In [180…
```
'''
Patrick Ballou
ID: 801130521
ECGR 4105
Homework 6
Problem 1
'''
```

Out[180]:
```
'\nPatrick Ballou\nID: 801130521\nECGR 4105\nHomework 6\nProblem 1\n'
```

In [181…
```
import torch
from torch import cuda
import torch.nn as nn
from torchvision import transforms
from torchvision import datasets
from sklearn import metrics
import matplotlib.pyplot as plt
```

In [182…
```
data_path = 'C:/Users/patri/Documents/School_Fall_2022/ECGR_4105/Homework/homework_6'
#normalized
cifar10 = datasets.CIFAR10(data_path, train=True, download=True, transform=transforms.
cifar10_val = datasets.CIFAR10(data_path, train=False, download=True, transform=transf
```

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

In [183…
```
#one hidden layer
model_a = nn.Sequential(
            nn.Linear(3072,512),
            nn.Tanh(),
            nn.Linear(512, 10),
            nn.Tanh(),
            nn.LogSoftmax(dim=1))
```

In [184…
```
device = 'cuda'
cuda.get_device_name()
```

Out[184]:
```
'Quadro T2000'
```

In [185…
```
#move to gpu
model_a.to(device)
```

Out[185]:
```
Sequential(
  (0): Linear(in_features=3072, out_features=512, bias=True)
  (1): Tanh()
  (2): Linear(in_features=512, out_features=10, bias=True)
  (3): Tanh()
  (4): LogSoftmax(dim=1)
)
```

In [186…
```
l_r = .002
num_epochs = 300
```

In [187…
```
optimizer = torch.optim.SGD(model_a.parameters(), lr=l_r)
train_loader = torch.utils.data.DataLoader(cifar10, batch_size=64, shuffle=True)
loss_fn = nn.NLLLoss()
```

In [188…

```
for epoch in range(num_epochs):
    for imgs, labels in train_loader:
        imgs, labels = imgs.to(device), labels.to(device)
        batch_size = imgs.shape[0]
        outputs = model_a(imgs.view(batch_size, -1))
        loss = loss_fn(outputs, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    print("Epoch: %d, Loss: %f" % (epoch, float(loss)))
```

```
Epoch: 0, Loss: 1.830380
Epoch: 1, Loss: 1.975224
Epoch: 2, Loss: 1.849329
Epoch: 3, Loss: 1.843616
Epoch: 4, Loss: 1.878878
Epoch: 5, Loss: 1.876666
Epoch: 6, Loss: 1.866058
Epoch: 7, Loss: 1.834244
Epoch: 8, Loss: 1.814626
Epoch: 9, Loss: 1.717482
Epoch: 10, Loss: 1.749159
Epoch: 11, Loss: 1.638357
Epoch: 12, Loss: 1.934950
Epoch: 13, Loss: 1.749440
Epoch: 14, Loss: 2.029092
Epoch: 15, Loss: 1.969803
Epoch: 16, Loss: 1.946956
Epoch: 17, Loss: 1.817966
Epoch: 18, Loss: 1.797036
Epoch: 19, Loss: 1.628987
Epoch: 20, Loss: 1.668926
Epoch: 21, Loss: 1.761377
Epoch: 22, Loss: 1.705829
Epoch: 23, Loss: 1.857571
Epoch: 24, Loss: 2.020938
Epoch: 25, Loss: 1.587944
Epoch: 26, Loss: 1.708810
Epoch: 27, Loss: 1.735573
Epoch: 28, Loss: 1.720546
Epoch: 29, Loss: 1.694691
Epoch: 30, Loss: 1.891228
Epoch: 31, Loss: 1.657503
Epoch: 32, Loss: 1.614375
Epoch: 33, Loss: 1.755880
Epoch: 34, Loss: 1.487533
Epoch: 35, Loss: 1.591486
Epoch: 36, Loss: 1.649389
Epoch: 37, Loss: 1.710340
Epoch: 38, Loss: 1.841892
Epoch: 39, Loss: 1.474391
Epoch: 40, Loss: 1.836172
Epoch: 41, Loss: 1.810567
Epoch: 42, Loss: 1.689059
Epoch: 43, Loss: 1.585100
Epoch: 44, Loss: 1.540840
Epoch: 45, Loss: 1.783811
Epoch: 46, Loss: 1.461412
Epoch: 47, Loss: 1.756535
Epoch: 48, Loss: 1.847187
Epoch: 49, Loss: 2.014337
Epoch: 50, Loss: 1.710624
Epoch: 51, Loss: 1.661636
Epoch: 52, Loss: 1.626320
Epoch: 53, Loss: 1.805043
Epoch: 54, Loss: 1.601557
Epoch: 55, Loss: 1.792169
Epoch: 56, Loss: 1.482822
Epoch: 57, Loss: 1.533759
Epoch: 58, Loss: 1.710453
Epoch: 59, Loss: 1.517948
```

```
Epoch: 60, Loss: 1.589384
Epoch: 61, Loss: 1.453722
Epoch: 62, Loss: 1.651487
Epoch: 63, Loss: 1.420582
Epoch: 64, Loss: 1.565559
Epoch: 65, Loss: 1.851331
Epoch: 66, Loss: 1.664756
Epoch: 67, Loss: 1.587819
Epoch: 68, Loss: 1.649750
Epoch: 69, Loss: 1.569828
Epoch: 70, Loss: 1.529196
Epoch: 71, Loss: 1.525078
Epoch: 72, Loss: 1.590867
Epoch: 73, Loss: 1.812189
Epoch: 74, Loss: 1.212054
Epoch: 75, Loss: 1.403548
Epoch: 76, Loss: 1.497714
Epoch: 77, Loss: 1.466328
Epoch: 78, Loss: 1.587299
Epoch: 79, Loss: 1.372588
Epoch: 80, Loss: 1.436152
Epoch: 81, Loss: 1.630036
Epoch: 82, Loss: 1.439608
Epoch: 83, Loss: 1.686939
Epoch: 84, Loss: 1.916663
Epoch: 85, Loss: 1.497437
Epoch: 86, Loss: 1.592362
Epoch: 87, Loss: 1.604804
Epoch: 88, Loss: 1.606800
Epoch: 89, Loss: 1.270441
Epoch: 90, Loss: 1.498697
Epoch: 91, Loss: 1.413336
Epoch: 92, Loss: 1.359442
Epoch: 93, Loss: 1.550365
Epoch: 94, Loss: 1.251211
Epoch: 95, Loss: 1.502292
Epoch: 96, Loss: 1.659042
Epoch: 97, Loss: 1.381871
Epoch: 98, Loss: 1.409628
Epoch: 99, Loss: 1.496023
Epoch: 100, Loss: 1.243275
Epoch: 101, Loss: 1.518440
Epoch: 102, Loss: 1.741562
Epoch: 103, Loss: 1.626041
Epoch: 104, Loss: 1.398134
Epoch: 105, Loss: 1.440250
Epoch: 106, Loss: 1.463718
Epoch: 107, Loss: 1.488112
Epoch: 108, Loss: 1.621097
Epoch: 109, Loss: 1.446499
Epoch: 110, Loss: 1.612020
Epoch: 111, Loss: 1.731316
Epoch: 112, Loss: 1.391259
Epoch: 113, Loss: 1.398203
Epoch: 114, Loss: 1.531208
Epoch: 115, Loss: 1.595702
Epoch: 116, Loss: 1.628711
Epoch: 117, Loss: 1.633103
Epoch: 118, Loss: 1.281495
Epoch: 119, Loss: 1.420043
```

```
Epoch: 120, Loss: 1.642820
Epoch: 121, Loss: 1.517546
Epoch: 122, Loss: 1.229197
Epoch: 123, Loss: 1.455593
Epoch: 124, Loss: 1.590193
Epoch: 125, Loss: 1.287894
Epoch: 126, Loss: 1.420781
Epoch: 127, Loss: 1.471048
Epoch: 128, Loss: 1.406068
Epoch: 129, Loss: 1.402471
Epoch: 130, Loss: 1.556539
Epoch: 131, Loss: 1.317857
Epoch: 132, Loss: 1.384552
Epoch: 133, Loss: 1.362206
Epoch: 134, Loss: 1.371155
Epoch: 135, Loss: 1.329015
Epoch: 136, Loss: 1.405928
Epoch: 137, Loss: 1.237124
Epoch: 138, Loss: 1.340280
Epoch: 139, Loss: 1.633500
Epoch: 140, Loss: 1.295563
Epoch: 141, Loss: 1.263401
Epoch: 142, Loss: 1.357279
Epoch: 143, Loss: 1.426428
Epoch: 144, Loss: 1.474170
Epoch: 145, Loss: 1.012608
Epoch: 146, Loss: 1.402853
Epoch: 147, Loss: 1.436497
Epoch: 148, Loss: 1.280151
Epoch: 149, Loss: 1.226379
Epoch: 150, Loss: 1.307105
Epoch: 151, Loss: 1.437686
Epoch: 152, Loss: 1.393417
Epoch: 153, Loss: 1.433767
Epoch: 154, Loss: 1.510463
Epoch: 155, Loss: 1.458329
Epoch: 156, Loss: 1.198451
Epoch: 157, Loss: 1.270291
Epoch: 158, Loss: 1.203069
Epoch: 159, Loss: 1.629848
Epoch: 160, Loss: 1.417962
Epoch: 161, Loss: 1.405342
Epoch: 162, Loss: 1.324113
Epoch: 163, Loss: 1.363252
Epoch: 164, Loss: 1.414532
Epoch: 165, Loss: 1.360867
Epoch: 166, Loss: 1.644938
Epoch: 167, Loss: 1.574692
Epoch: 168, Loss: 1.254041
Epoch: 169, Loss: 1.228406
Epoch: 170, Loss: 1.371582
Epoch: 171, Loss: 1.358553
Epoch: 172, Loss: 1.253830
Epoch: 173, Loss: 1.107036
Epoch: 174, Loss: 1.260457
Epoch: 175, Loss: 1.209090
Epoch: 176, Loss: 1.355409
Epoch: 177, Loss: 1.137098
Epoch: 178, Loss: 1.187573
Epoch: 179, Loss: 1.458739
```

```
Epoch: 180, Loss: 1.133879
Epoch: 181, Loss: 1.421390
Epoch: 182, Loss: 1.337532
Epoch: 183, Loss: 1.275644
Epoch: 184, Loss: 1.470158
Epoch: 185, Loss: 1.606486
Epoch: 186, Loss: 1.041858
Epoch: 187, Loss: 1.263257
Epoch: 188, Loss: 1.279440
Epoch: 189, Loss: 1.194176
Epoch: 190, Loss: 1.407727
Epoch: 191, Loss: 1.123659
Epoch: 192, Loss: 1.101905
Epoch: 193, Loss: 1.173969
Epoch: 194, Loss: 1.209534
Epoch: 195, Loss: 1.381774
Epoch: 196, Loss: 1.266635
Epoch: 197, Loss: 1.242110
Epoch: 198, Loss: 1.147398
Epoch: 199, Loss: 1.297130
Epoch: 200, Loss: 1.144265
Epoch: 201, Loss: 1.264098
Epoch: 202, Loss: 1.269959
Epoch: 203, Loss: 1.284696
Epoch: 204, Loss: 1.156545
Epoch: 205, Loss: 1.299049
Epoch: 206, Loss: 1.038579
Epoch: 207, Loss: 1.121245
Epoch: 208, Loss: 1.165666
Epoch: 209, Loss: 1.237504
Epoch: 210, Loss: 1.280493
Epoch: 211, Loss: 1.209827
Epoch: 212, Loss: 1.174733
Epoch: 213, Loss: 1.339459
Epoch: 214, Loss: 1.328829
Epoch: 215, Loss: 1.336193
Epoch: 216, Loss: 1.325338
Epoch: 217, Loss: 1.165435
Epoch: 218, Loss: 1.246009
Epoch: 219, Loss: 1.342925
Epoch: 220, Loss: 1.336461
Epoch: 221, Loss: 1.050348
Epoch: 222, Loss: 1.248122
Epoch: 223, Loss: 1.128356
Epoch: 224, Loss: 1.229037
Epoch: 225, Loss: 1.225406
Epoch: 226, Loss: 1.191774
Epoch: 227, Loss: 1.187903
Epoch: 228, Loss: 1.209955
Epoch: 229, Loss: 1.178248
Epoch: 230, Loss: 1.247566
Epoch: 231, Loss: 1.127708
Epoch: 232, Loss: 1.300509
Epoch: 233, Loss: 1.099195
Epoch: 234, Loss: 1.257511
Epoch: 235, Loss: 1.217770
Epoch: 236, Loss: 1.040972
Epoch: 237, Loss: 1.257901
Epoch: 238, Loss: 1.294452
Epoch: 239, Loss: 1.195047
```

```
Epoch: 240, Loss: 1.174431
Epoch: 241, Loss: 1.214147
Epoch: 242, Loss: 1.288073
Epoch: 243, Loss: 1.042904
Epoch: 244, Loss: 1.004369
Epoch: 245, Loss: 1.049875
Epoch: 246, Loss: 1.198838
Epoch: 247, Loss: 1.193267
Epoch: 248, Loss: 1.377081
Epoch: 249, Loss: 1.200194
Epoch: 250, Loss: 1.187020
Epoch: 251, Loss: 1.005733
Epoch: 252, Loss: 1.332182
Epoch: 253, Loss: 1.044926
Epoch: 254, Loss: 0.949741
Epoch: 255, Loss: 1.214462
Epoch: 256, Loss: 1.227793
Epoch: 257, Loss: 1.412740
Epoch: 258, Loss: 1.064264
Epoch: 259, Loss: 1.350299
Epoch: 260, Loss: 1.076716
Epoch: 261, Loss: 1.057215
Epoch: 262, Loss: 1.149963
Epoch: 263, Loss: 1.048952
Epoch: 264, Loss: 1.075009
Epoch: 265, Loss: 1.281075
Epoch: 266, Loss: 1.061598
Epoch: 267, Loss: 1.380394
Epoch: 268, Loss: 0.994005
Epoch: 269, Loss: 1.280600
Epoch: 270, Loss: 0.971542
Epoch: 271, Loss: 1.159168
Epoch: 272, Loss: 1.299902
Epoch: 273, Loss: 1.052519
Epoch: 274, Loss: 1.113684
Epoch: 275, Loss: 0.966753
Epoch: 276, Loss: 0.941142
Epoch: 277, Loss: 1.035878
Epoch: 278, Loss: 1.011306
Epoch: 279, Loss: 1.020775
Epoch: 280, Loss: 1.110091
Epoch: 281, Loss: 1.096480
Epoch: 282, Loss: 1.016505
Epoch: 283, Loss: 1.010025
Epoch: 284, Loss: 1.239608
Epoch: 285, Loss: 1.381129
Epoch: 286, Loss: 1.041666
Epoch: 287, Loss: 1.222587
Epoch: 288, Loss: 1.140830
Epoch: 289, Loss: 1.011175
Epoch: 290, Loss: 0.990518
Epoch: 291, Loss: 1.042813
Epoch: 292, Loss: 1.062636
Epoch: 293, Loss: 1.141650
Epoch: 294, Loss: 1.178998
Epoch: 295, Loss: 1.013862
Epoch: 296, Loss: 1.138582
Epoch: 297, Loss: 1.406964
Epoch: 298, Loss: 1.016502
Epoch: 299, Loss: 1.147049
```

In [189…
```
val_loader = torch.utils.data.DataLoader(cifar10_val, batch_size=64, shuffle=False)
correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
            imgs, labels = imgs.to(device), labels.to(device)
            batch_size = imgs.shape[0]
            outputs = model_a(imgs.view(batch_size, -1))
            _, predicted = torch.max(outputs, dim=1)
            total += labels.shape[0]
            correct += int((predicted == labels).sum())
    print("Accuracy: {:.2f}".format(correct / total))
```

```
Accuracy: 0.49
```

In [207…
```
cuda.empty_cache()
```

In [208…
```
#three hidden layers
model_b = nn.Sequential(
            nn.Linear(3072,1028),
            nn.LeakyReLU(),
            nn.Linear(1028, 256),
            nn.LeakyReLU(),
            nn.Linear(256,128),
            nn.LeakyReLU(),
            nn.Linear(128, 10),
            nn.LeakyReLU(),
            nn.LogSoftmax(dim=1))
```

In [209…
```
model_b.to(device)
```

Out[209]:
```
Sequential(
  (0): Linear(in_features=3072, out_features=1028, bias=True)
  (1): LeakyReLU(negative_slope=0.01)
  (2): Linear(in_features=1028, out_features=256, bias=True)
  (3): LeakyReLU(negative_slope=0.01)
  (4): Linear(in_features=256, out_features=128, bias=True)
  (5): LeakyReLU(negative_slope=0.01)
  (6): Linear(in_features=128, out_features=10, bias=True)
  (7): LeakyReLU(negative_slope=0.01)
  (8): LogSoftmax(dim=1)
)
```

In [212…
```
l_r = .002
num_epochs = 300
```

In [213…
```
optimizer = torch.optim.SGD(model_b.parameters(),lr=l_r)
train_loader = torch.utils.data.DataLoader(cifar10, batch_size=64, shuffle=True)
loss_fn = nn.NLLLoss()
```

In [214…
```
for epoch in range(num_epochs):
    for imgs, labels in train_loader:
        imgs, labels = imgs.to(device), labels.to(device)
        batch_size = imgs.shape[0]
        outputs = model_b(imgs.view(batch_size, -1))
        loss = loss_fn(outputs, labels)

        optimizer.zero_grad()
```

```
        loss.backward()
        optimizer.step()

    print("Epoch: %d, Loss: %f" % (epoch, float(loss)))
```

```
Epoch: 0, Loss: 2.285945
Epoch: 1, Loss: 2.135777
Epoch: 2, Loss: 2.190054
Epoch: 3, Loss: 2.230818
Epoch: 4, Loss: 1.935544
Epoch: 5, Loss: 2.305283
Epoch: 6, Loss: 2.045128
Epoch: 7, Loss: 1.467916
Epoch: 8, Loss: 2.027364
Epoch: 9, Loss: 1.493638
Epoch: 10, Loss: 1.303160
Epoch: 11, Loss: 1.815007
Epoch: 12, Loss: 1.256967
Epoch: 13, Loss: 1.604033
Epoch: 14, Loss: 1.311588
Epoch: 15, Loss: 1.388225
Epoch: 16, Loss: 1.227933
Epoch: 17, Loss: 1.781184
Epoch: 18, Loss: 1.533697
Epoch: 19, Loss: 1.347286
Epoch: 20, Loss: 1.151213
Epoch: 21, Loss: 1.406174
Epoch: 22, Loss: 1.322008
Epoch: 23, Loss: 1.036471
Epoch: 24, Loss: 1.042701
Epoch: 25, Loss: 1.200212
Epoch: 26, Loss: 1.334431
Epoch: 27, Loss: 1.262972
Epoch: 28, Loss: 0.866911
Epoch: 29, Loss: 1.512697
Epoch: 30, Loss: 1.009436
Epoch: 31, Loss: 1.081901
Epoch: 32, Loss: 1.071949
Epoch: 33, Loss: 0.969334
Epoch: 34, Loss: 0.844799
Epoch: 35, Loss: 1.138213
Epoch: 36, Loss: 0.860988
Epoch: 37, Loss: 0.821808
Epoch: 38, Loss: 1.159904
Epoch: 39, Loss: 0.896800
Epoch: 40, Loss: 0.953415
Epoch: 41, Loss: 1.144317
Epoch: 42, Loss: 0.909019
Epoch: 43, Loss: 0.881377
Epoch: 44, Loss: 0.689440
Epoch: 45, Loss: 1.110556
Epoch: 46, Loss: 2.073138
Epoch: 47, Loss: 0.921748
Epoch: 48, Loss: 0.684872
Epoch: 49, Loss: 1.233255
Epoch: 50, Loss: 0.955525
Epoch: 51, Loss: 0.884563
Epoch: 52, Loss: 0.635976
Epoch: 53, Loss: 0.535951
Epoch: 54, Loss: 0.581201
Epoch: 55, Loss: 0.382846
Epoch: 56, Loss: 0.343779
Epoch: 57, Loss: 0.720704
Epoch: 58, Loss: 0.917043
Epoch: 59, Loss: 0.416624
```

```
Epoch: 60, Loss: 0.701541
Epoch: 61, Loss: 0.743858
Epoch: 62, Loss: 0.651975
Epoch: 63, Loss: 0.714348
Epoch: 64, Loss: 0.473056
Epoch: 65, Loss: 0.558682
Epoch: 66, Loss: 0.204254
Epoch: 67, Loss: 0.295151
Epoch: 68, Loss: 0.403729
Epoch: 69, Loss: 0.506586
Epoch: 70, Loss: 0.491957
Epoch: 71, Loss: 0.365740
Epoch: 72, Loss: 0.522635
Epoch: 73, Loss: 0.415144
Epoch: 74, Loss: 0.472173
Epoch: 75, Loss: 0.168747
Epoch: 76, Loss: 0.208151
Epoch: 77, Loss: 0.563441
Epoch: 78, Loss: 0.550688
Epoch: 79, Loss: 0.672621
Epoch: 80, Loss: 0.444877
Epoch: 81, Loss: 0.262447
Epoch: 82, Loss: 1.264537
Epoch: 83, Loss: 0.247600
Epoch: 84, Loss: 0.206034
Epoch: 85, Loss: 0.702743
Epoch: 86, Loss: 0.464836
Epoch: 87, Loss: 0.301031
Epoch: 88, Loss: 0.234802
Epoch: 89, Loss: 0.180219
Epoch: 90, Loss: 0.225894
Epoch: 91, Loss: 0.092863
Epoch: 92, Loss: 0.103877
Epoch: 93, Loss: 0.177886
Epoch: 94, Loss: 0.228283
Epoch: 95, Loss: 0.211031
Epoch: 96, Loss: 0.064983
Epoch: 97, Loss: 0.102282
Epoch: 98, Loss: 0.272363
Epoch: 99, Loss: 0.084288
Epoch: 100, Loss: 0.268418
Epoch: 101, Loss: 0.110656
Epoch: 102, Loss: 0.196368
Epoch: 103, Loss: 0.076505
Epoch: 104, Loss: 0.051756
Epoch: 105, Loss: 0.292540
Epoch: 106, Loss: 0.090688
Epoch: 107, Loss: 0.094763
Epoch: 108, Loss: 0.035885
Epoch: 109, Loss: 0.108444
Epoch: 110, Loss: 0.039380
Epoch: 111, Loss: 0.032376
Epoch: 112, Loss: 0.020713
Epoch: 113, Loss: 0.012070
Epoch: 114, Loss: 0.073469
Epoch: 115, Loss: 0.056500
Epoch: 116, Loss: 0.221749
Epoch: 117, Loss: 0.014228
Epoch: 118, Loss: 0.029506
Epoch: 119, Loss: 0.020565
```

```
Epoch: 120, Loss: 0.021179
Epoch: 121, Loss: 0.028889
Epoch: 122, Loss: 0.026943
Epoch: 123, Loss: 0.018928
Epoch: 124, Loss: 0.023486
Epoch: 125, Loss: 0.468663
Epoch: 126, Loss: 0.051810
Epoch: 127, Loss: 0.027645
Epoch: 128, Loss: 0.012925
Epoch: 129, Loss: 0.022334
Epoch: 130, Loss: 0.024713
Epoch: 131, Loss: 0.033306
Epoch: 132, Loss: 0.034772
Epoch: 133, Loss: 0.013071
Epoch: 134, Loss: 0.015236
Epoch: 135, Loss: 0.009538
Epoch: 136, Loss: 0.007895
Epoch: 137, Loss: 0.009594
Epoch: 138, Loss: 0.014215
Epoch: 139, Loss: 0.020986
Epoch: 140, Loss: 0.026214
Epoch: 141, Loss: 0.009609
Epoch: 142, Loss: 0.010261
Epoch: 143, Loss: 0.006572
Epoch: 144, Loss: 0.007649
Epoch: 145, Loss: 0.009657
Epoch: 146, Loss: 0.010738
Epoch: 147, Loss: 0.010061
Epoch: 148, Loss: 0.006449
Epoch: 149, Loss: 0.010444
Epoch: 150, Loss: 0.007531
Epoch: 151, Loss: 0.011415
Epoch: 152, Loss: 0.008009
Epoch: 153, Loss: 0.006829
Epoch: 154, Loss: 0.007416
Epoch: 155, Loss: 0.006150
Epoch: 156, Loss: 0.014792
Epoch: 157, Loss: 0.002988
Epoch: 158, Loss: 0.003128
Epoch: 159, Loss: 0.005882
Epoch: 160, Loss: 0.004456
Epoch: 161, Loss: 0.002080
Epoch: 162, Loss: 0.005154
Epoch: 163, Loss: 0.005618
Epoch: 164, Loss: 0.003652
Epoch: 165, Loss: 0.002631
Epoch: 166, Loss: 0.004173
Epoch: 167, Loss: 0.005059
Epoch: 168, Loss: 0.005445
Epoch: 169, Loss: 0.001078
Epoch: 170, Loss: 0.004871
Epoch: 171, Loss: 0.006292
Epoch: 172, Loss: 0.004173
Epoch: 173, Loss: 0.005011
Epoch: 174, Loss: 0.004653
Epoch: 175, Loss: 0.005573
Epoch: 176, Loss: 0.004328
Epoch: 177, Loss: 0.003396
Epoch: 178, Loss: 0.003911
Epoch: 179, Loss: 0.002084
```

```
Epoch: 180, Loss: 0.007012
Epoch: 181, Loss: 0.003442
Epoch: 182, Loss: 0.004043
Epoch: 183, Loss: 0.002252
Epoch: 184, Loss: 0.003406
Epoch: 185, Loss: 0.007055
Epoch: 186, Loss: 0.004751
Epoch: 187, Loss: 0.003935
Epoch: 188, Loss: 0.006408
Epoch: 189, Loss: 0.004055
Epoch: 190, Loss: 0.005277
Epoch: 191, Loss: 0.004654
Epoch: 192, Loss: 0.003289
Epoch: 193, Loss: 0.002565
Epoch: 194, Loss: 0.003702
Epoch: 195, Loss: 0.001847
Epoch: 196, Loss: 0.003209
Epoch: 197, Loss: 0.003377
Epoch: 198, Loss: 0.003209
Epoch: 199, Loss: 0.006801
Epoch: 200, Loss: 0.003516
Epoch: 201, Loss: 0.002549
Epoch: 202, Loss: 0.004444
Epoch: 203, Loss: 0.001869
Epoch: 204, Loss: 0.003309
Epoch: 205, Loss: 0.002385
Epoch: 206, Loss: 0.000973
Epoch: 207, Loss: 0.003645
Epoch: 208, Loss: 0.005503
Epoch: 209, Loss: 0.002074
Epoch: 210, Loss: 0.002961
Epoch: 211, Loss: 0.001505
Epoch: 212, Loss: 0.002248
Epoch: 213, Loss: 0.002810
Epoch: 214, Loss: 0.001645
Epoch: 215, Loss: 0.002030
Epoch: 216, Loss: 0.002969
Epoch: 217, Loss: 0.001945
Epoch: 218, Loss: 0.001648
Epoch: 219, Loss: 0.002524
Epoch: 220, Loss: 0.002113
Epoch: 221, Loss: 0.002072
Epoch: 222, Loss: 0.003322
Epoch: 223, Loss: 0.002518
Epoch: 224, Loss: 0.002745
Epoch: 225, Loss: 0.002987
Epoch: 226, Loss: 0.001906
Epoch: 227, Loss: 0.001970
Epoch: 228, Loss: 0.001547
Epoch: 229, Loss: 0.001639
Epoch: 230, Loss: 0.001561
Epoch: 231, Loss: 0.002109
Epoch: 232, Loss: 0.003423
Epoch: 233, Loss: 0.002331
Epoch: 234, Loss: 0.004265
Epoch: 235, Loss: 0.002893
Epoch: 236, Loss: 0.002158
Epoch: 237, Loss: 0.001230
Epoch: 238, Loss: 0.002117
Epoch: 239, Loss: 0.001227
```

```
Epoch: 240, Loss: 0.002409
Epoch: 241, Loss: 0.001879
Epoch: 242, Loss: 0.001544
Epoch: 243, Loss: 0.003279
Epoch: 244, Loss: 0.002146
Epoch: 245, Loss: 0.001464
Epoch: 246, Loss: 0.001745
Epoch: 247, Loss: 0.000913
Epoch: 248, Loss: 0.004269
Epoch: 249, Loss: 0.002417
Epoch: 250, Loss: 0.001442
Epoch: 251, Loss: 0.001648
Epoch: 252, Loss: 0.001787
Epoch: 253, Loss: 0.001064
Epoch: 254, Loss: 0.001010
Epoch: 255, Loss: 0.002082
Epoch: 256, Loss: 0.000884
Epoch: 257, Loss: 0.002935
Epoch: 258, Loss: 0.001313
Epoch: 259, Loss: 0.001870
Epoch: 260, Loss: 0.003327
Epoch: 261, Loss: 0.001402
Epoch: 262, Loss: 0.001692
Epoch: 263, Loss: 0.001343
Epoch: 264, Loss: 0.001042
Epoch: 265, Loss: 0.001711
Epoch: 266, Loss: 0.001552
Epoch: 267, Loss: 0.002564
Epoch: 268, Loss: 0.001474
Epoch: 269, Loss: 0.000686
Epoch: 270, Loss: 0.002680
Epoch: 271, Loss: 0.002192
Epoch: 272, Loss: 0.002246
Epoch: 273, Loss: 0.001773
Epoch: 274, Loss: 0.001073
Epoch: 275, Loss: 0.002191
Epoch: 276, Loss: 0.001142
Epoch: 277, Loss: 0.002218
Epoch: 278, Loss: 0.001840
Epoch: 279, Loss: 0.001816
Epoch: 280, Loss: 0.001150
Epoch: 281, Loss: 0.000891
Epoch: 282, Loss: 0.001217
Epoch: 283, Loss: 0.000816
Epoch: 284, Loss: 0.001008
Epoch: 285, Loss: 0.000944
Epoch: 286, Loss: 0.001725
Epoch: 287, Loss: 0.001125
Epoch: 288, Loss: 0.001571
Epoch: 289, Loss: 0.000685
Epoch: 290, Loss: 0.001411
Epoch: 291, Loss: 0.000570
Epoch: 292, Loss: 0.000803
Epoch: 293, Loss: 0.000782
Epoch: 294, Loss: 0.001066
Epoch: 295, Loss: 0.001332
Epoch: 296, Loss: 0.001636
Epoch: 297, Loss: 0.000659
Epoch: 298, Loss: 0.000873
Epoch: 299, Loss: 0.001599
```

In [217…

```
val_loader = torch.utils.data.DataLoader(cifar10_val, batch_size=64, shuffle=False)
correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
            imgs, labels = imgs.to(device), labels.to(device)
            batch_size = imgs.shape[0]
            outputs = model_b(imgs.view(batch_size, -1))
            _, predicted = torch.max(outputs, dim=1)
            total += labels.shape[0]
            correct += int((predicted == labels).sum())
    print("Accuracy: {:.2f}".format(correct / total))
```

```
Accuracy: 0.54
```