

**A small experiment:**

A simple network with one input(x), one weight w and one bias b. The input will always be 1. The target value (teacher value) will be 0. The transfer function of the node is sigmoid and MSE cost function. , t = 0 in this case.



The role of the experiment is to implement simple gradient decent algorithm to find the weight(w) and bias(b).

(1a) when the initial weight: w = 0.6, b = 0.9, z=wx+b = 1.5. 0.81. With learning rate = 0.15. Training Epochs = 300. Plot the figure of cost v.s. epochs.

(1b) Try another initial weight: w= 2, b=2, z=4, . With learning rate = 0.15. Training Epochs = 300. Plot the figure of cost v.s. epochs.

Before implement the above experiment, please **take a guess** that which case (1a or 1b) will converge faster ?

2 Now let try another cost function: Cross Entropy, where n = number of training data. x = a training item. y= activation for training data x, t = teacher value for training data x.

(2a): when the initial weight: w = 0.6, b = 0.9, z=wx+b = 1.5. 0.81. Since the cost function is changed, please try the learning rate = 0.005. Plot the figure of cost v.s. epochs.

(2b) Try another initial weight: w= 2, b=2, z=4, . With learning rate = 0.005. Training Epochs = 300. Plot the figure of cost v.s. epochs.

**IRIS dataset exploration and visualization:**

This is perhaps the best known database to be found in the pattern recognition literature. Fisher’s paper is a classic in the field and is referenced frequently to this day. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

The role of this part is to practice data exploration and visualization using pandas, matplotlib, and seaborn.

1. First, read IRIS dataset from this URL: <https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data>, using pandas.[read\_csv](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html) (please specify column name as sepal-length, sepal-width, petal-length, petal-width, and class)
   1. What is the dimension of this dataset? ([shape](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.shape.html))
   2. Please print the 41th to 45th data
   3. Please print the last 5 data ([tail](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.tail.html))
   4. Please print the first 10 data ([head](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.head.html))
   5. Please print the description of the dataset (mean, std, min, 25%, 50%, 75% and max for each column, [describe](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.describe.html))
   6. How many data are there for each class? ([groupby](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.groupby.html))
2. Second, visualize IRIS dataset
   1. Plot the scatter graph of sepal-length versus sepal-width ([plot](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.html))
   2. Plot the box plot by class ([boxplot](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.boxplot.html))
   3. Plot the box plot (subplots) for each column ([plot](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.html))
   4. Plot the histogram for each column ([hist](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.hist.html))
   5. Plot the scatter matrix for each column ([scatter\_matrix](https://pandas.pydata.org/pandas-docs/stable/visualization.html" \l "scatter-matrix-plot))