Neural Networks

Objective

- Create a neural network that can create a sinusoidal wave out of polynomials
- Test the accuracy of previous machine learning algorithms using the fruits data.

Objective 1: Making a sine wave

- The general equation of a polynomial is given in figure 1 where **w_i** are coefficients of a polynomial
- The equation looks similar to what perceptron algorithm use where **w_i** are the weights **x_i** are **x** at **i** degree and **a** is the output shown in figure 2.
- Activation function **z** is used to compare with the desired function **d**. Iteration stops when an minimal error or a number of iterations is achieved.

$$f(x) = w_o + w_1 x + w_2 x^2 + w_3 x^3 + \ldots \quad .$$

Fig 1: General Equation of a polynomial

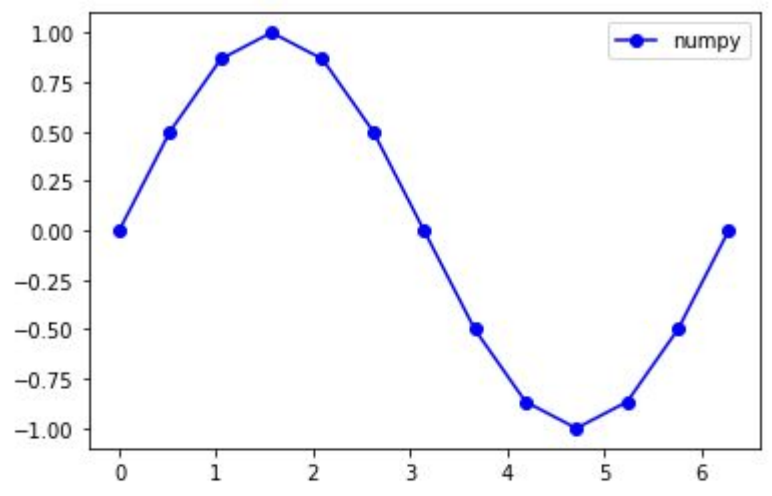
$$a = x_1 w_1 + x_2 w_2 + x_3 w_3 \ldots = \sum x_i w_i = \mathbf{x}^T \mathbf{w} \quad .$$

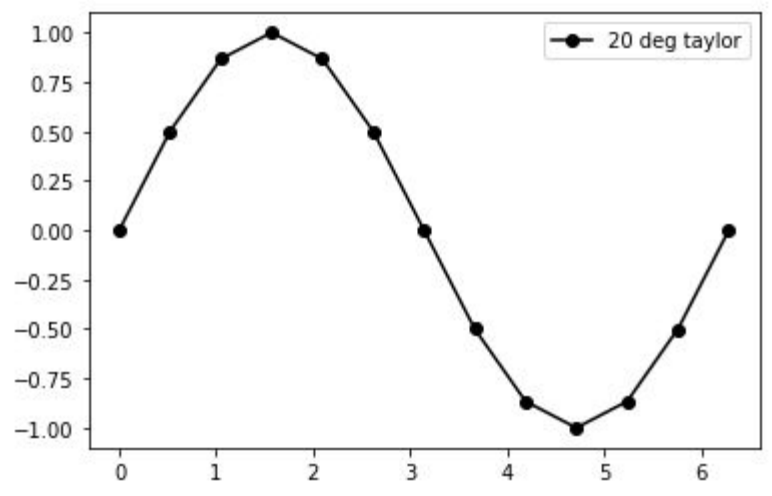
Fig 2: Output of Perceptron Algorithm

Objective 1: Making a sine wave

<u>What I tried:</u>

- Activation function: Tanh (range: [-1,1]
- Number of Layers: 1
- x = np.linspace(0,2*np.pi,N) (one period)
- degree: 20 (used 20 because that's where taylor series of sin is accurate)

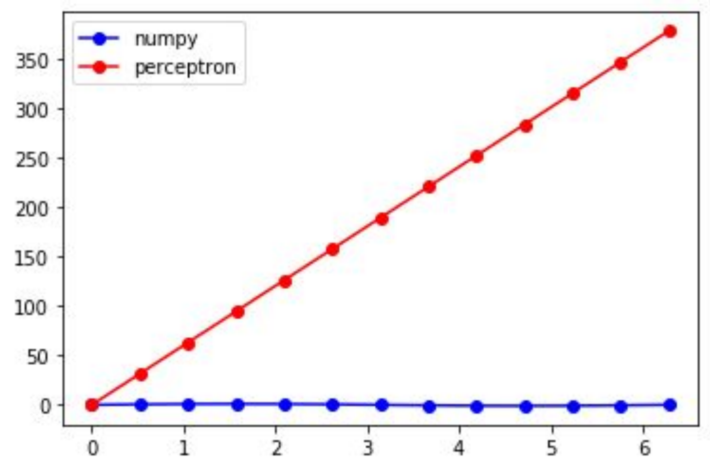


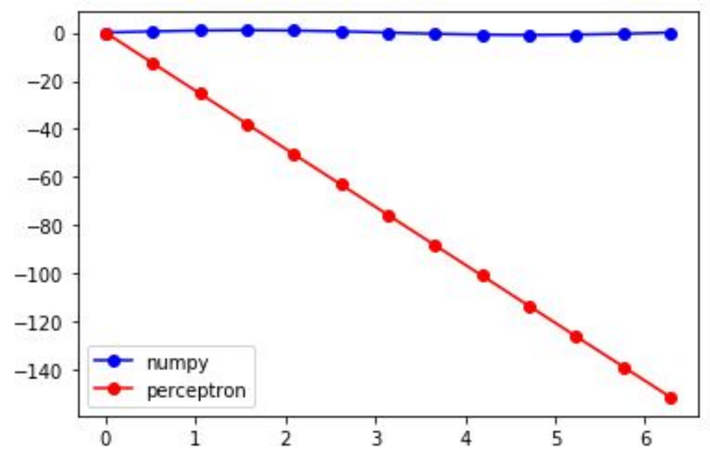


Result: It doesn't converge

- The figures show the result of the perceptron algorithm relative to the numpy equivalent.

- The error was not low enough to reach the threshold thus the result makes the output look linear.





Recommendations:

- Try to add more layers in the neural network
- Try other activation functions in conjunction with more layers
- Find a better way to compute for Δw_i

Objective 2: Testing accuracy

- In this objective I tested out two algorithms:

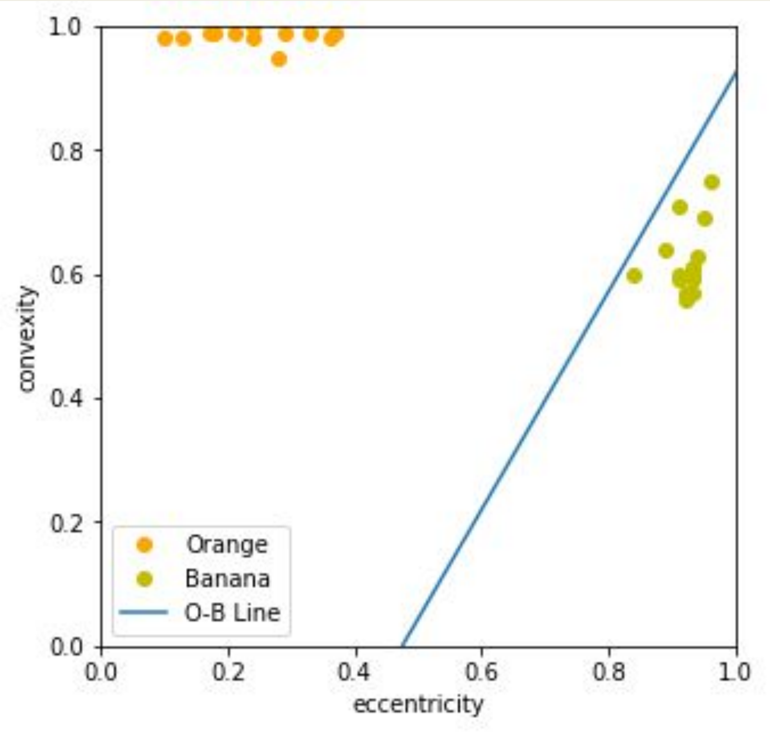
  1. Perceptron Algorithm
  2. Support Vector Machines

- Both algorithms are used to test how accurate is their identification between clusters

- Half of the samples from the fruits data were used for the training data while the other half were used for the test data. (I used 8 bananas and 6 mangoes for the training data while 7 bananas and 5 mangoes for the test data)

Overview: Perceptron

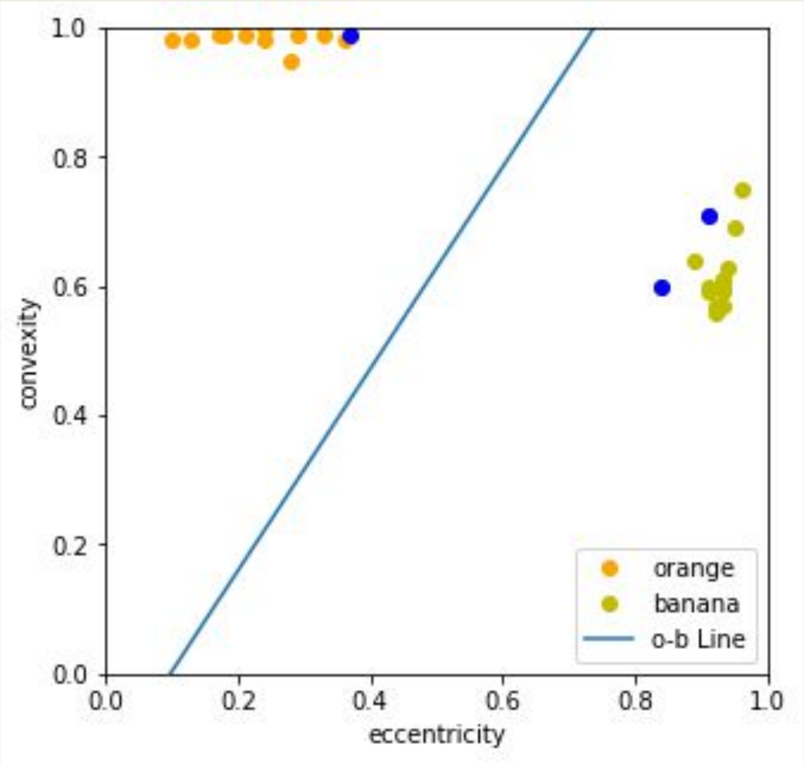
- In perceptron algorithm the weights are dependent on the samples used and activation function used.
- Iteration stops when the difference of **z** and **d** is minimal.
- Because of this the decision line is not unique and usually is biased to a side as shown in the figure (rectified activation was used)



Decision line not optimal

Overview: Support Vector Machines

- In perceptron algorithm, the decision line is not optimal to split between 2 clusters.
- It uses quadratic programming to to maximize the margin between the two clusters
- The decision line is unique since it is dependent by the support vectors (in blue) inferred by the quadratic programming.
- Although the decision line is better what I'm looking for is how accurate can they differentiate clusters.



Decision Line is optimized based on the support vectors in blue dots.

Results: Support Vector Machine

- For the support vector machines in if the gaussian clusters do collide, the data point should be at least 5-6 Standard deviations aways from both clusters which is statistically unlikely. Because of this, support vector machine has a high accuracy in terms on identifying clusters.

	Training Data		Test Data	
	Mean	StD	Mean	StD
Banana	-5.44	0.14	-5.80	0.28
Mango	-7.78	0.06	-7.76	0.08

mean and standard deviation for both training and test data

	0
0	-5.42414
1	-5.32665
2	-5.45023
3	-5.75267
4	-5.53468
5	-5.47941
6	-5.27138
7	-5.3136
8	-7.78839
9	-7.73312
10	-7.90579
11	-7.72317
12	-7.75303
13	-7.78839

training data

	0
0	-5.95075
1	-5.53468
2	-5.59681
3	-5.63218
4	-6.26933
5	-6.11347
6	-5.5055
7	-7.71012
8	-7.90888
9	-7.74926
10	-7.73931
11	-7.68404

test data

Results: Perceptron (Rectified)

- For perceptron, the iteration stops when all the banana clusters has the value **a** to be more than 0 and vice versa for mangoes.
- The training data shows that it was able to differentiate the two clusters but the test data has significant errors in identification.

	Training Data		Test Data	
	Mean	StD	Mean	StD
Banana	0.0054	0.0022	-0.0015	0.0095
Mango	-0.0139	0.0075	-0.0230	0.0059

mean and standard deviation for both training and test data

	0
0	0.004325
1	0.007725
2	0.006825
3	0.00395
4	0.002175
5	0.00325
6	0.0088
7	0.006475
8	-0.0176
9	-0.016525
10	-0.00635
11	-0.02385
12	-0.001875
13	-0.0176

training data

	0
0	-0.008925
1	0.002175
2	0.0145
3	-0.001225
4	-0.016625
5	-0.006075
6	0.00575
7	-0.0251
8	-0.012425
9	-0.02135
10	-0.028675
11	-0.0276

test data

Results: Perceptron (Rectified)

- The mean of the banana cluster of the test data is skewed to the mango region.
- The algorithm is dependent on the size of the data and activation function used.
- Since the decision line is not optimized, it is more likely to have wrong identification of clusters compared to SVM.

	Training Data		Test Data	
	Mean	StD	Mean	StD
Banana	0.0054	0.0022	-0.0015	0.0095
Mango	-0.0139	0.0075	-0.0230	0.0059

mean and standard deviation for both training and test data

	0
0	0.004325
1	0.007725
2	0.006825
3	0.00395
4	0.002175
5	0.00325
6	0.0088
7	0.006475
8	-0.0176
9	-0.016525
10	-0.00635
11	-0.02385
12	-0.001875
13	-0.0176

training data

	0
0	-0.008925
1	0.002175
2	0.0145
3	-0.001225
4	-0.016625
5	-0.006075
6	0.00575
7	-0.0251
8	-0.012425
9	-0.02135
10	-0.028675
11	-0.0276

test data