

**Ujian Tengah Semester**  
**Mata Kuliah Kecerdasan Buatan**



**Oleh :**

**Nama** : Aini Azzah  
**NIM** : 21091397006  
**Kelas** : 2021 B

**PRODI MANAJEMEN INFORMATIKA FAKULTAS VOKASI**  
**UNIVERSITAS NEGERI SURABAYA**  
**2022**

## 1. Buat Kodingan

### a. Single Neuron

#### Source Code

```
SingleNeuron.py X
SingleNeuron.py > ...
1  #Program Single Neuron
2
3  # Aini Azzah - 21091397006
4
5  #Inisialisasi numpy
6  import numpy as np
7
8  #Inisialisasi variabel
9  #Input layer feature = 10
10 input = [2.0, 1.5, 2.2, 2.5, 3.0, 1.0, 3.1, 1.2, 5.0, 4.5]
11
12 #Jumlah Neuron = 1
13 weight = [-0.1, 0.5, 1.5, -0.2, 0.2, -0.5, 1.0, 1.2, -1.5, 2.0]
14 bias = 0.5
15
16 #Output
17 output = np.dot(weight, input) + bias
18
19 #Print outputnya
20 print(output)
21
```

Program dimulai dengan inisialisasi numpy ke np pada baris ke-6. Kemudian membuat variable input (untuk menginput nilai pada neuron) pada baris ke-10, weight (bobot dari neuron tersebut) pada baris ke-13, serta bias (untuk mengimbangi output secara positif atau negatif) pada baris ke-14. Serta mengisi setiap variable dengan nilai yang telah diinputkan, yaitu 10 value untuk input layers, dan Weight serta bias = 1, karena single neuron. Pada baris ke-17 mendefinisikan output untuk menghitung dot.product (untuk mengkalikan nilai input dan weight) serta dijumlahkan dengan nilai biasnya. Dan pada baris ke-20 terdapat fungsi print untuk memanggil hasil dari perhitungan pada output.

#### Output

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  Python + - [ ] [X] ^ X
● PS C:\Users\Aini Azzah\My_Project\.vscode> & "C:/Users/Aini Azzah/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Aini Azzah/My_Project/.vscode/SingleNeuron.py"
9.99
○ PS C:\Users\Aini Azzah\My_Project\.vscode>
```

Penjelasan perhitungan single neuron :

Dengan array input yang memiliki 10 layer feature begitu juga array weight.

Kedua array tersebut dihitung pada perhitungan dengan rumus (yang telah disederhanakan menggunakan modul numpy) yang terdapat pada source code baris ke 17. Yakni :

$$\begin{aligned} \text{Output} = & (\text{input}[0] * \text{weight}[0]) + (\text{input}[1] * \text{weight}[1]) + (\text{input}[2] * \text{weight}[2]) + \\ & (\text{input}[3] * \text{weight}[3]) + (\text{input}[4] * \text{weight}[4]) + (\text{input}[5] * \text{weight}[5]) + \\ & (\text{input}[6] * \text{weight}[6]) + (\text{input}[7] * \text{weight}[7]) + (\text{input}[8] * \text{weight}[8]) + \\ & (\text{input}[9] * \text{weight}[9]) + \text{bias} \end{aligned}$$

Jadi perhitungan sesuai dengan source code diatas yaitu :

$$\begin{aligned} \text{Output} = & (2 * -0.1) + (1.5 * 0.5) + (2.2 * 1.5) + \\ & (2.5 * -0.2) + (3 * 0.2) + (1 * -0.5) + \\ & (3.1 * 1) + (1.2 * 1.2) + (5 * -1.5) + \\ & (4.5 * 2) + 0.5 \end{aligned}$$

$$\begin{aligned} \text{Output} = & (-0.2) + (0.75) + (3.3) + \\ & (-0.5) + (0.6) + (-0.5) + \\ & (3.1) + (1.44) + (-7.5) + \\ & (9) + 0.5 \end{aligned}$$

$$\begin{aligned} \text{Output} = & (3.85) + (-0.4) + (-2.96) + (9) + 0.5 \\ = & 9.99 \end{aligned}$$

Dengan input, weight, dan bias sedemikian seperti diatas dan dihitung menggunakan rumus tersebut, sehingga mendapatkan hasil pada output yakni 9.99

## b. Multi Neuron

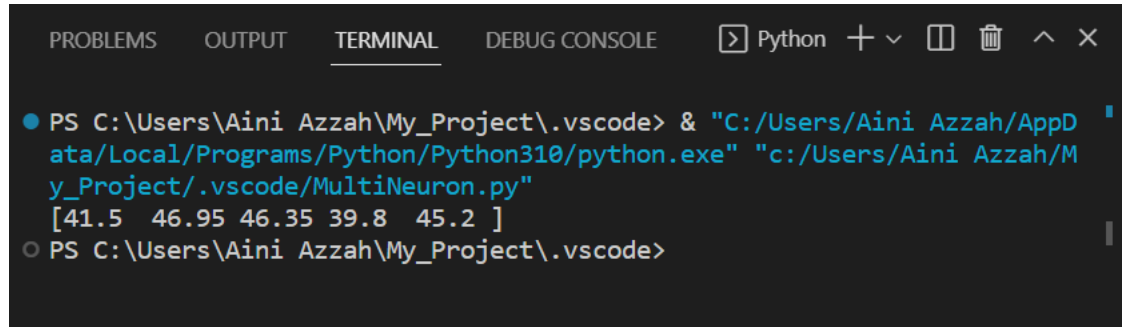
### Source Code

```
MultiNeuron.py X
MultiNeuron.py > ...
1  #Program Multli Neuron
2
3  # Aini Azzah - 21091397006
4
5  #Inisialisasi numpy
6  import numpy as np
7
8  #Inisialisasi variabel
9  #Input layer feature = 10
10 input = [0.5, 1.5, 4.0, 0.2, 3.5, 5.0, 2.5, 3.0, 2.0, 1.0]
11
12 #Jumlah Neuron = 5
13 weight = [[0.2, 0.8, -0.5, 1.0, 2.0, 4.0, 5.0, 2.5, -2.5, -1.0],
14           [2.0, -1.5, -1.0, 1.0, -0.5, 5.0, 4.5, 2.5, 3.0, 1.5],
15           [4.2, -0.5, 2.1, 3.0, 5.0, 1.0, 2.0, 3.5, -1.0, -2.0],
16           [-1.5, 1.0, -2.4, -3.5, 3.0, 3.5, 1.1, 1.2, 4.5, 5.5],
17           [2.2, -1.2, 0.5, 1.0, 0.2, 3.2, 4.4, 3.0, 5.0, -6.0]]
18
19 bias = [1.0, 2.5, 2.0, 0.5, 3.0]
20
21 #Output
22 layer_output = np.dot(weight, input) + bias
23
24 #Print output
25 print(layer_output)
26
```

Seperti pada single neuron, Multi Neuron diawali dengan inisialisasi numpy pada baris ke-6. Selanjutnya inisialisasi variable input (baris 10), weight (baris 13) dan bias (baris

19). Dengan input layer = 10 , weight dan bias = 5 (Neuron = 5). Kemudian pada baris ke-22 melakukan perhitungan dot.product (untuk mengkalikan weight dan input) pada layer output serta dijumlahkan dengan biasnya. Selanjutnya pada baris ke-25 memanggil hasil perhitungan pada output layer dengan perintah print.

## Output



```
PS C:\Users\Aini Azzah\My_Project\.vscode> & "C:/Users/Aini Azzah/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Aini Azzah/My_Project/.vscode/MultiNeuron.py"
[41.5 46.95 46.35 39.8 45.2 ]
PS C:\Users\Aini Azzah\My_Project\.vscode>
```

Penjelasan perhitungan Multi Neuron :

Masih seperti single neuron yang memiliki input layer feature = 10, begitu pula dengan weight-nya. Tetapi multi neuron tersebut memiliki 5 neuron, sehingga pada weight memiliki 5 array, begitu pula dengan jumlah bias, yakni = 5.

Dengan begitu maka perhitungan diatas menggunakan rumus yang terdapat pada source code diatas pada baris ke 22. Rincian rumus sebagai berikut.

- *Input \* weight 1*  
Output = (input[0] \* weight1[0]) + (input[1] \* weight1[1]) +  
(input[2] \* weight1[2]) + (input[3] \* weight1[3]) +  
(input[4] \* weight1[4]) + (input[5] \* weight1[5]) +  
(input[6] \* weight1[6]) + (input[7] \* weight1[7]) +  
(input[8] \* weight1[8]) + (input[9] \* weight1[9]) + bias1
- *Input \* weight 2*  
Output = (input[0] \* weight2[0]) + (input[1] \* weight2[1]) +  
(input[2] \* weight2[2]) + (input[3] \* weight2[3]) +  
(input[4] \* weight2[4]) + (input[5] \* weight2[5]) +  
(input[6] \* weight2[6]) + (input[7] \* weight2[7]) +  
(input[8] \* weight2[8]) + (input[9] \* weight2[9]) + bias2
- *Input \* weight 3*  
Output = (input[0] \* weight3[0]) + (input[1] \* weight3[1]) +  
(input[2] \* weight3[2]) + (input[3] \* weight3[3]) +  
(input[4] \* weight3[4]) + (input[5] \* weight3[5]) +  
(input[6] \* weight3[6]) + (input[7] \* weight3[7]) +  
(input[8] \* weight3[8]) + (input[9] \* weight3[9]) + bias3
- *Input \* weight 4*  
Output = (input[0] \* weight4[0]) + (input[1] \* weight4[1]) +  
(input[2] \* weight4[2]) + (input[3] \* weight4[3]) +

$$\begin{aligned}
 &(\text{input}[4] * \text{weight4}[4]) + (\text{input}[5] * \text{weight4}[5]) + \\
 &(\text{input}[6] * \text{weight4}[6]) + (\text{input}[7] * \text{weight4}[7]) + \\
 &(\text{input}[8] * \text{weight4}[8]) + (\text{input}[9] * \text{weight4}[9]) + \text{bias4}
 \end{aligned}$$

- *Input \* weight 5*

$$\begin{aligned}
 \text{Output} = &(\text{input}[0] * \text{weight5}[0]) + (\text{input}[1] * \text{weight5}[1]) + \\
 &(\text{input}[2] * \text{weight5}[2]) + (\text{input}[3] * \text{weight5}[3]) + \\
 &(\text{input}[4] * \text{weight5}[4]) + (\text{input}[5] * \text{weight5}[5]) + \\
 &(\text{input}[6] * \text{weight5}[6]) + (\text{input}[7] * \text{weight5}[7]) + \\
 &(\text{input}[8] * \text{weight5}[8]) + (\text{input}[9] * \text{weight5}[9]) + \text{bias5}
 \end{aligned}$$

Dengan rincian rumus diatas, maka perhitungan angka-angka dalam source code adalah :

- *Input \* weight 1*

$$\begin{aligned}
 \text{Output} = &(0.5 * 0.2) + (1.5 * 0.8) + (4 * -0.5) + (0.2 * 1) + (3.5 * 2) + (5 * 4) + \\
 &(2.5 * 5) + (3 * 2.5) + (2 * -2.5) + (1 * -1) + 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Output} = &(0.1) + (1.2) + (-2) + (0.2) + (7) + (20) + (12.5) + (7.5) + (-5) + (-1) + 1 \\
 = &41.5
 \end{aligned}$$

- *Input \* weight 2*

$$\begin{aligned}
 \text{Output} = &(0.5 * 2) + (1.5 * -1.5) + (4 * -1) + (0.2 * 1) + (3.5 * -0.5) + (5 * 5) + \\
 &(2.5 * 4.5) + (3 * 2.5) + (2 * 3) + (1 * 1.5) + 2.5
 \end{aligned}$$

$$\begin{aligned}
 \text{Output} = &(1) + (-2.25) + (-4) + (0.2) + (-1.75) + (25) + (11.25) + (7.5) + (6) + \\
 &(1.5) + 2.5 \\
 = &46.95
 \end{aligned}$$

- *Input \* weight 3*

$$\begin{aligned}
 \text{Output} = &(0.5 * 4.2) + (1.5 * -0.5) + (4 * 2.1) + (0.2 * 3) + (3.5 * 5) + (5 * 1) + \\
 &(2.5 * 2) + (3 * 3.5) + (2 * -1) + (1 * -2) + 2
 \end{aligned}$$

$$\begin{aligned}
 \text{Output} = &(2.1) + (-0.75) + (8.4) + (0.6) + (17.5) + (5) + (5) + (10.5) + (-2) + \\
 &(-2) + 2 \\
 = &46.35
 \end{aligned}$$

- *Input \* weight 4*

$$\begin{aligned}
 \text{Output} = &(0.5 * -1.5) + (1.5 * 1) + (4 * -2.4) + (0.2 * -3.5) + (3.5 * 3) + (5 * 3.5) + \\
 &(2.5 * 1.1) + (3 * 1.2) + (2 * 4.5) + (1 * 5.5) + 0.5
 \end{aligned}$$

$$\begin{aligned}
 \text{Output} = &(-0.75) + (1.5) + (-9.6) + (-0.7) + (10.5) + (17.5) + (2.75) + (3.6) + (9) + \\
 &(5.5) + 0.5 \\
 = &39.8
 \end{aligned}$$

- *Input \* weight 5*

$$\text{Output} = (0.5 * 2.2) + (1.5 * -1.2) + (4 * 0.5) + (0.2 * 1) + (3.5 * 0.2) + (5 * 3.2) + (2.5 * 4.4) + (3 * 3) + (2 * 5) + (1 * -6) + 3$$

$$\text{Output} = (1.1) + (-1.8) + (2) + (0.2) + (0.7) + (16) + (11) + (9) + (10) + (-6) + 3 = 45.2$$

Output layer merupakan hasil gabungan dari semua perhitungan di atas dalam satu array sebagai berikut.

Ouput layer = [41.5 46.95 46.35 39.8 45.2]

### c. Multi Neuron Batch Input

#### Source Code

```
MultiNeuronBatchInput.py X
MultiNeuronBatchInput.py > ...
1  #Program Multi Neuron Batch Input
2  # Aini Azzah - 21091397006
3
4  #Inisialisasi numpy
5  import numpy as np
6
7  #Inisialisasi variabel
8  #Input layer feature 10
9  #Per batch input = 6
10 inputs = [[4.0, 3.5, 5.0, 2.5, 1.0, 2.2, 2.0, 1.5, 2.0, 3.0],
11           [4.1, 1.0, 2.1, 5.0, 2.0, 3.0, 0.2, 2.7, 1.1, 1.0],
12           [3.0, 1.5, 4.1, 2.2, 0.8, 0.2, 3.5, 2.5, 1.5, 2.0],
13           [2.3, 4.2, 5.5, 3.0, 1.5, 2.5, 0.5, 0.9, 3.7, 1.0],
14           [0.5, 4.5, 3.0, 0.2, 2.5, 2.5, 1.5, 3.5, 5.0, 1.8],
15           [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5]]
16
17 #Jumlah Neuron = 5
18 weights = [[-2.1, 3.0, 4.2, 5.1, -1.0, 0.4, -0.6, 2.5, -0.5, 1.7],
19           [4.0, 4.4, -0.2, 1.3, 4.8, 3.1, 0.5, -2.5, 1.0, 5.0],
20           [-0.2, -0.5, 1.7, 0.8, 0.5, 1.0, -1.0, 2.0, 2.0, 2.1],
21           [1.5, -0.1, 0.2, -0.5, 5.0, 3.2, -1.6, 4.1, 2.2, -3.5],
22           [3.0, 4.2, -1.5, 1.5, 1.0, -2.0, 2.5, 2.4, -0.5, -1.0]]
23 biases = [0.5, 1.0, 2.5, 3.0, 1.5]
24
25 #Output
26 layer_outputs = np.dot(inputs, np.array(weights).T) + biases
27
28 #Print output
29 print(layer_outputs)
```

Pada baris ke-5 terlebih dahulu inisialisasi numpy (untuk nantinya digunakan pada perhitungan dot product). Kemudian inisialisasi variable inputs (baris 10), weights (baris 18), dan biases (baris 23). Dengan input layers = 10, per batch input = 6, weights dan biases = 5 (Neuron = 5). Kemudian pada baris ke-26 melakukan perhitungan menggunakan dot product (mengkalikan setiap value inputs dengan weights yang telah ditranspose (T)) dan dijumlahkan dengan setiap nilai biasnya. Sehingga akan menampilkan hasil perhitungan dengan array pada baris ke-29 dengan memanggil fungsi print output layers.

## Output

```
PROBLEMS TERMINAL ... Python + - [ ] [ ] ^ x
● PS C:\Users\Aini Azzah\My_Project\.vscode> & "C:/Users/Aini Azzah/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Aini Azzah/My_Project/.vscode/MultiNeuronBatchInput.py"
[[42.88 60.52 24.45 17.29 25.65]
 [36.19 46.23 22.25 36.24 23.78]
 [33.22 33.1 19.18 12.66 26.35]
 [47.97 53.13 25.74 28.66 19.35]
 [33.48 48.21 28.44 40.95 23.05]
 [38.15 62.95 38.5 33.5 20.35]]
○ PS C:\Users\Aini Azzah\My_Project\.vscode>
```

Penjelasan perhitungan dari Multi Neuron Batch Input :

Pada program multi neuron batch input ini memiliki masih memiliki feature layer = 10, dengan per batch input = 6, artinya memiliki banyak array input sama dengan 6. Dan pada neuron = 5 sehingga jumlah array dari weight sama dengan 5, dan dengan bias yang juga = 5.

Pada rumus perhitungan multi neuron batch input ini menggunakan perhitungan transpose pada rumus. Yakni pada weight yang ditranspose terlebih dahulu, sebelum dikalikan dengan input layer. Dengan gambaran sesuai angka pada source code diatas, sebagai berikut.

### INPUT

4.0	3.5	5.0	2.5	1.0	2.2	2.0	1.5	2.0	3.0
4.1	1.0	2.1	5.0	2.0	3.0	0.2	2.7	1.1	1.0
3.0	1.5	4.1	2.2	0.8	0.2	3.5	2.5	1.5	2.0
2.3	4.2	5.5	3.0	1.5	2.5	0.5	0.9	3.7	1.0
0.5	4.5	3.0	0.2	2.5	2.5	1.5	3.5	5.0	1.8
1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5

(6, 10)

### WEIGHT

-2.1	3.0	4.2	5.1	-1.0	0.4	-0.6	2.5	-0.5	1.7
4.0	4.4	-0.2	1.3	4.8	3.1	0.5	-2.5	1.0	5.0
-0.2	-0.5	1.7	0.8	0.5	1.0	-1.0	2.0	2.0	2.1
1.5	-0.1	0.2	-0.5	5.0	3.2	-1.6	4.1	2.2	-3.5
3.0	4.2	-1.5	1.5	1.0	-2.0	2.5	2.4	-0.5	-1.0

(5, 10)

Dengan matriks input dan weight diatas, pada perhitungan selanjutnya yaitu weight terlebih dahulu di transpose, sehingga menjadi

## INPUT

4.0	3.5	5.0	2.5	1.0	2.2	2.0	1.5	2.0	3.0
4.1	1.0	2.1	5.0	2.0	3.0	0.2	2.7	1.1	1.0
3.0	1.5	4.1	2.2	0.8	0.2	3.5	2.5	1.5	2.0
2.3	4.2	5.5	3.0	1.5	2.5	0.5	0.9	3.7	1.0
0.5	4.5	3.0	0.2	2.5	2.5	1.5	3.5	5.0	1.8
1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5

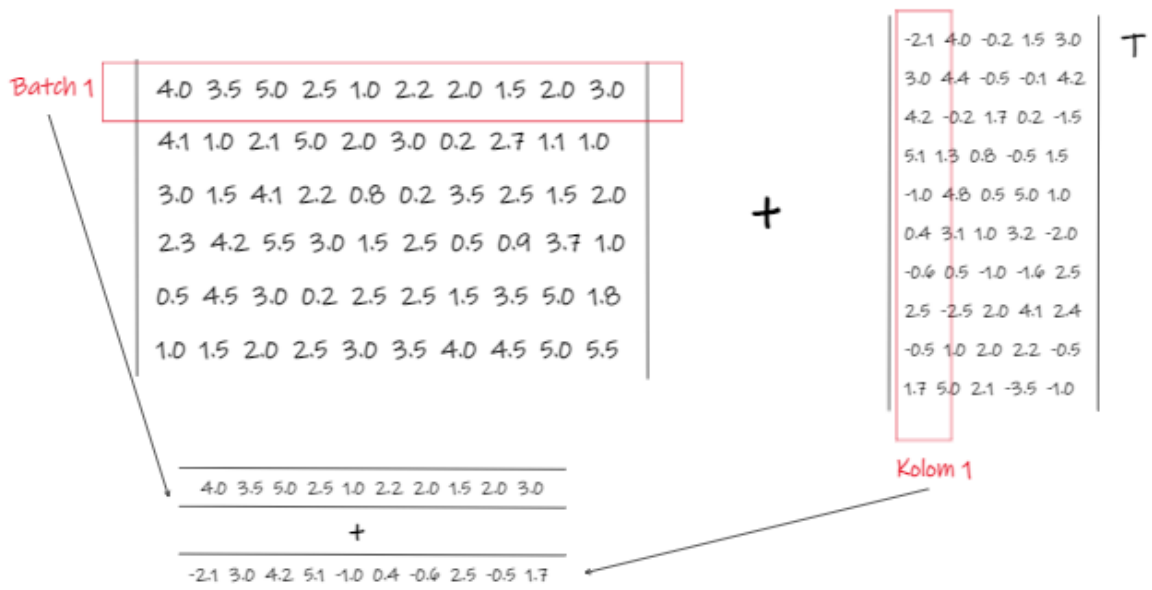
(6, 10)

## WEIGHT

-2.1	4.0	-0.2	1.5	3.0
3.0	4.4	-0.5	-0.1	4.2
4.2	-0.2	1.7	0.2	-1.5
5.1	1.3	0.8	-0.5	1.5
-1.0	4.8	0.5	5.0	1.0
0.4	3.1	1.0	3.2	-2.0
-0.6	0.5	-1.0	-1.6	2.5
2.5	-2.5	2.0	4.1	2.4
-0.5	1.0	2.0	2.2	-0.5
1.7	5.0	2.1	-3.5	-1.0

(10, 5)

Kemudian tiap 1 baris batch input di dikalikan dengan setiap kolom pada weight<sup>T</sup>, dan dilakukan berulang-ulang hingga batch ke 6. Dengan perhitungan seperti berikut.



$$\begin{aligned}
 &= (4.0 \cdot -2.1) + (3.5 \cdot 3.0) + (5.0 \cdot 4.2) + (2.5 \cdot 5.1) + (1.0 \cdot -1.0) + (2.2 \cdot 0.4) + (2.0 \cdot -0.6) + (1.5 \cdot 2.5) + (2.0 \cdot -0.5) + (3.0 \cdot 1.7) \\
 &= (-8.4) + (10.5) + (21) + (12.75) + (-1) + (0.88) + (-1.2) + (3.75) + (-1) + (5.1) \\
 &= 41.38
 \end{aligned}$$

41.38	59.52	21.95	14.29	24.15
35.69	45.23	19.75	33.24	22.28
32.72	32.1	16.68	9.66	24.85
47.47	52.13	19.75	25.66	17.85
32.98	47.21	25.94	37.95	21.55
37.65	61.95	36	30.5	18.85

Bias

+

0.5	1.0	2.5	3.0	1.5
-----	-----	-----	-----	-----



Setelah penjumlahan setiap batch input dengan weight, hasil dari layer output dijumlahkan dengan bias dari setiap neuron, dengan neuron = 5. Sehingga menjadi output layer seperti berikut.

**Layer output =**

42.88	60.52	24.45	17.29	25.65
36.19	46.23	22.25	36.24	23.78
33.22	33.1	19.18	12.66	26.35
47.97	53.13	25.74	28.66	19.35
33.48	48.21	28.44	40.95	23.05
38.15	62.95	38.5	33.5	20.35

## UTS 2

### d. Multi Neuron Batch Input 2 Layer

#### Source code

```
MultiNeuronBatchInput2.py X
MultiNeuronBatchInput2.py > ...
1 # Aini Azzah - 21091397006 - 2021B
2 # Program Multi Neuron Batch Input 2 layer
3
4 # Inisialisasi numpy
5 # Input layer feature 10
6 # Per batch input = 6
7 import numpy
8 inputs = [[4.0, 3.5, 5.0, 2.5, 1.0, 2.2, 2.0, 1.5, 2.0, 3.0],
9           [4.1, 1.0, 2.1, 5.0, 2.0, 3.0, 0.2, 2.7, 1.1, 1.0],
10          [3.0, 1.5, 4.1, 2.2, 0.8, 0.2, 3.5, 2.5, 1.5, 2.0],
11          [2.3, 4.2, 5.5, 3.0, 1.5, 2.5, 0.5, 0.9, 3.7, 1.0],
12          [0.5, 4.5, 3.0, 0.2, 2.5, 2.5, 1.5, 3.5, 5.0, 1.8],
13          [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5]]
14
15 # Layer 1, Neuron = 5
16 weights1 = [[-2.1, 3.0, 4.2, 5.1, -1.0, 0.4, -0.6, 2.5, -0.5, 1.7],
17             [4.0, 4.4, -0.2, 1.3, 4.8, 3.1, 0.5, -2.5, 1.0, 5.0],
18             [-0.2, -0.5, 1.7, 0.8, 0.5, 1.0, -1.0, 2.0, 2.0, 2.1],
19             [1.5, -0.1, 0.2, -0.5, 5.0, 3.2, -1.6, 4.1, 2.2, -3.5],
20             [3.0, 4.2, -1.5, 1.5, 1.0, -2.0, 2.5, 2.4, -0.5, -1.0]]
21 biases1 = [0.5, 1.0, 2.5, 3.0, 1.5]
```

```

22
23 # Layer 2, Neuron = 3
24 weights2 = [[0.1, 1.4, 1.5, 2.0, 1.0],
25             [1.5, 1.2, 2.3, 3.1, 2.4],
26             [2.4, 3.0, 2.0, 1.1, 1.7]]
27 biases2 = [1, 2, 0.5]
28
29 # Output layer 1
30 layer1_outputs = numpy.dot(inputs, numpy.array(weights1).T) + biases1
31
32 # Output layer 2
33 layer2_outputs = numpy.dot(layer1_outputs, numpy.array(weights2).T) + biases2
34
35 # Print Output layer 2
36 print(layer2_outputs)

```

Pada baris ke-7 terlebih dahulu inialisasi numpy (untuk nantinya digunakan pada perhitungan dot product).

Baris ke-8 inialisasi variable inputs, dengan input layers feature = 10, dan per batch input = 6.

Baris ke-16 inialisasi variable weight, untuk layer 1, dengan layer feature = 10, dan Neuron = 5, sehingga jumlah array = 5.

Baris ke-21 inialisasi variable biases 1, yakni bias dari layer 1 dan weight 1 dengan bias dari neuron = 5.

Baris ke-24 inialisasi variable weight, untuk layer 2, dengan layer feature berdasarkan hasil output dari perhitungan input dengan layer 1, dan jumlah array berdasarkan jumlah neuron dari layer 2, yakni neuron = 3.

Baris ke-27 inialisasi variable biases 2, yakni bias dari layer 2 dan weight 2 dengan bias dari neuron = 3.

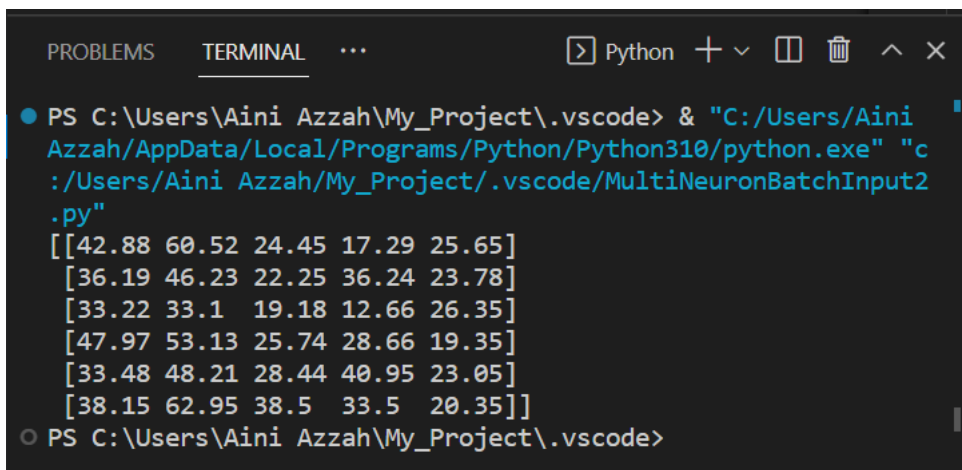
Baris ke-30 menghitung output dari perhitungan input dengan layer 1, dengan rumus menggunakan numpy, menghitung input, weight, dan biasnya.

Baris ke-33 menghitung output dari layer 2, yakni hasil dari perhitungan dengan rumus numpy dari hasil output dari layer 1 dan dijumlahkan dengan layer 2 dan weight layer 2 beserta biasnya.

Baris ke-36 menampilkan hasil dari perhitungan output layer 2.

## Output

- Output layer 1



```

PS C:\Users\Aini Azzah\My_Project\.vscode> & "C:/Users/Aini Azzah/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Aini Azzah/My_Project/.vscode/MultiNeuronBatchInput2.py"
[[42.88 60.52 24.45 17.29 25.65]
 [36.19 46.23 22.25 36.24 23.78]
 [33.22 33.1 19.18 12.66 26.35]
 [47.97 53.13 25.74 28.66 19.35]
 [33.48 48.21 28.44 40.95 23.05]
 [38.15 62.95 38.5 33.5 20.35]]
PS C:\Users\Aini Azzah\My_Project\.vscode>

```

## Penjelasan Perhitungan Multi Neuron Batch Input 2 layer, output layer 1 :

Pada program Multi Neuron Batch Input 2 layer ini memiliki 2 hidden layer, yang masing-masing memiliki 5 neuron untuk layer pertama dan 3 neuron untuk layer ke 2. Pada perhitungan output layer 1, yakni menghitung input dengan dijumlahkan dengan weight layer 1. Yaitu dengan detail input layer feature =10, per batch = 6, dan weight layer 1 memiliki neuron =5, sehingga memiliki 5 baris array pada weightnya, dan juga biasanya. Dengan detail penjelasan sebagai berikut.

INPUT

4.0	3.5	5.0	2.5	1.0	2.2	2.0	1.5	2.0	3.0
4.1	1.0	2.1	5.0	2.0	3.0	0.2	2.7	1.1	1.0
3.0	1.5	4.1	2.2	0.8	0.2	3.5	2.5	1.5	2.0
2.3	4.2	5.5	3.0	1.5	2.5	0.5	0.9	3.7	1.0
0.5	4.5	3.0	0.2	2.5	2.5	1.5	3.5	5.0	1.8
1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5

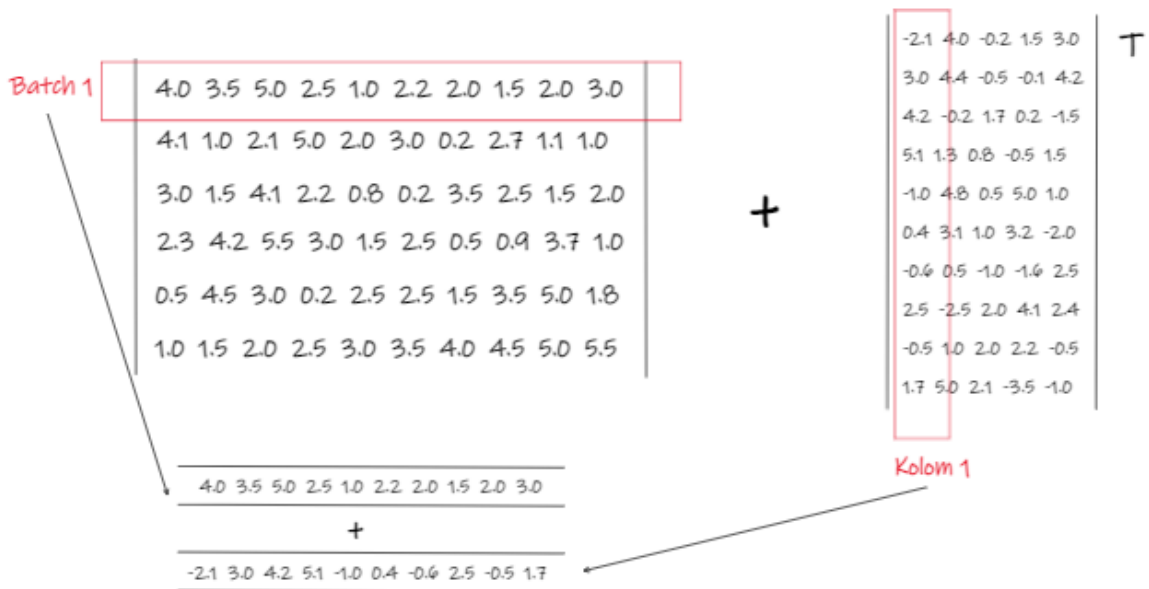
(6 , 10)

WEIGHT 1

-2.1	3.0	4.2	5.1	-1.0	0.4	-0.6	2.5	-0.5	1.7
4.0	4.4	-0.2	1.3	4.8	3.1	0.5	-2.5	1.0	5.0
-0.2	-0.5	1.7	0.8	0.5	1.0	-1.0	2.0	2.0	2.1
1.5	-0.1	0.2	-0.5	5.0	3.2	-1.6	4.1	2.2	-3.5
3.0	4.2	-1.5	1.5	1.0	-2.0	2.5	2.4	-0.5	-1.0

(5 , 10)

Sebelum dijumlahkan, weight layer 1 ditranspose terlebih dahulu, untuk menyamakan jumlah input layer fitur = 10, dengan jumlah pada weightnya sehingga menjadi (10, 5)



Setelah ditranspose baru dijumlahkan, yaitu dengan menjumlahkan per batch input dengan setiap kolom pada weight layer 1. Perhitungan berulang hingga batch input ke 6 selesai dijumlahkan dengan semua kolom pada weight 1. Sebagai berikut

$$\begin{aligned}
&= (4.0 * -2.1) + (3.5 * 3.0) + (5.0 * 4.2) + (2.5 * 5.1) + (1.0 * -1.0) + (2.2 * 0.4) + (2.0 * -0.6) + (1.5 * 2.5) + (2.0 * -0.5) + (3.0 * 1.7) \\
&= (-8.4) + (10.5) + (21) + (12.75) + (-1) + (0.88) + (-1.2) + (3.75) + (-1) + (-1) + (5.1) \\
&= 41.38
\end{aligned}$$

41.38	59.52	21.95	14.29	24.15
35.69	45.23	19.75	33.24	22.28
32.72	32.1	16.68	9.66	24.85
47.47	52.13	19.75	25.66	17.85
32.98	47.21	25.94	37.95	21.55
37.65	61.95	36	30.5	18.85

$$+ \begin{array}{|c|c|c|c|c|} \hline 0.5 & 1.0 & 2.5 & 3.0 & 1.5 \\ \hline \end{array}$$

Setelah penjumlahan setiap batch input dengan weight, hasilnya dijumlahkan dengan bias dari setiap neuron, dengan neuron = 5. Sehingga menjadi output layer 1 seperti berikut.

**Layer 1 Output =**

42.88	60.52	24.45	17.29	25.65
36.19	46.23	22.25	36.24	23.78
33.22	33.1	19.18	12.66	26.35
47.97	53.13	25.74	28.66	19.35
33.48	48.21	28.44	40.95	23.05
38.15	62.95	38.5	33.5	20.35

- Output layer 2

```

PROBLEMS  TERMINAL  ...
Python + - [ ] [ ] ^ x

● PS C:\Users\Aini Azzah\My_Project\.vscode> & "C:/Users/Aini Azzah/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Aini Azzah/My_Project/.vscode/MultiNeuronBatchInput2.py"
[186.921 310.338 396.496]
[198.976 332.352 350.836]
[131.102 238.15 276.609]
[195.459 332.199 390.919]
[219.452 357.749 366.592]
[238.045 376.005 429.355]]
○ PS C:\Users\Aini Azzah\My_Project\.vscode>

```

Penjelasan Perhitungan Multi Neuron Batch Input 2 layer, output layer 2 :  
 Dengan menjumlahkan hasil dari output layer 1 dengan layer 2 yang memiliki  
 weight = 5 neuron serta biasnya.

OUTPUT LAYER 1

42.88	60.52	24.45	17.29	25.65
36.19	46.23	22.25	36.24	23.78
33.22	33.1	19.18	12.66	26.35
47.97	53.13	25.74	28.66	19.35
33.48	48.21	28.44	40.95	23.05
38.15	62.95	38.5	33.5	20.35

(6, 5)

WEIGHT 2

0.1	1.4	1.5	2.0	1.0
1.5	1.2	2.3	3.1	2.4
2.4	3.0	2.0	1.1	1.7

(3, 5)

Selanjutnya weight 2 ditranspose terlebih dahulu sebelum dijumlahkan dengan hasil dari  
 output layer 1, sebagai berikut

42.88	60.52	24.45	17.29	25.65		0.1	1.5	2.4	T
36.19	46.23	22.25	36.24	23.78		1.4	1.2	3.0	
33.22	33.1	19.18	12.66	26.35		1.5	2.3	2.0	
47.97	53.13	25.74	28.66	19.35		2.0	3.1	1.1	
33.48	48.21	28.44	40.95	23.05		1.0	2.4	1.7	
38.15	62.95	38.5	33.5	20.35					

$$\begin{aligned}
 &= (42.88 * 0.1) + (60.52 * 1.4) + (24.45 * 1.5) + (17.29 * 2.0) + (25.65 * 1.0) \\
 &= (4.288) + (84.728) + (36.675) + (34.58) + (25.65) \\
 &= 185.921
 \end{aligned}$$

185.921	308.338	395.996
197.976	330.352	350.336
130.102	236.15	276.109
194.459	330.199	390.419
218.452	355.749	366.092
237.045	374.005	428.855

+ | 1.0 2.0 0.5 |

Setelah menjumlahkan output layer 1 dengan setiap array neuron pada weight 2, kemudian hasilnya dijumlahkan dengan bias dari setiap neuron pada layer 2 dengan jumlah bias = jumlah neuron dan weight, yakni 3 seperti diatas. Dengan hasil penjumlahan sebagai berikut

$$\text{Layer 2 Output} = \begin{pmatrix} 186.921 & 310.338 & 396.496 \\ 198.976 & 332.352 & 350.836 \\ 131.102 & 238.15 & 276.609 \\ 195.459 & 332.199 & 390.919 \\ 219.452 & 357.749 & 366.592 \\ 238.045 & 376.005 & 429.355 \end{pmatrix}$$