

**Kecerdasan Buatan**



**Oleh :**

**Reiznu Ahmad Tjandrida**

**21091397018 (2021B)**

**UNIVERSITAS NEGERI SURABAYA**

**MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**2022**

## Kode Program

```
1  # Reiznu Ahmad Tjandrida
2  # 21091397018
3
4  # UTS AI ke 2
5
6  # Mengimport Library numpy, dan memberi inisial
7  import numpy as np
8
9  # Layer input 10 featurese
10 inputs = [
11     [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.9, 4.5, 5.0, 5.5],
12     [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
13     [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5],
14     [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
15     [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
16     [12.5, 16.4, 17.2, 17.4, 18.2, 18.4, 19.2, 19.4, 10.2, 10.4],
17 ]
18
19 weights_1 = [
20     [1.0, 1.5, 2.0, 2.5, 3.7, 3.5, 4.7, 4.5, 5.0, 5.5],
21     [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
22     [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
23     [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
24     [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]
25 ]
```

Pada baris ke 7, terdapat sintaks untuk mengimport library dari python yaitu numpy yang kemudian diberikan nama alias np untuk memudahkan pemanggilan dan penggunaannya.

Pada baris ke baris 10 terdapat variabel inputs yang memiliki 6 batch dengan 10 layer per batch nya.

```

19 weights_1 = [
20     [1.0, 1.5, 2.0, 2.5, 3.7, 3.5, 4.7, 4.5, 5.0, 5.5],
21     [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
22     [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
23     [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
24     [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]
25 ]
26
27 biases_1 = [1.5, 2.3, 3.1, 4.7, 5.8]
28
29 weights_2 = [
30     [0.2, 0.4, 0.6, 0.8, 1.0],
31     [2.7, 1.8, 2.6, 2.8, 3.6],
32     [1.0, 1.5, 2.0, 2.5, 3.7]
33 ]
34 biases_2 = [1.9, 5.6, 4.3]
35
36 # np.array, membuat sebuah matriks dari array yang ada
37 outputs_1 = np.dot(inputs, np.array(weights_1) . T) + biases_1
38 outputs_2 = np.dot(outputs_1, np.array(weights_2) . T) + biases_2
39 print(outputs_2)
40

```

Kemudian pada baris 19 ada hidden layer yang diberi nama `weights_1` yang memiliki 5 batch yang dibarengi oleh variabel bias yang memiliki 5 layer, sesuai banyaknya batch pada `weights_1`.

Hal yang sama juga terjadi pada hidden layer kedua yaitu pada baris 29-34.

Kemudian pada baris 37 dan 38 terdapat rumus untuk menghitung hidden layer dan hidden layer 2. Bedanya pada `outputs_2`, hasil dari `outputs_1` nantinya akan dikalikan dengan `weights_2` kemudian ditambah dengan bias.

## Penjelasan perhitungan angka neuron

```
inputs = [
    [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5],
    [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
    [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5],
    [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
    [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
    [12.5, 16.4, 17.2, 17.4, 18.2, 18.4, 19.2, 19.4, 19.2, 10.4],
]
```

Hidden Layer 1

```
weights_1 = [
    [1.0, 1.5, 2.0, 2.5, 3.7, 3.5, 4.7, 4.5, 5.0, 5.5],
    [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
    [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
    [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
    [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]
]
```

```
biases_1 = [1.5, 2.3, 3.1, 4.7, 5.8]
```

Hidden Layer 2

```
weights_2 = [
    [0.2, 0.4, 0.6, 0.8, 1.0],
    [2.7, 1.8, 2.6, 2.8, 3.6],
    [1.0, 1.5, 2.0, 2.5, 3.7]
]
```

```
biases_2 = [1.9, 5.0, 4.3]
```

Hasil dari rumus  $\rightarrow$   $\text{np.dot(inputs, np.array(weights\_1).T) + biases\_1}$   
 $\text{np.dot(outputs\_1, np.array(weights\_2).T) + biases\_2}$

Setiap inputs dan hidden layer dikalikan dengan Transpose dan ditambahkan dengan bias

Setiap inputs dan hidden layer dikalikan dengan Transpose dan ditambahkan dengan bias

print(outputs\_1)

136.88	133.88	148.84	299.28	1227.05
134.48	132.11	147.43	295.01	1204.35
1235.75	1200.85	1325.25	2699.25	11225.3
148.84	146.63	164.83	328.73	1327.95
299.98	292.61	327.13	681.51	2700.35
533.98	519.61	594.13	1268.51	4715.35

Hasil tanpa menggunakan bias

np.dot(inputs, np.array(weights\_1).T)

1636.706	6252.908	5923.665
1608.556	6145.9	5821.125
14907.34	56912.685	53969.26
1778.252	6795.424	6433.685
3618.876	13816.67	13088.225
6401.276	24448.87	23119.725

print(outputs\_2)

1638.606	6258.508	5927.965
1610.456	6151.5	5825.425
14909.24	56918.285	53973.56
1780.152	6801.024	6437.985
3620.776	13822.27	13092.525
6403.176	24454.47	23124.025

Hasil tanpa menggunakan bias

np.dot(outputs\_1, np.array(weights\_2).T)

135.38	131.58	145.74	294.58	1221.25
132.98	129.81	144.33	290.31	1198.55
1234.25	1198.55	1322.15	2694.55	11219.5
148.84	146.63	164.83	328.73	1327.95
299.98	292.61	327.13	681.51	2700.35
533.98	519.61	594.13	1268.51	4715.35

```
inputs = [
    [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.9, 4.5, 5.0, 5.5],
    [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
    [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5],
    [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
    [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
    [12.5, 16.4, 17.2, 17.4, 18.2, 18.4, 19.2, 19.4, 10.2, 10.4],
]
```

#### Hidden Layer 1

```
weights_1 = [
    [1.0, 1.5, 2.0, 2.5, 3.7, 3.5, 4.7, 4.5, 5.0, 5.5],
    [1.5, 1.4, 2.2, 2.4, 3.2, 3.4, 4.2, 4.4, 5.2, 5.4],
    [2.7, 1.8, 2.6, 2.8, 3.6, 3.8, 4.6, 4.8, 5.6, 5.8],
    [2.5, 6.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
    [3.5, 18.5, 18.0, 20.5, 30.0, 30.5, 40.0, 40.5, 50.0, 50.5]
]
```

```
biases_1 = [1.5, 2.3, 3.1, 4.7, 5.8]
```

#### Hidden Layer 2

```
weights_2 = [
    [0.2, 0.4, 0.6, 0.8, 1.0],
    [2.7, 1.8, 2.6, 2.8, 3.6],
    [1.0, 1.5, 2.0, 2.5, 3.7]
]
```

```
biases_2 = [1.9, 5.6, 4.3]
```

Hasil dari rumus

→  $\text{np.dot}(\text{inputs}, \text{np.array}(\text{weights\_1}) . \text{T}) + \text{biases\_1}$   
 $\text{np.dot}(\text{outputs\_1}, \text{np.array}(\text{weights\_2}) . \text{T}) + \text{biases\_2}$



Setiap inputs dan hidden layer dikalikan dengan Transpose dan ditambahkan dengan bias

Pada hidden layer 1, setiap variabel data pada variabel input, dikalikan dengan hidden layer 1/weight\_1 ditambah dengan bias dengan menggunakan rumus diatas. Bias memiliki 3 angka yang sama dan harus sesuai dengan banyaknya batch pada neuronnya, yaitu weights\_1.

Pada hidden layer 2 hampir sama dengan hidden layer\_1. Bedanya Cuma terletak pada rumusnya. Pada rumus hidden layer 2, hasil dari hidden layer 1 akan dikalikan dengan weights\_2 lalu ditambahkan dengan bias.



Pada gambar diatas terdapat 2 hasil. Hasil pertama sebelum menggunakan bias. Dan hasil kedua tanpa menggunakan bias. Jadi setiap baris weights akan dikalikan dengan bias.