

LAPORAN UJIAN TENGAH SEMESTER KECERDASAN BUATAN



Oleh:

Fisma Meividianugraha Subani

21091397017

D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

TAHUN AJARAN 2022/2023

1. Buat Kodingan

a) Source Code Multi Neuron Batch Input

```
C:\Users\LENOVO\Downloads\Part 2\017_fisma_no1a_multineuronbatchinput.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

017_fisma_no1a_multineuronbatchinput.py x
1 # Nama : Fisma Meividianugraha Subani
2 # NIM : 21091397017
3 # Kelas : 2021 A
4
5 # No 1a. Multi Neuron Batch Input
6 # i. Input layer fature 10
7 # ii. Per batchnya 6 input
8 # iii. Hidden layer 1, 5 neuron
9 # iv. Hidden layer 2, 3 neuron
10
11 # inisialisasi library numpy
12 import numpy as np
13 # inisialisasi variabel inputs berdasarkan jumlah batch = 6 dan input layer feature = 10 (matriks 6x10)
14 inputs = [[1.3, 0.5, 2.0, 0.3, 3.1, 5.0, 0.2, 3.0, -1.3, -0.5],
15           [3.0, 0.9, 2.0, 0.3, 0.3, 9.0, 0.2, 3.0, -3.0, -0.9],
16           [8.0, 0.3, 2.0, 0.3, 0.8, 3.0, 0.2, 3.0, -8.0, -0.3],
17           [1.6, 1.2, 2.0, 2.0, 6.1, 2.1, 0.2, 0.2, -1.6, -1.2],
18           [6.0, 7.0, 2.0, 0.3, 0.6, 0.7, 0.2, 3.0, -6.0, -7.0],
19           [1.9, 1.0, 2.0, 2.2, 9.1, 0.1, 0.2, 2.2, -1.9, -1.0]]
20
21 # inisialisasi variabel weights1 [Hidden layer 1] berdasarkan jumlah neuron = 5 dan input layer feature = 10 (matriks 5x10)
22 weights1 = [[1.0, 0.1, 1.9, 9.5, 0.1, 1.0, 9.1, 5.9, -1.0, -0.1],
23             [1.9, 0.5, 1.9, 6.7, 9.1, 5.0, 9.1, 7.6, -1.9, -0.5],
24             [1.3, 1.1, 1.9, 6.7, 3.1, 1.1, 9.1, 7.6, -1.3, -1.1],
25             [1.0, 0.5, 2.0, 0.1, 0.1, 5.0, 0.2, 1.0, -1.0, -0.5],
26             [2.8, 1.1, 2.0, 0.2, 8.2, 1.1, 0.2, 2.0, -2.0, -1.1]]
27
28 # inisialisasi variabel biases1 berdasarkan panjang neuron = 5 [Hidden layer 1]
29 biases1 = [5.0, 9.0, 3.0, 2.1, 0.7]
30
31 # inisialisasi variabel weights2 [Hidden layer 2] berdasarkan jumlah neuron = 3 dan input layer feature = 5 (matriks 3x5)
32 weights2 = [[2.4, 1.0, 2.0, 2.2, 4.2],
33             [1.7, 0.8, 1.9, 4.5, 7.1],
34             [1.0, 1.1, 1.9, 4.5, 0.1]]
35
36 # inisialisasi variabel biases2 berdasarkan panjang neuron = 3 [Hidden layer 2]
37 biases2 = [0.1, 0.0, 1.1]
38
39 # Rumus dot product vector [inputs batch * hasil transpose weights1] + [biases]
40 layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
41
42 # Rumus dot product vector [layer_outputs1 * hasil transpose weights2] + [biases]
43 layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
44
45 # Menampilkan hasil output
46 print(layer2_outputs)
47
48
```

Hasil Output:

```
C:\Users\LENOVO\Downloads\Part 2\017_fisma_no1a_multineuronbatchinput.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

017_fisma_no1a_multineuronbatchinput.py x
1 # Nama : Fisma Meividianugraha Subani
2 # NIM : 21091397017
3 # Kelas : 2021 A
4
5 # No 1a. Multi Neuron Batch Input
6 # i. Input layer fature 10
7 # ii. Per batchnya 6 input
8 # iii. Hidden layer 1, 5 neuron
9 # iv. Hidden layer 2, 3 neuron
10
11 # inisialisasi library numpy
12 import numpy as np
13 # inisialisasi variabel inputs berdasarkan jumlah batch = 6 dan input layer feature = 10 (matriks 6x10)
14 inputs = [[1.3, 0.5, 2.0, 0.3, 3.1, 5.0, 0.2, 3.0, -1.3, -0.5],
15           [3.0, 0.9, 2.0, 0.3, 0.3, 9.0, 0.2, 3.0, -3.0, -0.9],
16           [8.0, 0.3, 2.0, 0.3, 0.8, 3.0, 0.2, 3.0, -8.0, -0.3],
17           [1.6, 1.2, 2.0, 2.0, 6.1, 2.1, 0.2, 0.2, -1.6, -1.2],
18           [6.0, 7.0, 2.0, 0.3, 0.6, 0.7, 0.2, 3.0, -6.0, -7.0],
19           [1.9, 1.0, 2.0, 2.2, 9.1, 0.1, 0.2, 2.2, -1.9, -1.0]]
20
21 # inisialisasi variabel weights1 [Hidden layer 1] berdasarkan jumlah neuron = 5 dan input layer feature = 10 (matriks 5x10)
22 weights1 = [[1.0, 0.1, 1.9, 9.5, 0.1, 1.0, 9.1, 5.9, -1.0, -0.1],
23             [1.9, 0.5, 1.9, 6.7, 9.1, 5.0, 9.1, 7.6, -1.9, -0.5],
24             [1.3, 1.1, 1.9, 6.7, 3.1, 1.1, 9.1, 7.6, -1.3, -1.1],
25             [1.0, 0.5, 2.0, 0.1, 0.1, 5.0, 0.2, 1.0, -1.0, -0.5],
26             [2.8, 1.1, 2.0, 0.2, 8.2, 1.1, 0.2, 2.0, -2.0, -1.1]]
27
28 # inisialisasi variabel biases1 berdasarkan panjang neuron = 5 [Hidden layer 1]
29 biases1 = [5.0, 9.0, 3.0, 2.1, 0.7]
30
31 # inisialisasi variabel weights2 [Hidden layer 2] berdasarkan jumlah neuron = 3 dan input layer feature = 5 (matriks 3x5)
32 weights2 = [[2.4, 1.0, 2.0, 2.2, 4.2],
33             [1.7, 0.8, 1.9, 4.5, 7.1],
34             [1.0, 1.1, 1.9, 4.5, 0.1]]
35
36 # inisialisasi variabel biases2 berdasarkan panjang neuron = 3 [Hidden layer 2]
37 biases2 = [0.1, 0.0, 1.1]
38
39 # Rumus dot product vector [inputs batch * hasil transpose weights1] + [biases]
40 layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
41
42 # Rumus dot product vector [layer_outputs1 * hasil transpose weights2] + [biases]
43 layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
44
45 # Menampilkan hasil output
46 print(layer2_outputs)
47
48
```

```
[[ 591.348  778.628  423.026]
 [ 629.52   841.814  538.706]
 [ 701.508  934.655  457.421]
 [ 635.014  843.505  355.299]
 [ 666.066  882.791  409.187]
 [ 848.138 1102.011  421.687]]
[Finished in 25.8s]
```

2. Buat dokumentasi dengan bahasa kalian sendiri, jelaskan:

- a) Masukkan Source Code kodingan dan jelaskan dengan bahasa kalian sendiri step by step pengerjaannya. (Dijelaskan dibagian komen)
- b) Dijelaskan cara kerja setiap penghitungan output setiap matrix nya.
 - i. Multi perceptron batch input → dot product, transpose dan penambahan bias.

Pada multi perceptron atau neuron batch input terdapat inputs, weights, dan bias. Jumlah batch input pada multi neuron dapat mempengaruhi jumlah inputs, misalnya jumlah per batch nya 6 dengan input layer feature 10, maka:

inputs1 [1.3, 0.5, 2.0, 0.3, 3.1, 5.0, 0.2, 3.0, -1.3, -0.5],

inputs2 [3.0, 0.9, 2.0, 0.3, 0.3, 9.0, 0.2, 3.0, -3.0, -0.9],

inputs3 [8.0, 0.3, 2.0, 0.3, 0.8, 3.0, 0.2, 3.0, -8.0, -0.3],

inputs4 [1.6, 1.2, 2.0, 2.0, 6.1, 2.1, 0.2, 0.2, -1.6, -1.2],

inputs5 [6.0, 7.0, 2.0, 0.3, 0.6, 0.7, 0.2, 3.0, -6.0, -7.0],

inputs6 [1.9, 1.0, 2.0, 2.2, 9.1, 0.1, 0.2, 2.2, -1.9, -1.0]

Dapat diketahui banyaknya jumlah batch inputs yaitu 6 sehingga bisa juga diartikan inputs-inputs tersebut sebagai matriks (6x10). Input layer feature atau panjang input dapat mempengaruhi panjang weights1 pada hidden layer 1, misalnya input layer feature di atas sebanyak 10, maka:

Weights1[1] [1.0, 0.1, 1.9, 9.5, 0.1, 1.0, 9.1, 5.9, -1.0, -0.1],

Weights1[2] [1.9, 0.5, 1.9, 6.7, 9.1, 5.0, 9.1, 7.6, -1.9, -0.5],

Weights1[3] [1.3, 1.1, 1.9, 6.7, 3.1, 1.1, 9.1, 7.6, -1.3, -1.1],

Weights1[4] [1.0, 0.5, 2.0, 0.1, 0.1, 5.0, 0.2, 1.0, -1.0, -0.5],

Weights1[5] [2.8, 1.1, 2.0, 0.2, 8.2, 1.1, 0.2, 2.0, -2.8, -1.1]

Berdasarkan hal tersebut dapat diketahui bahwa weights1[1], weights1[2], weights1[3], weights1[4], weights1[5] memiliki input layer feature atau panjang yang sama seperti inputs, yaitu sebanyak 10 sehingga dapat diartikan weights-weights tersebut sebagai matriks (5x10). Jumlah neuron pada multi neuron batch input dapat berjumlah 2 atau lebih dan jumlah tersebut dapat mempengaruhi jumlah weights dan bias. Pada hidden layer 1 jumlah neuron pada multi neuron batch input dapat mempengaruhi jumlah weights dan bias di hidden layer 1,

misalnya jumlah neuron berdasarkan contoh diatas adalah 5 karena contoh tersebut merupakan multi neuron pada hidden layer 1, maka:

Weights1[1]	Biases1[1] [5.0]
Weights1[2]	Biases1[2] [9.0]
Weights1[3]	Biases1[3] [3.0]
Weights1[4]	Biases1[4] [2.1]
Weights1[5]	Biases1[5] [0.7]

Pada hidden layer 2 jumlah neuron pada multi neuron batch input dapat mempengaruhi jumlah weights dan bias di hidden layer 2, misalnya jumlah neuron berdasarkan contoh diatas adalah 3 karena contoh tersebut merupakan multi neuron pada hidden layer 2, maka:

Weights2[1] [2.4, 1.0, 2.0, 2.2, 4.2],	Biases2[1] [0.1]
Weights2[2] [1.7, 0.8, 1.9, 4.5, 7.1],	Biases2[2] [8.0]
Weights2[3] [1.0, 1.1, 1.9, 4.5, 0.1]	Biases2[3] [1.1]

Input layer feature pada hidden layer 2 berbeda dengan input layer feature pada hidden layer 1. Dapat diketahui input layer feature weights2 berjumlah 5 dan input layer feature weights1 berjumlah 10. Hal tersebut dipengaruhi oleh hasil perhitungan pada layer1 outputs yaitu input batch (matriks 6x10) * hasil transpose weights1 (matriks 10x5) yang akan menghasilkan (matriks 6x5). Kemudian rumus perhitungan pada layer2 outputs menggunakan hasil dari layer1 outputs (matriks 6x5) * hasil transpose weights2 (matriks 5x3) yang akan menghasilkan matriks (6x3). Agar hasil dari layer1 outputs (matriks 6x5) di hidden layer 1 dapat dikalikan dengan weights2 di hidden layer 2 maka jumlah input layer featurenya harus disesuaikan dengan input layer feature pada hasil layer1 outputs yaitu input layer featurenya 5 sehingga menjadi (matriks 3x5). Jadi setelah weights2 ditransposekan, maka dapat dikalikan dengan hasil layer1 outputs yaitu (matriks 6x5) * (matriks 5x3) dan menghasilkan matriks (6x3).

Kalikan sesuai urutannya

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \times \begin{pmatrix} p & q \\ r & s \\ t & u \end{pmatrix} = \begin{pmatrix} ap+br+ct & aq+bs+cu \\ dp+er+ft & dq+es+fu \\ gp+hr+it & gq+hs+iu \end{pmatrix}$$

Gambar 2.b. 1 Rumus Perkalian Matriks

Cara menyelesaikan perhitungan pada multi neuron batch input di hidden layer 1 dapat menggunakan rumus perkalian matriks dengan salah satu matriksnya ditransposekan terlebih dahulu agar kedua matriks tersebut dapat dikalikan. Setelah itu, dijumlahkan dengan biases1 untuk menemukan matriks yang dihasilkan dari semua neuron di hidden layer 1 sehingga menghasilkan layer1 outputs. Dengan cara yang sama, pada hidden layer 2 juga menggunakan rumus perkalian matriks yang salah satu matriksnya menggunakan hasil dari hidden layer1 dan matriks yang lain ditransposekan terlebih dahulu agar kedua matriks tersebut dapat dikalikan. Kemudian dijumlahkan dengan biases2 untuk menemukan matriks yang dihasilkan dari semua neuron di hidden layer 2 sehingga menghasilkan layer2 outputs. Terdapat beberapa langkah mudah dalam menyelesaikan multi neuron batch input pada source code di atas:

1. Inputs1, inputs2, inputs3, inputs4, inputs5, inputs6 dimisalkan sebagai matriks (6x10), jawab:

$$\begin{bmatrix} 1.3 & 0.5 & 2.0 & 0.3 & 3.1 & 5.0 & 0.2 & 3.0 & -1.3 & -0.5 \\ 3.0 & 0.9 & 2.0 & 0.3 & 0.3 & 9.0 & 0.2 & 3.0 & -3.0 & -0.9 \\ 8.0 & 0.3 & 2.0 & 0.3 & 0.8 & 3.0 & 0.2 & 3.0 & -8.0 & -0.3 \\ 1.6 & 1.2 & 2.0 & 2.0 & 6.1 & 2.1 & 0.2 & 0.2 & -1.6 & -1.2 \\ 6.0 & 7.0 & 2.0 & 0.3 & 0.6 & 0.7 & 0.2 & 3.0 & -6.0 & -7.0 \\ 1.9 & 1.0 & 2.0 & 2.2 & 9.1 & 0.1 & 0.2 & 2.2 & -1.9 & -1.0 \end{bmatrix}$$

2. Weights1[1], weights1[2], weights1[3], weights1[4], weights1[5] dimisalkan sebagai matriks (5x10), jawab:

$$\begin{bmatrix} 1.0 & 0.1 & 1.9 & 9.5 & 0.1 & 1.0 & 9.1 & 5.9 & -1.0 & -0.1 \\ 1.9 & 0.5 & 1.9 & 6.7 & 9.1 & 5.0 & 9.1 & 7.6 & -1.9 & -0.5 \\ 1.3 & 1.1 & 1.9 & 6.7 & 3.1 & 1.1 & 9.1 & 7.6 & -1.3 & -1.1 \\ 1.0 & 0.5 & 2.0 & 0.1 & 0.1 & 5.0 & 0.2 & 1.0 & -1.0 & -0.5 \\ 2.8 & 1.1 & 2.0 & 0.2 & 8.2 & 1.1 & 0.2 & 2.0 & -2.8 & -1.1 \end{bmatrix}$$

3. Pada hidden layer 1 matriks inputs (6x10) dan matriks weights1 (5x10) tidak dapat dikalikan karena jumlah kolom (10) matriks inputs tidak sama dengan jumlah baris (5) pada matriks weights1, maka salah satu matriks yaitu matriks weights1 harus ditranposekan terlebih dahulu sehingga menjadi matriks (10x5), jawab:

$$\begin{bmatrix} 1.0 & 1.9 & 1.3 & 1.0 & 2.8 \\ 0.1 & 0.5 & 1.1 & 0.5 & 1.1 \\ 1.9 & 1.9 & 1.9 & 2.0 & 2.0 \\ 9.5 & 6.7 & 6.7 & 0.1 & 0.2 \\ 0.1 & 9.1 & 3.1 & 0.1 & 8.2 \\ 1.0 & 5.0 & 1.1 & 5.0 & 1.1 \\ 9.1 & 9.1 & 9.1 & 0.2 & 0.2 \\ 5.9 & 7.6 & 7.6 & 1.0 & 2.0 \\ -1.0 & -1.9 & -1.3 & -1.0 & -2.8 \\ -0.1 & -0.5 & -1.1 & -0.5 & -1.1 \end{bmatrix}$$

4. Kemudian setelah jumlah kolom (10) matriks inputs sama dengan jumlah baris (10) pada matriks weights1, maka matriks inputs (6x10) dan matriks weights1 (10x5) dapat dikalikan menggunakan rumus perkalian matriks dengan cara baris dikali kolom sehingga menjadi matriks (6x5) kemudian dijumlahkan dengan biases1. Hasil dari perhitungan ini disebut layer1 outputs yang terletak di hidden layer 1, jawab:

$$\begin{bmatrix} 39.18 & 98.08 & 53.02 & 37.58 & 50.1 \\ 46.38 & 99.46 & 54.04 & 61.1 & 41.94 \\ 50.31 & 92.41 & 60.67 & 40.55 & 66.12 \\ 36.95 & 102.83 & 51.56 & 22.05 & 69.47 \\ 45.33 & 78.19 & 67.06 & 31.73 & 65.49 \\ 49.51 & 137.61 & 75.54 & 14.77 & 97.15 \end{bmatrix}$$

5. Weights2[1], Weights2[2], Weights2[3] dimisalkan sebagai matriks (3x5), jawab:

$$\begin{bmatrix} 2.4 & 1.0 & 2.0 & 2.2 & 4.1 \\ 1.7 & 0.8 & 1.9 & 4.5 & 7.1 \\ 1.0 & 1.1 & 1.9 & 4.5 & 0.1 \end{bmatrix}$$

6. Pada hidden layer 2 hasil dari matriks layer1 outputs (6x5) dan matriks weights2 (3x5) tidak dapat dikalikan karena jumlah kolom (5) matriks layer1 outputs tidak sama dengan jumlah baris (3) pada matriks weights2, maka salah satu matriks yaitu matriks weights2 harus ditranposekan terlebih dahulu sehingga menjadi matriks (5x3), jawab:

$$\begin{bmatrix} 2.4 & 1.7 & 1.0 \\ 1.0 & 0.8 & 1.1 \\ 2.0 & 1.9 & 1.9 \\ 2.2 & 4.5 & 4.5 \\ 4.2 & 7.1 & 0.1 \end{bmatrix}$$

7. Kemudian setelah jumlah kolom (5) matriks layer1 outputs sama dengan jumlah baris (5) pada matriks weights2, maka matriks layer1 outputs (6x5) dan matriks weights2 (5x3) dapat dikalikan menggunakan rumus perkalian matriks dengan cara baris dikali kolom sehingga menjadi matriks (6x3) kemudian dijumlahkan dengan biases2. Hasil dari perhitungan ini disebut layer2 outputs yang terletak di hidden layer 2, jawab:

$$\begin{bmatrix} 591.348 & 778.628 & 423.026 \\ 629.52 & 841.814 & 538.706 \\ 701.508 & 934.655 & 457.421 \\ 635.014 & 843.505 & 355.299 \\ 666.066 & 882.791 & 409.187 \\ 848.138 & 1102.011 & 421.687 \end{bmatrix}$$