1. **What are the inputs and outputs for this neuron (physical meaning)?**
   - The inputs for this neuron are the hourly energy consumption data from 5:00 AM to 8:00 PM, in 1-hour intervals, for 3 days.
   - The outputs for this neuron are the predicted rates of energy consumption for a 4th day, in 1-hour intervals from 5:00 AM to 8:00 PM.

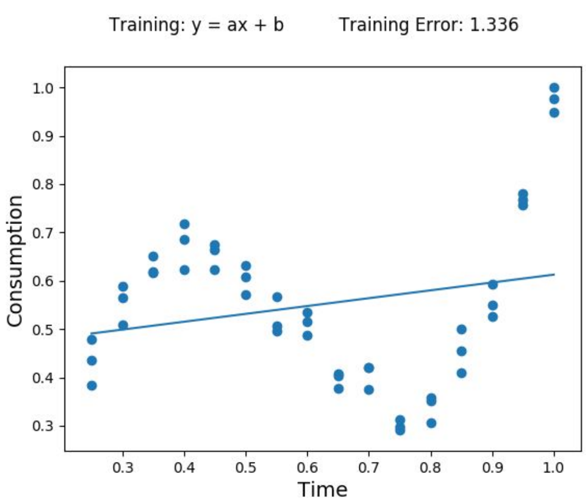
2. **Which activation function is used in the three architectures above? Why?**
   - A linear activation function is used for three architectures.
   - Because we can reduce a polynomial regression equation into a set of inputs for a neuron that utilizes a linear activation function, in order to achieve the same result as a polynomial equation would.

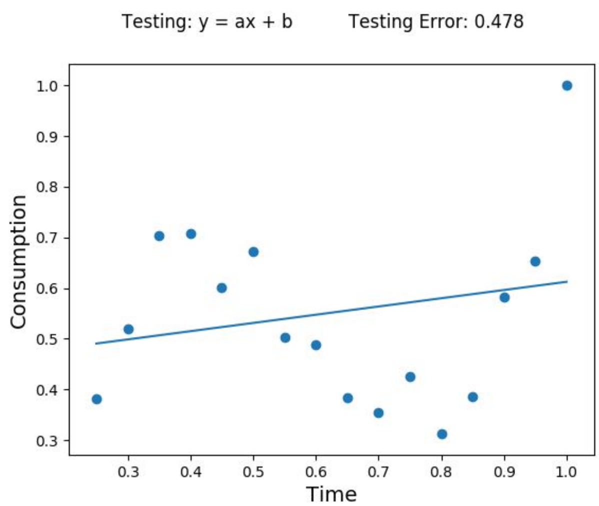
3. **Compare the training and testing total error obtained using the architectures on Figure 1:**
   - For each architecture used, training error was consistently higher than testing error, this is because there are more points in the training data.

y = ax + b (x neuron)

Training Total Error: 1.336

Testing Total Error: .478

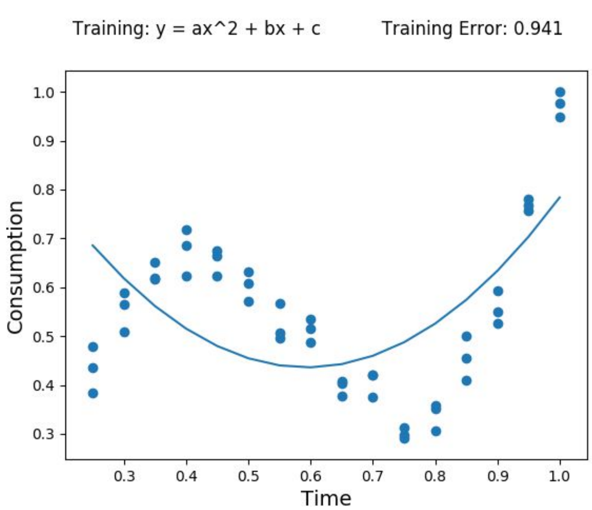


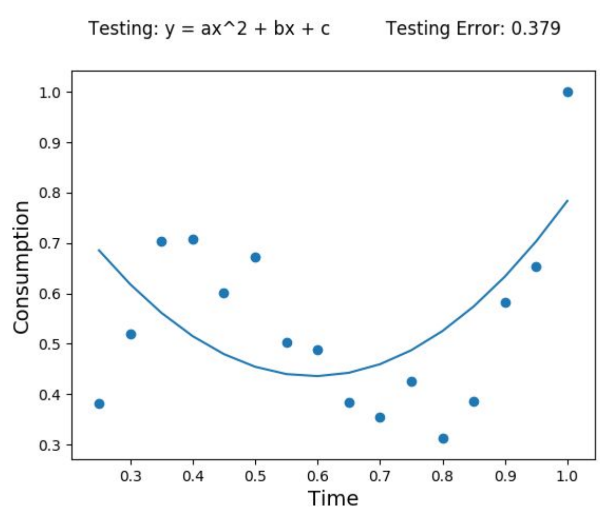


y = ax^2 +bx + c (x^2 neuron)

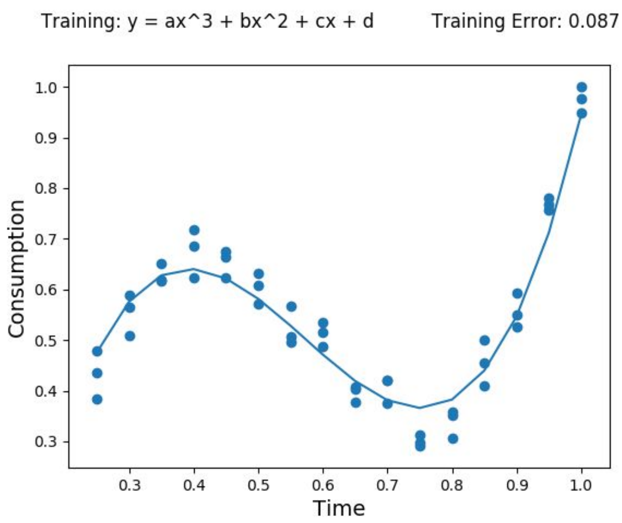
Training Total Error: .941

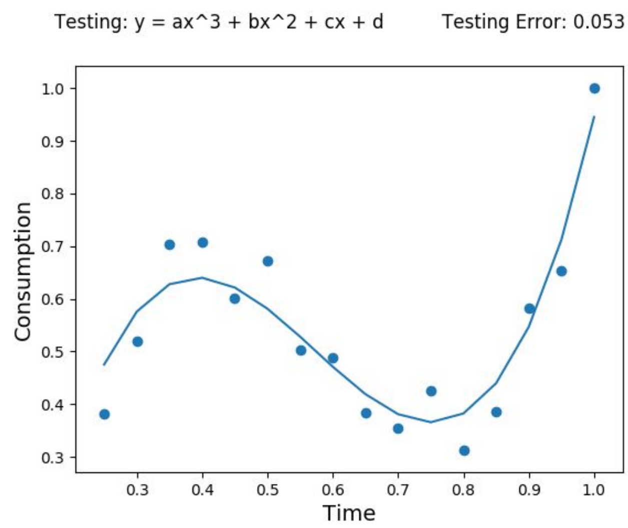
Testing Total Error: .379





y = ax^3 + bx^2 + cx + d (x^3 neuron)
Training Total Error: .087
Testing Total Error: .053





4. **Report the number of iterations, the learning rate,and data pre-processing steps you may have chosen (such as normalization of input data). Clearly explain why you selected these values and steps**

Iterations = 25000
Learning rate for linear and quadratic = .01
Learning rate for cubic = .3

Preprocessing steps.

1. Load data from file
2. Add preceding column of ones for bias input
3. Combine data for days 1, 2 and 3
4. Sort the data by hour of occurence (x) in ascending order
5. Save linear instance
6. Add column of x^2
7. Reorder columns
8. Save quadratic instance
9. Add column of x^3
10. Reorder columns
11. Save cubic instance
12. Normalization of input and testing data

Reasoning

Having a higher alpha for linear and quadratic anchored our regression lines to unfavorable positions due to being over influenced, meanwhile our 3rd degree improved with a higher alpha.

We utilized these preprocessing steps for our input in order to simplify the use of the linear activation function as a representation of polynomial regression. We represent each added degree as a new input column for our training data.