

- 1) Bir cam üretim tesisinde üretilen bardakların düzgün, hatalı veya kırık olarak sınıflandırılması için bir model oluşturulması istenmektedir. Düzgün durumu için  $D = [1 \ 0 \ 0]$  girişli "1" çıkışlı, hatalı durumu için  $H = [0 \ 1 \ 0]$  girişli "0.5" çıkışlı, kırık durumu için ise  $K = [0 \ 0 \ 1]$  girişli "0" çıkışlı durumlar belirlenmiştir. Model için verilen başlangıç parametreleri ise şu şekildedir;

$$w_1 = 0.5, w_2 = 0.4, w_3 = 0.5, \phi = 0.1, a = 0.3$$

Aktivasyon fonksiyonu;

$net > 0.5 \rightarrow 1, net = 0.5 \rightarrow 0.5, net < 0.5 \rightarrow 0$  olarak belirlenmiştir.

- a) Bu modelin doğru sınıflandırma yapabilmesi için ağırlık eğitimi sonucunda elde edilen yeni ağırlık değerleri ve eşik değerini hesaplayınız.

$$\begin{aligned} D &= [1 \ 0 \ 0] & a_{1k} &= 1 & w_1 &= 0.5, w_2 = 0.4, w_3 = 0.5, \phi = 0.1, a = 0.3 \\ H &= [0 \ 1 \ 0] & a_{1k} &= 0.5 & net > 0.5 &\rightarrow 1, net = 0.5 \rightarrow 0.5, net < 0.5 \rightarrow 0 \\ K &= [0 \ 0 \ 1] & a_{1k} &= 0 \end{aligned}$$

a-)  
Düzgün

$$\begin{aligned} net &= w_1 x_1 + w_2 x_2 + w_3 x_3 + \phi \\ &= 0.5(1) + 0.4(0) + (0.5)(0) + 0.1 = 0.6 \end{aligned}$$

$net > 0.6 \rightarrow 1$  gerçek durum = 1 ve beklenen durum = 1 *sağlandı ✓*

Hatalı

$$net = 0.5(0) + 0.4(1) + (0.5)(0) + 0.1 = 0.5$$

$net = 0.5 \rightarrow 0.5$  gerçek durum = 0.5 ve beklenen durum = 0.5 *sağlandı ✓*

Kırık

$$net = 0.5(0) + 0.4(0) + (0.5)(1) + 0.1 = 0.6$$

$net > 0.5 \rightarrow 1$  gerçek durum = 1 ve beklenen durum = 0 *sağlanmadı ✗*

yeni değerler bul

$$\bar{E} = (\text{beklenen}) - (\text{gerçek}) = 0 - 1 = -1$$

$$w_{1-\text{yeni}} = w_1 + a \bar{E} \cdot K[1,1] = 0.5 + 0.3(-1)(1) = 0.2$$

$$w_{2-\text{yeni}} = w_2 + a \bar{E} \cdot K[1,2] = 0.4 + 0.3(-1)(0) = 0.4$$

$$w_{3-\text{yeni}} = w_3 + a \bar{E} \cdot K[1,3] = 0.5 + 0.3(-1)(1) = 0.2$$

$$\phi_{\text{yeni}} = \phi + a \bar{E} = 0.1 + 0.3(-1) = -0.2$$

Kırık

$$net = 0.5(0) + 0.4(0) + 0.2(1) + (-0.2) = 0$$

$net < 0.5$  gerçek durum = 0 ve beklenen durum = 0 *sağlandı ✓*

Düzgün

$$net = 0.5(1) + 0.4(0) + 0.2(0) + (-0.2) = 0.3$$

$net < 0.5$  gerçek durum = 0 ve beklenen durum = 1 *sağlanmadı ✗*

yeni değerler bulun

$$\bar{E} = (\text{beklenen}) - (\text{gerçek}) = 1 - 0 = 1$$

$$w_{1-yeni} = w_1 + \alpha E D \Sigma_{i,j} = 0.5 + 0.3(1)(1) = 0.8$$

$$w_{2-yeni} = w_2 + \alpha E D \Sigma_{i,j} = 0.4 + 0.3(1)(0) = 0.4$$

$$w_{3-yeni} = w_3 + \alpha E D \Sigma_{i,j} = 0.2 + 0.3(1)(0) = 0.2$$

$$\phi_{-yeni} = \phi + \alpha E = -0.2 + 0.3(1) = 0.1$$

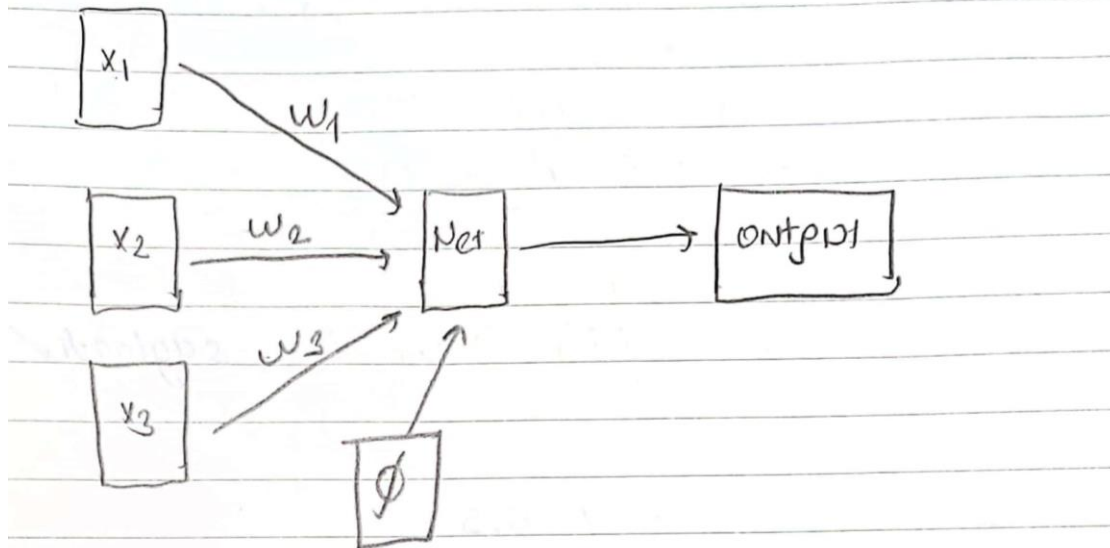
Diye  $net = 0.8(1) + 0.4(0) + 0.2(0) + 0.1 = 0.9$   
 $net > 0.5$  gerçek durum = 1 ve beklenen durum = 1 sağlandı ✓

Hıta  $net = 0.8(0) + 0.4(1) + 0.2(0) + 0.1 = 0.5$   
 $net = 0.5$  gerçek durum = 0.5 ve beklenen durum = 0.5 sağlandı ✓

Kıta  $net = 0.8(0) + 0.4(0) + 0.2(1) + 0.1 = 0.3$   
 $net < 0.5$  gerçek durum = 0 ve beklenen durum = 0 sağlandı ✓

Yeni ağırlık ve bias değerleri;  $w_1 = 0.8$  ,  $\phi = 0.1$  ,  
 $w_2 = 0.4$   
 $w_3 = 0.2$

b) Yapay sinir ağı modelini çiziniz.



Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

**c) Problemin çözümüne ilişkin olarak Matlab ve Python dilinde kodunu yazınız.**

## MATLAB

```
clc;
clear;
clear all;

%durumların tanitilmasi
%-----

B = [1 0 0; %duzgun olmasi
     0 1 0; %hatali olmasi
     0 0 1]; %kirik olmasi

beklenen_durumlar = [1 0.5 0]; %beklenen durumların sirayla ciktilari

%agirlik, esik ve ogrenme katsayilerinin tanitilmasi
%-----
W = [0.5 0.4 0.5];
fi = 0.1;
a = 0.3;

i = 1;

for sayici = 0 : 2

    while i < 4
        net = W(1,1)*B(i,1) + W(1,2)*B(i,2) + W(1,3)*B(i,3) + fi;
        fprintf('net : %f \n', net);
        if(net > 0.5)
            net = 1;
        elseif(net == 0.5)
            net = 0.5;
        elseif(net < 0.5)
            net = 0;
        end

        E = beklenen_durumlar(i) - net;
        fprintf('E : %f \n', E);
        if(E ~= 0) %esit degilse 0 a hatali durum
            fprintf('yeni degerler hesaplaniyor\n');
            fi = fi + a * E;
            for j = 1: 3
                W(1,j) = W(1,j) + a * E * B(i,j);
            end
            fprintf('bulunan degerler: w1:%f w2:%f w3:%f fi:%f \n', W(1,1), W(1,2), W(1,3), fi);
            i = 1;

        elseif (E == 0)
            fprintf('hata yok \n');
            i = i + 1;
        end
    end

end

fprintf('bulunan yeni degerler: w1:%f w2:%f w3:%f fi:%f \n', W(1,1), W(1,2), W(1,3), fi);
```

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

## SONUCLAR

### Command Window

```
net : 0.600000
E : 0.000000
hata yok
net : 0.500000
E : 0.000000
hata yok
net : 0.600000
E : -1.000000
yeni degerler hesaplaniyor
bulunan deęerler: w1:0.500000 w2:0.400000 w3:0.200000 fi:-0.200000
net : 0.300000
E : 1.000000
yeni degerler hesaplaniyor
bulunan deęerler: w1:0.800000 w2:0.400000 w3:0.200000 fi:0.100000
net : 0.900000
E : 0.000000
hata yok
net : 0.500000
E : 0.000000
hata yok
net : 0.300000
E : 0.000000
hata yok
bulunan yeni deęerler: w1:0.800000 w2:0.400000 w3:0.200000 fi:0.100000
fx >> |
```

## MATLAB ARAYUZU

hafta\_3\_deneme.m

hafta\_4.m

+

```
1 - clc;
2 - clear;
3 - clear all;
4
5 %durumların tanitilmesi
6 %-----
7
8 - B = [1 0 0; %duzgun olmasi
9         0 1 0; %hatali olmasi
10        0 0 1]; %kirik olmasi
11
12
13
14 - beklenen_durumlar = [1 0.5 0]; %beklenen durumların sirayla ciktilari
15
16 %agirlik, esik ve ogrenme katsayilerinin tanitilmesi
17 %-----
18 - W = [0.5 0.4 0.5];
19 - fi = 0.1;
20 - a = 0.3;
21
22 - i = 1;
```

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

```
24 - for sayici = 0 : 2
25 -
26 -
27 -     while i < 4
28 -         net = W(1,1)*B(i,1) + W(1,2)*B(i,2) + W(1,3)*B(i,3) +fi;
29 -         fprintf('net : %f \n' , net);
30 -         if(net > 0.5)
31 -             net = 1;
32 -         elseif(net == 0.5)
33 -             net = 0.5;
34 -         elseif(net < 0.5)
35 -             net = 0;
36 -         end
37 -
38 -         E = beklenen_durumlar(i) - net;
39 -         fprintf('E : %f \n' , E);
40 -         if(E ~= 0) %esit degilse 0 a hatali durum
41 -             fprintf('yeni degerler hesaplaniyor\n');
42 -             fi = fi + a * E;
43 -             for j = 1: 3
44 -                 W(1,j) = W(1,j) + a * E * B(i,j);
45 -             end
46 -             fprintf('bulunan değerler: w1:%f w2:%f w3:%f fi:%f \n' ,W(1,1),W(1,2),W(1,3),fi);
47 -             i = 1;
48 -
49 -         elseif (E == 0)
50 -             fprintf('hata yok \n');
51 -             i = i + 1;
52 -         end
53 -     end
54 - end
55 -
56 -
57 - fprintf('bulunan yeni değerler: w1:%f w2:%f w3:%f fi:%f \n' ,W(1,1),W(1,2),W(1,3),fi);
```

## PYTHON

"durumların tanitilmesi"

"-----"

```
B = [[1.0, 0.0, 0.0],
      [0.0, 1.0, 0.0],
      [0.0, 0.0, 1.0]];
```

```
net = 0.0;
```

```
beklenen_durumlar = [1, 0.5, 0]; "beklenen durumların sırayla çıktıları"
```

"ağırlık, esik ve öğrenme katsayılarının tanitilmesi"

"-----"

```
W = [0.5, 0.4, 0.5];
```

```
fi = 0.1;
```

```
a = 0.3;
```

```
for x in range(1):
```

```
    i = 0;
```

```
    while i < 3 :
```

```
        net = W[0] * B[i][0] + W[1] * B[i][1] + W[2] * B[i][2] +fi;
```

```
        print('net : ' , net, '\n');
```

```
        if(net > 0.5):
```

```
            net = 1;
```

```
        elif(net == 0.5):
```

```
            net = 0.5;
```

```
        elif(net < 0.5):
```

```
            net = 0;
```

```
    E = beklenen_durumlar[i] - net;
```

```
    print('hata : ' , E, '\n');
```

```
    if(E != 0):
```

```
        print('yeni degerler hesaplaniyor');
```

```
        fi = fi + a * E;
```

```
        for j in range(3):
```

```
            W[j] = W[j] + a * E * B[i][j];
```

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

```
print('bulunan degerler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'fi : ', fi);  
i = 0;  
elif (E == 0):  
    print('hata yok');  
    i = i + 1;
```

```
print('bulunan yeni degerler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'fi : ', fi);
```

## SONUC

```
In [12]: runfile('C:/Users/Onder/Desktop/ODEVLER/yapay_zeka_odevler/4.hafta/  
Users/Onder/Desktop/ODEVLER/yapay_zeka_odevler/4.hafta')  
net : 0.6  
  
hata : 0  
  
hata yok  
net : 0.5  
  
hata : 0.0  
  
hata yok  
net : 0.6  
  
hata : -1  
  
yeni degerler hesaplaniyor  
bulunan degerler: w1 : 0.5 w2 : 0.4 w3 : 0.2 fi : -0.19999999999999998  
net : 0.30000000000000004  
  
hata : 1  
  
yeni degerler hesaplaniyor  
bulunan degerler: w1 : 0.8 w2 : 0.4 w3 : 0.2 fi : 0.1  
net : 0.9  
  
hata : 0  
  
hata yok  
net : 0.5  
  
hata : 0.0  
  
hata yok  
net : 0.30000000000000004  
  
hata : 0  
  
hata yok  
bulunan yeni degerler: w1 : 0.8 w2 : 0.4 w3 : 0.2 fi : 0.1
```

## PYTHON ARAYÜZÜ

```

1  "durumların tanıtılması"
2  "-----"
3  B = [[1.0, 0.0, 0.0],
4        [0.0, 1.0, 0.0],
5        [0.0, 0.0, 1.0]];
6  net = 0.0;
7  beklenen_durumlar = [1, 0.5, 0]; "beklenen durumların sırayla çıktıları"
8
9
10 "ağırlık, eşik ve öğrenme katsayılarının tanıtılması"
11 "-----"
12 W = [0.5, 0.4, 0.5];
13 fi = 0.1;
14 a = 0.3;
15
16 for x in range(1):
17
18     i = 0;
19     while i < 3 :
20         net = W[0] * B[i][0] + W[1] * B[i][1] + W[2] * B[i][2] + fi;
21         print('net : ', net, '\n');
22         if(net > 0.5):
23             net = 1;
24         elif(net == 0.5):
25             net = 0.5;
26         elif(net < 0.5):
27             net = 0;
28
29         E = beklenen_durumlar[i] - net;
30         print('hata : ', E, '\n');
31
32         if(E != 0):
33             print('yeni değerler hesaplanıyor');
34             fi = fi + a * E;
35             for j in range(3):
36                 W[j] = W[j] + a * E * B[i][j];
37             print('bulunan değerler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'fi : ', fi);
38             i = 0;
39         elif (E == 0):
40             print('hata yok');
41             i = i + 1;
42
43
44 print('bulunan yeni değerler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'fi : ', fi);

```

2) 4 girişli 1 çıkışlı bir yapay sinir ağı modeli tasarlayarak; matematiksel çözümünü, model çizimini, matlab/python kodunu yazınız.

Bir kutu içerisinde bulunun kırmızı, mavi, yeşil, sarı olmak üzere 4 çeşit top rengi bulunmaktadır. Kutudan çekilen topların rengine göre bir model oluşturulması istenmektedir. Kırmızı için  $K = [1 \ 0 \ 0 \ 0]$  girişli "1" çıkışlı, Mavi durumu için  $M = [0 \ 1 \ 0 \ 0]$  girişli "0.8" çıkışlı, Yeşil durumu için  $Y = [0 \ 0 \ 1 \ 0]$  girişli "0.4" çıkışlı, Sarı durumu için  $S = [0 \ 0 \ 0 \ 1]$  girişli "0" çıkışlı durumlar belirtilmiştir. Model için başlangıç parametreleri şu şekildedir;

$w_1 = 0.7, w_2 = 0.8, w_3 = 0.3, w_4 = -0.2, f_i = 0.1, a = 0.9$

Aktivasyon Fonksiyonu

$net > 0.8$  ise 1;

$net \leq 0.8 \ \&\& \ net > 0.4$  ise 0.8;

$net \leq 0.4 \ \&\& \ net > 0$  ise 0.3;

$net \leq 0$  ise 0 olarak belirlenmiştir.



Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

MATEMATİKSEL ÇÖZÜM

$$\begin{array}{llll} K = [1 & 0 & 0 & 0] & c_k = 1 & w_1 = 0.7 & \phi = 0.1 \\ M = [0 & 1 & 0 & 0] & a_k = 0.8 & w_2 = 0.8 & q = 0.9 \\ Y = [0 & 0 & 1 & 0] & c_k = 0.4 & w_3 = 0.3 & \\ S = [0 & 0 & 0 & 1] & c_k = 0 & w_4 = -0.2 & \end{array}$$

aktüasyon fonks:  $net > 0.8 \rightarrow 1$   
 $0.8 > net > 0.4 \rightarrow 0.8$   
 $0.4 > net > 0 \rightarrow 0.4$   
 $net \leq 0 \rightarrow 0$

$\checkmark$   $net = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + \phi$   
 $= (0.7)(1) + 0.8(0) + 0.3(0) + (-0.2)(0) + 0.1 = 0.8$   
 $0.8 > net > 0.4 \rightarrow$  gerçek değer = 0.8 beklenen değer = 1 *saglanmadi X*

$$E = 8 - 6 = 1 - 0.8 = 0.2$$

$$\begin{aligned} w_{1-yeni} &= w_1 + aE x_1 = 0.7 + 0.9(0.2)(1) = \underline{0.88} \\ w_{2-yeni} &= w_2 + aE x_2 = 0.8 + 0.9(0.2)(0) = \underline{0.8} \\ w_{3-yeni} &= w_3 + aE x_3 = 0.3 + 0.9(0.2)(0) = \underline{0.3} \\ w_{4-yeni} &= w_4 + aE x_4 = -0.2 + 0.9(0.2)(0) = \underline{-0.2} \\ \phi_{yeni} &= \phi + aE = 0.1 + 0.9(0.2) = \underline{0.28} \end{aligned}$$

$\checkmark$   $net = (0.88)(1) + 0.8(0) + 0.3(0) + (-0.2)(0) + 0.28 = 1.16$   
 $net > 0.8$  gerçek durum = 1 ve beklenen durum = 1 *saglandı ✓*

$\checkmark$   $net = (0.88)(0) + 0.8(1) + 0.3(0) + (-0.2)(0) + 0.28 = 1.08$   
 $net > 0.8$  gerçek durum = 1 ve beklenen durum = 0.8 *saglanmadi X*

$$\bar{E} = 8 - 6 = 0.8 - 1 = -0.2$$

$$\begin{aligned} w_{1-yeni} &= 0.88 + 0.9(-0.2)(0) = \underline{0.88} \\ w_{2-yeni} &= 0.8 + 0.9(-0.2)(1) = \underline{0.62} \\ w_{3-yeni} &= 0.3 + 0.9(-0.2)(0) = \underline{0.3} \\ w_{4-yeni} &= -0.2 + 0.9(-0.2)(0) = \underline{-0.2} \\ \phi_{yeni} &= 0.28 + 0.9(-0.2) = \underline{0.1} \end{aligned}$$



$\checkmark$   $net = 10.88(1) + 0.02(0) + 0.3(0) + (-0.2)(0) + 0.1 = 0.98$   
 $net > 0.8$  gercek deger = 1 ve beklenen deger = 1 saglandi  $\checkmark$

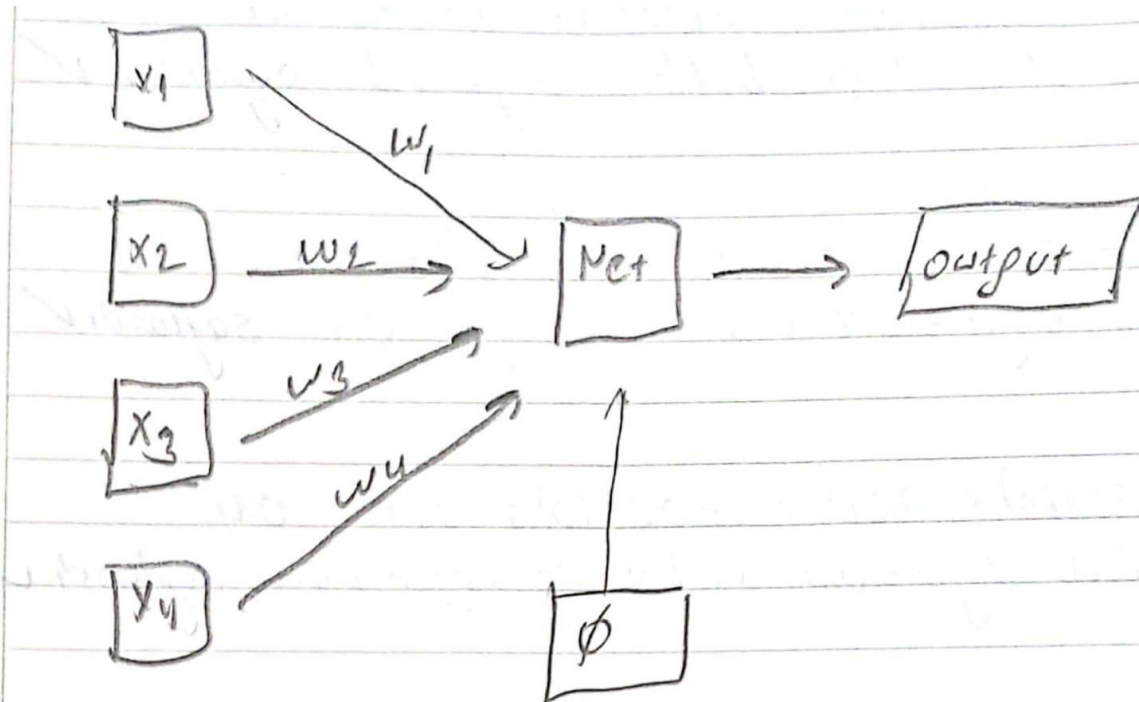
$M$   $net = 10.88(0) + 0.62(1) + 0.3(0) + (-0.2)(0) + 0.1 = 0.72$   
 $0.8 \geq net > 0.4$  gercek deger = 0.8 ve beklenen deger = 0.8 saglandi  $\checkmark$

$4$   $net = 10.88(0) + 0.62(0) + 0.3(1) + (-0.2)(0) + 0.1 = 0.4$   
 $0.4 \geq net > 0$  gercek deger = 0.4 ve beklenen deger = 0.4 saglandi  $\checkmark$

$S$   $net = 10.88(0) + 0.62(0) + 0.3(0) + (-0.2)(0) + 0.1 = -0.1$   
 $net < 0$  gercek deger = 0 ve beklenen deger = 0 saglandi  $\checkmark$

! yeni ağırlık ve eşik değerleri  $w_1 = 0.88$   $\phi = 0.1$   
 $w_2 = 0.62$   $\alpha = 0.9$   
 $w_3 = 0.3$   
 $w_4 = -0.2$

### YAPAY SİNİR AĞI MODELİ



Öğrenci No: 171110001  
Adı Soyadı: CAN AHMET ACAR

## MATLAB

```
clc;
clear;
clear all;

%durumların tanitilmasi
%-----

B = [1 0 0 0; %kirmizi durumu
      0 1 0 0; %mavi durumu
      0 0 1 0; %yesil durumu
      0 0 0 1]; %sari durumu

beklenen_durumlar = [1 0.8 0.4 0]; %beklenen durumların sirayla ciktıları

%agirlik, esik ve ogrenme katsayilerinin tanitilmasi
%-----

W = [0.7 0.8 0.3 -0.2];
fi = 0.1;
a = 0.9;

i = 1;

for sayici = 0 : 2

    while i < 5
        net = W(1,1)*B(i,1) + W(1,2)*B(i,2) + W(1,3)*B(i,3) + W(1,4)*B(i,4) + fi;
        fprintf('net : %f \n' , net);
        if(net > 0.8)
            net = 1;
        elseif(net <= 0.8 && net > 0.4)
            net = 0.8;
        elseif(net <= 0.4 && net > 0)
            net = 0.4;
        elseif(net <= 0)
            net = 0;
        end

        E = beklenen_durumlar(i) - net;
        fprintf('E : %f \n' , E);
        if(E ~= 0) %esit degilse 0 a hatali durum
            fprintf('yeni degerler hesaplanıyor\n');
            fi = fi + a * E;
            for j = 1: 4
                W(1,j) = W(1,j) + a * E * B(i,j);
            end
            fprintf('bulunan deęerler: w1:%f w2:%f w3:%f w4:%f fi:%f \n' ,W(1,1),W(1,2),W(1,3),W(1,4),fi);
            i = 1;

        elseif (E == 0)
            fprintf('hata yok \n');
            i = i + 1;
        end

    end

end

fprintf('bulunan yeni deęerler: w1:%f w2:%f w3:%f w4:%f fi:%f \n' ,W(1,1),W(1,2),W(1,3),W(1,4),fi);
```

**SONUCLAR**

```

net : 0.800000
E : 0.200000
yeni degerler hesaplaniyor
bulunan değerler: w1:0.880000 w2:0.800000 w3:0.300000 w4:-0.200000 fi:0.280000
net : 1.160000 |
E : 0.000000
hata yok
net : 1.080000
E : -0.200000
yeni degerler hesaplaniyor
bulunan değerler: w1:0.880000 w2:0.620000 w3:0.300000 w4:-0.200000 fi:0.100000
net : 0.980000
E : 0.000000
hata yok
net : 0.720000
E : 0.000000
hata yok
net : 0.400000
E : 0.000000
hata yok
net : -0.100000
E : 0.000000
hata yok
bulunan yeni değerler: w1:0.880000 w2:0.620000 w3:0.300000 w4:-0.200000 fi:0.100000

```

**MATLAB ARAYUZU**

```

1 -   clc;
2 -   clear;
3 -   clear all;
4
5   %durumların tanitilmesi
6   %-----
7
8 -   B = [1 0 0 0; %kirmizi durumu
9         0 1 0 0; %mavi durumu
10        0 0 1 0; %yesil durumu
11        0 0 0 1]; %sari durumu
12   |
13 -   beklenen_durumlar = [1 0.8 0.4 0]; %beklenen durumların sirayla ciktilari
14
15   %ağırlık, esik ve öğrenme katsayılarının tanitilmesi
16   %-----
17 -   W = [0.7 0.8 0.3 -0.2];
18 -   fi = 0.1;
19 -   a = 0.9;
20
21 -   i = 1;

```

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

```
23 - for sayici = 0 : 2
24 -
25 -     while i < 5
26 -         net = W(1,1)*B(i,1) + W(1,2)*B(i,2) + W(1,3)*B(i,3) + W(1,4)*B(i,4) + fi;
27 -         fprintf('net : %f \n' , net);
28 -         if(net > 0.8)
29 -             net = 1;
30 -         elseif(net <= 0.8 && net > 0.4)
31 -             net = 0.8;
32 -         elseif(net <= 0.4 && net > 0)
33 -             net = 0.4;
34 -         elseif(net <= 0)
35 -             net = 0;
36 -         end
37 -
38 -         E = beklenen_durumlar(i) - net;
39 -         fprintf('E : %f \n' , E);
40 -         if(E ~= 0) %esit degilse 0 a hatali durum
41 -             fprintf('yeni degerler hesaplaniyor\n');
42 -             fi = fi + a * E;
43 -             for j = 1: 4
44 -                 W(1,j) = W(1,j) + a * E * B(i,j);
45 -             end
46 -             fprintf('bulunan deęerler: w1:%f w2:%f w3:%f w4:%f fi:%f \n' ,W(1,1),W(1,2),W(1,3),W(1,4),fi);
47 -             i = 1;
48 -
49 -         elseif (E == 0)
50 -             fprintf('hata yok \n');
51 -             i = i + 1;
52 -         end
53 -     end
54 - end
55 -
56 -
57 - fprintf('bulunan yeni deęerler: w1:%f w2:%f w3:%f w4:%f fi:%f \n' ,W(1,1),W(1,2),W(1,3),W(1,4),fi);
```

## PYTHON

```
"""durumların tanitilmesi"
kirmizi durumlar"
sari durumlar
yesil durumlar
mavi durumlar"""
"-----"
B = [[1.0, 0.0, 0.0, 0.0],
      [0.0, 1.0, 0.0, 0.0],
      [0.0, 0.0, 1.0, 0.0],
      [0.0, 0.0, 0.0, 1.0]];
```

```
net = 0.0;
beklenen_durumlar = [1.0, 0.8, 0.4, 0]; "beklenen durumların sirayla ciktilari"
```

```
"ağırlık, esik ve öğrenme katsayılarının tanitilmesi"
"-----"
W = [0.7, 0.8, 0.3, -0.2];
fi = 0.1;
a = 0.9;
```

```
for x in range(1):

    i = 0;
    while i < 4 :
        net = W[0] * B[i][0] + W[1] * B[i][1] + W[2] * B[i][2] + W[3] * B[i][3]+fi;
        print('net : ' , net, '\n');
        if(net > 0.8):
            net = 1;
        elif(net <= 0.8 and net > 0.4):
            net = 0.8;
        elif(net <= 0.4 and net > 0):
            net = 0.4;
        elif(net < 0):
            net = 0;

        E = beklenen_durumlar[i] - net;
        print('hata : ' , E, '\n');

        if(E != 0):
```

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

```
print('yeni degerler hesaplaniyor');
fi = fi + a * E;
for j in range(4):
    W[j] = W[j] + a * E * B[i][j];
print('bulunan deg?erler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'w4 : ', W[3], 'fi : ', fi);
i = 0;
elif (E == 0):
    print('hata yok');
    i = i + 1;
```

```
print('bulunan yeni deg?erler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'w4 : ', W[3], 'fi : ', fi);
```

## SONUCLAR

```
In [30]: runfile('C:/Users/Onder/Desktop/ODEVLER/yapay_zeka_odevler/4.hafta/untitled3.py',
wdir='C:/Users/Onder/Desktop/ODEVLER/yapay_zeka_odevler/4.hafta')
net : 0.7999999999999999

hata : 0.19999999999999996

yeni degerler hesaplaniyor
bulunan degerler: w1 : 0.8799999999999999 w2 : 0.8 w3 : 0.3 w4 : -0.2 fi :
0.27999999999999997
net : 1.16

hata : 0.0

hata yok
net : 1.08

hata : -0.19999999999999996

yeni degerler hesaplaniyor
bulunan degerler: w1 : 0.8799999999999999 w2 : 0.6200000000000001 w3 : 0.3 w4 : -0.2 fi :
0.1
net : 0.9799999999999999

hata : 0.0

hata yok
net : 0.7200000000000001

hata : 0.0

hata yok
net : 0.4

hata : 0.0

hata yok
net : -0.1

hata : 0

hata yok
bulunan yeni degerler: w1 : 0.8799999999999999 w2 : 0.6200000000000001 w3 : 0.3 w4 :
-0.2 fi : 0.1
```

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

## PYTHON ARAYÜZÜ

```
1  """durumların tanitilmesi"
2  kırmızı durumlar"
3  sarı durumlar
4  yeşil durumlar
5  mavi durumlar"""
6  "-----"
7  B = [[1.0, 0.0, 0.0, 0.0],
8        [0.0, 1.0, 0.0, 0.0],
9        [0.0, 0.0, 1.0, 0.0],
10       [0.0, 0.0, 0.0, 1.0]];
11
12
13  net = 0.0;
14  beklenen_durumlar = [1.0, 0.8, 0.4, 0]; "beklenen durumların sırayla çıktıkları"
15
16
17  "ağırlık, esik ve öğrenme katsayılarının tanitilmesi"
18  "-----"
19  W = [0.7, 0.8, 0.3, -0.2];
20  fi = 0.1;
21  a = 0.9;
22
23  for x in range(1):
24
25      i = 0;
26      while i < 4 :
27          net = W[0] * B[i][0] + W[1] * B[i][1] + W[2] * B[i][2] + W[3] * B[i][3]+fi;
28          print('net : ', net, '\n');
29          if(net > 0.8):
30              net = 1;
31          elif(net <= 0.8 and net > 0.4):
32              net = 0.8;
33          elif(net <= 0.4 and net > 0):
34              net = 0.4;
35          elif(net < 0):
36              net = 0;
37
38          E = beklenen_durumlar[i] - net;
39          print('hata : ', E, '\n');
40
41          if(E != 0):
42              print('yeni değerler hesaplanıyor');
43              fi = fi + a * E;
44              for j in range(4):
45                  W[j] = W[j] + a * E * B[i][j];
46              print('bulunan değerler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'w4 : ', W[3], 'fi : ', fi);
47              i = 0;
48          elif (E == 0):
49              print('hata yok');
50              i = i +1;
51
52
53  print('bulunan yeni değerler: w1 : ', W[0], 'w2 : ', W[1], 'w3 : ', W[2], 'w4 : ', W[3], 'fi : ', fi);
54
55
```



- 3) <https://paperswithcode.com/sota> bağlantısında istediğiniz iki farklı alanda 2021 yılında yayımlanmış makalelerden; yapay zeka/makine öğrenmesi veya derin öğrenme ile ilgili olanlardan birer tanesini okuyup, en fazla 5'er sayfada Türkçe olarak özetleyiniz.

### EVOLUTIONARY SELF-REPLICATION AS A MECHANISM FOR PRODUCING ARTIFICIAL INTELLIGENCE

Yaşam öyküsü olarak bilinen mucizevi gerçek, ilkel organizmaların kendini kopyalamasıyla ortaya çıktı. Yaşama döngüsü evrim tarafından tetiklendi. Bu canlı organizmaların, bugün bildiğimiz gibi yaşama dönüşmesi, evrim tarafından körüklendi. Evrim, mikroskobik bakteri ve virüslerden bitkiler, kuşlar ve bizler gibi daha büyük çok hücreli organizmalara kadar her ölçekte organizmaya yol açan yaratıcı bir süreçtir; bu süreç, büyük ilgi çeken iki yeniliğe yol açtı: yaşam ve zeka.

Onlarca yıldır, makinelerimizin yakından takip etmesi için kuralları, mantığı, mimarileri ve hedefleri elde etmek için Yapay Zeka (AI) araştırmasının birincil stratejisi olmuştur ve makinenin yaratıcı çözümleri keşfetmesi için vazgeçilmez olmuştur. AI'daki büyük ilerleme, manuel tasarımın azaltılması ve makinenin kendi çözümlerini keşfetmesine izin veren bir model izler.

Mimariler ve öğrenme algoritmalarının tasarımını önemli ölçüde manuel çanalar harcanırken, bu tür nüanslar bir algoritmanın kendisi tarafından keşfedilebilir. AI üreten algoritmalar paradigması, AI üretmenin daha etkili yolunun; AI' nın kendini nasıl tasarlayacağını ve geliştireceğini öğrendiği bir süreç olması gerektiğini savunulur. Gerçekte, biyolojik organizmaların gelişme sürecine benzemektedir ve evrim en gelişmiş tasarım sürecidir.

Evrimin, hayatta kalma baskısı altında organizmaların üreyen bir özellik olarak ortaya çıktığı fikri ihlal etmek kolaydır çünkü organizmalara uygulanan bir algoritma varmış gibi görünebilir. Buna ek olarak üreme organizmalarının kendileri, doğal seçilimin baskısı altında evrim geçiriyorlar. Anbilim meselesinin ötesinde, evrim i kavramsallaştırmak

Evrimin, hayatta kalma baskısı altında organizmaları üreyen bir özellik olarak ortaya çıktığı fikrini ihmal etmek kolaydır; olduğu gibi, organizmalara uygulanan kapsayıcı bir algoritma varmış gibi görünebilir, oysa daha çok üreme organizmalarının kendileri, doğal seçilimin baskısı altında evrim geçiriyorlar. Bu sadece bir anlambilim meselesi gibi görünse de, evrimi bu şekilde kavramsallaştırmak, evrimin gücünden yararlanmaya çalışan algoritmaların uygulanma şeklini değiştirir. Evrim sadece organizmalara uygulanmaz, organizmaların kendileri doğal olarak evrimleşecek şekilde tasarlanmıştır. Bu çalışma, biyolojik kendini kopyalamanın özelliklerini soyutlayarak ve doğal seçim açısından modern öğrenme problemlerini formüle ederek yaşam ve ölüm sorusunu yanıtlıyor. Bu şekilde, atari oyunlarının yanı sıra robotik öğrenme problemlerinde ilginç davranışların yalnızca yaşam, ölüm ve replikasyona dayalı olarak ortaya çıktığı gözlemlenmiştir.

Evrimsel hesaplama ve evrimsel kendini kopyalama arasındaki ayrımın anlaşılması önemlidir, çünkü bunlar temelde farklı kavramlardır. Evrimsel kopyalamada oluşan organizmalar hayatta kalmak için koşulları sağlamalı, sağlayamazsa yok olmalı biçimindedir. Evrimsel hesaplama ise çeşitli faktörlerin sonucunda farklı popülasyonların oluşması neticesidir. Evrimsel kendini kopyalama Yapay Zeka alanında zengin bir tarihe sahiptir. Oyun tarihinde Darwin ve Core War gibi oyunlarda bulunan digital organizma gibi davranan, kendilerinin ve başkalarının talimatlarını mutasyona uğratarak, saldırgan saldırılarda bulunurlardı. Oyunu kazanılması için en başarılı strateji ise şuydu, diğer programların hesaplama kaynaklarından mahrum bırakılmasıdır.

Evrimsel kendini kopyalamada; başlangıçta, boş 1 boyutlu hücrelerin sonlu bir modeli somutlaştırılır. Buradan, bu temel organizmanın belirli niteliklerinin çevreye ve kendini kopyalayanın tanımına bağlı olduğu bir "temel organizma" ortaya çıkar; en basit durumda, başlangıçta rastgele davranış üreten küçük bir sinir ağı ortaya çıkar. En üst düzeyde basitliği korumak için, gerçek çoğaltma süreci soyutlanır ve rastgele gürültü yoluyla mutasyonlar eklenir. Organizmanın içinde bulunduğu çevre, hayatta kalmayı, yaşamı ve ölümü tanımlar ve varlığını sürdürmek için organizmanın hayatta kalmaya adapte olması gerekir. Organizma ölümle karşılaşır, varlığı sona erer ve 1 boyutlu modelden çıkarılır. 1-boyutlu modelde çoğalma şu şekilde çalışır: a) her ortamda zaman geçtikçe organizmaların bir çoğaltma olasılığı vardır, b) bir organizma başarılı bir şekilde çoğalırsa, iki komşu hücreden birinde rastgele kendisinin mutasyona uğramış bir kopyası üretilir, c) seçilen hücre doluyorsa, çoğaltma gerçekleşmez. Böylece, hücreler giderek daha fazla işgal edildiğinde, 1 boyutlu modeldeki çoğaltma, uzun ömür için bir rekabet haline gelir. Sonuç olarak organizma ya yaşar ya da ölür. Bir organizma ne kadar uzun süre hayatta kalabilirse, çoğalma olasılığı o kadar artar; Daha uzun ömürlü organizmalardan kopyalanan kopyaların, atalarının organizmalarına benzer, ancak aynı olmayan özellikler içeren, daha kısa ömürlü organizmalardan türetilenlere göre hayatta kalma olasılığının daha yüksek olması beklenir. CartPole-survival: Yaşam ve ölümün doğal bir tanımına sahip olan ele alınan ilk ortam CartPole-survival'dir. Bu ortamda, bir arabanın ortasına, organizmanın arabaya sağ veya sol taraftan kuvvet uygulayabileceği bir direk yerleştirilir. Direğin tepesi, arabadan belirli bir eşiğin ötesine düşerse, organizma ölür. Bir ölüm durumuna ulaşılan kadar organizma varlığını sürdürür.

Robotik öğrenmeye göz atacak olursak; bilinen modern robotik-öğrenme görevlerini çözmek için, öncelikle, görevler doğal seçilimin ortaya çıkışına uygun bir şekilde yeniden tanımlanır: yaşam ve ölüm. Üç yaklaşım daha ele alınmıştır hayatta kalma, yiyecek arama ve hayatta kalma-toplayıcılık. Sunulan ortamlar, deneyden önce seçilmiştir. Birçok ortamda, evrimsel kendini kopyalama yoluyla üretilen organizmalar, çözümlerinde en üst düzeyde yaratıcılığı sergilerler ve genellikle çevredeki nüanslardan yararlanarak, miras alınan davranışlar ile hayatta kalma olasılıklarını artırırlar. Küçük sinir ağları kullanılır, böylece organizmalar, aşırı parametrelili birçok modelin yapma eğiliminde olduğu gibi, bilgileri basitçe ezberlememeleri için baskı yapar. Her görev, gizli katmanda 32 nöronlu tek bir gizli katman tam bağlantılı tekrarlayan sinir ağı kullanır.

Biyolojik organizmalar sıfırdan başlayıp kendi problemlerini icat ettiğinden buna benzeyen robotik öğrenme uygulamaları da sabit bir morfoloji ile gelişim gösterir, robotik öğrenme evrimsel kendini yeniden üretmenin en sezgisel uygulaması olmasa da robotik öğrenme bu konuya iyi bir örnek teşkil eder.

CartPole-hayatta kalma ortamında görüldüğü gibi, çevre tarafından tanımlanan bir tür durumun korunmasını gerektirir. Belirli bir koşul, bir ölüm durumu karşılandığında, organizma sona erer ve daha fazla üremeyi önleyerek varlığı sona erer. Hayatta kalma deneylerinde görüldüğü gibi, hayatta kalmaya olan bu tek odak, her zaman olmasa da, çevre tarafından oluşan kusurları manipüle ederek ölüm durumlarından kaçabilen akıllı organizmalara yol açabilir.

In Boxing-survival-fight: Boks-hayatta kalma ortamı, mevcut bir Atari oyunundan Boks'tan uyarlanmıştır. Bu ortamda, iki oyuncu, organizmamız ve önceden tanımlanmış bir AI, hareket etmeyi ve yumruk ile puan lamayı hedefler 100 puana ulaşan kazanır ardından ek olarak 2 dk. süre konularak 100 puan veya sürenin bitmesi durumunda en yüksek puanı alanın kazanacağı bir durum belirlenmiştir. Gözlemler sonucunda yapay zeka hayatta kalmak için belli bir puandan sonrada gelen yumruklardan kaçmayı ve süreyi eritmeye de çalıştığı gözlemlenmiştir.

Pong-survival: In Boxing-survival-fight gibi, Pong ortamında da iki raketin karşı taraflara yerleştirildiği ve yatay olarak hareket eden bir topu hareket ettirmek zorunda olduğu mevcut Atari

Pong oyununa uyarlanır ve top rakibinin raketini geçtiğinde puan kazanılır. Organizma popülasyonu tarafından geliştirilen strateji, sonsuz bir ömre ulaşmak için çevre tasarımıdaki bir zayıflıktan yararlanır. Başlangıç stratejisi, topu nispeten öngörülebilir bir şekilde saptırmaktan, ara sıra puan kazanmaktan, ancak rakip raket üzerinde çok fazla stres oluşturmamaktan oluşur. Bununla birlikte, sonunda organizma, karşıt kanat deterministik olarak programlandığından, organizma kanadının hareket etmeden yerleştirilebileceği bir konum olduğunu ve karşıt kanatçının her zaman bu konuma geri döneceğini keşfeder. Bu konumu keşfederek ve onu nasıl kullanacağını öğrenerek, kendi kendini çoğaltıcının davranışı, nihayetinde hiçbir puanın alınmadığı sonsuz bir oyuna ve dolayısıyla sonsuz bir ömre yol açar

Yemek arama: Toplayıcılığa dayalı ortamlar, organizmanın hayatta kalabilmesi için düzenli aralıklarla bazı gereksinimleri karşılaması gereken bir metabolizma kavramını ortaya koymaktadır. Bu gereksinim, çok çeşitli olasılıklarla tanımlanabilir, örneğin: kaynak elde etme, bir oyunda puan kazanma veya bir yere doğru ilerleme. Bazı enerji gereksinimlerinin rutin olarak yerine getirilmesi gerekli hale geldiğinden, yiyecek arama görevleri çevre sömürsüne daha az eğilimli olma eğilimindedir. Fiziksel robotların büyük olasılıkla enerji, kaynak veya bakım elde etmeyi gerektireceği düşünüldüğünde, yiyecek aramanın getirilmesi, gerçek dünyada arzu edilen davranışı mümkün kılar. Reacher-forager: Reacher-forager ortamı, bir düzlem üzerinde duran iki bağlantılı bir kolun ikinci bağlantısının ucuna bir hedef pozisyonuna ulaşma görevi ile kontrol edildiği orijinal Reacher pekiştirmeli öğrenme kıyaslamasını uyarlar. Bu görev, doğal olarak, sadece enerji elde etmek için hedef konumuna ulaşılmasını gerektirerek yiyecek arama alanına genişletilebilir. Enerji bittiğinde, organizma ölür. Bir hedef pozisyonuna ulaşıldığında, organizmanın enerji deposuna enerji depolanır ve bu enerji zamanla yavaş yavaş tükenir ve hedef pozisyonu sıfırlanır. Bu görevin çözümü oldukça zekicedir ve organizmadan beklenenin ötesine geçer. İkinci halkayı dikkatli bir şekilde doğru yere yerleştirmek yerine, kol, enerji tükenmeden tüm olası hedef konumlara ulaşılacak şekilde ikinci halkayı ustaca sıkıştırıp genişletirken kolu hızla bir daire içinde döndürmek için daha başarılı bir strateji geliştirir. Bu çözümü kullanarak, Reacher-forager organizma popülasyonu süresiz olarak var olma yeteneğine sahip olur. Reacher'ın hareketine bir enerji cezası ekleyerek, bu davranış, beklenen hedef pozisyonlarına doğru kesin ve yavaş hareket stratejisine geri döner.

BattleZone-survival-forager: BattleZone, bir tank olan organizmanın diğer düşman tanklarıyla dolu bir ortama yerleştirildiği bir Atari oyunudur. Tank 5 canla başlar ve oyun puanı, tüm canlar kaybolmadan önce yok edilen düşman araçlarının sayısına bağlıdır. Bunu bir hayatta kalma-toplayıcı ortamına dönüştürürken, hayatta kalma bileşeni, tüm 5 canın kaybolmasından sonra ölümü içerir ve toplayıcı bileşen, bir düşman aracını yok ettikten sonra açlığı yeniler. Yiyecek arama kaldırıldığında, organizma stratejisi varsayılan olarak tüm tanklardan kaçınmaya ve amaçsızca etrafta dolaşmaya başlar, çünkü ne kadar çok araç yok edilirse oyun zorluğu o kadar yüksek olur. Gerçekten de bu organizmaların son teknoloji optimizasyon ile rekabetçi bir şekilde performans gösterme niyeti olmasa da ve bu organizmalar tanımlanmış amaç fonksiyonu için optimize edilmemiş olsa da, BattleZonesurvival-forager organizması tarafından elde edilen ödül, kaydedilen en yüksek puanların çoğuyla karşılaştırılabilirdi. Her düşman aracının imha edilmesinin 1000'lik bir ödül vermesiyle, ortaya çıkan organizma, beş canın tamamında yok edilen 102 araç anlamına gelen 102000'lik bir medyan ödül elde etti. Tank, ateşe karşılık verirken ve kaybedilen can başına ortalama 20 düşman avı gerçekleştirirken düşman ateşinden ustaca kaçtığından, bu tür bir performans elde etme stratejisini gözlemlemek nispeten basittir. Bu performansı elde etmek için kullanılan ağ yapısı, hiçbir ek değişiklik yapılmayan küçük, sabit ağırlıklı bir yapıydı. Performansta önemli iyileştirmelerin ek yetenekler yoluyla gerçekleştirilebilmesi muhtemeldir. Belki de bu, yalnızca zekanın kendisini değil, aynı zamanda insan tasarımının sınırlarıyla sınırlı olmayan zekayı da üretme mekanizması olarak nesnel rehberlik olmaksızın doğal seçim ve rastgele mutasyon potansiyeline bir bakıştır.

Öğrenci No: 171110001

Adı Soyadı: CAN AHMET ACAR

Her deneyin süresi, büyük ölçüde simülatörün hızına ve görevin karmaşıklığına bağlıydı. Örneğin, en hızlı deneyde, CartPolesurvival alanındaki basit evrimsel eşleyicide yaşam boyu farklılığını elde etmek için yalnızca yaklaşık bir dakikalık hesaplama gerektirdi. Öte yandan, en yavaş deneyde, BattleZone-hayatta kalma-toplayıcı ortamında elde edilen sonuçların yaklaşık bir günlük hesaplaması gerekiyordu. Bu deneylerin herhangi bir hız ivmesi olmadan standart bir tek çekirdekli işlemci üzerinde yürütüldüğü ve gradyan tabanlı deneylerin ve evrimsel hesaplama deneylerinin tipik hesaplama gereksinimleri göz önüne alındığında, evrimsel kendi kendini çoğaltıcılar, nispeten hızlı ilginç davranışlar elde etme eğilimindedir. Doğal sistemlerde anlamlı evrim çok yavaş bir hızda meydana gelme eğiliminde olduğundan, bu şaşırtıcı bir özelliktir.

Evrimsel kendini kopyalama, kendisini hem zekayı hem de yaşamı geliştirmeye yönelik güçlü bir süreç olarak sunar. Bu çalışmada, yaratıcı, anlamlı ve zeki davranışların basit yaşam, ölüm ve kopyalama süreciyle ortaya çıkabileceği gösterilmiştir. Bu yeteneğin, modern robotik öğrenme ortamlarında, doğal seçim yoluyla hayatta kalma açısından problemleri yeniden çerçeveleyerek hem pratik hem de ilginç davranışlar elde etmek için kullanılabileceği gösterilmiştir. Bu ortamlarda ortaya çıkan organizmalar, zaman zaman, evrimleşmiş organizmaların doğal olarak belirsiz ölçeklerde çalışmasını sağlayan ölümsüzlük çözümleri üretti. Evrimsel kendi kendini kopyalayan sistemlerin ilginç bir özelliği, gelişimin organizmaların hayatta kalma ve üreme yeteneklerini etkileyebilmesi ve belki de gelişim ile doğal seçim arasındaki etkileşimin keşfedilebilmesidir. Gelişim, yaşam boyu öğrenme ve evrimsel kendi kendini kopyalama, meta-seviye optimizasyonu olarak temsil edilebilir. Sorulması gereken ilginç bir soru, evrimsel kendini kopyalamanın geleneksel anlamda bir öğrenme algoritması olup olmadığıdır. Bazı yönlerden diğer öğrenme algoritmalarına benzer ve diğerlerinde büyük ölçüde farklılık gösterir. Görünüşe göre, evrimsel kendini kopyalama durumunda zekanın ortaya çıkışı, tasarımdan çok bir fenomen meselesidir; yapay zeka ve yaşamın gelişimi için büyük bir entrika olgusu.

<https://paperswithcode.com/paper/evolutionary-self-replication-as-a-mechanism>