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

**Problem 1 (20 pts):**

In our temperature prediction example, let's change our model to a non-linear system. Consider the following description for our model:

w2 * t_u ** 2 + w1 * t_u + b.

1.a Modify the training loop properly to accommodate this redefinition.

1.b Use 5000 epochs for your training. Explore different learning rates from 0.1 to 0.0001 (you need four separate trainings). Report your loss for every 500 epochs per training.

1.c Pick the best non-linear model and compare your final best loss against the linear model that we did during the lecture. For this, visualize the non-linear model against the linear model over the input dataset, as we did during the lecture. Is the actual result better or worse than our baseline linear model?

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.

```
Quadratic Model, Learning Rate = 0.1
Epoch 1 | Training Loss = 1.1159790754318237 | Validation Loss = 1.0119781494140625
Epoch 2 | Training Loss = 0.6310301423072815 | Validation Loss = 0.47319698333740234
Epoch 3 | Training Loss = 0.3597252368927002 | Validation Loss = 0.20560511946678162
Epoch 500 | Training Loss = 0.0025247116573154926 | Validation Loss = 0.001958972541615367
Epoch 1000 | Training Loss = 0.002356366254389286 | Validation Loss = 0.0014451019233092666
Epoch 1500 | Training Loss = 0.002238175133243203 | Validation Loss = 0.0014001831877976656
Epoch 2000 | Training Loss = 0.002155187539756298 | Validation Loss = 0.0016236853552982211
Epoch 2500 | Training Loss = 0.002096919110044837 | Validation Loss = 0.0019943180959671736
Epoch 3000 | Training Loss = 0.00205600680783391 | Validation Loss = 0.002433623420074582
Epoch 3500 | Training Loss = 0.002027279930189252 | Validation Loss = 0.002892129123210907
Epoch 4000 | Training Loss = 0.0020071109756827354 | Validation Loss = 0.0033397795632481575
Epoch 4500 | Training Loss = 0.0019929492846131325 | Validation Loss = 0.003759450279176235
Epoch 5000 | Training Loss = 0.001983006251975894 | Validation Loss = 0.0041424003429710865

Quadratic Model, Learning Rate = 0.01
Epoch 1 | Training Loss = 1.1159790754318237 | Validation Loss = 1.0119781494140625
Epoch 2 | Training Loss = 1.0611895322799683 | Validation Loss = 0.9488234519958496
Epoch 3 | Training Loss = 1.0091272592544556 | Validation Loss = 0.889259397983551
Epoch 500 | Training Loss = 0.006314016878604889 | Validation Loss = 0.032410189509391785
Epoch 1000 | Training Loss = 0.0036123048048466444 | Validation Loss = 0.013132939115166664
Epoch 1500 | Training Loss = 0.0029110193718224764 | Validation Loss = 0.006675686687231064
Epoch 2000 | Training Loss = 0.0027154150884598494 | Validation Loss = 0.004231832455843687
Epoch 2500 | Training Loss = 0.0026481228414922953 | Validation Loss = 0.003179206047207117
Epoch 3000 | Training Loss = 0.002613841090351343 | Validation Loss = 0.0026632335502654314
Epoch 3500 | Training Loss = 0.0025884974747896194 | Validation Loss = 0.002375128213316202
Epoch 4000 | Training Loss = 0.002565988339483738 | Validation Loss = 0.002191972453147173
Epoch 4500 | Training Loss = 0.002544754883274436 | Validation Loss = 0.0020609640050679445
Epoch 5000 | Training Loss = 0.002524381037801504 | Validation Loss = 0.0019581953529268503

Quadratic Model, Learning Rate = 0.001
Epoch 1 | Training Loss = 1.1159790754318237 | Validation Loss = 1.0119781494140625
Epoch 2 | Training Loss = 1.1104371547698975 | Validation Loss = 1.0055698156356812
Epoch 3 | Training Loss = 1.104923129081726 | Validation Loss = 0.9991985559463501
Epoch 500 | Training Loss = 0.103451661765557541 | Validation Loss = 0.020854901522397995
Epoch 1000 | Training Loss = 0.020564958453178406 | Validation Loss = 0.0340411551296711
Epoch 1500 | Training Loss = 0.012678220868110657 | Validation Loss = 0.05485355854034424
Epoch 2000 | Training Loss = 0.010952466167509556 | Validation Loss = 0.05727146938443184
Epoch 2500 | Training Loss = 0.009859978221356869 | Validation Loss = 0.053653694689273834
Epoch 3000 | Training Loss = 0.008940544910728931 | Validation Loss = 0.04879271239042282
Epoch 3500 | Training Loss = 0.008141514845192432 | Validation Loss = 0.04403477907180786
Epoch 4000 | Training Loss = 0.007445048075169325 | Validation Loss = 0.03969774395227432
Epoch 4500 | Training Loss = 0.006837758701294661 | Validation Loss = 0.03582017868757248
Epoch 5000 | Training Loss = 0.006308177951723337 | Validation Loss = 0.03237129747867584
```



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

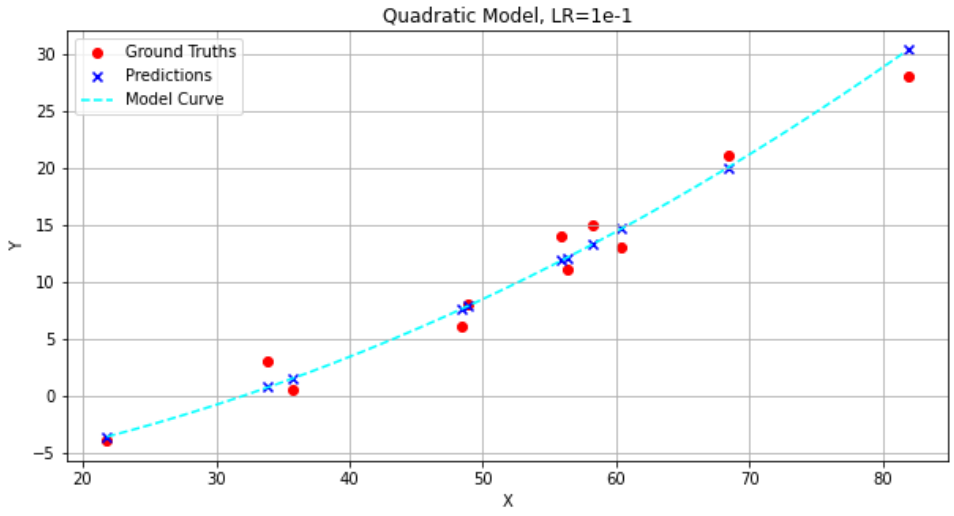
```
Quadratic Model, Learning Rate = 0.0001
Epoch 1 | Training Loss = 1.1159790754318237 | Validation Loss = 1.0119781494140625
Epoch 2 | Training Loss = 1.1154241561889648 | Validation Loss = 1.011336326599121
Epoch 3 | Training Loss = 1.1148697137832642 | Validation Loss = 1.0106948614120483
Epoch 500 | Training Loss = 0.8711451888084412 | Validation Loss = 0.7338039875030518
Epoch 1000 | Training Loss = 0.6804319620132446 | Validation Loss = 0.5259684324264526
Epoch 1500 | Training Loss = 0.5322036147117615 | Validation Loss = 0.3721737861633301
Epoch 2000 | Training Loss = 0.416988343000412 | Validation Loss = 0.2593950033187866
Epoch 2500 | Training Loss = 0.32742494344711304 | Validation Loss = 0.17763149738311768
Epoch 3000 | Training Loss = 0.2577933967113495 | Validation Loss = 0.11921809613704681
Epoch 3500 | Training Loss = 0.20365063846111298 | Validation Loss = 0.0782928392291069
Epoch 4000 | Training Loss = 0.1615433543920517 | Validation Loss = 0.050381358712911606
Epoch 4500 | Training Loss = 0.12878841161727905 | Validation Loss = 0.032078199088573456
Epoch 5000 | Training Loss = 0.10330072790384293 | Validation Loss = 0.020799720659852028
```

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

```
Linear Model, Learning Rate = 0.01
Epoch 1 | Training Loss = 0.4392632246017456 | Validation Loss = 0.08251953125
Epoch 2 | Training Loss = 0.42202311754226685 | Validation Loss = 0.076636403799057
Epoch 3 | Training Loss = 0.40559130907058716 | Validation Loss = 0.07138841599225998
Epoch 500 | Training Loss = 0.03325355052947998 | Validation Loss = 0.13743098080158234
Epoch 1000 | Training Loss = 0.01530188787728548 | Validation Loss = 0.06859982758760452
Epoch 1500 | Training Loss = 0.007968182675540447 | Validation Loss = 0.037055011838674545
Epoch 2000 | Training Loss = 0.004972160793840885 | Validation Loss = 0.021979261189699173
Epoch 2500 | Training Loss = 0.0037482124753296375 | Validation Loss = 0.014421489089727402
Epoch 3000 | Training Loss = 0.00324819702655077 | Validation Loss = 0.010439775884151459
Epoch 3500 | Training Loss = 0.003043925855308771 | Validation Loss = 0.008241587318480015
Epoch 4000 | Training Loss = 0.0029604758601635695 | Validation Loss = 0.006978275254368782
Epoch 4500 | Training Loss = 0.0029263850301504135 | Validation Loss = 0.006228701677173376
Epoch 5000 | Training Loss = 0.002912455704063177 | Validation Loss = 0.005773251876235008
```

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.






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

**Problem 2 (30 pts):**

2.a. Develop preprocessing and a training loop to train a linear regression model that predicts housing price based on the following input variables:

area, bedrooms, bathrooms, stories, parking

For this, you need to use the housing dataset. For training and validation use 80% (training) and 20% (validation) split. Identify the best parameters for your linear regression model, based on the above input variables. In this case, you will have six parameters:

U=W5*X5 + W4*X4 + W3*X3 + W2*X2 + W1*X1 + B

2.b Use 5000 epochs for your training. Explore different learning rates from 0.1 to 0.0001 (you need four separate trainings). Report your loss and validation accuracy for every 500 epochs per each training. Pick the best linear model.

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.

```
Learning Rate = 0.1
Epoch 1 | Training Loss = 1.2239868640899658 | Validation Loss = 1.2471128702163696
Epoch 2 | Training Loss = 0.7253420352935791 | Validation Loss = 0.7413965463638306
Epoch 3 | Training Loss = 0.4595836102962494 | Validation Loss = 0.47116509079933167
Epoch 500 | Training Loss = 0.011338168755173683 | Validation Loss = 0.013433423824608326
Epoch 1000 | Training Loss = 0.01107100211083889 | Validation Loss = 0.013250169344246387
Epoch 1500 | Training Loss = 0.011055151000618935 | Validation Loss = 0.01329437643289566
Epoch 2000 | Training Loss = 0.011054002679884434 | Validation Loss = 0.01331313606351614
Epoch 2500 | Training Loss = 0.011053916066884995 | Validation Loss = 0.013318805955350399
Epoch 3000 | Training Loss = 0.011053909547626972 | Validation Loss = 0.01332040410488844
Epoch 3500 | Training Loss = 0.011053909547626972 | Validation Loss = 0.013320842757821083
Epoch 4000 | Training Loss = 0.011053908616304398 | Validation Loss = 0.013320968486368656
Epoch 4500 | Training Loss = 0.011053909547626972 | Validation Loss = 0.01332099549472332
Epoch 5000 | Training Loss = 0.011053909547626972 | Validation Loss = 0.013321002013981342

Learning Rate = 0.01
Epoch 1 | Training Loss = 1.2239868640899658 | Validation Loss = 1.2471128702163696
Epoch 2 | Training Loss = 1.1670957803726196 | Validation Loss = 1.1894673109054565
Epoch 3 | Training Loss = 1.1132428646087646 | Validation Loss = 1.1348906755447388
Epoch 500 | Training Loss = 0.04896058142185211 | Validation Loss = 0.052956003695726395
Epoch 1000 | Training Loss = 0.021924735978245735 | Validation Loss = 0.025351444259285927
Epoch 1500 | Training Loss = 0.015495523810386658 | Validation Loss = 0.01833866350352764
Epoch 2000 | Training Loss = 0.013500482775270939 | Validation Loss = 0.01599494181573391
Epoch 2500 | Training Loss = 0.0126235606148839 | Validation Loss = 0.014928413555026054
Epoch 3000 | Training Loss = 0.012125825509428978 | Validation Loss = 0.014328173361718655
Epoch 3500 | Training Loss = 0.011806707829236984 | Validation Loss = 0.013953054323792458
Epoch 4000 | Training Loss = 0.011591452173888683 | Validation Loss = 0.013707597739994526
Epoch 4500 | Training Loss = 0.011442586779594421 | Validation Loss = 0.013543758541345596
Epoch 5000 | Training Loss = 0.011337925679981709 | Validation Loss = 0.013433655723929405

Learning Rate = 0.001
Epoch 1 | Training Loss = 1.2239868640899658 | Validation Loss = 1.2471128702163696
Epoch 2 | Training Loss = 1.2182273864746094 | Validation Loss = 1.2412774562835693
Epoch 3 | Training Loss = 1.2124992609024048 | Validation Loss = 1.2354736328125
Epoch 500 | Training Loss = 0.22195826470851898 | Validation Loss = 0.2284213751554489
Epoch 1000 | Training Loss = 0.1378132402896881 | Validation Loss = 0.14161814749240875
Epoch 1500 | Training Loss = 0.11642010509967804 | Validation Loss = 0.11988845467567444
Epoch 2000 | Training Loss = 0.10145088285207748 | Validation Loss = 0.10501861572265625
Epoch 2500 | Training Loss = 0.08889636397361755 | Validation Loss = 0.09261579811573029
Epoch 3000 | Training Loss = 0.07819823920726776 | Validation Loss = 0.08204194158315659
Epoch 3500 | Training Loss = 0.06906675547361374 | Validation Loss = 0.07299567759037018
Epoch 4000 | Training Loss = 0.06126762181520462 | Validation Loss = 0.06524623930454254
Epoch 4500 | Training Loss = 0.05460244044661522 | Validation Loss = 0.05860123783349991
Epoch 5000 | Training Loss = 0.04890259727835655 | Validation Loss = 0.05289783701300621
```



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

```
Learning Rate = 0.0001
Epoch 1 | Training Loss = 1.2239868640899658 | Validation Loss = 1.2471128702163696
Epoch 2 | Training Loss = 1.2234102487564087 | Validation Loss = 1.2465283870697021
Epoch 3 | Training Loss = 1.2228338718414307 | Validation Loss = 1.245944619178772
Epoch 500 | Training Loss = 0.9717539548873901 | Validation Loss = 0.9914497137069702
Epoch 1000 | Training Loss = 0.7787889838218689 | Validation Loss = 0.7956739068031311
Epoch 1500 | Training Loss = 0.6313768029212952 | Validation Loss = 0.6459552645683289
Epoch 2000 | Training Loss = 0.5186406970024109 | Validation Loss = 0.5313202738761902
Epoch 2500 | Training Loss = 0.43230369687080383 | Validation Loss = 0.4434148371219635
Epoch 3000 | Training Loss = 0.3660649359226227 | Validation Loss = 0.37587693333625793
Epoch 3500 | Training Loss = 0.3151301443576813 | Validation Loss = 0.3238632082939148
Epoch 4000 | Training Loss = 0.27585095167160034 | Validation Loss = 0.28368592262268066
Epoch 4500 | Training Loss = 0.24545010924339294 | Validation Loss = 0.25253599882125854
Epoch 5000 | Training Loss = 0.22181402146816254 | Validation Loss = 0.22827421128749847
```

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.



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

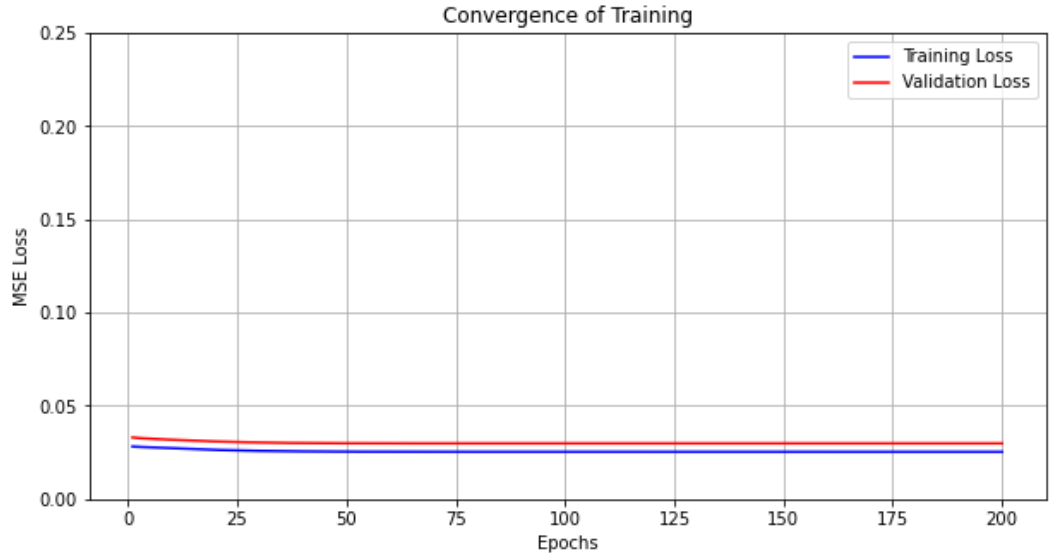
**Problem 3 (50 pts):**

3.a Build a fully connected neural network for the housing dataset you did in previous problem. 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 after 200 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. (15pts)

3.b Extend your network with two more additional hidden layers, like the example we did in lecture. Train your network for 200 epochs. Report your training time, training loss, and evaluation accuracy after 200 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. 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? Make sure to submit your code by providing the GitHub URL of your course repository for this course. (25pts)

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 = 0.028158618137240 41 | Validation Loss = 0.033026494085788 73
Epoch 2 | Training Loss = 0.027996810153126717 | Validation Loss = 0.03276772424578667
Epoch 3 | Training Loss = 0.027898808941245 08 | Validation Loss = 0.03260395675897598
Epoch 50 | Training Loss = 0.025389045476913452 | Validation Loss = 0.02988007850944996
Epoch 100 | Training Loss = 0.025302438065409 66 | 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
```




```
Final Training Loss = 0.025295691564679146 | Final Validation Loss = 0.02982373721897602
Model MSE Loss for whole dataset = 0.02620089240372181
```

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

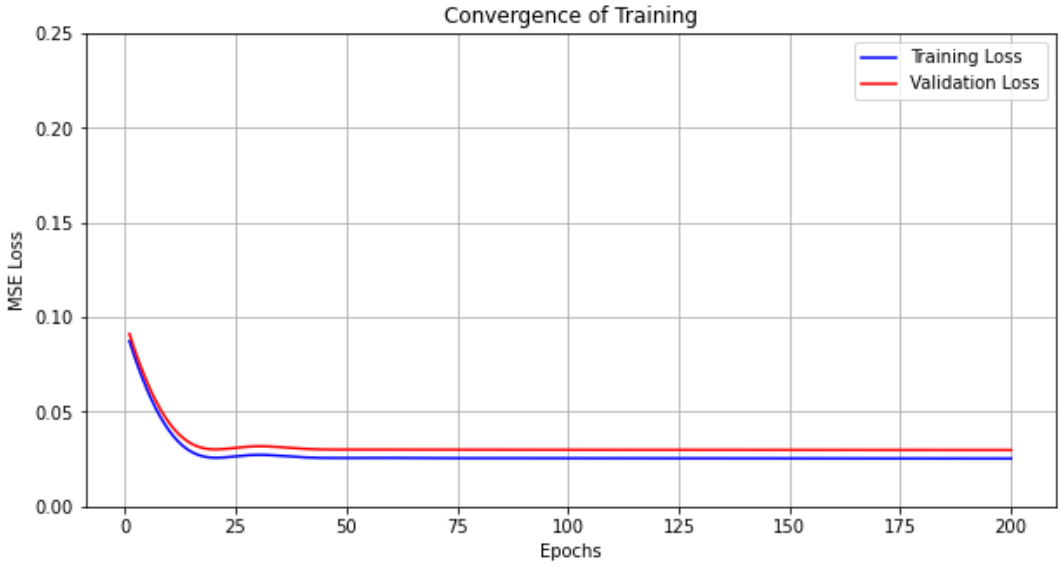


GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105

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.

```
Adam Optimizer, 3-layer net [in->13->21->5->out], LR=1e-3
Epoch 1 | Training Loss = 0.08736804872751236 | Validation Loss = 0.09115060418844223
Epoch 2 | Training Loss = 0.08022507280111313 | Validation Loss = 0.08406160771846771
Epoch 3 | Training Loss = 0.07354738563299179 | Validation Loss = 0.07743436098098755
Epoch 50 | Training Loss = 0.025625130161643028 | Validation Loss = 0.03006667084991932
Epoch 100 | Training Loss = 0.0254366397857666 | Validation Loss = 0.02992679923772812
Epoch 150 | Training Loss = 0.025343503803014755 | Validation Loss = 0.029858145862817764
Epoch 200 | Training Loss = 0.025298304855823517 | Validation Loss = 0.02982133999466896
Wall time: 333 ms
```




```
Final Training Loss = 0.025298304855823517 | Final Validation Loss = 0.02982133999466896
Model MSE Loss for whole dataset = 0.026202790439128876
```

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.



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