

KECERDASAN BUATAN

UTS Part 2

Ditulis Dalam Rangka Pemenuhan Ujian Tengah Semester Mata Kuliah Kecerdasan Buatan



Dosen Pengampu

I Gde Agung Sri Sidhimantra, S.Kom., M.Kom.

Disusun Oleh :

Aditya Ramadhan Wahyu Santoso (21091397055)

adityaramadhan.21055@mhs.unesa.ac.id

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2022

1. Single Neuron

Input :

```
1apy>...
1 #Single Neuron
2
3 #inisialisasi numpy
4 import numpy as np
5
6 #inisialisasi variabel
7 #Input layer feature 10
8 inputs = [4, 7, 3, 2, 8, 5, 6, 2, 6, 7]
9 #Neuron 1
10 weights = [0.3, 0.1, -0.4, 0.5, 0.6, 0.3, 0.2, 0.6, 0.3, -0.7]
11 #banyak bias tergantung dari berapa banyak neuron yang ada
12 bias = 6
13
14 #output dari rumus numpy
15 output = np.dot(weights, inputs) + bias
16
17 #print out
18 print(output)
```

Output :

```
18.1
PS F:\Vscode\AI> []
```

Analisis :

Metode Penghitungan Menggunakan Numpy sebagai np. Setelah memasukkan metode penghitungan selanjutnya memasukkan variable untuk inputs, weights, dan bias sesuai dengan ketentuan yang ada di soal yaitu dengan inputs 10*1, weights 1*10, neuron 1, dan bias 1. Selanjutnya membuat output untuk menghitung variable tersebut dengan mengalikan weights dan inputs lalu hasil dari perkalian weights dan inputs tersebut dijumlahkan dengan bias. Untuk terakhir membuat print out dengan fungsi print untuk melihat hasil dari penghitungan tersebut.

2. Multi Neuron

Input :

```
1b.py>...
1 #Multi Neuron
2
3 #inisialisasi numpy
4 import numpy as np
5
6 #inisialisasi variabel
7 #Input layer feature 10
8 inputs = [2.0, 0.4, 2.0, 0.43, -4.0, 0.22, 4.0, 7.0, 4.0, 11.0]
9
10 #Neuron 5
11 weights = [[3.0, 9.0, 3.0, 0.4, 0.7, 0.4, 0.2, 0.4, 0.9, -0.1],
12 [0.12, 0.24, 0.54, 0.2, 0.8, 0.25, -2.0, 0.33, 0.89, 0.46],
13 [0.32, 0.5, 0.3, 0.23, 0.24, -0.29, -0.46, 0.78, 0.99, -0.1],
14 [2.0, 0.21, 0.3, 7.0, 6.0, 2, 6, 0.55, 0.33, 0.22],
15 [1.0, 0.2, -2.0, 2.0, -0.2, -0.47, 1.0, -0.7, 0.4, -0.36]]
16
17 #banyak bias tergantung dari berapa banyak neuron yang ada
18 biases = [7.0, 6.0, 3.0, 4.0, 2.0]
19
20 #output dari rumus numpy
21 output = np.dot(weights, inputs) + biases
22
23 #print out
24 print(output)
```

Output :

```
[21.018  2.2915  9.3953 81.095  49.5412]
PS F:\Vscode\AI> []
```

Analisis:

Metode Penghitungan Menggunakan Numpy sebagai np. Setelah memasukkan metode penghitungan selanjutnya memasukkan variable untuk inputs, weights, dan bias sesuai

dengan ketentuan yang ada di soal yaitu dengan inputs 10, weights 5*10, neuron 5, dan bias 5. Selanjutnya membuat output untuk menghitung variable tersebut dengan mengalikan weights dan inputs lalu hasil dari perkalian weights dan inputs tersebut dijumlahkan dengan bias. Untuk terakhir membuat print out dengan fungsi print untuk melihat hasil dari penghitungan tersebut.

3. Multi Neuron Batch Input

Input :

```
1 # -*- coding: utf-8 -*-
2 #initia Neuron Batch Input
3 #inisialisasi numpy
4 import numpy as np
5
6 #inisialisasi variabel
7 #Input layer feature 10
8 #per batch nya 6 input
9 inputs = [[2.0, 3.0, 4.0, 8.0, 6.0, 12.0],
10 [2.0, 4.0, 1.4, 1.0, 0.4, 7.0],
11 [1.0, 0.3, 6.0, 0.38, -3.0, 0.22],
12 [-1.0, 2.0, 5.0, 5.0, 4.0, 0.3],
13 [0.1, 4.0, 7.0, 0.33, -7.0, 0.43],
14 [0.22, -1.0, 0.21, 6.0, 0.56, 0.2],
15 [-0.25, -0.3, 1.5, 0.2, 1.0, 5.0],
16 [-2.0, 8.0, 0.2, 0.3, 8.0, 6.0],
17 [0.1, -7.0, 3.0, -0.15, 0.75, 5.0],
18 [1.0, 2.0, 6.0, 4.0, 3.0, -2.0]]
19
20 #Neuron 5
21 weights = [[0.3, 0.4, 0.3, 2.1, 0.3, -0.2],
22 [-0.2, 1.0, 0.41, 0.0, -0.5, 1.0],
23 [3.0, -1.5, 4.0, 0.35, 0.28, -0.75],
24 [4.0, 4.5, -1.0, 4.0, 7.0, 0.1],
25 [5.0, 1.0, -6.5, 0.85, -3.0, -0.46]]
26
27 #Banyak bias tergantung dari berapa banyak neuron yang ada
28 biases = [2.0, 8.0, 3.0, 4.0, 2.0]
29
30 #output dari rumus numpy
31 output = np.dot(inputs, np.array(weights).T) + biases
32
33 #print out
34 print(output)
```

Output :

```
[[ 25.2   39.19  25.15  107.9   28.03 ]
 [ 15.35  35.085 30.225  58.25   61.88 ]
 [  9.828   5.528 29.357  26.573  31.9768]
 [ 29.52   35.47  -0.39  65.78   2.95 ]
 [  7.782  11.318 13.686 -11.312  25.279 ]
 [ 28.026  25.9337 12.17  75.82   1.554 ]
 [ 10.64   5.3925 13.3   22.52  -8.427 ]
 [ 10.93   11.767 -13.23  22.15 -34.859 ]
 [  4.59   6.975  12.1075 12.4   -6.7825]
 [ 19.5    17.79  28.45   60.3   47.98 ]]
```

Analisis :

Metode Penghitungan Menggunakan Numpy sebagai np. Setelah memasukkan metode penghitungan selanjutnya memasukkan variable untuk inputs, batch, weights, dan bias sesuai dengan ketentuan yang ada di soal yaitu dengan inputs 10, Batch 6 (Inputs menjadi matrix 6*10), weights 5*10, neuron 5, dan bias 5. Selanjutnya membuat output untuk menghitung variable tersebut dengan fungsi np.dot untuk mengalikan weights dan inputs lalu mentranspose keduanya kemudian ditambahkan dengan biases. Untuk terakhir membuat print out dengan fungsi print untuk melihat hasil dari penghitungan tersebut.

4. Multiple neuron batch and multiple layer

Input :

```

1  #multiple neuron batch and multiple layer
2
3  #inisialisasi numpy
4  import numpy as np
5
6  #inisialisasi variabel
7  #Input layer feature 10
8  #per batch nya 6 input
9  inputs = [[6.0, 1.5, 3.4, 2.7, 7.4, 2.4, 8.0, 4.9, 8.1, 1.7],
10 [6.3, 7.8, 6.7, 5.9, 1.3, 5.3, 2.9, 1.4, 3.9, 1.3],
11 [2.1, 1.1, 4.3, 5.1, 3.6, 7.2, 3.0, 6.0, 5.8, 7.0],
12 [0.7, 9.1, 2.3, 1.3, 8.6, 4.8, 1.3, 2.7, 6.2, 3.1],
13 [1.5, 7.3, 2.5, 4.0, 1.0, 4.3, 7.5, 2.8, 2.0, 6.2],
14 [0.6, 0.1, 0.8, 5.0, 3.4, 2.2, 2.9, 9.0, 1.3, 9.7]]
15
16 #Neuron 5 (layer1)
17 weights1 = [[0.3, 1.5, -3.1, 4.1, 1.37, -3.32, 7.5, -3.0, 4.8, -1.6],
18 [1.2, -1.0, 6.5, 3.1, -6.4, 4.1, -0.7, 0.9, -1.6, 1.5],
19 [-6.51, -6.0, 1.5, 4.5, -1.1, -6.2, 8.3, -3.5, 7.0, 1.0],
20 [-8.05, 2.15, -1.58, -4.35, 7.85, -4.0, 2.5, 0.1, -0.17, 0.34],
21 [0.5, 1.0, 7.4, 1.5, 4.1, -2.5, 6.1, 5.1, -5.2, 0.85]]
22
23 #banyak bias tergantung dari berapa banyak neuron pada weight1 yang ada
24 bias1 = [1.0, 0.45, 5.15, 1.5, 7.0]
25
26
27
28
29
30
31 #Neuron 3 (layer2)
32 weights2 = [[1.53, 4.15, 5.45, 1.7, 1.45],
33 [6.0, 1.37, 2.5, 5.0, 9.5],
34 [1.12, 7.82, 8.0, 6.0, 4.5]]
35
36 #banyak bias tergantung dari berapa banyak neuron pada weight2 yang ada
37 bias2 = [-5.5, 3, 2]
38
39
40
41 #output dari rumus numpy
42 #menghitung layer1
43 layer1_outputs = np.dot(inputs, np.array(weights1).T) + bias1
44
45 #menghitung layer2 berdasarkan hasil dari perhitungan dari layer1
46 layer2_outputs = np.dot(layer1_outputs, np.array(weights2).T) + bias2
47
48 #print output layer2
49 print(layer2_outputs)

```

Output :

```

[[ 555.57335 1638.5725  962.6017 ]
 [ 148.93105  577.783   231.0007 ]
 [ 388.46389  702.0386  713.79496]
 [ -91.85602  858.8582  110.72542]
 [ 456.71807 1245.1305  875.28688]
 [ 354.02737 1010.7724  809.13978]]
PS F:\Vscode\AI>

```

Analisis :

Metode Penghitungan Menggunakan Numpy sebagai np. Setelah memasukkan metode penghitungan selanjutnya memasukkan variable untuk inputs, batch, neuron, weights1, weight2, bias1 dan bias2 sesuai dengan ketentuan yang ada di soal yaitu dengan inputs 10, Batch 6 (Inputs menjadi matrix 6*10), weights1 5*10, weight2 3*5, neuron 5, bias1 5 dan bias2 3. Selanjutnya membuat output untuk menghitung variable tersebut dengan fungsi np.dot untuk mengalikan weights1 dan inputs lalu mentranspose keduanya kemudian ditambahkan dengan bias, kemudian mengalikan hasil dari perhitungan tersebut dan inputs lalu mentranspose keduanya kemudian ditambahkan dengan bias. Untuk terakhir membuat print out hasil dengan fungsi print(layer2) untuk melihat hasil dari penghitungan tersebut.