Final Projects

CSA01 Neural Networks I

Date: 2017/06/12

Team:

m5211102 Dai Funayama

m5211129 Sho Ikeda

m5211132 Shu takenoshita

We develop 2 programs to compare 3 models.
First model is Single-layer Perceptron Model.
Second model is Multi-layer Perceptron Model.
Third model is Linear Support Vector Machines Model.

We develop programs using python with scikit-learn.scikit-learn's official web site is this URL(http://scikit-learn.org/stable/index.html).

We use 3 classes on scikit-learn to realize 3 models.
Single-layer Perceptron Model is defined by sklearn.linear_model.Perceptron.
Multi-layer Perceptron Model is defined by sklearn.neural_network.MLPClassifier.
Linear Support Vector Machines Model is defined by sklearn.svm.LinearSVC.
Then, we use Perceptron and LinearSVC with default parameter. In addition, we use MLPClassifier with sgd parameter (basic stochastic gradient descent) and default parameter (Adam).

Identification-rate.py

This program is to calculate Identification rate using score method. score method returns the mean accuracy on the given test data and labels.
Training data to learn and Test data to confirm identification rate is created by train_test_split method with test-size="0.3" .
"rate-result.txt" is execution result file. "X_train" means training data. "y_train" means training data's class label."X_test" means test data. "y_test" means test data's class label.
After the running program, we obtain the identification rate by each models.

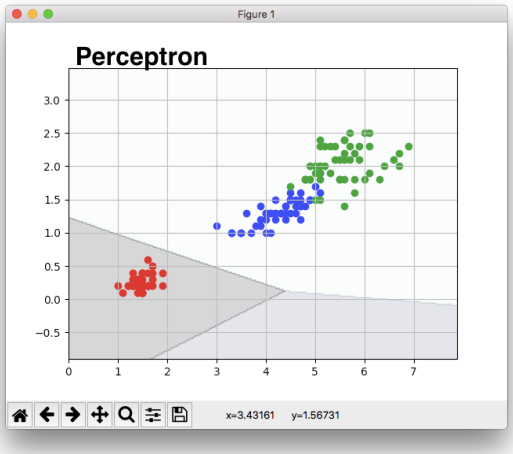
(Perceptron Identification rate:', 0.59999999999999998)
('MLPC Identification rate:', 0.93333333333333335)

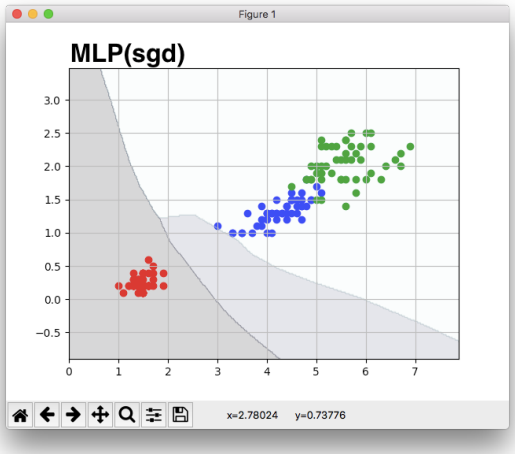
('MLPC(sgd) Identification rate:', 0.82222222222222219)
('LinearSVC Identification rate:', 0.93333333333333335)

From this result, MLPC with adam and LSVC can identify with high probability. However, perceptron identify hardly. When we compare MLPC with adam and MLPC with basic stochastic gradient descent, we can know that adam method is better than basic stochastic gradient descent.

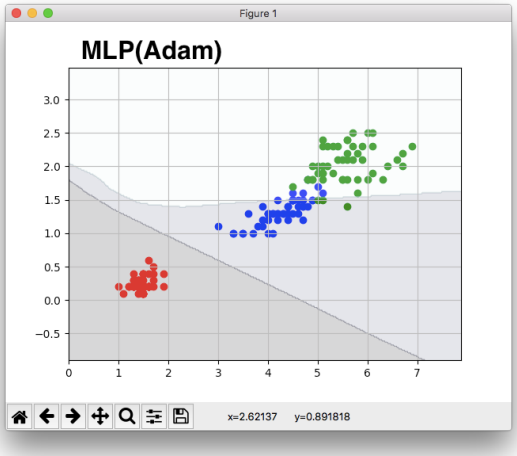
plot.py

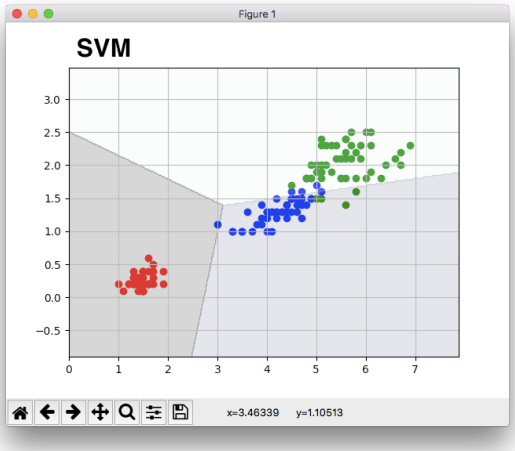
This program is to plot the graph. This graph expain how classification is done.
All iris data is used for training data. The graphs line means each class's border
From these figures, perceptron cannot identify three classes and MLP(sgd) also can hardly identify three classes. However, MLP(Adam) and SVM can classify three classes. From this result, MLP is influenced by the type stochastic gradient descent and SVM is excellent classifier





.





-----appendix-------

“rate-result.txt”

X_train

```
[[ 5.   2.   3.5  1. ]
 [ 6.5  3.   5.5  1.8]
 [ 6.7  3.3  5.7  2.5]
 [ 6.   2.2  5.   1.5]
 [ 6.7  2.5  5.8  1.8]
 [ 5.6  2.5  3.9  1.1]
 [ 7.7  3.   6.1  2.3]
 [ 6.3  3.3  4.7  1.6]
 [ 5.5  2.4  3.8  1.1]
 [ 6.3  2.7  4.9  1.8]
 [ 6.3  2.8  5.1  1.5]
 [ 4.9  2.5  4.5  1.7]
 [ 6.3  2.5  5.   1.9]
 [ 7.   3.2  4.7  1.4]
 [ 6.5  3.   5.2  2. ]
 [ 6.   3.4  4.5  1.6]
 [ 4.8  3.1  1.6  0.2]
 [ 5.8  2.7  5.1  1.9]
 [ 5.6  2.7  4.2  1.3]
 [ 5.6  2.9  3.6  1.3]
 [ 5.5  2.5  4.   1.3]
 [ 6.1  3.   4.6  1.4]
 [ 7.2  3.2  6.   1.8]
 [ 5.3  3.7  1.5  0.2]
 [ 4.3  3.   1.1  0.1]
 [ 6.4  2.7  5.3  1.9]
 [ 5.7  3.   4.2  1.2]
 [ 5.4  3.4  1.7  0.2]
 [ 5.7  4.4  1.5  0.4]
 [ 6.9  3.1  4.9  1.5]
 [ 4.6  3.1  1.5  0.2]
 [ 5.9  3.   5.1  1.8]
 [ 5.1  2.5  3.   1.1]
 [ 4.6  3.4  1.4  0.3]
 [ 6.2  2.2  4.5  1.5]
 [ 7.2  3.6  6.1  2.5]
 [ 5.7  2.9  4.2  1.3]
 [ 4.8  3.   1.4  0.1]
 [ 7.1  3.   5.9  2.1]
 [ 6.9  3.2  5.7  2.3]
 [ 6.5  3.   5.8  2.2]
 [ 6.4  2.8  5.6  2.1]
```

[5.1 3.8 1.6 0.2]
[4.8 3.4 1.6 0.2]
[6.5 3.2 5.1 2.]
[6.7 3.3 5.7 2.1]
[4.5 2.3 1.3 0.3]
[6.2 3.4 5.4 2.3]
[4.9 3.   1.4 0.2]
[5.7 2.5 5.   2.]
[6.9 3.1 5.4 2.1]
[4.4 3.2 1.3 0.2]
[5.   3.6 1.4 0.2]
[7.2 3.   5.8 1.6]
[5.1 3.5 1.4 0.3]
[4.4 3.   1.3 0.2]
[5.4 3.9 1.7 0.4]
[5.5 2.3 4.   1.3]
[6.8 3.2 5.9 2.3]
[7.6 3.   6.6 2.1]
[5.1 3.5 1.4 0.2]
[4.9 3.1 1.5 0.1]
[5.2 3.4 1.4 0.2]
[5.7 2.8 4.5 1.3]
[6.6 3.   4.4 1.4]
[5.   3.2 1.2 0.2]
[5.1 3.3 1.7 0.5]
[6.4 2.9 4.3 1.3]
[5.4 3.4 1.5 0.4]
[7.7 2.6 6.9 2.3]
[4.9 2.4 3.3 1.]
[7.9 3.8 6.4 2.]
[6.7 3.1 4.4 1.4]
[5.2 4.1 1.5 0.1]
[6.   3.   4.8 1.8]
[5.8 4.   1.2 0.2]
[7.7 2.8 6.7 2.]
[5.1 3.8 1.5 0.3]
[4.7 3.2 1.6 0.2]
[7.4 2.8 6.1 1.9]
[5.   3.3 1.4 0.2]
[6.3 3.4 5.6 2.4]
[5.7 2.8 4.1 1.3]
[5.8 2.7 3.9 1.2]
[5.7 2.6 3.5 1.]
[6.4 3.2 5.3 2.3]
[6.7 3.   5.2 2.3]
[6.3 2.5 4.9 1.5]

```
 [ 6.7  3.   5.   1.7]
 [ 5.   3.   1.6  0.2]
 [ 5.5  2.4  3.7  1. ]
 [ 6.7  3.1  5.6  2.4]
 [ 5.8  2.7  5.1  1.9]
 [ 5.1  3.4  1.5  0.2]
 [ 6.6  2.9  4.6  1.3]
 [ 5.6  3.   4.1  1.3]
 [ 5.9  3.2  4.8  1.8]
 [ 6.3  2.3  4.4  1.3]
 [ 5.5  3.5  1.3  0.2]
 [ 5.1  3.7  1.5  0.4]
 [ 4.9  3.1  1.5  0.1]
 [ 6.3  2.9  5.6  1.8]
 [ 5.8  2.7  4.1  1. ]
 [ 7.7  3.8  6.7  2.2]
 [ 4.6  3.2  1.4  0.2]]
y_train
[1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 0 2 1 1 1 1 2 0 0 2 1 0 0 1 0 2 1 0 1 2 1
 0 2 2 2 2 0 0 2 2 0 2 0 2 2 0 0 2 0 0 0 1 2 2 0 0 0 1 1 0 0 1 0 2 1 2 1 0
 2 0 2 0 0 2 0 2 1 1 1 2 2 1 1 0 1 2 2 0 1 1 1 1 0 0 0 2 1 2 0]
X_test
[[ 5.8  2.8  5.1  2.4]
 [ 6.   2.2  4.   1. ]
 [ 5.5  4.2  1.4  0.2]
 [ 7.3  2.9  6.3  1.8]
 [ 5.   3.4  1.5  0.2]
 [ 6.3  3.3  6.   2.5]
 [ 5.   3.5  1.3  0.3]
 [ 6.7  3.1  4.7  1.5]
 [ 6.8  2.8  4.8  1.4]
 [ 6.1  2.8  4.   1.3]
 [ 6.1  2.6  5.6  1.4]
 [ 6.4  3.2  4.5  1.5]
 [ 6.1  2.8  4.7  1.2]
 [ 6.5  2.8  4.6  1.5]
 [ 6.1  2.9  4.7  1.4]
 [ 4.9  3.1  1.5  0.1]
 [ 6.   2.9  4.5  1.5]
 [ 5.5  2.6  4.4  1.2]
 [ 4.8  3.   1.4  0.3]
 [ 5.4  3.9  1.3  0.4]
 [ 5.6  2.8  4.9  2. ]
 [ 5.6  3.   4.5  1.5]
 [ 4.8  3.4  1.9  0.2]
 [ 4.4  2.9  1.4  0.2]
```

```
 [ 6.2  2.8  4.8  1.8]
 [ 4.6  3.6  1.   0.2]
 [ 5.1  3.8  1.9  0.4]
 [ 6.2  2.9  4.3  1.3]
 [ 5.   2.3  3.3  1. ]
 [ 5.   3.4  1.6  0.4]
 [ 6.4  3.1  5.5  1.8]
 [ 5.4  3.   4.5  1.5]
 [ 5.2  3.5  1.5  0.2]
 [ 6.1  3.   4.9  1.8]
 [ 6.4  2.8  5.6  2.2]
 [ 5.2  2.7  3.9  1.4]
 [ 5.7  3.8  1.7  0.3]
 [ 6.   2.7  5.1  1.6]
 [ 5.9  3.   4.2  1.5]
 [ 5.8  2.6  4.   1.2]
 [ 6.8  3.   5.5  2.1]
 [ 4.7  3.2  1.3  0.2]
 [ 6.9  3.1  5.1  2.3]
 [ 5.   3.5  1.6  0.6]
 [ 5.4  3.7  1.5  0.2]]
y_test
[2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 1 1 1 2 0 2 0 0]
('Perceptron Identification rate:', 0.59999999999999998)
('MLPC       Identification rate:', 0.93333333333333335)
('MLPC(sgd)  Identification rate:', 0.82222222222222219)
('LinearSVC  Identification rate:', 0.93333333333333335)
```