

**Laporan Kecerdasan Buatan
(Multiple Neuron Batch Input)**



Disusun Oleh:

Diego Athalla Samudero (21091397042)

Program Studi D4 Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

2022

Source code

```
Multi Neuron Batch Inputpy <
1  #Diego Athalla Samudero
2  #21091397042
3
4  #mengimport numpy library sebagai np
5  import numpy as np
6
7  #menginisialisasi input
8  inputs = [ [1.4, 3.4, 3.4, 1.6, 2.8, 3.9, 4.4, 5.9, 6.7, 7.9],
9            [2.5, 1.7, 8.3, 6.5, 7.6, 5.6, 8.5, 1.9, 3.9, 2.7],
10           [1.3, 8.5, 7.6, 9.8, 4.5, 5.8, 4.7, 8.9, 8.8, 7.7],
11           [2.1, 3.2, 4.1, 1.1, 6.2, 7.2, 5.9, 9.1, 4.3, 6.6],
12           [3.3, 4.4, 6.3, 1.7, 1.3, 2.6, 7.3, 8.2, 6.2, 3.1],
13           [5.8, 2.5, 1.4, 5.8, 3.1, 0.2, 6.3, 6.9, 7.9, 5.8]]
14
15  #panjang weights1
16  weights1 = [[1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 1.5, 2.6, 3.1, 4.2],
17             [7.8, 8.9, 9.1, 2.4, 2.5, 7.5, 7.3, 6.7, 5.5, 2.5],
18             [9.3, 1.2, 5.2, 4.9, 9.8, 3.7, 1.4, 2.1, 5.6, 7.7],
19             [2.6, 7.5, 5.4, 4.3, 3.2, 2.1, 5.7, 6.1, 2.2, 1.1],
20             [4.4, 5.4, 5.9, 6.1, 9.2, 8.3, 3.6, 6.3, 2.1, 0.3]]
21
22  #jumlah biases pada layer1
23  biases1 = [1.4, 9.3, 2.5, 4.9, 2.9]
24
25  #panjang weight2
26  weights2 = [ [0.1, 3.2, 4.4, 6.4, 8.6],
27              [4.2, 1.1, 1.5, 7.8, 1.5],
28              [9.6, 8.4, 6.8, 3.7, 2.4]]
29
30  #jumlah biases pada layer2
31  biases2 = [3.4, 2.1, 1.7]
32
33  #menghitung layer1 menggunakan inputs, weights1, dan biases1
34  layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
```

```
Multi Neuron Batch Inputpy <
12           [3.3, 4.4, 6.3, 1.7, 1.3, 2.6, 7.3, 8.2, 6.2, 3.1],
13           [5.8, 2.5, 1.4, 5.8, 3.1, 0.2, 6.3, 6.9, 7.9, 5.8]]
14
15  #panjang weights1
16  weights1 = [[1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 1.5, 2.6, 3.1, 4.2],
17             [7.8, 8.9, 9.1, 2.4, 2.5, 7.5, 7.3, 6.7, 5.5, 2.5],
18             [9.3, 1.2, 5.2, 4.9, 9.8, 3.7, 1.4, 2.1, 5.6, 7.7],
19             [2.6, 7.5, 5.4, 4.3, 3.2, 2.1, 5.7, 6.1, 2.2, 1.1],
20             [4.4, 5.4, 5.9, 6.1, 9.2, 8.3, 3.6, 6.3, 2.1, 0.3]]
21
22  #jumlah biases pada layer1
23  biases1 = [1.4, 9.3, 2.5, 4.9, 2.9]
24
25  #panjang weight2
26  weights2 = [ [0.1, 3.2, 4.4, 6.4, 8.6],
27              [4.2, 1.1, 1.5, 7.8, 1.5],
28              [9.6, 8.4, 6.8, 3.7, 2.4]]
29
30  #jumlah biases pada layer2
31  biases2 = [3.4, 2.1, 1.7]
32
33  #menghitung layer1 menggunakan inputs, weights1, dan biases1
34  layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
35
36  #menghitung layer2 dari hasil perhitungan layer1
37  layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
38
39  #print output layer2
40  print(layer2_outputs)
```

```
#Diego Athalla Samudero
```

```
#21091397042
```

```
#mengimport numpy library sebagai np
```

```
import numpy as np
```

```
#menginisialisasi input
```

```
inputs = [ [1.4, 3.4, 3.4, 1.6, 2.8, 3.9, 4.4, 5.9, 6.7, 7.9],
           [2.5, 1.7, 8.3, 6.5, 7.6, 5.6, 8.5, 1.9, 3.9, 2.7],
```

```

        [1.3, 8.5, 7.6, 9.8, 4.5, 5.8, 4.7, 0.9, 8.8, 7.7],
        [2.1, 3.2, 4.1, 1.1, 6.2, 7.2, 5.9, 9.1, 4.3, 6.5],
        [3.3, 4.4, 6.3, 1.7, 1.3, 2.6, 7.3, 8.2, 6.2, 3.1],
        [5.8, 2.5, 1.4, 5.8, 3.1, 0.2, 6.3, 6.9, 7.9, 5.8]]

#panjang weights1
weights1 = [[1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 1.5, 2.6, 3.1, 4.2],
            [7.8, 8.9, 9.1, 2.4, 2.5, 7.5, 7.3, 6.7, 5.5, 2.5],
            [9.3, 1.2, 5.2, 4.9, 9.8, 3.7, 1.4, 2.1, 5.6, 7.7],
            [2.6, 7.5, 5.4, 4.3, 3.2, 2.1, 5.7, 6.1, 2.2, 1.1],
            [4.4, 5.4, 5.9, 6.1, 9.2, 8.3, 3.6, 6.3, 2.1, 0.3]]

#jumlah biases pada layer1
biases1 = [1.4, 9.3, 2.5, 4.9, 2.9]

#panjang weight2
weights2 = [ [0.1, 3.2, 4.4, 6.4, 8.6],
             [4.2, 1.1, 1.5, 7.8, 1.5],
             [9.6, 8.4, 6.8, 3.7, 2.4]]

#jumlah biases pada layer2
biases2= [3.4, 2.1, 1.7]

#menghitung layer1 menggunakan inputs, weights1, dan biases1
layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1

#menghitung layer2 dari hasil perhitungan layer1
layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2

#print output layer2
print(layer2_outputs)

```

Output

```

[[4333.888 2734.067 5939.801]
 [5822.252 3516.447 7481.171]
 [6505.152 4102.15  8712.474]
 [5654.477 3460.096 7402.562]
 [4991.558 3113.984 6431.319]
 [4803.047 2966.888 6333.389]]

```

```

[[4333.888 2734.067 5939.801]
 [5822.252 3516.447 7481.171]
 [6505.152 4102.15  8712.474]
 [5654.477 3460.096 7402.562]
 [4991.558 3113.984 6431.319]
 [4803.047 2966.888 6333.389]]

```

Memasukkan numpy dan menginisialisasi np sebagai method perhitungan

Input=10*6

Weight1=5*10

Neuron1=5

Bias1=5

Weight2=3*5

Bias2=3

Layer output berfungsi menghitung variable yang di inputkan

np.dot berfungsi menghitung vector weight dan vector input