

Laporan Kecerdasan Buatan
Ujian Tengah Semester
2022



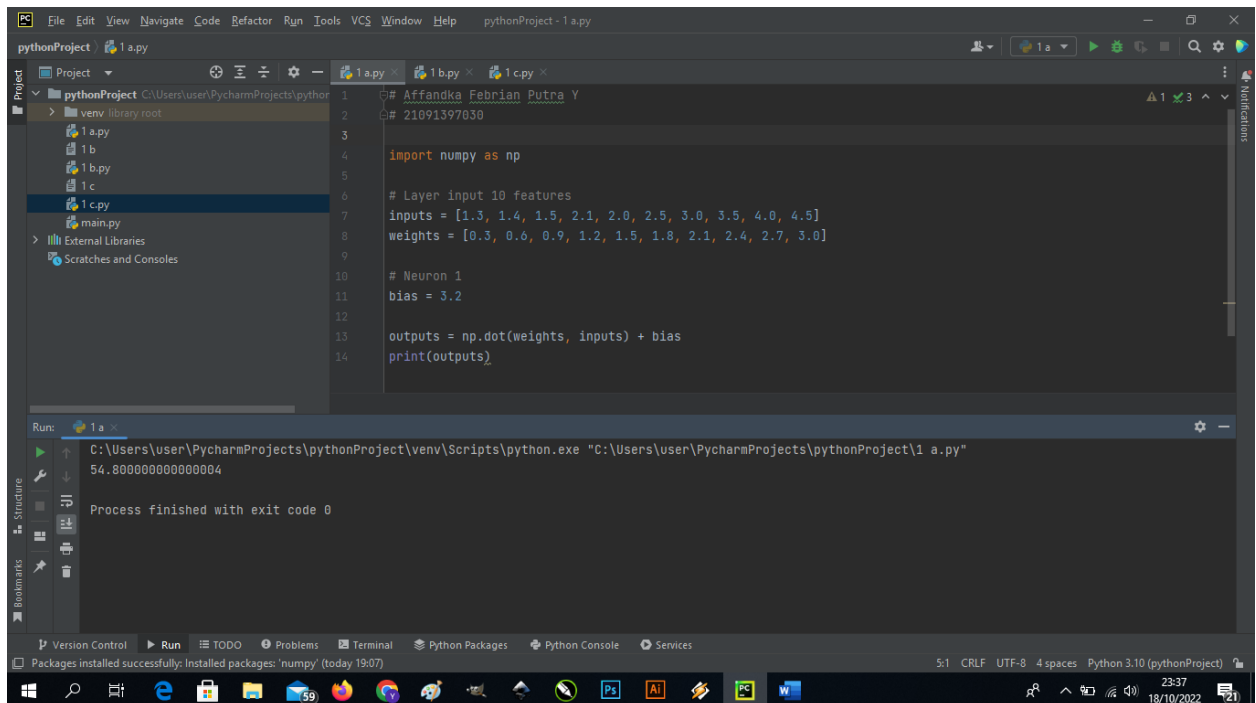
Oleh :

Affandika Febrian Putra Yunanto(21091397030)

Manajemen Informatika

Fakultas Vokasi

1. A.



```
pythonProject - 1 a.py
1  # Affandka Febrian Putra Y
2  # 21091397030
3
4  import numpy as np
5
6  # Layer input 10 features
7  inputs = [1.3, 1.4, 1.5, 2.1, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5]
8  weights = [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0]
9
10 # Neuron 1
11 bias = 3.2
12
13 outputs = np.dot(weights, inputs) + bias
14 print(outputs)
```

Run: 1 a

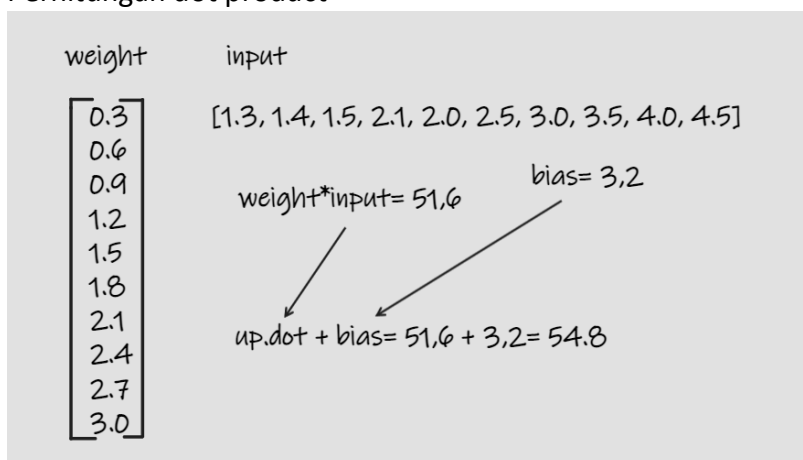
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\user\PycharmProjects\pythonProject\1 a.py"

54.800000000000004

Process finished with exit code 0

Sintax pada baris 4 yang berguna untuk meng-impor numpy (library python). Sintax variable input pada baris 7 yang bersifat data array yang berguna untuk menyimpan data layer dengan jumlah 10. Pada baris ke 8 variable weight yang berisi nilai neuron. Variable bias pada baris 11 yang berguna untuk menghitung hasil output. Pada baris 13 variable output yang berguna untuk menghitung single neuron yang dicetak pada baris 14, juga terdapat np.dot yang memiliki fungsi untuk mengembalikan nilai array. Saat dilakukan run dari source kode tersebut ditampilkan 54.800000000000004 sebagai hasil.

Perhitungan dot product



B.

```

1 import numpy as np
2
3 inputs = [1.3, 1.6, 1.9, 2.1, 2.4, 2.7, 3.0, 3.3, 3.6, 3.9]
4 weights = [
5     [0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0],
6     [2.4, 2.3, 2.2, 3.1, 3.4, 3.6, 3.8, 4.2, 4.6, 4.8],
7     [0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0],
8     [2.4, 2.3, 2.2, 3.1, 3.4, 3.6, 3.8, 4.2, 4.6, 4.8],
9     [0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0],
10 ]
11
12 biases = [1.3, 1.5, 1.4, 1.6, 1.8]
13
14 outputs = np.dot(weights, inputs) + biases
15 print(outputs)

```

Run: 1 b x

C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\user\PycharmProjects\pythonProject\1 b.py"

[67.54 97.41 67.64 97.51 68.04]

Process finished with exit code 0

Sintax pada baris pertama yang berguna untuk meng-impor numpy (library python).
 Sintax variable input pada baris 3 yang bersifat data array yang berguna untuk menyimpan data layer dengan jumlah 10. Pada baris ke 4 sampai baris 9 terdapat 5 baris variable weight yang berisi nilai neuron. Variable bias pada baris 12 yang berguna untuk menghitung hasil output. Pada baris 14 variable output yang berguna untuk menghitung multi neuron yang dicetak pada baris 15, juga terdapat np.dot yang memiliki fungsi untuk mengembalikan nilai array. Saat dilakukan run dari source kode tersebut ditampilkan [67.54 97.41 67.64 97.51 68.04] sebagai hasil.

Perhitungan dot product

weight	input
$\begin{bmatrix} 0.4 & 0.8 & 1.2 & 1.6 & 2.0 & 2.4 & 2.8 & 3.2 & 3.6 & 4.0 \\ 2.4 & 2.3 & 2.2 & 3.1 & 3.4 & 3.6 & 3.8 & 4.2 & 4.6 & 4.8 \\ 0.4 & 0.8 & 1.2 & 1.6 & 2.0 & 2.4 & 2.8 & 3.2 & 3.6 & 4.0 \\ 2.4 & 2.3 & 2.2 & 3.1 & 3.4 & 3.6 & 3.8 & 4.2 & 4.6 & 4.8 \\ 0.4 & 0.8 & 1.2 & 1.6 & 2.0 & 2.4 & 2.8 & 3.2 & 3.6 & 4.0 \end{bmatrix}$	$[1.3, 1.6, 1.9, 2.1, 2.4, 2.7, 3.0, 3.3, 3.6, 3.9]$
$\begin{bmatrix} (1.3*0.4) & (1.6*0.8) & (1.9*1.2) & (2.1*1.6) & (2.4*2.0) & (2.7*2.4) & (3*2.8) & (3.3*3.2) & (3.6*3.6) & (3.9*4.0) \\ (1.3*2.4) & (1.6*2.3) & (1.9*2.2) & (2.1*3.1) & (2.4*3.4) & (2.7*3.6) & (3*3.8) & (3.3*4.2) & (3.6*4.6) & (3.9*4.8) \\ (1.3*0.4) & (1.6*0.8) & (1.9*1.2) & (2.1*1.6) & (2.4*2.0) & (2.7*2.4) & (3*2.8) & (3.3*3.2) & (3.6*3.6) & (3.9*4.0) \\ (1.3*2.4) & (1.6*2.3) & (1.9*2.2) & (2.1*3.1) & (2.4*3.4) & (2.7*3.6) & (3*3.8) & (3.3*4.2) & (3.6*4.6) & (3.9*4.8) \\ (1.3*0.4) & (1.6*0.8) & (1.9*1.2) & (2.1*1.6) & (2.4*2.0) & (2.7*2.4) & (3*2.8) & (3.3*3.2) & (3.6*3.6) & (3.9*4.0) \end{bmatrix}$	
<div style="display: flex; justify-content: space-between;"> <div> $weight*input = [66.24 \ 95.91 \ 66.24 \ 95.91 \ 66.24]$ </div> <div> $bias = 1.3, 1.5, 1.4, 1.6, 1.8$ </div> </div>	
$np.dot + bias = [66.24 \ 95.91 \ 66.24 \ 95.91 \ 66.24] + [1.3, 1.5, 1.4, 1.6, 1.8]$ $= [67.54 \ 97.41 \ 67.64 \ 97.51 \ 68.04]$	

— neuron 1
— neuron 2
— neuron 3
— neuron 4
— neuron 5

C.

```
pythonProject - 1 c.py
1 import numpy as np
2
3 inputs = [
4     [1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0],
5     [1.3, 1.6, 1.9, 2.1, 2.4, 2.7, 3.0, 3.3, 3.6, 3.9],
6     [3.3, 13.4, 14.0, 10.6, 20.5, 30.4, 30.5, 40.0, 40.5, 45.0],
7     [2.4, 2.3, 2.2, 3.1, 3.4, 3.6, 3.8, 4.2, 4.6, 4.8],
8     [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0],
9     [10.0, 14.4, 15.5, 16.5, 18.7, 19.4, 19.8, 20.5, 20.8, 30.5],
10 ]
11
12 weights = [
13     [1.2, 1.4, 2.2, 2.6, 2.8, 3.3, 3.7, 4.5, 5.4, 5.9],
14     [1.3, 1.6, 2.3, 2.6, 2.8, 3.3, 4.6, 3.8, 4.2, 4.8],
15     [2.2, 2.4, 2.8, 3.4, 3.6, 3.8, 4.3, 4.6, 5.3, 5.6],
16 ]
17
18
19
20 biases = [1.3, 1.5, 1.8, 2.3, 2.8]
21
22 outputs = np.dot(inputs, np.array(weights).T) + biases
23 print(outputs)
```

Run: 1 c.py

```
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\user\PycharmProjects\pythonProject\1 c.py"
[[ 79.16  73.6   87.88 164.24 566.92]
 [ 98.77  91.39 108.86 204.39 710.9 ]
 [1017.15 923.11 1087.58 2072.01 7530.45]
 [ 127.82 118.62 142.14 265.09 916.04]
 [ 68.59  62.7   73.92 139.61 495.73]
 [ 684.62 635.21 759.38 1427.51 4926.9 ]]
```

Process finished with exit code 0

Python Packages: numpy (today 19:07)

2:1 CRLF UTF-8 4 spaces Python 3.10 (pythonProject)

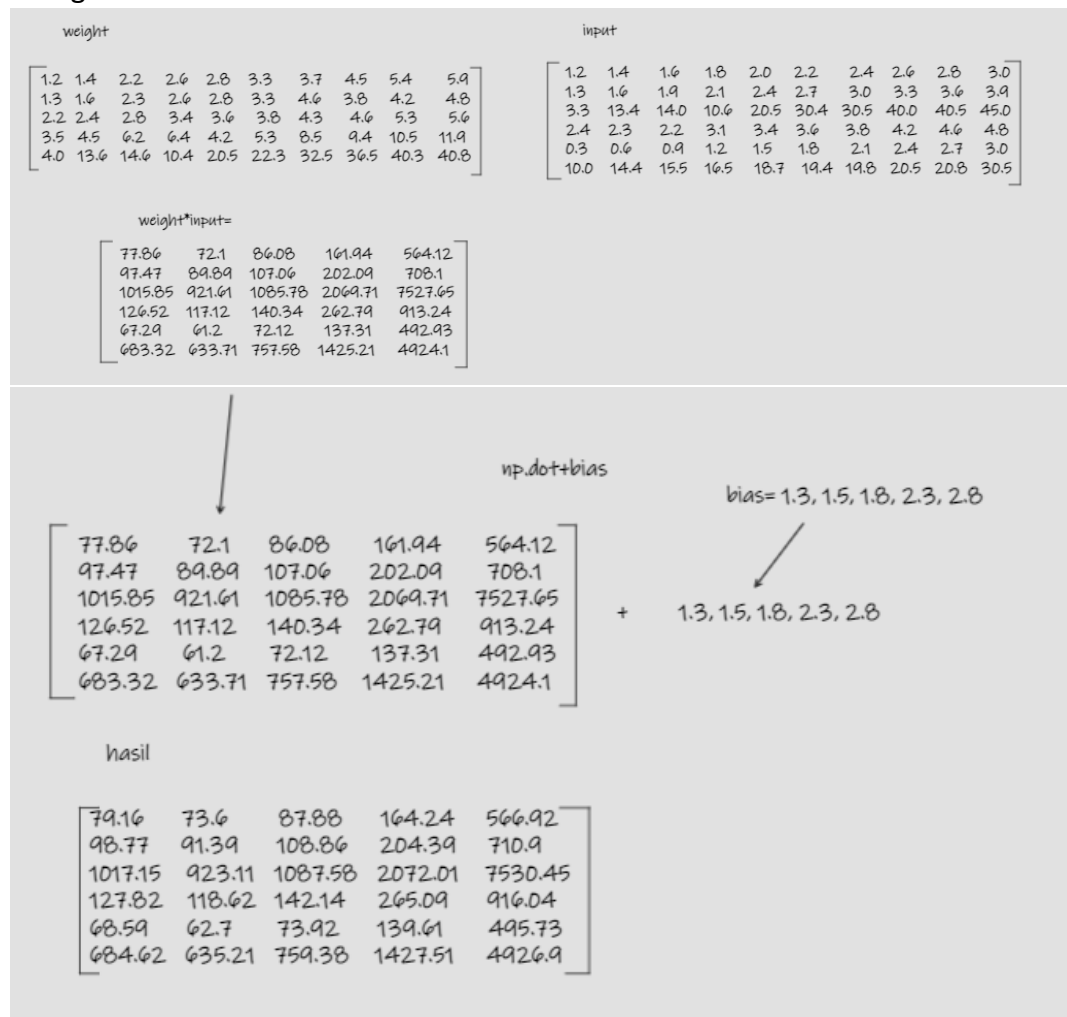
23:38 18/10/2022

Sintax pada baris pertama yang berguna untuk meng-impor numpy (library python). Terdapat 6 batch sintax variable input pada baris 3 sampai baris 9 yang bersifat data array yang berguna untuk menyimpan data layer dengan jumlah 10 pada tiap barisnya. Pada baris ke 12 sampai baris 17 terdapat 5 baris variable weight yang berisi nilai neuron. Variable bias pada baris 20 yang berguna untuk menghitung hasil output. Pada baris 22 terdapat np.dot yang memiliki fungsi untuk mengembalikan nilai array dan juga

terdapat np.array yang memiliki fungsi untuk mengembalikan variable weights yang mempunyai data array 2 dimensi. Saat dilakukan run dari source kode tersebut ditampilkan

```
[[ 79.16  73.6   87.88 164.24 566.92]
 [ 98.77  91.39 108.86 204.39 710.9 ]
 [1017.15 923.11 1087.58 2072.01 7530.45]
 [ 127.82 118.62 142.14 265.09 916.04]
 [ 68.59  62.7   73.92 139.61 495.73]
 [ 684.62 635.21 759.38 1427.51 4926.9 ]]
```

sebagai hasil.



UTS 2

1. Source Code

```
UTS2.py
1 import numpy as np
2 inputs = [
3     [1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0],
4     [1.3, 1.6, 1.9, 2.1, 2.4, 2.7, 3.0, 3.3, 3.6, 3.9],
5     [2.4, 2.3, 2.2, 3.1, 3.4, 3.6, 3.8, 4.2, 4.6, 4.8],
6     [2.4, 2.3, 2.2, 3.1, 3.4, 3.6, 3.8, 4.2, 4.6, 4.8],
7     [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0],
8     [10.0, 14.4, 15.5, 16.5, 18.7, 19.4, 19.8, 20.5, 20.8, 30.5],
9 ]
10 weights = [
11     [1.2, 1.4, 2.2, 2.6, 2.8, 3.3, 3.7, 4.5, 5.4, 5.9],
12     [1.3, 1.6, 2.3, 2.6, 2.8, 3.3, 4.6, 3.8, 4.2, 4.8],
13     [2.2, 2.4, 2.8, 3.4, 3.6, 3.8, 4.3, 4.6, 5.3, 5.6],
14     [3.5, 4.5, 6.2, 6.4, 4.2, 5.3, 8.5, 9.4, 10.5, 11.9],
15     [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0],
16 ]
17 biases = [1.3, 1.5, 1.8, 2.3, 2.8]
18 weights2 = [
19     [1.2, 1.4, 2.2, 2.6, 2.8],
20     [1.3, 1.6, 2.3, 2.6, 2.8],
21     [2.2, 2.4, 2.8, 3.4, 3.6],
22 ]
23 biases2 = [1.8, 2.2, 2.5]
24
25 layer1_outputs = np.dot(inputs, np.array(weights) . T) + biases
26 layer2_outputs = np.dot(layer1_outputs, np.array(weights2) . T) + biases2
```

```
UTS2.py
7     [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0],
8     [10.0, 14.4, 15.5, 16.5, 18.7, 19.4, 19.8, 20.5, 20.8, 30.5],
9 ]
10 weights = [
11     [1.2, 1.4, 2.2, 2.6, 2.8, 3.3, 3.7, 4.5, 5.4, 5.9],
12     [1.3, 1.6, 2.3, 2.6, 2.8, 3.3, 4.6, 3.8, 4.2, 4.8],
13     [2.2, 2.4, 2.8, 3.4, 3.6, 3.8, 4.3, 4.6, 5.3, 5.6],
14     [3.5, 4.5, 6.2, 6.4, 4.2, 5.3, 8.5, 9.4, 10.5, 11.9],
15     [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0],
16 ]
17 biases = [1.3, 1.5, 1.8, 2.3, 2.8]
18 weights2 = [
19     [1.2, 1.4, 2.2, 2.6, 2.8],
20     [1.3, 1.6, 2.3, 2.6, 2.8],
21     [2.2, 2.4, 2.8, 3.4, 3.6],
22 ]
23 biases2 = [1.8, 2.2, 2.5]
24
25 layer1_outputs = np.dot(inputs, np.array(weights) . T) + biases
26 layer2_outputs = np.dot(layer1_outputs, np.array(weights2) . T) + biases2
27 print(layer2_outputs)
```

Output:

```
UTS2
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\user\PycharmProjects\pythonProject\UTS 2.py"
[[ 938.912  970.736 1310.412]
 [1166.12 1205.561 1627.792]
 [1510.962 1562.082 2109.106]
 [1510.962 1562.082 2109.106]
 [ 802.358  829.549 1128.348]
 [8073.68 8345.522 11271.526]]

Process finished with exit code 0
```

Analysis:

Pada baris 1 terdapat syntax untuk import library python yaitu numpy. Terdapat 6 batch variable input pada baris 2 hingga 8 yang bertipe data array untuk menyimpan data yang berjumlah 10. Pada baris 10 hingga 15 terdapat 5 batch variabel weights

pertama yang berisi nilai neuron. Variable bias pertama pada baris 17 yang berjumlah 5 berguna untuk menghitung hasil output. Terdapat 3 batch variabel kedua yang berisi nilai neuron pada baris ke 19 hingga 21. Terdapat variabel bias kedua pada baris 23 yang berjumlah 3 berguna untuk menghitung hasil output. Pada baris 25 dan 26 variable output yang berguna untuk menghitung multi neuron yang masing-masing dicetak pada baris 27, juga terdapat np.dot yang memiliki fungsi untuk mengembalikan nilai array. Saat dilakukan run dari source kode tersebut ditampilkan output yang tertera diatas sebagai hasil.

Perhitungan layer 1

weight 10*5	input 6*10
$\begin{bmatrix} 1.2 & 1.4 & 2.2 & 2.6 & 2.8 & 3.3 & 3.7 & 4.5 & 5.4 & 5.9 \\ 1.3 & 1.6 & 2.3 & 2.6 & 2.8 & 3.3 & 4.6 & 3.8 & 4.2 & 4.8 \\ 2.2 & 2.4 & 2.8 & 3.4 & 3.6 & 3.8 & 4.3 & 4.6 & 5.3 & 5.6 \\ 3.5 & 4.5 & 6.2 & 6.4 & 4.2 & 5.3 & 8.5 & 9.4 & 10.5 & 11.9 \\ 0.3 & 0.6 & 0.9 & 1.2 & 1.5 & 1.8 & 2.1 & 2.4 & 2.7 & 3.0 \end{bmatrix}$	$\begin{bmatrix} 1.2 & 1.4 & 1.6 & 1.8 & 2.0 & 2.2 & 2.4 & 2.6 & 2.8 & 3.0 \\ 1.3 & 1.6 & 1.9 & 2.1 & 2.4 & 2.7 & 3.0 & 3.3 & 3.6 & 3.9 \\ 2.4 & 2.3 & 2.2 & 3.1 & 3.4 & 3.6 & 3.8 & 4.2 & 4.6 & 4.8 \\ 2.4 & 2.3 & 2.2 & 3.1 & 3.4 & 3.6 & 3.8 & 4.2 & 4.6 & 4.8 \\ 0.3 & 0.6 & 0.9 & 1.2 & 1.5 & 1.8 & 2.1 & 2.4 & 2.7 & 3.0 \\ 10.0 & 14.4 & 15.5 & 16.5 & 18.7 & 19.4 & 19.8 & 20.5 & 20.8 & 30.5 \end{bmatrix}$
weight*input=	
$\begin{bmatrix} 77.86 & 72.1 & 86.08 & 161.94 & 39.6 \\ 97.47 & 89.89 & 107.06 & 202.09 & 49.68 \\ 126.52 & 117.12 & 140.34 & 262.79 & 64.26 \\ 126.52 & 117.12 & 140.34 & 262.79 & 64.26 \\ 67.29 & 61.2 & 72.12 & 137.31 & 34.65 \\ 683.32 & 633.71 & 757.58 & 1425.21 & 346.8 \end{bmatrix}$	
	np.dot + bias=
$\begin{bmatrix} 77.86 & 72.1 & 86.08 & 161.94 & 39.6 \\ 97.47 & 89.89 & 107.06 & 202.09 & 49.68 \\ 126.52 & 117.12 & 140.34 & 262.79 & 64.26 \\ 126.52 & 117.12 & 140.34 & 262.79 & 64.26 \\ 67.29 & 61.2 & 72.12 & 137.31 & 34.65 \\ 683.32 & 633.71 & 757.58 & 1425.21 & 346.8 \end{bmatrix}$	$+ \begin{bmatrix} 1.3 & 1.5 & 1.8 & 2.3 & 2.8 \end{bmatrix}$
hasil	
$\begin{bmatrix} 79.16 & 73.6 & 87.88 & 164.24 & 42.4 \\ 98.77 & 91.39 & 108.86 & 204.39 & 52.48 \\ 127.82 & 118.62 & 142.14 & 265.09 & 67.06 \\ 127.82 & 118.62 & 142.14 & 265.09 & 67.06 \\ 68.59 & 62.7 & 73.92 & 139.61 & 37.45 \\ 684.62 & 635.21 & 759.38 & 1427.51 & 349.6 \end{bmatrix}$	

Perhitungan layer 2

weights 2 5*3

$$\begin{bmatrix} 1.2 & 1.4 & 2.2 & 2.6 & 2.8 \\ 1.3 & 1.6 & 2.3 & 2.6 & 2.8 \\ 2.2 & 2.4 & 2.8 & 3.4 & 3.6 \end{bmatrix}$$

output layer 1 5*6

$$\begin{bmatrix} 79.16 & 73.6 & 87.88 & 164.24 & 42.4 \\ 98.77 & 91.39 & 108.86 & 204.39 & 52.48 \\ 127.82 & 118.62 & 142.14 & 265.09 & 67.06 \\ 127.82 & 118.62 & 142.14 & 265.09 & 67.06 \\ 68.59 & 62.7 & 73.92 & 139.61 & 37.45 \\ 684.62 & 635.21 & 759.38 & 1427.51 & 349.6 \end{bmatrix}$$

weights2 * output layer1

$$\begin{bmatrix} 937.112 & 968.536 & 1307.912 \\ 1164.32 & 1203.361 & 1625.292 \\ 1509.162 & 1559.882 & 2106.606 \\ 1509.162 & 1559.882 & 2106.606 \\ 800.558 & 827.349 & 1117.848 \\ 8071.88 & 8343.322 & 11269.026 \end{bmatrix}$$

np.dot + bias

bias=[1.8, 2.2, 2.5]

$$\begin{bmatrix} 937.112 & 968.536 & 1307.912 \\ 1164.32 & 1203.361 & 1625.292 \\ 1509.162 & 1559.882 & 2106.606 \\ 1509.162 & 1559.882 & 2106.606 \\ 800.558 & 827.349 & 1117.848 \\ 8071.88 & 8343.322 & 11269.026 \end{bmatrix}$$

+ [1.8, 2.2, 2.5]

hasil

$$\begin{bmatrix} 938.912 & 970.736 & 1310.412 \\ 1166.12 & 1205.561 & 1627.792 \\ 1510.962 & 1562.082 & 2109.106 \\ 1510.962 & 1562.082 & 2109.106 \\ 802.358 & 829.549 & 1120.348 \\ 8073.68 & 8345.522 & 11271.526 \end{bmatrix}$$