LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : MADALINE
NAMA : ARION SYEMAEL SIAHAAN
NIM : 225150207111060
TANGGAL : 02/10/2024
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. Import Modul

```
import numpy as np
```

b. Fungsi Aktivasi

```
def aktivasi(x):
  if x < 0:
    return -1
  else:
    return 1
```

c. Fungsi Training Madaline

```
def train(train_data, train_target, alpha=0.1, max_epoch=10):
    w = np.random.random((2, 2))
    v = np.array([0.5, 0.5])
    b = np.random.random(2)
    b = np.append(b, 0.5)
    epoch = 0
    v_aktivasi = np.vectorize(aktivasi)
    weight_updated = True

    while weight_updated and epoch < max_epoch:
        weight_updated = False

        for data, target in zip(train_data, train_target):
            z_in = np.dot(data, w)
            z_in = z_in + b[:-1]
            z = v_aktivasi(z_in)
            y_in = np.dot(z, v) + b[-1]
            y = v_aktivasi(y_in)

            if y != target:
                weight_updated = True
                if target == 1:
                    index = np.argmin(np.abs(z_in))
```

```
                    b[index] = b[index] + alpha * (1 - z_in[index])
                         w[:, index] = w[:, index] + alpha * (1 -
z_in[index]) * data
                elif target == -1:
                    index = np.where(z_in > 0)[0]
                    if len(index) == 1:
                        index = index[0]
                    b[index] = b[index] + alpha * (-1 - z_in[index])
                         w[:, index] = w[:, index] + alpha * (-1 -
z_in[index]) * data
        epoch += 1

    return (w, v, b)
```

d. Fungsi Testing Madaline

```
def test(w,v,b,test_data):
      v_aktivasi = np.vectorize(aktivasi)
      z_in = np.dot(test_data, w)
      z_in = z_in + b[:-1]
      z = v_aktivasi(z_in)
      y_in = np.dot(z, v) + b[-1]
      y = v_aktivasi(y_in)
      return y
```

e. 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)
```

f. Logika AND

```
data = np.array([[1, 1], [1, -1], [-1, 1], [-1, -1]])
target = np.array([1, -1, -1, -1])
(w, v, b) = train(data, target, alpha=0.8, max_epoch=10)
output = test(w, v, b, data)
accuracy = calc_accuracy(output, target)

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

g. Logika OR

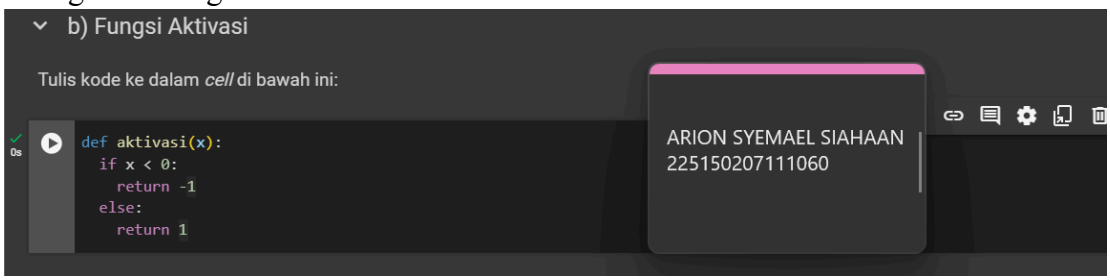
```
target = np.array([1, 1, 1, -1])
(w, v, b) = train(data, target, alpha=0.2, max_epoch=10)
output = test(w, v, b, data)
accuracy = calc_accuracy(output, target)

print('Output OR:', output)
```

```
print('Target OR:', target)
print('Accuracy OR:', accuracy)
```

B. Screenshot

a. Import Modul

b. Fungsi Training Madaline



c. Fungsi Training Madaline

d. Fungsi Testing Madaline

d) Fungsi *Testing* Madaline

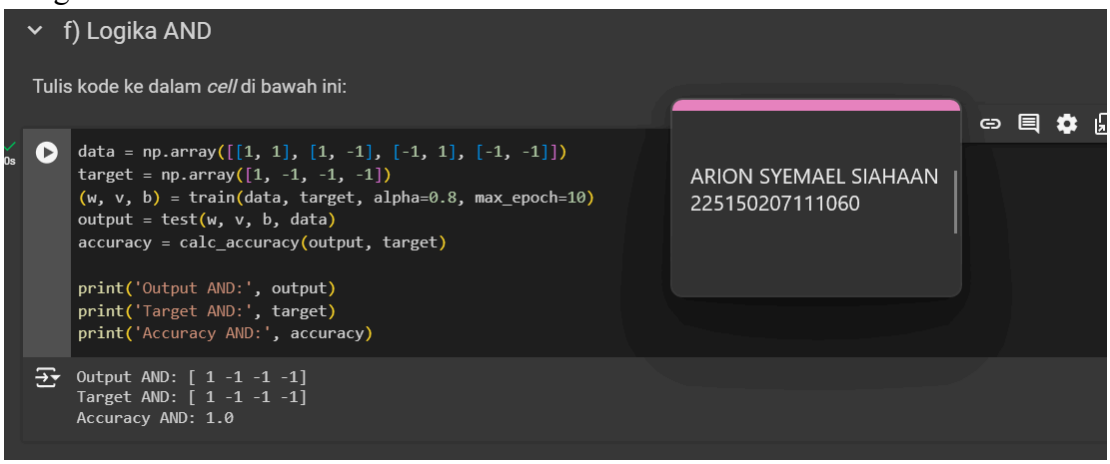
Tulis kode ke dalam *cell* di bawah ini:

```
[5]  def test(w,v,b,test_data):
         v_aktivasi = np.vectorize(aktivasi)
         z_in = np.dot(test_data, w)
         z_in = z_in + b[:-1]
         z = v_aktivasi(z_in)
         y_in = np.dot(z, v) + b[-1]
         y = v_aktivasi(y_in)
         return y
```

ARION SYEMAEL SIAHAAN
225150207111060

e. Fungsi Hitung Akurasi

f. Logika AND



g. Logika OR

C. Analisis

1. Berdasarkan source code yang ada, kapan proses training pada Madaline akan berhenti?

Proses training Madaline akan berhenti ketika model mencapai jumlah maksimum epoch yang telah ditentukan, atau ketika tidak ada lagi pembaruan bobot yang dilakukan. Ini menandakan bahwa model telah mencapai konvergensi dan prediksi model sesuai dengan target yang diinginkan.

2. Cobalah mengganti nilai alpha pada logika AND dengan rentang nilai 0,1 - 1.0. Apakah ada pengaruh alpha terhadap akurasi?

Nilai alpha mempengaruhi kecepatan konvergensi, bukan akurasi akhir. Jika alpha kecil, pembaruan bobot akan lebih lambat tetapi lebih stabil. Sebaliknya, jika alpha besar, pembaruan akan lebih cepat namun berpotensi menyebabkan overshooting, meskipun akurasi akhirnya akan tetap sama setelah mencapai konvergensi.

3. Apakah jaringan Madaline dapat mengenali logika XOR dengan tepat? Mengapa demikian?

Madaline tidak dapat mengenali logika XOR dengan tepat karena XOR adalah masalah non-linear, sedangkan Madaline pada dasarnya adalah model linear. Untuk memecahkan masalah seperti XOR, diperlukan jaringan dengan lapisan tersembunyi yang mampu memodelkan hubungan non-linear.

4. Ubahlah data dan target pada Logika OR menggunakan bilangan biner (bukan bipolar). Jangan lupa mengubah pula fungsi aktivasi agar menghasilkan bilangan biner. Apakah Madaline dapat mengenali Logika OR dengan data dan target biner? Mengapa demikian?

Jika data dan target pada logika OR diubah menjadi bilangan biner, dan fungsi aktivasi diubah untuk mengeluarkan bilangan biner (0 dan 1 alih-alih -1 dan 1), Madaline masih bisa mengenali logika OR. Hal ini karena logika OR adalah masalah linear, yang dapat dipecahkan oleh Madaline dengan baik. Perubahan dari representasi bipolar ke biner tidak mengubah linearitas masalah tersebut, sehingga Madaline tetap dapat mempelajarinya secara efektif asalkan fungsi aktivasi disesuaikan untuk bekerja dengan output biner (misalnya, 0 jika input < 0, dan 1 jika input ≥ 0).

D. Kesimpulan

Adaline (Adaptive Linear Neuron) adalah model dengan satu neuron yang menggunakan aturan delta untuk meminimalkan kesalahan prediksi. Model ini efektif untuk masalah yang dapat dipisahkan secara linear, di mana hubungan input-output bisa dipetakan dengan garis lurus. Sementara itu, Madaline (Multiple Adaptive Linear Neurons) lebih kompleks karena memiliki beberapa neuron yang tersusun dalam lapisan-lapisan. Madaline mampu menangani lebih banyak fitur input dan mempelajari pola yang lebih rumit dibandingkan Adaline, meskipun keduanya sama-sama menggunakan aturan pembaruan bobot untuk meminimalkan kesalahan.

Madaline cocok untuk mempelajari masalah-masalah yang bersifat linear atau hampir linear. Beberapa contohnya adalah logika AND dan OR, yang merupakan masalah linear sederhana. Selain itu, Madaline juga bisa digunakan dalam klasifikasi biner, seperti memprediksi apakah suatu objek aman atau berbahaya berdasarkan fitur-fitur yang bisa dipisahkan secara linear, seperti suhu dan tekanan dalam pengelompokan objek. Dalam kasus seperti ini, Madaline dapat mempelajari pola dengan baik asalkan batas pemisahnya linear.