

LAPORAN KECERDASAN BUATAN

“Ujian Tengah Semester”



Oleh :

MOCH. EMIL IDAD

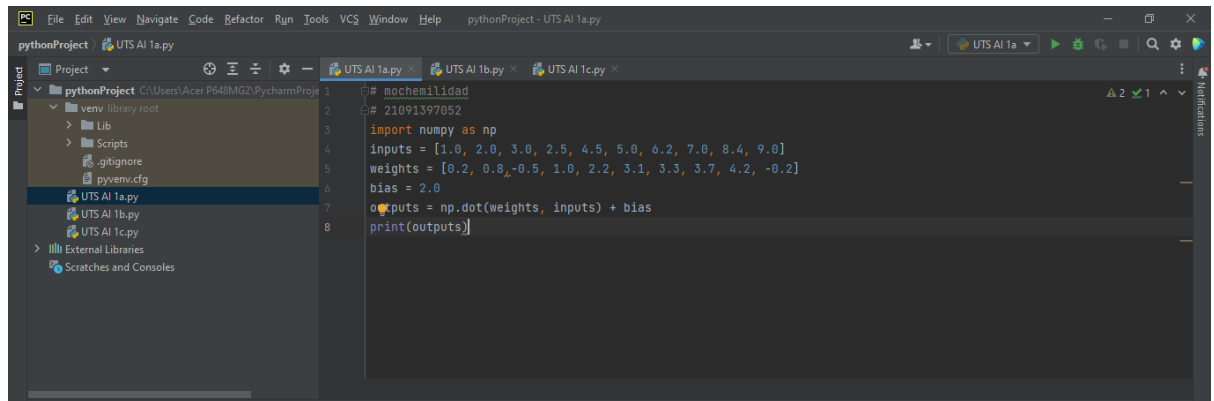
21091397052

UNIVERSITAS NEGERI SURABAYA

D4 MANAJEMEN INFORMATIKA

2022

1. A- single neuron
 - Input layer feature 10
 - Neuron 1



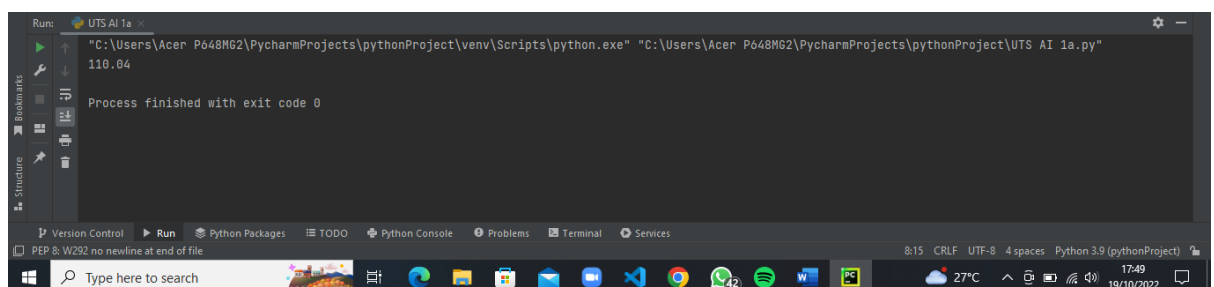
```
1 # mochemilidad
2 # 21091397052
3 import numpy as np
4 inputs = [1.0, 2.0, 3.0, 2.5, 4.5, 5.0, 6.2, 7.0, 8.4, 9.0]
5 weights = [0.2, 0.8, -0.5, 1.0, 2.2, 3.1, 3.3, 3.7, 4.2, -0.2]
6 bias = 2.0
7 outputs = np.dot(weights, inputs) + bias
8 print(outputs)
```

Pada baris ke 6 terdapat sintaks untuk mengimpor library dari python bernama numpy dan diberi insial np untuk memudahkan saat nanti ingin digunakan

Pada baris ke 10 terdapat input yang memiliki 1 baris layer input yang berjumlah 10 angka. Pada baris ke 8 terdapat variable weights(neuron) yng memiliki 1 neuron. Pada baris ke 9 terdapat bias yang nantinya diginakan pada baris ke 10

Varial outputs pada baris ke 10 terdapat dot dan dilakukan pada setiap elemen pada input dan tiap weight yang terhubung dengan input dan ditambahkan dengan bias. Np.dot yang berfungsi untuk mengalihkan dua matriks antara variable inputs dan weight. Bias sendiri berfungsi sebagai haidden layer dan output layer yang bertujuan untuk tambahan “input”. Kemudian hasilnya akan dicetak di konsol melalui fungsi print baris ke 11

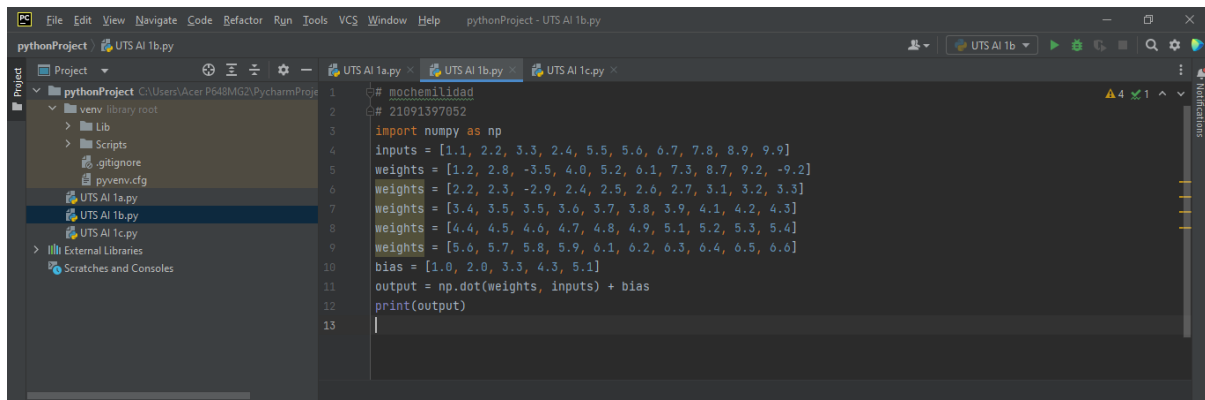
Hasil



```
Run UTS AI 1a
"C:\Users\Acer P648M62\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:\Users\Acer P648M62\PycharmProjects\pythonProject\UTS AI 1a.py"
110.04
Process finished with exit code 0
```

1. B- multi neuron

- Inputs layer feature 10
- Neuron 5



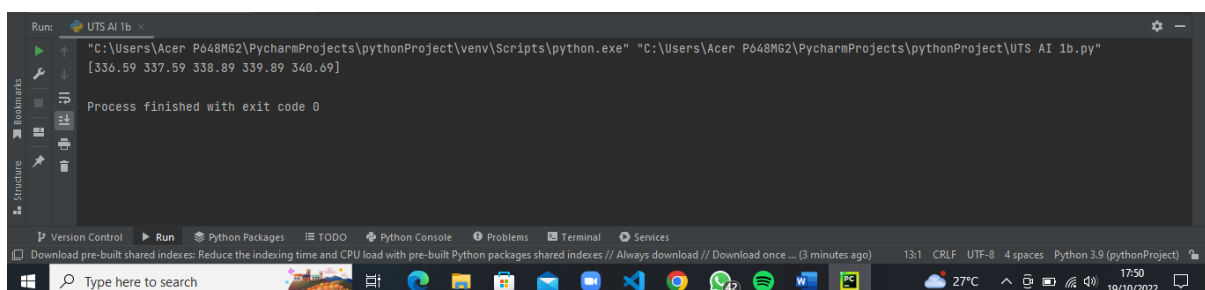
```
1 # mochemilidad
2 # 21091397052
3 import numpy as np
4 inputs = [1.1, 2.2, 3.3, 2.4, 5.5, 5.6, 6.7, 7.8, 8.9, 9.9]
5 weights = [1.2, 2.8, -3.5, 4.0, 5.2, 6.1, 7.3, 8.7, 9.2, -9.2]
6 weights = [2.2, 2.3, -2.9, 2.4, 2.5, 2.6, 2.7, 3.1, 3.2, 3.3]
7 weights = [3.4, 3.5, 3.5, 3.6, 3.7, 3.8, 3.9, 4.1, 4.2, 4.3]
8 weights = [4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5.1, 5.2, 5.3, 5.4]
9 weights = [5.6, 5.7, 5.8, 5.9, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6]
10 bias = [1.0, 2.0, 3.3, 4.3, 5.1]
11 output = np.dot(weights, inputs) + bias
12 print(output)
13
```

Pada baris ke 6 terdapat sintaks untuk mengimpor library dari python bernama numpy dan diberi np untuk memudahkan saat nanti ingin digunakan.

Pada baris ke 7 terdapat input yang memiliki 1 baris layer input yang berjumlah 10 angka. Pada baris ke 8-12 terdapat baris variable weights yang memiliki 6 neuron nya memiliki 10 layer. Jadi kita menggunakan multi neuron maka kita harus membungkusnya dengan array lagi. Jadi keenam neuron tersebut terbungkus dalam sebuah array. Jadi bisa dikatakan weights menampung array dua dimensi. Pada baris ke 8 bias yang nantinya digunakan pada baris ke 12.

Variable outputs pada baris ke 14 terdapat operasi dot dan dilakukan pada setiap elemen pada input dan tiap weight yang terhubung dengan input dan ditambahkan dengan bias. Np.dot yang berfungsi untuk mengalihkan dua matriks antara variable inputs dan weights. Bias sendiri berfungsi sebagai hidden layer dan output layer yang bertujuan untuk tambahan "input". Kemudian hasilnya akan dicetak melalui fungsi baris ke 15

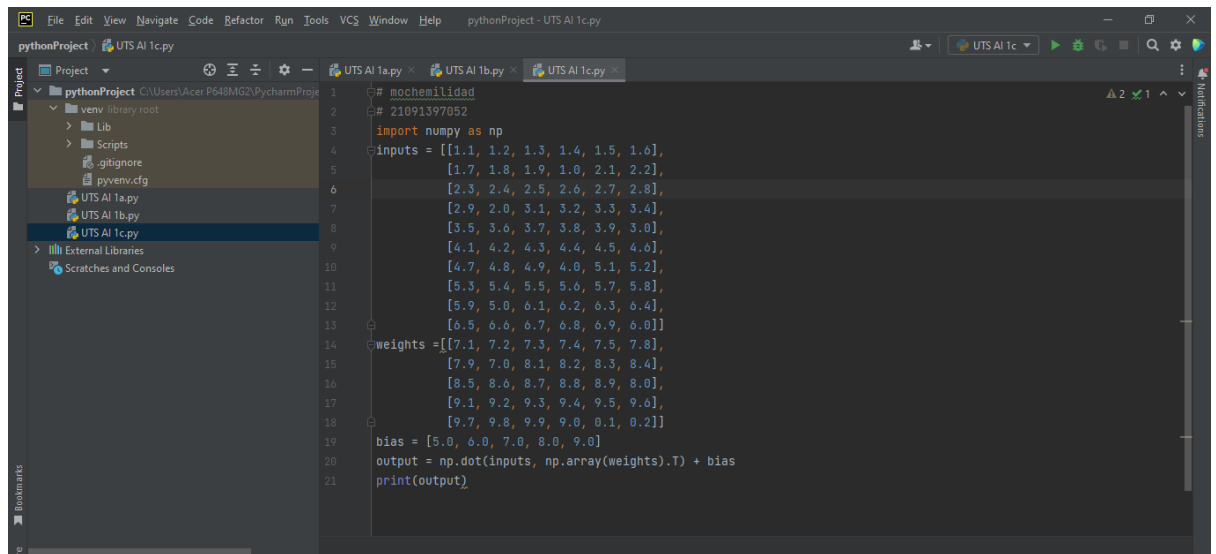
Hasil



```
Run: UTS AI 1b x
"C:\Users\Acer P648M62\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:\Users\Acer P648M62\PycharmProjects\pythonProject\UTS AI 1b.py"
[336.59 337.59 338.89 339.89 340.69]
Process finished with exit code 0
```

1. C- multi neuron batch input

- Inputs layer feature 10
- Neuron



```
1 # mochemilidad
2 # 21091397052
3 import numpy as np
4 inputs = [[1.1, 1.2, 1.3, 1.4, 1.5, 1.6],
5           [1.7, 1.8, 1.9, 1.0, 2.1, 2.2],
6           [2.3, 2.4, 2.5, 2.6, 2.7, 2.8],
7           [2.9, 2.0, 3.1, 3.2, 3.3, 3.4],
8           [3.5, 3.6, 3.7, 3.8, 3.9, 3.0],
9           [4.1, 4.2, 4.3, 4.4, 4.5, 4.6],
10          [4.7, 4.8, 4.9, 4.0, 5.1, 5.2],
11          [5.3, 5.4, 5.5, 5.6, 5.7, 5.8],
12          [5.9, 5.0, 6.1, 6.2, 6.3, 6.4],
13          [6.5, 6.6, 6.7, 6.8, 6.9, 6.0]]
14 weights = [[7.1, 7.2, 7.3, 7.4, 7.5, 7.8],
15            [7.9, 7.0, 8.1, 8.2, 8.3, 8.4],
16            [8.5, 8.6, 8.7, 8.8, 8.9, 8.0],
17            [9.1, 9.2, 9.3, 9.4, 9.5, 9.6],
18            [9.7, 9.8, 9.9, 9.0, 0.1, 0.2]]
19 bias = [5.0, 6.0, 7.0, 8.0, 9.0]
20 output = np.dot(inputs, np.array(weights).T) + bias
21 print(output)
```

Pada baris ke 5 terdapat sintaks untuk mengimpor library dari python bernama numpy dan diberi untuk memudahkan saat nanti ingin digunakan

Pada baris ke 6 terdapat variable dengan multiple batch yang berjumlah 6 batch. Pada baris ke 17-21 terdapat variable weights yang memiliki 5 neuron yang 1 neuron nya memiliki layer. Jika kita menggunakan multi neuron, maka kita harus membungkusnya dengan array lagi. Jadi keenak neuron tersebut terbungkus dalam sebuah array. Jadi bias dikatakan weights menampung array dua dimensi. Dari baris ke 22 terdapat bias yang nantinya digunakan pada baris 21. Jumlah angka pada bias, tergantung banyaknya neuron yang kita punya.

Variable output pada baris ke 23 terdapat operasi dot dan dilakukan pada setiap elemen pada input dan tiap weight yang terhubung dengan input dan ditambahkan dengan bias. Variable weight dibungkus dengan fungsi np.array karena weight merupakan array dua dimensi disini. Np.dot yang berfungsi untuk mengalihkan dua matriks antara variable input dan weight. Bias sendiri berfungsi sebagai hidden layer dan output layer yang bertujuan untuk tambahan "input". Kemudian hasilnya akan dicetak di konsol melalui fungsi print baris ke 24

Hasil

```
Run: UTS AI 1c
"C:\Users\Acer P648MG2\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:\Users\Acer P648MG2\PycharmProjects\pythonProject\UTS AI 1c.py"
[[ 45.03  70.99  76.45  83.91  57.37]
 [ 84.21  91.53  98.55 108.17  71.59]
 [118.19 128.47 138.29 151.23 163.81]
 [137.57 150.21 168.55 175.69 117.23]
 [163.55 177.55 192.05 208.95 150.05]
 [197.93 214.69 230.95 252.21 173.47]
 [217.11 235.23 253.05 276.47 187.69]
 [251.09 272.17 292.75 319.53 219.91]
 [278.47 293.91 315.05 343.99 233.33]
 [296.45 321.25 346.55 377.25 266.15]]

Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once ... (3 minutes ago)

16:1 CRLF UTF-8 4 spaces Python 3.9 (pythonProject)

Type here to search

27°C 17:51 19/10/2022