https://github.com/Christian-Martens-UNCC/ECGR-4105/tree/main/Homework_5-Pytorch_%26_%20Neural_Networks

Problem 1:

A)

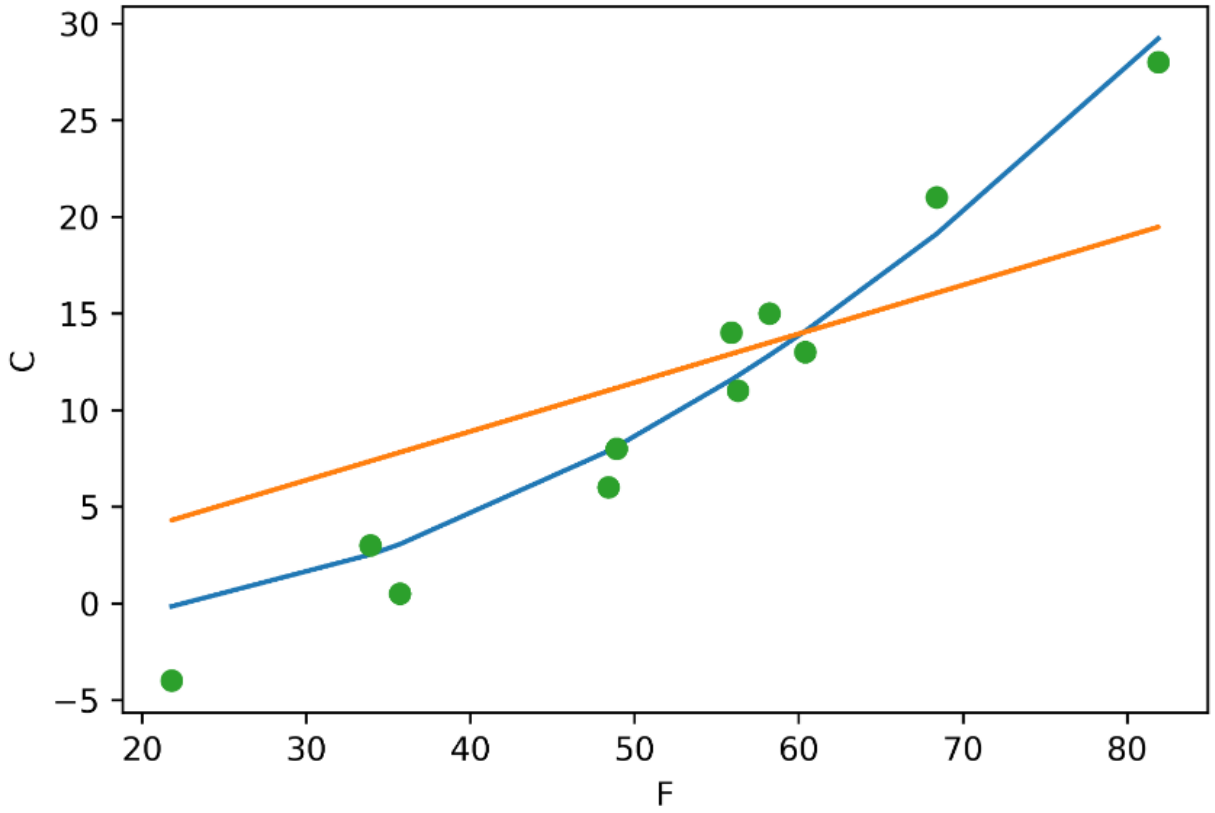
```
def model(t_u, w1, w2, b):
    return w1 * t_u ** 2 + w2 * t_u + b
```

B)

```
=============== Learning Rate = 0.1 ===============
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
Epoch 5000 --> Loss = nan
=============== Learning Rate = 0.01 ===============
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
Epoch 5000 --> Loss = nan
=============== Learning Rate = 0.001 ===============
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
Epoch 5000 --> Loss = nan
=============== Learning Rate = 0.0001 ===============
Epoch 500 --> Loss = 10.708597183227539
Epoch 1000 --> Loss = 8.642083168029785
Epoch 1500 --> Loss = 7.1710052490234375
Epoch 2000 --> Loss = 6.123476028442383
Epoch 2500 --> Loss = 5.377227783203125
Epoch 3000 --> Loss = 4.845285892486572
Epoch 3500 --> Loss = 4.465786933898926
Epoch 4000 --> Loss = 4.194724082946777
Epoch 4500 --> Loss = 4.0008015632629395
Epoch 5000 --> Loss = 3.8617441654205322
```

C) The quadratic model is a better model than our linear model at the same learning rate and training epochs because it has a lower loss. The loss after 5000 epochs for the quadratic model is 3.86 whereas the loss after 5000 epochs of the linear model is 25.64.

```
=============== Learning Rate = 0.0001 ===============
Epoch 500 --> Loss = 29.505889892578125
Epoch 1000 --> Loss = 28.94377326965332
Epoch 1500 --> Loss = 28.505281448364258
Epoch 2000 --> Loss = 28.074451446533203
Epoch 2500 --> Loss = 27.650876998901367
Epoch 3000 --> Loss = 27.23444366455078
Epoch 3500 --> Loss = 26.82501983642578
Epoch 4000 --> Loss = 26.422496795654297
Epoch 4500 --> Loss = 26.02674674987793
Epoch 5000 --> Loss = 25.637672424316406
```



Problem 2:

A)

```
def lin_model_2(x5, x4, x3, x2, x1, w5, w4, w3, w2, w1, b):
    return w5 * x5 + w4 * x4 + w3 * x3 + w2 * x2 + w1 * x1 + b
```

B) The models all seemed to converge on the same weights. Thus, a learning rate of 0.1 was the best model because it was the quickest to train.

```
=============== Learning Rate = 0.1 ===============
Epoch 500 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 1000 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 1500 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 2000 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 2500 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 3000 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 3500 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 4000 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 4500 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
Epoch 5000 --> Training Loss = 0.4420248393303922, Validation Loss = 0.4312726671658319
The Final Trained Parameters are tensor([ 0.3715,  0.0556,  0.2951,  0.2569,  0.1863, -0.0317],
      dtype=torch.float64)
=============== Learning Rate = 0.01 ===============
Epoch 500 --> Training Loss = 0.44202496642194095, Validation Loss = 0.4313072501209946
Epoch 1000 --> Training Loss = 0.4420248393307736, Validation Loss = 0.43127271665597483
Epoch 1500 --> Training Loss = 0.4420248393303924, Validation Loss = 0.4312726672346113
Epoch 2000 --> Training Loss = 0.4420248393303924, Validation Loss = 0.4312726671659207
Epoch 2500 --> Training Loss = 0.4420248393303924, Validation Loss = 0.431272667165832
Epoch 3000 --> Training Loss = 0.4420248393303923, Validation Loss = 0.43127266716583224
Epoch 3500 --> Training Loss = 0.4420248393303923, Validation Loss = 0.43127266716583224
Epoch 4000 --> Training Loss = 0.4420248393303923, Validation Loss = 0.43127266716583224
Epoch 4500 --> Training Loss = 0.4420248393303923, Validation Loss = 0.43127266716583224
Epoch 5000 --> Training Loss = 0.4420248393303923, Validation Loss = 0.43127266716583224
The Final Trained Parameters are tensor([ 0.3715,  0.0556,  0.2951,  0.2569,  0.1863, -0.0317],
      dtype=torch.float64)
=============== Learning Rate = 0.001 ===============
Epoch 500 --> Training Loss = 0.5600673162546, Validation Loss = 0.5033758127299941
Epoch 1000 --> Training Loss = 0.44770464818335665, Validation Loss = 0.43923141171369245
Epoch 1500 --> Training Loss = 0.4430861834304211, Validation Loss = 0.43542888553802406
Epoch 2000 --> Training Loss = 0.44230695423098887, Validation Loss = 0.43335217957183203
Epoch 2500 --> Training Loss = 0.44210291092882753, Validation Loss = 0.4322975299263287
Epoch 3000 --> Training Loss = 0.44204659972747895, Validation Loss = 0.43178513249342737
Epoch 3500 --> Training Loss = 0.44203091952990914, Validation Loss = 0.43153302092470214
Epoch 4000 --> Training Loss = 0.44202654063257757, Validation Loss = 0.4314064302747236
Epoch 4500 --> Training Loss = 0.4420253159108877, Validation Loss = 0.4313418692554751
Epoch 5000 --> Training Loss = 0.44202497297754434, Validation Loss = 0.43130861380171914
The Final Trained Parameters are tensor([ 0.3713,  0.0558,  0.2948,  0.2569,  0.1865, -0.0317],
      dtype=torch.float64)
=============== Learning Rate = 0.0001 ===============
Epoch 500 --> Training Loss = 4.430014264589534, Validation Loss = 3.2485209868302483
Epoch 1000 --> Training Loss = 3.111623193897309, Validation Loss = 2.297395965098786
Epoch 1500 --> Training Loss = 2.2314200638500035, Validation Loss = 1.666687513984712
Epoch 2000 --> Training Loss = 1.643432778155617, Validation Loss = 1.2488955137944027
Epoch 2500 --> Training Loss = 1.2503674932696385, Validation Loss = 0.9724855336492405
Epoch 3000 --> Training Loss = 0.9873670740769539, Validation Loss = 0.789874620300832
Epoch 3500 --> Training Loss = 0.8111900863754851, Validation Loss = 0.6694273651045203
Epoch 4000 --> Training Loss = 0.693000954146267, Validation Loss = 0.590124055042491
Epoch 4500 --> Training Loss = 0.6135659360842904, Validation Loss = 0.5380092211679076
Epoch 5000 --> Training Loss = 0.5600519466492027, Validation Loss = 0.5038264176074896
The Final Trained Parameters are tensor([ 0.4376,  0.2092,  0.3160,  0.3598,  0.3714, -0.0296],
      dtype=torch.float64)
```

Problem 3:

A) The training time was nearly instant for the linear NN.

```
linear_mod = nn.Sequential(nn.Linear(5, 8), nn.Tanh(), nn.Linear(8, 5))
```

```
For 1 Hidden Layer of 8 nodes:
Epoch 200 --> Training Loss = 1.0446717865453987, Validation Loss = 1.3203037976970269
```

B) Training time was also nearly instant. Since the validation and training losses both decreased, it does not appear that there is much overfitting. However, the previous method used in problem 2 was more accurate/had less loss.

```
linear_mod_2 = nn.Sequential(nn.Linear(5, 8), nn.Tanh(), nn.Linear(8, 12), nn.Tanh(),
                             nn.Linear(12, 8), nn.Tanh(), nn.Linear(8, 5))
```

```
For 3 Hidden Layers of 8, 12, and 8 nodes, respectively:
Epoch 200 --> Training Loss = 0.9786599650175829, Validation Loss = 1.2652099585577825
```