

Tugas Ujian Tengah Semester Kecerdasan Buatan
(Single Neuron Multiple Neuron, Multiple Neuron Batch)



Disusun oleh:

Diego Athalla Samudero (21091397042)

Program Studi D4 Manajemen Informatika

Program Vokasi

Universitas Negeri Surabaya

2022

a.

Source Code

```
# No. 1A

#single perceptron, single neuron

#memulai numpy
import numpy as np

#memasukkan variabel yang diinginkan
inputs = [1.0, 2.0, 1.5, 4.2, 6.0, 7.4, 2.1, 1.3, 6.1, 3.7]
weights = [-1.2, 3.5, 5.3, 7.2, -3.1, 8.0, 1.4, 7.3, 9.9, 5.1]
bias = 4.0

#proses perhitungan output
output = np.dot(weights, inputs) + bias

#pemanggilan output
print(output)
```

Output:



```
180.28
```

Analysis:

1. Memakai numpy sebagai method perhitungan untuk mempermudah penulisan code
2. Inputkan variabel dengan ketentuan input layer feature 10 dan neuron 1
3. Masukkan bobot untuk per neuronnya
4. Masukkan variabel bias yang diinginkan
5. Membuat output untuk menghitung semua variabel yang sudah kita masukkan tadi dengan bantuan numpy
6. Lalu munculkan hasil yang sudah dihitung sebelumnya dengan command print

Source Code

b.

```
# No. 1B

#multi neuron, multi perceptron

##memulai numpy
import numpy as np

#memasukkan variabel yang diinginkan
inputs = [1.5, 3.5, 2.5, 1.2, 3.3, 3.8, 4.1, 4.0, 5.1, 6.4]

weights = [[-0.25, 2.3, 4.3, 2.2, -2.1, 8.0, 4.5, 5.3, 2.9, 4.4],
            [-1.5, 2.6, 4.8, -1.2, 9.0, 2.4, 1.3, 0.5, 1.9, 2.7],
            [9.5, -0.3, 0.65, -7.3, 6.1, 2.0, 7.0, -6.0, -2.6, -1.0],
            [-0.2, 1.1, 2.1, 3.9, 1.8, -8.6, 4.6, -9.0, 3.1, 0.35],
            [-4.5, 4.2, 3.0, -8.1, 6.4, 9.7, 7.3, 4.5, 6.2, -0.32]]

#penggunaan bias untuk neuronnya
biases = [9.6, 0.3, 0.2, 8.8, 8.4]

#proses perhitungan output
output = np.dot(weights, inputs) + biases

#pemanggilan output
print(output)
```

Output:

```
[136.735  90.83  19.035  -3.55 149.612]
```

Analisis:

1. Penggunaan numpy sebagai method perhitungan untuk mempermudah penulisan code
2. Memasukkan variabel dengan ketentuan input layer feature 10 dan neuron 5
3. Masukkan bobot untuk per neuronnya
4. Masukkan variabel bias yang diinginkan
5. Membuat output untuk menghitung seluruh variabel yang sudah kita masukkan tadi dengan bantuan numpy
6. Lalu munculkan hasil yang sudah dihitung sebelumnya dengan command print

Source Code

c.

```
# No. 1C

#multi perceptron, neuron batch

import numpy as np

#memasukkan variabel yang diinginkan
inputs = [[9.2, 8.1, 6.7, 1.3, 7.4, 2.1, 6.1, 3.7, 9.1, 1.0],
          [1.2, 2.2, 3.2, 4.6, 6.5, 7.0, 9.9, 1.7, 7.9, 3.0],
          [6.6, 0.3, 5.0, 6.1, 8.8, 5.1, 9.8, 4.3, 4.0, 8.2],
          [2.1, 5.3, 7.0, 6.3, 7.8, 5.5, 6.1, 3.7, 9.2, 6.8],
          [3.1, 2.4, 6.1, 8.2, 5.3, 6.4, 5.0, 6.9, 0.7, 0.5],
          [4.2, 8.7, 1.9, 4.1, 9.6, 6.7, 8.9, 8.6, 8.3, 1.9]]

#isi weights mengikuti inputs dan neuronnya
weights = [[1.3, 3.7, 3.3, 9.9, -8.9, 4.8, 3.7, 6.9, 5.3, 5.6],
           [0.25, 5.8, 1.9, 2.7, 1.3, 0.7, 7.3, 1.0, 7.9, 8.9],
           [7.4, 8.2, 5.6, -2.7, 5.1, 8.3, 1.9, 9.3, 3.2, 7.8],
           [6.7, 5.1, 3.2, 6.7, 9.1, 9.3, 8.2, 7.8, -6.5, 4.8],
           [6.3, 1.0, 6.3, -4.8, 2.3, 7.2, 9.1, 6.7, 9.4, 9.9]]

#penggunaan bias untuk neuronnya
biases = [7.8, 6.4, 3.9, 2.2, 9.8]

#proses penghitungan output
layer_outputs = np.dot(inputs, np.array(weights).T) + biases

#pemanggilan output
print(layer_outputs)
```

Output:

```
[[130.86  212.03  310.5   246.7   319.71 ]
 [156.38  214.39  210.87  244.26  288.43 ]
 [173.59  231.19  289.31  359.59  347.63 ]
 [207.53  263.395 299.     281.76  333.83 ]
 [178.19  119.575 218.5   310.09  192.33 ]
 [185.93  245.81  348.73  353.16  343.01 ]]
```

Analisis:

1. Penggunaan numpy sebagai method perhitungan untuk mempermudah penulisan code
2. Memasukkan variabel dengan ketentuan input layer feature 10 dengan per batch nya 6 input beserta dengan 5 neuron lalu digunakan matrix 6×10
3. Masukkan bobot untuk per neuronnya menggunakan panjang input x jumlah weights = jumlah neuron (5)
4. Masukkan variabel bias yang diinginkan menyesuaikan dengan jumlah neuronnya

d.

5. Membuat output untuk menghitung seluruh variabel yang sudah kita masukkan tadi dengan bantuan numpy
6. Lalu munculkan hasil yang sudah dihitung sebelumnya dengan command print