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

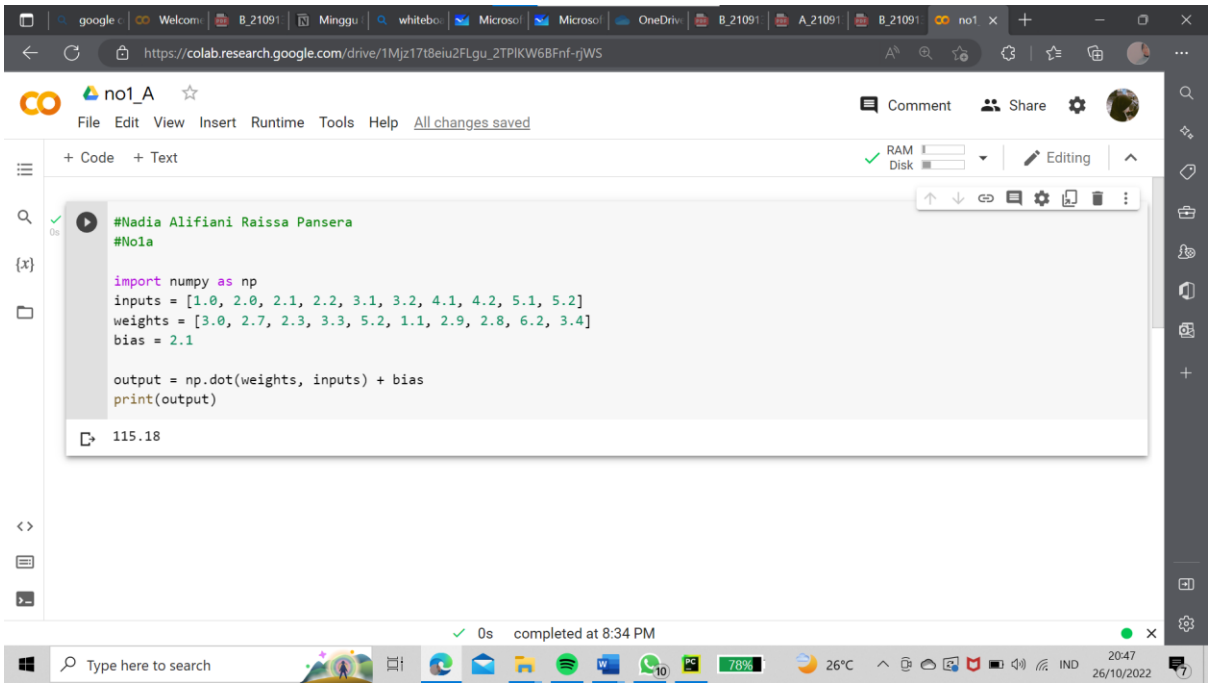
Fakultas Vokasi

Universitas Negeri Surabaya

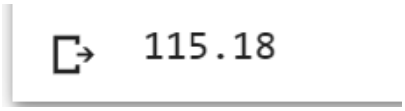
2022

UTS 1

a. Source Code:



Output :



- Inisialisasi numpy sebagai method perhitungan.

-Memasukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan di soal.

Inputs = 10 × 1

Weights = 1 × 10

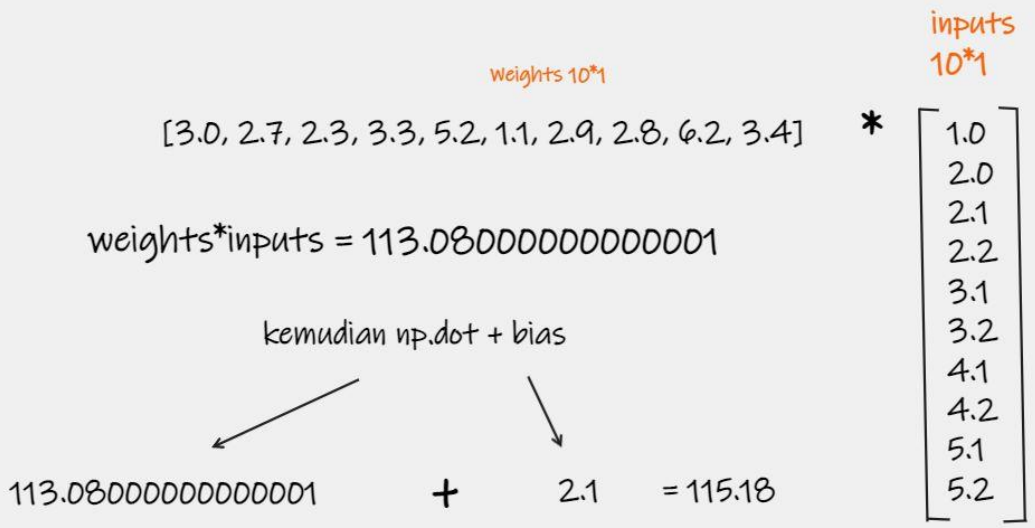
Neuron = 1

Bias = 1

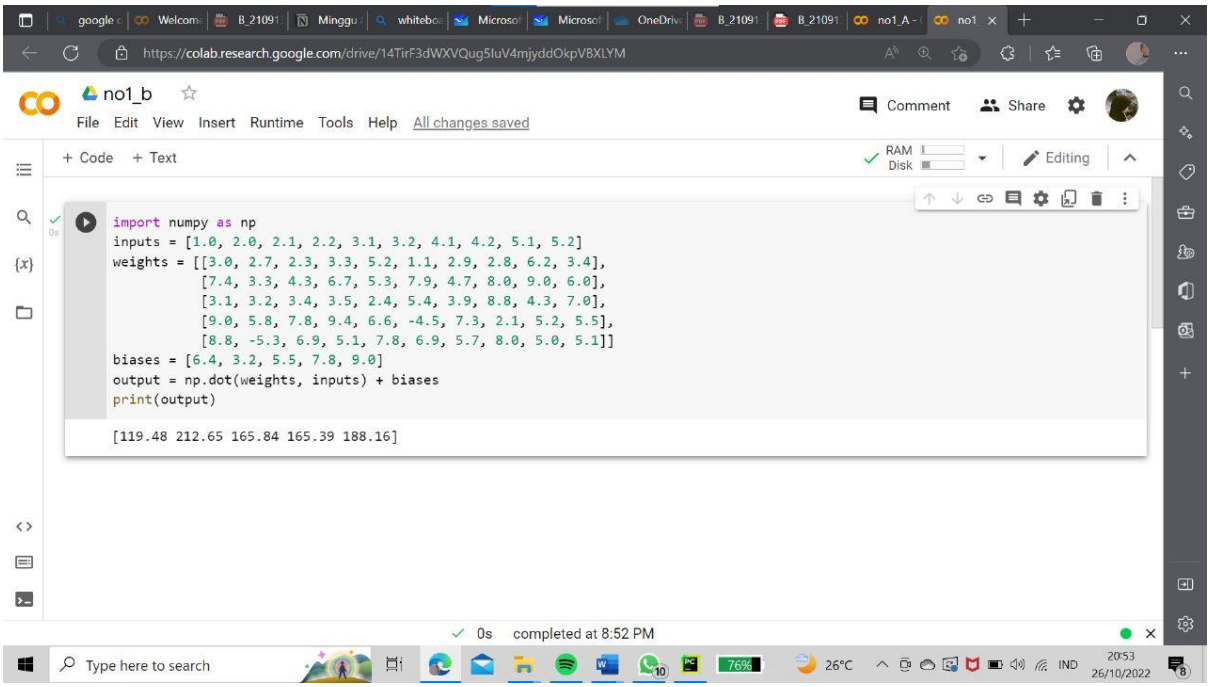
-Buat output untuk menghitung variabel yang telah kita masukkan/buat np.dot = untuk penghitungan vektor weight dan input kemudian hasil penghitungan vektor ditambahkan dengan bias.

-Buatlah command print untuk menampilkan hasil perhitungan output.

Perhitungan dot product :



b. source code :



Output :

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

Analisis:

- 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

$$\begin{bmatrix} 3.0, 2.7, 2.3, 3.3, 5.2, 1.1, 2.9, 2.8, 6.2, 3.4 \\ 7.4, 3.3, 4.3, 6.7, 5.3, 7.9, 4.7, 8.0, 9.0, 6.0 \\ 3.1, 3.2, 3.4, 3.5, 2.4, 5.4, 3.9, 8.8, 4.3, 7.0 \\ 9.0, 5.8, 7.8, 9.4, 6.6, -4.5, 7.3, 2.1, 5.2, 5.5 \\ 8.8, -5.3, 6.9, 5.1, 7.8, 6.9, 5.7, 8.0, 5.0, 5.1 \end{bmatrix} * \begin{matrix} \text{inputs} \\ 10*1 \end{matrix} \quad [1.0, 2.0, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2]$$

weights * inputs = [113.08 209.45 160.34 157.59 179.16]

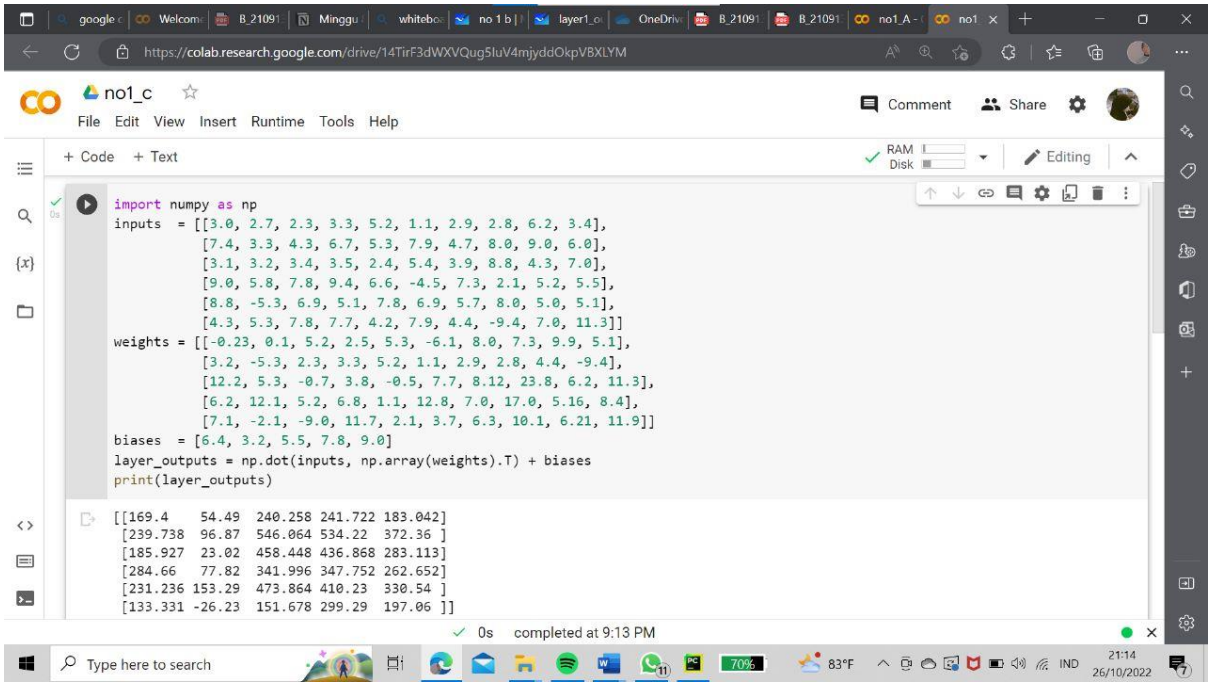
lalu np.dot + biases

[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 :



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

weights
10*5

```
[-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]
```

*

inputs
6*10

```
[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 * inputs =

```
[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]
```

lalu np.dot + biases

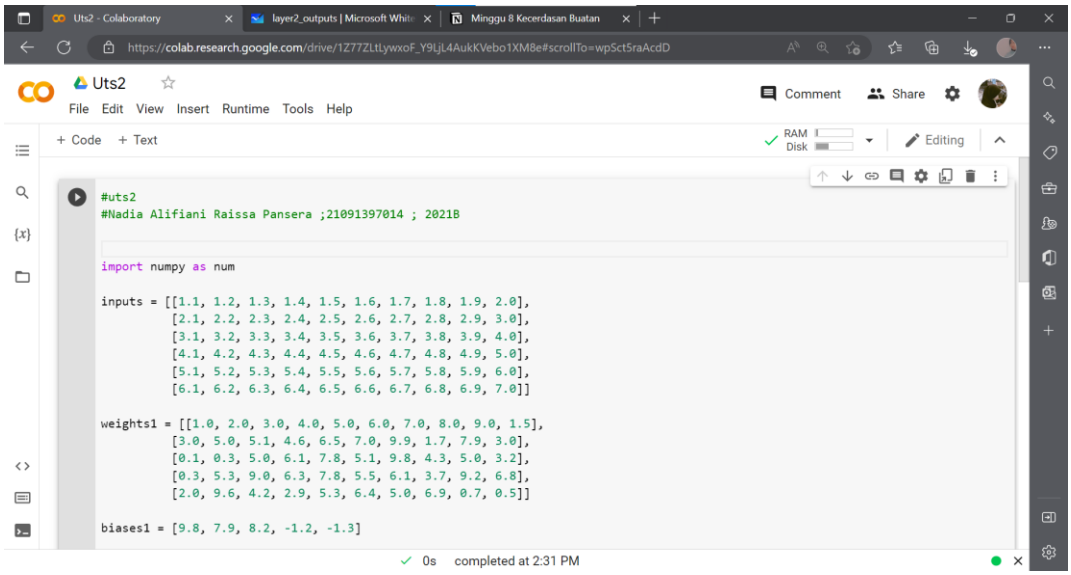
```
[163.    51.29  234.758 233.922 174.042
 233.338  93.67  540.564 526.42  363.36
 179.527  19.82  452.948 429.068 274.113
 278.26   74.62  336.496 339.952 253.652
 224.836 150.09  468.364 402.43  321.54
 126.931 -29.43  146.178 291.49  188.06]
```

+ [6.4, 3.2, 5.5, 7.8, 9.0]

```
[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]
```

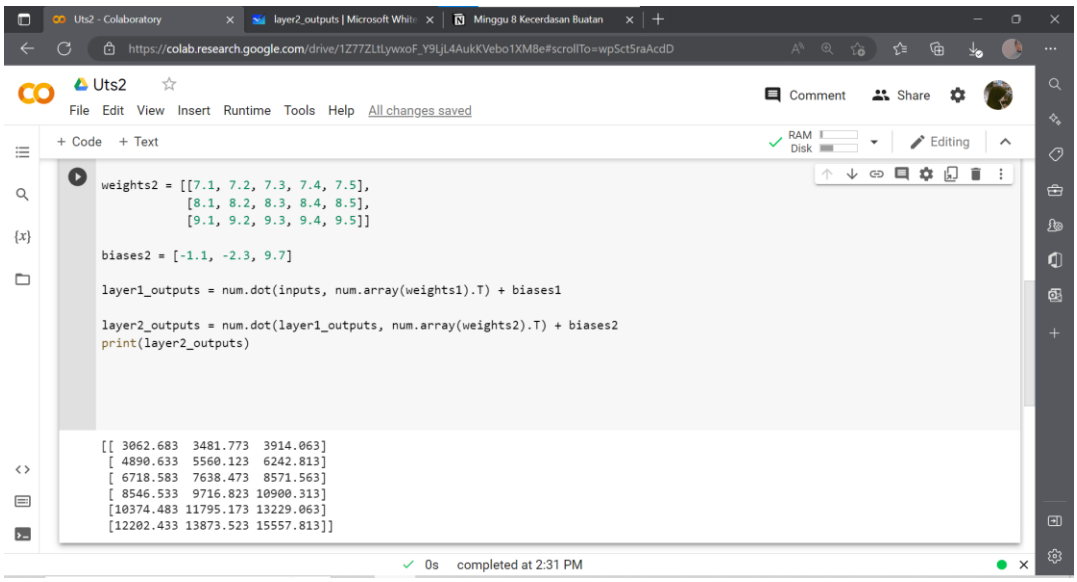
UTS 2

1. Source 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]
```

0s completed at 2:31 PM



```
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)
```

```
[[ 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]]
```

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]]
```

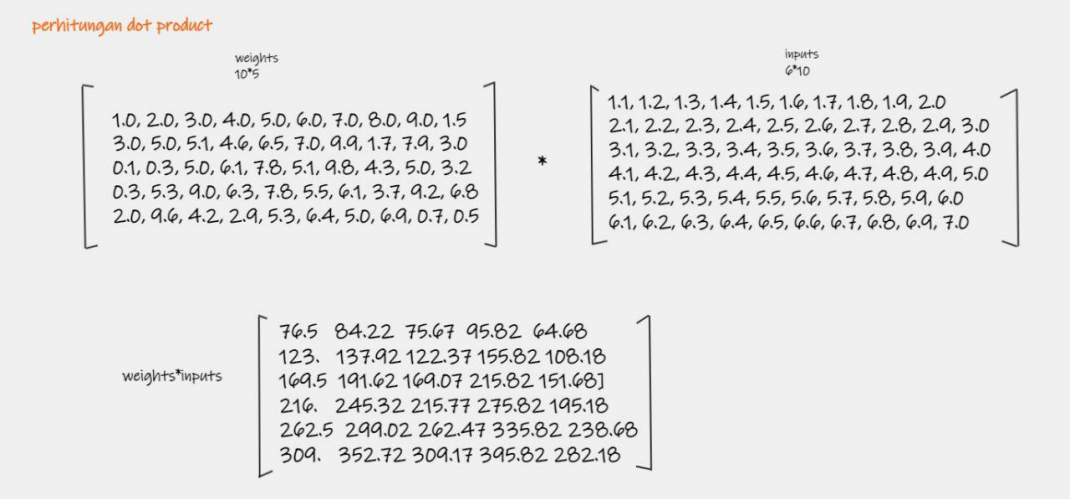
Output result:

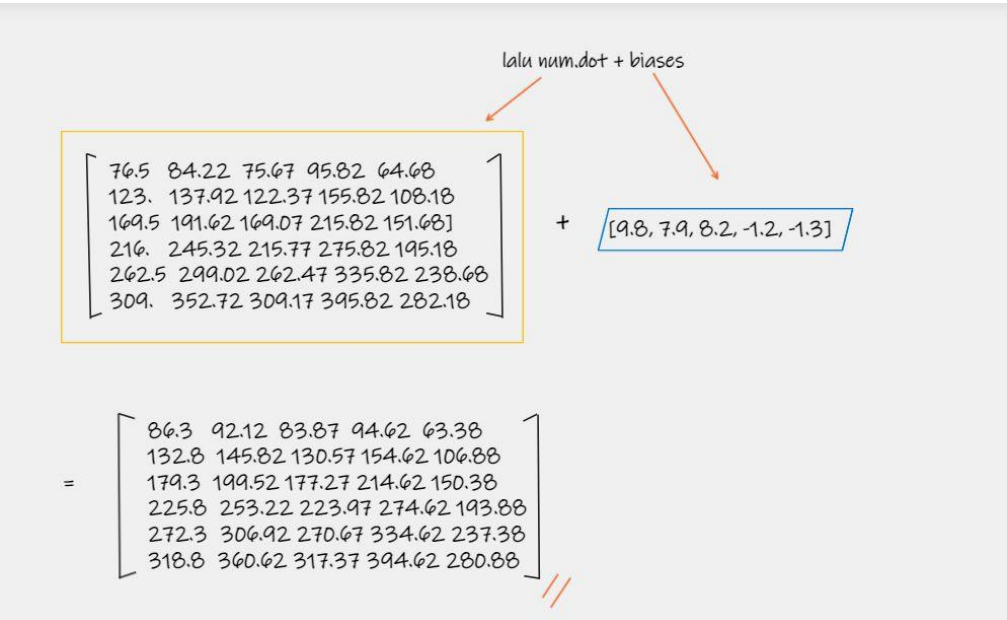
```
[ 3062.683  3481.773  3914.063]
[ 4890.633  5560.123  6242.813]
[ 6718.583  7638.473  8571.563]
[ 8546.533  9716.823 10900.313]
[10374.483 11795.173 13229.063]
[12202.433 13873.523 15557.813]
```

Analisis:

1. Inisialisasi numpy digunakan sebagai method perhitungan
2. Memasukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan soal
   Inputs = 10
   Batch = 6 } Inputs menjadi matric 6×10
   Weights1 = 5×10 Biases1 = 5
   Weights2 = 3×5 Biases2 = 3
3. Buatlah output untuk menghitung variabel yang sudah kita masukkan/buat
   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 layer 2 :