In [35]:

```
import ssl
import torch
ssl._create_default_https_context = ssl._create_unverified_context
from torchvision import datasets
data_path = "./data/"
cifar10 = datasets.CIFAR10(data_path, train=True, download=True)
cifar10_val = datasets.CIFAR10(data_path, train=False, download=True)
```

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

In [36]:

```
from torchvision import transforms
tensor_cifar10 = datasets.CIFAR10(data_path, train=True, download=False, transform=transfo
imgs = torch.stack([img_t for img_t, _ in tensor_cifar10], dim=3)
imgs.shape
tensor_cifar10_val = datasets.CIFAR10(data_path, train=False, download=False, transform=tr
imgs_val = torch.stack([img_t for img_t, _ in tensor_cifar10_val], dim=3)
imgs_val.shape
```

Out[36]:

```
torch.Size([3, 32, 32, 10000])
```

In [37]:

```
transforms.Normalize((0.4914, 0.4822, 0.4465),  (0.2470, 0.2435, 0.2616))
```

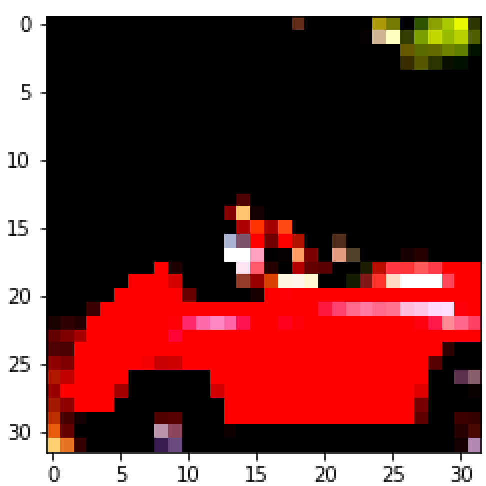
Out[37]:

```
Normalize(mean=(0.4914, 0.4822, 0.4465), std=(0.247, 0.2435, 0.2616))
```

In [38]:

```
import matplotlib.pyplot as plt
transformed_cifar10 = datasets.CIFAR10(data_path, train=True, download=False, transform=tr
img_t, _  = transformed_cifar10[99]
plt.imshow(img_t.permute(1, 2, 0))
plt.show()
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
255] for integers).
```



In [39]:

```
transforms.Normalize((0.4942, 0.4851, 0.4504),  (0.2467, 0.2429, 0.2616))
val_transformed_cifar10 = datasets.CIFAR10(data_path, train=False, download=False, transfo
```

In [40]:

```
import torch.nn.functional as F
import torch.nn as nn
class CNN(nn.Module):
    def __init__(self, n_chans1=32, n_blocks=5):
        super().__init__()
```

```python
        self.n_chans1 = n_chans1
        self.conv1 = nn.Conv2d(3, n_chans1, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(n_chans1, n_chans1, kernel_size=3, padding=1)
        self.fc1 = nn.Linear(8 * 8 * n_chans1, 32)
        self.fc2 = nn.Linear(32, 10)
    def forward(self, x):
        out = F.max_pool2d(torch.tanh(self.conv1(x)), 2)
        out = F.max_pool2d(torch.tanh(self.conv2(out)), 2)
        out = out.view(-1, 8 * 8 * self.n_chans1)
        out = torch.tanh(self.fc1(out))
        out = self.fc2(out)
        return out
```

In [41]:

```python
import torch
device = torch.device('cuda:0')
train_loader = torch.utils.data.DataLoader(transformed_cifar10, batch_size=64, shuffle=Tru

model = CNN()
model.to(device)
loss_function = nn.CrossEntropyLoss()
learning_rate = 1e-4
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

In [42]:

```python
import time
def training(model, optimizer, loss_fn, n_epochs, device, train_loader):
    start = time.time()

    for epoch in range(n_epochs):
        for imgs, labels in train_loader:
            imgs = imgs.to(device)
            labels = labels.to(device)
            batch_size = imgs.shape[0]
            outputs = model(imgs)
            loss = loss_fn((outputs), labels)

            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
        print("Epoch: %d, Loss: %f" % (epoch, float(loss)))
    end = time.time()
    print(end - start)
```

In [43]:

```python
n_epochs = 300
training(model, optimizer, loss_function, n_epochs, device, train_loader)
```

```
Epoch: 0, Loss: 1.498447
Epoch: 1, Loss: 1.497896
Epoch: 2, Loss: 1.313029
Epoch: 3, Loss: 0.725449
Epoch: 4, Loss: 1.140860
Epoch: 5, Loss: 0.979368
Epoch: 6, Loss: 0.955193
Epoch: 7, Loss: 0.922603
Epoch: 8, Loss: 0.877520
Epoch: 9, Loss: 1.048637
Epoch: 10, Loss: 0.969260
Epoch: 11, Loss: 0.982045
Epoch: 12, Loss: 0.776672
Epoch: 13, Loss: 0.345156
Epoch: 14, Loss: 0.983632
Epoch: 15, Loss: 0.762592
```

```
Epoch: 16, Loss: 0.937057
Epoch: 17, Loss: 0.873066
Epoch: 18, Loss: 0.701364
Epoch: 19, Loss: 0.904475
Epoch: 20, Loss: 0.554761
Epoch: 21, Loss: 1.172717
Epoch: 22, Loss: 0.972105
Epoch: 23, Loss: 0.560831
Epoch: 24, Loss: 0.362465
Epoch: 25, Loss: 0.488543
Epoch: 26, Loss: 0.514368
Epoch: 27, Loss: 0.417470
Epoch: 28, Loss: 0.634012
Epoch: 29, Loss: 0.876366
Epoch: 30, Loss: 0.836783
Epoch: 31, Loss: 0.840493
Epoch: 32, Loss: 0.869559
Epoch: 33, Loss: 0.839005
Epoch: 34, Loss: 1.090934
Epoch: 35, Loss: 0.555672
Epoch: 36, Loss: 0.724887
Epoch: 37, Loss: 0.855388
Epoch: 38, Loss: 0.477666
Epoch: 39, Loss: 0.570756
Epoch: 40, Loss: 0.477735
Epoch: 41, Loss: 0.197177
Epoch: 42, Loss: 0.531599
Epoch: 43, Loss: 0.571237
Epoch: 44, Loss: 0.346088
Epoch: 45, Loss: 0.477955
Epoch: 46, Loss: 0.348622
Epoch: 47, Loss: 0.376266
Epoch: 48, Loss: 0.588658
Epoch: 49, Loss: 0.534189
Epoch: 50, Loss: 0.496581
Epoch: 51, Loss: 0.524668
Epoch: 52, Loss: 0.512729
Epoch: 53, Loss: 0.410188
Epoch: 54, Loss: 0.632845
Epoch: 55, Loss: 0.855601
Epoch: 56, Loss: 0.321324
Epoch: 57, Loss: 0.616294
Epoch: 58, Loss: 0.762693
Epoch: 59, Loss: 0.605400
Epoch: 60, Loss: 0.404862
Epoch: 61, Loss: 0.392822
Epoch: 62, Loss: 0.694334
Epoch: 63, Loss: 0.506469
Epoch: 64, Loss: 0.725394
Epoch: 65, Loss: 0.351106
Epoch: 66, Loss: 0.326640
Epoch: 67, Loss: 0.358067
Epoch: 68, Loss: 0.347434
Epoch: 69, Loss: 0.277963
Epoch: 70, Loss: 0.242718
Epoch: 71, Loss: 0.523904
Epoch: 72, Loss: 0.572263
Epoch: 73, Loss: 0.399414
Epoch: 74, Loss: 0.857042
Epoch: 75, Loss: 0.556908
Epoch: 76, Loss: 0.580345
Epoch: 77, Loss: 0.475995
Epoch: 78, Loss: 0.675956
Epoch: 79, Loss: 0.484926
Epoch: 80, Loss: 0.627784
Epoch: 81, Loss: 0.701546
```

```
Epoch: 82, Loss: 0.192024
Epoch: 83, Loss: 0.220258
Epoch: 84, Loss: 0.600897
Epoch: 85, Loss: 0.427027
Epoch: 86, Loss: 0.357047
Epoch: 87, Loss: 0.270161
Epoch: 88, Loss: 0.182802
Epoch: 89, Loss: 0.320588
Epoch: 90, Loss: 0.169319
Epoch: 91, Loss: 0.604976
Epoch: 92, Loss: 0.244069
Epoch: 93, Loss: 0.278588
Epoch: 94, Loss: 0.361861
Epoch: 95, Loss: 0.168244
Epoch: 96, Loss: 0.515295
Epoch: 97, Loss: 0.380831
Epoch: 98, Loss: 0.101507
Epoch: 99, Loss: 0.417905
Epoch: 100, Loss: 0.584851
Epoch: 101, Loss: 0.358312
Epoch: 102, Loss: 0.163078
Epoch: 103, Loss: 0.154197
Epoch: 104, Loss: 0.166170
Epoch: 105, Loss: 0.150583
Epoch: 106, Loss: 0.218236
Epoch: 107, Loss: 0.378485
Epoch: 108, Loss: 0.255147
Epoch: 109, Loss: 0.297278
Epoch: 110, Loss: 0.192649
Epoch: 111, Loss: 0.205934
Epoch: 112, Loss: 0.101475
Epoch: 113, Loss: 0.155968
Epoch: 114, Loss: 0.240980
Epoch: 115, Loss: 0.129733
Epoch: 116, Loss: 0.067875
Epoch: 117, Loss: 0.116800
Epoch: 118, Loss: 0.168921
Epoch: 119, Loss: 0.152258
Epoch: 120, Loss: 0.223537
Epoch: 121, Loss: 0.289648
Epoch: 122, Loss: 0.386241
Epoch: 123, Loss: 0.199642
Epoch: 124, Loss: 0.389101
Epoch: 125, Loss: 0.112741
Epoch: 126, Loss: 0.455202
Epoch: 127, Loss: 0.115323
Epoch: 128, Loss: 0.124189
Epoch: 129, Loss: 0.096847
Epoch: 130, Loss: 0.146255
Epoch: 131, Loss: 0.137659
Epoch: 132, Loss: 0.323041
Epoch: 133, Loss: 0.142082
Epoch: 134, Loss: 0.188160
Epoch: 135, Loss: 0.113059
Epoch: 136, Loss: 0.033434
Epoch: 137, Loss: 0.097937
Epoch: 138, Loss: 0.085505
Epoch: 139, Loss: 0.216704
Epoch: 140, Loss: 0.075677
Epoch: 141, Loss: 0.151857
Epoch: 142, Loss: 0.299328
Epoch: 143, Loss: 0.122950
Epoch: 144, Loss: 0.108964
Epoch: 145, Loss: 0.172743
Epoch: 146, Loss: 0.218228
Epoch: 147, Loss: 0.084273
```

```
Epoch: 148, Loss: 0.156723
Epoch: 149, Loss: 0.103930
Epoch: 150, Loss: 0.064997
Epoch: 151, Loss: 0.123085
Epoch: 152, Loss: 0.084942
Epoch: 153, Loss: 0.187394
Epoch: 154, Loss: 0.091330
Epoch: 155, Loss: 0.109375
Epoch: 156, Loss: 0.140404
Epoch: 157, Loss: 0.149521
Epoch: 158, Loss: 0.124805
Epoch: 159, Loss: 0.106408
Epoch: 160, Loss: 0.039498
Epoch: 161, Loss: 0.106182
Epoch: 162, Loss: 0.039681
Epoch: 163, Loss: 0.095973
Epoch: 164, Loss: 0.086228
Epoch: 165, Loss: 0.085590
Epoch: 166, Loss: 0.054242
Epoch: 167, Loss: 0.243868
Epoch: 168, Loss: 0.038838
Epoch: 169, Loss: 0.039462
Epoch: 170, Loss: 0.139226
Epoch: 171, Loss: 0.110857
Epoch: 172, Loss: 0.036935
Epoch: 173, Loss: 0.106023
Epoch: 174, Loss: 0.040010
Epoch: 175, Loss: 0.016204
Epoch: 176, Loss: 0.047791
Epoch: 177, Loss: 0.083080
Epoch: 178, Loss: 0.095875
Epoch: 179, Loss: 0.056258
Epoch: 180, Loss: 0.035833
Epoch: 181, Loss: 0.028847
Epoch: 182, Loss: 0.040698
Epoch: 183, Loss: 0.054574
Epoch: 184, Loss: 0.052165
Epoch: 185, Loss: 0.035904
Epoch: 186, Loss: 0.018172
Epoch: 187, Loss: 0.012073
Epoch: 188, Loss: 0.036133
Epoch: 189, Loss: 0.026720
Epoch: 190, Loss: 0.056869
Epoch: 191, Loss: 0.024018
Epoch: 192, Loss: 0.037674
Epoch: 193, Loss: 0.036731
Epoch: 194, Loss: 0.097569
Epoch: 195, Loss: 0.060853
Epoch: 196, Loss: 0.019426
Epoch: 197, Loss: 0.031833
Epoch: 198, Loss: 0.047987
Epoch: 199, Loss: 0.036798
Epoch: 200, Loss: 0.046242
Epoch: 201, Loss: 0.020834
Epoch: 202, Loss: 0.025545
Epoch: 203, Loss: 0.067831
Epoch: 204, Loss: 0.010615
Epoch: 205, Loss: 0.034919
Epoch: 206, Loss: 0.180730
Epoch: 207, Loss: 0.021324
Epoch: 208, Loss: 0.028950
Epoch: 209, Loss: 0.021941
Epoch: 210, Loss: 0.025035
Epoch: 211, Loss: 0.046879
Epoch: 212, Loss: 0.011836
Epoch: 213, Loss: 0.020497
```

```
Epoch: 214, Loss: 0.021238
Epoch: 215, Loss: 0.017562
Epoch: 216, Loss: 0.015915
Epoch: 217, Loss: 0.035125
Epoch: 218, Loss: 0.030752
Epoch: 219, Loss: 0.037759
Epoch: 220, Loss: 0.012466
Epoch: 221, Loss: 0.026364
Epoch: 222, Loss: 0.021804
Epoch: 223, Loss: 0.015536
Epoch: 224, Loss: 0.010265
Epoch: 225, Loss: 0.019381
Epoch: 226, Loss: 0.013898
Epoch: 227, Loss: 0.013814
Epoch: 228, Loss: 0.016014
Epoch: 229, Loss: 0.016466
Epoch: 230, Loss: 0.007997
Epoch: 231, Loss: 0.009654
Epoch: 232, Loss: 0.010657
Epoch: 233, Loss: 0.017986
Epoch: 234, Loss: 0.008797
Epoch: 235, Loss: 0.007140
Epoch: 236, Loss: 0.006865
Epoch: 237, Loss: 0.016767
Epoch: 238, Loss: 0.009168
Epoch: 239, Loss: 0.017544
Epoch: 240, Loss: 0.048089
Epoch: 241, Loss: 0.016886
Epoch: 242, Loss: 0.013881
Epoch: 243, Loss: 0.028045
Epoch: 244, Loss: 0.012124
Epoch: 245, Loss: 0.003064
Epoch: 246, Loss: 0.025037
Epoch: 247, Loss: 0.003216
Epoch: 248, Loss: 0.007882
Epoch: 249, Loss: 0.005541
Epoch: 250, Loss: 0.001212
Epoch: 251, Loss: 0.005447
Epoch: 252, Loss: 0.007220
Epoch: 253, Loss: 0.008085
Epoch: 254, Loss: 0.005604
Epoch: 255, Loss: 0.006485
Epoch: 256, Loss: 0.002781
Epoch: 257, Loss: 0.002156
Epoch: 258, Loss: 0.011245
Epoch: 259, Loss: 0.003058
Epoch: 260, Loss: 0.004969
Epoch: 261, Loss: 0.005466
Epoch: 262, Loss: 0.006512
Epoch: 263, Loss: 0.001763
Epoch: 264, Loss: 0.004633
Epoch: 265, Loss: 0.001871
Epoch: 266, Loss: 0.008104
Epoch: 267, Loss: 0.014022
Epoch: 268, Loss: 0.011453
Epoch: 269, Loss: 0.001730
Epoch: 270, Loss: 0.005452
Epoch: 271, Loss: 0.003399
Epoch: 272, Loss: 0.002999
Epoch: 273, Loss: 0.001531
Epoch: 274, Loss: 0.013616
Epoch: 275, Loss: 0.004004
Epoch: 276, Loss: 0.015762
Epoch: 277, Loss: 0.002352
Epoch: 278, Loss: 0.002196
Epoch: 279, Loss: 0.010473
```

```
Epoch: 280, Loss: 0.007576
Epoch: 281, Loss: 0.002678
Epoch: 282, Loss: 0.003726
Epoch: 283, Loss: 0.001975
Epoch: 284, Loss: 0.006700
Epoch: 285, Loss: 0.002702
Epoch: 286, Loss: 0.004033
Epoch: 287, Loss: 0.004906
Epoch: 288, Loss: 0.005658
Epoch: 289, Loss: 0.002335
Epoch: 290, Loss: 0.002274
Epoch: 291, Loss: 0.002801
Epoch: 292, Loss: 0.002769
Epoch: 293, Loss: 0.002543
Epoch: 294, Loss: 0.005466
Epoch: 295, Loss: 0.004295
Epoch: 296, Loss: 0.004386
Epoch: 297, Loss: 0.003075
Epoch: 298, Loss: 0.000987
Epoch: 299, Loss: 0.002263
2617.700516939163
```

In [44]:

```
from ptflops import get_model_complexity_info


macs, params = get_model_complexity_info(model, ( 3, 32,32), as_strings=True, print_per_la
print('{:<30}  {:<8}'.format('Computational complexity: ', macs))
print('{:<30}  {:<8}'.format('Number of parameters: ', params))
```

```
Computational complexity:       0.0 GMac
Number of parameters:           76.04 k
```

In [45]:

```
val_loader = torch.utils.data.DataLoader(val_transformed_cifar10, batch_size=64, shuffle=F

correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
        imgs = imgs.to(device)
        labels = labels.to(device)
        batch_size=imgs.shape[0]
        outputs = model(imgs)
        _, predicted = torch.max(outputs, dim=1)
#          print(predicted)
#          print("\n")
#          print(labels)
        total += labels.shape[0]
        correct += int((predicted==labels).sum())
    print("Accuracy ", correct/total)
```

```
Accuracy  0.6443
```

In [46]:

```
#Part 1b
import torch.nn.functional as F
class CNN2(nn.Module):
    def __init__(self, n_chans1=32, n_blocks=5):
        super().__init__()
        self.n_chans1 = n_chans1
        self.conv1 = nn.Conv2d(3, n_chans1, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(n_chans1, 16, kernel_size=3, padding=1)
        self.conv3 = nn.Conv2d(16, n_chans1, kernel_size=3, padding=1)
        self.fc1 = nn.Linear(4 * 4 * n_chans1, 32)
        self.fc2 = nn.Linear(32, 10)
    def forward(self, x):
```

```python
        out = F.max_pool2d(torch.tanh(self.conv1(x)), 2)
        out = F.max_pool2d(torch.tanh(self.conv2(out)), 2)
        out = F.max_pool2d(torch.tanh(self.conv3(out)), 2)
        out = out.view(-1, 4 * 4 * self.n_chans1)
        out = torch.tanh(self.fc1(out))
        out = self.fc2(out)
        return out
```

In [47]:

```python
import torch
device = torch.device('cuda:0')
train_loader = torch.utils.data.DataLoader(transformed_cifar10, batch_size=64, shuffle=Tru

model = CNN2()
model.to(device)
loss_function = nn.CrossEntropyLoss()
learning_rate = 1e-4
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

In [48]:

```python
import time
def training(model, optimizer, loss_fn, n_epochs, device, train_loader):
    start = time.time()

    for epoch in range(n_epochs):
        for imgs, labels in train_loader:
            imgs = imgs.to(device)
            labels = labels.to(device)

            batch_size = imgs.shape[0]
            outputs = model(imgs)
            loss = loss_fn((outputs), labels)

            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
        print("Epoch: %d, Loss: %f" % (epoch, float(loss)))
    end = time.time()
    print(end - start)
```

In [49]:

```python
n_epochs = 300
training(model, optimizer, loss_function, n_epochs, device, train_loader)
```

```
Epoch: 0, Loss: 1.896323
Epoch: 1, Loss: 1.648372
Epoch: 2, Loss: 1.083320
Epoch: 3, Loss: 1.322986
Epoch: 4, Loss: 1.201101
Epoch: 5, Loss: 1.132529
Epoch: 6, Loss: 1.267703
Epoch: 7, Loss: 1.157219
Epoch: 8, Loss: 1.527774
Epoch: 9, Loss: 0.892078
Epoch: 10, Loss: 0.720922
Epoch: 11, Loss: 1.120539
Epoch: 12, Loss: 0.693970
Epoch: 13, Loss: 1.180377
Epoch: 14, Loss: 1.237997
Epoch: 15, Loss: 0.881170
Epoch: 16, Loss: 0.662154
Epoch: 17, Loss: 1.143472
Epoch: 18, Loss: 0.830828
Epoch: 19, Loss: 0.953425
```

```
Epoch: 20, Loss: 1.368803
Epoch: 21, Loss: 1.404832
Epoch: 22, Loss: 0.600543
Epoch: 23, Loss: 0.858952
Epoch: 24, Loss: 0.946415
Epoch: 25, Loss: 0.583561
Epoch: 26, Loss: 0.418478
Epoch: 27, Loss: 0.705524
Epoch: 28, Loss: 0.718551
Epoch: 29, Loss: 0.548431
Epoch: 30, Loss: 0.621175
Epoch: 31, Loss: 0.785009
Epoch: 32, Loss: 0.765577
Epoch: 33, Loss: 0.861708
Epoch: 34, Loss: 0.878913
Epoch: 35, Loss: 0.491723
Epoch: 36, Loss: 0.741196
Epoch: 37, Loss: 0.602203
Epoch: 38, Loss: 0.510127
Epoch: 39, Loss: 0.564570
Epoch: 40, Loss: 0.663044
Epoch: 41, Loss: 0.779188
Epoch: 42, Loss: 0.820207
Epoch: 43, Loss: 0.959233
Epoch: 44, Loss: 1.153374
Epoch: 45, Loss: 0.873901
Epoch: 46, Loss: 0.847194
Epoch: 47, Loss: 0.733507
Epoch: 48, Loss: 0.779626
Epoch: 49, Loss: 0.694241
Epoch: 50, Loss: 0.260936
Epoch: 51, Loss: 0.729337
Epoch: 52, Loss: 0.960977
Epoch: 53, Loss: 1.128400
Epoch: 54, Loss: 0.529357
Epoch: 55, Loss: 1.252581
Epoch: 56, Loss: 0.933282
Epoch: 57, Loss: 0.348256
Epoch: 58, Loss: 0.422800
Epoch: 59, Loss: 0.563232
Epoch: 60, Loss: 0.424046
Epoch: 61, Loss: 0.856917
Epoch: 62, Loss: 0.405780
Epoch: 63, Loss: 0.423738
Epoch: 64, Loss: 0.815334
Epoch: 65, Loss: 0.363746
Epoch: 66, Loss: 0.484086
Epoch: 67, Loss: 0.666310
Epoch: 68, Loss: 0.289109
Epoch: 69, Loss: 0.574485
Epoch: 70, Loss: 0.468145
Epoch: 71, Loss: 0.252282
Epoch: 72, Loss: 0.709677
Epoch: 73, Loss: 0.909277
Epoch: 74, Loss: 0.386946
Epoch: 75, Loss: 0.479924
Epoch: 76, Loss: 0.933227
Epoch: 77, Loss: 0.691403
Epoch: 78, Loss: 0.275495
Epoch: 79, Loss: 0.697377
Epoch: 80, Loss: 0.633488
Epoch: 81, Loss: 0.812485
Epoch: 82, Loss: 0.430187
Epoch: 83, Loss: 0.645006
Epoch: 84, Loss: 1.022087
Epoch: 85, Loss: 0.619434
```

```
Epoch: 86, Loss: 0.336618
Epoch: 87, Loss: 0.451006
Epoch: 88, Loss: 0.533008
Epoch: 89, Loss: 0.709450
Epoch: 90, Loss: 0.401480
Epoch: 91, Loss: 0.607547
Epoch: 92, Loss: 0.498062
Epoch: 93, Loss: 0.641506
Epoch: 94, Loss: 0.450096
Epoch: 95, Loss: 0.859753
Epoch: 96, Loss: 0.669792
Epoch: 97, Loss: 0.406585
Epoch: 98, Loss: 0.705167
Epoch: 99, Loss: 0.330190
Epoch: 100, Loss: 0.901043
Epoch: 101, Loss: 0.464197
Epoch: 102, Loss: 0.640942
Epoch: 103, Loss: 0.424082
Epoch: 104, Loss: 0.682475
Epoch: 105, Loss: 0.550172
Epoch: 106, Loss: 0.548426
Epoch: 107, Loss: 0.263066
Epoch: 108, Loss: 0.268184
Epoch: 109, Loss: 0.438816
Epoch: 110, Loss: 0.346763
Epoch: 111, Loss: 0.345796
Epoch: 112, Loss: 0.416866
Epoch: 113, Loss: 0.662212
Epoch: 114, Loss: 0.549651
Epoch: 115, Loss: 0.440759
Epoch: 116, Loss: 0.792771
Epoch: 117, Loss: 0.307351
Epoch: 118, Loss: 0.638000
Epoch: 119, Loss: 0.490885
Epoch: 120, Loss: 0.375789
Epoch: 121, Loss: 0.548913
Epoch: 122, Loss: 0.714649
Epoch: 123, Loss: 0.360686
Epoch: 124, Loss: 0.433685
Epoch: 125, Loss: 0.368262
Epoch: 126, Loss: 0.721726
Epoch: 127, Loss: 0.535415
Epoch: 128, Loss: 0.627891
Epoch: 129, Loss: 0.270918
Epoch: 130, Loss: 0.550155
Epoch: 131, Loss: 0.671316
Epoch: 132, Loss: 0.600900
Epoch: 133, Loss: 0.427695
Epoch: 134, Loss: 0.190060
Epoch: 135, Loss: 0.468132
Epoch: 136, Loss: 0.363780
Epoch: 137, Loss: 0.755758
Epoch: 138, Loss: 0.239808
Epoch: 139, Loss: 0.167057
Epoch: 140, Loss: 0.425273
Epoch: 141, Loss: 0.223723
Epoch: 142, Loss: 0.458519
Epoch: 143, Loss: 0.376198
Epoch: 144, Loss: 0.405882
Epoch: 145, Loss: 0.495435
Epoch: 146, Loss: 0.384176
Epoch: 147, Loss: 0.464494
Epoch: 148, Loss: 0.250836
Epoch: 149, Loss: 0.417796
Epoch: 150, Loss: 0.526215
Epoch: 151, Loss: 0.267406
```

```
Epoch: 152, Loss: 0.148104
Epoch: 153, Loss: 0.211547
Epoch: 154, Loss: 0.406436
Epoch: 155, Loss: 0.495581
Epoch: 156, Loss: 0.177228
Epoch: 157, Loss: 0.251501
Epoch: 158, Loss: 0.266415
Epoch: 159, Loss: 0.231716
Epoch: 160, Loss: 0.729060
Epoch: 161, Loss: 0.401378
Epoch: 162, Loss: 0.234087
Epoch: 163, Loss: 0.551553
Epoch: 164, Loss: 0.478218
Epoch: 165, Loss: 0.371285
Epoch: 166, Loss: 0.105495
Epoch: 167, Loss: 0.349293
Epoch: 168, Loss: 0.516784
Epoch: 169, Loss: 0.424125
Epoch: 170, Loss: 0.270033
Epoch: 171, Loss: 0.198812
Epoch: 172, Loss: 0.236980
Epoch: 173, Loss: 0.407601
Epoch: 174, Loss: 0.317935
Epoch: 175, Loss: 0.923415
Epoch: 176, Loss: 1.208022
Epoch: 177, Loss: 0.412102
Epoch: 178, Loss: 0.307048
Epoch: 179, Loss: 0.221508
Epoch: 180, Loss: 0.191000
Epoch: 181, Loss: 0.635185
Epoch: 182, Loss: 0.353264
Epoch: 183, Loss: 0.386095
Epoch: 184, Loss: 0.789814
Epoch: 185, Loss: 0.616831
Epoch: 186, Loss: 0.338618
Epoch: 187, Loss: 0.424749
Epoch: 188, Loss: 0.352902
Epoch: 189, Loss: 0.327687
Epoch: 190, Loss: 0.128037
Epoch: 191, Loss: 0.136699
Epoch: 192, Loss: 0.275025
Epoch: 193, Loss: 0.267811
Epoch: 194, Loss: 0.188329
Epoch: 195, Loss: 0.245726
Epoch: 196, Loss: 0.379887
Epoch: 197, Loss: 0.360750
Epoch: 198, Loss: 0.563856
Epoch: 199, Loss: 0.469363
Epoch: 200, Loss: 0.541940
Epoch: 201, Loss: 0.302168
Epoch: 202, Loss: 0.258219
Epoch: 203, Loss: 0.419230
Epoch: 204, Loss: 0.348259
Epoch: 205, Loss: 0.491737
Epoch: 206, Loss: 0.589031
Epoch: 207, Loss: 0.350189
Epoch: 208, Loss: 0.156740
Epoch: 209, Loss: 0.306332
Epoch: 210, Loss: 0.126633
Epoch: 211, Loss: 0.236091
Epoch: 212, Loss: 0.550422
Epoch: 213, Loss: 0.662598
Epoch: 214, Loss: 0.488895
Epoch: 215, Loss: 0.399305
Epoch: 216, Loss: 0.159977
Epoch: 217, Loss: 0.372328
```

```
Epoch: 218, Loss: 0.192008
Epoch: 219, Loss: 0.123877
Epoch: 220, Loss: 0.390310
Epoch: 221, Loss: 0.397420
Epoch: 222, Loss: 0.759552
Epoch: 223, Loss: 0.293446
Epoch: 224, Loss: 0.221819
Epoch: 225, Loss: 0.352326
Epoch: 226, Loss: 0.176500
Epoch: 227, Loss: 0.688282
Epoch: 228, Loss: 0.507022
Epoch: 229, Loss: 0.734138
Epoch: 230, Loss: 0.217383
Epoch: 231, Loss: 0.125206
Epoch: 232, Loss: 0.372049
Epoch: 233, Loss: 0.564945
Epoch: 234, Loss: 0.303220
Epoch: 235, Loss: 0.191691
Epoch: 236, Loss: 0.656408
Epoch: 237, Loss: 0.297436
Epoch: 238, Loss: 0.143998
Epoch: 239, Loss: 0.242424
Epoch: 240, Loss: 0.367371
Epoch: 241, Loss: 0.249586
Epoch: 242, Loss: 0.334356
Epoch: 243, Loss: 0.179882
Epoch: 244, Loss: 0.172921
Epoch: 245, Loss: 0.163452
Epoch: 246, Loss: 0.383987
Epoch: 247, Loss: 0.441124
Epoch: 248, Loss: 0.672070
Epoch: 249, Loss: 0.188298
Epoch: 250, Loss: 0.351831
Epoch: 251, Loss: 0.100963
Epoch: 252, Loss: 0.217324
Epoch: 253, Loss: 0.330780
Epoch: 254, Loss: 0.350682
Epoch: 255, Loss: 0.420344
Epoch: 256, Loss: 0.273676
Epoch: 257, Loss: 0.162933
Epoch: 258, Loss: 0.640054
Epoch: 259, Loss: 0.284244
Epoch: 260, Loss: 0.328823
Epoch: 261, Loss: 0.357020
Epoch: 262, Loss: 0.296699
Epoch: 263, Loss: 0.395608
Epoch: 264, Loss: 0.365300
Epoch: 265, Loss: 0.418496
Epoch: 266, Loss: 0.095528
Epoch: 267, Loss: 0.730917
Epoch: 268, Loss: 0.223145
Epoch: 269, Loss: 0.129105
Epoch: 270, Loss: 0.529553
Epoch: 271, Loss: 0.438484
Epoch: 272, Loss: 0.220422
Epoch: 273, Loss: 0.406033
Epoch: 274, Loss: 0.497470
Epoch: 275, Loss: 0.611705
Epoch: 276, Loss: 0.405171
Epoch: 277, Loss: 0.298941
Epoch: 278, Loss: 0.060136
Epoch: 279, Loss: 0.278914
Epoch: 280, Loss: 0.272528
Epoch: 281, Loss: 0.319582
Epoch: 282, Loss: 0.272578
Epoch: 283, Loss: 0.196376
```

```
Epoch: 284, Loss: 0.249674
Epoch: 285, Loss: 0.664153
Epoch: 286, Loss: 0.230481
Epoch: 287, Loss: 0.176848
Epoch: 288, Loss: 0.213954
Epoch: 289, Loss: 0.899344
Epoch: 290, Loss: 0.942388
Epoch: 291, Loss: 0.361399
Epoch: 292, Loss: 0.130273
Epoch: 293, Loss: 0.407304
Epoch: 294, Loss: 0.219181
Epoch: 295, Loss: 0.260964
Epoch: 296, Loss: 0.206191
Epoch: 297, Loss: 0.304515
Epoch: 298, Loss: 0.171799
Epoch: 299, Loss: 0.435424
2682.2555429935455
```

In [50]:

```python
from ptflops import get_model_complexity_info


macs, params = get_model_complexity_info(model, ( 3, 32,32), as_strings=True, print_per_la
print('{:<30}  {:<8}'.format('Computational complexity: ', macs))
print('{:<30}  {:<8}'.format('Number of parameters: ', params))
```

```
Computational complexity:       0.0 GMac
Number of parameters:           26.91 k
```

In [51]:

```python
val_loader = torch.utils.data.DataLoader(val_transformed_cifar10, batch_size=64, shuffle=F

correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
        imgs = imgs.to(device)
        labels = labels.to(device)
        batch_size=imgs.shape[0]
        outputs = model(imgs)
        _, predicted = torch.max(outputs, dim=1)
#         print(predicted)
#         print("\n")
#         print(labels)
        total += labels.shape[0]
        correct += int((predicted==labels).sum())
    print("Accuracy ", correct/total)
```

```
Accuracy  0.6989
```

In [52]:

```python
#Part 2
import torch.nn as nn
class ResBlock(nn.Module):
    def __init__(self, n_chans):
        super(ResBlock, self).__init__()
        self.conv = nn.Conv2d(n_chans, n_chans, kernel_size=3, padding=1, bias=False)
        self.batch_norm = nn.BatchNorm2d(num_features=n_chans)
        torch.nn.init.kaiming_normal_(self.conv.weight, nonlinearity='relu')
        torch.nn.init.constant_(self.batch_norm.weight, 0.5)
        torch.nn.init.zeros_(self.batch_norm.bias)

    def forward(self, x):
        out = self.conv(x)
        out = self.batch_norm(out)
        out = torch.relu(out)
```

```
        return out
```

In [53]:

```
import torch.nn.functional as F
class ResNet10(nn.Module):
    def __init__(self, n_chans1=32, n_blocks=10):
        super().__init__()
        self.n_chans1 = n_chans1
        self.conv1 = nn.Conv2d(3, n_chans1, kernel_size=3, padding=1)
        self.ResNetBlocks = nn.Sequential(*(n_blocks * [ResBlock(n_chans=n_chans1)]))
        self.fc1 = nn.Linear(16 * 16 * n_chans1, 32)
        self.fc2 = nn.Linear(32, 10)
    def forward(self, x):
        out = F.max_pool2d(torch.relu(self.conv1(x)), 2)
        out = self.ResNetBlocks(out)
        out = out.view(-1, 16 * 16 * self.n_chans1)
        out = torch.relu(self.fc1(out))
        out = self.fc2(out)
        return out
```

In [54]:

```
import torch
device = torch.device('cuda:0')
train_loader = torch.utils.data.DataLoader(transformed_cifar10, batch_size=64, shuffle=Tru

model = ResNet10()
model.to(device)
loss_function = nn.CrossEntropyLoss()
learning_rate = 3e-3
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

In [55]:

```
n_epochs = 300
training(model, optimizer, loss_function, n_epochs, device, train_loader)
```

```
Epoch: 0, Loss: 1.380295
Epoch: 1, Loss: 1.534053
Epoch: 2, Loss: 1.522577
Epoch: 3, Loss: 1.108553
Epoch: 4, Loss: 0.830498
Epoch: 5, Loss: 1.134960
Epoch: 6, Loss: 0.875195
Epoch: 7, Loss: 0.962270
Epoch: 8, Loss: 0.826196
Epoch: 9, Loss: 0.816926
Epoch: 10, Loss: 0.934372
Epoch: 11, Loss: 1.005834
Epoch: 12, Loss: 1.560953
Epoch: 13, Loss: 0.384691
Epoch: 14, Loss: 1.155713
Epoch: 15, Loss: 0.883912
Epoch: 16, Loss: 0.866532
Epoch: 17, Loss: 0.759370
Epoch: 18, Loss: 0.868821
Epoch: 19, Loss: 0.997360
Epoch: 20, Loss: 0.648042
Epoch: 21, Loss: 0.841270
Epoch: 22, Loss: 0.368993
Epoch: 23, Loss: 0.815533
Epoch: 24, Loss: 0.946853
Epoch: 25, Loss: 0.451778
Epoch: 26, Loss: 0.729739
Epoch: 27, Loss: 0.728167
```

```
Epoch: 28, Loss: 0.459222
Epoch: 29, Loss: 0.848244
Epoch: 30, Loss: 0.874603
Epoch: 31, Loss: 0.883124
Epoch: 32, Loss: 0.394543
Epoch: 33, Loss: 0.528460
Epoch: 34, Loss: 0.280844
Epoch: 35, Loss: 0.585605
Epoch: 36, Loss: 0.984623
Epoch: 37, Loss: 0.662719
Epoch: 38, Loss: 0.800092
Epoch: 39, Loss: 0.422235
Epoch: 40, Loss: 0.761066
Epoch: 41, Loss: 0.978466
Epoch: 42, Loss: 0.361117
Epoch: 43, Loss: 0.944220
Epoch: 44, Loss: 0.596822
Epoch: 45, Loss: 0.702380
Epoch: 46, Loss: 0.494088
Epoch: 47, Loss: 0.754879
Epoch: 48, Loss: 1.341822
Epoch: 49, Loss: 0.843196
Epoch: 50, Loss: 0.692155
Epoch: 51, Loss: 0.495611
Epoch: 52, Loss: 0.222088
Epoch: 53, Loss: 0.703717
Epoch: 54, Loss: 1.299031
Epoch: 55, Loss: 0.772099
Epoch: 56, Loss: 0.555656
Epoch: 57, Loss: 0.949775
Epoch: 58, Loss: 0.557609
Epoch: 59, Loss: 1.331509
Epoch: 60, Loss: 0.536458
Epoch: 61, Loss: 0.465959
Epoch: 62, Loss: 1.159207
Epoch: 63, Loss: 0.631564
Epoch: 64, Loss: 0.692085
Epoch: 65, Loss: 0.599146
Epoch: 66, Loss: 0.493434
Epoch: 67, Loss: 0.681162
Epoch: 68, Loss: 0.945495
Epoch: 69, Loss: 0.317505
Epoch: 70, Loss: 1.246249
Epoch: 71, Loss: 0.554352
Epoch: 72, Loss: 1.080858
Epoch: 73, Loss: 0.322223
Epoch: 74, Loss: 1.397937
Epoch: 75, Loss: 0.558204
Epoch: 76, Loss: 1.345222
Epoch: 77, Loss: 0.284252
Epoch: 78, Loss: 0.370521
Epoch: 79, Loss: 0.449116
Epoch: 80, Loss: 0.880541
Epoch: 81, Loss: 0.267190
Epoch: 82, Loss: 0.515562
Epoch: 83, Loss: 0.834472
Epoch: 84, Loss: 0.251172
Epoch: 85, Loss: 0.347078
Epoch: 86, Loss: 0.733065
Epoch: 87, Loss: 0.691058
Epoch: 88, Loss: 0.332971
Epoch: 89, Loss: 0.771989
Epoch: 90, Loss: 0.355433
Epoch: 91, Loss: 0.210443
Epoch: 92, Loss: 0.776835
Epoch: 93, Loss: 0.366610
```

```
Epoch: 94, Loss: 0.753169
Epoch: 95, Loss: 0.511624
Epoch: 96, Loss: 0.379635
Epoch: 97, Loss: 0.367830
Epoch: 98, Loss: 0.638768
Epoch: 99, Loss: 1.086178
Epoch: 100, Loss: 0.596224
Epoch: 101, Loss: 1.402197
Epoch: 102, Loss: 0.950927
Epoch: 103, Loss: 0.580121
Epoch: 104, Loss: 0.605557
Epoch: 105, Loss: 0.667210
Epoch: 106, Loss: 0.715795
Epoch: 107, Loss: 0.540268
Epoch: 108, Loss: 0.384047
Epoch: 109, Loss: 0.218978
Epoch: 110, Loss: 0.905232
Epoch: 111, Loss: 0.808130
Epoch: 112, Loss: 0.106361
Epoch: 113, Loss: 0.463803
Epoch: 114, Loss: 1.025647
Epoch: 115, Loss: 0.613296
Epoch: 116, Loss: 0.773259
Epoch: 117, Loss: 0.230441
Epoch: 118, Loss: 0.362173
Epoch: 119, Loss: 0.227840
Epoch: 120, Loss: 0.465431
Epoch: 121, Loss: 1.140233
Epoch: 122, Loss: 0.666230
Epoch: 123, Loss: 0.449102
Epoch: 124, Loss: 1.061858
Epoch: 125, Loss: 0.562275
Epoch: 126, Loss: 1.303998
Epoch: 127, Loss: 0.315353
Epoch: 128, Loss: 0.515697
Epoch: 129, Loss: 0.409228
Epoch: 130, Loss: 0.411534
Epoch: 131, Loss: 1.004170
Epoch: 132, Loss: 0.176068
Epoch: 133, Loss: 0.411302
Epoch: 134, Loss: 0.859111
Epoch: 135, Loss: 1.019657
Epoch: 136, Loss: 0.485963
Epoch: 137, Loss: 0.321783
Epoch: 138, Loss: 0.318216
Epoch: 139, Loss: 0.284407
Epoch: 140, Loss: 0.488205
Epoch: 141, Loss: 1.084039
Epoch: 142, Loss: 0.529898
Epoch: 143, Loss: 0.150954
Epoch: 144, Loss: 0.415855
Epoch: 145, Loss: 0.247873
Epoch: 146, Loss: 0.415892
Epoch: 147, Loss: 0.495871
Epoch: 148, Loss: 0.493417
Epoch: 149, Loss: 0.656050
Epoch: 150, Loss: 1.123717
Epoch: 151, Loss: 0.385549
Epoch: 152, Loss: 0.639023
Epoch: 153, Loss: 1.584278
Epoch: 154, Loss: 0.354603
Epoch: 155, Loss: 0.330090
Epoch: 156, Loss: 0.429722
Epoch: 157, Loss: 0.632479
Epoch: 158, Loss: 0.248089
Epoch: 159, Loss: 0.332424
```

```
Epoch: 160, Loss: 0.794612
Epoch: 161, Loss: 0.615460
Epoch: 162, Loss: 0.419100
Epoch: 163, Loss: 0.269721
Epoch: 164, Loss: 0.525057
Epoch: 165, Loss: 0.285052
Epoch: 166, Loss: 0.779525
Epoch: 167, Loss: 0.204980
Epoch: 168, Loss: 1.227870
Epoch: 169, Loss: 0.718987
Epoch: 170, Loss: 0.558725
Epoch: 171, Loss: 0.328941
Epoch: 172, Loss: 0.057317
Epoch: 173, Loss: 0.287118
Epoch: 174, Loss: 0.533803
Epoch: 175, Loss: 0.427484
Epoch: 176, Loss: 0.524672
Epoch: 177, Loss: 1.293799
Epoch: 178, Loss: 0.753082
Epoch: 179, Loss: 0.210859
Epoch: 180, Loss: 0.190630
Epoch: 181, Loss: 0.659434
Epoch: 182, Loss: 1.065244
Epoch: 183, Loss: 0.738482
Epoch: 184, Loss: 0.649073
Epoch: 185, Loss: 0.554764
Epoch: 186, Loss: 0.253485
Epoch: 187, Loss: 0.287929
Epoch: 188, Loss: 0.709250
Epoch: 189, Loss: 0.530483
Epoch: 190, Loss: 1.068691
Epoch: 191, Loss: 0.288138
Epoch: 192, Loss: 0.522530
Epoch: 193, Loss: 0.754178
Epoch: 194, Loss: 0.144283
Epoch: 195, Loss: 0.854065
Epoch: 196, Loss: 0.343788
Epoch: 197, Loss: 0.407730
Epoch: 198, Loss: 0.659166
Epoch: 199, Loss: 0.984298
Epoch: 200, Loss: 0.055736
Epoch: 201, Loss: 1.379359
Epoch: 202, Loss: 2.160732
Epoch: 203, Loss: 0.166857
Epoch: 204, Loss: 0.645467
Epoch: 205, Loss: 0.663232
Epoch: 206, Loss: 0.363973
Epoch: 207, Loss: 0.454711
Epoch: 208, Loss: 0.470384
Epoch: 209, Loss: 0.649272
Epoch: 210, Loss: 0.259131
Epoch: 211, Loss: 0.682679
Epoch: 212, Loss: 0.198253
Epoch: 213, Loss: 0.265053
Epoch: 214, Loss: 0.846186
Epoch: 215, Loss: 0.492594
Epoch: 216, Loss: 0.508281
Epoch: 217, Loss: 0.109077
Epoch: 218, Loss: 0.461323
Epoch: 219, Loss: 0.880210
Epoch: 220, Loss: 1.335081
Epoch: 221, Loss: 0.448995
Epoch: 222, Loss: 0.827476
Epoch: 223, Loss: 0.520554
Epoch: 224, Loss: 0.243618
Epoch: 225, Loss: 0.280187
```

```
Epoch: 226, Loss: 0.348860
Epoch: 227, Loss: 1.193089
Epoch: 228, Loss: 0.437516
Epoch: 229, Loss: 0.912597
Epoch: 230, Loss: 0.350146
Epoch: 231, Loss: 0.357153
Epoch: 232, Loss: 0.930340
Epoch: 233, Loss: 1.110108
Epoch: 234, Loss: 0.904539
Epoch: 235, Loss: 0.595233
Epoch: 236, Loss: 1.050200
Epoch: 237, Loss: 0.680218
Epoch: 238, Loss: 0.464325
Epoch: 239, Loss: 0.656830
Epoch: 240, Loss: 0.375612
Epoch: 241, Loss: 0.760939
Epoch: 242, Loss: 0.368319
Epoch: 243, Loss: 0.333448
Epoch: 244, Loss: 0.393145
Epoch: 245, Loss: 1.114257
Epoch: 246, Loss: 1.624054
Epoch: 247, Loss: 0.087357
Epoch: 248, Loss: 0.260697
Epoch: 249, Loss: 1.418267
Epoch: 250, Loss: 0.588495
Epoch: 251, Loss: 1.753193
Epoch: 252, Loss: 0.306724
Epoch: 253, Loss: 0.372830
Epoch: 254, Loss: 0.403171
Epoch: 255, Loss: 0.408812
Epoch: 256, Loss: 0.668673
Epoch: 257, Loss: 0.597615
Epoch: 258, Loss: 0.762671
Epoch: 259, Loss: 0.030158
Epoch: 260, Loss: 0.151636
Epoch: 261, Loss: 0.428615
Epoch: 262, Loss: 0.499738
Epoch: 263, Loss: 0.731224
Epoch: 264, Loss: 0.677875
Epoch: 265, Loss: 0.277742
Epoch: 266, Loss: 0.620186
Epoch: 267, Loss: 0.808300
Epoch: 268, Loss: 1.481186
Epoch: 269, Loss: 1.378914
Epoch: 270, Loss: 0.751533
Epoch: 271, Loss: 0.126849
Epoch: 272, Loss: 0.097029
Epoch: 273, Loss: 0.837152
Epoch: 274, Loss: 0.554728
Epoch: 275, Loss: 0.562224
Epoch: 276, Loss: 0.804801
Epoch: 277, Loss: 0.250821
Epoch: 278, Loss: 1.323176
Epoch: 279, Loss: 0.245550
Epoch: 280, Loss: 0.390145
Epoch: 281, Loss: 0.809721
Epoch: 282, Loss: 0.172840
Epoch: 283, Loss: 0.641828
Epoch: 284, Loss: 0.760464
Epoch: 285, Loss: 0.963606
Epoch: 286, Loss: 0.500598
Epoch: 287, Loss: 0.623114
Epoch: 288, Loss: 0.832079
Epoch: 289, Loss: 1.291583
Epoch: 290, Loss: 0.434085
Epoch: 291, Loss: 0.546361
```

```
Epoch: 292, Loss: 0.177244
Epoch: 293, Loss: 0.202407
Epoch: 294, Loss: 0.247989
Epoch: 295, Loss: 0.487519
Epoch: 296, Loss: 0.644584
Epoch: 297, Loss: 0.389295
Epoch: 298, Loss: 0.566370
Epoch: 299, Loss: 0.682870
3296.5056784152985
```

In [56]:

```
from ptflops import get_model_complexity_info


macs, params = get_model_complexity_info(model, ( 3, 32,32), as_strings=True, print_per_la
print('{:<30}  {:<8}'.format('Computational complexity: ', macs))
print('{:<30}  {:<8}'.format('Number of parameters: ', params))
```

```
Computational complexity:       0.02 GMac
Number of parameters:           272.68 k
```

In [57]:

```
val_loader = torch.utils.data.DataLoader(val_transformed_cifar10, batch_size=64, shuffle=F

correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
        imgs = imgs.to(device)
        labels = labels.to(device)
        batch_size=imgs.shape[0]
        outputs = model(imgs)
        _, predicted = torch.max(outputs, dim=1)
#         print(predicted)
#         print("\n")
#         print(labels)
        total += labels.shape[0]
        correct += int((predicted==labels).sum())
    print("Accuracy ", correct/total)
```

```
Accuracy  0.1
```

In [58]:

```
#part 2B

import time
def training(model, optimizer, loss_fn, n_epochs, device, train_loader, l2_lambda):
    start = time.time()

    for epoch in range(n_epochs):
        for imgs, labels in train_loader:
            imgs = imgs.to(device)
            labels = labels.to(device)

            batch_size = imgs.shape[0]
            outputs = model(imgs)
            loss = loss_fn((outputs), labels)
            l2_norm = sum(p.pow(2.0).sum() for p in model.parameters())
            loss = loss + l2_lambda* l2_norm
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
        print("Epoch: %d, Loss: %f" % (epoch, float(loss)))
    end = time.time()
    print(end - start)
```

In [59]:
```
n_epochs = 300
l2_lambda = .001
training(model, optimizer, loss_function, n_epochs, device, train_loader, l2_lambda)
```

```
Epoch: 0, Loss: 10.951440
Epoch: 1, Loss: 4.647394
Epoch: 2, Loss: 2.800955
Epoch: 3, Loss: 2.225659
Epoch: 4, Loss: 2.161125
Epoch: 5, Loss: 1.862766
Epoch: 6, Loss: 1.583517
Epoch: 7, Loss: 2.222004
Epoch: 8, Loss: 1.820010
Epoch: 9, Loss: 1.967250
Epoch: 10, Loss: 2.207386
Epoch: 11, Loss: 1.357456
Epoch: 12, Loss: 1.738413
Epoch: 13, Loss: 1.820579
Epoch: 14, Loss: 1.710661
Epoch: 15, Loss: 1.451937
Epoch: 16, Loss: 1.800370
Epoch: 17, Loss: 1.454486
Epoch: 18, Loss: 1.609495
Epoch: 19, Loss: 1.340552
Epoch: 20, Loss: 1.648403
Epoch: 21, Loss: 1.213099
Epoch: 22, Loss: 1.931675
Epoch: 23, Loss: 1.811163
Epoch: 24, Loss: 1.763996
Epoch: 25, Loss: 1.617107
Epoch: 26, Loss: 1.084703
Epoch: 27, Loss: 1.235137
Epoch: 28, Loss: 1.430830
Epoch: 29, Loss: 2.334033
Epoch: 30, Loss: 1.967194
Epoch: 31, Loss: 1.908156
Epoch: 32, Loss: 1.707417
Epoch: 33, Loss: 1.488826
Epoch: 34, Loss: 1.145470
Epoch: 35, Loss: 1.285830
Epoch: 36, Loss: 1.456136
Epoch: 37, Loss: 1.784907
Epoch: 38, Loss: 1.521228
Epoch: 39, Loss: 1.845839
Epoch: 40, Loss: 1.705061
Epoch: 41, Loss: 1.698319
Epoch: 42, Loss: 1.780094
Epoch: 43, Loss: 1.521727
Epoch: 44, Loss: 1.572895
Epoch: 45, Loss: 1.475441
Epoch: 46, Loss: 1.538025
Epoch: 47, Loss: 1.333474
Epoch: 48, Loss: 1.578037
Epoch: 49, Loss: 1.451524
Epoch: 50, Loss: 1.436068
Epoch: 51, Loss: 0.996346
Epoch: 52, Loss: 1.321064
Epoch: 53, Loss: 1.198278
Epoch: 54, Loss: 1.344646
Epoch: 55, Loss: 1.375412
Epoch: 56, Loss: 1.390983
Epoch: 57, Loss: 1.547436
Epoch: 58, Loss: 1.510586
Epoch: 59, Loss: 1.208568
Epoch: 60, Loss: 1.203075
```

```
Epoch: 61, Loss: 1.943881
Epoch: 62, Loss: 1.362887
Epoch: 63, Loss: 1.667814
Epoch: 64, Loss: 1.431761
Epoch: 65, Loss: 1.401344
Epoch: 66, Loss: 1.655278
Epoch: 67, Loss: 1.185472
Epoch: 68, Loss: 1.493755
Epoch: 69, Loss: 1.803148
Epoch: 70, Loss: 1.430788
Epoch: 71, Loss: 1.498374
Epoch: 72, Loss: 1.565392
Epoch: 73, Loss: 1.876347
Epoch: 74, Loss: 1.605134
Epoch: 75, Loss: 1.304337
Epoch: 76, Loss: 1.166213
Epoch: 77, Loss: 1.801992
Epoch: 78, Loss: 1.346815
Epoch: 79, Loss: 1.754570
Epoch: 80, Loss: 1.262118
Epoch: 81, Loss: 1.594926
Epoch: 82, Loss: 1.379508
Epoch: 83, Loss: 1.905716
Epoch: 84, Loss: 1.546248
Epoch: 85, Loss: 1.469356
Epoch: 86, Loss: 1.541683
Epoch: 87, Loss: 1.653567
Epoch: 88, Loss: 1.279009
Epoch: 89, Loss: 1.289350
Epoch: 90, Loss: 1.771722
Epoch: 91, Loss: 1.597301
Epoch: 92, Loss: 1.254401
Epoch: 93, Loss: 1.243417
Epoch: 94, Loss: 1.182143
Epoch: 95, Loss: 1.399967
Epoch: 96, Loss: 1.461592
Epoch: 97, Loss: 1.601644
Epoch: 98, Loss: 1.570920
Epoch: 99, Loss: 1.374875
Epoch: 100, Loss: 1.627290
Epoch: 101, Loss: 1.276383
Epoch: 102, Loss: 1.752806
Epoch: 103, Loss: 1.550049
Epoch: 104, Loss: 1.343803
Epoch: 105, Loss: 1.420080
Epoch: 106, Loss: 1.926564
Epoch: 107, Loss: 1.481239
Epoch: 108, Loss: 1.833448
Epoch: 109, Loss: 1.469544
Epoch: 110, Loss: 1.259187
Epoch: 111, Loss: 1.662703
Epoch: 112, Loss: 1.338640
Epoch: 113, Loss: 2.187330
Epoch: 114, Loss: 2.158751
Epoch: 115, Loss: 1.472168
Epoch: 116, Loss: 1.559178
Epoch: 117, Loss: 2.076247
Epoch: 118, Loss: 1.676108
Epoch: 119, Loss: 1.110409
Epoch: 120, Loss: 1.289116
Epoch: 121, Loss: 1.369417
Epoch: 122, Loss: 1.616939
Epoch: 123, Loss: 1.760504
Epoch: 124, Loss: 1.690705
Epoch: 125, Loss: 1.244495
Epoch: 126, Loss: 1.286316
```

```
Epoch: 127, Loss: 1.418093
Epoch: 128, Loss: 1.628275
Epoch: 129, Loss: 1.500955
Epoch: 130, Loss: 1.162256
Epoch: 131, Loss: 1.317757
Epoch: 132, Loss: 1.796748
Epoch: 133, Loss: 1.024727
Epoch: 134, Loss: 1.348021
Epoch: 135, Loss: 1.352397
Epoch: 136, Loss: 1.374675
Epoch: 137, Loss: 1.262214
Epoch: 138, Loss: 1.797882
Epoch: 139, Loss: 1.275564
Epoch: 140, Loss: 1.452496
Epoch: 141, Loss: 1.616353
Epoch: 142, Loss: 1.674169
Epoch: 143, Loss: 1.384351
Epoch: 144, Loss: 1.395098
Epoch: 145, Loss: 1.692765
Epoch: 146, Loss: 1.141497
Epoch: 147, Loss: 1.567030
Epoch: 148, Loss: 1.460711
Epoch: 149, Loss: 1.191043
Epoch: 150, Loss: 1.356780
Epoch: 151, Loss: 1.553092
Epoch: 152, Loss: 1.711885
Epoch: 153, Loss: 1.393295
Epoch: 154, Loss: 1.509276
Epoch: 155, Loss: 1.445006
Epoch: 156, Loss: 1.536040
Epoch: 157, Loss: 1.182578
Epoch: 158, Loss: 1.224824
Epoch: 159, Loss: 1.398164
Epoch: 160, Loss: 1.022205
Epoch: 161, Loss: 1.629232
Epoch: 162, Loss: 1.157836
Epoch: 163, Loss: 1.522558
Epoch: 164, Loss: 1.416380
Epoch: 165, Loss: 1.139839
Epoch: 166, Loss: 1.407486
Epoch: 167, Loss: 1.403350
Epoch: 168, Loss: 2.084922
Epoch: 169, Loss: 1.414070
Epoch: 170, Loss: 1.167084
Epoch: 171, Loss: 1.256077
Epoch: 172, Loss: 1.405317
Epoch: 173, Loss: 1.585081
Epoch: 174, Loss: 1.205439
Epoch: 175, Loss: 0.914562
Epoch: 176, Loss: 1.597933
Epoch: 177, Loss: 1.265217
Epoch: 178, Loss: 1.070268
Epoch: 179, Loss: 1.284259
Epoch: 180, Loss: 1.493295
Epoch: 181, Loss: 1.650852
Epoch: 182, Loss: 1.669943
Epoch: 183, Loss: 1.203025
Epoch: 184, Loss: 1.286689
Epoch: 185, Loss: 1.103081
Epoch: 186, Loss: 1.475908
Epoch: 187, Loss: 1.200938
Epoch: 188, Loss: 2.041421
Epoch: 189, Loss: 1.208506
Epoch: 190, Loss: 1.568819
Epoch: 191, Loss: 1.304908
Epoch: 192, Loss: 1.618936
```

```
Epoch: 193, Loss: 1.374373
Epoch: 194, Loss: 1.343892
Epoch: 195, Loss: 0.999015
Epoch: 196, Loss: 1.532312
Epoch: 197, Loss: 1.981481
Epoch: 198, Loss: 1.471740
Epoch: 199, Loss: 1.551695
Epoch: 200, Loss: 1.424155
Epoch: 201, Loss: 1.567466
Epoch: 202, Loss: 1.498433
Epoch: 203, Loss: 1.241199
Epoch: 204, Loss: 1.980607
Epoch: 205, Loss: 1.659364
Epoch: 206, Loss: 1.243863
Epoch: 207, Loss: 1.782366
Epoch: 208, Loss: 0.993867
Epoch: 209, Loss: 1.186438
Epoch: 210, Loss: 1.961484
Epoch: 211, Loss: 1.464996
Epoch: 212, Loss: 1.462134
Epoch: 213, Loss: 1.528311
Epoch: 214, Loss: 1.373449
Epoch: 215, Loss: 1.661284
Epoch: 216, Loss: 1.714193
Epoch: 217, Loss: 1.325982
Epoch: 218, Loss: 1.585445
Epoch: 219, Loss: 1.252668
Epoch: 220, Loss: 1.697926
Epoch: 221, Loss: 1.716879
Epoch: 222, Loss: 1.265929
Epoch: 223, Loss: 1.222662
Epoch: 224, Loss: 1.691228
Epoch: 225, Loss: 1.460888
Epoch: 226, Loss: 1.316936
Epoch: 227, Loss: 1.205209
Epoch: 228, Loss: 1.040003
Epoch: 229, Loss: 1.535553
Epoch: 230, Loss: 2.005908
Epoch: 231, Loss: 1.772653
Epoch: 232, Loss: 1.402008
Epoch: 233, Loss: 1.206209
Epoch: 234, Loss: 1.277411
Epoch: 235, Loss: 1.304767
Epoch: 236, Loss: 1.241860
Epoch: 237, Loss: 1.222328
Epoch: 238, Loss: 1.531687
Epoch: 239, Loss: 1.206513
Epoch: 240, Loss: 1.273515
Epoch: 241, Loss: 1.677120
Epoch: 242, Loss: 1.482058
Epoch: 243, Loss: 1.442981
Epoch: 244, Loss: 1.362632
Epoch: 245, Loss: 1.430273
Epoch: 246, Loss: 1.685722
Epoch: 247, Loss: 1.389421
Epoch: 248, Loss: 1.456757
Epoch: 249, Loss: 1.518691
Epoch: 250, Loss: 1.511637
Epoch: 251, Loss: 1.814562
Epoch: 252, Loss: 1.442732
Epoch: 253, Loss: 1.977664
Epoch: 254, Loss: 1.585663
Epoch: 255, Loss: 1.716352
Epoch: 256, Loss: 1.832749
Epoch: 257, Loss: 0.968053
Epoch: 258, Loss: 2.134270
```

```
Epoch: 259, Loss: 1.772189
Epoch: 260, Loss: 1.343420
Epoch: 261, Loss: 1.490398
Epoch: 262, Loss: 2.132844
Epoch: 263, Loss: 1.196839
Epoch: 264, Loss: 1.784396
Epoch: 265, Loss: 1.505852
Epoch: 266, Loss: 1.551780
Epoch: 267, Loss: 1.474594
Epoch: 268, Loss: 1.533204
Epoch: 269, Loss: 1.299223
Epoch: 270, Loss: 1.888834
Epoch: 271, Loss: 1.288868
Epoch: 272, Loss: 1.426370
Epoch: 273, Loss: 1.314751
Epoch: 274, Loss: 0.965519
Epoch: 275, Loss: 1.622324
Epoch: 276, Loss: 1.654595
Epoch: 277, Loss: 1.545431
Epoch: 278, Loss: 1.257667
Epoch: 279, Loss: 1.529096
Epoch: 280, Loss: 1.988781
Epoch: 281, Loss: 1.947571
Epoch: 282, Loss: 1.140765
Epoch: 283, Loss: 1.313967
Epoch: 284, Loss: 1.178080
Epoch: 285, Loss: 1.361185
Epoch: 286, Loss: 1.345096
Epoch: 287, Loss: 1.243309
Epoch: 288, Loss: 2.127151
Epoch: 289, Loss: 1.609862
Epoch: 290, Loss: 1.359417
Epoch: 291, Loss: 1.526035
Epoch: 292, Loss: 1.822676
Epoch: 293, Loss: 1.628549
Epoch: 294, Loss: 1.414407
Epoch: 295, Loss: 2.173963
Epoch: 296, Loss: 1.359991
Epoch: 297, Loss: 1.186405
Epoch: 298, Loss: 1.215213
Epoch: 299, Loss: 1.284183
3420.2066872119904
```

In [60]:

```
from ptflops import get_model_complexity_info


macs, params = get_model_complexity_info(model, ( 3, 32,32), as_strings=True, print_per_la
print('{:<30}  {:<8}'.format('Computational complexity: ', macs))
print('{:<30}  {:<8}'.format('Number of parameters: ', params))
```

```
Warning: variables __flops__ or __params__ are already defined for the moduleConv2d ptflop
s can affect your code!
Warning: variables __flops__ or __params__ are already defined for the moduleConv2d ptflop
s can affect your code!
Warning: variables __flops__ or __params__ are already defined for the moduleBatchNorm2d p
tflops can affect your code!
Warning: variables __flops__ or __params__ are already defined for the moduleLinear ptflop
s can affect your code!
Warning: variables __flops__ or __params__ are already defined for the moduleLinear ptflop
s can affect your code!
Computational complexity:       0.02 GMac
Number of parameters:           272.68 k
```

In [61]:

```
val_loader = torch.utils.data.DataLoader(val_transformed_cifar10, batch_size=64, shuffle=F
```

```
correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
        imgs = imgs.to(device)
        labels = labels.to(device)
        batch_size=imgs.shape[0]
        outputs = model(imgs)
        _, predicted = torch.max(outputs, dim=1)
#          print(predicted)
#          print("\n")
#          print(labels)
        total += labels.shape[0]
        correct += int((predicted==labels).sum())
    print("Accuracy ", correct/total)
```

```
Accuracy  0.5052
```

In [62]:

```
import torch.nn.functional as F
class ResNet10(nn.Module):
    def __init__(self, n_chans1=32, n_blocks=10):
        super().__init__()
        self.n_chans1 = n_chans1
        self.conv1 = nn.Conv2d(3, n_chans1, kernel_size=3, padding=1)
        self.conv1_dropout = nn.Dropout2d(p=0.3)
        self.ResNetBlocks = nn.Sequential(*(n_blocks * [ResBlock(n_chans=n_chans1)]))
        self.fc1 = nn.Linear(16 * 16 * n_chans1, 32)
        self.fc2 = nn.Linear(32, 10)
    def forward(self, x):
        out = F.max_pool2d(torch.relu(self.conv1(x)), 2)
        out = self.conv1_dropout(out)
        out = self.ResNetBlocks(out)
        out = out.view(-1, 16 * 16 * self.n_chans1)
        out = torch.relu(self.fc1(out))
        out = self.fc2(out)
        return out
```

In [63]:

```
import torch
device = torch.device('cuda:0')
train_loader = torch.utils.data.DataLoader(transformed_cifar10, batch_size=64, shuffle=Tru

model = ResNet10()
model.to(device)
loss_function = nn.CrossEntropyLoss()
learning_rate = 3e-3
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

In [64]:

```
n_epochs = 300
l2_lambda = .001
training(model, optimizer, loss_function, n_epochs, device, train_loader, l2_lambda)
```

```
Epoch: 0, Loss: 2.109835
Epoch: 1, Loss: 1.969299
Epoch: 2, Loss: 1.010406
Epoch: 3, Loss: 1.695881
Epoch: 4, Loss: 1.560346
Epoch: 5, Loss: 1.402911
Epoch: 6, Loss: 0.806718
Epoch: 7, Loss: 1.900599
Epoch: 8, Loss: 1.594776
Epoch: 9, Loss: 1.694569
```

```
Epoch: 10, Loss: 1.355907
Epoch: 11, Loss: 1.411947
Epoch: 12, Loss: 1.280041
Epoch: 13, Loss: 1.848790
Epoch: 14, Loss: 1.218277
Epoch: 15, Loss: 1.878471
Epoch: 16, Loss: 1.468370
Epoch: 17, Loss: 1.517360
Epoch: 18, Loss: 0.981155
Epoch: 19, Loss: 1.447148
Epoch: 20, Loss: 1.244213
Epoch: 21, Loss: 1.141342
Epoch: 22, Loss: 1.386391
Epoch: 23, Loss: 1.805052
Epoch: 24, Loss: 1.588807
Epoch: 25, Loss: 2.014440
Epoch: 26, Loss: 1.912367
Epoch: 27, Loss: 1.130355
Epoch: 28, Loss: 1.580149
Epoch: 29, Loss: 1.610882
Epoch: 30, Loss: 1.176360
Epoch: 31, Loss: 1.310100
Epoch: 32, Loss: 1.648198
Epoch: 33, Loss: 1.367503
Epoch: 34, Loss: 1.422516
Epoch: 35, Loss: 1.680829
Epoch: 36, Loss: 1.376226
Epoch: 37, Loss: 1.313788
Epoch: 38, Loss: 1.669272
Epoch: 39, Loss: 1.532687
Epoch: 40, Loss: 1.132806
Epoch: 41, Loss: 1.431460
Epoch: 42, Loss: 1.331438
Epoch: 43, Loss: 1.178844
Epoch: 44, Loss: 0.870595
Epoch: 45, Loss: 1.118547
Epoch: 46, Loss: 1.479602
Epoch: 47, Loss: 1.420517
Epoch: 48, Loss: 1.183153
Epoch: 49, Loss: 0.880000
Epoch: 50, Loss: 1.091831
Epoch: 51, Loss: 0.771163
Epoch: 52, Loss: 1.255601
Epoch: 53, Loss: 1.547179
Epoch: 54, Loss: 1.380214
Epoch: 55, Loss: 1.583119
Epoch: 56, Loss: 2.063123
Epoch: 57, Loss: 1.509463
Epoch: 58, Loss: 1.347073
Epoch: 59, Loss: 1.602739
Epoch: 60, Loss: 1.518044
Epoch: 61, Loss: 1.236717
Epoch: 62, Loss: 1.634024
Epoch: 63, Loss: 1.215928
Epoch: 64, Loss: 1.501999
Epoch: 65, Loss: 1.213793
Epoch: 66, Loss: 1.092679
Epoch: 67, Loss: 1.476160
Epoch: 68, Loss: 1.120238
Epoch: 69, Loss: 1.734018
Epoch: 70, Loss: 1.145930
Epoch: 71, Loss: 1.509084
Epoch: 72, Loss: 1.183794
Epoch: 73, Loss: 1.915933
Epoch: 74, Loss: 1.407423
Epoch: 75, Loss: 1.564566
```

```
Epoch: 76, Loss: 1.179301
Epoch: 77, Loss: 1.385219
Epoch: 78, Loss: 1.373884
Epoch: 79, Loss: 0.790598
Epoch: 80, Loss: 1.290121
Epoch: 81, Loss: 1.340641
Epoch: 82, Loss: 1.470451
Epoch: 83, Loss: 1.253623
Epoch: 84, Loss: 1.728054
Epoch: 85, Loss: 1.188489
Epoch: 86, Loss: 1.468468
Epoch: 87, Loss: 1.127864
Epoch: 88, Loss: 1.158275
Epoch: 89, Loss: 1.346945
Epoch: 90, Loss: 1.487638
Epoch: 91, Loss: 1.399106
Epoch: 92, Loss: 1.279044
Epoch: 93, Loss: 1.052002
Epoch: 94, Loss: 0.909184
Epoch: 95, Loss: 1.250015
Epoch: 96, Loss: 1.938959
Epoch: 97, Loss: 0.815221
Epoch: 98, Loss: 1.646101
Epoch: 99, Loss: 1.135005
Epoch: 100, Loss: 1.137235
Epoch: 101, Loss: 1.173342
Epoch: 102, Loss: 1.185027
Epoch: 103, Loss: 1.522191
Epoch: 104, Loss: 1.248118
Epoch: 105, Loss: 1.270838
Epoch: 106, Loss: 1.597983
Epoch: 107, Loss: 1.039197
Epoch: 108, Loss: 1.122006
Epoch: 109, Loss: 1.330083
Epoch: 110, Loss: 1.480005
Epoch: 111, Loss: 1.080469
Epoch: 112, Loss: 1.274531
Epoch: 113, Loss: 0.971847
Epoch: 114, Loss: 1.260521
Epoch: 115, Loss: 1.541933
Epoch: 116, Loss: 1.061544
Epoch: 117, Loss: 1.525583
Epoch: 118, Loss: 1.332189
Epoch: 119, Loss: 1.221895
Epoch: 120, Loss: 0.968208
Epoch: 121, Loss: 1.353856
Epoch: 122, Loss: 1.041356
Epoch: 123, Loss: 0.954831
Epoch: 124, Loss: 0.986697
Epoch: 125, Loss: 1.223044
Epoch: 126, Loss: 0.969452
Epoch: 127, Loss: 1.212687
Epoch: 128, Loss: 1.454601
Epoch: 129, Loss: 1.529407
Epoch: 130, Loss: 1.328114
Epoch: 131, Loss: 1.505339
Epoch: 132, Loss: 1.812648
Epoch: 133, Loss: 1.285132
Epoch: 134, Loss: 2.102280
Epoch: 135, Loss: 1.088109
Epoch: 136, Loss: 1.301905
Epoch: 137, Loss: 1.242243
Epoch: 138, Loss: 1.954775
Epoch: 139, Loss: 1.343727
Epoch: 140, Loss: 0.998247
Epoch: 141, Loss: 0.811761
```

```
Epoch: 142, Loss: 1.386384
Epoch: 143, Loss: 1.136151
Epoch: 144, Loss: 1.094962
Epoch: 145, Loss: 1.735127
Epoch: 146, Loss: 1.389385
Epoch: 147, Loss: 1.093279
Epoch: 148, Loss: 1.628156
Epoch: 149, Loss: 1.913773
Epoch: 150, Loss: 0.941347
Epoch: 151, Loss: 1.662392
Epoch: 152, Loss: 1.674738
Epoch: 153, Loss: 1.326700
Epoch: 154, Loss: 1.376500
Epoch: 155, Loss: 1.489647
Epoch: 156, Loss: 0.778387
Epoch: 157, Loss: 1.237298
Epoch: 158, Loss: 1.619611
Epoch: 159, Loss: 1.182442
Epoch: 160, Loss: 1.477567
Epoch: 161, Loss: 1.383962
Epoch: 162, Loss: 1.603646
Epoch: 163, Loss: 1.462832
Epoch: 164, Loss: 0.936300
Epoch: 165, Loss: 1.054512
Epoch: 166, Loss: 1.039273
Epoch: 167, Loss: 1.412014
Epoch: 168, Loss: 1.179628
Epoch: 169, Loss: 1.377108
Epoch: 170, Loss: 1.263451
Epoch: 171, Loss: 1.265061
Epoch: 172, Loss: 0.842233
Epoch: 173, Loss: 1.277047
Epoch: 174, Loss: 1.631402
Epoch: 175, Loss: 1.294199
Epoch: 176, Loss: 1.347221
Epoch: 177, Loss: 0.980425
Epoch: 178, Loss: 1.452324
Epoch: 179, Loss: 1.372367
Epoch: 180, Loss: 1.193663
Epoch: 181, Loss: 1.599889
Epoch: 182, Loss: 1.442709
Epoch: 183, Loss: 1.431270
Epoch: 184, Loss: 1.603526
Epoch: 185, Loss: 1.297375
Epoch: 186, Loss: 1.476580
Epoch: 187, Loss: 0.599081
Epoch: 188, Loss: 1.633477
Epoch: 189, Loss: 1.531051
Epoch: 190, Loss: 1.644556
Epoch: 191, Loss: 1.040234
Epoch: 192, Loss: 1.111907
Epoch: 193, Loss: 1.339648
Epoch: 194, Loss: 1.118119
Epoch: 195, Loss: 1.031549
Epoch: 196, Loss: 0.994083
Epoch: 197, Loss: 1.013652
Epoch: 198, Loss: 1.638794
Epoch: 199, Loss: 0.999518
Epoch: 200, Loss: 1.249751
Epoch: 201, Loss: 1.151665
Epoch: 202, Loss: 1.548531
Epoch: 203, Loss: 1.376494
Epoch: 204, Loss: 1.067082
Epoch: 205, Loss: 1.049309
Epoch: 206, Loss: 1.796296
Epoch: 207, Loss: 1.446187
```

```
Epoch: 208, Loss: 1.177490
Epoch: 209, Loss: 1.031091
Epoch: 210, Loss: 1.066871
Epoch: 211, Loss: 1.884002
Epoch: 212, Loss: 1.110114
Epoch: 213, Loss: 1.568319
Epoch: 214, Loss: 1.074750
Epoch: 215, Loss: 1.125444
Epoch: 216, Loss: 1.228867
Epoch: 217, Loss: 1.397890
Epoch: 218, Loss: 1.406899
Epoch: 219, Loss: 1.507887
Epoch: 220, Loss: 1.587396
Epoch: 221, Loss: 0.914172
Epoch: 222, Loss: 1.340848
Epoch: 223, Loss: 1.132545
Epoch: 224, Loss: 1.208206
Epoch: 225, Loss: 1.443970
Epoch: 226, Loss: 1.755221
Epoch: 227, Loss: 1.952862
Epoch: 228, Loss: 1.550981
Epoch: 229, Loss: 1.101821
Epoch: 230, Loss: 0.943626
Epoch: 231, Loss: 1.144663
Epoch: 232, Loss: 1.095440
Epoch: 233, Loss: 1.313185
Epoch: 234, Loss: 1.364339
Epoch: 235, Loss: 1.315221
Epoch: 236, Loss: 1.641995
Epoch: 237, Loss: 0.912288
Epoch: 238, Loss: 1.330724
Epoch: 239, Loss: 1.219863
Epoch: 240, Loss: 1.508999
Epoch: 241, Loss: 1.090021
Epoch: 242, Loss: 0.967507
Epoch: 243, Loss: 1.563182
Epoch: 244, Loss: 0.896861
Epoch: 245, Loss: 1.356174
Epoch: 246, Loss: 1.422906
Epoch: 247, Loss: 1.248907
Epoch: 248, Loss: 1.101489
Epoch: 249, Loss: 0.914826
Epoch: 250, Loss: 1.269861
Epoch: 251, Loss: 1.207112
Epoch: 252, Loss: 1.068681
Epoch: 253, Loss: 1.244215
Epoch: 254, Loss: 0.994129
Epoch: 255, Loss: 1.568863
Epoch: 256, Loss: 1.565098
Epoch: 257, Loss: 0.835318
Epoch: 258, Loss: 1.050843
Epoch: 259, Loss: 1.115909
Epoch: 260, Loss: 1.590399
Epoch: 261, Loss: 1.007654
Epoch: 262, Loss: 1.444347
Epoch: 263, Loss: 1.025030
Epoch: 264, Loss: 1.234481
Epoch: 265, Loss: 0.971877
Epoch: 266, Loss: 2.059131
Epoch: 267, Loss: 1.163352
Epoch: 268, Loss: 0.956094
Epoch: 269, Loss: 1.457070
Epoch: 270, Loss: 0.743331
Epoch: 271, Loss: 1.233499
Epoch: 272, Loss: 1.804929
Epoch: 273, Loss: 1.001853
```

```
Epoch: 274, Loss: 0.890160
Epoch: 275, Loss: 1.881496
Epoch: 276, Loss: 1.298632
Epoch: 277, Loss: 1.216773
Epoch: 278, Loss: 1.150815
Epoch: 279, Loss: 1.744982
Epoch: 280, Loss: 1.267296
Epoch: 281, Loss: 0.864310
Epoch: 282, Loss: 1.406229
Epoch: 283, Loss: 1.300003
Epoch: 284, Loss: 1.054250
Epoch: 285, Loss: 1.160117
Epoch: 286, Loss: 1.144608
Epoch: 287, Loss: 1.161726
Epoch: 288, Loss: 1.653617
Epoch: 289, Loss: 1.863817
Epoch: 290, Loss: 1.072395
Epoch: 291, Loss: 1.317519
Epoch: 292, Loss: 1.249313
Epoch: 293, Loss: 1.250967
Epoch: 294, Loss: 1.150312
Epoch: 295, Loss: 1.073376
Epoch: 296, Loss: 1.139743
Epoch: 297, Loss: 1.501464
Epoch: 298, Loss: 1.289799
Epoch: 299, Loss: 1.058628
3473.059977054596
```

In [65]:

```
from ptflops import get_model_complexity_info


macs, params = get_model_complexity_info(model, ( 3, 32,32), as_strings=True, print_per_la
print('{:<30}  {:<8}'.format('Computational complexity: ', macs))
print('{:<30}  {:<8}'.format('Number of parameters: ', params))
```

```
Computational complexity:       0.02 GMac
Number of parameters:           272.68 k
```

In [66]:

```
val_loader = torch.utils.data.DataLoader(val_transformed_cifar10, batch_size=64, shuffle=F

correct = 0
total = 0
with torch.no_grad():
    for imgs, labels in val_loader:
        imgs = imgs.to(device)
        labels = labels.to(device)
        batch_size=imgs.shape[0]
        outputs = model(imgs)
        _, predicted = torch.max(outputs, dim=1)
#          print(predicted)
#          print("\n")
#          print(labels)
        total += labels.shape[0]
        correct += int((predicted==labels).sum())
    print("Accuracy ", correct/total)
```

```
Accuracy  0.3615
```

In [ ]:

In [ ]: