

Exercise - Vægttræning

Disse opgaver i vægtudregning skal give indblik i hvordan vægtene i et neuralt netværk påvirker inputtet.

Gennem øvelserne skal følgende formel anvendes

$$\sum_j w_j \cdot x_j$$

hvor

$$\text{output} = \begin{cases} 0 & \text{if } x = \sum_j w_j \cdot x_j \leq \text{threshold} \\ 1 & \text{if } x = \sum_j w_j \cdot x_j > \text{threshold} \end{cases}$$

Opgave 1

Beregn output, når

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 0 \\ x_3 &= 0.4 \\ w_1 &= 1 \\ w_2 &= 1 \\ w_3 &= 0.7 \\ \text{threshold} &= 0.67 \end{aligned}$$

```
In [1]: def weight_calculator(X, W, threshold):
        """
        Denne funktion udregner outputtet givet et input og tilhørende vægte.
        """
        sum_wx = 0
        for idx, x in enumerate(X):
            sum_wx += x*W[idx]

        if sum_wx > threshold:
            return 1
        else:
            return 0

x1, x2, x3 = 1, 0, .4
w1, w2, w3 = 1, 1, .7
X = [x1, x2, x3]
W = [w1, w2, w3]
threshold = .67
print(weight_calculator(X, W, threshold))
```

1

Således kan det ses at output = 1

Opgave 2

Beregn output, når

$$\begin{aligned}x_1 &= 1 \\x_2 &= 0 \\x_3 &= 0.4 \\w_1 &= -1 \\w_2 &= 1 \\w_3 &= -0.35 \\ \text{threshold} &= 0.1\end{aligned}$$

```
In [2]: # Using the previous function
x1, x2, x3 = 1, 0, .4
w1, w2, w3 = -1, 1, -.35
X = [x1, x2, x3]
W = [w1, w2, w3]
threshold = .1
print(weight_calculator(X, W, threshold))

0
```

Exercise 3

Beregn output, når

$$\begin{aligned}w_1 &= 1 \\w_2 &= 1 \\w_3 &= 1 \\ \text{threshold} &= 2.5\end{aligned}$$

```
In [3]: def x_tables(W, threshold):
        """
        Denne funktion udregner x-tabellen.
        """
        outputs = {}
        for x1 in range(2):
            for x2 in range(2):
                for x3 in range(2):
                    outputs[x1,x2,x3] = weight_calculator([x1,x2,x3],W, threshold)

        return outputs

W = [1,1,1]
threshold = 2.5
print(x_tables(W, threshold))

{(0, 0, 0): 0, (0, 0, 1): 0, (0, 1, 0): 0, (0, 1, 1): 0, (1, 0, 0): 0, (1, 0, 1): 0, (1, 1, 0): 0, (1, 1, 1): 1}
```

Opgave 4

Beregn output, når

$$w_1 = 2$$

$$w_2 = 2$$

$$w_3 = 1$$

$$\text{threshold} = 4.1$$

```
In [4]: W = [2,2,1]
threshold = 4.1
print(x_tables(W, threshold))
```

```
{(0, 0, 0): 0, (0, 0, 1): 0, (0, 1, 0): 0, (0, 1, 1): 0, (1, 0, 0): 0, (1, 0, 1): 0, (1, 1, 0): 0, (1, 1, 1): 1}
```

Opgave 5

Udtænk værdier for w_1, w_2, w_3 og threshold, der skal give følgende tabel:

x_1	x_2	x_3	Output
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
1	1	1	1

```
In [5]: W = [1,1,0]
threshold = 1.1
output = x_tables(W, threshold)
print(("x1", "x2", "x3"), "output")
for k in output:
    print(k, output[k])
```

```
('x1', 'x2', 'x3') output
(0, 0, 0) 0
(0, 0, 1) 0
(0, 1, 0) 0
(0, 1, 1) 0
(1, 0, 0) 0
(1, 0, 1) 0
(1, 1, 0) 1
(1, 1, 1) 1
```

Opgave 6

Udfordring: Lav et 3:2:1 netværk. Input er x_1, x_2, x_3 . Det skal give følgende outputtabel:

x_1	x_2	x_3	Output
0	0	0	1
0	0	1	0
0	1	0	0

x_1	x_2	x_3	Output
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1

```
In [6]: def three_two_one_network(X, W1, W2, W3, threshold):
        """
        Denne funktion udregner i 3:2:1 netværk.
        """
        w1x = 0 #N1

        for idx, x in enumerate(X):
            w1x += x*W1[idx]

        if w1x > threshold[0]:
            w1x = 1
        else:
            w1x = 0
        #print("N1", w1x)

        w2x = 0
        for idx, x in enumerate(X):
            w2x += x*W2[idx]

        if w2x > threshold[1]:
            w2x = 1
        else:
            w2x = 0
        #print("N2", w2x)

        w3x = 0
        for idx, x in enumerate([w1x,w2x]):
            w3x += x*W3[idx]
        #print("N3",w3x)

        if w3x > threshold[2]:
            #print(1)
            return 1
        else:
            #print(0)
            return 0

        W1 = [1, 1, 1] # griber den ene case
        W2 = [-1, -1, -1] # griber den anden case
        W3 = [1, 1]
        thresholds = [2.5, -0.5, .5]
        outputs = {}

        for x1 in range(2):
            for x2 in range(2):
                for x3 in range(2):

                    outputs[x1,x2,x3] = three_two_one_network([x1,x2,x3],W1,W2,W3, threshold)

        print(("x1","x2","x3"), "output")
        for k in outputs:
            print(k, outputs[k])
```

```
('x1', 'x2', 'x3') output  
(0, 0, 0) 1  
(0, 0, 1) 0  
(0, 1, 0) 0  
(0, 1, 1) 0  
(1, 0, 0) 0  
(1, 0, 1) 0  
(1, 1, 0) 0  
(1, 1, 1) 1
```

Jeg håber, det giver mening.