Assignment 9: CNN Classification

Mohammed Jasam

Summary

The assignment was on CNN Classification, there are two different parts in this assignment. First, we have to build our own CNN model from the scratch. Here we specify various parameters in order to make the model work accurately and achieve the best possible accuracy. Secondly, we have to use the technique of Transfer Learning (reuse a pre-trained model that is ready to classify 10 classes) to classify our testing data into either face or background.

Algorithmic Approach:

Problem 1:

1. Load the data as an ImageDatastore object. This will help us load the data one by one with out the problem of over usage of the memory.
2. Set the different Parameters utilized in the training of the CNN model.
3. Define the CNN Architecture, in this process we define how the CNN architecture should be. We define different parameters like DataAugmentation in imageInputLayer, we also add different layers like the crossChannelNormalizationLayer and dropoutLayer, etc. We also make the model deeper to better classify the images.
4. Next, use the defined parameters in the trainingOptions, such as, MiniBatchSize, InitialLearnRate, etc.
5. Then we train the network using the defined architecture and parameters.
6. Finally, we calculate the accuracy
7.

Problem 2:

1. Here we first load the pre-trained network made available by the professor.
2. Next, we use the first N-3 layers of the network and define the last 3 layers to solve out 2 class problem (faces/backgrounds)
3. Then, we set the parameters like WeightLearnRateFactor, BiasLearnRateFactor, MiniBatchSize, InitialLearnRate, MaxEpochs.
4. Then we train the network again with the new set of parameters.
5. Finally, we classify the test set and calculate the accuracy.

Design additions:

I tried to change the number of channels in the deeper network in order to achieve better accuracy in classification. The first layer had 20 channels, the next layer has 40 channels and the final convolution layer has 80 channels in it. Since, the data isn't big enough, I think increase the number of channels helped me improve the accuracy.

Final Results:

Problem 1:

Task 1:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       28.13% |       1.2357 |          0.0100 |
|       5 |          50 |       00:00:02 |       56.25% |          NaN |          0.0100 |
|      10 |         100 |       00:00:04 |       56.25% |          NaN |          0.0100 |
|      15 |         150 |       00:00:07 |       56.25% |          NaN |          0.0100 |
|      20 |         200 |       00:00:09 |       56.25% |          NaN |          0.0100 |
|=========================================================================================|
Elapsed time is 12.685668 seconds.
Elapsed time is 0.303173 seconds.

MiniBatchSize = 32

Accuracy = 0.000000
```

Task 2:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       68.75% |       0.8781 |          0.0010 |
|       5 |          50 |       00:00:02 |       81.25% |       2.9892 |          0.0010 |
|      10 |         100 |       00:00:05 |       75.00% |       3.9856 |          0.0010 |
|      15 |         150 |       00:00:07 |       96.88% |       0.4982 |          0.0010 |
|      20 |         200 |       00:00:09 |       90.63% |       1.4946 |          0.0010 |
|=========================================================================================|
Elapsed time is 14.123093 seconds.
Elapsed time is 0.266428 seconds.

MiniBatchSize = 32
InitialLearRate = 0.001000
MaxEpochs = 20

Accuracy = 0.835580
```

Task 3:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       75.00% |       0.6365 |          0.0010 |
|       5 |          50 |       00:00:02 |       96.88% |       0.4982 |          0.0010 |
|      10 |         100 |       00:00:05 |       90.63% |       1.4946 |          0.0010 |
|      15 |         150 |       00:00:08 |      100.00% |  -0.0000e+00 |          0.0010 |
|      20 |         200 |       00:00:10 |       90.63% |       1.4946 |          0.0010 |
|=========================================================================================|
Elapsed time is 14.135101 seconds.
Elapsed time is 0.297016 seconds.

MiniBatchSize = 32
InitialLearRate = 0.001000
MaxEpochs = 20

Accuracy = 0.900270
```

Task 4:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       43.75% |       0.9595 |          0.0010 |
|       5 |          50 |       00:00:03 |       96.88% |       0.1702 |          0.0010 |
|      10 |         100 |       00:00:05 |      100.00% |       0.0182 |          0.0010 |
|      15 |         150 |       00:00:08 |       90.63% |       1.4946 |          0.0010 |
|      20 |         200 |       00:00:11 |       96.88% |       0.4982 |          0.0010 |
|=========================================================================================|
Elapsed time is 14.805012 seconds.
Elapsed time is 0.290020 seconds.

MiniBatchSize = 32
InitialLearRate = 0.001000
MaxEpochs = 20
DropOut = 0.300000

Accuracy = 0.943396
```

Task 5:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       53.13% |       0.7564 |          0.0010 |
|       5 |          50 |       00:00:03 |       50.00% |       7.9712 |          0.0010 |
|      10 |         100 |       00:00:05 |       78.13% |       3.4874 |          0.0010 |
|      15 |         150 |       00:00:08 |      100.00% |  -0.0000e+00 |          0.0010 |
|      20 |         200 |       00:00:11 |      100.00% |  -0.0000e+00 |          0.0010 |
|=========================================================================================|
Elapsed time is 15.138445 seconds.
Elapsed time is 0.296884 seconds.

MiniBatchSize = 32
InitialLearRate = 0.001000
MaxEpochs = 20
DropOut = 0.300000
DataAugmentation = randcrop

Accuracy = 0.902965
```

Task 6:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       53.13% |       0.7564 |          0.0010 |
|       5 |          50 |       00:00:03 |       50.00% |       7.9712 |          0.0010 |
|      10 |         100 |       00:00:05 |       78.13% |       3.4874 |          0.0010 |
|      15 |         150 |       00:00:08 |      100.00% |  -0.0000e+00 |          0.0010 |
|      20 |         200 |       00:00:11 |      100.00% |  -0.0000e+00 |          0.0010 |
|=========================================================================================|
Elapsed time is 15.138445 seconds.
Elapsed time is 0.296884 seconds.

MiniBatchSize = 32
InitialLearRate = 0.001000
MaxEpochs = 20
DropOut = 0.300000
DataAugmentation = randcrop

Accuracy = 0.902965
```

Problem 2:

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       40.63% |       0.8327 |      1.0000e-04 |
|       5 |          50 |       00:00:07 |       90.63% |       0.3077 |      1.0000e-04 |
|      10 |         100 |       00:00:15 |       81.25% |       0.4053 |      1.0000e-04 |
|      15 |         150 |       00:00:22 |       84.38% |       0.2561 |      1.0000e-04 |
|      20 |         200 |       00:00:29 |       90.63% |       0.2682 |      1.0000e-04 |
|=========================================================================================|

MiniBatchSize = 32
InitialLearRate = 0.000100
MaxEpochs = 20
WeightLearnRateFactor = 2
BiasLearnRateFactor = 2

Accuracy = 0.964960
```

```
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       43.75% |       1.0861 |      1.0000e-04 |
|       5 |          50 |       00:00:08 |       87.50% |       0.3050 |      1.0000e-04 |
|      10 |         100 |       00:00:15 |       87.50% |       0.4036 |      1.0000e-04 |
|      15 |         150 |       00:00:22 |       87.50% |       0.1761 |      1.0000e-04 |
|      20 |         200 |       00:00:28 |       90.63% |       0.2074 |      1.0000e-04 |
|      25 |         250 |       00:00:35 |       93.75% |       0.1160 |      1.0000e-04 |
|      30 |         300 |       00:00:42 |       93.75% |       0.0941 |      1.0000e-04 |
|      35 |         350 |       00:00:48 |      100.00% |       0.0476 |      1.0000e-04 |
|      40 |         400 |       00:00:55 |       93.75% |       0.1208 |      1.0000e-04 |
|=========================================================================================|

MiniBatchSize = 32
InitialLearRate = 0.000100
MaxEpochs = 40
WeightLearnRateFactor = 3
BiasLearnRateFactor = 3

Accuracy = 0.970350
```

Observations:

In the problem 1, I have observed that just setting the batch size will not give any result. Next, when we set the initial learning rate to 0.1 and 0.01 I faced the same problem so, I decreased it to 0.001 where it fetched me 83%. Next, after adding the crossChannelNormalizationLayer I got an increase in accuracy, also by adding the dropout layer just before the FullyConnectedLayer increased the accuracy a little more, The DataAugmentation technique however, reduced my accuracy to 90% and then finally I made the layer deeper and that gave me the same result as the settings in task 5. Next, in problem 1, I applied the last 3 layers for classifying into face and background. Then set the parameters where, decreasing the initial rate gave me good results, but increasing the WeightLearnRateFactor and BiasLearnRateFactor to 3 gave me much better results. Thus, ***pre-trained > model from scratch!***