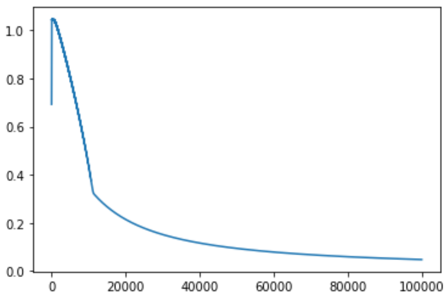
Binary classification using 2-layerd network #3-1

컴퓨터소프트웨어학과 2017029470 김종원

1. Testing result

- $w = 0\,,\ b = 0\,,\ \alpha = 0.001$: *used sine function to generate dataset.*

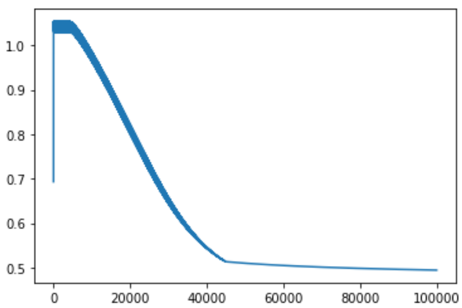
	m = 10, n = 1000, k = 5000	m = 100, n = 1000, k = 5000	m = 10000, n = 1000, k = 5000
Accuracy (m train samples)	**80.0**	**83.0**	**49.49**
Accuracy (n test samples)	**47.599999999999994**	**86.0**	**50.0**
	m = 10000, n = 1000, k = 10	**m = 10000, n = 1000, k = 100**	**m = 1000, n = 1000, k = 5000**
Accuracy (m train samples)	**49.62**	**49.71**	**99.0**
Accuracy (n test samples)	**50.1**	**48.699999999999996**	**97.7**



The graph on the left shows the value of cost function according to the k value when a 2-layer neural network is trained with a training set with m = 10000, and with learning rate 0.001. until k = 15000, the cost function value decreased sharply, but the amount of decrease thereafter became smaller. However, when all iterations were completed, the cost value was very small, about 0.04, and the testing accuracy was 100%.

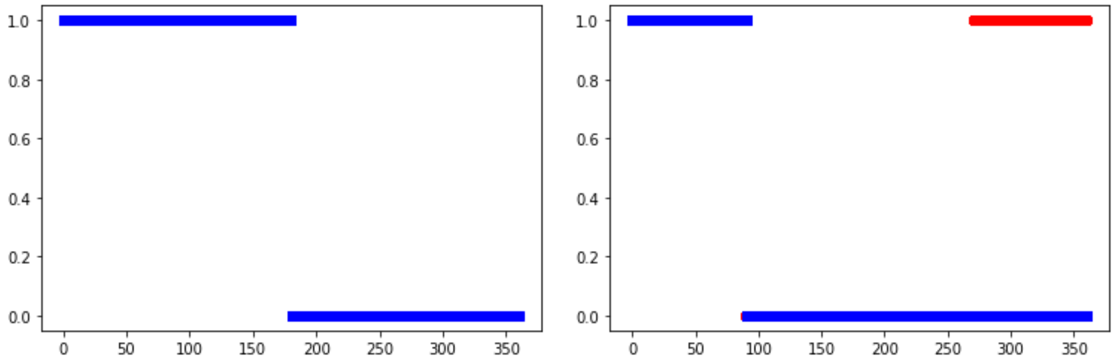
- $w = 0\,,\ b = 0\,,\ \alpha = 0.001$: *used cosine function to generate dataset.*

	m = 10, n = 1000, k = 5000	m = 100, n = 1000, k = 5000	m = 10000, n = 1000, k = 5000
Accuracy (m train samples)	**70.0**	**56.1**	**49.81**
Accuracy (n test samples)	**48.6**	**49.4**	**48.9**
	m = 10000, n = 1000, k = 10	**m = 10000, n = 1000, k = 100**	**m = 1000, n = 1000, k = 5000**
Accuracy (m train samples)	**49.78**	**49.58**	**50.1**
Accuracy (n test samples)	**49.6**	**51.6**	**52.7**



The graph on the left shows the value of the cost function according to the k value when a 2-layer neural network is trained with a training set with m = 10000 and with learning rate 0.001. until k = 40000, the cost value was drastically reduced by the gradient decent method, but after that, as the gradient of the graph became gentle, the amount of decreases became smaller. When comparing the experimental results, when the dataset was created using the cosine function, the accuracy of the result was lower than when the dataset was created using the sine function.

2. empirical discussion



The two graphs above show the results of classifying the test dataset using a 2-layer neural network trained as a training dataset (when m = 10,000, n = 1,000, k = 100,000 and learning rate = 0.001). The red line represents the class label of the training dataset, and the blue line represents the class label of the test dataset. When the training data was created using the sine function, the accuracy was 100% and the cost value was only 0.04, but when the cosine function was used, the accuracy was about 75% and the cost value was 0.4. in the case of the cosine function, even if the value of k was increased to 300,000, the accuracy did not increase any more, and the cost value did not decrease. **The reason because there was only one perceptron in one layer, it was insufficient to learn the cosine function that has more complex class label than the sine function.**

- **Empirically determined (best) hyper parameter,** $\boldsymbol{\alpha = 0.001}$

  Although graphs of the results of other experiments were not attached to the report, when the learning rate was 0.01, it changed so rapidly that it was not possible to find appropriate values for w and b. even when the learning rate was 0.0001, appropriate w and b could not be found, because the change occurred too little, and the k value had to be very large.

- **Estimated unknown function parameters,** $\boldsymbol{w_1 = -1.12044874}$, $\boldsymbol{b_1 = 00.61686392}$, $\boldsymbol{w_2 = 4.84586487}$, $\boldsymbol{b_2 = -0.6283422}$

  Since the accuracy of the trained neural network was no longer improved at about 75%, I wrote the w and b values when k = 300,000.