

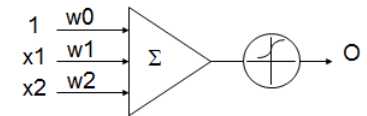
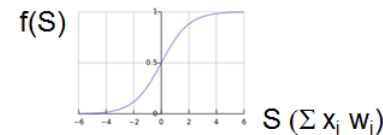
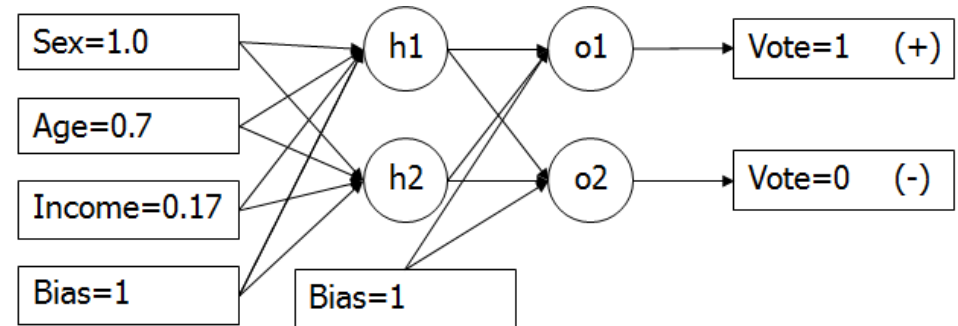
Neural Networks

Practice6

실습 1. Backpropagation

■ Solving a backpropagation example

example no.	sex	age	income	vote(+)	vote(-)
1	1.0	0.7	0.17	1	0
2	-1.0	-1	-0.6	0	1
3	-1.0	0.25	0.4	1	0
4	1.0	-0.1	0.6	0	1
5	-1.0	-0.75	-1	1	0
6	1.0	-0.5	0.27	0	1
7	1.0	0.4	-0.27	1	0
8	-1.0	-0.4	-0.33	1	0
New data	1.0	-0.55	-0.23	0	1

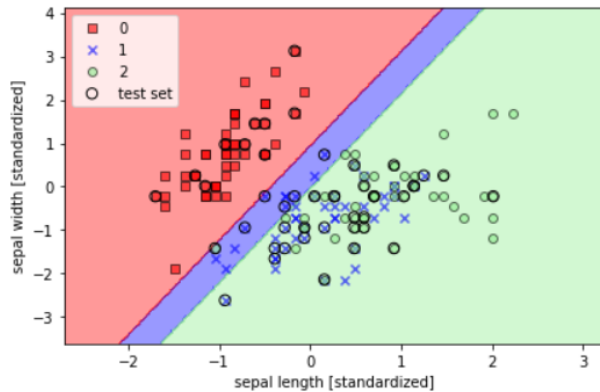


실습 2. Decision Region of Neural Networks

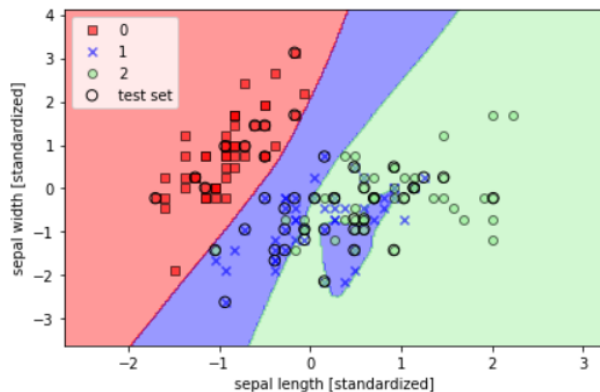
■ Sklearn

■ Make a decision Region of MLP

One Neuron

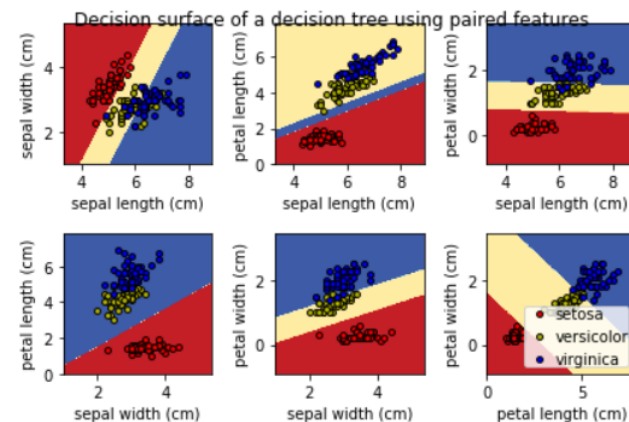
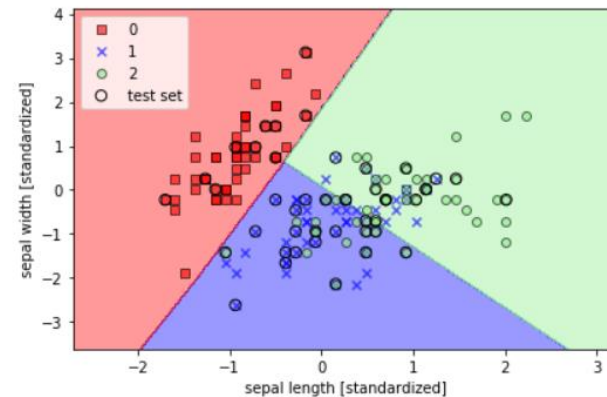


Multi-layer Perceptron



■ Make Following Results

Accuracy
0.9733333333333334

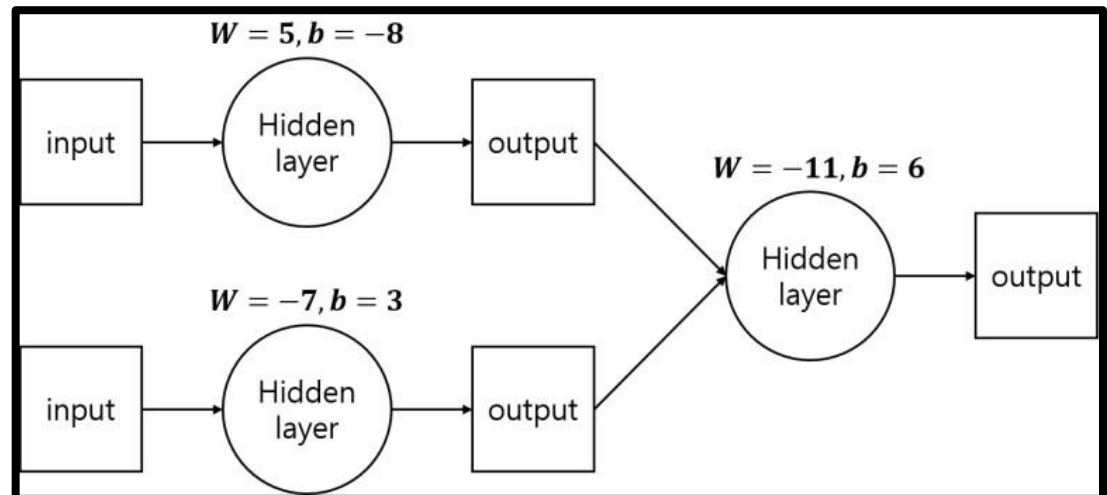
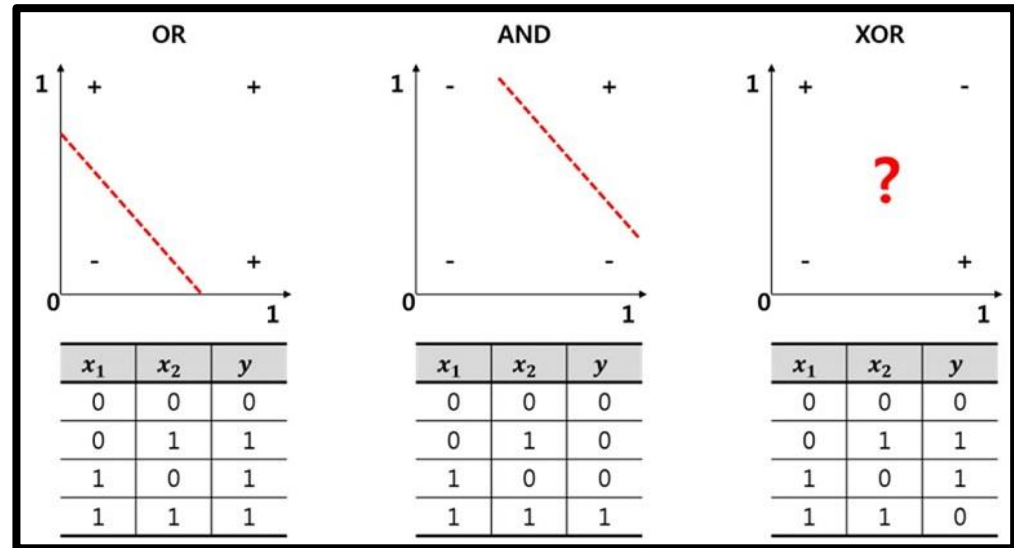


실습 3. Solving a XOR Problem

■ XOR Problem

■ Multi-layer Perceptron

```
#####  
Final Results...  
#####  
  
Predicted data based on trained weights:  
  
<Train data>  
Input:  
[[0. 0.]  
 [0. 1.]  
 [1. 0.]  
 [1. 1.]]  
Actual Output:  
[[0.]  
 [1.]  
 [1.]  
 [0.]]  
Predicted Output:  
[[0.04423811]  
 [0.95518522]  
 [0.95042826]  
 [0.05113156]]  
  
<Test data>  
Input:  
[[0. 0.]  
 [0. 1.]  
 [1. 0.]  
 [1. 1.]]  
Actual Output:  
[[0.]  
 [1.]  
 [1.]  
 [0.]]  
Predicted Output:  
[[0.04423811]  
 [0.95518522]  
 [0.95042826]  
 [0.05113156]]
```



실습 4. Custom Neural Networks

■ Forward & backward function Implementation

example no.	sex	age	income	vote(+)	vote(-)
1	1.0	0.7	0.17	1	0
2	-1.0	-1	-0.6	0	1
3	-1.0	0.25	0.4	1	0
4	1.0	-0.1	0.6	0	1
5	-1.0	-0.75	-1	1	0
6	1.0	-0.5	0.27	0	1
7	1.0	0.4	-0.27	1	0
8	-1.0	-0.4	-0.33	1	0
New data	1.0	-0.55	-0.23	0	1

```
<Train data>
Input:
[[ 1.    0.7    0.17]
 [-1.   -1.   -0.6 ]
 [-1.    0.25  0.4 ]
 [ 1.   -0.1   0.6 ]
 [-1.   -0.75 -1.  ]
 [ 1.   -0.5   0.27]
 [ 1.    0.4  -0.27]
 [-1.   -0.4  -0.33]]
Actual Output:
[[1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]]
Predicted Output:
[[0.98837464 0.01509107]
 [0.09117726 0.90956967]
 [0.99463774 0.00519892]
 [0.02085673 0.97989383]
 [0.94265666 0.05274743]
 [0.01183164 0.98911128]
 [0.97740921 0.02788956]
 [0.9385171  0.06110966]]

<Test data>
Input:
[ 1.   -0.55 -0.23]
Actual Output:
[0. 1.]
Predicted Output:
[0.01547253 0.98759013]
```

```
class Neural_Network(object):
    def __init__(self):
        # fill the following (1), (2), (3)
        #parameters
        (1)
        (2)
        (3)
```

```
def forward(self, X):
    #implementation forward function
    (4)
```

```
def backward(self, X, y, o, alpha):
    #implementation backward function
    (5)
```