Nathon Tadeo
Student ID: 801265462
11/27/24
Homework 6
Github:https://github.com/nathon-tadeo/Intro-to-ML/blob/main/homework_6_intro_to_ml.ipynb

Problem 1 (30 pts):

1a. Build a fully connected neural network for the housing dataset you did in the 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 200 epochs. Report your training time, training loss, and evaluation accuracy for many epochs Analyze your results in your report. Compare your results against the linear regression and support vector regression from the previous homework (15pts).

The preprocessing of the data was similar to previous homework. Setting up the neural network was fairly simple. The housing neural network was initially set up with only one hidden layer with 8 nodes. The model was trained with 200 epochs. For the activation, I used the ReLU function rather than the Tanh function. After some research I came to the conclusion that the ReLU function was the better choice because it was a non-saturating function that does not become flat at the extremes of the input range. Given the restriction of 200 epochs, I adjusted the learning rate to be 0.5. This learning rate still didn't fair well and equated to a much higher training loss and validation loss compared to the model from homework 5. With the neural network model my training time was 0.28 seconds, the validation MSE was 20,636,573,368,320, validation R^2 Score was -6.1617. The training loss from this model was 21,750,251,257,856, and the validation loss was 2,063,657,127,116. This was much higher compared to homework 5 linear regression model which had a training loss of 1,196,440,813,568 and validation loss of 968,281,161,728. This discrepancy could be due to the difference in epoch being too large. 200 for the neural network and 5000 for the linear regression model.

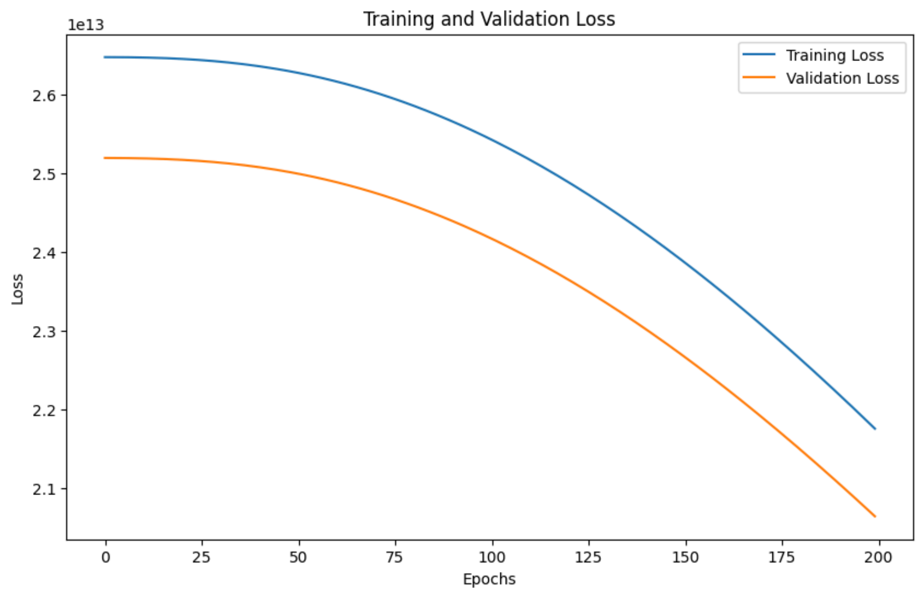


Figure 1.1 Loss for Neural Network Housing Data

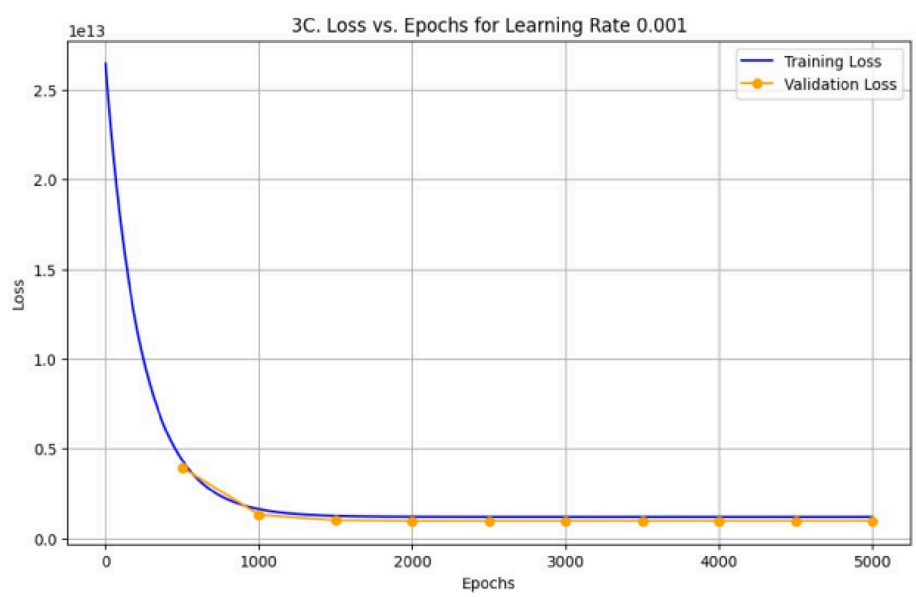
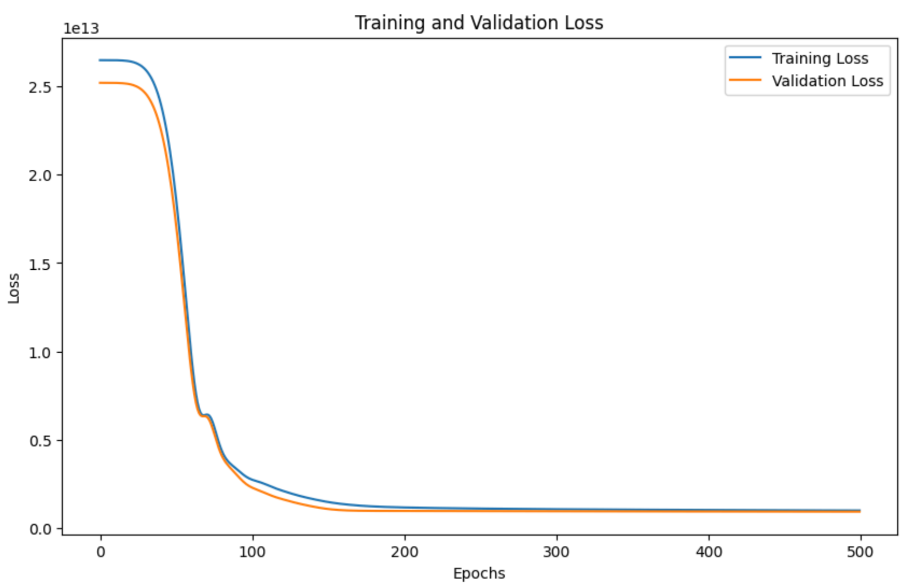


Figure 1.2 Loss for Linear Regression Housing Data

1b. Extend your network with two more additional hidden layers, like the example we did in the lecture. 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 model size and accuracy over the baseline implementation in Problem 1a. Do you see any over-fitting? Compare your results against the linear regression and support vector regression from previous homework (25pts).

Since this model had 3 hidden layers with the ReLU activation function, I decided to train the model with 500 epochs with a learning rate of 0.1. This model had a learning rate that was much lower, but had higher epochs compared to problem 1a. The training time increased from 0.28 seconds to 2.07 seconds when comparing it to part a. This could be attributed to the amount of hidden layers added. The validation MSE was 949,774,516,224, validation R^2 Score was 0.6704. Overall this model was much better than the one from problem 1a. Visually from the graph, I do not see any overfitting. This model was also better compared to the one from homework 5. The training loss for this model was 1,016,315,707, 392, while the one from homework 5 was 1,196,440,813,568. This model had a lower training loss at lower epochs at a learning rate of 0.01. While the linear regression model had a learning rate of 0.001 training over 5000 epochs.



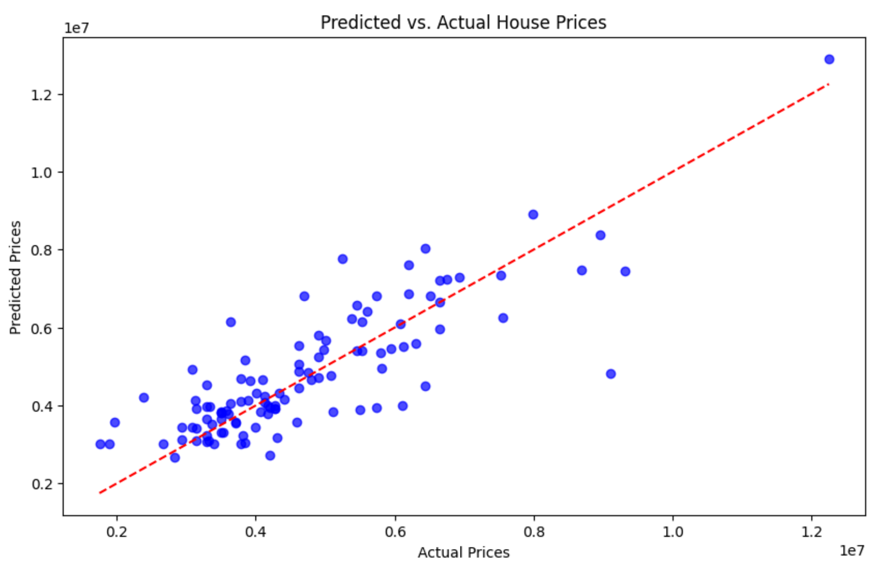


Figure 1.3 Loss for Neural Network Housing Data (3 Hidden Layers)

Problem 2 (40 pts):

2a. Use the cancer dataset to build an SVM classifier 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 (15pts).

The preprocessing was simple and similar to problem 1. The biggest difference for this model was the amount of nodes included in the neural network. To keep a constant with the other models, I decided to train the model over 500 epochs with a training rate of 0.01 similar to problem 1b. This model had a training time of 0.81 seconds and a test accuracy of 0.9649. The training and validation loss was very small being 0.0004 and 0.1514 respectively. Comparing this model to the PCA feature extraction model from homework 4, their accuracy scores were quite high when K=28 being 0.9825. This gives us a difference of 0.0176 between the two model's accuracies.

Figure 2.1 Loss for Neural Network Cancer Dataset (32 Nodes)

Figure 2.2 PCA Model Cancer Dataset (HW 4)

2b. 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. 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. (25pts)

All parameters stayed the same, except two additional hidden layers were added. After training the model over 500 epochs, the training loss and validation loss were 0.0003 and 0.1921 respectively. The training loss is lower than problem 2a, but its validation loss is higher. Overall, I would recommend the model from problem 2a to be sufficient enough because this model also had a longer training time and lower test accuracy. The training time for this model was 2.27 seconds with an accuracy of 0.9561. The model from problem 2a and 2b are both equate to a lower accuracy compared to the model from homework 4.

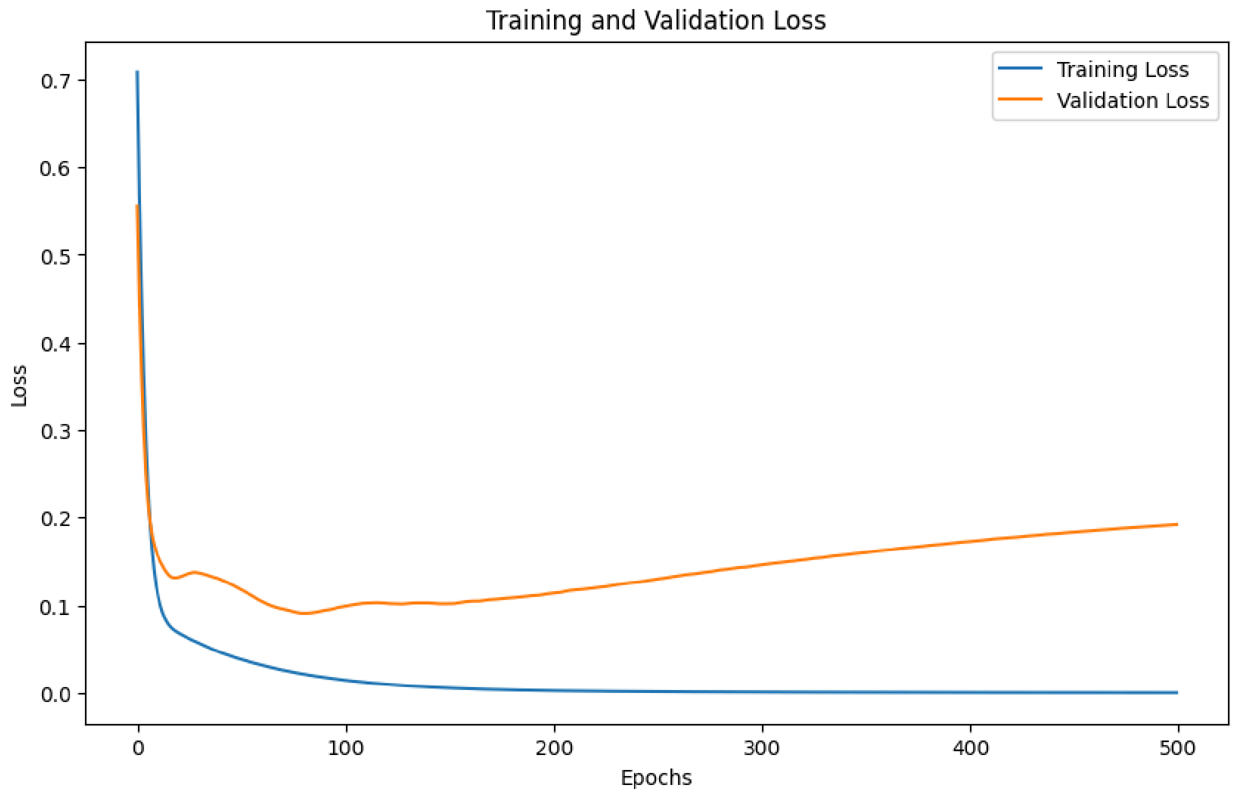


Figure 2.3 Loss for Neural Network Cancer Dataset (32 Nodes 3 Layers)

Problem 3 (40 pts):

3a. 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 (25pt).

Setting up the CIFAR-10 was quite difficult and time consuming because of the pure size of the dataset and download speed. After the dataset was downloaded, the CIFAR-10 dataset had to be preprocessed a little differently because it contained images that were 32x32x3. Each image had a tensor shape of [3,32,32] because the 3 collar channels needed to be accounted for in each 32x32 image. After the normalization, the model was trained with a learning rate of 0.001 and a hidden layer with 256 nodes. Due to the dataset being images and the model training with 256 nodes in the neural network, training was a lengthy process and the model trained for 2037.51 seconds (34 minutes). After 100 epochs, the training loss was 0.1488 and the accuracy was 0.4889.

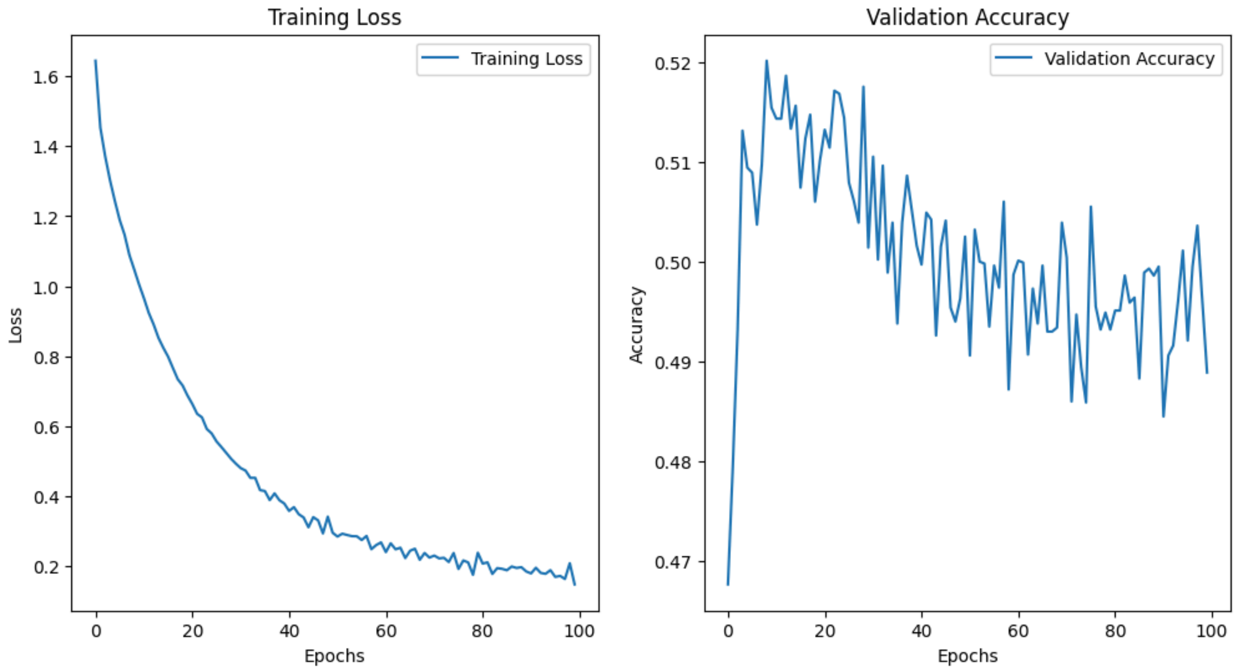


Figure 3.1 CIFAR-10 Training Loss and Validation Accuracy (256 Nodes)

3b. 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 Problem1. a. Do you see any over-fitting? (35pt)

Similarly to problem 1b and 2b, the process was relatively the same. This model was trained with 100 epochs and learning rate of 0.001, except the model was extended to 3 hidden layers. Surprisingly enough this change improved the model only a little bit. The final accuracy after 100 epochs was 0.5198, increasing the accuracy by 0.0309. The training loss was 0.1115, being about 0.0338 lower than the model from 3a. Luckily the training time was only 20 seconds longer than the model from problem 3a. There seems to be no over-fitting in both models as well.

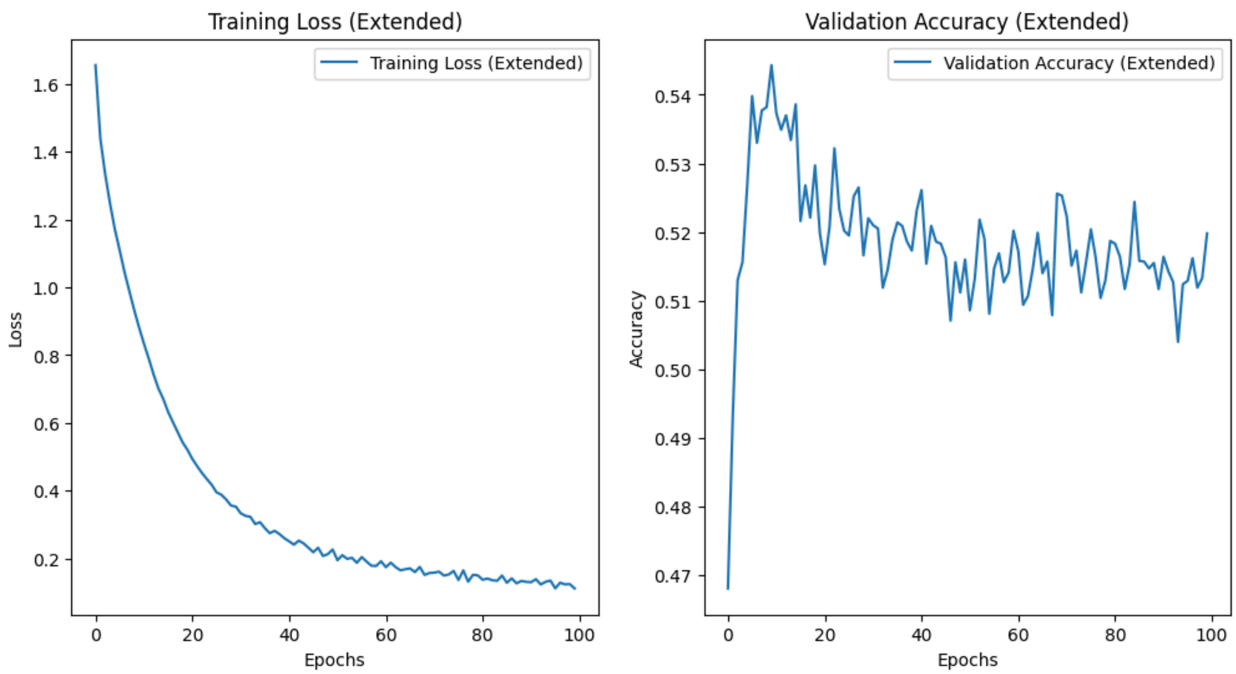


Figure 3.2 CIFAR-10 Training Loss and Validation Accuracy (256 Nodes 3 Hidden Layers)