

**Ujian Tengah Semester Kecerdasan Buatan**  
**Single Neuron, Multiple Neuron, Multiple Neuron Batch**  
**Multiple Neuron Batch and Layers**



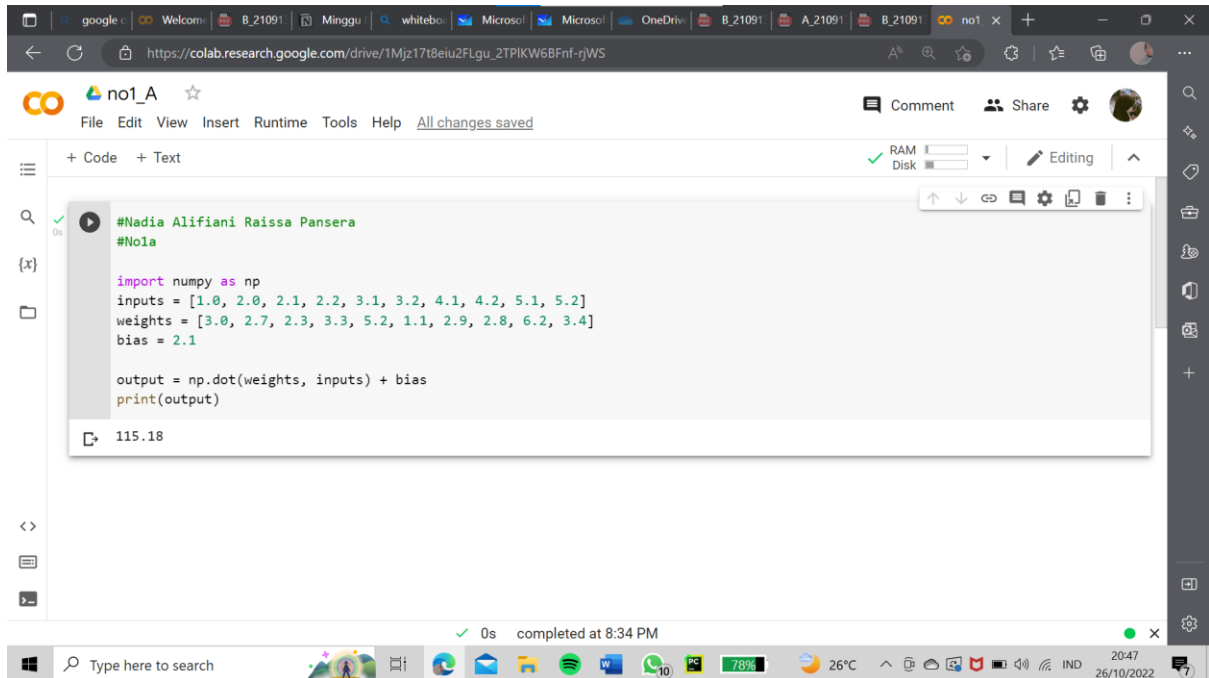
Disusun oleh :

Nadia Alifiani Raissa Pansera (21091397014)

**Program Studi D4 Manajemen Informatika**  
**Fakultas Vokasi**  
**Universitas Negeri Surabaya**  
**2022**

# UTS 1

## a. Source Code:



```
#Nadia Alifiani Raissa Pansera
#No1a

import numpy as np
inputs = [1.0, 2.0, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2]
weights = [3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4]
bias = 2.1

output = np.dot(weights, inputs) + bias
print(output)
```

115.18

Output :

115.18

- Inisialisasi numpy sebagai method perhitungan.
- Memasukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan di soal.
- Inputs =  $10 \times 1$
- Weights =  $1 \times 10$
- Neuron = 1
- Bias = 1
- Buat output untuk menghitung variabel yang telah kita masukkan/buat np.dot = untuk penghitungan vektor weight dan input kemudian hasil penghitungan vektor ditambahkan dengan bias.
- Buatlah command print untuk menampilkan hasil perhitungan output.

Perhitungan dot product :

Weights 10\*1

[3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4]

inputs  
10\*1

\*

$\begin{bmatrix} 1.0 \\ 2.0 \\ 2.1 \\ 2.2 \\ 3.1 \\ 3.2 \\ 4.1 \\ 4.2 \\ 5.1 \\ 5.2 \end{bmatrix}$

weights\*inputs = 113.08000000000001

kemudian np.dot + bias

113.08000000000001

+

2.1

= 115.18

b. source code :

```
import numpy as np
inputs = [1.0, 2.0, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2]
weights = [[3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4],
           [7.4, 3.3, 4.3, 6.7, 5.3, 7.9, 4.7, 8.0, 9.0, 6.0],
           [3.1, 3.2, 3.4, 3.5, 2.4, 5.4, 3.9, 8.8, 4.3, 7.0],
           [9.0, 5.8, 7.8, 9.4, 6.6, -4.5, 7.3, 2.1, 5.2, 5.5],
           [8.8, -5.3, 6.9, 5.1, 7.8, 6.9, 5.7, 8.0, 5.0, 5.1]]
biases = [6.4, 3.2, 5.5, 7.8, 9.0]
output = np.dot(weights, inputs) + biases
print(output)

[119.48 212.65 165.84 165.39 188.16]
```

Output :

```
[119.48 212.65 165.84 165.39 188.16]
```

Analysis:

- Inisialisasi numpy sebagai method perhitungan
- Memasukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan.  
Inputs = 10    Weights = 5×10    Neuron = 5    Biases = 5
- Buat output untuk menghitung variabel yang telah kita masukkan/buat  
np.dot = untuk penghitungan vektor weight dan input kemudian hasil penghitungan vektor ditambahkan dengan biases
- Buat command print untuk menampilkan hasil perhitungan output.

## Perhitungan dot product

weights  
10\*5

$$\begin{bmatrix} 3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4 \\ 7.4, 3.3, 4.3, 6.7, 5.3, 7.9, 4.7, 8.0, 9.0, 6.0 \\ 3.1, 3.2, 3.4, 3.5, 2.4, 5.4, 3.9, 8.8, 4.3, 7.0 \\ 9.0, 5.8, 7.8, 9.4, 6.6, -4.5, 7.3, 2.1, 5.2, 5.5 \\ 8.8, -5.3, 6.9, 5.1, 7.8, 6.9, 5.7, 8.0, 5.0, 5.1 \end{bmatrix}$$

inputs  
10\*1

\*

[1.0, 2.0, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2]

weights \* inputs = [113.08 209.45 160.34 157.59 179.16]

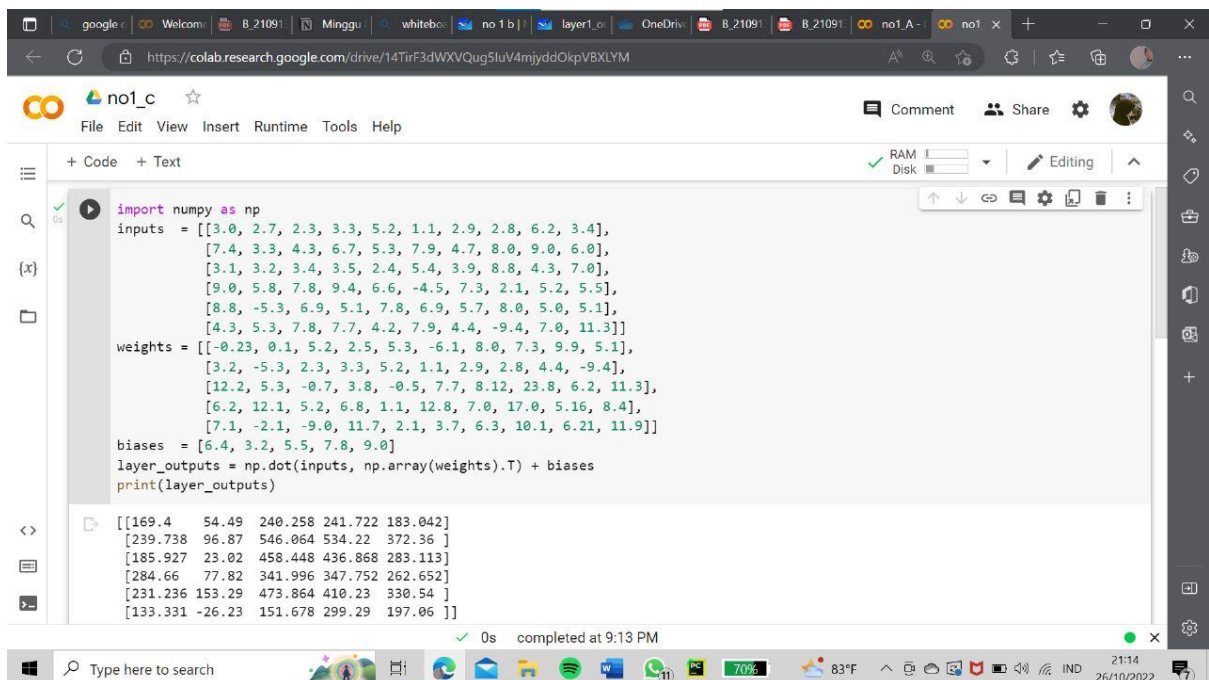
lalu np.dot + biases

[113.08 209.45 160.34 157.59 179.16]

[6.4, 3.2, 5.5, 7.8, 9.0]

= [119.48 212.65 165.84 165.39 188.16]

c. source code :



```
import numpy as np
inputs = [[3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4],
          [7.4, 3.3, 4.3, 6.7, 5.3, 7.9, 4.7, 8.0, 9.0, 6.0],
          [3.1, 3.2, 3.4, 3.5, 2.4, 5.4, 3.9, 8.8, 4.3, 7.0],
          [9.0, 5.8, 7.8, 9.4, 6.6, -4.5, 7.3, 2.1, 5.2, 5.5],
          [8.8, -5.3, 6.9, 5.1, 7.8, 6.9, 5.7, 8.0, 5.0, 5.1],
          [4.3, 5.3, 7.8, 7.7, 4.2, 7.9, 4.4, -9.4, 7.0, 11.3]]
weights = [[-0.23, 0.1, 5.2, 2.5, 5.3, -6.1, 8.0, 7.3, 9.9, 5.1],
           [3.2, -5.3, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 4.4, -9.4],
           [12.2, 5.3, -0.7, 3.8, -0.5, 7.7, 8.12, 23.8, 6.2, 11.3],
           [6.2, 12.1, 5.2, 6.8, 1.1, 12.8, 7.0, 17.0, 5.16, 8.4],
           [7.1, -2.1, -9.0, 11.7, 2.1, 3.7, 6.3, 10.1, 6.21, 11.9]]
biases = [6.4, 3.2, 5.5, 7.8, 9.0]
layer_outputs = np.dot(inputs, np.array(weights).T) + biases
print(layer_outputs)
```

[[169.4 54.49 240.258 241.722 183.042]  
[239.738 96.87 546.064 534.22 372.36]  
[185.927 23.02 458.448 436.868 283.113]  
[284.66 77.82 341.996 347.752 262.652]  
[231.236 153.29 473.864 410.23 330.54]  
[133.331 -26.23 151.678 299.29 197.06]]

Output :

```
[[169.4 54.49 240.258 241.722 183.042]
 [239.738 96.87 546.064 534.22 372.36 ]
 [185.927 23.02 458.448 436.868 283.113]
 [284.66 77.82 341.996 347.752 262.652]
 [231.236 153.29 473.864 410.23 330.54 ]
 [133.331 -26.23 151.678 299.29 197.06 ]]
```

Analisis:

- Inisialisasi numpy sebagai method perhitungan.
- Memasukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan.

Inputs = 10 Batch = 6 Weights = 5\*10 Neuron = 5 Biases = 5

- Buat output untuk menghitung variabel yang telah kita masukkan/buat

np.dot = untuk penghitungan vektor weight dan input kemudian hasil penghitungan vektor ditambahkan dengan biases.

- Buat command print untuk menampilkan hasil perhitungan output.

## perhitungan dot product

$$\begin{array}{c} \text{weights} \\ 10 \times 5 \end{array} \begin{bmatrix} -0.23, 0.1, 5.2, 2.5, 5.3, -6.1, 8.0, 7.3, 9.9, 5.1 \\ 3.2, -5.3, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 4.4, -9.4 \\ 12.2, 5.3, -0.7, 3.8, -0.5, 7.7, 8.12, 23.8, 6.2, 11.3 \\ 6.2, 12.1, 5.2, 6.8, 1.1, 12.8, 7.0, 17.0, 5.16, 8.4 \\ 7.1, -2.1, -9.0, 11.7, 2.1, 3.7, 6.3, 10.1, 6.21, 11.9 \end{bmatrix} * \begin{array}{c} \text{inputs} \\ 6 \times 10 \end{array} \begin{bmatrix} 3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4 \\ 7.4, 3.3, 4.3, 6.7, 5.3, 7.9, 4.7, 8.0, 9.0, 6.0 \\ 3.1, 3.2, 3.4, 3.5, 2.4, 5.4, 3.9, 8.8, 4.3, 7.0 \\ 9.0, 5.8, 7.8, 9.4, 6.6, -4.5, 7.3, 2.1, 5.2, 5.5 \\ 8.8, -5.3, 6.9, 5.1, 7.8, 6.9, 5.7, 8.0, 5.0, 5.1 \\ 4.3, 5.3, 7.8, 7.7, 4.2, 7.9, 4.4, -9.4, 7.0, 11.3 \end{bmatrix}$$

weights \* inputs =

$$\begin{bmatrix} 163. & 51.29 & 234.758 & 233.922 & 174.042 \\ 233.338 & 93.67 & 540.564 & 526.42 & 363.36 \\ 179.527 & 19.82 & 452.948 & 429.068 & 274.113 \\ 278.26 & 74.62 & 336.496 & 339.952 & 253.652 \\ 224.836 & 150.09 & 468.364 & 402.43 & 321.54 \\ 126.931 & -29.43 & 146.178 & 291.49 & 188.06 \end{bmatrix}$$

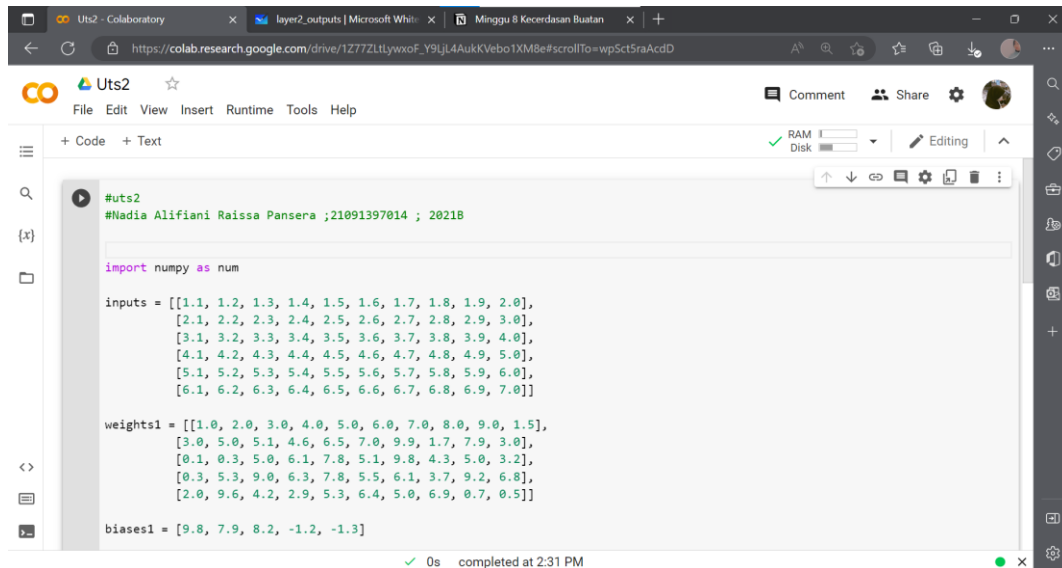
lalu np.dot + biases

$$\begin{bmatrix} 163. & 51.29 & 234.758 & 233.922 & 174.042 \\ 233.338 & 93.67 & 540.564 & 526.42 & 363.36 \\ 179.527 & 19.82 & 452.948 & 429.068 & 274.113 \\ 278.26 & 74.62 & 336.496 & 339.952 & 253.652 \\ 224.836 & 150.09 & 468.364 & 402.43 & 321.54 \\ 126.931 & -29.43 & 146.178 & 291.49 & 188.06 \end{bmatrix} + [6.4, 3.2, 5.5, 7.8, 9.0]$$

$$\begin{bmatrix} 169.4 & 54.49 & 240.258 & 241.722 & 183.042 \\ 239.738 & 96.87 & 546.064 & 534.22 & 372.36 \\ 185.927 & 23.02 & 458.448 & 436.868 & 283.113 \\ 284.66 & 77.82 & 341.996 & 347.752 & 262.652 \\ 231.236 & 153.29 & 473.864 & 410.23 & 330.54 \\ 133.331 & -26.23 & 151.678 & 299.29 & 197.06 \end{bmatrix}$$

# UTS 2

## 1. Source code :



The screenshot shows a Google Colab notebook titled 'Uts2'. The code cell contains the following Python code:

```
#uts2
#Nadia Alifiani Raissa Pansera ;21091397014 ; 2021B

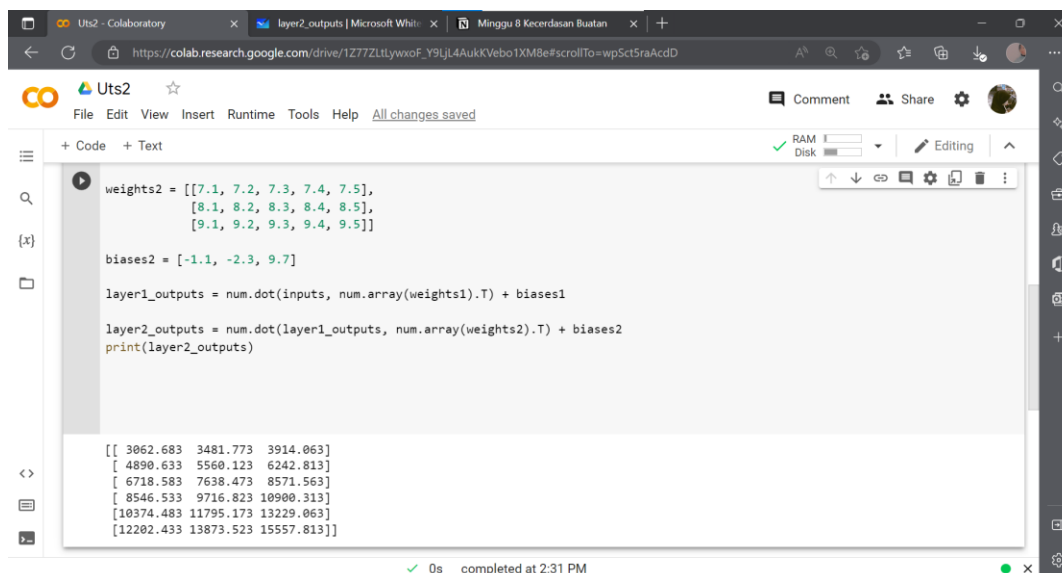
import numpy as num

inputs = [[1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0],
[2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0],
[3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0],
[4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5.0],
[5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0],
[6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 7.0]]

weights1 = [[1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 1.5],
[3.0, 5.0, 5.1, 4.6, 6.5, 7.0, 9.9, 1.7, 7.9, 3.0],
[0.1, 0.3, 5.0, 6.1, 7.8, 5.1, 9.8, 4.3, 5.0, 3.2],
[0.3, 5.3, 9.0, 6.3, 7.8, 5.5, 6.1, 3.7, 9.2, 6.8],
[2.0, 9.6, 4.2, 2.9, 5.3, 6.4, 5.0, 6.9, 0.7, 0.5]]

biases1 = [9.8, 7.9, 8.2, -1.2, -1.3]
```

The code is executed successfully, as indicated by the status bar showing '0s completed at 2:31 PM'.



The screenshot shows the same Google Colab notebook with additional code for calculating the outputs of two layers. The code cell contains the following Python code:

```
weights2 = [[7.1, 7.2, 7.3, 7.4, 7.5],
[8.1, 8.2, 8.3, 8.4, 8.5],
[9.1, 9.2, 9.3, 9.4, 9.5]]

biases2 = [-1.1, -2.3, 9.7]

layer1_outputs = num.dot(inputs, num.array(weights1).T) + biases1

layer2_outputs = num.dot(layer1_outputs, num.array(weights2).T) + biases2
print(layer2_outputs)
```

The output of the code is displayed in the output cell:

```
[[ 3062.683  3481.773  3914.063]
 [ 4890.633  5560.123  6242.813]
 [ 6718.583  7638.473  8571.563]
 [ 8546.533  9716.823 10900.313]
 [10374.483 11795.173 13229.063]
 [12202.433 13873.523 15557.813]]
```

The code is executed successfully, as indicated by the status bar showing '0s completed at 2:31 PM'.

## Output :

```
[[ 3062.683  3481.773  3914.063]
 [ 4890.633  5560.123  6242.813]
 [ 6718.583  7638.473  8571.563]
 [ 8546.533  9716.823 10900.313]
 [10374.483 11795.173 13229.063]
 [12202.433 13873.523 15557.813]]
```

## Output result:

```
[ 3062.683  3481.773  3914.063]
[ 4890.633  5560.123  6242.813]
[ 6718.583  7638.473  8571.563]
[ 8546.533  9716.823 10900.313]
[10374.483 11795.173 13229.063]
[12202.433 13873.523 15557.813]
```



## Analysis:

1. Inisialisasi numpy digunakan sebagai method perhitungan
2. Memasukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan soal  

$$\left. \begin{array}{l} \text{Inputs} = 10 \\ \text{Batch} = 6 \end{array} \right\} \text{Inputs menjadi matric } 6 \times 10$$

$$\begin{array}{ll} \text{Weights1} = 5 \times 10 & \text{Biases1} = 5 \\ \text{Weights2} = 3 \times 5 & \text{Biases2} = 3 \end{array}$$
3. Buatlah output untuk menghitung variabel yang sudah kita masukkan/buat  
 $\text{num.dot}$  = untuk menghitung vektor weight dan input  
kemudian hasil dari hitungan vektor ditambah dengan biases
4. Untuk output yang diinginkan adalah output layer2 yang berasal dari hasil perhitungan layer1 lalu dihitung lagi dengan weights2 dan biases2
5. Buat command print untuk menampilkan hasil perhitungan output

## Perhitungan Layer 1:

perhitungan dot product

$$\begin{array}{c} \text{weights} \\ 10 \times 5 \end{array} \begin{bmatrix} 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 1.5 \\ 3.0, 5.0, 5.1, 4.6, 6.5, 7.0, 9.9, 1.7, 7.9, 3.0 \\ 0.1, 0.3, 5.0, 6.1, 7.8, 5.1, 9.8, 4.3, 5.0, 3.2 \\ 0.3, 5.3, 9.0, 6.3, 7.8, 5.5, 6.1, 3.7, 9.2, 6.8 \\ 2.0, 9.6, 4.2, 2.9, 5.3, 6.4, 5.0, 6.9, 0.7, 0.5 \end{bmatrix} * \begin{array}{c} \text{inputs} \\ 6 \times 10 \end{array} \begin{bmatrix} 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0 \\ 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0 \\ 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0 \\ 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5.0 \\ 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0 \\ 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 7.0 \end{bmatrix}$$

$$\text{weights} * \text{inputs} \begin{bmatrix} 76.5 & 84.22 & 75.67 & 95.82 & 64.68 \\ 123. & 137.92 & 122.37 & 155.82 & 108.18 \\ 169.5 & 191.62 & 169.07 & 215.82 & 151.68 \\ 216. & 245.32 & 215.77 & 275.82 & 195.18 \\ 262.5 & 299.02 & 262.47 & 335.82 & 238.68 \\ 309. & 352.72 & 309.17 & 395.82 & 282.18 \end{bmatrix}$$

lalu  $\text{num.dot} + \text{biases}$

$$\begin{bmatrix} 76.5 & 84.22 & 75.67 & 95.82 & 64.68 \\ 123. & 137.92 & 122.37 & 155.82 & 108.18 \\ 169.5 & 191.62 & 169.07 & 215.82 & 151.68 \\ 216. & 245.32 & 215.77 & 275.82 & 195.18 \\ 262.5 & 299.02 & 262.47 & 335.82 & 238.68 \\ 309. & 352.72 & 309.17 & 395.82 & 282.18 \end{bmatrix} + \begin{bmatrix} 9.8 & 7.9 & 8.2 & -1.2 & -1.3 \end{bmatrix}$$

$$= \begin{bmatrix} 86.3 & 92.12 & 83.87 & 94.62 & 63.38 \\ 132.8 & 145.82 & 130.57 & 154.62 & 106.88 \\ 179.3 & 199.52 & 177.27 & 214.62 & 150.38 \\ 225.8 & 253.22 & 223.97 & 274.62 & 193.88 \\ 272.3 & 306.92 & 270.67 & 334.62 & 237.38 \\ 318.8 & 360.62 & 317.37 & 394.62 & 280.88 \end{bmatrix}$$

Perhitungan layer 2 :

### perhitungan dot product

$$\begin{array}{c} \text{weights2} \\ 5 \times 3 \end{array} \begin{bmatrix} 7.1, 7.2, 7.3, 7.4, 7.5 \\ 8.1, 8.2, 8.3, 8.4, 8.5 \\ 9.1, 9.2, 9.3, 9.4, 9.5 \end{bmatrix} * \begin{array}{c} \text{output layer1} \\ 5 \times 6 \end{array} \begin{bmatrix} 86.3 & 92.12 & 83.87 & 94.62 & 63.38 & \\ 132.8 & 145.82 & 130.57 & 154.62 & 106.88 & \\ 179.3 & 199.52 & 177.27 & 214.62 & 150.38 & \\ 225.8 & 253.22 & 223.97 & 274.62 & 193.88 & \\ 272.3 & 306.92 & 270.67 & 334.62 & 237.38 & \\ 318.8 & 360.62 & 317.37 & 394.62 & 280.88 & \end{bmatrix}$$

$$\text{weights2} * \text{output layer1} = \begin{bmatrix} 3063.783 & 3484.073 & 3904.363 \\ 4891.733 & 5562.423 & 6233.113 \\ 6719.683 & 7640.773 & 8561.863 \\ 8547.633 & 9719.123 & 10890.613 \\ 10375.583 & 11797.473 & 13219.363 \\ 12203.533 & 13875.823 & 15548.113 \end{bmatrix}$$

lalu num.dot + biases2

$$\begin{bmatrix} 3063.783 & 3484.073 & 3904.363 \\ 4891.733 & 5562.423 & 6233.113 \\ 6719.683 & 7640.773 & 8561.863 \\ 8547.633 & 9719.123 & 10890.613 \\ 10375.583 & 11797.473 & 13219.363 \\ 12203.533 & 13875.823 & 15548.113 \end{bmatrix} + [-1.1, -2.3, 9.7]$$

$$= \begin{bmatrix} 3062.683 & 3481.773 & 3914.063 \\ 4890.633 & 5560.123 & 6242.813 \\ 6718.583 & 7638.473 & 8571.563 \\ 8546.533 & 9716.823 & 10900.313 \\ 10374.483 & 11795.173 & 13229.063 \\ 12202.433 & 13873.523 & 15557.813 \end{bmatrix}$$