기계학습원론 HW3

2017313260 이재민

1. 환경

OS : Windows 10

Version : Python 3.7.5

2. 코드

a) Numpy만을 이용한 regression

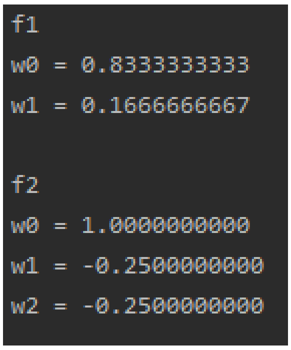
```
import numpy as np


def getXY(data):
    x_data = []
    y_data = []
    for index in range(len(data)):
        x_data.append(data[index][0])
        y_data.append(data[index][1])
    return x_data, y_data


def f1_matrix(x_data):
    # f1 = [1, cos(pi*x)]
    fmatrix = []
    for x in x_data:
        fmatrix.append([1, np.cos(np.pi*x)])
    return fmatrix


def f2_matrix(x_data):
    # f2 = [1, x, x**2]
    fmatrix = []
    for x in x_data:
        fmatrix.append([1, x, x**2])
    return fmatrix


def calcCoef(hMatrix, y_data):
    matrix = np.array(hMatrix)
    y = np.array(y_data)
    hTh = np.dot(matrix.transpose(), matrix)
    inverse_hTh = np.linalg.inv(hTh)
    hTy = np.dot(matrix.transpose(), y)
    w = np.dot(inverse_hTh, hTy)
    return w


if __name__ == '__main__':
    sample_data = [[-1, 1], [0, 1], [1, 1], [1, 0]]
    xData, yData = getXY(sample_data)

    w1 = calcCoef(f1_matrix(xData), yData)
    w2 = calcCoef(f2_matrix(xData), yData)

    print('f1')
    for i in range(len(w1)):
        print('w{} = {:.10f}'.format(i, w1[i]))

    print('\nf2')
    for i in range(len(w2)):
        print('w{} = {:.10f}'.format(i, w2[i]))
```

b) Scikit-Learn을 이용한 regression

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import FunctionTransformer
from sklearn.preprocessing import PolynomialFeatures


def kernel(x_):
    xnew = np.hstack([np.cos(np.pi*x_)])
    return xnew


def model1(x_, y_):
    xx = FunctionTransformer(kernel).fit_transform(x_)
    coef_output(xx, y_)


def model2(x_, y_):
    poly_features = PolynomialFeatures(degree=2, include_bias=False)
    xx = poly_features.fit_transform(x_)
    coef_output(xx, y_)


def coef_output(x_, y_):
    model = LinearRegression()
    model.fit(x_, y_)
    print('constant {}'.format(model.intercept_))
    print('coef {}'.format(model.coef_))


if __name__ == '__main__':
    data = np.array([[-1, 1], [0, 1], [1, 1], [1, 0]])

    x, y = data[:, 0], data[:, 1]
    x = x.reshape((len(x), 1))
    y = y.reshape((len(y), 1))

    print('f1')
    model1(x, y)
    print('f2')
    model2(x, y)
```

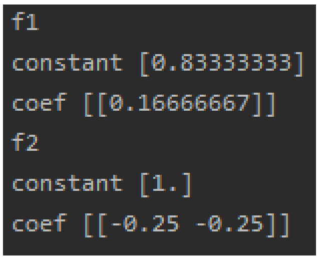
3. 결과

a) Numpy만을 이용한 regression



```
f1
w0 = 0.8333333333
w1 = 0.1666666667

f2
w0 = 1.0000000000
w1 = -0.2500000000
w2 = -0.2500000000
```

b) Scikit-Learn을 이용한 regression



```
f1
constant [0.83333333]
coef [[0.16666667]]
f2
constant [1.]
coef [[-0.25 -0.25]]
```

손으로 푼 것과 Numpy를 이용한 것, 그리고 Scikit-Learn을 이용한 결과가 동일하게 나온다.