MP3_P1A_Introduction

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,
        ])
```

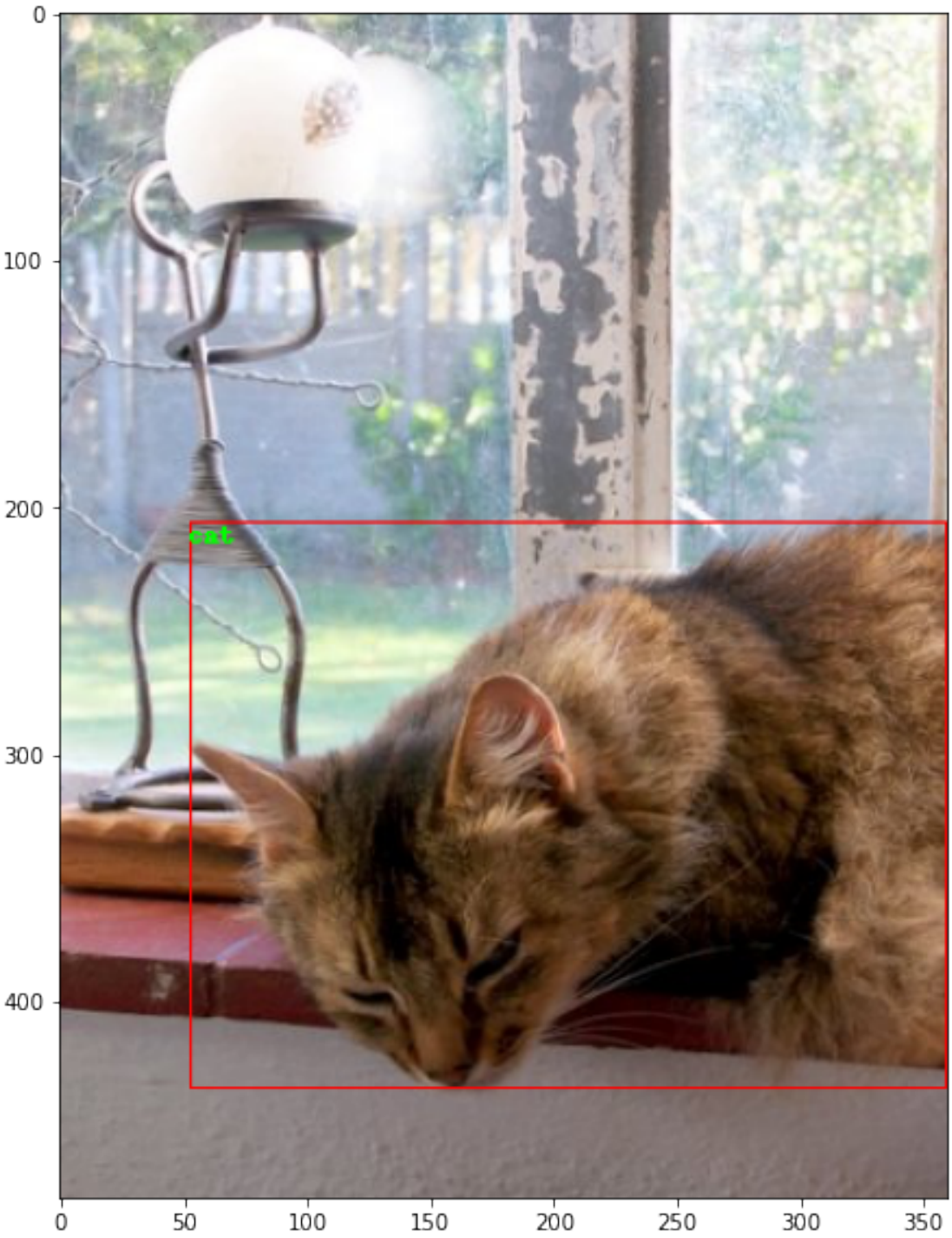
[5]:
```
ds_val = VocDataset('VOCdevkit_2007/VOC2007/','val',test_transform)
```

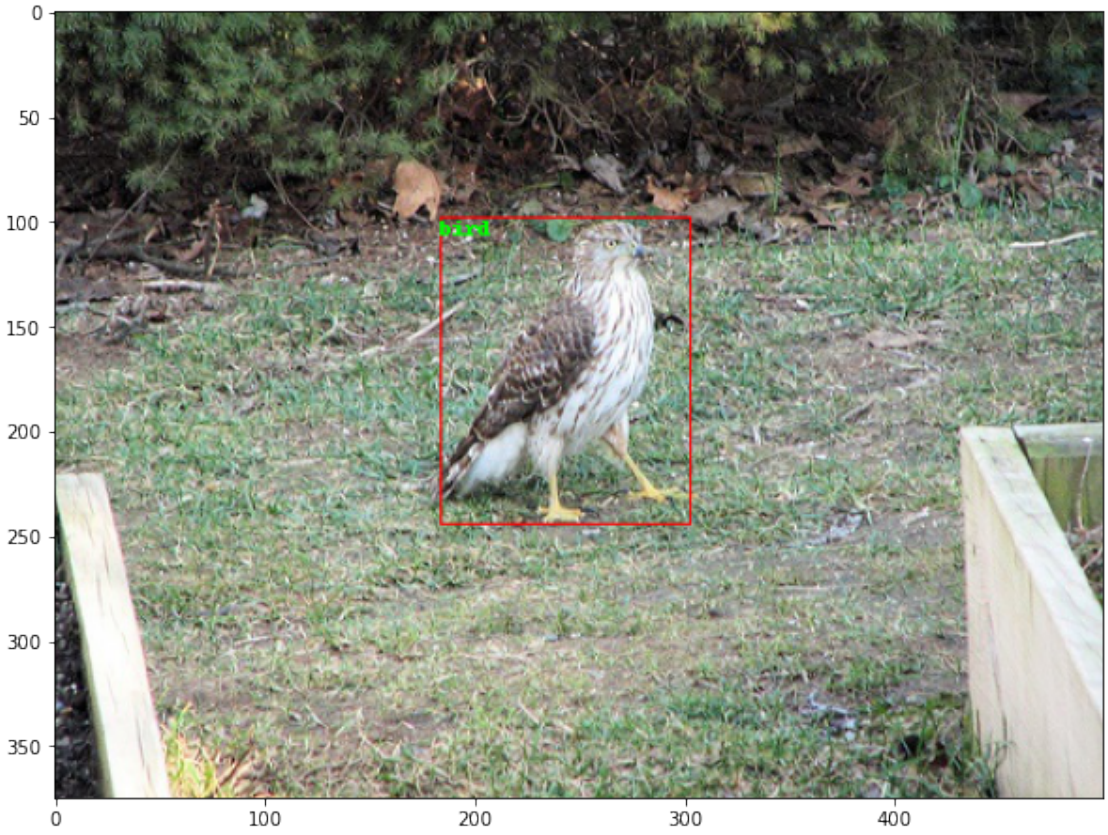
2.2.3 Visualizing the Data

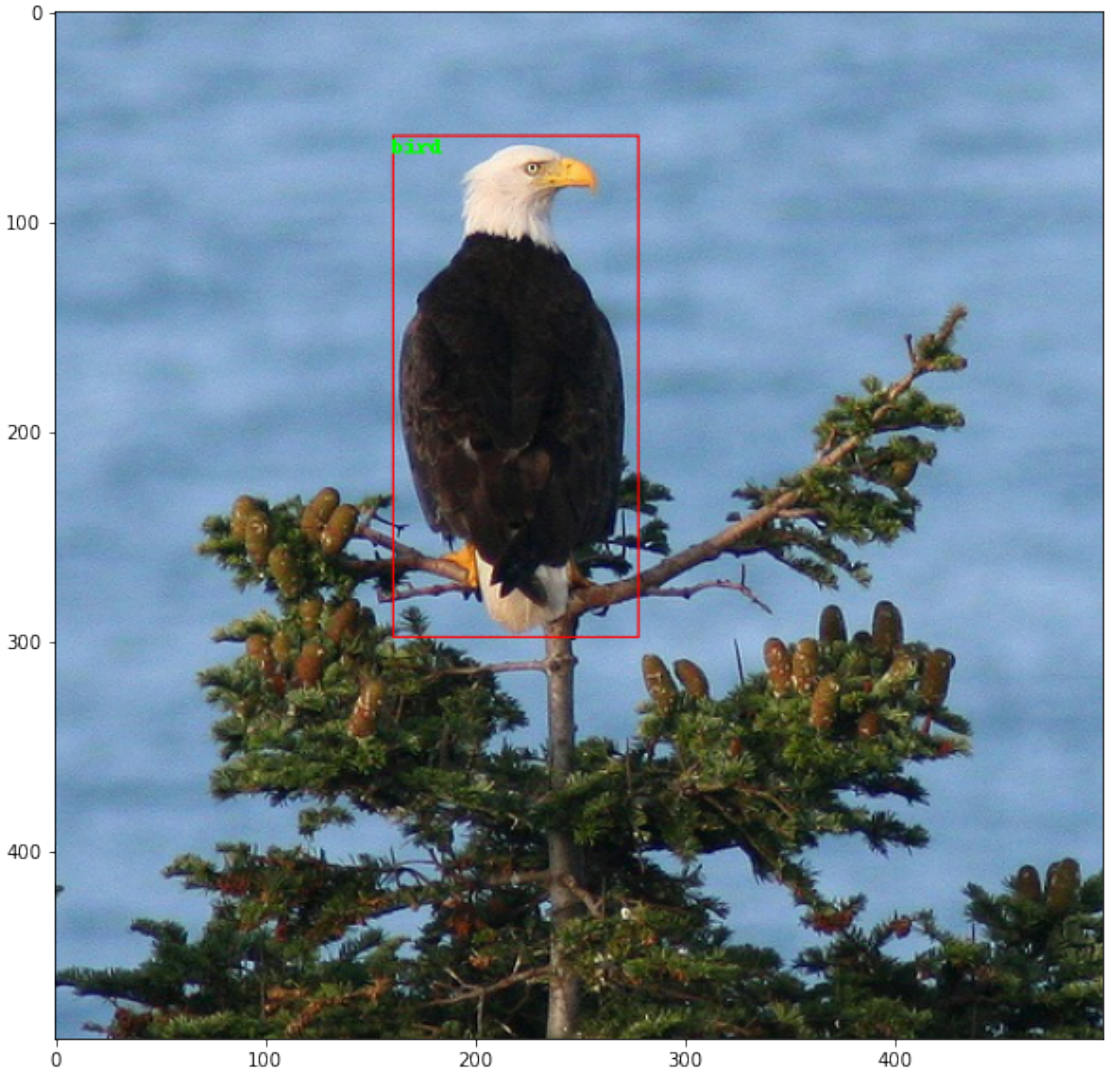
PASCAL VOC has bounding box annotations in addition to class labels. Use the following code to visualize some random examples and corresponding annotations from the train set.

[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))
```


person
0
100
200
300
400
0
50
100
150
200
250
300


cat
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


bird
0
100
200
300
400
0
100
200
300
400

3 Classification

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

[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)
```

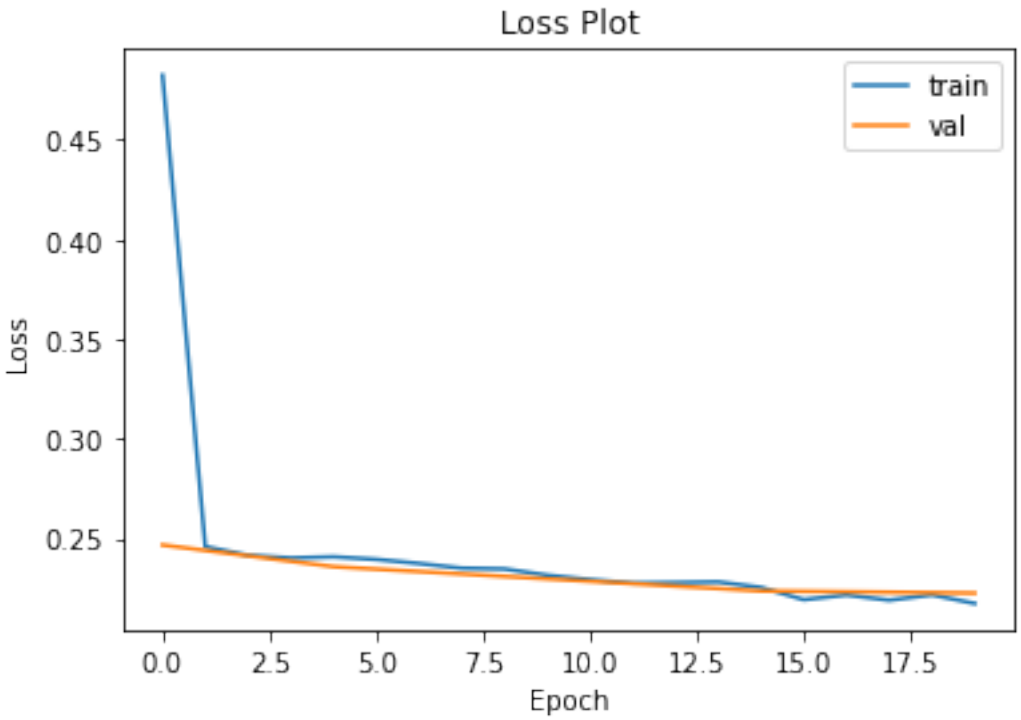
```
Starting epoch number 1
Loss for Training on Epoch 1 is 0.4822922646999359
-------  Class: aeroplane        AP:   0.1013  -------
-------  Class: bicycle          AP:   0.0489  -------
-------  Class: bird             AP:   0.1196  -------
-------  Class: boat             AP:   0.0904  -------
-------  Class: bottle           AP:   0.0425  -------
-------  Class: bus              AP:   0.0294  -------
-------  Class: car              AP:   0.1126  -------
-------  Class: cat              AP:   0.0846  -------
-------  Class: chair            AP:   0.1193  -------
-------  Class: cow              AP:   0.0316  -------
-------  Class: diningtable      AP:   0.0526  -------
-------  Class: dog              AP:   0.1179  -------
-------  Class: horse            AP:   0.0531  -------
-------  Class: motorbike        AP:   0.0458  -------
-------  Class: person           AP:   0.3519  -------
-------  Class: pottedplant      AP:   0.0472  -------
-------  Class: sheep            AP:   0.0287  -------
-------  Class: sofa             AP:   0.0907  -------
-------  Class: train            AP:   0.0359  -------
-------  Class: tvmonitor        AP:   0.0497  -------
mAP: 0.0827
Avg loss: 0.24675959494768404
Evaluating classifier
```

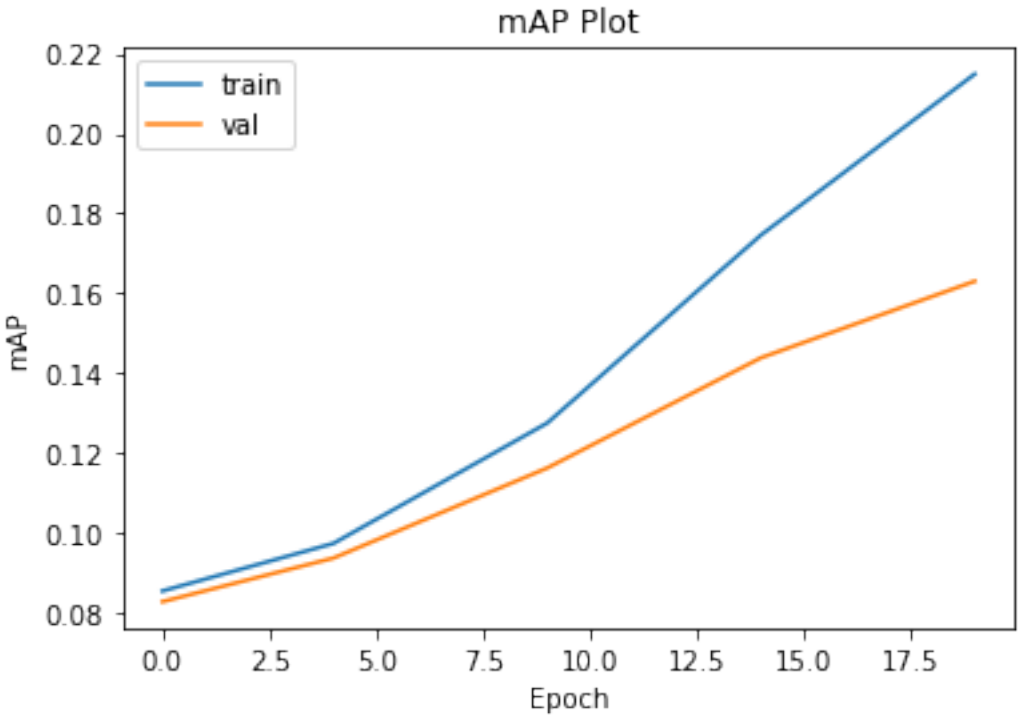
```
Mean Precision Score for Testing on Epoch 1 is 0.08267893031359375
Starting epoch number 2
Loss for Training on Epoch 2 is 0.24593831598758698
Starting epoch number 3
Loss for Training on Epoch 3 is 0.24157248437404633
Starting epoch number 4
Loss for Training on Epoch 4 is 0.24024754762649536
Starting epoch number 5
Loss for Training on Epoch 5 is 0.24083521962165833
-------  Class: aeroplane        AP:   0.1055  -------
-------  Class: bicycle          AP:   0.0457  -------
-------  Class: bird             AP:   0.1161  -------
-------  Class: boat             AP:   0.0815  -------
-------  Class: bottle           AP:   0.0465  -------
-------  Class: bus              AP:   0.0290  -------
-------  Class: car              AP:   0.1463  -------
-------  Class: cat              AP:   0.1064  -------
-------  Class: chair            AP:   0.1351  -------
-------  Class: cow              AP:   0.0326  -------
-------  Class: diningtable      AP:   0.0573  -------
-------  Class: dog              AP:   0.1346  -------
-------  Class: horse            AP:   0.0537  -------
-------  Class: motorbike        AP:   0.0436  -------
-------  Class: person           AP:   0.4713  -------
-------  Class: pottedplant      AP:   0.0471  -------
-------  Class: sheep            AP:   0.0316  -------
-------  Class: sofa             AP:   0.0967  -------
-------  Class: train            AP:   0.0376  -------
-------  Class: tvmonitor        AP:   0.0542  -------
mAP: 0.0936
Avg loss: 0.235886572914965
Evaluating classifier
Mean Precision Score for Testing on Epoch 5 is 0.09361205219984012
Starting epoch number 6
Loss for Training on Epoch 6 is 0.23953580856323242
Starting epoch number 7
Loss for Training on Epoch 7 is 0.23749369382858276
Starting epoch number 8
Loss for Training on Epoch 8 is 0.23513050377368927
Starting epoch number 9
Loss for Training on Epoch 9 is 0.23474645614624023
Starting epoch number 10
Loss for Training on Epoch 10 is 0.2316085249185562
-------  Class: aeroplane        AP:   0.2544  -------
-------  Class: bicycle          AP:   0.0508  -------
-------  Class: bird             AP:   0.1199  -------
-------  Class: boat             AP:   0.1318  -------
-------  Class: bottle           AP:   0.0584  -------
```

```
-------  Class: bus              AP:   0.0387  -------
-------  Class: car              AP:   0.1777  -------
-------  Class: cat              AP:   0.1286  -------
-------  Class: chair            AP:   0.1809  -------
-------  Class: cow              AP:   0.0356  -------
-------  Class: diningtable      AP:   0.0795  -------
-------  Class: dog              AP:   0.1390  -------
-------  Class: horse            AP:   0.0621  -------
-------  Class: motorbike        AP:   0.0460  -------
-------  Class: person           AP:   0.4926  -------
-------  Class: pottedplant      AP:   0.0511  -------
-------  Class: sheep            AP:   0.0314  -------
-------  Class: sofa             AP:   0.1156  -------
-------  Class: train            AP:   0.0688  -------
-------  Class: tvmonitor        AP:   0.0607  -------
mAP: 0.1162
Avg loss: 0.22978955594932332
Evaluating classifier
Mean Precision Score for Testing on Epoch 10 is 0.11618564980489962
Starting epoch number 11
Loss for Training on Epoch 11 is 0.22937935590744019
Starting epoch number 12
Loss for Training on Epoch 12 is 0.2276429831981659
Starting epoch number 13
Loss for Training on Epoch 13 is 0.22791102528572083
Starting epoch number 14
Loss for Training on Epoch 14 is 0.22824092209339142
Starting epoch number 15
Loss for Training on Epoch 15 is 0.22547674179077148
-------  Class: aeroplane        AP:   0.3760  -------
-------  Class: bicycle          AP:   0.0758  -------
-------  Class: bird             AP:   0.1038  -------
-------  Class: boat             AP:   0.1807  -------
-------  Class: bottle           AP:   0.0667  -------
-------  Class: bus              AP:   0.0622  -------
-------  Class: car              AP:   0.2333  -------
-------  Class: cat              AP:   0.1362  -------
-------  Class: chair            AP:   0.2149  -------
-------  Class: cow              AP:   0.0446  -------
-------  Class: diningtable      AP:   0.1122  -------
-------  Class: dog              AP:   0.1433  -------
-------  Class: horse            AP:   0.0923  -------
-------  Class: motorbike        AP:   0.0538  -------
-------  Class: person           AP:   0.5038  -------
-------  Class: pottedplant      AP:   0.0634  -------
-------  Class: sheep            AP:   0.0446  -------
-------  Class: sofa             AP:   0.1661  -------
-------  Class: train            AP:   0.1162  -------
```

```
-------  Class: tvmonitor        AP:   0.0863  -------
mAP: 0.1438
Avg loss: 0.2237169859456081
Evaluating classifier
Mean Precision Score for Testing on Epoch 15 is 0.1438042743055859
Starting epoch number 16
Loss for Training on Epoch 16 is 0.21934661269187927
Starting epoch number 17
Loss for Training on Epoch 17 is 0.22170685231685638
Starting epoch number 18
Loss for Training on Epoch 18 is 0.21908095479011536
Starting epoch number 19
Loss for Training on Epoch 19 is 0.22175471484661102
Starting epoch number 20
Loss for Training on Epoch 20 is 0.21743857860565186
-------  Class: aeroplane        AP:   0.3747  -------
-------  Class: bicycle          AP:   0.0901  -------
-------  Class: bird             AP:   0.0994  -------
-------  Class: boat             AP:   0.2435  -------
-------  Class: bottle           AP:   0.0780  -------
-------  Class: bus              AP:   0.0763  -------
-------  Class: car              AP:   0.2705  -------
-------  Class: cat              AP:   0.1480  -------
-------  Class: chair            AP:   0.2450  -------
-------  Class: cow              AP:   0.0561  -------
-------  Class: diningtable      AP:   0.1664  -------
-------  Class: dog              AP:   0.1459  -------
-------  Class: horse            AP:   0.1130  -------
-------  Class: motorbike        AP:   0.0798  -------
-------  Class: person           AP:   0.5148  -------
-------  Class: pottedplant      AP:   0.0705  -------
-------  Class: sheep            AP:   0.0482  -------
-------  Class: sofa             AP:   0.1818  -------
-------  Class: train            AP:   0.1487  -------
-------  Class: tvmonitor        AP:   0.1094  -------
mAP: 0.1630
Avg loss: 0.22264221807320914
Evaluating classifier
Mean Precision Score for Testing on Epoch 20 is 0.16300340788757076
```

[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
0.45
0.40
0.35
0.30
0.25
Loss
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch


mAP Plot
train
val
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
mAP
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:   0.3399  -------
-------  Class: bicycle          AP:   0.0696  -------
-------  Class: bird             AP:   0.0991  -------
-------  Class: boat             AP:   0.1920  -------
-------  Class: bottle           AP:   0.0792  -------
-------  Class: bus              AP:   0.0697  -------
-------  Class: car              AP:   0.3238  -------
-------  Class: cat              AP:   0.1259  -------
-------  Class: chair            AP:   0.2312  -------
-------  Class: cow              AP:   0.0673  -------
-------  Class: diningtable      AP:   0.1058  -------
-------  Class: dog              AP:   0.1541  -------
-------  Class: horse            AP:   0.1208  -------
-------  Class: motorbike        AP:   0.0950  -------
-------  Class: person           AP:   0.5366  -------
-------  Class: pottedplant      AP:   0.0643  -------
-------  Class: sheep            AP:   0.0950  -------
-------  Class: sofa             AP:   0.1851  -------
-------  Class: train            AP:   0.1704  -------
-------  Class: tvmonitor        AP:   0.0985  -------
mAP: 0.1612
```

```
Avg loss: 0.2186693897843361
```

[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)
```

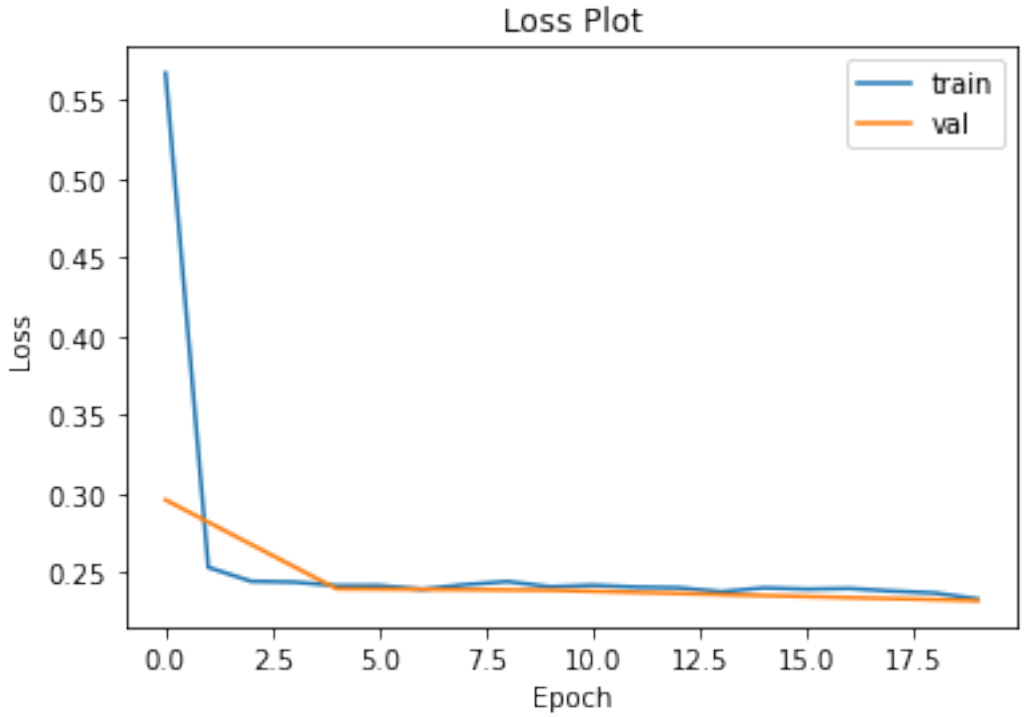
```
Starting epoch number 1
Loss for Training on Epoch 1 is 0.5670338273048401
-------  Class: aeroplane        AP:   0.1682  -------
-------  Class: bicycle          AP:   0.0462  -------
-------  Class: bird             AP:   0.1023  -------
-------  Class: boat             AP:   0.1485  -------
-------  Class: bottle           AP:   0.0363  -------
-------  Class: bus              AP:   0.0307  -------
-------  Class: car              AP:   0.1802  -------
-------  Class: cat              AP:   0.0899  -------
-------  Class: chair            AP:   0.1059  -------
-------  Class: cow              AP:   0.0353  -------
-------  Class: diningtable      AP:   0.0454  -------
-------  Class: dog              AP:   0.1061  -------
-------  Class: horse            AP:   0.0460  -------
-------  Class: motorbike        AP:   0.0347  -------
```

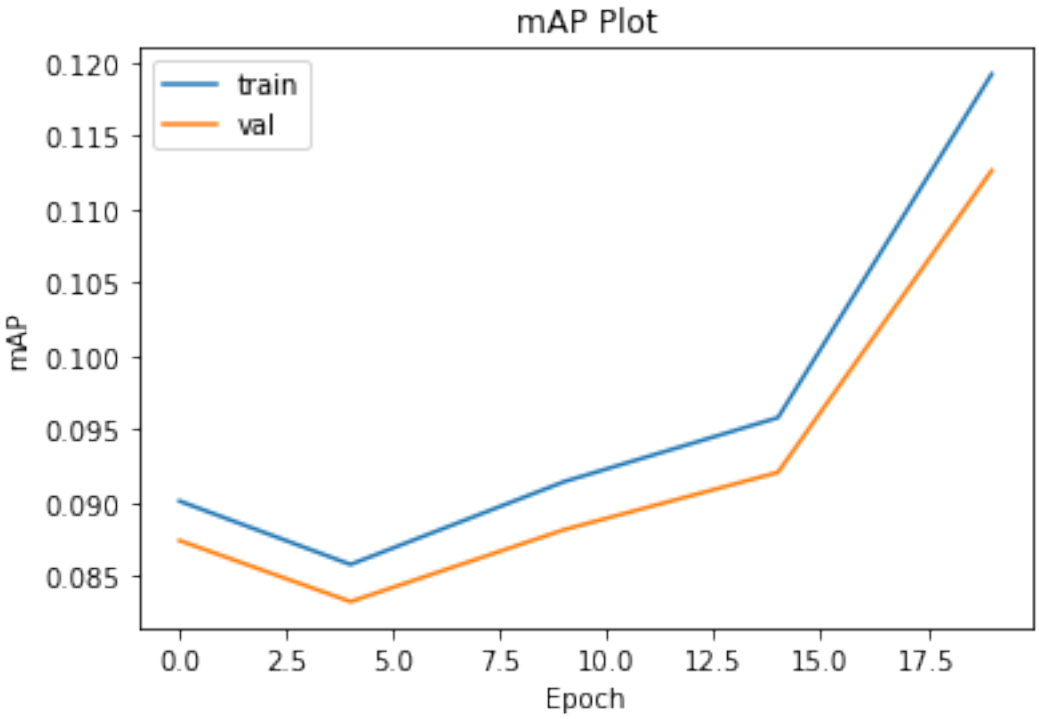
```
-------  Class: person           AP:   0.3218  -------
-------  Class: pottedplant      AP:   0.0419  -------
-------  Class: sheep            AP:   0.0367  -------
-------  Class: sofa             AP:   0.0839  -------
-------  Class: train            AP:   0.0414  -------
-------  Class: tvmonitor        AP:   0.0470  -------
mAP: 0.0874
Avg loss: 0.29586591615396385
Evaluating classifier
Mean Precision Score for Testing on Epoch 1 is 0.087432536811534
Starting epoch number 2
Loss for Training on Epoch 2 is 0.2530347406864166
Starting epoch number 3
Loss for Training on Epoch 3 is 0.24417918920516968
Starting epoch number 4
Loss for Training on Epoch 4 is 0.2436884641647339
Starting epoch number 5
Loss for Training on Epoch 5 is 0.24142777919769287
-------  Class: aeroplane        AP:   0.0943  -------
-------  Class: bicycle          AP:   0.0465  -------
-------  Class: bird             AP:   0.1170  -------
-------  Class: boat             AP:   0.1001  -------
-------  Class: bottle           AP:   0.0416  -------
-------  Class: bus              AP:   0.0290  -------
-------  Class: car              AP:   0.1094  -------
-------  Class: cat              AP:   0.0933  -------
-------  Class: chair            AP:   0.1218  -------
-------  Class: cow              AP:   0.0355  -------
-------  Class: diningtable      AP:   0.0546  -------
-------  Class: dog              AP:   0.1167  -------
-------  Class: horse            AP:   0.0491  -------
-------  Class: motorbike        AP:   0.0367  -------
-------  Class: person           AP:   0.3449  -------
-------  Class: pottedplant      AP:   0.0470  -------
-------  Class: sheep            AP:   0.0420  -------
-------  Class: sofa             AP:   0.0967  -------
-------  Class: train            AP:   0.0404  -------
-------  Class: tvmonitor        AP:   0.0488  -------
mAP: 0.0833
Avg loss: 0.23943003953671924
Evaluating classifier
Mean Precision Score for Testing on Epoch 5 is 0.08326895212000034
Starting epoch number 6
Loss for Training on Epoch 6 is 0.24145732820034027
Starting epoch number 7
Loss for Training on Epoch 7 is 0.23909330368041992
Starting epoch number 8
Loss for Training on Epoch 8 is 0.2420089691877365
```

```
Starting epoch number 9
Loss for Training on Epoch 9 is 0.24403002858161926
Starting epoch number 10
Loss for Training on Epoch 10 is 0.24069006741046906
-------  Class: aeroplane        AP:   0.0827  -------
-------  Class: bicycle          AP:   0.0490  -------
-------  Class: bird             AP:   0.1249  -------
-------  Class: boat             AP:   0.0907  -------
-------  Class: bottle           AP:   0.0415  -------
-------  Class: bus              AP:   0.0296  -------
-------  Class: car              AP:   0.1152  -------
-------  Class: cat              AP:   0.0929  -------
-------  Class: chair            AP:   0.1290  -------
-------  Class: cow              AP:   0.0366  -------
-------  Class: diningtable      AP:   0.0562  -------
-------  Class: dog              AP:   0.1195  -------
-------  Class: horse            AP:   0.0521  -------
-------  Class: motorbike        AP:   0.0391  -------
-------  Class: person           AP:   0.4176  -------
-------  Class: pottedplant      AP:   0.0496  -------
-------  Class: sheep            AP:   0.0490  -------
-------  Class: sofa             AP:   0.0955  -------
-------  Class: train            AP:   0.0419  -------
-------  Class: tvmonitor        AP:   0.0511  -------
mAP: 0.0882
Avg loss: 0.2385225970955456
Evaluating classifier
Mean Precision Score for Testing on Epoch 10 is 0.0881786049095123
Starting epoch number 11
Loss for Training on Epoch 11 is 0.24164117872714996
Starting epoch number 12
Loss for Training on Epoch 12 is 0.2404957264661789
Starting epoch number 13
Loss for Training on Epoch 13 is 0.24006013572216034
Starting epoch number 14
Loss for Training on Epoch 14 is 0.23754532635211945
Starting epoch number 15
Loss for Training on Epoch 15 is 0.23998771607875824
-------  Class: aeroplane        AP:   0.0690  -------
-------  Class: bicycle          AP:   0.0525  -------
-------  Class: bird             AP:   0.1345  -------
-------  Class: boat             AP:   0.0762  -------
-------  Class: bottle           AP:   0.0444  -------
-------  Class: bus              AP:   0.0304  -------
-------  Class: car              AP:   0.1237  -------
-------  Class: cat              AP:   0.0918  -------
-------  Class: chair            AP:   0.1405  -------
-------  Class: cow              AP:   0.0370  -------
```

```
-------  Class: diningtable      AP:   0.0629  -------
-------  Class: dog              AP:   0.1203  -------
-------  Class: horse            AP:   0.0553  -------
-------  Class: motorbike        AP:   0.0431  -------
-------  Class: person           AP:   0.4576  -------
-------  Class: pottedplant      AP:   0.0548  -------
-------  Class: sheep            AP:   0.0513  -------
-------  Class: sofa             AP:   0.0982  -------
-------  Class: train            AP:   0.0453  -------
-------  Class: tvmonitor        AP:   0.0527  -------
mAP: 0.0921
Avg loss: 0.2351163757197997
Evaluating classifier
Mean Precision Score for Testing on Epoch 15 is 0.0920795272784576
Starting epoch number 16
Loss for Training on Epoch 16 is 0.2391771823167801
Starting epoch number 17
Loss for Training on Epoch 17 is 0.23961228132247925
Starting epoch number 18
Loss for Training on Epoch 18 is 0.23796388506889343
Starting epoch number 19
Loss for Training on Epoch 19 is 0.2367224544286728
Starting epoch number 20
Loss for Training on Epoch 20 is 0.23296068608760834
-------  Class: aeroplane        AP:   0.1752  -------
-------  Class: bicycle          AP:   0.0591  -------
-------  Class: bird             AP:   0.1324  -------
-------  Class: boat             AP:   0.1520  -------
-------  Class: bottle           AP:   0.0565  -------
-------  Class: bus              AP:   0.0406  -------
-------  Class: car              AP:   0.1727  -------
-------  Class: cat              AP:   0.0997  -------
-------  Class: chair            AP:   0.1710  -------
-------  Class: cow              AP:   0.0383  -------
-------  Class: diningtable      AP:   0.0811  -------
-------  Class: dog              AP:   0.1244  -------
-------  Class: horse            AP:   0.0558  -------
-------  Class: motorbike        AP:   0.0452  -------
-------  Class: person           AP:   0.5024  -------
-------  Class: pottedplant      AP:   0.0606  -------
-------  Class: sheep            AP:   0.0473  -------
-------  Class: sofa             AP:   0.1173  -------
-------  Class: train            AP:   0.0682  -------
-------  Class: tvmonitor        AP:   0.0534  -------
mAP: 0.1127
Avg loss: 0.23188597694331525
Evaluating classifier
Mean Precision Score for Testing on Epoch 20 is 0.11265499050337262
```

[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:   0.1462  -------
-------  Class: bicycle          AP:   0.0567  -------
-------  Class: bird             AP:   0.1084  -------
-------  Class: boat             AP:   0.1115  -------
-------  Class: bottle           AP:   0.0549  -------
-------  Class: bus              AP:   0.0382  -------
-------  Class: car              AP:   0.1735  -------
-------  Class: cat              AP:   0.0981  -------
-------  Class: chair            AP:   0.1530  -------
-------  Class: cow              AP:   0.0441  -------
-------  Class: diningtable      AP:   0.0635  -------
-------  Class: dog              AP:   0.1293  -------
-------  Class: horse            AP:   0.0537  -------
-------  Class: motorbike        AP:   0.0402  -------
-------  Class: person           AP:   0.5094  -------
-------  Class: pottedplant      AP:   0.0532  -------
-------  Class: sheep            AP:   0.0611  -------
-------  Class: sofa             AP:   0.1192  -------
-------  Class: train            AP:   0.0799  -------
-------  Class: tvmonitor        AP:   0.0540  -------
mAP: 0.1074
Avg loss: 0.22864371940493583
Test mAP:  0.10741163339990774
```

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)
```

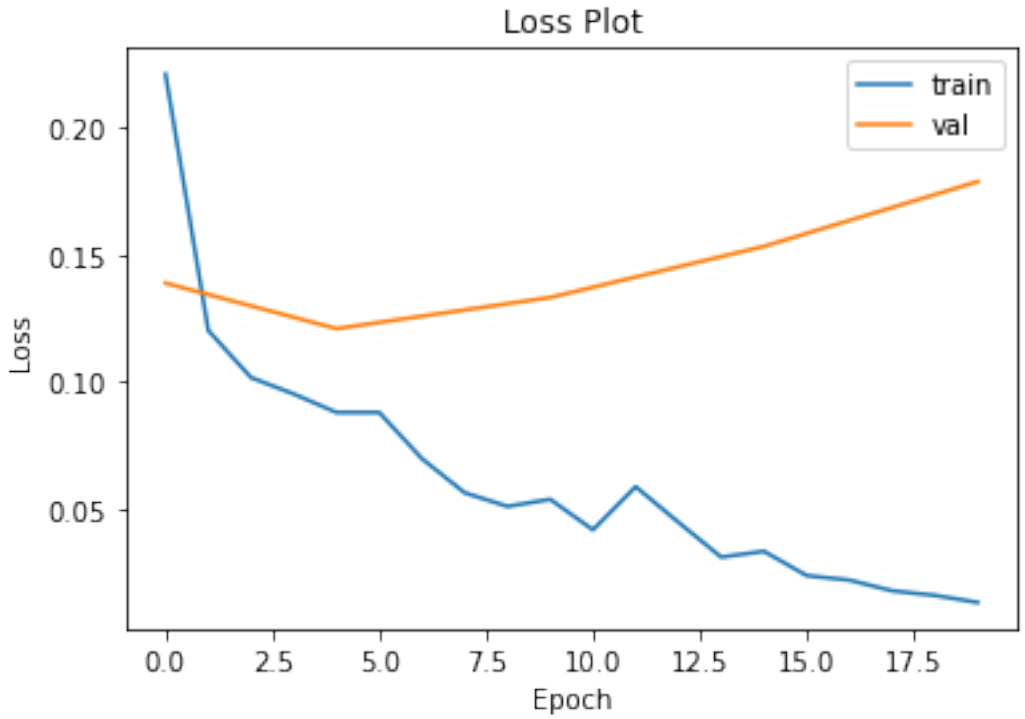
```
Starting epoch number 1
Loss for Training on Epoch 1 is 0.22088713943958282
-------  Class: aeroplane        AP:   0.7777  -------
-------  Class: bicycle          AP:   0.5748  -------
-------  Class: bird             AP:   0.8083  -------
-------  Class: boat             AP:   0.6698  -------
-------  Class: bottle           AP:   0.2634  -------
-------  Class: bus              AP:   0.5101  -------
-------  Class: car              AP:   0.7644  -------
-------  Class: cat              AP:   0.7025  -------
-------  Class: chair            AP:   0.5885  -------
-------  Class: cow              AP:   0.2592  -------
-------  Class: diningtable      AP:   0.4800  -------
-------  Class: dog              AP:   0.5996  -------
-------  Class: horse            AP:   0.6081  -------
-------  Class: motorbike        AP:   0.7115  -------
-------  Class: person           AP:   0.8985  -------
-------  Class: pottedplant      AP:   0.3279  -------
-------  Class: sheep            AP:   0.2182  -------
-------  Class: sofa             AP:   0.4274  -------
-------  Class: train            AP:   0.8056  -------
-------  Class: tvmonitor        AP:   0.5279  -------
mAP: 0.5762
Avg loss: 0.13878947788593815
Evaluating classifier
Mean Precision Score for Testing on Epoch 1 is 0.5761676358775671
Starting epoch number 2
Loss for Training on Epoch 2 is 0.12018540501594543
Starting epoch number 3
Loss for Training on Epoch 3 is 0.10166820883750916
Starting epoch number 4
Loss for Training on Epoch 4 is 0.09522087872028351
Starting epoch number 5
Loss for Training on Epoch 5 is 0.08795271068811417
-------  Class: aeroplane        AP:   0.8640  -------
-------  Class: bicycle          AP:   0.7139  -------
-------  Class: bird             AP:   0.8590  -------
-------  Class: boat             AP:   0.7756  -------
-------  Class: bottle           AP:   0.2846  -------
-------  Class: bus              AP:   0.5940  -------
-------  Class: car              AP:   0.8178  -------
-------  Class: cat              AP:   0.7880  -------
```

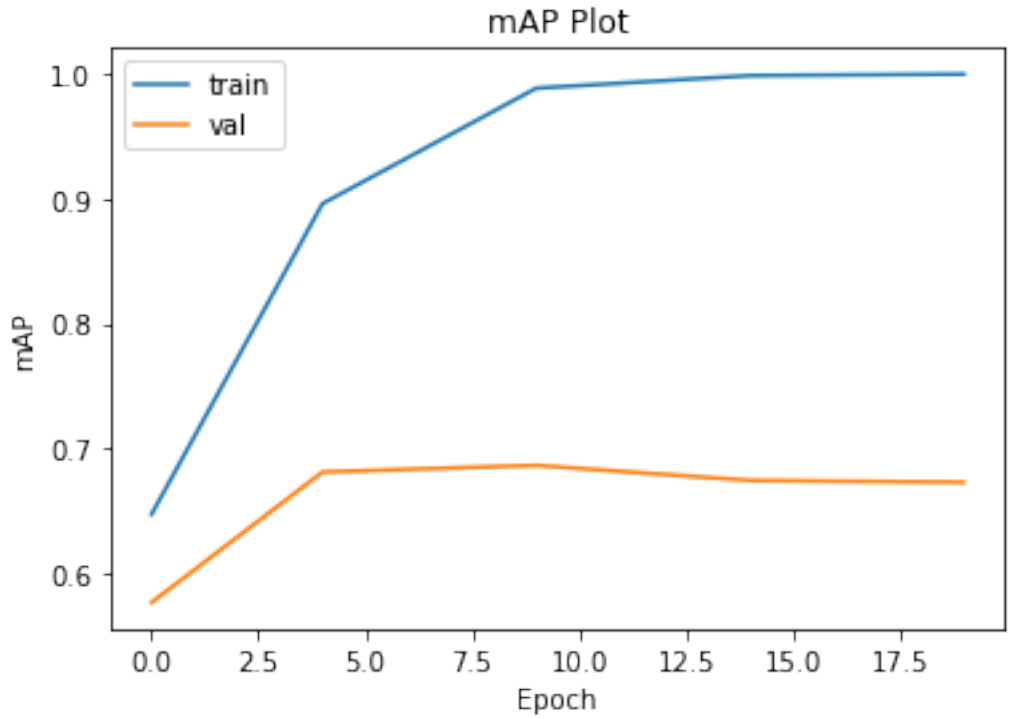
```
-------  Class: chair            AP:   0.5990  -------
-------  Class: cow              AP:   0.4832  -------
-------  Class: diningtable      AP:   0.5598  -------
-------  Class: dog              AP:   0.7058  -------
-------  Class: horse            AP:   0.7380  -------
-------  Class: motorbike        AP:   0.8057  -------
-------  Class: person           AP:   0.9085  -------
-------  Class: pottedplant      AP:   0.4744  -------
-------  Class: sheep            AP:   0.5537  -------
-------  Class: sofa             AP:   0.5406  -------
-------  Class: train            AP:   0.8929  -------
-------  Class: tvmonitor        AP:   0.6584  -------
mAP: 0.6809
Avg loss: 0.12090012825587217
Evaluating classifier
Mean Precision Score for Testing on Epoch 5 is 0.6808549336018612
Starting epoch number 6
Loss for Training on Epoch 6 is 0.08797134459018707
Starting epoch number 7
Loss for Training on Epoch 7 is 0.0697883740067482
Starting epoch number 8
Loss for Training on Epoch 8 is 0.05638810992240906
Starting epoch number 9
Loss for Training on Epoch 9 is 0.05104634538292885
Starting epoch number 10
Loss for Training on Epoch 10 is 0.05381588265299797
-------  Class: aeroplane        AP:   0.8709  -------
-------  Class: bicycle          AP:   0.7105  -------
-------  Class: bird             AP:   0.8568  -------
-------  Class: boat             AP:   0.7792  -------
-------  Class: bottle           AP:   0.3163  -------
-------  Class: bus              AP:   0.5721  -------
-------  Class: car              AP:   0.8224  -------
-------  Class: cat              AP:   0.7939  -------
-------  Class: chair            AP:   0.5884  -------
-------  Class: cow              AP:   0.5358  -------
-------  Class: diningtable      AP:   0.5985  -------
-------  Class: dog              AP:   0.6791  -------
-------  Class: horse            AP:   0.7479  -------
-------  Class: motorbike        AP:   0.8069  -------
-------  Class: person           AP:   0.9061  -------
-------  Class: pottedplant      AP:   0.4823  -------
-------  Class: sheep            AP:   0.6047  -------
-------  Class: sofa             AP:   0.5212  -------
-------  Class: train            AP:   0.8915  -------
-------  Class: tvmonitor        AP:   0.6387  -------
mAP: 0.6862
Avg loss: 0.13315457558515026
```

```
Evaluating classifier
Mean Precision Score for Testing on Epoch 10 is 0.6861694954976929
Starting epoch number 11
Loss for Training on Epoch 11 is 0.041838936507701874
Starting epoch number 12
Loss for Training on Epoch 12 is 0.05879150703549385
Starting epoch number 13
Loss for Training on Epoch 13 is 0.04479841887950897
Starting epoch number 14
Loss for Training on Epoch 14 is 0.031119735911488533
Starting epoch number 15
Loss for Training on Epoch 15 is 0.033401962369680405
-------  Class: aeroplane        AP:   0.8597  -------
-------  Class: bicycle          AP:   0.7225  -------
-------  Class: bird             AP:   0.8538  -------
-------  Class: boat             AP:   0.7474  -------
-------  Class: bottle           AP:   0.3183  -------
-------  Class: bus              AP:   0.5576  -------
-------  Class: car              AP:   0.8065  -------
-------  Class: cat              AP:   0.7931  -------
-------  Class: chair            AP:   0.5770  -------
-------  Class: cow              AP:   0.5200  -------
-------  Class: diningtable      AP:   0.6007  -------
-------  Class: dog              AP:   0.6963  -------
-------  Class: horse            AP:   0.7547  -------
-------  Class: motorbike        AP:   0.7860  -------
-------  Class: person           AP:   0.9041  -------
-------  Class: pottedplant      AP:   0.4596  -------
-------  Class: sheep            AP:   0.5663  -------
-------  Class: sofa             AP:   0.4959  -------
-------  Class: train            AP:   0.8670  -------
-------  Class: tvmonitor        AP:   0.5957  -------
mAP: 0.6741
Avg loss: 0.1532426086418769
Evaluating classifier
Mean Precision Score for Testing on Epoch 15 is 0.6741043706564072
Starting epoch number 16
Loss for Training on Epoch 16 is 0.023852219805121422
Starting epoch number 17
Loss for Training on Epoch 17 is 0.02207884192466736
Starting epoch number 18
Loss for Training on Epoch 18 is 0.017864355817437172
Starting epoch number 19
Loss for Training on Epoch 19 is 0.016078734770417213
Starting epoch number 20
Loss for Training on Epoch 20 is 0.013291938230395317
-------  Class: aeroplane        AP:   0.8582  -------
-------  Class: bicycle          AP:   0.6973  -------
```

```
-------  Class: bird             AP:   0.8488  -------
-------  Class: boat             AP:   0.7461  -------
-------  Class: bottle           AP:   0.3403  -------
-------  Class: bus              AP:   0.5739  -------
-------  Class: car              AP:   0.8055  -------
-------  Class: cat              AP:   0.7965  -------
-------  Class: chair            AP:   0.5790  -------
-------  Class: cow              AP:   0.5240  -------
-------  Class: diningtable      AP:   0.5661  -------
-------  Class: dog              AP:   0.6877  -------
-------  Class: horse            AP:   0.7652  -------
-------  Class: motorbike        AP:   0.7979  -------
-------  Class: person           AP:   0.9085  -------
-------  Class: pottedplant      AP:   0.4662  -------
-------  Class: sheep            AP:   0.5466  -------
-------  Class: sofa             AP:   0.4711  -------
-------  Class: train            AP:   0.8722  -------
-------  Class: tvmonitor        AP:   0.6013  -------
mAP: 0.6726
Avg loss: 0.17868691492898792
Evaluating classifier
Mean Precision Score for Testing on Epoch 20 is 0.6726102870797436
```

[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:   0.8533  -------
-------  Class: bicycle          AP:   0.7585  -------
-------  Class: bird             AP:   0.8297  -------
-------  Class: boat             AP:   0.7963  -------
-------  Class: bottle           AP:   0.3068  -------
-------  Class: bus              AP:   0.6619  -------
-------  Class: car              AP:   0.8345  -------
-------  Class: cat              AP:   0.7908  -------
-------  Class: chair            AP:   0.5684  -------
-------  Class: cow              AP:   0.5241  -------
-------  Class: diningtable      AP:   0.6135  -------
-------  Class: dog              AP:   0.7110  -------
-------  Class: horse            AP:   0.8458  -------
-------  Class: motorbike        AP:   0.7674  -------
-------  Class: person           AP:   0.9205  -------
-------  Class: pottedplant      AP:   0.4462  -------
-------  Class: sheep            AP:   0.6102  -------
-------  Class: sofa             AP:   0.4838  -------
```

```
-------  Class: train            AP:   0.8556  -------
-------  Class: tvmonitor        AP:   0.5950  -------
mAP: 0.6887
Avg loss: 0.17193814761936665
Test mAP:  0.6886657450667757
```