

CSDS 391 Programming Assignment 2

Writeup

Ningjia Huang, Tianxi Zhao

Due On: December 4, 2020

Problem 1:

Problem 2:

(a)

The mean squared error is calculated using the following equation:

$$MSE = \frac{1}{n} \sum_n (Y_i - \hat{Y}_i)^2$$

, where Y_i is the actual category of the i th item and \hat{Y}_i is the predicted category by using our neural network.

The following codes compute the **mean squared error** for iris data. The parameter "data_vector" are the attributes we would like to take into account, it should be a dataframe of attributes. w_0, w_1, w_2 define the weights of neural network. The parameter "pattern_classes" is a dataframe of the category corresponding to the data_vectors. The **mean_square_error** makes use of the logistic non-linearity function in Problem 1.

```
1 # data vectors in dataframe, pattern classes in list
2 def mean_square_error(data_vectors, w0, w1, w2,
3   pattern_classes):
4     n = data_vectors.shape[0]
5     data_vectors_list = data_vectors.values.tolist()
6     pattern_classes_list = pattern_classes.tolist()
7     temp_mse = 0
```

```

7     for i in range(n):
8         temp_mse = temp_mse + np.square(pattern_classes_list[
          i] - one_layer_network(w0, w1, w2, data_vectors_list[i
          ] [0], data_vectors_list[i] [1]))
9     mse = temp_mse/n
10    return mse

```

Listing 1: Mean Squared Error Calculation

(b)

To plot the linear boundary, we find the x-intercept and y-intercept by using the following code:

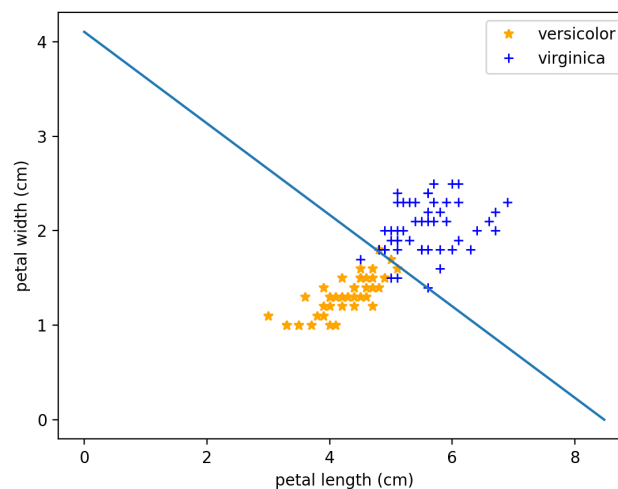
```

1 plot([0, -w0/w1], [-w0/w2, 0])

```

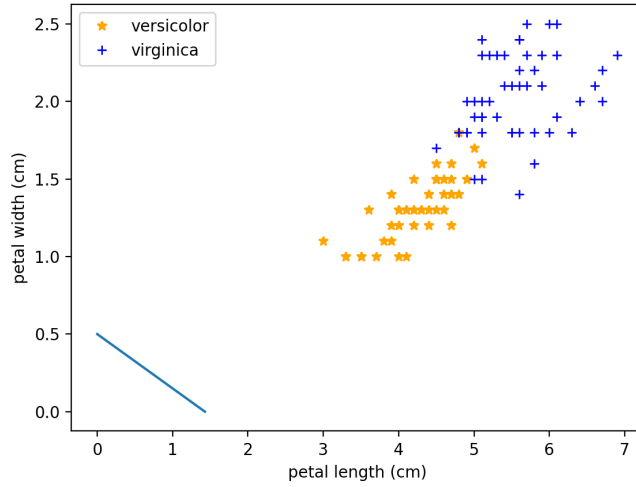
2.b.1

The first set of weights we've chosen is $[-3.9, 0.46, 0.95]$. The mean squared error we concluded is 0.05. The following figure shows the linear boundary with our chosen set of weights. As we can tell from the graph, the linear boundary generally separates the two categories of iris.



2.b.2

The second set of weights we've chosen is $[-1, 0.7, 2]$. The mean squared error we concluded is 0.5. The following figure shows the linear boundary with our chosen set of weights. As we can tell from the graph, the linear boundary does not separate these two categories at all.



(c)

To compute the gradient of the objective function, we need to take the derivative of the objective function. Assume \mathbf{x}_i is the i th row with attribute petal length and petal width such that $x_i = [1, x_{i1}, x_{i2}]$ (1 is placed here because w_0 requires 1 to be its coefficient). \mathbf{w} is the set of weights for the attributes. The logistic function we get is:

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{w \cdot x}}$$

Therefore, the mean squared function becomes:

$$MSE = \frac{1}{N} \sum_{i=1}^n (\sigma(w \cdot x_i) - y_i)^2$$

By taking the derivative of this objective function with respect to w_0 , we have:

$$\begin{aligned}\frac{\partial MSE}{\partial w_0} &= \frac{2}{N} \sum_{i=1}^N (\sigma(w \cdot x_i) - y_i) \frac{\partial \sigma}{\partial w_0} \\ \frac{\partial MSE}{\partial w_0} &= \frac{2}{N} \sum_{i=1}^N (\sigma(w \cdot x_i) - y_i) (1 - \sigma(w \cdot x_i)) (\sigma(w \cdot x_i)) \frac{\partial(w \cdot x)}{\partial w_0} \\ \frac{\partial MSE}{\partial w_0} &= \frac{2}{N} \sum_{i=1}^N (\sigma(w \cdot x_i) - y_i) (1 - \sigma(w \cdot x_i)) (\sigma(w \cdot x_i))\end{aligned}$$

Similarly, by only changing the last derivative $\frac{\partial(w \cdot x)}{\partial w_0}$ to $\frac{\partial(w \cdot x)}{\partial w_1}$ and $\frac{\partial(w \cdot x)}{\partial w_2}$, we can compute the derivative of the objective function with respect to w_1 and w_2 :

$$\begin{aligned}\frac{\partial MSE}{\partial w_1} &= \frac{2}{N} \sum_{i=1}^N (\sigma(w \cdot x_i) - y_i) (1 - \sigma(w \cdot x_i)) (\sigma(w \cdot x_i)) x_{i1} \\ \frac{\partial MSE}{\partial w_2} &= \frac{2}{N} \sum_{i=1}^N (\sigma(w \cdot x_i) - y_i) (1 - \sigma(w \cdot x_i)) (\sigma(w \cdot x_i)) x_{i2}\end{aligned}$$

(d)

vector form:

scalar form: