ztl2103_coms6998_a4

November 21, 2020

1 COMS 6998 - Practical Deep Learning System Performance

1.1 Assignment 4

- **Name**: Zach Lawless
- **UNI**: ztl2103

1.1.1 Problem 1: *Transfer Learning: Shallow Learning vs Fine Tuning, PyTorch* (30 points)

Q1 Using this PyTorch tutorial as reference.

- a: (4 points)

Using the `vgg-flowers` Visual Domain Decathalon dataset.

[1]:
```
from __future__ import print_function, division

import torch
import torch.nn as nn
import torch.optim as optim
from torch.optim import lr_scheduler
import numpy as np
import torchvision
from torchvision import datasets, models, transforms
import matplotlib.pyplot as plt
import time
import os
import copy

plt.ion()
```

[2]:
```
# Data augmentation and normalization for training
# Just normalization for validation
data_transforms = {
    'train': transforms.Compose([
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
```

```python
    ]),
    'val': transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ]),
}

data_dir = '~/data/vgg-flowers'
image_datasets = {x: datasets.ImageFolder(os.path.join(data_dir, x),
                                          data_transforms[x])
                  for x in ['train', 'val']}

dataloaders = {x: torch.utils.data.DataLoader(image_datasets[x], batch_size=64,
                                              shuffle=True, num_workers=4)
               for x in ['train', 'val']}

dataset_sizes = {x: len(image_datasets[x]) for x in ['train', 'val']}
class_names = image_datasets['train'].classes

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
```

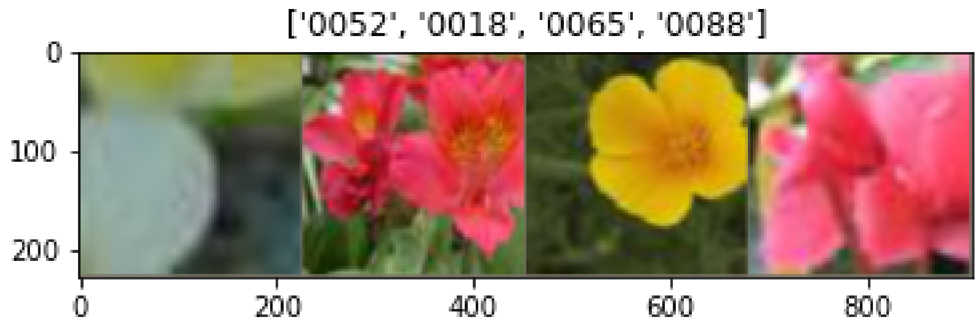
[3]:

```python
# Funtion to plot image
def imshow(inp, title=None):
    """Imshow for Tensor."""
    inp = inp.numpy().transpose((1, 2, 0))
    mean = np.array([0.485, 0.456, 0.406])
    std = np.array([0.229, 0.224, 0.225])
    inp = std * inp + mean
    inp = np.clip(inp, 0, 1)
    plt.imshow(inp)
    if title is not None:
        plt.title(title)
    plt.pause(0.001)  # pause a bit so that plots are updated


# Get a batch of training data
inputs, classes = next(iter(dataloaders['train']))

# Make a grid from batch
out = torchvision.utils.make_grid(inputs[:4])

imshow(out, title=[class_names[x] for x in classes[:4]])
```



[4]:
```
# Show how many classes there are in the training dataset
print(f"There are {len(class_names)} target classes in the training dataset")
```

```
There are 102 target classes in the training dataset
```

[5]:
```
# Show the distribution of training images per class
base_dir = '/home/jupyter/data/vgg-flowers/train/'

print("Distribution of training images per class")
for cn in os.listdir(base_dir):
    cn_images = os.listdir(os.path.join(base_dir, cn))
    print(f'Class {cn}: {len(cn_images)} images')
```

```
Distribution of training images per class
Class 0028: 10 images
Class 0075: 10 images
Class 0029: 10 images
Class 0009: 10 images
Class 0092: 10 images
Class 0085: 10 images
Class 0061: 10 images
Class 0056: 10 images
Class 0074: 10 images
Class 0050: 10 images
Class 0013: 10 images
Class 0069: 10 images
Class 0040: 10 images
Class 0072: 10 images
Class 0003: 10 images
Class 0017: 10 images
Class 0101: 10 images
Class 0094: 10 images
Class 0026: 10 images
Class 0046: 10 images
Class 0008: 10 images
```

```
Class 0071: 10 images
Class 0049: 10 images
Class 0080: 10 images
Class 0067: 10 images
Class 0044: 10 images
Class 0100: 10 images
Class 0102: 10 images
Class 0007: 10 images
Class 0089: 10 images
Class 0015: 10 images
Class 0087: 10 images
Class 0004: 10 images
Class 0077: 10 images
Class 0022: 10 images
Class 0030: 10 images
Class 0090: 10 images
Class 0060: 10 images
Class 0005: 10 images
Class 0025: 10 images
Class 0036: 10 images
Class 0064: 10 images
Class 0053: 10 images
Class 0099: 10 images
Class 0034: 10 images
Class 0065: 10 images
Class 0024: 10 images
Class 0066: 10 images
Class 0059: 10 images
Class 0032: 10 images
Class 0097: 10 images
Class 0096: 10 images
Class 0084: 10 images
Class 0042: 10 images
Class 0016: 10 images
Class 0086: 10 images
Class 0012: 10 images
Class 0051: 10 images
Class 0095: 10 images
Class 0045: 10 images
Class 0037: 10 images
Class 0058: 10 images
Class 0057: 10 images
Class 0055: 10 images
Class 0020: 10 images
Class 0014: 10 images
Class 0019: 10 images
Class 0073: 10 images
Class 0070: 10 images
```

```
Class 0078: 10 images
Class 0018: 10 images
Class 0011: 10 images
Class 0039: 10 images
Class 0083: 10 images
Class 0082: 10 images
Class 0047: 10 images
Class 0023: 10 images
Class 0063: 10 images
Class 0021: 10 images
Class 0001: 10 images
Class 0006: 10 images
Class 0062: 10 images
Class 0031: 10 images
Class 0033: 10 images
Class 0041: 10 images
Class 0027: 10 images
Class 0093: 10 images
Class 0052: 10 images
Class 0098: 10 images
Class 0081: 10 images
Class 0038: 10 images
Class 0002: 10 images
Class 0010: 10 images
Class 0076: 10 images
Class 0088: 10 images
Class 0054: 10 images
Class 0035: 10 images
Class 0048: 10 images
Class 0043: 10 images
Class 0091: 10 images
Class 0079: 10 images
Class 0068: 10 images
```

[6]:

```
# Load pretrained Resnet50 and change final layer
model_ft = models.resnet50(pretrained=True)
num_ftrs = model_ft.fc.in_features
model_ft.fc = nn.Linear(num_ftrs, len(class_names))
```

- b: (8 points)

[7]:

```
# Function to train the model
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
```

```
    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]:
                inputs = inputs.to(device)
                labels = labels.to(device)

                # 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]

            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 model
```

[8]:
```
# Set up criterion
criterion = nn.CrossEntropyLoss()

# Observe that all parameters are being optimized
optimizer_ft = optim.SGD(model_ft.parameters(), lr=0.001, momentum=0.9)

# Decay LR by a factor of 0.1 every 60 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=60, gamma=0.1)

# Send model to device
model_ft = model_ft.to(device)
```

[9]:
```
# Train model
model_ft = train_model(model_ft, criterion, optimizer_ft, exp_lr_scheduler,␣
 ↪num_epochs=200)
```

```
Epoch 0/199
----------
train Loss: 4.6525 Acc: 0.0225
val Loss: 4.5929 Acc: 0.0196

Epoch 1/199
----------
train Loss: 4.5687 Acc: 0.0265
val Loss: 4.4863 Acc: 0.0578

Epoch 2/199
----------
train Loss: 4.4612 Acc: 0.0882
val Loss: 4.3703 Acc: 0.1412

Epoch 3/199
```

```
----------
train Loss: 4.3490 Acc: 0.1373
val Loss: 4.2245 Acc: 0.2235

Epoch 4/199
----------
train Loss: 4.2027 Acc: 0.2225
val Loss: 4.0441 Acc: 0.2716

Epoch 5/199
----------
train Loss: 4.0379 Acc: 0.2804
val Loss: 3.8313 Acc: 0.2902

Epoch 6/199
----------
train Loss: 3.8418 Acc: 0.3157
val Loss: 3.6239 Acc: 0.3412

Epoch 7/199
----------
train Loss: 3.6518 Acc: 0.3657
val Loss: 3.4105 Acc: 0.3824

Epoch 8/199
----------
train Loss: 3.4888 Acc: 0.4137
val Loss: 3.2118 Acc: 0.3961

Epoch 9/199
----------
train Loss: 3.2932 Acc: 0.4333
val Loss: 3.0344 Acc: 0.4353

Epoch 10/199
----------
train Loss: 3.0963 Acc: 0.4500
val Loss: 2.8744 Acc: 0.4637

Epoch 11/199
----------
train Loss: 2.9870 Acc: 0.4765
val Loss: 2.7030 Acc: 0.4843

Epoch 12/199
----------
train Loss: 2.8186 Acc: 0.4892
val Loss: 2.5716 Acc: 0.5049
```

```
Epoch 13/199
----------
train Loss: 2.6741 Acc: 0.5353
val Loss: 2.4413 Acc: 0.5294

Epoch 14/199
----------
train Loss: 2.5149 Acc: 0.5696
val Loss: 2.3390 Acc: 0.5382

Epoch 15/199
----------
train Loss: 2.3622 Acc: 0.5804
val Loss: 2.2405 Acc: 0.5686

Epoch 16/199
----------
train Loss: 2.2649 Acc: 0.5922
val Loss: 2.1320 Acc: 0.5853

Epoch 17/199
----------
train Loss: 2.1877 Acc: 0.6127
val Loss: 2.0334 Acc: 0.5990

Epoch 18/199
----------
train Loss: 2.0731 Acc: 0.6314
val Loss: 1.9588 Acc: 0.5961

Epoch 19/199
----------
train Loss: 1.9700 Acc: 0.6471
val Loss: 1.8846 Acc: 0.6167

Epoch 20/199
----------
train Loss: 1.8955 Acc: 0.6745
val Loss: 1.7982 Acc: 0.6402

Epoch 21/199
----------
train Loss: 1.8297 Acc: 0.6696
val Loss: 1.7503 Acc: 0.6343

Epoch 22/199
----------
```

```
train Loss: 1.6957 Acc: 0.6961
val Loss: 1.6790 Acc: 0.6696

Epoch 23/199
----------
train Loss: 1.6827 Acc: 0.6980
val Loss: 1.6344 Acc: 0.6784

Epoch 24/199
----------
train Loss: 1.5710 Acc: 0.7216
val Loss: 1.5978 Acc: 0.6765

Epoch 25/199
----------
train Loss: 1.5661 Acc: 0.7157
val Loss: 1.5337 Acc: 0.6814

Epoch 26/199
----------
train Loss: 1.4684 Acc: 0.7392
val Loss: 1.4947 Acc: 0.6931

Epoch 27/199
----------
train Loss: 1.3775 Acc: 0.7716
val Loss: 1.4557 Acc: 0.7029

Epoch 28/199
----------
train Loss: 1.3484 Acc: 0.7676
val Loss: 1.3992 Acc: 0.7078

Epoch 29/199
----------
train Loss: 1.2873 Acc: 0.7755
val Loss: 1.3689 Acc: 0.7157

Epoch 30/199
----------
train Loss: 1.2200 Acc: 0.7892
val Loss: 1.3266 Acc: 0.7186

Epoch 31/199
----------
train Loss: 1.1552 Acc: 0.8098
val Loss: 1.3034 Acc: 0.7343
```

```
Epoch 32/199
----------
train Loss: 1.1471 Acc: 0.8127
val Loss: 1.2778 Acc: 0.7167

Epoch 33/199
----------
train Loss: 1.0886 Acc: 0.8078
val Loss: 1.2153 Acc: 0.7333

Epoch 34/199
----------
train Loss: 1.0536 Acc: 0.8343
val Loss: 1.2033 Acc: 0.7284

Epoch 35/199
----------
train Loss: 0.9640 Acc: 0.8412
val Loss: 1.1998 Acc: 0.7324

Epoch 36/199
----------
train Loss: 0.9604 Acc: 0.8284
val Loss: 1.1479 Acc: 0.7333

Epoch 37/199
----------
train Loss: 0.8995 Acc: 0.8549
val Loss: 1.1398 Acc: 0.7431

Epoch 38/199
----------
train Loss: 0.8513 Acc: 0.8716
val Loss: 1.1092 Acc: 0.7598

Epoch 39/199
----------
train Loss: 0.8372 Acc: 0.8667
val Loss: 1.1065 Acc: 0.7480

Epoch 40/199
----------
train Loss: 0.8257 Acc: 0.8529
val Loss: 1.0717 Acc: 0.7539

Epoch 41/199
----------
train Loss: 0.8377 Acc: 0.8578
```

```
val Loss: 1.0633 Acc: 0.7598

Epoch 42/199
----------
train Loss: 0.7829 Acc: 0.8735
val Loss: 1.0501 Acc: 0.7627

Epoch 43/199
----------
train Loss: 0.7282 Acc: 0.8873
val Loss: 1.0434 Acc: 0.7569

Epoch 44/199
----------
train Loss: 0.7156 Acc: 0.8833
val Loss: 0.9955 Acc: 0.7686

Epoch 45/199
----------
train Loss: 0.6386 Acc: 0.9078
val Loss: 0.9926 Acc: 0.7667

Epoch 46/199
----------
train Loss: 0.6950 Acc: 0.8755
val Loss: 0.9833 Acc: 0.7686

Epoch 47/199
----------
train Loss: 0.6490 Acc: 0.9049
val Loss: 0.9869 Acc: 0.7608

Epoch 48/199
----------
train Loss: 0.6291 Acc: 0.9000
val Loss: 0.9786 Acc: 0.7667

Epoch 49/199
----------
train Loss: 0.6319 Acc: 0.8882
val Loss: 0.9523 Acc: 0.7814

Epoch 50/199
----------
train Loss: 0.5856 Acc: 0.9127
val Loss: 0.9507 Acc: 0.7745

Epoch 51/199
```

```
----------
train Loss: 0.5804 Acc: 0.9127
val Loss: 0.9392 Acc: 0.7657

Epoch 52/199
----------
train Loss: 0.5209 Acc: 0.9216
val Loss: 0.9366 Acc: 0.7716

Epoch 53/199
----------
train Loss: 0.5545 Acc: 0.9049
val Loss: 0.9032 Acc: 0.7784

Epoch 54/199
----------
train Loss: 0.5601 Acc: 0.9088
val Loss: 0.9176 Acc: 0.7804

Epoch 55/199
----------
train Loss: 0.4757 Acc: 0.9255
val Loss: 0.8888 Acc: 0.7922

Epoch 56/199
----------
train Loss: 0.5129 Acc: 0.9196
val Loss: 0.8959 Acc: 0.7833

Epoch 57/199
----------
train Loss: 0.4034 Acc: 0.9363
val Loss: 0.8909 Acc: 0.7814

Epoch 58/199
----------
train Loss: 0.4339 Acc: 0.9284
val Loss: 0.8798 Acc: 0.7784

Epoch 59/199
----------
train Loss: 0.4116 Acc: 0.9412
val Loss: 0.8729 Acc: 0.7902

Epoch 60/199
----------
train Loss: 0.4593 Acc: 0.9382
val Loss: 0.8544 Acc: 0.7931
```

```
Epoch 61/199
----------
train Loss: 0.4205 Acc: 0.9461
val Loss: 0.8548 Acc: 0.7941

Epoch 62/199
----------
train Loss: 0.4051 Acc: 0.9382
val Loss: 0.8613 Acc: 0.7931

Epoch 63/199
----------
train Loss: 0.4255 Acc: 0.9373
val Loss: 0.8595 Acc: 0.7912

Epoch 64/199
----------
train Loss: 0.4773 Acc: 0.9186
val Loss: 0.8565 Acc: 0.7863

Epoch 65/199
----------
train Loss: 0.4133 Acc: 0.9363
val Loss: 0.8591 Acc: 0.7941

Epoch 66/199
----------
train Loss: 0.4452 Acc: 0.9265
val Loss: 0.8596 Acc: 0.7931

Epoch 67/199
----------
train Loss: 0.4269 Acc: 0.9265
val Loss: 0.8584 Acc: 0.7941

Epoch 68/199
----------
train Loss: 0.4521 Acc: 0.9186
val Loss: 0.8563 Acc: 0.7922

Epoch 69/199
----------
train Loss: 0.4023 Acc: 0.9392
val Loss: 0.8564 Acc: 0.7902

Epoch 70/199
----------
```

```
train Loss: 0.4228 Acc: 0.9402
val Loss: 0.8595 Acc: 0.7882

Epoch 71/199
----------
train Loss: 0.4395 Acc: 0.9284
val Loss: 0.8572 Acc: 0.7912

Epoch 72/199
----------
train Loss: 0.4447 Acc: 0.9373
val Loss: 0.8573 Acc: 0.7873

Epoch 73/199
----------
train Loss: 0.3785 Acc: 0.9441
val Loss: 0.8507 Acc: 0.7931

Epoch 74/199
----------
train Loss: 0.4256 Acc: 0.9382
val Loss: 0.8571 Acc: 0.7892

Epoch 75/199
----------
train Loss: 0.3887 Acc: 0.9382
val Loss: 0.8555 Acc: 0.7882

Epoch 76/199
----------
train Loss: 0.4061 Acc: 0.9451
val Loss: 0.8547 Acc: 0.7902

Epoch 77/199
----------
train Loss: 0.3863 Acc: 0.9422
val Loss: 0.8563 Acc: 0.7892

Epoch 78/199
----------
train Loss: 0.4352 Acc: 0.9284
val Loss: 0.8537 Acc: 0.7912

Epoch 79/199
----------
train Loss: 0.4198 Acc: 0.9275
val Loss: 0.8543 Acc: 0.7902
```

```
Epoch 80/199
----------
train Loss: 0.4190 Acc: 0.9392
val Loss: 0.8531 Acc: 0.7892

Epoch 81/199
----------
train Loss: 0.3878 Acc: 0.9431
val Loss: 0.8544 Acc: 0.7922

Epoch 82/199
----------
train Loss: 0.4196 Acc: 0.9382
val Loss: 0.8522 Acc: 0.7902

Epoch 83/199
----------
train Loss: 0.3927 Acc: 0.9422
val Loss: 0.8523 Acc: 0.7882

Epoch 84/199
----------
train Loss: 0.4125 Acc: 0.9343
val Loss: 0.8529 Acc: 0.7902

Epoch 85/199
----------
train Loss: 0.4130 Acc: 0.9314
val Loss: 0.8532 Acc: 0.7892

Epoch 86/199
----------
train Loss: 0.4139 Acc: 0.9333
val Loss: 0.8587 Acc: 0.7902

Epoch 87/199
----------
train Loss: 0.3835 Acc: 0.9461
val Loss: 0.8578 Acc: 0.7961

Epoch 88/199
----------
train Loss: 0.3967 Acc: 0.9412
val Loss: 0.8483 Acc: 0.7912

Epoch 89/199
----------
train Loss: 0.3783 Acc: 0.9412
```

```
val Loss: 0.8462 Acc: 0.7931

Epoch 90/199
----------
train Loss: 0.3730 Acc: 0.9451
val Loss: 0.8487 Acc: 0.7941

Epoch 91/199
----------
train Loss: 0.4188 Acc: 0.9265
val Loss: 0.8492 Acc: 0.7922

Epoch 92/199
----------
train Loss: 0.4076 Acc: 0.9353
val Loss: 0.8544 Acc: 0.7873

Epoch 93/199
----------
train Loss: 0.4050 Acc: 0.9353
val Loss: 0.8470 Acc: 0.7922

Epoch 94/199
----------
train Loss: 0.4261 Acc: 0.9275
val Loss: 0.8484 Acc: 0.7912

Epoch 95/199
----------
train Loss: 0.4144 Acc: 0.9353
val Loss: 0.8511 Acc: 0.7882

Epoch 96/199
----------
train Loss: 0.3830 Acc: 0.9471
val Loss: 0.8470 Acc: 0.7882

Epoch 97/199
----------
train Loss: 0.3897 Acc: 0.9343
val Loss: 0.8451 Acc: 0.7902

Epoch 98/199
----------
train Loss: 0.3899 Acc: 0.9314
val Loss: 0.8452 Acc: 0.7961

Epoch 99/199
```

```
----------
train Loss: 0.3802 Acc: 0.9471
val Loss: 0.8495 Acc: 0.7951

Epoch 100/199
----------
train Loss: 0.3821 Acc: 0.9412
val Loss: 0.8418 Acc: 0.7941

Epoch 101/199
----------
train Loss: 0.3760 Acc: 0.9471
val Loss: 0.8371 Acc: 0.7990

Epoch 102/199
----------
train Loss: 0.3931 Acc: 0.9402
val Loss: 0.8451 Acc: 0.7931

Epoch 103/199
----------
train Loss: 0.4081 Acc: 0.9363
val Loss: 0.8471 Acc: 0.7980

Epoch 104/199
----------
train Loss: 0.4061 Acc: 0.9343
val Loss: 0.8468 Acc: 0.7922

Epoch 105/199
----------
train Loss: 0.3722 Acc: 0.9392
val Loss: 0.8472 Acc: 0.7951

Epoch 106/199
----------
train Loss: 0.3839 Acc: 0.9314
val Loss: 0.8454 Acc: 0.7922

Epoch 107/199
----------
train Loss: 0.3413 Acc: 0.9510
val Loss: 0.8452 Acc: 0.7902

Epoch 108/199
----------
train Loss: 0.4032 Acc: 0.9275
val Loss: 0.8430 Acc: 0.7971
```

```
Epoch 109/199
----------
train Loss: 0.3556 Acc: 0.9500
val Loss: 0.8440 Acc: 0.7912

Epoch 110/199
----------
train Loss: 0.3781 Acc: 0.9422
val Loss: 0.8491 Acc: 0.7912

Epoch 111/199
----------
train Loss: 0.3975 Acc: 0.9324
val Loss: 0.8428 Acc: 0.7941

Epoch 112/199
----------
train Loss: 0.3658 Acc: 0.9412
val Loss: 0.8387 Acc: 0.7971

Epoch 113/199
----------
train Loss: 0.3477 Acc: 0.9510
val Loss: 0.8367 Acc: 0.7912

Epoch 114/199
----------
train Loss: 0.3902 Acc: 0.9412
val Loss: 0.8393 Acc: 0.7922

Epoch 115/199
----------
train Loss: 0.3894 Acc: 0.9314
val Loss: 0.8395 Acc: 0.7912

Epoch 116/199
----------
train Loss: 0.4131 Acc: 0.9382
val Loss: 0.8390 Acc: 0.7961

Epoch 117/199
----------
train Loss: 0.3654 Acc: 0.9451
val Loss: 0.8366 Acc: 0.7951

Epoch 118/199
----------
```

```
train Loss: 0.3776 Acc: 0.9382
val Loss: 0.8374 Acc: 0.7990

Epoch 119/199
----------
train Loss: 0.3712 Acc: 0.9392
val Loss: 0.8449 Acc: 0.7941

Epoch 120/199
----------
train Loss: 0.3853 Acc: 0.9392
val Loss: 0.8451 Acc: 0.7961

Epoch 121/199
----------
train Loss: 0.3655 Acc: 0.9382
val Loss: 0.8400 Acc: 0.7971

Epoch 122/199
----------
train Loss: 0.3933 Acc: 0.9441
val Loss: 0.8440 Acc: 0.7922

Epoch 123/199
----------
train Loss: 0.3730 Acc: 0.9431
val Loss: 0.8399 Acc: 0.7990

Epoch 124/199
----------
train Loss: 0.3344 Acc: 0.9461
val Loss: 0.8377 Acc: 0.7971

Epoch 125/199
----------
train Loss: 0.4217 Acc: 0.9235
val Loss: 0.8398 Acc: 0.7922

Epoch 126/199
----------
train Loss: 0.3726 Acc: 0.9373
val Loss: 0.8421 Acc: 0.7961

Epoch 127/199
----------
train Loss: 0.3788 Acc: 0.9382
val Loss: 0.8434 Acc: 0.8000
```

```
Epoch 128/199
----------
train Loss: 0.3359 Acc: 0.9451
val Loss: 0.8367 Acc: 0.8000

Epoch 129/199
----------
train Loss: 0.3554 Acc: 0.9441
val Loss: 0.8430 Acc: 0.7941

Epoch 130/199
----------
train Loss: 0.3421 Acc: 0.9480
val Loss: 0.8430 Acc: 0.7931

Epoch 131/199
----------
train Loss: 0.3747 Acc: 0.9422
val Loss: 0.8425 Acc: 0.7961

Epoch 132/199
----------
train Loss: 0.3460 Acc: 0.9480
val Loss: 0.8407 Acc: 0.7941

Epoch 133/199
----------
train Loss: 0.3476 Acc: 0.9402
val Loss: 0.8378 Acc: 0.7951

Epoch 134/199
----------
train Loss: 0.3410 Acc: 0.9490
val Loss: 0.8370 Acc: 0.7990

Epoch 135/199
----------
train Loss: 0.3360 Acc: 0.9520
val Loss: 0.8431 Acc: 0.7971

Epoch 136/199
----------
train Loss: 0.3532 Acc: 0.9500
val Loss: 0.8425 Acc: 0.7922

Epoch 137/199
----------
train Loss: 0.3655 Acc: 0.9441
```

```
val Loss: 0.8388 Acc: 0.7922

Epoch 138/199
----------
train Loss: 0.3634 Acc: 0.9451
val Loss: 0.8400 Acc: 0.7951

Epoch 139/199
----------
train Loss: 0.3530 Acc: 0.9510
val Loss: 0.8362 Acc: 0.7931

Epoch 140/199
----------
train Loss: 0.3265 Acc: 0.9618
val Loss: 0.8365 Acc: 0.7980

Epoch 141/199
----------
train Loss: 0.3588 Acc: 0.9451
val Loss: 0.8373 Acc: 0.7961

Epoch 142/199
----------
train Loss: 0.3743 Acc: 0.9275
val Loss: 0.8353 Acc: 0.7951

Epoch 143/199
----------
train Loss: 0.3650 Acc: 0.9412
val Loss: 0.8411 Acc: 0.7971

Epoch 144/199
----------
train Loss: 0.3754 Acc: 0.9382
val Loss: 0.8389 Acc: 0.7971

Epoch 145/199
----------
train Loss: 0.3967 Acc: 0.9363
val Loss: 0.8373 Acc: 0.7941

Epoch 146/199
----------
train Loss: 0.3502 Acc: 0.9451
val Loss: 0.8393 Acc: 0.8000

Epoch 147/199
```

```
----------
train Loss: 0.3402 Acc: 0.9520
val Loss: 0.8376 Acc: 0.7971

Epoch 148/199
----------
train Loss: 0.3091 Acc: 0.9647
val Loss: 0.8376 Acc: 0.7990

Epoch 149/199
----------
train Loss: 0.3174 Acc: 0.9549
val Loss: 0.8400 Acc: 0.7990

Epoch 150/199
----------
train Loss: 0.4095 Acc: 0.9343
val Loss: 0.8453 Acc: 0.7912

Epoch 151/199
----------
train Loss: 0.3929 Acc: 0.9412
val Loss: 0.8398 Acc: 0.7922

Epoch 152/199
----------
train Loss: 0.3611 Acc: 0.9471
val Loss: 0.8330 Acc: 0.7971

Epoch 153/199
----------
train Loss: 0.3684 Acc: 0.9373
val Loss: 0.8316 Acc: 0.7941

Epoch 154/199
----------
train Loss: 0.3547 Acc: 0.9392
val Loss: 0.8353 Acc: 0.8000

Epoch 155/199
----------
train Loss: 0.3536 Acc: 0.9471
val Loss: 0.8350 Acc: 0.7951

Epoch 156/199
----------
train Loss: 0.3669 Acc: 0.9471
val Loss: 0.8393 Acc: 0.7951
```

```
Epoch 157/199
----------
train Loss: 0.3987 Acc: 0.9304
val Loss: 0.8391 Acc: 0.7951

Epoch 158/199
----------
train Loss: 0.3833 Acc: 0.9343
val Loss: 0.8345 Acc: 0.7931

Epoch 159/199
----------
train Loss: 0.3608 Acc: 0.9500
val Loss: 0.8393 Acc: 0.7912

Epoch 160/199
----------
train Loss: 0.3820 Acc: 0.9363
val Loss: 0.8410 Acc: 0.7961

Epoch 161/199
----------
train Loss: 0.3983 Acc: 0.9294
val Loss: 0.8383 Acc: 0.7892

Epoch 162/199
----------
train Loss: 0.3605 Acc: 0.9402
val Loss: 0.8410 Acc: 0.7902

Epoch 163/199
----------
train Loss: 0.3227 Acc: 0.9578
val Loss: 0.8352 Acc: 0.7922

Epoch 164/199
----------
train Loss: 0.3506 Acc: 0.9529
val Loss: 0.8373 Acc: 0.7961

Epoch 165/199
----------
train Loss: 0.3462 Acc: 0.9480
val Loss: 0.8374 Acc: 0.7951

Epoch 166/199
----------
```

```
train Loss: 0.3663 Acc: 0.9402
val Loss: 0.8327 Acc: 0.8029

Epoch 167/199
----------
train Loss: 0.3568 Acc: 0.9500
val Loss: 0.8397 Acc: 0.7912

Epoch 168/199
----------
train Loss: 0.3592 Acc: 0.9461
val Loss: 0.8389 Acc: 0.7931

Epoch 169/199
----------
train Loss: 0.3619 Acc: 0.9422
val Loss: 0.8361 Acc: 0.7951

Epoch 170/199
----------
train Loss: 0.3644 Acc: 0.9422
val Loss: 0.8387 Acc: 0.7951

Epoch 171/199
----------
train Loss: 0.3583 Acc: 0.9294
val Loss: 0.8372 Acc: 0.7990

Epoch 172/199
----------
train Loss: 0.3168 Acc: 0.9627
val Loss: 0.8314 Acc: 0.7980

Epoch 173/199
----------
train Loss: 0.3773 Acc: 0.9461
val Loss: 0.8342 Acc: 0.7961

Epoch 174/199
----------
train Loss: 0.3731 Acc: 0.9382
val Loss: 0.8347 Acc: 0.7961

Epoch 175/199
----------
train Loss: 0.3558 Acc: 0.9441
val Loss: 0.8324 Acc: 0.7971
```

```
Epoch 176/199
----------
train Loss: 0.3510 Acc: 0.9490
val Loss: 0.8350 Acc: 0.7941

Epoch 177/199
----------
train Loss: 0.3476 Acc: 0.9510
val Loss: 0.8382 Acc: 0.7941

Epoch 178/199
----------
train Loss: 0.3565 Acc: 0.9520
val Loss: 0.8345 Acc: 0.8000

Epoch 179/199
----------
train Loss: 0.3517 Acc: 0.9510
val Loss: 0.8334 Acc: 0.7990

Epoch 180/199
----------
train Loss: 0.3577 Acc: 0.9490
val Loss: 0.8365 Acc: 0.7961

Epoch 181/199
----------
train Loss: 0.3496 Acc: 0.9441
val Loss: 0.8354 Acc: 0.7980

Epoch 182/199
----------
train Loss: 0.3709 Acc: 0.9392
val Loss: 0.8350 Acc: 0.8000

Epoch 183/199
----------
train Loss: 0.3740 Acc: 0.9441
val Loss: 0.8339 Acc: 0.7941

Epoch 184/199
----------
train Loss: 0.3320 Acc: 0.9510
val Loss: 0.8356 Acc: 0.7922

Epoch 185/199
----------
train Loss: 0.3351 Acc: 0.9529
```

```
val Loss: 0.8321 Acc: 0.7980

Epoch 186/199
----------
train Loss: 0.3526 Acc: 0.9431
val Loss: 0.8325 Acc: 0.7941

Epoch 187/199
----------
train Loss: 0.4063 Acc: 0.9343
val Loss: 0.8335 Acc: 0.7922

Epoch 188/199
----------
train Loss: 0.3833 Acc: 0.9373
val Loss: 0.8385 Acc: 0.7951

Epoch 189/199
----------
train Loss: 0.3564 Acc: 0.9471
val Loss: 0.8422 Acc: 0.7961

Epoch 190/199
----------
train Loss: 0.3463 Acc: 0.9461
val Loss: 0.8366 Acc: 0.7971

Epoch 191/199
----------
train Loss: 0.3160 Acc: 0.9578
val Loss: 0.8336 Acc: 0.7931

Epoch 192/199
----------
train Loss: 0.3915 Acc: 0.9402
val Loss: 0.8357 Acc: 0.7951

Epoch 193/199
----------
train Loss: 0.3690 Acc: 0.9441
val Loss: 0.8376 Acc: 0.8000

Epoch 194/199
----------
train Loss: 0.3561 Acc: 0.9441
val Loss: 0.8369 Acc: 0.7951

Epoch 195/199
```

```
----------
train Loss: 0.3283 Acc: 0.9647
val Loss: 0.8405 Acc: 0.7980

Epoch 196/199
----------
train Loss: 0.3810 Acc: 0.9284
val Loss: 0.8399 Acc: 0.7961

Epoch 197/199
----------
train Loss: 0.4119 Acc: 0.9275
val Loss: 0.8354 Acc: 0.7931

Epoch 198/199
----------
train Loss: 0.3580 Acc: 0.9441
val Loss: 0.8381 Acc: 0.7961

Epoch 199/199
----------
train Loss: 0.3720 Acc: 0.9422
val Loss: 0.8426 Acc: 0.7931

Training complete in 26m 20s
Best val Acc: 0.802941
```

- c: (6 points)

Repeating the exercise above except with uniform learning rate equal to 0.1 and 0.01 and no learning rate decay.

[10]:
```
LR_SET = [0.1, 0.01]

for lr in LR_SET:
    print(f'----- Performing experiment with fixed lr={lr} -----\n')

    # Load pretrained Resnet50 and change final layer
    model_ft = models.resnet50(pretrained=True)
    num_ftrs = model_ft.fc.in_features
    model_ft.fc = nn.Linear(num_ftrs, len(class_names))

    # Set up criterion
    criterion = nn.CrossEntropyLoss()

    # Observe that all parameters are being optimized
    optimizer_ft = optim.SGD(model_ft.parameters(), lr=lr, momentum=0.9)
```

```
    # Decay LR by a factor of 0.1 every 1000 epochs (never going to step,␣
↪setting artifically high for fixing lr)
    exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=1000,␣
↪gamma=0.1)

    # Send model to device
    model_ft = model_ft.to(device)

    # Train model
    model_ft = train_model(model_ft, criterion, optimizer_ft, exp_lr_scheduler,␣
↪num_epochs=200)
```

```
----- Performing experiment with fixed lr=0.1 -----

Epoch 0/199
----------
train Loss: 5.2498 Acc: 0.0196
val Loss: 749843.8510 Acc: 0.0098

Epoch 1/199
----------
train Loss: 5.0352 Acc: 0.0206
val Loss: 795.6690 Acc: 0.0108

Epoch 2/199
----------
train Loss: 4.4090 Acc: 0.0363
val Loss: 10.2650 Acc: 0.0373

Epoch 3/199
----------
train Loss: 4.0696 Acc: 0.0422
val Loss: 5.0215 Acc: 0.0500

Epoch 4/199
----------
train Loss: 3.8868 Acc: 0.0559
val Loss: 3.8237 Acc: 0.0824

Epoch 5/199
----------
train Loss: 3.6877 Acc: 0.0882
val Loss: 4.1769 Acc: 0.1176

Epoch 6/199
----------
train Loss: 3.5217 Acc: 0.1245
```

```
val Loss: 3.7040 Acc: 0.1314

Epoch 7/199
----------
train Loss: 3.3368 Acc: 0.1480
val Loss: 3.5623 Acc: 0.1686

Epoch 8/199
----------
train Loss: 3.2987 Acc: 0.1647
val Loss: 3.5789 Acc: 0.1647

Epoch 9/199
----------
train Loss: 3.2518 Acc: 0.1667
val Loss: 3.5565 Acc: 0.1735

Epoch 10/199
----------
train Loss: 3.0891 Acc: 0.1941
val Loss: 3.9557 Acc: 0.1451

Epoch 11/199
----------
train Loss: 3.0188 Acc: 0.2392
val Loss: 3.5213 Acc: 0.1892

Epoch 12/199
----------
train Loss: 2.8823 Acc: 0.2324
val Loss: 3.8216 Acc: 0.1765

Epoch 13/199
----------
train Loss: 2.7834 Acc: 0.2480
val Loss: 3.6240 Acc: 0.2039

Epoch 14/199
----------
train Loss: 2.6396 Acc: 0.3029
val Loss: 3.1627 Acc: 0.2500

Epoch 15/199
----------
train Loss: 2.6257 Acc: 0.3069
val Loss: 3.8536 Acc: 0.2020

Epoch 16/199
```

```
----------
train Loss: 2.5366 Acc: 0.3186
val Loss: 3.5336 Acc: 0.2461

Epoch 17/199
----------
train Loss: 2.4416 Acc: 0.3324
val Loss: 3.7555 Acc: 0.2059

Epoch 18/199
----------
train Loss: 2.3806 Acc: 0.3667
val Loss: 3.2821 Acc: 0.2520

Epoch 19/199
----------
train Loss: 2.2433 Acc: 0.3735
val Loss: 3.3031 Acc: 0.2863

Epoch 20/199
----------
train Loss: 2.2374 Acc: 0.3784
val Loss: 3.2288 Acc: 0.2725

Epoch 21/199
----------
train Loss: 2.0776 Acc: 0.4314
val Loss: 3.0558 Acc: 0.3157

Epoch 22/199
----------
train Loss: 2.0227 Acc: 0.4324
val Loss: 4.3364 Acc: 0.2569

Epoch 23/199
----------
train Loss: 2.0176 Acc: 0.4343
val Loss: 3.0314 Acc: 0.3284

Epoch 24/199
----------
train Loss: 1.8823 Acc: 0.4598
val Loss: 3.1605 Acc: 0.3176

Epoch 25/199
----------
train Loss: 1.9202 Acc: 0.4500
val Loss: 3.2581 Acc: 0.3049
```

```
Epoch 26/199
----------
train Loss: 1.8703 Acc: 0.4863
val Loss: 5.9186 Acc: 0.2118

Epoch 27/199
----------
train Loss: 1.8041 Acc: 0.4824
val Loss: 3.4629 Acc: 0.3078

Epoch 28/199
----------
train Loss: 1.7214 Acc: 0.5235
val Loss: 3.4604 Acc: 0.3127

Epoch 29/199
----------
train Loss: 1.6890 Acc: 0.5196
val Loss: 3.4055 Acc: 0.3235

Epoch 30/199
----------
train Loss: 1.6670 Acc: 0.5147
val Loss: 3.7097 Acc: 0.3049

Epoch 31/199
----------
train Loss: 1.6647 Acc: 0.5088
val Loss: 3.5649 Acc: 0.3000

Epoch 32/199
----------
train Loss: 1.6227 Acc: 0.5402
val Loss: 4.2674 Acc: 0.2627

Epoch 33/199
----------
train Loss: 1.5345 Acc: 0.5422
val Loss: 3.2455 Acc: 0.3412

Epoch 34/199
----------
train Loss: 1.5738 Acc: 0.5412
val Loss: 3.9550 Acc: 0.2725

Epoch 35/199
----------
```

```
train Loss: 1.4978 Acc: 0.5627
val Loss: 3.4374 Acc: 0.3176

Epoch 36/199
----------
train Loss: 1.3860 Acc: 0.5833
val Loss: 3.1897 Acc: 0.3814

Epoch 37/199
----------
train Loss: 1.3081 Acc: 0.6284
val Loss: 3.2801 Acc: 0.3510

Epoch 38/199
----------
train Loss: 1.3171 Acc: 0.6088
val Loss: 3.6173 Acc: 0.3392

Epoch 39/199
----------
train Loss: 1.3667 Acc: 0.6098
val Loss: 3.9286 Acc: 0.2696

Epoch 40/199
----------
train Loss: 1.2711 Acc: 0.6225
val Loss: 3.7613 Acc: 0.3225

Epoch 41/199
----------
train Loss: 1.1994 Acc: 0.6245
val Loss: 3.3390 Acc: 0.3578

Epoch 42/199
----------
train Loss: 1.1891 Acc: 0.6431
val Loss: 3.0645 Acc: 0.3755

Epoch 43/199
----------
train Loss: 1.0711 Acc: 0.6961
val Loss: 3.5324 Acc: 0.3402

Epoch 44/199
----------
train Loss: 1.1806 Acc: 0.6657
val Loss: 3.4956 Acc: 0.3520
```

```
Epoch 45/199
----------
train Loss: 0.9744 Acc: 0.7167
val Loss: 3.2732 Acc: 0.4020

Epoch 46/199
----------
train Loss: 1.0943 Acc: 0.6824
val Loss: 3.3572 Acc: 0.4020

Epoch 47/199
----------
train Loss: 0.9824 Acc: 0.7147
val Loss: 3.2358 Acc: 0.3843

Epoch 48/199
----------
train Loss: 0.9305 Acc: 0.7284
val Loss: 3.2426 Acc: 0.3843

Epoch 49/199
----------
train Loss: 0.9876 Acc: 0.7167
val Loss: 7.3918 Acc: 0.2206

Epoch 50/199
----------
train Loss: 0.9070 Acc: 0.7343
val Loss: 3.2567 Acc: 0.3941

Epoch 51/199
----------
train Loss: 0.9001 Acc: 0.7471
val Loss: 3.6451 Acc: 0.3500

Epoch 52/199
----------
train Loss: 0.9538 Acc: 0.7167
val Loss: 3.7217 Acc: 0.3510

Epoch 53/199
----------
train Loss: 0.9866 Acc: 0.7069
val Loss: 3.9140 Acc: 0.3725

Epoch 54/199
----------
train Loss: 0.9311 Acc: 0.7304
```

```
val Loss: 3.9310 Acc: 0.3451

Epoch 55/199
----------
train Loss: 0.8976 Acc: 0.7265
val Loss: 3.3994 Acc: 0.3941

Epoch 56/199
----------
train Loss: 0.9367 Acc: 0.7431
val Loss: 3.5609 Acc: 0.3980

Epoch 57/199
----------
train Loss: 0.8457 Acc: 0.7539
val Loss: 3.2680 Acc: 0.3882

Epoch 58/199
----------
train Loss: 0.7462 Acc: 0.7686
val Loss: 4.0766 Acc: 0.3814

Epoch 59/199
----------
train Loss: 0.6597 Acc: 0.7912
val Loss: 3.4203 Acc: 0.4069

Epoch 60/199
----------
train Loss: 0.7366 Acc: 0.7814
val Loss: 3.4418 Acc: 0.4314

Epoch 61/199
----------
train Loss: 0.6132 Acc: 0.8147
val Loss: 3.6396 Acc: 0.4137

Epoch 62/199
----------
train Loss: 0.6955 Acc: 0.7961
val Loss: 3.6491 Acc: 0.3990

Epoch 63/199
----------
train Loss: 0.6091 Acc: 0.8118
val Loss: 3.4732 Acc: 0.4078

Epoch 64/199
```

```
----------
train Loss: 0.5812 Acc: 0.8255
val Loss: 3.5415 Acc: 0.4137

Epoch 65/199
----------
train Loss: 0.6948 Acc: 0.7971
val Loss: 4.1944 Acc: 0.3892

Epoch 66/199
----------
train Loss: 0.6311 Acc: 0.8314
val Loss: 3.3628 Acc: 0.4333

Epoch 67/199
----------
train Loss: 0.5960 Acc: 0.8088
val Loss: 3.6900 Acc: 0.4324

Epoch 68/199
----------
train Loss: 0.6471 Acc: 0.8157
val Loss: 4.1339 Acc: 0.3735

Epoch 69/199
----------
train Loss: 0.5607 Acc: 0.8284
val Loss: 3.7407 Acc: 0.4275

Epoch 70/199
----------
train Loss: 0.5317 Acc: 0.8461
val Loss: 4.2721 Acc: 0.3775

Epoch 71/199
----------
train Loss: 0.5809 Acc: 0.8343
val Loss: 3.7498 Acc: 0.4314

Epoch 72/199
----------
train Loss: 0.6328 Acc: 0.8069
val Loss: 3.6281 Acc: 0.4049

Epoch 73/199
----------
train Loss: 0.5156 Acc: 0.8490
val Loss: 3.1814 Acc: 0.4716
```

```
Epoch 74/199
----------
train Loss: 0.5180 Acc: 0.8559
val Loss: 3.4923 Acc: 0.4363

Epoch 75/199
----------
train Loss: 0.4815 Acc: 0.8520
val Loss: 3.8130 Acc: 0.4304

Epoch 76/199
----------
train Loss: 0.5183 Acc: 0.8392
val Loss: 3.7529 Acc: 0.4343

Epoch 77/199
----------
train Loss: 0.4936 Acc: 0.8529
val Loss: 3.1415 Acc: 0.4667

Epoch 78/199
----------
train Loss: 0.4275 Acc: 0.8775
val Loss: 3.5126 Acc: 0.4745

Epoch 79/199
----------
train Loss: 0.5677 Acc: 0.8275
val Loss: 4.1178 Acc: 0.3980

Epoch 80/199
----------
train Loss: 0.5304 Acc: 0.8490
val Loss: 3.6072 Acc: 0.4569

Epoch 81/199
----------
train Loss: 0.4828 Acc: 0.8471
val Loss: 3.6504 Acc: 0.4314

Epoch 82/199
----------
train Loss: 0.4239 Acc: 0.8814
val Loss: 3.3109 Acc: 0.4647

Epoch 83/199
----------
```

```
train Loss: 0.4520 Acc: 0.8676
val Loss: 3.5952 Acc: 0.4353

Epoch 84/199
----------
train Loss: 0.4575 Acc: 0.8667
val Loss: 3.8310 Acc: 0.4069

Epoch 85/199
----------
train Loss: 0.4239 Acc: 0.8716
val Loss: 3.6362 Acc: 0.4402

Epoch 86/199
----------
train Loss: 0.4274 Acc: 0.8745
val Loss: 3.9087 Acc: 0.4098

Epoch 87/199
----------
train Loss: 0.4322 Acc: 0.8578
val Loss: 4.0313 Acc: 0.4363

Epoch 88/199
----------
train Loss: 0.4434 Acc: 0.8784
val Loss: 3.4245 Acc: 0.4647

Epoch 89/199
----------
train Loss: 0.3615 Acc: 0.8971
val Loss: 3.3488 Acc: 0.4618

Epoch 90/199
----------
train Loss: 0.4242 Acc: 0.8686
val Loss: 3.8551 Acc: 0.4471

Epoch 91/199
----------
train Loss: 0.4617 Acc: 0.8667
val Loss: 3.7298 Acc: 0.4441

Epoch 92/199
----------
train Loss: 0.3903 Acc: 0.8902
val Loss: 3.5832 Acc: 0.4480
```

```
Epoch 93/199
----------
train Loss: 0.3795 Acc: 0.8941
val Loss: 3.9082 Acc: 0.4265

Epoch 94/199
----------
train Loss: 0.3918 Acc: 0.8833
val Loss: 3.7573 Acc: 0.4206

Epoch 95/199
----------
train Loss: 0.4024 Acc: 0.8755
val Loss: 3.7049 Acc: 0.4529

Epoch 96/199
----------
train Loss: 0.3746 Acc: 0.8824
val Loss: 3.4581 Acc: 0.4598

Epoch 97/199
----------
train Loss: 0.4179 Acc: 0.8853
val Loss: 3.5201 Acc: 0.4569

Epoch 98/199
----------
train Loss: 0.3839 Acc: 0.8863
val Loss: 3.5684 Acc: 0.4363

Epoch 99/199
----------
train Loss: 0.3827 Acc: 0.8990
val Loss: 3.7491 Acc: 0.4618

Epoch 100/199
----------
train Loss: 0.3857 Acc: 0.8902
val Loss: 3.7695 Acc: 0.4304

Epoch 101/199
----------
train Loss: 0.3273 Acc: 0.9059
val Loss: 3.4107 Acc: 0.4549

Epoch 102/199
----------
train Loss: 0.3823 Acc: 0.8745
```

```
val Loss: 3.7824 Acc: 0.4284

Epoch 103/199
----------
train Loss: 0.3889 Acc: 0.8755
val Loss: 3.7604 Acc: 0.4353

Epoch 104/199
----------
train Loss: 0.3542 Acc: 0.8941
val Loss: 4.1533 Acc: 0.4029

Epoch 105/199
----------
train Loss: 0.3811 Acc: 0.8971
val Loss: 3.7874 Acc: 0.4314

Epoch 106/199
----------
train Loss: 0.4145 Acc: 0.8853
val Loss: 3.4110 Acc: 0.4500

Epoch 107/199
----------
train Loss: 0.3383 Acc: 0.8951
val Loss: 3.6860 Acc: 0.4637

Epoch 108/199
----------
train Loss: 0.2602 Acc: 0.9176
val Loss: 4.0530 Acc: 0.4500

Epoch 109/199
----------
train Loss: 0.3316 Acc: 0.9098
val Loss: 3.6794 Acc: 0.4637

Epoch 110/199
----------
train Loss: 0.3544 Acc: 0.8912
val Loss: 4.0061 Acc: 0.4382

Epoch 111/199
----------
train Loss: 0.4155 Acc: 0.8814
val Loss: 4.0666 Acc: 0.4441

Epoch 112/199
```

```
----------
train Loss: 0.3127 Acc: 0.9078
val Loss: 3.9296 Acc: 0.4490

Epoch 113/199
----------
train Loss: 0.3207 Acc: 0.9029
val Loss: 3.5122 Acc: 0.5010

Epoch 114/199
----------
train Loss: 0.2970 Acc: 0.9088
val Loss: 3.5558 Acc: 0.4686

Epoch 115/199
----------
train Loss: 0.2980 Acc: 0.9137
val Loss: 3.8435 Acc: 0.4598

Epoch 116/199
----------
train Loss: 0.2943 Acc: 0.9294
val Loss: 3.5092 Acc: 0.4853

Epoch 117/199
----------
train Loss: 0.2666 Acc: 0.9294
val Loss: 3.7671 Acc: 0.4686

Epoch 118/199
----------
train Loss: 0.2975 Acc: 0.9078
val Loss: 3.5571 Acc: 0.4794

Epoch 119/199
----------
train Loss: 0.3030 Acc: 0.9118
val Loss: 3.6217 Acc: 0.4745

Epoch 120/199
----------
train Loss: 0.2654 Acc: 0.9294
val Loss: 3.7679 Acc: 0.4559

Epoch 121/199
----------
train Loss: 0.2541 Acc: 0.9216
val Loss: 3.6834 Acc: 0.4882
```

```
Epoch 122/199
----------
train Loss: 0.3081 Acc: 0.9069
val Loss: 3.7195 Acc: 0.4627

Epoch 123/199
----------
train Loss: 0.3327 Acc: 0.8941
val Loss: 3.7218 Acc: 0.4637

Epoch 124/199
----------
train Loss: 0.2549 Acc: 0.9245
val Loss: 3.6883 Acc: 0.4618

Epoch 125/199
----------
train Loss: 0.2322 Acc: 0.9304
val Loss: 3.6057 Acc: 0.4559

Epoch 126/199
----------
train Loss: 0.2570 Acc: 0.9265
val Loss: 3.4722 Acc: 0.4922

Epoch 127/199
----------
train Loss: 0.2894 Acc: 0.9225
val Loss: 3.5445 Acc: 0.4882

Epoch 128/199
----------
train Loss: 0.2819 Acc: 0.9186
val Loss: 3.7723 Acc: 0.4696

Epoch 129/199
----------
train Loss: 0.2217 Acc: 0.9373
val Loss: 3.9951 Acc: 0.4784

Epoch 130/199
----------
train Loss: 0.2032 Acc: 0.9500
val Loss: 3.3078 Acc: 0.5000

Epoch 131/199
----------
```

```
train Loss: 0.2573 Acc: 0.9304
val Loss: 3.7207 Acc: 0.4873

Epoch 132/199
----------
train Loss: 0.2524 Acc: 0.9255
val Loss: 3.7365 Acc: 0.4784

Epoch 133/199
----------
train Loss: 0.2628 Acc: 0.9157
val Loss: 3.3411 Acc: 0.5069

Epoch 134/199
----------
train Loss: 0.2339 Acc: 0.9265
val Loss: 3.5268 Acc: 0.4941

Epoch 135/199
----------
train Loss: 0.2771 Acc: 0.9265
val Loss: 3.8559 Acc: 0.4667

Epoch 136/199
----------
train Loss: 0.2307 Acc: 0.9225
val Loss: 3.6794 Acc: 0.4696

Epoch 137/199
----------
train Loss: 0.2065 Acc: 0.9431
val Loss: 3.5744 Acc: 0.4804

Epoch 138/199
----------
train Loss: 0.2897 Acc: 0.9186
val Loss: 3.4794 Acc: 0.4931

Epoch 139/199
----------
train Loss: 0.2878 Acc: 0.9108
val Loss: 3.6485 Acc: 0.4657

Epoch 140/199
----------
train Loss: 0.2893 Acc: 0.9108
val Loss: 3.4630 Acc: 0.4824
```

```
Epoch 141/199
----------
train Loss: 0.2526 Acc: 0.9304
val Loss: 3.7093 Acc: 0.4735

Epoch 142/199
----------
train Loss: 0.2526 Acc: 0.9333
val Loss: 3.6844 Acc: 0.4775

Epoch 143/199
----------
train Loss: 0.2365 Acc: 0.9275
val Loss: 3.5997 Acc: 0.4696

Epoch 144/199
----------
train Loss: 0.2528 Acc: 0.9284
val Loss: 3.4315 Acc: 0.4863

Epoch 145/199
----------
train Loss: 0.2260 Acc: 0.9294
val Loss: 3.4859 Acc: 0.4980

Epoch 146/199
----------
train Loss: 0.2481 Acc: 0.9333
val Loss: 3.6822 Acc: 0.4755

Epoch 147/199
----------
train Loss: 0.2271 Acc: 0.9265
val Loss: 3.7619 Acc: 0.4824

Epoch 148/199
----------
train Loss: 0.2657 Acc: 0.9206
val Loss: 3.4394 Acc: 0.4824

Epoch 149/199
----------
train Loss: 0.2411 Acc: 0.9314
val Loss: 3.6095 Acc: 0.4745

Epoch 150/199
----------
train Loss: 0.2251 Acc: 0.9373
```

```
val Loss: 3.4996 Acc: 0.4912

Epoch 151/199
----------
train Loss: 0.2274 Acc: 0.9373
val Loss: 3.6188 Acc: 0.4912

Epoch 152/199
----------
train Loss: 0.2494 Acc: 0.9265
val Loss: 3.6437 Acc: 0.4882

Epoch 153/199
----------
train Loss: 0.2674 Acc: 0.9255
val Loss: 3.8681 Acc: 0.4765

Epoch 154/199
----------
train Loss: 0.2132 Acc: 0.9343
val Loss: 3.4962 Acc: 0.5000

Epoch 155/199
----------
train Loss: 0.1827 Acc: 0.9500
val Loss: 3.3931 Acc: 0.4902

Epoch 156/199
----------
train Loss: 0.1863 Acc: 0.9451
val Loss: 3.7097 Acc: 0.4647

Epoch 157/199
----------
train Loss: 0.2251 Acc: 0.9422
val Loss: 3.5575 Acc: 0.4843

Epoch 158/199
----------
train Loss: 0.1920 Acc: 0.9480
val Loss: 3.5869 Acc: 0.4951

Epoch 159/199
----------
train Loss: 0.2669 Acc: 0.9216
val Loss: 3.7659 Acc: 0.4873

Epoch 160/199
```

```
----------
train Loss: 0.2064 Acc: 0.9373
val Loss: 3.5788 Acc: 0.4922

Epoch 161/199
----------
train Loss: 0.2289 Acc: 0.9275
val Loss: 3.7677 Acc: 0.4686

Epoch 162/199
----------
train Loss: 0.2223 Acc: 0.9294
val Loss: 4.0917 Acc: 0.4559

Epoch 163/199
----------
train Loss: 0.2547 Acc: 0.9225
val Loss: 3.4315 Acc: 0.5127

Epoch 164/199
----------
train Loss: 0.1909 Acc: 0.9441
val Loss: 3.6746 Acc: 0.4775

Epoch 165/199
----------
train Loss: 0.2565 Acc: 0.9245
val Loss: 3.2990 Acc: 0.5147

Epoch 166/199
----------
train Loss: 0.1984 Acc: 0.9500
val Loss: 3.9395 Acc: 0.4735

Epoch 167/199
----------
train Loss: 0.2030 Acc: 0.9422
val Loss: 3.1920 Acc: 0.5196

Epoch 168/199
----------
train Loss: 0.2041 Acc: 0.9402
val Loss: 3.5791 Acc: 0.4814

Epoch 169/199
----------
train Loss: 0.1433 Acc: 0.9637
val Loss: 3.3980 Acc: 0.4980
```

```
Epoch 170/199
----------
train Loss: 0.1438 Acc: 0.9627
val Loss: 3.2865 Acc: 0.5167

Epoch 171/199
----------
train Loss: 0.1830 Acc: 0.9520
val Loss: 3.3839 Acc: 0.5157

Epoch 172/199
----------
train Loss: 0.1848 Acc: 0.9471
val Loss: 3.5344 Acc: 0.5049

Epoch 173/199
----------
train Loss: 0.1994 Acc: 0.9451
val Loss: 3.7443 Acc: 0.4873

Epoch 174/199
----------
train Loss: 0.2195 Acc: 0.9392
val Loss: 3.8128 Acc: 0.4804

Epoch 175/199
----------
train Loss: 0.2170 Acc: 0.9431
val Loss: 3.6214 Acc: 0.4961

Epoch 176/199
----------
train Loss: 0.2584 Acc: 0.9265
val Loss: 3.6759 Acc: 0.5010

Epoch 177/199
----------
train Loss: 0.2036 Acc: 0.9431
val Loss: 3.4050 Acc: 0.4980

Epoch 178/199
----------
train Loss: 0.1703 Acc: 0.9490
val Loss: 3.4007 Acc: 0.4912

Epoch 179/199
----------
```

```
train Loss: 0.1909 Acc: 0.9431
val Loss: 3.7264 Acc: 0.4931

Epoch 180/199
----------
train Loss: 0.2022 Acc: 0.9480
val Loss: 3.5900 Acc: 0.4951

Epoch 181/199
----------
train Loss: 0.2054 Acc: 0.9373
val Loss: 3.4971 Acc: 0.5108

Epoch 182/199
----------
train Loss: 0.1649 Acc: 0.9480
val Loss: 3.4710 Acc: 0.5196

Epoch 183/199
----------
train Loss: 0.1917 Acc: 0.9510
val Loss: 3.5507 Acc: 0.5049

Epoch 184/199
----------
train Loss: 0.1887 Acc: 0.9461
val Loss: 3.2824 Acc: 0.5147

Epoch 185/199
----------
train Loss: 0.1764 Acc: 0.9539
val Loss: 3.5367 Acc: 0.5000

Epoch 186/199
----------
train Loss: 0.1844 Acc: 0.9461
val Loss: 3.3338 Acc: 0.4990

Epoch 187/199
----------
train Loss: 0.2095 Acc: 0.9471
val Loss: 3.5892 Acc: 0.4951

Epoch 188/199
----------
train Loss: 0.1999 Acc: 0.9412
val Loss: 3.5408 Acc: 0.4912
```

```
Epoch 189/199
----------
train Loss: 0.1650 Acc: 0.9510
val Loss: 3.4062 Acc: 0.5088

Epoch 190/199
----------
train Loss: 0.1944 Acc: 0.9392
val Loss: 3.4182 Acc: 0.5010

Epoch 191/199
----------
train Loss: 0.1182 Acc: 0.9637
val Loss: 3.7257 Acc: 0.4971

Epoch 192/199
----------
train Loss: 0.2010 Acc: 0.9343
val Loss: 3.7699 Acc: 0.4833

Epoch 193/199
----------
train Loss: 0.1785 Acc: 0.9510
val Loss: 3.6733 Acc: 0.4980

Epoch 194/199
----------
train Loss: 0.1558 Acc: 0.9569
val Loss: 3.9372 Acc: 0.4765

Epoch 195/199
----------
train Loss: 0.1507 Acc: 0.9559
val Loss: 3.5078 Acc: 0.5098

Epoch 196/199
----------
train Loss: 0.1335 Acc: 0.9608
val Loss: 3.6874 Acc: 0.4990

Epoch 197/199
----------
train Loss: 0.1836 Acc: 0.9500
val Loss: 3.5239 Acc: 0.5167

Epoch 198/199
----------
train Loss: 0.1490 Acc: 0.9520
```

```
val Loss: 3.6698 Acc: 0.5127

Epoch 199/199
----------
train Loss: 0.1720 Acc: 0.9510
val Loss: 3.6226 Acc: 0.5147

Training complete in 26m 32s
Best val Acc: 0.519608
----- Performing experiment with fixed lr=0.01 -----

Epoch 0/199
----------
train Loss: 4.6535 Acc: 0.0196
val Loss: 4.2255 Acc: 0.1118

Epoch 1/199
----------
train Loss: 3.8918 Acc: 0.1706
val Loss: 2.8557 Acc: 0.3490

Epoch 2/199
----------
train Loss: 2.7883 Acc: 0.3853
val Loss: 2.1084 Acc: 0.4941

Epoch 3/199
----------
train Loss: 1.9877 Acc: 0.5451
val Loss: 1.6396 Acc: 0.6049

Epoch 4/199
----------
train Loss: 1.5040 Acc: 0.6353
val Loss: 1.5362 Acc: 0.6049

Epoch 5/199
----------
train Loss: 1.1264 Acc: 0.7343
val Loss: 1.2352 Acc: 0.6990

Epoch 6/199
----------
train Loss: 0.8980 Acc: 0.8000
val Loss: 1.0860 Acc: 0.7098

Epoch 7/199
----------
```

```
train Loss: 0.6885 Acc: 0.8392
val Loss: 1.0197 Acc: 0.7382

Epoch 8/199
----------
train Loss: 0.6251 Acc: 0.8549
val Loss: 0.9793 Acc: 0.7480

Epoch 9/199
----------
train Loss: 0.6186 Acc: 0.8598
val Loss: 0.9167 Acc: 0.7686

Epoch 10/199
----------
train Loss: 0.4995 Acc: 0.8902
val Loss: 0.8827 Acc: 0.7784

Epoch 11/199
----------
train Loss: 0.4431 Acc: 0.9020
val Loss: 0.9743 Acc: 0.7549

Epoch 12/199
----------
train Loss: 0.4877 Acc: 0.8843
val Loss: 0.9543 Acc: 0.7412

Epoch 13/199
----------
train Loss: 0.4208 Acc: 0.8980
val Loss: 0.9568 Acc: 0.7490

Epoch 14/199
----------
train Loss: 0.3943 Acc: 0.9088
val Loss: 0.9183 Acc: 0.7608

Epoch 15/199
----------
train Loss: 0.3478 Acc: 0.9088
val Loss: 0.8956 Acc: 0.7667

Epoch 16/199
----------
train Loss: 0.3351 Acc: 0.9137
val Loss: 0.9187 Acc: 0.7676
```

```
Epoch 17/199
----------
train Loss: 0.3085 Acc: 0.9255
val Loss: 0.9120 Acc: 0.7676

Epoch 18/199
----------
train Loss: 0.2862 Acc: 0.9333
val Loss: 0.9498 Acc: 0.7500

Epoch 19/199
----------
train Loss: 0.2495 Acc: 0.9402
val Loss: 0.9972 Acc: 0.7598

Epoch 20/199
----------
train Loss: 0.3256 Acc: 0.9275
val Loss: 0.9184 Acc: 0.7618

Epoch 21/199
----------
train Loss: 0.2706 Acc: 0.9314
val Loss: 0.9351 Acc: 0.7676

Epoch 22/199
----------
train Loss: 0.3113 Acc: 0.9324
val Loss: 0.8850 Acc: 0.7745

Epoch 23/199
----------
train Loss: 0.2549 Acc: 0.9441
val Loss: 0.8549 Acc: 0.7863

Epoch 24/199
----------
train Loss: 0.2683 Acc: 0.9314
val Loss: 0.9606 Acc: 0.7578

Epoch 25/199
----------
train Loss: 0.2679 Acc: 0.9314
val Loss: 0.9476 Acc: 0.7647

Epoch 26/199
----------
train Loss: 0.2703 Acc: 0.9353
```

```
val Loss: 0.8371 Acc: 0.7833

Epoch 27/199
----------
train Loss: 0.2684 Acc: 0.9382
val Loss: 0.8780 Acc: 0.7833

Epoch 28/199
----------
train Loss: 0.2319 Acc: 0.9461
val Loss: 0.8606 Acc: 0.7873

Epoch 29/199
----------
train Loss: 0.2271 Acc: 0.9471
val Loss: 0.8427 Acc: 0.7833

Epoch 30/199
----------
train Loss: 0.2447 Acc: 0.9422
val Loss: 0.8703 Acc: 0.7716

Epoch 31/199
----------
train Loss: 0.2019 Acc: 0.9480
val Loss: 0.8136 Acc: 0.7961

Epoch 32/199
----------
train Loss: 0.1975 Acc: 0.9490
val Loss: 0.8630 Acc: 0.7735

Epoch 33/199
----------
train Loss: 0.1833 Acc: 0.9559
val Loss: 0.8066 Acc: 0.7961

Epoch 34/199
----------
train Loss: 0.1863 Acc: 0.9490
val Loss: 0.8058 Acc: 0.7980

Epoch 35/199
----------
train Loss: 0.1897 Acc: 0.9520
val Loss: 0.7909 Acc: 0.7971

Epoch 36/199
```

```
----------
train Loss: 0.2230 Acc: 0.9382
val Loss: 0.9319 Acc: 0.7618

Epoch 37/199
----------
train Loss: 0.2080 Acc: 0.9441
val Loss: 0.9643 Acc: 0.7647

Epoch 38/199
----------
train Loss: 0.1903 Acc: 0.9549
val Loss: 0.8941 Acc: 0.7794

Epoch 39/199
----------
train Loss: 0.1582 Acc: 0.9647
val Loss: 0.7836 Acc: 0.8000

Epoch 40/199
----------
train Loss: 0.2118 Acc: 0.9451
val Loss: 0.7886 Acc: 0.8020

Epoch 41/199
----------
train Loss: 0.2027 Acc: 0.9490
val Loss: 0.8985 Acc: 0.7696

Epoch 42/199
----------
train Loss: 0.2023 Acc: 0.9529
val Loss: 0.9072 Acc: 0.7745

Epoch 43/199
----------
train Loss: 0.1558 Acc: 0.9578
val Loss: 0.8512 Acc: 0.7892

Epoch 44/199
----------
train Loss: 0.1878 Acc: 0.9500
val Loss: 0.9067 Acc: 0.7873

Epoch 45/199
----------
train Loss: 0.2235 Acc: 0.9422
val Loss: 0.8608 Acc: 0.7971
```

```
Epoch 46/199
----------
train Loss: 0.1827 Acc: 0.9520
val Loss: 0.8987 Acc: 0.7745

Epoch 47/199
----------
train Loss: 0.1603 Acc: 0.9569
val Loss: 0.8393 Acc: 0.7990

Epoch 48/199
----------
train Loss: 0.1824 Acc: 0.9520
val Loss: 0.8874 Acc: 0.7755

Epoch 49/199
----------
train Loss: 0.2245 Acc: 0.9402
val Loss: 0.9322 Acc: 0.7775

Epoch 50/199
----------
train Loss: 0.2070 Acc: 0.9461
val Loss: 0.8474 Acc: 0.7980

Epoch 51/199
----------
train Loss: 0.1970 Acc: 0.9490
val Loss: 0.8773 Acc: 0.7824

Epoch 52/199
----------
train Loss: 0.1359 Acc: 0.9676
val Loss: 0.8410 Acc: 0.7951

Epoch 53/199
----------
train Loss: 0.1690 Acc: 0.9539
val Loss: 0.9226 Acc: 0.7814

Epoch 54/199
----------
train Loss: 0.2067 Acc: 0.9451
val Loss: 0.8882 Acc: 0.7951

Epoch 55/199
----------
```

```
train Loss: 0.1611 Acc: 0.9569
val Loss: 0.8185 Acc: 0.8010

Epoch 56/199
----------
train Loss: 0.1533 Acc: 0.9618
val Loss: 0.8919 Acc: 0.7873

Epoch 57/199
----------
train Loss: 0.1513 Acc: 0.9549
val Loss: 0.9020 Acc: 0.7882

Epoch 58/199
----------
train Loss: 0.1412 Acc: 0.9627
val Loss: 0.9397 Acc: 0.7873

Epoch 59/199
----------
train Loss: 0.1849 Acc: 0.9500
val Loss: 0.8706 Acc: 0.8049

Epoch 60/199
----------
train Loss: 0.1615 Acc: 0.9539
val Loss: 0.8590 Acc: 0.7863

Epoch 61/199
----------
train Loss: 0.1412 Acc: 0.9608
val Loss: 0.9284 Acc: 0.7725

Epoch 62/199
----------
train Loss: 0.1583 Acc: 0.9578
val Loss: 0.8657 Acc: 0.7882

Epoch 63/199
----------
train Loss: 0.1539 Acc: 0.9618
val Loss: 0.8341 Acc: 0.7853

Epoch 64/199
----------
train Loss: 0.1560 Acc: 0.9539
val Loss: 0.8384 Acc: 0.7951
```

```
Epoch 65/199
----------
train Loss: 0.1303 Acc: 0.9627
val Loss: 0.8228 Acc: 0.7951

Epoch 66/199
----------
train Loss: 0.1751 Acc: 0.9510
val Loss: 0.8517 Acc: 0.8059

Epoch 67/199
----------
train Loss: 0.1158 Acc: 0.9706
val Loss: 0.9028 Acc: 0.7912

Epoch 68/199
----------
train Loss: 0.1247 Acc: 0.9696
val Loss: 0.8860 Acc: 0.7941

Epoch 69/199
----------
train Loss: 0.1291 Acc: 0.9667
val Loss: 0.8688 Acc: 0.8098

Epoch 70/199
----------
train Loss: 0.1083 Acc: 0.9735
val Loss: 0.8919 Acc: 0.8118

Epoch 71/199
----------
train Loss: 0.1105 Acc: 0.9696
val Loss: 0.9020 Acc: 0.7931

Epoch 72/199
----------
train Loss: 0.1322 Acc: 0.9706
val Loss: 0.9909 Acc: 0.7696

Epoch 73/199
----------
train Loss: 0.1347 Acc: 0.9618
val Loss: 0.9382 Acc: 0.7843

Epoch 74/199
----------
train Loss: 0.1670 Acc: 0.9539
```

```
val Loss: 0.9405 Acc: 0.7824

Epoch 75/199
----------
train Loss: 0.1615 Acc: 0.9510
val Loss: 0.9438 Acc: 0.7765

Epoch 76/199
----------
train Loss: 0.1284 Acc: 0.9706
val Loss: 0.9022 Acc: 0.7892

Epoch 77/199
----------
train Loss: 0.1924 Acc: 0.9480
val Loss: 0.9512 Acc: 0.7843

Epoch 78/199
----------
train Loss: 0.1622 Acc: 0.9569
val Loss: 0.9486 Acc: 0.7843

Epoch 79/199
----------
train Loss: 0.1284 Acc: 0.9686
val Loss: 0.9317 Acc: 0.7902

Epoch 80/199
----------
train Loss: 0.1330 Acc: 0.9686
val Loss: 0.9510 Acc: 0.7941

Epoch 81/199
----------
train Loss: 0.1171 Acc: 0.9676
val Loss: 0.8280 Acc: 0.7961

Epoch 82/199
----------
train Loss: 0.1026 Acc: 0.9735
val Loss: 0.8234 Acc: 0.8049

Epoch 83/199
----------
train Loss: 0.1035 Acc: 0.9735
val Loss: 0.8953 Acc: 0.7745

Epoch 84/199
```

```
----------
train Loss: 0.0987 Acc: 0.9745
val Loss: 0.8729 Acc: 0.7833

Epoch 85/199
----------
train Loss: 0.0980 Acc: 0.9716
val Loss: 0.8394 Acc: 0.8049

Epoch 86/199
----------
train Loss: 0.1200 Acc: 0.9686
val Loss: 0.8471 Acc: 0.8088

Epoch 87/199
----------
train Loss: 0.1150 Acc: 0.9706
val Loss: 0.8261 Acc: 0.8020

Epoch 88/199
----------
train Loss: 0.0916 Acc: 0.9765
val Loss: 0.8530 Acc: 0.8010

Epoch 89/199
----------
train Loss: 0.1133 Acc: 0.9706
val Loss: 0.8710 Acc: 0.8039

Epoch 90/199
----------
train Loss: 0.0902 Acc: 0.9755
val Loss: 0.8878 Acc: 0.7902

Epoch 91/199
----------
train Loss: 0.0904 Acc: 0.9784
val Loss: 0.9202 Acc: 0.7863

Epoch 92/199
----------
train Loss: 0.0986 Acc: 0.9765
val Loss: 0.8792 Acc: 0.7941

Epoch 93/199
----------
train Loss: 0.0891 Acc: 0.9735
val Loss: 0.8462 Acc: 0.8020
```

```
Epoch 94/199
----------
train Loss: 0.0834 Acc: 0.9775
val Loss: 0.8161 Acc: 0.7990

Epoch 95/199
----------
train Loss: 0.1207 Acc: 0.9667
val Loss: 0.8602 Acc: 0.7941

Epoch 96/199
----------
train Loss: 0.1185 Acc: 0.9657
val Loss: 0.9602 Acc: 0.7902

Epoch 97/199
----------
train Loss: 0.1437 Acc: 0.9627
val Loss: 0.8959 Acc: 0.8029

Epoch 98/199
----------
train Loss: 0.1163 Acc: 0.9667
val Loss: 0.8369 Acc: 0.8098

Epoch 99/199
----------
train Loss: 0.1405 Acc: 0.9588
val Loss: 0.8714 Acc: 0.7971

Epoch 100/199
----------
train Loss: 0.1176 Acc: 0.9696
val Loss: 0.8425 Acc: 0.7941

Epoch 101/199
----------
train Loss: 0.1223 Acc: 0.9647
val Loss: 0.8530 Acc: 0.7882

Epoch 102/199
----------
train Loss: 0.1247 Acc: 0.9578
val Loss: 0.8856 Acc: 0.7990

Epoch 103/199
----------
```

```
train Loss: 0.1159 Acc: 0.9637
val Loss: 0.9700 Acc: 0.7804

Epoch 104/199
----------
train Loss: 0.1173 Acc: 0.9716
val Loss: 0.9461 Acc: 0.7725

Epoch 105/199
----------
train Loss: 0.1146 Acc: 0.9686
val Loss: 0.9951 Acc: 0.7706

Epoch 106/199
----------
train Loss: 0.1307 Acc: 0.9657
val Loss: 0.8993 Acc: 0.7775

Epoch 107/199
----------
train Loss: 0.1285 Acc: 0.9676
val Loss: 0.9888 Acc: 0.7853

Epoch 108/199
----------
train Loss: 0.1062 Acc: 0.9686
val Loss: 0.8805 Acc: 0.7971

Epoch 109/199
----------
train Loss: 0.0814 Acc: 0.9735
val Loss: 0.8543 Acc: 0.7971

Epoch 110/199
----------
train Loss: 0.0934 Acc: 0.9735
val Loss: 0.8965 Acc: 0.7971

Epoch 111/199
----------
train Loss: 0.1148 Acc: 0.9676
val Loss: 0.9464 Acc: 0.7873

Epoch 112/199
----------
train Loss: 0.1087 Acc: 0.9725
val Loss: 0.9104 Acc: 0.7755
```

```
Epoch 113/199
----------
train Loss: 0.0691 Acc: 0.9814
val Loss: 0.9380 Acc: 0.7833

Epoch 114/199
----------
train Loss: 0.0837 Acc: 0.9775
val Loss: 0.8992 Acc: 0.7951

Epoch 115/199
----------
train Loss: 0.0623 Acc: 0.9804
val Loss: 0.9035 Acc: 0.7931

Epoch 116/199
----------
train Loss: 0.0843 Acc: 0.9833
val Loss: 0.8910 Acc: 0.7931

Epoch 117/199
----------
train Loss: 0.1011 Acc: 0.9696
val Loss: 0.9090 Acc: 0.8000

Epoch 118/199
----------
train Loss: 0.1111 Acc: 0.9676
val Loss: 0.9687 Acc: 0.7804

Epoch 119/199
----------
train Loss: 0.1038 Acc: 0.9735
val Loss: 0.9768 Acc: 0.7804

Epoch 120/199
----------
train Loss: 0.0918 Acc: 0.9745
val Loss: 0.9067 Acc: 0.7853

Epoch 121/199
----------
train Loss: 0.0999 Acc: 0.9755
val Loss: 0.9458 Acc: 0.7667

Epoch 122/199
----------
train Loss: 0.1120 Acc: 0.9686
```

```
val Loss: 0.9427 Acc: 0.7814

Epoch 123/199
----------
train Loss: 0.0942 Acc: 0.9716
val Loss: 0.9125 Acc: 0.7814

Epoch 124/199
----------
train Loss: 0.1169 Acc: 0.9735
val Loss: 0.9099 Acc: 0.7804

Epoch 125/199
----------
train Loss: 0.0810 Acc: 0.9784
val Loss: 0.9144 Acc: 0.7990

Epoch 126/199
----------
train Loss: 0.0933 Acc: 0.9745
val Loss: 0.8732 Acc: 0.8010

Epoch 127/199
----------
train Loss: 0.0840 Acc: 0.9735
val Loss: 0.9325 Acc: 0.7843

Epoch 128/199
----------
train Loss: 0.0803 Acc: 0.9843
val Loss: 0.8639 Acc: 0.8049

Epoch 129/199
----------
train Loss: 0.0737 Acc: 0.9824
val Loss: 0.8860 Acc: 0.8098

Epoch 130/199
----------
train Loss: 0.0912 Acc: 0.9824
val Loss: 0.8698 Acc: 0.8059

Epoch 131/199
----------
train Loss: 0.0887 Acc: 0.9765
val Loss: 0.9184 Acc: 0.7882

Epoch 132/199
```

```
----------
train Loss: 0.1109 Acc: 0.9725
val Loss: 0.9169 Acc: 0.7843

Epoch 133/199
----------
train Loss: 0.1217 Acc: 0.9647
val Loss: 0.9421 Acc: 0.7873

Epoch 134/199
----------
train Loss: 0.1191 Acc: 0.9667
val Loss: 0.8454 Acc: 0.8059

Epoch 135/199
----------
train Loss: 0.0851 Acc: 0.9735
val Loss: 0.9001 Acc: 0.8000

Epoch 136/199
----------
train Loss: 0.0971 Acc: 0.9716
val Loss: 0.9537 Acc: 0.7765

Epoch 137/199
----------
train Loss: 0.0968 Acc: 0.9745
val Loss: 0.9460 Acc: 0.7843

Epoch 138/199
----------
train Loss: 0.0873 Acc: 0.9794
val Loss: 0.9132 Acc: 0.7980

Epoch 139/199
----------
train Loss: 0.0892 Acc: 0.9804
val Loss: 0.9264 Acc: 0.8020

Epoch 140/199
----------
train Loss: 0.0942 Acc: 0.9755
val Loss: 0.9117 Acc: 0.8010

Epoch 141/199
----------
train Loss: 0.0804 Acc: 0.9833
val Loss: 0.9318 Acc: 0.8039
```

```
Epoch 142/199
----------
train Loss: 0.0962 Acc: 0.9735
val Loss: 0.9076 Acc: 0.7990

Epoch 143/199
----------
train Loss: 0.0709 Acc: 0.9775
val Loss: 0.8777 Acc: 0.8059

Epoch 144/199
----------
train Loss: 0.0949 Acc: 0.9775
val Loss: 0.8631 Acc: 0.7941

Epoch 145/199
----------
train Loss: 0.0719 Acc: 0.9833
val Loss: 0.8163 Acc: 0.8078

Epoch 146/199
----------
train Loss: 0.0855 Acc: 0.9784
val Loss: 0.8598 Acc: 0.8069

Epoch 147/199
----------
train Loss: 0.1010 Acc: 0.9725
val Loss: 0.8324 Acc: 0.8078

Epoch 148/199
----------
train Loss: 0.1040 Acc: 0.9706
val Loss: 0.8894 Acc: 0.7931

Epoch 149/199
----------
train Loss: 0.0806 Acc: 0.9784
val Loss: 0.8633 Acc: 0.8010

Epoch 150/199
----------
train Loss: 0.0669 Acc: 0.9804
val Loss: 0.8433 Acc: 0.8020

Epoch 151/199
----------
```

```
train Loss: 0.0660 Acc: 0.9843
val Loss: 0.8534 Acc: 0.8049

Epoch 152/199
----------
train Loss: 0.0728 Acc: 0.9794
val Loss: 0.8677 Acc: 0.8000

Epoch 153/199
----------
train Loss: 0.0726 Acc: 0.9814
val Loss: 0.8342 Acc: 0.8078

Epoch 154/199
----------
train Loss: 0.1060 Acc: 0.9696
val Loss: 0.8034 Acc: 0.8078

Epoch 155/199
----------
train Loss: 0.0684 Acc: 0.9794
val Loss: 0.8457 Acc: 0.8088

Epoch 156/199
----------
train Loss: 0.0720 Acc: 0.9794
val Loss: 0.8542 Acc: 0.8010

Epoch 157/199
----------
train Loss: 0.0586 Acc: 0.9853
val Loss: 0.8311 Acc: 0.8049

Epoch 158/199
----------
train Loss: 0.0613 Acc: 0.9814
val Loss: 0.8499 Acc: 0.8059

Epoch 159/199
----------
train Loss: 0.0788 Acc: 0.9775
val Loss: 0.8651 Acc: 0.8069

Epoch 160/199
----------
train Loss: 0.0838 Acc: 0.9784
val Loss: 0.9157 Acc: 0.7971
```

```
Epoch 161/199
----------
train Loss: 0.0779 Acc: 0.9814
val Loss: 0.8370 Acc: 0.8088

Epoch 162/199
----------
train Loss: 0.0555 Acc: 0.9843
val Loss: 0.7870 Acc: 0.8127

Epoch 163/199
----------
train Loss: 0.0645 Acc: 0.9824
val Loss: 0.8713 Acc: 0.7971

Epoch 164/199
----------
train Loss: 0.1015 Acc: 0.9696
val Loss: 0.9098 Acc: 0.7990

Epoch 165/199
----------
train Loss: 0.0911 Acc: 0.9735
val Loss: 0.9173 Acc: 0.8000

Epoch 166/199
----------
train Loss: 0.0935 Acc: 0.9716
val Loss: 0.8590 Acc: 0.8078

Epoch 167/199
----------
train Loss: 0.0846 Acc: 0.9794
val Loss: 0.9291 Acc: 0.7912

Epoch 168/199
----------
train Loss: 0.0858 Acc: 0.9735
val Loss: 0.9659 Acc: 0.7882

Epoch 169/199
----------
train Loss: 0.0930 Acc: 0.9725
val Loss: 1.0006 Acc: 0.7725

Epoch 170/199
----------
train Loss: 0.1077 Acc: 0.9696
```

```
val Loss: 0.9939 Acc: 0.7725

Epoch 171/199
----------
train Loss: 0.0974 Acc: 0.9725
val Loss: 0.9288 Acc: 0.7971

Epoch 172/199
----------
train Loss: 0.0871 Acc: 0.9725
val Loss: 0.8784 Acc: 0.7980

Epoch 173/199
----------
train Loss: 0.0674 Acc: 0.9843
val Loss: 0.9035 Acc: 0.7951

Epoch 174/199
----------
train Loss: 0.1042 Acc: 0.9765
val Loss: 0.9110 Acc: 0.7882

Epoch 175/199
----------
train Loss: 0.0608 Acc: 0.9853
val Loss: 0.8832 Acc: 0.7951

Epoch 176/199
----------
train Loss: 0.0963 Acc: 0.9716
val Loss: 0.9273 Acc: 0.7843

Epoch 177/199
----------
train Loss: 0.0933 Acc: 0.9735
val Loss: 0.9509 Acc: 0.7853

Epoch 178/199
----------
train Loss: 0.0799 Acc: 0.9814
val Loss: 0.9393 Acc: 0.7843

Epoch 179/199
----------
train Loss: 0.0792 Acc: 0.9765
val Loss: 0.9259 Acc: 0.7843

Epoch 180/199
```

```
----------
train Loss: 0.0967 Acc: 0.9745
val Loss: 0.8909 Acc: 0.7980

Epoch 181/199
----------
train Loss: 0.1013 Acc: 0.9716
val Loss: 0.9797 Acc: 0.7735

Epoch 182/199
----------
train Loss: 0.0998 Acc: 0.9745
val Loss: 0.8821 Acc: 0.7892

Epoch 183/199
----------
train Loss: 0.0682 Acc: 0.9784
val Loss: 0.9600 Acc: 0.7873

Epoch 184/199
----------
train Loss: 0.0727 Acc: 0.9755
val Loss: 0.9821 Acc: 0.7755

Epoch 185/199
----------
train Loss: 0.0656 Acc: 0.9794
val Loss: 0.9541 Acc: 0.7755

Epoch 186/199
----------
train Loss: 0.0742 Acc: 0.9794
val Loss: 0.9153 Acc: 0.7882

Epoch 187/199
----------
train Loss: 0.0888 Acc: 0.9765
val Loss: 0.9702 Acc: 0.7804

Epoch 188/199
----------
train Loss: 0.0708 Acc: 0.9833
val Loss: 0.9473 Acc: 0.7833

Epoch 189/199
----------
train Loss: 0.0873 Acc: 0.9755
val Loss: 0.9878 Acc: 0.7882
```

```
Epoch 190/199
----------
train Loss: 0.0674 Acc: 0.9784
val Loss: 0.9714 Acc: 0.7784

Epoch 191/199
----------
train Loss: 0.0975 Acc: 0.9775
val Loss: 0.9702 Acc: 0.7922

Epoch 192/199
----------
train Loss: 0.0668 Acc: 0.9794
val Loss: 0.9936 Acc: 0.7961

Epoch 193/199
----------
train Loss: 0.0732 Acc: 0.9804
val Loss: 0.9708 Acc: 0.7882

Epoch 194/199
----------
train Loss: 0.0870 Acc: 0.9755
val Loss: 0.9667 Acc: 0.7990

Epoch 195/199
----------
train Loss: 0.0665 Acc: 0.9784
val Loss: 0.9840 Acc: 0.7931

Epoch 196/199
----------
train Loss: 0.0825 Acc: 0.9794
val Loss: 0.9631 Acc: 0.7971

Epoch 197/199
----------
train Loss: 0.0612 Acc: 0.9853
val Loss: 0.9282 Acc: 0.7931

Epoch 198/199
----------
train Loss: 0.0708 Acc: 0.9784
val Loss: 0.9146 Acc: 0.7912

Epoch 199/199
----------
```

```
train Loss: 0.0899 Acc: 0.9775
val Loss: 0.9275 Acc: 0.7931

Training complete in 26m 32s
Best val Acc: 0.812745
```

The results for the three learning rate approaches are below in the table.

Learning Rate Setting	Best Validation Accuracy
0.01 with decay	0.803
0.1 fixed	0.520
0.01 fixed	0.813

The fixed learning rate of 0.01 for all 200 epochs gave the best validation accuracy for the three experiments.

Q2

- a: (8 points)

Treating `ResNet50` as a feature extractor and only training the final layer with various learning rates.

[11]:
```
LR_SET = [10**i for i in range(-3, 1)]

for lr in LR_SET:
    print(f'\n----- Performing experiment with fixed lr={lr} -----\n')

    model_conv = torchvision.models.resnet50(pretrained=True)
    for param in model_conv.parameters():
        param.requires_grad = False

    # Parameters of newly constructed modules have requires_grad=True by default
    num_ftrs = model_conv.fc.in_features
    model_conv.fc = nn.Linear(num_ftrs, len(class_names))

    model_conv = model_conv.to(device)

    criterion = nn.CrossEntropyLoss()

    # Observe that only parameters of final layer are being optimized as
    # opposed to before.
    optimizer_conv = optim.SGD(model_conv.fc.parameters(), lr=0.001, momentum=0.
↪9)

    # Decay LR by a factor of 0.1 every 60 epochs
    exp_lr_scheduler = lr_scheduler.StepLR(optimizer_conv, step_size=60,␣
↪gamma=0.1)
```

```python
# Train model
model_conv = train_model(model_conv, criterion, optimizer_conv,
                         exp_lr_scheduler, num_epochs=200)
```

```
----- Performing experiment with fixed lr=0.001 -----

Epoch 0/199
----------
train Loss: 4.6785 Acc: 0.0147
val Loss: 4.6396 Acc: 0.0127

Epoch 1/199
----------
train Loss: 4.6253 Acc: 0.0127
val Loss: 4.5816 Acc: 0.0255

Epoch 2/199
----------
train Loss: 4.5631 Acc: 0.0284
val Loss: 4.5245 Acc: 0.0441

Epoch 3/199
----------
train Loss: 4.5190 Acc: 0.0412
val Loss: 4.4712 Acc: 0.0745

Epoch 4/199
----------
train Loss: 4.4662 Acc: 0.0755
val Loss: 4.4178 Acc: 0.1069

Epoch 5/199
----------
train Loss: 4.4225 Acc: 0.0912
val Loss: 4.3649 Acc: 0.1441

Epoch 6/199
----------
train Loss: 4.3570 Acc: 0.1490
val Loss: 4.3146 Acc: 0.1755

Epoch 7/199
----------
train Loss: 4.3206 Acc: 0.1961
val Loss: 4.2642 Acc: 0.2059
```

```
Epoch 8/199
----------
train Loss: 4.2621 Acc: 0.2343
val Loss: 4.2143 Acc: 0.2461

Epoch 9/199
----------
train Loss: 4.2095 Acc: 0.2392
val Loss: 4.1662 Acc: 0.2735

Epoch 10/199
----------
train Loss: 4.1708 Acc: 0.2853
val Loss: 4.1173 Acc: 0.3029

Epoch 11/199
----------
train Loss: 4.1370 Acc: 0.3000
val Loss: 4.0754 Acc: 0.3216

Epoch 12/199
----------
train Loss: 4.0773 Acc: 0.3422
val Loss: 4.0217 Acc: 0.3431

Epoch 13/199
----------
train Loss: 4.0473 Acc: 0.3647
val Loss: 3.9758 Acc: 0.3549

Epoch 14/199
----------
train Loss: 3.9868 Acc: 0.3853
val Loss: 3.9305 Acc: 0.3657

Epoch 15/199
----------
train Loss: 3.9625 Acc: 0.3804
val Loss: 3.8916 Acc: 0.3892

Epoch 16/199
----------
train Loss: 3.9057 Acc: 0.3971
val Loss: 3.8446 Acc: 0.4196

Epoch 17/199
----------
```

```
train Loss: 3.8594 Acc: 0.4529
val Loss: 3.8022 Acc: 0.4147

Epoch 18/199
----------
train Loss: 3.8225 Acc: 0.4382
val Loss: 3.7600 Acc: 0.4353

Epoch 19/199
----------
train Loss: 3.7966 Acc: 0.4529
val Loss: 3.7217 Acc: 0.4559

Epoch 20/199
----------
train Loss: 3.7670 Acc: 0.4382
val Loss: 3.6766 Acc: 0.4549

Epoch 21/199
----------
train Loss: 3.7112 Acc: 0.4716
val Loss: 3.6358 Acc: 0.4696

Epoch 22/199
----------
train Loss: 3.6610 Acc: 0.4725
val Loss: 3.5992 Acc: 0.4608

Epoch 23/199
----------
train Loss: 3.6250 Acc: 0.4892
val Loss: 3.5590 Acc: 0.4686

Epoch 24/199
----------
train Loss: 3.5829 Acc: 0.4961
val Loss: 3.5194 Acc: 0.4931

Epoch 25/199
----------
train Loss: 3.5724 Acc: 0.4873
val Loss: 3.4879 Acc: 0.4951

Epoch 26/199
----------
train Loss: 3.5307 Acc: 0.5088
val Loss: 3.4473 Acc: 0.5010
```

```
Epoch 27/199
----------
train Loss: 3.4682 Acc: 0.5186
val Loss: 3.4181 Acc: 0.5049

Epoch 28/199
----------
train Loss: 3.4580 Acc: 0.5186
val Loss: 3.3808 Acc: 0.4951

Epoch 29/199
----------
train Loss: 3.3998 Acc: 0.5412
val Loss: 3.3397 Acc: 0.5127

Epoch 30/199
----------
train Loss: 3.3858 Acc: 0.5343
val Loss: 3.3068 Acc: 0.5049

Epoch 31/199
----------
train Loss: 3.3589 Acc: 0.5265
val Loss: 3.2781 Acc: 0.5108

Epoch 32/199
----------
train Loss: 3.3057 Acc: 0.5510
val Loss: 3.2438 Acc: 0.5235

Epoch 33/199
----------
train Loss: 3.3018 Acc: 0.5422
val Loss: 3.2249 Acc: 0.5225

Epoch 34/199
----------
train Loss: 3.2481 Acc: 0.5431
val Loss: 3.1865 Acc: 0.5225

Epoch 35/199
----------
train Loss: 3.2440 Acc: 0.5373
val Loss: 3.1503 Acc: 0.5402

Epoch 36/199
----------
train Loss: 3.1851 Acc: 0.5431
```

```
val Loss: 3.1142 Acc: 0.5363

Epoch 37/199
----------
train Loss: 3.1600 Acc: 0.5676
val Loss: 3.0943 Acc: 0.5382

Epoch 38/199
----------
train Loss: 3.0908 Acc: 0.6000
val Loss: 3.0607 Acc: 0.5441

Epoch 39/199
----------
train Loss: 3.1136 Acc: 0.5578
val Loss: 3.0366 Acc: 0.5539

Epoch 40/199
----------
train Loss: 3.0684 Acc: 0.5961
val Loss: 3.0099 Acc: 0.5461

Epoch 41/199
----------
train Loss: 3.0535 Acc: 0.5627
val Loss: 2.9838 Acc: 0.5588

Epoch 42/199
----------
train Loss: 2.9852 Acc: 0.6000
val Loss: 2.9471 Acc: 0.5529

Epoch 43/199
----------
train Loss: 3.0238 Acc: 0.5706
val Loss: 2.9315 Acc: 0.5657

Epoch 44/199
----------
train Loss: 2.9405 Acc: 0.6245
val Loss: 2.9010 Acc: 0.5627

Epoch 45/199
----------
train Loss: 2.9553 Acc: 0.5863
val Loss: 2.8781 Acc: 0.5637

Epoch 46/199
```

```
----------
train Loss: 2.9092 Acc: 0.5843
val Loss: 2.8521 Acc: 0.5765

Epoch 47/199
----------
train Loss: 2.8975 Acc: 0.5853
val Loss: 2.8434 Acc: 0.5716

Epoch 48/199
----------
train Loss: 2.8776 Acc: 0.5971
val Loss: 2.8084 Acc: 0.5637

Epoch 49/199
----------
train Loss: 2.8474 Acc: 0.6029
val Loss: 2.7833 Acc: 0.5657

Epoch 50/199
----------
train Loss: 2.8232 Acc: 0.5951
val Loss: 2.7598 Acc: 0.5814

Epoch 51/199
----------
train Loss: 2.8043 Acc: 0.6127
val Loss: 2.7509 Acc: 0.5745

Epoch 52/199
----------
train Loss: 2.7742 Acc: 0.6049
val Loss: 2.7249 Acc: 0.5784

Epoch 53/199
----------
train Loss: 2.7621 Acc: 0.6137
val Loss: 2.7020 Acc: 0.5853

Epoch 54/199
----------
train Loss: 2.7341 Acc: 0.6167
val Loss: 2.6768 Acc: 0.5814

Epoch 55/199
----------
train Loss: 2.7224 Acc: 0.6069
val Loss: 2.6626 Acc: 0.5824
```

```
Epoch 56/199
----------
train Loss: 2.6762 Acc: 0.6186
val Loss: 2.6368 Acc: 0.5931

Epoch 57/199
----------
train Loss: 2.6908 Acc: 0.6225
val Loss: 2.6164 Acc: 0.5873

Epoch 58/199
----------
train Loss: 2.6495 Acc: 0.6471
val Loss: 2.5969 Acc: 0.5804

Epoch 59/199
----------
train Loss: 2.6298 Acc: 0.6235
val Loss: 2.5721 Acc: 0.5922

Epoch 60/199
----------
train Loss: 2.5988 Acc: 0.6245
val Loss: 2.5702 Acc: 0.5912

Epoch 61/199
----------
train Loss: 2.6115 Acc: 0.6451
val Loss: 2.5739 Acc: 0.5902

Epoch 62/199
----------
train Loss: 2.6147 Acc: 0.6392
val Loss: 2.5774 Acc: 0.5882

Epoch 63/199
----------
train Loss: 2.5960 Acc: 0.6167
val Loss: 2.5658 Acc: 0.5873

Epoch 64/199
----------
train Loss: 2.6088 Acc: 0.6431
val Loss: 2.5670 Acc: 0.5931

Epoch 65/199
----------
```

```
train Loss: 2.5992 Acc: 0.6324
val Loss: 2.5660 Acc: 0.5912

Epoch 66/199
----------
train Loss: 2.6220 Acc: 0.6137
val Loss: 2.5690 Acc: 0.5922

Epoch 67/199
----------
train Loss: 2.6198 Acc: 0.6196
val Loss: 2.5676 Acc: 0.5902

Epoch 68/199
----------
train Loss: 2.6002 Acc: 0.6196
val Loss: 2.5614 Acc: 0.5912

Epoch 69/199
----------
train Loss: 2.6124 Acc: 0.6225
val Loss: 2.5680 Acc: 0.5902

Epoch 70/199
----------
train Loss: 2.5829 Acc: 0.6471
val Loss: 2.5586 Acc: 0.5990

Epoch 71/199
----------
train Loss: 2.5804 Acc: 0.6373
val Loss: 2.5536 Acc: 0.5990

Epoch 72/199
----------
train Loss: 2.5976 Acc: 0.6235
val Loss: 2.5523 Acc: 0.5961

Epoch 73/199
----------
train Loss: 2.6177 Acc: 0.6167
val Loss: 2.5554 Acc: 0.5902

Epoch 74/199
----------
train Loss: 2.6381 Acc: 0.6118
val Loss: 2.5522 Acc: 0.5912
```

```
Epoch 75/199
----------
train Loss: 2.5715 Acc: 0.6392
val Loss: 2.5422 Acc: 0.5980

Epoch 76/199
----------
train Loss: 2.5930 Acc: 0.6196
val Loss: 2.5500 Acc: 0.5912

Epoch 77/199
----------
train Loss: 2.5747 Acc: 0.6373
val Loss: 2.5468 Acc: 0.5980

Epoch 78/199
----------
train Loss: 2.5767 Acc: 0.6147
val Loss: 2.5483 Acc: 0.5951

Epoch 79/199
----------
train Loss: 2.6146 Acc: 0.6167
val Loss: 2.5409 Acc: 0.5980

Epoch 80/199
----------
train Loss: 2.5857 Acc: 0.6363
val Loss: 2.5385 Acc: 0.5971

Epoch 81/199
----------
train Loss: 2.5666 Acc: 0.6245
val Loss: 2.5357 Acc: 0.5980

Epoch 82/199
----------
train Loss: 2.5822 Acc: 0.6363
val Loss: 2.5367 Acc: 0.6020

Epoch 83/199
----------
train Loss: 2.5768 Acc: 0.6186
val Loss: 2.5358 Acc: 0.5961

Epoch 84/199
----------
train Loss: 2.5833 Acc: 0.6500
```

```
val Loss: 2.5353 Acc: 0.5980

Epoch 85/199
----------
train Loss: 2.5749 Acc: 0.6343
val Loss: 2.5316 Acc: 0.5931

Epoch 86/199
----------
train Loss: 2.5821 Acc: 0.6245
val Loss: 2.5341 Acc: 0.5971

Epoch 87/199
----------
train Loss: 2.5655 Acc: 0.6275
val Loss: 2.5225 Acc: 0.6029

Epoch 88/199
----------
train Loss: 2.5446 Acc: 0.6471
val Loss: 2.5213 Acc: 0.6020

Epoch 89/199
----------
train Loss: 2.5689 Acc: 0.6441
val Loss: 2.5243 Acc: 0.5971

Epoch 90/199
----------
train Loss: 2.5374 Acc: 0.6353
val Loss: 2.5135 Acc: 0.5980

Epoch 91/199
----------
train Loss: 2.5171 Acc: 0.6608
val Loss: 2.5222 Acc: 0.6000

Epoch 92/199
----------
train Loss: 2.5368 Acc: 0.6373
val Loss: 2.5179 Acc: 0.5990

Epoch 93/199
----------
train Loss: 2.5337 Acc: 0.6461
val Loss: 2.5105 Acc: 0.6020

Epoch 94/199
```

```
----------
train Loss: 2.5646 Acc: 0.6314
val Loss: 2.5120 Acc: 0.6069

Epoch 95/199
----------
train Loss: 2.5858 Acc: 0.6314
val Loss: 2.5138 Acc: 0.5980

Epoch 96/199
----------
train Loss: 2.5153 Acc: 0.6520
val Loss: 2.5112 Acc: 0.6049

Epoch 97/199
----------
train Loss: 2.5257 Acc: 0.6353
val Loss: 2.5052 Acc: 0.6059

Epoch 98/199
----------
train Loss: 2.5102 Acc: 0.6608
val Loss: 2.5055 Acc: 0.6059

Epoch 99/199
----------
train Loss: 2.5395 Acc: 0.6314
val Loss: 2.5055 Acc: 0.5980

Epoch 100/199
----------
train Loss: 2.5571 Acc: 0.6402
val Loss: 2.5055 Acc: 0.6049

Epoch 101/199
----------
train Loss: 2.5433 Acc: 0.6314
val Loss: 2.5032 Acc: 0.6049

Epoch 102/199
----------
train Loss: 2.5581 Acc: 0.6324
val Loss: 2.5045 Acc: 0.6010

Epoch 103/199
----------
train Loss: 2.5425 Acc: 0.6392
val Loss: 2.5036 Acc: 0.6059
```

```
Epoch 104/199
----------
train Loss: 2.5300 Acc: 0.6461
val Loss: 2.4945 Acc: 0.6049

Epoch 105/199
----------
train Loss: 2.5357 Acc: 0.6216
val Loss: 2.5061 Acc: 0.6010

Epoch 106/199
----------
train Loss: 2.5239 Acc: 0.6451
val Loss: 2.4923 Acc: 0.6059

Epoch 107/199
----------
train Loss: 2.5006 Acc: 0.6608
val Loss: 2.4993 Acc: 0.6029

Epoch 108/199
----------
train Loss: 2.5097 Acc: 0.6422
val Loss: 2.4866 Acc: 0.6039

Epoch 109/199
----------
train Loss: 2.5278 Acc: 0.6500
val Loss: 2.4917 Acc: 0.6069

Epoch 110/199
----------
train Loss: 2.5264 Acc: 0.6412
val Loss: 2.4840 Acc: 0.6069

Epoch 111/199
----------
train Loss: 2.5223 Acc: 0.6549
val Loss: 2.4780 Acc: 0.6029

Epoch 112/199
----------
train Loss: 2.5125 Acc: 0.6373
val Loss: 2.4846 Acc: 0.5990

Epoch 113/199
----------
```

```
train Loss: 2.6108 Acc: 0.6059
val Loss: 2.4908 Acc: 0.5990

Epoch 114/199
----------
train Loss: 2.5036 Acc: 0.6353
val Loss: 2.4863 Acc: 0.6088

Epoch 115/199
----------
train Loss: 2.5662 Acc: 0.6078
val Loss: 2.4863 Acc: 0.6039

Epoch 116/199
----------
train Loss: 2.4893 Acc: 0.6559
val Loss: 2.4732 Acc: 0.6059

Epoch 117/199
----------
train Loss: 2.5219 Acc: 0.6471
val Loss: 2.4752 Acc: 0.6108

Epoch 118/199
----------
train Loss: 2.5121 Acc: 0.6275
val Loss: 2.4753 Acc: 0.6098

Epoch 119/199
----------
train Loss: 2.5205 Acc: 0.6441
val Loss: 2.4758 Acc: 0.6059

Epoch 120/199
----------
train Loss: 2.5001 Acc: 0.6225
val Loss: 2.4765 Acc: 0.6088

Epoch 121/199
----------
train Loss: 2.5129 Acc: 0.6245
val Loss: 2.4690 Acc: 0.6088

Epoch 122/199
----------
train Loss: 2.5009 Acc: 0.6441
val Loss: 2.4696 Acc: 0.6029
```

```
Epoch 123/199
----------
train Loss: 2.5148 Acc: 0.6392
val Loss: 2.4802 Acc: 0.5990

Epoch 124/199
----------
train Loss: 2.4891 Acc: 0.6500
val Loss: 2.4765 Acc: 0.6049

Epoch 125/199
----------
train Loss: 2.4829 Acc: 0.6559
val Loss: 2.4794 Acc: 0.6059

Epoch 126/199
----------
train Loss: 2.5295 Acc: 0.6245
val Loss: 2.4731 Acc: 0.6039

Epoch 127/199
----------
train Loss: 2.4569 Acc: 0.6735
val Loss: 2.4629 Acc: 0.6137

Epoch 128/199
----------
train Loss: 2.4588 Acc: 0.6422
val Loss: 2.4609 Acc: 0.6088

Epoch 129/199
----------
train Loss: 2.5348 Acc: 0.6412
val Loss: 2.4735 Acc: 0.6059

Epoch 130/199
----------
train Loss: 2.4560 Acc: 0.6676
val Loss: 2.4612 Acc: 0.6137

Epoch 131/199
----------
train Loss: 2.5180 Acc: 0.6284
val Loss: 2.4723 Acc: 0.6049

Epoch 132/199
----------
train Loss: 2.5202 Acc: 0.6353
```

```
val Loss: 2.4776 Acc: 0.6088

Epoch 133/199
----------
train Loss: 2.4624 Acc: 0.6598
val Loss: 2.4680 Acc: 0.6118

Epoch 134/199
----------
train Loss: 2.4851 Acc: 0.6627
val Loss: 2.4699 Acc: 0.6088

Epoch 135/199
----------
train Loss: 2.4820 Acc: 0.6480
val Loss: 2.4748 Acc: 0.6049

Epoch 136/199
----------
train Loss: 2.5066 Acc: 0.6569
val Loss: 2.4819 Acc: 0.6059

Epoch 137/199
----------
train Loss: 2.5121 Acc: 0.6353
val Loss: 2.4750 Acc: 0.6078

Epoch 138/199
----------
train Loss: 2.4620 Acc: 0.6598
val Loss: 2.4719 Acc: 0.6098

Epoch 139/199
----------
train Loss: 2.4814 Acc: 0.6647
val Loss: 2.4726 Acc: 0.6069

Epoch 140/199
----------
train Loss: 2.4939 Acc: 0.6559
val Loss: 2.4702 Acc: 0.6049

Epoch 141/199
----------
train Loss: 2.5012 Acc: 0.6363
val Loss: 2.4643 Acc: 0.6137

Epoch 142/199
```

```
----------
train Loss: 2.4747 Acc: 0.6412
val Loss: 2.4706 Acc: 0.6088

Epoch 143/199
----------
train Loss: 2.4927 Acc: 0.6520
val Loss: 2.4613 Acc: 0.6127

Epoch 144/199
----------
train Loss: 2.4830 Acc: 0.6471
val Loss: 2.4667 Acc: 0.6049

Epoch 145/199
----------
train Loss: 2.4738 Acc: 0.6745
val Loss: 2.4578 Acc: 0.6088

Epoch 146/199
----------
train Loss: 2.4938 Acc: 0.6647
val Loss: 2.4714 Acc: 0.6118

Epoch 147/199
----------
train Loss: 2.4787 Acc: 0.6637
val Loss: 2.4652 Acc: 0.6069

Epoch 148/199
----------
train Loss: 2.5016 Acc: 0.6363
val Loss: 2.4711 Acc: 0.6088

Epoch 149/199
----------
train Loss: 2.4993 Acc: 0.6461
val Loss: 2.4599 Acc: 0.6118

Epoch 150/199
----------
train Loss: 2.4774 Acc: 0.6422
val Loss: 2.4681 Acc: 0.6059

Epoch 151/199
----------
train Loss: 2.5320 Acc: 0.6235
val Loss: 2.4763 Acc: 0.6039
```

```
Epoch 152/199
----------
train Loss: 2.4967 Acc: 0.6392
val Loss: 2.4631 Acc: 0.6108

Epoch 153/199
----------
train Loss: 2.4981 Acc: 0.6363
val Loss: 2.4624 Acc: 0.6088

Epoch 154/199
----------
train Loss: 2.4938 Acc: 0.6451
val Loss: 2.4701 Acc: 0.6039

Epoch 155/199
----------
train Loss: 2.4969 Acc: 0.6431
val Loss: 2.4681 Acc: 0.6029

Epoch 156/199
----------
train Loss: 2.4641 Acc: 0.6608
val Loss: 2.4593 Acc: 0.6088

Epoch 157/199
----------
train Loss: 2.5052 Acc: 0.6686
val Loss: 2.4723 Acc: 0.6010

Epoch 158/199
----------
train Loss: 2.4991 Acc: 0.6363
val Loss: 2.4635 Acc: 0.6039

Epoch 159/199
----------
train Loss: 2.4834 Acc: 0.6559
val Loss: 2.4642 Acc: 0.6137

Epoch 160/199
----------
train Loss: 2.4939 Acc: 0.6392
val Loss: 2.4648 Acc: 0.6088

Epoch 161/199
----------
```

```
train Loss: 2.4670 Acc: 0.6627
val Loss: 2.4667 Acc: 0.6108

Epoch 162/199
----------
train Loss: 2.4582 Acc: 0.6402
val Loss: 2.4698 Acc: 0.6078

Epoch 163/199
----------
train Loss: 2.4771 Acc: 0.6529
val Loss: 2.4698 Acc: 0.6049

Epoch 164/199
----------
train Loss: 2.4889 Acc: 0.6373
val Loss: 2.4695 Acc: 0.6078

Epoch 165/199
----------
train Loss: 2.5065 Acc: 0.6500
val Loss: 2.4692 Acc: 0.6108

Epoch 166/199
----------
train Loss: 2.4831 Acc: 0.6373
val Loss: 2.4650 Acc: 0.6078

Epoch 167/199
----------
train Loss: 2.5031 Acc: 0.6333
val Loss: 2.4734 Acc: 0.6078

Epoch 168/199
----------
train Loss: 2.4925 Acc: 0.6480
val Loss: 2.4678 Acc: 0.6098

Epoch 169/199
----------
train Loss: 2.4819 Acc: 0.6363
val Loss: 2.4583 Acc: 0.6098

Epoch 170/199
----------
train Loss: 2.4960 Acc: 0.6471
val Loss: 2.4673 Acc: 0.6098
```

```
Epoch 171/199
----------
train Loss: 2.5166 Acc: 0.6392
val Loss: 2.4589 Acc: 0.6127

Epoch 172/199
----------
train Loss: 2.4743 Acc: 0.6471
val Loss: 2.4646 Acc: 0.6127

Epoch 173/199
----------
train Loss: 2.4695 Acc: 0.6461
val Loss: 2.4567 Acc: 0.6088

Epoch 174/199
----------
train Loss: 2.4806 Acc: 0.6422
val Loss: 2.4600 Acc: 0.6098

Epoch 175/199
----------
train Loss: 2.4763 Acc: 0.6431
val Loss: 2.4545 Acc: 0.6137

Epoch 176/199
----------
train Loss: 2.4923 Acc: 0.6441
val Loss: 2.4596 Acc: 0.6078

Epoch 177/199
----------
train Loss: 2.4888 Acc: 0.6569
val Loss: 2.4613 Acc: 0.6069

Epoch 178/199
----------
train Loss: 2.5311 Acc: 0.6412
val Loss: 2.4635 Acc: 0.6010

Epoch 179/199
----------
train Loss: 2.5046 Acc: 0.6422
val Loss: 2.4554 Acc: 0.6078

Epoch 180/199
----------
train Loss: 2.4944 Acc: 0.6412
```

```
val Loss: 2.4550 Acc: 0.6127

Epoch 181/199
----------
train Loss: 2.4803 Acc: 0.6539
val Loss: 2.4610 Acc: 0.6059

Epoch 182/199
----------
train Loss: 2.4983 Acc: 0.6549
val Loss: 2.4652 Acc: 0.6039

Epoch 183/199
----------
train Loss: 2.5404 Acc: 0.6422
val Loss: 2.4694 Acc: 0.6020

Epoch 184/199
----------
train Loss: 2.5035 Acc: 0.6578
val Loss: 2.4579 Acc: 0.6020

Epoch 185/199
----------
train Loss: 2.4968 Acc: 0.6471
val Loss: 2.4666 Acc: 0.6069

Epoch 186/199
----------
train Loss: 2.5102 Acc: 0.6471
val Loss: 2.4723 Acc: 0.6029

Epoch 187/199
----------
train Loss: 2.5208 Acc: 0.6461
val Loss: 2.4722 Acc: 0.6069

Epoch 188/199
----------
train Loss: 2.5255 Acc: 0.6402
val Loss: 2.4720 Acc: 0.6069

Epoch 189/199
----------
train Loss: 2.5115 Acc: 0.6363
val Loss: 2.4673 Acc: 0.6118

Epoch 190/199
```

```
----------
train Loss: 2.4454 Acc: 0.6745
val Loss: 2.4611 Acc: 0.6127

Epoch 191/199
----------
train Loss: 2.4731 Acc: 0.6549
val Loss: 2.4660 Acc: 0.6029

Epoch 192/199
----------
train Loss: 2.4508 Acc: 0.6608
val Loss: 2.4606 Acc: 0.6108

Epoch 193/199
----------
train Loss: 2.4795 Acc: 0.6490
val Loss: 2.4631 Acc: 0.6108

Epoch 194/199
----------
train Loss: 2.4951 Acc: 0.6451
val Loss: 2.4675 Acc: 0.6098

Epoch 195/199
----------
train Loss: 2.4941 Acc: 0.6392
val Loss: 2.4593 Acc: 0.6088

Epoch 196/199
----------
train Loss: 2.4905 Acc: 0.6412
val Loss: 2.4543 Acc: 0.6147

Epoch 197/199
----------
train Loss: 2.4933 Acc: 0.6578
val Loss: 2.4572 Acc: 0.6069

Epoch 198/199
----------
train Loss: 2.5139 Acc: 0.6275
val Loss: 2.4689 Acc: 0.6069

Epoch 199/199
----------
train Loss: 2.5129 Acc: 0.6353
val Loss: 2.4605 Acc: 0.6020
```

```
Training complete in 15m 9s
Best val Acc: 0.614706

----- Performing experiment with fixed lr=0.01 -----

Epoch 0/199
----------
train Loss: 4.6726 Acc: 0.0088
val Loss: 4.6353 Acc: 0.0127

Epoch 1/199
----------
train Loss: 4.6122 Acc: 0.0127
val Loss: 4.5764 Acc: 0.0147

Epoch 2/199
----------
train Loss: 4.5601 Acc: 0.0225
val Loss: 4.5218 Acc: 0.0451

Epoch 3/199
----------
train Loss: 4.5078 Acc: 0.0461
val Loss: 4.4702 Acc: 0.0627

Epoch 4/199
----------
train Loss: 4.4573 Acc: 0.0667
val Loss: 4.4173 Acc: 0.1010

Epoch 5/199
----------
train Loss: 4.4130 Acc: 0.1020
val Loss: 4.3640 Acc: 0.1461

Epoch 6/199
----------
train Loss: 4.3567 Acc: 0.1373
val Loss: 4.3136 Acc: 0.1784

Epoch 7/199
----------
train Loss: 4.3205 Acc: 0.1735
val Loss: 4.2670 Acc: 0.2049

Epoch 8/199
----------
```

```
train Loss: 4.2540 Acc: 0.2294
val Loss: 4.2170 Acc: 0.2324

Epoch 9/199
----------
train Loss: 4.2261 Acc: 0.2412
val Loss: 4.1688 Acc: 0.2686

Epoch 10/199
----------
train Loss: 4.1687 Acc: 0.2951
val Loss: 4.1185 Acc: 0.2980

Epoch 11/199
----------
train Loss: 4.1278 Acc: 0.3255
val Loss: 4.0732 Acc: 0.3255

Epoch 12/199
----------
train Loss: 4.0882 Acc: 0.3304
val Loss: 4.0258 Acc: 0.3696

Epoch 13/199
----------
train Loss: 4.0430 Acc: 0.3549
val Loss: 3.9778 Acc: 0.3794

Epoch 14/199
----------
train Loss: 3.9808 Acc: 0.3745
val Loss: 3.9351 Acc: 0.3922

Epoch 15/199
----------
train Loss: 3.9535 Acc: 0.3892
val Loss: 3.8898 Acc: 0.4088

Epoch 16/199
----------
train Loss: 3.9042 Acc: 0.4353
val Loss: 3.8438 Acc: 0.4353

Epoch 17/199
----------
train Loss: 3.8674 Acc: 0.4167
val Loss: 3.8048 Acc: 0.4157
```

```
Epoch 18/199
----------
train Loss: 3.8186 Acc: 0.4324
val Loss: 3.7613 Acc: 0.4422

Epoch 19/199
----------
train Loss: 3.7751 Acc: 0.4578
val Loss: 3.7209 Acc: 0.4510

Epoch 20/199
----------
train Loss: 3.7415 Acc: 0.4618
val Loss: 3.6766 Acc: 0.4569

Epoch 21/199
----------
train Loss: 3.7153 Acc: 0.4480
val Loss: 3.6415 Acc: 0.4725

Epoch 22/199
----------
train Loss: 3.6591 Acc: 0.4716
val Loss: 3.5967 Acc: 0.4745

Epoch 23/199
----------
train Loss: 3.6209 Acc: 0.5000
val Loss: 3.5604 Acc: 0.4716

Epoch 24/199
----------
train Loss: 3.5950 Acc: 0.4843
val Loss: 3.5188 Acc: 0.4853

Epoch 25/199
----------
train Loss: 3.5683 Acc: 0.4765
val Loss: 3.4849 Acc: 0.4912

Epoch 26/199
----------
train Loss: 3.4946 Acc: 0.5118
val Loss: 3.4438 Acc: 0.4912

Epoch 27/199
----------
train Loss: 3.4726 Acc: 0.5275
```

```
val Loss: 3.4147 Acc: 0.5078

Epoch 28/199
----------
train Loss: 3.4343 Acc: 0.5284
val Loss: 3.3720 Acc: 0.5020

Epoch 29/199
----------
train Loss: 3.4258 Acc: 0.5255
val Loss: 3.3451 Acc: 0.5059

Epoch 30/199
----------
train Loss: 3.3929 Acc: 0.5069
val Loss: 3.3148 Acc: 0.5157

Epoch 31/199
----------
train Loss: 3.3377 Acc: 0.5275
val Loss: 3.2758 Acc: 0.5078

Epoch 32/199
----------
train Loss: 3.3292 Acc: 0.5255
val Loss: 3.2425 Acc: 0.5206

Epoch 33/199
----------
train Loss: 3.2892 Acc: 0.5304
val Loss: 3.2171 Acc: 0.5186

Epoch 34/199
----------
train Loss: 3.2619 Acc: 0.5382
val Loss: 3.1900 Acc: 0.5275

Epoch 35/199
----------
train Loss: 3.2244 Acc: 0.5529
val Loss: 3.1575 Acc: 0.5294

Epoch 36/199
----------
train Loss: 3.1814 Acc: 0.5578
val Loss: 3.1252 Acc: 0.5373

Epoch 37/199
```

```
----------
train Loss: 3.1784 Acc: 0.5471
val Loss: 3.0874 Acc: 0.5461

Epoch 38/199
----------
train Loss: 3.1453 Acc: 0.5451
val Loss: 3.0560 Acc: 0.5529

Epoch 39/199
----------
train Loss: 3.1065 Acc: 0.5725
val Loss: 3.0389 Acc: 0.5500

Epoch 40/199
----------
train Loss: 3.0903 Acc: 0.5539
val Loss: 3.0101 Acc: 0.5500

Epoch 41/199
----------
train Loss: 3.0351 Acc: 0.5843
val Loss: 2.9770 Acc: 0.5510

Epoch 42/199
----------
train Loss: 3.0385 Acc: 0.5637
val Loss: 2.9497 Acc: 0.5618

Epoch 43/199
----------
train Loss: 2.9951 Acc: 0.5922
val Loss: 2.9457 Acc: 0.5598

Epoch 44/199
----------
train Loss: 3.0032 Acc: 0.5833
val Loss: 2.9094 Acc: 0.5618

Epoch 45/199
----------
train Loss: 2.9435 Acc: 0.6029
val Loss: 2.8845 Acc: 0.5696

Epoch 46/199
----------
train Loss: 2.9065 Acc: 0.6029
val Loss: 2.8527 Acc: 0.5775
```

```
Epoch 47/199
----------
train Loss: 2.8872 Acc: 0.6167
val Loss: 2.8313 Acc: 0.5794

Epoch 48/199
----------
train Loss: 2.8873 Acc: 0.5784
val Loss: 2.8085 Acc: 0.5843

Epoch 49/199
----------
train Loss: 2.8579 Acc: 0.6137
val Loss: 2.7868 Acc: 0.5853

Epoch 50/199
----------
train Loss: 2.8738 Acc: 0.5853
val Loss: 2.7637 Acc: 0.5784

Epoch 51/199
----------
train Loss: 2.7714 Acc: 0.6294
val Loss: 2.7252 Acc: 0.5882

Epoch 52/199
----------
train Loss: 2.7641 Acc: 0.6069
val Loss: 2.7106 Acc: 0.5922

Epoch 53/199
----------
train Loss: 2.7484 Acc: 0.6176
val Loss: 2.6927 Acc: 0.5971

Epoch 54/199
----------
train Loss: 2.7512 Acc: 0.5980
val Loss: 2.6767 Acc: 0.5902

Epoch 55/199
----------
train Loss: 2.7321 Acc: 0.5971
val Loss: 2.6572 Acc: 0.5853

Epoch 56/199
----------
```

```
train Loss: 2.6814 Acc: 0.6088
val Loss: 2.6394 Acc: 0.5941

Epoch 57/199
----------
train Loss: 2.6973 Acc: 0.6010
val Loss: 2.6196 Acc: 0.5941

Epoch 58/199
----------
train Loss: 2.6467 Acc: 0.6324
val Loss: 2.6015 Acc: 0.6029

Epoch 59/199
----------
train Loss: 2.6433 Acc: 0.6196
val Loss: 2.5805 Acc: 0.6049

Epoch 60/199
----------
train Loss: 2.6406 Acc: 0.6235
val Loss: 2.5743 Acc: 0.6049

Epoch 61/199
----------
train Loss: 2.6197 Acc: 0.6029
val Loss: 2.5696 Acc: 0.6059

Epoch 62/199
----------
train Loss: 2.5862 Acc: 0.6422
val Loss: 2.5764 Acc: 0.6069

Epoch 63/199
----------
train Loss: 2.6080 Acc: 0.6382
val Loss: 2.5617 Acc: 0.6069

Epoch 64/199
----------
train Loss: 2.5657 Acc: 0.6441
val Loss: 2.5663 Acc: 0.6069

Epoch 65/199
----------
train Loss: 2.5760 Acc: 0.6471
val Loss: 2.5631 Acc: 0.6078
```

```
Epoch 66/199
----------
train Loss: 2.5970 Acc: 0.6314
val Loss: 2.5593 Acc: 0.6059

Epoch 67/199
----------
train Loss: 2.5645 Acc: 0.6392
val Loss: 2.5525 Acc: 0.6059

Epoch 68/199
----------
train Loss: 2.6207 Acc: 0.6324
val Loss: 2.5630 Acc: 0.6059

Epoch 69/199
----------
train Loss: 2.5826 Acc: 0.6333
val Loss: 2.5594 Acc: 0.6088

Epoch 70/199
----------
train Loss: 2.6078 Acc: 0.6392
val Loss: 2.5560 Acc: 0.6010

Epoch 71/199
----------
train Loss: 2.5945 Acc: 0.6333
val Loss: 2.5552 Acc: 0.6010

Epoch 72/199
----------
train Loss: 2.5796 Acc: 0.6422
val Loss: 2.5505 Acc: 0.6049

Epoch 73/199
----------
train Loss: 2.6162 Acc: 0.6255
val Loss: 2.5550 Acc: 0.6088

Epoch 74/199
----------
train Loss: 2.5629 Acc: 0.6706
val Loss: 2.5529 Acc: 0.6098

Epoch 75/199
----------
train Loss: 2.5399 Acc: 0.6735
```

```
val Loss: 2.5442 Acc: 0.6078

Epoch 76/199
----------
train Loss: 2.6022 Acc: 0.6167
val Loss: 2.5408 Acc: 0.6078

Epoch 77/199
----------
train Loss: 2.5471 Acc: 0.6500
val Loss: 2.5436 Acc: 0.6088

Epoch 78/199
----------
train Loss: 2.5686 Acc: 0.6598
val Loss: 2.5432 Acc: 0.6118

Epoch 79/199
----------
train Loss: 2.5217 Acc: 0.6451
val Loss: 2.5350 Acc: 0.6088

Epoch 80/199
----------
train Loss: 2.5622 Acc: 0.6284
val Loss: 2.5280 Acc: 0.6157

Epoch 81/199
----------
train Loss: 2.5934 Acc: 0.6343
val Loss: 2.5430 Acc: 0.6010

Epoch 82/199
----------
train Loss: 2.5811 Acc: 0.6284
val Loss: 2.5322 Acc: 0.6069

Epoch 83/199
----------
train Loss: 2.5588 Acc: 0.6294
val Loss: 2.5229 Acc: 0.6059

Epoch 84/199
----------
train Loss: 2.5627 Acc: 0.6324
val Loss: 2.5306 Acc: 0.6069

Epoch 85/199
```

```
----------
train Loss: 2.5484 Acc: 0.6480
val Loss: 2.5203 Acc: 0.6108

Epoch 86/199
----------
train Loss: 2.5314 Acc: 0.6441
val Loss: 2.5187 Acc: 0.6078

Epoch 87/199
----------
train Loss: 2.5808 Acc: 0.6275
val Loss: 2.5258 Acc: 0.6127

Epoch 88/199
----------
train Loss: 2.5404 Acc: 0.6598
val Loss: 2.5181 Acc: 0.6059

Epoch 89/199
----------
train Loss: 2.5157 Acc: 0.6598
val Loss: 2.5103 Acc: 0.6069

Epoch 90/199
----------
train Loss: 2.5693 Acc: 0.6196
val Loss: 2.5111 Acc: 0.6127

Epoch 91/199
----------
train Loss: 2.5544 Acc: 0.6343
val Loss: 2.5169 Acc: 0.6078

Epoch 92/199
----------
train Loss: 2.5615 Acc: 0.6343
val Loss: 2.5102 Acc: 0.6118

Epoch 93/199
----------
train Loss: 2.5681 Acc: 0.6353
val Loss: 2.5110 Acc: 0.6059

Epoch 94/199
----------
train Loss: 2.5314 Acc: 0.6373
val Loss: 2.5134 Acc: 0.6118
```

```
Epoch 95/199
----------
train Loss: 2.5264 Acc: 0.6510
val Loss: 2.5072 Acc: 0.6108

Epoch 96/199
----------
train Loss: 2.5691 Acc: 0.6392
val Loss: 2.5254 Acc: 0.6069

Epoch 97/199
----------
train Loss: 2.5201 Acc: 0.6490
val Loss: 2.5167 Acc: 0.6167

Epoch 98/199
----------
train Loss: 2.4966 Acc: 0.6559
val Loss: 2.4953 Acc: 0.6127

Epoch 99/199
----------
train Loss: 2.4851 Acc: 0.6402
val Loss: 2.4977 Acc: 0.6088

Epoch 100/199
----------
train Loss: 2.4948 Acc: 0.6461
val Loss: 2.5023 Acc: 0.6059

Epoch 101/199
----------
train Loss: 2.5561 Acc: 0.6324
val Loss: 2.5060 Acc: 0.6108

Epoch 102/199
----------
train Loss: 2.5127 Acc: 0.6490
val Loss: 2.5043 Acc: 0.6137

Epoch 103/199
----------
train Loss: 2.5081 Acc: 0.6559
val Loss: 2.4953 Acc: 0.6118

Epoch 104/199
----------
```

```
train Loss: 2.5030 Acc: 0.6598
val Loss: 2.4922 Acc: 0.6127

Epoch 105/199
----------
train Loss: 2.5056 Acc: 0.6529
val Loss: 2.4917 Acc: 0.6137

Epoch 106/199
----------
train Loss: 2.5023 Acc: 0.6412
val Loss: 2.4893 Acc: 0.6108

Epoch 107/199
----------
train Loss: 2.5245 Acc: 0.6510
val Loss: 2.4871 Acc: 0.6088

Epoch 108/199
----------
train Loss: 2.5127 Acc: 0.6353
val Loss: 2.4825 Acc: 0.6098

Epoch 109/199
----------
train Loss: 2.5012 Acc: 0.6618
val Loss: 2.4802 Acc: 0.6088

Epoch 110/199
----------
train Loss: 2.5330 Acc: 0.6275
val Loss: 2.4927 Acc: 0.6118

Epoch 111/199
----------
train Loss: 2.5277 Acc: 0.6461
val Loss: 2.4845 Acc: 0.6118

Epoch 112/199
----------
train Loss: 2.5471 Acc: 0.6196
val Loss: 2.4796 Acc: 0.6078

Epoch 113/199
----------
train Loss: 2.5264 Acc: 0.6578
val Loss: 2.4777 Acc: 0.6137
```

Epoch 114/199
----
train Loss: 2.5586 Acc: 0.6284
val Loss: 2.4863 Acc: 0.6049

Epoch 115/199
----
train Loss: 2.5003 Acc: 0.6618
val Loss: 2.4773 Acc: 0.6127

Epoch 116/199
----
train Loss: 2.5162 Acc: 0.6196
val Loss: 2.4730 Acc: 0.6196

Epoch 117/199
----
train Loss: 2.5164 Acc: 0.6431
val Loss: 2.4805 Acc: 0.6137

Epoch 118/199
----
train Loss: 2.5132 Acc: 0.6618
val Loss: 2.4824 Acc: 0.6157

Epoch 119/199
----
train Loss: 2.4918 Acc: 0.6490
val Loss: 2.4735 Acc: 0.6137

Epoch 120/199
----
train Loss: 2.5124 Acc: 0.6412
val Loss: 2.4721 Acc: 0.6127

Epoch 121/199
----
train Loss: 2.4964 Acc: 0.6500
val Loss: 2.4734 Acc: 0.6118

Epoch 122/199
----
train Loss: 2.5042 Acc: 0.6608
val Loss: 2.4815 Acc: 0.6118

Epoch 123/199
----
train Loss: 2.5076 Acc: 0.6353

```
val Loss: 2.4763 Acc: 0.6108

Epoch 124/199
----------
train Loss: 2.5041 Acc: 0.6373
val Loss: 2.4656 Acc: 0.6127

Epoch 125/199
----------
train Loss: 2.4646 Acc: 0.6549
val Loss: 2.4581 Acc: 0.6127

Epoch 126/199
----------
train Loss: 2.5090 Acc: 0.6588
val Loss: 2.4691 Acc: 0.6157

Epoch 127/199
----------
train Loss: 2.5068 Acc: 0.6490
val Loss: 2.4598 Acc: 0.6167

Epoch 128/199
----------
train Loss: 2.5126 Acc: 0.6343
val Loss: 2.4693 Acc: 0.6098

Epoch 129/199
----------
train Loss: 2.4619 Acc: 0.6569
val Loss: 2.4611 Acc: 0.6147

Epoch 130/199
----------
train Loss: 2.5096 Acc: 0.6275
val Loss: 2.4792 Acc: 0.6147

Epoch 131/199
----------
train Loss: 2.4714 Acc: 0.6588
val Loss: 2.4720 Acc: 0.6127

Epoch 132/199
----------
train Loss: 2.4891 Acc: 0.6333
val Loss: 2.4679 Acc: 0.6147

Epoch 133/199
```

```
----------
train Loss: 2.4688 Acc: 0.6529
val Loss: 2.4694 Acc: 0.6108

Epoch 134/199
----------
train Loss: 2.4890 Acc: 0.6392
val Loss: 2.4681 Acc: 0.6206

Epoch 135/199
----------
train Loss: 2.4865 Acc: 0.6578
val Loss: 2.4655 Acc: 0.6137

Epoch 136/199
----------
train Loss: 2.5060 Acc: 0.6363
val Loss: 2.4694 Acc: 0.6137

Epoch 137/199
----------
train Loss: 2.5068 Acc: 0.6392
val Loss: 2.4572 Acc: 0.6147

Epoch 138/199
----------
train Loss: 2.5295 Acc: 0.6392
val Loss: 2.4632 Acc: 0.6088

Epoch 139/199
----------
train Loss: 2.4798 Acc: 0.6627
val Loss: 2.4653 Acc: 0.6137

Epoch 140/199
----------
train Loss: 2.4817 Acc: 0.6304
val Loss: 2.4547 Acc: 0.6235

Epoch 141/199
----------
train Loss: 2.5104 Acc: 0.6373
val Loss: 2.4788 Acc: 0.6108

Epoch 142/199
----------
train Loss: 2.4779 Acc: 0.6412
val Loss: 2.4661 Acc: 0.6147
```

```
Epoch 143/199
----------
train Loss: 2.4981 Acc: 0.6245
val Loss: 2.4762 Acc: 0.6078

Epoch 144/199
----------
train Loss: 2.4859 Acc: 0.6402
val Loss: 2.4698 Acc: 0.6167

Epoch 145/199
----------
train Loss: 2.5204 Acc: 0.6284
val Loss: 2.4643 Acc: 0.6157

Epoch 146/199
----------
train Loss: 2.5126 Acc: 0.6441
val Loss: 2.4591 Acc: 0.6186

Epoch 147/199
----------
train Loss: 2.4826 Acc: 0.6451
val Loss: 2.4555 Acc: 0.6196

Epoch 148/199
----------
train Loss: 2.5189 Acc: 0.6373
val Loss: 2.4601 Acc: 0.6127

Epoch 149/199
----------
train Loss: 2.4872 Acc: 0.6461
val Loss: 2.4679 Acc: 0.6137

Epoch 150/199
----------
train Loss: 2.4957 Acc: 0.6412
val Loss: 2.4629 Acc: 0.6167

Epoch 151/199
----------
train Loss: 2.4916 Acc: 0.6451
val Loss: 2.4665 Acc: 0.6059

Epoch 152/199
----------
```

```
train Loss: 2.5284 Acc: 0.6422
val Loss: 2.4679 Acc: 0.6108

Epoch 153/199
----------
train Loss: 2.4562 Acc: 0.6500
val Loss: 2.4651 Acc: 0.6147

Epoch 154/199
----------
train Loss: 2.5334 Acc: 0.6353
val Loss: 2.4669 Acc: 0.6137

Epoch 155/199
----------
train Loss: 2.4922 Acc: 0.6598
val Loss: 2.4614 Acc: 0.6127

Epoch 156/199
----------
train Loss: 2.5090 Acc: 0.6294
val Loss: 2.4662 Acc: 0.6137

Epoch 157/199
----------
train Loss: 2.5197 Acc: 0.6422
val Loss: 2.4749 Acc: 0.6167

Epoch 158/199
----------
train Loss: 2.4842 Acc: 0.6637
val Loss: 2.4727 Acc: 0.6118

Epoch 159/199
----------
train Loss: 2.4932 Acc: 0.6618
val Loss: 2.4662 Acc: 0.6157

Epoch 160/199
----------
train Loss: 2.4899 Acc: 0.6529
val Loss: 2.4589 Acc: 0.6157

Epoch 161/199
----------
train Loss: 2.4673 Acc: 0.6510
val Loss: 2.4637 Acc: 0.6206
```

```
Epoch 162/199
----------
train Loss: 2.4913 Acc: 0.6510
val Loss: 2.4637 Acc: 0.6186

Epoch 163/199
----------
train Loss: 2.5052 Acc: 0.6471
val Loss: 2.4672 Acc: 0.6147

Epoch 164/199
----------
train Loss: 2.5013 Acc: 0.6412
val Loss: 2.4665 Acc: 0.6118

Epoch 165/199
----------
train Loss: 2.4961 Acc: 0.6549
val Loss: 2.4741 Acc: 0.6098

Epoch 166/199
----------
train Loss: 2.5221 Acc: 0.6343
val Loss: 2.4672 Acc: 0.6098

Epoch 167/199
----------
train Loss: 2.5023 Acc: 0.6461
val Loss: 2.4767 Acc: 0.6088

Epoch 168/199
----------
train Loss: 2.5188 Acc: 0.6402
val Loss: 2.4672 Acc: 0.6118

Epoch 169/199
----------
train Loss: 2.4619 Acc: 0.6559
val Loss: 2.4628 Acc: 0.6147

Epoch 170/199
----------
train Loss: 2.4836 Acc: 0.6510
val Loss: 2.4694 Acc: 0.6167

Epoch 171/199
----------
train Loss: 2.4895 Acc: 0.6500
```

```
val Loss: 2.4580 Acc: 0.6176

Epoch 172/199
----------
train Loss: 2.5119 Acc: 0.6500
val Loss: 2.4529 Acc: 0.6167

Epoch 173/199
----------
train Loss: 2.4666 Acc: 0.6422
val Loss: 2.4608 Acc: 0.6137

Epoch 174/199
----------
train Loss: 2.4847 Acc: 0.6333
val Loss: 2.4558 Acc: 0.6176

Epoch 175/199
----------
train Loss: 2.4838 Acc: 0.6451
val Loss: 2.4597 Acc: 0.6137

Epoch 176/199
----------
train Loss: 2.5134 Acc: 0.6588
val Loss: 2.4637 Acc: 0.6167

Epoch 177/199
----------
train Loss: 2.4713 Acc: 0.6461
val Loss: 2.4532 Acc: 0.6167

Epoch 178/199
----------
train Loss: 2.4927 Acc: 0.6559
val Loss: 2.4536 Acc: 0.6206

Epoch 179/199
----------
train Loss: 2.4815 Acc: 0.6549
val Loss: 2.4522 Acc: 0.6118

Epoch 180/199
----------
train Loss: 2.4957 Acc: 0.6392
val Loss: 2.4588 Acc: 0.6176

Epoch 181/199
```

```
----------
train Loss: 2.4771 Acc: 0.6549
val Loss: 2.4560 Acc: 0.6157

Epoch 182/199
----------
train Loss: 2.4800 Acc: 0.6441
val Loss: 2.4505 Acc: 0.6176

Epoch 183/199
----------
train Loss: 2.4965 Acc: 0.6431
val Loss: 2.4563 Acc: 0.6186

Epoch 184/199
----------
train Loss: 2.4816 Acc: 0.6500
val Loss: 2.4585 Acc: 0.6147

Epoch 185/199
----------
train Loss: 2.4823 Acc: 0.6363
val Loss: 2.4660 Acc: 0.6186

Epoch 186/199
----------
train Loss: 2.5032 Acc: 0.6304
val Loss: 2.4567 Acc: 0.6108

Epoch 187/199
----------
train Loss: 2.5043 Acc: 0.6480
val Loss: 2.4649 Acc: 0.6078

Epoch 188/199
----------
train Loss: 2.5035 Acc: 0.6255
val Loss: 2.4571 Acc: 0.6157

Epoch 189/199
----------
train Loss: 2.4879 Acc: 0.6373
val Loss: 2.4544 Acc: 0.6108

Epoch 190/199
----------
train Loss: 2.4848 Acc: 0.6588
val Loss: 2.4480 Acc: 0.6206
```

```
Epoch 191/199
----------
train Loss: 2.4907 Acc: 0.6402
val Loss: 2.4599 Acc: 0.6176

Epoch 192/199
----------
train Loss: 2.4602 Acc: 0.6569
val Loss: 2.4555 Acc: 0.6157

Epoch 193/199
----------
train Loss: 2.5013 Acc: 0.6324
val Loss: 2.4612 Acc: 0.6127

Epoch 194/199
----------
train Loss: 2.5311 Acc: 0.6216
val Loss: 2.4689 Acc: 0.6127

Epoch 195/199
----------
train Loss: 2.4902 Acc: 0.6549
val Loss: 2.4696 Acc: 0.6127

Epoch 196/199
----------
train Loss: 2.4778 Acc: 0.6510
val Loss: 2.4650 Acc: 0.6176

Epoch 197/199
----------
train Loss: 2.4910 Acc: 0.6412
val Loss: 2.4629 Acc: 0.6157

Epoch 198/199
----------
train Loss: 2.5037 Acc: 0.6431
val Loss: 2.4577 Acc: 0.6206

Epoch 199/199
----------
train Loss: 2.4951 Acc: 0.6382
val Loss: 2.4696 Acc: 0.6127

Training complete in 15m 7s
Best val Acc: 0.623529
```

```
----- Performing experiment with fixed lr=0.1 -----

Epoch 0/199
----------
train Loss: 4.6838 Acc: 0.0059
val Loss: 4.6210 Acc: 0.0127

Epoch 1/199
----------
train Loss: 4.6140 Acc: 0.0137
val Loss: 4.5573 Acc: 0.0284

Epoch 2/199
----------
train Loss: 4.5528 Acc: 0.0255
val Loss: 4.4999 Acc: 0.0471

Epoch 3/199
----------
train Loss: 4.5063 Acc: 0.0382
val Loss: 4.4476 Acc: 0.0706

Epoch 4/199
----------
train Loss: 4.4571 Acc: 0.0686
val Loss: 4.3970 Acc: 0.1216

Epoch 5/199
----------
train Loss: 4.4003 Acc: 0.1157
val Loss: 4.3453 Acc: 0.1578

Epoch 6/199
----------
train Loss: 4.3546 Acc: 0.1500
val Loss: 4.2947 Acc: 0.2000

Epoch 7/199
----------
train Loss: 4.3016 Acc: 0.1931
val Loss: 4.2440 Acc: 0.2441

Epoch 8/199
----------
train Loss: 4.2490 Acc: 0.2569
val Loss: 4.1939 Acc: 0.2804
```

```
Epoch 9/199
----------
train Loss: 4.2124 Acc: 0.2529
val Loss: 4.1457 Acc: 0.2990

Epoch 10/199
----------
train Loss: 4.1681 Acc: 0.2912
val Loss: 4.0945 Acc: 0.3363

Epoch 11/199
----------
train Loss: 4.1077 Acc: 0.3255
val Loss: 4.0510 Acc: 0.3451

Epoch 12/199
----------
train Loss: 4.0636 Acc: 0.3657
val Loss: 4.0014 Acc: 0.3696

Epoch 13/199
----------
train Loss: 4.0371 Acc: 0.3824
val Loss: 3.9577 Acc: 0.3814

Epoch 14/199
----------
train Loss: 3.9758 Acc: 0.3912
val Loss: 3.9085 Acc: 0.4049

Epoch 15/199
----------
train Loss: 3.9360 Acc: 0.3912
val Loss: 3.8657 Acc: 0.4245

Epoch 16/199
----------
train Loss: 3.9135 Acc: 0.4020
val Loss: 3.8250 Acc: 0.4225

Epoch 17/199
----------
train Loss: 3.8360 Acc: 0.4412
val Loss: 3.7778 Acc: 0.4529

Epoch 18/199
----------
train Loss: 3.8113 Acc: 0.4382
```

```
val Loss: 3.7441 Acc: 0.4431

Epoch 19/199
----------
train Loss: 3.7753 Acc: 0.4520
val Loss: 3.6976 Acc: 0.4588

Epoch 20/199
----------
train Loss: 3.7317 Acc: 0.4490
val Loss: 3.6539 Acc: 0.4824

Epoch 21/199
----------
train Loss: 3.7037 Acc: 0.4559
val Loss: 3.6221 Acc: 0.4735

Epoch 22/199
----------
train Loss: 3.6714 Acc: 0.4588
val Loss: 3.5802 Acc: 0.4853

Epoch 23/199
----------
train Loss: 3.6224 Acc: 0.4706
val Loss: 3.5417 Acc: 0.4990

Epoch 24/199
----------
train Loss: 3.5730 Acc: 0.4931
val Loss: 3.5034 Acc: 0.5137

Epoch 25/199
----------
train Loss: 3.5439 Acc: 0.5127
val Loss: 3.4679 Acc: 0.5088

Epoch 26/199
----------
train Loss: 3.5151 Acc: 0.5029
val Loss: 3.4271 Acc: 0.5206

Epoch 27/199
----------
train Loss: 3.4842 Acc: 0.5176
val Loss: 3.4021 Acc: 0.5196

Epoch 28/199
```

```
----------
train Loss: 3.4475 Acc: 0.5186
val Loss: 3.3603 Acc: 0.5245

Epoch 29/199
----------
train Loss: 3.4114 Acc: 0.5235
val Loss: 3.3255 Acc: 0.5402

Epoch 30/199
----------
train Loss: 3.3805 Acc: 0.5314
val Loss: 3.2979 Acc: 0.5373

Epoch 31/199
----------
train Loss: 3.3171 Acc: 0.5353
val Loss: 3.2631 Acc: 0.5392

Epoch 32/199
----------
train Loss: 3.3007 Acc: 0.5373
val Loss: 3.2286 Acc: 0.5471

Epoch 33/199
----------
train Loss: 3.2847 Acc: 0.5324
val Loss: 3.1919 Acc: 0.5559

Epoch 34/199
----------
train Loss: 3.2709 Acc: 0.5363
val Loss: 3.1720 Acc: 0.5627

Epoch 35/199
----------
train Loss: 3.1996 Acc: 0.5775
val Loss: 3.1348 Acc: 0.5647

Epoch 36/199
----------
train Loss: 3.1896 Acc: 0.5676
val Loss: 3.1184 Acc: 0.5627

Epoch 37/199
----------
train Loss: 3.1518 Acc: 0.5833
val Loss: 3.0800 Acc: 0.5775
```

```
Epoch 38/199
----------
train Loss: 3.1369 Acc: 0.5520
val Loss: 3.0545 Acc: 0.5804

Epoch 39/199
----------
train Loss: 3.1292 Acc: 0.5529
val Loss: 3.0348 Acc: 0.5706

Epoch 40/199
----------
train Loss: 3.0802 Acc: 0.5647
val Loss: 2.9957 Acc: 0.5745

Epoch 41/199
----------
train Loss: 3.0544 Acc: 0.5578
val Loss: 2.9702 Acc: 0.5775

Epoch 42/199
----------
train Loss: 2.9834 Acc: 0.5961
val Loss: 2.9340 Acc: 0.5892

Epoch 43/199
----------
train Loss: 2.9784 Acc: 0.6098
val Loss: 2.9160 Acc: 0.5931

Epoch 44/199
----------
train Loss: 2.9579 Acc: 0.5765
val Loss: 2.8906 Acc: 0.5941

Epoch 45/199
----------
train Loss: 2.9821 Acc: 0.5696
val Loss: 2.8796 Acc: 0.5863

Epoch 46/199
----------
train Loss: 2.8893 Acc: 0.6108
val Loss: 2.8468 Acc: 0.5892

Epoch 47/199
----------
```

```
train Loss: 2.8855 Acc: 0.6000
val Loss: 2.8084 Acc: 0.6039

Epoch 48/199
----------
train Loss: 2.8648 Acc: 0.5980
val Loss: 2.7904 Acc: 0.5922

Epoch 49/199
----------
train Loss: 2.8380 Acc: 0.5961
val Loss: 2.7698 Acc: 0.5990

Epoch 50/199
----------
train Loss: 2.8231 Acc: 0.5922
val Loss: 2.7435 Acc: 0.6059

Epoch 51/199
----------
train Loss: 2.7902 Acc: 0.6029
val Loss: 2.7200 Acc: 0.5980

Epoch 52/199
----------
train Loss: 2.7197 Acc: 0.6343
val Loss: 2.6892 Acc: 0.6157

Epoch 53/199
----------
train Loss: 2.7520 Acc: 0.6127
val Loss: 2.6768 Acc: 0.6029

Epoch 54/199
----------
train Loss: 2.7254 Acc: 0.6000
val Loss: 2.6597 Acc: 0.6088

Epoch 55/199
----------
train Loss: 2.7085 Acc: 0.6167
val Loss: 2.6340 Acc: 0.6127

Epoch 56/199
----------
train Loss: 2.6882 Acc: 0.6176
val Loss: 2.6137 Acc: 0.6069
```

```
Epoch 57/199
----------
train Loss: 2.6455 Acc: 0.6353
val Loss: 2.6032 Acc: 0.6157

Epoch 58/199
----------
train Loss: 2.6717 Acc: 0.6088
val Loss: 2.5874 Acc: 0.6216

Epoch 59/199
----------
train Loss: 2.6210 Acc: 0.6284
val Loss: 2.5656 Acc: 0.6167

Epoch 60/199
----------
train Loss: 2.5804 Acc: 0.6480
val Loss: 2.5625 Acc: 0.6167

Epoch 61/199
----------
train Loss: 2.5720 Acc: 0.6422
val Loss: 2.5617 Acc: 0.6157

Epoch 62/199
----------
train Loss: 2.5999 Acc: 0.6500
val Loss: 2.5655 Acc: 0.6167

Epoch 63/199
----------
train Loss: 2.5924 Acc: 0.6402
val Loss: 2.5553 Acc: 0.6196

Epoch 64/199
----------
train Loss: 2.6185 Acc: 0.6088
val Loss: 2.5585 Acc: 0.6147

Epoch 65/199
----------
train Loss: 2.5852 Acc: 0.6431
val Loss: 2.5430 Acc: 0.6196

Epoch 66/199
----------
train Loss: 2.5994 Acc: 0.6235
```

```
val Loss: 2.5588 Acc: 0.6176

Epoch 67/199
----------
train Loss: 2.5923 Acc: 0.6255
val Loss: 2.5570 Acc: 0.6157

Epoch 68/199
----------
train Loss: 2.5691 Acc: 0.6471
val Loss: 2.5522 Acc: 0.6206

Epoch 69/199
----------
train Loss: 2.5873 Acc: 0.6275
val Loss: 2.5475 Acc: 0.6225

Epoch 70/199
----------
train Loss: 2.6123 Acc: 0.6324
val Loss: 2.5461 Acc: 0.6206

Epoch 71/199
----------
train Loss: 2.5958 Acc: 0.6373
val Loss: 2.5458 Acc: 0.6186

Epoch 72/199
----------
train Loss: 2.5952 Acc: 0.6284
val Loss: 2.5410 Acc: 0.6216

Epoch 73/199
----------
train Loss: 2.5321 Acc: 0.6559
val Loss: 2.5286 Acc: 0.6265

Epoch 74/199
----------
train Loss: 2.5697 Acc: 0.6500
val Loss: 2.5287 Acc: 0.6216

Epoch 75/199
----------
train Loss: 2.6061 Acc: 0.6157
val Loss: 2.5318 Acc: 0.6216

Epoch 76/199
```

```
----------
train Loss: 2.5610 Acc: 0.6392
val Loss: 2.5380 Acc: 0.6225

Epoch 77/199
----------
train Loss: 2.5826 Acc: 0.6618
val Loss: 2.5372 Acc: 0.6186

Epoch 78/199
----------
train Loss: 2.5371 Acc: 0.6529
val Loss: 2.5131 Acc: 0.6245

Epoch 79/199
----------
train Loss: 2.5820 Acc: 0.6304
val Loss: 2.5362 Acc: 0.6186

Epoch 80/199
----------
train Loss: 2.5746 Acc: 0.6373
val Loss: 2.5256 Acc: 0.6186

Epoch 81/199
----------
train Loss: 2.5568 Acc: 0.6324
val Loss: 2.5209 Acc: 0.6294

Epoch 82/199
----------
train Loss: 2.5486 Acc: 0.6422
val Loss: 2.5180 Acc: 0.6186

Epoch 83/199
----------
train Loss: 2.5168 Acc: 0.6471
val Loss: 2.5094 Acc: 0.6265

Epoch 84/199
----------
train Loss: 2.5754 Acc: 0.6392
val Loss: 2.5211 Acc: 0.6137

Epoch 85/199
----------
train Loss: 2.5623 Acc: 0.6559
val Loss: 2.5153 Acc: 0.6235
```

```
Epoch 86/199
----------
train Loss: 2.5337 Acc: 0.6520
val Loss: 2.5085 Acc: 0.6216

Epoch 87/199
----------
train Loss: 2.5357 Acc: 0.6725
val Loss: 2.5026 Acc: 0.6196

Epoch 88/199
----------
train Loss: 2.5674 Acc: 0.6373
val Loss: 2.5077 Acc: 0.6176

Epoch 89/199
----------
train Loss: 2.5498 Acc: 0.6510
val Loss: 2.5050 Acc: 0.6137

Epoch 90/199
----------
train Loss: 2.5206 Acc: 0.6324
val Loss: 2.5093 Acc: 0.6206

Epoch 91/199
----------
train Loss: 2.5421 Acc: 0.6363
val Loss: 2.5050 Acc: 0.6235

Epoch 92/199
----------
train Loss: 2.5321 Acc: 0.6412
val Loss: 2.5107 Acc: 0.6167

Epoch 93/199
----------
train Loss: 2.5165 Acc: 0.6490
val Loss: 2.5059 Acc: 0.6196

Epoch 94/199
----------
train Loss: 2.5149 Acc: 0.6353
val Loss: 2.4929 Acc: 0.6245

Epoch 95/199
----------
```

```
train Loss: 2.5099 Acc: 0.6627
val Loss: 2.4996 Acc: 0.6225

Epoch 96/199
----------
train Loss: 2.5025 Acc: 0.6647
val Loss: 2.4933 Acc: 0.6265

Epoch 97/199
----------
train Loss: 2.5154 Acc: 0.6618
val Loss: 2.4840 Acc: 0.6235

Epoch 98/199
----------
train Loss: 2.5729 Acc: 0.6225
val Loss: 2.4999 Acc: 0.6196

Epoch 99/199
----------
train Loss: 2.5443 Acc: 0.6186
val Loss: 2.4926 Acc: 0.6235

Epoch 100/199
----------
train Loss: 2.5612 Acc: 0.6245
val Loss: 2.4969 Acc: 0.6167

Epoch 101/199
----------
train Loss: 2.5179 Acc: 0.6647
val Loss: 2.4836 Acc: 0.6216

Epoch 102/199
----------
train Loss: 2.5030 Acc: 0.6549
val Loss: 2.4792 Acc: 0.6225

Epoch 103/199
----------
train Loss: 2.5270 Acc: 0.6333
val Loss: 2.4863 Acc: 0.6225

Epoch 104/199
----------
train Loss: 2.5336 Acc: 0.6294
val Loss: 2.4861 Acc: 0.6245
```

```
Epoch 105/199
----------
train Loss: 2.5016 Acc: 0.6471
val Loss: 2.4673 Acc: 0.6314

Epoch 106/199
----------
train Loss: 2.5474 Acc: 0.6304
val Loss: 2.4714 Acc: 0.6255

Epoch 107/199
----------
train Loss: 2.4759 Acc: 0.6549
val Loss: 2.4598 Acc: 0.6324

Epoch 108/199
----------
train Loss: 2.4845 Acc: 0.6569
val Loss: 2.4631 Acc: 0.6255

Epoch 109/199
----------
train Loss: 2.5076 Acc: 0.6431
val Loss: 2.4785 Acc: 0.6275

Epoch 110/199
----------
train Loss: 2.5723 Acc: 0.6167
val Loss: 2.4931 Acc: 0.6206

Epoch 111/199
----------
train Loss: 2.5562 Acc: 0.6353
val Loss: 2.4750 Acc: 0.6206

Epoch 112/199
----------
train Loss: 2.4933 Acc: 0.6461
val Loss: 2.4651 Acc: 0.6265

Epoch 113/199
----------
train Loss: 2.5219 Acc: 0.6422
val Loss: 2.4649 Acc: 0.6255

Epoch 114/199
----------
train Loss: 2.4992 Acc: 0.6706
```

```
val Loss: 2.4722 Acc: 0.6176

Epoch 115/199
----------
train Loss: 2.4936 Acc: 0.6549
val Loss: 2.4625 Acc: 0.6118

Epoch 116/199
----------
train Loss: 2.4876 Acc: 0.6324
val Loss: 2.4562 Acc: 0.6265

Epoch 117/199
----------
train Loss: 2.5169 Acc: 0.6284
val Loss: 2.4694 Acc: 0.6235

Epoch 118/199
----------
train Loss: 2.5015 Acc: 0.6451
val Loss: 2.4626 Acc: 0.6304

Epoch 119/199
----------
train Loss: 2.4571 Acc: 0.6647
val Loss: 2.4472 Acc: 0.6314

Epoch 120/199
----------
train Loss: 2.4864 Acc: 0.6382
val Loss: 2.4466 Acc: 0.6255

Epoch 121/199
----------
train Loss: 2.5224 Acc: 0.6402
val Loss: 2.4566 Acc: 0.6196

Epoch 122/199
----------
train Loss: 2.5149 Acc: 0.6471
val Loss: 2.4511 Acc: 0.6235

Epoch 123/199
----------
train Loss: 2.4808 Acc: 0.6402
val Loss: 2.4555 Acc: 0.6255

Epoch 124/199
```

```
----------
train Loss: 2.5301 Acc: 0.6382
val Loss: 2.4597 Acc: 0.6275

Epoch 125/199
----------
train Loss: 2.4620 Acc: 0.6647
val Loss: 2.4570 Acc: 0.6275

Epoch 126/199
----------
train Loss: 2.4811 Acc: 0.6402
val Loss: 2.4552 Acc: 0.6255

Epoch 127/199
----------
train Loss: 2.5076 Acc: 0.6422
val Loss: 2.4542 Acc: 0.6216

Epoch 128/199
----------
train Loss: 2.4946 Acc: 0.6441
val Loss: 2.4624 Acc: 0.6235

Epoch 129/199
----------
train Loss: 2.4944 Acc: 0.6510
val Loss: 2.4506 Acc: 0.6255

Epoch 130/199
----------
train Loss: 2.4822 Acc: 0.6598
val Loss: 2.4574 Acc: 0.6284

Epoch 131/199
----------
train Loss: 2.4447 Acc: 0.6559
val Loss: 2.4490 Acc: 0.6304

Epoch 132/199
----------
train Loss: 2.4720 Acc: 0.6686
val Loss: 2.4541 Acc: 0.6284

Epoch 133/199
----------
train Loss: 2.5135 Acc: 0.6529
val Loss: 2.4532 Acc: 0.6235
```

```
Epoch 134/199
----------
train Loss: 2.5129 Acc: 0.6382
val Loss: 2.4566 Acc: 0.6225

Epoch 135/199
----------
train Loss: 2.4914 Acc: 0.6422
val Loss: 2.4531 Acc: 0.6255

Epoch 136/199
----------
train Loss: 2.5138 Acc: 0.6461
val Loss: 2.4580 Acc: 0.6265

Epoch 137/199
----------
train Loss: 2.4714 Acc: 0.6353
val Loss: 2.4571 Acc: 0.6265

Epoch 138/199
----------
train Loss: 2.4760 Acc: 0.6539
val Loss: 2.4539 Acc: 0.6186

Epoch 139/199
----------
train Loss: 2.4578 Acc: 0.6578
val Loss: 2.4499 Acc: 0.6225

Epoch 140/199
----------
train Loss: 2.5081 Acc: 0.6490
val Loss: 2.4531 Acc: 0.6235

Epoch 141/199
----------
train Loss: 2.5079 Acc: 0.6265
val Loss: 2.4391 Acc: 0.6275

Epoch 142/199
----------
train Loss: 2.4933 Acc: 0.6441
val Loss: 2.4468 Acc: 0.6196

Epoch 143/199
----------
```

```
train Loss: 2.5370 Acc: 0.6324
val Loss: 2.4618 Acc: 0.6167

Epoch 144/199
----------
train Loss: 2.5012 Acc: 0.6500
val Loss: 2.4634 Acc: 0.6196

Epoch 145/199
----------
train Loss: 2.4852 Acc: 0.6441
val Loss: 2.4584 Acc: 0.6275

Epoch 146/199
----------
train Loss: 2.5108 Acc: 0.6324
val Loss: 2.4551 Acc: 0.6196

Epoch 147/199
----------
train Loss: 2.4540 Acc: 0.6618
val Loss: 2.4519 Acc: 0.6275

Epoch 148/199
----------
train Loss: 2.4689 Acc: 0.6608
val Loss: 2.4427 Acc: 0.6284

Epoch 149/199
----------
train Loss: 2.5092 Acc: 0.6480
val Loss: 2.4478 Acc: 0.6255

Epoch 150/199
----------
train Loss: 2.4954 Acc: 0.6324
val Loss: 2.4579 Acc: 0.6255

Epoch 151/199
----------
train Loss: 2.5340 Acc: 0.6265
val Loss: 2.4483 Acc: 0.6216

Epoch 152/199
----------
train Loss: 2.5354 Acc: 0.6353
val Loss: 2.4594 Acc: 0.6245
```

```
Epoch 153/199
----------
train Loss: 2.4619 Acc: 0.6716
val Loss: 2.4581 Acc: 0.6265

Epoch 154/199
----------
train Loss: 2.5153 Acc: 0.6373
val Loss: 2.4511 Acc: 0.6284

Epoch 155/199
----------
train Loss: 2.4236 Acc: 0.6735
val Loss: 2.4315 Acc: 0.6275

Epoch 156/199
----------
train Loss: 2.4502 Acc: 0.6559
val Loss: 2.4416 Acc: 0.6275

Epoch 157/199
----------
train Loss: 2.4622 Acc: 0.6588
val Loss: 2.4547 Acc: 0.6294

Epoch 158/199
----------
train Loss: 2.4698 Acc: 0.6431
val Loss: 2.4527 Acc: 0.6216

Epoch 159/199
----------
train Loss: 2.4719 Acc: 0.6402
val Loss: 2.4433 Acc: 0.6235

Epoch 160/199
----------
train Loss: 2.4933 Acc: 0.6471
val Loss: 2.4465 Acc: 0.6255

Epoch 161/199
----------
train Loss: 2.5100 Acc: 0.6500
val Loss: 2.4509 Acc: 0.6225

Epoch 162/199
----------
train Loss: 2.4398 Acc: 0.6618
```

```
val Loss: 2.4454 Acc: 0.6294

Epoch 163/199
----------
train Loss: 2.4914 Acc: 0.6608
val Loss: 2.4502 Acc: 0.6206

Epoch 164/199
----------
train Loss: 2.4597 Acc: 0.6745
val Loss: 2.4504 Acc: 0.6255

Epoch 165/199
----------
train Loss: 2.4940 Acc: 0.6490
val Loss: 2.4574 Acc: 0.6225

Epoch 166/199
----------
train Loss: 2.4978 Acc: 0.6451
val Loss: 2.4525 Acc: 0.6225

Epoch 167/199
----------
train Loss: 2.4993 Acc: 0.6520
val Loss: 2.4556 Acc: 0.6235

Epoch 168/199
----------
train Loss: 2.5156 Acc: 0.6265
val Loss: 2.4523 Acc: 0.6245

Epoch 169/199
----------
train Loss: 2.5003 Acc: 0.6275
val Loss: 2.4532 Acc: 0.6304

Epoch 170/199
----------
train Loss: 2.4739 Acc: 0.6608
val Loss: 2.4510 Acc: 0.6294

Epoch 171/199
----------
train Loss: 2.5165 Acc: 0.6431
val Loss: 2.4531 Acc: 0.6255

Epoch 172/199
```

```
----------
train Loss: 2.5012 Acc: 0.6392
val Loss: 2.4499 Acc: 0.6216

Epoch 173/199
----------
train Loss: 2.4703 Acc: 0.6618
val Loss: 2.4505 Acc: 0.6255

Epoch 174/199
----------
train Loss: 2.4929 Acc: 0.6343
val Loss: 2.4517 Acc: 0.6235

Epoch 175/199
----------
train Loss: 2.4904 Acc: 0.6402
val Loss: 2.4516 Acc: 0.6275

Epoch 176/199
----------
train Loss: 2.5286 Acc: 0.6225
val Loss: 2.4517 Acc: 0.6245

Epoch 177/199
----------
train Loss: 2.4563 Acc: 0.6451
val Loss: 2.4308 Acc: 0.6275

Epoch 178/199
----------
train Loss: 2.4828 Acc: 0.6343
val Loss: 2.4409 Acc: 0.6284

Epoch 179/199
----------
train Loss: 2.4968 Acc: 0.6510
val Loss: 2.4425 Acc: 0.6284

Epoch 180/199
----------
train Loss: 2.4800 Acc: 0.6441
val Loss: 2.4482 Acc: 0.6235

Epoch 181/199
----------
train Loss: 2.4583 Acc: 0.6627
val Loss: 2.4458 Acc: 0.6235
```

```
Epoch 182/199
----------
train Loss: 2.4920 Acc: 0.6431
val Loss: 2.4608 Acc: 0.6245

Epoch 183/199
----------
train Loss: 2.4762 Acc: 0.6549
val Loss: 2.4446 Acc: 0.6235

Epoch 184/199
----------
train Loss: 2.4660 Acc: 0.6333
val Loss: 2.4485 Acc: 0.6225

Epoch 185/199
----------
train Loss: 2.4632 Acc: 0.6618
val Loss: 2.4530 Acc: 0.6245

Epoch 186/199
----------
train Loss: 2.4641 Acc: 0.6480
val Loss: 2.4511 Acc: 0.6265

Epoch 187/199
----------
train Loss: 2.4969 Acc: 0.6539
val Loss: 2.4594 Acc: 0.6216

Epoch 188/199
----------
train Loss: 2.4645 Acc: 0.6686
val Loss: 2.4498 Acc: 0.6196

Epoch 189/199
----------
train Loss: 2.4586 Acc: 0.6598
val Loss: 2.4390 Acc: 0.6245

Epoch 190/199
----------
train Loss: 2.4738 Acc: 0.6304
val Loss: 2.4366 Acc: 0.6245

Epoch 191/199
----------
```

```
train Loss: 2.4518 Acc: 0.6686
val Loss: 2.4364 Acc: 0.6304

Epoch 192/199
----------
train Loss: 2.4806 Acc: 0.6529
val Loss: 2.4410 Acc: 0.6275

Epoch 193/199
----------
train Loss: 2.4819 Acc: 0.6284
val Loss: 2.4500 Acc: 0.6275

Epoch 194/199
----------
train Loss: 2.4723 Acc: 0.6539
val Loss: 2.4438 Acc: 0.6275

Epoch 195/199
----------
train Loss: 2.5020 Acc: 0.6461
val Loss: 2.4570 Acc: 0.6255

Epoch 196/199
----------
train Loss: 2.4813 Acc: 0.6461
val Loss: 2.4553 Acc: 0.6196

Epoch 197/199
----------
train Loss: 2.4526 Acc: 0.6647
val Loss: 2.4489 Acc: 0.6245

Epoch 198/199
----------
train Loss: 2.4445 Acc: 0.6696
val Loss: 2.4446 Acc: 0.6265

Epoch 199/199
----------
train Loss: 2.5059 Acc: 0.6196
val Loss: 2.4440 Acc: 0.6206

Training complete in 15m 8s
Best val Acc: 0.632353

----- Performing experiment with fixed lr=1 -----
```

```
Epoch 0/199
----------
train Loss: 4.6827 Acc: 0.0108
val Loss: 4.6661 Acc: 0.0078

Epoch 1/199
----------
train Loss: 4.6365 Acc: 0.0118
val Loss: 4.5956 Acc: 0.0235

Epoch 2/199
----------
train Loss: 4.5635 Acc: 0.0363
val Loss: 4.5384 Acc: 0.0441

Epoch 3/199
----------
train Loss: 4.5194 Acc: 0.0539
val Loss: 4.4867 Acc: 0.0676

Epoch 4/199
----------
train Loss: 4.4689 Acc: 0.0843
val Loss: 4.4343 Acc: 0.1000

Epoch 5/199
----------
train Loss: 4.4224 Acc: 0.0990
val Loss: 4.3836 Acc: 0.1422

Epoch 6/199
----------
train Loss: 4.3596 Acc: 0.1637
val Loss: 4.3286 Acc: 0.1892

Epoch 7/199
----------
train Loss: 4.3257 Acc: 0.1706
val Loss: 4.2814 Acc: 0.2275

Epoch 8/199
----------
train Loss: 4.2674 Acc: 0.2382
val Loss: 4.2329 Acc: 0.2490

Epoch 9/199
----------
train Loss: 4.2323 Acc: 0.2275
```

```
val Loss: 4.1847 Acc: 0.2804

Epoch 10/199
----------
train Loss: 4.1655 Acc: 0.2892
val Loss: 4.1332 Acc: 0.2961

Epoch 11/199
----------
train Loss: 4.1309 Acc: 0.3137
val Loss: 4.0911 Acc: 0.3304

Epoch 12/199
----------
train Loss: 4.0929 Acc: 0.3137
val Loss: 4.0433 Acc: 0.3314

Epoch 13/199
----------
train Loss: 4.0325 Acc: 0.3500
val Loss: 3.9948 Acc: 0.3451

Epoch 14/199
----------
train Loss: 3.9970 Acc: 0.3657
val Loss: 3.9483 Acc: 0.3784

Epoch 15/199
----------
train Loss: 3.9415 Acc: 0.4069
val Loss: 3.9049 Acc: 0.3912

Epoch 16/199
----------
train Loss: 3.9304 Acc: 0.3784
val Loss: 3.8600 Acc: 0.3941

Epoch 17/199
----------
train Loss: 3.8653 Acc: 0.4186
val Loss: 3.8199 Acc: 0.3990

Epoch 18/199
----------
train Loss: 3.8501 Acc: 0.4147
val Loss: 3.7777 Acc: 0.4176

Epoch 19/199
```

```
----------
train Loss: 3.8024 Acc: 0.4461
val Loss: 3.7325 Acc: 0.4265

Epoch 20/199
----------
train Loss: 3.7549 Acc: 0.4608
val Loss: 3.6894 Acc: 0.4363

Epoch 21/199
----------
train Loss: 3.6925 Acc: 0.4686
val Loss: 3.6509 Acc: 0.4451

Epoch 22/199
----------
train Loss: 3.6657 Acc: 0.4951
val Loss: 3.6110 Acc: 0.4559

Epoch 23/199
----------
train Loss: 3.6181 Acc: 0.5029
val Loss: 3.5692 Acc: 0.4549

Epoch 24/199
----------
train Loss: 3.5837 Acc: 0.4941
val Loss: 3.5392 Acc: 0.4775

Epoch 25/199
----------
train Loss: 3.5604 Acc: 0.4961
val Loss: 3.4992 Acc: 0.4902

Epoch 26/199
----------
train Loss: 3.5125 Acc: 0.5324
val Loss: 3.4612 Acc: 0.4902

Epoch 27/199
----------
train Loss: 3.5040 Acc: 0.5098
val Loss: 3.4324 Acc: 0.4843

Epoch 28/199
----------
train Loss: 3.4469 Acc: 0.5069
val Loss: 3.3874 Acc: 0.4980
```

```
Epoch 29/199
----------
train Loss: 3.4101 Acc: 0.5176
val Loss: 3.3564 Acc: 0.4931

Epoch 30/199
----------
train Loss: 3.4114 Acc: 0.5108
val Loss: 3.3227 Acc: 0.5157

Epoch 31/199
----------
train Loss: 3.3876 Acc: 0.5069
val Loss: 3.2991 Acc: 0.5196

Epoch 32/199
----------
train Loss: 3.3126 Acc: 0.5392
val Loss: 3.2596 Acc: 0.5127

Epoch 33/199
----------
train Loss: 3.2860 Acc: 0.5382
val Loss: 3.2180 Acc: 0.5284

Epoch 34/199
----------
train Loss: 3.2642 Acc: 0.5412
val Loss: 3.1880 Acc: 0.5392

Epoch 35/199
----------
train Loss: 3.2347 Acc: 0.5343
val Loss: 3.1557 Acc: 0.5422

Epoch 36/199
----------
train Loss: 3.2251 Acc: 0.5490
val Loss: 3.1449 Acc: 0.5353

Epoch 37/199
----------
train Loss: 3.1220 Acc: 0.6069
val Loss: 3.0985 Acc: 0.5520

Epoch 38/199
----------
```

```
train Loss: 3.1496 Acc: 0.5559
val Loss: 3.0827 Acc: 0.5490

Epoch 39/199
----------
train Loss: 3.1216 Acc: 0.5647
val Loss: 3.0431 Acc: 0.5510

Epoch 40/199
----------
train Loss: 3.0757 Acc: 0.5667
val Loss: 3.0082 Acc: 0.5667

Epoch 41/199
----------
train Loss: 3.0488 Acc: 0.5686
val Loss: 2.9830 Acc: 0.5667

Epoch 42/199
----------
train Loss: 3.0274 Acc: 0.5833
val Loss: 2.9686 Acc: 0.5598

Epoch 43/199
----------
train Loss: 2.9691 Acc: 0.6078
val Loss: 2.9369 Acc: 0.5647

Epoch 44/199
----------
train Loss: 2.9635 Acc: 0.5990
val Loss: 2.9153 Acc: 0.5696

Epoch 45/199
----------
train Loss: 2.9425 Acc: 0.5784
val Loss: 2.8851 Acc: 0.5814

Epoch 46/199
----------
train Loss: 2.9621 Acc: 0.5608
val Loss: 2.8687 Acc: 0.5775

Epoch 47/199
----------
train Loss: 2.9179 Acc: 0.5922
val Loss: 2.8387 Acc: 0.5775
```

```
Epoch 48/199
----------
train Loss: 2.8655 Acc: 0.5951
val Loss: 2.8207 Acc: 0.5794

Epoch 49/199
----------
train Loss: 2.8556 Acc: 0.5922
val Loss: 2.7934 Acc: 0.5863

Epoch 50/199
----------
train Loss: 2.8496 Acc: 0.5980
val Loss: 2.7771 Acc: 0.5735

Epoch 51/199
----------
train Loss: 2.7989 Acc: 0.6000
val Loss: 2.7463 Acc: 0.5873

Epoch 52/199
----------
train Loss: 2.8234 Acc: 0.5902
val Loss: 2.7263 Acc: 0.5873

Epoch 53/199
----------
train Loss: 2.7248 Acc: 0.6314
val Loss: 2.7028 Acc: 0.5912

Epoch 54/199
----------
train Loss: 2.7610 Acc: 0.6069
val Loss: 2.6947 Acc: 0.5980

Epoch 55/199
----------
train Loss: 2.7360 Acc: 0.6206
val Loss: 2.6668 Acc: 0.5892

Epoch 56/199
----------
train Loss: 2.6724 Acc: 0.6382
val Loss: 2.6353 Acc: 0.6010

Epoch 57/199
----------
train Loss: 2.6879 Acc: 0.6196
```

```
val Loss: 2.6248 Acc: 0.6069

Epoch 58/199
----------
train Loss: 2.6774 Acc: 0.6196
val Loss: 2.6154 Acc: 0.6078

Epoch 59/199
----------
train Loss: 2.6075 Acc: 0.6520
val Loss: 2.5862 Acc: 0.6029

Epoch 60/199
----------
train Loss: 2.5955 Acc: 0.6343
val Loss: 2.5822 Acc: 0.6069

Epoch 61/199
----------
train Loss: 2.6415 Acc: 0.6167
val Loss: 2.5842 Acc: 0.5990

Epoch 62/199
----------
train Loss: 2.6167 Acc: 0.5961
val Loss: 2.5901 Acc: 0.6029

Epoch 63/199
----------
train Loss: 2.6357 Acc: 0.6147
val Loss: 2.5851 Acc: 0.6069

Epoch 64/199
----------
train Loss: 2.6234 Acc: 0.6265
val Loss: 2.5788 Acc: 0.6049

Epoch 65/199
----------
train Loss: 2.5845 Acc: 0.6245
val Loss: 2.5707 Acc: 0.6098

Epoch 66/199
----------
train Loss: 2.6021 Acc: 0.6314
val Loss: 2.5770 Acc: 0.6049

Epoch 67/199
```

```
----------
train Loss: 2.6113 Acc: 0.6343
val Loss: 2.5774 Acc: 0.5990

Epoch 68/199
----------
train Loss: 2.5920 Acc: 0.6412
val Loss: 2.5794 Acc: 0.6078

Epoch 69/199
----------
train Loss: 2.5828 Acc: 0.6333
val Loss: 2.5682 Acc: 0.6059

Epoch 70/199
----------
train Loss: 2.5960 Acc: 0.6412
val Loss: 2.5652 Acc: 0.6088

Epoch 71/199
----------
train Loss: 2.5993 Acc: 0.6314
val Loss: 2.5640 Acc: 0.6059

Epoch 72/199
----------
train Loss: 2.5688 Acc: 0.6490
val Loss: 2.5533 Acc: 0.6078

Epoch 73/199
----------
train Loss: 2.6109 Acc: 0.6304
val Loss: 2.5553 Acc: 0.6039

Epoch 74/199
----------
train Loss: 2.5718 Acc: 0.6461
val Loss: 2.5499 Acc: 0.6098

Epoch 75/199
----------
train Loss: 2.5837 Acc: 0.6373
val Loss: 2.5605 Acc: 0.6127

Epoch 76/199
----------
train Loss: 2.5964 Acc: 0.6225
val Loss: 2.5557 Acc: 0.6069
```

```
Epoch 77/199
----------
train Loss: 2.5793 Acc: 0.6431
val Loss: 2.5423 Acc: 0.6098

Epoch 78/199
----------
train Loss: 2.5620 Acc: 0.6304
val Loss: 2.5588 Acc: 0.6088

Epoch 79/199
----------
train Loss: 2.5734 Acc: 0.6225
val Loss: 2.5574 Acc: 0.6127

Epoch 80/199
----------
train Loss: 2.5789 Acc: 0.6382
val Loss: 2.5560 Acc: 0.6069

Epoch 81/199
----------
train Loss: 2.5477 Acc: 0.6363
val Loss: 2.5396 Acc: 0.6088

Epoch 82/199
----------
train Loss: 2.5587 Acc: 0.6480
val Loss: 2.5421 Acc: 0.6098

Epoch 83/199
----------
train Loss: 2.5358 Acc: 0.6382
val Loss: 2.5442 Acc: 0.6167

Epoch 84/199
----------
train Loss: 2.5472 Acc: 0.6588
val Loss: 2.5376 Acc: 0.6118

Epoch 85/199
----------
train Loss: 2.5574 Acc: 0.6304
val Loss: 2.5433 Acc: 0.6098

Epoch 86/199
----------
```

```
train Loss: 2.5592 Acc: 0.6412
val Loss: 2.5433 Acc: 0.6118

Epoch 87/199
----------
train Loss: 2.5960 Acc: 0.6471
val Loss: 2.5455 Acc: 0.6078

Epoch 88/199
----------
train Loss: 2.5712 Acc: 0.6373
val Loss: 2.5339 Acc: 0.6167

Epoch 89/199
----------
train Loss: 2.5952 Acc: 0.6353
val Loss: 2.5424 Acc: 0.6167

Epoch 90/199
----------
train Loss: 2.5453 Acc: 0.6657
val Loss: 2.5287 Acc: 0.6108

Epoch 91/199
----------
train Loss: 2.5488 Acc: 0.6529
val Loss: 2.5365 Acc: 0.6108

Epoch 92/199
----------
train Loss: 2.5353 Acc: 0.6451
val Loss: 2.5260 Acc: 0.6127

Epoch 93/199
----------
train Loss: 2.5645 Acc: 0.6402
val Loss: 2.5211 Acc: 0.6069

Epoch 94/199
----------
train Loss: 2.6140 Acc: 0.6275
val Loss: 2.5281 Acc: 0.6137

Epoch 95/199
----------
train Loss: 2.5413 Acc: 0.6510
val Loss: 2.5251 Acc: 0.6118
```

```
Epoch 96/199
----------
train Loss: 2.5486 Acc: 0.6461
val Loss: 2.5243 Acc: 0.6127

Epoch 97/199
----------
train Loss: 2.5314 Acc: 0.6529
val Loss: 2.5063 Acc: 0.6118

Epoch 98/199
----------
train Loss: 2.5422 Acc: 0.6324
val Loss: 2.5164 Acc: 0.6127

Epoch 99/199
----------
train Loss: 2.5901 Acc: 0.6255
val Loss: 2.5229 Acc: 0.6098

Epoch 100/199
----------
train Loss: 2.5350 Acc: 0.6402
val Loss: 2.5142 Acc: 0.6137

Epoch 101/199
----------
train Loss: 2.5611 Acc: 0.6324
val Loss: 2.5156 Acc: 0.6147

Epoch 102/199
----------
train Loss: 2.5229 Acc: 0.6402
val Loss: 2.5199 Acc: 0.6098

Epoch 103/199
----------
train Loss: 2.5334 Acc: 0.6441
val Loss: 2.5146 Acc: 0.6118

Epoch 104/199
----------
train Loss: 2.5157 Acc: 0.6451
val Loss: 2.5057 Acc: 0.6078

Epoch 105/199
----------
train Loss: 2.5200 Acc: 0.6471
```

```
val Loss: 2.4935 Acc: 0.6157

Epoch 106/199
----------
train Loss: 2.5307 Acc: 0.6441
val Loss: 2.5043 Acc: 0.6147

Epoch 107/199
----------
train Loss: 2.5794 Acc: 0.6294
val Loss: 2.5043 Acc: 0.6098

Epoch 108/199
----------
train Loss: 2.5550 Acc: 0.6343
val Loss: 2.5003 Acc: 0.6157

Epoch 109/199
----------
train Loss: 2.5371 Acc: 0.6490
val Loss: 2.4874 Acc: 0.6147

Epoch 110/199
----------
train Loss: 2.5161 Acc: 0.6569
val Loss: 2.4941 Acc: 0.6157

Epoch 111/199
----------
train Loss: 2.5184 Acc: 0.6490
val Loss: 2.4873 Acc: 0.6137

Epoch 112/199
----------
train Loss: 2.4683 Acc: 0.6510
val Loss: 2.4883 Acc: 0.6118

Epoch 113/199
----------
train Loss: 2.4730 Acc: 0.6598
val Loss: 2.4905 Acc: 0.6108

Epoch 114/199
----------
train Loss: 2.5260 Acc: 0.6529
val Loss: 2.4918 Acc: 0.6147

Epoch 115/199
```

```
----------
train Loss: 2.4946 Acc: 0.6441
val Loss: 2.4914 Acc: 0.6176

Epoch 116/199
----------
train Loss: 2.5232 Acc: 0.6539
val Loss: 2.4860 Acc: 0.6137

Epoch 117/199
----------
train Loss: 2.4998 Acc: 0.6343
val Loss: 2.4832 Acc: 0.6167

Epoch 118/199
----------
train Loss: 2.5149 Acc: 0.6343
val Loss: 2.4741 Acc: 0.6118

Epoch 119/199
----------
train Loss: 2.5170 Acc: 0.6363
val Loss: 2.4754 Acc: 0.6176

Epoch 120/199
----------
train Loss: 2.4998 Acc: 0.6480
val Loss: 2.4794 Acc: 0.6147

Epoch 121/199
----------
train Loss: 2.4860 Acc: 0.6382
val Loss: 2.4593 Acc: 0.6216

Epoch 122/199
----------
train Loss: 2.4955 Acc: 0.6637
val Loss: 2.4782 Acc: 0.6167

Epoch 123/199
----------
train Loss: 2.5204 Acc: 0.6373
val Loss: 2.4906 Acc: 0.6157

Epoch 124/199
----------
train Loss: 2.4794 Acc: 0.6559
val Loss: 2.4818 Acc: 0.6196
```

```
Epoch 125/199
----------
train Loss: 2.4573 Acc: 0.6549
val Loss: 2.4597 Acc: 0.6127

Epoch 126/199
----------
train Loss: 2.5148 Acc: 0.6343
val Loss: 2.4764 Acc: 0.6176

Epoch 127/199
----------
train Loss: 2.4868 Acc: 0.6510
val Loss: 2.4739 Acc: 0.6147

Epoch 128/199
----------
train Loss: 2.5134 Acc: 0.6569
val Loss: 2.4834 Acc: 0.6157

Epoch 129/199
----------
train Loss: 2.4829 Acc: 0.6510
val Loss: 2.4723 Acc: 0.6176

Epoch 130/199
----------
train Loss: 2.5179 Acc: 0.6471
val Loss: 2.4819 Acc: 0.6118

Epoch 131/199
----------
train Loss: 2.5262 Acc: 0.6363
val Loss: 2.4868 Acc: 0.6167

Epoch 132/199
----------
train Loss: 2.5145 Acc: 0.6343
val Loss: 2.4831 Acc: 0.6157

Epoch 133/199
----------
train Loss: 2.5227 Acc: 0.6343
val Loss: 2.4722 Acc: 0.6176

Epoch 134/199
----------
```

```
train Loss: 2.4962 Acc: 0.6471
val Loss: 2.4745 Acc: 0.6216

Epoch 135/199
----------
train Loss: 2.5225 Acc: 0.6265
val Loss: 2.4764 Acc: 0.6176

Epoch 136/199
----------
train Loss: 2.4955 Acc: 0.6402
val Loss: 2.4712 Acc: 0.6176

Epoch 137/199
----------
train Loss: 2.4943 Acc: 0.6598
val Loss: 2.4708 Acc: 0.6167

Epoch 138/199
----------
train Loss: 2.5255 Acc: 0.6402
val Loss: 2.4813 Acc: 0.6108

Epoch 139/199
----------
train Loss: 2.4900 Acc: 0.6333
val Loss: 2.4782 Acc: 0.6196

Epoch 140/199
----------
train Loss: 2.5189 Acc: 0.6578
val Loss: 2.4810 Acc: 0.6167

Epoch 141/199
----------
train Loss: 2.4751 Acc: 0.6343
val Loss: 2.4731 Acc: 0.6167

Epoch 142/199
----------
train Loss: 2.5183 Acc: 0.6441
val Loss: 2.4733 Acc: 0.6186

Epoch 143/199
----------
train Loss: 2.5393 Acc: 0.6422
val Loss: 2.4766 Acc: 0.6167
```

```
Epoch 144/199
----------
train Loss: 2.4801 Acc: 0.6588
val Loss: 2.4791 Acc: 0.6157

Epoch 145/199
----------
train Loss: 2.4677 Acc: 0.6520
val Loss: 2.4730 Acc: 0.6147

Epoch 146/199
----------
train Loss: 2.4774 Acc: 0.6520
val Loss: 2.4869 Acc: 0.6196

Epoch 147/199
----------
train Loss: 2.5136 Acc: 0.6333
val Loss: 2.4880 Acc: 0.6137

Epoch 148/199
----------
train Loss: 2.5107 Acc: 0.6422
val Loss: 2.4843 Acc: 0.6157

Epoch 149/199
----------
train Loss: 2.5188 Acc: 0.6402
val Loss: 2.4792 Acc: 0.6147

Epoch 150/199
----------
train Loss: 2.5123 Acc: 0.6324
val Loss: 2.4740 Acc: 0.6147

Epoch 151/199
----------
train Loss: 2.4805 Acc: 0.6382
val Loss: 2.4756 Acc: 0.6147

Epoch 152/199
----------
train Loss: 2.4733 Acc: 0.6627
val Loss: 2.4728 Acc: 0.6147

Epoch 153/199
----------
train Loss: 2.4612 Acc: 0.6706
```

```
val Loss: 2.4686 Acc: 0.6167

Epoch 154/199
----------
train Loss: 2.5240 Acc: 0.6382
val Loss: 2.4729 Acc: 0.6147

Epoch 155/199
----------
train Loss: 2.5261 Acc: 0.6373
val Loss: 2.4778 Acc: 0.6176

Epoch 156/199
----------
train Loss: 2.5219 Acc: 0.6304
val Loss: 2.4863 Acc: 0.6147

Epoch 157/199
----------
train Loss: 2.5301 Acc: 0.6275
val Loss: 2.4774 Acc: 0.6147

Epoch 158/199
----------
train Loss: 2.5498 Acc: 0.6167
val Loss: 2.4758 Acc: 0.6147

Epoch 159/199
----------
train Loss: 2.4908 Acc: 0.6539
val Loss: 2.4771 Acc: 0.6137

Epoch 160/199
----------
train Loss: 2.5146 Acc: 0.6578
val Loss: 2.4804 Acc: 0.6108

Epoch 161/199
----------
train Loss: 2.4773 Acc: 0.6451
val Loss: 2.4672 Acc: 0.6147

Epoch 162/199
----------
train Loss: 2.4950 Acc: 0.6324
val Loss: 2.4785 Acc: 0.6137

Epoch 163/199
```

```
----------
train Loss: 2.5164 Acc: 0.6471
val Loss: 2.4834 Acc: 0.6127

Epoch 164/199
----------
train Loss: 2.5432 Acc: 0.6078
val Loss: 2.4743 Acc: 0.6176

Epoch 165/199
----------
train Loss: 2.5119 Acc: 0.6451
val Loss: 2.4784 Acc: 0.6147

Epoch 166/199
----------
train Loss: 2.5248 Acc: 0.6510
val Loss: 2.4742 Acc: 0.6127

Epoch 167/199
----------
train Loss: 2.5270 Acc: 0.6451
val Loss: 2.4747 Acc: 0.6167

Epoch 168/199
----------
train Loss: 2.5174 Acc: 0.6392
val Loss: 2.4738 Acc: 0.6127

Epoch 169/199
----------
train Loss: 2.4991 Acc: 0.6333
val Loss: 2.4791 Acc: 0.6147

Epoch 170/199
----------
train Loss: 2.5028 Acc: 0.6431
val Loss: 2.4759 Acc: 0.6147

Epoch 171/199
----------
train Loss: 2.4976 Acc: 0.6382
val Loss: 2.4700 Acc: 0.6157

Epoch 172/199
----------
train Loss: 2.5270 Acc: 0.6333
val Loss: 2.4824 Acc: 0.6186
```

```
Epoch 173/199
----------
train Loss: 2.4761 Acc: 0.6500
val Loss: 2.4742 Acc: 0.6196

Epoch 174/199
----------
train Loss: 2.4694 Acc: 0.6461
val Loss: 2.4631 Acc: 0.6167

Epoch 175/199
----------
train Loss: 2.4980 Acc: 0.6422
val Loss: 2.4703 Acc: 0.6127

Epoch 176/199
----------
train Loss: 2.5295 Acc: 0.6569
val Loss: 2.4763 Acc: 0.6137

Epoch 177/199
----------
train Loss: 2.4489 Acc: 0.6627
val Loss: 2.4658 Acc: 0.6167

Epoch 178/199
----------
train Loss: 2.5286 Acc: 0.6294
val Loss: 2.4797 Acc: 0.6157

Epoch 179/199
----------
train Loss: 2.4623 Acc: 0.6402
val Loss: 2.4693 Acc: 0.6157

Epoch 180/199
----------
train Loss: 2.5146 Acc: 0.6363
val Loss: 2.4750 Acc: 0.6167

Epoch 181/199
----------
train Loss: 2.4471 Acc: 0.6775
val Loss: 2.4635 Acc: 0.6147

Epoch 182/199
----------
```

```
train Loss: 2.4750 Acc: 0.6431
val Loss: 2.4611 Acc: 0.6167

Epoch 183/199
----------
train Loss: 2.5209 Acc: 0.6490
val Loss: 2.4838 Acc: 0.6196

Epoch 184/199
----------
train Loss: 2.5077 Acc: 0.6422
val Loss: 2.4661 Acc: 0.6157

Epoch 185/199
----------
train Loss: 2.5045 Acc: 0.6520
val Loss: 2.4730 Acc: 0.6167

Epoch 186/199
----------
train Loss: 2.5086 Acc: 0.6265
val Loss: 2.4725 Acc: 0.6147

Epoch 187/199
----------
train Loss: 2.5202 Acc: 0.6255
val Loss: 2.4629 Acc: 0.6147

Epoch 188/199
----------
train Loss: 2.4849 Acc: 0.6363
val Loss: 2.4678 Acc: 0.6176

Epoch 189/199
----------
train Loss: 2.5059 Acc: 0.6471
val Loss: 2.4723 Acc: 0.6127

Epoch 190/199
----------
train Loss: 2.4624 Acc: 0.6598
val Loss: 2.4677 Acc: 0.6147

Epoch 191/199
----------
train Loss: 2.5146 Acc: 0.6559
val Loss: 2.4734 Acc: 0.6167
```

```
Epoch 192/199
----------
train Loss: 2.5260 Acc: 0.6412
val Loss: 2.4804 Acc: 0.6167

Epoch 193/199
----------
train Loss: 2.4933 Acc: 0.6559
val Loss: 2.4715 Acc: 0.6167

Epoch 194/199
----------
train Loss: 2.4944 Acc: 0.6461
val Loss: 2.4773 Acc: 0.6137

Epoch 195/199
----------
train Loss: 2.4801 Acc: 0.6422
val Loss: 2.4681 Acc: 0.6216

Epoch 196/199
----------
train Loss: 2.5208 Acc: 0.6353
val Loss: 2.4736 Acc: 0.6127

Epoch 197/199
----------
train Loss: 2.4832 Acc: 0.6627
val Loss: 2.4701 Acc: 0.6137

Epoch 198/199
----------
train Loss: 2.5156 Acc: 0.6265
val Loss: 2.4733 Acc: 0.6147

Epoch 199/199
----------
train Loss: 2.5080 Acc: 0.6549
val Loss: 2.4779 Acc: 0.6147

Training complete in 15m 9s
Best val Acc: 0.621569
```

The best validation accuracy when using `ResNet50` as a feature extractor on `vgg-flowers` for the various learning rates are below.

Learning Rate Setting	Best Validation Accuracy
0.001	0.615

Learning Rate Setting	Best Validation Accuracy
0.01	0.624
0.1	0.632
1	0.622

From above, we can see that 0.1 is the best learning rate for using `ResNet50` as a feature extractor for the `vgg-flowers` VDD dataset.

- b: (4 points)

Given the two transfer learning approaches (Fine Tuning and Feature Extraction) and the various learning rate schemes, using Fine Tuning with a fixed learning rate equal to 0.01 resulted in the best validation accuracy on the holdout dataset. This isn't surprising, as the `ResNet50` model was pretrained on `ImageNet`, not the `vgg-flowers` dataset. Allowing for weight updates on the layers prior to the final Dense layers allows for `ResNet50` to learn features and representations specific to the `vgg-flowers` dataset.

1.1.2 Problem 2: ***Weakly and Semi-Supervised Learning for Image Classification*** (20 points)

Q1 (2 points) The difference between weakly and semi-supervised pretraining is related to the labeling mechanism. Weakly supervised learning uses tags or labels, which can often be noisy, innacurate, and imbalanced, to train models on. The noise and imbalance in the tags can reduce performance of models. Semi-supervised learning takes advantage of a teacher-student architecture to select the Top-K images per class with the teacher model and train the student model on the subsampled dataset.

Q2

- a: (2 points)

The model trained using hashtags is robust against noise in the labels. To test this, Mahajan et al randomly injected, 10%, 25%, and 50% label noise into the targets, and observed that there was only a 1 and 2 percent degredation in performance when injection 10 and 25 percent label noise, respectively. This showed that label noise is a miniscule issue to models trained on billions of images.

- b: (2 points)

Hashtags exhibit a Zipfian distribution, and it has been shown that resampling Zipfian distributions reduces the impact of the head of the word distribution when pretraining for transfer learning.

Q3

- a: (4 points)

There are two models, the teacher and the student, for the purposes of selecting the best data to train on per class (teacher) and training on the best data (student). The student model leverages the teacher in the sense that the output of the teacher model is a subsample of images the teacher has the most confidence in per class, and this is the data the student is trained on. This can be thought of as a distillation technique because the more inaccurate or noisy images that the teacher has less confidence in are not fed to the student for learning; only the most "pure" training images are left after the teacher distillation process.

- b: (4 points)

K and P are hyperparameters for selecting which data to feed from the teacher to the student. K represents the Top-K images per class labelled by the teacher on the unlabelled dataset that will be fed to the student, and P represents the Top-P softmax probabilities per image that are kept for ranking the Top-K images per class. We use P > 1 to account for uncertainty in selecting only the most likely label assigned from the teacher, as well as to capture some of the less represented or infrequent labels that were learned by the teacher. A higher value for P leads to a more robust dataset to select the Top-K images per class from for the student.

- c: (4 points)

A new labelled dataset is created from the unlabbeled data by using the teacher to generate the Top-P softmax probabilities per image. From here, we rank and select the Top-K images per class from the Top-P softmax probabilities per image. It is possible for an image in the unlabelled dataset to belong to multiple classes with this approach, and Yalzin et al simply allow for an image with multiple Top-K class labels to be replicated in the dataset that is fed to the student.

- d: (2 points)

The accuracy of the student model demonstrates an inverted parabola shape with increasing K hyperparameter value because at a certain point, the original noise from the weakly labelled dataset is reintroduced with larger K values selected by the teacher model.

1.1.3 Problem 3: *PALEO, FLOPs, Platform Percent of Peak (PPP)* (20 points)

Q1 (4 points) Achieving peak FLOPs from GPU hardware is a difficult proposition in real systems because you typically need to develop customized libraries that take advantage of intimate knowledge of the underlying GPU hardware, which most people outside of the developers of the GPU hardware possess. Platform Percent of Peak (PPP) helps capture this inefficiency by taking the average relative inefficiency of the platform compared to peak FLOPS.

Q2 (6 points) The difference between VGG16 and VGG19 is the additional convolutional layer in the third and fourth convolution block of VGG19.

The two additional convolution layers are 3x3x512 layers, where the input size to the first is 28x28x512 and the input size to the second is 14x14x512.

From this, we can calculate the additional number of multiplications as follows:

$$2x(3x3x512)x512 = 4718592$$

Adding this number to the number of FLOPs from Table 3 in Lu et al results in roughly 15365M FLOPs for the CONVs in VGG19.

Adding the above to the unchanged number of FLOPs for POOL, ReLU, and FC layers in VGG19 results in a total of 15508M FLOPs for VGG19.

Q3 (4 points) The measured time and sum of layerwise timings for forward pass did not match on GPUs for AlexNet, VGG16, GoogleNet, and ResNet50 due to communication and measurement overhead differences between TK1 and TX1. The authors keep GPUs iteratively running the matrix multiplication in a way that GPU cores can continuously perform multiply-add operations without synchronization, before recording the end time. After this, the measurement overhead is amortized over all the iterations, giving accurate timing estimates. When the number of iterations is large enough, the overhead is negligible.

Q4 (6 points) The peak double-precision floating point performance of a NVIDIA Tesla K80 is 1.87 TFLOPS.

The FLOPs for VGG16, GoogleNet, and ResNet50 are in the table below.

Model Architecture	FLOPS
VGG16	15503M
GoogleNet	1606M
ResNet50	3922M

The computation time formula is calculated as the FLOP counts of the operation divided by the FLOPS of the device.

Using this formula, the computation time for the three models above is:

Model Architecture	FLOPS	K80 TFLOPS	Computation Time
VGG16	15503M	1.87	8 ms
GoogleNet	1606M	1.87	0.8 ms
ResNet50	3922M	1.87	2 ms

Given these computation times, we can divide by 1 second to get throughput, or the number of images processed per second.

Model Architecture	FLOPS	K80 TFLOPS	Computation Time	Throughput
VGG16	15503M	1.87	8 ms	125
GoogleNet	1606M	1.87	0.8 ms	1250
ResNet50	3922M	1.87	2 ms	500

1.1.4 Problem 4: *Optimus, Learning and Resource Models, Performance-Cost Trade-offs* (30 points)

Q1: (15 points) I worked on this question with Shawn Pachgade, Roberto Ponce, Nick Christman, and Wang Yao.

I used this code repository for building out my `ResNet20` implementation.

So far, I've testing on the following GPUs (this is for my own record keeping):

- [X] K80

- [X] P100
- [X] V100

Import the necessary packages and set up the ResNet20 module.

[2]:
```
import copy

import torch
from torch.autograd import Variable
import torch.nn as nn
import torch.nn.functional as F
import torch.nn.init as init
import torch.optim as optim
from torch.optim import lr_scheduler
import torchvision
import torchvision.transforms as transforms

import pandas as pd

import time
```

[2]:
```
def _weights_init(m):
    classname = m.__class__.__name__
    #print(classname)
    if isinstance(m, nn.Linear) or isinstance(m, nn.Conv2d):
        init.kaiming_normal_(m.weight)

class LambdaLayer(nn.Module):
    def __init__(self, lambd):
        super(LambdaLayer, self).__init__()
        self.lambd = lambd

    def forward(self, x):
        return self.lambd(x)

class BasicBlock(nn.Module):
    expansion = 1

    def __init__(self, in_planes, planes, stride=1, option='A'):
        super(BasicBlock, self).__init__()
        self.conv1 = nn.Conv2d(in_planes, planes, kernel_size=3, stride=stride,␣
 ↪padding=1, bias=False)
        self.bn1 = nn.BatchNorm2d(planes)
        self.conv2 = nn.Conv2d(planes, planes, kernel_size=3, stride=1,␣
 ↪padding=1, bias=False)
        self.bn2 = nn.BatchNorm2d(planes)

        self.shortcut = nn.Sequential()
```

```
        if stride != 1 or in_planes != planes:
            if option == 'A':
                """
                For CIFAR10 ResNet paper uses option A.
                """
                self.shortcut = LambdaLayer(lambda x:
                                            F.pad(x[:, :, ::2, ::2], (0, 0, 0,␣
↪0, planes//4, planes//4), "constant", 0))
            elif option == 'B':
                self.shortcut = nn.Sequential(
                     nn.Conv2d(in_planes, self.expansion * planes,␣
↪kernel_size=1, stride=stride, bias=False),
                     nn.BatchNorm2d(self.expansion * planes)
                )

    def forward(self, x):
        out = F.relu(self.bn1(self.conv1(x)))
        out = self.bn2(self.conv2(out))
        out += self.shortcut(x)
        out = F.relu(out)
        return out

class ResNet(nn.Module):
    def __init__(self, block, num_blocks, num_classes=10):
        super(ResNet, self).__init__()
        self.in_planes = 16

        self.conv1 = nn.Conv2d(3, 16, kernel_size=3, stride=1, padding=1,␣
↪bias=False)
        self.bn1 = nn.BatchNorm2d(16)
        self.layer1 = self._make_layer(block, 16, num_blocks[0], stride=1)
        self.layer2 = self._make_layer(block, 32, num_blocks[1], stride=2)
        self.layer3 = self._make_layer(block, 64, num_blocks[2], stride=2)
        self.linear = nn.Linear(64, num_classes)

        self.apply(_weights_init)

    def _make_layer(self, block, planes, num_blocks, stride):
        strides = [stride] + [1]*(num_blocks-1)
        layers = []
        for stride in strides:
            layers.append(block(self.in_planes, planes, stride))
            self.in_planes = planes * block.expansion

        return nn.Sequential(*layers)

    def forward(self, x):
```

```
        out = F.relu(self.bn1(self.conv1(x)))
        out = self.layer1(out)
        out = self.layer2(out)
        out = self.layer3(out)
        out = F.avg_pool2d(out, out.size()[3])
        out = out.view(out.size(0), -1)
        out = self.linear(out)
        return out

def resnet20():
    return ResNet(BasicBlock, [3, 3, 3])
```

Create a function to train the model

```
[3]: def train_model(model, resnet_layers, hardware, dataloaders, criterion,␣
     ↪optimizer, scheduler, num_epochs=350):
         since = time.time()

         best_model_wts = copy.deepcopy(model.state_dict())
         best_acc = 0.0
         metrics = []
         training_step = 0

         for epoch in range(num_epochs):
             print(f'Epoch {epoch}/{num_epochs - 1}')
             print('-' * 10)

             for phase in ['train', 'val']:
                 if phase == 'train':
                     model.train()
                 else:
                     model.eval()

                 epoch_phase_start_time = time.time()
                 running_loss = 0.0
                 running_corrects = 0

                 for inputs, labels in dataloaders[phase]:
                     step_start_time = time.time()

                     inputs = inputs.to(device)
                     labels = labels.to(device)

                     optimizer.zero_grad()

                     # forward
                     # track history if only in train
```

```python
                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()

                        metrics.append({
                            'resnet_layers': resnet_layers,
                            'hardware': hardware,
                            'epoch': epoch,
                            'training_step': training_step,
                            'training_step_loss': loss.item(),
                            'training_step_time': time.time() - step_start_time
                        })

                        training_step += 1

                # 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]
            epoch_phase_end_time = time.time()

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

            # deep copy the model
            if phase == 'val' and epoch_acc > best_acc:
                best_acc = epoch_acc.item()
                best_model_wts = copy.deepcopy(model.state_dict())

        print()

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

    # load best model weights
    model.load_state_dict(best_model_wts)
```

```
    # set up return structure
    return_df = pd.DataFrame(data=metrics)

    return model, return_df
```

Load CIFAR-10 data.

[4]:
```
BATCHSIZE = 128
DATA_DIR = '~/data/cifar10'

data_transforms = {
    'train': transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
    ]),
    'val': transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
    ]),
}


train_set = torchvision.datasets.CIFAR10(root=DATA_DIR, train=True,
                                        download=True,␣
 ↪transform=data_transforms['train'])

val_set = torchvision.datasets.CIFAR10(root=DATA_DIR, train=False,
                                       download=True,␣
 ↪transform=data_transforms['val'])

image_datasets = {'train': train_set, 'val': val_set}

dataloaders = {x: torch.utils.data.DataLoader(image_datasets[x],␣
 ↪batch_size=BATCHSIZE,
                                              shuffle=True, num_workers=4)
               for x in ['train', 'val']}

dataset_sizes = {x: len(image_datasets[x]) for x in ['train', 'val']}
class_names = image_datasets['train'].classes

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

print(f"Dataset sizes: {dataset_sizes}")
print(f"Class names: {class_names}")
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to
/home/jupyter/data/cifar10/cifar-10-python.tar.gz
```

```
HBox(children=(HTML(value=''), FloatProgress(value=1.0, bar_style='info', layout=Layout(width=
```

```
Extracting /home/jupyter/data/cifar10/cifar-10-python.tar.gz to
/home/jupyter/data/cifar10
Files already downloaded and verified
Dataset sizes: {'train': 50000, 'val': 10000}
Class names: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
'horse', 'ship', 'truck']
```

Set up learning criteria and train.

[5]:
```
# Fetch model
model = resnet20()

# Set up criterion
criterion = nn.CrossEntropyLoss()

# Observe that all parameters are being optimized
optimizer = optim.SGD(model.parameters(), lr=0.1, momentum=0.9)

# Decay LR by a factor of 0.1 every 60 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer, step_size=100, gamma=0.1)

# Send model to device
model = model.to(device)

# Train
num_epochs = 350
resnet_layers = 20
hardware = 'K80'
model, results_df = train_model(model, resnet_layers, hardware, dataloaders,
                                criterion, optimizer, exp_lr_scheduler,␣
 ↪num_epochs)
```

```
Epoch 0/349
----------

train Loss: 1.473 Acc: 0.4655
val Loss: 1.2322 Acc: 0.562

Epoch 1/349
----------
train Loss: 0.9335 Acc: 0.6677
val Loss: 0.9706 Acc: 0.6647

Epoch 2/349
----------
train Loss: 0.7192 Acc: 0.7495
```

```
val Loss: 0.7323 Acc: 0.7429

Epoch 3/349
----------
train Loss: 0.6051 Acc: 0.789
val Loss: 0.6667 Acc: 0.7655

Epoch 4/349
----------
train Loss: 0.5261 Acc: 0.8163
val Loss: 0.7175 Acc: 0.7629

Epoch 5/349
----------
train Loss: 0.4531 Acc: 0.8407
val Loss: 0.6559 Acc: 0.7821

Epoch 6/349
----------
train Loss: 0.396 Acc: 0.8598
val Loss: 0.6327 Acc: 0.7971

Epoch 7/349
----------
train Loss: 0.347 Acc: 0.8766
val Loss: 0.6986 Acc: 0.7891

Epoch 8/349
----------
train Loss: 0.311 Acc: 0.8891
val Loss: 0.7333 Acc: 0.7777

Epoch 9/349
----------
train Loss: 0.2625 Acc: 0.9075
val Loss: 0.6963 Acc: 0.7885

Epoch 10/349
----------
train Loss: 0.2284 Acc: 0.9186
val Loss: 0.7719 Acc: 0.7766

Epoch 11/349
----------
train Loss: 0.2075 Acc: 0.9257
val Loss: 0.7753 Acc: 0.7935

Epoch 12/349
```

```
----------
train Loss: 0.1811 Acc: 0.935
val Loss: 0.7936 Acc: 0.786

Epoch 13/349
----------
train Loss: 0.1557 Acc: 0.9448
val Loss: 0.8015 Acc: 0.7967

Epoch 14/349
----------
train Loss: 0.1347 Acc: 0.9519
val Loss: 0.9735 Acc: 0.7693

Epoch 15/349
----------
train Loss: 0.1167 Acc: 0.9588
val Loss: 0.8532 Acc: 0.7873

Epoch 16/349
----------
train Loss: 0.109 Acc: 0.961
val Loss: 0.8809 Acc: 0.7977

Epoch 17/349
----------
train Loss: 0.094 Acc: 0.9662
val Loss: 1.0137 Acc: 0.7793

Epoch 18/349
----------
train Loss: 0.0833 Acc: 0.9703
val Loss: 1.239 Acc: 0.7669

Epoch 19/349
----------
train Loss: 0.0781 Acc: 0.9724
val Loss: 1.1143 Acc: 0.7876

Epoch 20/349
----------
train Loss: 0.0655 Acc: 0.9777
val Loss: 0.9729 Acc: 0.8037

Epoch 21/349
----------
train Loss: 0.061 Acc: 0.9783
val Loss: 1.0509 Acc: 0.7917
```

```
Epoch 22/349
----------
train Loss: 0.0619 Acc: 0.9784
val Loss: 1.0793 Acc: 0.7982

Epoch 23/349
----------
train Loss: 0.0527 Acc: 0.9822
val Loss: 1.0762 Acc: 0.8016

Epoch 24/349
----------
train Loss: 0.0463 Acc: 0.9837
val Loss: 1.0887 Acc: 0.8028

Epoch 25/349
----------
train Loss: 0.0361 Acc: 0.9872
val Loss: 1.2955 Acc: 0.7805

Epoch 26/349
----------
train Loss: 0.0408 Acc: 0.9865
val Loss: 1.1367 Acc: 0.7965

Epoch 27/349
----------
train Loss: 0.0373 Acc: 0.9867
val Loss: 1.137 Acc: 0.8012

Epoch 28/349
----------
train Loss: 0.0301 Acc: 0.9897
val Loss: 1.2268 Acc: 0.7968

Epoch 29/349
----------
train Loss: 0.0359 Acc: 0.9873
val Loss: 1.1602 Acc: 0.8083

Epoch 30/349
----------
train Loss: 0.0448 Acc: 0.9848
val Loss: 1.1677 Acc: 0.7998

Epoch 31/349
----------
```

```
train Loss: 0.0401 Acc: 0.986
val Loss: 1.098 Acc: 0.8057

Epoch 32/349
----------
train Loss: 0.0231 Acc: 0.992
val Loss: 1.1118 Acc: 0.8103

Epoch 33/349
----------
train Loss: 0.0193 Acc: 0.9934
val Loss: 1.1394 Acc: 0.81

Epoch 34/349
----------
train Loss: 0.0141 Acc: 0.9952
val Loss: 1.1895 Acc: 0.8148

Epoch 35/349
----------
train Loss: 0.0152 Acc: 0.9951
val Loss: 1.4298 Acc: 0.7978

Epoch 36/349
----------
train Loss: 0.0221 Acc: 0.9918
val Loss: 1.371 Acc: 0.7949

Epoch 37/349
----------
train Loss: 0.0174 Acc: 0.994
val Loss: 1.2277 Acc: 0.8147

Epoch 38/349
----------
train Loss: 0.0221 Acc: 0.9926
val Loss: 1.2235 Acc: 0.8079

Epoch 39/349
----------
train Loss: 0.0229 Acc: 0.9919
val Loss: 1.2149 Acc: 0.8117

Epoch 40/349
----------
train Loss: 0.021 Acc: 0.9925
val Loss: 1.2686 Acc: 0.807
```

```
Epoch 41/349
----------
train Loss: 0.0207 Acc: 0.9926
val Loss: 1.2501 Acc: 0.817

Epoch 42/349
----------
train Loss: 0.0171 Acc: 0.994
val Loss: 1.5065 Acc: 0.7885

Epoch 43/349
----------
train Loss: 0.0215 Acc: 0.9925
val Loss: 1.2041 Acc: 0.8117

Epoch 44/349
----------
train Loss: 0.0168 Acc: 0.9944
val Loss: 1.3165 Acc: 0.804

Epoch 45/349
----------
train Loss: 0.0192 Acc: 0.9933
val Loss: 1.2706 Acc: 0.8101

Epoch 46/349
----------
train Loss: 0.0127 Acc: 0.9955
val Loss: 1.2671 Acc: 0.813

Epoch 47/349
----------
train Loss: 0.009 Acc: 0.9971
val Loss: 1.2593 Acc: 0.8159

Epoch 48/349
----------
train Loss: 0.008 Acc: 0.9971
val Loss: 1.2555 Acc: 0.817

Epoch 49/349
----------
train Loss: 0.0054 Acc: 0.9982
val Loss: 1.2806 Acc: 0.8148

Epoch 50/349
----------
train Loss: 0.0046 Acc: 0.9987
```

```
val Loss: 1.2735 Acc: 0.8197

Epoch 51/349
----------
train Loss: 0.0041 Acc: 0.9987
val Loss: 1.2752 Acc: 0.8208

Epoch 52/349
----------
train Loss: 0.0061 Acc: 0.998
val Loss: 1.295 Acc: 0.824

Epoch 53/349
----------
train Loss: 0.0019 Acc: 0.9996
val Loss: 1.3282 Acc: 0.8238

Epoch 54/349
----------
train Loss: 0.0033 Acc: 0.999
val Loss: 1.2936 Acc: 0.8193

Epoch 55/349
----------
train Loss: 0.0017 Acc: 0.9996
val Loss: 1.307 Acc: 0.8223

Epoch 56/349
----------
train Loss: 0.0008 Acc: 0.9999
val Loss: 1.2929 Acc: 0.8263

Epoch 57/349
----------
train Loss: 0.0006 Acc: 0.9998
val Loss: 1.2995 Acc: 0.8246

Epoch 58/349
----------
train Loss: 0.0005 Acc: 0.9999
val Loss: 1.2982 Acc: 0.8243

Epoch 59/349
----------
train Loss: 0.0006 Acc: 0.9998
val Loss: 1.3821 Acc: 0.8206

Epoch 60/349
```

```
----------
train Loss: 0.0007 Acc: 0.9999
val Loss: 1.3258 Acc: 0.8256

Epoch 61/349
----------
train Loss: 0.0003 Acc: 1.0
val Loss: 1.3314 Acc: 0.8261

Epoch 62/349
----------
train Loss: 0.0002 Acc: 1.0
val Loss: 1.3048 Acc: 0.8271

Epoch 63/349
----------
train Loss: 0.0002 Acc: 1.0
val Loss: 1.3083 Acc: 0.8259

Epoch 64/349
----------
train Loss: 0.0002 Acc: 1.0
val Loss: 1.3134 Acc: 0.8266

Epoch 65/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3136 Acc: 0.8274

Epoch 66/349
----------
train Loss: 0.0002 Acc: 1.0
val Loss: 1.3265 Acc: 0.8262

Epoch 67/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3388 Acc: 0.8268

Epoch 68/349
----------
train Loss: 0.0002 Acc: 1.0
val Loss: 1.3271 Acc: 0.8263

Epoch 69/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3436 Acc: 0.8288
```

Epoch 70/349
----
train Loss: 0.0002 Acc: 0.9999
val Loss: 1.3269 Acc: 0.828

Epoch 71/349
----
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3397 Acc: 0.8271

Epoch 72/349
----
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3308 Acc: 0.8277

Epoch 73/349
----
train Loss: 0.0002 Acc: 1.0
val Loss: 1.3418 Acc: 0.8285

Epoch 74/349
----
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3453 Acc: 0.829

Epoch 75/349
----
train Loss: 0.0001 Acc: 1.0
val Loss: 1.346 Acc: 0.8286

Epoch 76/349
----
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3432 Acc: 0.8285

Epoch 77/349
----
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3453 Acc: 0.8278

Epoch 78/349
----
train Loss: 0.0 Acc: 1.0
val Loss: 1.3431 Acc: 0.827

Epoch 79/349

```
train Loss: 0.0 Acc: 1.0
val Loss: 1.3468 Acc: 0.8285

Epoch 80/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3634 Acc: 0.8264

Epoch 81/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3513 Acc: 0.8293

Epoch 82/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3513 Acc: 0.8303

Epoch 83/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3447 Acc: 0.8297

Epoch 84/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3634 Acc: 0.8298

Epoch 85/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3513 Acc: 0.8304

Epoch 86/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.36 Acc: 0.8295

Epoch 87/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3532 Acc: 0.8274

Epoch 88/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3651 Acc: 0.8294
```

```
Epoch 89/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3617 Acc: 0.829

Epoch 90/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3557 Acc: 0.8302

Epoch 91/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3539 Acc: 0.8301

Epoch 92/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.366 Acc: 0.8303

Epoch 93/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.373 Acc: 0.8286

Epoch 94/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3692 Acc: 0.831

Epoch 95/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3657 Acc: 0.8312

Epoch 96/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3726 Acc: 0.8307

Epoch 97/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3762 Acc: 0.8303

Epoch 98/349
----------
train Loss: 0.0 Acc: 1.0
```

```
val Loss: 1.3706 Acc: 0.8313

Epoch 99/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3651 Acc: 0.8304

Epoch 100/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3684 Acc: 0.831

Epoch 101/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3731 Acc: 0.8304

Epoch 102/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3746 Acc: 0.8311

Epoch 103/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3636 Acc: 0.8304

Epoch 104/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3632 Acc: 0.8296

Epoch 105/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3757 Acc: 0.8311

Epoch 106/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3722 Acc: 0.8305

Epoch 107/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3667 Acc: 0.8314

Epoch 108/349
```

```
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3727 Acc: 0.83

Epoch 109/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3668 Acc: 0.8301

Epoch 110/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3695 Acc: 0.8308

Epoch 111/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3714 Acc: 0.8305

Epoch 112/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3701 Acc: 0.8293

Epoch 113/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3706 Acc: 0.8319

Epoch 114/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3819 Acc: 0.831

Epoch 115/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3665 Acc: 0.8304

Epoch 116/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3753 Acc: 0.8303

Epoch 117/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3654 Acc: 0.8309
```

```
Epoch 118/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3743 Acc: 0.8314

Epoch 119/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3644 Acc: 0.8303

Epoch 120/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3714 Acc: 0.8309

Epoch 121/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3787 Acc: 0.8311

Epoch 122/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3688 Acc: 0.8299

Epoch 123/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3702 Acc: 0.8307

Epoch 124/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3734 Acc: 0.8304

Epoch 125/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3716 Acc: 0.8302

Epoch 126/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.374 Acc: 0.8315

Epoch 127/349
----------
```

```
train Loss: 0.0 Acc: 1.0
val Loss: 1.3789 Acc: 0.8313

Epoch 128/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3743 Acc: 0.831

Epoch 129/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3655 Acc: 0.8299

Epoch 130/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3845 Acc: 0.8303

Epoch 131/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3741 Acc: 0.8308

Epoch 132/349
----------
train Loss: 0.0001 Acc: 1.0
val Loss: 1.3662 Acc: 0.83

Epoch 133/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3752 Acc: 0.8314

Epoch 134/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3707 Acc: 0.8319

Epoch 135/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3693 Acc: 0.8316

Epoch 136/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3722 Acc: 0.8298
```

```
Epoch 137/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3664 Acc: 0.8302

Epoch 138/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.377 Acc: 0.8304

Epoch 139/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3722 Acc: 0.8296

Epoch 140/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3649 Acc: 0.8299

Epoch 141/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.384 Acc: 0.8296

Epoch 142/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3665 Acc: 0.8315

Epoch 143/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3712 Acc: 0.8292

Epoch 144/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3761 Acc: 0.8301

Epoch 145/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3709 Acc: 0.8317

Epoch 146/349
----------
train Loss: 0.0 Acc: 1.0
```

```
val Loss: 1.3805 Acc: 0.8304

Epoch 147/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.37 Acc: 0.8315

Epoch 148/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3714 Acc: 0.8311

Epoch 149/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3738 Acc: 0.8315

Epoch 150/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3738 Acc: 0.8314

Epoch 151/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3626 Acc: 0.8299

Epoch 152/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3816 Acc: 0.8303

Epoch 153/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3691 Acc: 0.8298

Epoch 154/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3719 Acc: 0.831

Epoch 155/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3768 Acc: 0.8314

Epoch 156/349
```

```
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.381 Acc: 0.8303

Epoch 157/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3741 Acc: 0.8306

Epoch 158/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3777 Acc: 0.8304

Epoch 159/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3657 Acc: 0.8318

Epoch 160/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3751 Acc: 0.8304

Epoch 161/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3702 Acc: 0.8305

Epoch 162/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3756 Acc: 0.8313

Epoch 163/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3764 Acc: 0.8299

Epoch 164/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.368 Acc: 0.8313

Epoch 165/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3754 Acc: 0.8301
```

```
Epoch 166/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3744 Acc: 0.8295

Epoch 167/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3815 Acc: 0.831

Epoch 168/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3729 Acc: 0.8306

Epoch 169/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3684 Acc: 0.8309

Epoch 170/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3709 Acc: 0.8302

Epoch 171/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3707 Acc: 0.8302

Epoch 172/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3721 Acc: 0.8307

Epoch 173/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3846 Acc: 0.8299

Epoch 174/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3876 Acc: 0.8309

Epoch 175/349
----------
```

```
train Loss: 0.0 Acc: 1.0
val Loss: 1.3753 Acc: 0.8312

Epoch 176/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3772 Acc: 0.8315

Epoch 177/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3678 Acc: 0.831

Epoch 178/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3775 Acc: 0.8312

Epoch 179/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3812 Acc: 0.831

Epoch 180/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.37 Acc: 0.8301

Epoch 181/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3783 Acc: 0.8312

Epoch 182/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3807 Acc: 0.8294

Epoch 183/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3847 Acc: 0.8299

Epoch 184/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3693 Acc: 0.8299
```

```
Epoch 185/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3659 Acc: 0.8311

Epoch 186/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3786 Acc: 0.8311

Epoch 187/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3765 Acc: 0.8308

Epoch 188/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3723 Acc: 0.8305

Epoch 189/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3783 Acc: 0.8309

Epoch 190/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3769 Acc: 0.8291

Epoch 191/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3745 Acc: 0.8301

Epoch 192/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.37 Acc: 0.8311

Epoch 193/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3772 Acc: 0.8308

Epoch 194/349
----------
train Loss: 0.0 Acc: 1.0
```

```
val Loss: 1.3765 Acc: 0.831

Epoch 195/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3817 Acc: 0.8307

Epoch 196/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3818 Acc: 0.8305

Epoch 197/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3756 Acc: 0.8309

Epoch 198/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3736 Acc: 0.8295

Epoch 199/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3766 Acc: 0.8304

Epoch 200/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.378 Acc: 0.8307

Epoch 201/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3805 Acc: 0.8306

Epoch 202/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3667 Acc: 0.8306

Epoch 203/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.381 Acc: 0.8294

Epoch 204/349
```

```
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3735 Acc: 0.8306

Epoch 205/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3909 Acc: 0.8296

Epoch 206/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3775 Acc: 0.8298

Epoch 207/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3793 Acc: 0.8302

Epoch 208/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3771 Acc: 0.8301

Epoch 209/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.373 Acc: 0.8299

Epoch 210/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3785 Acc: 0.8309

Epoch 211/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3804 Acc: 0.8303

Epoch 212/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.374 Acc: 0.8321

Epoch 213/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3709 Acc: 0.8309
```

```
Epoch 214/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3815 Acc: 0.8307

Epoch 215/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3714 Acc: 0.8309

Epoch 216/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.383 Acc: 0.832

Epoch 217/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3757 Acc: 0.8306

Epoch 218/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3799 Acc: 0.829

Epoch 219/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3822 Acc: 0.8309

Epoch 220/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3764 Acc: 0.829

Epoch 221/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3659 Acc: 0.8287

Epoch 222/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3745 Acc: 0.8296

Epoch 223/349
----------
```

```
train Loss: 0.0 Acc: 1.0
val Loss: 1.3671 Acc: 0.8302

Epoch 224/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3735 Acc: 0.8314

Epoch 225/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3737 Acc: 0.8308

Epoch 226/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3673 Acc: 0.8301

Epoch 227/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3758 Acc: 0.8311

Epoch 228/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.379 Acc: 0.8288

Epoch 229/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3813 Acc: 0.8303

Epoch 230/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3816 Acc: 0.8306

Epoch 231/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3842 Acc: 0.8302

Epoch 232/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3651 Acc: 0.8307
```

```
Epoch 233/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3776 Acc: 0.8308

Epoch 234/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3851 Acc: 0.8309

Epoch 235/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3754 Acc: 0.831

Epoch 236/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3724 Acc: 0.8305

Epoch 237/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3787 Acc: 0.8306

Epoch 238/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3753 Acc: 0.8316

Epoch 239/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3771 Acc: 0.8322

Epoch 240/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3703 Acc: 0.8304

Epoch 241/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3804 Acc: 0.8318

Epoch 242/349
----------
train Loss: 0.0 Acc: 1.0
```

```
val Loss: 1.3765 Acc: 0.8299

Epoch 243/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3745 Acc: 0.8314

Epoch 244/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3808 Acc: 0.8303

Epoch 245/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3764 Acc: 0.8315

Epoch 246/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3912 Acc: 0.8296

Epoch 247/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3801 Acc: 0.8305

Epoch 248/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3793 Acc: 0.83

Epoch 249/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3737 Acc: 0.829

Epoch 250/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.388 Acc: 0.8312

Epoch 251/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3834 Acc: 0.8297

Epoch 252/349
```

```
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3707 Acc: 0.8304

Epoch 253/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.376 Acc: 0.831

Epoch 254/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3768 Acc: 0.8303

Epoch 255/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3779 Acc: 0.831

Epoch 256/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3731 Acc: 0.8307

Epoch 257/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3689 Acc: 0.8308

Epoch 258/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3774 Acc: 0.8312

Epoch 259/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3689 Acc: 0.8315

Epoch 260/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3701 Acc: 0.8306

Epoch 261/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3741 Acc: 0.831
```

```
Epoch 262/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3712 Acc: 0.8301

Epoch 263/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3727 Acc: 0.8301

Epoch 264/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3673 Acc: 0.831

Epoch 265/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3726 Acc: 0.8309

Epoch 266/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3816 Acc: 0.8307

Epoch 267/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3788 Acc: 0.8316

Epoch 268/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3901 Acc: 0.8314

Epoch 269/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3775 Acc: 0.8303

Epoch 270/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3791 Acc: 0.8302

Epoch 271/349
----------
```

```
train Loss: 0.0 Acc: 1.0
val Loss: 1.3772 Acc: 0.8308

Epoch 272/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3667 Acc: 0.8312

Epoch 273/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3688 Acc: 0.8309

Epoch 274/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3833 Acc: 0.8308

Epoch 275/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3792 Acc: 0.8317

Epoch 276/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.382 Acc: 0.8312

Epoch 277/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3829 Acc: 0.8294

Epoch 278/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3678 Acc: 0.8316

Epoch 279/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3811 Acc: 0.8302

Epoch 280/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3785 Acc: 0.83
```

```
Epoch 281/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3841 Acc: 0.8296

Epoch 282/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3712 Acc: 0.8313

Epoch 283/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3748 Acc: 0.8297

Epoch 284/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3767 Acc: 0.8316

Epoch 285/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3802 Acc: 0.8309

Epoch 286/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3691 Acc: 0.8313

Epoch 287/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3765 Acc: 0.8314

Epoch 288/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3795 Acc: 0.8304

Epoch 289/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3799 Acc: 0.8299

Epoch 290/349
----------
train Loss: 0.0 Acc: 1.0
```

```
val Loss: 1.3745 Acc: 0.8322

Epoch 291/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3835 Acc: 0.8311

Epoch 292/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3729 Acc: 0.8304

Epoch 293/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3739 Acc: 0.8312

Epoch 294/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3733 Acc: 0.831

Epoch 295/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3927 Acc: 0.8304

Epoch 296/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.378 Acc: 0.8295

Epoch 297/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3707 Acc: 0.83

Epoch 298/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3804 Acc: 0.831

Epoch 299/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3683 Acc: 0.8307

Epoch 300/349
```

```
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3757 Acc: 0.8298

Epoch 301/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3744 Acc: 0.8311

Epoch 302/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3768 Acc: 0.831

Epoch 303/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3806 Acc: 0.831

Epoch 304/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3673 Acc: 0.8297

Epoch 305/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3763 Acc: 0.8292

Epoch 306/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3692 Acc: 0.8307

Epoch 307/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3849 Acc: 0.8312

Epoch 308/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3796 Acc: 0.8299

Epoch 309/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3692 Acc: 0.8309
```

```
Epoch 310/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.371 Acc: 0.8288

Epoch 311/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3838 Acc: 0.8304

Epoch 312/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3696 Acc: 0.831

Epoch 313/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.375 Acc: 0.831

Epoch 314/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3829 Acc: 0.8304

Epoch 315/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3742 Acc: 0.8316

Epoch 316/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3782 Acc: 0.8307

Epoch 317/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.377 Acc: 0.8298

Epoch 318/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3771 Acc: 0.8306

Epoch 319/349
----------
```

```
train Loss: 0.0 Acc: 1.0
val Loss: 1.375 Acc: 0.8308

Epoch 320/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3791 Acc: 0.8299

Epoch 321/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.375 Acc: 0.831

Epoch 322/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3799 Acc: 0.8298

Epoch 323/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3827 Acc: 0.83

Epoch 324/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3678 Acc: 0.8299

Epoch 325/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3855 Acc: 0.8306

Epoch 326/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3752 Acc: 0.8307

Epoch 327/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3778 Acc: 0.8305

Epoch 328/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3718 Acc: 0.8304
```

```
Epoch 329/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3776 Acc: 0.8322

Epoch 330/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3824 Acc: 0.8308

Epoch 331/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3727 Acc: 0.8318

Epoch 332/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3752 Acc: 0.8325

Epoch 333/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3716 Acc: 0.8304

Epoch 334/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3764 Acc: 0.8298

Epoch 335/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3874 Acc: 0.8295

Epoch 336/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.377 Acc: 0.8314

Epoch 337/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3708 Acc: 0.8303

Epoch 338/349
----------
train Loss: 0.0 Acc: 1.0
```

```
val Loss: 1.3797 Acc: 0.8301

Epoch 339/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3778 Acc: 0.8297

Epoch 340/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3812 Acc: 0.8311

Epoch 341/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3829 Acc: 0.8301

Epoch 342/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.386 Acc: 0.8305

Epoch 343/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3824 Acc: 0.8306

Epoch 344/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3896 Acc: 0.8301

Epoch 345/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3836 Acc: 0.8304

Epoch 346/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3806 Acc: 0.8318

Epoch 347/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3602 Acc: 0.8308

Epoch 348/349
```

```
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3793 Acc: 0.8308

Epoch 349/349
----------
train Loss: 0.0 Acc: 1.0
val Loss: 1.3779 Acc: 0.8308

Training complete in 182.0m 43.079378604888916s
Best val Acc: 0.8325
```

[6]:
```python
results_df.shape
```

[6]:
```
(136850, 6)
```

[7]:
```python
results_df.head(20)
```

[7]:
```
    resnet_layers hardware  epoch  training_step  training_step_loss  \
0              20      K80      0              0            3.425021
1              20      K80      0              1            2.860192
2              20      K80      0              2            3.165385
3              20      K80      0              3            3.740157
4              20      K80      0              4            3.208768
5              20      K80      0              5            3.059932
6              20      K80      0              6            3.161976
7              20      K80      0              7            2.510218
8              20      K80      0              8            2.822329
9              20      K80      0              9            2.698619
10             20      K80      0             10            2.532782
11             20      K80      0             11            2.341161
12             20      K80      0             12            2.716353
13             20      K80      0             13            2.485975
14             20      K80      0             14            2.691406
15             20      K80      0             15            2.547162
16             20      K80      0             16            2.274678
17             20      K80      0             17            2.420792
18             20      K80      0             18            2.174370
19             20      K80      0             19            2.205040

    training_step_time
0             2.043396
1             0.096694
2             0.091998
3             0.087504
4             0.085457
5             0.080485
```

```
6              0.078340
7              0.076159
8              0.075174
9              0.073195
10             0.071628
11             0.071436
12             0.072030
13             0.072090
14             0.072622
15             0.073154
16             0.072544
17             0.073413
18             0.074306
19             0.073762
```

[8]:
```
results_df.to_csv(f'q4_data/layers_{resnet_layers}_{hardware}.csv', index=False)
```

Now that me and my teammates have trained all of the various `ResNet` models on `K80`, `P100`, and `V100`, load in all of the training data and fit each model.

[1]:
```
import numpy as np
import pandas as pd
from scipy.optimize import curve_fit
```

[2]:
```
def f(k, b0, b1, b2):
    l = (1 / (b0 * k + b1)) + b2
    return l
```

[3]:
```
resnets = [18, 20, 32, 44, 56]
gpus = ['K80', 'P100', 'V100']

models = {}

for r in resnets:
    for g in gpus:
        print(f'fitting resnet{r} from gpu {g}')
        df_r_g = pd.read_csv(f'q4_data/resnet{r}_{g}.csv')
        df_r_g = df_r_g.groupby(['resnet_layers', 'hardware',␣
↪'epoch'])['training_step_loss'].mean().reset_index()
        df_r_g.rename(columns={'training_step_loss': 'epoch_loss'},␣
↪inplace=True)

        # fit training loss function and store
        l = df_r_g['epoch_loss'].values
        k = df_r_g['epoch'].values
        betas, _ = curve_fit(f, xdata=k, ydata=l, bounds=(0, np.inf))
        models[(r, g)] = betas
```

```
fitting resnet18 from gpu K80
fitting resnet18 from gpu P100
fitting resnet18 from gpu V100
fitting resnet20 from gpu K80
fitting resnet20 from gpu P100
fitting resnet20 from gpu V100
fitting resnet32 from gpu K80
fitting resnet32 from gpu P100
fitting resnet32 from gpu V100
fitting resnet44 from gpu K80
fitting resnet44 from gpu P100
fitting resnet44 from gpu V100
fitting resnet56 from gpu K80
fitting resnet56 from gpu P100
fitting resnet56 from gpu V100
```

The training loss model parameters learned for the various configurations of `ResNet` layers and GPU are below.

[4]:
```
models
```

[4]:
```
{(18, 'K80'): array([2.79606191e-01, 4.38387877e-01, 1.10788921e-15]),
 (18, 'P100'): array([2.89971602e-01, 4.58898067e-01, 8.40719734e-22]),
 (18, 'V100'): array([2.59958482e-01, 4.67686329e-01, 8.29936723e-20]),
 (20, 'K80'): array([4.29991087e-01, 6.58764805e-01, 7.68529319e-22]),
 (20, 'P100'): array([4.22362166e-01, 6.36503622e-01, 5.20377854e-21]),
 (20, 'V100'): array([4.22374333e-01, 6.56266751e-01, 1.12818064e-21]),
 (32, 'K80'): array([3.73487534e-01, 5.30811749e-01, 3.78300354e-18]),
 (32, 'P100'): array([4.00031023e-01, 5.80132607e-01, 5.13036268e-17]),
 (32, 'V100'): array([3.66265454e-01, 5.15098275e-01, 5.44771001e-19]),
 (44, 'K80'): array([2.08930924e-01, 4.00375364e-01, 1.14601918e-19]),
 (44, 'P100'): array([3.51199957e-01, 4.88508870e-01, 9.08728878e-24]),
 (44, 'V100'): array([3.36027035e-01, 4.48961778e-01, 6.14817529e-17]),
 (56, 'K80'): array([3.24700516e-01, 4.64460939e-01, 2.80048358e-21]),
 (56, 'P100'): array([3.02823087e-01, 4.37064899e-01, 1.46106375e-22]),
 (56, 'V100'): array([2.61575412e-01, 3.94904233e-01, 9.74388361e-25])}
```

Q2: (8 points) Training `ResNet50` to 92% training accuracy.

So far, I've trained on the following GPUs (this is for my own record keeping):

- [X] K80
- [X] P100
- [X] V100

[1]:
```python
import copy

import torch
from torch.autograd import Variable
```

```
import torch.nn as nn
import torch.nn.functional as F
import torch.nn.init as init
import torch.optim as optim
from torch.optim import lr_scheduler
import torchvision
import torchvision.transforms as transforms

import pandas as pd

import time
```

[2]:
```
# load ResNet50
model = torch.hub.load('pytorch/vision:v0.6.0', 'resnet50', pretrained=True)
```

```
Downloading: "https://github.com/pytorch/vision/archive/v0.6.0.zip" to
/home/jupyter/.cache/torch/hub/v0.6.0.zip
Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to
/home/jupyter/.cache/torch/hub/checkpoints/resnet50-19c8e357.pth

HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=102502400.0), HTML(value='')))
```

[3]:
```
# set up data preproceesing
BATCHSIZE = 128
DATA_DIR = '~/data/cifar10'

data_transforms = {
    'train': transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
    ]),
    'val': transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
    ]),
}


train_set = torchvision.datasets.CIFAR10(root=DATA_DIR, train=True,
                                         download=True,␣
 ↪transform=data_transforms['train'])

val_set = torchvision.datasets.CIFAR10(root=DATA_DIR, train=False,
                                       download=True,␣
 ↪transform=data_transforms['val'])
```

```
image_datasets = {'train': train_set, 'val': val_set}

dataloaders = {x: torch.utils.data.DataLoader(image_datasets[x],␣
↪batch_size=BATCHSIZE,
                                              shuffle=True, num_workers=4)
               for x in ['train', 'val']}

dataset_sizes = {x: len(image_datasets[x]) for x in ['train', 'val']}
class_names = image_datasets['train'].classes

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

print(f"Dataset sizes: {dataset_sizes}")
print(f"Class names: {class_names}")
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to
/home/jupyter/data/cifar10/cifar-10-python.tar.gz

HBox(children=(HTML(value=''), FloatProgress(value=1.0, bar_style='info', layout=Layout(width=

Extracting /home/jupyter/data/cifar10/cifar-10-python.tar.gz to
/home/jupyter/data/cifar10
Files already downloaded and verified
Dataset sizes: {'train': 50000, 'val': 10000}
Class names: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
'horse', 'ship', 'truck']
```

[4]:
```
# function to perform training with early stopping at 92% accuarcy
def train_model(model, resnet_layers, hardware, dataloaders, criterion,␣
↪optimizer, scheduler, num_epochs=350, acc_thresh=0.92):
    since = time.time()

    best_model_wts = copy.deepcopy(model.state_dict())
    best_acc = 0.0
    metrics = []
    training_step = 0
    epoch = 0
    threshold_not_met = True

    while epoch < num_epochs and threshold_not_met:
        print(f'Epoch {epoch}/{num_epochs - 1}')
        print('-' * 10)

        for phase in ['train', 'val']:
            if phase == 'train':
                model.train()
```

```
            else:
                model.eval()

            epoch_phase_start_time = time.time()
            running_loss = 0.0
            running_corrects = 0

            for inputs, labels in dataloaders[phase]:
                step_start_time = time.time()

                inputs = inputs.to(device)
                labels = labels.to(device)

                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()

                        metrics.append({
                            'resnet_layers': resnet_layers,
                            'hardware': hardware,
                            'epoch': epoch,
                            'training_step': training_step,
                            'training_step_loss': loss.item(),
                            'training_step_time': time.time() - step_start_time
                        })

                        training_step += 1

                # 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]
            epoch_phase_end_time = time.time()
```

```
            print(f'{phase} Loss: {round(epoch_loss, 4)} Acc: {round(epoch_acc.
↪item(), 4)}')

            # deep copy the model
            if phase == 'val' and epoch_acc > best_acc:
                best_acc = epoch_acc.item()
                best_model_wts = copy.deepcopy(model.state_dict())

            if phase == 'train' and epoch_acc >= acc_thresh:
                print("!!! MET ACCURACY THRESHOLD, STOPPING !!!")
                threshold_not_met = False

        epoch += 1
        print()

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

    # load best model weights
    model.load_state_dict(best_model_wts)

    # set up return structure
    return_df = pd.DataFrame(data=metrics)

    return model, return_df
```

[5]:

```
# Set up criterion
criterion = nn.CrossEntropyLoss()

# Observe that all parameters are being optimized
optimizer = optim.SGD(model.parameters(), lr=0.1, momentum=0.9)

# Decay LR by a factor of 0.1 every 60 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer, step_size=100, gamma=0.1)

# Send model to device
model = model.to(device)

# Train
num_epochs = 350
resnet_layers = 50
acc_thresh = 0.92
hardware = 'V100'
model, results_df = train_model(model, resnet_layers, hardware, dataloaders,
```

```
                                    criterion, optimizer, exp_lr_scheduler,␣
↪num_epochs, acc_thresh)
```

```
Epoch 0/349
----------
train Loss: 2.387 Acc: 0.2429
val Loss: 1.8514 Acc: 0.3165

Epoch 1/349
----------
train Loss: 1.7665 Acc: 0.3467
val Loss: 1.6653 Acc: 0.3798

Epoch 2/349
----------
train Loss: 1.6171 Acc: 0.4065
val Loss: 1.5176 Acc: 0.4398

Epoch 3/349
----------
train Loss: 1.5003 Acc: 0.4518
val Loss: 1.4791 Acc: 0.462

Epoch 4/349
----------
train Loss: 1.4098 Acc: 0.4858
val Loss: 1.4081 Acc: 0.4943

Epoch 5/349
----------
train Loss: 1.3187 Acc: 0.5236
val Loss: 1.3106 Acc: 0.5224

Epoch 6/349
----------
train Loss: 1.2383 Acc: 0.5517
val Loss: 1.2523 Acc: 0.5514

Epoch 7/349
----------
train Loss: 1.1777 Acc: 0.5768
val Loss: 1.2196 Acc: 0.5709

Epoch 8/349
----------
train Loss: 1.0925 Acc: 0.6067
```

```
val Loss: 1.1322 Acc: 0.6047

Epoch 9/349
----------
train Loss: 1.0778 Acc: 0.612
val Loss: 1.081 Acc: 0.6114

Epoch 10/349
----------
train Loss: 0.9746 Acc: 0.6513
val Loss: 1.0486 Acc: 0.6281

Epoch 11/349
----------
train Loss: 0.9105 Acc: 0.6744
val Loss: 1.0428 Acc: 0.6422

Epoch 12/349
----------
train Loss: 0.8622 Acc: 0.6939
val Loss: 1.0024 Acc: 0.6552

Epoch 13/349
----------
train Loss: 0.8284 Acc: 0.7039
val Loss: 1.0167 Acc: 0.6547

Epoch 14/349
----------
train Loss: 0.7682 Acc: 0.7256
val Loss: 0.9409 Acc: 0.676

Epoch 15/349
----------
train Loss: 0.7031 Acc: 0.7485
val Loss: 0.9449 Acc: 0.6786

Epoch 16/349
----------
train Loss: 0.6368 Acc: 0.7709
val Loss: 0.9232 Acc: 0.69

Epoch 17/349
----------
train Loss: 0.5869 Acc: 0.7881
val Loss: 0.9782 Acc: 0.6823

Epoch 18/349
```

```
----------
train Loss: 0.5313 Acc: 0.809
val Loss: 0.9941 Acc: 0.6846

Epoch 19/349
----------
train Loss: 0.4826 Acc: 0.8282
val Loss: 0.9619 Acc: 0.7002

Epoch 20/349
----------
train Loss: 0.432 Acc: 0.845
val Loss: 1.0157 Acc: 0.6895

Epoch 21/349
----------
train Loss: 0.3767 Acc: 0.865
val Loss: 1.1024 Acc: 0.6915

Epoch 22/349
----------
train Loss: 0.3169 Acc: 0.8877
val Loss: 1.1041 Acc: 0.6945

Epoch 23/349
----------
train Loss: 0.2812 Acc: 0.9003
val Loss: 1.1727 Acc: 0.6913

Epoch 24/349
----------
train Loss: 0.2427 Acc: 0.914
val Loss: 1.1827 Acc: 0.7

Epoch 25/349
----------
train Loss: 0.209 Acc: 0.9262
!!! MET ACCURACY THRESHOLD, STOPPING !!!
val Loss: 1.2001 Acc: 0.7014

Training complete in 10.0m 1.1729986667633057s
Best val Acc: 0.7014
```

[6]:
```
results_df.shape
```

[6]:
```
(10166, 6)
```

[7]:
```
results_df.head(20)
```

[7]:
```
    resnet_layers hardware  epoch  training_step  training_step_loss  \
0              50     V100      0              0           13.322277
1              50     V100      0              1            7.019900
2              50     V100      0              2            4.346029
3              50     V100      0              3            3.709551
4              50     V100      0              4            3.560764
5              50     V100      0              5            4.032091
6              50     V100      0              6            2.668718
7              50     V100      0              7            4.557339
8              50     V100      0              8            5.248215
9              50     V100      0              9            3.843186
10             50     V100      0             10           10.561168
11             50     V100      0             11            7.067451
12             50     V100      0             12           10.018452
13             50     V100      0             13            3.212391
14             50     V100      0             14            6.883983
15             50     V100      0             15            3.155358
16             50     V100      0             16            4.572664
17             50     V100      0             17            5.905703
18             50     V100      0             18            3.942033
19             50     V100      0             19            2.786420

    training_step_time
0             6.653620
1             0.064077
2             0.051719
3             0.053324
4             0.053852
5             0.052706
6             0.053849
7             0.052973
8             0.058157
9             0.053047
10            0.053207
11            0.051462
12            0.043801
13            0.043869
14            0.043779
15            0.043885
16            0.049052
17            0.053222
18            0.058126
19            0.057351
```

[8]:
```
results_df.to_csv(f'q4_data/resnet{resnet_layers}_{hardware}.csv', index=False)
```

Now that I have trained ResNet50 on K80, P100, and V100, load in all of the training data.

[5]:
```
resnet50_training_data = []
for g in gpus:
    df_50_g = pd.read_csv(f'q4_data/resnet50_{g}.csv')
    df_50_g = df_50_g.groupby(['resnet_layers', 'hardware',␣
↪'epoch'])['training_step_loss'].mean().reset_index()
    df_50_g.rename(columns={'training_step_loss': 'epoch_loss'}, inplace=True)
    resnet50_training_data.append(df_50_g)

df_resnet50_training_data = pd.concat(resnet50_training_data, axis=0,␣
↪ignore_index=True)
print(df_resnet50_training_data.shape)
df_resnet50_training_data.head()
```

```
(77, 4)
```

[5]:

	resnet_layers	hardware	epoch	epoch_loss
0	50	K80	0	2.509837
1	50	K80	1	1.711982
2	50	K80	2	1.538258
3	50	K80	3	1.440099
4	50	K80	4	1.354824

Next, fit linear models given the number of ResNet layers and type of GPU for β_0, β_1, and β_2.

[6]:
```
# format data
beta_data = []
for (r, g), betas in models.items():
    beta_data.append({
        'layers': r,
        'K80': 1 if g == 'K80' else 0,
        'P100': 1 if g == 'P100' else 0,
        'V100': 1 if g == 'V100' else 0,
        'beta0': betas[0],
        'beta1': betas[1],
        'beta2': betas[2]
    })
df_beta = pd.DataFrame(beta_data)
df_beta
```

[6]:

	layers	K80	P100	V100	beta0	beta1	beta2
0	18	1	0	0	0.279606	0.438388	1.107889e-15
1	18	0	1	0	0.289972	0.458898	8.407197e-22
2	18	0	0	1	0.259958	0.467686	8.299367e-20
3	20	1	0	0	0.429991	0.658765	7.685293e-22
4	20	0	1	0	0.422362	0.636504	5.203779e-21
5	20	0	0	1	0.422374	0.656267	1.128181e-21

6	32	1	0	0	0.373488	0.530812	3.783004e-18
7	32	0	1	0	0.400031	0.580133	5.130363e-17
8	32	0	0	1	0.366265	0.515098	5.447710e-19
9	44	1	0	0	0.208931	0.400375	1.146019e-19
10	44	0	1	0	0.351200	0.488509	9.087289e-24
11	44	0	0	1	0.336027	0.448962	6.148175e-17
12	56	1	0	0	0.324701	0.464461	2.800484e-21
13	56	0	1	0	0.302823	0.437065	1.461064e-22
14	56	0	0	1	0.261575	0.394904	9.743884e-25

[7]:
```python
from sklearn import linear_model
```

[8]:
```python
# fit models
beta_models = {}
for i in range(3):
    reg = linear_model.LinearRegression()
    X = df_beta[['layers', 'K80', 'P100', 'V100']].values
    y = df_beta[[f'beta{i}']].values
    reg.fit(X, y)
    beta_models[i] = reg
beta_models
```

[8]: {0: LinearRegression(), 1: LinearRegression(), 2: LinearRegression()}

Now that the regression models have been trained, predict the training loss curve for `ResNet50` on the various GPUs.

[9]:
```python
df_predict = pd.DataFrame({
    'layers': 50,
    'K80': [1, 0, 0],
    'P100': [0, 1, 0],
    'V100': [0, 0, 1],
})

df_predict
```

[9]:

	layers	K80	P100	V100
0	50	1	0	0
1	50	0	1	0
2	50	0	0	1

[10]:
```python
pred_betas = {}
for i in range(3):
    reg = beta_models[i]
    X = df_predict.values
    y_pred = reg.predict(X)
    pred_betas[f'beta{i}'] = y_pred.reshape(1, -1).tolist()[0]
```

```
df_pred_betas = pd.DataFrame(pred_betas)
df_pred_betas
```

[10]:

```
      beta0     beta1         beta2
0  0.298335  0.444646  1.340355e-16
1  0.328269  0.466308 -7.806061e-17
2  0.304232  0.442670 -7.590045e-17
```

[11]:

```
df_predict_w_betas = pd.concat([df_predict, df_pred_betas], axis=1)
df_predict_w_betas
```

[11]:

```
   layers  K80  P100  V100     beta0     beta1         beta2
0      50    1     0     0  0.298335  0.444646  1.340355e-16
1      50    0     1     0  0.328269  0.466308 -7.806061e-17
2      50    0     0     1  0.304232  0.442670 -7.590045e-17
```

[12]:

```
resnet50_pred_training_loss = {}
for idx, g in enumerate(gpus):
    k = np.arange(max(df_r_g['epoch'].values))
    b0 = df_predict_w_betas['beta0'][idx]
    b1 = df_predict_w_betas['beta1'][idx]
    b2 = df_predict_w_betas['beta2'][idx]
    resnet50_pred_training_loss[g] = f(k, b0, b1, b2)
```

[13]:

```
resnet50_pred_training_loss
```

[13]:

```
{'K80': array([2.24897889, 1.34592871, 0.96032276, 0.74646267, 0.6105055 ,
        0.51644302, 0.44749593, 0.39478998, 0.35319121, 0.31952325,
        0.29171547, 0.26836034, 0.24846769, 0.23132066, 0.21638752,
        0.20326551, 0.19164398, 0.18127948, 0.17197853, 0.16358542,
        0.1559734 , 0.1490383 , 0.14269366, 0.13686715, 0.1314978 ,
        0.12653382, 0.12193099, 0.11765127, 0.1136618 , 0.10993401,
        0.10644298, 0.10316685, 0.10008636, 0.0971845 , 0.09444617,
        0.09185793, 0.08940776, 0.0870849 , 0.08487968, 0.08278339,
        0.08078815, 0.07888683, 0.07707294, 0.07534059, 0.0736844 ,
        0.07209946, 0.07058127, 0.0691257 , 0.06772895, 0.06638753,
        0.06509821, 0.06385802, 0.0626642 , 0.0615142 , 0.06040564,
        0.05933633, 0.05830423, 0.05730741, 0.0563441 , 0.05541265,
        0.05451149, 0.05363917, 0.05279433, 0.0519757 , 0.05118206,
        0.05041229, 0.04966534, 0.04894019, 0.04823592, 0.04755163,
        0.04688649, 0.04623969, 0.0456105 , 0.0449982 , 0.04440212,
        0.04382163, 0.04325612, 0.04270501, 0.04216778, 0.04164389,
        0.04113287, 0.04063423, 0.04014753, 0.03967236, 0.03920831,
        0.03875498, 0.03831202, 0.03787907, 0.03745579, 0.03704187,
        0.036637  , 0.03624088, 0.03585324, 0.0354738 , 0.03510231,
        0.03473852, 0.03438219, 0.0340331 , 0.03369103, 0.03335576,
        0.0330271 , 0.03270486, 0.03238884, 0.03207887, 0.03177478,
```

```
0.0314764 , 0.03118357, 0.03089613, 0.03061395, 0.03033688,
0.03006478, 0.02979751, 0.02953496, 0.02927699, 0.02902349,
0.02877434, 0.02852943, 0.02828865, 0.02805191, 0.0278191 ,
0.02759011, 0.02736487, 0.02714328, 0.02692524, 0.02671068,
0.02649951, 0.02629166, 0.02608704, 0.02588558, 0.02568721,
0.02549185, 0.02529945, 0.02510993, 0.02492322, 0.02473927,
0.02455802, 0.0243794 , 0.02420337, 0.02402986, 0.02385881,
0.02369019, 0.02352393, 0.02335999, 0.02319832, 0.02303887,
0.0228816 , 0.02272646, 0.02257341, 0.0224224 , 0.02227341,
0.02212638, 0.02198128, 0.02183807, 0.02169672, 0.02155718,
0.02141942, 0.02128342, 0.02114913, 0.02101653, 0.02088557,
0.02075624, 0.02062851, 0.02050233, 0.02037769, 0.02025455,
0.0201329 , 0.02001269, 0.01989392, 0.01977654, 0.01966055,
0.0195459 , 0.01943258, 0.01932057, 0.01920985, 0.01910039,
0.01899216, 0.01888516, 0.01877935, 0.01867473, 0.01857126,
0.01846893, 0.01836773, 0.01826763, 0.01816861, 0.01807066,
0.01797376, 0.0178779 , 0.01778305, 0.0176892 , 0.01759634,
0.01750445, 0.01741351, 0.01732352, 0.01723445, 0.01714629,
0.01705902, 0.01697264, 0.01688714, 0.01680248, 0.01671868,
0.0166357 , 0.01655355, 0.0164722 , 0.01639165, 0.01631188,
0.01623288, 0.01615465, 0.01607716, 0.01600042, 0.0159244 ,
0.01584911, 0.01577452, 0.01570063, 0.01562743, 0.01555491,
0.01548306, 0.01541187, 0.01534133, 0.01527144, 0.01520218,
0.01513354, 0.01506552, 0.01499811, 0.0149313 , 0.01486509,
0.01479945, 0.0147344 , 0.01466991, 0.01460599, 0.01454262,
0.0144798 , 0.01441752, 0.01435577, 0.01429455, 0.01423385,
0.01417366, 0.01411398, 0.0140548 , 0.01399611, 0.01393791,
0.0138802 , 0.01382296, 0.01376619, 0.01370988, 0.01365403,
0.01359864, 0.01354369, 0.01348919, 0.01343512, 0.01338149,
0.01332828, 0.01327549, 0.01322312, 0.01317116, 0.01311961,
0.01306846, 0.01301771, 0.01296735, 0.01291737, 0.01286778,
0.01281858, 0.01276974, 0.01272128, 0.01267318, 0.01262544,
0.01257807, 0.01253105, 0.01248437, 0.01243805, 0.01239206,
0.01234642, 0.01230111, 0.01225613, 0.01221148, 0.01216716,
0.01212315, 0.01207946, 0.01203609, 0.01199302, 0.01195026,
0.01190781, 0.01186566, 0.0118238 , 0.01178224, 0.01174097,
0.01169999, 0.01165929, 0.01161888, 0.01157874, 0.01153888,
0.0114993 , 0.01145998, 0.01142093, 0.01138215, 0.01134363,
0.01130537, 0.01126737, 0.01122962, 0.01119213, 0.01115488,
0.01111788, 0.01108113, 0.01104461, 0.01100834, 0.01097231,
0.01093651, 0.01090094, 0.0108656 , 0.0108305 , 0.01079561,
0.01076096, 0.01072652, 0.0106923 , 0.0106583 , 0.01062452,
0.01059095, 0.01055759, 0.01052444, 0.0104915 , 0.01045877,
0.01042624, 0.0103939 , 0.01036177, 0.01032984, 0.01029811,
0.01026656, 0.01023521, 0.01020406, 0.01017309, 0.01014231,
0.01011171, 0.0100813 , 0.01005107, 0.01002102, 0.00999115,
0.00996146, 0.00993194, 0.0099026 , 0.00987343, 0.00984443,
```

```
       0.0098156 , 0.00978694, 0.00975845, 0.00973012, 0.00970196,
       0.00967396, 0.00964612, 0.00961844, 0.00959092]),
'P100': array([2.14450671, 1.25853098, 0.89059356, 0.68912472, 0.56199173,
       0.47446093, 0.41052174, 0.3617691 , 0.32336679, 0.292335  ,
       0.26673762, 0.24526203, 0.22698686, 0.21124629, 0.19754724,
       0.18551671, 0.17486738, 0.1653743 , 0.15685885, 0.14917741,
       0.14221318, 0.13587018, 0.13006884, 0.12474263, 0.11983546,
       0.11529976, 0.11109489, 0.10718592, 0.10354268, 0.10013897,
       0.09695191, 0.09396146, 0.09114997, 0.08850184, 0.08600323,
       0.08364184, 0.08140665, 0.07928782, 0.07727648, 0.07536467,
       0.07354517, 0.07181145, 0.07015759, 0.06857819, 0.06706834,
       0.06562354, 0.06423968, 0.06291297, 0.06163996, 0.06041744,
       0.05924247, 0.05811233, 0.0570245 , 0.05597666, 0.05496662,
       0.05399239, 0.05305209, 0.05214398, 0.05126644, 0.05041795,
       0.04959708, 0.04880252, 0.04803301, 0.04728739, 0.04656457,
       0.04586351, 0.04518325, 0.04452288, 0.04388153, 0.04325839,
       0.04265271, 0.04206375, 0.04149083, 0.04093331, 0.04039058,
       0.03986205, 0.03934717, 0.03884542, 0.03835631, 0.03787937,
       0.03741414, 0.03696019, 0.03651713, 0.03608457, 0.03566214,
       0.03524948, 0.03484626, 0.03445216, 0.03406688, 0.03369012,
       0.0333216 , 0.03296106, 0.03260824, 0.03226288, 0.03192477,
       0.03159367, 0.03126937, 0.03095166, 0.03064034, 0.03033522,
       0.03003611, 0.02974285, 0.02945526, 0.02917318, 0.02889644,
       0.02862491, 0.02835844, 0.02809688, 0.0278401 , 0.02758797,
       0.02734037, 0.02709717, 0.02685826, 0.02662353, 0.02639286,
       0.02616616, 0.02594332, 0.02572424, 0.02550883, 0.025297  ,
       0.02508866, 0.02488372, 0.0246821 , 0.02448373, 0.02428851,
       0.02409639, 0.02390728, 0.02372112, 0.02353783, 0.02335735,
       0.02317962, 0.02300458, 0.02283216, 0.0226623 , 0.02249495,
       0.02233006, 0.02216756, 0.02200742, 0.02184957, 0.02169397,
       0.02154057, 0.02138932, 0.02124018, 0.02109311, 0.02094806,
       0.02080499, 0.02066387, 0.02052464, 0.02038728, 0.02025175,
       0.020118  , 0.01998601, 0.01985574, 0.01972716, 0.01960023,
       0.01947493, 0.01935121, 0.01922906, 0.01910844, 0.01898933,
       0.01887169, 0.0187555 , 0.01864073, 0.01852736, 0.01841536,
       0.0183047 , 0.01819537, 0.01808733, 0.01798057, 0.01787507,
       0.01777079, 0.01766772, 0.01756585, 0.01746514, 0.01736557,
       0.01726714, 0.01716982, 0.01707359, 0.01697843, 0.01688432,
       0.01679125, 0.01669921, 0.01660816, 0.01651811, 0.01642902,
       0.01634089, 0.01625371, 0.01616744, 0.01608209, 0.01599764,
       0.01591406, 0.01583136, 0.01574951, 0.0156685 , 0.01558832,
       0.01550896, 0.0154304 , 0.01535264, 0.01527565, 0.01519943,
       0.01512397, 0.01504925, 0.01497527, 0.01490202, 0.01482947,
       0.01475763, 0.01468648, 0.01461602, 0.01454622, 0.0144771 ,
       0.01440862, 0.01434079, 0.01427359, 0.01420703, 0.01414108,
       0.01407574, 0.014011  , 0.01394685, 0.01388329, 0.0138203 ,
       0.01375788, 0.01369603, 0.01363473, 0.01357397, 0.01351376,
```

```
       0.01345407, 0.01339491, 0.01333627, 0.01327814, 0.01322052,
       0.01316339, 0.01310675, 0.0130506 , 0.01299493, 0.01293973,
       0.012885  , 0.01283073, 0.01277691, 0.01272355, 0.01267062,
       0.01261814, 0.01256609, 0.01251447, 0.01246327, 0.01241248,
       0.01236211, 0.01231215, 0.01226259, 0.01221342, 0.01216465,
       0.01211627, 0.01206827, 0.01202064, 0.0119734 , 0.01192652,
       0.01188001, 0.01183386, 0.01178807, 0.01174263, 0.01169753,
       0.01165279, 0.01160838, 0.01156432, 0.01152058, 0.01147718,
       0.0114341 , 0.01139134, 0.0113489 , 0.01130678, 0.01126497,
       0.01122346, 0.01118226, 0.01114137, 0.01110077, 0.01106046,
       0.01102045, 0.01098072, 0.01094128, 0.01090213, 0.01086325,
       0.01082465, 0.01078632, 0.01074826, 0.01071047, 0.01067295,
       0.01063568, 0.01059868, 0.01056193, 0.01052544, 0.0104892 ,
       0.0104532 , 0.01041746, 0.01038195, 0.01034669, 0.01031167,
       0.01027688, 0.01024233, 0.010208  , 0.01017391, 0.01014005,
       0.01010641, 0.01007299, 0.01003979, 0.01000681, 0.00997404,
       0.00994149, 0.00990916, 0.00987703, 0.00984511, 0.00981339,
       0.00978188, 0.00975057, 0.00971946, 0.00968855, 0.00965783,
       0.00962731, 0.00959698, 0.00956684, 0.00953689, 0.00950712,
       0.00947755, 0.00944815, 0.00941894, 0.00938991, 0.00936105,
       0.00933237, 0.00930387, 0.00927554, 0.00924738, 0.0092194 ,
       0.00919158, 0.00916393, 0.00913644, 0.00910912, 0.00908197,
       0.00905497, 0.00902813, 0.00900146, 0.00897494, 0.00894857,
       0.00892236, 0.00889631, 0.0088704 , 0.00884465, 0.00881904,
       0.00879358, 0.00876827, 0.00874311, 0.00871809]),
'V100': array([2.25902136, 1.33886442, 0.95135379, 0.73780824, 0.60255564,
       0.50920909, 0.44090507, 0.38875806, 0.34764156, 0.31439046,
       0.28694486, 0.2639064 , 0.24429246, 0.2273923 , 0.21267915,
       0.19975429, 0.18831035, 0.17810662, 0.16895184, 0.16069217,
       0.15320245, 0.14637982, 0.14013895, 0.13440848, 0.12912825,
       0.1242472 , 0.11972172, 0.11551432, 0.11159261, 0.10792843,
       0.10449723, 0.10127748, 0.09825021, 0.09539866, 0.09270797,
       0.09016489, 0.08775761, 0.08547552, 0.08330912, 0.08124982,
       0.07928987, 0.07742225, 0.07564059, 0.07393908, 0.07231244,
       0.07075582, 0.06926482, 0.06783535, 0.06646369, 0.06514641,
       0.06388032, 0.06266251, 0.06149027, 0.06036107, 0.0592726 ,
       0.0582227 , 0.05720934, 0.05623065, 0.05528488, 0.0543704 ,
       0.05348568, 0.05262929, 0.0517999 , 0.05099624, 0.05021714,
       0.04946148, 0.04872823, 0.0480164 , 0.04732507, 0.04665336,
       0.04600046, 0.04536558, 0.04474798, 0.04414697, 0.0435619 ,
       0.04299213, 0.04243707, 0.04189616, 0.04136886, 0.04085468,
       0.04035312, 0.03986372, 0.03938605, 0.0389197 , 0.03846426,
       0.03801935, 0.03758462, 0.03715972, 0.03674432, 0.0363381 ,
       0.03594077, 0.03555203, 0.03517161, 0.03479925, 0.03443469,
       0.03407769, 0.03372801, 0.03338544, 0.03304975, 0.03272075,
       0.03239824, 0.03208202, 0.03177191, 0.03146775, 0.03116935,
       0.03087655, 0.03058921, 0.03030716, 0.03003027, 0.02975839,
```

0.0294914 , 0.02922914, 0.02897152, 0.02871839, 0.02846965,
0.02822518, 0.02798488, 0.02774863, 0.02751633, 0.0272879 ,
0.02706322, 0.02684222, 0.02662479, 0.02641086, 0.02620034,
0.02599315, 0.02578921, 0.02558844, 0.02539078, 0.02519615,
0.02500448, 0.0248157 , 0.02462975, 0.02444657, 0.02426609,
0.02408826, 0.02391302, 0.0237403 , 0.02357007, 0.02340225,
0.02323682, 0.0230737 , 0.02291286, 0.02275424, 0.02259781,
0.02244351, 0.0222913 , 0.02214115, 0.021993 , 0.02184682,
0.02170258, 0.02156022, 0.02141972, 0.02128104, 0.02114415,
0.021009 , 0.02087558, 0.02074383, 0.02061374, 0.02048527,
0.02035839, 0.02023307, 0.02010929, 0.01998701, 0.01986621,
0.01974686, 0.01962894, 0.01951241, 0.01939727, 0.01928347,
0.019171 , 0.01905984, 0.01894995, 0.01884133, 0.01873394,
0.01862777, 0.0185228 , 0.01841901, 0.01831637, 0.01821487,
0.01811449, 0.0180152 , 0.017917 , 0.01781987, 0.01772378,
0.01762873, 0.01753468, 0.01744164, 0.01734958, 0.01725848,
0.01716834, 0.01707913, 0.01699085, 0.01690347, 0.01681699,
0.01673138, 0.01664665, 0.01656277, 0.01647973, 0.01639752,
0.01631612, 0.01623553, 0.01615573, 0.01607671, 0.01599846,
0.01592097, 0.01584423, 0.01576822, 0.01569294, 0.01561837,
0.01554451, 0.01547134, 0.01539886, 0.01532706, 0.01525592,
0.01518544, 0.01511561, 0.01504641, 0.01497785, 0.01490991,
0.01484258, 0.01477586, 0.01470974, 0.0146442 , 0.01457925,
0.01451487, 0.01445105, 0.0143878 , 0.01432509, 0.01426293,
0.01420131, 0.01414022, 0.01407965, 0.01401959, 0.01396005,
0.01390101, 0.01384247, 0.01378442, 0.01372686, 0.01366977,
0.01361315, 0.01355701, 0.01350132, 0.01344609, 0.01339131,
0.01333697, 0.01328308, 0.01322962, 0.01317658, 0.01312397,
0.01307178, 0.01302   , 0.01296863, 0.01291766, 0.0128671 ,
0.01281692, 0.01276714, 0.01271774, 0.01266873, 0.01262008,
0.01257182, 0.01252392, 0.01247638, 0.0124292 , 0.01238238,
0.01233591, 0.01228978, 0.012244 , 0.01219856, 0.01215346,
0.01210869, 0.01206425, 0.01202013, 0.01197633, 0.01193285,
0.01188969, 0.01184684, 0.01180429, 0.01176205, 0.01172011,
0.01167847, 0.01163713, 0.01159607, 0.0115553 , 0.01151482,
0.01147463, 0.01143471, 0.01139507, 0.0113557 , 0.0113166 ,
0.01127778, 0.01123921, 0.01120091, 0.01116287, 0.01112509,
0.01108757, 0.01105029, 0.01101327, 0.01097649, 0.01093996,
0.01090366, 0.01086761, 0.0108318 , 0.01079622, 0.01076088,
0.01072576, 0.01069088, 0.01065622, 0.01062178, 0.01058757,
0.01055358, 0.0105198 , 0.01048624, 0.01045289, 0.01041976,
0.01038683, 0.01035411, 0.0103216 , 0.01028929, 0.01025718,
0.01022527, 0.01019356, 0.01016204, 0.01013072, 0.0100996 ,
0.01006866, 0.01003791, 0.01000735, 0.00997697, 0.00994678,
0.00991677, 0.00988694, 0.00985729, 0.00982782, 0.00979853,
0.0097694 , 0.00974045, 0.00971167, 0.00968306, 0.00965462,
0.00962635, 0.00959824, 0.00957029, 0.00954251, 0.00951488,

```
        0.00948742, 0.00946011, 0.00943297, 0.00940597])}
```

Make the dataframe for the plotting the predicted loss curve with the actual loss curve for each `ResNet50` trained on the different GPUs, with percent error.

[14]:
```
viz_data = []
perc_error = {}
for g in gpus:
    print(f'{g}')

    actual_train_loss = df_resnet50_training_data[df_resnet50_training_data.
 ↪hardware==g]['epoch_loss'].values
    pred_training_loss_g = resnet50_pred_training_loss[g]
    pred_training_loss_g_sub = pred_training_loss_g[:len(actual_train_loss)]
    perc_diff = pred_training_loss_g_sub - actual_train_loss / actual_train_loss

    df_g_actual = pd.DataFrame(data={
        'gpu': g,
        'mode': 'actual',
        'epoch': df_resnet50_training_data[df_resnet50_training_data.
 ↪hardware==g]['epoch'].values.tolist(),
        'epoch_loss': df_resnet50_training_data[df_resnet50_training_data.
 ↪hardware==g]['epoch_loss'].values.tolist(),
    })
    df_g_pred = pd.DataFrame(data={
        'gpu': g,
        'mode': 'predict',
        'epoch': np.arange(pred_training_loss_g.shape[0]).tolist(),
        'epoch_loss': pred_training_loss_g.tolist()
    })

    viz_data.append(df_g_actual)
    viz_data.append(df_g_pred)

    perc_error[g] = perc_diff

df_viz = pd.concat(viz_data, axis=0, ignore_index=True)
print(df_viz.shape)
df_viz.head()
```

```
K80
P100
V100
(1124, 4)
```

[14]:

	gpu	mode	epoch	epoch_loss
0	K80	actual	0	2.509837

```
1   K80   actual      1    1.711982
2   K80   actual      2    1.538258
3   K80   actual      3    1.440099
4   K80   actual      4    1.354824
```

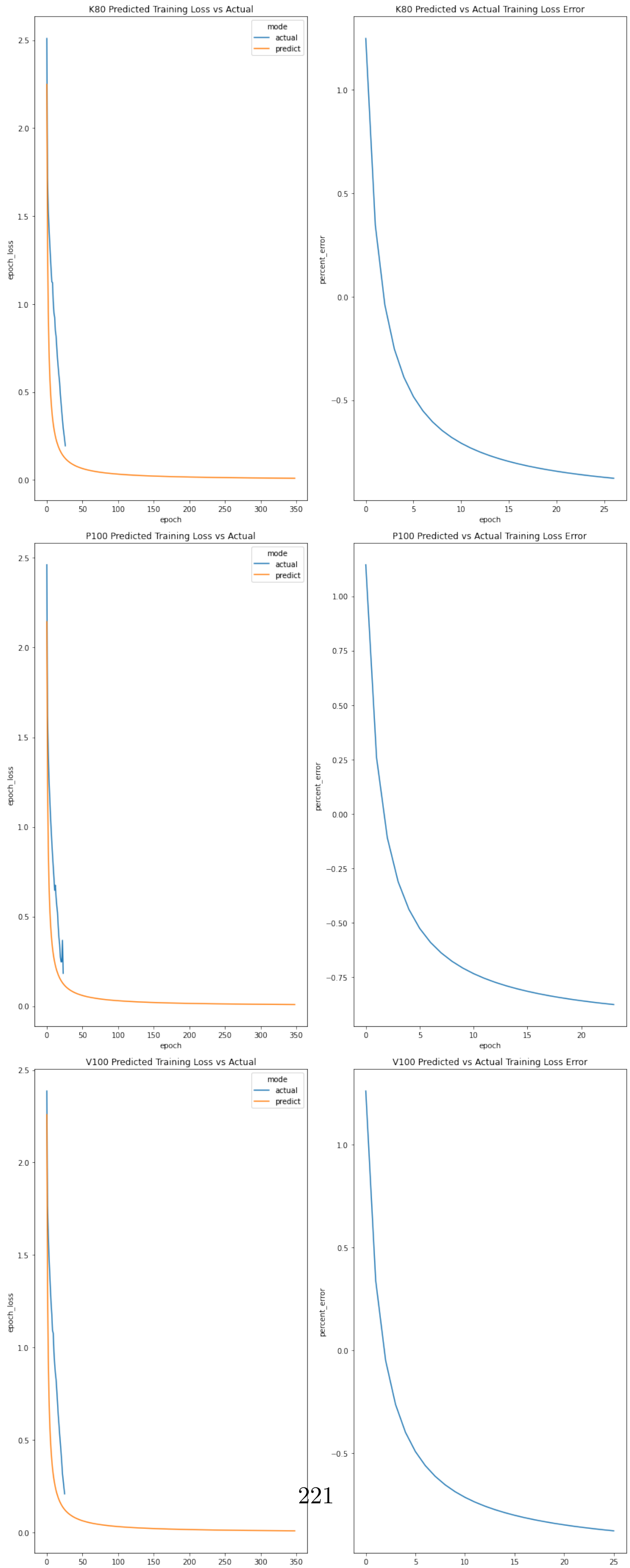
[15]:
```
perc_error
```

[15]:
```
{'K80': array([ 1.24897889,  0.34592871, -0.03967724, -0.25353733, -0.3894945 ,
        -0.48355698, -0.55250407, -0.60521002, -0.64680879, -0.68047675,
        -0.70828453, -0.73163966, -0.75153231, -0.76867934, -0.78361248,
        -0.79673449, -0.80835602, -0.81872052, -0.82802147, -0.83641458,
        -0.8440266 , -0.8509617 , -0.85730634, -0.86313285, -0.8685022 ,
        -0.87346618, -0.87806901]),
 'P100': array([ 1.14450671,  0.25853098, -0.10940644, -0.31087528, -0.43800827,
        -0.52553907, -0.58947826, -0.6382309 , -0.67663321, -0.707665  ,
        -0.73326238, -0.75473797, -0.77301314, -0.78875371, -0.80245276,
        -0.81448329, -0.82513262, -0.8346257 , -0.84314115, -0.85082259,
        -0.85778682, -0.86412982, -0.86993116, -0.87525737]),
 'V100': array([ 1.25902136,  0.33886442, -0.04864621, -0.26219176, -0.39744436,
        -0.49079091, -0.55909493, -0.61124194, -0.65235844, -0.68560954,
        -0.71305514, -0.7360936 , -0.75570754, -0.7726077 , -0.78732085,
        -0.80024571, -0.81168965, -0.82189338, -0.83104816, -0.83930783,
        -0.84679755, -0.85362018, -0.85986105, -0.86559152, -0.87087175,
        -0.8757528 ])}
```

[16]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

[18]:
```
fig, axs = plt.subplots(nrows=len(gpus), ncols=2, figsize=(12, 30))
for idx, g in enumerate(gpus):
    df_for_plot = df_viz[df_viz.gpu == g]
    df_g_error = pd.DataFrame(data={
        'epoch': np.arange(len(perc_error[g])),
        'percent_error': perc_error[g]
    })
    sns.lineplot(x='epoch', y='epoch_loss', hue='mode', data=df_for_plot,␣
 ↪ax=axs[idx, 0])
    sns.lineplot(x='epoch', y='percent_error', data=df_g_error, ax=axs[idx, 1])

    axs[idx, 0].set_title(f'{g} Predicted Training Loss vs Actual')
    axs[idx, 1].set_title(f'{g} Predicted vs Actual Training Loss Error')
fig.tight_layout()
```


K80 Predicted Training Loss vs Actual
mode
actual
predict
epoch_loss
epoch
K80 Predicted vs Actual Training Loss Error
percent_error
epoch
P100 Predicted Training Loss vs Actual
mode
actual
predict
epoch_loss
epoch
P100 Predicted vs Actual Training Loss Error
percent_error
epoch
V100 Predicted Training Loss vs Actual
mode
actual
predict
epoch_loss
epoch
V100 Predicted vs Actual Training Loss Error
percent_error
epoch

From above we can see the performance of the ResNet50 model in actuality on various hardware versus the predicted performance based on the other versions of ResNet that were trained.

In each of the cases above, the model converges to 92% accuracy around the same epoch regardless of what hardware it was trained on.

[19]:
```
threshold_acc_loss_val = min(df_resnet50_training_data['epoch_loss'].values)
print(f'The training loss associated with 92% accuracy in the actual training␣
 ↪data is {threshold_acc_loss_val}')
```

```
The training loss associated with 92% accuracy in the actual training data is
0.18296788097418787
```

Find the first epoch for the predicted ResNet50 loss lower than the threshold above.

[20]:
```
for g in gpus:
    pred_training_loss_g = resnet50_pred_training_loss[g].tolist()
    lower_than_thresh_mask = [i for i, x in enumerate(pred_training_loss_g) if␣
 ↪x <= threshold_acc_loss_val]
    print(f'For {g} we predict that at epoch {lower_than_thresh_mask[0]} we␣
 ↪would achieve 92% accuracy')
```

```
For K80 we predict that at epoch 17 we would achieve 92% accuracy
For P100 we predict that at epoch 16 we would achieve 92% accuracy
For V100 we predict that at epoch 17 we would achieve 92% accuracy
```

From above we can see that based on the predicted training loss curves from the learned beta values, we would have expected to achieve 92% accuracy when training ResNet50 by epoch 17, when in reality we achieved 92% accuracy between 23rd and 26th epochs.

Q3: (7 points) Using equation 4 from Peng et al with the learned synchronous theta parameters from Table 2.

[44]:
```
def f(w, p, M,
      theta0=1.02,
      theta1=2.78,
      theta2=4.92,
      theta3=0.00,
      theta4=0.02):
    return (theta0 * M/w + theta1 + theta2 * w/p + theta3 * w + theta4 * p)**-1
```

[45]:
```
ps = [2, 4]
w = np.arange(1, 21)
data = []
M = 128
predicted_epochs = 17
training_data_size = 50000
```

```python
for p in ps:
    f_p = f(w, p, M)
    time_to_acc = predicted_epochs / f_p
    df_p = pd.DataFrame(data={
        'parameter servers': p,
        'workers': w.tolist(),
        'time to accuracy (s)': time_to_acc
    })
    data.append(df_p)
df = pd.concat(data, axis=0, ignore_index=True)
df.head()
```

[45]:

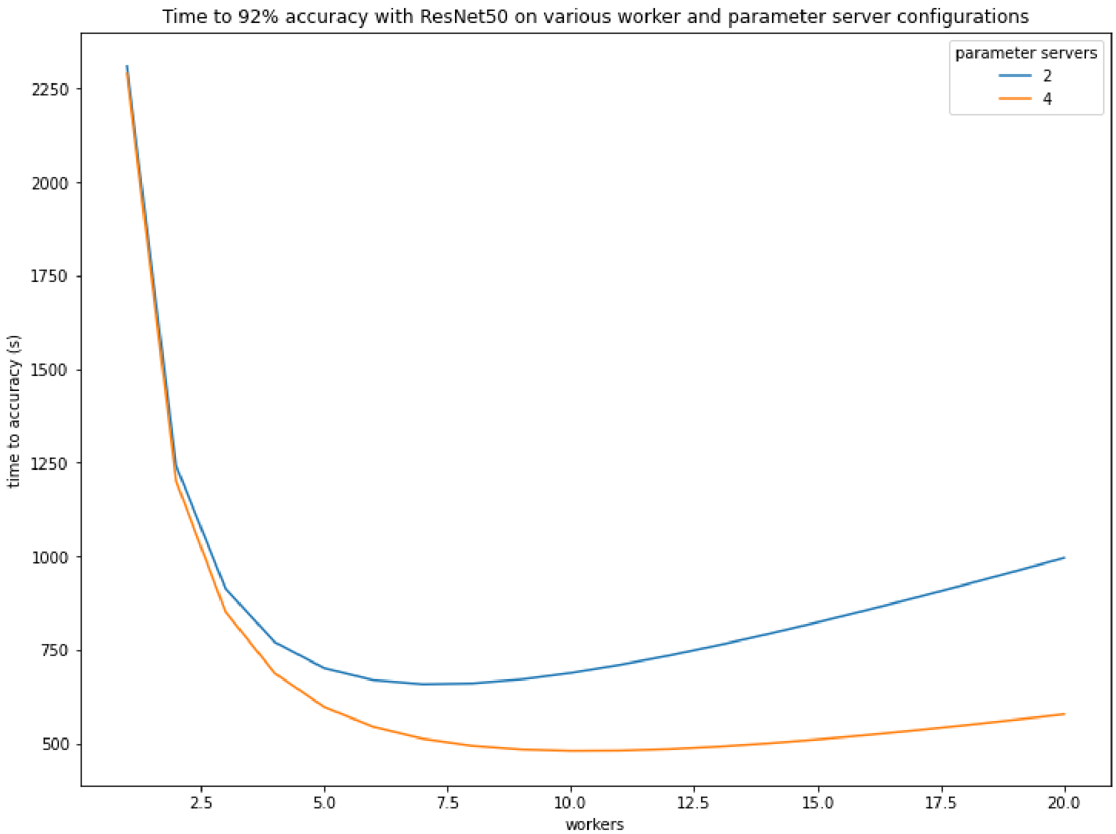
	parameter servers	workers	time to accuracy (s)
0	2	1	2309.280
1	2	2	1241.340
2	2	3	913.240
3	2	4	770.100
4	2	5	700.944

[46]:

```python
fig, ax = plt.subplots(figsize=(12, 9))
df['parameter servers'] = df['parameter servers'].astype('str')
sns.lineplot(x='workers', y='time to accuracy (s)', hue='parameter servers',␣
 ↪data=df)
plt.title('Time to 92% accuracy with ResNet50 on various worker and parameter␣
 ↪server configurations')
```

[46]:

```
Text(0.5, 1.0, 'Time to 92% accuracy with ResNet50 on various worker and
parameter server configurations')
```


Time to 92% accuracy with ResNet50 on various worker and parameter server configurations
parameter servers
2
4
time to accuracy (s)
2250
2000
1750
1500
1250
1000
750
500
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
workers