ImageClasssification

December 8, 2021

[ ]:
```
#installs
!pip install -U scikit-learn
!pip install seaborn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-
packages (1.0.1)
Requirement already satisfied: scipy>=1.1.0 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn) (1.4.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from scikit-learn) (3.0.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn) (1.19.5)
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages
(0.11.2)
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-
packages (from seaborn) (1.4.1)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-
packages (from seaborn) (1.19.5)
Requirement already satisfied: matplotlib>=2.2 in /usr/local/lib/python3.7/dist-
packages (from seaborn) (3.2.2)
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-
packages (from seaborn) (1.1.5)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2->seaborn) (1.3.2)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2->seaborn) (3.0.6)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-
packages (from matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-
packages (from pandas>=0.23->seaborn) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-
packages (from python-dateutil>=2.1->matplotlib>=2.2->seaborn) (1.15.0)
```

[ ]:
```
#imports

import torch
import torch.nn as nn
from torch.utils.data import DataLoader
from torchvision.transforms.transforms import Compose, Normalize, RandomCrop,␣
 ↪RandomHorizontalFlip
from torch.utils.data.dataset import random_split
import torch.optim as optim
import torch.optim.lr_scheduler as lr_scheduler

from torchvision import datasets
from torchvision.transforms import ToTensor
from torchvision.datasets.utils import download_url
from torchvision.datasets import ImageFolder

import torch.nn.functional as F

import matplotlib.pyplot as plt

import copy
import time

from sklearn.metrics import confusion_matrix
import seaborn as sns
```

[ ]:
```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

[ ]:
```
%cd /content/gdrive/MyDrive/Colab\ Notebooks/img_classification
```

```
/content/gdrive/MyDrive/Colab Notebooks/img_classification
```

[ ]:
```
device = 'cpu'
if torch.cuda.is_available():
  device = 'cuda'

torch.cuda.is_available()
```

[ ]:
```
True
```

1 Dataset and Dataloaders

1.1 Datasets

[ ]:
```
data_dir = './data'

stats = ((0.491, 0.482, 0.446), (0.247, 0.243, 0.261))

augmentation_transform = Compose([
        RandomCrop(32, padding=4, padding_mode='reflect'),
        RandomHorizontalFlip(),
        ToTensor(),
        Normalize(*stats, inplace=True)
    ])

train_transform = Compose([
        ToTensor(),
        Normalize(*stats, inplace=True)
    ])

val_transform = Compose([
                        ToTensor(),
                        Normalize(*stats)])

train_ds =  datasets.CIFAR10(
    root=data_dir,
    train=True,
    download=True,
    transform=train_transform
)

val_ds =  datasets.CIFAR10(
    root=data_dir,
    train=False,
    download=False,
    transform=val_transform
)


batch_sizes = [32, 64, 128]
```

```
Files already downloaded and verified
```

1.2 Dataloaders

1.2.1 Device Dataloader

[ ]:
```
loaders = {}

for batch_size in batch_sizes:

  train_dl = DataLoader(train_ds, batch_size, shuffle=True, num_workers=2,␣
 ↪pin_memory=True)
  val_dl = DataLoader(val_ds, batch_size*2, num_workers=2, pin_memory=True)

  train_dl = DeviceDataLoader(train_dl, device)
  val_dl = DeviceDataLoader(val_dl, device)

  loaders[batch_size] = [train_dl, val_dl]
```

[ ]:
```
loaders
```

[ ]:
```
{32: [<__main__.DeviceDataLoader at 0x7f1a2729bd90>,
  <__main__.DeviceDataLoader at 0x7f1a2729b690>],
 64: [<__main__.DeviceDataLoader at 0x7f1a2729b190>,
  <__main__.DeviceDataLoader at 0x7f1a2729ba10>],
 128: [<__main__.DeviceDataLoader at 0x7f1a2729bb10>,
  <__main__.DeviceDataLoader at 0x7f1a2729b7d0>]}
```

[ ]:
```
loaders[32][0].device
```

[ ]:
```
'cuda'
```

[ ]:
```
def get_default_device():
    """Pick GPU if available, else CPU"""
    if torch.cuda.is_available():
        return torch.device('cuda')
    else:
        return torch.device('cpu')

def to_device(data, device):
    """Move tensor(s) to chosen device"""
    if isinstance(data, (list,tuple)):
        return [to_device(x, device) for x in data]
    return data.to(device, non_blocking=True)

class DeviceDataLoader():
    """Wrap a dataloader to move data to a device"""
    def __init__(self, dl, device):
        self.dl = dl
        self.device = device
```

```
    def __iter__(self):
        """Yield a batch of data after moving it to device"""
        for b in self.dl:
            yield to_device(b, self.device)

    def __len__(self):
        """Number of batches"""
        return len(self.dl)
```

2 Models

2.1 Base

[ ]:
```
class ImageClassificationBase(nn.Module):

    def training_step(self, batch):
        imgs, lbls = batch
        out = self(imgs) #Generate predictions
        loss = F.cross_entropy(out, lbls) # Calculate Loss
        return loss

    def validation_step(self, batch):

        imgs, lbls = batch
        out = self(imgs)
        loss = F.cross_entropy(out, lbls)
        acc = accuracy(out, lbls)
        return {"val_loss": loss.detach(), "val_acc": acc}

    def validation_epoch_end(self, outputs):
        batch_losses = [x["val_loss"] for x in outputs]
        epoch_loss = torch.stack(batch_losses).mean()
        batch_accs = [x["val_acc"] for x in outputs]
        epoch_acc = torch.stack(batch_accs).mean()
        return {"val_loss": epoch_loss.item(), "val_acc": epoch_acc.item()}

    def epoch_end(self, epoch, result):
        print(f"Epoch: {epoch}, last_lr: {result['lrs'][-1]}, train_loss:␣
 ↪{result['train_loss']}, val_loss: {result['val_loss']}, val_acc:␣
 ↪{result['val_acc']}")


def accuracy(outputs, lbls):
    _, preds = torch.max(outputs, dim=1)
    return torch.tensor(torch.sum(preds == lbls).item() / len(preds))
```

2.2 Simple CNN

[ ]:
```
class Net(ImageClassificationBase):

    def __init__(self) -> None:
        super().__init__()

        self.network = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, padding=1),
            nn.ReLU(), # activation, iterates all elements and if a value is␣
↪negative, changes to 0 introduces non linearity
            nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2,2),

            nn.Conv2d(64, 128, kernel_size=3, padding=1),
            nn.ReLU(), # activation, iterates all elements and if a value is␣
↪negative, changes to 0 introduces non linearity
            nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2,2),

            nn.Conv2d(128, 256, kernel_size=3, padding=1),
            nn.ReLU(), # activation, iterates all elements and if a value is␣
↪negative, changes to 0 introduces non linearity
            nn.Conv2d(256, 256, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2,2),

            nn.Flatten(),
            nn.Linear(256*4*4, 1024),
            nn.ReLU(),
            nn.Linear(1024,512),
            nn.ReLU(),
            nn.Linear(512, 10)
        )

    def forward(self, xb):
        return self.network(xb)

cnnModel = Net()
```

[ ]:
```
device
```

[ ]:
```
'cuda'
```

[ ]:
```
cnnModel.to(device)
cnnModel.cuda
```

[ ]:
```
<bound method Module.cuda of Net(
  (network): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU()
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU()
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU()
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU()
    (14): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (15): Flatten(start_dim=1, end_dim=-1)
    (16): Linear(in_features=4096, out_features=1024, bias=True)
    (17): ReLU()
    (18): Linear(in_features=1024, out_features=512, bias=True)
    (19): ReLU()
    (20): Linear(in_features=512, out_features=10, bias=True)
  )
)>
```

2.3 ResNet

[ ]:
```
def conv_block(in_channels, out_channels, pool=False):
  layers = [
            nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True)
  ]
  if pool: layers.append(nn.MaxPool2d(2))
  return nn.Sequential(*layers)


class ResNet9(ImageClassificationBase):
  def __init__(self, in_channels, num_classes):
    super().__init__()
```

```
    self.conv1 = conv_block(in_channels, 64)
    self.conv2 = conv_block(64, 128, pool=True)
    self.res1 = nn.Sequential(conv_block(128, 128), conv_block(128, 128))

    self.conv3 = conv_block(128, 256, pool=True)
    self.conv4 = conv_block(256, 512, pool=True)
    self.res2 = nn.Sequential(conv_block(512, 512), conv_block(512, 512))

    self.classifier = nn.Sequential(nn.MaxPool2d(4),
                                    nn.Flatten(),
                                    nn.Linear(512, num_classes))


  def forward(self, X):

    out = self.conv1(X)
    out = self.conv2(out)
    out = self.res1(out) + out
    out = self.conv3(out)
    out = self.conv4(out)
    out = self.res2(out) + out
    out = self.classifier(out)

    return out


resNetModel = ResNet9(3,10)
```

[ ]:
```
resNetModel.to(device)
```

[ ]:
```
ResNet9(
  (conv1): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
  )
  (conv2): Sequential(
    (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (res1): Sequential(
```

```
    (0): Sequential(
      (0): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (1): Sequential(
      (0): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (conv3): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (conv4): Sequential(
    (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (res2): Sequential(
    (0): Sequential(
      (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (1): Sequential(
      (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (classifier): Sequential(
    (0): MaxPool2d(kernel_size=4, stride=4, padding=0, dilation=1,
ceil_mode=False)
    (1): Flatten(start_dim=1, end_dim=-1)
```

```
    (2): Linear(in_features=512, out_features=10, bias=True)
  )
)
```

[ ]:
```
resNetModel.__
```

3 Training and Evaluation

[ ]:
```
@torch.no_grad()
def evaluate(model, val_loader):
    model.eval()
    outputs = [model.validation_step(batch) for batch in val_loader]
    return model.validation_epoch_end(outputs)

def get_lr(optimizer):
    for param_group in optimizer.param_groups:
        return param_group['lr']

def fit_one_cycle(epochs, max_lr, model, train_loader, val_loader,
                  weight_decay=0, grad_clip=None, opt_func=torch.optim.SGD):

    best_acc = [0.0, 0]
    torch.cuda.empty_cache()
    history = []

    # Set up cutom optimizer with weight decay
    optimizer = opt_func(model.parameters(), max_lr, weight_decay=weight_decay)
    # Set up one-cycle learning rate scheduler
    sched = torch.optim.lr_scheduler.OneCycleLR(optimizer, max_lr,␣
↪epochs=epochs,
                                                ␣
↪steps_per_epoch=len(train_loader))


    for epoch in range(epochs):

        print(f'Epoch {epoch}/{epochs-1}')
        print('-' * 10)

        # Training Phase
        model.train()
        train_losses = []
        lrs = []

        for batch in train_loader:
            loss = model.training_step(batch)
```

```
            train_losses.append(loss)
            #loss.requires_grad = True
            loss.backward()

            # Gradient clipping
            if grad_clip:
                nn.utils.clip_grad_value_(model.parameters(), grad_clip)

            optimizer.step()
            optimizer.zero_grad()

            # Record & update learning rate
            lrs.append(get_lr(optimizer))
            sched.step()

        # Validation phase
        result = evaluate(model, val_loader)
        result['train_loss'] = torch.stack(train_losses).mean().item()
        result['lrs'] = lrs
        model.epoch_end(epoch, result)
        history.append(result)

        epoch_acc = result['val_acc']

        if epoch_acc > best_acc[0]:
          best_acc[0] = epoch_acc
          best_acc[1] = epoch
          torch.save(model, 'best_modelrn9.pt')

        print(f"Best Acc: {best_acc[0]} in Epoch no. {best_acc[1]}")

    return history
```

[ ]:
```
history = [evaluate(resNetModel, val_dl)]
history
```

[ ]:
```
[{'val_acc': 0.10830078274011612, 'val_loss': 2.304202079772949}]
```

3.1 Hyperparametres

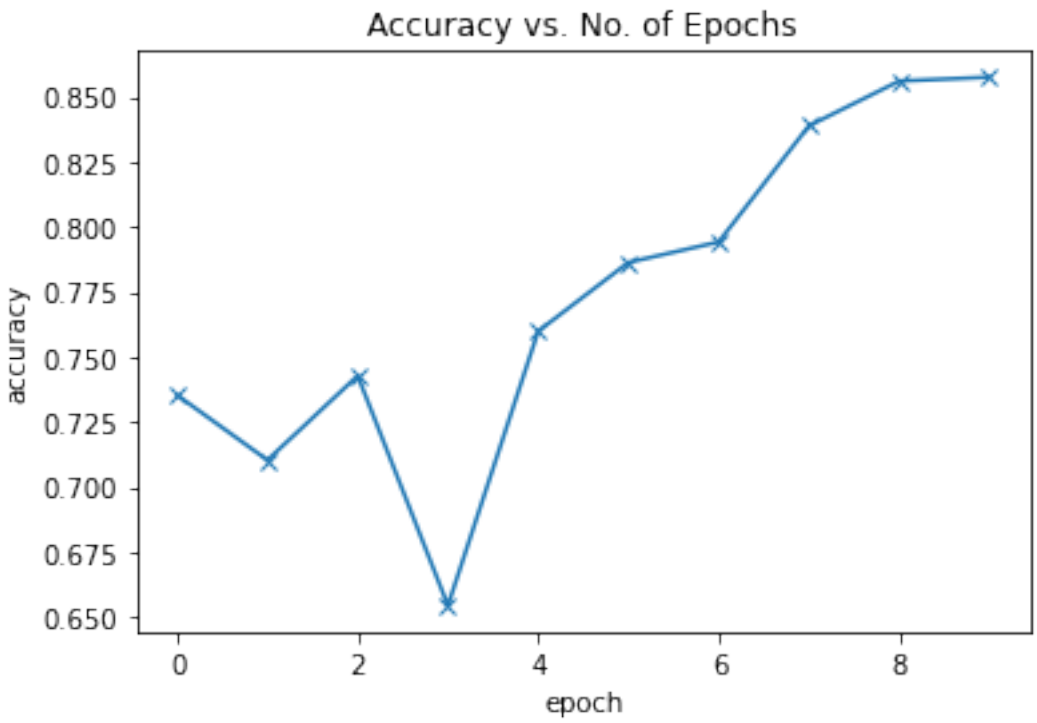
[ ]:
```
epochs = 10
max_lr = 0.01
grad_clip = 0.1
weight_decay = 1e-4
opt_func = torch.optim.Adam
```

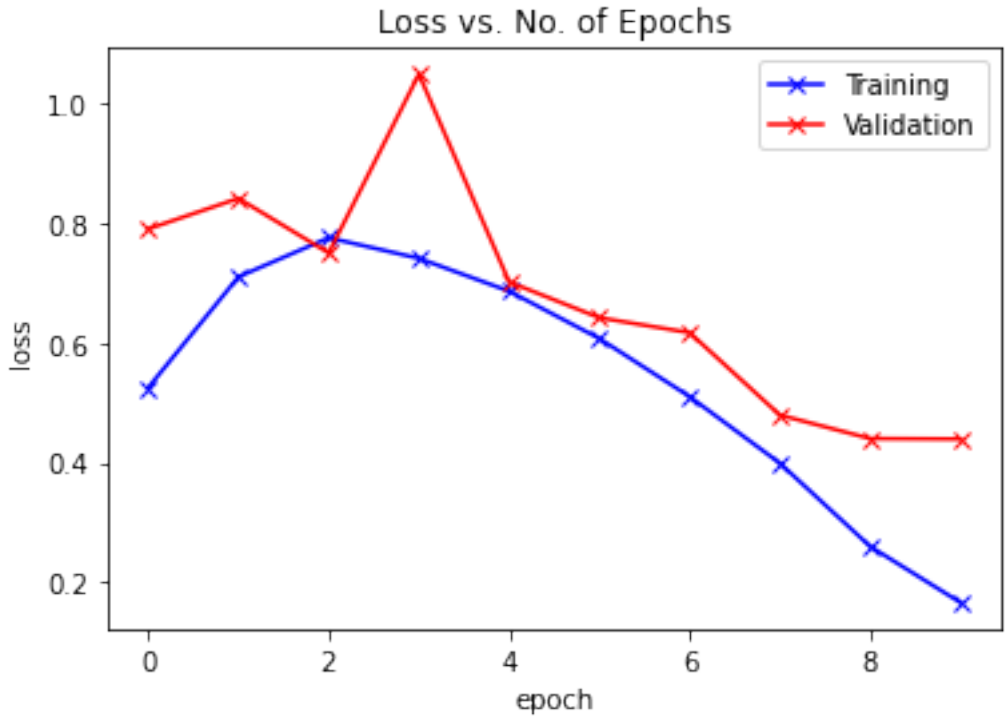
[ ]:
```
def plot_accuracies(history, filename):
    accuracies = [x['val_acc'] for x in history]
    plt.plot(accuracies, '-x')
    plt.xlabel('epoch')
    plt.ylabel('accuracy')
    plt.title('Accuracy vs. No. of Epochs')
    plt.savefig(f'acc_{filename}.png')
    plt.show()


def plot_losses(history, filename):
    train_losses = [ x.get('train_loss') for x in history]
    val_losses = [x['val_loss'] for x in history]
    plt.plot(train_losses, '-bx')
    plt.plot(val_losses, '-rx')
    plt.xlabel('epoch')
    plt.ylabel('loss')
    plt.legend(['Training', 'Validation'])
    plt.title('Loss vs. No. of Epochs')
    plt.savefig(f'loss_{filename}.png')
    plt.show()


def visualize_model(model, val_loader, num_images=6):
    was_training = model.training
    model.eval()
    images_so_far = 0
    fig = plt.figure()

    with torch.no_grad():
        for i, (inputs, labels) in enumerate(val_loader):

            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)

            for j in range(inputs.size()[0]):
                images_so_far += 1
                ax = plt.subplot(num_images//2, 2, images_so_far)
                ax.axis('off')
                ax.set_title('predicted: {}'.format(class_names[preds[j]]))
                imshow(inputs.cpu().data[j])

                if images_so_far == num_images:
                    model.train(mode=was_training)
                    return
        model.train(mode=was_training)
```

[ ]:
```
for key, val in loaders.items():

  print(f"""Parametres
  -----
  Num Epochs: {epochs},
  Batch size: {key},
  """)
  %%time
  history, best_model = fit_one_cycle(epochs, max_lr, resNetModel, val[0],␣
 ↪val[1], grad_clip=grad_clip,
                          weight_decay=weight_decay, opt_func=opt_func)

  plot_accuracies(history, f"resNetModel{key}")
  plot_losses(history, f"resNetModel{key}")

  torch.save(best_model, f"resNet{key}Dict.pt")
```

```
Parametres
  -----
  Num Epochs: 10,
  Batch size: 32,

CPU times: user 4 µs, sys: 1 µs, total: 5 µs
Wall time: 10 µs
Epoch 0/9
----------
Epoch: 0, last_lr: 0.0027981431071167143, train_loss: 0.5248221158981323,
val_loss: 0.7899484038352966, val_acc: 0.7352706789970398
Best Acc: 0.7352706789970398 in Epoch no. 0
Epoch 1/9
----------
Epoch: 1, last_lr: 0.007599071373902852, train_loss: 0.7099831104278564,
val_loss: 0.8410148024559021, val_acc: 0.7102906107902527
Best Acc: 0.7352706789970398 in Epoch no. 0
Epoch 2/9
----------
Epoch: 2, last_lr: 0.01, train_loss: 0.7752159237861633, val_loss:
0.7497961521148682, val_acc: 0.7427348494529724
Best Acc: 0.7427348494529724 in Epoch no. 2
Epoch 3/9
----------
Epoch: 3, last_lr: 0.009504846320134737, train_loss: 0.7408013343811035,
val_loss: 1.0484436750411987, val_acc: 0.6543591022491455
Best Acc: 0.7427348494529724 in Epoch no. 2
Epoch 4/9
----------
Epoch: 4, last_lr: 0.008117456539497631, train_loss: 0.6859750151634216,
```

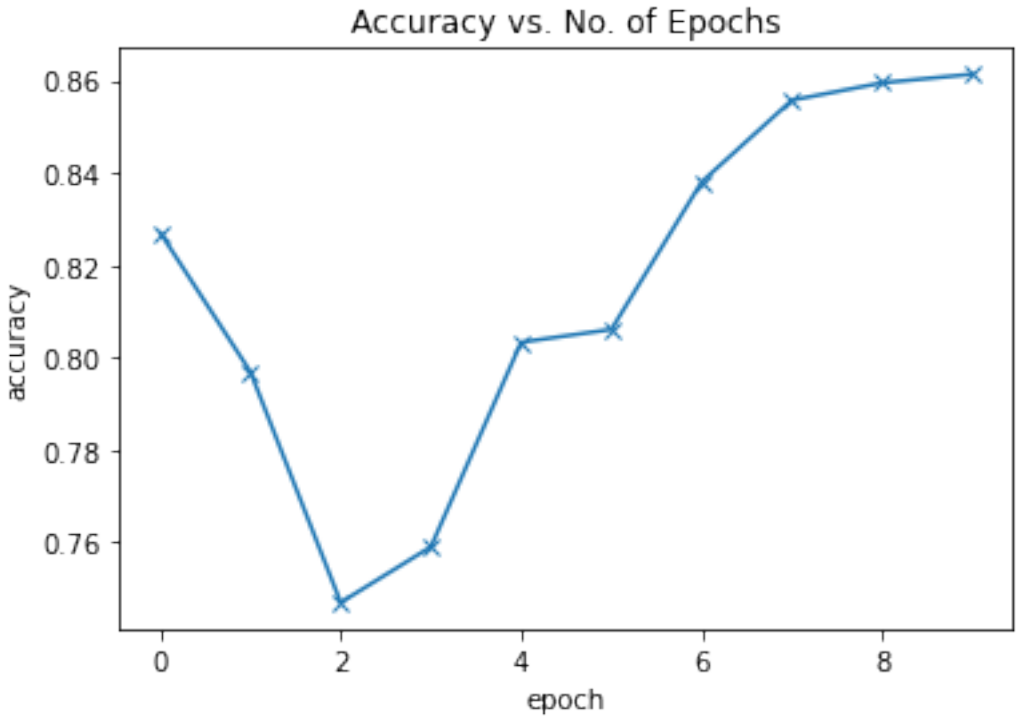
```
val_loss: 0.7013722658157349, val_acc: 0.7598527073860168
Best Acc: 0.7598527073860168 in Epoch no. 4
Epoch 5/9
----------
Epoch: 5, last_lr: 0.0061126202193628925, train_loss: 0.6069056987762451,
val_loss: 0.641569972038269, val_acc: 0.7863256335258484
Best Acc: 0.7863256335258484 in Epoch no. 5
Epoch 6/9
----------
Epoch: 6, last_lr: 0.003887419780637108, train_loss: 0.5099025964736938,
val_loss: 0.6167000532150269, val_acc: 0.7941879034042358
Best Acc: 0.7941879034042358 in Epoch no. 6
Epoch 7/9
----------
Epoch: 7, last_lr: 0.00188258346050237, train_loss: 0.3986726701259613,
val_loss: 0.4790746569633484, val_acc: 0.8388733863830566
Best Acc: 0.8388733863830566 in Epoch no. 7
Epoch 8/9
----------
Epoch: 8, last_lr: 0.0004951936798652628, train_loss: 0.2597155272960663,
val_loss: 0.4398353695869446, val_acc: 0.856090784072876
Best Acc: 0.856090784072876 in Epoch no. 8
Epoch 9/9
----------
Epoch: 9, last_lr: 4e-08, train_loss: 0.1660231500864029, val_loss:
0.4390949010848999, val_acc: 0.8575835824012756
Best Acc: 0.8575835824012756 in Epoch no. 9
```

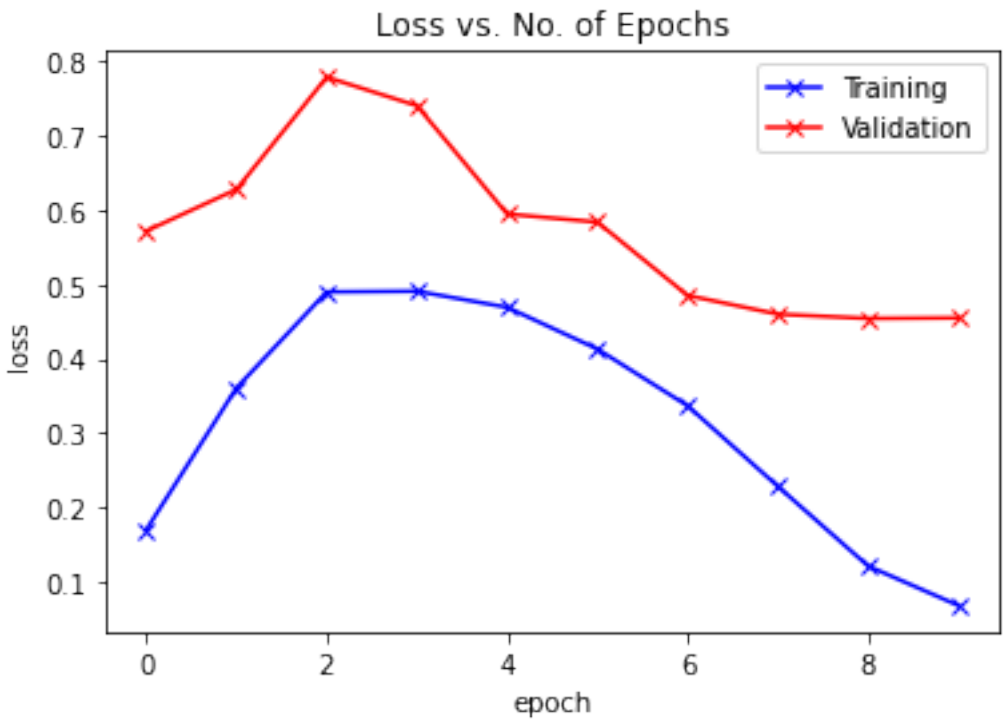

Accuracy vs. No. of Epochs
accuracy
epoch
0.850
0.825
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0
2
4
6
8


Loss vs. No. of Epochs
Training
Validation
loss
epoch
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8

```
Parametres
  -----
  Num Epochs: 10,
  Batch size: 64,

CPU times: user 0 ns, sys: 2 µs, total: 2 µs
Wall time: 6.68 µs
Epoch 0/9
----------
Epoch: 0, last_lr: 0.002796288276714347, train_loss: 0.16991613805294037,
val_loss: 0.5712572932243347, val_acc: 0.8270371556282043
Best Acc: 0.8270371556282043 in Epoch no. 0
Epoch 1/9
----------
Epoch: 1, last_lr: 0.007598143420255434, train_loss: 0.3605975806713104,
val_loss: 0.6271682977676392, val_acc: 0.796875
Best Acc: 0.8270371556282043 in Epoch no. 0
Epoch 2/9
----------
Epoch: 2, last_lr: 0.01, train_loss: 0.4894494116306305, val_loss:
0.7778607606887817, val_acc: 0.7469343543052673
Best Acc: 0.8270371556282043 in Epoch no. 0
Epoch 3/9
----------
Epoch: 3, last_lr: 0.009504846320134737, train_loss: 0.49041029810905457,
val_loss: 0.7395403385162354, val_acc: 0.758999228477478
Best Acc: 0.8270371556282043 in Epoch no. 0
Epoch 4/9
----------
Epoch: 4, last_lr: 0.008117456539497631, train_loss: 0.46883106231689453,
val_loss: 0.5941011309623718, val_acc: 0.8033030033111572
Best Acc: 0.8270371556282043 in Epoch no. 0
Epoch 5/9
----------
Epoch: 5, last_lr: 0.0061126202193628925, train_loss: 0.4128063917160034,
val_loss: 0.5832463502883911, val_acc: 0.8060719966888428
Best Acc: 0.8270371556282043 in Epoch no. 0
Epoch 6/9
----------
Epoch: 6, last_lr: 0.003887419780637108, train_loss: 0.3365459442138672,
val_loss: 0.48472481966018677, val_acc: 0.8380142450332642
Best Acc: 0.8380142450332642 in Epoch no. 6
Epoch 7/9
----------
Epoch: 7, last_lr: 0.00188258346050237, train_loss: 0.22834143042564392,
val_loss: 0.4595971405506134, val_acc: 0.8557159900665283
Best Acc: 0.8557159900665283 in Epoch no. 7
Epoch 8/9
```

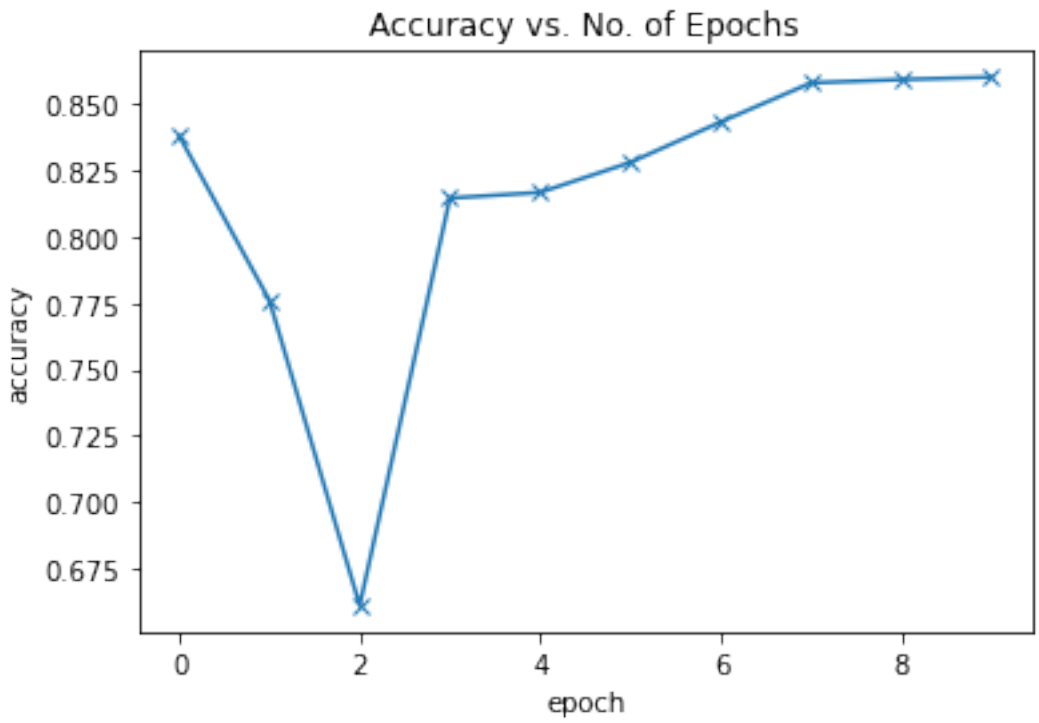
```
----------
Epoch: 8, last_lr: 0.0004951936798652628, train_loss: 0.1213671937584877,
val_loss: 0.4538663327693939, val_acc: 0.8594738841056824
Best Acc: 0.8594738841056824 in Epoch no. 8
Epoch 9/9
----------
Epoch: 9, last_lr: 4e-08, train_loss: 0.06839791685342789, val_loss:
0.4546057879924774, val_acc: 0.8613528609275818
Best Acc: 0.8613528609275818 in Epoch no. 9
```

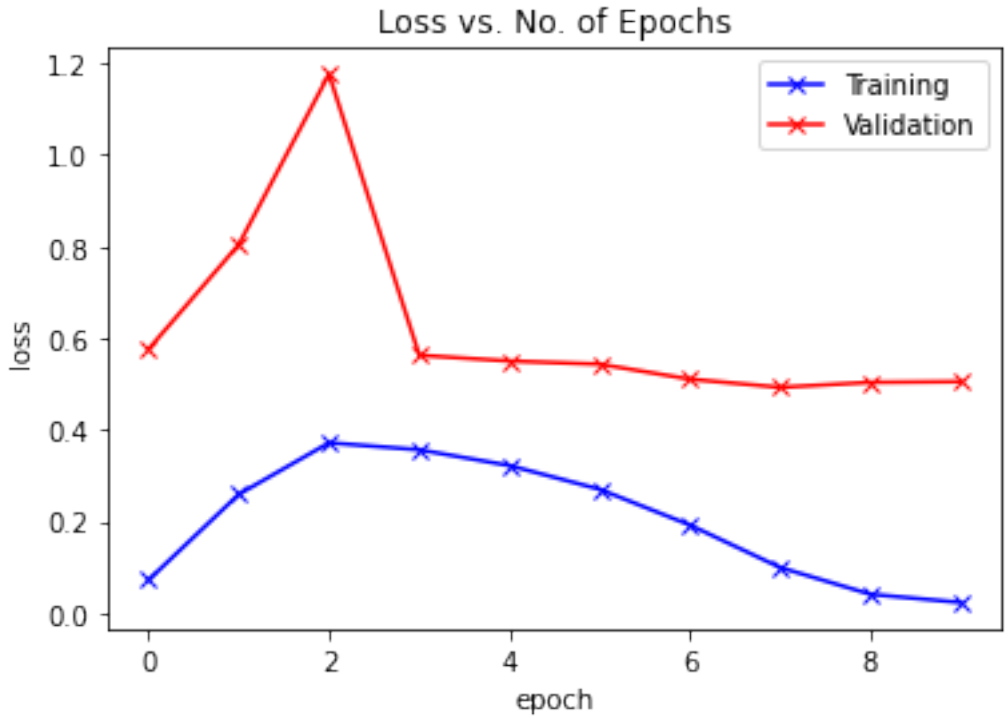





```
Parametres
  -----
  Num Epochs: 10,
  Batch size: 128,

CPU times: user 2 µs, sys: 1e+03 ns, total: 3 µs
Wall time: 5.96 µs
Epoch 0/9
----------
Epoch: 0, last_lr: 0.0027925753062899962, train_loss: 0.07193922996520996,
val_loss: 0.5758343935012817, val_acc: 0.8379882574081421
Best Acc: 0.8379882574081421 in Epoch no. 0
Epoch 1/9
----------
Epoch: 1, last_lr: 0.007596284777545438, train_loss: 0.25895464420318604,
val_loss: 0.8043265342712402, val_acc: 0.7759765386581421
Best Acc: 0.8379882574081421 in Epoch no. 0
Epoch 2/9
----------
Epoch: 2, last_lr: 0.01, train_loss: 0.37148746848106384, val_loss:
1.1762553453445435, val_acc: 0.660937488079071
Best Acc: 0.8379882574081421 in Epoch no. 0
Epoch 3/9
----------
```

```
Epoch: 3, last_lr: 0.009504846320134737, train_loss: 0.35620585083961487,
val_loss: 0.5625473856925964, val_acc: 0.8145507574081421
Best Acc: 0.8379882574081421 in Epoch no. 0
Epoch 4/9
----------
Epoch: 4, last_lr: 0.008117456539497631, train_loss: 0.32177823781967163,
val_loss: 0.5499623417854309, val_acc: 0.816699206829071
Best Acc: 0.8379882574081421 in Epoch no. 0
Epoch 5/9
----------
Epoch: 5, last_lr: 0.0061126202193628925, train_loss: 0.2699469327926636,
val_loss: 0.5431169867515564, val_acc: 0.8280273675918579
Best Acc: 0.8379882574081421 in Epoch no. 0
Epoch 6/9
----------
Epoch: 6, last_lr: 0.003887419780637108, train_loss: 0.19183219969272614,
val_loss: 0.5103043913841248, val_acc: 0.84326171875
Best Acc: 0.84326171875 in Epoch no. 6
Epoch 7/9
----------
Epoch: 7, last_lr: 0.00188258346050237, train_loss: 0.0991458147764206,
val_loss: 0.49252286553382874, val_acc: 0.8580077886581421
Best Acc: 0.8580077886581421 in Epoch no. 7
Epoch 8/9
----------
Epoch: 8, last_lr: 0.0004951936798652628, train_loss: 0.040739886462688446,
val_loss: 0.5034223198890686, val_acc: 0.8592773675918579
Best Acc: 0.8592773675918579 in Epoch no. 8
Epoch 9/9
----------
Epoch: 9, last_lr: 4e-08, train_loss: 0.02280818298459053, val_loss:
0.5049628615379333, val_acc: 0.860156238079071
Best Acc: 0.860156238079071 in Epoch no. 9
```


Accuracy vs. No. of Epochs
accuracy
0.850
0.825
0.800
0.775
0.750
0.725
0.700
0.675
0
2
4
6
8
epoch


Loss vs. No. of Epochs
loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
epoch
Training
Validation

[ ]:
```
%%time
history += fit_one_cycle(epochs, max_lr, resNetModel, train_dl, val_dl,␣
 ↪grad_clip=grad_clip,
                         weight_decay=weight_decay, opt_func=opt_func)
```

```
Epoch: 0, lat_lr: [0.0003999999999999993, 0.0004000269563249007,
0.0004001078249968384 5, 0.0004002426051075113, 0.0004004312951430937,
0.0004006738929842601 5, 0.0004009703959061955, 0.0004013208005786417,
0.000401725103065919, 0.0004021832988269861, 0.0004026953827154754,
0.0004032613489797637 3, 0.000403881191263028, 0.0004045549026033268,
0.0004052824754336645, 0.0004060639015820849 4, 0.0004068991722717702 6,
0.0004077882781211191, 0.0004087312091438748, 0.0004097279547492227 4,
0.0004107785037419115 6, 0.0004118828443223834, 0.0004130409640869022 6,
0.0004142528500277014, 0.0004155184885331116 7, 0.0004168378653877352,
0.0004182109657725908 3, 0.0004196377742652878, 0.0004211182748401956,
0.0004226524508686245, 0.0004242402851190128, 0.0004258817597571 23,
0.0004275768563462324 6, 0.0004293255558473537 5, 0.0004311278386194429,
0.0004329836844196143 6, 0.0004348930724033736 6, 0.0004368559811248548 5,
0.0004388723885370565 7, 0.0004409422271992090 3, 0.0004430656082414306,
0.0004452423734361901 6, 0.0004474725431273703, 0.0004497560922661469,
0.0004520929952041511 5, 0.0004544832256937452 6, 0.0004569267568883400 6,
0.0004594235613426724 7, 0.0004619736110131334, 0.0004645768772580643 3,
0.0004672333308380991 6, 0.0004699429419164746, 0.0004727056800593755,
0.0004755215142362746 7, 0.0004783904128202801, 0.0004813123435884886,
0.0004842872737223572, 0.0004873151569808053 5, 0.0004903959597836852 4,
0.0004935297232055002, 0.0004967163107166168, 0.0004999557245790806,
0.0005032479284084 37, 0.0005065928852273048, 0.0005099905574657889,
0.0005134409069619088, 0.0005169438949620235, 0.0005204994821212 62,
0.0005241076285039739, 0.0005277682935841757, 0.0005314814362459997,
0.0005352470147841643, 0.0005390649869044355, 0.0005429353097241053,
0.0005468579397724764, 0.0005508328329913419, 0.0005548599447354908,
0.0005589392297731983, 0.0005630706422867452, 0.0005672541358729 22,
0.0005714896635435583, 0.0005757771777260452, 0.0005801166302638727,
0.0005845079724171656, 0.0005889511548632428, 0.0005934461276971559,
0.0005979928404322606, 0.0006025912420007801, 0.0006072412802 80754 4,
0.0006119429044647365, 0.0006166960603241536, 0.0006215006949461 22,
0.0006263567543659341, 0.0006312641840412955, 0.0006362229288529257,
0.0006412329331051837, 0.0006462941405266948, 0.0006514064942709803,
0.0006565699369170952, 0.0006617844104702692, 0.0006670498563625723,
0.0006723662154535567, 0.0006777334280309282, 0.0006831514338112212,
0.0006886201719404645, 0.0006941395809948753, 0.0006997095989815429,
0.0007053301633391275, 0.0007110012109385645, 0.0007167226780837692,
0.0007224945005123544, 0.0007283166133963542, 0.0007341889513429473,
0.0007401114483952011, 0.0007460840380327 95, 0.0007521066531727891,
0.0007581792261703597, 0.0007643016888195693, 0.0007704739723541302,
0.0007766960074481778, 0.0007829677242170412, 0.0007892890520 52218 41,
0.0007956599204512731, 0.0008020802573604074, 0.0008085499908334869,
0.0008150690482037516, 0.0008216373562504355, 0.0008282548411996035,
```

0.0008349214287249763, 0.0008416370439487588, 0.000848401611442496,
0.0008552150552278998, 0.0008620772987777206, 0.0008689882650165966,
0.0008759478763219249, 0.0008829560545247257, 0.0008900127209105317,
0.0008971177796220261, 0.0009042712006511171, 0.0009114728538574726,
0.0009187226749517892, 0.0009260205825055105, 0.0009333664945499788,
0.0009407603285773707, 0.0009482020015416041, 0.0009556914298592782,
0.0009632285294106195, 0.000970813215540418, 0.0009784454030589817,
0.0009861250062430868, 0.0009938519388369548, 0.001001626114053208,
0.001009447445738466, 0.001017315842512367, 0.001025231219609089 3,
0.001033193486843456 1, 0.001041202554823728 4, 0.001049258333593639 7,
0.001057360732672275 5, 0.001065509661055097, 0.001073705027214952,
0.001081946739103108 9, 0.001090234704150294 3, 0.001098568829267719 9,
0.001106949020848149, 0.001115375184766925 8, 0.001123847226383043 3,
0.001132365050540209, 0.001140928561567907 9, 0.001149537663282480 1,
0.001158192258988197 8, 0.001166892251478356 5, 0.001175637543036362 6,
0.001184428035436833 5, 0.001193263629946695 5, 0.001202144227326302 4,
0.001211069727830542 7, 0.001220040031209962, 0.001229055036711888 5,
0.001238114643081560 5, 0.001247218748563278 7, 0.00125636725090153,
0.001265560047342146 5, 0.001274797034633458 7, 0.001284078109027454 7,
0.001293403166280944 1, 0.001302772101656730 6, 0.001312184809924784 5,
0.001321641185363429 6, 0.001331141121760531 4, 0.001340684512414683 6,
0.001350271250136411 7, 0.001359901227249375 7, 0.001369574335591579,
0.001379290466516582 3, 0.001389049510894727 4, 0.001398851359114359 3,
0.001408695901083060 3, 0.001418583026228883, 0.001428512623501595 7,
0.001438484581373926 7, 0.001448498787842818 2, 0.001458555130430687 4,
0.001468653496186682 3, 0.001478793771687960 9, 0.001488975843040956 5,
0.001499199595882657 1, 0.001509464915381895 6, 0.001519771686240633 4,
0.001530119792695263 5, 0.001540509118517901, 0.001550939547017696 2,
0.001561410961042140 9, 0.001571923242978383 3, 0.001582476274754555 2,
0.001593069937841093 8, 0.001603704113252071 9, 0.001614378681546535 9,
0.001625093522829846 5, 0.00163584851675503, 0.001646643542524120 8,
0.001657478478889530 9, 0.001668353204155396 6, 0.001679267596178956 8,
0.001690221532371921 3, 0.001701214889701846 4, 0.001712247544693519 5,
0.001723319373430343 1, 0.001734430251555729 7, 0.001745580054274496 4,
0.001756768656354263 2, 0.001767995932126868 4, 0.001779261755489772 2,
0.001790565999907479, 0.001801908538412951, 0.00181328924360904,
0.001824707987669921 6, 0.001836164642342520 7, 0.001847659078947960 8,
0.001859191168383003 3, 0.001870760781121506 8, 0.001882367787215866 5,
0.001894012056298490 6, 0.001905693457583252 5, 0.001917411859866966 6,
0.001929167131530858 1, 0.001940959140542042 3, 0.001952787754455006 4,
0.001964652840413102 7, 0.001976554265150034, 0.001988491894991355 5,
0.002000465595855967, 0.002012475233257635, 0.00202452067230649,
0.002036601777710542 5, 0.002048718413777208 4, 0.002060870444414832 4,
0.002073057733134209 3, 0.002085280143050125 7, 0.002097537536882894,
0.002109829776959891, 0.002122156725217112, 0.002134518243200709 8,
0.002146914192068567 4, 0.002159344432591838, 0.002171808825156518 5,
0.002184307229765024 6, 0.002196839506037746 3, 0.002209405513214638 5,
0.002222005110156793 4, 0.002234638155348035 4, 0.002247304504896501 7,

0.0022600040225362363, 0.0022727365596287965, 0.0022855019751648453,
0.0022983001257657603, 0.0023111308676852507, 0.0023239940568109624,
0.0023368895486660995, 0.0023498172198411052, 0.0023627768608450164,
0.0023757683904076 3, 0.0023887916411806055, 0.0024018464668893675,
0.0024149327209046 97, 0.0024280502562443 81, 0.0024411989255748584,
0.0024543785812128766, 0.0024675890751271554, 0.0024808302589400403,
0.0024941019839291763, 0.0025074041010291 76, 0.0025207364608332934,
0.0025340989135951026, 0.0025474913092301803, 0.0025609134973177865,
0.0025743653271025 64, 0.0025878466474962237, 0.0026013573070792426,
0.0026148971541025 69, 0.0026284660364893 19, 0.0026420638018364914,
0.0026556902974166794, 0.0026693453701797836, 0.0026830288667547 25,
0.0026967406334511834, 0.0027104805162613058, 0.0027242483608614513,
0.0027380440126139145, 0.0027518673165686635, 0.0027657181174650 89,
0.0027795962597337324, 0.0027935015874980497, 0.0028074339445761523,
0.0028213931744825622, 0.0028353791204299 74, 0.0028493916253310107,
0.0028634305317999896, 0.0028774956821546946, 0.0028915869184181 4,
0.0029057040823203505, 0.0029198470153001 34, 0.0029340155585068 69,
0.0029482095528022816, 0.0029624288837622 4, 0.0029766732566785343,
0.0029909426465606877, 0.0030052368481377 34, 0.0030195555700860 0334,
0.0030338990439010647, 0.0030482667161592387, 0.0030626585562597043,
0.0030770744025561 63, 0.0030915140931326 84, 0.0031059774658055185,
0.0031204643581249306, 0.0031349746073770 13, 0.0031494508050585518,
0.0031640645245136 88, 0.0031786438656666 09, 0.0031932459102904497,
0.0032078704943794938, 0.0032225174536727862, 0.0032371866236585826,
0.0032518778395756688, 0.0032665909364152142, 0.0032813257489226295,
0.0032960821115994 17, 0.0033108598587050 31, 0.0033256588242587424,
0.0033404788420415017, 0.0033553197455977986, 0.0033701813682375462,
0.0033850635430379 38, 0.0033999661028453355, 0.0034148888802771355,
0.0034298317077236544, 0.0034447944173500 15, 0.0034597768410980 23,
0.0034747788106880622, 0.0034898001576209 79, 0.0035048407131799 78,
0.0035199003084325177, 0.0035349787742322046, 0.0035500759412206 95,
0.0035651916398295987, 0.0035802325700282381, 0.0035954779525962724,
0.0036106482265841726, 0.0036258363518565 67, 0.0036410421578234 4,
0.0036562654736961 88, 0.0036715061284895417, 0.0036868639510234 85,
0.0037020387699251717, 0.0037173304136308635, 0.0037326387103878422,
0.0037479634882563 51, 0.0037633045751115186, 0.0037786617986452 94,
0.0037940349863683 82, 0.0038094239656121832, 0.0038248285635307244,
0.0038402486071026174, 0.0038556839231329 81, 0.0038711343382554 03,
0.0038865996789338 79, 0.0039020797714647 62, 0.0039175744419787 2,
0.0039330835116442 68, 0.0039486068206617 88], train_loss: 0.5334901213645935,
val_loss: 0.7030841112136841, val_acc: 0.7627929449081421
Epoch: 1, lat_lr: [0.003964144180281361, 0.003979695420788856,
0.0039952603675158 16, 0.0040108388456398 42, 0.0040264306801865 49,
0.0040420356960315 4, 0.0040576537179023 64, 0.0040732845703804 94,
0.0040889280779032 87, 0.0041045840647659 63, 0.0041202523551235 79,
0.0041359327729929 97, 0.0041516251422548 69, 0.0041673292866556 11,
0.0041830450298093 79, 0.0041987721952000 566, 0.0042145106061832 39,
0.0042302600859882 07, 0.0042460204577199 22, 0.0042617915443610085,

0.004277573168773744, 0.004293365153702046, 0.0043091673217734655,
0.004324979495501181, 0.004340801497285985, 0.00435633149418288,
0.004372474274080105, 0.004388324693347061, 0.00440418229190384,
0.004420052703478906, 0.004435929937981067, 0.004451815754366910 5,
0.004467709974210092, 0.004483612418989882, 0.00449952291009316 7,
0.004515441268816462, 0.004531367316367912, 0.00454730087386930 1,
0.004563241762358072, 0.004579189802789313, 0.00459514481603779 8,
0.004611106622899973, 0.004627075044095982 5, 0.0046430499002716825,
0.00465903101200065, 0.004675018199786202, 0.00469101128406341 1,
0.004707010085201117, 0.004723014423503957, 0.00473902411921436 9,
0.00475503899251462, 0.004771058863528823, 0.00478708355232495 85,
0.004803112878916896, 0.004819146663266409 5, 0.004835184725285208,
0.004851226884836956, 0.004867272961739289 5, 0.00488332277576585,
0.004899376146648306, 0.004915432894078373, 0.00493149283770984 1,
0.004947555797160602, 0.004963621592014675, 0.00497969004182423 5,
0.004995760966111633 5, 0.00501183418437143, 0.005027909516072419,
0.005043986780659655, 0.005060065797556489 5, 0.00507614638616658 2,
0.005092228365875948, 0.005108311556054975, 0.00512439577606045 4,
0.00514048084523761, 0.00515656658292213, 0.00517265280844219 15,
0.005188739341120497, 0.005204826000276292, 0.00522091260522740 75,
0.005236998975292281, 0.005253084929791988, 0.00526917028805227 3,
0.005285254869405576, 0.005301338493193061, 0.00531742097876665 3,
0.005333502145491054, 0.005349581812745785 6, 0.00536565979992720 8,
0.005381735926450549, 0.005397810011751943, 0.00541388187529044 05,
0.005429951336550056, 0.005446018215041779, 0.00546208233030561 2,
0.005478143501912593, 0.005494201549466819, 0.00551025629260748 2,
0.005526307551010886, 0.005542355144392469, 0.00555839889250884 4,
0.005574438615159804, 0.005590474132190363 5, 0.00560650526349276 7,
0.005622531829008522, 0.005638553648730417, 0.00565457054270454 3,
0.005670582331032318, 0.005686588833872509, 0.00570258987144324 14,
0.005718585264024028, 0.005734574831957789, 0.00575055839565285 9,
0.005766535775585016, 0.005782506792299494, 0.00579847126641298 9,
0.005814429018615694, 0.005830379869673292, 0.00584632364042898 4,
0.005862260151805489, 0.005878189224807069, 0.00589411068052152 9,
0.005910024340122229, 0.005925930024870097, 0.00594182755611562 9,
0.005957716755300899, 0.005973597443961571, 0.00598946944372889,
0.006005332576331701, 0.006021186663598435, 0.00603703152745912 3,
0.006052866989947393, 0.006068692873202461, 0.00608450899947114 4,
0.00610031519110984, 0.006116111270586531, 0.00613189706048278 3,
0.006147672383495728, 0.006163437062440061, 0.00617919092025002 8,
0.006194933779981418, 0.006210665464813544, 0.00622638579805124,
0.006242094603126832, 0.006257791703602132, 0.00627347692317041 7,
0.006289150085658401, 0.00630481101502823, 0.00632045953537944 3,
0.006336095470950959, 0.006351718646123044, 0.00636732888541928 7,
0.00638292601350857, 0.006398509855207037, 0.00641408023548006 8,
0.006429636979444229, 0.006445179912369252, 0.00646070885967999 3,
0.006476223646958388, 0.006491724099945419, 0.00650721004454306 65,
0.006522681306816264, 0.00653813771299486, 0.006553579089475556,

0.00656900526282387, 0.006584416059776079, 0.00659981130724116,
0.006615190832302741, 0.006630554462221041, 0.006645902024434810 5,
0.006661233465632684, 0.006676548256408041 5, 0.006691846581955091,
0.00670712815137665 6, 0.006722392793033169, 0.006737640335475203,
0.006752870607445376, 0.006768083437880287, 0.006783278655912439,
0.006798456090872149, 0.006813615572289473, 0.006828756929896117,
0.006843879993627348 6, 0.006858984593623911, 0.006874070560233925,
0.006889137724014803, 0.006904185915735141, 0.006919214966376630 5,
0.00693422470713595, 0.00694921496942666, 0.006964185584881106,
0.006979136385352294, 0.006994067202915797, 0.007008977869871624,
0.00702386821874612, 0.00703873808229383, 0.00705358729349939 2,
0.007068415685579406, 0.007083223091984306, 0.007098009346400235 5,
0.00711277428275091, 0.007127517735199491 4, 0.007142239538150435 6,
0.007156939526251369, 0.007171617534394935, 0.00718627339772065,
0.007200906951616759, 0.007215518031722085, 0.007230106473927864,
0.007244672114379604, 0.007259214789478913, 0.007273734335885341,
0.007288230590518223, 0.007302703390558492 5, 0.007317152573450525,
0.00733157797690396, 0.007345979438895518, 0.007360356797670833,
0.007374709891746256, 0.007389038559910674, 0.007403342641227325,
0.007417621975035595, 0.007431876400952837, 0.007446105758876163,
0.007460309888984239, 0.007474488631739086, 0.007488641827887874,
0.00750276931846470 4, 0.007516870944792 4, 0.007530946548484288,
0.007544995971445969, 0.00755901905587711, 0.007573015644273200 5,
0.007586985579427332, 0.007600928704431957, 0.007614844862680655,
0.007628733897869893, 0.007642595654000776, 0.007656429975380803,
0.007670236706625617, 0.007684015692660744, 0.007697766778723339,
0.007711489810363927, 0.007725184633448135 5, 0.007738851094158419 5,
0.007752489038995799, 0.007766098314781577, 0.00777967876865906,
0.007793230248095282, 0.007806752600882707, 0.007820245675140943,
0.007833709319318457, 0.007847143382194258, 0.00786054771287961,
0.00787392216081973, 0.00788726657579546, 0.007900580807924978,
0.00791386470766546, 0.007927118125814774, 0.007940340913513155,
0.007953532922244867, 0.007966694003839876, 0.007979824010475525,
0.007992922794678173, 0.008005990209324873, 0.00801902610764501,
0.00803203034322195, 0.0080450027699947, 0.008057943242259525,
0.008070851614671606, 0.008083727742246657, 0.008096571480362563,
0.00810938268476 1, 0.008122161211549054, 0.008134906917200846,
0.008147619658559131, 0.008160299292836914, 0.008172945677619052,
0.008185558670863858, 0.008198138130904687, 0.008210683916451533,
0.00822319588659261 8, 0.008235673900795968, 0.00824811781891 1,
0.008260527501170086, 0.008272902808190137, 0.008285243600974148,
0.00829754974091278, 0.008309821089785908, 0.008322057509764164,
0.008334258863410509, 0.008346425013681748, 0.008358555823930094,
0.008370651157904686, 0.008382710879753123, 0.008394734854023004,
0.008406722945663424, 0.008418675020026516, 0.008430590942868943,
0.008442470580353419, 0.008454313799050208, 0.008466120465938622,
0.008477890448408517, 0.008489623614261779, 0.00850131983171381,
0.008512978969395011, 0.008524600896352258, 0.008536185482050368,

0.008547732596373567, 0.008559242109626952, 0.008570713892537949,
0.008582147816257763, 0.008593543752362827, 0.008604901572856243,
0.008616221150169218, 0.0086275023571625, 0.008638745067127807,
0.008649949153789247, 0.008661114491304734, 0.00867224095426741,
0.00868332841770704, 0.008694376757091437, 0.008705385848327835,
0.008716355567764303, 0.008727285792191123, 0.008738176398842178,
0.00874902726539633, 0.008759838269978793, 0.008770609291162506,
0.00878134020796949, 0.008792030899872209, 0.008802681246794933,
0.008813291129115068, 0.008823860427664523, 0.008834389023731022,
0.008844876799059462, 0.008855323635853224, 0.00886572941677551,
0.008876094024950645, 0.008886417343965402, 0.008896699257870306,
0.008906939651180935, 0.008917138408879218, 0.008927295416414729,
0.00893741055970597, 0.008947483725141652, 0.008957514799581975,
0.0089675036703599, 0.008977450225282404, 0.008987354352631751,
0.008997215941166746, 0.009007034880123978, 0.009016811059219067,
0.009026544368647907, 0.009036234699087889, 0.009045881941699144,
0.009055485988125752, 0.009065046730496962, 0.009074564061428411,
0.009084037874023324, 0.009093468061873708, 0.009102854519061562,
0.009112197140160058, 0.009121495820234724, 0.009130750454844627,
0.009139960940043546, 0.009149127172381128, 0.00915824904890407,
0.009167326467157262, 0.009176359325184937, 0.00918534752153182,
0.009194290955244272, 0.009203189525871413, 0.009212043133466264,
0.009220851678586852, 0.009229615062297346, 0.009238333186169151,
0.009247005952282028, 0.009255633263225186, 0.009264215022098373,
0.009272751132512974, 0.009281241498593088, 0.009289686024976599,
0.009298084616816266, 0.00930643717978076, 0.009314743620055754,
0.009323003844344951, 0.009331217759871148, 0.009339385274377271,
0.009347506296127418, 0.009355580733907874], train_loss: 0.7064589858055115,
val_loss: 0.931765079498291, val_acc: 0.6939452886581421
Epoch: 2, lat_lr: [0.009363608497028155, 0.009371589495322017,
0.009379523639148464, 0.009387410839392766, 0.009395251007467449,
0.009403044055313297, 0.00941078989540034, 0.009418488440728836,
0.009426139604830247, 0.009433743301768214, 0.009441299446139516,
0.009448807953075037, 0.009456268738240712, 0.00946368171783848,
0.009471046808607221, 0.009478363927823695, 0.009485632993303465,
0.009492853923401828, 0.00950000026637014726, 0.00950715105357966,
0.009514227093076593, 0.00952125467602885, 0.009528233723504012,
0.009535164157114797, 0.00954204589901995, 0.009548878871925105,
0.009555662999083663, 0.009562398204297651, 0.009569084411918581,
0.009575721546848286, 0.009582309534539786, 0.009588848300998105,
0.009595337772781114, 0.009601777877000348, 0.009608168541321835,
0.009614509693966893, 0.009620801263712956, 0.009627043179894356,
0.009633235372403124, 0.009639377771689784, 0.00964547030876412,
0.009651512915195963, 0.009657505523115954, 0.009663448065216303,
0.00966934047475156, 0.009675182685539341, 0.009680974631961096,
0.00968671624896283, 0.009692407472055832, 0.009698048237317415,
0.00970363848139162, 0.00970917814148993, 0.00971466715539198,
0.009720105461446254, 0.009725492998570773, 0.00973082970625379,

0.009736115524554457, 0.009741350394103511, 0.009746534256103933,
0.00975166705233161, 0.009756748725135994, 0.009761779217440739,
0.009766758472744353, 0.009771686435120825, 0.009776563049220257,
0.009781388260269487, 0.009786162014072697, 0.009790884257012031,
0.00979555493604819, 0.009800173998721032, 0.00980474139315016,
0.009809257068035504, 0.009813720972657897, 0.009818133056879645,
0.009822493271145094, 0.009826801566481178, 0.009831057894497978,
0.00983526220738926, 0.009839414457933012, 0.009843514599491981,
0.009847562586014188, 0.009851558372033449, 0.009855501912669887,
0.009859393163630431, 0.009863232081209325, 0.009867018622288604,
0.00987075274433859, 0.009874434405418363, 0.009878063564176233,
0.009881640179850208, 0.00988516421226845, 0.009888635621849723,
0.009892054369603843, 0.00989542041713211, 0.009898733726627746,
0.009901994260876314, 0.00990520198325614, 0.00990835685773872,
0.00991145884888913, 0.009914507921866416, 0.009917504042423999,
0.009920447176910041, 0.00992333729226784, 0.009926174356036193,
0.00992895833634976, 0.009931689201939427, 0.009934366922132652,
0.009936991466853812, 0.009939562806624537, 0.009942080912564053,
0.009944545756389488, 0.009946957310416206, 0.00994931554755811,
0.009951620441327948, 0.009953871965837607, 0.00995607009579841,
0.009958214806521398, 0.009960306073917603, 0.009962343874498325,
0.00996432818537539, 0.009966258984261412, 0.00996813624947004,
0.009969959959916203, 0.009971730095116345, 0.009973446635188659,
0.009975109560853306, 0.009976718853432636, 0.00997827449485139,
0.009979776467636915, 0.009981224754919346, 0.009982619340431811,
0.009983960208510601, 0.009985247344095349, 0.009986480732729203,
0.009987660360558988, 0.009988786214335356, 0.009989858281412942,
0.0099908765497505, 0.009991841007911041, 0.00999275164506196 3,
0.009993608450975168, 0.009994411416027183, 0.009995160531199263,
0.009995855788077495, 0.009996497178852893, 0.009997084696321478,
0.009997618333884375, 0.00999809808554787, 0.009998523945923488,
0.00999889591022805, 0.009999213974283725, 0.009999478134518082,
0.009999688387964126, 0.00999984473226033, 0.009999947165650667,
0.009999995686984625, 0.00999999814727517, 0.009999986825072844,
0.009999965209983048, 0.009999933302050273, 0.009999891101340209,
0.009999838607939727, 0.00999977582195689, 0.009999702743520952,
0.009999619372782346, 0.009999525709912704, 0.009999421755104836,
0.009999307508572742, 0.00999918297055161, 0.009999048141297814,
0.00999890302108891, 0.009998747610223643, 0.009998581909021942,
0.009998405917824913, 0.009998219636994855, 0.00999802306691524,
0.009997816207990728, 0.009997599060647157, 0.009997371625331543,
0.009997133902512084, 0.009996885892678152, 0.009996627596340301,
0.009996359014030256, 0.009996080146300917, 0.009995790993726357,
0.009995491556901829, 0.009995181836443746, 0.009994861832989697,
0.009994531547198439, 0.009994190979749894, 0.009993840131345154,
0.009993479002706468, 0.009993107594577257, 0.009992725907722096,
0.009992333942926722, 0.009991931700998031, 0.009991519182764072,
0.009991096389074056, 0.00999066332079834, 0.009990219978828433,

0.009989766364076998, 0.009989302477477839, 0.009988828319985913,
0.009988343892577313, 0.009987849196249277, 0.009987344232020185,
0.009986829000929549, 0.009986303504038022, 0.009985767742427385,
0.009985221717200553, 0.009984665429481567, 0.0099840988804156,
0.00998352207116894, 0.009982935002929008, 0.009982337676904329,
0.009981730094324561, 0.009981112256440463, 0.009980484164523911,
0.009979845819867892, 0.009979197223786495, 0.009978538377614915,
0.009977869282709445, 0.00997718994044748, 0.009976500352227507,
0.009975800519469107, 0.00997509044361295, 0.009974370126120789,
0.009973639568475464, 0.009972898772180896, 0.009972147738762083,
0.00997138646976509 4, 0.009970614966757068, 0.00996983323132622,
0.009969041265081817, 0.009968239069654201, 0.009967426646694756,
0.009966603997875937, 0.009965771124891236, 0.009964928029455202,
0.00996407471330342, 0.009963211178192521, 0.009962337425900171,
0.009961453458225068, 0.00996055927698694, 0.009959654884026541,
0.009958740281205646, 0.00995781547040705, 0.009956880453534557,
0.009955935232512987, 0.009954979809288162, 0.009954014185826908,
0.009953038364117052, 0.009952052346167408, 0.009951056134007786,
0.009950049729688978, 0.009949033135282762, 0.009948006352881887,
0.00994696938460008, 0.009945922232572032, 0.009944864898953404,
0.009943797385920813, 0.009942719695671834, 0.009941631830424983,
0.009940533792419739, 0.009939425583916507, 0.009938307207196637,
0.009937178664562412, 0.009936039958337037, 0.009934891090864643,
0.009933732064510278, 0.009932562881659907, 0.009931383544720394,
0.009930194056119513, 0.009928994418305939, 0.00992778463374923,
0.009926564704939843, 0.009925334634389108, 0.009924094424629241,
0.009922844078213325, 0.00992158359771531, 0.009920312985730016,
0.009919032244873107, 0.009917741377781111, 0.009916440387111392,
0.009915129275542155, 0.009913808045772448, 0.00991247670052214,
0.009911135242531928, 0.009909783674563325, 0.009908421999398656,
0.009907050219841056, 0.009905668338714455, 0.009904276358863583,
0.009902874283153958, 0.00990146211447188, 0.009900039855724425,
0.009898607509839445, 0.009897165079765555, 0.009895712568472124,
0.009894249978949285, 0.00989277731420791, 0.00989129457727961,
0.00988980177121674, 0.009888298899092372, 0.009886785964000307,
0.00988526296905506, 0.009883729917391858, 0.009882186812166623,
0.009880633656555981, 0.009879070453757242, 0.009877497206988403,
0.009875913919488135, 0.00987432059451578, 0.009872717235351342,
0.009871103845295482, 0.009869480427669514, 0.009867846985815385,
0.009866203523095689, 0.009864550042893641, 0.009862886548613083,
0.009861213043678468, 0.00985952953153486, 0.009857836015647919,
0.009856132499503905, 0.00985441898660966, 0.009852695480492606,
0.00985096198470074, 0.009849218502802617, 0.009847465038387357,
0.009845701595064624, 0.00984392817646463, 0.009842144786238112,
0.00984035142805635, 0.009838548105611128, 0.009836734822614756,
0.009834911582800037, 0.00983307838992028, 0.009831235247749278,
0.009829382160081307, 0.009827519130731117, 0.009825646163533925,
0.009823763262345402, 0.009821870431041674, 0.009819967673519305,

0.009818054993695298, 0.009816132395507075, 0.009814199882912484,
0.009812257459889773, 0.009810305130437602, 0.009808342898575016,
0.009806370768341448, 0.009804388743796707, 0.009802396829020977,
0.009800395028114788, 0.009798383345199037, 0.00979636178441495,
0.009794330349924099, 0.009792289045908375, 0.009790237876569992,
0.009788176846131463, 0.00978610595883561, 0.009784025218945544,
0.009781934630744657, 0.009779834198536615, 0.00977772392664535,
0.009775603819415047, 0.009773473881210143, 0.00977133411641531,
0.009769184529435444, 0.00976702512469567, 0.00976485590664132,
0.009762676879737925, 0.009760488048471213, 0.009758289417347087,
0.009756080990891636, 0.009753862773651105, 0.009751634770191895,
0.009749396985100555, 0.009747149422983771, 0.009744892088468354,
0.00974262498620123, 0.00974034812084944, 0.009738061497100116,
0.009735765119660482, 0.009733458993257837, 0.009731143122639561,
0.009728817512573078, 0.009726482167845869, 0.009724137093265455,
0.009721782293659387, 0.009719417773875232], train_loss: 0.6915974617004395,
val_loss: 0.8637287020683289, val_acc: 0.69873046875
Epoch: 3, lat_lr: [0.00971704353878057, 0.009714659593262984,
0.009712265942230038, 0.009709862590609285, 0.00970744954334824,
0.009705026805414383, 0.00970259438179514, 0.009700152277497879,
0.009697700497549889, 0.009695239046998387, 0.009692767930910492,
0.009690287154373223, 0.009687796722493482, 0.009685296640398052,
0.00968278691323358, 0.009680267546166566, 0.00967773854438336,
0.009675199913090139, 0.009672651657512911, 0.009670093782897488,
0.009667526294509492, 0.00966494919763433, 0.00966236249757719,
0.009659766199663033, 0.009657160309236572, 0.00965454483166227,
0.009651919772324326, 0.009649285136626667, 0.009646640929992926,
0.009643987157866447, 0.00964132382571026, 0.009638650939007079,
0.009635968503259283, 0.009633276523988911, 0.009630575006737647,
0.009627863957066813, 0.009625143380557349, 0.00962241328280981,
0.009619673669444355, 0.009616924546100723, 0.009614165918438236,
0.009611397792135783, 0.009608620172891802, 0.009605833066424275,
0.009603036478470715, 0.009600230414788156, 0.009597414881153133,
0.009594589883361681, 0.009591755427229313, 0.009588911518591016,
0.009586058163301235, 0.009583195367233864, 0.009580323136282225,
0.009577441476359073, 0.009574550393396565, 0.009571649893346256,
0.009568739982179095, 0.009565820665885395, 0.009562891950474838,
0.00955995384197645, 0.009557006346438596, 0.009554049469928962,
0.00955108321853455, 0.009548107598361657, 0.009545122615535872,
0.00954212827620205, 0.009539124586524315, 0.009536111552686034,
0.009533089180889812, 0.009530057477357474, 0.009527016448330063,
0.009523966100067813, 0.009520906438850145, 0.009517837470975647,
0.00951475920276207, 0.009511671640546313, 0.009508574790684402,
0.009505468659551482, 0.009502353253541814, 0.009499228579068738,
0.009496094642564686, 0.009492951450481148, 0.009489799009288678,
0.009486637325476855, 0.0094834664055543, 0.009480286256048636,
0.009477096883506496, 0.009473898294493488, 0.009470690495594205,
0.009467473493412188, 0.009464247294569933, 0.009461011905708862,

0.00945776733348932, 0.009454513584590554, 0.009451250665710703,
0.009447978583566781, 0.009444697344894667, 0.009441406956449095,
0.009438107425003624, 0.009434798757350642, 0.009431480960301344,
0.009428154040685718, 0.009424818005352529, 0.009421472861169313,
0.009418118615022349, 0.009414755273816664, 0.009411382844476,
0.00940800133394281, 0.009404610749178243, 0.009401211097162125,
0.009397802384892951, 0.009394384619387864, 0.009390957807682648,
0.009387521956831707, 0.009384077073908053, 0.009380623166003289,
0.009377160240227604, 0.009373688303709741, 0.009370207363597003,
0.009366717427055221, 0.009363218501268743, 0.009359710593440432,
0.009356193710791634, 0.009352667860562174, 0.009349133050010333,
0.009345589286412844, 0.009342036577064866, 0.009338474929279975,
0.009334904350390147, 0.009331324847745745, 0.0093277364287155,
0.009324139100686503, 0.009320532871064179, 0.00931691774727228,
0.009313293736752869, 0.009309660846966302, 0.009306019085391214,
0.009302368459524502, 0.009298708976881313, 0.009295040644995027,
0.009291363471417236, 0.009287677463717739, 0.009283982629484521,
0.009280278976323731, 0.00927656651185968, 0.009272845243734812,
0.009269115179609698, 0.009265376327163014, 0.00926162869409153,
0.00925787228811009, 0.009254107116951597, 0.009250333188367,
0.009246550510125275, 0.009242759090013408, 0.009238958935836386,
0.00923515005541717, 0.009231332456596693, 0.009227506147233822,
0.009223671135205369, 0.009219827428406054, 0.009215975034748499,
0.009212113962163206, 0.009208244218598545, 0.00920436581202074,
0.009200478750413837, 0.009196583041779711, 0.009192678694138035,
0.009188765715526261, 0.009184844113999615, 0.009180913897631069,
0.009176975074511333, 0.009173027652748833, 0.009169071640469695,
0.009165107045817734, 0.009161133876954428, 0.009157152142058905,
0.009153161849327933, 0.00914916300697589, 0.009145155623234761,
0.009141139706354107, 0.009137115264601063, 0.009133082306260304,
0.009129040839634048, 0.009124990873042018, 0.009120932414821443,
0.009116865473327027, 0.009112790056930938, 0.009108706174022797,
0.009104613833009645, 0.009100513042315942, 0.009096403810383536,
0.009092286145671659, 0.009088160056656898, 0.00908402555183318,
0.009079882639711767, 0.009075731328821213, 0.009071571627707378,
0.00906740354493338, 0.0090632270890796, 0.009059042268743652,
0.00905484909254037, 0.009050647569101792, 0.009046437707077134,
0.00904221951513278, 0.009037993001952266, 0.009033758176236252,
0.009029515046702515, 0.009025263622085921, 0.009021003911138418,
0.00901673592262901, 0.009012459665343735, 0.009008175148085662,
0.009003882379674862, 0.008999581368948387, 0.008995272124760263,
0.008990954655981456, 0.008986628971499875, 0.008982295080220333,
0.008977952991064542, 0.008973602712971087, 0.008969244254895412,
0.008964877625809801, 0.008960502834703355, 0.008956119890581985,
0.008951728802468379, 0.00894732957940199, 0.008942922230439023,
0.008938506764652406, 0.008934083191131776, 0.008929651518983468,
0.008925211757330477, 0.008920763915312458, 0.008916308002085704,
0.008911844026823114, 0.008907371998714187, 0.008902891926965004,

0.008898403820798201, 0.00889390768945295, 0.00888940354218495,
0.0088848913882664, 0.00888037123698598, 0.008875843097648828,
0.008871306979576537, 0.008866762892107118, 0.00886221084459499,
0.008857650846410955, 0.008853082906942188, 0.00884850703559221,
0.008843923241780863, 0.008839331534944312, 0.008834731924534997,
0.008830124420021642, 0.00882550903088921, 0.008820885766638902,
0.008816254636788128, 0.008811615650870493, 0.00880696881843577,
0.00880231414904989, 0.008797651652294914, 0.008792981337769012,
0.008788303215086456, 0.008783617293877588, 0.008778923583788803,
0.008774222094482531, 0.008769512835637218, 0.0087647958169473,
0.00876007104812319, 0.008755338538891258, 0.008750598298993804,
0.008745850338189042, 0.008741094666251087, 0.008736331292969919,
0.008731560228151377, 0.008726781481617133, 0.008721995063204674,
0.008717200982767275, 0.008712399250173992, 0.008707589875309625,
0.008702772868074712, 0.008697948238385505, 0.008693115996173938,
0.008688276151387626, 0.008683428713989829, 0.00867857369395944,
0.008673711101290962, 0.00866884094599448, 0.008663963238095656,
0.008659077987635697, 0.008654185204671335, 0.00864928489927481,
0.008644377081533846, 0.008639461761551637, 0.008634538949446812,
0.008629608655353433, 0.008624670889420958, 0.008619725661814228,
0.008614772982713443, 0.008609812862314149, 0.008604845310827204,
0.008599870338478763, 0.008594887955510264, 0.008589898172178393,
0.008584900998755075, 0.008579896445527447, 0.008574884522797841,
0.008569865240883752, 0.008564838610117833, 0.008559804640847858,
0.008554763343436717, 0.008549714728262372, 0.008544658805717866,
0.008539595586211273, 0.008534525080165691, 0.008529447298019221,
0.008524362250224937, 0.008519269947250876, 0.00851417039958001,
0.008509063617710217, 0.008503949612154277, 0.008498828393439834,
0.00849369997210939, 0.008488564358720259, 0.008483421563844573,
0.008478271598069246, 0.008473114471995951, 0.0084679501962411,
0.008462778781435828, 0.00845760023822596, 0.008452414577272005,
0.008447221809249113, 0.008442021944847075, 0.008436814994770283,
0.008431600969737718, 0.008426379880482927, 0.008421151737753999,
0.008415916552313541, 0.00841067433493866, 0.008405425096420938,
0.00840016884756641, 0.008394905599195547, 0.008389635362143223,
0.0083843581472587, 0.008379073965405607, 0.008373782827461916,
0.008368484744319914, 0.008363179726886188, 0.008357867786081604,
0.00835254893284127, 0.008347223178114536, 0.008341890532864951,
0.00833655100807025, 0.008331204614722336, 0.008325851363827241,
0.008320491266405125, 0.008315124333490237, 0.008309750576130894,
0.008304370005389465, 0.00829898263234235, 0.008293588468079946,
0.008288187523706629, 0.008282779810340738, 0.008277365339114542,
0.008271944121174225, 0.00826651616767986, 0.008261081489805379,
0.008255640098738566, 0.00825019200568102, 0.008244737221848134,
0.008239275758469083, 0.008233807626786783, 0.008228332838057883,
0.008222851403552734, 0.00821736333455537, 0.00821186864236348,
0.00820636733828839, 0.008200859433655033, 0.008195344939801936,
0.008189823868081186, 0.008184296229858414, 0.008178762036512764,

0.008173221299436878, 0.00816767403003687, 0.0081621202397323,
0.00815655993995615, 0.008150993142154806, 0.008145419857788026,
0.008139840098328925, 0.00813425387526395, 0.00812866120009284,
0.008123062084328638, 0.00811745653949763], train_loss: 0.6257481575012207,
val_loss: 0.897084653377533, val_acc: 0.696582019329071
Epoch: 4, lat_lr: [0.00811184457713934, 0.008106226208806506,
0.008100601446065048, 0.008094970300494058, 0.00808933278368576,
0.008083688907245493, 0.008078038682791696, 0.008072382121955867,
0.008066719236382554, 0.008061050037729325, 0.008055374537666738,
0.008049692747878331, 0.008044004680060585, 0.008038310345922907,
0.0080326097571876, 0.008026902925589847, 0.008021189862877682,
0.008015470580811966, 0.00800974509116636, 0.008004013405727307,
0.007998275536294005, 0.00799253149467838, 0.007986781292705067,
0.007981024942211378, 0.007975262455047288, 0.007969493843075402,
0.007963719118170932, 0.007957938292221677, 0.007952151377127993,
0.007946358384802772, 0.007940559327171418, 0.00793475421617182,
0.007928943063754327, 0.007923125881881725, 0.007917302682529216,
0.007911473477684384, 0.007905638279347182, 0.007899797099529891,
0.00789394995025712, 0.007888096843565753, 0.007882237791504947,
0.007876372806136093, 0.007870501899532801, 0.007864625083780863,
0.007858742370978241, 0.007852853773235037, 0.007846959302673465,
0.007841058971427832, 0.007835152791644507, 0.007829240775481895,
0.007823322935110425, 0.007817399282712512, 0.007811469830482531,
0.007805534590626801, 0.007799593575363556, 0.007793646796922913,
0.007787694267546862, 0.0077817359994892255, 0.00777577200501564,
0.007769802296403535, 0.007763826885942097, 0.007757845785932254,
0.007751859008686646, 0.007745866566529601, 0.007739868471797103,
0.007733864736836782, 0.00772785537400787, 0.00772184039568119,
0.007715819814239123, 0.007709793642075584, 0.0077037618915959975,
0.007697724575217273, 0.007691681705367774, 0.007685633294487302,
0.0076795793550270604, 0.0076735198994496365, 0.007667454940228972,
0.007661384489850336, 0.007655308560810306, 0.0076492271656167345,
0.0076431403167887295, 0.007637048026856624, 0.007630950308361951,
0.007624847173857419, 0.0076187386359068895, 0.007612624707085342,
0.0076065053999788594, 0.007600380727184589, 0.00759425070131073,
0.007588115334976502, 0.0075819746408121134, 0.007575828631458741,
0.007569677319568509, 0.00756352071780445, 0.007557358838840493,
0.007551191695361428, 0.007545019300062878, 0.007538841665651285,
0.007532658804843871, 0.007526470730368618, 0.007520277454964245,
0.0075140789913801685, 0.007507875352376495, 0.007501666550723977,
0.007495452599204004, 0.007489233510608558, 0.0074830092977401985,
0.007476779973412039, 0.007470545550447708, 0.007464306041681336,
0.0074580614599575185, 0.007451811818131298, 0.00744555712906813,
0.007439297405643864, 0.00743303266074471, 0.007426762907267217,
0.007420488158118244, 0.007414208426214932, 0.007407923724484685,
0.0074016340658651335, 0.00739533946330411, 0.007389039929759631,
0.007382735478199861, 0.007376426121603085, 0.007370111872957692,
0.007363792745262137, 0.007357468751524919, 0.007351139904764557,

0.007344806218009559, 0.007338467704298396, 0.007332124376679477,
0.007325776248211119, 0.007319423319615245, 0.007313065641008749,
0.00730670318844068, 0.007300335987355005, 0.00729396405085919,
0.007287587392070445 4, 0.007281206024115704, 0.007274819960131597,
0.007268429213264415, 0.007262033796670094, 0.007255633723514183,
0.007249229006971815, 0.007242819660227684, 0.007236405696476014 5,
0.007229987128920533, 0.007223563970774449, 0.007217136235260417,
0.007210703935610518, 0.007204267085066227, 0.007197825696878385 4,
0.007191379784307181, 0.007184929360622111, 0.00717847443910196,
0.007172015033034772, 0.007165551155717825, 0.0071590828204576,
0.007152610040569751, 0.007146132829379089, 0.007139651200219543,
0.007133165166434135, 0.007126674741374959, 0.007120179938403147,
0.007113680770888843, 0.007107177252211174, 0.007100669395758228,
0.007094157214927018, 0.007087640723123467, 0.00708111993376236,
0.007074594860267342, 0.00706806551607087, 0.007061531914614194,
0.007054994069347327 6, 0.007048451993729022, 0.007041905701226735,
0.007035355205316608, 0.007028800519483435, 0.007022241657220631,
0.007015678632030214, 0.00700911145742277, 0.007002540146917424,
0.00699596471404182, 0.006989385172332085, 0.006982801535332806,
0.006976213816596998 6, 0.006969622029686084, 0.006963026188169855,
0.006956426305626454, 0.006949822395642339, 0.006943214471812262,
0.006936602547739236, 0.006929986637034509, 0.006923366753317539,
0.006916742910215956 5, 0.006910115121365548, 0.006903483400410221,
0.006896847761001976, 0.006890208216800885, 0.006883564781475051,
0.006876917468700594, 0.006870266292161612, 0.006863611265550158,
0.006856952402566212, 0.006850289716917649, 0.006843623222320217,
0.0068369529324975, 0.0068302788611809, 0.006823601022109599,
0.006816919429030546 4, 0.006810234095698389, 0.006803545035875516,
0.006796852263331959, 0.0067901557918454, 0.006783455635201137,
0.006776751807192053, 0.006770044321618589, 0.006763333192288715,
0.006756618433017904, 0.006749900057629097, 0.006743178079952686,
0.006736452513826473, 0.006729723373095647, 0.006722990671612761,
0.006716254423237692, 0.006709514641837624, 0.006702771341287007,
0.006696024535467544 5, 0.006689274238268148 5, 0.006682520463584921,
0.006675763225321124, 0.00666900253738715, 0.006662238413700489,
0.006655470868185706, 0.006648699914774412, 0.00664192556740523,
0.006635147840023773, 0.00662836674658261, 0.0066215823010412 4,
0.006614794517366062, 0.006608003409530348, 0.006601208991514213,
0.006594411277304585, 0.006587610280895178, 0.006580806016286464,
0.006573998497485643, 0.006567187738506609 5, 0.006560373753369935,
0.006553556556102829, 0.006546736160739111, 0.006539912581319188,
0.006533085831890026, 0.006526255926505094, 0.00651942287922439,
0.006512586704114361, 0.006505747415247898, 0.006498905026704297,
0.006492059552569238, 0.006485211006934759 5, 0.006478359403899211,
0.006471504757567246, 0.006464647082049776, 0.006457786391463952 5,
0.006450922699331316, 0.006444056021586848, 0.006437186370560785,
0.006430313760996746, 0.006423438207042624, 0.006416559722852375,
0.006409678322585982 5, 0.006402794020409439, 0.006395906830494705,

0.0063890167670196925, 0.006382123844168223, 0.006375228076130008,
0.00636832947710061 3, 0.006361428061281433, 0.006354523842879662,
0.006347616836108264, 0.006340707055185939, 0.006333794514337104,
0.006326879227791854 5, 0.006319961209785936, 0.006313040474560719 5,
0.006306117036363169, 0.006299190909445817, 0.006292262108066723,
0.006285330646489457, 0.006278396538983065, 0.00627145979982204,
0.00626452044328629, 0.006257578483661115, 0.006250633935237171,
0.006243686812310443, 0.006236737129182218, 0.006229784900159051,
0.00622283013955274, 0.006215872861680296, 0.006208913080863903,
0.006201950811430911, 0.006194986067713783, 0.006188018864050081,
0.006181049214782425 4, 0.006174077134258478, 0.006167102636830899 5,
0.00616012573685733, 0.006153146448700355, 0.006146164786727472,
0.006139180765311072, 0.006132194398828397, 0.006125205701661523,
0.006118214688197316, 0.006111221372827415, 0.0061042257699482,
0.006097227893960755 5, 0.006090227759270847, 0.00608322538028889,
0.00607622077142992, 0.006069213947113563, 0.006062204921764005,
0.006055193709809965, 0.00604818032568466, 0.00604116478382578,
0.006034147098675461, 0.006027127284680245, 0.006020105356291057,
0.006013081327963179, 0.006006055214156216, 0.0059990270293340 6,
0.005991996787964871, 0.005984964504521044, 0.005977930193479171,
0.005970893869320026, 0.005963855465285235, 0.0059568152395936 9,
0.005949772963008642, 0.005942728731270546, 0.005935682558880596,
0.005928634460343979, 0.0059215844501698 5, 0.005914532542871296,
0.005907478752965311 5, 0.005900423094972768, 0.005893365583418377,
0.005886306232830672, 0.005879245057741969, 0.00587218207268834,
0.005865117292209584, 0.005858050730849196, 0.005850982403154335,
0.005843912323675799, 0.005836840506967993, 0.005829766967588894,
0.005822691720100026, 0.005815614779066434, 0.005808536159056646,
0.005801455874642645, 0.005794373940399842, 0.005787290370907046,
0.005780205180746428, 0.005773118384503 5, 0.005766029996767077 5,
0.005758940032129 25, 0.005751848505185361, 0.005744755430533961,
0.005737660822776792, 0.005730564696518751, 0.005723467066367 86,
0.005716367946935239, 0.005709267352835073, 0.005702165298684581,
0.005695061799103987, 0.005687956868716495, 0.005680850522148251,
0.005673742774028317, 0.005666633638988641, 0.005659523131664026,
0.005652411266692097, 0.005645298058713279, 0.005638183522370757,
0.005631067672310452, 0.005623950523180994, 0.005616832089633 68,
0.005609712386322455, 0.005602591427903878, 0.005595469229037089,
0.005588345804383787, 0.005581221168608186, 0.005574095336377004,
0.005566968322359 41, 0.005559840141227017], train_loss: 0.5475258827209473,
val_loss: 0.5482883453369141, val_acc: 0.813183605670929
Epoch: 5, lat_lr: [0.005552710807653834, 0.005545580336316243 6,
0.005538448741892971, 0.005531316039065054, 0.005524182242515811,
0.005517047366930813, 0.005509911426997 85, 0.005502774443740690 95,
0.005495636412850129, 0.005488497368021787, 0.005481357317618255 6,
0.005474216276337981, 0.005467074258881447, 0.005459931279951147,
0.005452787354251556, 0.005445642496489093, 0.005438496721372102,
0.005431350043610812, 0.005424202477917 31, 0.005417054039005512,

0.005409904741591129, 0.0054027546003916445, 0.005395603630126274,
0.00538451845515942, 0.005381299261283249, 0.005374145892152443,
0.00536691752849384, 0.005359836858101522, 0.0053526812226378576,
0.0053455248611889205, 0.005338367788486734, 0.005331210019264784,
0.00532405156825798 7, 0.005316892450202674, 0.005309732679836535,
0.005302572271898616, 0.005295411241129265, 0.005288249602270118,
0.005281087370064063, 0.0052739245592552065, 0.005266761184588847,
0.005259597260811443, 0.005252432802670588, 0.005245267824914971,
0.00523810234229435, 0.005230936369559526, 0.005223769921462306,
0.005216603012755478, 0.005209435658192775, 0.005202267872528852,
0.005195099670519247, 0.005187931066920358, 0.005180762076489409,
0.00517359271398442, 0.005166422994164178, 0.005159252931788204,
0.005152082541616725, 0.005144911838410645, 0.005137740836931507,
0.005130569551941472, 0.005123397998203285, 0.005116226190480242,
0.005109054143536163, 0.005101881872135362, 0.005094709391042611,
0.005087536715023118, 0.0050803638588424885, 0.005073190837266703,
0.005066017665062078, 0.005058844356995246, 0.005051670927833111,
0.005044497392342833, 0.005037323765291791, 0.0050301500614475456,
0.005022976295577821, 0.005015802482450 4726, 0.005008628636833444,
0.005001454773494751, 0.004994280907202446, 0.004987107052724587,
0.004979933224829207, 0.004972759438284285, 0.004965585707857715,
0.004958412048317273, 0.004951238474430592, 0.004944065000965129,
0.0049368916426881316, 0.004929718414366613, 0.004922545330767315,
0.004915372406656686, 0.004908199656800845, 0.004901027095965548,
0.004893854738916169, 0.004886682600417657, 0.004879510695234513,
0.004872339038130759, 0.004865167643869902, 0.004857996527214913,
0.004850825702928188, 0.004843655185771524, 0.004836484990506083,
0.004829315131892366, 0.004822145624690179, 0.004814976483658608,
0.004807807723555982, 0.004800639359139849, 0.00479347140516694,
0.00478630387639314, 0.004779136787573462, 0.00477197015346201,
0.004764803988811956, 0.004757638308375501, 0.004750473126903852,
0.004743308459147188, 0.004736144319854632, 0.004728980723774216,
0.004721817685652856, 0.004714655220236319, 0.004707493342269193,
0.004700332066494857, 0.0046931714076554505, 0.004686011380491841,
0.004678851999743597, 0.004671693280148958, 0.0046645352364447995,
0.004657377883366608, 0.004650221235648446, 0.004643065308022926,
0.004635910115221176, 0.004628755671972814, 0.004621601993005913,
0.004614449093046975, 0.0046072969868208935, 0.0046001456890509326,
0.004592995214458691, 0.0045858455777640725, 0.004578696793685256,
0.004571548876938664, 0.004564401842238935, 0.004557255704298892,
0.004550110477829512, 0.0045429661775398915, 0.00453582281813723,
0.004528680414326777, 0.004521538980811829, 0.004514398532293671,
0.004507259083471571, 0.004500120649042736, 0.004492983243702281,
0.004485846882143207, 0.004478711579056365, 0.004471577349130427,
0.0044644442070518545, 0.004457312167504871, 0.004450181245171429,
0.004443051454731187, 0.004435922810861462, 0.004428795328237222,
0.004421669021531036, 0.004414543905413059, 0.00440741999455099,
0.00440029730361005, 0.004393175847252948, 0.00438605564013985,

0.004378936696928351, 0.004371819032273445, 0.004364702660827495,
0.004357587597240197, 0.004350473856158561, 0.00434336145222687,
0.004336250400086658, 0.004329140714376669, 0.004322032409732846,
0.004314925500788275, 0.0043078200021731804, 0.004300715928514879,
0.004293613294437751, 0.0042865121145632176, 0.0042794124035097055,
0.004272314175892617, 0.004265217446324299, 0.00425812229414019,
0.004251028539767924, 0.004243936391989026, 0.004236845800677152,
0.004229756780428935, 0.004222669345837766, 0.004215583511493777,
0.0042084992919838035, 0.004201416701891358, 0.004194335755796596,
0.004187256468276292, 0.004180178853903804, 0.0041731029272490455,
0.004166028702878459, 0.004158956195354977, 0.00415188541923800 5,
0.004144816389083376, 0.004137749119443336, 0.0041306836248665,
0.004123619919897839, 0.004116558019078625, 0.004109497936946432,
0.00410243968803508, 0.004095383286874615, 0.004088328747991288,
0.004081276085907504, 0.004074225315141816, 0.004067176450208874,
0.0040601295056194135, 0.004053084495880208, 0.004046041435494055,
0.004039000338959734 5, 0.004031961220771987, 0.004024924095421476,
0.004017888977394771, 0.004010855881174299, 0.0040038248212383295,
0.003996795812060945, 0.003989768868111997, 0.003982744003857093,
0.003975721233757553, 0.003968700572270394, 0.003961682033848282,
0.003954665632939525, 0.003947651383988017, 0.003940639301433233 5,
0.003933629399710183, 0.003926621693249389, 0.003919616196476853,
0.003912612923814027, 0.003905611889677792, 0.003898613108480408,
0.003891616594629509 3, 0.003884622362528053, 0.003877630426574307 5,
0.003870640801161806, 0.003863653500679334 7, 0.003856668539510883 6,
0.003849685932035636, 0.003842705692627922, 0.003835727835657205,
0.003828752375488035, 0.003821779326480035, 0.003814808702987864 4,
0.003807840519361182 4, 0.003800874789944634, 0.003793911529077804 6,
0.003786950751095206, 0.003779992470326228 7, 0.003773036701095133 3,
0.003766083457721, 0.003759132754517718 4, 0.003752184605793941,
0.003745239025853068, 0.003738296028993206, 0.003731355629507149 5,
0.003724417841682344, 0.003717482679800854 3, 0.003710550158139348,
0.003703620290969048 7, 0.003696693092555723 5, 0.003689768577159637 7,
0.003682846759035542, 0.003675927652432624 7, 0.003669011271594502 8,
0.003662097630759171 5, 0.003655186744158996, 0.003648278626020661,
0.003641373290565161, 0.003634470752007759, 0.003627571024557956 4,
0.003620674122419474 7, 0.003613780059790212, 0.003606888885086228 7,
0.003600000509821702, 0.003593115050848914 5, 0.003586232488118207,
0.003579352835797965 6, 0.003572476108050578 7, 0.003565602319032420 5,
0.003558731482893808 6, 0.003551863613778989 6, 0.003544998725826093 4,
0.003538136833167119 4, 0.003531277949927902 6, 0.003524422090228075,
0.003517569268181051 5, 0.003510719497893988, 0.003503872793467765,
0.003497029168996942 8, 0.003490188638569751 3, 0.003483351216268041 2,
0.003476516916167273, 0.003469685752336474, 0.003462857738838221,
0.003456032889728597 6, 0.003449211219057182, 0.003442392740867001 7,
0.003435577469194517 4, 0.003428765418069588 3, 0.003421956601515439 3,
0.003415151033548644 3, 0.003408348728179082, 0.003401549699409917 4,
0.003394753961237571 7, 0.003387961527651694, 0.003381172412635125,

0.0033743866301638795, 0.0033676041942071054, 0.0033608251187270677,
0.0033540494176791135, 0.0033472771050116357, 0.0033405081946660626,
0.0033337427005768086, 0.0033269806366712645, 0.00332320222016869751,
0.0033313466855085507, 0.0033306715165224644, 0.0033299966961186136,
0.0033293222256861771, 0.0033286481066136443, 0.0033279743402866046,
0.0033273009280983251, 0.0033266278714288833, 0.0033259551716658877,
0.0033252828301941705 7, 0.0033246108483977794, 0.0033239392276600615,
0.0033232679693636089 6, 0.0033225970748902686 3, 0.0033219265456211374,
0.0033212563829365614, 0.0033205865882161311 5, 0.0033199171628386805 5,
0.0033192481081822824, 0.0033185794256242471 3, 0.0033179111165411182 2,
0.0033172431823086711, 0.0033165756243019095, 0.0033159084438950619,
0.0033152416424615805 5, 0.0033145752213741365, 0.0033139091820046187,
0.0033132435257241295, 0.0033125782539029834 3, 0.0033119133679107028 5,
0.0033112488691160165, 0.0033105847588868553, 0.0033099210385903511 6,
0.0033092577095928324 7, 0.0033085947732598224 6, 0.0033079322309560363 2,
0.0033072700840453773 2, 0.0033066083338909357, 0.0033059469818549840 3,
0.0033052860292989761 5, 0.0033046254775835427 6, 0.0033039653280684901 8,
0.0033033055821127959 2, 0.0033026462410746076 6, 0.0033019873063112385 3,
0.0033013287791791663, 0.0033006706610340287, 0.0033000129532306221 5,
0.0029935565712289787, 0.0029869877406396, 0.0029804230540606234,
0.0029738625250060511, 0.0029673061669813341, 0.0029607539934833286,
0.0029542060180002857, 0.0029476622540118053, 0.0029411227149888264,
0.0029345874143935817, 0.0029280563656795871, 0.0029215295822915981,
0.0029150070776655971, 0.0029084888652287527, 0.0029019749583993971,
0.0028954653705870054, 0.0028888960115192154, 0.0028828245920560651 7,
0.0028759626552127761, 0.0028694704773847045, 0.0028629826854870335,
0.0028564992928754701, 0.0028500203128966734, 0.0028434545758888211,
0.0028370756441785444, 0.0028306099820869924], train_loss: 0.45073404908180237,
val_loss: 0.41739627718925476, val_acc: 0.8578125238418579
Epoch: 6, lat_lr: [0.0028241487859237131, 0.0028176920689896635,
0.0028111239844576589, 0.0028047921259669761, 0.0027983489264340466,
0.0027919102592417096, 0.0027854761376445501, 0.0027790465748877965,
0.0027726215842072857, 0.0027662011788829451, 0.0027597853719712761,
0.0027533741768402905, 0.0027469676066345186, 0.0027405656745424741,
0.0027341683937431136, 0.0027277757757740631, 0.0027213878386903906,
0.0027150045907469666, 0.0027086260467160473, 0.0027022522197284495,
0.0026958831229052776, 0.0026895187693578951, 0.0026831591721879077,
0.0026768043444871216, 0.0026704542993375331, 0.0026641090498112853,
0.0026577686089706561, 0.0026514329898680181, 0.0026451021022055458,
0.0026387762690365645, 0.0026324551933627634, 0.0026261389159153691 2,
0.0026198276765615324, 0.0026135212614289997, 0.0026072197591216561,
0.0026009231826117215, 0.0025946315448612141, 0.0025883448588222141,
0.0025820631374362663, 0.0025757863936349271, 0.0025695146403943941 5,
0.0025632478904607883, 0.0025569861568996411, 0.0025507294525463522,
0.0025444777902809148, 0.0025382311829729461, 0.0025319896434481661,
0.0025257531846558296, 0.0025195218193337781, 0.0025132955603433351,
0.0025070744205018263, 0.0025008584126160311, 0.0024946475494821721,
0.0024884418438838741, 0.0024822413086021504, 0.0024760459563953653,

0.0024698558000192166, 0.002463670852216709, 0.0024574911257201166,
0.0024513166332509736, 0.0024451473875200325, 0.0024389834012272502,
0.0024328246870617518, 0.0024266712577018157, 0.0024205231258148325,
0.0024143804304057296, 0.0024082428805074762, 0.0024021106641501834,
0.0023959838259621272, 0.0023898623710682510, 0.0023837462894217803,
0.0023776355936132223, 0.0023715302962220063, 0.0023654304098164394,
0.0023593359946953696, 0.0023532469201797820, 0.0023471633420295170,
0.0023410852250264963, 0.0023350125816830832, 0.0023289454245003626,
0.0023228837659681350, 0.0023168276185648725, 0.0023107769947577094,
0.0023047319070024076, 0.0022986923677433280, 0.0022926583894134164,
0.0022866299844341630, 0.0022806071652155925, 0.0022745899441562246,
0.0022685783336430600, 0.0022625723460515440, 0.0022565719937455526,
0.0022505772890773540, 0.0022445882443875990, 0.0022386048720052783,
0.0022326271842477100, 0.0022266551934200120, 0.0022206889118175674,
0.0022147283517210156, 0.0022087735254012080, 0.0022028244451167012,
0.0021968811231142162, 0.0021909435716286263, 0.0021850118028829190,
0.0021790858290881843, 0.0021731656624357650, 0.0021672513151363014,
0.0021613427993415780, 0.0021554401271726270, 0.0021495433109306390,
0.0021436523623626040, 0.0021377672943719995, 0.0021318881183473516,
0.0021260148466336237, 0.0021201474913214767, 0.0021142860644894007,
0.0021084305782036734, 0.0021025810445183510, 0.0020967374754752273,
0.0020908998831038240, 0.0020850682792794130, 0.0020792426764327165,
0.0020734230861304353, 0.0020676095204946680, 0.0020618019914931730,
0.0020560005110812755, 0.0020502050912018545, 0.0020444157437853067,
0.0020386324807495355, 0.0020328553139999112, 0.0020270842554292613,
0.0020213193169178330, 0.0020155605103332790, 0.0020098078475306296,
0.0020040613403522620, 0.0019983210006278883, 0.0019925868401745182,
0.0019868588707964464, 0.0019811371042852170, 0.0019754215524196110,
0.0019697122269656106, 0.0019640091396763847, 0.0019583123022922553,
0.0019526217265406847, 0.0019469374241362393, 0.0019412594067805744,
0.0019355876861624085, 0.0019299222739574923, 0.0019242631818285963,
0.0019186104214254738, 0.0019129640043848517, 0.0019073239423303900,
0.0019016902468726740, 0.0018960629296091757, 0.0018904420021242452,
0.0018848274759890695, 0.0018792193627616657, 0.0018736176739868441,
0.0018680224211961916, 0.0018624336159080493, 0.0018568512696274780,
0.0018512753938462500, 0.0018457060000004280, 0.0018401430996822681,
0.0018345867042163560, 0.0018290368250834248, 0.0018234934737084035,
0.0018179566615027882, 0.0018124263998646095, 0.0018069027001784178,
0.0018013855738152498, 0.0017958750321326138, 0.0017903710864744641,
0.0017848737481771150, 0.0017793830285395109, 0.0017738989388826260,
0.0017684214904900187, 0.0017629506946375125, 0.0017574865625872426,
0.0017520291055876194, 0.0017465783348733191, 0.0017411342616652466,
0.0017356968971705246, 0.0017302662525824654, 0.0017248423390805424,
0.0017194251678303790, 0.0017140147499837134, 0.0017086110966783867,
0.0017032142190383087, 0.0016978241281734470, 0.0016924408351797913,
0.0016870643511393457, 0.0016816946871200877, 0.0016763318541759644,
0.0016709758633468529, 0.0016656267256585497, 0.0016602844521227446,
0.0016549490537369910, 0.0016496205415414696, 0.0016442989263508460,

0.0016389842192431905, 0.0016336764311498182, 0.001628375572981538,
0.0016230816556506448, 0.001617794690055155, 0.001612514687078764 8,
0.0016072416575908445, 0.0016019756124464044, 0.0015967165624860783,
0.0015914645185361022, 0.0015862194914082845, 0.0015809814918999944,
0.0015757505307941277, 0.0015705266188590 99, 0.0015653097668488032,
0.0015600999855026 09, 0.0015548972855453232, 0.0015497016776871 81,
0.0015445131726238115, 0.0015393317810362286, 0.0015341575135907957,
0.0015289903809392 16, 0.0015238303937185047, 0.0015186775625509624,
0.0015135318980441645, 0.0015083934107909282, 0.0015032621113693003,
0.0014981380103425259, 0.0014930211182590369, 0.0014879114456524188,
0.0014828090030414012, 0.0014777138009298251, 0.0014726258498066 31,
0.0014675451601458274, 0.0014624717424064 78, 0.0014574056070326778,
0.0014523467644535248, 0.0014472952250831105, 0.0014422509993204866,
0.0014372140975496544, 0.0014321845301395327, 0.0014271623074439469,
0.0014221474398015976, 0.0014171399375360492, 0.0014121398109557,
0.0014071470703537683, 0.0014021617260082625, 0.0013971837881819726,
0.0013922132671224 33, 0.0013872501730619166, 0.0013822945162174063,
0.0013773463067905 71, 0.0013724055549677527, 0.0013674722709199373,
0.0013625464648027417, 0.0013576281467563 85, 0.0013527173269056757,
0.0013478140153599812, 0.0013429182222132181, 0.0013380299575438 2,
0.0013331492314147281, 0.0013282760538733593, 0.0013233410434951595,
0.0013185523846657569, 0.0013137019130165812, 0.0013088590299892091,
0.0013040237455531 54, 0.0012991960696622925, 0.0012943760122548322,
0.0012895635832533038, 0.0012847587925645279, 0.0012799616500796076,
0.0012751721656738945, 0.0012703903492069 82, 0.0012656162105226 72,
0.0012608497594489665, 0.0012560910057980395, 0.0012513399593662175,
0.0012465966299339648, 0.0012418610272658553, 0.0012371331611105607,
0.0012324130412008224, 0.0012277006772534395, 0.0012229960789692407,
0.0012182992560330727, 0.0012136102181137714, 0.0012089289748641 51,
0.0012042555359209747, 0.0011995899109049 46, 0.0011949321094206758,
0.0011902821410566743, 0.0011856400153853275, 0.0011810057419628706,
0.0011763793303293 81, 0.0011717607900087457, 0.0011671501305086545,
0.0011625473613205667, 0.0011579524919197053, 0.0011533655317650 25,
0.0011487864902992042, 0.0011442153769486141, 0.0011396522011233 11,
0.0011350969722170067, 0.0011305496996070548, 0.0011260103926544328,
0.0011214790607037145, 0.0011169557130830633, 0.0011124403591041 99,
0.0011079330080623927, 0.0011034336692364347, 0.0010989423518886266,
0.0010944590652647518, 0.0010899838185940668, 0.0010855166210892 73,
0.0010810574819465 06, 0.0010766064103453 07, 0.0010721634154486149,
0.0010677285064027 4, 0.0010633016923373446, 0.0010588828982365432,
0.0010544723855833 16, 0.0010500699110706157, 0.0010456756789022 32,
0.0010412893650882963, 0.0010368911311694232, 0.0010325414167206542,
0.0010281796891633 87, 0.0010238261380014 48, 0.0010194807721970154,
0.0010151436006954 24, 0.0010108146324251336, 0.0010064938762977217,
0.0010021813412078 6, 0.000997877036033292, 0.0009935809696348247,
0.0009892931508563008, 0.000985013588524588, 0.0009807422914495528,
0.0009764792684240521, 0.0009722245282239056, 0.0009679780796078855,
0.0009637399313176912, 0.0009595100920779 4, 0.0009552885705961385,

0.0009510753755626752, 0.000946870515650797, 0.0009426739995165888,
0.000938484858357989644, 0.0009343060331196383, 0.0009301346000831172,
0.0009259715452766742, 0.0009218168772703388, 0.0009176706046168728,
0.0009135327358517573, 0.0009094032794931709, 0.0009052822440419779,
0.0009011696379817033, 0.0008970654697785228, 0.0008929697478812416,
0.000888882480721274, 0.0008848036767126353, 0.0008807333442519119,
0.0008766714917182576, 0.000872618127473363, 0.00086857325986145,
0.0008645368972092459, 0.0008605090478259728, 0.0008564897200033237,
0.000852478922015453, 0.0008484766621189526, 0.0008444829485528415,
0.0008404977895385401, 0.0008365211932798629, 0.0008325531679629974,
0.0008285937217564828, 0.0008246428628112025, 0.0008207005992603574,
0.0008167669392194581, 0.0008128418907863009, 0.0008089254620409578,
0.0008050176610457516, 0.0008011184958452498, 0.0007972279744662368,
0.0007933461049177081, 0.0007894728951908429, 0.0007856083532589984,
0.0007817524870776875, 0.0007779053045845597, 0.00077404066813699393,
0.00077023702232406 9, 0.0007664159383425639], train_loss: 0.33767396211624146,
val_loss: 0.32699212431907654, val_acc: 0.887499988079071
Epoch: 7, lat_lr: [0.0007626035696209259, 0.0007587999240072654,
0.0007550050093317311, 0.0007512188334065033, 0.0007474414040257682,
0.0007436727289657105, 0.0007399128159844893, 0.0007361616728222289,
0.0007324193072010013, 0.0007286857268248043, 0.0007249609393795556,
0.0007212449525330682, 0.0007175377739350411, 0.0007138394112170384,
0.0007101498719924773, 0.00070646916385661, 0.0007027972943865111,
0.0006991342711410567, 0.0006954801016609157, 0.0006918347934685 28,
0.0006881983540680935, 0.0006845707909455532, 0.0006809521115685773,
0.0006773423233865477, 0.0006737414338305412, 0.0006701494503133182,
0.0006665663802293027, 0.0006629922309545733, 0.0006594270098468399,
0.0006558707242454366, 0.0006523233814713004, 0.0006487848988826961,
0.0006452555535965205, 0.0006417350830456448, 0.0006382235844215421,
0.0006347210649529529, 0.0006312275318501336, 0.0006277429923048388,
0.0006242674534903134, 0.0006208009225612673, 0.0006173434066538 74,
0.0006138949128857422, 0.0006104554483559 13, 0.0006070250201448356,
0.0006036036353143616, 0.0006001913009077205, 0.0005967880239495169,
0.0005933938114457033, 0.0005900086703835781, 0.0005866326077317625,
0.0005832656304401876, 0.0005799077454400857, 0.0005765589596439657,
0.0005732192799456119, 0.0005698888713220057, 0.0005665672663235777,
0.0005632549460936731, 0.0005599517593490589, 0.0005566577128896441,
0.0005533728134965254, 0.0005500970679319653, 0.0005468304829393874,
0.0005435730652433521, 0.0005403248215495518, 0.0005370857585447934,
0.0005338558828969806, 0.0005306352012551099, 0.0005274237202492457,
0.000524422144649 0516, 0.0005210283865710911, 0.0005178445470641784,
0.000514669934524, 0.0005115045554857864, 0.0005083484164657575,
0.0005052015239611151, 0.0005020638844500218, 0.0004989355043915 96,
0.0004958163902258939, 0.0004927065483738951, 0.0004896059852374 93,
0.0004865147071994 7916, 0.0004834327206235321, 0.0004803600318542002,
0.0004772966472168956, 0.0004742425730178735, 0.0004711978155442246,
0.0004681623810638589, 0.0004651362758254 9634, 0.0004621195060586482,
0.0004591120779736 1196, 0.0004561139977614527, 0.0004531252715939899,

0.0004501459056237907, 0.00044717590598415056, 0.0004442152787890848,
0.0004412640301333138, 0.0004383221660922534, 0.00043538969272199663,
0.0004324666160593093, 0.0004295529421216095, 0.00042664867690696235,
0.00042375382639406106, 0.00042086839654222025, 0.00041799239329135923,
0.0004151258225619939, 0.00041226869025522134, 0.0004094210022527076,
0.0004065827644166799, 0.00040375398258990784, 0.0004009346625956983,
0.0003981248102378765, 0.0003953244313007817, 0.00039253353154924736,
0.00038975211672859683, 0.0003869801925646242, 0.00038421776476358944,
0.00038146483901220133, 0.0003787214209776079, 0.00037598751630738724,
0.00037326313062952877, 0.00037054826955243105, 0.00036784293866488816,
0.0003651471435360512, 0.0003624608897154794, 0.0003597841827330657,
0.00035711702809905363, 0.0003544594313040258, 0.00035181139781888634,
0.0003491729330948545, 0.00034654404256344887, 0.00034392473163648144,
0.00034131500570604357, 0.00033871487014449134, 0.00033612433030444433,
0.00033354339151876, 0.0003309720591005408, 0.0003284103383431057,
0.0003258582345199931, 0.0003233157528849388, 0.0003207828986718751,
0.00031825967709491203, 0.00031574609334833156, 0.0003132421526065743,
0.000310747860024232, 0.0003082632207360318, 0.00030578823985683056,
0.0003033229224816029, 0.00030086727368542834, 0.00029842129852348457,
0.00029598500203103367, 0.00029355838922341475, 0.0002911414650960303,
0.0002887342346243404, 0.0002863367027638468, 0.00028394887445008945,
0.0002815707545986284, 0.0002792023481050424, 0.00027684365984490954,
0.00027449469467380596, 0.00027215545742729184, 0.00026982595292089824,
0.0002675061859501233, 0.0002651961612904189, 0.00026289588369718164,
0.00026060535790574136, 0.00025832458863135616, 0.000256053580569195,
0.00025379233839433764, 0.0002515408667617553, 0.0002492991703063086,
0.00024706725364273313, 0.00024484512136563357, 0.00024263277804947189,
0.0002404302282485573, 0.00023823747649704137, 0.0002360545273089008,
0.00023388138517793793, 0.00023171805457776243, 0.000229564539961788,
0.0002274208457632207, 0.00022528697639505118, 0.00022316293625004245,
0.00022104872970072655, 0.00021894436109938848, 0.00021684983477806454,
0.00021476515504852727, 0.00021269032620227943, 0.0002106253525105473,
0.00020857023822426512, 0.00020652498757407458, 0.00020448960477030924,
0.00020246409400299123, 0.0002004484594418184, 0.00019844270523615946,
0.00019644683551504, 0.00019446085438714253, 0.00019248476594078756,
0.0001905185742439359, 0.00018856228334416974, 0.00018661589726869375,
0.00018467942002432176, 0.00018275285559746676, 0.0001808362079541388,
0.00017892948103993088, 0.00017703267878001525, 0.0001751458050791306,
0.00017326886382157983, 0.0001714018588712167, 0.00016954479407144254,
0.0001676976732451923, 0.00016586050019493475, 0.00016403327870265604,
0.00016221601252986004, 0.00016040870541755376, 0.00015861136108624336,
0.00015682398323592716, 0.00015504657554608436, 0.00015327914167567123,
0.0001515216852631111, 0.00014977420992628936, 0.00014803671926254128,
0.0001463092168486514, 0.0001445917062408397, 0.00014288419097475987,
0.0001411866745654861, 0.0001394991605075119, 0.0001378216522747378,
0.00013615415332046766, 0.00013449666707740076, 0.00013284919695762291,
0.00013121174635260212, 0.00012958431863317903, 0.00012796691714956257,
0.00012635954523132047, 0.00012476220618737487, 0.00012317490330599333,

0.00012159763985478455, 0.00012003041908068767, 0.0001184732442099707,
0.00011692611844821884, 0.00011538904498033277, 0.00011386202697051763,
0.00011234506756227962, 0.00011083816987841770, 0.00010934133702101748,
0.00010785457207144679, 0.00010637787809034623, 0.00010491125811762584,
0.00010345471517245567, 0.00010200825225326246, 0.00010057187233772126,
9.914557838275219e-05, 9.772937332451034e-05, 9.632326007838422e-05,
9.492724153898457e-05, 9.354132058014326e-05, 9.216550005490612e-05,
9.079978279552404e-05, 8.94441716134507e-05, 8.809866929933431e-05,
8.676327862301594e-05, 8.54380023335178e-05, 8.412284315904218e-05,
8.28178038069637e-05, 8.152288696382591e-05, 8.02380952953319e-05,
7.896343144634376e-05, 7.769889804087144e-05, 7.644449768207387e-05,
7.520023295224788e-05, 7.396610641282591e-05, 7.274212060437003e-05,
7.152827804656626e-05, 7.032458123822079e-05, 6.913103265725209e-05,
6.794763476068991e-05, 6.677438998466581e-05, 6.561130074441202e-05,
6.445836943425258e-05, 6.331559842760334e-05, 6.21829900769603e-05,
6.106054671390294e-05, 5.994827064908259e-05, 5.88461641722207e-05,
5.775422955210446e-05, 5.667246903658012 6e-05, 5.560088485255 1314e-05,
5.453947920597019e-05, 5.34882542818385e-05, 5.244721224419765e-05,
5.141635523612866e-05, 5.0395685379743297e-05, 4.938520477618 4076e-05,
4.83849155056176e-05, 4.739481962723067e-05, 4.6414919179225294e-05,
4.5445216178815886e-05, 4.4485712622225436e-05, 4.3536410484678235e-05,
4.259731172039991e-05, 4.166841826261131e-05, 4.0749732023523495e-05,
3.984125489433554e-05, 3.894298874523174e-05, 3.805493542537386e-05,
3.717709676290109e-05, 3.630947456492346e-05, 3.545207061752175e-05,
3.4604886685738683e-05, 3.376792451358054e-05, 3.294118582400999e-05,
3.2124672318943816e-05, 3.131838567925019e-05, 3.052232756474307e-05,
2.9736499614180584e-05, 2.8960903445261128e-05, 2.8195540654620574e-05,
2.7440412817827294e-05, 2.669552148938159e-05, 2.5960868202709597e-05,
2.5236454470162733e-05, 2.4522281783011573e-05, 2.3818351611446425e-05,
2.3124665404572328e-05, 2.2441224590404045e-05, 2.1768030575868296e-05,
2.1105084746795433e-05, 2.0452388467919982e-05, 1.9809943082876766e-05,
1.9177749914198687e-05, 1.8555810263311725e-05, 1.7944125410537156e-05,
1.7342696615082124e-05, 1.6751525115044075e-05, 1.6170612127402988e-05,
1.5599958848020816e-05, 1.503956645163983e-05, 1.4489436091878168e-05,
1.394956890123039e-05, 1.3419965991061386e-05, 1.2900628451608021e-05,
1.2391557351972492e-05, 1.1892753740125645e-05, 1.1404218642898657e-05,
1.0925953065986919e-05, 1.0457957993942815e-05, 1.0000234390177398e-05,
9.55278319695594e-06, 9.11560533539793e-06, 8.688701705473757e-06,
8.272073186003581e-06, 7.865720634654579e-06, 7.46964488794092e-06,
7.083846761221015e-06, 6.708327048694722e-06, 6.343086523404464e-06,
5.988125937230236e-06, 5.643446020891263e-06, 5.309047483942 6756e-06,
4.984931014774395e-06, 4.671097280608919e-06, 4.367546927502 4224e-06,
4.074280580340882e-06, 3.791298842839 5137e-06, 3.5186022975422208e-06,
3.2561915058199277e-06, 3.00406700786836e-06, 2.7622293227097105e-06,
2.5306789481881955e-06, 2.3094163609717233e-06, 2.098442016549 1175e-06,
1.8977563492312269e-06, 1.707359772147039e-06, 1.5272526772447925e-06,
1.3574354352925305e-06, 1.19790839587366e-06, 1.0486718873897277e-06,
9.097262170576451e-07, 7.810716709091318e-07, 6.627085137912717e-07,

```
5.546369893665131e-07, 4.568573201087824e-07, 3.693697073068151e-07,
2.9217433106249027e-07, 2.2527135028805503e-07, 1.686609027111207e-07,
1.2234310486800144e-07, 8.63180521092651e-08, 6.058581859473759e-08,
4.514645729738839e-08, 4e-08], train_loss: 0.2512175142765045, val_loss:
0.306847482919693, val_acc: 0.8960937261581421
CPU times: user 6min 45s, sys: 4min 46s, total: 11min 32s
Wall time: 11min 39s
```

4 Validation

4.1 Data augmentation

[ ]:
```
batch_size_ft = 256

train_ds =  datasets.CIFAR10(
    root=data_dir,
    train=True,
    download=True,
    transform=augmentation_transform
)

val_ds =  datasets.CIFAR10(
    root=data_dir,
    train=False,
    download=False,
    transform=val_transform
)

train_dl = DataLoader(train_ds, batch_size_ft, shuffle=True, num_workers=2,␣
 ↪pin_memory=True)
val_dl = DataLoader(val_ds, batch_size_ft*2, num_workers=2, pin_memory=True)
train_dl_ft = DeviceDataLoader(train_dl, device)
val_dl_ft = DeviceDataLoader(val_dl, device)
```

```
Files already downloaded and verified
```

4.2 Prediction

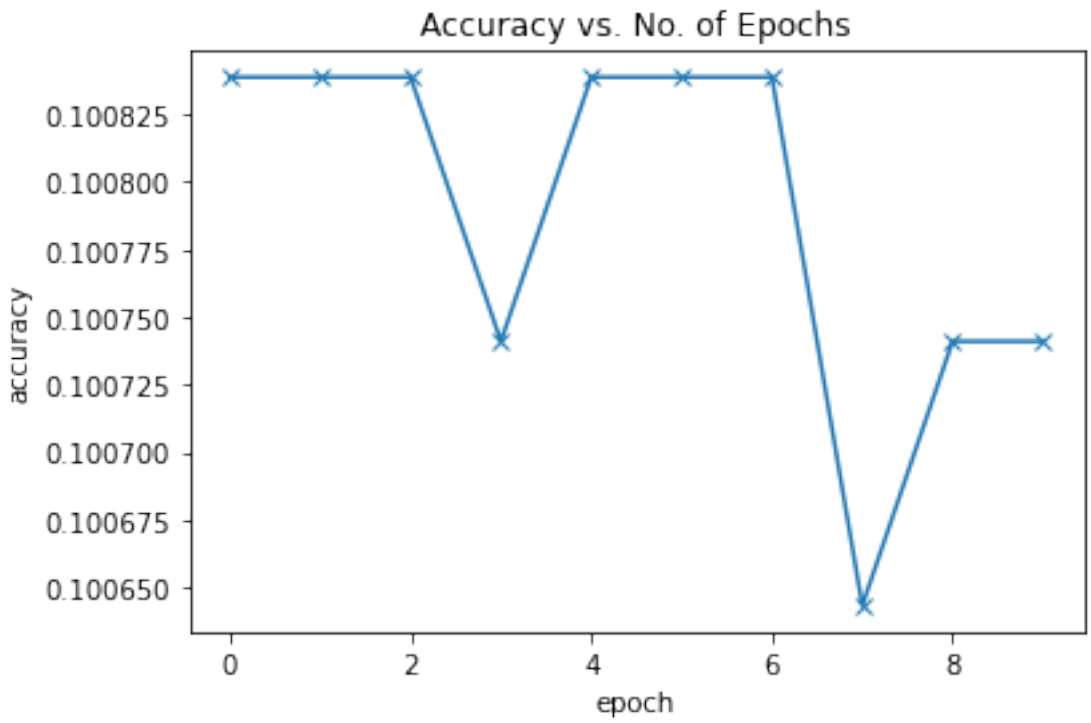
[ ]:
```
def predict(img, model):
    # Convert to a batch of 1
    xb = to_device(img.unsqueeze(0), device)
    # Get predictions from model
    yb = model(xb)
    # Pick index with highest probability
    _, preds  = torch.max(yb, dim=1)
    # Retrieve the class label
    return val_ds.classes[preds[0].item()]
```

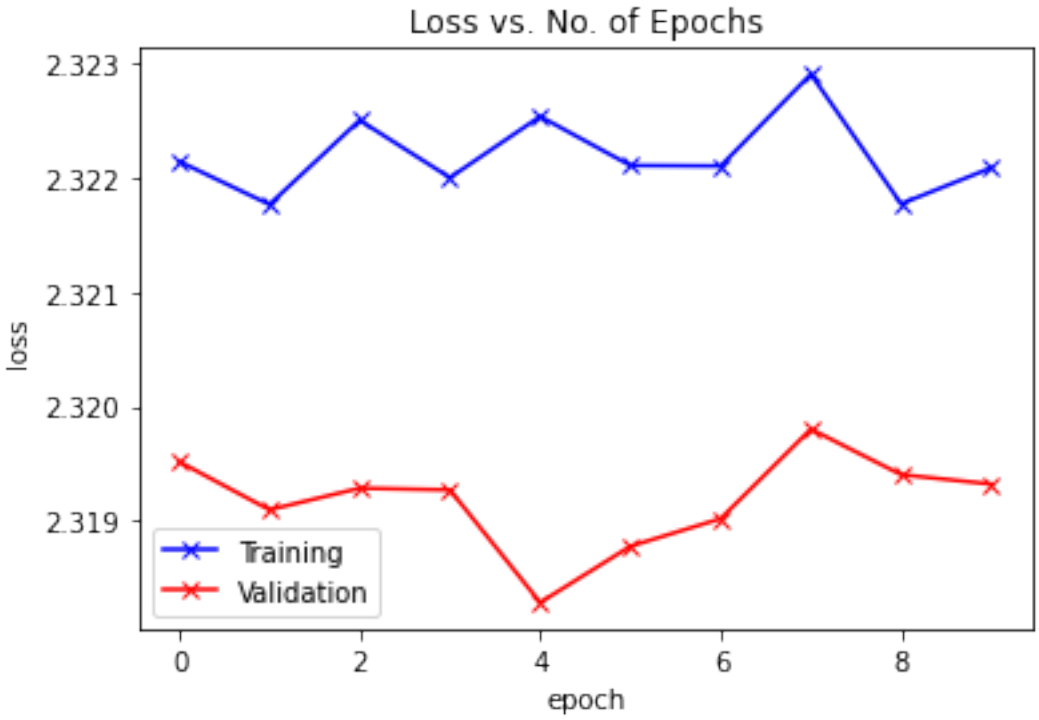
```
#img, label = val_ds[467]
#plt.imshow(img.permute(1, 2, 0))
#print('Label:', val_ds.classes[label], ', Predicted:', predict(img,␣
 ↪resNetFineTunning))
```

[ ]:
```
resNet9 = ResNet9(3, 10)
resNet9.to(device)



history = fit_one_cycle(epochs, max_lr, resNet9, train_dl_ft, val_dl_ft,␣
 ↪grad_clip=grad_clip,
                         opt_func=opt_func)

plot_accuracies(history_ft, 'resNet_best')
plot_losses(history_ft, 'resNest_best')

y = []
y_pred = []

for img, lbl in val_ds:
  y.append(val_ds.classes[lbl])
  y_pred.append(predict(img, resNet9))

confusion_matrix(y, y_pred, labels=val_ds.classes)
```

```
Epoch 0/9
----------
Epoch: 0, last_lr: 0.0027851835584105904, train_loss: 1.2554885149002075,
val_loss: 1.082831859588623, val_acc: 0.6149126887321472
Best Acc: 0.6149126887321472 in Epoch no. 0
Epoch 1/9
----------
Epoch: 1, last_lr: 0.007592580310080607, train_loss: 0.9218870997428894,
val_loss: 1.8074153661727905, val_acc: 0.5475987792015076
Best Acc: 0.6149126887321472 in Epoch no. 0
Epoch 2/9
----------
Epoch: 2, last_lr: 0.01, train_loss: 0.7938281297683716, val_loss:
1.2265682220458984, val_acc: 0.6216853857040405
Best Acc: 0.6216853857040405 in Epoch no. 2
Epoch 3/9
----------
Epoch: 3, last_lr: 0.009504846320134737, train_loss: 0.6052901148796082,
val_loss: 1.1297537088394165, val_acc: 0.6879020929336548
Best Acc: 0.6879020929336548 in Epoch no. 3
Epoch 4/9
```

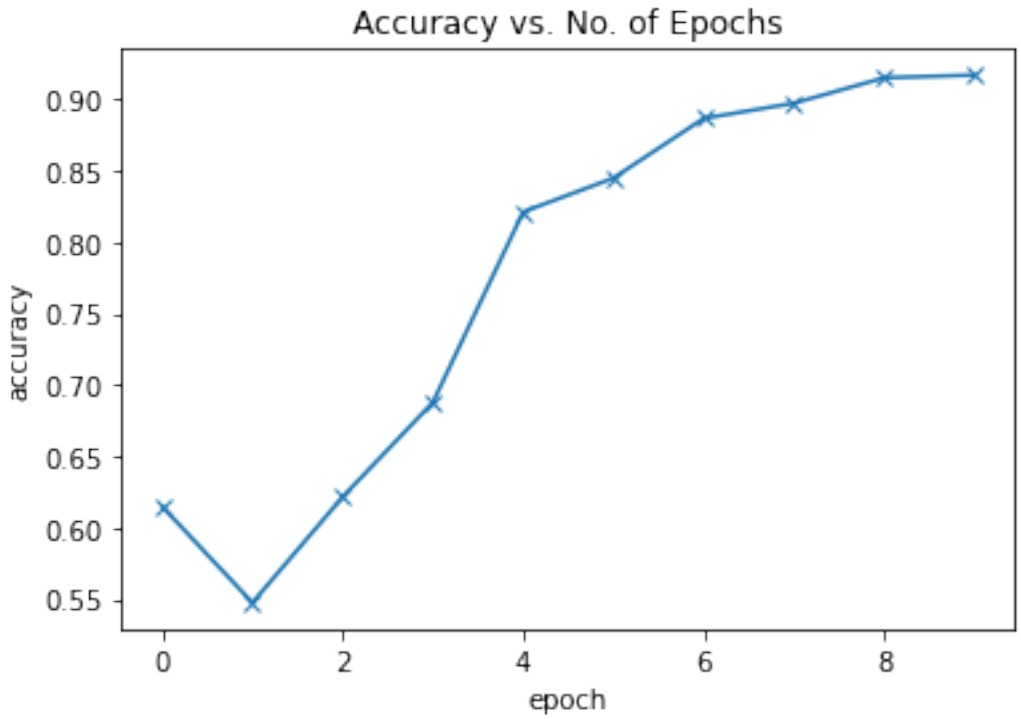
```
----------
Epoch: 4, last_lr: 0.008117456539497631, train_loss: 0.4544855058193207,
val_loss: 0.5621666312217712, val_acc: 0.8212717771530151
Best Acc: 0.8212717771530151 in Epoch no. 4
Epoch 5/9
----------
Epoch: 5, last_lr: 0.0061126202193628925, train_loss: 0.34964555501937866,
val_loss: 0.46824899315834045, val_acc: 0.8449621200561523
Best Acc: 0.8449621200561523 in Epoch no. 5
Epoch 6/9
----------
Epoch: 6, last_lr: 0.003887419780637108, train_loss: 0.28146857023239136,
val_loss: 0.3418613076210022, val_acc: 0.8867474794387817
Best Acc: 0.8867474794387817 in Epoch no. 6
Epoch 7/9
----------
Epoch: 7, last_lr: 0.00188258346050237, train_loss: 0.20665232837200165,
val_loss: 0.3106996715068817, val_acc: 0.897426426410675
Best Acc: 0.897426426410675 in Epoch no. 7
Epoch 8/9
----------
Epoch: 8, last_lr: 0.0004951936798652628, train_loss: 0.15422232449054718,
val_loss: 0.2627998888492584, val_acc: 0.9152171015739441
Best Acc: 0.9152171015739441 in Epoch no. 8
Epoch 9/9
----------
Epoch: 9, last_lr: 4e-08, train_loss: 0.12455056607723236, val_loss:
0.2577187716960907, val_acc: 0.9171243906021118
Best Acc: 0.9171243906021118 in Epoch no. 9
```

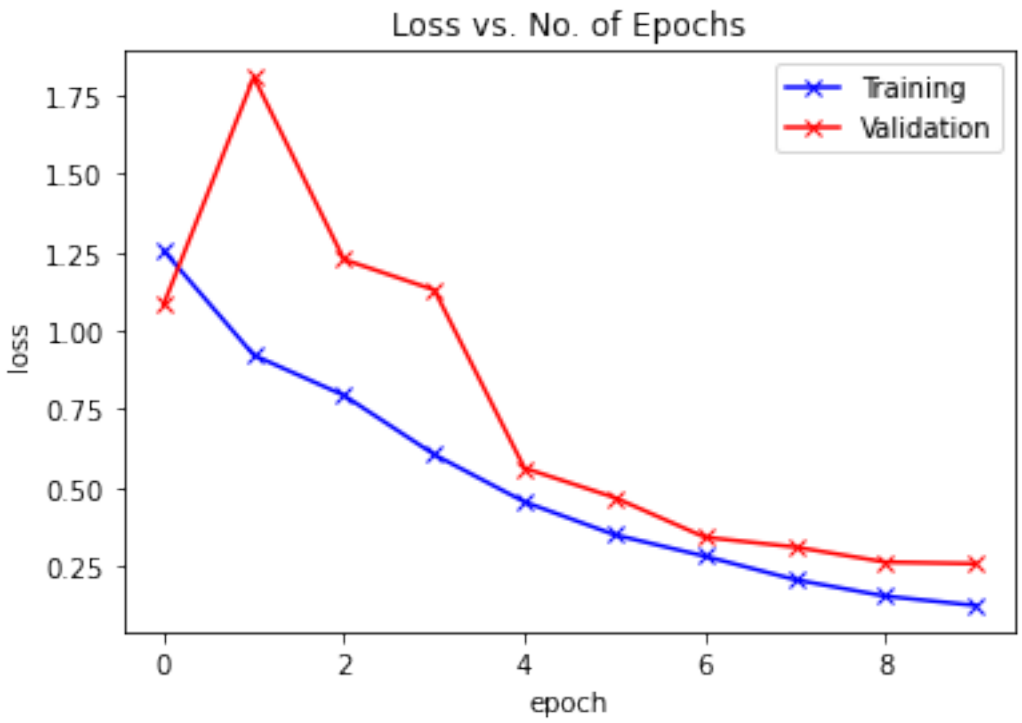

Accuracy vs. No. of Epochs
accuracy
0.100825
0.100800
0.100775
0.100750
0.100725
0.100700
0.100675
0.100650
0
2
4
6
8
epoch


Loss vs. No. of Epochs
loss
2.323
2.322
2.321
2.320
2.319
Training
Validation
0
2
4
6
8
epoch

```
[ ]: array([[938,   3,  15,   7,   1,   1,   1,   3,  21,  10],
            [  6, 977,   0,   1,   0,   0,   0,   0,   3,  13],
            [ 25,   0, 877,  23,  23,  21,  21,   6,   2,   2],
            [  8,   2,  19, 800,  19,  93,  27,  14,   6,  12],
            [  6,   1,  17,  20, 919,  16,   9,  10,   2,   0],
            [  4,   0,  11,  66,  19, 879,   6,  13,   0,   2],
            [  4,   0,  16,  19,   6,   3, 946,   4,   1,   1],
            [  4,   0,   7,  16,  13,  17,   3, 938,   0,   2],
            [ 35,   8,   2,   1,   0,   0,   2,   1, 943,   8],
            [  6,  28,   1,   1,   0,   1,   0,   1,  10, 952]])
```

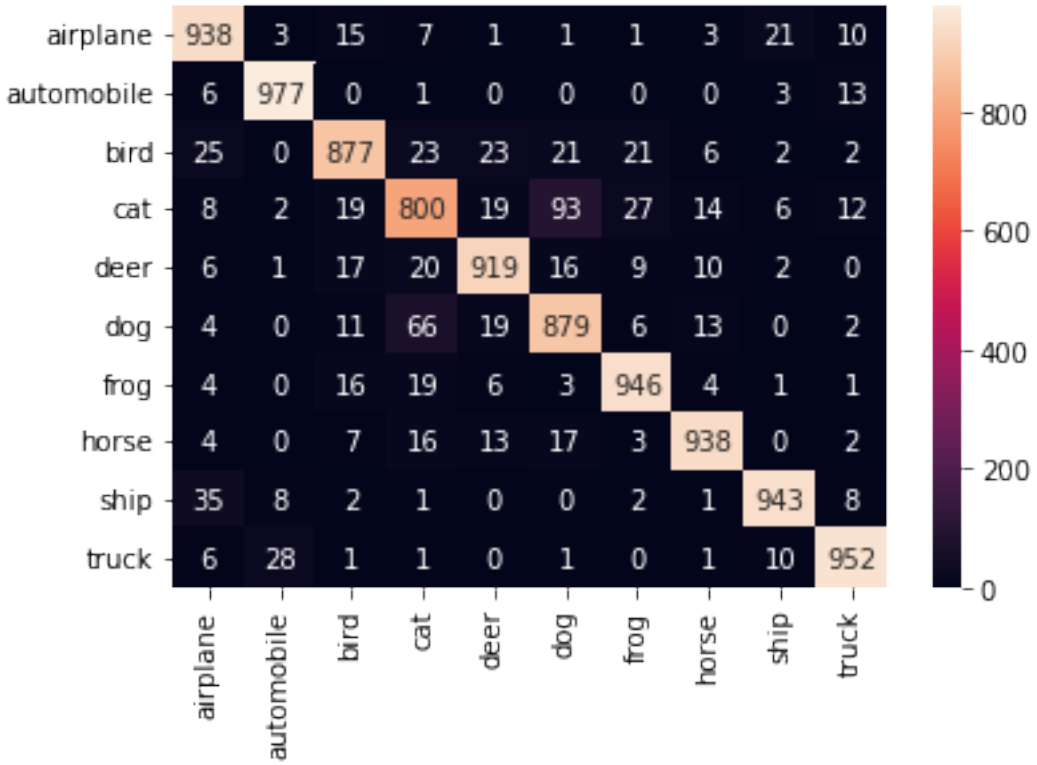
```
[ ]: plot_accuracies(history, 'resNet_best')
     plot_losses(history, 'resNest_best')
```





[ ]:
```
sns.heatmap(confusion_matrix(y, y_pred, labels=val_ds.classes), annot=True,␣
↪xticklabels=val_ds.classes, yticklabels=val_ds.classes, fmt='g')
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff720d144d0>
```



5 Transfer Learning

5.1 Fixed Feature Extractor

5.1.1 tf training function

[ ]:
```
dataloaders = {
    'train': train_dl_ft,
    'val': val_dl_ft
}
```

[ ]:
```
def train_model(model, criterion, optimizer, scheduler, num_epochs=25):
    since = time.time()

    best_model_wts = copy.deepcopy(model.state_dict())
    best_acc = 0.0

    accs = []
    loss = []

    for epoch in range(num_epochs):
        print('Epoch {}/{}'.format(epoch, num_epochs - 1))
```

```
        print('-' * 10)

        # Each epoch has a training and validation phase
        for phase in ['train', 'val']:
            if phase == 'train':
                model.train()  # Set model to training mode
            else:
                model.eval()   # Set model to evaluate mode

            running_loss = 0.0
            running_corrects = 0

            # Iterate over data.
            for inputs, labels in dataloaders[phase]:

                # zero the parameter gradients
                optimizer.zero_grad()

                # forward
                # track history if only in train
                with torch.set_grad_enabled(phase == 'train'):
                    outputs = model(inputs)
                    _, preds = torch.max(outputs, 1)
                    loss = criterion(outputs, labels)

                    # backward + optimize only if in training phase
                    if phase == 'train':
                        loss.backward()
                        optimizer.step()

                # statistics
                running_loss += loss.item() * inputs.size(0)
                running_corrects += torch.sum(preds == labels.data)
            if phase == 'train':
                scheduler.step()

            epoch_loss = running_loss / dataset_sizes[phase]
            epoch_acc = running_corrects.double() / dataset_sizes[phase]

            accs.append(epoch_acc)
            accs.append(epoch_loss)

            print('{} Loss: {:.4f} Acc: {:.4f}'.format(
                phase, epoch_loss, epoch_acc))

            # deep copy the model
            if phase == 'val' and epoch_acc > best_acc:
```

```
                best_acc = epoch_acc
                best_model_wts = copy.deepcopy(model.state_dict())

            print()

        time_elapsed = time.time() - since
        print('Training complete in {:.0f}m {:.0f}s'.format(
            time_elapsed // 60, time_elapsed % 60))
        print('Best val Acc: {:4f}'.format(best_acc))

        # load best model weights
        model.load_state_dict(best_model_wts)
        return accs, loss
```

[ ]:

[ ]:
```
<bound method Module.cuda of ResNet9(
  (conv1): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
  )
  (conv2): Sequential(
    (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (res1): Sequential(
    (0): Sequential(
      (0): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (1): Sequential(
      (0): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (conv3): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

```
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (conv4): Sequential(
    (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (res2): Sequential(
    (0): Sequential(
      (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (1): Sequential(
      (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (classifier): Linear(in_features=512, out_features=10, bias=True)
)>
```

[ ]:
```
for param in resNet9.parameters():
    param.requires_grad = False

resNet9.classifier = nn.Linear(512, 10)

criterion = nn.CrossEntropyLoss().
optimizer_conv = optim.SGD(resNet9.classifier.parameters(), lr=0.001,␣
 ↪momentum=0.9)
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_conv, step_size=7, gamma=0.1)


accs, loss = train_model(resNet9, criterion, optimizer_conv,
                         exp_lr_scheduler, num_epochs=20)
```

```
Epoch 0/19
----------
```

```
      ␣
↪---------------------------------------------------------------------------

      RuntimeError                              Traceback (most recent call␣
↪last)

      <ipython-input-107-2d5a6a597405> in <module>()
       15
       16 accs, loss = train_model(resNet9, criterion, optimizer_conv,
   ---> 17                         exp_lr_scheduler, num_epochs=20)


      <ipython-input-106-88adc23a8554> in train_model(model, criterion,␣
↪optimizer, scheduler, num_epochs)
       31                 # track history if only in train
       32                 with torch.set_grad_enabled(phase == 'train'):
   ---> 33                     outputs = model(inputs)
       34                     _, preds = torch.max(outputs, 1)
       35                     loss = criterion(outputs, labels)


      /usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py in␣
↪_call_impl(self, *input, **kwargs)
     1100         if not (self._backward_hooks or self._forward_hooks or self.
↪_forward_pre_hooks or _global_backward_hooks
     1101                 or _global_forward_hooks or␣
↪_global_forward_pre_hooks):
   -> 1102             return forward_call(*input, **kwargs)
     1103         # Do not call functions when jit is used
     1104         full_backward_hooks, non_full_backward_hooks = [], []


      <ipython-input-14-8e2b8e2ee3f0> in forward(self, X)
       34     out = self.conv4(out)
       35     out = self.res2(out) + out
   ---> 36     out = self.classifier(out)
       37
       38     return out


      /usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py in␣
↪_call_impl(self, *input, **kwargs)
     1100         if not (self._backward_hooks or self._forward_hooks or self.
↪_forward_pre_hooks or _global_backward_hooks
     1101                 or _global_forward_hooks or␣
↪_global_forward_pre_hooks):
```

```
   -> 1102             return forward_call(*input, **kwargs)
      1103         # Do not call functions when jit is used
      1104         full_backward_hooks, non_full_backward_hooks = [], []


      /usr/local/lib/python3.7/dist-packages/torch/nn/modules/linear.py in␣
↪forward(self, input)
       101
       102     def forward(self, input: Tensor) -> Tensor:
   --> 103         return F.linear(input, self.weight, self.bias)
       104
       105     def extra_repr(self) -> str:


      /usr/local/lib/python3.7/dist-packages/torch/nn/functional.py in␣
↪linear(input, weight, bias)
      1846     if has_torch_function_variadic(input, weight, bias):
      1847         return handle_torch_function(linear, (input, weight, bias),␣
↪input, weight, bias=bias)
   -> 1848     return torch._C._nn.linear(input, weight, bias)
      1849
      1850


      RuntimeError: Expected all tensors to be on the same device, but found␣
↪at least two devices, cuda:0 and cpu! (when checking argument for argument␣
↪mat2 in method wrapper_mm)
```