

**LAPORAN UJIAN TENGAH SEMESTER  
MATA KULIAH KECERDASAN BUATAN**



Disusun Oleh :  
Aditya Putra Pratama  
21091397043

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA  
FAKULTAS VOKASI  
UNIVERSITAS NEGERI SURABAYA  
2022**

# UTS Part 1

## A. Single Neuron

- i. Input layer feature 10
- ii. Neuron 1
- iii. Coding :

```
1  #Aditya Putra Pratama/21091397043
2
3  #Single Neuron
4
5  #Inisialisasi Numpy
6  import numpy as np
7
8  #Inisialisasi Variabel
9  inputs = [1.0, 1.5, 2.0, 3.5, 3.0, 1.5, 4.0, 2.5, 3.0, 7.2]
10 weights = [1.2, 3.4, 4.0, 2.4, 1.0, 2.2, 5.0, 3.2, 2.0, 2.0]
11 bias = 3.5
12
13 #Output
14 outputs = np.dot(weights, inputs) + bias
15
16 #PrintOutput
17 print(outputs)
```

Output :

80.9

Analisis :

- Pada baris ke-6 terdapat inisialisasi numpy, digunakan untuk mempermudah perhitungan atau dapat disebut juga dengan metode perhitungan.
- Pada baris ke-9 dan baris ke-10 terdapat input layer dan weights masing-masing berjumlah 10.
- Pada baris ke-11 terdapat bias = 3.5.
- Pada baris ke-14 terdapat cara perhitungannya, yaitu : buat np.dot untuk menghitung variable weight dan input yang telah dibuat, lalu hasil dari perhitungan tersebut akan ditambahkan dengan biasnya.
- Pada baris ke-17 terdapat Print Output untuk mencetak hasil perhitungannya.

## B. Multi Neuron

- i. Input layer feature 10
- ii. Neuron 5
- iii. Coding :

```
1  #Aditya Putra Pratama/21091397043
2
3  #Multi Neuron
4
5  #Inisialisasi Numpy
6  import numpy as np
7
8  #Inisialisasi Variabel
9  inputs = [3.6, 7.3, 2.1, 5.6, 3.1, 1.0, 2.5, 1.9, 4.3, 6.0]
10 weights = [
11     [1.3, 5.3, 7.1, 5.7, 8.5, 3.2, 1.4, 5.0, 2.3, 4.0],
12     [2.4, 5.2, 4.4, 2.0, 4.6, 7.0, 5.5, 1.5, 9.0, 4.5],
13     [2.0, 1.8, 4.5, 2.0, 4.0, 2.5, 4.5, 3.0, 2.5, 8.5],
14     [2.0, 5.7, 3.0, 1.0, 2.0, 4.0, 2.0, 4.5, 1.5, 6.5],
15     [4.3, 5.0, 4.2, 2.6, 3.7, 4.8, 3.0, 4.0, 3.0, 6.0],
16 ]
17 biases = [3.0, 1.0, 2.5, 4.5, 1.5]
18
19 #Output
20 outputs = np.dot(weights, inputs) + biases
21
22 #PrintOutput
23 print(outputs)
```

**Output :**

```
[169.64 171.6 137.09 134.41 157.13]
```

**Analisis :**

- Pada baris ke-6 terdapat inisialisasi numpy untuk mempermudah perhitungan atau dapat disebut juga dengan metode perhitungan.
- Pada baris ke-9 terdapat input layer berjumlah 10.
- Pada baris ke-10 sampai baris ke-16 terdapat Weights 5 x 10 (Matriks). Kenapa? Karena panjang weights terdiri dari jumlah inputnya = 10, sedangkan jumlah weights terdiri dari jumlah neuronnya 5.
- Pada baris ke-17 terdapat Bias berjumlah 5.
- Pada baris ke-20 terdapat cara perhitungannya, yaitu : buat np.dot untuk

menghitung variable weight dan input yang telah dibuat, lalu hasil dari perhitungan tersebut akan ditambahkan dengan biasnya.

- Pada baris ke-23 terdapat Print Output untuk mencetak hasil perhitungannya.

### C. Multi Neuron Batch Input

- Input layer Feature 10
- Per batch nya 6 input
- Neuron 5
- Coding :

```
1  #Aditya Putra Pratama/21091397043
2
3  #Multi Neuron Batch Input
4
5  #Inisialisasi Numpy
6  import numpy as np
7
8  #Inisialisasi Variabel
9  inputs = [
10     [1.3, 4.5, 2.0, 6.1, 2.4, 3.8, 1.2, 5.2, 4.3, 2.5],
11     [3.6, 2.5, 2.1, 1.0, 5.3, 7.2, 1.5, 2.5, 3.0, 5.0],
12     [2.5, 9.2, 23.0, 11.0, 23.5, 19.5, 30.2, 10.5, 43.5, 50.0],
13     [3.8, 5.6, 4.3, 5.5, 1.5, 7.8, 6.1, 2.0, 6.0, 5.7],
14     [4.4, 8.3, 2.7, 6.3, 8.8, 9.1, 3.5, 5.6, 12.0, 9.5],
15     [10.0, 13.1, 19.4, 11.0, 20.0, 14.1, 32.2, 21.3, 11.1, 13.0],
16 ]
17 weights = [
18     [3.1, 3.1, 3.2, 4.1, 1.0, 5.4, 1.0, 3.0, 5.1, 7.0],
19     [3.4, 5.7, 8.3, 1.0, 2.3, 5.5, 6.7, 3.2, 4.9, 4.0],
20     [7.6, 5.9, 4.3, 2.1, 1.0, 5.6, 7.5, 5.9, 1.9, 4.0],
21     [2.1, 3.9, 4.1, 9.8, 5.7, 7.1, 8.0, 9.0, 12.4, 19.1],
22     [2.4, 24.3, 11.5, 13.4, 23.4, 41.0, 34.1, 32.5, 45.0, 41.2]
23 ]
24 biases = [1.0, 4.5, 3.0, 4.0, 5.2]
25
26 #Output
27 outputs = np.dot(inputs, np.array(weights) . T) + biases
28
29 #PrintOutput
30 print(outputs)
```

Output :

```
[[ 129.54  139.44  142.37  290.39  940.79]
 [ 134.21  153.96  153.56  288.25 1004.76]
 [ 918.32 1077.73  902.08 2350.13 7384.63]
 [ 192.67  236.35  231.89  419.35 1406.3 ]
 [ 280.78  303.94  290.1   641.84 2144.69]
 [ 519.64  799.03  802.03 1312.78 4589.5 ]]
```

#### Analisis :

- Pada baris ke-6 terdapat inisialisasi numpy untuk mempermudah perhitungan atau dapat disebut juga dengan metode perhitungan.
- Pada baris ke-9 s a m p a i b a r i s k e - 1 6 terdapat input layer 10 dan per batchnya adalah 6 jadi 6x10 (Matriks).
- Pada baris ke-17 sampai baris ke-23 terdapat panjang Weights terdiri dari jumlah inputnya = 10, sedangkan jumlah weights terdiri dari jumlah neuronnya 5. Jadi 5x10 (Matriks).
- Pada baris ke-24 terdapat bias berjumlah 5.
- Pada baris ke-27 terdapat cara perhitungannya, yaitu : buat np.dot untuk menghitung variable weight dan input yang telah dibuat, lalu hasil dari perhitungan tersebut akan ditambahkan dengan biasnya.
- Pada baris ke-30 terdapat Print Output untuk mencetak hasil perhitungannya.

## UTS Part 2

### 1. Multi Neuron Batch Input

1. Input layer feature 10
2. Per batch nya 6 input
3. Hidden layer 1, 5 neuron
4. Hidden layer 2, 3 neuron

#### Source Code :

```
1  # Multi Neuron Batch Input
2
3  #Inisialisasi numpy
4  import numpy as np
5
6  # Inisialisasi Variabel Input
7  # 6 batch Inputs setiap batch berisi 10
8  inputs =[
9      # Inputs 1
10     [4.2,-5.0,4.0,-1.22,3.5,6.5,4.9,2.13,5.8,-3.55],
11     # Inputs 2
12     [2.5,3.3,-2.25,4.2,3.0,5.25,4.1,3.2,7.0,-1.11],
13     # Inputs 3
14     [1.1,-3.05,3.2,9.0,1.25,3.75,-1.23,0.15,-4.23,1.15],
15     # Inputs 4
16     [5.25,-5.25,1.15,-1.25,8.9,10.3,1.88,3.4,-0.17,0.21],
17     # Inputs 5
18     [3.0,0.49,0.33,0.34,-0.12,0.46,0.93,-0.28,3.44,1.92],
19     # Inputs 6
20     [2.34,-1.14,2.27,8.09,6.61,7.24,1.91,-2.22,4.78,0.75]
21 ]
22
```

```
23 # Inisialisasi Variabel Weights 1
24 weights_1 =[
25     # Neuron 1
26     [1.25,-4.1,-2.3,-9.19,4.45,1.67,2.39,-7.14,0.99,0.3],
27     # Neuron 2
28     [-0.19,9.81,9.99,1.34,2.55,-2.82,1.56,-0.85,2.09,7.04],
29     # Neuron 3
30     [1.12,7.56,-2.25,5.67,1.04,9.15,3.14,9.91,-1.55,2.45],
31     # Neuron 4
32     [-1.2,4.44,5.35,-1.17,3.55,9.02,-1.45,2.49,-1.11,9.02],
33     # Neuron 5
34     [-6.78,2.18,-4.44,8.06,-1.01,2.09,1.23,-4.04,7.34,1.50]
35 ]
36
37 # Inisialisasi bias 1
38 # Jumlah bias pada layer 1 berisi 5
39 biases_1 = [1.54,2.25,2.15,5.91,1.23]
40
```

```

41 # Inisialisasi Variabel Weights 2
42 # Jumlah neuron sesuai dengan jumlah bias pada layer ke 2, yaitu 3
43 # Di setiap neuron sesuai dengan jumlah bias pada layer 1, yaitu 5
44 Weights_2 = [
45     # Neuron 1
46     [2.05, 3.45, 2.51, 9.12, 2.46],
47     # Neuron 2
48     [3.5, 6.8, 2.59, 6.12, 3.66],
49     # Neuron3
50     [4.05, 2.34, 0.39, 1.45, 3.24]
51 ]
52
53 # Inisialisasi bias 2
54 # Jumlah bias pada layer 2 berisi 3
55 biases_2 = [0.5, 3.99, 4.25]
56
57 # Perhitungan output layer 1
58 layer_outputs_1 = np.dot(inputs, np.array(weights_1).T) + biases_1
59
60 # Perhitungan output layer 1
61 layer_outputs_2 = np.dot(layer_outputs_1, np.array(Weights_2).T) +
    biases_2
62
63 # Print layer_output 2
64 print(layer_outputs_2)

```

### Output :

```

[[ 359.244797  239.817604  166.960743]
 [ 971.062871  989.891011  337.513299]
 [ 578.938958  370.614042 -98.252242]
 [1165.536731  712.022142  214.331468]
 [ 386.776676  461.274635  192.245398]
 [1340.726464 1347.373753  544.015508]]
>

```

### Analisis :

- Inisialisasi numpy digunakan untuk mempermudah perhitungan atau dapat disebut juga dengan metode perhitungan.
- Pada Multi Neuron Batch Input terdapat input layer 10 dan per batchnya ada 6 jadi 6x10 (Matriks).
- Terdapat 2 Weights dan Biases
  - Panjang weights 1 terdiri dari jumlah inputnya = 10, sedangkan jumlah weights 1 terdiri dari jumlah neuronnya 5. Jadi 5x10 (Matriks).
  - Panjang weights 2 terdiri dari hidden layer 1 = 5, sedangkan jumlah weights 2 terdiri dari jumlah hidden layer 2 = 3. Jadi 3x5 (Matriks).
  - Biases layer 1 adalah = 5 neuron
  - Biases layer 2 adalah = 3 neuron

- Cara perhitungannya dengan np.dot untuk menghitung variable (input, weight 1) yang telah dibuat, lalu hasil dari perhitungan tersebut akan ditambahkan dengan biases 1.
- Setelah perhitungan selesai, maka dihitunglah layer 2 yang berasal dari hasil perhitungan layer 1 lalu dihitung kembali dengan weight 2 dan biases 2.
- Terakhir print Output untuk layer 2 untuk mencetak hasil perhitungannya, karena yang diinginkan adalah output dari layer 2.