Problem 1:

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

$$w2 \cdot t_u^{\,2} + w1 \cdot t_u + b$$

(a) Modify the training loop properly to accommodate this redefinition.

ECGR 4105 - HW5 - Problem 1.ipynb

(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.

```
Training with learning rate: 0.1
Epoch 0, Loss: 52230.57421875
Epoch 500, Loss: nan
Epoch 1000, Loss: nan
Epoch 1500, Loss: nan
Epoch 2000, Loss: nan
Epoch 2500, Loss: nan
Epoch 3000, Loss: nan
Epoch 3500, Loss: nan
Epoch 4000, Loss: nan
Epoch 4500, Loss: nan

Training with learning rate: 0.01
Epoch 0, Loss: 56430.42578125
Epoch 500, Loss: nan
Epoch 1000, Loss: nan
Epoch 1500, Loss: nan
Epoch 2000, Loss: nan
Epoch 2500, Loss: nan
Epoch 3000, Loss: nan
Epoch 3500, Loss: nan
Epoch 4000, Loss: nan
Epoch 4500, Loss: nan
```

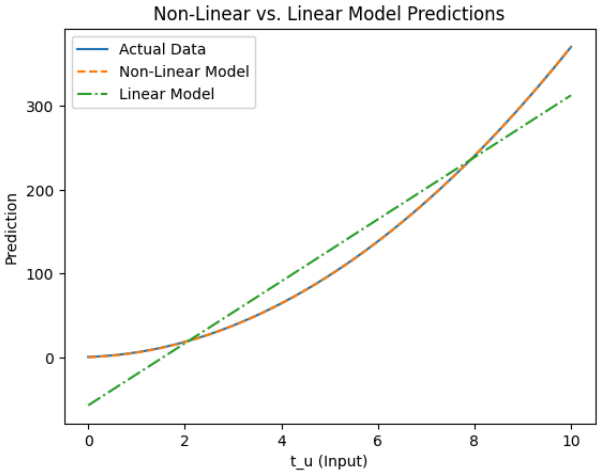
```
Training with learning rate: 0.001
Epoch 0, Loss: 35192.46875
Epoch 500, Loss: nan
Epoch 1000, Loss: nan
Epoch 1500, Loss: nan
Epoch 2000, Loss: nan
Epoch 2500, Loss: nan
Epoch 3000, Loss: nan
Epoch 3500, Loss: nan
Epoch 4000, Loss: nan
Epoch 4500, Loss: nan

Training with learning rate: 0.0001
Epoch 0, Loss: 16955.552734375
Epoch 500, Loss: 2.676705837249756
Epoch 1000, Loss: 1.656842827796936
Epoch 1500, Loss: 1.0265941619873047
Epoch 2000, Loss: 0.6370910406112671
Epoch 2500, Loss: 0.3963528871536255
Epoch 3000, Loss: 0.2475438266992569
Epoch 3500, Loss: 0.15553589165210724
Epoch 4000, Loss: 0.09863147884607315
Epoch 4500, Loss: 0.06342044472694397
```

For the learning rates of 0.1, 0.01, and 0.001, the loss quickly becomes nan (not a number), due to the gradients becoming too large and causing instability in the training process (the learning rate is too high, leading to divergence instead of convergence).

For the learning rate of 0.0001, the loss does not become nan and decreases. It continues to decrease over the epochs, showing that the model is converging. By the end of 5000 epochs, the loss reaches a small value, indicating successful convergence.

Therefore, the best learning rate for convergence is 0.0001.

(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?



For the temperature prediction task, the non-linear model performs better than the linear model, as demonstrated by the graph.

The non-linear model outperforms the linear model in minimizing the prediction error, as evidenced by its reduced final loss. On the other hand, the linear model differs significantly from the real data, especially in areas where the correlations are not linear.

The non-linear model is a better option than the baseline linear model because it captures the quadratic trend in the data and offers a far better fit overall.

Problem 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.

ECGR 4105 - HW5- Problem 2.ipynb

(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.

The best linear model I picked is the Linear Regression Model using Gradient Descent.

```
Training with Learning Rate: 0.1
Epoch 500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 1000/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 1500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 2000/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 2500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 3000/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 3500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 4000/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 4500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662
Epoch 5000/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3662

Training with Learning Rate: 0.01
Epoch 500/5000: Train Loss = 1350009010876.5654, Validation Loss = 2292813178774.1748
Epoch 1000/5000: Train Loss = 1350008211333.5713, Validation Loss = 2292721692029.6787
Epoch 1500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721546204.0869
Epoch 2000/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545726.9561
Epoch 2500/5000: Train Loss = 1350008211326.5803, Validation Loss = 2292721545725.3770
Epoch 3000/5000: Train Loss = 1350008211326.5801, Validation Loss = 2292721545725.3726
Epoch 3500/5000: Train Loss = 1350008211326.5801, Validation Loss = 2292721545725.3726
Epoch 4000/5000: Train Loss = 1350008211326.5801, Validation Loss = 2292721545725.3726
Epoch 4500/5000: Train Loss = 1350008211326.5801, Validation Loss = 2292721545725.3726
Epoch 5000/5000: Train Loss = 1350008211326.5801, Validation Loss = 2292721545725.3726

Training with Learning Rate: 0.001
Epoch 500/5000: Train Loss = 4428910450828.4600, Validation Loss = 6026504766422.3066
Epoch 1000/5000: Train Loss = 1767420758090.9146, Validation Loss = 2894953483806.9399
Epoch 1500/5000: Train Loss = 1407960624252.7695, Validation Loss = 2413169011139.9165
Epoch 2000/5000: Train Loss = 1358342820260.3408, Validation Loss = 2323576806063.2661
Epoch 2500/5000: Train Loss = 1351290148936.9075, Validation Loss = 2302396127424.1353
Epoch 3000/5000: Train Loss = 1350229453600.0400, Validation Loss = 2296156481221.6343
Epoch 3500/5000: Train Loss = 1350052956926.5608, Validation Loss = 2294032466154.8330
Epoch 4000/5000: Train Loss = 1350018866646.1482, Validation Loss = 2293247896566.0298
Epoch 4500/5000: Train Loss = 1350011086718.7402, Validation Loss = 2292942769433.6924
Epoch 5000/5000: Train Loss = 1350009048453.4460, Validation Loss = 2292818935060.5659

Training with Learning Rate: 0.0001
Epoch 500/5000: Train Loss = 20705542685265.9922, Validation Loss = 24839773808034.9570
Epoch 1000/5000: Train Loss = 17056916524493.8281, Validation Loss = 20604240673950.2227
Epoch 1500/5000: Train Loss = 14115695294628.7383, Validation Loss = 17201089604829.7383
Epoch 2000/5000: Train Loss = 11738794306343.0312, Validation Loss = 14458005933892.8828
Epoch 2500/5000: Train Loss = 9813833385830.5566, Validation Loss = 12240607727733.4473
Epoch 3000/5000: Train Loss = 8252043228973.9297, Validation Loss = 10443501933882.5371
Epoch 3500/5000: Train Loss = 6982947086155.4678, Validation Loss = 8983600956273.5117
Epoch 4000/5000: Train Loss = 5950339148635.6992, Validation Loss = 7795091394560.7373
Epoch 4500/5000: Train Loss = 5109218014923.4170, Validation Loss = 6825618923305.2949
Epoch 5000/5000: Train Loss = 4423429686043.9248, Validation Loss = 6033374860840.7676
```

For a learning rate of 0.1, the training and validation loss remain unchanged at high values, showing that the model is not learning effectively.

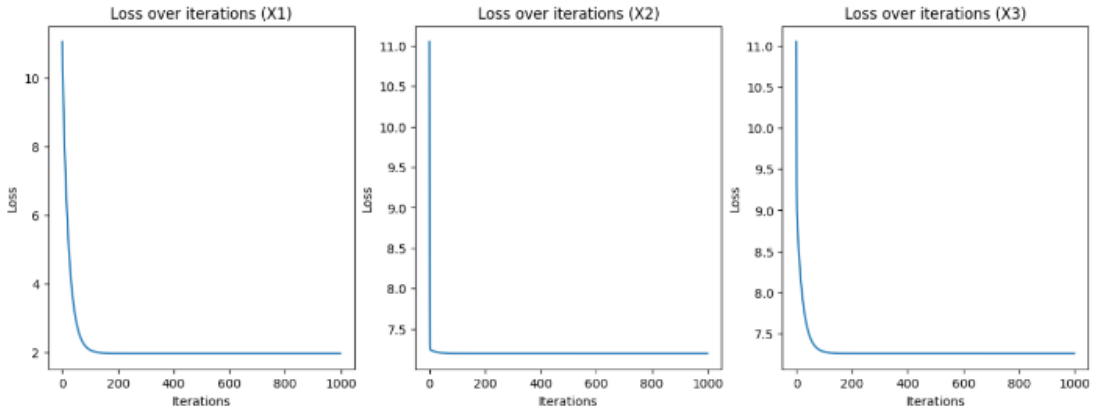
For a learning rate of 0.01, it is similar to the learning rate of 0.1.

For a learning rate of 0.001, the training and validation losses steadily decrease over epochs, indicating gradual and effective learning.

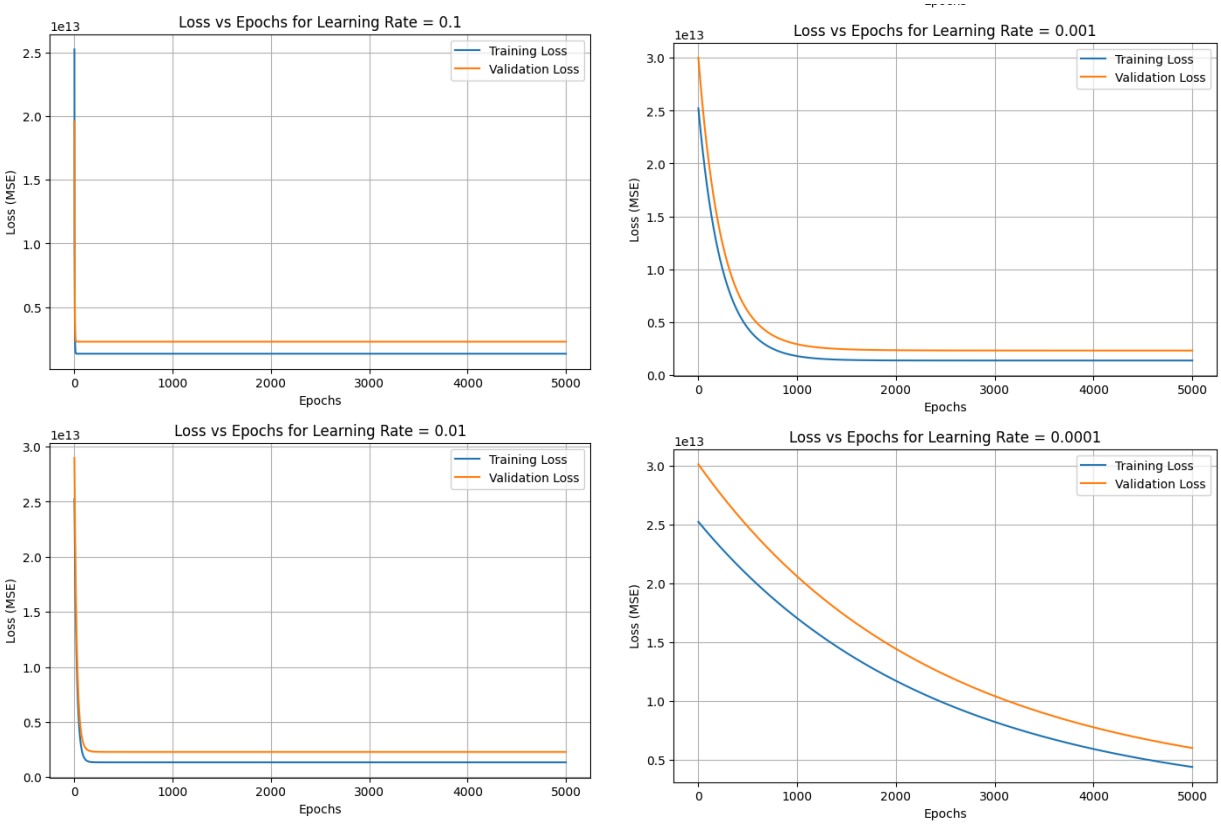
For a learning rate of 0.0001, the loss decreases slowly, showing that the learning rate is too small for efficient training.

(c) Compare your results against the linear regression done in homework 1. Do you see meaningful differences?

From HW1



From this HW5



In Homework 1, linear regression with gradient descent was applied to a simpler dataset with three explanatory variables, both individually and combined, showing that a learning rate of 0.05 provided the best balance between convergence speed and stability.

In Homework 2, a more complex dataset with five variables was used, requiring more iterations for convergence. A learning rate of 0.01 was optimal, as smaller rates slowed convergence while higher rates caused instability.

Problem 3:

Repeat all sections of problem 2, this time use all the input features from the housing price dataset.

(a) Develop preprocessing and a training loop to train a linear regression model that predicts housing price.

For this, you need to use the housing dataset. For training and validation use 80% (training) and 20% (validation) split.

ECGR 4105 - HW5 - Problem 3.ipynb

(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.

```
Training with Learning Rate: 0.1
Epoch 500/5000: Train Loss = 968434161689.2682, Validation Loss = 1752041921466.2380
Epoch 1000/5000: Train Loss = 968358211635.4539, Validation Loss = 1754267726692.2485
Epoch 1500/5000: Train Loss = 968358188448.7123, Validation Loss = 1754317676955.1101
Epoch 2000/5000: Train Loss = 968358188440.7271, Validation Loss = 1754318667921.8079
Epoch 2500/5000: Train Loss = 968358188440.7242, Validation Loss = 1754318686962.3716
Epoch 3000/5000: Train Loss = 968358188440.7242, Validation Loss = 1754318687323.7144
Epoch 3500/5000: Train Loss = 968358188440.7242, Validation Loss = 1754318687330.5369
Epoch 4000/5000: Train Loss = 968358188440.7242, Validation Loss = 1754318687330.6660
Epoch 4500/5000: Train Loss = 968358188440.7242, Validation Loss = 1754318687330.6667
Epoch 5000/5000: Train Loss = 968358188440.7242, Validation Loss = 1754318687330.6667

Training with Learning Rate: 0.01
Epoch 500/5000: Train Loss = 1126812705584.7939, Validation Loss = 1864882938866.8564
Epoch 1000/5000: Train Loss = 1034222523182.0872, Validation Loss = 1783095574037.8821
Epoch 1500/5000: Train Loss = 996025000245.2867, Validation Loss = 1754351165564.1201
Epoch 2000/5000: Train Loss = 980070937678.3365, Validation Loss = 1745953348499.3706
Epoch 2500/5000: Train Loss = 973352621383.3152, Validation Loss = 1744947959221.3137
Epoch 3000/5000: Train Loss = 970502319906.0071, Validation Loss = 1746298753364.7158
Epoch 3500/5000: Train Loss = 969284498205.9944, Validation Loss = 1748113942705.9753
Epoch 4000/5000: Train Loss = 968760708956.8226, Validation Loss = 1749750630628.0725
Epoch 4500/5000: Train Loss = 968534031375.7451, Validation Loss = 1751047495054.4617
Epoch 5000/5000: Train Loss = 968435375202.5380, Validation Loss = 1752015045436.3079

Training with Learning Rate: 0.001
Epoch 500/5000: Train Loss = 1557195750487.2029, Validation Loss = 2509610277593.2358
Epoch 1000/5000: Train Loss = 1312632078636.7546, Validation Loss = 2020294324473.3899
Epoch 1500/5000: Train Loss = 1271585921785.9302, Validation Loss = 1976872130463.8455
Epoch 2000/5000: Train Loss = 1241752937832.5464, Validation Loss = 1958062045888.1929
Epoch 2500/5000: Train Loss = 1216788563819.6497, Validation Loss = 1941054251011.5286
Epoch 3000/5000: Train Loss = 1194870094556.2161, Validation Loss = 1924266472406.2026
Epoch 3500/5000: Train Loss = 1175226584094.1443, Validation Loss = 1908000367923.1870
Epoch 4000/5000: Train Loss = 1157459618181.5737, Validation Loss = 1892582903948.4824
Epoch 4500/5000: Train Loss = 1141318144747.9006, Validation Loss = 1878198875281.1880
Epoch 5000/5000: Train Loss = 1126618061370.3796, Validation Loss = 1864920167418.7158

Training with Learning Rate: 0.0001
Epoch 500/5000: Train Loss = 16065928578311.4355, Validation Loss = 19911901612572.9141
Epoch 1000/5000: Train Loss = 10420133303641.8184, Validation Loss = 13497870348108.5254
Epoch 1500/5000: Train Loss = 6950225612638.8555, Validation Loss = 9450530187395.6895
Epoch 2000/5000: Train Loss = 4815665716233.3779, Validation Loss = 6880434797981.2031
Epoch 2500/5000: Train Loss = 3500842573130.8516, Validation Loss = 5236080909958.4600
Epoch 3000/5000: Train Loss = 2689419370649.0488, Validation Loss = 4174660009824.1914
Epoch 3500/5000: Train Loss = 2187283181208.8916, Validation Loss = 3482439958082.9482
Epoch 4000/5000: Train Loss = 1875294909295.2151, Validation Loss = 3025671940520.8364
Epoch 4500/5000: Train Loss = 1680311488942.8508, Validation Loss = 2720283851060.4966
Epoch 5000/5000: Train Loss = 1557412548047.2776, Validation Loss = 2513144177860.3071
```

For a learning rate of 0.1, the loss values stagnate quickly, indicating that the learning rate is too high for effective training.

For a learning rate of 0.01, the training and validation losses stabilize after around 2000 epochs, demonstrating effective learning and convergence.

For a learning rate of 0.001, the loss decreases steadily but requires more epochs to converge compared to a learning rate of 0.01.

For a learning rate of 0.0001, the loss decreases very slowly, confirming that this learning rate is too small for practical use.

(c) Compare your results against the linear regression done in homework 1. Do you see meaningful differences?

In this problem, linear regression was applied using all features from the housing dataset, including both categorical and numerical variables, making the model more complex than in Homework 1.

While Homework 1 demonstrated effective learning with fewer features and a learning rate of 0.05, Problem 3 required extensive preprocessing (one-hot encoding) and showed optimal performance with a learning rate of 0.01 due to the larger feature set and higher dimensionality.