

Homework 5

GitHub Repository Link: <https://github.com/NaraPvP/IntroToML>

For all the problems, an 80/20 training split was done on the dataset. Along with this, the features were scaled using standardization.

Problem 1:

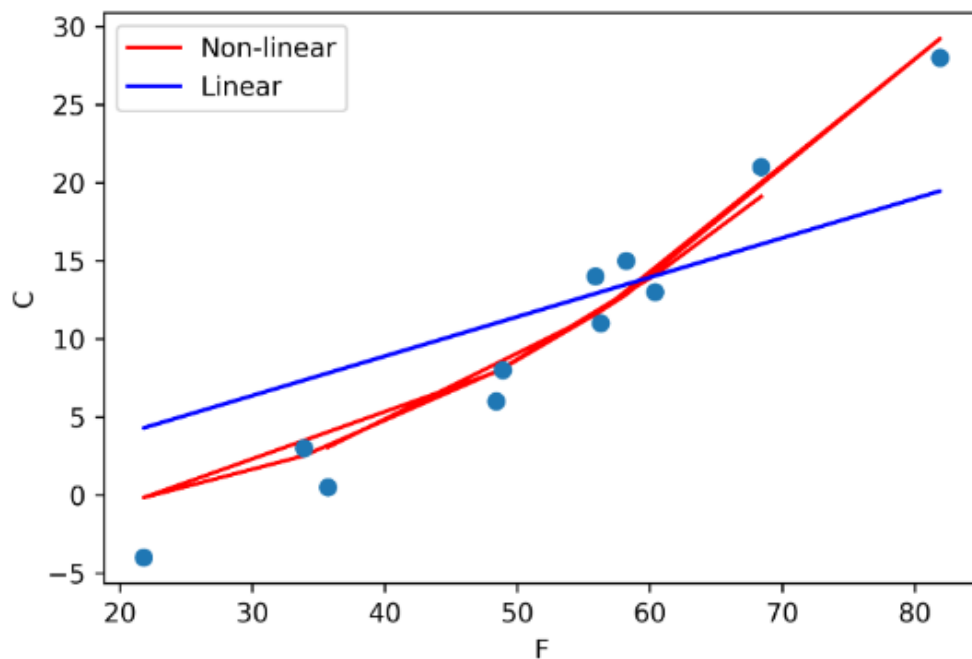
To accommodate for a non-linear system, the following changes were made to the training loop:

- Model equation was changed from $w * t_u + b$ to $w_1 * t_u^2 + w_2 * t_u + b$
- Loss equation stayed the same: $mean(t_p - t_c)^2$
- The training loop function looked similar to the linear version aside from another parameter to feed into model function.

The training loop was performed 4 times with different learning rate varying from 0.1 to 0.0001. The loss was reported every 500 epochs until 5000 epochs have been reached. The following are the calculated losses at 5000 epochs for each loop using the non-linear model:

- Learning Rate of 0.1: nan (too large of a loss)
- Learning Rate of 0.01: nan (too large of a loss)
- Learning Rate of 0.001: nan (too large of a loss)
- Learning Rate of 0.0001: 3.8617%

Compared to the linear model (which reaches a loss of 25.6377%), the loss has decreased with the use of the non-linear model. Here is a plot that visualizes the models with the actual datapoints, which helps determine which model better generalizes the data.



With the provided datapoints, the non-linear model is better at capturing the dataset. Due to a lack of datapoints, the model may not perform as well as the linear due to its nature of having to go away from certain datapoints.

Problem 2:

Due to the increase of input variables, the training loop must be adjusted to account for this change.

- Model equation was changed from the non-linear model $w_1 * t_u^2 + w_2 * t_u + b$ to the linear model $w_5 * x_5 + w_4 * x_4 + w_3 * x_3 + w_2 * x_2 + w_1 * x_1 + b$
- Loss equation stayed the same: $mean(t_p - t_c)^2$
- The training loop function looked similar to the other versions aside from further parameters to feed into model function.

After performing the training loop (similarly as to what was done in Problem 1), the best parameters seemed to come from the ones trained with a learning rate of 0.1. This was concluded since the training and validation losses do not go any lower than the ones calculated for that learning rate. The model equation from the trained parameters would be the following:

$$U = 0.4027x_5 + 0.0904x_4 + 0.3196x_3 + 0.244x_2 + 0.1612x_1 + 0.0037$$

Here are the losses for every training loop iteration:

- Learning Rate: 0.1
 - Epoch 500: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 1000: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 1500: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 2000: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 2500: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 3000: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 3500: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 4000: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 4500: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
 - Epoch 5000: Training Loss = 0.4656211385036159, Validation Loss = 0.33612493606266414
- Learning Rate: 0.01
 - Epoch 500: Training Loss = 0.4656211911638495, Validation Loss = 0.33608939914455876
 - Epoch 1000: Training Loss = 0.4656211385053994, Validation Loss = 0.3361246801262018
 - Epoch 1500: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 2000: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 2500: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 3000: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 3500: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 4000: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 4500: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
 - Epoch 5000: Training Loss = 0.4656211385053993, Validation Loss = 0.33612468017729846
- Learning Rate: 0.001
 - Epoch 500: Training Loss = 0.5958591135965695, Validation Loss = 0.5246754716962263
 - Epoch 1000: Training Loss = 0.4702922581622703, Validation Loss = 0.3443575953609255
 - Epoch 1500: Training Loss = 0.46617636499456966, Validation Loss = 0.33440361284543263
 - Epoch 2000: Training Loss = 0.46575331713992985, Validation Loss = 0.3344553617000288
 - Epoch 2500: Training Loss = 0.4656565555889337, Validation Loss = 0.33514112657788514
 - Epoch 3000: Training Loss = 0.4656308043796804, Validation Loss = 0.3355980888324366
 - Epoch 3500: Training Loss = 0.4656237901638312, Validation Loss = 0.33585066519656703

- Epoch 4000: Training Loss = 0.4656218677693654, Validation Loss = 0.3359836149290369
- Epoch 4500: Training Loss = 0.46562133949321666, Validation Loss = 0.33605247592957704
- Epoch 5000: Training Loss = 0.46562119412966596, Validation Loss = 0.3360877949161908
- Learning Rate: 0.0001
 - Epoch 500: Training Loss = 4.234316403132837, Validation Loss = 4.558631478859912
 - Epoch 1000: Training Loss = 3.047319484760979, Validation Loss = 3.274183271843407
 - Epoch 1500: Training Loss = 2.235223821323195, Validation Loss = 2.387479698023647
 - Epoch 2000: Training Loss = 1.6794750637026417, Validation Loss = 1.7739929439900128
 - Epoch 2500: Training Loss = 1.2990205510347188, Validation Loss = 1.348407999508614
 - Epoch 3000: Training Loss = 1.0384622245113642, Validation Loss = 1.0522559859839897
 - Epoch 3500: Training Loss = 0.8599228673590468, Validation Loss = 0.8454234567758347
 - Epoch 4000: Training Loss = 0.7375058628429059, Validation Loss = 0.7003660886166537
 - Epoch 4500: Training Loss = 0.6535032127781812, Validation Loss = 0.5981467494850922
 - Epoch 5000: Training Loss = 0.5958039789120795, Validation Loss = 0.5257262447050344

Problem 3:

For Part A, a fully connected neural network was built with one hidden layer. The training time was instantaneous compared to the Problem 1 and 2. On Epoch 200, the training loss was 0.4372 and the validation loss was 0.3327. As shown below, the validation loss begins to diverge as training loss continues converge. This means that the model is starting to overfit the training data.

```
Epoch 50: Training Loss = 0.47148941289167073, Validation Loss = 0.33611141205837697
Epoch 100: Training Loss = 0.450174478223789, Validation Loss = 0.3290381312005883
Epoch 150: Training Loss = 0.44366449117398926, Validation Loss = 0.33176357222739494
Epoch 200: Training Loss = 0.43715101325502864, Validation Loss = 0.3327354334632003
```

For Part B, two more hidden layers were added to the neural network. Similarly, the training time seemed instant. On Epoch 200, the training loss was 0.4151 and the validation loss was 0.3367. Compared to one hidden layer, this model was harder to prevent from overfitting and decrease overall loss. As shown in the results below, the validation loss begins to slightly diverge while training loss is at the lowest out of the two neural networks tested. Due to the increase in hidden layers, the loss was decreased while keeping the same amount of validation loss.

```
Epoch 50: Training Loss = 0.46459866631956587, Validation Loss = 0.338609394025321
Epoch 100: Training Loss = 0.44075205098537407, Validation Loss = 0.3332733334843175
Epoch 150: Training Loss = 0.42625675727643253, Validation Loss = 0.3354831613873975
Epoch 200: Training Loss = 0.41510950814978975, Validation Loss = 0.3367185847365864
```