GitHub Code Repository: <https://github.com/pharmon0/Harmon_ECGR5105>

Graphical user interface, text, application

Description automatically generated

The training loop and model used can be found in the Github repository under directory /HW5/HW5\_Part\_1. The model training progression for four different learning rates over 5000 epochs can be seen below.

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

For accurate comparison, the linear training from the lecture was performed again using my methods, and the results can be found below.

Text

Description automatically generated

The best quadratic model found was the learning rate of 0.1, with a final MSELoss for the whole dataset of 0.0023757393937557936. Comparing this to the final loss of the linear model of 0.0034324584994465113, it can be seen that the quadratic model produced 30.8% less loss than the linear model after 5k epochs.

The visual representation of the quadratic model can be seen below.

Chart, scatter chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

As before, the training loop developed can be found on the Github repository under /HW5/HW5\_Part\_2. The model training progression for the four learning rates can be seen below.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

The best linear model produced was the model with a learning rate of 0.1. This produced the lowest final MSE Loss of 0.0115 after 5k epochs.

Text

Description automatically generated

The single-layer network was constructed and trained with the following results using the Adam optimizer and a learning rate of 0.001

Adam Optimizer, 1-layer net, LR=1e-3

Epoch 1 | Training Loss = 0.02815861813724041 | Validation Loss = 0.03302649408578873

Epoch 2 | Training Loss = 0.027996810153126717 | Validation Loss = 0.03276772424578667

Epoch 3 | Training Loss = 0.02789880894124508 | Validation Loss = 0.03260395675897598

Epoch 50 | Training Loss = 0.025389045476913452 | Validation Loss = 0.02988007850944996

Epoch 100 | Training Loss = 0.02530243806540966 | Validation Loss = 0.029824895784258842

Epoch 150 | Training Loss = 0.025298453867435455 | Validation Loss = 0.02982637658715248

Epoch 200 | Training Loss = 0.025295691564679146 | Validation Loss = 0.02982373721897602

Wall time: 252 ms

Chart

Description automatically generated

This model performed somewhat worse than the non-networked linear model from Problem 2, producing a final error 56.1% greater than the earlier linear model. The neural-net model is not overfit nor is it undertrained (as the loss appears to have completely converged), but it has presented a significantly worse end-result. It did, however, produce this result in far fewer epochs than its linear counterpart. The neural net appears to have converged to roughly its end value in only approximately 50 epochs, where the linear model took ten times more epochs to do so.

A deeper neural net was produced in an attempt to improve the results above. This network was selected to have three layers of 13, 21, and 5 nodes, respectively. An 8, 13, 5 network was tested as well, but the former performed better. The results of this expanded neural network can be seen below.

Text

Description automatically generated

Chart, line chart

Description automatically generated

It can be seen in the final MSE Loss that this expanded model did not outperform the previous, single-layer model. This leads me to believe that this neural network is not successfully improving the training and its added complexity is only worsening the outcome for this specific problem.