

LAPORAN UTS KECERDASAN BUATAN



Oleh:

Alvin Febrianto

21091397031

D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

TAHUN AJARAN 2022/2023

A. Single Neuron

- i. Input layer feature 10
- ii. Neuron 1

- Source code

```
1a_SingleNeuron_031_AlvinFebrianto.py x
1 # Alvin Febrianto
2 # 21091397031
3 # 2021 A / D4 Manajemen Informatika
4
5 # a. Single Neuron
6 # i. Input layer feature 10
7 # ii. Neuron 1
8
9 import numpy as np
10
11 # Inisialisasi variabel inputs
12 inputs = [0.8, 4, 7, -3.2, 1.3, 3, -8, -2.0, 4.5, 2.4]
13
14 # Inisialisasi variabel weights
15 weights = [-3.6, 3.2, 6, -1.8, 2.8, -7, 5.7, 3.4, 3.1, 4.2]
16
17 # Inisialisasi variabel bias
18 bias = 6
19
20 # Penghitungan output menggunakan rumus dot product vector (inputs*weights)+bias
21 output = np.dot(weights, inputs) + bias
22
23 # Menampilkan output
24 print(output)

17.95
[Finished in 4.3s]
```

Program pertama kali akan melakukan *import* NumPy (salah satu library Python) yang bisa digunakan untuk melakukan operasi vektor dan matriks. Selanjutnya mengisi sebanyak sepuluh angka pada variabel *inputs* sesuai soal. Diikuti dengan pengisian variabel *weights* sebanyak sepuluh angka, sesuai banyaknya angka pada variabel *inputs*. Lalu isi variabel *bias* yang berfungsi sebagai tambahan *inputs*. Karena hanya ada satu set *weights* maka isi *bias* hanya sebanyak satu angka, sehingga dapat didefinisikan sebagai satu neuron. Selanjutnya dilakukan penghitungan operasi dot product vector yaitu mengalikan *inputs* dan *weights* dari nilai indeks yang sama, lalu dilakukan penambahan *bias*. Langkah terakhir, cetak outputnya dengan memanggil fungsi *print*.

B. Multi Neuron

- i. Input layer feature 10
- ii. Neuron 5

- Source code

```
1b_MultiNeuron_031_AlvinFebrianto.py x
1 # Alvin Febrianto
2 # 21091397031
3 # 2021 A / D4 Manajemen Informatika
4
5 # b. Multi Neuron
6 # i. Input layer feature 10
7 # ii. Neuron 5
8
9 import numpy as np
10
11 # Inisialisasi variabel inputs
12 inputs = [-8, 7.1, -5, 4.2, 5.3, 2.0, 7.9, 2.4, 8.4, -4]
13
14 # Inisialisasi variabel weights
15 weights = [[1.4, 2.0, 2.8, -4.2, 6.0, 2.4, -3.8, 1.2, 6.2, 5.1],
16            [6.0, 6.6, 2.8, -8.2, 6.2, 1.6, 5.1, 3.6, -7.1, 4.0],
17            [8.6, -2.3, 5.1, 7.4, 1.4, -2.4, 7.5, 5.7, 1.9, 8.4],
18            [5.6, 4.0, -6.5, 8.2, 2.6, 1.8, -7.3, 7.8, 3.7, 5.9],
19            [6.0, 3.0, 6.6, 1.3, -3.2, 7.1, 7.4, 9.4, 1.1, -7.6]]
20
21 # Inisialisasi variabel biases
22 biases = [2.0, 4, 0.8, 5.2, 3]
23
24 # Penghitungan output menggunakan rumus dot product vector (inputs*weights)+biases
25 output = np.dot(weights, inputs) + biases
26
27 # Menampilkan output
28 print(output)
```

```
[ 14.5 -36.23 -20.84  41.65  66.66]
```

Program pertama kali akan melakukan import NumPy (salah satu library Python) yang bisa digunakan untuk melakukan operasi vektor dan matriks. Selanjutnya mengisi sebanyak sepuluh angka pada variabel inputs sesuai soal. Diikuti dengan pengisian variabel weights sebanyak sepuluh angka, sesuai banyaknya angka pada variabel inputs. Karena jumlah neuron pada soal adalah lima, maka tambahkan dengan total lima set pada variabel weights. Lalu isi variabel biases yang berfungsi sebagai tambahan inputs. Karena di sini ada lima set weights maka isi variabel biases sebanyak lima angka, sehingga dapat didefinisikan sebagai lima neuron. Selanjutnya dilakukan penghitungan operasi dot

product vector yaitu mengalikan inputs dan weights dari nilai indeks yang sama, lalu dilakukan penambahan biases. Langkah terakhir, cetak outputnya dengan memanggil fungsi print.

C. Multi Neuron Batch Input

- i. Input layer feature 10
 - ii. Per batch nya 6 input
 - iii. Neuron 5
- Source code

```
1c_MultiNeuronBatchInput_031_AlvinFebrianto.py x
1  # Alvin Febrianto
2  # 21091397031
3  # 2021 A / D4 Manajemen Informatika
4
5  # c. Multi Neuron Batch Input
6  # i. Input layer feature 10
7  # ii. Per batch nya 6 input
8  # iii. Neuron 5
9
10 import numpy as np
11
12 # Inisialisasi variabel inputs
13 inputs = [[-6.6, 7.2, 2.1, 6.8, 5.4, 1.9, 1.0, 6.9, 1.1, -4.3],
14           [3.5, -8.6, 5.3, 6.8, 2.5, 2.0, 4.7, 1.3, -9.5, 7.0],
15           [2.0, 7.9, -8.1, 2.3, 9.9, 6.3, 4.3, -8.8, 6.7, 8.2],
16           [1.9, 4.7, 9.1, -3.3, 6.8, 5.2, -7.8, 5.0, 2.8, 3.7],
17           [3.0, 4.8, 5.5, 8.6, -8.5, -7.0, 5.8, 5.0, 3.9, 3.6],
18           [9.0, 7.5, 9.6, -2.2, 8.8, 2.3, -6.6, 8.6, 2.8, 1.7]]
19
20 # Inisialisasi variabel weights
21 weights = [[1.4, 2.0, 2.8, -4.2, 6.0, 2.4, -3.8, 1.2, 6.2, 5.1],
22           [6.0, 6.6, 2.8, -8.2, 6.2, 1.6, 5.1, 3.6, -7.1, 4.0],
23           [8.6, -2.3, 5.1, 7.4, 1.4, -2.4, 7.5, 5.7, 1.9, 8.4],
24           [5.6, 4.0, -6.5, 8.2, 2.6, 1.8, -7.3, 7.8, 3.7, 5.9],
25           [6.0, 3.0, 6.6, 1.3, -3.2, 7.1, 7.4, 9.4, 1.1, -7.6]]
26
27 # Inisialisasi variabel biases
28 biases = [2.0, 4, 0.8, 5.2, 3]
29
30 # Penghitungan output menggunakan rumus dot product vector
31 output = np.dot(inputs, np.array(weights).T) + biases
```

```

32
33 # Menampilkan output
34 print(output)

```

```

[[ 10.81   3.49   4.31  81.83 110.06]
 [ -43.72  70.12 210.14   3.79  31.57]
 [ 119.24  73.54  37.98 129.73 -104.57]
 [ 178.55 122.58  31.76 103.6   63.67]
 [ -46.22 -31.05 221.8   73.6  127.23]
 [ 185.47 194.88 119.55 167.83 150.33]]

```

Program pertama kali akan melakukan import NumPy (salah satu library Python) yang bisa digunakan untuk melakukan operasi vektor dan matriks. Selanjutnya mengisi sebanyak sepuluh angka pada variabel inputs sesuai ketentuan soal, namun kali ini dengan jumlah per batch-nya enam inputs. Di sini diketahui banyaknya jumlah inputs yaitu enam baris sehingga bisa diartikan sebagai matriks (6×10). Berikutnya isikan variabel weights sebanyak sepuluh angka, sesuai banyaknya angka pada variabel inputs. Karena jumlah neuron pada soal adalah lima, maka isi variabel weights dengan total lima set. Lalu isi variabel biases yang berfungsi sebagai tambahan inputs. Karena di sini ada lima set weights maka isi variabel biases sebanyak lima angka, sehingga dapat didefinisikan sebagai lima neuron. Untuk menyelesaikan perhitungan pada multi neuron batch input dapat menggunakan rumus perkalian matriks dengan salah satu matriksnya ditransposekan lebih dahulu agar kedua matriks tersebut dapat dikalikan. Selanjutnya tinggal melakukan penghitungan operasi dot product yaitu mengalikan inputs dan weights dari nilai indeks yang sama, lalu dilakukan penambahan biases. Langkah terakhir, cetak outputnya dengan memanggil fungsi print.