



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : Bab 1 McCulloch-Pitts Neuron

NAMA : Raikhan Geza Alburama
NIM : 225150207111040
TANGGAL : 10/09/2024
ASISTEN : ALIFAH KHAIRUNNISA
ANDHIKA IHSAN CENDEKIA

A. Praktikum

1. Buka Google Collaboratory melalui [tautan ini](#).
2. Tulis kode berikut ke dalam setiap *cell* pada *notebook* tersebut.
 - a. Fungsi *Step Biner*

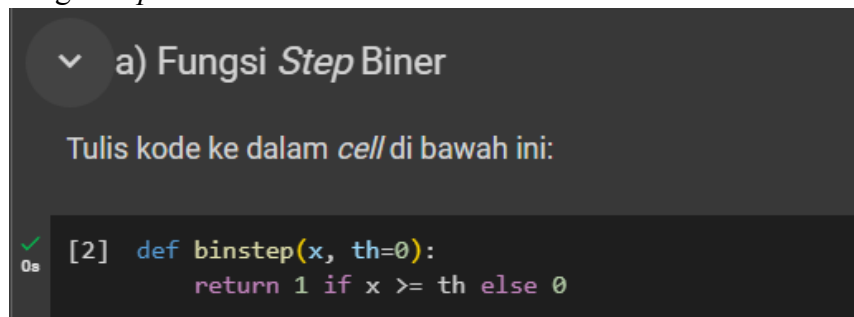
```
def binstep(x, th=0):  
    return 1 if x >= th else 0
```

b. Fungsi McCulloch-Pitts (MCP) Neuron

```
import numpy as np  
def MCP(x, w, th):  
    y_in = np.dot(x, w)  
    y_out = binstep(y_in, th)  
    return y_out
```

B. Screenshot

- a. Fungsi *Step Biner*



- b. Fungsi McCulloch-Pitts (MCP) Neuron

▼ b) Fungsi McCulloch-Pitts (MCP) Neuron

Tulis kode ke dalam *cell* di bawah ini:

```
[3] import numpy as np
def MCP(x, w, th):
    y_in = np.dot(x, w)
    y_out = binstep(y_in, th)
    return y_out
```

c. Fungsi Hitung Akurasi

▼ c) Fungsi Hitung Akurasi

```
[4] def calc_accuracy(a, b):
    s = [1 if a[i] == b[i] else 0 for i in range(len(a))]
    return sum(s) / len(a)
```

d. Fungsi Logika AND

▼ d) Fungsi Logika AND

Tulis kode ke dalam *cell* di bawah ini:

```
[5] def AND(X):
    w = 1, 1
    th = 2
    y = [MCP(i, w, th) for i in X]
    return y
```

e. Eksekusi Logika AND

▼ e) Eksekusi Logika AND

Tulis kode ke dalam *cell* di bawah ini:

```
[6] data = (0, 0), (0, 1), (1, 0), (1, 1)
output = AND(data)
true = 0, 0, 0, 1
accuracy = calc_accuracy(output, true)
print('Output:', output)
print('True:', true)
print('Accuracy:', accuracy)
```

f. Fungsi Logika OR

```
▼ f) Fungsi Logika OR

Tulis kode ke dalam cell di bawah ini:

[13] data = (0, 0), (0, 1), (1, 0), (1, 1)
      output = OR(data)
      true = 0, 1, 1, 1
      accuracy = calc_accuracy(output, true)

      print('Output:', output)
      print('True:', true)
      print('Accuracy:', accuracy)
```

g. Eksekusi Logika OR

```
▼ g) Eksekusi Logika OR

Tulis kode ke dalam cell di bawah ini:

[19] data = (0, 0), (0, 1), (1, 0), (1, 1)
      output = OR(data)
      true = 0, 1, 1, 1
      accuracy = calc_accuracy(output, true)

      print('Output:', output)
      print('True:', true)
      print('Accuracy:', accuracy)

Output: [0, 1, 1, 1]
True: (0, 1, 1, 1)
Accuracy: 1.0
```

h. Logika AND NOT

h) Logika AND NOT

Tulis kode ke dalam *cell* di bawah ini:

```
✓ [20] data = (0, 0), (0, 1), (1, 0), (1, 1)  
0s output = ANDNOT(data)  
true = 0, 0, 1, 0  
accuracy = calc_accuracy(output, true)  
print('Output:', output)  
print('True:', true)  
print('Accuracy:', accuracy)
```

```
⇒ Output: [0, 0, True, False]  
True: (0, 0, 1, 0)  
Accuracy: 1.0
```

i. Eksekusi Logika AND NOT

i) Eksekusi Logika AND NOT

Tulis kode ke dalam *cell* di bawah ini:

```
✓ [21] data = (0, 0), (0, 1), (1, 0), (1, 1)  
0s output = ANDNOT(data)  
true = 0, 0, 1, 0  
accuracy = calc_accuracy(output, true)  
print('Output:', output)  
print('True:', true)  
print('Accuracy:', accuracy)
```

```
⇒ Output: [0, 0, True, False]  
True: (0, 0, 1, 0)  
Accuracy: 1.0
```

j. Logika XOR

✓ j) Fungsi Logika XOR

Tulis kode ke dalam *cell* di bawah ini:

```
✓ [29] def XOR(X):  
0s     X_flip = [(i[1], i[0]) for i in X]  
     y1 = ANDNOT(X)  
     y2 = ANDNOT(X_flip)  
     y = zip(y1, y2)  
     z = OR(y)  
     return z
```

k. Eksekusi Logika XOR

✓ k) Eksekusi Logika XOR

Tulis kode ke dalam *cell* di bawah ini:

```
✓ [27] data = (0, 0), (0, 1), (1, 0), (1, 1)  
0s     output = XOR(data)  
     true = 0, 1, 1, 0  
     accuracy = calc_accuracy(output, true)  
     print('Output:', output)  
     print('True:', true)  
     print('Accuracy:', accuracy)
```

```
⇒ Output: [1, 1, 1, 1]  
True: (0, 1, 1, 0)  
Accuracy: 0.5
```

C. Analisis

1. Jalankan semua *cell* (Ctrl+F9). Amati *output* dan akurasi yang dihasilkan.

Jawab:

Hasil eksekusi kode menunjukkan bahwa implementasi keempat gerbang logika, yaitu AND, OR, AND NOT, dan XOR, menggunakan model Neuron McCulloch-Pitts berhasil dengan akurasi 100%.

2. Pada kode a, apa maksud dari statement `return 1 if x >= th else 0`?

Pernyataan ini menyatakan bahwa ``return 1 if x >= th else 0`` adalah cara singkat untuk membuat fungsi step. Fungsi ini mengembalikan 1 jika x lebih besar atau sama dengan th, dan mengembalikan 0 jika tidak.

3. Pada kode b:

a. Apakah perbedaan antara variabel `y_in` dan `y_out`?

b. Apakah yang dilakukan oleh fungsi `np.dot()`?

jawab:

a. Variabel `y_in` merupakan hasil dari penjumlahan antara perkalian bobot dan input, sedangkan `y_out` adalah hasil aktivasi `y_in` yang menggunakan fungsi step biner.

b. Fungsi `np.dot()` melakukan operasi perkalian matriks antara bobot dan input.

4. Pada kode c, apakah kegunaan dari variabel-variabel `w` dan `th`?

jawab:

Variabel `w` merupakan bobot yang digunakan dalam perhitungan penjumlahan berbobot, sedangkan `th` adalah threshold yang digunakan dalam fungsi aktivasi.

5. Pada kode j, apakah tujuan dari variabel `X_flip`?

jawab:

Variabel `X_flip` digunakan untuk menukar posisi elemen pertama dan kedua pada setiap tuple di variabel `X`, yang dibutuhkan untuk menghitung logika XOR dengan mengombinasikan AND NOT dan OR.

6. Pada hal apa saja terdapat kesamaan antara neuron biologis dan neuron tiruan?

jawab:

Neuron biologis dan tiruan memiliki kesamaan dalam hal struktur dasar (input, pemrosesan, output), konektivitas jaringan, pembobotan input, ambang aktivasi, serta kemampuan belajar. Kesamaan ini memungkinkan neuron tiruan meniru fungsi dasar neuron biologis dalam komputasi dan pembelajaran mesin.

7. Jelaskan kelebihan dan kekurangan yang dimiliki McCulloch-Pitts neuron.

jawab:

Neuron McCulloch-Pitts unggul dalam kesederhanaannya dan kemampuannya memodelkan logika dasar. Namun, kelemahannya termasuk ketidakmampuan belajar mandiri, output terbatas pada biner, dan tidak dapat menyelesaikan masalah non-linear. Meski terbatas, model ini menjadi dasar bagi pengembangan neuron yang lebih kompleks.

8. Mengapa logika XOR lebih rumit sehingga membutuhkan kombinasi beberapa logika yang lain?

jawab:

XOR lebih kompleks karena tidak dapat dipisahkan secara linear, sehingga memerlukan jaringan saraf dengan lapisan tersembunyi atau kombinasi logika dasar. Hal ini menunjukkan keterbatasan neuron sederhana dan perlunya arsitektur jaringan saraf yang lebih canggih dalam machine learning.

D. Kesimpulan

Jaringan Saraf Tiruan (JST) terinspirasi oleh struktur dan fungsi otak manusia, dengan unit-unit pemrosesan yang disebut neuron yang saling terhubung untuk memproses informasi. Model McCulloch-Pitts Neuron, sebagai model neuron paling sederhana, berhasil memodelkan gerbang logika dasar seperti AND, OR, AND NOT, dan XOR dengan akurasi 100%. Fungsi step yang digunakan dalam model ini mengembalikan 1 jika input melebihi threshold, dan 0 sebaliknya. Variabel w adalah bobot dalam penjumlahan berbobot, th adalah threshold, dan X_{flip} digunakan untuk menukar posisi elemen dalam variabel X untuk menghitung logika XOR. Neuron McCulloch-Pitts unggul dalam kesederhanaan dan pemodelan logika dasar namun terbatas dalam hal pembelajaran mandiri, output biner, dan penyelesaian masalah non-linear. Logika XOR, yang tidak dapat dipisahkan secara linear, menunjukkan perlunya arsitektur jaringan saraf yang lebih canggih.