

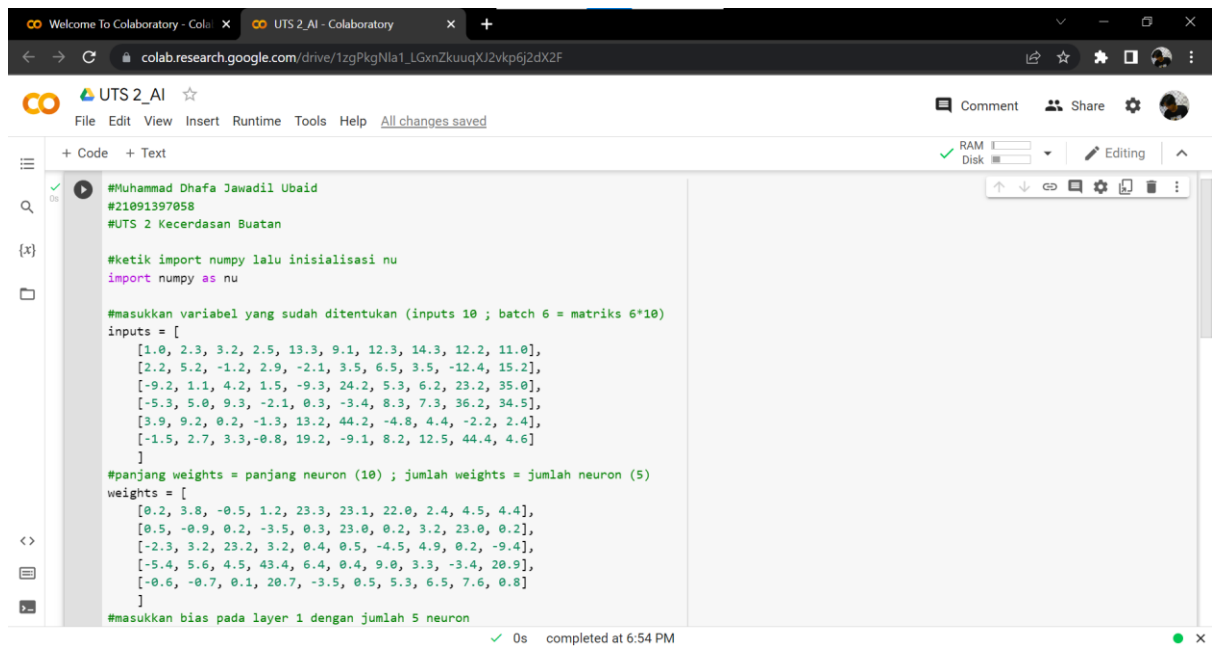
UTS 2
Kecerdasan Buatan



Disusun oleh :
Muhammad Dhafa Jawadil Ubaid
21091397058

Program Studi D4 Manajemen Informatika
Fakultas Vokasi
Universitas Negeri Surabaya
2022

Source Code



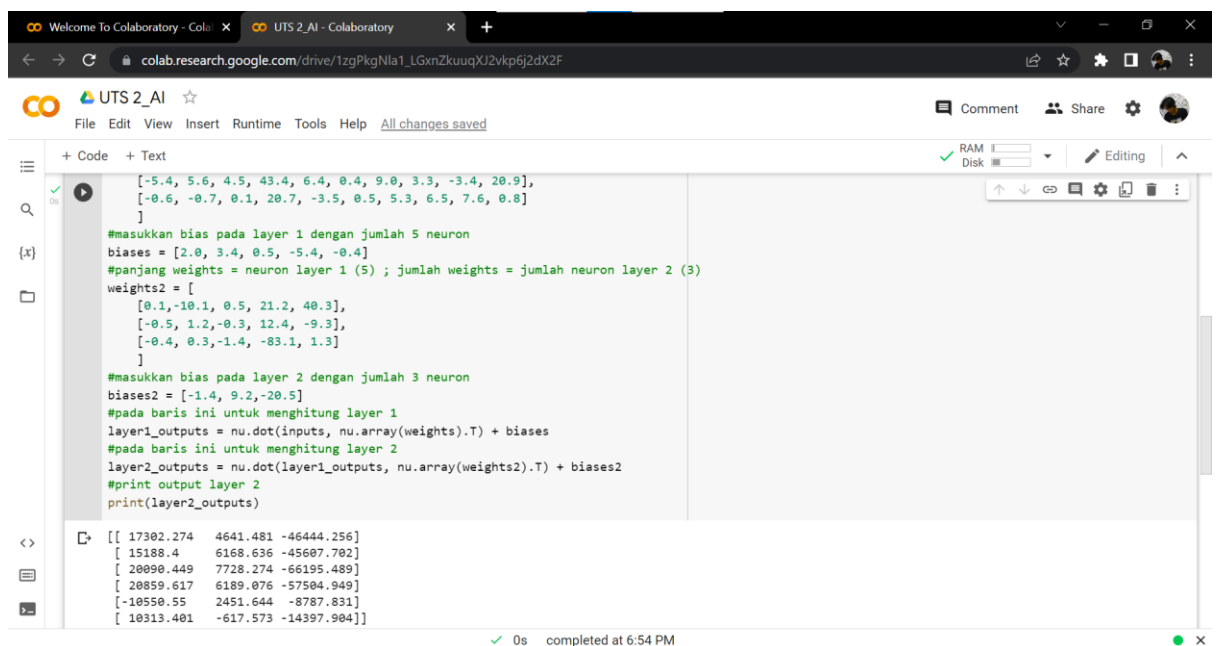
```
UTS_2_AI
Welcome To Colaboratory - Colab x UTS_2_AI - Colaboratory x +
colab.research.google.com/drive/1zgPkgNla1_LGxnZkuuqXJ2vKp6j2dX2F
Comment Share
+ Code + Text
#Muhammad Dhafa Jawadil Ubaid
#21091397058
#UTS 2 Kecerdasan Buatan

#ketik import numpy lalu inisialisasi nu
import numpy as nu

#masukkan variabel yang sudah ditentukan (inputs 10 ; batch 6 = matriks 6*10)
inputs = [
    [1.0, 2.3, 3.2, 2.5, 13.3, 9.1, 12.3, 14.3, 12.2, 11.0],
    [2.2, 5.2, -1.2, 2.9, -2.1, 3.5, 6.5, 3.5, -12.4, 15.2],
    [-9.2, 1.1, 4.2, 1.5, -9.3, 24.2, 5.3, 6.2, 23.2, 35.0],
    [-5.3, 5.0, 9.3, -2.1, 0.3, -3.4, 8.3, 7.3, 36.2, 34.5],
    [3.9, 9.2, 0.2, -1.3, 13.2, 44.2, -4.8, 4.4, -2.2, 2.4],
    [-1.5, 2.7, 3.3, -0.8, 19.2, -9.1, 8.2, 12.5, 44.4, 4.6]
]

#panjang weights = panjang neuron (10) ; jumlah weights = jumlah neuron (5)
weights = [
    [0.2, 3.8, -0.5, 1.2, 23.3, 23.1, 22.0, 2.4, 4.5, 4.4],
    [0.5, -0.9, 0.2, -3.5, 0.3, 23.0, 0.2, 3.2, 23.0, 0.2],
    [-2.3, 3.2, 23.2, 3.2, 0.4, 0.5, -4.5, 4.9, 0.2, -9.4],
    [-5.4, 5.6, 4.5, 43.4, 6.4, 0.4, 9.0, 3.3, -3.4, 20.9],
    [-0.6, -0.7, 0.1, 20.7, -3.5, 0.5, 5.3, 6.5, 7.6, 0.8]
]

#masukkan bias pada layer 1 dengan jumlah 5 neuron
```



```
UTS_2_AI
Welcome To Colaboratory - Colab x UTS_2_AI - Colaboratory x +
colab.research.google.com/drive/1zgPkgNla1_LGxnZkuuqXJ2vKp6j2dX2F
Comment Share
+ Code + Text
[-5.4, 5.6, 4.5, 43.4, 6.4, 0.4, 9.0, 3.3, -3.4, 20.9],
[-0.6, -0.7, 0.1, 20.7, -3.5, 0.5, 5.3, 6.5, 7.6, 0.8]
]

#masukkan bias pada layer 1 dengan jumlah 5 neuron
biases = [2.0, 3.4, 0.5, -5.4, -0.4]
#panjang weights = neuron layer 1 (5) ; jumlah weights = jumlah neuron layer 2 (3)
weights2 = [
    [0.1, -10.1, 0.5, 21.2, 40.3],
    [-0.5, 1.2, -0.3, 12.4, -9.3],
    [-0.4, 0.3, -1.4, -83.1, 1.3]
]

#masukkan bias pada layer 2 dengan jumlah 3 neuron
biases2 = [-1.4, 9.2, -20.5]
#pada baris ini untuk menghitung layer 1
layer1_outputs = nu.dot(inputs, nu.array(weights).T) + biases
#pada baris ini untuk menghitung layer 2
layer2_outputs = nu.dot(layer1_outputs, nu.array(weights2).T) + biases2
#print output layer 2
print(layer2_outputs)

[[ 17302.274  4641.481 -46444.256]
 [ 15188.4   6168.636 -45607.702]
 [ 20090.449  7728.274 -66195.489]
 [ 20859.617  6189.076 -57504.949]
 [-10550.55   2451.644 -8787.831]
 [ 10313.401 -617.573 -14397.904]]
```

Output

```
[[ 17302.274  4641.481 -46444.256]
 [ 15188.4   6168.636 -45607.702]
 [ 20090.449  7728.274 -66195.489]
 [ 20859.617  6189.076 -57504.949]
 [-10550.55   2451.644 -8787.831]
 [ 10313.401 -617.573 -14397.904]]
```

Penjelasan

- Mengimport numpy sebagai library dan memberi inisial nu
- Masukkan variabel untuk inputs, weights, dan bias sesuai dengan ketentuan soal
Inputs = 10
Batch = 6 } Inputs menjadi matrix 6×10
Weights1 = 5×10 Biases1 = 5
Weights2 = 3×5 Biases2 = 3
- Buatlah output untuk menghitung variabel yang sudah kita masukkan/buat nu.dot untuk menghitung vektor weight dan input kemudian hasil dari hitungan vector ditambah dengan biases
- Untuk output yang diinginkan adalah output layer2 yang berasal dari hasil perhitungan layer1 lalu dihitung lagi dengan weights2 dan biases2
- Buat command print untuk menampilkan hasil perhitungan output

Perhitungan Layer 1

Perhitungan dot product

Weights 10*5		Inputs 6*10
<div style="display: flex; justify-content: space-between;"><div>0.2, 3.8, -0.5, 1.2, 23.3, 23.1, 22.0, 2.4, 4.5, 4.4</div><div>0.5, -0.9, 0.2, -3.5, 0.3, 23.0, 0.2, 3.2, 23.0, 0.2</div><div>-2.3, 3.2, 23.2, 3.2, 0.4, 0.5, -4.5, 4.9, 0.2, -9.4</div><div>-5.4, 5.6, 4.5, 43.4, 6.4, 0.4, 9.0, 3.3, -3.4, 20.9</div><div>-0.6, -0.7, 0.1, 20.7, -3.5, 0.5, 5.3, 6.5, 7.6, 0.8</div></div>	×	<div style="display: flex; justify-content: space-between;"><div>1.0, 2.3, 3.2, 2.5, 13.3, 9.1, 12.3, 14.3, 12.2, 11.0</div><div>2.2, 5.2, -1.2, 2.9, -2.1, 3.5, 6.5, 3.5, -12.4, 15.2</div><div>-9.2, 1.1, 4.2, 1.5, -9.3, 2.4, 2, 5.3, 6.2, 23.2, 35.0</div><div>-5.3, 5.0, 9.3, -2.1, 0.3, -3.4, 8.3, 7.3, 36.2, 34.5</div><div>3.9, 9.2, 0.2, -1.3, 13.2, 44.2, -4.8, 4.4, -2.2, 2.4</div><div>-1.5, 2.7, 3.3, -0.8, 19.2, -9.1, 8.2, 12.5, 44.4, 4.6</div></div>
Weights*Inputs		
<div style="display: flex; justify-content: space-between;"><div>938.66, 534.63, 10.93, 565.45, 267.52</div><div>218.68, -203.76, -163.53, 555.55, 39.17</div><div>734.25, 1105.31, -182.53, 810.78, 353.58</div><div>454.04, 788.47, -82.99, 704.65, 348.55</div><div>1268.28, 981.82, 68.49, 106.06, -71.41</div><div>674.94, 860.5, 79.21, 182.82, 376.86</div></div>		

nu.dot + biases

$$\begin{bmatrix} 938.66, 534.63, 10.93, 565.45, 267.52 \\ 218.68, -203.76, -163.53, 555.55, 39.17 \\ 734.25, 1105.31, -182.53, 810.78, 353.58 \\ 454.04, 788.47, -82.99, 704.65, 348.55 \\ 1268.28, 981.82, 68.49, 106.06, -71.41 \\ 674.94, 860.5, 79.21, 182.82, 376.86 \end{bmatrix} + [2.0, 3.4, 0.5, -5.4, -0.4]$$

$$= \begin{bmatrix} 940.66, 538.03, 11.43, 560.05, 267.12 \\ 220.68, -200.36, -163.03, 550.15, 38.77 \\ 736.25, 1108.71, -182.03, 805.38, 353.18 \\ 456.04, 791.87, -82.49, 699.25, 348.15 \\ 1270.28, 985.22, 68.99, 100.66, -71.81 \\ 676.94, 863.9, 79.71, 177.42, 376.46 \end{bmatrix}$$

Perhitungan Layer 2

Perhitungan dot product

Weights 3*5

$$\begin{bmatrix} 0.1, -10.1, 0.5, 21.2, 40.3 \\ -0.5, 1.2, -0.3, 12.4, -9.3 \\ -0.4, 0.3, -1.4, -83.1, 1.3 \end{bmatrix}$$

Output layer 1
5*6

$$\begin{bmatrix} 940.66, 538.03, 11.43, 560.05, 267.12 \\ 220.68, -200.36, -163.03, 550.15, 38.77 \\ 736.25, 1108.71, -182.03, 805.38, 353.18 \\ 456.04, 791.87, -82.49, 699.25, 348.15 \\ 1270.28, 985.22, 68.99, 100.66, -71.81 \\ 676.94, 863.9, 79.71, 177.42, 376.46 \end{bmatrix}$$

Weights 2 * output layer 1

$$\begin{bmatrix} 17303.674, 4632.281, -46423.756 \\ 15189.8, 6159.436, -45587.202 \\ 20091.849, 7719.074, -66174.989 \\ 20861.017, 6179.876, -57484.449 \\ -10549.15, 2442.444, -8767.331 \\ 10314.801, -626.773, -14377.404 \end{bmatrix}$$

nu.dot + biases 2

$$\begin{bmatrix} 17303.674, & 4632.281, & -46423.756 \\ 15189.8, & 6159.436, & -45587.202 \\ 20091.849, & 7719.074, & -66174.989 \\ 20861.017, & 6179.876, & -57484.449 \\ -10549.15, & 2442.444, & -8767.331 \\ 10314.801, & -626.773, & -14377.404 \end{bmatrix} + [-1.4, 9.2, -20.5]$$

=

$$\begin{bmatrix} 17302.274, & 4641.481, & -46444.256 \\ 15188.4, & 6168.636, & -45607.702 \\ 20090.449, & 7728.274, & -66195.489 \\ 20859.617, & 6189.076, & -57504.949 \\ -10550.55, & 2451.644, & -8787.831 \\ 10313.401, & -617.573, & -14397.904 \end{bmatrix}$$