

# Homework 4

Jian Huang

2023/10/07

## Task 1: Softmax cost function on linear separable 2-class data

Softmax cost function:

$$g(w) = \frac{1}{P} * \sum_{p=1}^P \log(1 + e^{(-y_p * \vec{x}_p^T * w)})$$

1. Report both the accuracy and the misclassifications of my model.

The confusion matrix of my prediction using trained softmax cost function:

	class1 (label: 1)	class2 (label: -1)	Total of actual data points
class1 (label: 1)	6	0	6
class2 (label: -1)	0	5	5
Total of prediction	6	5	11

Based on the above matrix, the total accuracy is then calculated as follows:

$$Misclassifications = 0 + 0 = 0$$

$$Accuracy = (6 + 5)/11 = 100\%$$

2. A figure that shows cost history over iterations.

Using the cost history obtained during training, below is the plot of cost vs. iterations:

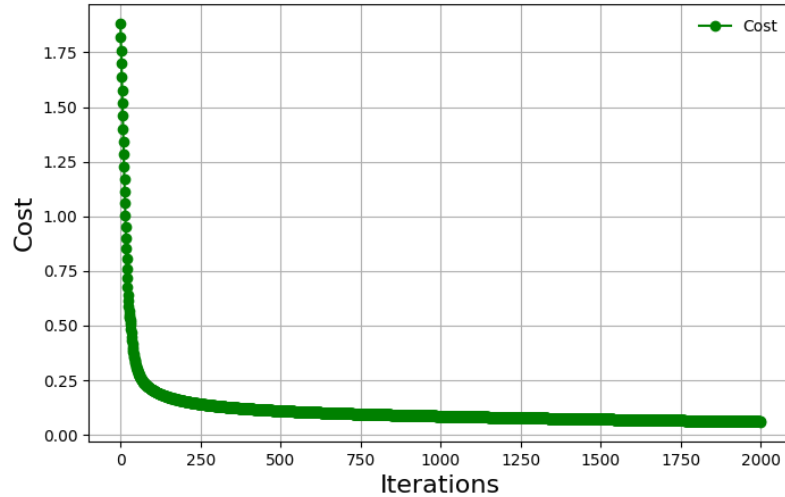


Figure 1: Cost history during training

3. A figure that shows the original data and the fitted tanh curve.

The final weight defines the separation boundary as:

$$x = 1.03995$$

Using the final weights, here is the fitting result.

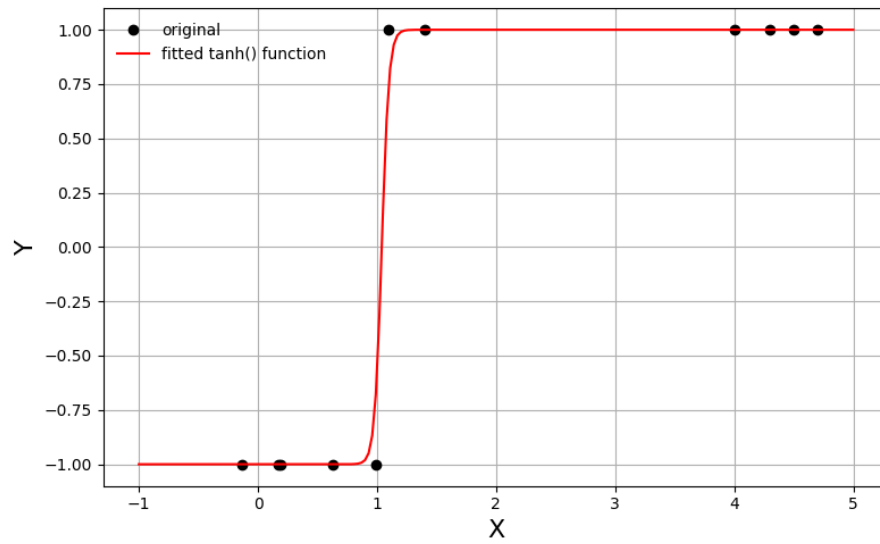


Figure 2: The fitted  $\tanh(x)$  curve Vs. the original data

## Task 2: Compare the efficacy of two-class cost functions

1. Initial values for weight, alpha and max iterations.

Cost Func.	weights	alpha	max iterations
Softmax	random between (0, 1)	0.1	500
Perceptron	all 1.0	0.1	500

### Reasonings for my choices:

- The Softmax cost function has gradient everywhere. Therefore, it does not matter where to start. That's why I chose to use random numbers between (0, 1). However, the Perceptron cost function has an undifferentiable point at all zeros of weight, and also this point results in the cost function to reach 0. Therefore, I used all ones for all dimensions of the weights.
- The choice of alpha depends on the iterations and also the accuracy of prediction. I found 500 max iterations works great with alpha=0.1, based on the cost history plot (shown below).

2. Report both the accuracies and the misclassifications obtained from the two cost functions separately. Here I summarized the confusion matrices for both softmax and perceptron training:

	class1 (label: 1)	class2 (label: -1)	Total of actual data points
class1 (label: 1)	444	14	458
class2 (label: -1)	12	229	241
Total of prediction	456	243	699

Table 1: Confusion matrix from **Softmax** cost function training

	class1 (label: 1)	class2 (label: -1)	Total of actual data points
class1 (label: 1)	442	16	458
class2 (label: -1)	10	231	241
Total of prediction	452	247	699

Table 2: Confusion matrix from **Perceptron** cost function training

Based on the above matrices, the accuracy values of both cost functions are then calculated as follows:

Cost Func.	Misclassifications	Overall Accuracy
Softmax	26	96.3%
Perceptron	26	96.3%

Table 3: Accuracy of training results of the two cost functions

3. A figure that compares the cost over iterations between two cost functions.

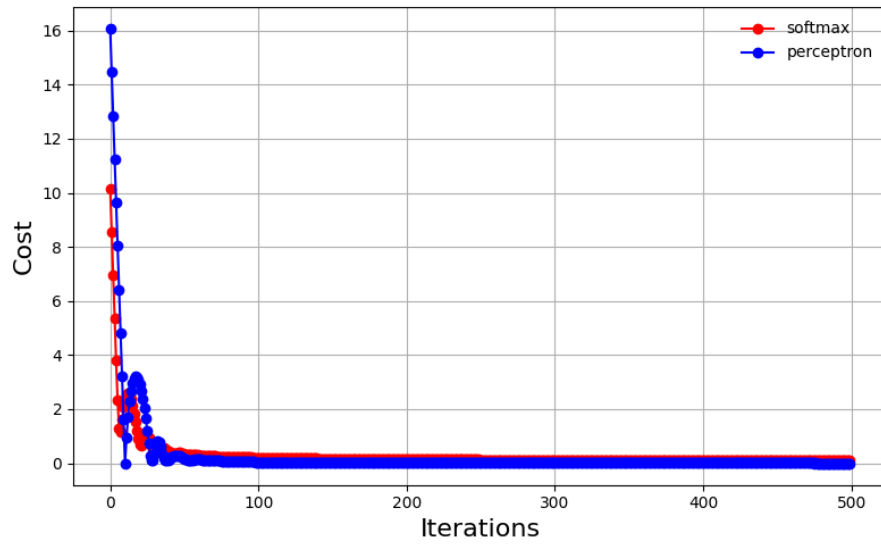


Figure 3: Cost comparison between two cost functions

4. A figure that compares the accuracy over iterations between two cost functions.

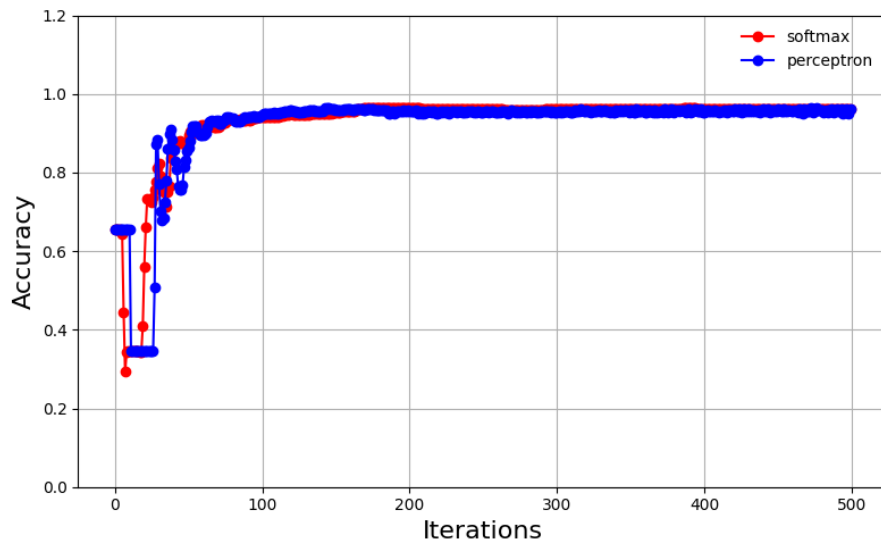


Figure 4: Accuracy comparison between two cost functions