

NIM : 442023611012
NAME : RIZKY CAHYONO PUTRA

Tugas 3: Membuat FeedForward dengan Numpy

1. Pendahuluan

Jaringan Saraf Tiruan (JST) adalah model komputasi yang terinspirasi dari cara kerja otak manusia. Pada laporan ini, kita mengimplementasikan JST sederhana dengan **feedforward** menggunakan **2 input neurons, 2 hidden neurons, dan 1 output neuron**.

2. Struktur Jaringan

Jaringan saraf yang diimplementasikan memiliki struktur sebagai berikut:

- **Input Layer:** 2 neuron (menerima input acak)
- **Hidden Layer:** 2 neuron (menggunakan bobot dan bias acak)
- **Output Layer:** 1 neuron (menghasilkan keluaran akhir)

Jaringan ini menggunakan **fungsi aktivasi sigmoid** pada hidden layer dan output layer untuk memastikan bahwa output berada dalam rentang (0,1).

3. Implementasi dalam kode

a. Inisialisasi Bobot dan Bias

Bobot dan bias diinisialisasikan dengan random, dengan menggunakan “np.random.rand(2, 1)”

```
1 # inisialisasi input layer (input layer => 2 node)
2 X = np.random.rand(2, 1) # 2 input neurons, 1 sample(column vector)
3
4 # inisialisasi bobot dan bias secara random
5 np.random.seed(42)
6
7 # inisialisasi hidden layer (hidden layer => 2 node)
8 W_hidden = np.random.rand(2, 2) # bobot untuk hidden layer (2x2)
9 b_hidden = np.random.rand(2, 1) # bias untuk hidden layer (2x1)
10
11 # inisialisasi output layer (output layer => 1 node)
12 W_output = np.random.rand(1, 2) # bobot untuk hidden layer (1x2)
13 b_output = np.random.rand(1, 1) # bias untuk output layer (1x1)
```

b. Fungsi Aktivasi Sigmoid

Fungsi Sigmoid digunakan untuk menghitung aktivasi Neuron



```
1 # Definisi aktivasi sigmoid
2 def sigmoid(x):
3     return 1 / (1 + np.exp(-x))
```

c. Proses FeedForward

proses FeedForward dibagi menjadi dua bagian:

1. proses di Hidden Layer

- input di kalikan dengan bobot dan ditambah dengan bias
- hasilnya di proses dengan fungsi sigmoid



```
1 # --- FeedForward ---
2 # 1. perhitungan di hidden layer
3 z_hidden = np.dot(W_hidden, X) + b_hidden # perhitungan formula: z=W.x+b;
4 a_hidden = sigmoid(z_hidden)
```

2. Proses di Output Layer

- output dari hidden layer dikalikan dengan bobot output dan ditambah dengan bias
- Hasilnya diproses lagi dengan sigmoid, untuk hasil akhir

```
1 # 2. perhitungan di output layer
2 z_output = np.dot(W_output, a_hidden) + b_output # perhitungan formula: z=W.x+b;
3 a_output = sigmoid(z_output)
```

d. Menampilkan Hasil

Mencetak untuk input, output dari hidden layer dan output akhir

```
1 # mencetak hasil
2 print("input: \n", X)
3 print("Hidden Layer Output (a_hidden): \n", a_hidden)
4 print("Final Output (a_hidden): \n", a_output)
```

e. Hasil

```
(venv) ch4rl0tt3@dell:~/Documents/Code/Artifical Intelligent/Feedforward$ "
e/ch4rl0tt3/Documents/Code/Artifical Intelligent/Feedforward/script.py"
input:
[[0.61873805]
 [0.61612668]]
Hidden Layer Output (a_hidden):
[[0.72582081]
 [0.72666283]]
Final Output (a_hidden):
[[0.78120311]]
```