MP3_P1A_Introduction_1.nbconvert

October 19, 2020

1 Assignment 3 Part 1A Introduction: Multi-label Image Classification

[1]:
```
import os
import numpy as np
import torch
import torch.nn as nn
import torchvision

from torchvision import transforms
from sklearn.metrics import average_precision_score
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt
from kaggle_submission import output_submission_csv
from classifier import SimpleClassifier, Classifier#, AlexNet
from voc_dataloader import VocDataset, VOC_CLASSES

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

2 Multi-label Classification

In this assignment, you train a classifier to do multi-label classificaton on the PASCAL VOC 2007 dataset. The dataset has 20 different class which can appear in any given image. Your classifier will predict whether each class appears in an image. This task is slightly different from exclusive multiclass classification like the ImageNet competition where only a single most appropriate class is predicted for an image.

2.1 Part 1A

You will use this notebook to warm up with pytorch and the code+dataset that we will use for assignment3.

2.1.1 What to do

In part 1A, You are asked to run below experiments. You don't need to change hyperparameters for this Part 1A's experiments. (the following code provides everything that you will need.) 1. to

train a simple network (defined in `classifiers.py`) 2. to train the AlexNet (PyTorch built-in) - from scratch - finetuning AlexNet pretrained on ImageNet

2.1.2 What to submit

We ask you to run the following code and report the results in your homework submission. You may want to leverage this part 1A get yourself familiar with PyTorch.

You will the need the numbers and plots this notebook outputs for reports, but you are not required to submit this notebook as a printed pdf.

2.2 Reading Pascal Data

2.2.1 Loading Training Data

In the following cell we will load the training data and also apply some transforms to the data.

[2]:
```
# Transforms applied to the training data
normalize = transforms.Normalize(mean=[0.485, 0.456, 0.406],
                                 std= [0.229, 0.224, 0.225])

train_transform = transforms.Compose([
            transforms.Resize(227),
            transforms.CenterCrop(227),
            transforms.ToTensor(),
            normalize
        ])
```

[3]:
```
ds_train = VocDataset('VOCdevkit_2007/VOC2007/','train',train_transform)
```

2.2.2 Loading Validation Data

We will load the test data for the PASCAL VOC 2007 dataset. Do **NOT** add data augmentation transforms to validation data.

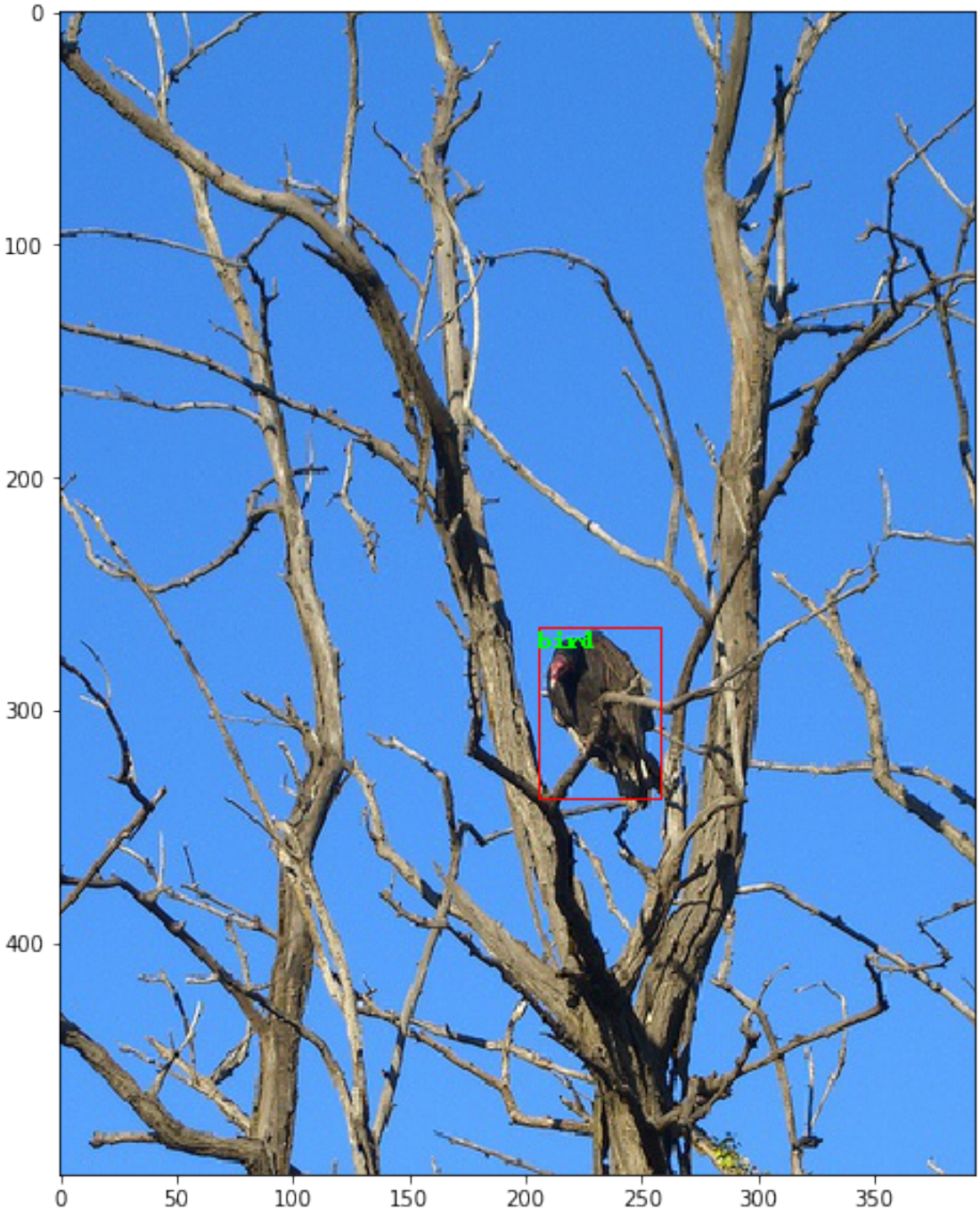
[4]:
```
# Transforms applied to the testing data
test_transform = transforms.Compose([
            transforms.Resize(227),
            transforms.CenterCrop(227),
            transforms.ToTensor(),
            normalize,
        ])
```

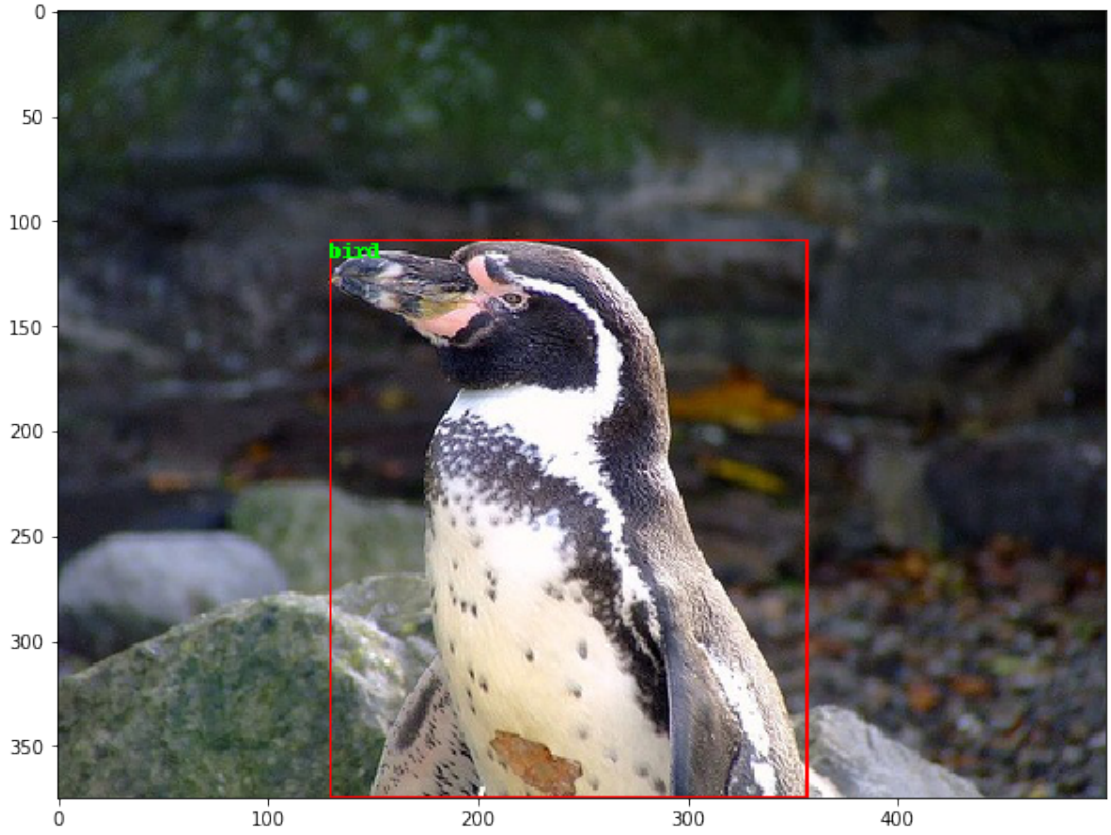
[5]:
```
ds_val = VocDataset('VOCdevkit_2007/VOC2007/','val',test_transform)
```

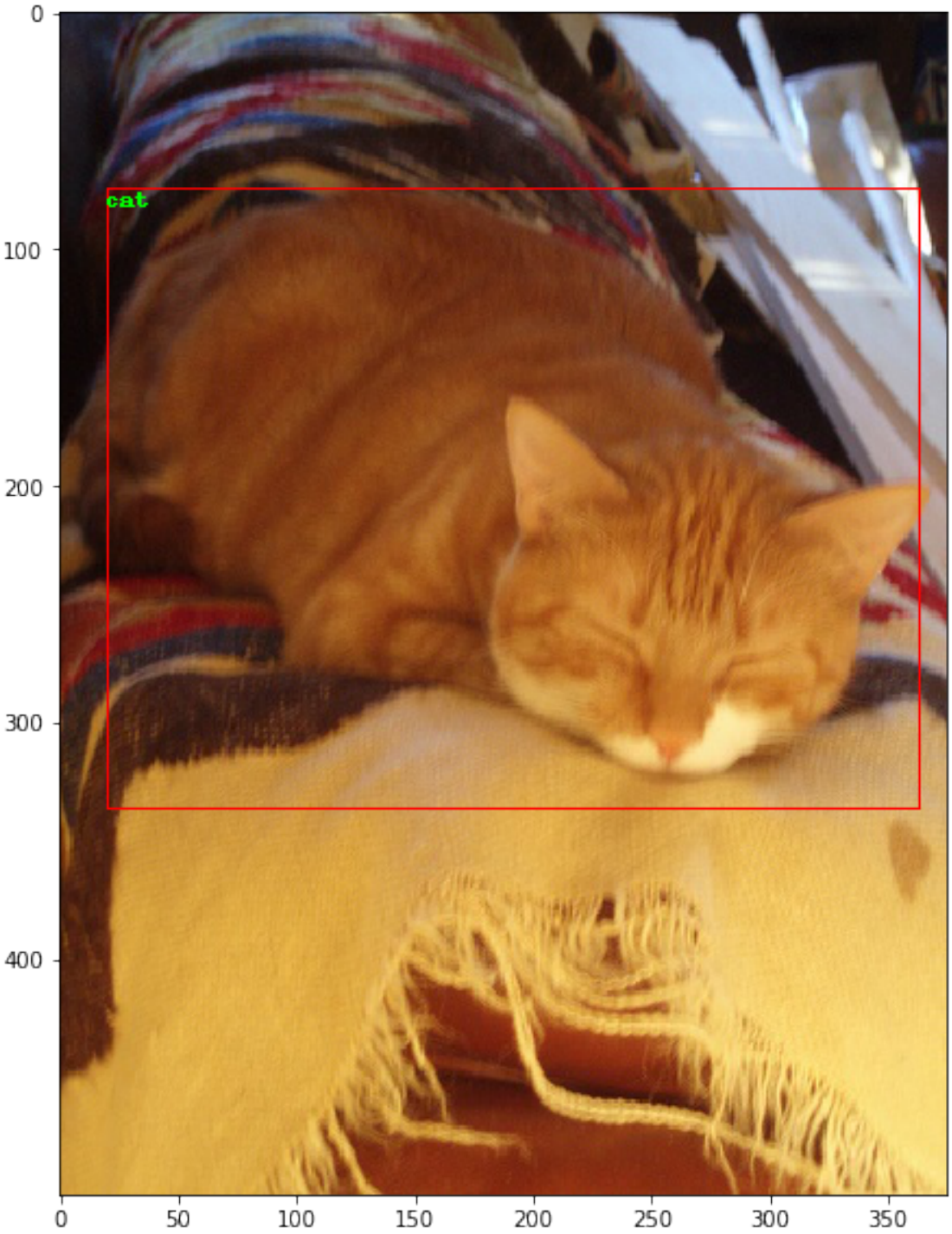
2.2.3 Visualizing the Data

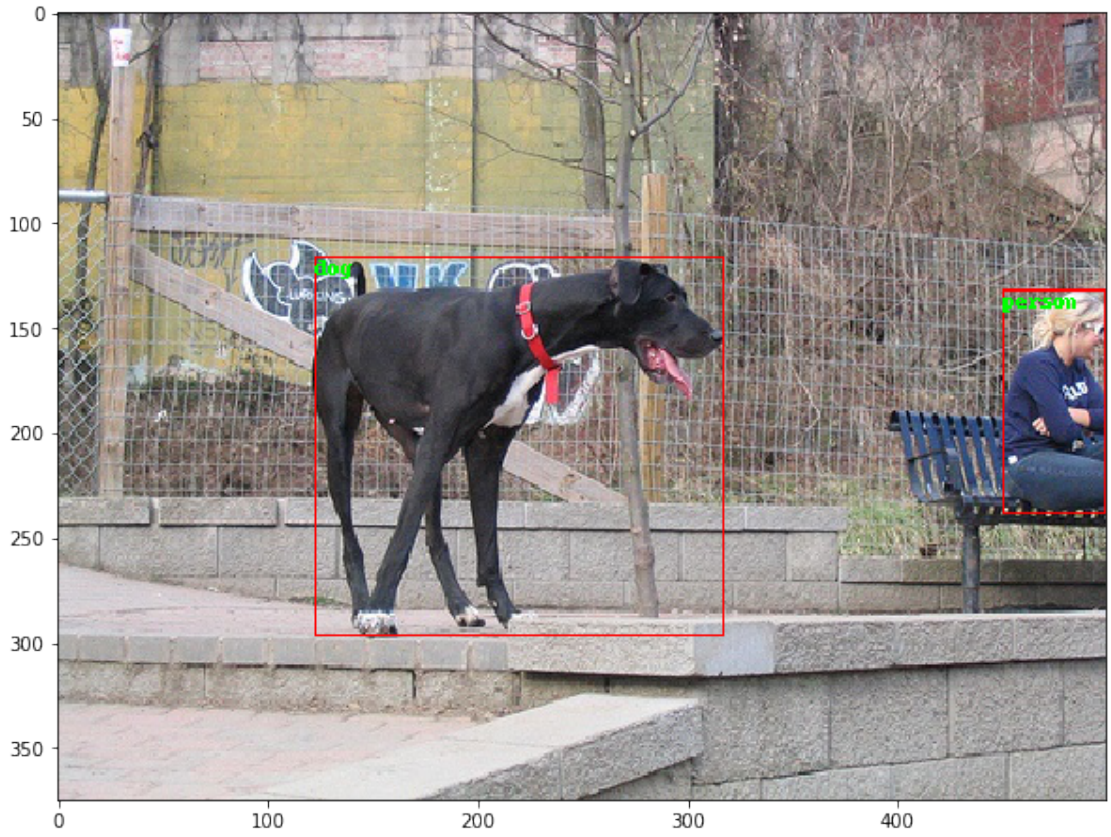
PASCAL VOC has bounding box annotations in addition to class labels. Use the following code to visualize some random examples and corresponding annotations from the train set.

[6]:
```
for i in range(5):
    idx = np.random.randint(0, len(ds_train.names)+1)
    _imgpath = os.path.join('VOCdevkit_2007/VOC2007/', 'JPEGImages', ds_train.
↪names[idx]+'.jpg')
    img = Image.open(_imgpath).convert('RGB')
    draw = ImageDraw.Draw(img)
    for j in range(len(ds_train.box_indices[idx])):
        obj = ds_train.box_indices[idx][j]
        draw.rectangle(list(obj), outline=(255,0,0))
        draw.text(list(obj[0:2]), ds_train.classes[ds_train.
↪label_order[idx][j]], fill=(0,255,0))
    plt.figure(figsize = (10,10))
    plt.imshow(np.array(img))
```


bird
0
100
200
300
400
0
50
100
150
200
250
300
350


bird
0
50
100
150
200
250
300
350
0
100
200
300
400


cat
0
100
200
300
400
0
50
100
150
200
250
300
350



3 Classification

[7]:
```
# declare what device to use: gpu/cpu
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(device)
```

```
cuda:0
```

[8]:
```
train_loader = torch.utils.data.DataLoader(dataset=ds_train,
                                           batch_size=50,
                                           shuffle=True,
                                           num_workers=1)
```

[9]:
```
val_loader = torch.utils.data.DataLoader(dataset=ds_val,
                                           batch_size=50,
                                           shuffle=True,
                                           num_workers=1)
```

[10]:
```
def train_classifier(train_loader, classifier, criterion, optimizer):
    classifier.train()
    loss_ = 0.0
    losses = []
    for i, (images, labels, _) in enumerate(train_loader):
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        logits = classifier(images)
        loss = criterion(logits, labels)
        loss.backward()
        optimizer.step()
        losses.append(loss)
    return torch.stack(losses).mean().item()
```

[11]:
```
def test_classifier(test_loader, classifier, criterion, print_ind_classes=True,␣
↪print_total=True):
    classifier.eval()
    losses = []
    with torch.no_grad():
        y_true = np.zeros((0,21))
        y_score = np.zeros((0,21))
        for i, (images, labels, _) in enumerate(test_loader):
            images, labels = images.to(device), labels.to(device)
            logits = classifier(images)
            y_true = np.concatenate((y_true, labels.cpu().numpy()), axis=0)
            y_score = np.concatenate((y_score, logits.cpu().numpy()), axis=0)
            loss = criterion(logits, labels)
            losses.append(loss.item())
        aps = []
        # ignore first class which is background
        for i in range(1, y_true.shape[1]):
            ap = average_precision_score(y_true[:, i], y_score[:, i])
            if print_ind_classes:
                print('-------  Class: {:<12}     AP: {:>8.4f}  -------'.
↪format(VOC_CLASSES[i], ap))
            aps.append(ap)

        mAP = np.mean(aps)
        test_loss = np.mean(losses)
        if print_total:
            print('mAP: {0:.4f}'.format(mAP))
            print('Avg loss: {}'.format(test_loss))

    return mAP, test_loss, aps
```

[12]:
```
# plot functions
def plot_losses(train, val, test_frequency, num_epochs):
```

```
    plt.plot(train, label="train")
    indices = [i for i in range(num_epochs) if ((i+1)%test_frequency == 0 or i
↪==0)]
    plt.plot(indices, val, label="val")
    plt.title("Loss Plot")
    plt.ylabel("Loss")
    plt.xlabel("Epoch")
    plt.legend()
    plt.show()

def plot_mAP(train, val, test_frequency, num_epochs):
    indices = [i for i in range(num_epochs) if ((i+1)%test_frequency == 0 or i
↪==0)]
    plt.plot(indices, train, label="train")
    plt.plot(indices, val, label="val")
    plt.title("mAP Plot")
    plt.ylabel("mAP")
    plt.xlabel("Epoch")
    plt.legend()
    plt.show()
```

3.1 Training the network

The simple network you are given as is will allow you to reach around 0.15-0.2 mAP. In this project, you will find ways to design a better network. Save plots and final test mAP scores as you will be adding these to the writeup.

[13]:
```
def train(classifier, num_epochs, train_loader, val_loader, criterion,
↪optimizer, test_frequency=5):
    train_losses = []
    train_mAPs = []
    val_losses = []
    val_mAPs = []

    for epoch in range(1,num_epochs+1):
        print("Starting epoch number " + str(epoch))
        train_loss = train_classifier(train_loader, classifier, criterion,
↪optimizer)
        train_losses.append(train_loss)
        print("Loss for Training on Epoch " +str(epoch) + " is "+
↪str(train_loss))
        if(epoch%test_frequency==0 or epoch==1):
            mAP_train, _, _ = test_classifier(train_loader, classifier,
↪criterion, False, False)
            train_mAPs.append(mAP_train)
```

```
            mAP_val, val_loss, _ = test_classifier(val_loader, classifier,␣
 ↪criterion)
            print('Evaluating classifier')
            print("Mean Precision Score for Testing on Epoch " +str(epoch) + "␣
 ↪is "+ str(mAP_val))
            val_losses.append(val_loss)
            val_mAPs.append(mAP_val)

    return classifier, train_losses, val_losses, train_mAPs, val_mAPs
```

[14]:
```
classifier = SimpleClassifier().to(device)
# You can can use this function to reload a network you have already saved␣
 ↪previously
#classifier.load_state_dict(torch.load('voc_classifier.pth'))
```

[15]:
```
criterion = nn.MultiLabelSoftMarginLoss()
optimizer = torch.optim.SGD(classifier.parameters(), lr=0.01, momentum=0.9)
```

[16]:
```
# Training the Classifier
num_epochs = 20
test_frequency = 5

classifier, train_losses, val_losses, train_mAPs, val_mAPs = train(classifier,␣
 ↪num_epochs, train_loader, val_loader, criterion, optimizer, test_frequency)
```

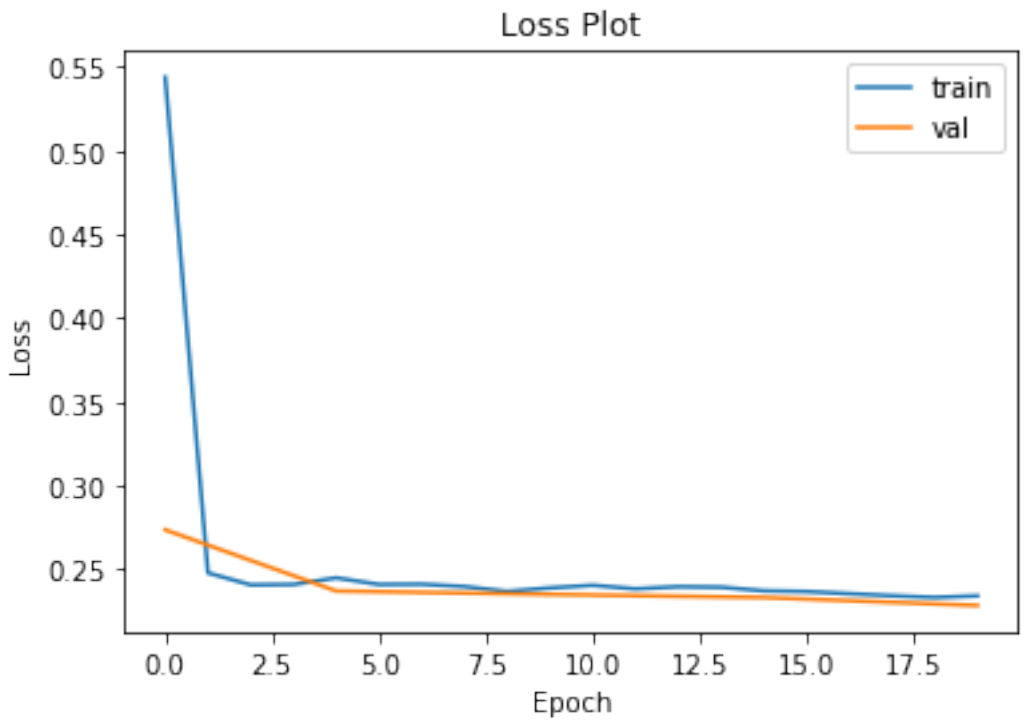
```
Starting epoch number 1
Loss for Training on Epoch 1 is 0.5439279079437256
-------  Class: aeroplane        AP:   0.1492  -------
-------  Class: bicycle          AP:   0.0513  -------
-------  Class: bird             AP:   0.1055  -------
-------  Class: boat             AP:   0.1174  -------
-------  Class: bottle           AP:   0.0393  -------
-------  Class: bus              AP:   0.0316  -------
-------  Class: car              AP:   0.1154  -------
-------  Class: cat              AP:   0.0791  -------
-------  Class: chair            AP:   0.1124  -------
-------  Class: cow              AP:   0.0298  -------
-------  Class: diningtable      AP:   0.0470  -------
-------  Class: dog              AP:   0.1108  -------
-------  Class: horse            AP:   0.0495  -------
-------  Class: motorbike        AP:   0.0468  -------
-------  Class: person           AP:   0.4366  -------
-------  Class: pottedplant      AP:   0.0443  -------
-------  Class: sheep            AP:   0.0237  -------
-------  Class: sofa             AP:   0.0838  -------
-------  Class: train            AP:   0.0363  -------
-------  Class: tvmonitor        AP:   0.0502  -------
```

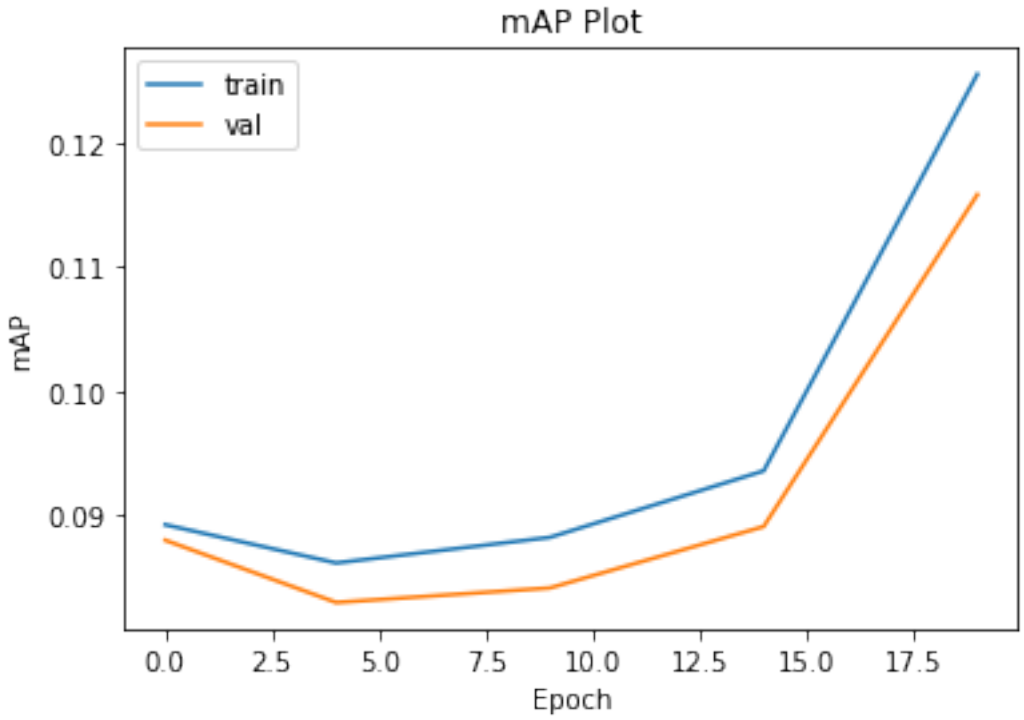
```
mAP: 0.0880
Avg loss: 0.27371764679749805
Evaluating classifier
Mean Precision Score for Testing on Epoch 1 is 0.08801461347977 26
Starting epoch number 2
Loss for Training on Epoch 2 is 0.24799811840057373
Starting epoch number 3
Loss for Training on Epoch 3 is 0.24086320400238037
Starting epoch number 4
Loss for Training on Epoch 4 is 0.2410968840122223
Starting epoch number 5
Loss for Training on Epoch 5 is 0.24504314363002777
-------  Class: aeroplane        AP:   0.0861  -------
-------  Class: bicycle          AP:   0.0455  -------
-------  Class: bird             AP:   0.1038  -------
-------  Class: boat             AP:   0.0756  -------
-------  Class: bottle           AP:   0.0446  -------
-------  Class: bus              AP:   0.0284  -------
-------  Class: car              AP:   0.1113  -------
-------  Class: cat              AP:   0.1066  -------
-------  Class: chair            AP:   0.1316  -------
-------  Class: cow              AP:   0.0305  -------
-------  Class: diningtable      AP:   0.0553  -------
-------  Class: dog              AP:   0.1298  -------
-------  Class: horse            AP:   0.0486  -------
-------  Class: motorbike        AP:   0.0414  -------
-------  Class: person           AP:   0.3543  -------
-------  Class: pottedplant      AP:   0.0466  -------
-------  Class: sheep            AP:   0.0263  -------
-------  Class: sofa             AP:   0.1045  -------
-------  Class: train            AP:   0.0353  -------
-------  Class: tvmonitor        AP:   0.0539  -------
mAP: 0.0830
Avg loss: 0.23721890998821632
Evaluating classifier
Mean Precision Score for Testing on Epoch 5 is 0.08300479102381039
Starting epoch number 6
Loss for Training on Epoch 6 is 0.24111507833003998
Starting epoch number 7
Loss for Training on Epoch 7 is 0.2412501573562622
Starting epoch number 8
Loss for Training on Epoch 8 is 0.23983041942119598
Starting epoch number 9
Loss for Training on Epoch 9 is 0.2369721233844757
Starting epoch number 10
Loss for Training on Epoch 10 is 0.2390352338552475
-------  Class: aeroplane        AP:   0.0745  -------
-------  Class: bicycle          AP:   0.0440  -------
```

```
-------  Class: bird             AP:   0.1023  -------
-------  Class: boat             AP:   0.0600  -------
-------  Class: bottle           AP:   0.0456  -------
-------  Class: bus              AP:   0.0278  -------
-------  Class: car              AP:   0.1143  -------
-------  Class: cat              AP:   0.1177  -------
-------  Class: chair            AP:   0.1397  -------
-------  Class: cow              AP:   0.0315  -------
-------  Class: diningtable      AP:   0.0585  -------
-------  Class: dog              AP:   0.1352  -------
-------  Class: horse            AP:   0.0501  -------
-------  Class: motorbike        AP:   0.0411  -------
-------  Class: person           AP:   0.3610  -------
-------  Class: pottedplant      AP:   0.0480  -------
-------  Class: sheep            AP:   0.0297  -------
-------  Class: sofa             AP:   0.1106  -------
-------  Class: train            AP:   0.0358  -------
-------  Class: tvmonitor        AP:   0.0560  -------
mAP: 0.0842
Avg loss: 0.23516886199221892
Evaluating classifier
Mean Precision Score for Testing on Epoch 10 is 0.08417308886982525
Starting epoch number 11
Loss for Training on Epoch 11 is 0.24059483408927917
Starting epoch number 12
Loss for Training on Epoch 12 is 0.23849020898342133
Starting epoch number 13
Loss for Training on Epoch 13 is 0.23976543545722961
Starting epoch number 14
Loss for Training on Epoch 14 is 0.2394586205482483
Starting epoch number 15
Loss for Training on Epoch 15 is 0.23732246458530426
-------  Class: aeroplane        AP:   0.0859  -------
-------  Class: bicycle          AP:   0.0446  -------
-------  Class: bird             AP:   0.1060  -------
-------  Class: boat             AP:   0.0545  -------
-------  Class: bottle           AP:   0.0467  -------
-------  Class: bus              AP:   0.0285  -------
-------  Class: car              AP:   0.1269  -------
-------  Class: cat              AP:   0.1185  -------
-------  Class: chair            AP:   0.1529  -------
-------  Class: cow              AP:   0.0331  -------
-------  Class: diningtable      AP:   0.0633  -------
-------  Class: dog              AP:   0.1312  -------
-------  Class: horse            AP:   0.0531  -------
-------  Class: motorbike        AP:   0.0424  -------
-------  Class: person           AP:   0.3945  -------
-------  Class: pottedplant      AP:   0.0508  -------
```

```
-------  Class: sheep            AP:   0.0410  -------
-------  Class: sofa             AP:   0.1130  -------
-------  Class: train            AP:   0.0385  -------
-------  Class: tvmonitor        AP:   0.0568  -------
mAP: 0.0891
Avg loss: 0.23332357172872506
Evaluating classifier
Mean Precision Score for Testing on Epoch 15 is 0.08911148094568158
Starting epoch number 16
Loss for Training on Epoch 16 is 0.2368842512369156
Starting epoch number 17
Loss for Training on Epoch 17 is 0.23557999730110168
Starting epoch number 18
Loss for Training on Epoch 18 is 0.23437541723251343
Starting epoch number 19
Loss for Training on Epoch 19 is 0.23336078226566315
Starting epoch number 20
Loss for Training on Epoch 20 is 0.23432353138923645
-------  Class: aeroplane        AP:   0.2138  -------
-------  Class: bicycle          AP:   0.0576  -------
-------  Class: bird             AP:   0.1294  -------
-------  Class: boat             AP:   0.1173  -------
-------  Class: bottle           AP:   0.0618  -------
-------  Class: bus              AP:   0.0552  -------
-------  Class: car              AP:   0.1758  -------
-------  Class: cat              AP:   0.1182  -------
-------  Class: chair            AP:   0.1766  -------
-------  Class: cow              AP:   0.0392  -------
-------  Class: diningtable      AP:   0.0909  -------
-------  Class: dog              AP:   0.1269  -------
-------  Class: horse            AP:   0.0603  -------
-------  Class: motorbike        AP:   0.0464  -------
-------  Class: person           AP:   0.4821  -------
-------  Class: pottedplant      AP:   0.0538  -------
-------  Class: sheep            AP:   0.0408  -------
-------  Class: sofa             AP:   0.1204  -------
-------  Class: train            AP:   0.0926  -------
-------  Class: tvmonitor        AP:   0.0572  -------
mAP: 0.1158
Avg loss: 0.22855493046489417
Evaluating classifier
Mean Precision Score for Testing on Epoch 20 is 0.11581629322238615
```

[17]:
```
# Compare train and validation metrics
plot_losses(train_losses, val_losses, test_frequency, num_epochs)
plot_mAP(train_mAPs, val_mAPs, test_frequency, num_epochs)
```


Loss Plot
train
val
Loss
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch


mAP Plot
train
val
mAP
0.12
0.11
0.10
0.09
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

[18]:
```
# Save the clssifier network
# Suggestion: you can save checkpoints of your network during training and␣
 ↪reload them later
torch.save(classifier.state_dict(), './voc_simple_classifier.pth')
```

4 Evaluate on test set

[19]:
```
ds_test = VocDataset('VOCdevkit_2007/VOC2007test/','test', test_transform)

test_loader = torch.utils.data.DataLoader(dataset=ds_test,
                                                batch_size=50,
                                                shuffle=False,
                                                num_workers=1)

# Transforms applied to the testing data
test_transform = transforms.Compose([
            transforms.Resize(227),
            transforms.CenterCrop(227),
            transforms.ToTensor(),
            normalize,
        ])

mAP_test, test_loss, test_aps = test_classifier(test_loader, classifier,␣
 ↪criterion)
```

```
-------  Class: aeroplane        AP:   0.1757  -------
-------  Class: bicycle          AP:   0.0535  -------
-------  Class: bird             AP:   0.1020  -------
-------  Class: boat             AP:   0.1069  -------
-------  Class: bottle           AP:   0.0565  -------
-------  Class: bus              AP:   0.0493  -------
-------  Class: car              AP:   0.1842  -------
-------  Class: cat              AP:   0.1158  -------
-------  Class: chair            AP:   0.1562  -------
-------  Class: cow              AP:   0.0473  -------
-------  Class: diningtable      AP:   0.0665  -------
-------  Class: dog              AP:   0.1348  -------
-------  Class: horse            AP:   0.0573  -------
-------  Class: motorbike        AP:   0.0455  -------
-------  Class: person           AP:   0.4929  -------
-------  Class: pottedplant      AP:   0.0504  -------
-------  Class: sheep            AP:   0.0595  -------
-------  Class: sofa             AP:   0.1157  -------
-------  Class: train            AP:   0.1170  -------
-------  Class: tvmonitor        AP:   0.0565  -------
mAP: 0.1122
```

```
Avg loss: 0.2248046140372753
```

[20]:
```
output_submission_csv('my_solution.csv', test_aps)
```

5 AlexNet Baselines (From Scratch)

AlexNet was one of the earliest deep learning models to have success in classification. In this section we will be running classification with AlexNet as a baseline. Furthermore, we will run an ImageNet-pretrained AlexNet to observe the impact of well-trained features. Save plots and final test mAP scores as you will be adding these to the writeup.

5.1 Running AlexNet

In this section, we train AlexNet from scratch using the same hyperparameters as our previous experiment.

[21]:
```
num_epochs = 20
test_frequency = 5

# Change classifier to AlexNet
classifier = torchvision.models.alexnet(pretrained=False)
classifier.classifier._modules['6'] = nn.Linear(4096, 21)
classifier = classifier.to(device)

criterion = nn.MultiLabelSoftMarginLoss()

optimizer = torch.optim.SGD(classifier.parameters(), lr=0.01, momentum=0.9)
```

[22]:
```
classifier, train_losses, val_losses, train_mAPs, val_mAPs = train(classifier,␣
↪num_epochs, train_loader, val_loader, criterion, optimizer, test_frequency)
```

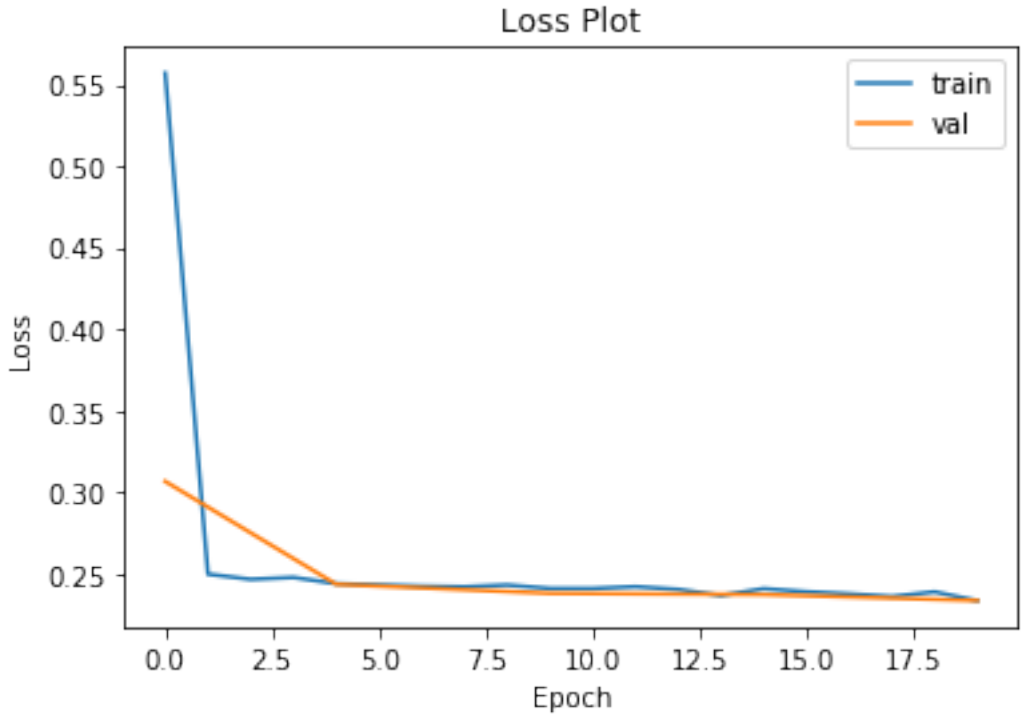
```
Starting epoch number 1
Loss for Training on Epoch 1 is 0.5570489168167114
-------  Class: aeroplane        AP:   0.1382  -------
-------  Class: bicycle          AP:   0.0430  -------
-------  Class: bird             AP:   0.1053  -------
-------  Class: boat             AP:   0.1274  -------
-------  Class: bottle           AP:   0.0376  -------
-------  Class: bus              AP:   0.0292  -------
-------  Class: car              AP:   0.1091  -------
-------  Class: cat              AP:   0.0995  -------
-------  Class: chair            AP:   0.1093  -------
-------  Class: cow              AP:   0.0361  -------
-------  Class: diningtable      AP:   0.0458  -------
-------  Class: dog              AP:   0.1130  -------
-------  Class: horse            AP:   0.0451  -------
-------  Class: motorbike        AP:   0.0339  -------
-------  Class: person           AP:   0.3249  -------
```

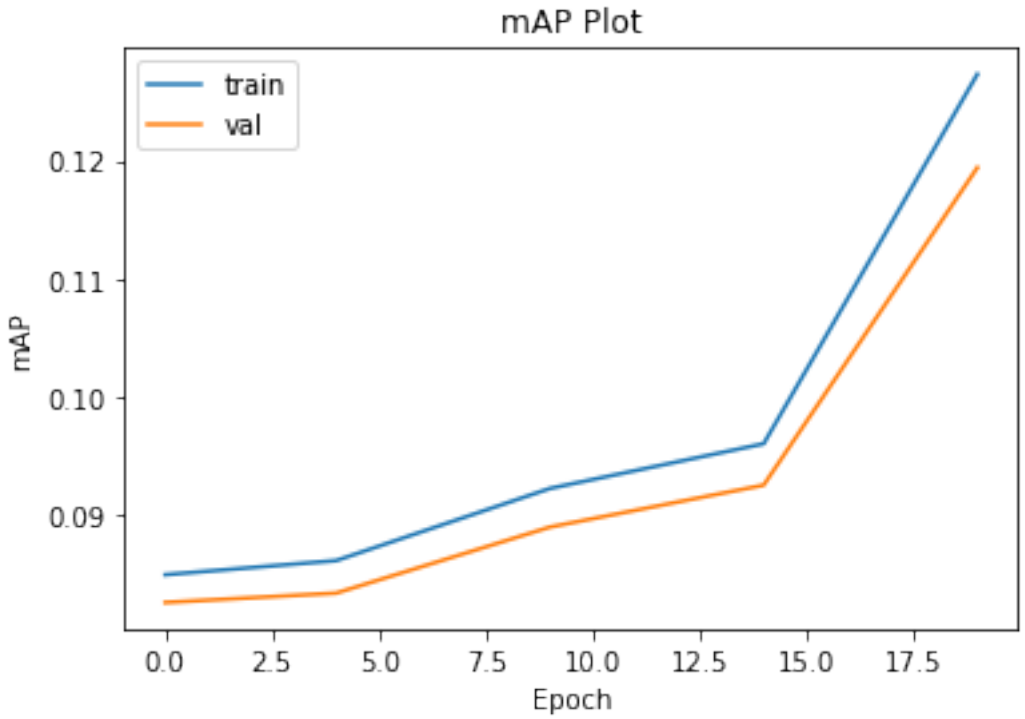
```
-------  Class: pottedplant      AP:   0.0409  -------
-------  Class: sheep            AP:   0.0379  -------
-------  Class: sofa             AP:   0.0885  -------
-------  Class: train            AP:   0.0405  -------
-------  Class: tvmonitor        AP:   0.0473  -------
mAP: 0.0826
Avg loss: 0.30657491087913513
Evaluating classifier
Mean Precision Score for Testing on Epoch 1 is 0.08262655529072144
Starting epoch number 2
Loss for Training on Epoch 2 is 0.24965687096118927
Starting epoch number 3
Loss for Training on Epoch 3 is 0.24649912118911743
Starting epoch number 4
Loss for Training on Epoch 4 is 0.2476574331521988
Starting epoch number 5
Loss for Training on Epoch 5 is 0.24382369220256805
-------  Class: aeroplane        AP:   0.0945  -------
-------  Class: bicycle          AP:   0.0463  -------
-------  Class: bird             AP:   0.1172  -------
-------  Class: boat             AP:   0.1018  -------
-------  Class: bottle           AP:   0.0414  -------
-------  Class: bus              AP:   0.0292  -------
-------  Class: car              AP:   0.1124  -------
-------  Class: cat              AP:   0.0929  -------
-------  Class: chair            AP:   0.1203  -------
-------  Class: cow              AP:   0.0358  -------
-------  Class: diningtable      AP:   0.0527  -------
-------  Class: dog              AP:   0.1176  -------
-------  Class: horse            AP:   0.0485  -------
-------  Class: motorbike        AP:   0.0371  -------
-------  Class: person           AP:   0.3513  -------
-------  Class: pottedplant      AP:   0.0459  -------
-------  Class: sheep            AP:   0.0414  -------
-------  Class: sofa             AP:   0.0938  -------
-------  Class: train            AP:   0.0403  -------
-------  Class: tvmonitor        AP:   0.0480  -------
mAP: 0.0834
Avg loss: 0.243446968349756
Evaluating classifier
Mean Precision Score for Testing on Epoch 5 is 0.08341850320114871
Starting epoch number 6
Loss for Training on Epoch 6 is 0.2430320680141449
Starting epoch number 7
Loss for Training on Epoch 7 is 0.2422732263803482
Starting epoch number 8
Loss for Training on Epoch 8 is 0.24190865457057953
Starting epoch number 9
```

```
Loss for Training on Epoch 9 is 0.24298124015331268
Starting epoch number 10
Loss for Training on Epoch 10 is 0.2408696413040161
-------  Class: aeroplane        AP:   0.0836  -------
-------  Class: bicycle          AP:   0.0487  -------
-------  Class: bird             AP:   0.1243  -------
-------  Class: boat             AP:   0.0913  -------
-------  Class: bottle           AP:   0.0413  -------
-------  Class: bus              AP:   0.0298  -------
-------  Class: car              AP:   0.1171  -------
-------  Class: cat              AP:   0.0922  -------
-------  Class: chair            AP:   0.1276  -------
-------  Class: cow              AP:   0.0365  -------
-------  Class: diningtable      AP:   0.0546  -------
-------  Class: dog              AP:   0.1200  -------
-------  Class: horse            AP:   0.0515  -------
-------  Class: motorbike        AP:   0.0395  -------
-------  Class: person           AP:   0.4411  -------
-------  Class: pottedplant      AP:   0.0494  -------
-------  Class: sheep            AP:   0.0475  -------
-------  Class: sofa             AP:   0.0935  -------
-------  Class: train            AP:   0.0412  -------
-------  Class: tvmonitor        AP:   0.0496  -------
mAP: 0.0890
Avg loss: 0.23791137483774447
Evaluating classifier
Mean Precision Score for Testing on Epoch 10 is 0.08900571633952556
Starting epoch number 11
Loss for Training on Epoch 11 is 0.24085105955600739
Starting epoch number 12
Loss for Training on Epoch 12 is 0.24197809398174286
Starting epoch number 13
Loss for Training on Epoch 13 is 0.24029192328453064
Starting epoch number 14
Loss for Training on Epoch 14 is 0.2365572601556778
Starting epoch number 15
Loss for Training on Epoch 15 is 0.24076105654239655
-------  Class: aeroplane        AP:   0.0848  -------
-------  Class: bicycle          AP:   0.0520  -------
-------  Class: bird             AP:   0.1327  -------
-------  Class: boat             AP:   0.0803  -------
-------  Class: bottle           AP:   0.0438  -------
-------  Class: bus              AP:   0.0312  -------
-------  Class: car              AP:   0.1278  -------
-------  Class: cat              AP:   0.0920  -------
-------  Class: chair            AP:   0.1425  -------
-------  Class: cow              AP:   0.0373  -------
-------  Class: diningtable      AP:   0.0606  -------
```

```
-------  Class: dog              AP:   0.1209  -------
-------  Class: horse            AP:   0.0549  -------
-------  Class: motorbike        AP:   0.0435  -------
-------  Class: person           AP:   0.4484  -------
-------  Class: pottedplant      AP:   0.0547  -------
-------  Class: sheep            AP:   0.0502  -------
-------  Class: sofa             AP:   0.0982  -------
-------  Class: train            AP:   0.0438  -------
-------  Class: tvmonitor        AP:   0.0518  -------
mAP: 0.0926
Avg loss: 0.2371828433345346
Evaluating classifier
Mean Precision Score for Testing on Epoch 15 is 0.092551745332285
Starting epoch number 16
Loss for Training on Epoch 16 is 0.2388356626033783
Starting epoch number 17
Loss for Training on Epoch 17 is 0.2375527024269104
Starting epoch number 18
Loss for Training on Epoch 18 is 0.2359590083360672
Starting epoch number 19
Loss for Training on Epoch 19 is 0.23879577219486237
Starting epoch number 20
Loss for Training on Epoch 20 is 0.23340480029582977
-------  Class: aeroplane        AP:   0.2291  -------
-------  Class: bicycle          AP:   0.0613  -------
-------  Class: bird             AP:   0.1364  -------
-------  Class: boat             AP:   0.1809  -------
-------  Class: bottle           AP:   0.0589  -------
-------  Class: bus              AP:   0.0441  -------
-------  Class: car              AP:   0.1714  -------
-------  Class: cat              AP:   0.1006  -------
-------  Class: chair            AP:   0.1746  -------
-------  Class: cow              AP:   0.0399  -------
-------  Class: diningtable      AP:   0.0879  -------
-------  Class: dog              AP:   0.1312  -------
-------  Class: horse            AP:   0.0580  -------
-------  Class: motorbike        AP:   0.0465  -------
-------  Class: person           AP:   0.5024  -------
-------  Class: pottedplant      AP:   0.0672  -------
-------  Class: sheep            AP:   0.0485  -------
-------  Class: sofa             AP:   0.1202  -------
-------  Class: train            AP:   0.0756  -------
-------  Class: tvmonitor        AP:   0.0536  -------
mAP: 0.1194
Avg loss: 0.23350198099426195
Evaluating classifier
Mean Precision Score for Testing on Epoch 20 is 0.11941861495429547
```

[23]:
```
plot_losses(train_losses, val_losses, test_frequency, num_epochs)
plot_mAP(train_mAPs, val_mAPs, test_frequency, num_epochs)
```





[24]:
```
mAP_test, test_loss, test_aps = test_classifier(test_loader, classifier,␣
↪criterion)
print("Test mAP: ", mAP_test)
```

```
-------  Class: aeroplane        AP:   0.2001  -------
-------  Class: bicycle          AP:   0.0579  -------
-------  Class: bird             AP:   0.1106  -------
-------  Class: boat             AP:   0.1395  -------
-------  Class: bottle           AP:   0.0568  -------
-------  Class: bus              AP:   0.0414  -------
-------  Class: car              AP:   0.1696  -------
-------  Class: cat              AP:   0.1009  -------
-------  Class: chair            AP:   0.1592  -------
-------  Class: cow              AP:   0.0478  -------
-------  Class: diningtable      AP:   0.0700  -------
-------  Class: dog              AP:   0.1367  -------
-------  Class: horse            AP:   0.0553  -------
-------  Class: motorbike        AP:   0.0417  -------
-------  Class: person           AP:   0.5110  -------
-------  Class: pottedplant      AP:   0.0569  -------
-------  Class: sheep            AP:   0.0651  -------
-------  Class: sofa             AP:   0.1209  -------
-------  Class: train            AP:   0.0885  -------
-------  Class: tvmonitor        AP:   0.0537  -------
mAP: 0.1142
Avg loss: 0.23103498727083205
Test mAP:  0.11418412719617296
```

You should notice somewhat poor performance. You could try running AlexNet with an Adam optimizer instead with learning rate 1e-4 to see if that makes a difference. This experiment is not required for the writeup, but it may show you the importance of a good learning rate and optimizer.

5.2 Pretrained AlexNet

Here we look at the impact of pretrained features. This model's weights were trained on ImageNet, which is a much larger dataset. How do pretrained features perform on VOC? Why do you think there is such a large difference in performance?

[25]:
```
num_epochs = 20
test_frequency = 5

# Load Pretrained AlexNet
classifier = torchvision.models.alexnet(pretrained=True)
classifier.classifier._modules['6'] = nn.Linear(4096, 21)
classifier = classifier.to(device)
optimizer = torch.optim.SGD(classifier.parameters(), lr=0.01, momentum=0.9)
```

[26]:
```
classifier, train_losses, val_losses, train_mAPs, val_mAPs = train(classifier,␣
↪num_epochs, train_loader, val_loader, criterion, optimizer, test_frequency)
```

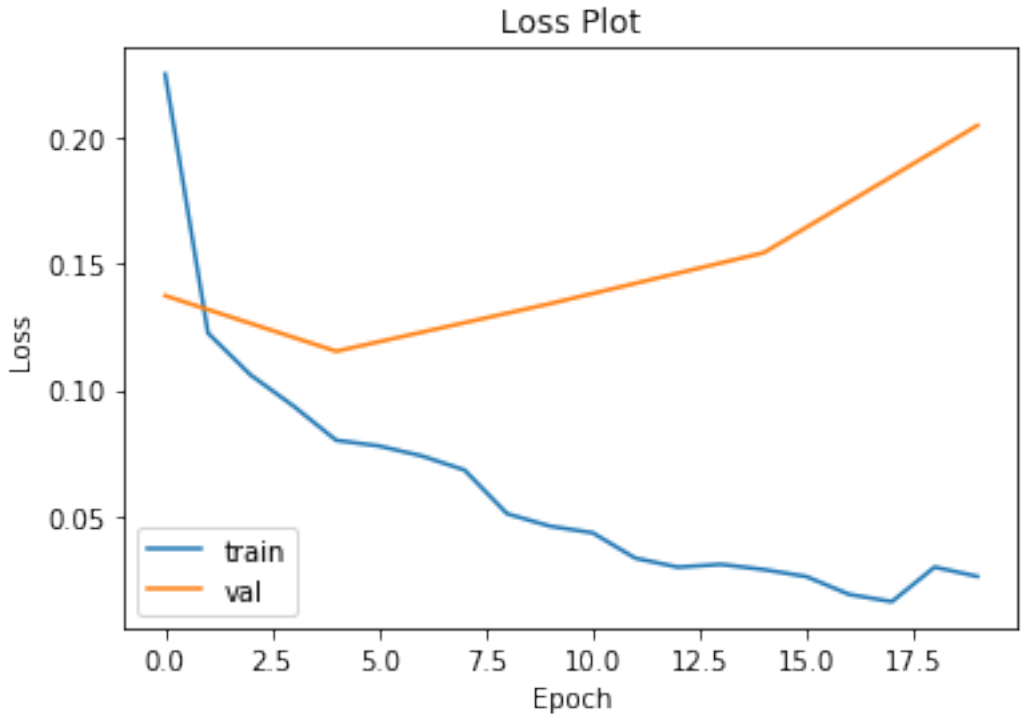
```
Starting epoch number 1
Loss for Training on Epoch 1 is 0.2252361923456192
-------  Class: aeroplane        AP:   0.7824  -------
-------  Class: bicycle          AP:   0.5411  -------
-------  Class: bird             AP:   0.8137  -------
-------  Class: boat             AP:   0.6326  -------
-------  Class: bottle           AP:   0.3017  -------
-------  Class: bus              AP:   0.5275  -------
-------  Class: car              AP:   0.7794  -------
-------  Class: cat              AP:   0.7153  -------
-------  Class: chair            AP:   0.5954  -------
-------  Class: cow              AP:   0.2724  -------
-------  Class: diningtable      AP:   0.4843  -------
-------  Class: dog              AP:   0.6379  -------
-------  Class: horse            AP:   0.6265  -------
-------  Class: motorbike        AP:   0.7089  -------
-------  Class: person           AP:   0.9019  -------
-------  Class: pottedplant      AP:   0.3515  -------
-------  Class: sheep            AP:   0.2086  -------
-------  Class: sofa             AP:   0.4565  -------
-------  Class: train            AP:   0.8223  -------
-------  Class: tvmonitor        AP:   0.5432  -------
mAP: 0.5852
Avg loss: 0.13741200285799363
Evaluating classifier
Mean Precision Score for Testing on Epoch 1 is 0.5851549360899801
Starting epoch number 2
Loss for Training on Epoch 2 is 0.1226029172539711
Starting epoch number 3
Loss for Training on Epoch 3 is 0.10597190260887146
Starting epoch number 4
Loss for Training on Epoch 4 is 0.09369628131389618
Starting epoch number 5
Loss for Training on Epoch 5 is 0.08010298013687134
-------  Class: aeroplane        AP:   0.8580  -------
-------  Class: bicycle          AP:   0.7248  -------
-------  Class: bird             AP:   0.8553  -------
-------  Class: boat             AP:   0.7782  -------
-------  Class: bottle           AP:   0.4114  -------
-------  Class: bus              AP:   0.6188  -------
-------  Class: car              AP:   0.8194  -------
-------  Class: cat              AP:   0.7910  -------
-------  Class: chair            AP:   0.6239  -------
-------  Class: cow              AP:   0.4844  -------
```

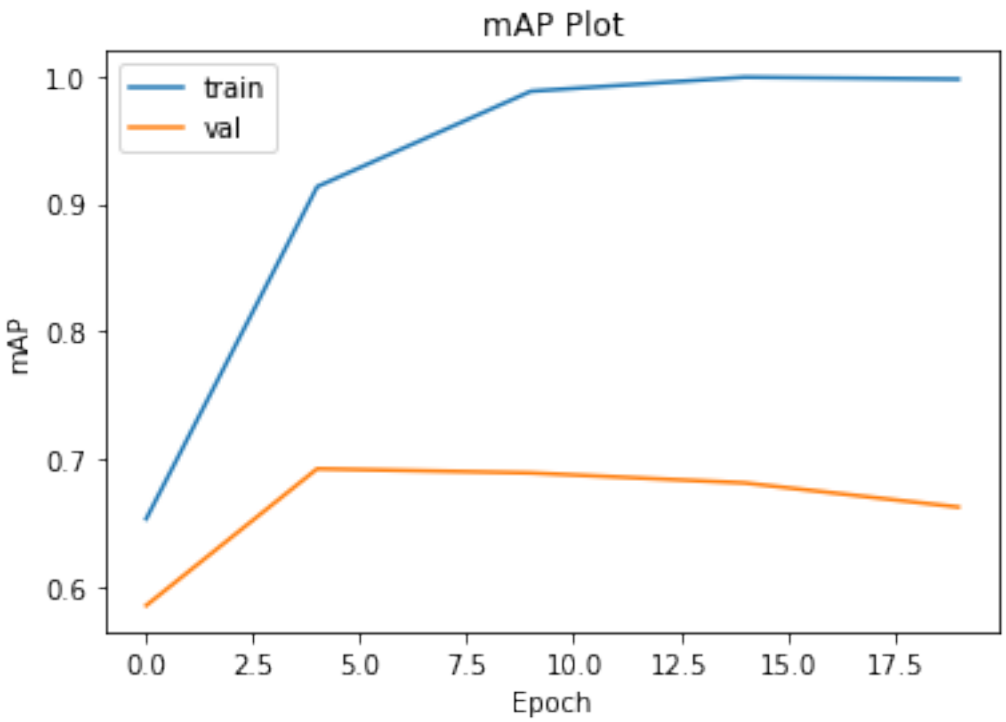
```
-------  Class: diningtable      AP:   0.6018  -------
-------  Class: dog              AP:   0.7409  -------
-------  Class: horse            AP:   0.7250  -------
-------  Class: motorbike        AP:   0.8055  -------
-------  Class: person           AP:   0.9139  -------
-------  Class: pottedplant      AP:   0.4691  -------
-------  Class: sheep            AP:   0.5601  -------
-------  Class: sofa             AP:   0.5679  -------
-------  Class: train            AP:   0.8608  -------
-------  Class: tvmonitor        AP:   0.6367  -------
mAP: 0.6923
Avg loss: 0.11538420646798377
Evaluating classifier
Mean Precision Score for Testing on Epoch 5 is 0.6923415540683364
Starting epoch number 6
Loss for Training on Epoch 6 is 0.07785788178443909
Starting epoch number 7
Loss for Training on Epoch 7 is 0.07391707599163055
Starting epoch number 8
Loss for Training on Epoch 8 is 0.06834276020526886
Starting epoch number 9
Loss for Training on Epoch 9 is 0.05098617821931839
Starting epoch number 10
Loss for Training on Epoch 10 is 0.04607788100838661
-------  Class: aeroplane        AP:   0.8754  -------
-------  Class: bicycle          AP:   0.7110  -------
-------  Class: bird             AP:   0.8585  -------
-------  Class: boat             AP:   0.7523  -------
-------  Class: bottle           AP:   0.4112  -------
-------  Class: bus              AP:   0.6179  -------
-------  Class: car              AP:   0.8107  -------
-------  Class: cat              AP:   0.7785  -------
-------  Class: chair            AP:   0.6147  -------
-------  Class: cow              AP:   0.5179  -------
-------  Class: diningtable      AP:   0.6029  -------
-------  Class: dog              AP:   0.7296  -------
-------  Class: horse            AP:   0.7399  -------
-------  Class: motorbike        AP:   0.8026  -------
-------  Class: person           AP:   0.9033  -------
-------  Class: pottedplant      AP:   0.4436  -------
-------  Class: sheep            AP:   0.5588  -------
-------  Class: sofa             AP:   0.5481  -------
-------  Class: train            AP:   0.8862  -------
-------  Class: tvmonitor        AP:   0.6183  -------
mAP: 0.6891
Avg loss: 0.13423261776858686
Evaluating classifier
Mean Precision Score for Testing on Epoch 10 is 0.6890731384548342
```

```
Starting epoch number 11
Loss for Training on Epoch 11 is 0.043439846485853195
Starting epoch number 12
Loss for Training on Epoch 12 is 0.03338000923395157
Starting epoch number 13
Loss for Training on Epoch 13 is 0.02982497774064541
Starting epoch number 14
Loss for Training on Epoch 14 is 0.030931679531931877
Starting epoch number 15
Loss for Training on Epoch 15 is 0.028879933059215546
-------  Class: aeroplane        AP:   0.8632  -------
-------  Class: bicycle          AP:   0.7091  -------
-------  Class: bird             AP:   0.8466  -------
-------  Class: boat             AP:   0.7231  -------
-------  Class: bottle           AP:   0.3739  -------
-------  Class: bus              AP:   0.6240  -------
-------  Class: car              AP:   0.8063  -------
-------  Class: cat              AP:   0.7535  -------
-------  Class: chair            AP:   0.6111  -------
-------  Class: cow              AP:   0.5133  -------
-------  Class: diningtable      AP:   0.5854  -------
-------  Class: dog              AP:   0.7100  -------
-------  Class: horse            AP:   0.7559  -------
-------  Class: motorbike        AP:   0.7934  -------
-------  Class: person           AP:   0.9076  -------
-------  Class: pottedplant      AP:   0.4340  -------
-------  Class: sheep            AP:   0.5684  -------
-------  Class: sofa             AP:   0.5459  -------
-------  Class: train            AP:   0.8865  -------
-------  Class: tvmonitor        AP:   0.6136  -------
mAP: 0.6812
Avg loss: 0.15442244226441665
Evaluating classifier
Mean Precision Score for Testing on Epoch 15 is 0.6812323136585167
Starting epoch number 16
Loss for Training on Epoch 16 is 0.026101112365722656
Starting epoch number 17
Loss for Training on Epoch 17 is 0.01903168484568596
Starting epoch number 18
Loss for Training on Epoch 18 is 0.016157908365130424
Starting epoch number 19
Loss for Training on Epoch 19 is 0.029855649918317795
Starting epoch number 20
Loss for Training on Epoch 20 is 0.026157978922128677
-------  Class: aeroplane        AP:   0.8720  -------
-------  Class: bicycle          AP:   0.6885  -------
-------  Class: bird             AP:   0.8077  -------
-------  Class: boat             AP:   0.7238  -------
```

```
-------  Class: bottle           AP:   0.3165  -------
-------  Class: bus              AP:   0.5907  -------
-------  Class: car              AP:   0.7979  -------
-------  Class: cat              AP:   0.7405  -------
-------  Class: chair            AP:   0.5722  -------
-------  Class: cow              AP:   0.5337  -------
-------  Class: diningtable      AP:   0.5388  -------
-------  Class: dog              AP:   0.6850  -------
-------  Class: horse            AP:   0.7453  -------
-------  Class: motorbike        AP:   0.7601  -------
-------  Class: person           AP:   0.8922  -------
-------  Class: pottedplant      AP:   0.4017  -------
-------  Class: sheep            AP:   0.5830  -------
-------  Class: sofa             AP:   0.5245  -------
-------  Class: train            AP:   0.8641  -------
-------  Class: tvmonitor        AP:   0.6079  -------
mAP: 0.6623
Avg loss: 0.20485579748364055
Evaluating classifier
Mean Precision Score for Testing on Epoch 20 is 0.6622990875393091
```

[27]:
```
plot_losses(train_losses, val_losses, test_frequency, num_epochs)
plot_mAP(train_mAPs, val_mAPs, test_frequency, num_epochs)
```





[28]:
```
mAP_test, test_loss, test_aps = test_classifier(test_loader, classifier,␣
 ↪criterion)
print("Test mAP: ", mAP_test)
```

```
-------  Class: aeroplane        AP:   0.8483  -------
-------  Class: bicycle          AP:   0.7503  -------
-------  Class: bird             AP:   0.7964  -------
-------  Class: boat             AP:   0.7894  -------
-------  Class: bottle           AP:   0.2776  -------
-------  Class: bus              AP:   0.6580  -------
-------  Class: car              AP:   0.8213  -------
-------  Class: cat              AP:   0.7488  -------
-------  Class: chair            AP:   0.5427  -------
-------  Class: cow              AP:   0.5058  -------
-------  Class: diningtable      AP:   0.5546  -------
-------  Class: dog              AP:   0.6797  -------
-------  Class: horse            AP:   0.8363  -------
-------  Class: motorbike        AP:   0.7136  -------
-------  Class: person           AP:   0.9069  -------
-------  Class: pottedplant      AP:   0.4180  -------
-------  Class: sheep            AP:   0.6622  -------
-------  Class: sofa             AP:   0.4921  -------
-------  Class: train            AP:   0.8469  -------
-------  Class: tvmonitor        AP:   0.5784  -------
```

```
mAP: 0.6714
Avg loss: 0.19662140294909478
Test mAP:  0.6713662652003508
```

[ ]: