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

Problem 1:

(a) Build a fully connected neural network for the housing dataset you did in previous homework. For training and validation use 80% (training) and 20% (validation) split. For this part, only use one hidden layer with 8 nodes. Train your network for as many epochs you want to. Report your training time, training loss, and evaluation accuracy for many epochs. Analyze your results in your report. Make sure to submit your code by providing the GitHub URL of your course repository for this course. Compare your results against the linear regression and support vector regression from previous homework. (15pts)

ECGR 4105 - HW6 - Problem 1.ipynb

```
(9.537226676940918, 25234312790016.0, 4706488.5, 30129436753920.0, 5007497.5)
```

The training results show a training time of approximately 9.54 seconds.

The training loss is quite high at 25,243,127,900,016.0, with a training mean absolute error (MAE) of 4,706,488.5.

The validation loss and validation MAE are also significantly large, at 30,129,436,753,920.0 and 5,007,497.5, respectively.

These high error values suggest that the model is not performing well, possibly due to insufficient training epochs or lack of feature scaling.

(b) Extend your network with two more additional hidden layers, like the example we did in lecture. Train your network for many epochs as needed. Report your training time, training loss, and evaluation accuracy. Analyze your results in your report. Make sure to submit your code by providing the GitHub URL of your course repository for this course. 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? Compare your results against the linear regression and support vector regression from previous homework. Make sure to submit your code by providing the GitHub URL of your course repository for this course. (25pts)

```
{'Extended (3 Hidden Layers)': {'Training Time (s)': 11.375105381011963,
  'Training Loss': 24964273012736.0,
  'Training MAE': 4682190.5,
  'Validation Loss': 29835818696704.0,
  'Validation MAE': 4984039.5}}
```

The extended model with 3 hidden layers achieved a slightly lower training loss (2,496,427,301,2736.0) and training MAE (4,682,190.5) compared to the baseline model with 1 hidden layer (training loss: 25,243,127,900,016.0, training MAE: 4,706,488.5).

The validation loss and MAE also decreased marginally to 29,835,818,696,704.0 and 4,984,039.5, respectively, from the baseline values of 30,129,436,753,920.0 and 5,007,497.5.

The extended model took slightly more training time (11.38 seconds vs. 9.54 seconds), which is expected due to the additional layers.

The improvements in both training and validation metrics indicate a better model fit, though the marginal difference suggests that adding more layers provided diminishing returns.

There is no significant overfitting observed as the gap between training and validation loss remains small, showing that the model generalizes reasonably well to the validation data.

Problem 2:

(a) Use the cancer dataset to build a fully connected neural network to classify the type of cancer (Malignant vs. benign). For training and validation use 80% (training) and 20% (validation) split. For this part, only use one hidden layer with 32 nodes. Train your network for many epochs as needed. Report your training time, training loss, and evaluation accuracy. Analyze your results in your report. Compare your results against the logistic regression and support vector classification from previous homework. Make sure to submit your code by providing the GitHub URL of your course repository for this course. (15pts)

ECGR 4105 - HW6 - Problem 2.ipynb

```
(6.114588499069214, 0.013174124993383884, 0.9561403393745422)
```

The results show a training time of approximately 6.11 seconds, indicating computational efficiency.

The training loss is around 0.013, suggesting the model is learning effectively.

The validation accuracy is high, at 95.6%, indicating consistent performance on unseen data and demonstrating that the model maintains its generalization capability.

These results affirm that the single hidden-layer architecture with 32 nodes is both efficient and accurate for the cancer classification task, balancing computational speed and predictive performance.

(b) Extend your network with two more additional hidden layers, like the example we did in lecture. Train your network for many epochs as needed. Report your training time, training loss, and evaluation accuracy for many epochs as needed. Analyze your results in your report. Make sure to submit your code by providing the GitHub URL of your course repository for this course. Analyze your results in your report and compare your model size and accuracy over the baseline implementation in Problem1. a. Do you see any over-fitting? Compare your results against the logistic regression and support vector classification from previous homework. Make sure to submit your code by providing the GitHub URL of your course repository for this course. (25pts)

```
(7.382623195648193, 0.00022640445968136191, 0.9649122953414917)
```

The results for the extended network with two additional hidden layers demonstrate improved performance.

The training time increased slightly to approximately 7.38 seconds, which is expected due to the added complexity of the model. The training loss dropped significantly to 0.000226, indicating that the model has learned the training data very effectively.

The validation accuracy improved to 96.49%, showing a slight boost in generalization compared to the single-hidden-layer model.

These results highlight that increasing the model complexity with additional hidden layers can lead to marginal improvements in performance while requiring more computational resources.

Problem 3:

(a) Create a fully connected Neural Network for all 10 classes in CIFAR-10 with only one hidden layer with the size of 256. Train your network for 100 epochs. Report your training time, training loss and evaluation accuracy after 100 epochs. Analyze your results in your report. Make sure to submit your code by providing the GitHub URL of your course repository for this course. (25pt)

ECGR 4105 - HW6 - Problem 3.ipynb

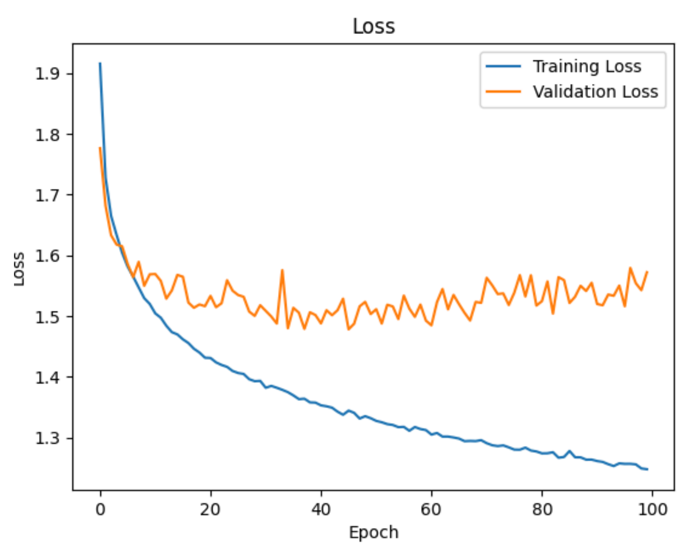
```
Epoch 89/100
782/782 - 10s - 13ms/step - accuracy: 0.5487 - loss: 1.2635 - val_accuracy: 0.4663 - val_loss: 1.5410
Epoch 90/100
782/782 - 10s - 13ms/step - accuracy: 0.5497 - loss: 1.2635 - val_accuracy: 0.4728 - val_loss: 1.5546
Epoch 91/100
782/782 - 11s - 14ms/step - accuracy: 0.5502 - loss: 1.2611 - val_accuracy: 0.4805 - val_loss: 1.5196
Epoch 92/100
782/782 - 11s - 14ms/step - accuracy: 0.5501 - loss: 1.2597 - val_accuracy: 0.4830 - val_loss: 1.5176
Epoch 93/100
782/782 - 19s - 25ms/step - accuracy: 0.5520 - loss: 1.2560 - val_accuracy: 0.4746 - val_loss: 1.5352
Epoch 94/100
782/782 - 11s - 14ms/step - accuracy: 0.5531 - loss: 1.2529 - val_accuracy: 0.4775 - val_loss: 1.5331
Epoch 95/100
782/782 - 21s - 26ms/step - accuracy: 0.5525 - loss: 1.2573 - val_accuracy: 0.4734 - val_loss: 1.5500
Epoch 96/100
782/782 - 20s - 26ms/step - accuracy: 0.5543 - loss: 1.2565 - val_accuracy: 0.4812 - val_loss: 1.5158
Epoch 97/100
782/782 - 12s - 16ms/step - accuracy: 0.5518 - loss: 1.2566 - val_accuracy: 0.4624 - val_loss: 1.5790
Epoch 98/100
782/782 - 11s - 14ms/step - accuracy: 0.5515 - loss: 1.2557 - val_accuracy: 0.4741 - val_loss: 1.5538
Epoch 99/100
782/782 - 20s - 26ms/step - accuracy: 0.5545 - loss: 1.2490 - val_accuracy: 0.4767 - val_loss: 1.5424
Epoch 100/100
782/782 - 11s - 14ms/step - accuracy: 0.5555 - loss: 1.2477 - val_accuracy: 0.4731 - val_loss: 1.5718
Training Time: 1389.30 seconds
```

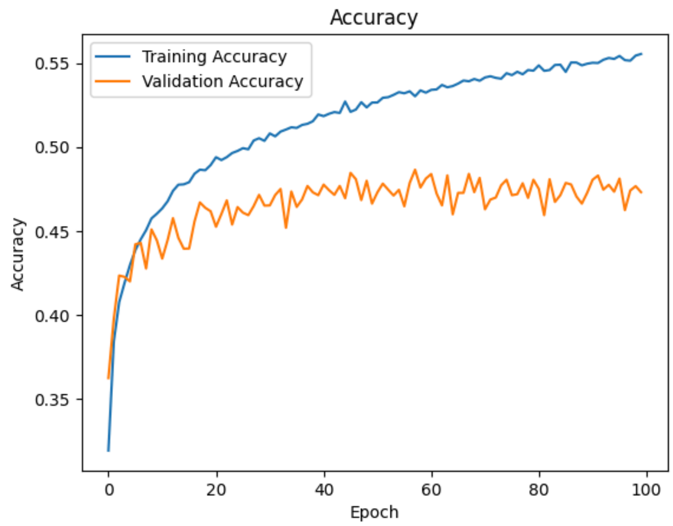
Based on the model/result graph, the training accuracy reaches approximately 55.55% (0.5555) and the validation accuracy gets around 47.31% (0.4731).

The validation loss fluctuates and remains higher than the training loss, showing that the model cannot generalize well to unknown data.

```
Test Loss: 1.5718
Test Accuracy: 0.4731
```

The test accuracy of approximately 47.31% (0.4731) and a test loss of 1.5718 suggests that the model performs slightly better than random guessing, possibly due to CIFAR-10 having 10 classes (baseline random accuracy = 10%).





(b) Extend your network with two more additional hidden layers, like the example we did in lecture. Train your network for 100 epochs. Report your training time, loss, and evaluation accuracy after 100 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. (35pt)

```
Epoch 89/100
782/782 - 20s - 25ms/step - accuracy: 0.6953 - loss: 0.8393 - val_accuracy: 0.4833 - val_loss: 1.8785
Epoch 90/100
782/782 - 22s - 28ms/step - accuracy: 0.6956 - loss: 0.8427 - val_accuracy: 0.4871 - val_loss: 1.8120
Epoch 91/100
782/782 - 12s - 16ms/step - accuracy: 0.6958 - loss: 0.8369 - val_accuracy: 0.4902 - val_loss: 1.8486
Epoch 92/100
782/782 - 12s - 15ms/step - accuracy: 0.6952 - loss: 0.8336 - val_accuracy: 0.4839 - val_loss: 1.8628
Epoch 93/100
782/782 - 12s - 15ms/step - accuracy: 0.6984 - loss: 0.8293 - val_accuracy: 0.4871 - val_loss: 1.8522
Epoch 94/100
782/782 - 23s - 29ms/step - accuracy: 0.6996 - loss: 0.8279 - val_accuracy: 0.4919 - val_loss: 1.8605
Epoch 95/100
782/782 - 18s - 23ms/step - accuracy: 0.6976 - loss: 0.8311 - val_accuracy: 0.4873 - val_loss: 1.8649
Epoch 96/100
782/782 - 12s - 15ms/step - accuracy: 0.6989 - loss: 0.8235 - val_accuracy: 0.4891 - val_loss: 1.8954
Epoch 97/100
782/782 - 11s - 14ms/step - accuracy: 0.7022 - loss: 0.8201 - val_accuracy: 0.4747 - val_loss: 1.9599
Epoch 98/100
782/782 - 23s - 29ms/step - accuracy: 0.7042 - loss: 0.8140 - val_accuracy: 0.4823 - val_loss: 1.9594
Epoch 99/100
782/782 - 12s - 15ms/step - accuracy: 0.7079 - loss: 0.8090 - val_accuracy: 0.4886 - val_loss: 1.8757
Epoch 100/100
782/782 - 12s - 16ms/step - accuracy: 0.7067 - loss: 0.8091 - val_accuracy: 0.4839 - val_loss: 1.9097
Extended Model Training Time: 1637.40 seconds
```

Based on the model/result graph, the training accuracy reaches approximately 70.67% (0.7067) and the validation accuracy gets around 48.39% (0.4839).

```
Extended Model Test Loss: 1.9097
Extended Model Test Accuracy: 0.4839
```

The test accuracy of approximately 48.39% (0.4839) and a test loss of 1.9097.

The extended model outperforms the baseline in terms of training accuracy (70.67% vs. 55.55%) and test accuracy (48.39% vs. 47.31%), indicating better learning capacity due to the additional hidden layers.

However, the extended model suffers from severe overfitting, as the gap between training and validation accuracy is significant, and validation loss increases after early epochs.

The training time for the extended model is also considerably longer (~1637.40 seconds vs. ~1389.30 seconds), reflecting the increased complexity and number of parameters.

While the extended model demonstrates improved expressiveness, the marginal gain in validation accuracy highlights the need for regularization techniques to balance model complexity and generalization.