Wes Bosman

AI Project 3

11/9/16

AI Project Three

1. The inputs I used for this project were the times from 5:00 AM to 13:00PM. The output for the unit that I created were the temperatures that corresponded to the time of day that was input into the unit.
2. The activation function that I used for this problem was a linear one. I chose this activation function because when the time of day was input as an x value and the temperature for that time of day was plotted as a y value the graph appeared to be linear. Choosing a linear function gave me continuous values that I could use ranging from negative infinity to positive infinity.
3. I trained my decision unit on the first three days of input. I iterated over the input 10,000 times updating the weights iteratively. Here is some sample data from the last iteration of training.

Training Iteration 10000

|--------------------------------------------------------------------|

| Time | Learn | Net | PredictTemp | ActualTemp | Error | % Change |

|--------------------------------------------------------------------|

5:00 -0.0003 1.46 0.04 0.03 -0.0081 26.6987

6:00 -0.0011 1.84 0.15 0.12 -0.0320 26.4145

7:00 0.0027 2.19 0.26 0.33 0.0775 -23.2544

8:00 0.0002 2.72 0.42 0.42 0.0070 -1.6508

9:00 0.0011 3.14 0.54 0.58 0.0324 -5.6318

10:00 -0.0005 3.65 0.70 0.68 -0.0138 2.0254

11:00 -0.0005 4.01 0.80 0.79 -0.0130 1.6438

12:00 -0.0021 4.36 0.91 0.85 -0.0602 7.0946

13:00 0.0003 4.44 0.93 0.94 0.0085 -0.9071

5:00 0.0007 1.42 0.03 0.05 0.0194 -42.6916

6:00 -0.0020 1.82 0.15 0.09 -0.0565 62.1069

7:00 0.0030 2.13 0.24 0.32 0.0863 -26.6287

8:00 0.0009 2.67 0.40 0.43 0.0252 -5.9050

9:00 0.0007 3.13 0.54 0.56 0.0204 -3.6412

10:00 -0.0005 3.60 0.68 0.67 -0.0140 2.0944

11:00 -0.0002 3.96 0.79 0.78 -0.0054 0.6905

12:00 -0.0027 4.34 0.90 0.82 -0.0768 9.3213

13:00 0.0037 4.32 0.90 1.00 0.1048 -10.4811

5:00 -0.0028 1.60 0.08 0.00 -0.0791 inf

6:00 -0.0018 1.93 0.18 0.13 -0.0515 40.4467

7:00 0.0009 2.26 0.28 0.30 0.0263 -8.6717

8:00 0.0017 2.70 0.41 0.46 0.0486 -10.5883

9:00 0.0014 3.23 0.57 0.61 0.0414 -6.8000

10:00 -0.0006 3.77 0.73 0.72 -0.0161 2.2530

11:00 -0.0006 4.13 0.84 0.82 -0.0162 1.9621

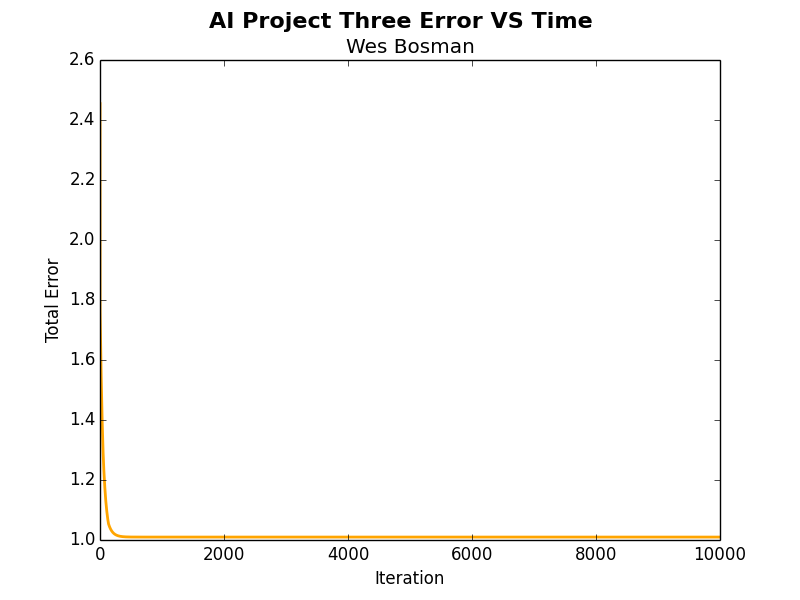
12:00 -0.0022 4.48 0.94 0.88 -0.0622 7.0593

13:00 0.0002 4.55 0.97 0.97 0.0069 -0.7102

Total Error = 1.009759

The Training error got down to just about 1. This was the result of adding the absolute values of all the errors together. The percent change is how much the prediction was off from the actual value.

Below is a graph of the error throughout each iteration. You can see that the overall error decreases as the number of iterations increases but at a point it levels off and approaches an asymptote.



1. Below is my decision unit’s prediction for the weather on the fourth day. The text output is shown below and underneath that is a graph the visually explains how accurately my decision function could predict the weather of the fourth day

Testing Neuron

|--------------------------------------------------------------------|

| Time | Predicted Temp | Actual Temp | Error | % Change

|--------------------------------------------------------------------|

5:00AM | 0.0384 | 0.0152 | -0.023 | 153.40 |

6:00AM | 0.1559 | 0.1515 | -0.0043 | 2.87 |

7:00AM | 0.2733 | 0.2939 | 0.0206 | -7.01 |

8:00AM | 0.3908 | 0.4439 | 0.0531 | -11.97 |

9:00AM | 0.5083 | 0.5888 | 0.0805 | -13.67 |

10:00AM| 0.6257 | 0.6970 | 0.0712 | -10.22 |

11:00AM| 0.7432 | 0.7939 | 0.0507 | -6.39 |

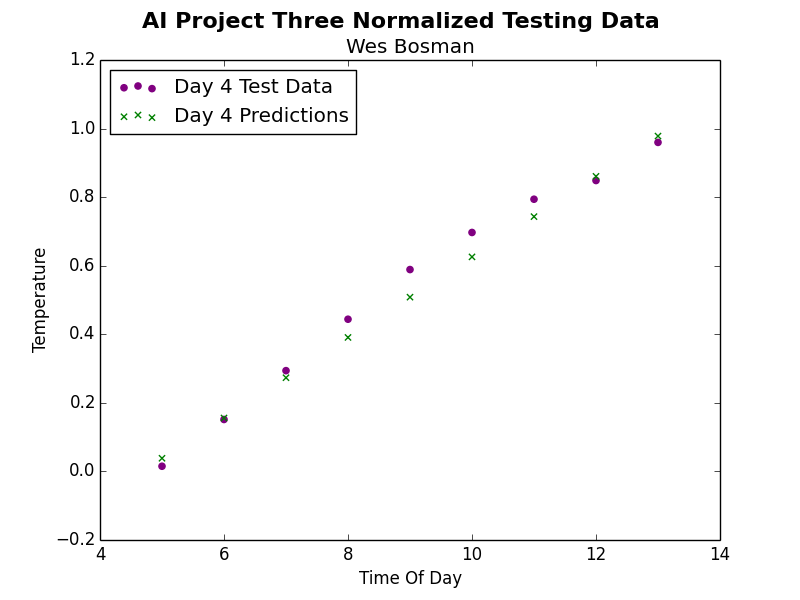
12:00PM| 0.8607 | 0.8485 | -0.0122 | 1.44 |

13:00PM| 0.9782 | 0.9597 | -0.0185 | 1.92 |

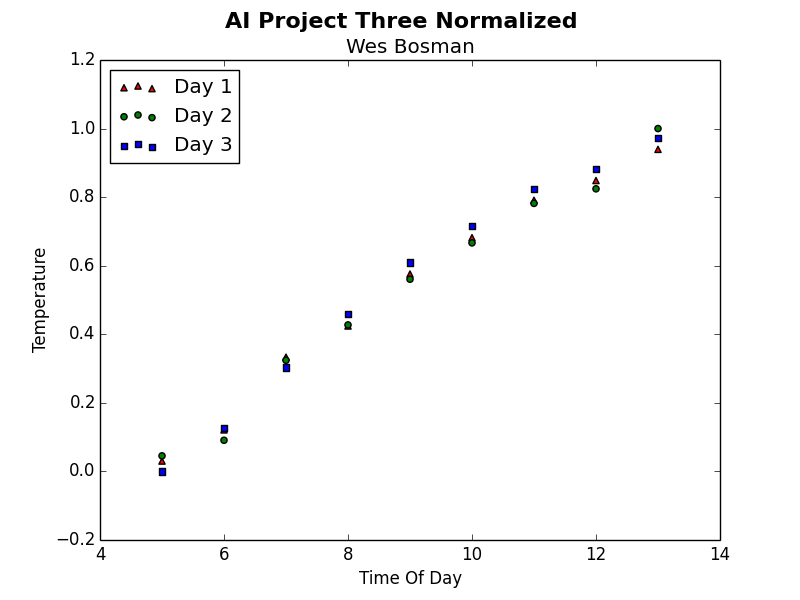
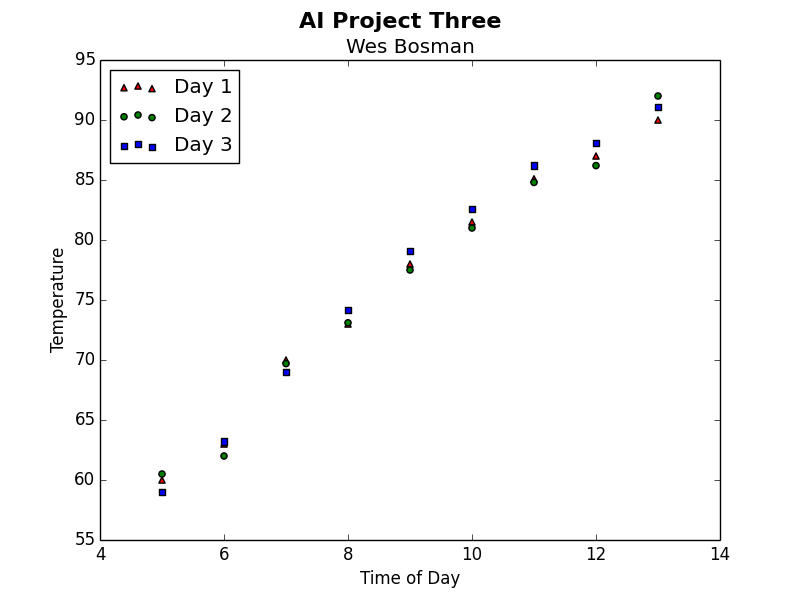
Testing Error = 0.334455

Testing Error Average Percent = 3.716172%

Below is a graph of the temperatures on the fourth day and the predicted temperatures that the unit produced for that day.



1. The number of iterations my decision unit went through was 10,000. I used a learning rate of 0.35. This seemed to give me the most accurate prediction for testing with the least amount of error. I selected this value after trying values of 0.1 and 0.5. neither of these learning constants were a best fit for the prediction of the testing data. A learning constant of 0.35 seemed as close to as many of the data points as I could get. I chose to normalize the data before I trained my neuron. Below is a graph of the data from the first three days compared to the data from the first three days that had been normalized. The figure on the top is not normalized. The figure on the bottom is normalized. The normalization seemed to make the data points more linear which led to a more accurate prediction by the unit for testing. It also made the activation function easier to write because the y values were not significantly larger than the x values.



1. As I showed in the first graph, the prediction will not get any better with more iterations the reason this is true is because the function that we are using for prediction is linear and the unit will never be 100 % accurate unless the data points that we are testing on form a straight line. I tested the unit with 5000 iterations versus 10,000 iterations and there was no significant difference in error between the two.
2. I think that a neural network could reduce the error. In a neural network you could have multiple decision units making training for predicting temperatures for different times during the day. This could make the predictions far more accurate. Most of all weather prediction would require non-linear functions to predict the pattern. Weather is volatile and constantly changing, it does not always follow a linear trend such as we have seen in this project. A neural network would be better at accurately predicting the trend for nonlinear weather data than a single neuron would be.