task

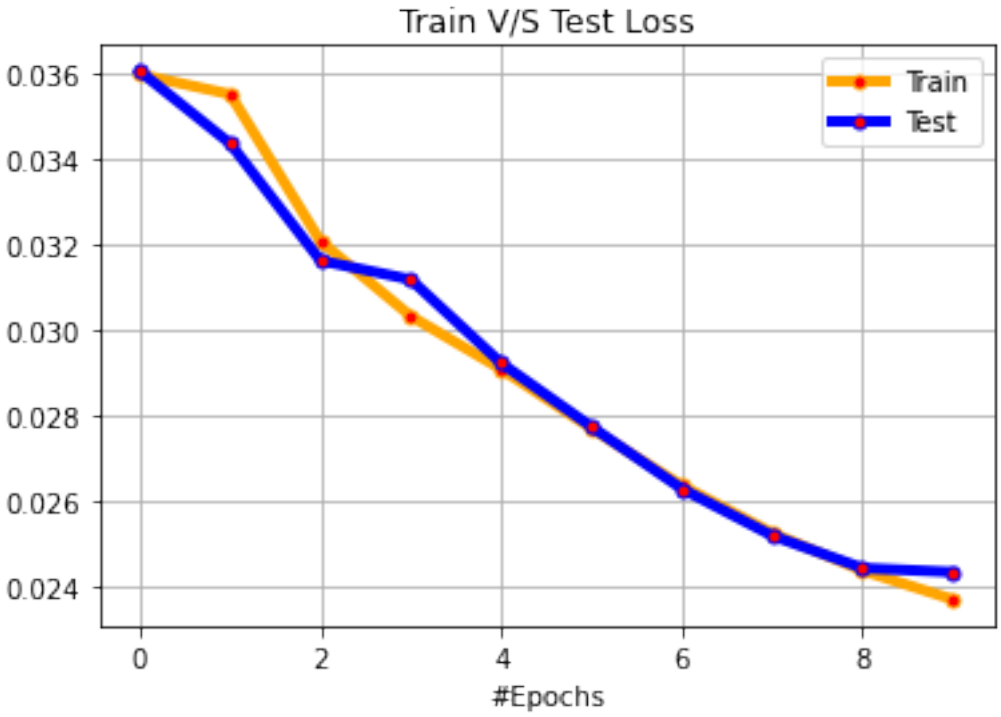
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader,optimi = 'SGD')
```

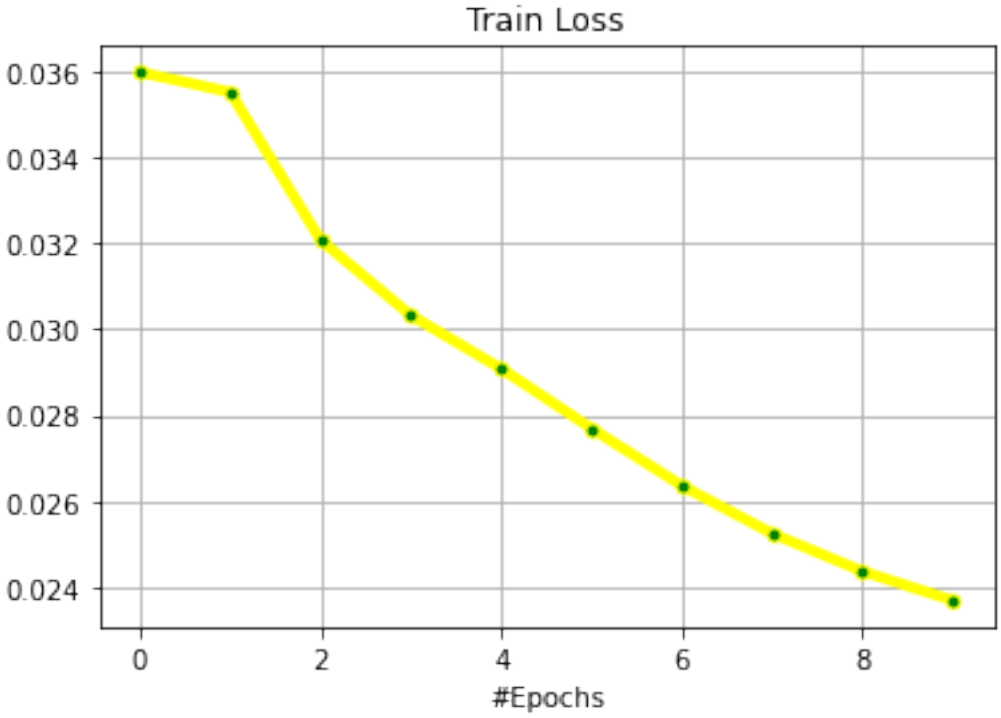
```
Epoch 0 Train loss 0.03599019549846649
Train Accuracy:  11.77 %
Epoch 0 Test loss 0.03606726760864258
Test Accuracy:  10.41 %
Epoch 2 Train loss 0.0320806671333313
Train Accuracy:  24.1 %
Epoch 2 Test loss 0.03162817757129669
Test Accuracy:  25.87 %
Epoch 4 Train loss 0.029084838860034944
Train Accuracy:  33.06 %
Epoch 4 Test loss 0.02922531386613846
Test Accuracy:  32.43 %
```

```
Epoch 6 Train loss 0.02635736324310303
Train Accuracy:  39.17 %
Epoch 6 Test loss 0.026279090547561647
Test Accuracy:  40.05 %
Epoch 8 Train loss 0.024362156608104705
Train Accuracy:  44.04 %
Epoch 8 Test loss 0.024422195732593535
Test Accuracy:  43.79 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.036
0.034
0.032
0.030
0.028
0.026
0.024
0
2
4
6
8
#Epochs


Train Loss
0.036
0.034
0.032
0.030
0.028
0.026
0.024
0
2
4
6
8
#Epochs

1.7 Adagrad

- When a learning rate of 1e-4 is used, accuracy is ~ 16%.
- When learning rate is increased to 1e-2, accuracy considerably increases to ~ 50% for 10 epochs.

[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader,optimi = 'Adagrad')
```

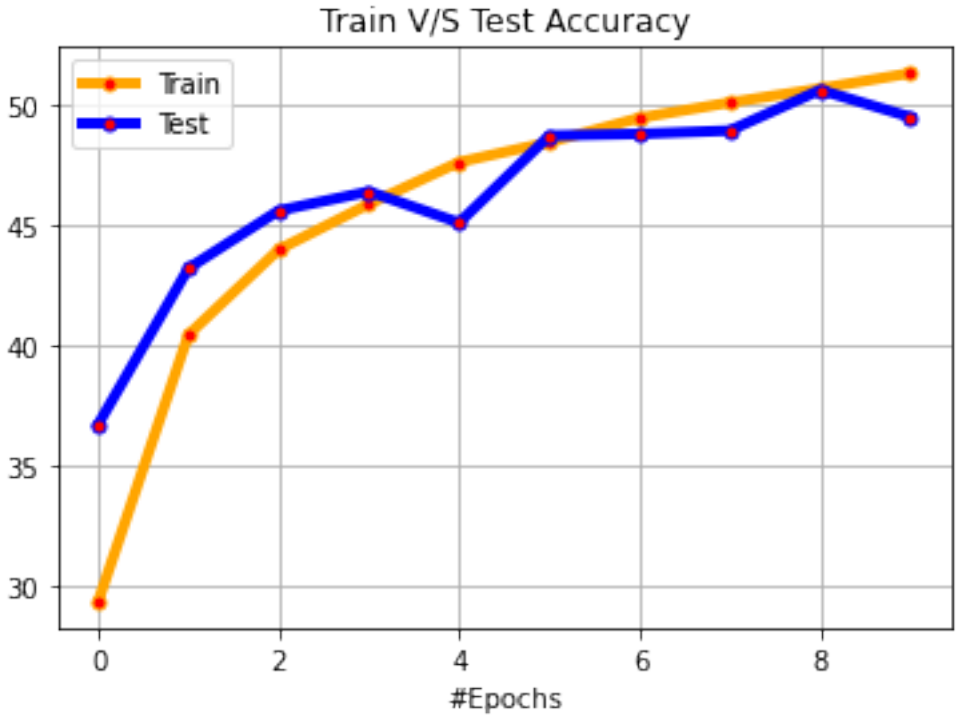
```
Epoch 0 Train loss 0.03600227933883667
Train Accuracy:  10.0 %
Epoch 0 Test loss 0.03609655613899231
Test Accuracy:  10.0 %
Epoch 2 Train loss 0.035766343007087704
Train Accuracy:  15.05 %
Epoch 2 Test loss 0.035822373747825625
Test Accuracy:  14.8 %
Epoch 4 Train loss 0.03544133864879608
Train Accuracy:  13.42 %
Epoch 4 Test loss 0.03547430272102356
Test Accuracy:  13.28 %
Epoch 6 Train loss 0.03504983952999115
Train Accuracy:  13.01 %
Epoch 6 Test loss 0.035072908067703246
Test Accuracy:  13.23 %
Epoch 8 Train loss 0.03467727119922638
Train Accuracy:  16.17 %
Epoch 8 Test loss 0.03469627468585968
Test Accuracy:  16.93 %
```

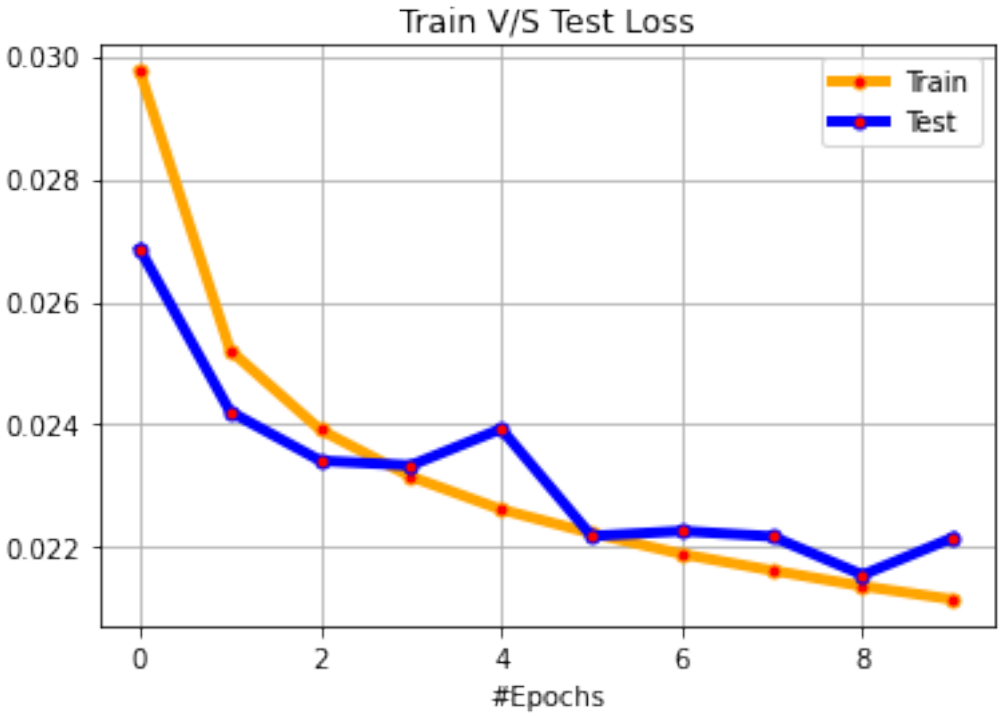
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader,optimi = 'Adagrad', lr = 0.01)
```

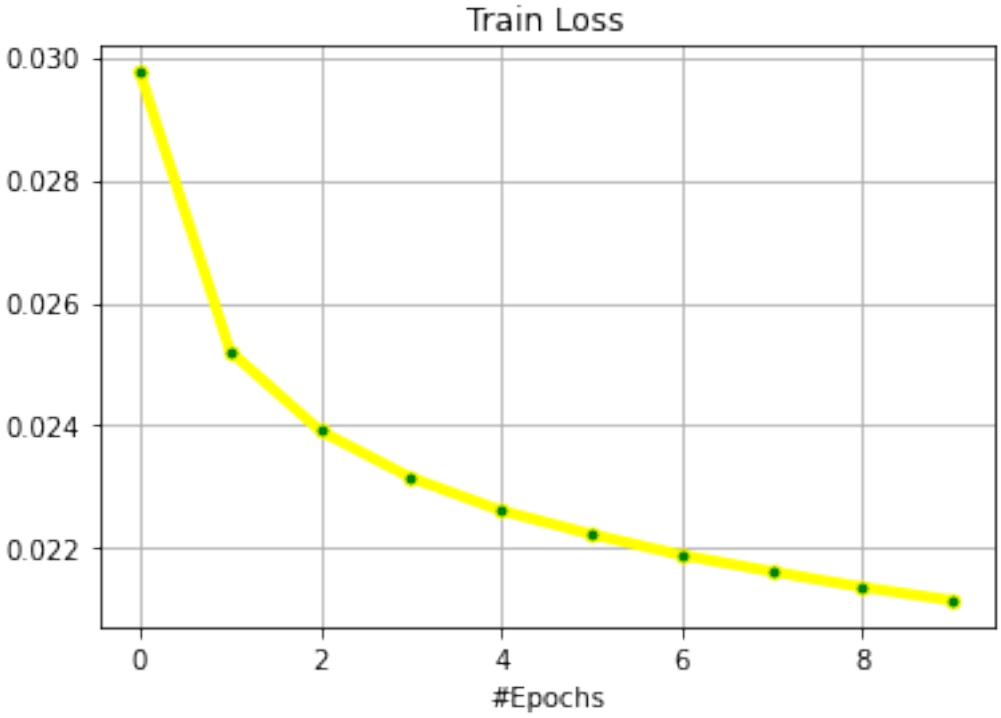
```
Epoch 0 Train loss 0.029755968663692473
Train Accuracy:  29.38 %
Epoch 0 Test loss 0.02685710277557373
Test Accuracy:  36.72 %
Epoch 2 Train loss 0.0239123738861084
Train Accuracy:  43.97 %
Epoch 2 Test loss 0.02341057974100113
Test Accuracy:  45.58 %
Epoch 4 Train loss 0.022616039016246797
Train Accuracy:  47.59 %
Epoch 4 Test loss 0.023929133760929106
Test Accuracy:  45.08 %
Epoch 6 Train loss 0.021890447549819947
```

```
Train Accuracy:  49.42 %
Epoch 6 Test loss 0.022265618538856505
Test Accuracy:  48.76 %
Epoch 8 Train loss 0.02136837214946747
Train Accuracy:  50.64 %
Epoch 8 Test loss 0.02155127100944519
Test Accuracy:  50.58 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.030
0.028
0.026
0.024
0.022
0
2
4
6
8
#Epochs


Train Loss
0.030
0.028
0.026
0.024
0.022
0
2
4
6
8
#Epochs

1.8 Adding fully connected layers

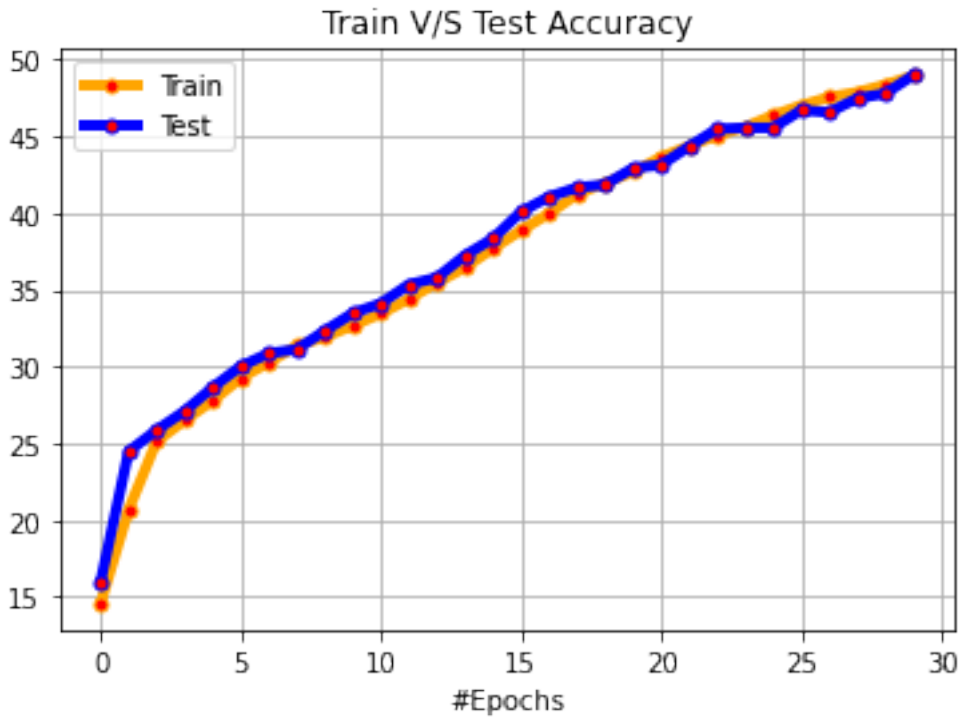
- We add 4 more fully connected layers, making it a total of 7
- We can see that both the training and testing loss are not increasing as much for the same number of epochs.
- This implies that the model is not learning as fast due to the increased number of parameters it has to train.

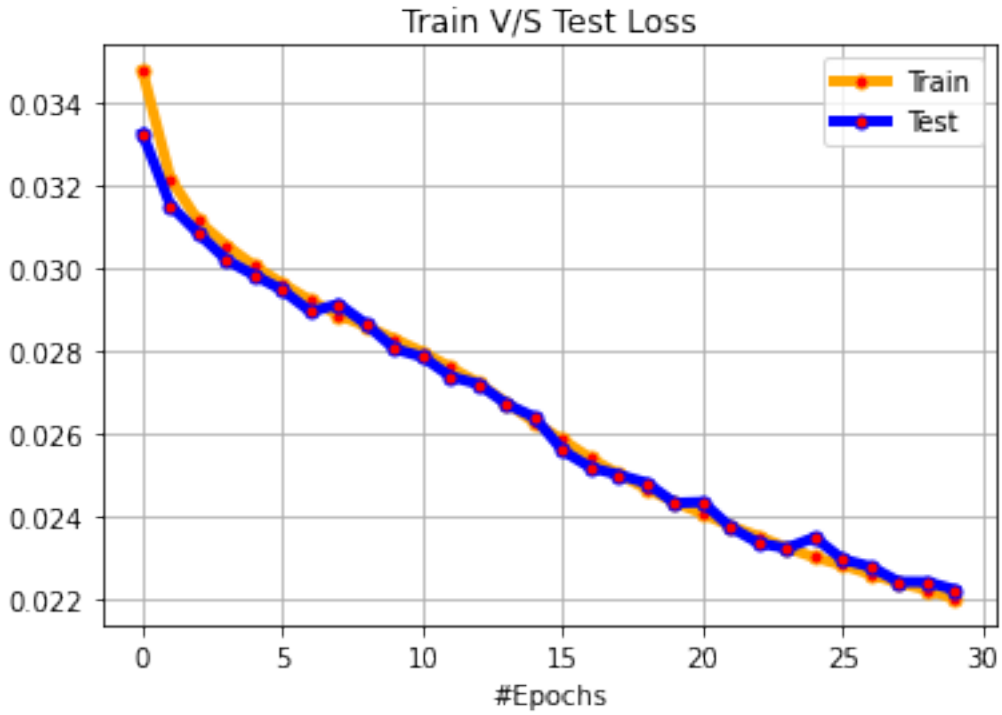
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 3, num_epochs = 30)
```

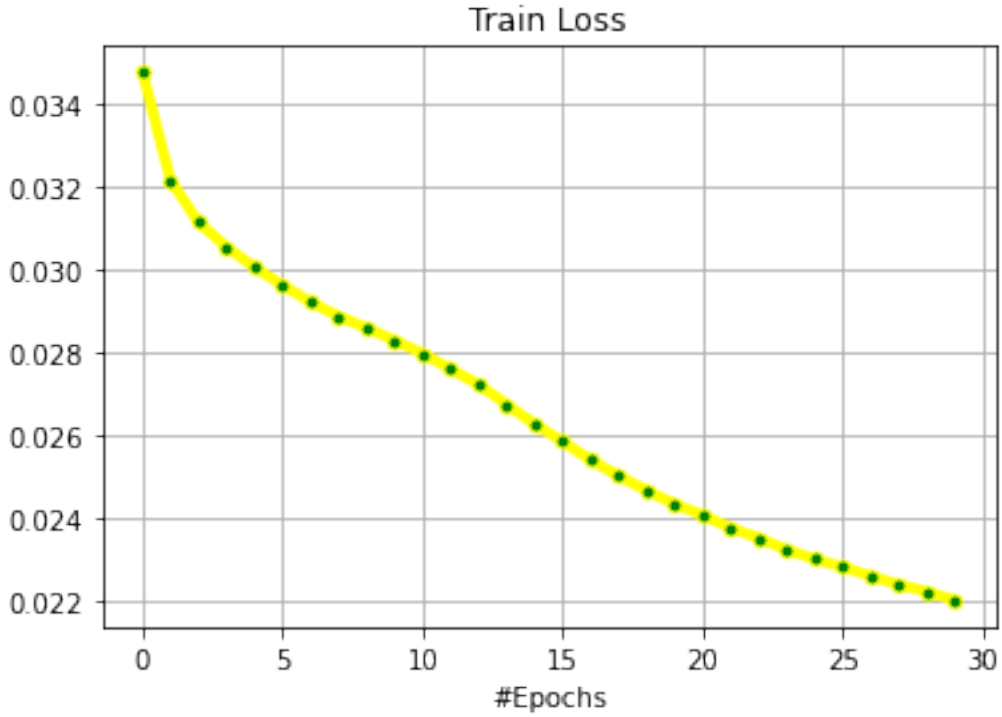
```
Epoch 0 Train loss 0.03475888429164886
Train Accuracy:  14.56 %
Epoch 0 Test loss 0.03324731366634369
Test Accuracy:  15.98 %
Epoch 2 Train loss 0.031134864294528962
Train Accuracy:  25.12 %
Epoch 2 Test loss 0.030835563039779662
Test Accuracy:  25.86 %
Epoch 4 Train loss 0.030053937683105467
Train Accuracy:  27.77 %
Epoch 4 Test loss 0.02982536404132843
Test Accuracy:  28.64 %
Epoch 6 Train loss 0.029219264731407166
Train Accuracy:  30.27 %
Epoch 6 Test loss 0.028946864330768585
Test Accuracy:  30.86 %
Epoch 8 Train loss 0.02859239072084427
Train Accuracy:  31.91 %
Epoch 8 Test loss 0.02864276648759842
Test Accuracy:  32.33 %
Epoch 10 Train loss 0.027974882340431214
Train Accuracy:  33.44 %
Epoch 10 Test loss 0.027858073580265045
Test Accuracy:  34.11 %
Epoch 12 Train loss 0.027226446244716644
Train Accuracy:  35.46 %
Epoch 12 Test loss 0.027201344347000122
Test Accuracy:  35.79 %
Epoch 14 Train loss 0.026276116499900817
Train Accuracy:  37.69 %
Epoch 14 Test loss 0.026402254414558412
Test Accuracy:  38.34 %
Epoch 16 Train loss 0.025409018359184264
Train Accuracy:  39.96 %
Epoch 16 Test loss 0.025185823464393617
Test Accuracy:  41.06 %
```

```
Epoch 18 Train loss 0.024672216272354126
Train Accuracy:  41.96 %
Epoch 18 Test loss 0.024817278575897217
Test Accuracy:  41.85 %
Epoch 20 Train loss 0.024075806136131285
Train Accuracy:  43.64 %
Epoch 20 Test loss 0.024354607260227205
Test Accuracy:  43.13 %
Epoch 22 Train loss 0.023524482069015504
Train Accuracy:  45.05 %
Epoch 22 Test loss 0.023363175892829897
Test Accuracy:  45.48 %
Epoch 24 Train loss 0.023017869918346406
Train Accuracy:  46.37 %
Epoch 24 Test loss 0.023500524723529814
Test Accuracy:  45.55 %
Epoch 26 Train loss 0.022612378101348876
Train Accuracy:  47.55 %
Epoch 26 Test loss 0.022793904852867128
Test Accuracy:  46.56 %
Epoch 28 Train loss 0.02223988809347153
Train Accuracy:  48.35 %
Epoch 28 Test loss 0.022428051161766053
Test Accuracy:  47.78 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.034
0.032
0.030
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs


Train Loss
0.034
0.032
0.030
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs

1.9 Adding batch normalization

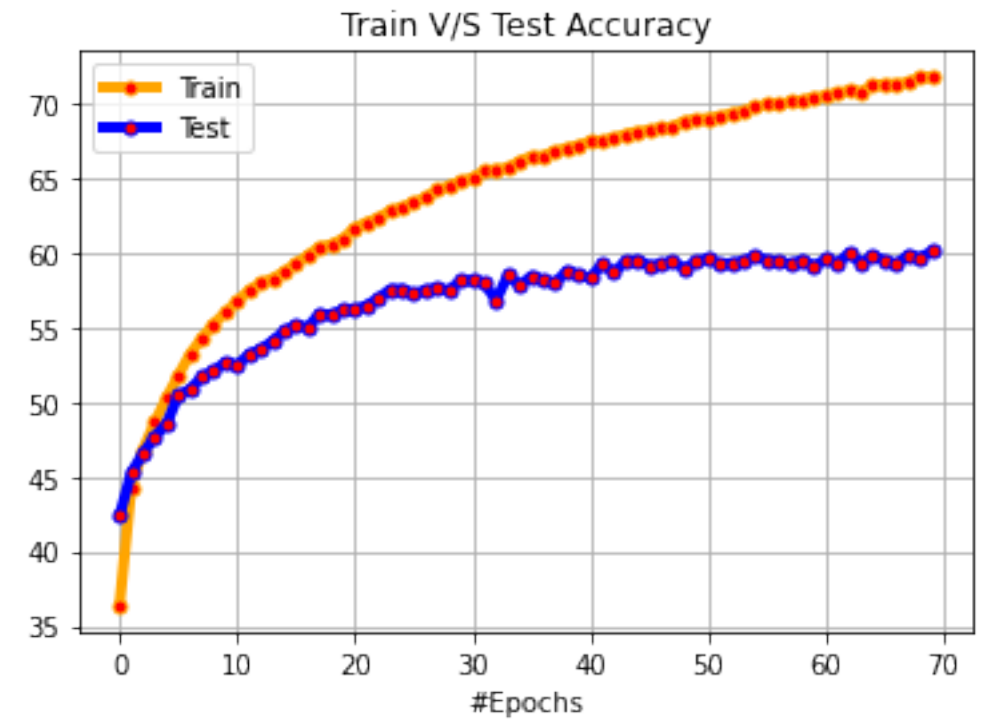
- We add batch normalization to the preexisting model of method 1.
- The model's training accuracy increases due to the addition of the two batch normalization layers
- The batch normalization allows each layer of the network to learn more independently, hence making the model better.
- Model starts to overfit around 40 epochs

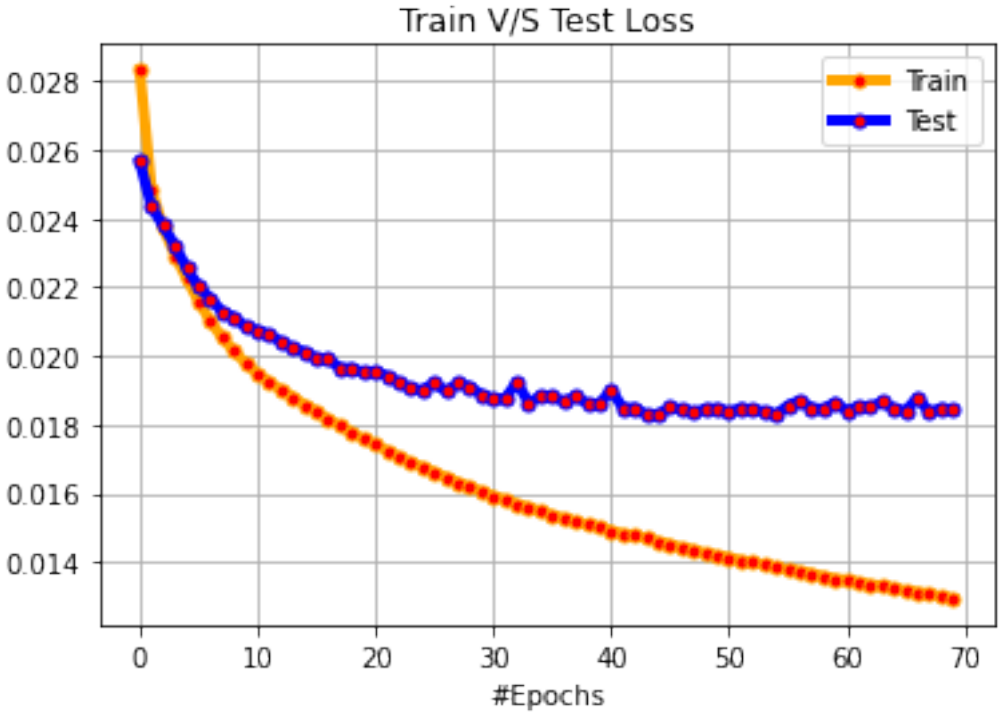
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 4, num_epochs = 30)
```

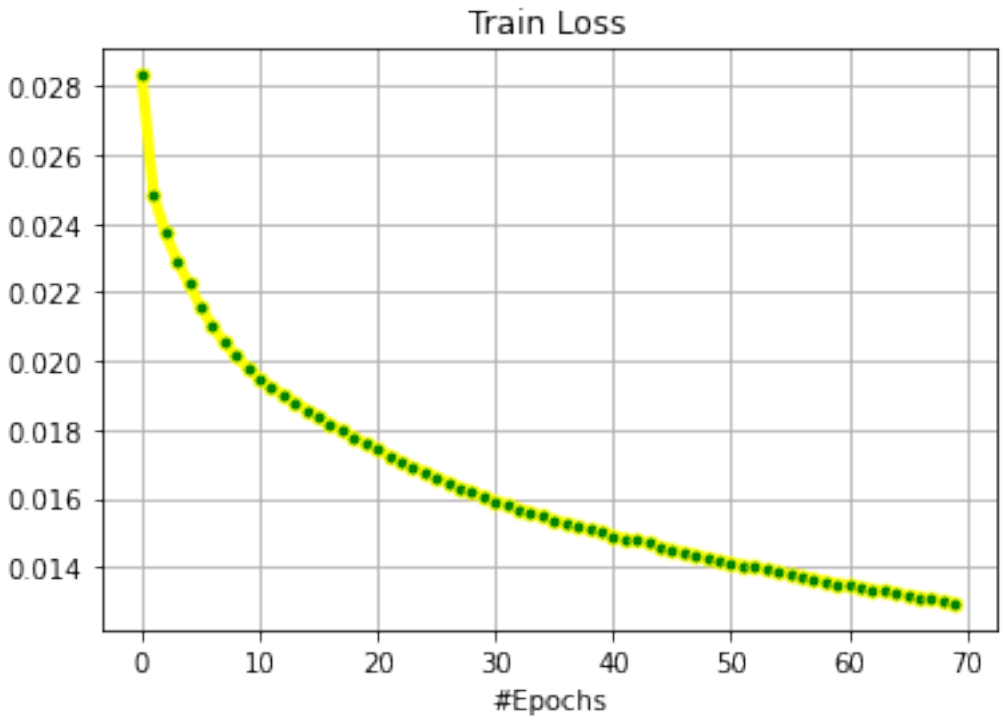
```
Epoch 0 Train loss 0.027917139530181885
Train Accuracy:  35.07 %
Epoch 0 Test loss 0.024068545293807985
Test Accuracy:  44.15 %
Epoch 2 Train loss 0.021506977634429932
Train Accuracy:  50.74 %
Epoch 2 Test loss 0.0209488321185112
Test Accuracy:  52.25 %
Epoch 4 Train loss 0.019615175763368606
Train Accuracy:  55.29 %
Epoch 4 Test loss 0.019947964930534363
Test Accuracy:  54.53 %
Epoch 6 Train loss 0.018599477890729903
Train Accuracy:  57.5 %
Epoch 6 Test loss 0.01908717193007469
Test Accuracy:  57.47 %
Epoch 8 Train loss 0.017855527951717378
Train Accuracy:  59.55 %
Epoch 8 Test loss 0.01837403621673584
Test Accuracy:  58.86 %
Epoch 10 Train loss 0.0172367767226696
Train Accuracy:  60.93 %
Epoch 10 Test loss 0.018369607251882553
Test Accuracy:  58.45 %
Epoch 12 Train loss 0.016702423726320265
Train Accuracy:  62.29 %
Epoch 12 Test loss 0.017879072564840316
Test Accuracy:  59.54 %
Epoch 14 Train loss 0.01625943939805031
Train Accuracy:  63.31 %
Epoch 14 Test loss 0.017618257963657378
```

```
Test Accuracy:  60.22 %
Epoch 16 Train loss 0.015855387351512908
Train Accuracy:  64.43 %
Epoch 16 Test loss 0.01746900560259819
Test Accuracy:  60.25 %
Epoch 18 Train loss 0.015497917870283127
Train Accuracy:  65.07 %
Epoch 18 Test loss 0.017683476507663727
Test Accuracy:  59.98 %
Epoch 20 Train loss 0.015117243509292602
Train Accuracy:  65.91 %
Epoch 20 Test loss 0.01682152062058449
Test Accuracy:  62.03 %
Epoch 22 Train loss 0.014830109993219376
Train Accuracy:  66.46 %
Epoch 22 Test loss 0.01678799704313278
Test Accuracy:  62.24 %
Epoch 24 Train loss 0.014491198724508285
Train Accuracy:  67.46 %
Epoch 24 Test loss 0.016992759639024736
Test Accuracy:  61.79 %
Epoch 26 Train loss 0.014193999980688096
Train Accuracy:  68.05 %
Epoch 26 Test loss 0.01712386245727539
Test Accuracy:  62.1 %
Epoch 28 Train loss 0.013910324320793152
Train Accuracy:  68.64 %
Epoch 28 Test loss 0.01620204409956932
Test Accuracy:  63.61 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```


Train V/S Test Accuracy
Train
Test
70
65
60
55
50
45
40
35
0
10
20
30
40
50
60
70
#Epochs


Train V/S Test Loss
Train
Test
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0
10
20
30
40
50
60
70
#Epochs



1.10 Removing Fully Connected layers

- On removing all fully connected layers, no real classification is seen
- After this, we remove all but one fully connected layer from the first model to see the need of fully connected layers in a CNN, and we get a sharp increase in accuracy.
- This implies that fully connected layers are of prime importance so that the input features are encapsulated.
- Hence, the fully connected layers over time help in capturing the elements of the input required for classification.
- The increase in FC layers increases classification at the risk of overfitting

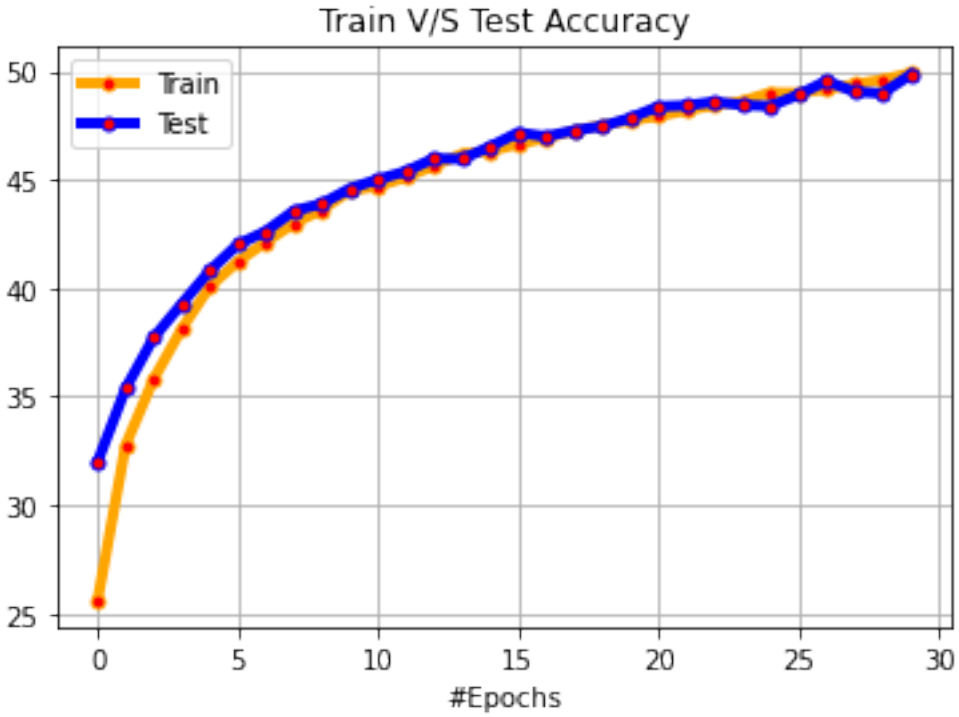
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 5, num_epochs = 30)
```

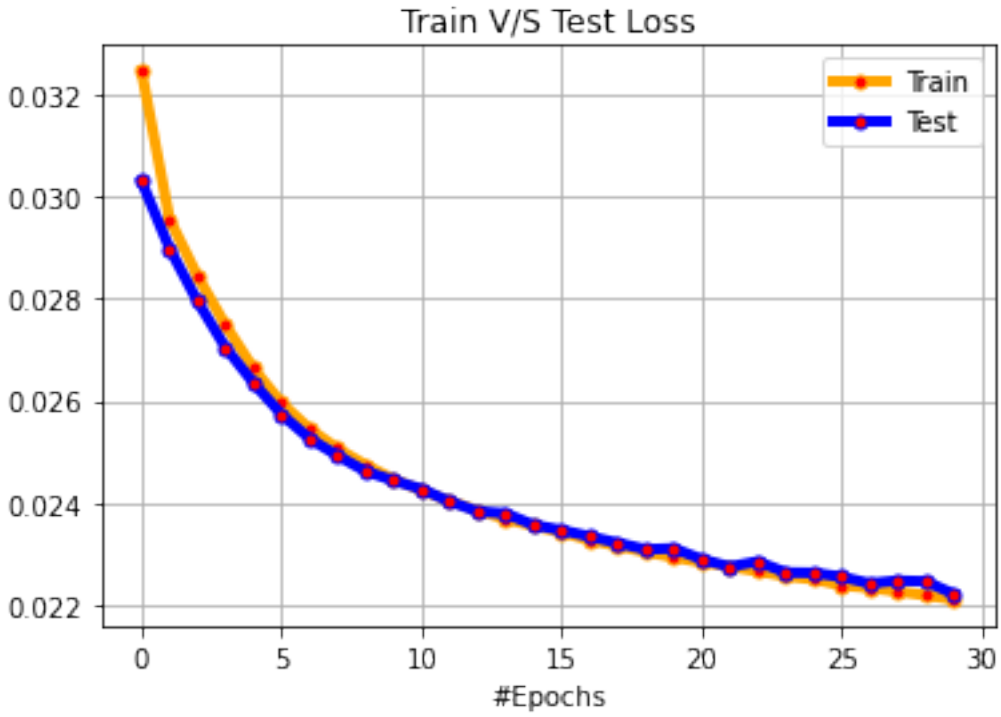
```
Epoch 0 Train loss 0.03245091633081436
Train Accuracy:  25.59 %
Epoch 0 Test loss 0.030293566823005675
Test Accuracy:  31.99 %
Epoch 2 Train loss 0.02843973103761673
Train Accuracy:  35.8 %
Epoch 2 Test loss 0.027943662679195402
Test Accuracy:  37.72 %
```

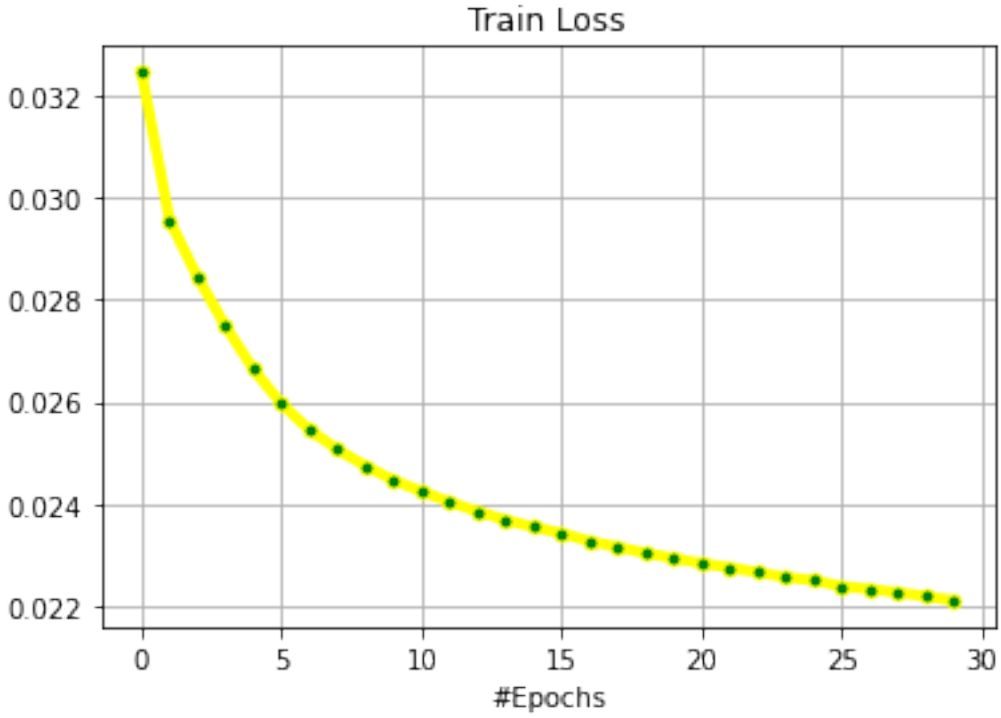
```
Epoch 4 Train loss 0.026630586347579956
Train Accuracy:  40.06 %
Epoch 4 Test loss 0.026346282386779784
Test Accuracy:  40.8 %
Epoch 6 Train loss 0.02545036435365677
Train Accuracy:  42.12 %
Epoch 6 Test loss 0.025257973790168763
Test Accuracy:  42.62 %
Epoch 8 Train loss 0.024742556970119477
Train Accuracy:  43.61 %
Epoch 8 Test loss 0.024617441391944887
Test Accuracy:  43.89 %
Epoch 10 Train loss 0.024244724669456482
Train Accuracy:  44.7 %
Epoch 10 Test loss 0.024270060181617738
Test Accuracy:  45.01 %
Epoch 12 Train loss 0.023850639278888704
Train Accuracy:  45.64 %
Epoch 12 Test loss 0.023834529948234557
Test Accuracy:  45.98 %
Epoch 14 Train loss 0.023547777893543242
Train Accuracy:  46.31 %
Epoch 14 Test loss 0.023567615354061128
Test Accuracy:  46.52 %
Epoch 16 Train loss 0.023268902716636657
Train Accuracy:  46.92 %
Epoch 16 Test loss 0.023344457960128785
Test Accuracy:  46.99 %
Epoch 18 Train loss 0.023036395609378816
Train Accuracy:  47.5 %
Epoch 18 Test loss 0.023090436065196993
Test Accuracy:  47.46 %
Epoch 20 Train loss 0.02283580857515335
Train Accuracy:  47.92 %
Epoch 20 Test loss 0.02289699239730835
Test Accuracy:  48.37 %
Epoch 22 Train loss 0.022661435062885284
Train Accuracy:  48.42 %
Epoch 22 Test loss 0.022853617095947267
Test Accuracy:  48.59 %
Epoch 24 Train loss 0.022504608232975006
Train Accuracy:  49.01 %
Epoch 24 Test loss 0.02263120641708374
Test Accuracy:  48.39 %
Epoch 26 Train loss 0.022335357122421264
Train Accuracy:  49.17 %
Epoch 26 Test loss 0.022425109112262726
Test Accuracy:  49.55 %
```

```
Epoch 28 Train loss 0.022191653356552123
Train Accuracy:  49.6 %
Epoch 28 Test loss 0.02246375209093094
Test Accuracy:  48.97 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.032
0.030
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs


Train Loss
0.032
0.030
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs

1.11 Varying learning rate

- We test for different learning rates, 0.1, 0.001, 0.001, 0.0001
- For a learning rate of 0.1, no learning is done since the training is too fast to capture any information.
- For a learning rate of 0.01, the model improves from 0.1, but is not as good.
- For a learning rate of 0.001, the model starts to overfit slightly above the 20 epochs, indicating that it is learning too many details from the training set.
- For learning rate of 0.0001, the learning is slower than the other tests. But there is no overfitting for 30 epochs

1.11.1 Learning rate = 0.1

[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 1, num_epochs = 30, lr = 0.1)
```

```
Epoch 0 Train loss 0.11740410886287689
Train Accuracy:  9.84 %
Epoch 0 Test loss 0.036261409997940065
Test Accuracy:  10.0 %
Epoch 2 Train loss 0.03614556882858276
Train Accuracy:  10.04 %
Epoch 2 Test loss 0.03628281333446503
Test Accuracy:  10.0 %
Epoch 4 Train loss 0.036152034974098204
Train Accuracy:  9.99 %
Epoch 4 Test loss 0.03633590083122253
Test Accuracy:  10.0 %
Epoch 6 Train loss 0.036148593683242795
Train Accuracy:  10.13 %
Epoch 6 Test loss 0.03636818180084229
Test Accuracy:  10.0 %
Epoch 8 Train loss 0.03615383156299591
Train Accuracy:  10.01 %
Epoch 8 Test loss 0.036272672605514526
Test Accuracy:  10.0 %
Epoch 10 Train loss 0.036164373202323914
Train Accuracy:  9.94 %
Epoch 10 Test loss 0.03619877026081085
Test Accuracy:  10.0 %
Epoch 12 Train loss 0.03614847599506378
Train Accuracy:  10.07 %
Epoch 12 Test loss 0.0362496337890625
Test Accuracy:  10.0 %
Epoch 14 Train loss 0.036165206780433654
Train Accuracy:  10.15 %
```

```
Epoch 14 Test loss 0.03622400331497192
Test Accuracy:  10.0 %
Epoch 16 Train loss 0.03616562909603119
Train Accuracy:  10.08 %
Epoch 16 Test loss 0.03624142308235168
Test Accuracy:  10.0 %
Epoch 18 Train loss 0.036144770860672
Train Accuracy:  9.75 %
Epoch 18 Test loss 0.03628265163898468
Test Accuracy:  10.0 %
Epoch 20 Train loss 0.03616015038490295
Train Accuracy:  10.1 %
Epoch 20 Test loss 0.03632381880283356
Test Accuracy:  10.0 %
Epoch 22 Train loss 0.03614922878265381
Train Accuracy:  10.07 %
Epoch 22 Test loss 0.03625841827392578
Test Accuracy:  10.0 %
Epoch 24 Train loss 0.03614446442604065
Train Accuracy:  9.99 %
Epoch 24 Test loss 0.03619375011920929
Test Accuracy:  10.0 %
Epoch 26 Train loss 0.0361514474773407
Train Accuracy:  9.96 %
Epoch 26 Test loss 0.03634953601360321
Test Accuracy:  10.0 %
Epoch 28 Train loss 0.036148710951805114
Train Accuracy:  10.01 %
Epoch 28 Test loss 0.036201337242126465
Test Accuracy:  10.0 %
```

1.11.2 Learning rate = 0.01

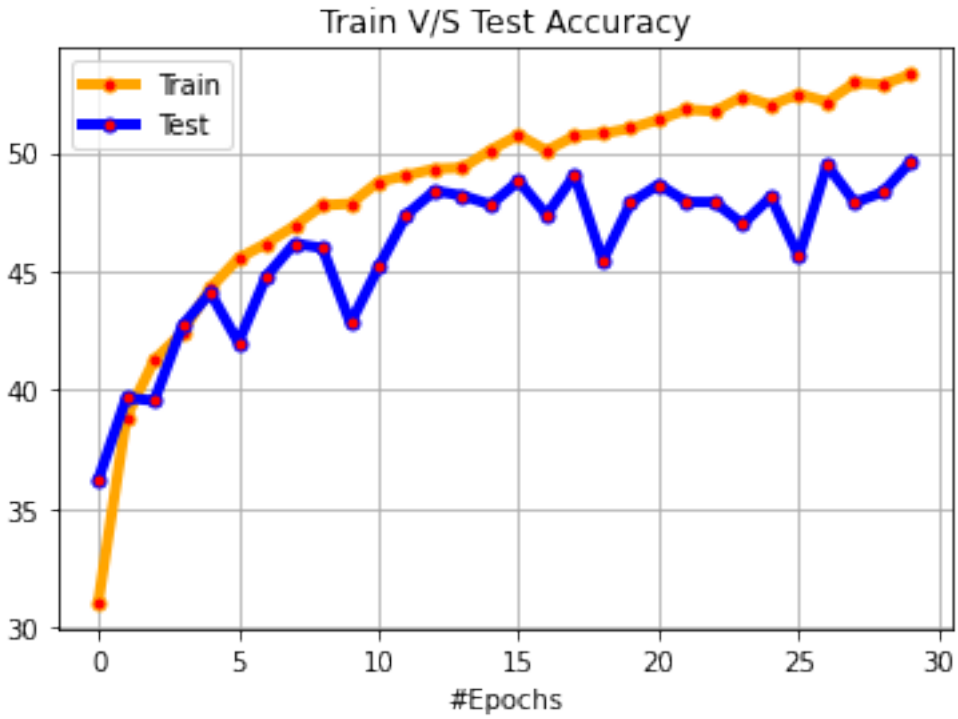
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 1, num_epochs = 30, lr = 0.01)
```

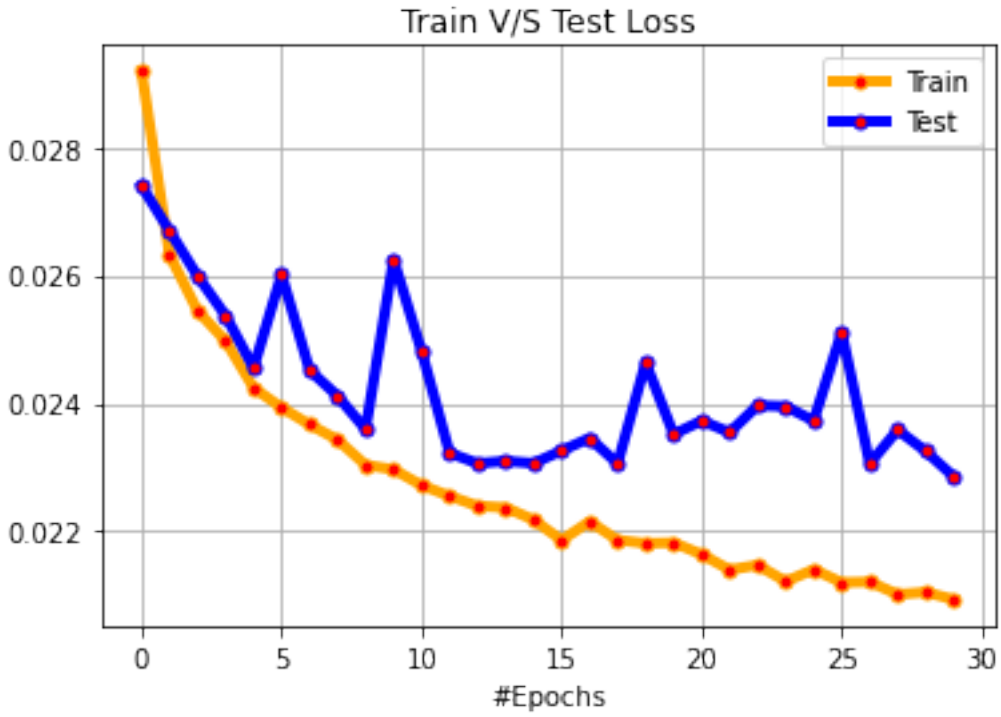
```
Epoch 0 Train loss 0.029189408905506135
Train Accuracy:  31.04 %
Epoch 0 Test loss 0.027398265397548676
Test Accuracy:  36.21 %
Epoch 2 Train loss 0.025446906151771545
Train Accuracy:  41.26 %
Epoch 2 Test loss 0.025989460039138795
Test Accuracy:  39.54 %
Epoch 4 Train loss 0.02424355677127838
Train Accuracy:  44.28 %
```

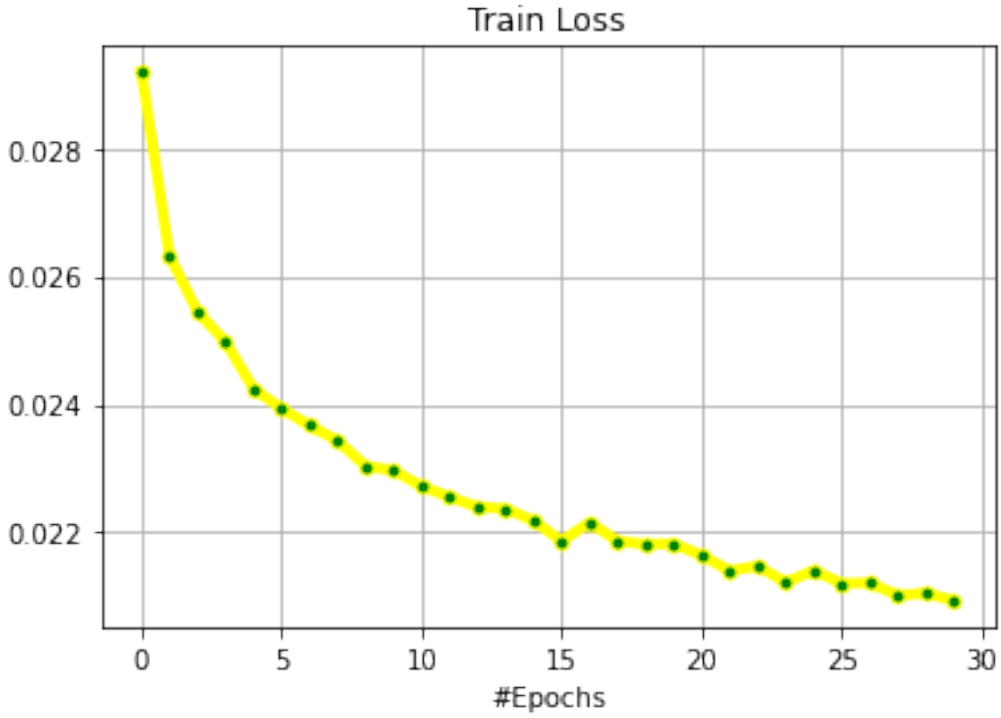
```
Epoch 4 Test loss 0.024555451393127443
Test Accuracy:  44.09 %
Epoch 6 Train loss 0.023669011359214783
Train Accuracy:  46.16 %
Epoch 6 Test loss 0.024519493961334227
Test Accuracy:  44.79 %
Epoch 8 Train loss 0.023037680280208587
Train Accuracy:  47.77 %
Epoch 8 Test loss 0.023595652079582214
Test Accuracy:  45.97 %
Epoch 10 Train loss 0.022720283029079438
Train Accuracy:  48.76 %
Epoch 10 Test loss 0.02481651603579521
Test Accuracy:  45.16 %
Epoch 12 Train loss 0.02239755865097046
Train Accuracy:  49.31 %
Epoch 12 Test loss 0.023063282513618468
Test Accuracy:  48.38 %
Epoch 14 Train loss 0.022180982081890108
Train Accuracy:  50.1 %
Epoch 14 Test loss 0.023065522587299346
Test Accuracy:  47.8 %
Epoch 16 Train loss 0.022156054856777192
Train Accuracy:  50.08 %
Epoch 16 Test loss 0.02345835942029953
Test Accuracy:  47.37 %
Epoch 18 Train loss 0.02181518461704254
Train Accuracy:  50.79 %
Epoch 18 Test loss 0.02463557460308075
Test Accuracy:  45.43 %
Epoch 20 Train loss 0.02164169592142105
Train Accuracy:  51.38 %
Epoch 20 Test loss 0.02372876281738281
Test Accuracy:  48.65 %
Epoch 22 Train loss 0.02147416175842285
Train Accuracy:  51.73 %
Epoch 22 Test loss 0.023967721176147462
Test Accuracy:  47.9 %
Epoch 24 Train loss 0.021405132743120194
Train Accuracy:  51.99 %
Epoch 24 Test loss 0.023728196024894715
Test Accuracy:  48.13 %
Epoch 26 Train loss 0.021211635674238206
Train Accuracy:  52.13 %
Epoch 26 Test loss 0.02306110110282898
Test Accuracy:  49.47 %
Epoch 28 Train loss 0.021050327529907226
Train Accuracy:  52.86 %
```

```
Epoch 28 Test loss 0.023253227043151855
Test Accuracy:  48.33 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs


Train Loss
0.028
0.026
0.024
0.022
0
5
10
15
20
25
30
#Epochs

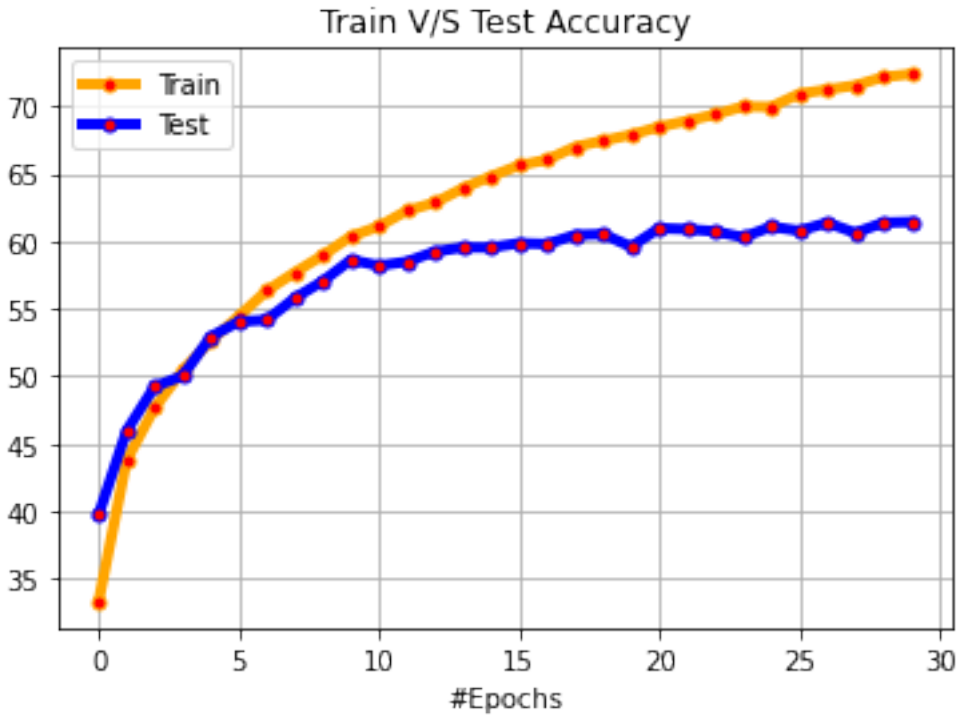
1.11.3 Learning rate = 0.001

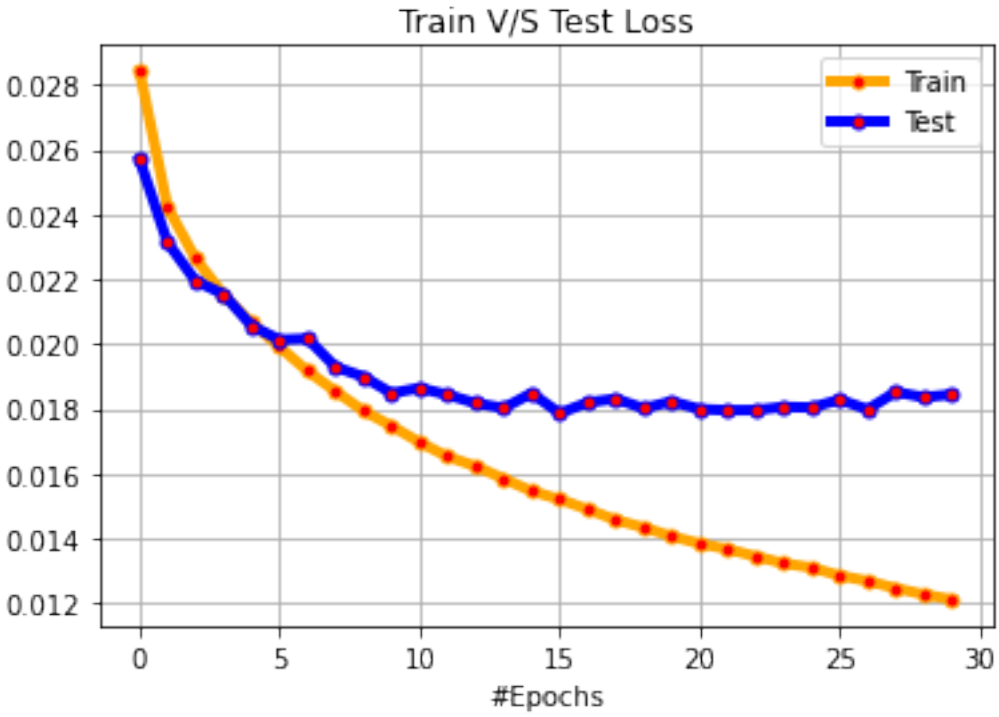
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 1, num_epochs = 30, lr = 0.001)
```

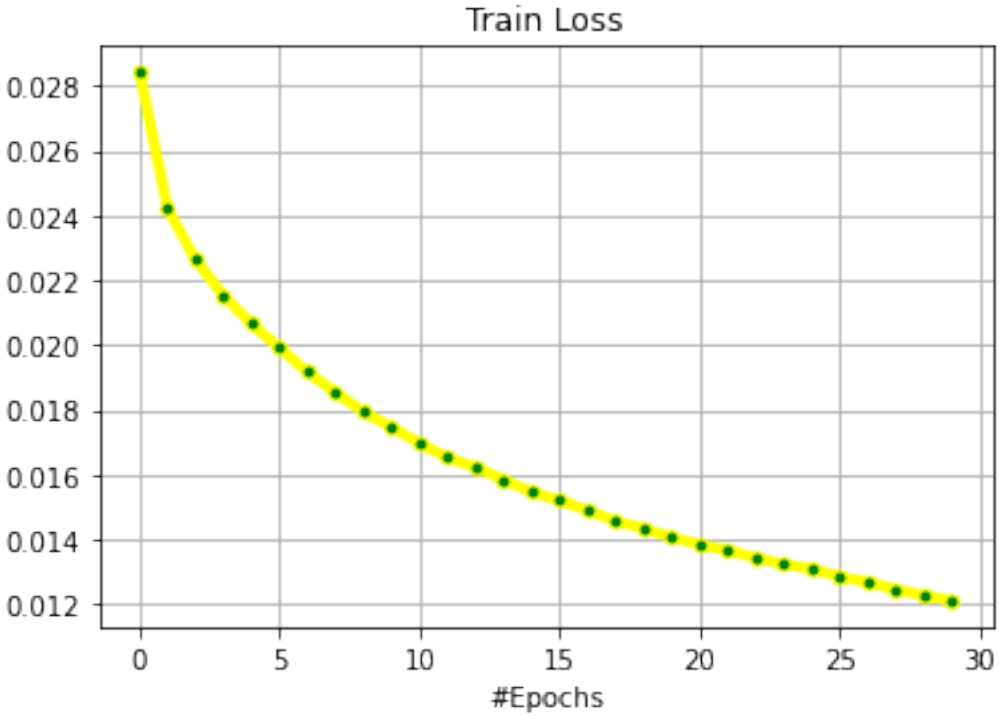
```
Epoch 0 Train loss 0.0284243941283226
Train Accuracy:  33.29 %
Epoch 0 Test loss 0.02569301972389221
Test Accuracy:  39.73 %
Epoch 2 Train loss 0.02263741551041603
Train Accuracy:  47.65 %
Epoch 2 Test loss 0.021951331090927124
Test Accuracy:  49.23 %
Epoch 4 Train loss 0.02065375601053238
Train Accuracy:  52.62 %
Epoch 4 Test loss 0.020563525515794753
Test Accuracy:  52.92 %
Epoch 6 Train loss 0.01916860643863678
Train Accuracy:  56.32 %
Epoch 6 Test loss 0.020171793049573897
Test Accuracy:  54.16 %
Epoch 8 Train loss 0.01793928391456604
Train Accuracy:  58.94 %
Epoch 8 Test loss 0.018995503002405167
Test Accuracy:  57.03 %
Epoch 10 Train loss 0.016971045863628388
Train Accuracy:  61.09 %
Epoch 10 Test loss 0.018650246173143387
Test Accuracy:  58.18 %
Epoch 12 Train loss 0.016234444193840028
Train Accuracy:  62.84 %
Epoch 12 Test loss 0.01818048803806305
Test Accuracy:  59.22 %
Epoch 14 Train loss 0.01546493481516838
Train Accuracy:  64.79 %
Epoch 14 Test loss 0.018487155705690383
Test Accuracy:  59.52 %
Epoch 16 Train loss 0.014903024500608444
Train Accuracy:  66.02 %
Epoch 16 Test loss 0.018200888890028
Test Accuracy:  59.78 %
Epoch 18 Train loss 0.01434364363193512
Train Accuracy:  67.44 %
Epoch 18 Test loss 0.01801983703970909
Test Accuracy:  60.51 %
Epoch 20 Train loss 0.013847226552963257
```

```
Train Accuracy:  68.46 %
Epoch 20 Test loss 0.017996345204114913
Test Accuracy:  60.95 %
Epoch 22 Train loss 0.013450068492889404
Train Accuracy:  69.38 %
Epoch 22 Test loss 0.017967092764377593
Test Accuracy:  60.72 %
Epoch 24 Train loss 0.013105946396589279
Train Accuracy:  69.94 %
Epoch 24 Test loss 0.01804545001387596
Test Accuracy:  61.07 %
Epoch 26 Train loss 0.012693150154352188
Train Accuracy:  71.22 %
Epoch 26 Test loss 0.017987996673583983
Test Accuracy:  61.36 %
Epoch 28 Train loss 0.012277896779179573
Train Accuracy:  72.14 %
Epoch 28 Test loss 0.018360096603631974
Test Accuracy:  61.36 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0.012
0
5
10
15
20
25
30
#Epochs


Train Loss
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0.012
0
5
10
15
20
25
30
#Epochs

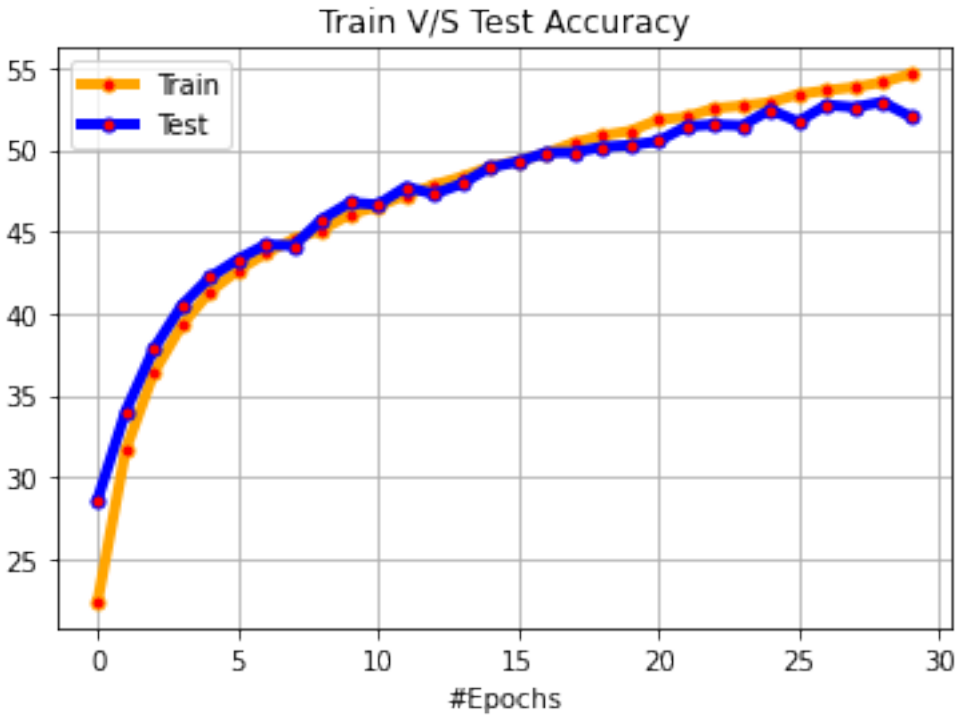
1.11.4 Learning rate = 0.0001

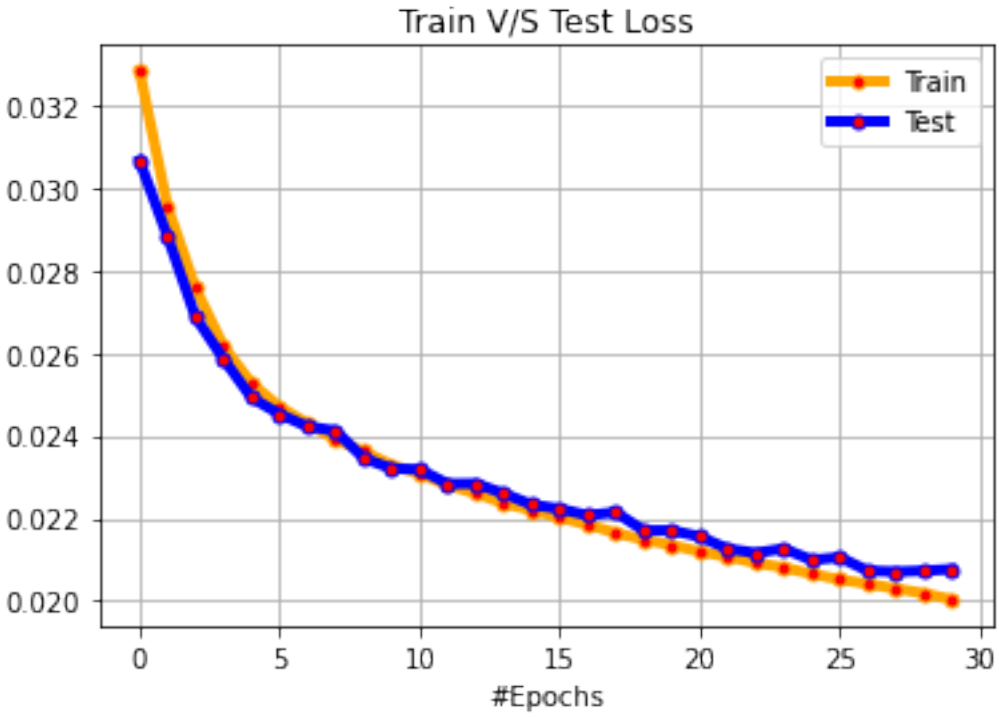
[0]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, method = 1, num_epochs = 30, lr = 0.0001)
```

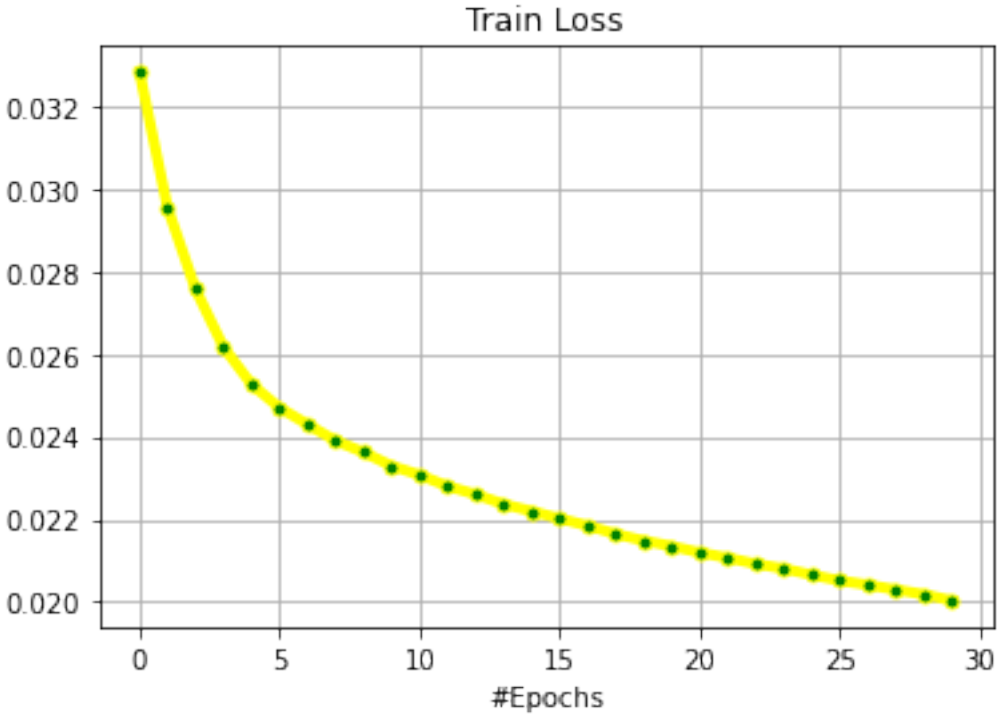
```
Epoch 0 Train loss 0.032827669267654416
Train Accuracy:  22.38 %
Epoch 0 Test loss 0.030659768772125243
Test Accuracy:  28.63 %
Epoch 2 Train loss 0.02757111728191376
Train Accuracy:  36.35 %
Epoch 2 Test loss 0.0268716757774353
Test Accuracy:  37.81 %
Epoch 4 Train loss 0.02526780061006546
Train Accuracy:  41.3 %
Epoch 4 Test loss 0.024918532204627992
Test Accuracy:  42.19 %
Epoch 6 Train loss 0.02429858787059784
Train Accuracy:  43.66 %
Epoch 6 Test loss 0.024223096048831938
Test Accuracy:  44.19 %
Epoch 8 Train loss 0.023624930171966554
Train Accuracy:  45.1 %
Epoch 8 Test loss 0.0234566109418869
Test Accuracy:  45.74 %
Epoch 10 Train loss 0.023075134482383727
Train Accuracy:  46.53 %
Epoch 10 Test loss 0.023185414004325867
Test Accuracy:  46.63 %
Epoch 12 Train loss 0.022604630215168
Train Accuracy:  47.83 %
Epoch 12 Test loss 0.022843438243865966
Test Accuracy:  47.28 %
Epoch 14 Train loss 0.022197305936813353
Train Accuracy:  48.95 %
Epoch 14 Test loss 0.022332274508476257
Test Accuracy:  48.91 %
Epoch 16 Train loss 0.02184705468893051
Train Accuracy:  49.77 %
Epoch 16 Test loss 0.02207356948852539
Test Accuracy:  49.81 %
Epoch 18 Train loss 0.021483549307584764
Train Accuracy:  50.86 %
Epoch 18 Test loss 0.02170699874162674
Test Accuracy:  50.16 %
Epoch 20 Train loss 0.02119532025337219
```

```
Train Accuracy:  51.81 %
Epoch 20 Test loss 0.021558650064468384
Test Accuracy:  50.5 %
Epoch 22 Train loss 0.020922310481071474
Train Accuracy:  52.52 %
Epoch 22 Test loss 0.021162275183200838
Test Accuracy:  51.52 %
Epoch 24 Train loss 0.020655539088249207
Train Accuracy:  52.88 %
Epoch 24 Test loss 0.020991155409812928
Test Accuracy:  52.43 %
Epoch 26 Train loss 0.020413880299329758
Train Accuracy:  53.61 %
Epoch 26 Test loss 0.020732024466991423
Test Accuracy:  52.72 %
Epoch 28 Train loss 0.020184305658340455
Train Accuracy:  54.11 %
Epoch 28 Test loss 0.020731239438056947
Test Accuracy:  52.9 %
```

[0]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```




Train V/S Test Loss
Train
Test
0.032
0.030
0.028
0.026
0.024
0.022
0.020
0
5
10
15
20
25
30
#Epochs


Train Loss
0.032
0.030
0.028
0.026
0.024
0.022
0.020
0
5
10
15
20
25
30
#Epochs

1.12 Varying losses

- Cross Entropy
- L1
- L2
- We see that the L1 losses does not work as well as Cross Entropy and L2 losses, which work better for classification tasks.

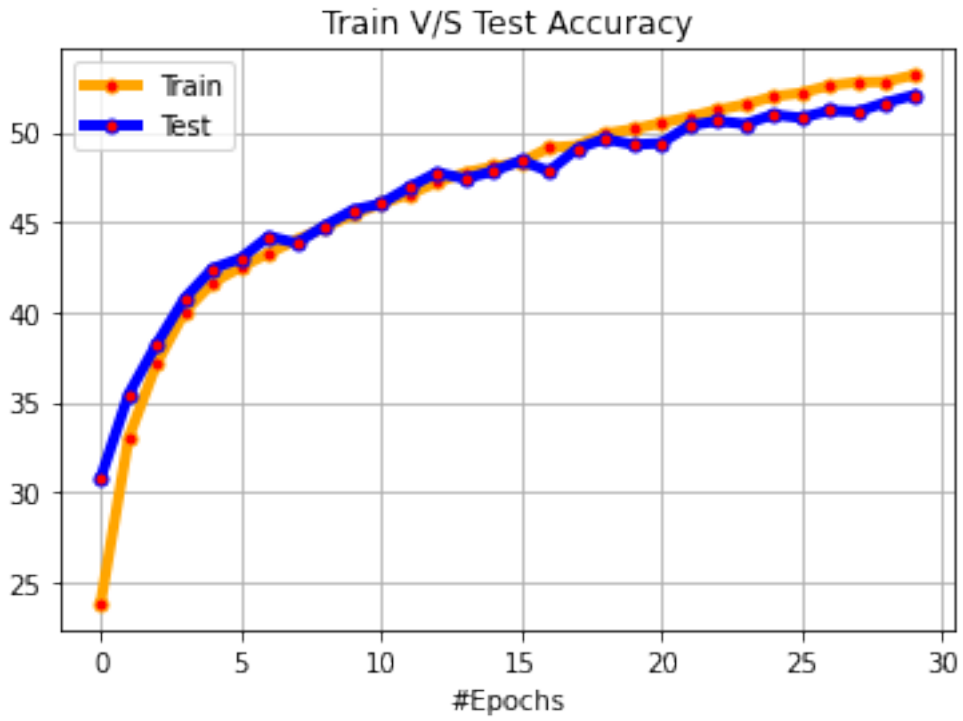
1.12.1 Cross Entropy

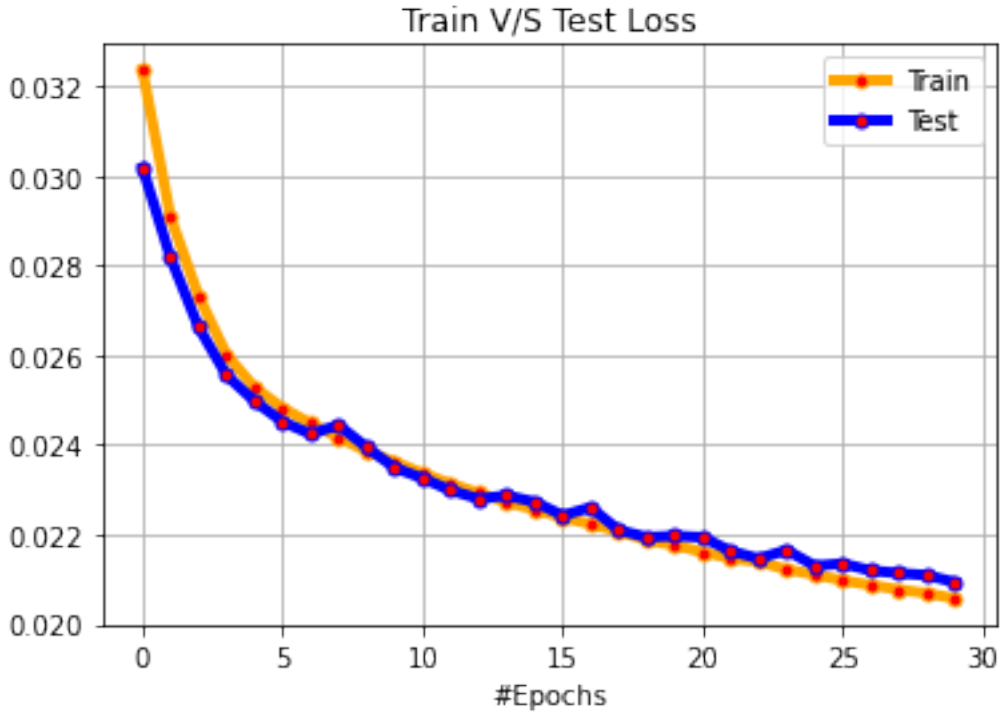
[35]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, num_epochs = 30)
```

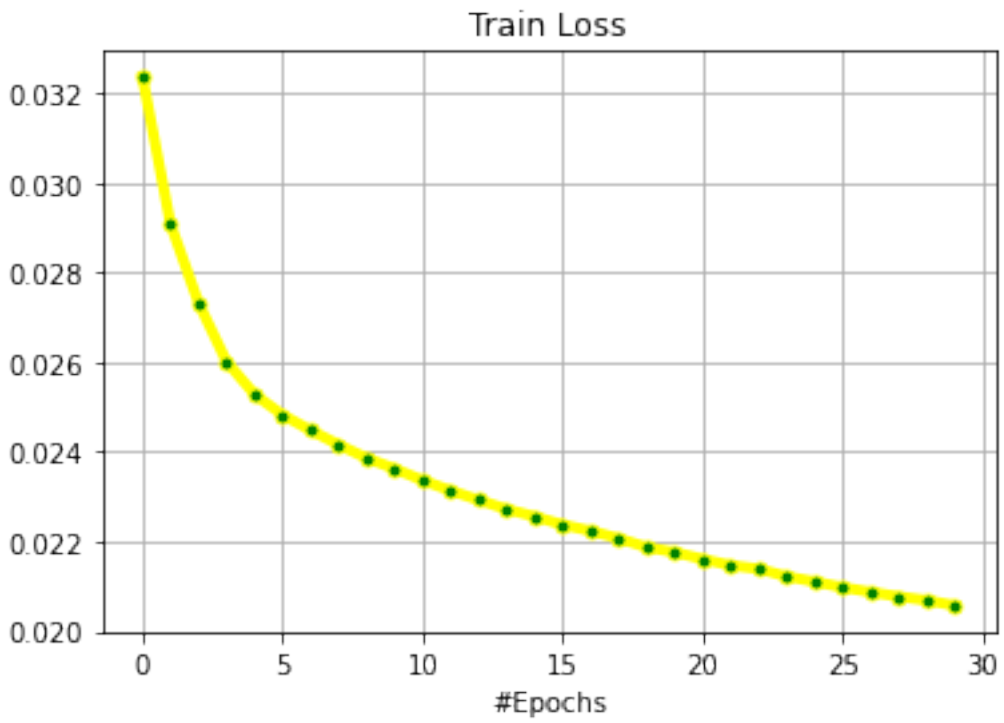
```
Epoch 0 Train loss 0.03235381579399109
Train Accuracy:  23.8 %
Epoch 0 Test loss 0.03017472482919693
Test Accuracy:  30.76 %
Epoch 2 Train loss 0.027322877237796785
Train Accuracy:  37.14 %
Epoch 2 Test loss 0.026626333487033842
Test Accuracy:  38.23 %
Epoch 4 Train loss 0.025284030783176422
Train Accuracy:  41.64 %
Epoch 4 Test loss 0.024975963521003723
Test Accuracy:  42.43 %
Epoch 6 Train loss 0.02448316572189331
Train Accuracy:  43.28 %
Epoch 6 Test loss 0.024246564590930938
Test Accuracy:  44.21 %
Epoch 8 Train loss 0.023863536615371703
Train Accuracy:  44.71 %
Epoch 8 Test loss 0.023949354028701783
Test Accuracy:  44.81 %
Epoch 10 Train loss 0.023368281440734863
Train Accuracy:  46.02 %
Epoch 10 Test loss 0.023258136773109436
Test Accuracy:  46.02 %
Epoch 12 Train loss 0.02292942199230194
Train Accuracy:  47.21 %
Epoch 12 Test loss 0.022802755713462828
Test Accuracy:  47.76 %
Epoch 14 Train loss 0.022562573735713957
Train Accuracy:  48.09 %
Epoch 14 Test loss 0.02273227859735489
Test Accuracy:  47.83 %
Epoch 16 Train loss 0.022242012321949004
```

```
Train Accuracy:  49.14 %
Epoch 16 Test loss 0.02260816181898117
Test Accuracy:  47.83 %
Epoch 18 Train loss 0.021878147711753844
Train Accuracy:  49.93 %
Epoch 18 Test loss 0.02194477069377899
Test Accuracy:  49.64 %
Epoch 20 Train loss 0.021612877440452576
Train Accuracy:  50.51 %
Epoch 20 Test loss 0.021944054985046386
Test Accuracy:  49.39 %
Epoch 22 Train loss 0.021395591418743132
Train Accuracy:  51.25 %
Epoch 22 Test loss 0.021468857419490815
Test Accuracy:  50.63 %
Epoch 24 Train loss 0.021110317282676698
Train Accuracy:  51.99 %
Epoch 24 Test loss 0.021306962156295777
Test Accuracy:  50.98 %
Epoch 26 Train loss 0.02087784893155098
Train Accuracy:  52.6 %
Epoch 26 Test loss 0.021201864421367646
Test Accuracy:  51.21 %
Epoch 28 Train loss 0.020694269416332244
Train Accuracy:  52.81 %
Epoch 28 Test loss 0.02109709233045578
Test Accuracy:  51.65 %
```

[36]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```


Train V/S Test Accuracy
Train
Test
50
45
40
35
30
25
0
5
10
15
20
25
30
#Epochs


Train V/S Test Loss
Train
Test
0.032
0.030
0.028
0.026
0.024
0.022
0.020
0
5
10
15
20
25
30
#Epochs



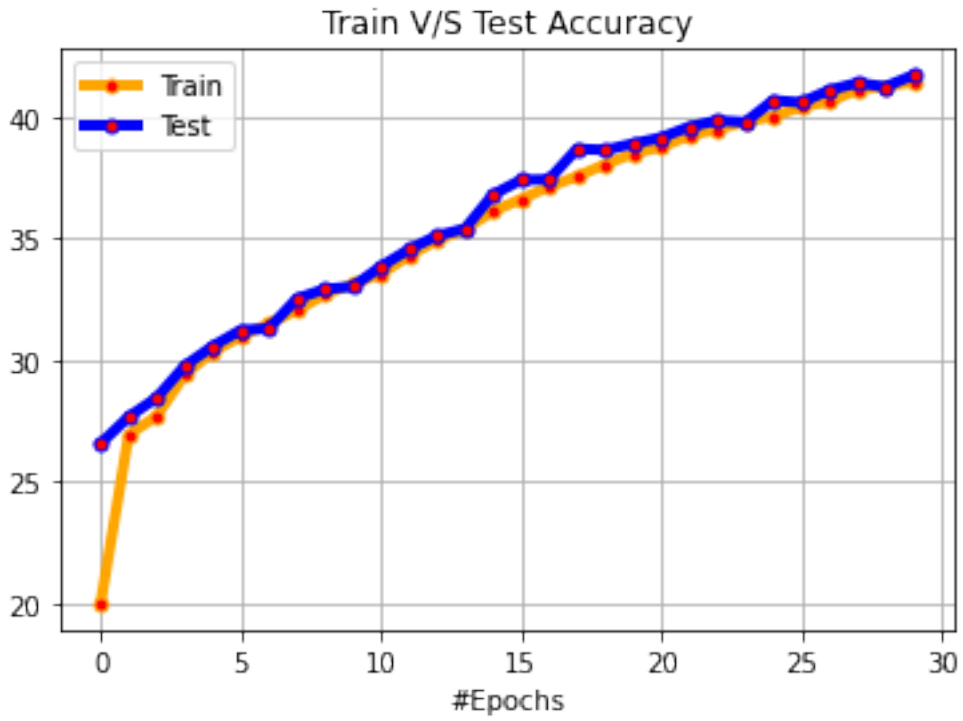
1.12.2 L1 Loss

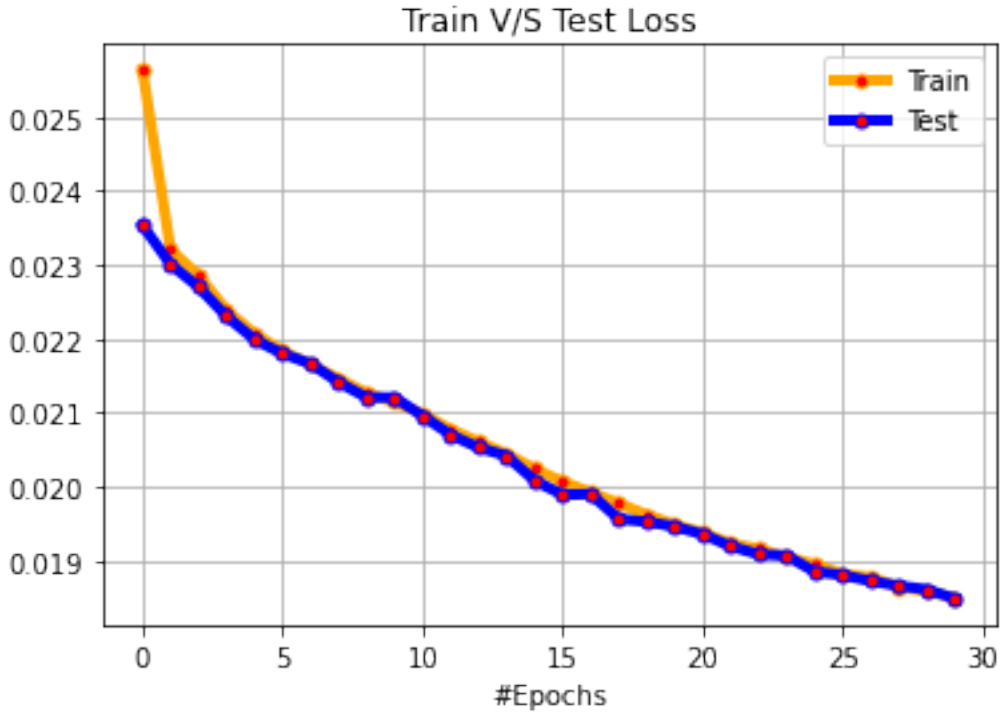
[31]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, num_epochs = 30, loss_type = 'l1')
```

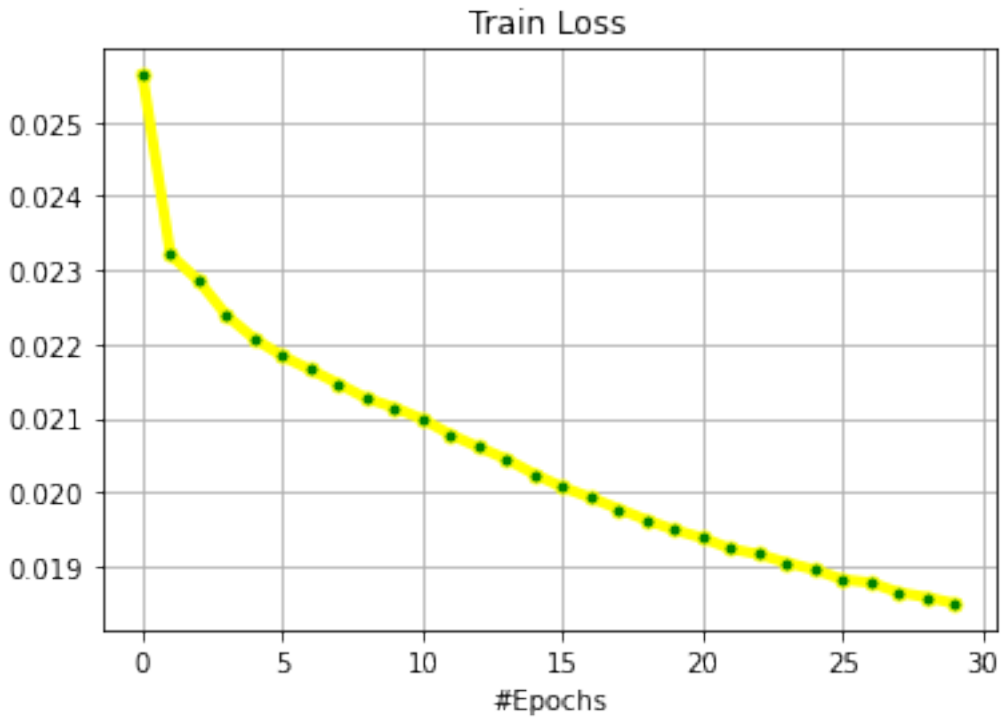
```
Epoch 0 Train loss 0.02562958305835724
Train Accuracy:  20.02 %
Epoch 0 Test loss 0.02353138518333435
Test Accuracy:  26.58 %
Epoch 2 Train loss 0.022855551342964173
Train Accuracy:  27.7 %
Epoch 2 Test loss 0.0226971852183342
Test Accuracy:  28.46 %
Epoch 4 Train loss 0.02206899701356888
Train Accuracy:  30.35 %
Epoch 4 Test loss 0.021987938463687896
Test Accuracy:  30.58 %
Epoch 6 Train loss 0.021650766837596893
Train Accuracy:  31.49 %
Epoch 6 Test loss 0.0216632110953331
Test Accuracy:  31.3 %
Epoch 8 Train loss 0.02126389032125473
```

```
Train Accuracy:  32.7 %
Epoch 8 Test loss 0.021204432106018068
Test Accuracy:  32.92 %
Epoch 10 Train loss 0.020983206026554107
Train Accuracy:  33.54 %
Epoch 10 Test loss 0.020956089627742768
Test Accuracy:  33.85 %
Epoch 12 Train loss 0.020606279680728912
Train Accuracy:  34.91 %
Epoch 12 Test loss 0.020526439011096954
Test Accuracy:  35.11 %
Epoch 14 Train loss 0.020241376444101334
Train Accuracy:  36.11 %
Epoch 14 Test loss 0.020079648721218108
Test Accuracy:  36.8 %
Epoch 16 Train loss 0.019924928871393202
Train Accuracy:  37.12 %
Epoch 16 Test loss 0.019899650716781615
Test Accuracy:  37.42 %
Epoch 18 Train loss 0.019622547734975813
Train Accuracy:  38.04 %
Epoch 18 Test loss 0.019524008399248123
Test Accuracy:  38.63 %
Epoch 20 Train loss 0.01938046478152275
Train Accuracy:  38.75 %
Epoch 20 Test loss 0.01935564943552017
Test Accuracy:  39.14 %
Epoch 22 Train loss 0.019160412759780882
Train Accuracy:  39.48 %
Epoch 22 Test loss 0.019085144102573396
Test Accuracy:  39.84 %
Epoch 24 Train loss 0.018951653730869295
Train Accuracy:  40.0 %
Epoch 24 Test loss 0.018849332690238953
Test Accuracy:  40.67 %
Epoch 26 Train loss 0.01877772992134094
Train Accuracy:  40.61 %
Epoch 26 Test loss 0.018722592914104462
Test Accuracy:  41.09 %
Epoch 28 Train loss 0.018580496969223023
Train Accuracy:  41.23 %
Epoch 28 Test loss 0.018609993720054625
Test Accuracy:  41.23 %
```

[32]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```


Train V/S Test Accuracy
Train
Test
40
35
30
25
20
0
5
10
15
20
25
30
#Epochs


Train V/S Test Loss
Train
Test
0.025
0.024
0.023
0.022
0.021
0.020
0.019
0
5
10
15
20
25
30
#Epochs



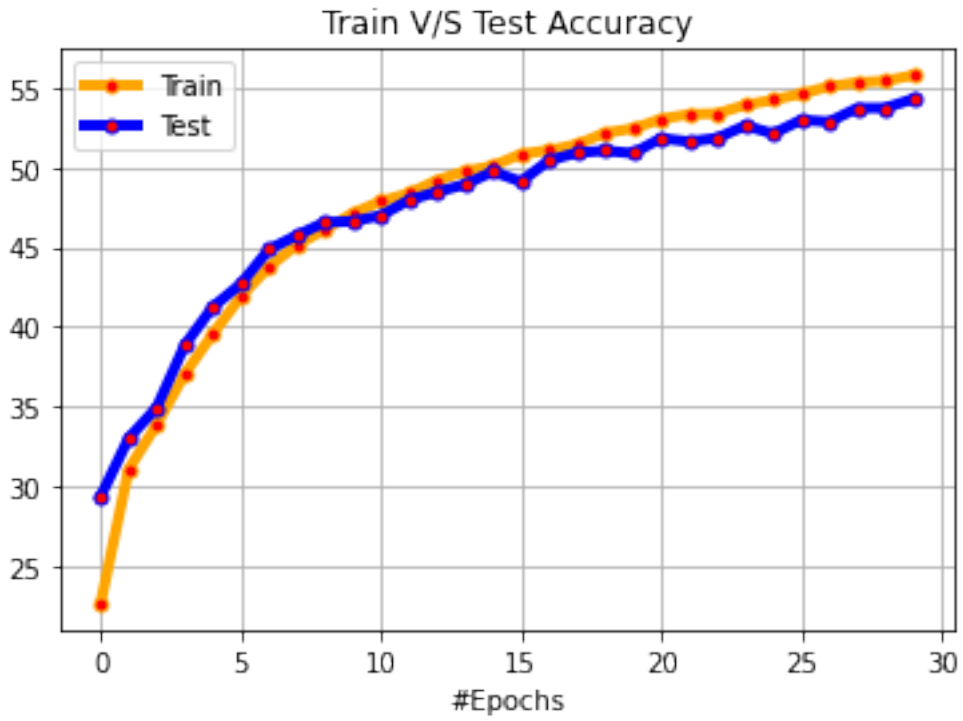
1.12.3 L2 Loss

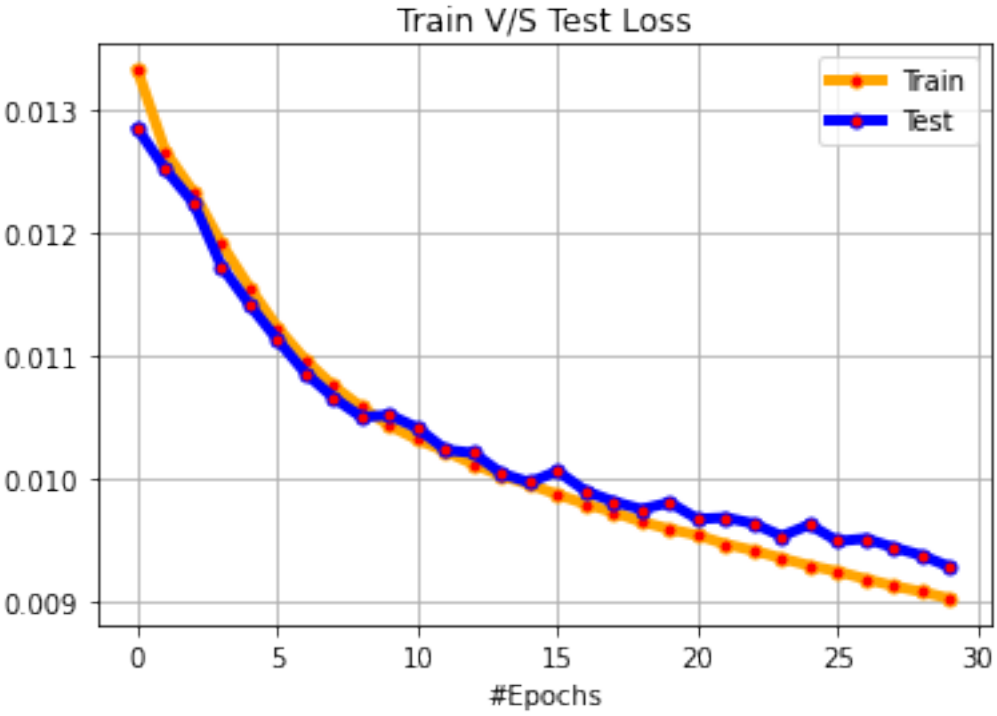
[33]:
```
cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss =␣
↪train_test(trainloader = cifar10_trainloader, testloader =␣
↪cifar10_testloader, num_epochs = 30, loss_type = 'l2')
```

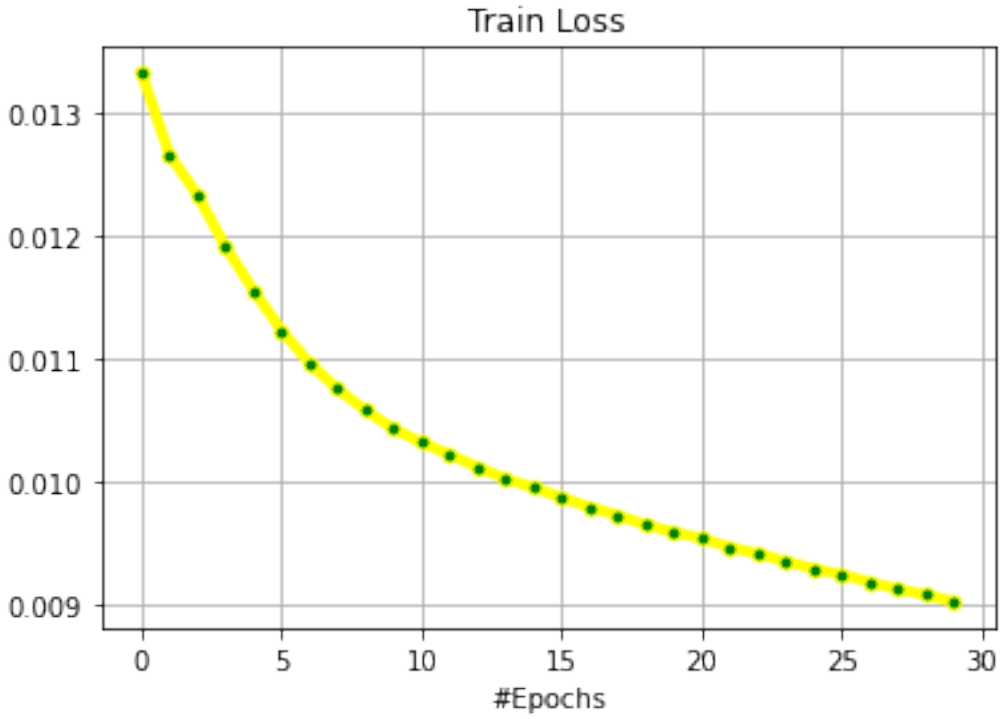
```
Epoch 0 Train loss 0.01331799992442131
Train Accuracy:  22.65 %
Epoch 0 Test loss 0.012842462611198426
Test Accuracy:  29.36 %
Epoch 2 Train loss 0.012329465761184693
Train Accuracy:  33.86 %
Epoch 2 Test loss 0.01224024885892868
Test Accuracy:  34.94 %
Epoch 4 Train loss 0.011555669370889664
Train Accuracy:  39.55 %
Epoch 4 Test loss 0.011414718073606491
Test Accuracy:  41.25 %
Epoch 6 Train loss 0.010965514270067214
Train Accuracy:  43.76 %
Epoch 6 Test loss 0.010858898329734802
Test Accuracy:  44.85 %
Epoch 8 Train loss 0.010582988983392716
```

```
Train Accuracy:  46.1 %
Epoch 8 Test loss 0.010502567887306214
Test Accuracy:  46.56 %
Epoch 10 Train loss 0.010320935755968094
Train Accuracy:  47.88 %
Epoch 10 Test loss 0.010414966160058976
Test Accuracy:  46.98 %
Epoch 12 Train loss 0.010121560986042022
Train Accuracy:  49.19 %
Epoch 12 Test loss 0.010211314725875855
Test Accuracy:  48.46 %
Epoch 14 Train loss 0.009955947295427323
Train Accuracy:  50.1 %
Epoch 14 Test loss 0.009973411387205124
Test Accuracy:  49.74 %
Epoch 16 Train loss 0.009797630146741867
Train Accuracy:  51.08 %
Epoch 16 Test loss 0.009902750474214553
Test Accuracy:  50.44 %
Epoch 18 Train loss 0.009652903554439545
Train Accuracy:  52.17 %
Epoch 18 Test loss 0.009755272018909454
Test Accuracy:  51.05 %
Epoch 20 Train loss 0.009544406747221948
Train Accuracy:  53.04 %
Epoch 20 Test loss 0.009675943917036056
Test Accuracy:  51.83 %
Epoch 22 Train loss 0.009420580766797066
Train Accuracy:  53.38 %
Epoch 22 Test loss 0.00963899363875389
Test Accuracy:  51.84 %
Epoch 24 Train loss 0.009297923620343209
Train Accuracy:  54.24 %
Epoch 24 Test loss 0.009630371636152267
Test Accuracy:  52.12 %
Epoch 26 Train loss 0.009188574499487876
Train Accuracy:  55.08 %
Epoch 26 Test loss 0.009516917929053306
Test Accuracy:  52.86 %
Epoch 28 Train loss 0.00908512634396553
Train Accuracy:  55.42 %
Epoch 28 Test loss 0.009381013187766076
Test Accuracy:  53.73 %
```

[34]:
```
plot(cifar10_trainAcc, cifar10_trainLoss, cifar10_testAcc, cifar10_testLoss)
```


Train V/S Test Accuracy
Train
Test
55
50
45
40
35
30
25
0
5
10
15
20
25
30
#Epochs


Train V/S Test Loss
Train
Test
0.013
0.012
0.011
0.010
0.009
0
5
10
15
20
25
30
#Epochs


Train Loss
0.013
0.012
0.011
0.010
0.009
0
5
10
15
20
25
30
#Epochs