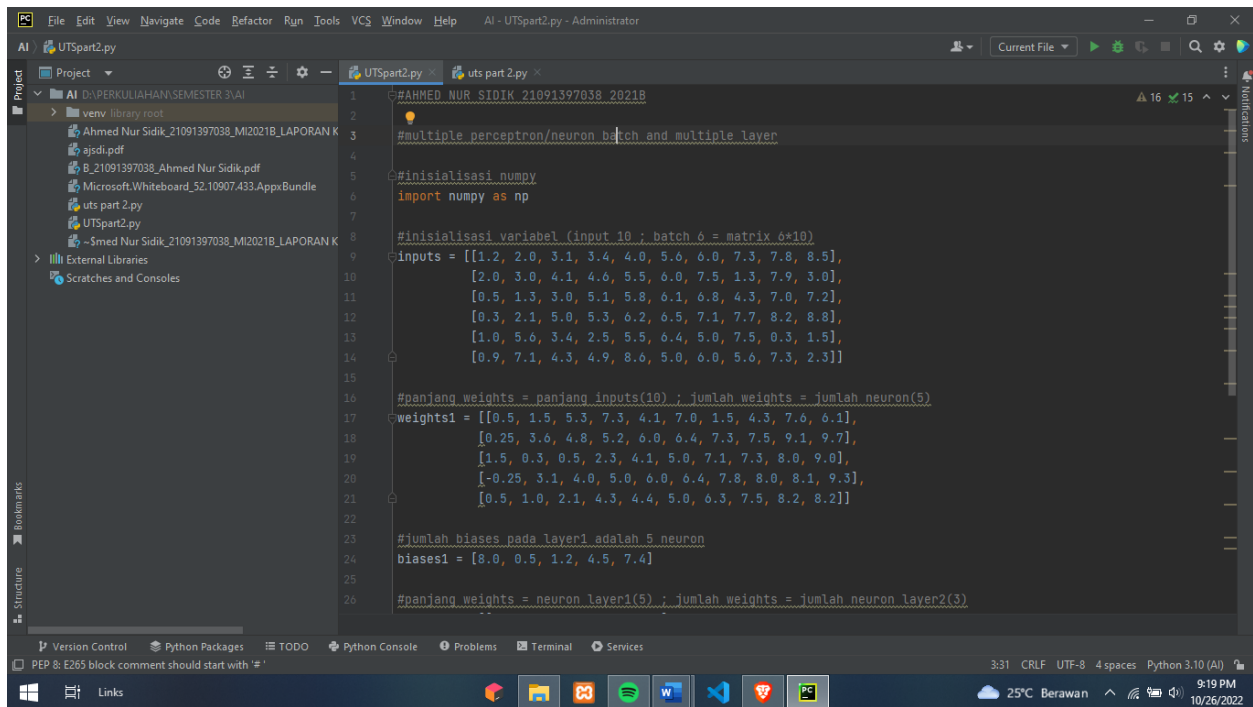


LAPORAN KECERDASAN BUATAN
UJIAN TENGAH SEMESTER PART 2
2022

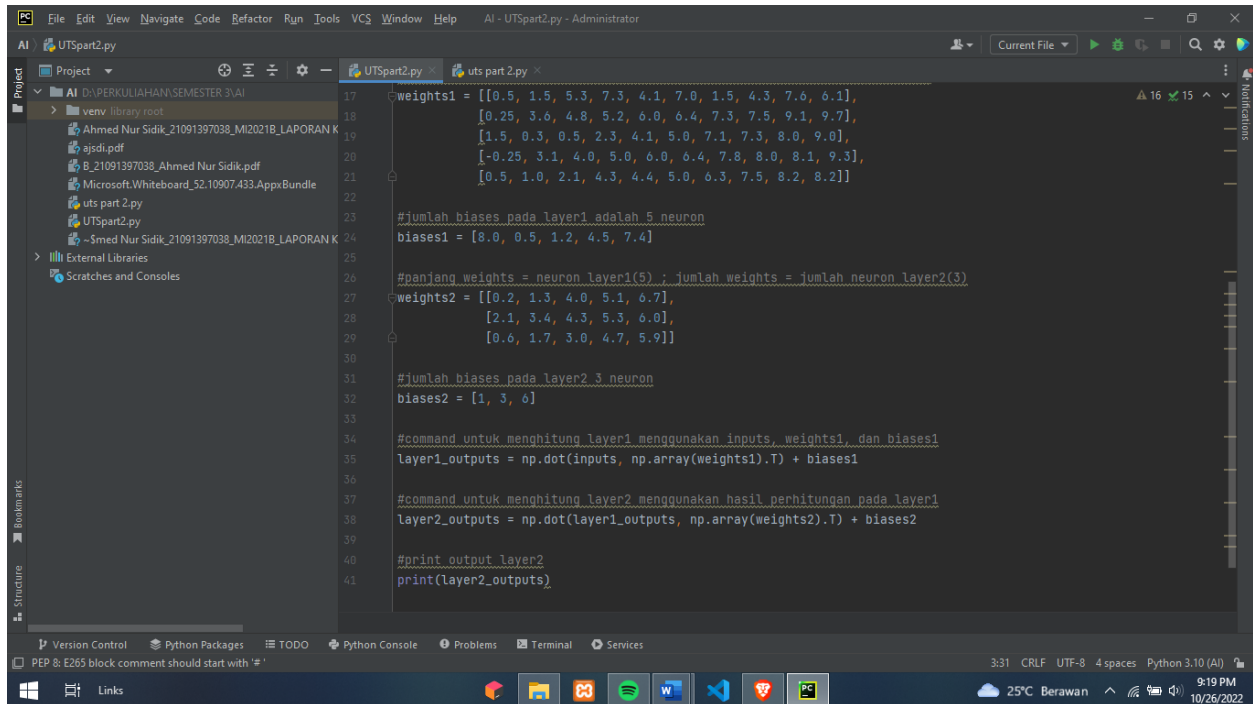


Oleh :
Ahmed Nur Sidik
(21091397038)
D4 Manajemen Informatika
Fakultas Vokasi

1. A. source code



```
1  #AHMED_NUR_SIDIK_21091397038_2021B
2
3  #multiple perceptron/neuron batch and multiple layer
4
5  #inisialisasi numpy
6  import numpy as np
7
8  #inisialisasi variabel (input 10 ; batch 6 = matrix 6*10)
9  inputs = [[1.2, 2.0, 3.1, 3.4, 4.0, 5.0, 6.0, 7.3, 7.8, 8.5],
10           [2.0, 3.0, 4.1, 4.6, 5.5, 6.0, 7.5, 1.3, 7.9, 3.0],
11           [0.5, 1.3, 3.0, 5.1, 5.8, 6.1, 6.8, 4.3, 7.0, 7.2],
12           [0.3, 2.1, 5.0, 5.3, 6.2, 6.5, 7.1, 7.7, 8.2, 8.8],
13           [1.0, 5.6, 3.4, 2.5, 5.5, 6.4, 5.0, 7.5, 0.3, 1.5],
14           [0.9, 7.1, 4.3, 4.9, 8.6, 5.0, 6.0, 5.6, 7.3, 2.3]]
15
16  #panjang weights = panjang inputs(10) ; jumlah weights = jumlah neuron(5)
17  weights1 = [[0.5, 1.5, 5.3, 7.3, 4.1, 7.0, 1.5, 4.3, 7.6, 6.1],
18             [0.25, 3.6, 4.8, 5.2, 6.0, 6.4, 7.3, 7.5, 9.1, 9.7],
19             [1.5, 0.3, 0.5, 2.3, 4.1, 5.0, 7.1, 7.3, 8.0, 9.0],
20             [-0.25, 3.1, 4.0, 5.0, 6.0, 6.4, 7.8, 8.0, 8.1, 9.3],
21             [0.5, 1.0, 2.1, 4.3, 4.4, 5.0, 6.3, 7.5, 8.2, 8.2]]
22
23  #jumlah biases pada layer1 adalah 5 neuron
24  biases1 = [8.0, 0.5, 1.2, 4.5, 7.4]
25
26  #panjang weights = neuron layer1(5) ; jumlah weights = jumlah neuron layer2(3)
```



```
17  weights1 = [[0.5, 1.5, 5.3, 7.3, 4.1, 7.0, 1.5, 4.3, 7.6, 6.1],
18             [0.25, 3.6, 4.8, 5.2, 6.0, 6.4, 7.3, 7.5, 9.1, 9.7],
19             [1.5, 0.3, 0.5, 2.3, 4.1, 5.0, 7.1, 7.3, 8.0, 9.0],
20             [-0.25, 3.1, 4.0, 5.0, 6.0, 6.4, 7.8, 8.0, 8.1, 9.3],
21             [0.5, 1.0, 2.1, 4.3, 4.4, 5.0, 6.3, 7.5, 8.2, 8.2]]
22
23  #jumlah biases pada layer1 adalah 5 neuron
24  biases1 = [8.0, 0.5, 1.2, 4.5, 7.4]
25
26  #panjang weights = neuron layer1(5) ; jumlah weights = jumlah neuron layer2(3)
27  weights2 = [[0.2, 1.3, 4.0, 5.1, 6.7],
28             [2.1, 3.4, 4.3, 5.3, 6.0],
29             [0.6, 1.7, 3.0, 4.7, 5.9]]
30
31  #jumlah biases pada layer2 3 neuron
32  biases2 = [1, 3, 6]
33
34  #command untuk menghitung layer1 menggunakan inputs, weights1, dan biases1
35  layer1_outputs = np.dot(inputs, np.array(weights1).T) + biases1
36
37  #command untuk menghitung layer2 menggunakan hasil 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

```
Run: UTSpant2
"D:\PERKULIAHAN\SEMESTER 3\AI\venv\Scripts\python.exe" "D:\PERKULIAHAN\SEMESTER 3\AI\UTSpant2.py"
[[5479.483 6660.428 5056.083 ]
 [4382.983 5388.703 4066.843 ]
 [5164.548 6304.7195 4775.552 ]
 [6254.488 7633.9945 5782.998 ]
 [3512.571 4298.762 3260.509 ]
 [4880.413 6003.0625 4534.285 ]]
Process finished with exit code 0
Version Control Run Python Packages TODO Python Console Problems Terminal Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't s... (36 minutes ago) 331 CRLF UTF-8 4 spaces Python 3.10 (AI)
```

Penghitungan layer 1

Perhitungan dot product

$$\begin{array}{c} \text{weight} \\ 10 \times 5 \end{array} \begin{bmatrix} 0.5 & 1.5 & 5.3 & 7.3 & 4.1 & 7.0 & 1.5 & 4.3 & 7.6 & 6.1 \\ 0.25 & 3.6 & 4.8 & 5.2 & 6.0 & 6.4 & 7.3 & 7.5 & 9.1 & 9.7 \\ 1.5 & 0.3 & 0.5 & 2.3 & 4.1 & 5.0 & 7.1 & 7.3 & 8.0 & 9.0 \\ -0.25 & 3.1 & 4.0 & 5.0 & 6.0 & 6.4 & 7.8 & 8.0 & 8.1 & 9.3 \\ 0.5 & 1.0 & 2.1 & 4.3 & 4.4 & 5.0 & 6.3 & 7.5 & 8.2 & 8.2 \end{bmatrix} \begin{array}{c} \text{inputs} \\ 6 \times 10 \end{array} \begin{bmatrix} 1.2 & 2.0 & 3.1 & 3.4 & 4.0 & 5.6 & 6.0 & 7.3 & 7.8 & 8.5 \\ 2.0 & 3.0 & 4.1 & 4.6 & 5.5 & 6.0 & 7.5 & 1.3 & 7.9 & 3.0 \\ 0.5 & 1.3 & 3.0 & 5.1 & 5.8 & 6.1 & 6.8 & 4.3 & 7.0 & 7.2 \\ 0.3 & 2.1 & 5.0 & 5.3 & 6.2 & 6.5 & 7.1 & 7.7 & 8.2 & 8.8 \\ 1.0 & 5.6 & 3.4 & 2.5 & 5.5 & 6.4 & 5.0 & 7.5 & 0.3 & 1.5 \\ 0.9 & 7.1 & 4.3 & 4.9 & 8.6 & 5.0 & 6.0 & 5.6 & 7.3 & 2.3 \end{bmatrix}$$

$$\text{weight} \times \text{inputs} = \begin{bmatrix} 251.97 & 351.88 & 290.96 & 342.57 & 295.54 \\ 220.54 & 291.79 & 222.02 & 280.39 & 232.97 \\ 247.62 & 334.995 & 269.12 & 326.345 & 277.33 \\ 299.17 & 407.555 & 325.11 & 396.975 & 337.2 \\ 163.7 & 233.72 & 171.33 & 232.55 & 182.7 \\ 242.51 & 330.045 & 239.74 & 319.205 & 259.01 \end{bmatrix}$$

kemudian np.dot + bias

$$\begin{bmatrix} 251.97 & 351.88 & 290.96 & 342.57 & 295.54 \\ 220.54 & 291.79 & 222.02 & 280.39 & 232.97 \\ 247.62 & 334.995 & 269.12 & 326.345 & 277.33 \\ 299.17 & 407.555 & 325.11 & 396.975 & 337.2 \\ 163.7 & 233.72 & 171.33 & 232.55 & 182.7 \\ 242.51 & 330.045 & 239.74 & 319.205 & 259.01 \end{bmatrix}$$

$$+ [8.0, 0.5, 1.2, 4.5, 7.4]$$

$$= \begin{bmatrix} 259.97 & 352.38 & 292.16 & 347.07 & 302.94 \\ 228.54 & 292.29 & 223.22 & 284.89 & 240.37 \\ 255.62 & 335.495 & 270.32 & 330.845 & 284.73 \\ 307.17 & 408.055 & 326.31 & 401.475 & 344.6 \\ 171.7 & 234.22 & 172.53 & 237.05 & 190.1 \\ 250.51 & 330.545 & 240.94 & 323.705 & 266.41 \end{bmatrix}$$

Perhitungan layer 2

Perhitungan dot product

$$\begin{array}{c} \text{Wight2} \\ 5 \times 3 \end{array} \begin{bmatrix} 0.2 & 1.3 & 4.0 & 5.1 & 6.7 \\ 2.1 & 3.4 & 4.3 & 5.3 & 6.0 \\ 0.6 & 1.7 & 3.0 & 4.7 & 5.9 \end{bmatrix} * \begin{array}{c} \text{Output layer1} \\ 5 \times 6 \end{array} \begin{bmatrix} 259.97 & 352.38 & 292.16 & 347.07 & 302.94 \\ 228.54 & 292.29 & 223.22 & 284.89 & 240.37 \\ 255.62 & 335.495 & 270.32 & 330.845 & 284.73 \\ 307.17 & 408.055 & 326.31 & 401.475 & 344.6 \\ 171.7 & 234.22 & 172.53 & 237.05 & 190.1 \\ 250.51 & 330.545 & 240.94 & 323.705 & 266.41 \end{bmatrix}$$

$$\text{Wight2} * \text{output layer1} = \begin{bmatrix} 5478.483 & 6657.428 & 5050.083 \\ 4381.983 & 5385.703 & 4060.843 \\ 5163.548 & 6301.7195 & 4769.552 \\ 6253.488 & 7630.9945 & 5776.998 \\ 3511.571 & 4295.762 & 3254.509 \\ 4879.413 & 6000.0625 & 4528.285 \end{bmatrix}$$

kemudian np.dot + biases2

$$\begin{bmatrix} 5478.483 & 6657.428 & 5050.083 \\ 4381.983 & 5385.703 & 4060.843 \\ 5163.548 & 6301.7195 & 4769.552 \\ 6253.488 & 7630.9945 & 5776.998 \\ 3511.571 & 4295.762 & 3254.509 \\ 4879.413 & 6000.0625 & 4528.285 \end{bmatrix}$$

$$+ [1, 3, 6]$$

$$= \begin{bmatrix} 5479.483 & 6660.428 & 5056.083 \\ 4382.983 & 5388.703 & 4066.843 \\ 5164.548 & 6304.7195 & 4775.552 \\ 6254.488 & 7633.9945 & 5782.998 \\ 3512.571 & 4298.762 & 3260.509 \\ 4880.413 & 6003.0625 & 4534.285 \end{bmatrix}$$