LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : ADALINE
NAMA : ARION SYEMAEL SIAHAAN
NIM : 225150207111060
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 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,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
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=True,
max_epoch=500)
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

s

c. Fungsi testing Adaline

c) Fungsi *Testing* Adaline

Tulis kode ke dalam *cell* di bawah ini:

```
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
```

ARION SYEMAEL SIAHAAN
225150207111060

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)
```

ARION SYEMAEL SIAHAAN
225150207111060

e. Logika AND

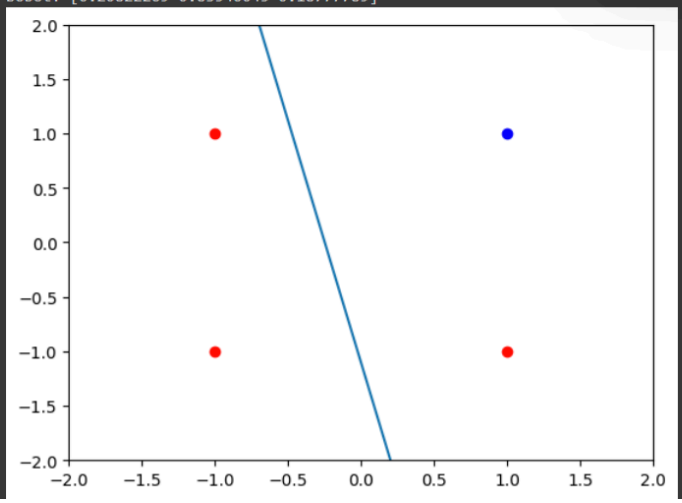
e) Logika AND

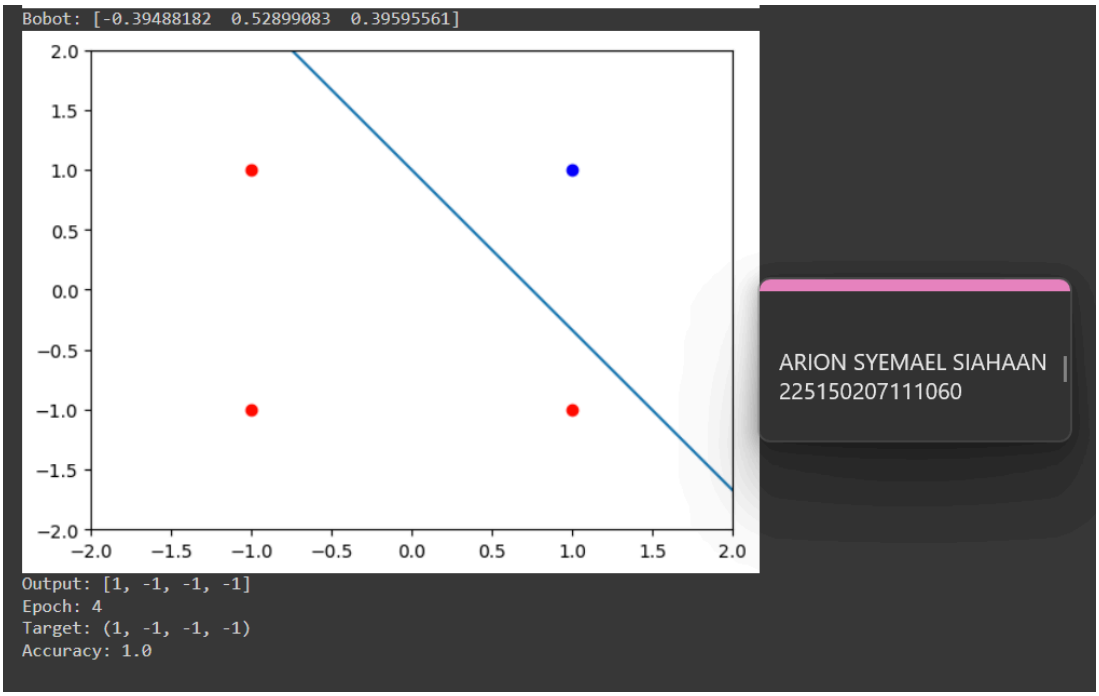
Tulis kode ke dalam *cell* di bawah ini:

```
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)
```

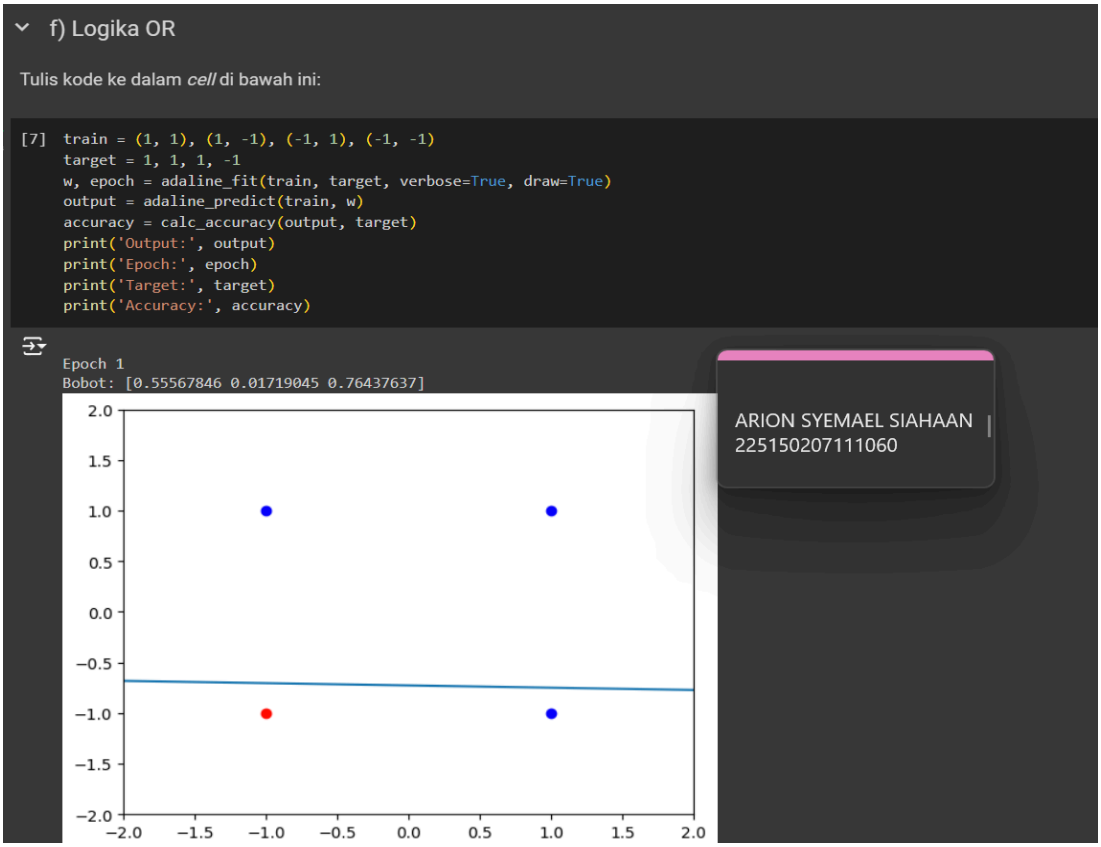
ARION SYEMAEL SIAHAAN
225150207111060

```
Epoch 1
Bobot: [0.20822269 0.83546045 0.18777789]
```

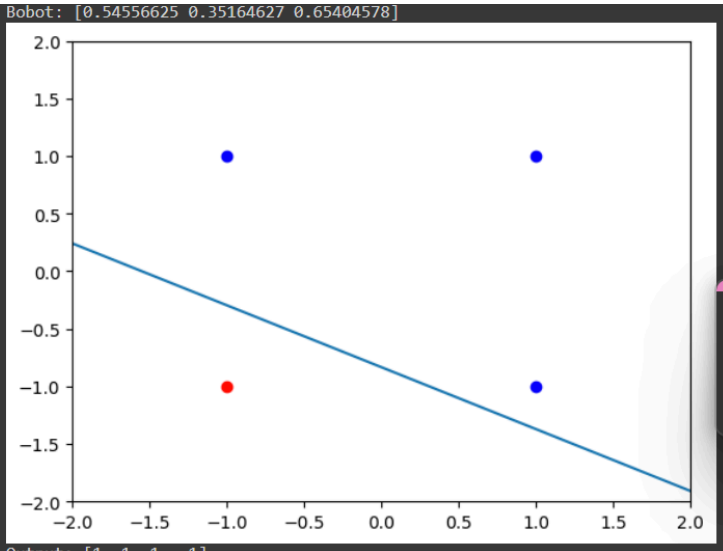



Bobot: [-0.39488182 0.52899083 0.39595561]
ARION SYEMAEL SIAHAAN
225150207111060
Output: [1, -1, -1, -1]
Epoch: 4
Target: (1, -1, -1, -1)
Accuracy: 1.0

f. Logika OR


f) Logika OR
Tulis kode ke dalam cell di bawah ini:
[7] 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: [0.55567846 0.01719045 0.76437637]
ARION SYEMAEL SIAHAAN
225150207111060

```
Bobot: [0.54556625 0.35164627 0.65404578]
```



ARION SYEMAEL SIAHAAN
225150207111060

```
Output: [1, 1, 1, -1]
Epoch: 2
Target: (1, 1, 1, -1)
Accuracy: 1.0
```

g. Logika AND NOT

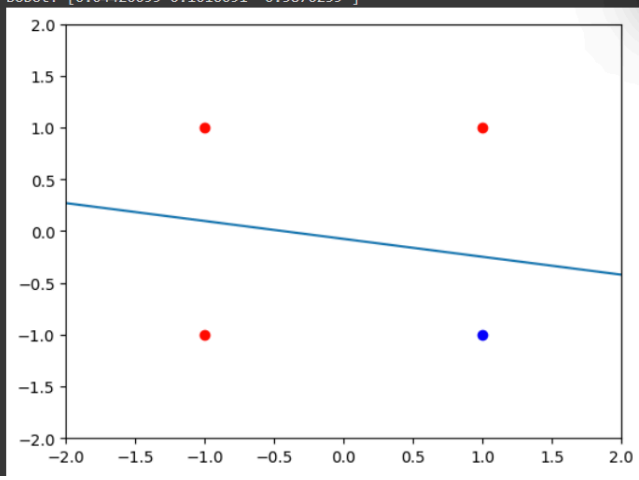
g) Logika AND NOT

Tulis kode ke dalam *cell* di bawah ini:

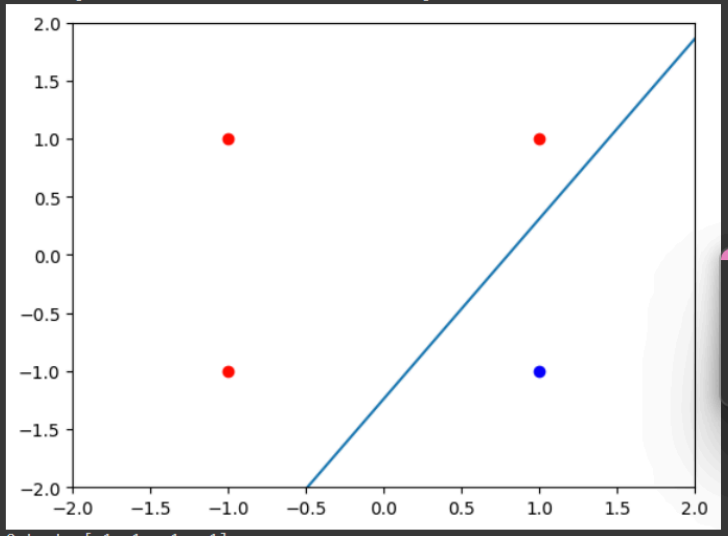
```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = -1, 1, -1, -1
w, epoch = adaline_fit(train, target, verbose=True, draw=True)
output = adaline_predict(train, w)
accuracy = calc_accuracy(output, target)

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

ARION SYEMAEL SIAHAAN
225150207111060

```
Epoch 1
Bobot: [0.04420099 0.1016091  0.5876259 ]
```



```
Bobot: [-0.41105576  0.51488732 -0.33216729]
```




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

ARION SYEMAEL SIAHAAN
225150207111060

h. Logika XOR

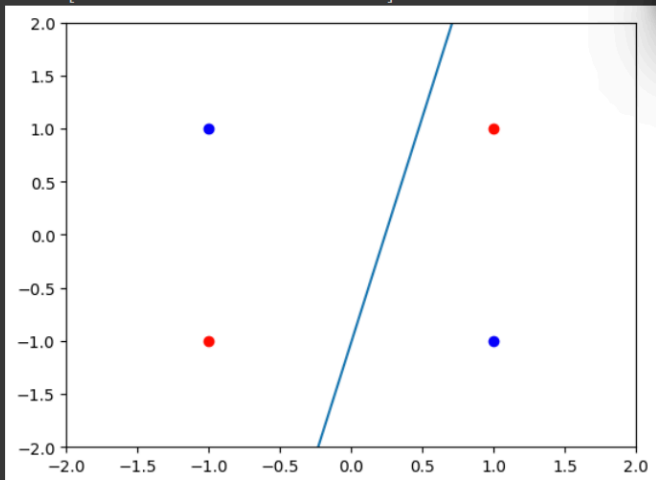
h) Logika XOR

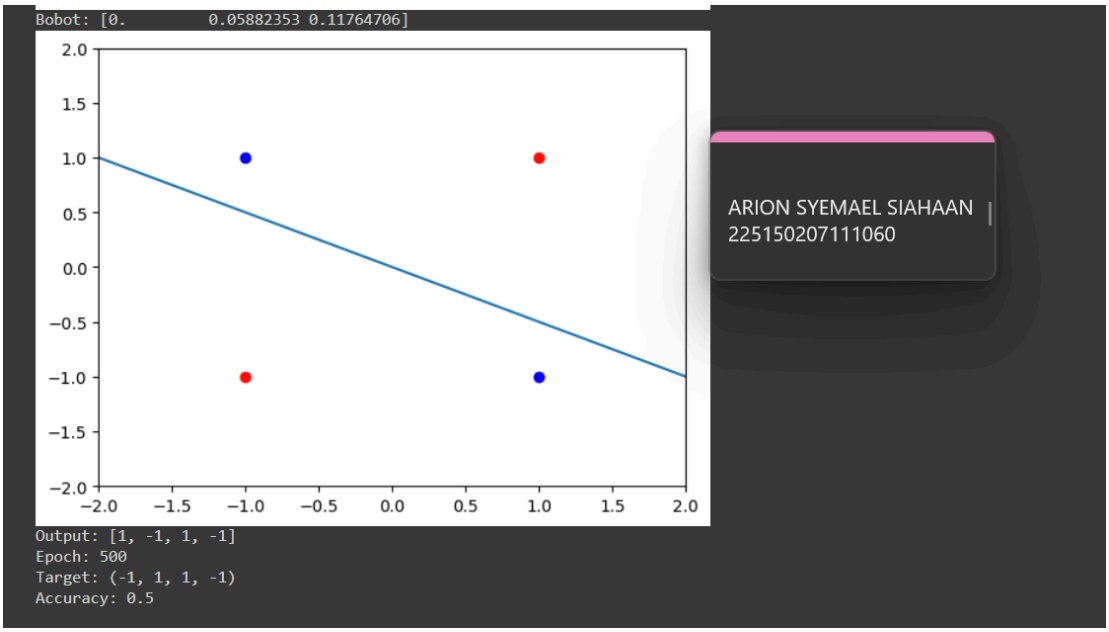
Tulis kode ke dalam *cell* di bawah ini:

```
train = (1, 1), (1, -1), (-1, 1), (-1, -1)
target = -1, 1, 1, -1
w, epoch = adaline_fit(train, target, verbose=True, draw=True, max_epoch=500)
output = adaline_predict(train, w)
accuracy = calc_accuracy(output, target)

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

ARION SYEMAEL SIAHAAN
225150207111060

```
Epoch 1
Bobot: [-0.12223943  0.51396195 -0.12099661]
```





C. Analisis

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

1.alpha (learning rate): Mengatur seberapa besar langkah perubahan bobot di setiap iterasi. Kecil = lambat, besar = cepat tapi bisa nggak stabil.

2.max_err (error maksimal): Menentukan seberapa kecil perubahan bobot sebelum pelatihan berhenti. Kalau perubahannya udah sangat kecil, berhenti.

3.max_epoch (maksimum epoch): Batas maksimal jumlah putaran (epoch) pelatihan. Jadi, kalau sudah mencapai jumlah ini, pelatihan akan berhenti, walau belum konvergen

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

Kalau max_epoch diisi -1, artinya looping akan jalan terus tanpa batas, sampai kondisi lain (seperti max_err) tercapai. Jadi, loop nggak akan berhenti hanya karena jumlah 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?

Jumlah epoch berbeda karena target dari tiap logika (AND, OR, AND NOT) berbeda, sehingga perubahan bobot (`delta w`) juga beda di setiap iterasi. Proses pelatihan berhenti ketika perubahan bobot sudah kecil (di bawah batas yang ditentukan), jadi logika dengan perubahan yang lebih cepat atau lambat akan membutuhkan epoch yang berbeda.

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

Ketika melatih data logika XOR, model Adaline tidak dapat menemukan solusi karena XOR tak dapat dipisahkan dengan garis lurus (non-linear). Adaline hanya bisa memisahkan data linear, jadi pelatihan terus berjalan tanpa hasil yang benar. Itulah kenapa XOR tidak cocok untuk Adaline.

D. Kesimpulan

Adaline (Adaptive Linear Neuron) adalah algoritma dalam jaringan saraf tiruan (JST) yang diperkenalkan oleh Bernard Widrow dan Ted Hoff pada tahun 1960. Adaline merupakan perbaikan dari Perceptron, menggunakan metode pembaruan bobot dengan aturan Least Mean Squares (LMS). Adaline beroperasi dengan menyesuaikan bobot berdasarkan kesalahan antara output aktual dan target, yang membuatnya lebih efektif untuk mempelajari pola linear. Secara struktural, Adaline memiliki beberapa neuron input dan satu neuron output dengan fungsi aktivasi linear.

Threshold digunakan sebagai nilai ambang yang menentukan kapan neuron akan mengeluarkan sinyal. Jika input terakumulasi melebihi nilai threshold, neuron akan mengeluarkan sinyal output, sebaliknya jika tidak, maka tidak ada output yang dikeluarkan. Threshold penting untuk mengontrol kapan keputusan dibuat dalam proses klasifikasi.

Learning rate diperlukan untuk mengatur kecepatan pembaruan bobot selama pelatihan. Jika learning rate terlalu kecil, proses pembelajaran akan berlangsung sangat lambat karena perubahan bobot di setiap iterasi sangat kecil. Jika terlalu besar, algoritma bisa mengalami osilasi atau bahkan gagal untuk konvergen karena perubahan bobot yang terlalu drastis. Dengan kata lain, learning rate menentukan seberapa besar penyesuaian bobot dilakukan berdasarkan kesalahan yang dihasilkan pada setiap iterasi.

Adaline tidak bisa menyelesaikan masalah logika XOR karena cara klasifikasinya bersifat linear. Masalah XOR tidak dapat diselesaikan oleh model yang hanya dapat

memisahkan data secara linear (linearly separable), seperti Adaline. Logika XOR membutuhkan model yang mampu menangani pemisahan data yang non-linear, yang tidak bisa dilakukan oleh Adaline. Algoritma Adaline hanya bisa menangani masalah yang dapat dipisahkan dengan garis lurus, sedangkan XOR membutuhkan model yang bisa memisahkan data dengan garis non-linear.