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 Un
corr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  C
ompute 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                  G
PU Memory |
|        ID   ID                                                   U
sage      |
|=============================================================================
==========|
+-----------------------------------------------------------------------------
----------+
```

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 [10]:

```
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 [15]:

```
run_experiment(model_name="lenet")
run_experiment(model_name="vgg",model_cfg="vgg16")
run_experiment(model_name="resnet",model_cfg="resnet18")
```

```
Progress: [====================>] 100% [train loss] 2.05892
Progress: [====================>] 100% [test loss] 1.89343 [test acc]
0.28240
test acc improved from 0 to 0.2824000120162964
Progress: [====================>] 100% [train loss] 1.91568
Progress: [====================>] 100% [test loss] 1.87559 [test acc]
0.28090
Progress: [====================>] 100% [train loss] 1.90090
Progress: [====================>] 100% [test loss] 1.85571 [test acc]
0.26190
Progress: [====================>] 100% [train loss] 1.86789
Progress: [====================>] 100% [test loss] 1.91435 [test acc]
0.28530
test acc improved from 0.2824000120162964 to 0.28529998660087585
Progress: [====================>] 100% [train loss] 1.86541
Progress: [====================>] 100% [test loss] 1.81164 [test acc]
0.31840
test acc improved from 0.28529998660087585 to 0.31839999556541443
Progress: [====================>] 100% [train loss] 1.85255
Progress: [====================>] 100% [test loss] 1.75928 [test acc]
0.33490
test acc improved from 0.31839999556541443 to 0.33489999175071716
Progress: [====================>] 100% [train loss] 1.82484
Progress: [====================>] 100% [test loss] 1.73692 [test acc]
0.36360
test acc improved from 0.33489999175071716 to 0.3635999858379364
Progress: [====================>] 100% [train loss] 1.83350
Progress: [====================>] 100% [test loss] 1.80799 [test acc]
0.32810
Progress: [====================>] 100% [train loss] 1.82900
Progress: [====================>] 100% [test loss] 1.70090 [test acc]
0.35850
Progress: [====================>] 100% [train loss] 1.80538
Progress: [====================>] 100% [test loss] 1.69859 [test acc]
0.38040
test acc improved from 0.3635999858379364 to 0.38040000200271606
Progress: [====================>] 100% [train loss] 1.80941
Progress: [====================>] 100% [test loss] 1.74209 [test acc]
0.34890
Progress: [====================>] 100% [train loss] 1.80695
Progress: [====================>] 100% [test loss] 1.76213 [test acc]
0.34230
Progress: [====================>] 100% [train loss] 1.79239
Progress: [====================>] 100% [test loss] 1.65901 [test acc]
0.39150
test acc improved from 0.38040000200271606 to 0.39149999618530273
Progress: [====================>] 100% [train loss] 1.80244
Progress: [====================>] 100% [test loss] 1.70791 [test acc]
0.36140
Progress: [====================>] 100% [train loss] 1.80360
Progress: [====================>] 100% [test loss] 1.65222 [test acc]
0.39640
test acc improved from 0.39149999618530273 to 0.39640000462532043
Progress: [====================>] 100% [train loss] 1.79699
Progress: [====================>] 100% [test loss] 1.78248 [test acc]
0.33110
Progress: [====================>] 100% [train loss] 1.76489
Progress: [====================>] 100% [test loss] 1.71936 [test acc]
0.37370
Progress: [====================>] 100% [train loss] 1.75925
Progress: [====================>] 100% [test loss] 1.71337 [test acc]
```

```
0.37390
Progress: [====================>] 100% [train loss] 1.76143
Progress: [====================>] 100% [test loss] 1.64131 [test acc]
0.37950
Progress: [====================>] 100% [train loss] 1.77533
Progress: [====================>] 100% [test loss] 1.69804 [test acc]
0.37570
Progress: [====================>] 100% [train loss] 1.77281
Progress: [====================>] 100% [test loss] 1.67897 [test acc]
0.37940
Progress: [====================>] 100% [train loss] 1.77340
Progress: [====================>] 100% [test loss] 1.64549 [test acc]
0.39810
test acc improved from 0.39640000462532043 to 0.39809998869895935
Progress: [====================>] 100% [train loss] 1.76340
Progress: [====================>] 100% [test loss] 1.63413 [test acc]
0.39310
Progress: [====================>] 100% [train loss] 1.74679
Progress: [====================>] 100% [test loss] 1.69400 [test acc]
0.39340
Progress: [====================>] 100% [train loss] 1.74443
Progress: [====================>] 100% [test loss] 1.70349 [test acc]
0.38630
Progress: [====================>] 100% [train loss] 1.73466
Progress: [====================>] 100% [test loss] 1.60272 [test acc]
0.42000
test acc improved from 0.39809998869895935 to 0.41999998688697815
Progress: [====================>] 100% [train loss] 1.73571
Progress: [====================>] 100% [test loss] 1.59391 [test acc]
0.42720
test acc improved from 0.41999998688697815 to 0.42719998955726624
Progress: [====================>] 100% [train loss] 1.72224
Progress: [====================>] 100% [test loss] 1.62424 [test acc]
0.41240
Progress: [====================>] 100% [train loss] 1.73026
Progress: [====================>] 100% [test loss] 1.64642 [test acc]
0.39700
Progress: [====================>] 100% [train loss] 1.73157
Progress: [====================>] 100% [test loss] 1.66879 [test acc]
0.38280
Progress: [====================>] 100% [train loss] 1.72325
Progress: [====================>] 100% [test loss] 1.68796 [test acc]
0.39100
Progress: [====================>] 100% [train loss] 1.73695
Progress: [====================>] 100% [test loss] 1.65546 [test acc]
0.40140
Progress: [====================>] 100% [train loss] 1.72891
Progress: [====================>] 100% [test loss] 1.63130 [test acc]
0.37780
Progress: [====================>] 100% [train loss] 1.70969
Progress: [====================>] 100% [test loss] 1.64474 [test acc]
0.38440
Progress: [====================>] 100% [train loss] 1.69703
Progress: [====================>] 100% [test loss] 1.66651 [test acc]
0.39570
Progress: [====================>] 100% [train loss] 1.71385
Progress: [====================>] 100% [test loss] 1.86380 [test acc]
0.35140
Progress: [====================>] 100% [train loss] 1.71266
Progress: [====================>] 100% [test loss] 1.69020 [test acc]
0.39100
```

```
Progress: [====================>] 100% [train loss] 1.70406
Progress: [====================>] 100% [test loss] 1.61599 [test acc]
0.40820
Progress: [====================>] 100% [train loss] 1.71529
Progress: [====================>] 100% [test loss] 1.67400 [test acc]
0.38260
Progress: [====================>] 100% [train loss] 1.68100
Progress: [====================>] 100% [test loss] 1.66185 [test acc]
0.39600
Progress: [====================>] 100% [train loss] 1.69857
Progress: [====================>] 100% [test loss] 1.66545 [test acc]
0.39530
Progress: [====================>] 100% [train loss] 1.67632
Progress: [====================>] 100% [test loss] 1.62683 [test acc]
0.40500
Progress: [====================>] 100% [train loss] 1.67521
Progress: [====================>] 100% [test loss] 1.62419 [test acc]
0.42410
Progress: [====================>] 100% [train loss] 1.66212
Progress: [====================>] 100% [test loss] 1.56780 [test acc]
0.43490
test acc improved from 0.42719998955726624 to 0.4348999857902527
Progress: [====================>] 100% [train loss] 1.67713
Progress: [====================>] 100% [test loss] 1.52603 [test acc]
0.44810
test acc improved from 0.4348999857902527 to 0.4481000006198883
Progress: [====================>] 100% [train loss] 1.68116
Progress: [====================>] 100% [test loss] 1.59115 [test acc]
0.44190
Progress: [====================>] 100% [train loss] 1.68547
Progress: [====================>] 100% [test loss] 1.58456 [test acc]
0.42820
Progress: [====================>] 100% [train loss] 1.66500
Progress: [====================>] 100% [test loss] 1.59398 [test acc]
0.42460
Progress: [====================>] 100% [train loss] 1.67832
Progress: [====================>] 100% [test loss] 1.53644 [test acc]
0.45160
test acc improved from 0.4481000006198883 to 0.45159998536109924
Progress: [====================>] 100% [train loss] 1.66458
Progress: [====================>] 100% [test loss] 1.52033 [test acc]
0.46350
test acc improved from 0.45159998536109924 to 0.4634999930858612
Progress: [====================>] 100% [train loss] 1.65687
Progress: [====================>] 100% [test loss] 1.65384 [test acc]
0.40440
Progress: [====================>] 100% [train loss] 1.65790
Progress: [====================>] 100% [test loss] 1.63487 [test acc]
0.42680
Progress: [====================>] 100% [train loss] 1.63622
Progress: [====================>] 100% [test loss] 1.53909 [test acc]
0.43580
Progress: [====================>] 100% [train loss] 1.63308
Progress: [====================>] 100% [test loss] 1.55011 [test acc]
0.44290
Progress: [====================>] 100% [train loss] 1.64432
Progress: [====================>] 100% [test loss] 1.50457 [test acc]
0.45320
Progress: [====================>] 100% [train loss] 1.64082
Progress: [====================>] 100% [test loss] 1.53467 [test acc]
0.44990
```

```
Progress: [====================>] 100% [train loss] 1.62246
Progress: [====================>] 100% [test loss] 1.59203 [test acc]
0.41850
Progress: [====================>] 100% [train loss] 1.62340
Progress: [====================>] 100% [test loss] 1.52067 [test acc]
0.45090
Progress: [====================>] 100% [train loss] 1.63326
Progress: [====================>] 100% [test loss] 1.67111 [test acc]
0.41880
Progress: [====================>] 100% [train loss] 1.61057
Progress: [====================>] 100% [test loss] 1.53929 [test acc]
0.45470
Progress: [====================>] 100% [train loss] 1.60288
Progress: [====================>] 100% [test loss] 1.61193 [test acc]
0.43140
Progress: [====================>] 100% [train loss] 1.62835
Progress: [====================>] 100% [test loss] 1.63933 [test acc]
0.42780
Progress: [====================>] 100% [train loss] 1.62287
Progress: [====================>] 100% [test loss] 1.48743 [test acc]
0.47460
test acc improved from 0.4634999930858612 to 0.4745999872684479
Progress: [====================>] 100% [train loss] 1.59911
Progress: [====================>] 100% [test loss] 1.53145 [test acc]
0.44750
Progress: [====================>] 100% [train loss] 1.59970
Progress: [====================>] 100% [test loss] 1.54008 [test acc]
0.45770
Progress: [====================>] 100% [train loss] 1.60175
Progress: [====================>] 100% [test loss] 1.53160 [test acc]
0.44270
Progress: [====================>] 100% [train loss] 1.60843
Progress: [====================>] 100% [test loss] 1.46069 [test acc]
0.47410
Progress: [====================>] 100% [train loss] 1.58844
Progress: [====================>] 100% [test loss] 1.45188 [test acc]
0.48690
test acc improved from 0.4745999872684479 to 0.4869000017642975
Progress: [====================>] 100% [train loss] 1.59179
Progress: [====================>] 100% [test loss] 1.56721 [test acc]
0.46780
Progress: [====================>] 100% [train loss] 1.57762
Progress: [====================>] 100% [test loss] 1.48965 [test acc]
0.46700
Progress: [====================>] 100% [train loss] 1.58772
Progress: [====================>] 100% [test loss] 1.51657 [test acc]
0.45660
Progress: [====================>] 100% [train loss] 1.58202
Progress: [====================>] 100% [test loss] 1.43750 [test acc]
0.49070
test acc improved from 0.4869000017642975 to 0.49070000648498535
Progress: [====================>] 100% [train loss] 1.56597
Progress: [====================>] 100% [test loss] 1.50265 [test acc]
0.47720
Progress: [====================>] 100% [train loss] 1.57003
Progress: [====================>] 100% [test loss] 1.46314 [test acc]
0.48910
Progress: [====================>] 100% [train loss] 1.57484
Progress: [====================>] 100% [test loss] 1.51094 [test acc]
0.45890
Progress: [====================>] 100% [train loss] 1.56657
```

```
Progress: [====================>] 100% [test loss] 1.43764 [test acc]
0.49280
test acc improved from 0.49070000648498535 to 0.4927999973297119
Progress: [====================>] 100% [train loss] 1.56641
Progress: [====================>] 100% [test loss] 1.47923 [test acc]
0.47820
Progress: [====================>] 100% [train loss] 1.55452
Progress: [====================>] 100% [test loss] 1.45320 [test acc]
0.47860
Progress: [====================>] 100% [train loss] 1.55549
Progress: [====================>] 100% [test loss] 1.55178 [test acc]
0.46380
Progress: [====================>] 100% [train loss] 1.54035
Progress: [====================>] 100% [test loss] 1.46051 [test acc]
0.48510
Progress: [====================>] 100% [train loss] 1.52888
Progress: [====================>] 100% [test loss] 1.47509 [test acc]
0.47950
Progress: [====================>] 100% [train loss] 1.52817
Progress: [====================>] 100% [test loss] 1.51626 [test acc]
0.46130
Progress: [====================>] 100% [train loss] 1.53813
Progress: [====================>] 100% [test loss] 1.46743 [test acc]
0.45830
Progress: [====================>] 100% [train loss] 1.52668
Progress: [====================>] 100% [test loss] 1.46797 [test acc]
0.46710
Progress: [====================>] 100% [train loss] 1.52022
Progress: [====================>] 100% [test loss] 1.48119 [test acc]
0.47210
Progress: [====================>] 100% [train loss] 1.50961
Progress: [====================>] 100% [test loss] 1.40074 [test acc]
0.50050
test acc improved from 0.4927999973297119 to 0.5005000233650208
Progress: [====================>] 100% [train loss] 1.51692
Progress: [====================>] 100% [test loss] 1.53131 [test acc]
0.45210
Progress: [====================>] 100% [train loss] 1.49947
Progress: [====================>] 100% [test loss] 1.51801 [test acc]
0.46730
Progress: [====================>] 100% [train loss] 1.49665
Progress: [====================>] 100% [test loss] 1.45766 [test acc]
0.48550
Progress: [====================>] 100% [train loss] 1.48703
Progress: [====================>] 100% [test loss] 1.40369 [test acc]
0.49470
Progress: [====================>] 100% [train loss] 1.48952
Progress: [====================>] 100% [test loss] 1.45512 [test acc]
0.49260
Progress: [====================>] 100% [train loss] 1.46952
Progress: [====================>] 100% [test loss] 1.43528 [test acc]
0.48800
Progress: [====================>] 100% [train loss] 1.47376
Progress: [====================>] 100% [test loss] 1.35199 [test acc]
0.52510
test acc improved from 0.5005000233650208 to 0.5250999927520752
Progress: [====================>] 100% [train loss] 1.46161
Progress: [====================>] 100% [test loss] 1.50190 [test acc]
0.46650
Progress: [====================>] 100% [train loss] 1.46767
Progress: [====================>] 100% [test loss] 1.34419 [test acc]
```

```
0.53250
test acc improved from 0.5250999927520752 to 0.5325000286102295
Progress: [====================>] 100% [train loss] 1.43663
Progress: [====================>] 100% [test loss] 1.39795 [test acc]
0.50250
Progress: [====================>] 100% [train loss] 1.45605
Progress: [====================>] 100% [test loss] 1.33838 [test acc]
0.53040
Progress: [====================>] 100% [train loss] 1.43675
Progress: [====================>] 100% [test loss] 1.33263 [test acc]
0.53780
test acc improved from 0.5325000286102295 to 0.5378000140190125
Progress: [====================>] 100% [train loss] 1.43168
Progress: [====================>] 100% [test loss] 1.41656 [test acc]
0.50630
Progress: [====================>] 100% [train loss] 1.42232
Progress: [====================>] 100% [test loss] 1.47138 [test acc]
0.49910
Progress: [====================>] 100% [train loss] 1.41444
Progress: [====================>] 100% [test loss] 1.43215 [test acc]
0.49930
Progress: [====================>] 100% [train loss] 1.40749
Progress: [====================>] 100% [test loss] 1.28730 [test acc]
0.54170
test acc improved from 0.5378000140190125 to 0.541700005531311
Progress: [====================>] 100% [train loss] 1.38981
Progress: [====================>] 100% [test loss] 1.38513 [test acc]
0.51360
Progress: [====================>] 100% [train loss] 1.39431
Progress: [====================>] 100% [test loss] 1.32692 [test acc]
0.52610
Progress: [====================>] 100% [train loss] 1.38964
Progress: [====================>] 100% [test loss] 1.27094 [test acc]
0.55600
test acc improved from 0.541700005531311 to 0.5559999942779541
Progress: [====================>] 100% [train loss] 1.38959
Progress: [====================>] 100% [test loss] 1.33606 [test acc]
0.53160
Progress: [====================>] 100% [train loss] 1.37088
Progress: [====================>] 100% [test loss] 1.26485 [test acc]
0.55520
Progress: [====================>] 100% [train loss] 1.37658
Progress: [====================>] 100% [test loss] 1.30781 [test acc]
0.55400
Progress: [====================>] 100% [train loss] 1.35585
Progress: [====================>] 100% [test loss] 1.30151 [test acc]
0.54970
Progress: [====================>] 100% [train loss] 1.36629
Progress: [====================>] 100% [test loss] 1.27020 [test acc]
0.56020
test acc improved from 0.5559999942779541 to 0.5601999759674072
Progress: [====================>] 100% [train loss] 1.34943
Progress: [====================>] 100% [test loss] 1.23935 [test acc]
0.56520
test acc improved from 0.5601999759674072 to 0.5651999711990356
Progress: [====================>] 100% [train loss] 1.35024
Progress: [====================>] 100% [test loss] 1.22203 [test acc]
0.56610
test acc improved from 0.5651999711990356 to 0.566100001335144
Progress: [====================>] 100% [train loss] 1.34370
Progress: [====================>] 100% [test loss] 1.23398 [test acc]
```

```
0.57090
test acc improved from 0.566100001335144 to 0.570900022983551
Progress: [====================>] 100% [train loss] 1.34038
Progress: [====================>] 100% [test loss] 1.27838 [test acc]
0.55200
Progress: [====================>] 100% [train loss] 1.32455
Progress: [====================>] 100% [test loss] 1.19224 [test acc]
0.59370
test acc improved from 0.570900022983551 to 0.5936999917030334
Progress: [====================>] 100% [train loss] 1.31361
Progress: [====================>] 100% [test loss] 1.20793 [test acc]
0.59170
Progress: [====================>] 100% [train loss] 1.29297
Progress: [====================>] 100% [test loss] 1.23746 [test acc]
0.57440
Progress: [====================>] 100% [train loss] 1.30066
Progress: [====================>] 100% [test loss] 1.18255 [test acc]
0.58750
Progress: [====================>] 100% [train loss] 1.29737
Progress: [====================>] 100% [test loss] 1.19221 [test acc]
0.58890
Progress: [====================>] 100% [train loss] 1.27934
Progress: [====================>] 100% [test loss] 1.17876 [test acc]
0.59650
test acc improved from 0.5936999917030334 to 0.5964999794960022
Progress: [====================>] 100% [train loss] 1.27740
Progress: [====================>] 100% [test loss] 1.21710 [test acc]
0.57650
Progress: [====================>] 100% [train loss] 1.26568
Progress: [====================>] 100% [test loss] 1.19322 [test acc]
0.58320
Progress: [====================>] 100% [train loss] 1.26339
Progress: [====================>] 100% [test loss] 1.22020 [test acc]
0.57600
Progress: [====================>] 100% [train loss] 1.25872
Progress: [====================>] 100% [test loss] 1.16937 [test acc]
0.59460
Progress: [====================>] 100% [train loss] 1.26743
Progress: [====================>] 100% [test loss] 1.17575 [test acc]
0.58920
Progress: [====================>] 100% [train loss] 1.24012
Progress: [====================>] 100% [test loss] 1.14183 [test acc]
0.61010
test acc improved from 0.5964999794960022 to 0.6100999712944031
Progress: [====================>] 100% [train loss] 1.22818
Progress: [====================>] 100% [test loss] 1.17511 [test acc]
0.59030
Progress: [====================>] 100% [train loss] 1.23742
Progress: [====================>] 100% [test loss] 1.17816 [test acc]
0.59120
Progress: [====================>] 100% [train loss] 1.23259
Progress: [====================>] 100% [test loss] 1.11657 [test acc]
0.61480
test acc improved from 0.6100999712944031 to 0.614799976348877
Progress: [====================>] 100% [train loss] 1.21451
Progress: [====================>] 100% [test loss] 1.10920 [test acc]
0.61760
test acc improved from 0.614799976348877 to 0.6176000237464905
Progress: [====================>] 100% [train loss] 1.20464
Progress: [====================>] 100% [test loss] 1.14015 [test acc]
0.60630
```

```
Progress: [====================>] 100% [train loss] 1.19785
Progress: [====================>] 100% [test loss] 1.15798 [test acc]
0.59550
Progress: [====================>] 100% [train loss] 1.19128
Progress: [====================>] 100% [test loss] 1.12608 [test acc]
0.61060
Progress: [====================>] 100% [train loss] 1.18509
Progress: [====================>] 100% [test loss] 1.11405 [test acc]
0.61530
Progress: [====================>] 100% [train loss] 1.18987
Progress: [====================>] 100% [test loss] 1.09312 [test acc]
0.62050
test acc improved from 0.6176000237464905 to 0.6205000281333923
Progress: [====================>] 100% [train loss] 1.16611
Progress: [====================>] 100% [test loss] 1.05647 [test acc]
0.62960
test acc improved from 0.6205000281333923 to 0.6295999884605408
Progress: [====================>] 100% [train loss] 1.16165
Progress: [====================>] 100% [test loss] 1.08008 [test acc]
0.62350
Progress: [====================>] 100% [train loss] 1.15785
Progress: [====================>] 100% [test loss] 1.05282 [test acc]
0.63340
test acc improved from 0.6295999884605408 to 0.633400022983551
Progress: [====================>] 100% [train loss] 1.14377
Progress: [====================>] 100% [test loss] 1.06171 [test acc]
0.63240
Progress: [====================>] 100% [train loss] 1.13533
Progress: [====================>] 100% [test loss] 1.06940 [test acc]
0.63080
Progress: [====================>] 100% [train loss] 1.13917
Progress: [====================>] 100% [test loss] 1.04003 [test acc]
0.63700
test acc improved from 0.633400022983551 to 0.6370000243186951
Progress: [====================>] 100% [train loss] 1.12842
Progress: [====================>] 100% [test loss] 1.07322 [test acc]
0.63290
Progress: [====================>] 100% [train loss] 1.11796
Progress: [====================>] 100% [test loss] 1.05707 [test acc]
0.62470
Progress: [====================>] 100% [train loss] 1.11290
Progress: [====================>] 100% [test loss] 1.13223 [test acc]
0.60870
Progress: [====================>] 100% [train loss] 1.10927
Progress: [====================>] 100% [test loss] 1.00808 [test acc]
0.64830
test acc improved from 0.6370000243186951 to 0.6482999920845032
Progress: [====================>] 100% [train loss] 1.10599
Progress: [====================>] 100% [test loss] 1.11685 [test acc]
0.61800
Progress: [====================>] 100% [train loss] 1.09533
Progress: [====================>] 100% [test loss] 1.01485 [test acc]
0.64510
Progress: [====================>] 100% [train loss] 1.08879
Progress: [====================>] 100% [test loss] 1.00489 [test acc]
0.65110
test acc improved from 0.6482999920845032 to 0.6510999798774719
Progress: [====================>] 100% [train loss] 1.07969
Progress: [====================>] 100% [test loss] 1.03925 [test acc]
0.63760
Progress: [====================>] 100% [train loss] 1.07875
```

```
Progress: [====================>] 100% [test loss] 1.02330 [test acc]
0.64690
Progress: [====================>] 100% [train loss] 1.06824
Progress: [====================>] 100% [test loss] 1.02713 [test acc]
0.64420
Progress: [====================>] 100% [train loss] 1.06602
Progress: [====================>] 100% [test loss] 1.05484 [test acc]
0.63440
Progress: [====================>] 100% [train loss] 1.05153
Progress: [====================>] 100% [test loss] 0.97563 [test acc]
0.66460
test acc improved from 0.6510999798774719 to 0.6646000146865845
Progress: [====================>] 100% [train loss] 1.04605
Progress: [====================>] 100% [test loss] 0.98672 [test acc]
0.65270
Progress: [====================>] 100% [train loss] 1.03886
Progress: [====================>] 100% [test loss] 1.01814 [test acc]
0.64380
Progress: [====================>] 100% [train loss] 1.02964
Progress: [====================>] 100% [test loss] 1.01783 [test acc]
0.64090
Progress: [====================>] 100% [train loss] 1.02789
Progress: [====================>] 100% [test loss] 0.97628 [test acc]
0.65810
Progress: [====================>] 100% [train loss] 1.02439
Progress: [====================>] 100% [test loss] 1.04760 [test acc]
0.63530
Progress: [====================>] 100% [train loss] 1.01206
Progress: [====================>] 100% [test loss] 0.92979 [test acc]
0.67390
test acc improved from 0.6646000146865845 to 0.6739000082015991
Progress: [====================>] 100% [train loss] 0.99611
Progress: [====================>] 100% [test loss] 0.96185 [test acc]
0.66080
Progress: [====================>] 100% [train loss] 0.99459
Progress: [====================>] 100% [test loss] 0.92555 [test acc]
0.67760
test acc improved from 0.6739000082015991 to 0.6776000261306763
Progress: [====================>] 100% [train loss] 0.98891
Progress: [====================>] 100% [test loss] 0.93932 [test acc]
0.67000
Progress: [====================>] 100% [train loss] 0.98694
Progress: [====================>] 100% [test loss] 0.92591 [test acc]
0.67900
test acc improved from 0.6776000261306763 to 0.6790000200271606
Progress: [====================>] 100% [train loss] 0.97567
Progress: [====================>] 100% [test loss] 0.94578 [test acc]
0.67400
Progress: [====================>] 100% [train loss] 0.97499
Progress: [====================>] 100% [test loss] 0.90201 [test acc]
0.68390
test acc improved from 0.6790000200271606 to 0.683899998664856
Progress: [====================>] 100% [train loss] 0.96743
Progress: [====================>] 100% [test loss] 0.91665 [test acc]
0.67760
Progress: [====================>] 100% [train loss] 0.95616
Progress: [====================>] 100% [test loss] 0.90419 [test acc]
0.68420
test acc improved from 0.683899998664856 to 0.6841999888420105
Progress: [====================>] 100% [train loss] 0.95858
Progress: [====================>] 100% [test loss] 0.87846 [test acc]
```

```
0.69390
test acc improved from 0.6841999888420105 to 0.6938999891281128
Progress: [====================>] 100% [train loss] 0.94748
Progress: [====================>] 100% [test loss] 0.88945 [test acc]
0.69200
Progress: [====================>] 100% [train loss] 0.94161
Progress: [====================>] 100% [test loss] 0.88304 [test acc]
0.69110
Progress: [====================>] 100% [train loss] 0.93786
Progress: [====================>] 100% [test loss] 0.89225 [test acc]
0.68820
Progress: [====================>] 100% [train loss] 0.92747
Progress: [====================>] 100% [test loss] 0.87985 [test acc]
0.69320
Progress: [====================>] 100% [train loss] 0.92117
Progress: [====================>] 100% [test loss] 0.89641 [test acc]
0.68680
Progress: [====================>] 100% [train loss] 0.92117
Progress: [====================>] 100% [test loss] 0.85913 [test acc]
0.70120
test acc improved from 0.6938999891281128 to 0.701200008392334
Progress: [====================>] 100% [train loss] 0.91254
Progress: [====================>] 100% [test loss] 0.86963 [test acc]
0.69620
Progress: [====================>] 100% [train loss] 0.90264
Progress: [====================>] 100% [test loss] 0.88398 [test acc]
0.69120
Progress: [====================>] 100% [train loss] 0.89819
Progress: [====================>] 100% [test loss] 0.84334 [test acc]
0.70310
test acc improved from 0.701200008392334 to 0.7031000256538391
Progress: [====================>] 100% [train loss] 0.89744
Progress: [====================>] 100% [test loss] 0.84602 [test acc]
0.69890
Progress: [====================>] 100% [train loss] 0.89080
Progress: [====================>] 100% [test loss] 0.85155 [test acc]
0.70000
Progress: [====================>] 100% [train loss] 0.88510
Progress: [====================>] 100% [test loss] 0.84089 [test acc]
0.70820
test acc improved from 0.7031000256538391 to 0.7081999778747559
Progress: [====================>] 100% [train loss] 0.87888
Progress: [====================>] 100% [test loss] 0.85075 [test acc]
0.70070
Progress: [====================>] 100% [train loss] 0.87260
Progress: [====================>] 100% [test loss] 0.85519 [test acc]
0.69930
Progress: [====================>] 100% [train loss] 0.86864
Progress: [====================>] 100% [test loss] 0.83333 [test acc]
0.70930
test acc improved from 0.7081999778747559 to 0.7092999815940857
Progress: [====================>] 100% [train loss] 0.86233
Progress: [====================>] 100% [test loss] 0.83045 [test acc]
0.71170
test acc improved from 0.7092999815940857 to 0.7117000222206116
Progress: [====================>] 100% [train loss] 0.86057
Progress: [====================>] 100% [test loss] 0.82205 [test acc]
0.71480
test acc improved from 0.7117000222206116 to 0.7148000001907349
Progress: [====================>] 100% [train loss] 0.85572
Progress: [====================>] 100% [test loss] 0.82698 [test acc]
```

```
0.70930
Progress: [====================>] 100% [train loss] 0.85004
Progress: [====================>] 100% [test loss] 0.81922 [test acc]
0.71190
Progress: [====================>] 100% [train loss] 0.84975
Progress: [====================>] 100% [test loss] 0.81627 [test acc]
0.71280
Progress: [====================>] 100% [train loss] 0.84272
Progress: [====================>] 100% [test loss] 0.81623 [test acc]
0.71270
Progress: [====================>] 100% [train loss] 0.84305
Progress: [====================>] 100% [test loss] 0.81166 [test acc]
0.71420
Progress: [====================>] 100% [train loss] 0.83993
Progress: [====================>] 100% [test loss] 0.81454 [test acc]
0.71650
test acc improved from 0.7148000001907349 to 0.7164999842643738
Progress: [====================>] 100% [train loss] 0.83463
Progress: [====================>] 100% [test loss] 0.80506 [test acc]
0.71860
test acc improved from 0.7164999842643738 to 0.7185999751091003
Progress: [====================>] 100% [train loss] 0.83404
Progress: [====================>] 100% [test loss] 0.80047 [test acc]
0.71820
Progress: [====================>] 100% [train loss] 0.83759
Progress: [====================>] 100% [test loss] 0.80214 [test acc]
0.71910
test acc improved from 0.7185999751091003 to 0.7190999984741211
Progress: [====================>] 100% [train loss] 0.82717
Progress: [====================>] 100% [test loss] 0.80090 [test acc]
0.71700
Progress: [====================>] 100% [train loss] 0.83014
Progress: [====================>] 100% [test loss] 0.80202 [test acc]
0.71610
Progress: [====================>] 100% [train loss] 0.83196
Progress: [====================>] 100% [test loss] 0.80070 [test acc]
0.71740
Progress: [====================>] 100% [train loss] 0.83139
Progress: [====================>] 100% [test loss] 0.80000 [test acc]
0.71700
Progress: [====================>] 100% [train loss] 0.82608
Progress: [====================>] 100% [test loss] 0.80025 [test acc]
0.71710
Progress: [====================>] 100% [train loss] 0.82689
Progress: [====================>] 100% [test loss] 0.79965 [test acc]
0.71690
```

```
---------------------------------------------------------------------
-------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-15-0e55e786e156> 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)
      3     model = LeNet()
      4   elif model_name == "vgg":
----> 5     model = VGG(model_cfg)
      6   elif model_name == "resnet":
      7     model = ResNet(model_cfg)

<ipython-input-10-00622ca84552> in __init__(self, cfg)
     29     # TODO: add average pool to collapse spatial dimensions
     30     #avgpool = F.lp_pool2d(input, norm_type, kernel_size=2)
 """difference"""
---> 31     avgpool = nn.AvgPool2d(Kernel_size=1, stride=1)
     32     layers.append(avgpool)
     33     self.layers = nn.Sequential(*layers)

TypeError: __init__() got an unexpected keyword argument 'Kernel_siz
e'
```

In [ ]: