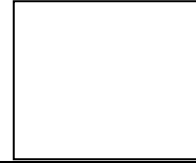




LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : ADALINE
NAMA : DIMAS TRI MUSTAKIM
NIM : 205150200111049
TANGGAL : 03/10/2022
ASISTEN : ANDIKA IRZA PRADANA



A. Praktikum

1. Buka Google Colaboratory 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 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
```

c. Fungsi Testing Adaline

```
def adaline_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

```
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)
```

f. Logika OR

```
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)
```

g. Logika AND NOT

```
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)
```

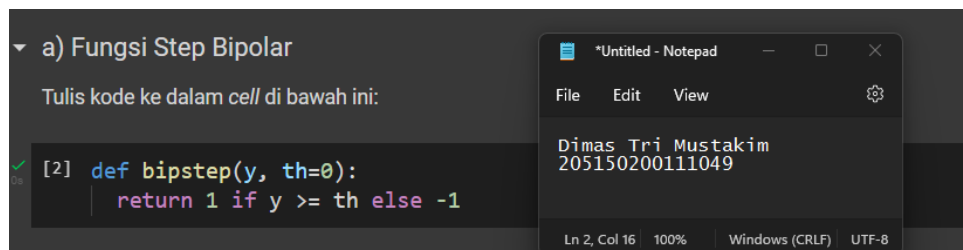
```
print('Epoch:', epoch)
print('Target:', target)
print('Accuracy:', accuracy)
```

h. Logika XOR

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

B. Screenshot

a. Fungsi Step Bipolar



b. Fungsi Training Adaline

```
[3] 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

▼ c) Fungsi *Testing* Adaline

Tulis kode ke dalam cell di bawah ini:

```
[4] def adaline_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
```

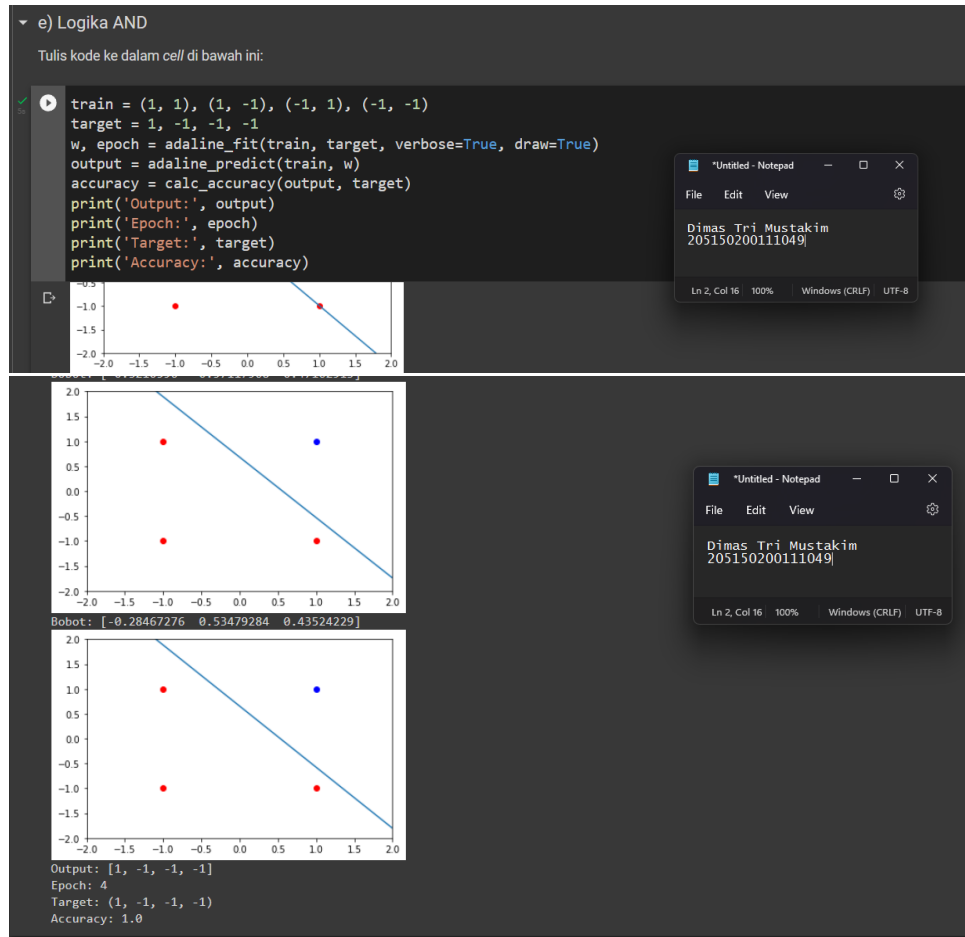
d. Fungsi Hitung Akurasi

▼ d) Fungsi Hitung Akurasi

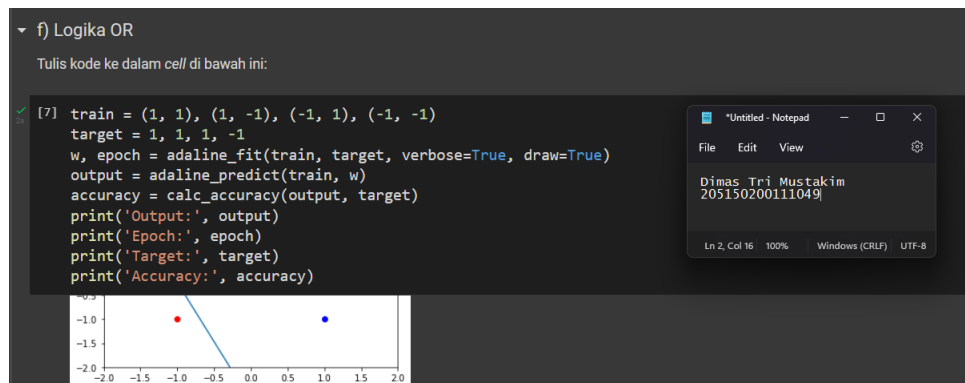
Tulis kode ke dalam cell di bawah ini:

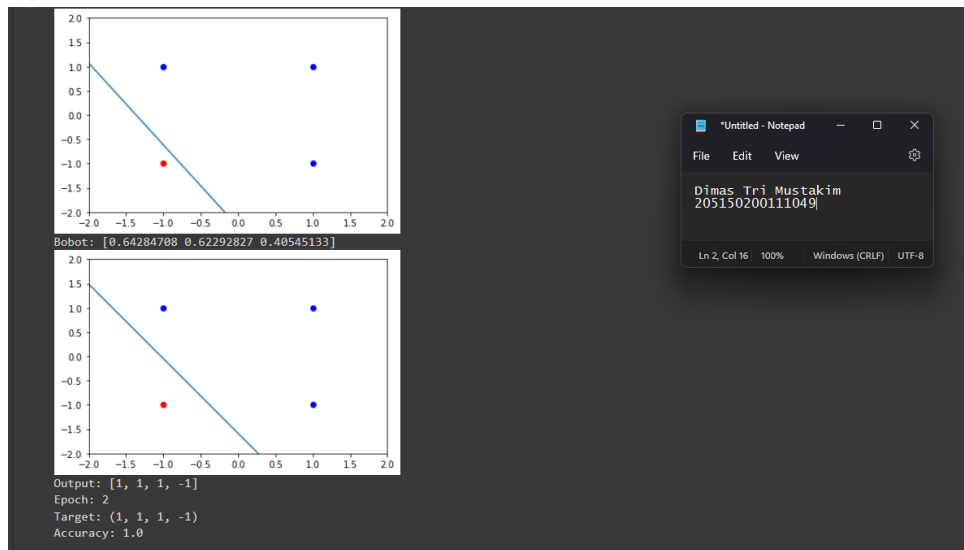
```
[5] def calc_accuracy(a, b):
    s = [1 if a[i] == b[i] else 0 for i in range(len(a))]
    return sum(s) / len(a)
```

e. Logika AND

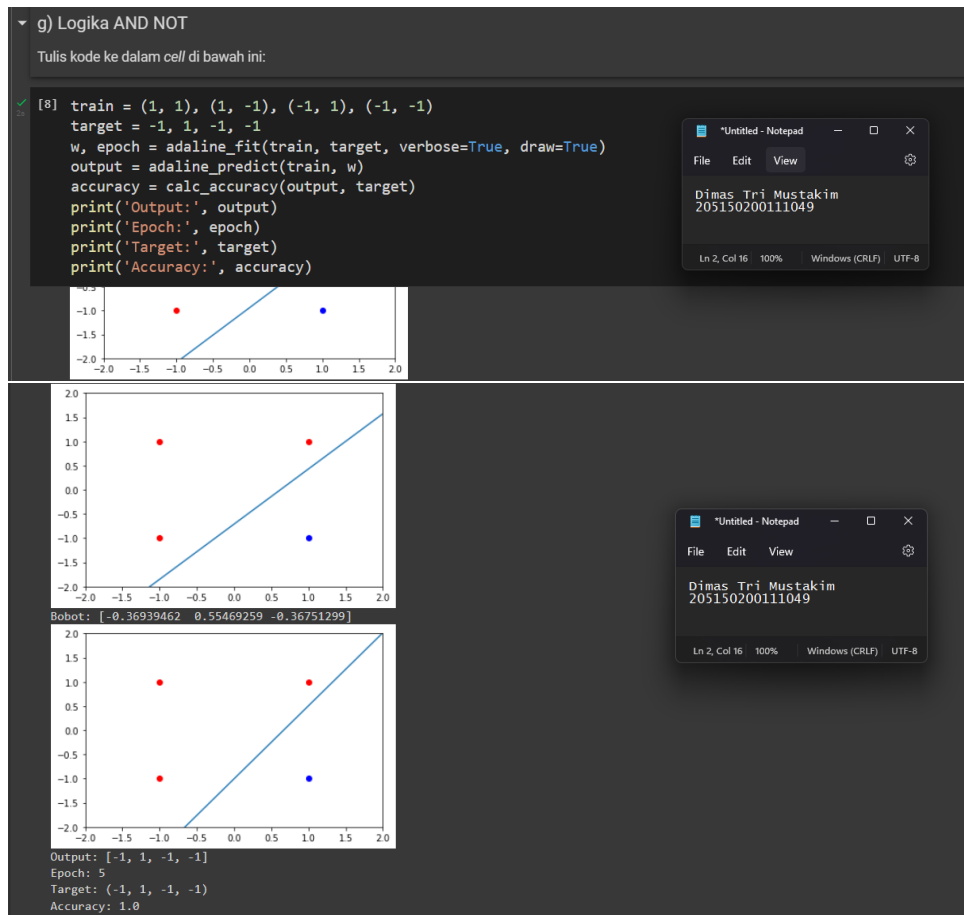


f. Logika OR

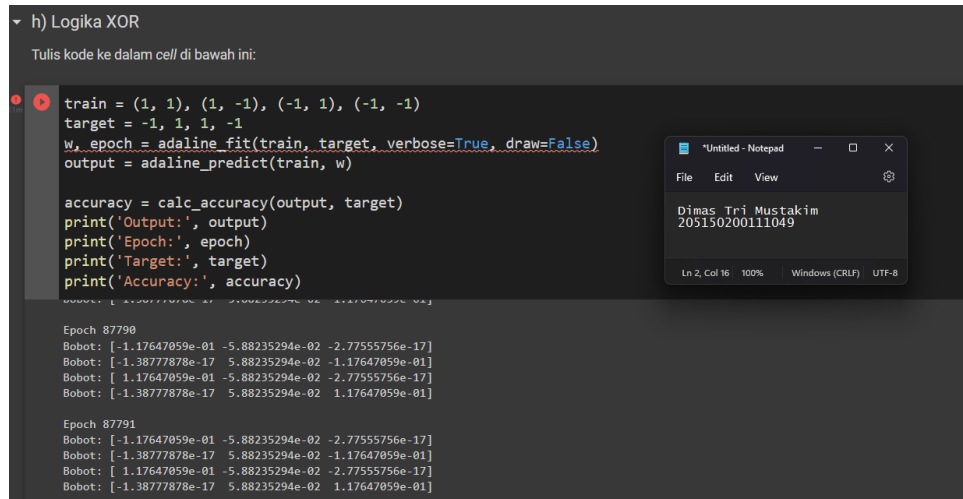




g. Logika AND NOT



h. Logika XOR



```
h) Logika XOR
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=False)
output = adaline_predict(train, w)

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

Epoch 87790
Bobot: [-1.17647059e-01 -5.88235294e-02 -2.77555756e-17]
Bobot: [-1.38777878e-17 5.88235294e-02 -1.17647059e-01]
Bobot: [ 1.17647059e-01 -5.88235294e-02 -2.77555756e-17]
Bobot: [-1.38777878e-17 5.88235294e-02 1.17647059e-01]

Epoch 87791
Bobot: [-1.17647059e-01 -5.88235294e-02 -2.77555756e-17]
Bobot: [-1.38777878e-17 5.88235294e-02 -1.17647059e-01]
Bobot: [ 1.17647059e-01 -5.88235294e-02 -2.77555756e-17]
Bobot: [-1.38777878e-17 5.88235294e-02 1.17647059e-01]
```

Untitled - Notepad

File Edit View

Dimas Tri Mustakim
205150200111049

Ln 2, Col 16 100% Windows (CRLF) UTF-8

C. Analisis

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

Jawab:

Parameter `alpha` berfungsi untuk mendeklarasikan seberapa besar bobot akan diperbarui atau decision boundary akan digeser pada setiap training. Parameter `max_err` berfungsi sebagai penentu kapan proses training akan dihentikan. Jika perubahan bobot tertinggi pada epoch lebih kecil dari `max_err`, maka proses pelatihan akan berhenti. Sedangkan parameter `max_epoch` merupakan parameter yang menentukan batas jumlah epoch yang bisa dilakukan pada proses pelatihan.

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

Jawab:

Jika `max_epoch` diberi nilai -1, berarti proses training tidak dibatasi untuk berapa kali epoch. Dapat dilihat di dalam kode program, terdapat while loop dengan kondisi yang menggunakan nilai `max_epoch`. Jika `max_epoch`nya adalah -1 maka akan seperti tidak diatur, dan jika `max_epoch`nya diatur, maka proses training akan tetap berjalan ketika jumlah epoch lebih kecil dari `max_epoch`.

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?

Jawab:

Jumlah epoch pada proses training untuk ketiga permasalahan tersebut tidak sama karena permasalahan yang ingin diklasifikasikan tidak sama dan nilai bobot awal yang digunakan oleh Adaline adalah random, sehingga membutuhkan epoch pelatihan yang berbeda.

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

Jawab:

Hal yang terjadi adalah yaitu proses pelatihan untuk data logika XOR tidak pernah berhenti meskipun telah mencapai epoch yang sangat besar. Hal tersebut karena Adaline juga merupakan yang hanya bisa mengklasifikasikan permasalahan yang *linearly separable* dan permasalahan XOR merupakan permasalahan yang *non linearly separable*.

Proses training yang tidak pernah selesai terjadi karena kondisi berhenti dari algoritma Adaline di program tersebut yaitu `max_err` dan `max_epoch`. `Max_epoch` tidak dibatasi dan `Max_err` tidak tercapai karena bobot selalu berubah setiap kali pelatihan dari minus ke positif dan sebaliknya seperti digambar dibawah yang menyebabkan nilai perubahan bobot cukup besar sehingga tidak pernah lebih kecil nilai `max_err`.

Epoch 49

Bobot: [-1.17647059e-01 -5.88235294e-02 -1.67978687e-11]

Bobot: [8.60360394e-12 5.88235294e-02 -1.17647059e-01]

Bobot: [1.17647059e-01 -5.88235294e-02 -7.86301579e-12]

Bobot: [1.15646659e-11 5.88235294e-02 1.17647059e-01]

Epoch 50

Bobot: [-1.17647059e-01 -5.88235294e-02 -1.06760850e-11]

Bobot: [5.14736864e-12 5.88235294e-02 -1.17647059e-01]

Bobot: [1.17647059e-01 -5.88235294e-02 -5.07824338e-12]

Bobot: [7.02074510e-12 5.88235294e-02 1.17647059e-01]

D. Kesimpulan

Adaline merupakan salah satu jenis algoritma jaringan saraf tiruan *single layer* yang terdiri dari beberapa input neuron dan satu output neuron (pengklasifikasi biner). Adaline memiliki bias yang memiliki nilai aktivasi konstan sebesar 1. Adaline memiliki algoritma pelatihannya sendiri untuk mengubah nilai-nilai bobot.

Fungsi dari threshold pada pelatihan Adaline adalah untuk menentukan waktu berhenti dari proses training. Jika perubahan bobot terbesar dalam satu epoch lebih kecil dari threshold, maka proses training akan dihentikan. Sedangkan learning rate

berfungsi untuk mengontrol seberapa banyak bobot dari jaringan disesuaikan untuk setiap proses training. Pada data yang kompleks, learning rate sangat penting agar pergeseran decision boundary tidak terlalu besar yang menyebabkan posisi decision boundary tersebut tidak baik.

Adaline tidak bisa menyelesaikan permasalahan logika XOR karena merupakan permasalahan yang termasuk not linearly separable. Adaline hanya bisa memisahkan data dengan satu buah garis linear, yang membuatnya tidak bisa melakukan klasifikasi pada permasalahan XOR. Dapat diamati juga pada saat proses training, terjadi kejadian bahwa proses training tersebut tidak akan selesai karena kondisi berhenti yaitu nilai perubahan bobot terbesar lebih kecil dari threshold tidak pernah terjadi.