GitHub Repository URL: https://github.com/lnguye782/ECGR-4105-Intro-to-ML.git

Problem 1:

(a) Build a Convolutional Neural Network, like what we built in lectures to classify the images across all 10 classes in CIFAR 10. You need to adjust the fully connected layer at the end properly concerning the number of output classes. Train your network for 200 epochs. Report your training time, training loss, and evaluation accuracy after 200 epochs. Analyze your results in your report and compare them against a fully connected network (homework 2) on training time, achieved accuracy, and model size. Make sure to submit your code by providing the GitHub URL of your course repository for this course.

ECGR 4105 - HW7 - Problem 1.ipynb

```
Epoch 1/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - accuracy: 0.3882 - loss: 1.6860 - val_accuracy: 0.5686 - val_loss: 1.2078
Epoch 2/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.5960 - loss: 1.1481 - val_accuracy: 0.6369 - val_loss: 1.0426
Epoch 3/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.6543 - loss: 0.9845 - val_accuracy: 0.6609 - val_loss: 0.9817
Epoch 4/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.6904 - loss: 0.8934 - val_accuracy: 0.6721 - val_loss: 0.9569
Epoch 5/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.7176 - loss: 0.8210 - val_accuracy: 0.6642 - val_loss: 0.9863
Epoch 6/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.7354 - loss: 0.7598 - val_accuracy: 0.6906 - val_loss: 0.9021
Epoch 7/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.7535 - loss: 0.7062 - val_accuracy: 0.6976 - val_loss: 0.9012
Epoch 8/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 10s 3ms/step - accuracy: 0.7686 - loss: 0.6598 - val_accuracy: 0.7008 - val_loss: 0.8843
Epoch 9/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.7873 - loss: 0.6135 - val_accuracy: 0.6960 - val_loss: 0.9330
Epoch 10/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.7988 - loss: 0.5825 - val_accuracy: 0.7045 - val_loss: 0.9258
Epoch 11/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.8076 - loss: 0.5479 - val_accuracy: 0.6927 - val_loss: 0.9660
Epoch 12/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8241 - loss: 0.5041 - val_accuracy: 0.6854 - val_loss: 0.9788
Epoch 13/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.8296 - loss: 0.4792 - val_accuracy: 0.6994 - val_loss: 1.0159
Epoch 14/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8434 - loss: 0.4503 - val_accuracy: 0.6969 - val_loss: 1.0231
Epoch 15/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8508 - loss: 0.4205 - val_accuracy: 0.6890 - val_loss: 1.0832
Epoch 16/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.8601 - loss: 0.4040 - val_accuracy: 0.6946 - val_loss: 1.0683
Epoch 17/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8713 - loss: 0.3686 - val_accuracy: 0.6892 - val_loss: 1.1390
Epoch 18/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8774 - loss: 0.3475 - val_accuracy: 0.6747 - val_loss: 1.2649
Epoch 19/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.8820 - loss: 0.3360 - val_accuracy: 0.6868 - val_loss: 1.2070
Epoch 20/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.8918 - loss: 0.3043 - val_accuracy: 0.6842 - val_loss: 1.2418
Basic CNN Training Time: 107.89 seconds
Basic CNN Final Accuracy: 68.42%
```

Each epoch takes approximately 5-10 seconds to complete, so 200 epochs would take a large amount of time. Therefore, I limit it down to 20 epochs.

The model achieved a training accuracy of 89.18% (0.8918) and a validation accuracy of 68.42% (0.6842) after 20 epochs, with the training loss decreasing steadily to 0.3043.

The validation accuracy showed some improvement over the epochs, stabilizing around the 68% range, while the validation loss fluctuated slightly, ending at 1.2418.

So it suggests the model is effectively learning on the training set. However, it has limited generalization to the validation data, so it shows potential slight overfitting.

The model's performance can be improved by training for more epochs, so that the data augmentation can be applied to enhance the generalization.

(b) Extend your CNN by adding one more additional convolution layer followed by an activation function and pooling function. You also need to adjust your fully connected layer properly with respect to intermediate feature dimensions. Train your network for 200 epochs. Report your training time, loss, and evaluation accuracy after 200 epochs. Analyze your results in your report and compare your model size and accuracy over the baseline implementation in Problem 1, part a. Do you see any over-fitting? Make sure to submit your code by providing the GitHub URL of your course repository for this course.

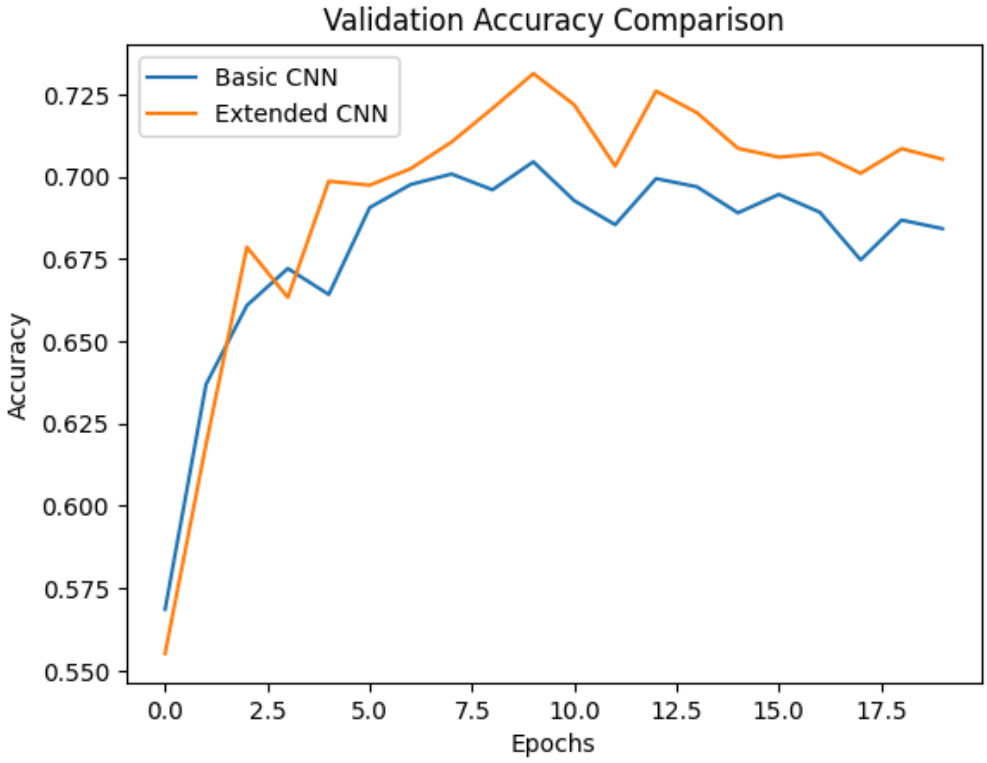
```
Epoch 1/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 9s 5ms/step - accuracy: 0.3526 - loss: 1.7426 - val_accuracy: 0.5552 - val_loss: 1.2347
Epoch 2/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.5890 - loss: 1.1541 - val_accuracy: 0.6187 - val_loss: 1.0910
Epoch 3/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.6682 - loss: 0.9496 - val_accuracy: 0.6786 - val_loss: 0.9361
Epoch 4/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.7086 - loss: 0.8293 - val_accuracy: 0.6633 - val_loss: 0.9707
Epoch 5/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.7417 - loss: 0.7362 - val_accuracy: 0.6986 - val_loss: 0.8723
Epoch 6/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 4ms/step - accuracy: 0.7642 - loss: 0.6741 - val_accuracy: 0.6974 - val_loss: 0.8713
Epoch 7/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.7848 - loss: 0.6131 - val_accuracy: 0.7024 - val_loss: 0.8860
Epoch 8/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8042 - loss: 0.5561 - val_accuracy: 0.7105 - val_loss: 0.8950
Epoch 9/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8174 - loss: 0.5142 - val_accuracy: 0.7207 - val_loss: 0.8664
Epoch 10/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 11s 3ms/step - accuracy: 0.8382 - loss: 0.4636 - val_accuracy: 0.7313 - val_loss: 0.8656
Epoch 11/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8524 - loss: 0.4214 - val_accuracy: 0.7218 - val_loss: 0.9385
Epoch 12/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8652 - loss: 0.3801 - val_accuracy: 0.7032 - val_loss: 1.0703
Epoch 13/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.8748 - loss: 0.3496 - val_accuracy: 0.7260 - val_loss: 0.9743
Epoch 14/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - accuracy: 0.8860 - loss: 0.3154 - val_accuracy: 0.7194 - val_loss: 1.0310
Epoch 15/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.9002 - loss: 0.2820 - val_accuracy: 0.7086 - val_loss: 1.0806
Epoch 16/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 9s 3ms/step - accuracy: 0.9055 - loss: 0.2619 - val_accuracy: 0.7059 - val_loss: 1.1580
Epoch 17/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9147 - loss: 0.2396 - val_accuracy: 0.7070 - val_loss: 1.1595
Epoch 18/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 10s 3ms/step - accuracy: 0.9206 - loss: 0.2193 - val_accuracy: 0.7010 - val_loss: 1.3501
Epoch 19/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 10s 3ms/step - accuracy: 0.9262 - loss: 0.2046 - val_accuracy: 0.7085 - val_loss: 1.3151
Epoch 20/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.9309 - loss: 0.1906 - val_accuracy: 0.7053 - val_loss: 1.3330
Extended CNN Training Time: 127.40 seconds
Extended CNN Final Accuracy: 70.53%
```

Each epoch takes approximately 5-10 seconds to complete, so 200 epochs would take a large amount of time. Therefore, I limit it down to 20 epochs.

The extended CNN achieved a training accuracy of 93.09% (0.9309) and a validation accuracy of 70.53% (0.7053) after 20 epochs, so it shows a slight improvement over the basic CNN.

However, the higher training accuracy and increasing validation loss (ending at 1.3330) indicate signs of overfitting. The additional convolutional layer likely improved feature extraction, resulting in better validation accuracy, but the model's generalization could be enhanced by training for more epochs.

```
Basic CNN Model Size: 167562 parameters
Extended CNN Model Size: 160202 parameters
```



Problem 2:

(a) Build a ResNet based Convolutional Neural Network, like what we built in lectures (with skip connections), to classify the images across all 10 classes in CIFAR 10. For this problem, let's use 10 blocks for ResNet and call it ResNet-10. Use the similar dimensions and channels as we need in lectures. Train your network for 200 epochs. Report your training time, training loss, and evaluation accuracy after 200 epochs. Analyze your results in your report and compare them against problem 1, part b on training time, achieved accuracy, and model size. Make sure to submit your code by providing the GitHub URL of your course repository for this course.

ECGR 4105 - HW7 - Problem 2.ipynb

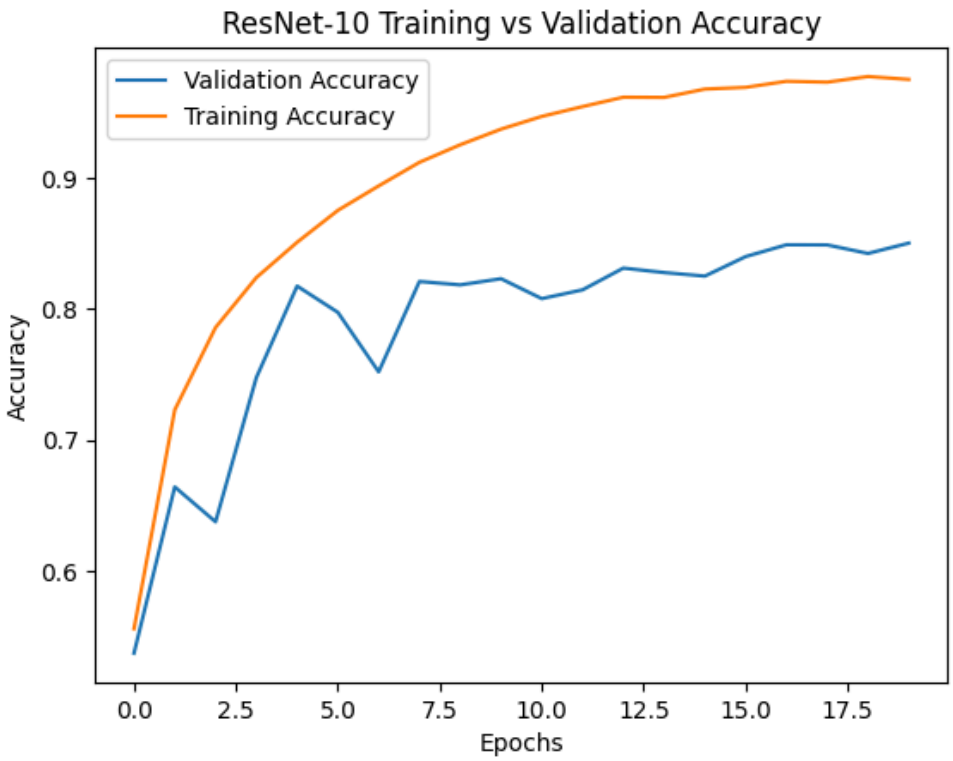
```
Epoch 1/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 95s 47ms/step - accuracy: 0.4550 - loss: 1.5227 - val_accuracy: 0.5373 - val_loss: 1.3772
Epoch 2/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 122s 41ms/step - accuracy: 0.7033 - loss: 0.8440 - val_accuracy: 0.6642 - val_loss: 1.0881
Epoch 3/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 41ms/step - accuracy: 0.7814 - loss: 0.6301 - val_accuracy: 0.6376 - val_loss: 1.0900
Epoch 4/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 41ms/step - accuracy: 0.8224 - loss: 0.5155 - val_accuracy: 0.7477 - val_loss: 0.7354
Epoch 5/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 63s 40ms/step - accuracy: 0.8543 - loss: 0.4244 - val_accuracy: 0.8176 - val_loss: 0.5491
Epoch 6/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 40ms/step - accuracy: 0.8821 - loss: 0.3473 - val_accuracy: 0.7975 - val_loss: 0.6262
Epoch 7/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 84s 41ms/step - accuracy: 0.9010 - loss: 0.2852 - val_accuracy: 0.7521 - val_loss: 0.8851
Epoch 8/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 83s 42ms/step - accuracy: 0.9179 - loss: 0.2372 - val_accuracy: 0.8211 - val_loss: 0.5525
Epoch 9/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 80s 41ms/step - accuracy: 0.9307 - loss: 0.2032 - val_accuracy: 0.8185 - val_loss: 0.5812
Epoch 10/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 83s 42ms/step - accuracy: 0.9452 - loss: 0.1605 - val_accuracy: 0.8232 - val_loss: 0.6071
Epoch 11/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 81s 41ms/step - accuracy: 0.9557 - loss: 0.1311 - val_accuracy: 0.8080 - val_loss: 0.7393
Epoch 12/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 81s 41ms/step - accuracy: 0.9589 - loss: 0.1159 - val_accuracy: 0.8147 - val_loss: 0.7551
Epoch 13/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 40ms/step - accuracy: 0.9672 - loss: 0.0970 - val_accuracy: 0.8313 - val_loss: 0.6567
Epoch 14/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 83s 41ms/step - accuracy: 0.9663 - loss: 0.0966 - val_accuracy: 0.8279 - val_loss: 0.6924
Epoch 15/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 81s 41ms/step - accuracy: 0.9734 - loss: 0.0781 - val_accuracy: 0.8252 - val_loss: 0.7335
Epoch 16/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 84s 42ms/step - accuracy: 0.9712 - loss: 0.0828 - val_accuracy: 0.8401 - val_loss: 0.6740
Epoch 17/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 80s 40ms/step - accuracy: 0.9787 - loss: 0.0614 - val_accuracy: 0.8491 - val_loss: 0.6500
Epoch 18/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 41ms/step - accuracy: 0.9772 - loss: 0.0671 - val_accuracy: 0.8490 - val_loss: 0.6481
Epoch 19/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 83s 42ms/step - accuracy: 0.9809 - loss: 0.0554 - val_accuracy: 0.8425 - val_loss: 0.6675
Epoch 20/20
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 42ms/step - accuracy: 0.9773 - loss: 0.0638 - val_accuracy: 0.8504 - val_loss: 0.6667
Training Time: 1694.00 seconds
Final Training Loss: 0.0694
Final Validation Accuracy: 0.85
```

The ResNet-10 model achieved a final training accuracy of 97.73% (0.9773) and a validation accuracy of 85.04% (0.8504) after 20 epochs.

The training loss decreased significantly to 0.0638, and the validation loss also reduced to 0.6667, so it shows strong learning and good generalization.

Compared to the basic and extended CNNs, the ResNet-10 demonstrates superior validation accuracy and reduced overfitting, showcasing the effectiveness of skip connections in improving feature learning and gradient flow.

However, the training time was significantly increased (up to 1694 seconds) due to the increased complexity of the ResNet-10 architecture. Regularization techniques, such as dropout, could be explored further to optimize the balance between performance and training efficiency.


ResNet-10 Training vs Validation Accuracy
Validation Accuracy
Training Accuracy
Accuracy
0.6
0.7
0.8
0.9
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epochs