

**UTS Part 2 Kecerdasan Buatan  
(Multi Neuron Batch Input)**



Disusun oleh :  
Muhammad Yuda Pratama (21091397025)

**Program Studi D4 Manajemen Informatika**

**Fakultas Vokasi**

**Universitas Negeri Surabaya**

**2022**

## Source Code :

```
1  #Muhammad Yuda Pratama/21091397025
2
3  #inisialisasi numpy
4  #multiple neuron batch and multiple layer dengan layers 10
5
6  import numpy as np
7
8  #inisialisasi variabel dengan Input Layer 10 dan per Batchnya 6 Input
9  inputs = [[4.3, 7.6, 1.2, 8.6, 4.9, 2.0, 1.0, 0.9, 4.4, 1.4],
10           [0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 1.1],
11           [0.1, 1.2, 1.3, 6.5, 7.2, 8.1, 5.0, 6.3, 5.2, 7.8],
12           [0.4, 8.4, 5.9, 2.4, 5.6, 6.7, 2.3, 1.0, 8.4, 9.4],
13           [0.5, 5.3, 6.4, 4.2, 1.9, 4.1, 4.3, 7.3, 1.0, 8.0],
14           [0.6, 9.0, 4.6, 7.0, 3.0, 2.0, 2.3, 9.7, 7.2, 6.0]]
15
16  #panjang weights = panjang inputs(10) ; jumlah weights = jumlah neuron(5)
17  weights1 = [[0.35, -2.5, 6.3, 9.4, 9.3, 2.4, 5.0, 1.0, 8.8, -6.6],
18             [3.4, 1.9, 4.5, 6.3, 7.3, -7.2, -8.6, 9.2, 0.9, 3.7],
19             [0.5, -5.0, 5.5, 9.2, -5.1, 0.2, 0.7, -7.9, 5.0, -4.0],
20             [8.4, 7.1, 1.5, 4.0, 7.0, 2.0, -2.8, 0.9, -4.1, 0.77],
21             [-0.55, 4.0, 5.0, 4.5, -6.4, 3.0, -7.3, 4.6, -6.2, 0.45]]
22
23  #jumlah biases pada hidden layer1 = 5 neuron
24  biases1 = [8.0, 0.4, 3.0, 1.2, 9.3]
25
26  #panjang weights = neuron layer1(5) ; jumlah weights = jumlah neuron layer2(3)
27  weights2 = [[4.5, 2.1, 8.4, 5.6, 2.4],
28             [6.5, 7.3, 2.1, 5.4, 9.0],
29             [1.0, 3.8, 5.0, 6.8, 4.0]]
30
31  #jumlah biases pada hidden layer2 = 3 neuron
32  biases2 = [5, -3, -4.0]
33
34  #perintah untuk menghitung layer1 menggunakan inputs, weights1, dan biases1
35  layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
36
37  #perintah untuk menghitung layer2 menggunakan hasil dari perhitungan pada layer1
38  layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
39
40  #print output layer2
41  print(layer2_outputs)
```

## Output :

```
[[2204.5113 3052.6597 1910.9694]
 [ 284.2687  481.1273  288.4906]
 [1284.6061 2202.0749  942.2118]
 [1287.2368 2105.0772 1112.7664]
 [ 981.4915 2500.1145 1229.239 ]
 [1639.097  3578.559 1745.638 ]]
[Finished in 649ms]
```

### Analisa :

- Inisialisasi numpy untuk mempermudah perhitungan dapat disebut juga dengan metode perhitungan.
- Pada Multi Neuron Batch Input terdapat input layer 10 dan per batchnya adalah 6 jadi  $6 \times 10$  (Matriks).
- Terdapat 2 Weights dan Biases
  - Panjang weights1 terdiri dari jumlah inputnya = 10, sedangkan jumlah weights1 terdiri dari jumlah neuronnya 5. Jadi  $5 \times 10$  (Matriks).
  - Panjang weights2 terdiri dari hidden layer 1 = 5, sedangkan jumlah weights2 terdiri dari jumlah hidden layer 2 = 3. Jadi  $3 \times 5$  (Matriks).
  - Biases layer 1 adalah = 5 neuron
  - Biases layer 2 adalah = 3 neuron
- Cara perhitungannya buat np.dot untuk menghitung variable (input, weight 1) yang telah dibuat, lalu hasil dari perhitungan tersebut akan ditambahkan dengan biases 1.
- Lalu setelah perhitungan diatas selesai maka dihitunglah layer2 yang berasal dari hasil perhitungan layer1 lalu dihitung kembali dengan weight 2 dan biases 2.
- Terakhir print Output untuk layer2 untuk mengeprint hasil perhitungannya, karena yang kita inginkan adalah output dari layer2.