HW 2 Single -layer Linear Neutral Networks

Report

5.

(a) Accuracy of prediction

All accuracy results are calculated in the main.py. The final accuracy results are attached below (only show 10 iterations)

1. perceptron binary classifier

Iteration	1	2	3	4	5	6	7	8	9	10
Misclassify	2	2	2	1	0	0	0	0	0	0
Accuracy	98.7%	98.7%	98.7%	99.3%	100%	100%	100%	100%	100%	100%

2. adaline binary classifier

Iteration	1	2	3	4	5	6	7	8	9	10
Misclassify	13	0	0	0	0	0	0	0	0	0
Accuracy	91.3%	100%	100%	100%	100%	100%	100%	100%	100%	100%

3. sgd binary classifier

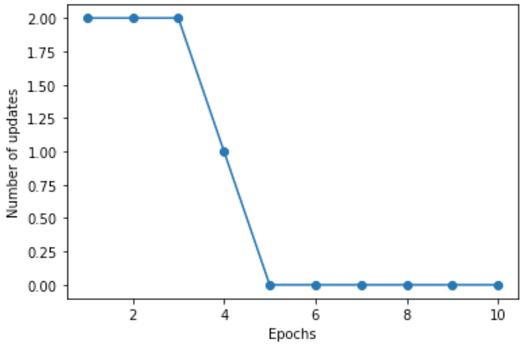
Iteration	1	2	3	4	5	6	7	8	9	10
Misclassify	4	0	0	0	0	0	0	0	0	0
Accuracy	97.3%	100%	100%	100%	100%	100%	100%	100%	100%	100%

All there three classifiers can predict 100% result after several iterations. Perceptron binary classifier need four iterations and the other two classifiers only need 1 iteration.

(b) Error or cost analysis

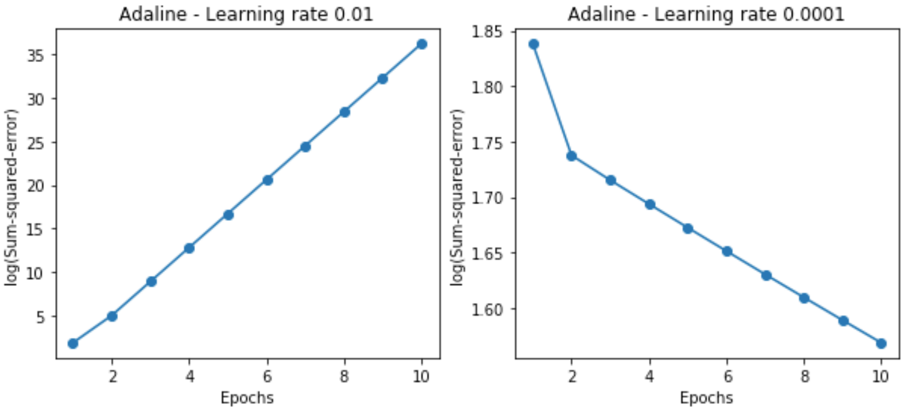
1. Iris

- perceptron classifier



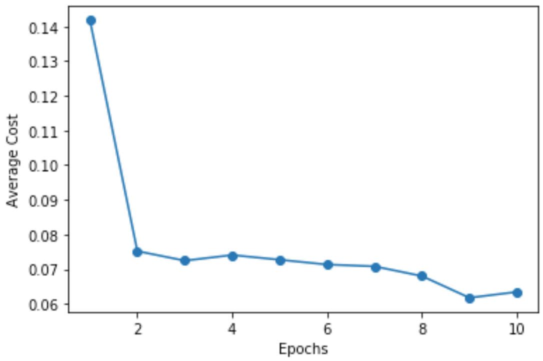
Updates reduce to 0 at 5 epochs, which mean the trained model can perfectly provide accurate predictions after 5 epochs.

- adaline classifier



Based on the error curves at learning rate of 0.01 and 0.0001, the error keeps increasing at the rate of 0.01, which means the step is too large. As for the rate of 0.0001, the error gradually reduces, and it will converge after certain epochs.

- sgd classifier

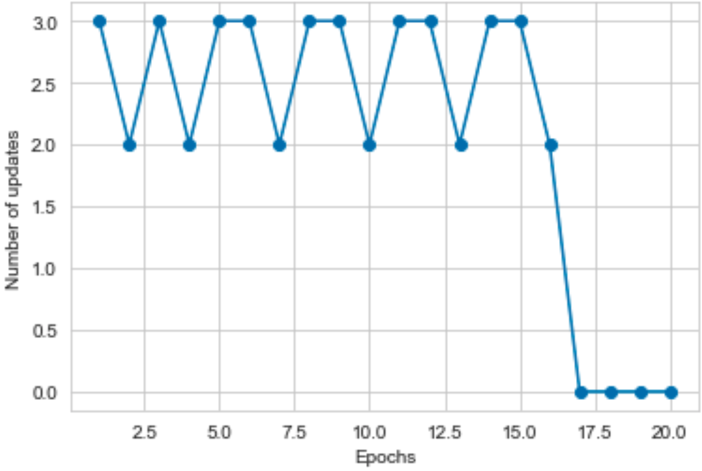


As shown in figure, the error reduces to a low level (<0.08) right after the first epoch.

2. Wine Data Set

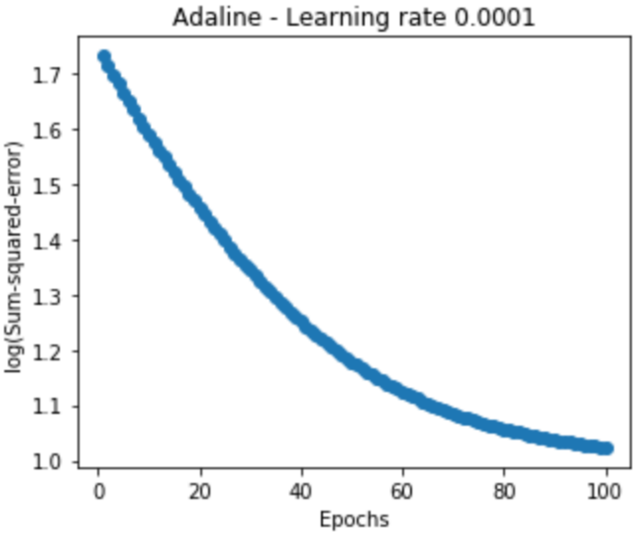
- perceptron classifier:

  it takes 17 epochs to reach the optimization (no error).



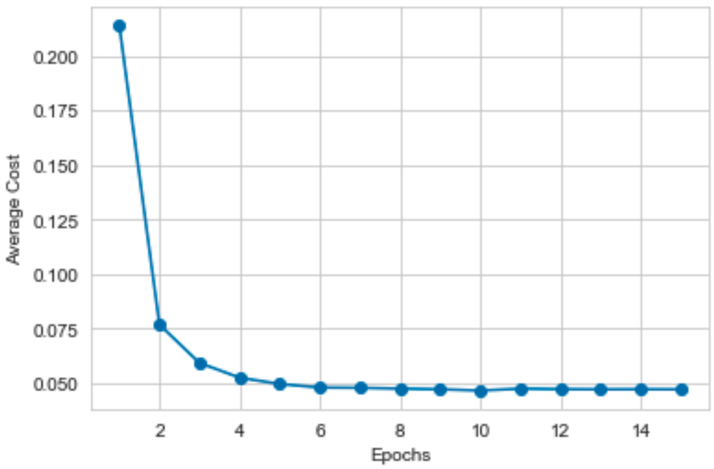
- adaline classifier

  As indicated in the below figure, the error keeps decreasing and can converge nearly at 100 epochs.



- sgd classifier

  sgd classifier only takes 4 epochs to reach a very error (about 0.05).


0.200
0.175
0.150
0.125
0.100
0.075
0.050
Average Cost
2
4
6
8
10
12
14
Epochs

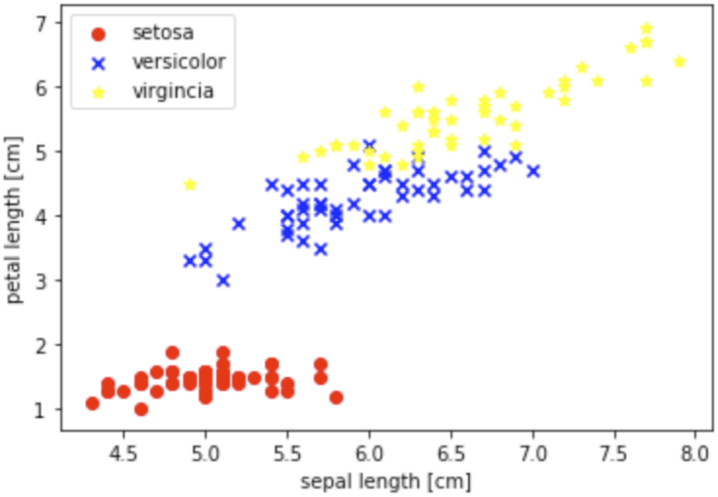
(c) Datasets

1. Iris

   Number of Instances: 150

   Number of Attributes: 4 features + label

   Treat "Iris-setosa" as 1 and the other two as -1. It can be found that the type of setosa can be successfully separated from the other groups.

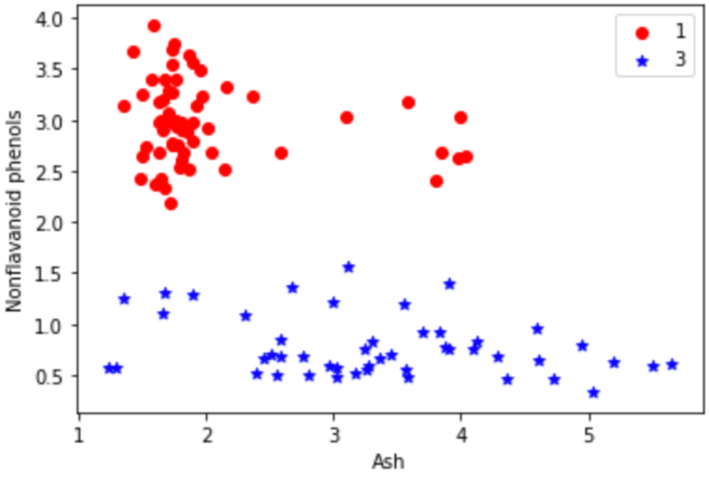


2. Wine Data Set

   Number of Instances: 178

   Number of Attributes: 13 features + label (1,2,3) [first column is the label]

   In the testing, treat type 1 as 1 and 3 as -1. In addition, two features are selected as they are proved linearly reparably, as shown below.



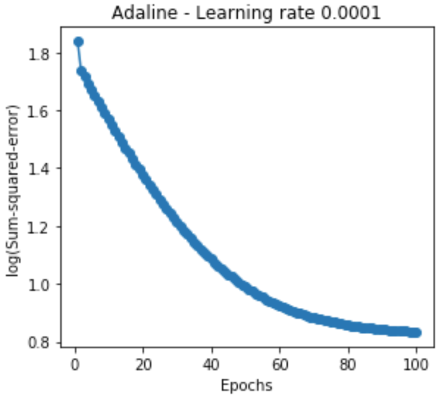
(d) Classifiers’ behavior analysis

- Classifiers convergence

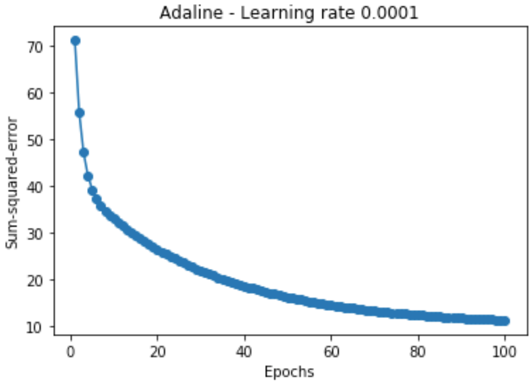
  According to the figure in (b), the fast-convergent algorithm is the sgd, followed by adaline and perceptron classifier. (convergence rate: sgd > adaline > perceptron)

- The effect of feature scaling:

  The optimizer goes through fewer steps to find a good or optimal solution.



(1) no feature scaling



(2) with feature scaling

When I increase the iteration to 100 times, after featuring, the convergence rate will be much fast, as shown in (2).

(e) Other aspects

When selecting a classifier for two classes, the number of training data is also an important consideration. For example, mini-batch learning show benefits for smaller subsets of data. However, as for large-scale data, stochastic gradient descent is a better alternative.

- Learning rate

Similar with the convergence rate as discussed above, the learning rate for these three classifiers are (sgd > adaline > perceptron)

- Running time

Since the data selected in the analysis is very small, the running time for each classifier doesn’t vary too much.

7.

As for the Iris data set, I create a multiclass classifier based on training three sgd classifiers. All features are selected to train the classifier. In addition, another function, terms as ‘decision_function’ is defined to achieve the multiclass classifier purpose.

As for the wine data set, it is hard to classify the types of wine if all features are considered. To this end, I pre-plot them and choose two features (column: 1 and 7 in the original data set)

All cost curves are plotted in the main.py as well as the prediction result with the trained classifiers. Accuracy and predicted results are both printed. All figures and results will be shown after you execute the main.py.