

**LAPORAN PRAKTIKUM UTS  
KECERDASAN BUATAN  
NEURAL NETWORKS**



Disusun oleh :  
Sayyidina Auliya

NIM/Kelas :  
21091397034/2021B

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

## A. SOURCE CODE

### 1. Single Neuron



```
#Sayyidina Auliya
#21091397034

#inisialisasi library
import numpy as np

#input layer feature 10
inputs = [1.2,5.1,2.1,3.7,4.2,9.1,6.2,7.8,1]
weights = [3.1,2.1,8.7,4.2,8.4,3.3,7.1,9.4,6]

#neuron 1
bias = 3

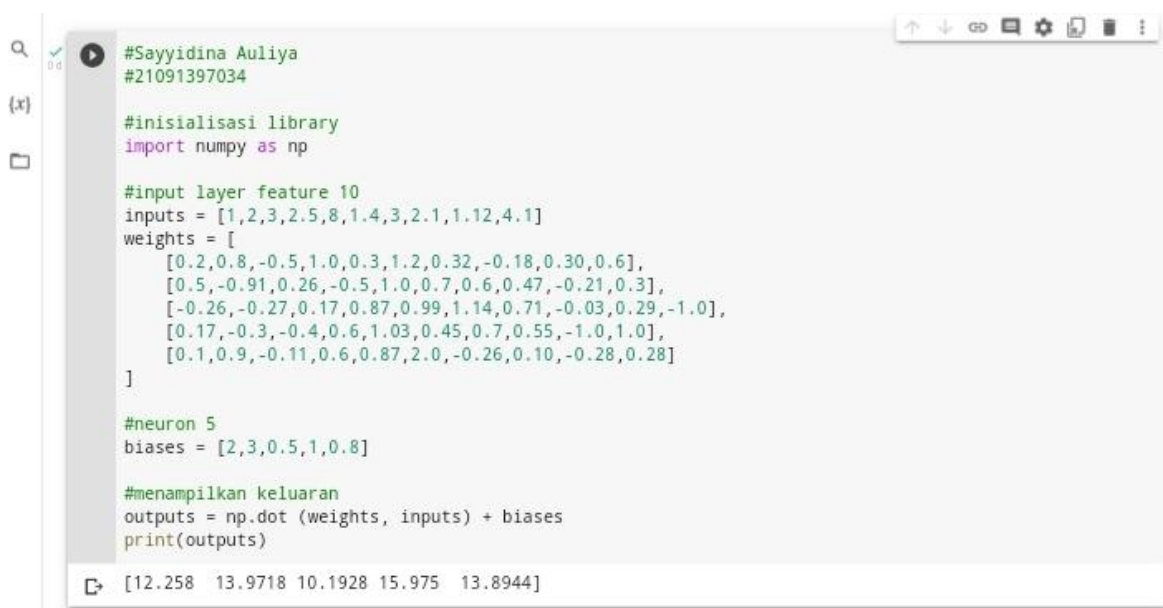
#menampilkan keluaran
outputs = np.dot(weights, inputs) + bias
print(outputs)
```

205.66

Pengerjaan :

- import numpy sebagai library python
- setiap neuron memiliki koneksi yang unik ke setiap neuron sebelumnya, yang outputnya menjadi neuron yang inputnya kita kodekan. Dan hanya perlu membuat beberapa angka sebagai input (input 10)
- setiap input unik memiliki bobot unik yang juga saling terkait. Karena inputnya sebanyak 10, kita perlu menyatakan bobot sebanyak 10
- setiap neuron (inputs dan weights) unik memiliki bias yang unik juga. Jadi diinputkan bias sama dengan 3
- langkah pertama sebuah neuron adalah menjumlahkan semua inputs kali bobot ditambah bias, jadi sederhananya kita masukkan  $\text{outputs} = \text{np.dot}(\text{weights}, \text{inputs}) + \text{bias}$
- mencetak output, jalankan. Dengan `print (outputs)`

### 2. Multi Neuron



```
#Sayyidina Auliya
#21091397034

#inisialisasi library
import numpy as np

#input layer feature 10
inputs = [1,2,3,2.5,8,1.4,3,2.1,1.12,4.1]
weights = [
    [0.2,0.8,-0.5,1.0,0.3,1.2,0.32,-0.18,0.30,0.6],
    [0.5,-0.91,0.26,-0.5,1.0,0.7,0.6,0.47,-0.21,0.3],
    [-0.26,-0.27,0.17,0.87,0.99,1.14,0.71,-0.03,0.29,-1.0],
    [0.17,-0.3,-0.4,0.6,1.03,0.45,0.7,0.55,-1.0,1.0],
    [0.1,0.9,-0.11,0.6,0.87,2.0,-0.26,0.10,-0.28,0.28]
]

#neuron 5
biases = [2,3,0.5,1,0.8]

#menampilkan keluaran
outputs = np.dot(weights, inputs) + biases
print(outputs)
```

[12.258 13.9718 10.1928 15.975 13.8944]

Pengerjaan :

- import numpy sebagai library python
- memodelkan 5 neuron dengan 10 input. Lima neuron berarti perlu memasukkan lima set bobot (weights). Dimana setiap set nya memiliki 10 nilai karena ada 10 input
- kemudian, kita membutuhkan lima bias unik atau terpisah
- selanjutnya menampilkan keluaran atau neuron. Karena memodelkan lima neuron, outputnya pun akan ada lima, bukan nilai tunggal seperti output single neuron

### 3. Multi Neuron Batch Input

```
#Sayyidina Auliya
#21091397034

#inisialisasi library
import numpy as np

#input layer feature 10
#per batch 6 input
inputs = [
    [0.4,0.8,1.2,1.5,-0.7,4.4,1.8,2.1,1.3,2.4],
    [3.8,-1.0,-1.6,2.7,3.2,3.5,-1.9,-4.2,1.4,-0.9],
    [1.7,2.5,3.4,2.8,2.0,3.6,3.9,3.1,2.3,2.9],
    [3.3,3.7,2.6,4.1,3.0,1.1,2.2,0.81,-1.99,3.74],
    [0.2,-0.27,3.8,1.4,6.667,5.1,0.73,0.11,2.78,4.0],
    [-0.8,-0.9,-0.10,3.7,2.3,4.8,1,3.33,0.65,4.6]
]

weights = [
    [2.5,1.3,-1.3,2.1,0.1,3.2,-2.1,1.1,0.34,-0.77],
    [0.3,3.6,-6.0,2.7,1.31,3.9,0.76,9.0,4.9,2.5],
    [0.81,1.5,-1.5,5.8,9.2,1.6,0.71,-0.5,2,1.0],
    [0.2,0.57,5.5,9.8,9.01,6.6,0.85,7.3,5.90,4.1],
    [7.1,3.29,4,6,9.9,0.27,1.3,-1.7,-0.2,1.7]
]

#neuron 5
biases = [4,5,0.3,1.7,0.2]

#menampilkan keluaran
outputs = np.dot(inputs, np.array(weights).T)+ biases
print(outputs)
```

Output :

```
[[ 18.764    53.731    14.552    80.639    16.32    ]
 [ 32.009     2.638    57.629    43.777    69.175    ]
 [ 18.449    67.714    49.446   142.79     75.937    ]
 [ 18.8246    31.561    58.68    104.615   109.039    ]
 [ 15.5889    38.85857   81.9967   164.16127   98.718    ]
 [  5.092    81.534    57.458   137.749    49.768    ]]
```

0 d selesai pada 13.37

Pengerjaan :

- import numpy sebagai library python dan menambahkan input sebanyak 10
- memodelkan lima neuron dengan enam input per batch-nya, sehingga menjadi kumpulan input (matriks) yang terdiri dari input 6 set, dan bobot 5 set yang masing-masing set nya memiliki 10 nilai karena ada 10 input
- kita perlu melakukan dot product dan menempatkan hasilnya dalam array output

- tambahkan bias, dan tampilkan hasilnya dengan print (outputs)

## B. CARA KERJA SETIAP PERHITUNGAN

### 1. Single perceptron

- Dot product (kombinasi inputs dan weights) sebagai cara ringkas untuk operasi yang dibutuhkan dengan mengalikan weights dan input dari nilai indeks yang sama dan menambahkan nilai yang dihasilkan, dan
- penambahan vektor akan digunakan untuk penambahan bias

### 2. Multi perceptron

- Dot product (kombinasi inputs dan weights) sebagai cara ringkas untuk operasi yang dibutuhkan dengan mengalikan weights dan input dari nilai indeks yang sama dan menambahkan nilai yang dihasilkan, dan
- penambahan vektor akan digunakan untuk penambahan bias

### 3. Multi perceptron batch input

- Sebelumnya dot product dilakukan tanpa transposisi, setelah adanya matriks (karena kombinasi inputs dan weight) dilanjutkan ke produk matriks. Setelah itu di transpose array kedua sehingga berubah posisi, baris menjadi kolom dan sebaliknya.
- Dot product mengambil baris dari array pertama dan kolom dari array kedua. Matriks produk dari input dan bobot akan menghasilkan matriks yang berisi dot product yang perlu dihitung. Dengan  $\text{outputs} = \text{np.dot}(\text{inputs}, \text{np.array(weights).T}) + \text{biases}$ .
- Selanjutnya, tambahkan bias