

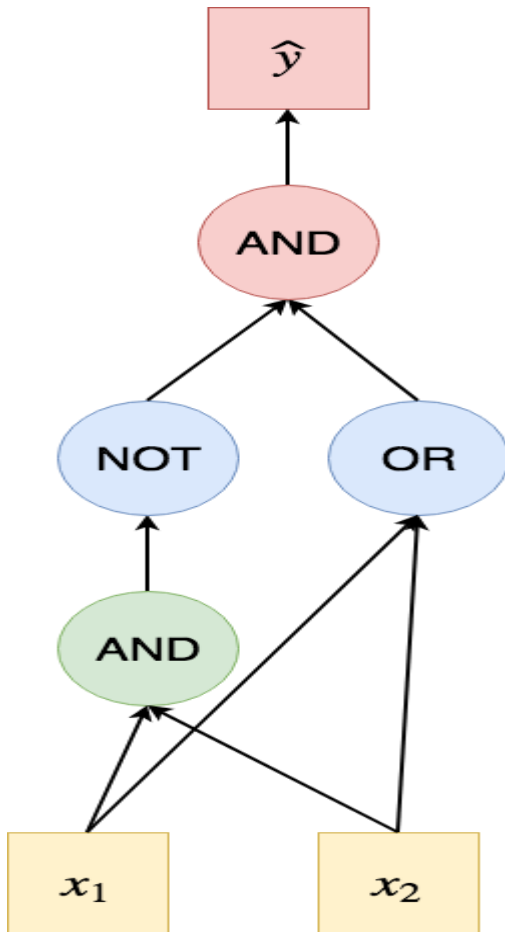
Ad-Soyad:Ömer Battek

Öğrenci No:170260614

Bilgisayar mühendiliği lap ödevi

Soru1:Xor için perceptron tasarlayın,başarınızı yorumlayın?

$$XOR(x_1, x_2) = \text{AND}(\text{NOT}(\text{AND}(x_1, x_2)), \text{OR}(x_1, x_2))$$



Xor elde etmek için OR ve And Kullanarak elde edebiliriz

X1	X2	And(x1,x2)	Not And(x1,x2)	OR(x1,x2)	And(NotAnd,OR)
0	0	0	1	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	0	1	0

Soru2:Öğrenci no son hanesine göre veri setini kullanarak:

A-Sınıflandırmayı gerçekleştir

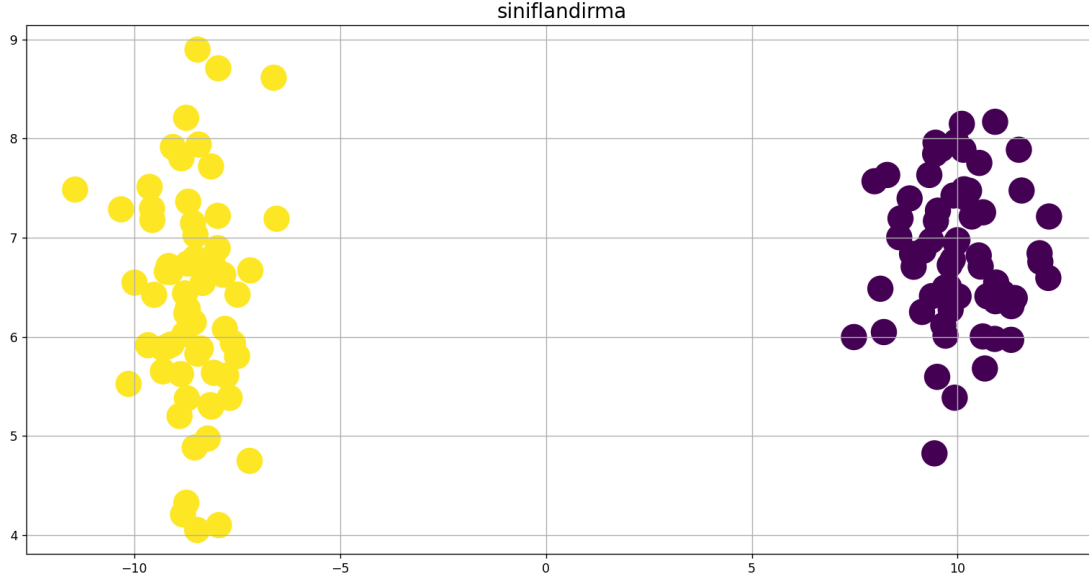
B-Kodları açıkla

C-Rasgele 5 örnek için elde ettiğin ağırlıklarının başarısının göster

D-Sonuçları yorumla

Çözüm:

Dosyadan 1. Ve 3. Özellikler sınıflandıracağım.



Yukarıdaki grafiğe göre sınıflandırma yapabileceğimiz kolayca anlaşılır.

Kod kısmı:

```
#gerekli kütüphaneler gir
```

```
import numpy as np
```

```
import pandas
```

```
from matplotlib import pyplot as plt
```

```
df = pandas.read_csv('data_456.csv', header=None)#gerekli dosya panda  
kütüphanesi kullanarak oku
```

```
data_giris = df.iloc[1:127, [1,3]].values #Dosyadan Özellik1 ve Özellik3 giriş  
olarak al
```

```
giris = data_giris.astype('float64') #veriler float tipine dönüştür
```

```
data_cikis = df.iloc[1:127, 6].values#çıkış veriler al
```

```
cikis = data_cikis.astype('float64')
```

```
#graf için gerekli kodlar
```

```
plt.title('siniflandirma', fontsize=16)
```

```
plt.scatter(giris[:,0], giris[:,1], s=400, c = cikis)
```

```
plt.grid()
```

```
plt.show()
```

```
class Perceptron(object):
```

```
    def __init__(self, ogrenme_orani=0.1, iter_sayisi=10):
```

```
        self.ogrenme_orani = ogrenme_orani
```

```
        self.iter_sayisi = iter_sayisi
```

```
    def ogren(self, X, y):
```

```
        self.w = np.zeros(1 + X.shape[1])#[0,0,0]oluşturup w ya at
```

```
        self.hatalar = []
```

```
        for _ in range(self.iter_sayisi):
```

```
            hata = 0
```

```
            for xi, hedef in zip(X, y):#xi ya X at ve hadefe Y at her seferinde xi iki  
                özellik1 ve özellik3 atacak, y ye ise çıkış değeri atacak
```

degisim = self.ogrenme_orani * (hedef - self.tahmin(xi))#öğrenme oranı 0.1 daha önce beliledik, eğer bilgisayar tahmin ettiği değer hedef değeri aynı ise değişim 0 olacak değil is degisime bir değer atacak

self.w[1:] += degisim * xi#eğer degisim 0 değil ise w1 ve w2 değerlerini değiştirecek

self.w[0] += degisim X#eğer değişim 0 değil ise denklemde b değerini değiştirecek(denklem = $w_1 \cdot x_1 + w_2 \cdot x_2 + b$)

hata += int(degisim != 0.0) #eğer değişim değeri 0 değil ise hata değişkene 1 arttırılacak 0 ise hata değerine bir şey olmayacak

self.hatalar.append(hata)#her iterasyonda kaç tane hata var elde etmek için bu satır kullanılıyor

return self

def net_input(self, X):

return np.dot(X, self.w[1:]) + self.w[0]# $y = x_1 \cdot w_1 + x_2 \cdot w_2 + b$

def tahmin(self, x):

return np.where(self.net_input(x) >= 0.0, 1, 0)#eğer net_input değeri büyük eşit sıfır ise 1 return 0 dan küçük ise 0 return

siniflandirici = Perceptron(ogrenme_orani=0.1, iter_sayisi=10)#nesne tanımla

siniflandirici.ogren(giris, cikis)#ogren fonksiyonu çağır

print(siniflandirici.w)#ağırlı yaz [b=-0.1 , w1= -0.97058155 ,w2=-0.60121962]

print(siniflandirici.hatalar)# [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

$$y = w1 \cdot x1 + w2 \cdot x2 + b$$

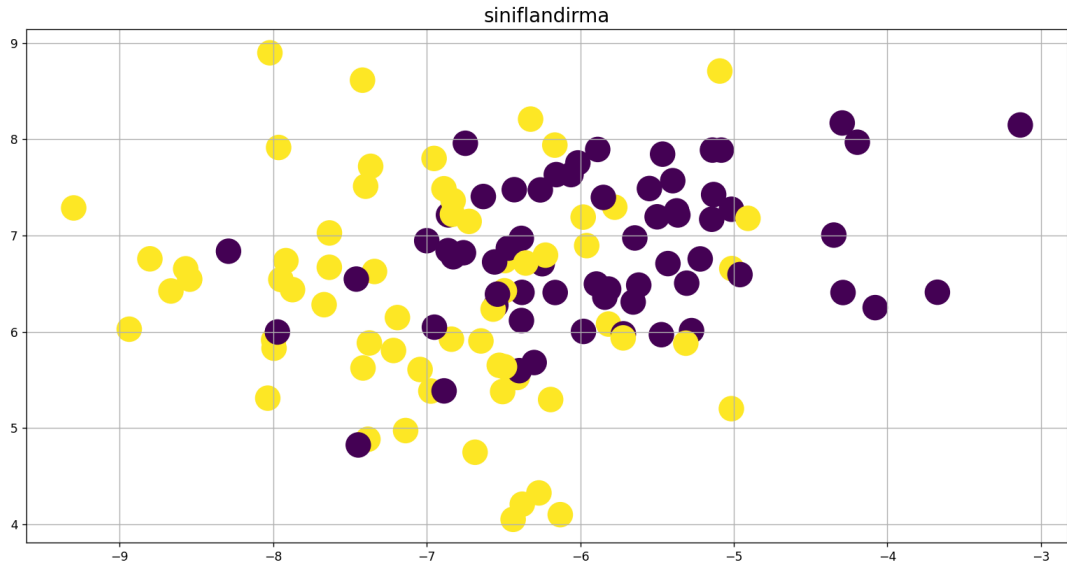
X1	X2	W1	W2	b	net	y	çıkış
9.691730829197661	6.496519111386209	-0.97058155	0.60121962	-0.1	-5.605	0	0
12.220106132644755	7.214077386407947	-0.97058155	0.60121962	-0.1	-7.627	0	0
10.723326054371057	6.409550908182716	-0.97058155	0.60121962	-0.1	-6.658	0	0
7.981281196449051	7.569833142967	-0.97058155	0.60121962	-0.1	-3.304	0	0
-8.513329664873226	7.027073064258198	-0.97058155	0.60121962	-0.1	12.366	1	1

$$Y = 1, \text{net} \geq 0 \quad | \quad 0, \text{net} < 0$$

Yorum:

Sınıflandırma her zaman elde edilmeyebilir.

Örneğin veri setimizde eğer özellik 2 ve özellik 3 alırsak aşağıdaki graf çıkacak.



Bu graf sınıflandırması imkansız ve ondan bir sonuca varamayız