

LAPORAN KECERDASAN BUATAN

UJIAN TENGAH SEMESTER



Disusun Oleh :

Rifqi Yudo Dewantoro

21091397074

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2022

UTS 2

D. Multi Neuron Batch Input

Input layer feature 10, batchnya 6 input, hidden layer1 (5 neuron), hidden layer2 (3 neuron)

- Source Code

```
uts part 2_074_Rifqi yudo.py x
1  #Rifqi yudo dewantoro 21091397074
2  #Multiple perceptron / Neuron batch and multiple layer 2
3  |
4  #inisialisasi numpy
5  import numpy as np
6
7  # inisialisasi variabel
8  # memasukkan nilai variabel layer feature 10 dengan batch sejumlah 6
9  inputs = [
10     [5.0, 1.3, 2.3, 2.6, 3.1, 3.8, 4.6, 4.9, 5.2, 5.2],
11     [1.3, 1.8, 1.2, 2.4, 4.2, 2.4, 3.2, 4.2, 5.1, 5.4],
12     [3.1, 20.5, 10.0, 25.5, 35.0, 31.5, 18.0, 50.5, 51.0, 57.5],
13     [2.8, 2.8, 2.5, 4.8, 3.2, 3.1, 2.6, 4.9, 5.2, 5.1],
14     [2.2, 5.4, 7.2, 7.4, 8.2, 8.4, 9.2, 9.4, 10.2, 10.4],
15     [16.5, 18.4, 15.2, 16.4, 19.2, 12.4, 17.2, 12.4, 22.2, 11.4]
16 ]
17
18 # memberikan nilai bobot pada variabel sesuai dengan jumlah input
19 # memasukkan jumlah weight sesuai dengan jumlah neuron yaitu sejumlah 5
20 weights1 = [
21     [6.0, 4.8, 8.4, 2.5, 0.1, 3.5, 9.7, 4.5, 6.2, 15.5],
22     [7.4, 9.7, 4.10, 2.84, 3.52, 38.4, 45.2, 4.4, 5.2, 5.4],
23     [3.3, 6.1, 2.3, 10.9, 31.6, 3.82, 4.26, 4.8, 56.6, 55.8],
24     [5.8, 4.3, 4.2, 7.8, 0.2, 7.4, 3.5, 0.7, 40.3, 71.1],
```

```

27
28 # inisialisasi biases pada layer1 sesuai dengan neuron yang ditentukan yaitu layer 1 = 5 neuron
29 biases1 = [4.7, 2.8, 1.0, 9.6, 3.1]
30
31 # inisialisasi jumlah weight 2, weight layer 2 = neuron layer 1 yaitu 5
32 # memasukkan jumlah weight sesuai dengan neuron layer 2 yaitu 3 neuron
33 weights2 = [
34     [15.3, 7.4, 5.9, 8.2, 12.6],
35     [7.0, 1.2, 4.7, 7.4, 9.7],
36     [2.1, 6.9, 3.7, 4.0, 3.4]]
37
38 # inisialisasi biases pada layer2 dengan neuron yang ditentukan yaitu 3
39 biases2 = [8.8, 4.6, 5.3]
40
41
42 # output
43 # menghitung layer1 dengan (inputs*weight1) dan biases1
44 layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
45
46 # menghitung layer2 dengan hasil perhitungan pada layer1
47 layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
48
49 #print output layer2
50 print(layer2_outputs)

```

- Line 4 menginisialisasikan numpy ke np guna memudahkan dalam pengoperasionalkan source code ini, proses komputasi angka
- Line 9 menginisialisasikan variabel, memasukkan nilai input dengan jumlah 10 baris angka sesuai dengan yang diminta yaitu feature layer 10 dan 6 kolom angka dengan perintah batch = 6
- Line 18 memasukan nilai weight dengan 5 kolom karena sesuai dengan neuron yang diminta yaitu 5 dan 10 baris angka
- Line 25 memasukkan nilai bias sejumlah dengan banyaknya neuron yang ditetapkan yaitu 5
- Line 29 memasukkan nilai weight2 dengan jumlah weight samadengan neuron atau ordo di layer1 yaitu 5 dengan 3 kolom sesuai dengan neuron layer2 yaitu 3
- Line 34 memasukkan nilai biases sejumlah dengan banyaknya neuron yang ditetapkan di layer2 yaitu 3
- Line 38 melakukan perhitungan di layer1 dengan weight di transpose terlebih dahulu
- Line 41 melakukan perhitungan dari hasil layer1 dikali weight transpose ditambah biases2
- Line 44 untuk menampilkan hasil output dari source code tersebut

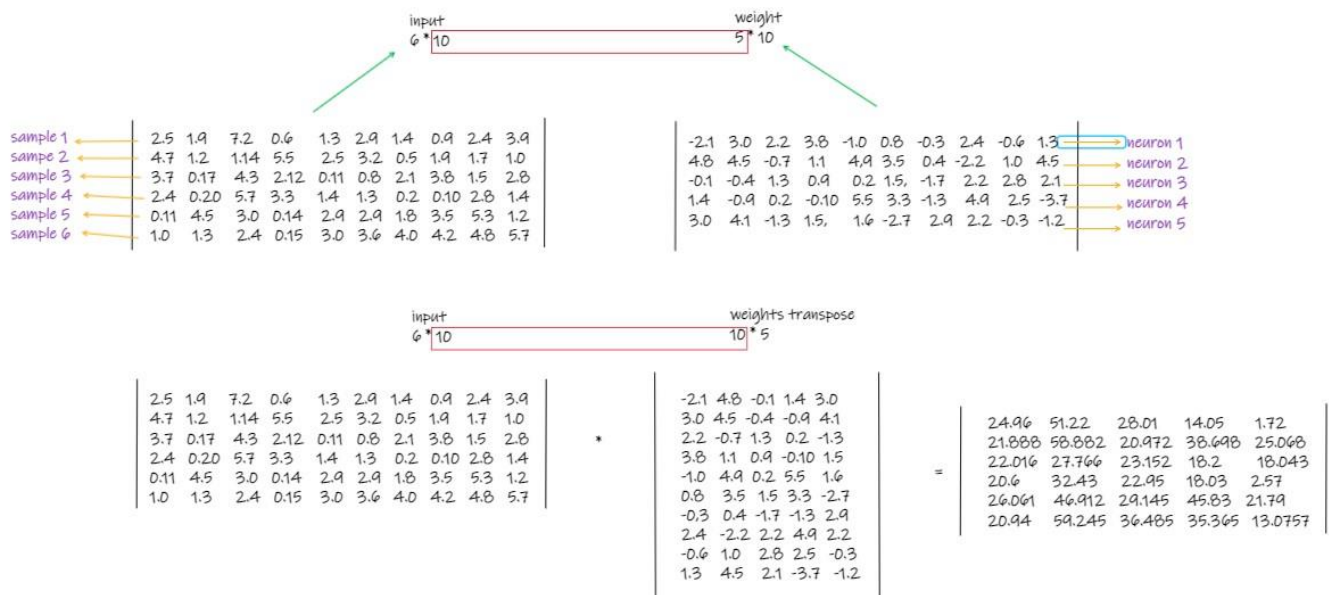
- Output

```
"C:\Users\windows_10\PycharmProjects\ots_
[[ 37541.7464  26249.975  15048.5216]
 [ 35301.0514  25369.8448  13809.4306]
 [344681.497  249778.0115 134314.086 ]
 [ 36926.5752  26380.5185  14391.0414]
 [ 79716.0324  56030.0128  31581.3696]
 [143219.6294  99837.0218  56837.6606]]

Process finished with exit code 0
```

```
# output
# menghitung layer1 dengan (inputs*weight1) dan biases1
layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
```

np.array (weight) T adalah mentranspose weight, dilakukan transpose pada weight karena untuk menyamakan ordo dengan input. Jika ordo tidak sama maka tidak akan dapat dihitung atau dioperasikan. Setelah weight di transpose maka operasi hitung dapat dilakukan dengan cara mengkali setiap baris input dengan kolom weight.



Saat hasil layer 1 sudah ditemukan dengan mengkali input dan transpose weight 1 hasil np.dot ditambah dengan biases 1

$$\begin{array}{c} \text{np.dot} \\ \left| \begin{array}{ccccc} 24.96 & 51.22 & 28.01 & 14.05 & 1.72 \\ 21.888 & 58.882 & 20.972 & 38.698 & 25.068 \\ 22.016 & 27.766 & 23.152 & 18.2 & 18.043 \\ 20.6 & 32.43 & 22.95 & 18.03 & 2.57 \\ 26.061 & 46.912 & 24.145 & 45.83 & 21.79 \\ 20.94 & 59.245 & 36.485 & 35.365 & 13.0757 \end{array} \right| \end{array} + \begin{array}{c} \text{biases} \\ \left| \begin{array}{ccccc} 0.3 & 1.9 & 2.3 & 3.5 & 1.7 \end{array} \right| \end{array} = \begin{array}{c} \left| \begin{array}{ccccc} 25.26 & 53.12 & 30.31 & 17.55 & 3.42 \\ 22.188 & 68.782 & 23.272 & 42.198 & 26.768 \\ 22.316 & 29.666 & 25.452 & 21.7 & 19.743 \\ 20.9 & 34.33 & 25.25 & 21.53 & 4.27 \\ 26.361 & 48.812 & 31.445 & 48.33 & 23.49 \\ 21.24 & 61.145 & 38.785 & 38.865 & 14.775 \end{array} \right| \end{array}$$

Menghitung hidden layer 2

```
# menghitung layer2 dengan hasil perhitungan pada layer1
layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + biases2
```

Saat hasil dari layer1 didapatkan lalu lanjut masuk ke penghitungan layer 2, mencari nilai np.dot layer 2 dengan cara hasil layer 1 dikali dengan weight transpose 2. Weights 2 di transpose untuk menyamakan ordo output hidden layer 1

$$\begin{array}{c} \text{Output Layer 1} \\ 6 \times 5 \end{array} \begin{array}{c} \text{Weights 2} \\ 3 \times 5 \end{array} \rightarrow \begin{array}{c} \left| \begin{array}{ccccc} 25.26 & 53.12 & 30.31 & 17.55 & 3.42 \\ 22.188 & 68.782 & 23.272 & 42.198 & 26.768 \\ 22.316 & 29.666 & 25.452 & 21.7 & 19.743 \\ 20.9 & 34.33 & 25.25 & 21.53 & 4.27 \\ 26.361 & 48.812 & 31.445 & 48.33 & 23.49 \\ 21.24 & 61.145 & 38.785 & 38.865 & 14.775 \end{array} \right| \end{array} \begin{array}{c} \left| \begin{array}{ccccc} -0.2 & 1.5 & 1.1 & 2.1 & -1.3 \\ 0.4 & 1.6 & 2.4 & 3.0 & 2.2 \\ 2.4 & 2.5 & 3.0 & 1.1 & 1.5 \end{array} \right| \end{array}$$

$$\begin{array}{c} \text{Output Layer 1} \\ 6 \times 5 \end{array} \begin{array}{c} \text{Weights 2 transpose} \\ 5 \times 3 \end{array} \rightarrow \begin{array}{c} \left| \begin{array}{ccccc} 25.26 & 53.12 & 30.31 & 17.55 & 3.42 \\ 22.188 & 68.782 & 23.272 & 42.198 & 26.768 \\ 22.316 & 29.666 & 25.452 & 21.7 & 19.743 \\ 20.9 & 34.33 & 25.25 & 21.53 & 4.27 \\ 26.361 & 48.812 & 31.445 & 48.33 & 23.49 \\ 21.24 & 61.145 & 38.785 & 38.865 & 14.775 \end{array} \right| \end{array} * \begin{array}{c} \left| \begin{array}{ccc} -0.2 & 0.4 & 2.4 \\ 1.5 & 1.6 & 2.5 \\ 1.1 & 2.4 & 3.0 \\ 2.1 & 3.0 & 1.1 \\ -1.3 & 2.2 & 1.5 \end{array} \right| \end{array} = \begin{array}{c} \left| \begin{array}{ccc} 140.378 & 228.014 & 308.789 \\ 166.152 & 383.4628 & 361.592 \\ 87.9371 & 226.0114 & 257.5639 \\ 114.752 & 197.872 & 241.823 \\ 175.5913 & 363.7796 & 369.1294 \\ 192.542 & 348.512 & 385.1075 \end{array} \right| \end{array}$$

Setelah di dapatkan hasil np.dot layer 2 lalu ditambah dengan biases 2 dan hasil layer 2 didapatkan

$$\begin{array}{c} \text{np.dot layer 2} \\ \left| \begin{array}{ccc} 140.378 & 228.014 & 308.789 \\ 166.152 & 383.4628 & 361.592 \\ 87.9371 & 226.0114 & 257.5639 \\ 114.752 & 197.872 & 241.823 \\ 175.5913 & 363.7796 & 369.1294 \\ 192.542 & 348.512 & 385.1075 \end{array} \right| \end{array} + \begin{array}{c} \text{biases 2} \\ \left| \begin{array}{ccc} 2.0 & 1.0 & 0.4 \end{array} \right| \end{array} = \begin{array}{c} \left| \begin{array}{ccc} 142.378 & 229.014 & 309.189 \\ 168.152 & 384.4628 & 361.992 \\ 89.9371 & 227.0114 & 257.9639 \\ 116.752 & 198.872 & 242.223 \\ 177.5913 & 364.7796 & 369.5294 \\ 194.542 & 349.512 & 385.5075 \end{array} \right| \end{array}$$

