

Laporan Analisa UTS Neuron/Peceptron  
Kecerdasan Buatan



Di susun Oleh :  
Ariel Fikri Ramadhani (A\_21091397067)

Program Studi D4 Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

2022

## Soal

2

1. Buat kodingan
  - a. Single Neuron,
    - i. Input layer feature 10
    - ii. Neuron 1
  - b. Multi Neuron
    - i. Input layer feature 10
    - ii. Neuron 5
  - c. Multi Neuron Batch Input
    - i. Input layer feature 10
    - ii. Per batch nya 6 input
    - iii. Neuron 5
2. Buat dokumentasi dengan bahasa kalian sendiri, jelaskan :
  - a. Masukkan Source Code kodingan dan jelaskan dengan bahasa kalian sendiri step by step pengerjaannya
  - b. Dijelaskan cara kerja setiap penghitungan output setiap matrix nya.
    - i. single perceptron → dot product dan penambahan bias
    - ii. multi perceptron → dot product dan penambahan bias
    - iii. multi perceptron batch input → dot product, transpose dan penambahan bias

## Code

Single Neuron input layer 10, 1 neuron

Pertama melakukan inisialisasi numpy & Library

---

Kedua melakukan input variable dan input

---

```
#Layer Input 10 Features
inputs = [0.5, 0.7, 0.7, 3.4, 0.6, 0.7, 1.5, 0.3, 3.9, 1.6]
```

Ketiga melakukan perhitungan matrix, neuron, dan output

Weights . Input  
1X10 . 10X1

```
#Neuron 1
weights = [[2.7, 3.6, 3.5, 1.8, 0.0, 2.1, 0.4, 1.8, 0.5, 0.3],]
```

Keempat mendapat hasil dari perhitungan

Outputs = np.dot(weights, input) + bias

```
#Bias from Layer
biases = [2.0, 3.0, 1.0, 1.5, 2.5]

#Calculating Output
layer_outputs = np.dot(weights, inputs) + biases

#Printing Output
print(layer_outputs)
```

Outputs = [np.dot(weights, input) + bias]

```
#Calculating Output
layer_outputs = np.dot(weights, inputs) + biases

#Printing Output
print(layer_outputs)
```

b. Multi neuron input layer 10, 5 neuron

Pertama inialisasi numpy dengan cara :

```
#Intializing Library
import numpy as np
```

Kedua melakukan inialisasi variable dan menentukan input

```
#Input Layer 10 Features
inputs = [1.4, 1.2, 2.7, 4.5, 3.5, 1.7, 2.0, 1.5, 0.3, 4.0]
```

Ketiga melakukan perhitungan matrix, neuron, dan output

```
#Neuron 5
weights = [[0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0],
           [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0],
           [0.03, 0.04, 0.97, 0.44, 0.27, 0.3, 0.35, 0.52, 0.53, 0.47],
           [0.95, 0.48, 0.07, 0.45, 0.09, 0.01, 0.55, 0.63, 0.33, 0.27],
           [0.64, 0.07, 0.09, 0.21, 0.86, 0.25, 0.07, 0.01, 0.31, 0.28]]

#Bias dari Layer

biases = [1.0, 2.0, 0.5, 2.5, 1.5]

#Calculating Layer Output

layer_outputs = np.dot(weights, inputs) + biases

#Printing Out
print(layer_outputs)
```

Keempat hasil dari perhitungan

[26.26 65.15 10.163 10.176 8.471]

c. Multi Neuron Batch input, 10 layer input, perbatch 6, 5 neuron

Pertama melakukan inisialisasi numpy

```
#Intializing Library |  
import numpy as np
```

Kedua melakukan inisialisasi validasi dan menentukan input

Ketiga melakukan perhitungan matrix, neuron, batch, dan output

```
weights = [[0.2, 0.8, 0.5, 0.2, 0.1, 0.4],  
           [0.5, 0.91, 0.26, 0.5, 0.13, 0.15],  
           [0.26, 0.27, 0.17, 0.87, 0.77, 0.11],  
           [0.31, 0.33, 0.34, 0.35, 0.37, 0.38],  
           [0.21, 0.22, 0.23, 0.24, 0.25, 0.26]]  
  
biases = [2.0, 3.0, 0.5, 1.5, 2.0]
```

Keempat mendapatkan hasil dari perhitungan

```
[[4.22  5.922  2.436  3.441  3.314 ]  
 [3.027  3.8771 1.8647 2.7005 2.8156]  
 [2.762  4.0557 1.2442 2.1909 2.4679]  
 [2.755  3.8238 1.3587 2.2244 2.4912]  
 [2.46   3.5305 1.066  1.9572 2.3101]  
 [3.12   4.081  1.061  2.22   2.486 ]  
 [3.071  4.2851 1.2105 2.2188 2.4845]  
 [2.648  3.9715 2.0313 2.3587 2.583 ]  
 [2.641  3.6587 1.2159 2.1426 2.4358]  
 [2.51   3.5565 1.083  1.9912 2.3331]]
```

```
** Process exited - Return Code: 0 **  
Press Enter to exit terminal
```