LAPORAN UJIAN TENGAH SEMESTER
KECERDASAN BUATAN

Disusun Oleh :

Nama : Shandy Ilham Alamsyah

NIM : 21091397015

Program Studi D4 Manajemen Informatika
Program Vokasi
Universitas Negeri Surabaya
2022

- **Single Neuron**

Source Code :

```
1   # Program Single Neuron,
2   # 1. Input layer feature 10
3   # 2. Neuron 1
4
5   # inisialisasi numpy
6   import numpy as np
7
8   # inisialisasi variable
9   inputs  = [2, 5.5, 9, 4, 1, 6.5, 3, 7, 2.5, 6]
10
11  # inisialisasi bobot variable
12  weights = [-3.3, 4.2, 1.8, 3.8, -4.3, -0.3, 0.9, 1.9, 2.1, 2.8]
13
14  # inisialisasi bias
15  bias = 7
16
17  # penghitungan  output = (input*weight)+bias
18  output = np.dot(weights, inputs) + bias
19
20  # cetak output
21  print(output)
```

```
>>>
86.69999999999999
```

Penjelasan step by step:

- Memanggil library Numpy yang telah diinstall, untuk memproses komputasi numeric / angka.

```
# inisialisasi numpy
import numpy as np
```

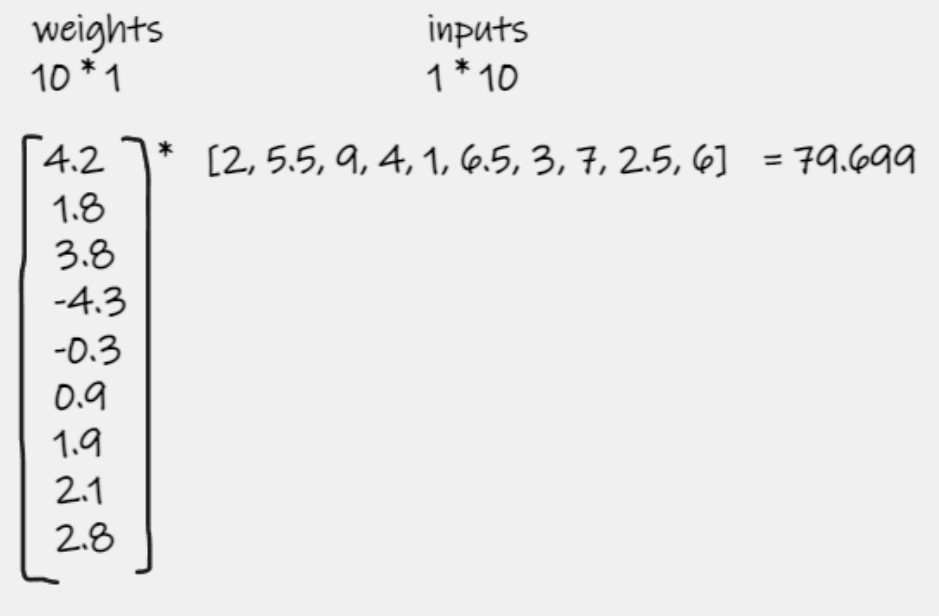
- Mengset nilai dari variabel inputs, weight, dan juga bias. Dengan ketentuan jumlah tiap input layer 10 dengan 1 neuron. Kita bisa memanfaatkan numeric generator di internet.

```
# inisialisasi variable
inputs  = [2, 5.5, 9, 4, 1, 6.5, 3, 7, 2.5, 6]

# inisialisasi bobot variable
weights = [-3.3, 4.2, 1.8, 3.8, -4.3, -0.3, 0.9, 1.9, 2.1, 2.8]

# inisialisasi bias
bias = 7
```

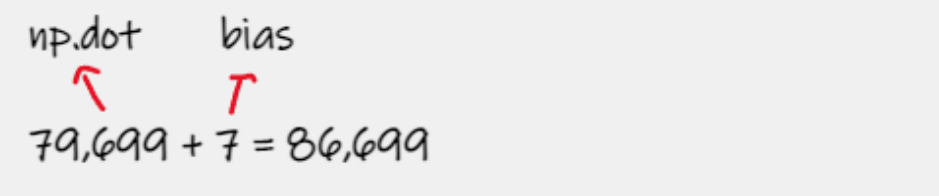
- Melakukan penghitungan untuk mendapatkan output. Dengan rumus (inputs*weights) + bias.

```
# penghitungan  output = (input*weight)+bias
output = np.dot(weights, inputs) + bias
```

Perhitungan dot product.



Kemudian akan ditambahkan dengan bias.



- Mencetak output.

```
# cetak output
print(output)
```

```
>>>
86.69999999999999
```

- **Multi Neuron**

Source Code :

```
# Program Multi Neuron
# 1. Input layer feature 10
# 2. Neuron 5

# inisialisasi numpy
import numpy as np

# inisialisasi variable
inputs  = [2, 5.5, 9, 4, 1, 6.5, 3, 7, 2.5, 6]

# inisialisasi bobot variable
weights =  [[-3.3, 4.2, 1.8, 3.8, -4.3, -0.3, 0.9, 1.9, 2.1, 2.8],
            [4, 3.6, 4.4, -3.3, -1.8, -2.1, 1.1, 2.2, -3.9, 3.4],
            [1, -0.8, 3, 1.5, 1.2, -1.9, -2.7, 4, 0.2, 2.3],
            [2.1, -2.6, 3.9, 4.6, 0.3, -3.5, 2.2, -4.8, 4, 2.3],
            [4.1, -2.2, 0.7, 1.7, 2, 0.2, 4.6, 2.6, -2.3, 3]]

# inisialisasi bias
bias = [7, 3, 0.5, 1.5, 4.5]

# penghitungan output = (input*weight)+bias
output = np.dot(weights, inputs) + bias

# cetak output
print(output)
```

```
>>>
[86.7  71.1  54.15 19.25 61.25]
```

Penjelasan step by step:

- Memanggil library Numpy yang telah diinstall, untuk memproses komputasi numeric / angka.

```
# inisialisasi numpy
import numpy as np
```

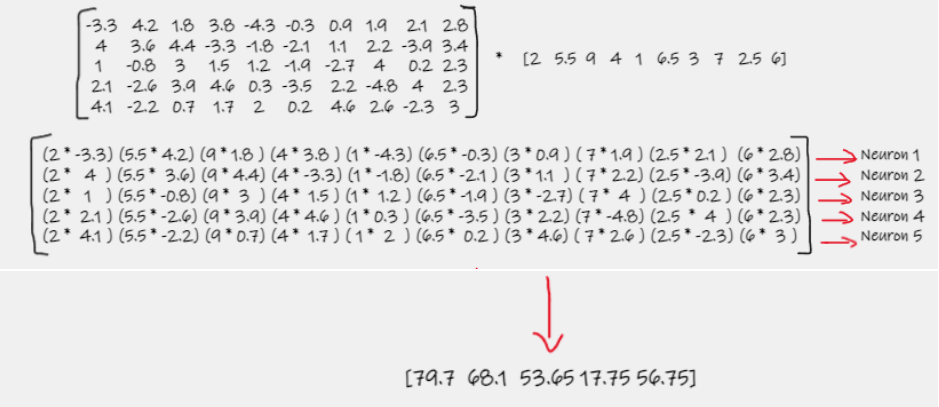
- Mengset nilai dari variabel inputs, weight, dan juga bias. Dengan ketentuan jumlah tiap input layer 10 dengan 5 neuron.

```
# inisialisasi variable
inputs  = [2, 5.5, 9, 4, 1, 6.5, 3, 7, 2.5, 6]

# inisialisasi bobot variable
weights =  [[-3.3, 4.2, 1.8, 3.8, -4.3, -0.3, 0.9, 1.9, 2.1, 2.8],
            [4, 3.6, 4.4, -3.3, -1.8, -2.1, 1.1, 2.2, -3.9, 3.4],
            [1, -0.8, 3, 1.5, 1.2, -1.9, -2.7, 4, 0.2, 2.3],
            [2.1, -2.6, 3.9, 4.6, 0.3, -3.5, 2.2, -4.8, 4, 2.3],
            [4.1, -2.2, 0.7, 1.7, 2, 0.2, 4.6, 2.6, -2.3, 3]]

# inisialisasi bias
bias = [7, 3, 0.5, 1.5, 4.5]
```

- Melakukan penghitungan untuk mendapatkan output. Dengan rumus (inputs*weights) + bias.

```
# penghitungan  output = (input*weight)+bias
output = np.dot(weights, inputs) + bias
```

Perhitungan dot product.



Kemudian akan ditambahkan dengan bias.

- Mencetak output.

```
# cetak output
print(output)
```

```
>>>
[86.7  71.1  54.15 19.25 61.25]
```

- **Multi Neuron Batch Input**

Source Code :

```
# Program Multi Neuron Batch Input
# 1. Input layer feature 10
# 2. Per batch nya 6 input
# 3. Neuron 5

# inisialisasi numpy
import numpy as np

# inisialisasi variable
inputs  =  [[-2, 5.5, 9, 4, 1, 6.5, -3, 7, 2.5, 6],
            [1.6, -1.7, 2.3, -0.2, 3.7, 3.3, -3.7, -3.3, 0.5, 4],
            [4.6, -2.6, 1.3, 1, -2.7, 0.5, 3.2, 4.2, -3.4, 4.7],
            [2.3, -4.1, -3.4, 1, 0.2, -4.5, 5, 1.4, 4, 36],
            [-1.3, 2, -5, 4.3, 4.5, 3.6, -0.6, -0.8, 0.5, 3.5],
            [1.2, 0.7, -4.6, -4.5, 2.2, -2.2, 3, 4.3, -1.5, 4]]

# inisialisasi bobot variable
weights =  [[-3.3, 4.2, 1.8, 3.8, -4.3, -0.3, 0.9, 1.9, 2.1, 2.8],
            [4, 3.6, 4.4, -3.3, -1.8, -2.1, 1.1, 2.2, -3.9, 3.4],
            [1, -0.8, 3, 1.5, 1.2, -1.9, -2.7, 4, 0.2, 2.3],
            [2.1, -2.6, 3.9, 4.6, 0.3, -3.5, 2.2, -4.8, 4, 2.3],
            [4.1, -2.2, 0.7, 1.7, 2, 0.2, 4.6, 2.6, -2.3, 3]]

# inisialisasi bias
bias = [7, 3, 0.5, 1.5, 4.5]

# penghitungan output
output = np.dot(inputs, np.array(weights).T) + bias

# cetak output
print(output)
```

```
>>>
[[ 94.5   48.5   66.35  -2.35  17.25]
 [-16.29   0.79  14.32  25.79   9.38]
 [ 15.38  60.27  26.68   9.12  73.95]
 [ 96.72 103.65  81.87 127.22 147.21]
 [ 15.39 -39.32  -5.82  -4.83  12.81]
 [ -9.28  37.8    5.41 -38.92  41.4 ]]
```

Penjelasan step by step:

- Memanggil library Numpy yang telah diinstall, untuk memproses komputasi numeric / angka.

```
# inisialisasi numpy
import numpy as np
```

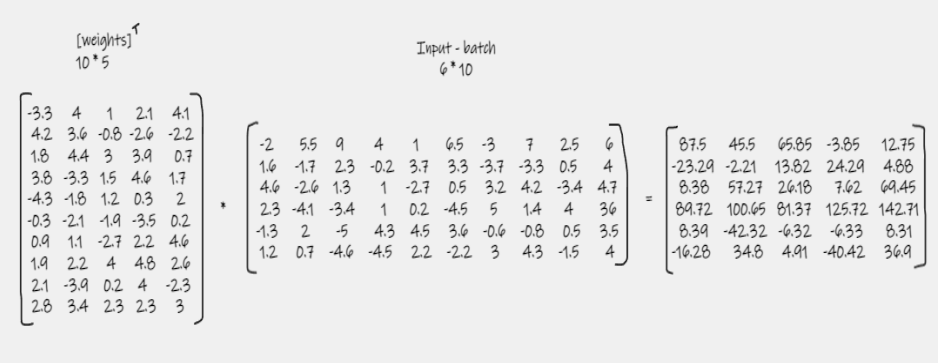
- Mengset nilai dari variabel inputs, weight, dan juga bias. Dengan ketentuan per batch 6 input dan tiap input – batch layer 10 jadi inputs = 6 * 10 dan 5 neuron.

```
# inisialisasi variable
inputs  =  [[-2, 5.5, 9, 4, 1, 6.5, -3, 7, 2.5, 6],
            [1.6, -1.7, 2.3, -0.2, 3.7, 3.3, -3.7, -3.3, 0.5, 4],
            [4.6, -2.6, 1.3, 1, -2.7, 0.5, 3.2, 4.2, -3.4, 4.7],
            [2.3, -4.1, -3.4, 1, 0.2, -4.5, 5, 1.4, 4, 36],
            [-1.3, 2, -5, 4.3, 4.5, 3.6, -0.6, -0.8, 0.5, 3.5],
            [1.2, 0.7, -4.6, -4.5, 2.2, -2.2, 3, 4.3, -1.5, 4]]

# inisialisasi bobot variable
weights =  [[-3.3, 4.2, 1.8, 3.8, -4.3, -0.3, 0.9, 1.9, 2.1, 2.8],
            [4, 3.6, 4.4, -3.3, -1.8, -2.1, 1.1, 2.2, -3.9, 3.4],
            [1, -0.8, 3, 1.5, 1.2, -1.9, -2.7, 4, 0.2, 2.3],
            [2.1, -2.6, 3.9, 4.6, 0.3, -3.5, 2.2, -4.8, 4, 2.3],
            [4.1, -2.2, 0.7, 1.7, 2, 0.2, 4.6, 2.6, -2.3, 3]]

# inisialisasi bias
bias = [7, 3, 0.5, 1.5, 4.5]
```

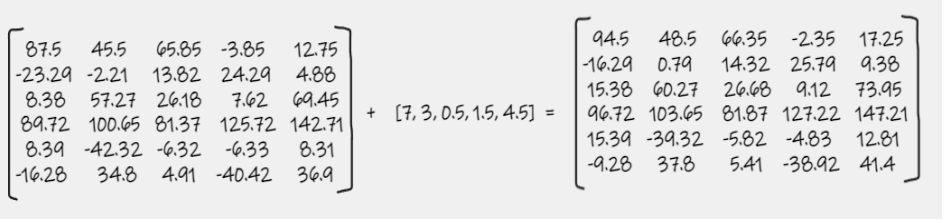
- Melakukan penghitungan untuk mendapatkan output. Dengan rumus :

```
# penghitungan output
output = np.dot(inputs, np.array(weights).T) + bias
```

Perhitungan dot product.



Kemudian akan ditambahkan dengan bias.



- Mencetak output.

```
# cetak output
print(output)
```

```
>>>
[[ 94.5   48.5   66.35  -2.35  17.25]
 [-16.29   0.79  14.32  25.79   9.38]
 [ 15.38  60.27  26.68   9.12  73.95]
 [ 96.72 103.65  81.87 127.22 147.21]
 [ 15.39 -39.32  -5.82  -4.83  12.81]
 [ -9.28  37.8    5.41 -38.92  41.4 ]]
```