1. The input, x for this problem is the hour (from 5am to 8pm) and the output, y is the expected energy consumption in kW.
2. When doing linear regression with a perceptron the best option for an activation function is a Linear Activation function because it can output from negative infinity to positive infinity.
3. Plots are in the zip file with the code and this file.
   1. The training for perceptron a was the fastest and had the highest learning rate as well as the fewest iterations. It was the only one of the three that consistently got relatively similar results when trained.
   2. The training for perceptron b was much slower than a but it was fairly consistent in its weights when being trained.
   3. Perceptron c was not able to train well and its weights frequently went to negative or positive infinity when training. S
   4. Perceptron A
      1. Number of iterations: 500
      2. Learning rate: 0.001
      3. Data pre-processing steps: None, because the data was already normalized enough for accurate comparison
   5. Perceptron B
      1. Number of iterations: 100000
      2. Learning rate: 0.0001
      3. Data pre-processing steps: None, because the data was already normalized enough for accurate comparison
   6. Perceptron C
      1. Number of iterations: 100000
      2. Learning rate: 0.0001
      3. Data pre-processing steps: None, because the data was already normalized enough for accurate comparison
4. If you used multiple neurons the error could be reduced. If you use a single neuron for each hour or a single neuron for each power of x then you could have a far more accurate model for predicting the consumption at a given time. Both of these methods make the training process quicker because you do not to worry about multiple weights being associated with the same inputs. Since for perceptrons b and c there are multiple weights that depend on the value of x it can cause the models to behave erratically when being trained.