EE21S055_Tutorial6

September 18, 2022

0.1 Imports

[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
```

0.2 Utilising GPU using Pytorch

[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
```

[3]:
```
torch.cuda.is_available()
```

[3]:
```
True
```

[4]:
```
!nvidia-smi
```

```
Sat Sep 17 19:40:21 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.43.04    Driver Version: 515.43.04    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
```

```
| 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  NVIDIA GeForce …  On     | 00000000:09:00.0 Off |                  N/A |
|  0%   39C    P2   109W / 350W |  11104MiB / 24576MiB |      5%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      1206      G   /usr/lib/xorg/Xorg                 35MiB |
|    0   N/A  N/A     72979      G   /usr/lib/xorg/Xorg                 76MiB |
|    0   N/A  N/A     73105      G   /usr/bin/gnome-shell               15MiB |
|    0   N/A  N/A    714142      C   …runima/aru_env/bin/python       1767MiB |
|    0   N/A  N/A    783901      C   …bhumika/DL/dl/bin/python3       3719MiB |
|    0   N/A  N/A    791518      C   …Deep_Learning/bin/python3       1933MiB |
|    0   N/A  N/A    810345      C   …ehasree/DLvenv/bin/python       1791MiB |
|    0   N/A  N/A    817238      C   …ehasree/DLvenv/bin/python       1691MiB |
+-----------------------------------------------------------------------------+
```

0.3 Dataset and Transforms

[5]:
```
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=train_transform, download=True)
test_dset = torchvision.datasets.CIFAR10(root="data/", train=False,␣
  ↪transform=test_transform, download=True)
```

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

[6]:
```
print(f"# of train samples: {len(train_dset)}")
print(f"# of test samples: {len(test_dset)}")
```

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

[7]:
```
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)
```

[8]:
```
print(f"# of train batches: {len(train_loader)}")
print(f"# of test batches: {len(test_loader)}")
```

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

[9]:
```
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])
```

0.4 LeNet

[10]:
```
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)
    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)
    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)
    out = self.activ(self.fc1(out))
    out = self.activ(self.fc2(out))
    out = self.fc3(out)
    return out
```

0.5 VGG

[11]:
```
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,␣
↪"pool", 512, 512, "pool"],
      "vgg16": [64, 64, "pool", 128, 128, "pool", 256, 256, 256, "pool", 512,␣
↪512, 512, "pool", 512, 512, 512, "pool"],
      "vgg19": [64, 64, "pool", 128, 128, "pool", 256, 256, 256, 256, "pool",␣
↪512, 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␣
↪each)
            # 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.
↪html?highlight=sequential#torch.nn.Sequential
            block = nn.Sequential(
                nn.
↪Conv2d(in_channels=in_dim,out_channels=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 = nn.AvgPool2d(kernel_size=1)
    layers.append(avgpool)
    self.layers = nn.Sequential(*layers)
    # TODO: missing output features
    self.fc = nn.Linear(512,10)

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

0.6 ResNet

[85]:
```
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=False)
    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 (in_dim!=dim):
        # TODO: add sequential module consisting of 1x1 convolution (given␣
↪stride, bias=False), batchnorm
        self.shortcut = nn.Sequential(
            nn.Conv2d(in_dim,dim,kernel_size=1,stride=stride,bias=False),
            nn.BatchNorm2d(dim)
        )

  def forward(self, x):
    out = self.activ(self.bn1(self.conv1(x)))
    out = self.bn2(self.conv2(out))
    # TODO: missing residual connection
    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,␣
 ↪bias=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 (in_dim!=dim*self.expansion):
        # TODO: add sequential module consisting of 1x1 convolution (given␣
 ↪stride, bias=False), batchnorm
        self.shortcut = nn.Sequential(
            nn.Conv2d(in_dim,dim*self.
 ↪expansion,kernel_size=1,stride=stride,bias=False),
            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 += 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
   out = F.avg_pool2d(out,out.shape[2])
   out = out.view(out.size(0),-1)

   out = self.linear(out)
   return out
```

0.7 Utility functions (can ignore)

[13]:
```
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 -␣
 ↪block),
        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()])
```

0.8 Training

[14]:
```
def train(model, optim, lr_sched=None, epochs=20, device=torch.device("cuda" if␣
 ↪torch.cuda.is_available() else "cpu"), criterion=None, metric_meter=None,␣
 ↪out_dir="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
    acc = (out.argmax(1) == target).type(torch.float).sum().item()

    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()
```

0.9 Run Experiments

[15]:
```
def run_experiment(model_name="lenet", model_cfg=None, epochs=20):
  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_decay=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)
```

[16]:
```
run_experiment(model_name="lenet")
```

```
Progress: [====================>] 100% [train loss] 1.99768
Progress: [====================>] 100% [test loss] 1.85006 [test acc]
28.98000
test acc improved from 0 to 28.98
Progress: [====================>] 100% [train loss] 1.91818
Progress: [====================>] 100% [test loss] 1.80415 [test acc]
32.53000
test acc improved from 28.98 to 32.53
Progress: [====================>] 100% [train loss] 1.86331
Progress: [====================>] 100% [test loss] 1.79053 [test acc]
32.07000
Progress: [====================>] 100% [train loss] 1.83782
Progress: [====================>] 100% [test loss] 1.76359 [test acc]
30.90000
Progress: [====================>] 100% [train loss] 1.80683
Progress: [====================>] 100% [test loss] 1.72777 [test acc]
36.10000
test acc improved from 32.53 to 36.1
Progress: [====================>] 100% [train loss] 1.77354
Progress: [====================>] 100% [test loss] 1.65982 [test acc]
38.66000
test acc improved from 36.1 to 38.66
```

```
Progress: [==================>] 100% [train loss] 1.74006
Progress: [==================>] 100% [test loss] 1.71353 [test acc]
38.25000
Progress: [==================>] 100% [train loss] 1.68507
Progress: [==================>] 100% [test loss] 1.61862 [test acc]
39.67000
test acc improved from 38.66 to 39.67
Progress: [==================>] 100% [train loss] 1.65915
Progress: [==================>] 100% [test loss] 1.58646 [test acc]
42.78000
test acc improved from 39.67 to 42.78
Progress: [==================>] 100% [train loss] 1.59603
Progress: [==================>] 100% [test loss] 1.55104 [test acc]
43.14000
test acc improved from 42.78 to 43.14
Progress: [==================>] 100% [train loss] 1.54667
Progress: [==================>] 100% [test loss] 1.52381 [test acc]
44.78000
test acc improved from 43.14 to 44.78
Progress: [==================>] 100% [train loss] 1.50150
Progress: [==================>] 100% [test loss] 1.47559 [test acc]
45.37000
test acc improved from 44.78 to 45.37
Progress: [==================>] 100% [train loss] 1.43652
Progress: [==================>] 100% [test loss] 1.32363 [test acc]
52.38000
test acc improved from 45.37 to 52.38
Progress: [==================>] 100% [train loss] 1.37409
Progress: [==================>] 100% [test loss] 1.32664 [test acc]
53.85000
test acc improved from 52.38 to 53.85
Progress: [==================>] 100% [train loss] 1.31185
Progress: [==================>] 100% [test loss] 1.21977 [test acc]
56.45000
test acc improved from 53.85 to 56.45
Progress: [==================>] 100% [train loss] 1.25784
Progress: [==================>] 100% [test loss] 1.17891 [test acc]
57.40000
test acc improved from 56.45 to 57.4
Progress: [==================>] 100% [train loss] 1.20363
Progress: [==================>] 100% [test loss] 1.12406 [test acc]
59.52000
test acc improved from 57.4 to 59.52
Progress: [==================>] 100% [train loss] 1.16398
Progress: [==================>] 100% [test loss] 1.08123 [test acc]
61.59000
test acc improved from 59.52 to 61.59
Progress: [==================>] 100% [train loss] 1.13158
```

```
Progress: [==================>] 100% [test loss] 1.05498 [test acc]
62.35000
test acc improved from 61.59 to 62.35
Progress: [==================>] 100% [train loss] 1.11515
Progress: [==================>] 100% [test loss] 1.04802 [test acc]
62.57000
test acc improved from 62.35 to 62.57
```

[17]:
```
run_experiment(model_name="vgg", model_cfg="vgg11")
```

```
Progress: [==================>] 100% [train loss] 2.49837
Progress: [==================>] 100% [test loss] 2.11753 [test acc]
18.64000
test acc improved from 0 to 18.64
Progress: [==================>] 100% [train loss] 1.82412
Progress: [==================>] 100% [test loss] 1.71281 [test acc]
32.83000
test acc improved from 18.64 to 32.83
Progress: [==================>] 100% [train loss] 1.49933
Progress: [==================>] 100% [test loss] 1.32061 [test acc]
52.28000
test acc improved from 32.83 to 52.28
Progress: [==================>] 100% [train loss] 1.18352
Progress: [==================>] 100% [test loss] 1.00651 [test acc]
64.18000
test acc improved from 52.28 to 64.18
Progress: [==================>] 100% [train loss] 0.99579
Progress: [==================>] 100% [test loss] 1.13784 [test acc]
59.84000
Progress: [==================>] 100% [train loss] 0.88512
Progress: [==================>] 100% [test loss] 0.97336 [test acc]
67.41000
test acc improved from 64.18 to 67.41
Progress: [==================>] 100% [train loss] 0.80116
Progress: [==================>] 100% [test loss] 1.12813 [test acc]
63.16000
Progress: [==================>] 100% [train loss] 0.73332
Progress: [==================>] 100% [test loss] 0.81526 [test acc]
72.04000
test acc improved from 67.41 to 72.04
Progress: [==================>] 100% [train loss] 0.66932
Progress: [==================>] 100% [test loss] 0.75485 [test acc]
74.10000
test acc improved from 72.04 to 74.1
Progress: [==================>] 100% [train loss] 0.61633
Progress: [==================>] 100% [test loss] 0.74455 [test acc]
74.44000
test acc improved from 74.1 to 74.44
```

```
Progress: [==================>] 100% [train loss] 0.57556
Progress: [==================>] 100% [test loss] 0.81263 [test acc]
73.85000
Progress: [==================>] 100% [train loss] 0.52683
Progress: [==================>] 100% [test loss] 0.67534 [test acc]
78.31000
test acc improved from 74.44 to 78.31
Progress: [==================>] 100% [train loss] 0.48040
Progress: [==================>] 100% [test loss] 0.56392 [test acc]
81.05000
test acc improved from 78.31 to 81.05
Progress: [==================>] 100% [train loss] 0.42881
Progress: [==================>] 100% [test loss] 0.52627 [test acc]
82.32000
test acc improved from 81.05 to 82.32
Progress: [==================>] 100% [train loss] 0.37935
Progress: [==================>] 100% [test loss] 0.47783 [test acc]
83.94000
test acc improved from 82.32 to 83.94
Progress: [==================>] 100% [train loss] 0.32716
Progress: [==================>] 100% [test loss] 0.49832 [test acc]
83.22000
Progress: [==================>] 100% [train loss] 0.28133
Progress: [==================>] 100% [test loss] 0.39997 [test acc]
86.63000
test acc improved from 83.94 to 86.63
Progress: [==================>] 100% [train loss] 0.23462
Progress: [==================>] 100% [test loss] 0.37468 [test acc]
87.58000
test acc improved from 86.63 to 87.58
Progress: [==================>] 100% [train loss] 0.20010
Progress: [==================>] 100% [test loss] 0.35865 [test acc]
88.12000
test acc improved from 87.58 to 88.12
Progress: [==================>] 100% [train loss] 0.17948
Progress: [==================>] 100% [test loss] 0.35253 [test acc]
88.49000
test acc improved from 88.12 to 88.49
```

[18]:
```
run_experiment(model_name="vgg", model_cfg="vgg13")
```

```
Progress: [==================>] 100% [train loss] 2.44108
Progress: [==================>] 100% [test loss] 2.07746 [test acc]
20.70000
test acc improved from 0 to 20.7
Progress: [==================>] 100% [train loss] 1.79503
Progress: [==================>] 100% [test loss] 1.62918 [test acc]
38.93000
```

```
test acc improved from 20.7 to 38.93
Progress: [====================>] 100% [train loss] 1.52694
Progress: [====================>] 100% [test loss] 1.38246 [test acc]
48.90000
test acc improved from 38.93 to 48.9
Progress: [====================>] 100% [train loss] 1.21315
Progress: [====================>] 100% [test loss] 1.06427 [test acc]
62.47000
test acc improved from 48.9 to 62.47
Progress: [====================>] 100% [train loss] 0.97325
Progress: [====================>] 100% [test loss] 1.19326 [test acc]
60.26000
Progress: [====================>] 100% [train loss] 0.81624
Progress: [====================>] 100% [test loss] 0.75401 [test acc]
73.96000
test acc improved from 62.47 to 73.96
Progress: [====================>] 100% [train loss] 0.71858
Progress: [====================>] 100% [test loss] 0.81863 [test acc]
72.17000
Progress: [====================>] 100% [train loss] 0.65010
Progress: [====================>] 100% [test loss] 0.75620 [test acc]
74.63000
test acc improved from 73.96 to 74.63
Progress: [====================>] 100% [train loss] 0.58490
Progress: [====================>] 100% [test loss] 0.86171 [test acc]
72.56000
Progress: [====================>] 100% [train loss] 0.54148
Progress: [====================>] 100% [test loss] 0.69478 [test acc]
77.01000
test acc improved from 74.63 to 77.01
Progress: [====================>] 100% [train loss] 0.49765
Progress: [====================>] 100% [test loss] 0.69928 [test acc]
76.93000
Progress: [====================>] 100% [train loss] 0.44990
Progress: [====================>] 100% [test loss] 0.64635 [test acc]
79.11000
test acc improved from 77.01 to 79.11
Progress: [====================>] 100% [train loss] 0.40234
Progress: [====================>] 100% [test loss] 0.58674 [test acc]
80.00000
test acc improved from 79.11 to 80.0
Progress: [====================>] 100% [train loss] 0.36221
Progress: [====================>] 100% [test loss] 0.47647 [test acc]
83.82000
test acc improved from 80.0 to 83.82
Progress: [====================>] 100% [train loss] 0.30554
Progress: [====================>] 100% [test loss] 0.39319 [test acc]
86.59000
```

```
test acc improved from 83.82 to 86.59
Progress: [==================>] 100% [train loss] 0.26545
Progress: [==================>] 100% [test loss] 0.38164 [test acc]
87.23000
test acc improved from 86.59 to 87.23
Progress: [==================>] 100% [train loss] 0.21576
Progress: [==================>] 100% [test loss] 0.32860 [test acc]
89.06000
test acc improved from 87.23 to 89.06
Progress: [==================>] 100% [train loss] 0.17796
Progress: [==================>] 100% [test loss] 0.30572 [test acc]
89.77000
test acc improved from 89.06 to 89.77
Progress: [==================>] 100% [train loss] 0.14826
Progress: [==================>] 100% [test loss] 0.29035 [test acc]
90.45000
test acc improved from 89.77 to 90.45
Progress: [==================>] 100% [train loss] 0.13208
Progress: [==================>] 100% [test loss] 0.28872 [test acc]
90.66000
test acc improved from 90.45 to 90.66
```

[19]:
```
run_experiment(model_name="vgg", model_cfg="vgg16")
```

```
Progress: [==================>] 100% [train loss] 2.49230
Progress: [==================>] 100% [test loss] 2.24595 [test acc]
14.35000
test acc improved from 0 to 14.35
Progress: [==================>] 100% [train loss] 1.97769
Progress: [==================>] 100% [test loss] 2.00301 [test acc]
23.58000
test acc improved from 14.35 to 23.58
Progress: [==================>] 100% [train loss] 1.65402
Progress: [==================>] 100% [test loss] 1.66933 [test acc]
40.78000
test acc improved from 23.58 to 40.78
Progress: [==================>] 100% [train loss] 1.33771
Progress: [==================>] 100% [test loss] 1.13723 [test acc]
60.03000
test acc improved from 40.78 to 60.03
Progress: [==================>] 100% [train loss] 1.06645
Progress: [==================>] 100% [test loss] 1.10258 [test acc]
61.46000
test acc improved from 60.03 to 61.46
Progress: [==================>] 100% [train loss] 0.90210
Progress: [==================>] 100% [test loss] 0.84698 [test acc]
71.38000
test acc improved from 61.46 to 71.38
```

```
Progress: [====================>] 100% [train loss] 0.78909
Progress: [====================>] 100% [test loss] 0.85340 [test acc]
71.80000
test acc improved from 71.38 to 71.8
Progress: [====================>] 100% [train loss] 0.71298
Progress: [====================>] 100% [test loss] 0.81018 [test acc]
73.08000
test acc improved from 71.8 to 73.08
Progress: [====================>] 100% [train loss] 0.64612
Progress: [====================>] 100% [test loss] 0.84514 [test acc]
71.12000
Progress: [====================>] 100% [train loss] 0.58225
Progress: [====================>] 100% [test loss] 0.72979 [test acc]
76.15000
test acc improved from 73.08 to 76.15
Progress: [====================>] 100% [train loss] 0.52955
Progress: [====================>] 100% [test loss] 0.60654 [test acc]
79.92000
test acc improved from 76.15 to 79.92
Progress: [====================>] 100% [train loss] 0.48558
Progress: [====================>] 100% [test loss] 0.52970 [test acc]
82.30000
test acc improved from 79.92 to 82.3
Progress: [====================>] 100% [train loss] 0.42712
Progress: [====================>] 100% [test loss] 0.55022 [test acc]
82.24000
Progress: [====================>] 100% [train loss] 0.37877
Progress: [====================>] 100% [test loss] 0.43747 [test acc]
85.24000
test acc improved from 82.3 to 85.24
Progress: [====================>] 100% [train loss] 0.33444
Progress: [====================>] 100% [test loss] 0.41765 [test acc]
85.75000
test acc improved from 85.24 to 85.75
Progress: [====================>] 100% [train loss] 0.28608
Progress: [====================>] 100% [test loss] 0.42421 [test acc]
86.08000
test acc improved from 85.75 to 86.08
Progress: [====================>] 100% [train loss] 0.23351
Progress: [====================>] 100% [test loss] 0.33544 [test acc]
88.79000
test acc improved from 86.08 to 88.79
Progress: [====================>] 100% [train loss] 0.19538
Progress: [====================>] 100% [test loss] 0.30805 [test acc]
90.05000
test acc improved from 88.79 to 90.05
Progress: [====================>] 100% [train loss] 0.16175
Progress: [====================>] 100% [test loss] 0.28934 [test acc]
```

```
90.63000
test acc improved from 90.05 to 90.63
Progress: [==================>] 100% [train loss] 0.14454
Progress: [==================>] 100% [test loss] 0.28464 [test acc]
90.81000
test acc improved from 90.63 to 90.81
```

[20]:
```
run_experiment(model_name="vgg", model_cfg="vgg19")
```

```
Progress: [==================>] 100% [train loss] 2.36738
Progress: [==================>] 100% [test loss] 1.97781 [test acc]
21.20000
test acc improved from 0 to 21.2
Progress: [==================>] 100% [train loss] 1.93916
Progress: [==================>] 100% [test loss] 1.93952 [test acc]
22.01000
test acc improved from 21.2 to 22.01
Progress: [==================>] 100% [train loss] 1.85289
Progress: [==================>] 100% [test loss] 1.98220 [test acc]
26.42000
test acc improved from 22.01 to 26.42
Progress: [==================>] 100% [train loss] 1.70820
Progress: [==================>] 100% [test loss] 1.70799 [test acc]
35.32000
test acc improved from 26.42 to 35.32
Progress: [==================>] 100% [train loss] 1.52197
Progress: [==================>] 100% [test loss] 1.39538 [test acc]
48.63000
test acc improved from 35.32 to 48.63
Progress: [==================>] 100% [train loss] 1.39908
Progress: [==================>] 100% [test loss] 1.59189 [test acc]
48.71000
test acc improved from 48.63 to 48.71
Progress: [==================>] 100% [train loss] 1.24281
Progress: [==================>] 100% [test loss] 1.48484 [test acc]
48.50000
Progress: [==================>] 100% [train loss] 1.05808
Progress: [==================>] 100% [test loss] 1.15496 [test acc]
58.28000
test acc improved from 48.71 to 58.28
Progress: [==================>] 100% [train loss] 0.91882
Progress: [==================>] 100% [test loss] 1.10952 [test acc]
62.64000
test acc improved from 58.28 to 62.64
Progress: [==================>] 100% [train loss] 0.80825
Progress: [==================>] 100% [test loss] 0.93130 [test acc]
67.77000
test acc improved from 62.64 to 67.77
```

```
Progress: [====================>] 100% [train loss] 0.72778
Progress: [====================>] 100% [test loss] 0.75594 [test acc]
74.14000
test acc improved from 67.77 to 74.14
Progress: [====================>] 100% [train loss] 0.65767
Progress: [====================>] 100% [test loss] 0.86275 [test acc]
72.95000
Progress: [====================>] 100% [train loss] 0.59681
Progress: [====================>] 100% [test loss] 0.63934 [test acc]
78.87000
test acc improved from 74.14 to 78.87
Progress: [====================>] 100% [train loss] 0.52449
Progress: [====================>] 100% [test loss] 0.58232 [test acc]
80.56000
test acc improved from 78.87 to 80.56
Progress: [====================>] 100% [train loss] 0.47651
Progress: [====================>] 100% [test loss] 0.61759 [test acc]
79.96000
Progress: [====================>] 100% [train loss] 0.41082
Progress: [====================>] 100% [test loss] 0.47925 [test acc]
84.07000
test acc improved from 80.56 to 84.07
Progress: [====================>] 100% [train loss] 0.36357
Progress: [====================>] 100% [test loss] 0.42612 [test acc]
85.93000
test acc improved from 84.07 to 85.93
Progress: [====================>] 100% [train loss] 0.31390
Progress: [====================>] 100% [test loss] 0.40071 [test acc]
87.11000
test acc improved from 85.93 to 87.11
Progress: [====================>] 100% [train loss] 0.27374
Progress: [====================>] 100% [test loss] 0.37058 [test acc]
88.27000
test acc improved from 87.11 to 88.27
Progress: [====================>] 100% [train loss] 0.25003
Progress: [====================>] 100% [test loss] 0.36040 [test acc]
88.21000
```

[91]:
```
run_experiment(model_name="resnet", model_cfg="resnet18")
```

```
Progress: [====================>] 100% [train loss] 1.85363
Progress: [====================>] 100% [test loss] 1.50344 [test acc]
45.19000
test acc improved from 0 to 45.19
Progress: [====================>] 100% [train loss] 1.39760
Progress: [====================>] 100% [test loss] 1.36675 [test acc]
51.39000
test acc improved from 45.19 to 51.39
```

```
Progress: [====================>] 100% [train loss] 1.10971
Progress: [====================>] 100% [test loss] 1.00291 [test acc]
64.05000
test acc improved from 51.39 to 64.05
Progress: [====================>] 100% [train loss] 0.91560
Progress: [====================>] 100% [test loss] 1.07837 [test acc]
63.15000
Progress: [====================>] 100% [train loss] 0.75763
Progress: [====================>] 100% [test loss] 0.81671 [test acc]
72.21000
test acc improved from 64.05 to 72.21
Progress: [====================>] 100% [train loss] 0.65539
Progress: [====================>] 100% [test loss] 0.73771 [test acc]
74.90000
test acc improved from 72.21 to 74.9
Progress: [====================>] 100% [train loss] 0.58522
Progress: [====================>] 100% [test loss] 0.76168 [test acc]
74.62000
Progress: [====================>] 100% [train loss] 0.53549
Progress: [====================>] 100% [test loss] 0.64815 [test acc]
78.03000
test acc improved from 74.9 to 78.03
Progress: [====================>] 100% [train loss] 0.49188
Progress: [====================>] 100% [test loss] 0.54227 [test acc]
81.41000
test acc improved from 78.03 to 81.41
Progress: [====================>] 100% [train loss] 0.44862
Progress: [====================>] 100% [test loss] 0.75183 [test acc]
77.86000
Progress: [====================>] 100% [train loss] 0.41324
Progress: [====================>] 100% [test loss] 0.45600 [test acc]
84.32000
test acc improved from 81.41 to 84.32
Progress: [====================>] 100% [train loss] 0.37006
Progress: [====================>] 100% [test loss] 0.52401 [test acc]
82.63000
Progress: [====================>] 100% [train loss] 0.32874
Progress: [====================>] 100% [test loss] 0.44176 [test acc]
85.46000
test acc improved from 84.32 to 85.46
Progress: [====================>] 100% [train loss] 0.28406
Progress: [====================>] 100% [test loss] 0.39078 [test acc]
86.84000
test acc improved from 85.46 to 86.84
Progress: [====================>] 100% [train loss] 0.24206
Progress: [====================>] 100% [test loss] 0.35797 [test acc]
87.93000
test acc improved from 86.84 to 87.93
```

```
Progress: [==================>] 100% [train loss] 0.20007
Progress: [==================>] 100% [test loss] 0.30316 [test acc]
90.00000
test acc improved from 87.93 to 90.0
Progress: [==================>] 100% [train loss] 0.15968
Progress: [==================>] 100% [test loss] 0.27374 [test acc]
91.14000
test acc improved from 90.0 to 91.14
Progress: [==================>] 100% [train loss] 0.12435
Progress: [==================>] 100% [test loss] 0.26047 [test acc]
91.40000
test acc improved from 91.14 to 91.4
Progress: [==================>] 100% [train loss] 0.09912
Progress: [==================>] 100% [test loss] 0.23887 [test acc]
92.20000
test acc improved from 91.4 to 92.2
Progress: [==================>] 100% [train loss] 0.08385
Progress: [==================>] 100% [test loss] 0.23715 [test acc]
92.26000
test acc improved from 92.2 to 92.26
```

[93]:
```
run_experiment(model_name="resnet", model_cfg="resnet50")
```

```
Progress: [==================>] 100% [train loss] 2.49105
Progress: [==================>] 100% [test loss] 1.95167 [test acc]
22.99000
test acc improved from 0 to 22.99
Progress: [==================>] 100% [train loss] 1.84636
Progress: [==================>] 100% [test loss] 1.71040 [test acc]
34.28000
test acc improved from 22.99 to 34.28
Progress: [==================>] 100% [train loss] 1.59796
Progress: [==================>] 100% [test loss] 1.41821 [test acc]
46.98000
test acc improved from 34.28 to 46.98
Progress: [==================>] 100% [train loss] 1.32265
Progress: [==================>] 100% [test loss] 1.18074 [test acc]
56.99000
test acc improved from 46.98 to 56.99
Progress: [==================>] 100% [train loss] 1.13106
Progress: [==================>] 100% [test loss] 1.17298 [test acc]
57.31000
test acc improved from 56.99 to 57.31
Progress: [==================>] 100% [train loss] 1.00609
Progress: [==================>] 100% [test loss] 1.40445 [test acc]
52.20000
Progress: [==================>] 100% [train loss] 0.91767
Progress: [==================>] 100% [test loss] 1.04850 [test acc]
```

```
64.88000
test acc improved from 57.31 to 64.88
Progress: [==================>] 100% [train loss] 0.84548
Progress: [==================>] 100% [test loss] 0.86910 [test acc]
68.86000
test acc improved from 64.88 to 68.86
Progress: [==================>] 100% [train loss] 0.77406
Progress: [==================>] 100% [test loss] 0.80187 [test acc]
72.25000
test acc improved from 68.86 to 72.25
Progress: [==================>] 100% [train loss] 0.71192
Progress: [==================>] 100% [test loss] 0.73169 [test acc]
75.04000
test acc improved from 72.25 to 75.04
Progress: [==================>] 100% [train loss] 0.63558
Progress: [==================>] 100% [test loss] 0.80377 [test acc]
73.56000
Progress: [==================>] 100% [train loss] 0.57921
Progress: [==================>] 100% [test loss] 0.62985 [test acc]
78.65000
test acc improved from 75.04 to 78.65
Progress: [==================>] 100% [train loss] 0.52649
Progress: [==================>] 100% [test loss] 0.64654 [test acc]
77.70000
Progress: [==================>] 100% [train loss] 0.47134
Progress: [==================>] 100% [test loss] 0.55946 [test acc]
80.83000
test acc improved from 78.65 to 80.83
Progress: [==================>] 100% [train loss] 0.42190
Progress: [==================>] 100% [test loss] 0.51426 [test acc]
82.56000
test acc improved from 80.83 to 82.56
Progress: [==================>] 100% [train loss] 0.36953
Progress: [==================>] 100% [test loss] 0.43794 [test acc]
85.34000
test acc improved from 82.56 to 85.34
Progress: [==================>] 100% [train loss] 0.32161
Progress: [==================>] 100% [test loss] 0.41156 [test acc]
85.87000
test acc improved from 85.34 to 85.87
Progress: [==================>] 100% [train loss] 0.28001
Progress: [==================>] 100% [test loss] 0.35456 [test acc]
87.98000
test acc improved from 85.87 to 87.98
Progress: [==================>] 100% [train loss] 0.24272
Progress: [==================>] 100% [test loss] 0.32878 [test acc]
89.06000
test acc improved from 87.98 to 89.06
```

```
Progress: [==================>] 100% [train loss] 0.21911
Progress: [==================>] 100% [test loss] 0.32036 [test acc]
89.35000
test acc improved from 89.06 to 89.35
```

[95]:
```
run_experiment(model_name="resnet", model_cfg="resnet152")
```

```
Progress: [==================>] 100% [train loss] 2.61806
Progress: [==================>] 100% [test loss] 1.98638 [test acc]
22.47000
test acc improved from 0 to 22.47
Progress: [==================>] 100% [train loss] 1.93957
Progress: [==================>] 100% [test loss] 1.79311 [test acc]
31.27000
test acc improved from 22.47 to 31.27
Progress: [==================>] 100% [train loss] 1.71470
Progress: [==================>] 100% [test loss] 1.57397 [test acc]
42.86000
test acc improved from 31.27 to 42.86
Progress: [==================>] 100% [train loss] 1.48099
Progress: [==================>] 100% [test loss] 1.40066 [test acc]
48.99000
test acc improved from 42.86 to 48.99
Progress: [==================>] 100% [train loss] 1.28673
Progress: [==================>] 100% [test loss] 1.25056 [test acc]
54.62000
test acc improved from 48.99 to 54.62
Progress: [==================>] 100% [train loss] 1.12041
Progress: [==================>] 100% [test loss] 1.02216 [test acc]
62.94000
test acc improved from 54.62 to 62.94
Progress: [==================>] 100% [train loss] 0.99068
Progress: [==================>] 100% [test loss] 1.01641 [test acc]
65.01000
test acc improved from 62.94 to 65.01
Progress: [==================>] 100% [train loss] 0.89672
Progress: [==================>] 100% [test loss] 1.08985 [test acc]
63.22000
Progress: [==================>] 100% [train loss] 0.80515
Progress: [==================>] 100% [test loss] 0.81202 [test acc]
71.26000
test acc improved from 65.01 to 71.26
Progress: [==================>] 100% [train loss] 0.72578
Progress: [==================>] 100% [test loss] 0.79400 [test acc]
72.15000
test acc improved from 71.26 to 72.15
Progress: [==================>] 100% [train loss] 0.65587
Progress: [==================>] 100% [test loss] 0.74817 [test acc]
```

```
74.46000
test acc improved from 72.15 to 74.46
Progress: [==================>] 100% [train loss] 0.58560
Progress: [==================>] 100% [test loss] 0.57894 [test acc]
80.03000
test acc improved from 74.46 to 80.03
Progress: [==================>] 100% [train loss] 0.53010
Progress: [==================>] 100% [test loss] 0.59598 [test acc]
79.81000
Progress: [==================>] 100% [train loss] 0.48260
Progress: [==================>] 100% [test loss] 0.61924 [test acc]
79.08000
Progress: [==================>] 100% [train loss] 0.42896
Progress: [==================>] 100% [test loss] 0.48719 [test acc]
83.62000
test acc improved from 80.03 to 83.62
Progress: [==================>] 100% [train loss] 0.37894
Progress: [==================>] 100% [test loss] 0.42114 [test acc]
85.93000
test acc improved from 83.62 to 85.93
Progress: [==================>] 100% [train loss] 0.32384
Progress: [==================>] 100% [test loss] 0.39248 [test acc]
86.80000
test acc improved from 85.93 to 86.8
Progress: [==================>] 100% [train loss] 0.27796
Progress: [==================>] 100% [test loss] 0.33872 [test acc]
88.75000
test acc improved from 86.8 to 88.75
Progress: [==================>] 100% [train loss] 0.24053
Progress: [==================>] 100% [test loss] 0.32085 [test acc]
89.48000
test acc improved from 88.75 to 89.48
Progress: [==================>] 100% [train loss] 0.21585
Progress: [==================>] 100% [test loss] 0.31588 [test acc]
89.77000
test acc improved from 89.48 to 89.77
```

0.10 Questions

- Train and report test set metrics on three model types - LeNet, VGG, ResNet.
- Which model performs the best and why?
- Which model performs the worst and why?
- Number of the Epochs run = 20

Sl No	Model Name	Test set Accuracies in Percentage(%)
1.	LeNEt	62.57
2.	VGG11	88.49

Sl No	Model Name	Test set Accuracies in Percentage(%)
3.	VGG13	90.66
4.	VGG16	90.81
5.	VGG19	88.27
6.	ResNet18	92.26
7.	ResNet50	89.35
8.	ResNet152	89.77

1. The performance of LeNet is not so good. Lenet performs the worst because of shallow architecture and no better optimisation to push the accuracy.
2. Resnet performs better ,The principle on which ResNets work is to build a deeper networks compared to other plain networks and simultaneously find a optimised number of layers to negate the vanishing gradient problem.

- Note : Increased Resenet performs worst than its lower versions , so one need to find the fit properly according to the dataset and model to be used

Bonus Marks Separate File (`EE21S055_Tutorial6_Bonus.ipynb`)
1 Increased the channel from 16 to 32 * 16->32 Accuracy Moved from 62.57% to 64.48%

2 Increased the channel from 32 to 64 * 32->64 Accuracy Moved from 64.48% to 71.33

This clearly shows that the increase in the channel number increased the accuracies