



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : Bab 2 Hebb

NAMA : Raikhan Geza Alburama
NIM : 225150207111040
TANGGAL : 17/09/2024
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 Bipolar

```
def bipstep(y, th=0):  
    return 1 if y >= th else -1
```

b. Fungsi training Hebb

```
def hebb_fit(train, target, verbose=False,  
            draw=False, draw_padding=1):  
    w = np.zeros(len(train[0]) + 1)  
    bias = np.ones((len(train), 1))  
    train = np.hstack((bias, train))  
  
    for r, row in enumerate(train):  
        w = [w[i] + row[i] * target[r] for i in range(len(row))]  
  
        if verbose:  
            print('Bobot:', w)  
  
        if draw:  
            plot(line(w, 0), train, target, draw_padding)  
  
    return w
```

c. Fungsi Testing Hebb

```
def hebb_predict(X, w):  
    Y = []  
  
    for x in X:  
        y_in = w[0] + np.dot(x, w[1:])  
        y = bipstep(y_in)
```

```
Y.append(y)

return Y
```

d. Fungsi Hitung Akurasi

```
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)
```

e. Logika AND

```
from sklearn.metrics import accuracy_score

train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = 1, -1, -1, -1
model = hebb_fit(train, target, verbose=False,
draw=False) output = hebb_predict(train, model)
accuracy = accuracy_score(output, target)

print('Output:', output)
print('Target:', target)
print('Accuracy:', accuracy)
```

f. Logika OR

```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = 1, 1, 1, -1
model = hebb_fit(train, target, verbose=True,
draw=True) output = hebb_predict(train, model)
accuracy = accuracy_score(output, target)

print('Output:', output)
print('Target:', target)
print('Accuracy:', accuracy)
```

g. Logika AND NOT

```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = -1, 1, -1, -1
model = hebb_fit(train, target, verbose=True,
draw=True) output = hebb_predict(train, model)
accuracy = accuracy_score(output, target)
```

```
print('Output:', output)
print('Target:', target)
print('Accuracy:', accuracy)
```

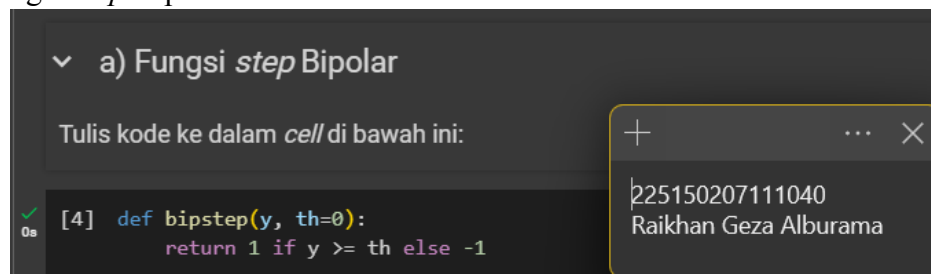
h. Logika XOR

```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = -1, 1, 1, -1
model = hebb_fit(train, target, verbose=True,
draw=True) output = hebb_predict(train, model)
accuracy = accuracy_score(output, target)

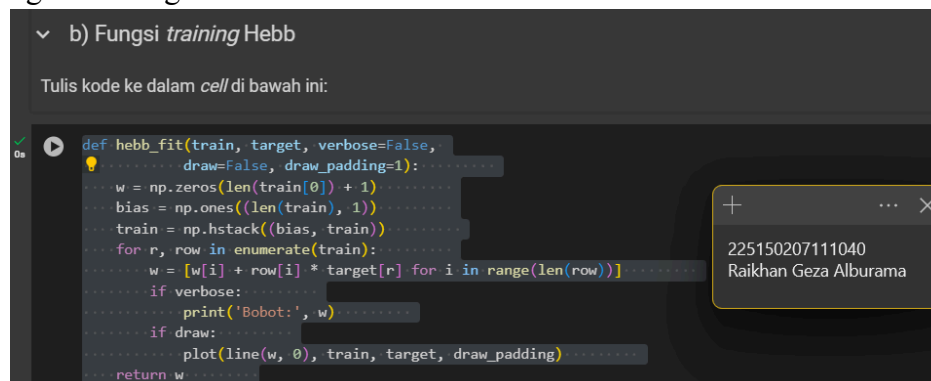
print('Output:', output)
print('Target:', target)
print('Accuracy:', accuracy)
```

B. Screenshot

a. Fungsi *Step* Bipolar



b. Fungsi Training Hebb



c. Fungsi Testing

▼ c) Fungsi *testing* Hebb

Tulis kode ke dalam *cell* di bawah ini:

```
[37] def hebb_predict(X, w):  
    Y = []  
    for x in X:  
        y_in = w[0] + np.dot(x, w[1:])  
        y = bipstep(y_in)  
        Y.append(y)  
    return Y
```

225150207111040
Raikhan Geza Alburama

d. Fungsi Hitung Akurasi

▼ d) Fungsi Hitung Akurasi

```
[38] 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)
```

225150207111040
Raikhan Geza Alburama

e. Fungsi Logika AND

▼ e) Logika AND

Tulis kode ke dalam *cell* di bawah ini:

```
from sklearn.metrics import accuracy_score  
train = (1, 1), (1, -1), (-1, 1), (-1, -1)  
target = 1, -1, -1, -1  
model = hebb_fit(train, target, verbose=False, draw=False)  
output = hebb_predict(train, model)  
accuracy = accuracy_score(output, target)  
print('Output:', output)  
print('Target:', target)  
print('Accuracy:', accuracy)
```

225150207111040
Raikhan Geza Alburama

f. Eksekusi Logika AND

+

...

×

Output: [1, 1, 1, 1]
Target: (1, -1, -1, -1)
Accuracy: 0.25

225150207111040
Raikhan Geza Alburama

g. Fungsi Logika OR

▼ f) Logika OR

Tulis kode ke dalam *cell* di bawah ini:

[67] def hebb_fit(train, target, verbose=False,
draw=False, draw_padding=1):
w = np.zeros(len(train[0]) + 1)
bias = np.ones((len(train), 1))
train = np.hstack((bias, train))

return w

+

...

×

225150207111040
Raikhan Geza Alburama

h. Eksekusi Logika OR

+

...

×

Output: [1, 1, 1, 1]
Target: (1, 1, 1, -1)
Accuracy: 0.75

225150207111040
Raikhan Geza Alburama

i. Logika AND NOT

g) Logika AND NOT

Tulis kode ke dalam *cell* di bawah ini:

```
[68] train = (1, 1), (1, -1), (-1, 1), (-1, -1)
      target = -1, 1, -1, -1
      model = hebb_fit(train, target, verbose=True, draw=True)
      output = hebb_predict(train, model)
      accuracy = accuracy_score(output, target)
      print('Output:', output)
      print('Target:', target)
      print('Accuracy:', accuracy)
```

225150207111040
Raikhan Geza Alburama

j. Eksekusi Logika AND NOT

Output: [1, 1, 1, 1]
Target: (-1, 1, -1, -1)
Accuracy: 0.25

225150207111040
Raikhan Geza Alburama

k. Logika XOR

h) Logika XOR

Tulis kode ke dalam *cell* di bawah ini:

```
[69] train = (1, 1), (1, -1), (-1, 1), (-1, -1)
      target = -1, 1, 1, -1
      model = hebb_fit(train, target, verbose=True, draw=True)
      output = hebb_predict(train, model)
      accuracy = accuracy_score(output, target)
      print('Output:', output)
      print('Target:', target)
      print('Accuracy:', accuracy)
```

225150207111040
Raikhan Geza Alburama

i. Eksekusi Logika XOR

Output: [1, 1, 1, 1]
Target: (-1, 1, 1, -1)
Accuracy: 0.5

225150207111040
Raikhan Geza Alburama

C. Analisis

1. Pada klasifikasi menggunakan logika XOR, mengapa akurasi yang didapatkan tidak mencapai 1 (100%)?
2. Lakukan proses training dan testing menggunakan data berikut. Training: (-1, .5), (.5, .3), (1, 1.5), (3, 1.9) Target: (-1, -1, 1, 1) Berapakah akurasi yang didapatkan? Mengapa tidak dapat mencapai akurasi 1 (100%)?

Jawaban

1. Klasifikasi XOR tidak mencapai akurasi 100% karena sifatnya yang non-linear. Model linear sederhana tidak mampu memisahkan data XOR dengan sempurna. Dibutuhkan model lebih kompleks, seperti neural network dengan minimal satu hidden layer, untuk menangani pola XOR. Faktor lain seperti arsitektur jaringan, inisialisasi bobot, jumlah iterasi, dan learning rate juga dapat mempengaruhi akurasi.

2. Pada dataset yang diberikan, akurasi tidak mencapai 100% karena beberapa alasan: pola non-linear dalam data, keterbatasan model linear, jumlah data yang sangat sedikit (hanya 4 titik), dan kompleksitas hubungan antara input (koordinat) dan output (klasifikasi biner). Untuk meningkatkan akurasi, bisa dicoba menggunakan model non-linear, menambah fitur polynomial, atau menerapkan teknik ensemble. Namun, dengan data yang sangat terbatas, peningkatan signifikan mungkin sulit dicapai tanpa menambah jumlah sampel atau mengubah pendekatan pemodelan

D. Kesimpulan

1. Apakah itu Hebb Net dan apa yang membedakan (kelebihan dan kekurangan) Hebb Net dengan McCulloch Pitts Neuron?
2. Apakah fungsi dari bias?
3. Jelaskan tentang Epoch

Jawaban

1. Hebb Net adalah model jaringan saraf sederhana yang dapat belajar dan menangani input kontinu, namun kurang efektif untuk masalah non-linear yang kompleks dan rentan overfitting. Sebaliknya, McCulloch-Pitts Neuron lebih mudah diimplementasikan, tapi terbatas pada input biner dan tidak bisa belajar.
2. Bias dalam jaringan saraf berfungsi sebagai nilai ambang batas yang memengaruhi kapan neuron akan aktif. Bias ini memungkinkan jaringan untuk menggeser fungsi aktivasi, sehingga memberikan fleksibilitas lebih dalam mengenali pola. Tanpa bias, neuron hanya akan aktif jika total input yang diterima melebihi nol, yang bisa membatasi kemampuan jaringan.
3. Epoch dalam jaringan saraf adalah satu putaran penuh di mana seluruh data pelatihan diproses melalui jaringan, baik secara batch maupun stochastic. Selama satu epoch, setiap sampel dalam dataset digunakan untuk memperbarui bobot jaringan. Jumlah epoch menunjukkan seberapa sering model akan melihat seluruh dataset selama pelatihan.