Imports

In [1]:

```
import torch
import torchvision
import torchvision.transforms as transforms
from torch.utils.data import Dataset, DataLoader
import torch.nn as nn
import torch.nn.functional as F
import sys
import numpy as np
import os
```

Utilising GPU using Pytorch

In [2]:

```
# cpu-gpu
a = torch.randn((3, 4))
print(a.device)

device = torch.device("cuda")
a = a.to(device)
print(a.device)

# a more generic code
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
cpu
cuda:0
```

In [3]:

```
!nvidia-smi
```

```
Sun Sep 18 08:59:58 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   54C    P0    28W /  70W |    612MiB / 15109MiB |      2%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```

Dataset and Transforms

In [4]:

```
train_transform = transforms.Compose([
  transforms.RandomCrop(32, padding=4),
  transforms.RandomHorizontalFlip(),
  transforms.ToTensor(),
  transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])
test_transform = transforms.Compose([
  transforms.ToTensor(),
  transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])

train_dset = torchvision.datasets.CIFAR10(root="data/", train=True, transform=tr
ain_transform, download=True)
test_dset = torchvision.datasets.CIFAR10(root="data/", train=False, transform=te
st_transform, download=True)
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
to data/cifar-10-python.tar.gz


Extracting data/cifar-10-python.tar.gz to data/
Files already downloaded and verified
```

In [5]:

```
print(f"# of train samples: {len(train_dset)}")
print(f"# of test samples: {len(test_dset)}")
```

```
# of train samples: 50000
# of test samples: 10000
```

In [6]:

```
train_loader = DataLoader(train_dset, batch_size=100, shuffle=True, num_workers=
2)
test_loader = DataLoader(test_dset, batch_size=100, shuffle=False, num_workers=2
)
```

In [7]:

```
print(f"# of train batches: {len(train_loader)}")
print(f"# of test batches: {len(test_loader)}")
```

```
# of train batches: 500
# of test batches: 100
```

In [8]:

```
print("sample i/o sizes")
data = next(iter(train_loader))
img, target = data
print(f"input size: {img.shape}")
print(f"output size: {target.shape}")
```

```
sample i/o sizes
input size: torch.Size([100, 3, 32, 32])
output size: torch.Size([100])
```

LeNet

In [9]:

```
class LeNet(nn.Module):
  def __init__(self):
    super(LeNet, self).__init__()
    self.conv1 = nn.Conv2d(3, 6, kernel_size=5)
    self.conv2 = nn.Conv2d(6, 16, kernel_size=5)
    # TODO: missing input feature size
    self.fc1   = nn.Linear(16*5*5, 120)
    self.fc2   = nn.Linear(120, 84)
    # TODO: missing output feature size
    self.fc3   = nn.Linear(84, 10)  ##10 CLASSES
    self.activ = nn.ReLU()

  # TODO: add maxpool operation of given kernel size
  # https://pytorch.org/docs/stable/nn.functional.html
  def pool(self, x, kernel_size=2):
    out = F.max_pool2d(x, kernel_size=2)
    return out

  def forward(self, x):
    out = self.activ(self.conv1(x))
    out = self.pool(out)
    out = self.activ(self.conv2(out))
    out = self.pool(out)

    # TODO: flatten
    out = out.view(out.size(0),-1) ##OR We can do out.view(out.size(0),-1)
    out = self.activ(self.fc1(out))
    out = self.activ(self.fc2(out))
    out = self.fc3(out)
    return out
```

VGG

In [17]:

```
class VGG(nn.Module):
  CONFIGS = {
      "vgg11": [64, "pool", 128, "pool", 256, 256, "pool", 512, 512, "pool", 512
, 512, "pool"],
      "vgg13": [64, 64, "pool", 128, 128, "pool", 256, 256, "pool", 512, 512, "p
ool", 512, 512, "pool"],
      "vgg16": [64, 64, "pool", 128, 128, "pool", 256, 256, 256, "pool", 512, 51
2, 512, "pool", 512, 512, 512, "pool"],
      "vgg19": [64, 64, "pool", 128, 128, "pool", 256, 256, 256, 256, "pool", 51
2, 512, 512, 512, "pool", 512, 512, 512, 512, "pool"],
  }
  def __init__(self, cfg):
    super(VGG, self).__init__()
    # TODO: missing input dimension
    in_dim = 3
    layers = []
    for layer in self.CONFIGS[cfg]:
        if layer == "pool":
            # TODO: add maxpool module of given kernel size, stride (here 2 eac
h)
            # https://pytorch.org/docs/stable/nn.html
            maxpool = nn.MaxPool2d(kernel_size=2, stride=2)
            layers.append(maxpool)
        else:
            # TODO: add sequential module consisting of convolution (kernel size
= 3, padding = 1), batchnorm, relu
            # https://pytorch.org/docs/stable/generated/torch.nn.Sequential.htm
l?highlight=sequential#torch.nn.Sequential
            block = nn.Sequential(
          nn.Conv2d(in_dim,layer,kernel_size = 3, padding = 1),
          nn.BatchNorm2d(layer),
          nn.ReLU()
        )
            layers.append(block)
            in_dim = layer
    # TODO: add average pool to collapse spatial dimensions
    #avgpool = F.lp_pool2d(input, norm_type, kernel_size=2) """difference"""
    avgpool = nn.AvgPool2d(kernel_size=1, stride=1)
    layers.append(avgpool)
    self.layers = nn.Sequential(*layers)
    # TODO: missing output features
    self.fc = nn.Linear(512, 10) ##10 CLASSES

  def forward(self, x):
    out = self.layers(x)
    # TODO: flatten
    out = out.view(out.size(0),-1)
    out = self.fc(out)
    return out
```

ResNet

In [11]:

```
class BasicBlock(nn.Module):
  expansion = 1

  def __init__(self, in_dim, dim, stride=1):
    super(BasicBlock, self).__init__()
    self.conv1 = nn.Conv2d(in_dim, dim, kernel_size=3, stride=stride, padding=1,
bias=False)
    self.bn1 = nn.BatchNorm2d(dim)
    self.conv2 = nn.Conv2d(dim, dim, kernel_size=3, stride=1, padding=1, bias=Fa
lse)
    self.bn2 = nn.BatchNorm2d(dim)
    self.activ = nn.ReLU()

    self.shortcut = nn.Identity()
    # TODO: missing condition for parameterized shortcut connection (hint: when
 input and output dimensions don't match - both spatial, feature)
    if (stride != 1 or in_dim != self.expansion*dim):
        # TODO: add sequential module consisting of 1x1 convolution (given strid
e, bias=False), batchnorm
        self.shortcut = nn.Sequential(
                nn.Conv2d(in_dim, self.expansion*dim, kernel_size=1, stride=stri
de),
                nn.BatchNorm2d(self.expansion*dim)
            )

  def forward(self, x):
    out = self.activ(self.bn1(self.conv1(x)))
    out = self.bn2(self.conv2(out))
    # TODO: missing residual connection
    out = out + self.shortcut(x)
    out = self.activ(out)
    return out


class Bottleneck(nn.Module):
  expansion = 4

  def __init__(self, in_dim, dim, stride=1):
    super(Bottleneck, self).__init__()
    self.conv1 = nn.Conv2d(in_dim, dim, kernel_size=1, bias=False)
    self.bn1 = nn.BatchNorm2d(dim)
    self.conv2 = nn.Conv2d(dim, dim, kernel_size=3, stride=stride, padding=1, bi
as=False)
    self.bn2 = nn.BatchNorm2d(dim)
    self.conv3 = nn.Conv2d(dim, self.expansion * dim, kernel_size=1, bias=False)
    self.bn3 = nn.BatchNorm2d(self.expansion*dim)
    self.activ = nn.ReLU()

    self.shortcut = nn.Identity()
    # TODO: missing condition for parameterized shortcut connection (hint: when
 input and output dimensions don't match - both spatial, feature)
    if (stride != 1 or in_dim != self.expansion*dim):
        # TODO: add sequential module consisting of 1x1 convolution (given strid
e, bias=False), batchnorm
        self.shortcut = nn.Sequential(
                nn.Conv2d(in_dim, self.expansion*dim, kernel_size=1, stride=stri
de),
                nn.BatchNorm2d(self.expansion*dim)
            )
```

```
  def forward(self, x):
    out = self.activ(self.bn1(self.conv1(x)))
    out = self.activ(self.bn2(self.conv2(out)))
    out = self.bn3(self.conv3(out))
    # TODO: missing residual connection
    out = out + self.shortcut(x)
    out = self.activ(out)
    return out


class ResNet(nn.Module):
  CONFIGS = {
      "resnet18": (BasicBlock, [2, 2, 2, 2]),
      "resnet34": (BasicBlock, [3, 4, 6, 3]),
      "resnet50": (Bottleneck, [3, 4, 6, 3]),
      "resnet101": (Bottleneck, [3, 4, 23, 3]),
      "resnet152": (Bottleneck, [3, 8, 36, 3]),
  }
  def __init__(self, cfg):
    super(ResNet, self).__init__()
    block, num_blocks = self.CONFIGS[cfg]
    self.in_dim = 64
    self.conv1 = nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1, bias=False
)
    self.bn1 = nn.BatchNorm2d(64)
    self.layer1 = self._make_layer(block, 64, num_blocks[0], stride=1)
    self.layer2 = self._make_layer(block, 128, num_blocks[1], stride=2)
    self.layer3 = self._make_layer(block, 256, num_blocks[2], stride=2)
    self.layer4 = self._make_layer(block, 512, num_blocks[3], stride=2)
    self.activ = nn.ReLU()
    # TODO: missing output features
    self.linear = nn.Linear(512*block.expansion,10)

  def _make_layer(self, block, dim, num_blocks, stride):
    strides = [stride] + [1]*(num_blocks-1)
    layers = []
    for stride in strides:
        # TODO: create layers within block
        layer = block(self.in_dim, dim, stride)
        layers.append(layer)
        # TODO: update in_dim based on block output size
        self.in_dim = dim * block.expansion
    return nn.Sequential(*layers)

  def forward(self, x):
    out = self.activ(self.bn1(self.conv1(x)))
    out = self.layer1(out)
    out = self.layer2(out)
    out = self.layer3(out)
    out = self.layer4(out)
    # TODO: average pool and flatten
    # pooling = nn.AvgPool2d(4)
    out = F.avg_pool2d(out,4)
    out = out.view(out.size(0),-1)
    out = self.linear(out)
    return out
```

Utility functions (can ignore)

In [12]:

```
def pbar(p=0, msg="", bar_len=20):
    sys.stdout.write("\033[K")
    sys.stdout.write("\x1b[2K" + "\r")
    block = int(round(bar_len * p))
    text = "Progress: [{}] {}% {}".format(
        "\x1b[32m" + "=" * (block - 1) + ">" + "\033[0m" + "-" * (bar_len - bloc
k),
        round(p * 100, 2),
        msg,
    )
    print(text, end="\r")
    if p == 1:
        print()


class AvgMeter:
    def __init__(self):
        self.reset()

    def reset(self):
        self.metrics = {}

    def add(self, batch_metrics):
        if self.metrics == {}:
            for key, value in batch_metrics.items():
                self.metrics[key] = [value]
        else:
            for key, value in batch_metrics.items():
                self.metrics[key].append(value)

    def get(self):
        return {key: np.mean(value) for key, value in self.metrics.items()}

    def msg(self):
        avg_metrics = {key: np.mean(value) for key, value in self.metrics.items
()}
        return "".join(["[{}] {:.5f} ".format(key, value) for key, value in avg_
metrics.items()])
```

Training

In [13]:

```
def train(model, optim, lr_sched=None, epochs=200, device=torch.device("cuda" if
torch.cuda.is_available() else "cpu"), criterion=None, metric_meter=None, out_di
r="out/"):
  model.to(device)
  best_acc = 0
  for epoch in range(epochs):
    model.train()
    metric_meter.reset()
    for indx, (img, target) in enumerate(train_loader):
      # TODO: send to device (cpu or gpu)
      img = img.to(device)
      target = target.to(device)

      # TODO: missing forward pass
      out = model(img)
      loss = criterion(out, target)
      # TODO: missing backward, parameter update
      optim.zero_grad()
      loss.backward()
      optim.step()

      metric_meter.add({"train loss": loss.item()})
      pbar(indx / len(train_loader), msg=metric_meter.msg())
    pbar(1, msg=metric_meter.msg())

    model.eval()
    metric_meter.reset()
    for indx, (img, target) in enumerate(test_loader):
      # TODO: send to device (cpu or gpu)
      img = img.to(device)
      target = target.to(device)

      # TODO: missing forward pass
      out = model(img)
      loss = criterion(out, target)
      # TODO: compute accuracy
      classes = torch.argmax(out, dim=1)
      acc_t = torch.mean((classes == target).float())
      acc=acc_t.cpu().detach().numpy()

      metric_meter.add({"test loss": loss.item(), "test acc": acc})
      pbar(indx / len(test_loader), msg=metric_meter.msg())
    pbar(1, msg=metric_meter.msg())

    test_metrics = metric_meter.get()
    if test_metrics["test acc"] > best_acc:
      print(
          "\x1b[33m"
          + f"test acc improved from {round(best_acc, 5)} to {round(test_metrics
['test acc'], 5)}"
          + "\033[0m"
      )
      best_acc = test_metrics['test acc']
      torch.save(model.state_dict(), os.path.join(out_dir, "best.ckpt"))
    lr_sched.step()
```

Run Experiments

In [14]:

```
def run_experiment(model_name="lenet", model_cfg=None, epochs=200):
  if model_name == "lenet":
    model = LeNet()
  elif model_name == "vgg":
    model = VGG(model_cfg)
  elif model_name == "resnet":
    model = ResNet(model_cfg)
  else:
    raise NotImplementedError()
  optim = torch.optim.SGD(model.parameters(), lr=1e-1, momentum=0.9, weight_deca
y=5e-4)
  lr_sched = torch.optim.lr_scheduler.CosineAnnealingLR(optim, T_max=epochs)
  criterion = nn.CrossEntropyLoss()
  metric_meter = AvgMeter()
  out_dir = f"{model_name}_{model_cfg}"
  os.makedirs(out_dir, exist_ok=True)
  train(model, optim, lr_sched, epochs=epochs, criterion=criterion, metric_meter
=metric_meter, out_dir=out_dir)
```

In [18]:

```
# run_experiment(model_name="lenet")
run_experiment(model_name="vgg",model_cfg="vgg16")
run_experiment(model_name="resnet",model_cfg="resnet18") ##Last 6 entries are of
resnet
```

```
Progress: [====================>] 100% [train loss] 2.36074
Progress: [====================>] 100% [test loss] 1.93528 [test acc]
0.22790
test acc improved from 0 to 0.22789999842643738
Progress: [====================>] 100% [train loss] 1.87238
Progress: [====================>] 100% [test loss] 1.73560 [test acc]
0.32020
test acc improved from 0.22789999842643738 to 0.32019999623298645
Progress: [====================>] 100% [train loss] 1.68946
Progress: [====================>] 100% [test loss] 1.57096 [test acc]
0.39430
test acc improved from 0.32019999623298645 to 0.39430001378059387
Progress: [====================>] 100% [train loss] 1.43035
Progress: [====================>] 100% [test loss] 1.34073 [test acc]
0.53620
test acc improved from 0.39430001378059387 to 0.5361999869346619
Progress: [====================>] 100% [train loss] 1.10646
Progress: [====================>] 100% [test loss] 1.24309 [test acc]
0.59480
test acc improved from 0.5361999869346619 to 0.5947999954223633
Progress: [====================>] 100% [train loss] 0.92909
Progress: [====================>] 100% [test loss] 1.02593 [test acc]
0.64930
test acc improved from 0.5947999954223633 to 0.6492999792098999
Progress: [====================>] 100% [train loss] 0.83323
Progress: [====================>] 100% [test loss] 1.12235 [test acc]
0.62200
Progress: [====================>] 100% [train loss] 0.79281
Progress: [====================>] 100% [test loss] 1.08844 [test acc]
0.64560
Progress: [====================>] 100% [train loss] 0.74353
Progress: [====================>] 100% [test loss] 0.83379 [test acc]
0.72750
test acc improved from 0.6492999792098999 to 0.7275000214576721
Progress: [====================>] 100% [train loss] 0.69805
Progress: [====================>] 100% [test loss] 1.04592 [test acc]
0.68350
Progress: [====================>] 100% [train loss] 0.67252
Progress: [====================>] 100% [test loss] 0.76194 [test acc]
0.75390
test acc improved from 0.7275000214576721 to 0.7538999915122986
Progress: [====================>] 100% [train loss] 0.64506
Progress: [====================>] 100% [test loss] 0.80785 [test acc]
0.73930
Progress: [====================>] 100% [train loss] 0.63206
Progress: [====================>] 100% [test loss] 0.73495 [test acc]
0.75830
test acc improved from 0.7538999915122986 to 0.7583000063896179
Progress: [====================>] 100% [train loss] 0.61424
Progress: [====================>] 100% [test loss] 0.75104 [test acc]
0.76070
test acc improved from 0.7583000063896179 to 0.760699987411499
Progress: [====================>] 100% [train loss] 0.59594
Progress: [====================>] 100% [test loss] 0.85933 [test acc]
0.72140
Progress: [====================>] 100% [train loss] 0.59034
Progress: [====================>] 100% [test loss] 0.74713 [test acc]
0.76390
test acc improved from 0.760699987411499 to 0.7638999819755554
Progress: [====================>] 100% [train loss] 0.57288
Progress: [====================>] 100% [test loss] 0.85982 [test acc]
```

```
0.73480
Progress: [====================>] 100% [train loss] 0.56566
Progress: [====================>] 100% [test loss] 0.64291 [test acc]
0.79150
test acc improved from 0.7638999819755554 to 0.7914999723434448
Progress: [====================>] 100% [train loss] 0.55636
Progress: [====================>] 100% [test loss] 0.83199 [test acc]
0.73350
Progress: [====================>] 100% [train loss] 0.54265
Progress: [====================>] 100% [test loss] 0.60783 [test acc]
0.80690
test acc improved from 0.7914999723434448 to 0.8069000244140625
Progress: [====================>] 100% [train loss] 0.53106
Progress: [====================>] 100% [test loss] 0.96370 [test acc]
0.69400
Progress: [====================>] 100% [train loss] 0.53020
Progress: [====================>] 100% [test loss] 0.67735 [test acc]
0.77830
Progress: [====================>] 100% [train loss] 0.52457
Progress: [====================>] 100% [test loss] 0.66913 [test acc]
0.78050
Progress: [====================>] 100% [train loss] 0.51410
Progress: [====================>] 100% [test loss] 0.59660 [test acc]
0.80230
Progress: [====================>] 100% [train loss] 0.50515
Progress: [====================>] 100% [test loss] 0.55662 [test acc]
0.81250
test acc improved from 0.8069000244140625 to 0.8125
Progress: [====================>] 100% [train loss] 0.51052
Progress: [====================>] 100% [test loss] 1.00276 [test acc]
0.69430
Progress: [====================>] 100% [train loss] 0.50234
Progress: [====================>] 100% [test loss] 0.64009 [test acc]
0.78520
Progress: [====================>] 100% [train loss] 0.49401
Progress: [====================>] 100% [test loss] 0.62338 [test acc]
0.79320
Progress: [====================>] 100% [train loss] 0.49651
Progress: [====================>] 100% [test loss] 0.63325 [test acc]
0.79770
Progress: [====================>] 100% [train loss] 0.49037
Progress: [====================>] 100% [test loss] 0.64087 [test acc]
0.78870
Progress: [====================>] 100% [train loss] 0.47975
Progress: [====================>] 100% [test loss] 0.63229 [test acc]
0.78650
Progress: [====================>] 100% [train loss] 0.47788
Progress: [====================>] 100% [test loss] 0.61555 [test acc]
0.79650
Progress: [====================>] 100% [train loss] 0.47488
Progress: [====================>] 100% [test loss] 0.92747 [test acc]
0.71240
Progress: [====================>] 100% [train loss] 0.47781
Progress: [====================>] 100% [test loss] 0.66246 [test acc]
0.77960
Progress: [====================>] 100% [train loss] 0.47490
Progress: [====================>] 100% [test loss] 0.55400 [test acc]
0.82010
test acc improved from 0.8125 to 0.8201000094413757
Progress: [====================>] 100% [train loss] 0.46972
Progress: [====================>] 100% [test loss] 0.65669 [test acc]
```

```
0.79260
Progress: [====================>] 100% [train loss] 0.46356
Progress: [====================>] 100% [test loss] 0.69271 [test acc]
0.78860
Progress: [====================>] 100% [train loss] 0.45414
Progress: [====================>] 100% [test loss] 0.54622 [test acc]
0.81820
Progress: [====================>] 100% [train loss] 0.45491
Progress: [====================>] 100% [test loss] 0.75468 [test acc]
0.76250
Progress: [====================>] 100% [train loss] 0.45930
Progress: [====================>] 100% [test loss] 0.71958 [test acc]
0.77310
Progress: [====================>] 100% [train loss] 0.45670
Progress: [====================>] 100% [test loss] 0.54065 [test acc]
0.82480
test acc improved from 0.8201000094413757 to 0.8248000144958496
Progress: [====================>] 100% [train loss] 0.45149
Progress: [====================>] 100% [test loss] 0.66545 [test acc]
0.79240
Progress: [====================>] 100% [train loss] 0.43991
Progress: [====================>] 100% [test loss] 0.60977 [test acc]
0.80120
Progress: [====================>] 100% [train loss] 0.44922
Progress: [====================>] 100% [test loss] 0.47960 [test acc]
0.83740
test acc improved from 0.8248000144958496 to 0.8374000191688538
Progress: [====================>] 100% [train loss] 0.44315
Progress: [====================>] 100% [test loss] 0.58409 [test acc]
0.80980
Progress: [====================>] 100% [train loss] 0.44463
Progress: [====================>] 100% [test loss] 0.75218 [test acc]
0.75770
Progress: [====================>] 100% [train loss] 0.43753
Progress: [====================>] 100% [test loss] 0.80535 [test acc]
0.74530
Progress: [====================>] 100% [train loss] 0.43741
Progress: [====================>] 100% [test loss] 0.61064 [test acc]
0.79930
Progress: [====================>] 100% [train loss] 0.43466
Progress: [====================>] 100% [test loss] 0.65468 [test acc]
0.78380
Progress: [====================>] 100% [train loss] 0.42496
Progress: [====================>] 100% [test loss] 0.53417 [test acc]
0.82250
Progress: [====================>] 100% [train loss] 0.42737
Progress: [====================>] 100% [test loss] 0.55928 [test acc]
0.81750
Progress: [====================>] 100% [train loss] 0.42744
Progress: [====================>] 100% [test loss] 0.71934 [test acc]
0.78540
Progress: [====================>] 100% [train loss] 0.42534
Progress: [====================>] 100% [test loss] 0.80829 [test acc]
0.74720
Progress: [====================>] 100% [train loss] 0.41781
Progress: [====================>] 100% [test loss] 0.57098 [test acc]
0.81850
Progress: [====================>] 100% [train loss] 0.41724
Progress: [====================>] 100% [test loss] 0.63991 [test acc]
0.79650
Progress: [====================>] 100% [train loss] 0.41982
```

```
Progress: [====================>] 100% [test loss] 0.77760 [test acc]
0.77020
Progress: [====================>] 100% [train loss] 0.41405
Progress: [====================>] 100% [test loss] 0.56651 [test acc]
0.81110
Progress: [====================>] 100% [train loss] 0.41363
Progress: [====================>] 100% [test loss] 0.46697 [test acc]
0.84590
test acc improved from 0.8374000191688538 to 0.8458999991416931
Progress: [====================>] 100% [train loss] 0.40676
Progress: [====================>] 100% [test loss] 0.53698 [test acc]
0.82650
Progress: [====================>] 100% [train loss] 0.40453
Progress: [====================>] 100% [test loss] 0.60523 [test acc]
0.80180
Progress: [====================>] 100% [train loss] 0.40910
Progress: [====================>] 100% [test loss] 0.69067 [test acc]
0.78300
Progress: [====================>] 100% [train loss] 0.39841
Progress: [====================>] 100% [test loss] 0.55237 [test acc]
0.81490
Progress: [====================>] 100% [train loss] 0.40132
Progress: [====================>] 100% [test loss] 0.68076 [test acc]
0.77770
Progress: [====================>] 100% [train loss] 0.39072
Progress: [====================>] 100% [test loss] 0.64541 [test acc]
0.80600
Progress: [====================>] 100% [train loss] 0.39358
Progress: [====================>] 100% [test loss] 0.66190 [test acc]
0.79290
Progress: [====================>] 100% [train loss] 0.38615
Progress: [====================>] 100% [test loss] 0.44727 [test acc]
0.85430
test acc improved from 0.8458999991416931 to 0.8543000221252441
Progress: [====================>] 100% [train loss] 0.38295
Progress: [====================>] 100% [test loss] 0.55130 [test acc]
0.81920
Progress: [====================>] 100% [train loss] 0.37606
Progress: [====================>] 100% [test loss] 0.60192 [test acc]
0.81270
Progress: [====================>] 100% [train loss] 0.38925
Progress: [====================>] 100% [test loss] 0.61588 [test acc]
0.80460
Progress: [====================>] 100% [train loss] 0.38153
Progress: [====================>] 100% [test loss] 0.74176 [test acc]
0.75860
Progress: [====================>] 100% [train loss] 0.37310
Progress: [====================>] 100% [test loss] 0.62272 [test acc]
0.80010
Progress: [====================>] 100% [train loss] 0.37784
Progress: [====================>] 100% [test loss] 0.60447 [test acc]
0.81150
Progress: [====================>] 100% [train loss] 0.37197
Progress: [====================>] 100% [test loss] 0.72424 [test acc]
0.76980
Progress: [====================>] 100% [train loss] 0.37219
Progress: [====================>] 100% [test loss] 0.79391 [test acc]
0.74970
Progress: [====================>] 100% [train loss] 0.36254
Progress: [====================>] 100% [test loss] 0.52206 [test acc]
0.83780
```

```
Progress: [====================>] 100% [train loss] 0.36199
Progress: [====================>] 100% [test loss] 0.45824 [test acc]
0.85200
Progress: [====================>] 100% [train loss] 0.36322
Progress: [====================>] 100% [test loss] 0.47469 [test acc]
0.83670
Progress: [====================>] 100% [train loss] 0.35975
Progress: [====================>] 100% [test loss] 0.54857 [test acc]
0.82170
Progress: [====================>] 100% [train loss] 0.35254
Progress: [====================>] 100% [test loss] 0.50116 [test acc]
0.83940
Progress: [====================>] 100% [train loss] 0.34594
Progress: [====================>] 100% [test loss] 0.51809 [test acc]
0.83170
Progress: [====================>] 100% [train loss] 0.35079
Progress: [====================>] 100% [test loss] 0.69369 [test acc]
0.78100
Progress: [====================>] 100% [train loss] 0.34939
Progress: [====================>] 100% [test loss] 0.47162 [test acc]
0.84610
Progress: [====================>] 100% [train loss] 0.34685
Progress: [====================>] 100% [test loss] 0.55572 [test acc]
0.81650
Progress: [====================>] 100% [train loss] 0.33749
Progress: [====================>] 100% [test loss] 0.60079 [test acc]
0.80170
Progress: [====================>] 100% [train loss] 0.33364
Progress: [====================>] 100% [test loss] 0.70597 [test acc]
0.77830
Progress: [====================>] 100% [train loss] 0.32925
Progress: [====================>] 100% [test loss] 0.47452 [test acc]
0.84600
Progress: [====================>] 100% [train loss] 0.33164
Progress: [====================>] 100% [test loss] 0.54601 [test acc]
0.83440
Progress: [====================>] 100% [train loss] 0.32758
Progress: [====================>] 100% [test loss] 0.45359 [test acc]
0.85150
Progress: [====================>] 100% [train loss] 0.31870
Progress: [====================>] 100% [test loss] 0.50159 [test acc]
0.83890
Progress: [====================>] 100% [train loss] 0.31965
Progress: [====================>] 100% [test loss] 0.56337 [test acc]
0.82210
Progress: [====================>] 100% [train loss] 0.32132
Progress: [====================>] 100% [test loss] 0.51222 [test acc]
0.83220
Progress: [====================>] 100% [train loss] 0.31571
Progress: [====================>] 100% [test loss] 0.59733 [test acc]
0.81930
Progress: [====================>] 100% [train loss] 0.31629
Progress: [====================>] 100% [test loss] 0.44948 [test acc]
0.85300
Progress: [====================>] 100% [train loss] 0.31256
Progress: [====================>] 100% [test loss] 0.43186 [test acc]
0.86080
test acc improved from 0.8543000221252441 to 0.86080002784729
Progress: [====================>] 100% [train loss] 0.30431
Progress: [====================>] 100% [test loss] 0.57545 [test acc]
0.82180
```

```
Progress: [====================>] 100% [train loss] 0.30482
Progress: [====================>] 100% [test loss] 0.48021 [test acc]
0.84500
Progress: [====================>] 100% [train loss] 0.29908
Progress: [====================>] 100% [test loss] 0.57887 [test acc]
0.82460
Progress: [====================>] 100% [train loss] 0.29896
Progress: [====================>] 100% [test loss] 0.43505 [test acc]
0.85750
Progress: [====================>] 100% [train loss] 0.29822
Progress: [====================>] 100% [test loss] 0.46274 [test acc]
0.85530
Progress: [====================>] 100% [train loss] 0.28974
Progress: [====================>] 100% [test loss] 0.49225 [test acc]
0.84690
Progress: [====================>] 100% [train loss] 0.28631
Progress: [====================>] 100% [test loss] 0.60463 [test acc]
0.81680
Progress: [====================>] 100% [train loss] 0.28626
Progress: [====================>] 100% [test loss] 0.48925 [test acc]
0.84290
Progress: [====================>] 100% [train loss] 0.27877
Progress: [====================>] 100% [test loss] 0.48399 [test acc]
0.84900
Progress: [====================>] 100% [train loss] 0.27935
Progress: [====================>] 100% [test loss] 0.38503 [test acc]
0.87070
test acc improved from 0.86080002784729 to 0.8707000017166138
Progress: [====================>] 100% [train loss] 0.26592
Progress: [====================>] 100% [test loss] 0.44920 [test acc]
0.86080
Progress: [====================>] 100% [train loss] 0.26227
Progress: [====================>] 100% [test loss] 0.42438 [test acc]
0.86260
Progress: [====================>] 100% [train loss] 0.27118
Progress: [====================>] 100% [test loss] 0.43679 [test acc]
0.86000
Progress: [====================>] 100% [train loss] 0.25990
Progress: [====================>] 100% [test loss] 0.42443 [test acc]
0.86250
Progress: [====================>] 100% [train loss] 0.25943
Progress: [====================>] 100% [test loss] 0.38018 [test acc]
0.87780
test acc improved from 0.8707000017166138 to 0.8777999877929688
Progress: [====================>] 100% [train loss] 0.25721
Progress: [====================>] 100% [test loss] 0.37584 [test acc]
0.87680
Progress: [====================>] 100% [train loss] 0.24725
Progress: [====================>] 100% [test loss] 0.40425 [test acc]
0.87300
Progress: [====================>] 100% [train loss] 0.24965
Progress: [====================>] 100% [test loss] 0.45026 [test acc]
0.85830
Progress: [====================>] 100% [train loss] 0.24404
Progress: [====================>] 100% [test loss] 0.36777 [test acc]
0.88220
test acc improved from 0.8777999877929688 to 0.8822000026702881
Progress: [====================>] 100% [train loss] 0.24058
Progress: [====================>] 100% [test loss] 0.41754 [test acc]
0.87100
Progress: [====================>] 100% [train loss] 0.23583
```

```
Progress: [====================>] 100% [test loss] 0.37524 [test acc]
0.87540
Progress: [====================>] 100% [train loss] 0.23202
Progress: [====================>] 100% [test loss] 0.51649 [test acc]
0.84240
Progress: [====================>] 100% [train loss] 0.22606
Progress: [====================>] 100% [test loss] 0.42929 [test acc]
0.86380
Progress: [====================>] 100% [train loss] 0.22864
Progress: [====================>] 100% [test loss] 0.49422 [test acc]
0.84720
Progress: [====================>] 100% [train loss] 0.23215
Progress: [====================>] 100% [test loss] 0.48078 [test acc]
0.85130
Progress: [====================>] 100% [train loss] 0.21730
Progress: [====================>] 100% [test loss] 0.48413 [test acc]
0.84520
Progress: [====================>] 100% [train loss] 0.21672
Progress: [====================>] 100% [test loss] 0.46294 [test acc]
0.85650
Progress: [====================>] 100% [train loss] 0.20781
Progress: [====================>] 100% [test loss] 0.39781 [test acc]
0.87420
Progress: [====================>] 100% [train loss] 0.20782
Progress: [====================>] 100% [test loss] 0.35576 [test acc]
0.88300
test acc improved from 0.8822000026702881 to 0.8830000162124634
Progress: [====================>] 100% [train loss] 0.20623
Progress: [====================>] 100% [test loss] 0.38947 [test acc]
0.87590
Progress: [====================>] 100% [train loss] 0.19255
Progress: [====================>] 100% [test loss] 0.38529 [test acc]
0.87720
Progress: [====================>] 100% [train loss] 0.19255
Progress: [====================>] 100% [test loss] 0.37757 [test acc]
0.87820
Progress: [====================>] 100% [train loss] 0.19096
Progress: [====================>] 100% [test loss] 0.41555 [test acc]
0.87310
Progress: [====================>] 100% [train loss] 0.18593
Progress: [====================>] 100% [test loss] 0.38630 [test acc]
0.87840
Progress: [====================>] 100% [train loss] 0.18273
Progress: [====================>] 100% [test loss] 0.35761 [test acc]
0.88900
test acc improved from 0.8830000162124634 to 0.8889999985694885
Progress: [====================>] 100% [train loss] 0.18155
Progress: [====================>] 100% [test loss] 0.35824 [test acc]
0.88560
Progress: [====================>] 100% [train loss] 0.17161
Progress: [====================>] 100% [test loss] 0.36684 [test acc]
0.88320
Progress: [====================>] 100% [train loss] 0.16573
Progress: [====================>] 100% [test loss] 0.35868 [test acc]
0.88630
Progress: [====================>] 100% [train loss] 0.16715
Progress: [====================>] 100% [test loss] 0.35142 [test acc]
0.88980
test acc improved from 0.8889999985694885 to 0.8898000121116638
Progress: [====================>] 100% [train loss] 0.16725
Progress: [====================>] 100% [test loss] 0.39336 [test acc]
```

```
0.87800
Progress: [====================>] 100% [train loss] 0.16074
Progress: [====================>] 100% [test loss] 0.31068 [test acc]
0.89770
test acc improved from 0.8898000121116638 to 0.8977000117301941
Progress: [====================>] 100% [train loss] 0.15557
Progress: [====================>] 100% [test loss] 0.40433 [test acc]
0.87840
Progress: [====================>] 100% [train loss] 0.15124
Progress: [====================>] 100% [test loss] 0.36454 [test acc]
0.89060
Progress: [====================>] 100% [train loss] 0.14549
Progress: [====================>] 100% [test loss] 0.39763 [test acc]
0.87760
Progress: [====================>] 100% [train loss] 0.14500
Progress: [====================>] 100% [test loss] 0.31713 [test acc]
0.90210
test acc improved from 0.8977000117301941 to 0.9021000266075134
Progress: [====================>] 100% [train loss] 0.13810
Progress: [====================>] 100% [test loss] 0.35882 [test acc]
0.88910
Progress: [====================>] 100% [train loss] 0.12998
Progress: [====================>] 100% [test loss] 0.41305 [test acc]
0.88260
Progress: [====================>] 100% [train loss] 0.13323
Progress: [====================>] 100% [test loss] 0.34622 [test acc]
0.89440
Progress: [====================>] 100% [train loss] 0.12565
Progress: [====================>] 100% [test loss] 0.34903 [test acc]
0.89330
Progress: [====================>] 100% [train loss] 0.12372
Progress: [====================>] 100% [test loss] 0.36460 [test acc]
0.89360
Progress: [====================>] 100% [train loss] 0.11262
Progress: [====================>] 100% [test loss] 0.37418 [test acc]
0.89210
Progress: [====================>] 100% [train loss] 0.11276
Progress: [====================>] 100% [test loss] 0.32965 [test acc]
0.90050
Progress: [====================>] 100% [train loss] 0.10611
Progress: [====================>] 100% [test loss] 0.31819 [test acc]
0.90490
test acc improved from 0.9021000266075134 to 0.9049000144004822
Progress: [====================>] 100% [train loss] 0.10206
Progress: [====================>] 100% [test loss] 0.33906 [test acc]
0.90530
test acc improved from 0.9049000144004822 to 0.9053000211715698
Progress: [====================>] 100% [train loss] 0.10386
Progress: [====================>] 100% [test loss] 0.45076 [test acc]
0.87660
Progress: [====================>] 100% [train loss] 0.09681
Progress: [====================>] 100% [test loss] 0.30854 [test acc]
0.91000
test acc improved from 0.9053000211715698 to 0.9100000262260437
Progress: [====================>] 100% [train loss] 0.09501
Progress: [====================>] 100% [test loss] 0.31087 [test acc]
0.90890
Progress: [====================>] 100% [train loss] 0.08845
Progress: [====================>] 100% [test loss] 0.35339 [test acc]
0.90280
Progress: [====================>] 100% [train loss] 0.08180
```

```
Progress: [====================>] 100% [test loss] 0.37437 [test acc]
0.89550
Progress: [====================>] 100% [train loss] 0.07976
Progress: [====================>] 100% [test loss] 0.44190 [test acc]
0.87920
Progress: [====================>] 100% [train loss] 0.07640
Progress: [====================>] 100% [test loss] 0.33868 [test acc]
0.90690
Progress: [====================>] 100% [train loss] 0.07378
Progress: [====================>] 100% [test loss] 0.32000 [test acc]
0.91060
test acc improved from 0.9100000262260437 to 0.9106000065803528
Progress: [====================>] 100% [train loss] 0.07086
Progress: [====================>] 100% [test loss] 0.32519 [test acc]
0.90980
Progress: [====================>] 100% [train loss] 0.06407
Progress: [====================>] 100% [test loss] 0.32337 [test acc]
0.90980
Progress: [====================>] 100% [train loss] 0.06326
Progress: [====================>] 100% [test loss] 0.29734 [test acc]
0.91380
test acc improved from 0.9106000065803528 to 0.9138000011444092
Progress: [====================>] 100% [train loss] 0.05973
Progress: [====================>] 100% [test loss] 0.31796 [test acc]
0.91630
test acc improved from 0.9138000011444092 to 0.9162999987602234
Progress: [====================>] 100% [train loss] 0.05587
Progress: [====================>] 100% [test loss] 0.32664 [test acc]
0.91060
Progress: [====================>] 100% [train loss] 0.04872
Progress: [====================>] 100% [test loss] 0.31626 [test acc]
0.91670
test acc improved from 0.9162999987602234 to 0.916700005531311
Progress: [====================>] 100% [train loss] 0.05017
Progress: [====================>] 100% [test loss] 0.32131 [test acc]
0.91540
Progress: [====================>] 100% [train loss] 0.04309
Progress: [====================>] 100% [test loss] 0.33786 [test acc]
0.91150
Progress: [====================>] 100% [train loss] 0.04482
Progress: [====================>] 100% [test loss] 0.30225 [test acc]
0.92130
test acc improved from 0.916700005531311 to 0.9212999939918518
Progress: [====================>] 100% [train loss] 0.03578
Progress: [====================>] 100% [test loss] 0.35043 [test acc]
0.91110
Progress: [====================>] 100% [train loss] 0.03433
Progress: [====================>] 100% [test loss] 0.30142 [test acc]
0.92080
Progress: [====================>] 100% [train loss] 0.03005
Progress: [====================>] 100% [test loss] 0.28928 [test acc]
0.92560
test acc improved from 0.9212999939918518 to 0.925599992275238
Progress: [====================>] 100% [train loss] 0.02764
Progress: [====================>] 100% [test loss] 0.29491 [test acc]
0.92660
test acc improved from 0.925599992275238 to 0.9265999794006348
Progress: [====================>] 100% [train loss] 0.02770
Progress: [====================>] 100% [test loss] 0.30249 [test acc]
0.92290
Progress: [====================>] 100% [train loss] 0.02256
```

```
Progress: [====================>] 100% [test loss] 0.30050 [test acc]
0.92540
Progress: [====================>] 100% [train loss] 0.01922
Progress: [====================>] 100% [test loss] 0.31272 [test acc]
0.92320
Progress: [====================>] 100% [train loss] 0.01818
Progress: [====================>] 100% [test loss] 0.28663 [test acc]
0.93130
test acc improved from 0.9265999794006348 to 0.9312999844551086
Progress: [====================>] 100% [train loss] 0.01567
Progress: [====================>] 100% [test loss] 0.30474 [test acc]
0.92750
Progress: [====================>] 100% [train loss] 0.01605
Progress: [====================>] 100% [test loss] 0.30057 [test acc]
0.92850
Progress: [====================>] 100% [train loss] 0.01226
Progress: [====================>] 100% [test loss] 0.29947 [test acc]
0.92910
Progress: [====================>] 100% [train loss] 0.00955
Progress: [====================>] 100% [test loss] 0.31190 [test acc]
0.93000
Progress: [====================>] 100% [train loss] 0.00867
Progress: [====================>] 100% [test loss] 0.30479 [test acc]
0.93030
Progress: [====================>] 100% [train loss] 0.00772
Progress: [====================>] 100% [test loss] 0.31223 [test acc]
0.93390
test acc improved from 0.9312999844551086 to 0.933899998664856
Progress: [====================>] 100% [train loss] 0.00604
Progress: [====================>] 100% [test loss] 0.30094 [test acc]
0.93550
test acc improved from 0.933899998664856 to 0.9355000257492065
Progress: [====================>] 100% [train loss] 0.00490
Progress: [====================>] 100% [test loss] 0.30571 [test acc]
0.93220
Progress: [====================>] 100% [train loss] 0.00508
Progress: [====================>] 100% [test loss] 0.29654 [test acc]
0.93760
test acc improved from 0.9355000257492065 to 0.9376000165939331
Progress: [====================>] 100% [train loss] 0.00481
Progress: [====================>] 100% [test loss] 0.29844 [test acc]
0.93660
Progress: [====================>] 100% [train loss] 0.00406
Progress: [====================>] 100% [test loss] 0.29741 [test acc]
0.93550
Progress: [====================>] 100% [train loss] 0.00295
Progress: [====================>] 100% [test loss] 0.29540 [test acc]
0.93640
Progress: [====================>] 100% [train loss] 0.00330
Progress: [====================>] 100% [test loss] 0.28955 [test acc]
0.93630
Progress: [====================>] 100% [train loss] 0.00320
Progress: [====================>] 100% [test loss] 0.29400 [test acc]
0.93760
Progress: [====================>] 100% [train loss] 0.00267
Progress: [====================>] 100% [test loss] 0.29150 [test acc]
0.93720
Progress: [====================>] 100% [train loss] 0.00252
Progress: [====================>] 100% [test loss] 0.29331 [test acc]
0.93860
test acc improved from 0.9376000165939331 to 0.9386000037193298
```

```
Progress: [====================>] 100% [train loss] 0.00210
Progress: [====================>] 100% [test loss] 0.29087 [test acc]
0.93890
test acc improved from 0.9386000037193298 to 0.9388999938964844
Progress: [====================>] 100% [train loss] 0.00228
Progress: [====================>] 100% [test loss] 0.28642 [test acc]
0.93830
Progress: [====================>] 100% [train loss] 0.00213
Progress: [====================>] 100% [test loss] 0.28671 [test acc]
0.93850
Progress: [====================>] 100% [train loss] 0.00203
Progress: [====================>] 100% [test loss] 0.28702 [test acc]
0.93880
Progress: [====================>] 100% [train loss] 0.00183
Progress: [====================>] 100% [test loss] 0.28619 [test acc]
0.93840
Progress: [====================>] 100% [train loss] 0.00179
Progress: [====================>] 100% [test loss] 0.28447 [test acc]
0.93900
test acc improved from 0.9388999938964844 to 0.9390000104904175
Progress: [====================>] 100% [train loss] 0.00197
Progress: [====================>] 100% [test loss] 0.28720 [test acc]
0.93830
Progress: [====================>] 100% [train loss] 0.00182
Progress: [====================>] 100% [test loss] 0.28461 [test acc]
0.93870
Progress: [====================>] 100% [train loss] 0.00145
Progress: [====================>] 100% [test loss] 0.28566 [test acc]
0.93920
test acc improved from 0.9390000104904175 to 0.9391999840736389
Progress: [====================>] 100% [train loss] 0.00190
Progress: [====================>] 100% [test loss] 0.28506 [test acc]
0.93870
Progress: [====================>] 100% [train loss] 0.00188
Progress: [====================>] 100% [test loss] 0.28582 [test acc]
0.93870
```

VGGNet

```
Progress: [====================>] 100% [train loss] 1.99895
Progress: [====================>] 100% [test loss] 1.56277 [test acc]
0.42020
test acc improved from 0 to 0.420199990272522
Progress: [====================>] 100% [train loss] 1.48508
Progress: [====================>] 100% [test loss] 1.37254 [test acc]
0.49390
test acc improved from 0.420199990272522 to 0.49390000104904175
```

```
---------------------------------------------------------------------
-------
KeyboardInterrupt                         Traceback (most recent cal
l last)
<ipython-input-18-3a585c35ebf3> in <module>
      1 # run_experiment(model_name="lenet")
      2 run_experiment(model_name="vgg",model_cfg="vgg16")
----> 3 run_experiment(model_name="resnet",model_cfg="resnet18")

<ipython-input-14-0d8cd2d0ba80> in run_experiment(model_name, model_
cfg, epochs)
     14   out_dir = f"{model_name}_{model_cfg}"
     15   os.makedirs(out_dir, exist_ok=True)
---> 16   train(model, optim, lr_sched, epochs=epochs, criterion=cri
terion, metric_meter=metric_meter, out_dir=out_dir)

<ipython-input-13-99c4e1e223f0> in train(model, optim, lr_sched, epo
chs, device, criterion, metric_meter, out_dir)
     18       optim.step()
     19
---> 20       metric_meter.add({"train loss": loss.item()})
     21       pbar(indx / len(train_loader), msg=metric_meter.msg())
     22     pbar(1, msg=metric_meter.msg())

KeyboardInterrupt:
```