

**Tugas Ujian Tengah Semester ke 2 Kecerdasan Buatan
(Multi Neuron Batch Input)**



Disusun oleh:

Rafie Mirza Ramadhan (21091397037)

Program Studi D4 Manajemen Informatika

Program Vokasi

Universitas Negeri Surabaya

2022

Source Code:

```
# Rafie Mirza Ramadhan - 21091397037 - MI 2021A
# UTS AI 2

# multi perceptron, neuron batch dan multi layer

# penginisialisasian numpy
import numpy as np

# penginisialisasian variabel dengan 10 input & 6 batch, lalu menggunakan matrix menjadi 6 dikalikan 10
inputs = [[3.8, 3.7, 2.6, 8.3, 7.3, 2.4, 5.2, 6.7, 8.2, 4.6],
          [1.3, 1.5, 7.4, 2.5, 7.9, 4.2, 8.7, 3.2, 1.4, 6.7],
          [7.4, 1.2, 5.4, 7.3, 8.1, 3.5, 7.1, 5.2, 6.9, 9.1],
          [8.4, 7.1, 8.3, 1.2, 2.7, 8.8, 9.9, 0.1, 8.1, 9.1],
          [2.3, 1.2, 7.1, 9.1, 1.3, 4.5, 7.9, 7.7, 4.3, 6.9],
          [2.3, 7.6, 8.1, 1.5, 8.4, 9.1, 3.2, 5.4, 2.4, 3.9]]

# panjang weights = pengisian variabel dengan 10 input & jumlah weights = jumlah neuron (5)
weights1 = [[6.7, 8.1, -9.4, 5.4, 6.4, 8.2, 9.1, -5.9, 5.2, 6.4],
            [6.4, 3.2, -1.9, 5.1, 9.2, 5.1, 2.5, 4.5, 7.4, -8.9],
            [-8.8, 5.1, -1.9, 5.9, 6.2, -7.8, 2.9, 4.7, 7.3, 3.1],
            [6.7, 9.9, -7.2, 8.1, -9.6, 6.1, 6.9, 1.1, 7.2, 8.2],
            [3.4, 2.1, -6.2, -7.9, -4.7, 6.2, 3.9, 6.9, 2.9, 8.9]]

# jumlah penggunaan bias pada lapisan layer 1 adalah dengan 5 neuron
biases1 = [8.1, 2.1, 4.1, 2.9, 8.9]

# panjang weights = pengisian variabel dengan 5 input neuron layer 1 & jumlah weights = jumlah neuron layer ke 2 (3)
weights2 = [[8.8, 6.1, 8.1, 1.2, 3.1],
            [1.1, 5.1, 8.4, 9.1, 2.7],
            [8.6, 4.8, 2.9, 5.1, 4.1]]

# jumlah penggunaan bias pada lapisan layer 2 adalah dengan 3 neuron
biases2 = [-6.7, 5.1, -9.2]

# perintah untuk penghitungan layer 1 menggunakan semua variable yang sudah kita masukkan di inputs, weights1, dan biases1
layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1

# perintah penghitungan inputs untuk layer ke 2 (untuk input yang sudah melalui layer 1 & menggunakan hasil perhitungan pada layer1)
layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2

# pemanggilan output yang sudah melewati beberapa layer sebelumnya
print(layer2_outputs)
```

```
[[5235.471 4852.055 4795.678]
 [3040.728 2231.169 2760.497]
 [4990.635 4417.609 4950.848]
 [4865.3   4569.404 5875.591]
 [3488.834 3928.506 3644.367]
 [3492.109 2717.76  3377.002]]
```

Output:

```
[[5235.471 4852.055 4795.678]
 [3040.728 2231.169 2760.497]
 [4990.635 4417.609 4950.848]
 [4865.3   4569.404 5875.591]
 [3488.834 3928.506 3644.367]
 [3492.109 2717.76  3377.002]]
```

Analisa:

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 5×10 untuk weights1 beserta 3×5 untuk weights2, dan 5 untuk bias1 beserta 3 untuk bias2
3. Masukkan bobot untuk per neuronnya menggunakan panjang input x jumlah weights pada per layernya
4. Masukkan variabel bias yang diinginkan menyesuaikan dengan jumlah neuronnya
5. Membuat perhitungan output layer pertama untuk menghitung seluruh variabel yang sudah dimasukkan sebelumnya dengan menggunakan metode numpy agar lebih singkat
6. Membuat perhitungan output untuk layer ke 2 dengan menggunakan hasil perhitungan dari layer ke 1 sebagai inputs, weights2 sebagai beban ditambah dengan bias kedua
7. Lalu munculkan hasil yang sudah dihitung sebelumnya dengan perintah print layer 2 output