



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : ADALINE

NAMA : RAIKHAN GEZA ALBURAMA
NIM : 225150207111040
TANGGAL : 25/09/2024
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 percep_step(input, th=0):  
    return 1 if input > th else -1 if input < -th else 0
```

b. Fungsi training Adaline

```
import sys  
  
def adaline_fit(x, t, alpha=.1, max_err=.1,  
max_epoch=-1, verbose=False, draw=False):  
  
    w = np.random.uniform(0, 1, len(x[0]) + 1)  
  
    b = np.ones((len(x), 1))  
  
    x = np.hstack((b, x))  
  
    stop = False  
  
    epoch = 0  
  
    while not stop and (max_epoch == -1 or epoch < max_epoch):  
  
        epoch += 1  
  
        max_ch = -sys.maxsize  
  
        if verbose:  
  
            print('\nEpoch', epoch)
```

```

for r, row in enumerate(x):
    y = np.dot(row, w)

    for i in range(len(row)):
        w_new = w[i] + alpha * (t[r] - y) * row[i]

        max_ch = max(abs(w[i] - w_new), max_ch)

        w[i] = w_new

    if verbose:
        print('Bobot:', w)

    if draw:
        plot(line(w), x, t)

    stop = max_ch < max_err

return w, epoch

```

c. Fungsi Testing Adaline

```

def percep_predict(X, w, th=0):
    Y = []

    for x in X:
        y_in = w[0] + np.dot(x, w[1:])

        y = percep_step(y_in, th)

        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

```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)  
target = 1, -1, -1, -1  
th = .2  
model, epoch = percep_fit(train, target, th, verbose=True,  
draw=True)  
output = percep_predict(train, model)  
accuracy = calc_accuracy(output, target)  
print('Epochs:', epoch)  
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  
th = .2  
model, epoch = percep_fit(train, target, th, verbose=True,  
draw=True)  
output = percep_predict(train, model)  
accuracy = calc_accuracy(output, target)  
  
print('Epochs:', epoch)  
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
th = .2
model, epoch = percep_fit(train, target, th, verbose=True,
draw=True)
output = percep_predict(train, model)
accuracy = calc_accuracy(output, target)
print('Epochs:', epoch)
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
th = .2
model, epoch = percep_fit(train, target, th, max_epoch=50,
verbose=True, draw=False)
output = percep_predict(train, model)
accuracy = calc_accuracy(output, target)

print('Epochs:', epoch)
print('Output:', output)
print('Accuracy:', accuracy)
```

B. Screenshot

a. Fungsi *Step* Perception

▼ a) Fungsi Step Bipolar

Tulis kode ke dalam *cell* di bawah ini:

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

+

...

×

225150207111040
Raikhan Geza Alburama

b. Fungsi Training Perceptron

▼ b) Fungsi *Training* Adaline

Tulis kode ke dalam *cell* di bawah ini:

```
import sys  
def adaline_fit(x, t, alpha=.1, max_err=.1, max_epoch=-1, verbose=False, draw=False):  
    w = np.random.uniform(0, 1, len(x[0]) + 1)  
    b = np.ones((len(x), 1))  
    x = np.hstack((b, x))  
    stop = False  
    epoch = 0  
    while not stop and (max_epoch == -1 or epoch < max_epoch):  
        epoch += 1  
        max_ch = -sys.maxsize  
        if verbose:  
            print('\nEpoch', epoch)  
        for r, row in enumerate(x):  
            y = np.dot(row, w)  
            for i in range(len(row)):  
                w_new = w[i] + alpha * (t[r] - y) * row[i]  
                max_ch = max(abs(w[i] - w_new), max_ch)  
                w[i] = w_new  
            if verbose:  
                print('Bobot:', w)  
            if draw:  
                plot(line(w), x, t)  
        stop = max_ch < max_err  
    return w, epoch
```

+

...

×

225150207111040
Raikhan Geza Alburama

c. Fungsi Testing Perceptron

✓ c) Fungsi *Testing* Adaline

Tulis kode ke dalam *cell* di bawah ini:

```
✓ [4] def adaline_predict(X, w):  
0s   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

Tulis kode ke dalam *cell* di bawah ini:

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

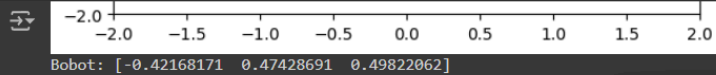
+ ... ✕
225150207111040
Raikhan Geza Alburama

e. Logika AND

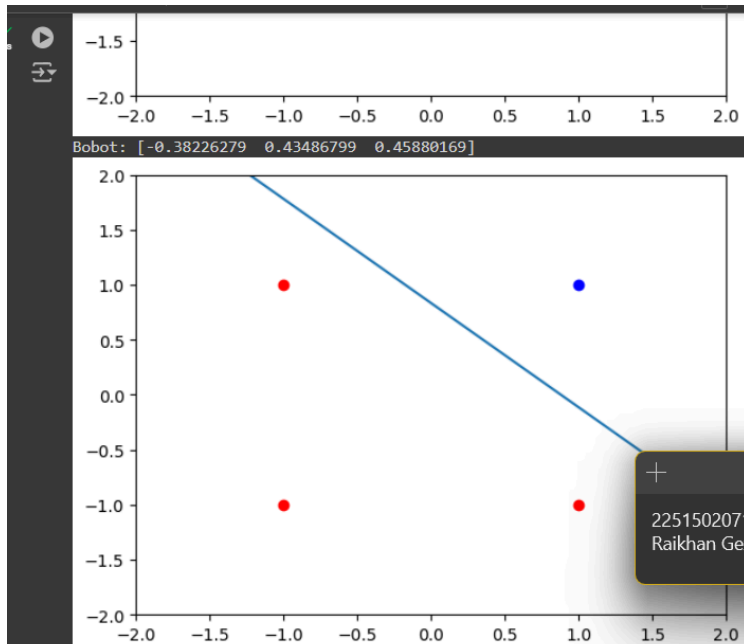
✓ e) Logika AND

Tulis kode ke dalam *cell* di bawah ini:

```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = 1, -1, -1, -1
w, epoch = adaline_fit(train, target, verbose=True, draw=True)
output = adaline_predict(train, w)
accuracy = calc_accuracy(output, target)
print('Output:', output)
print('Epoch:', epoch)
print('Target:', target)
print('Accuracy:', accuracy)
```



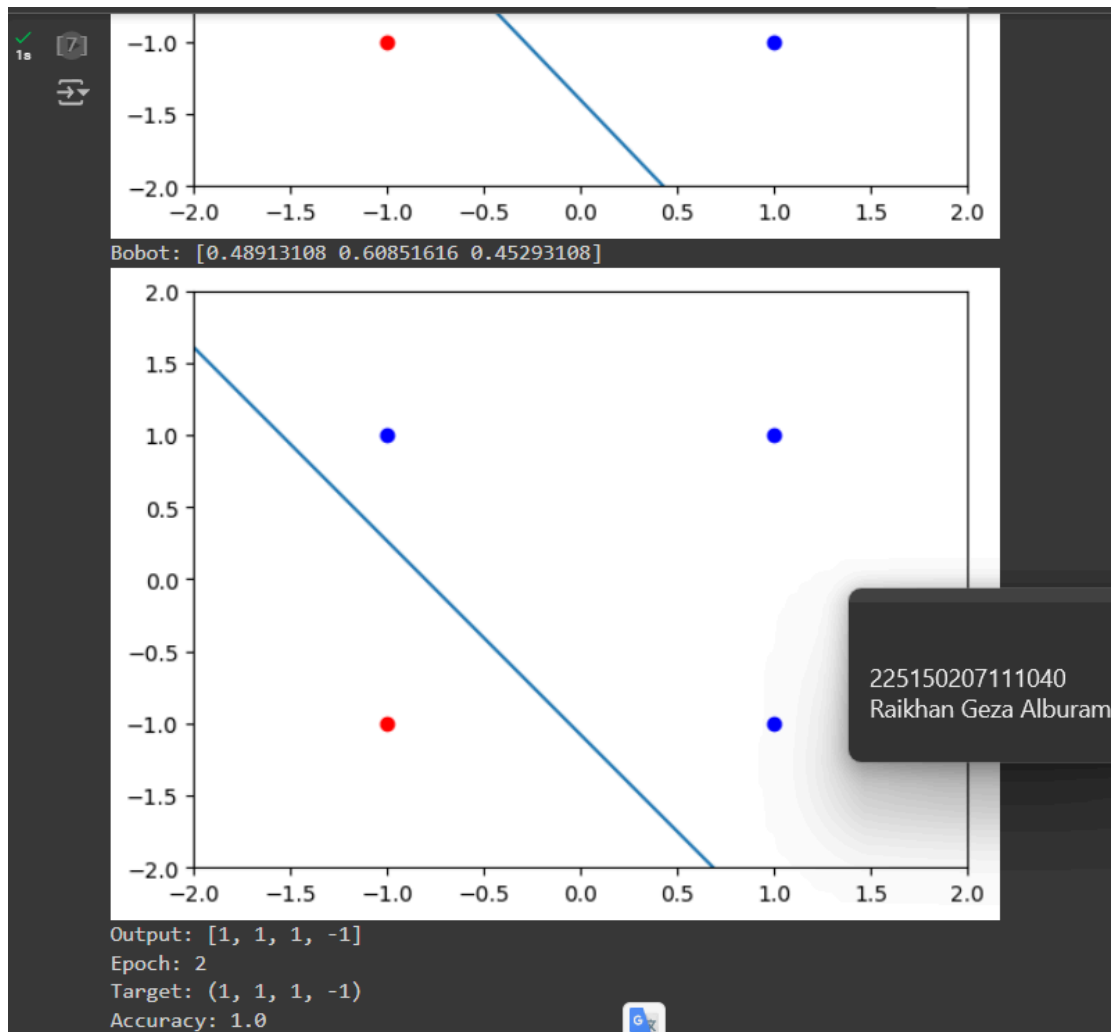
225150207111040
Raikhan Geza Alburama



225150207111040
Raikhan Geza Alburama

Output: [1, -1, -1, -1]
Epoch: 5
Target: (1, -1, -1, -1)
Accuracy: 1.0

f. Logika OR



g. Logika AND NOT

g) Logika AND NOT

Tulis kode ke dalam *cell* di bawah ini:

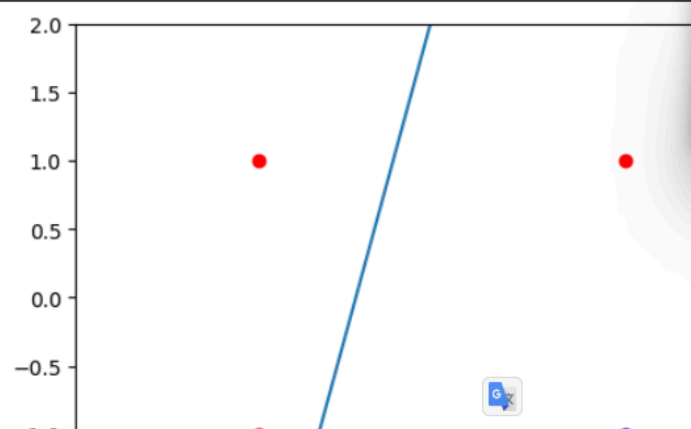
```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = -1, 1, -1, -1
w, epoch = adaline_fit(train, target, verbose=True, draw=True)
output = adaline_predict(train, w)
accuracy = calc_accuracy(output, target)

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

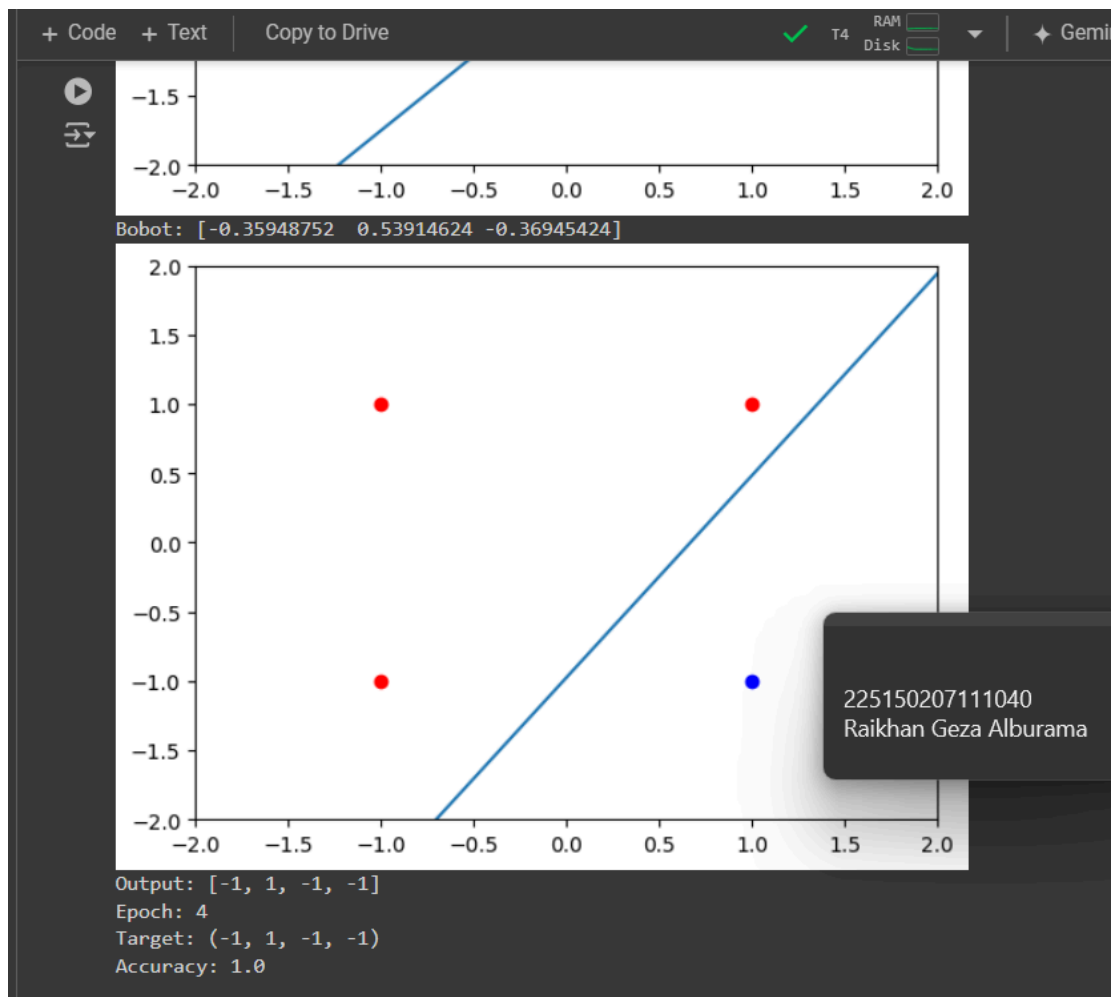


Epoch 1

Bobot: [0.17783458 0.37624369 -0.07660271]



225150207111040
Raikhan Geza Alburama



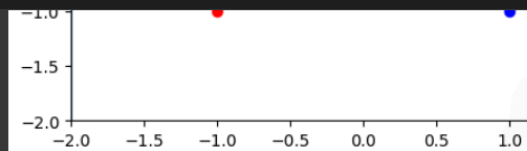
h. Logika XOR

h) Logika XOR

Tulis kode ke dalam *cell* di bawah ini:

```
[9] train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = -1, 1, 1, -1
w, epoch = adaline_fit(train, target, verbose=True, draw=True)
output = adaline_predict(train, w)
accuracy = calc_accuracy(output, target)

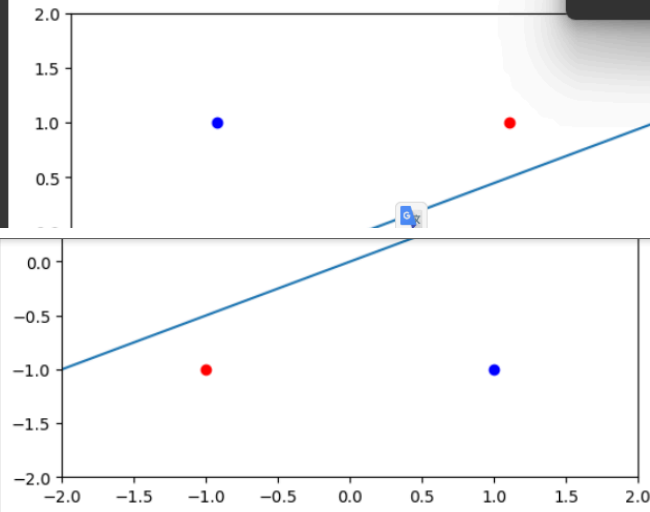
print('Output:', output)
print('Epoch:', epoch)
print('Target:', target)
print('Accuracy:', accuracy)
```



Bobot: [1.60066142e-05 5.88356176e-02 -1.17634934e-01]

225150207111040

Raikhan Geza Alburama



Bobot: [1.17661461e-01 -5.88098369e-02 1.05204122e-05]

KeyboardInterrupt Traceback (most recent call last)

<ipython-input-9-d853b4015744> in <cell line: 3>()

1 train = (1, 1), (1, -1), (-1, 1), (-1, -1)

2 target = -1, 1, 1, -1

----> 3 w, epoch = adaline_fit(train, target, verbose=True, draw=True)

4 output = adaline_predict(train, w)

5 accuracy = calc_accuracy(output, target)

27 frames

<decorator-gen-2> in __call__(self, obj)

/usr/local/lib/python3.10/dist-packages/matplotlib/ctbook/_init_.py in __init__(self, exception_handler, signals)

188 # _func_cid_map: signal -> {weakref-to-callback -> cid}

189

--> 190 def __init__(self, exception_handler=exception_printer, *, signals=None):

191 self.signals = None if signals is None else list(signals) # Copy it.

192 self.exception_handler = exception_handler

KeyboardInterrupt:



225150207111040

Raikhan Geza Alburama

C. Analisis

1. Jelaskan tujuan dari parameter-parameter α , \max_err , dan \max_epoch yang ada pada fungsi `adaline_fit`.

Jawaban:

Tujuan dari parameter diatas yang ada pada fungsi `adaline_fit` yaitu:

- α yaitu parameter yang berisikan nilai dari learning rate.
- \max_err yaitu parameter yang berisikan nilai batas pembanding dengan perubahan bobot maksimum.
- \max_epoch yaitu parameter yang merupakan batas maksimal epoch dalam suatu proses

2. Pada fungsi `adaline_fit`, apakah yang akan dilakukan oleh fungsi tersebut jika parameter \max_epoch diberi nilai -1?

Jawaban:

Ketika parameter \max_epoch diberi nilai -1, proses looping akan berhenti Ketika epoch memiliki nilai -1. Oleh karena itu output yang terjadi adalah looping tidak akan berhenti karena epoch tidak mungkin menjadi -1 karena iterasi +1

3. Amati jumlah epoch saat melakukan proses pelatihan menggunakan data logika AND, OR, dan AND NOT. Mengapa jumlah epoch pada ketiga proses pelatihan tersebut tidak sama?

Jawaban:

Epoch pada logika AND adalah 3, OR adalah 3, dan AND NOT adalah 5. Perbedaan epoch tersebut terjadi disebabkan karena perbedaan target. Perbedaan target menyebabkan nilai delta "w" yang berbeda pada tiap iterasinya. Nilai epoch akan berhenti bertambah jika nilai delta w tertinggi dari suatu epoch lebih kecil dari nilai batas yang ditentukan.

4. Apakah yang terjadi saat melakukan proses pelatihan untuk data logika XOR? Mengapa bisa terjadi demikian?

Jawaban:

Logika XOR menghasilkan output yang tidak terbatas (infinite). Hal tersebut dikarenakan Adaline bersifat linearly seperable sehingga Adaline hanya bisa membagi data dalam garis lurus, sedangkan logika XOR tidak bersifat linear.

D. Kesimpulan

Adaline merupakan salah satu metode algoritma dalam JST yang memiliki kepanjangan Adaptive Linear Neuron. Adaline digunakan dalam pengenalan pola karena merupakan perbaikan dari metode Perceptron. Algoritma Adaline memperbaikinya dengan memodifikasi bobot dengan menggunakan aturan Least Mean Square. Adaline menggunakan dua perubahan pada laju pemahaman dan batas toleransi sehingga lebih efektif dalam pengenalan pola yang dilakukan. Secara arsitektural, Adaline terdiri dari beberapa input neuron dan satu output neuron. Adaline juga menggunakan bias yang merupakan bobot dari sebuah neuron tambahan yang memiliki nilai aktivasi konstan sebesar 1.

Threshold digunakan sebagai nilai ambang internal dari node yang mempengaruhi aktivasi dari output. Sedangkan learning rate diperlukan karena dapat memengaruhi proses dan hasil pelatihan. Nilai yang terlalu rendah akan mengakibatkan pergeseran decision boundary menjadi sangat perlahan pada setiap iterasinya. Perubahan dan pergeseran decision boundary tidak menimbulkan masalah pada data yang sederhana, namun menimbulkan masalah pada data yang kompleks dengan jumlah data dan dimensi data yang tinggi.

Algoritma Adaline tidak bisa menyelesaikan logika XOR, karena cara klasifikasi Adaline pada praktikum ini masih linear, sehingga logika XOR belum bisa diselesaikan. Kondisinya, apabila pada code tidak diberi max_epoch maka prosesnya akan terus berlanjut (infinite loop). Adaline bersifat linearly seperable sehingga Adaline hanya bisa membagi data dalam garis lurus, sedangkan logika XOR tidak bersifat linear