

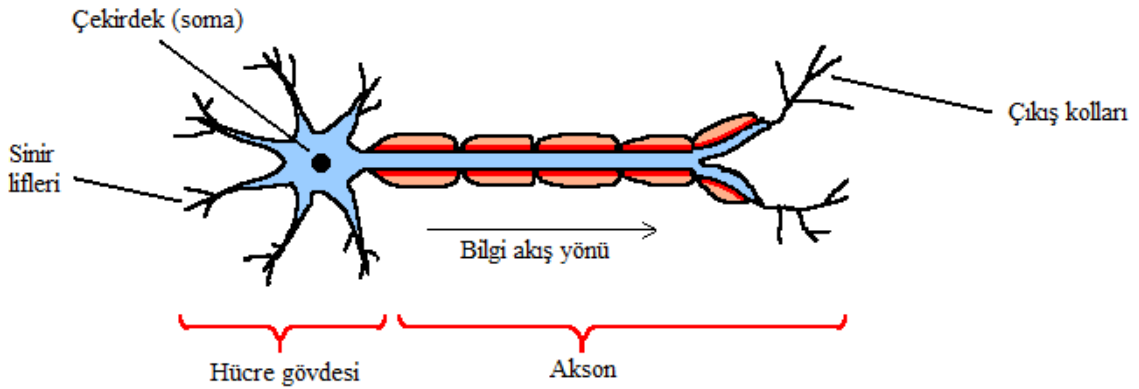
Lineer Ayrılabilen Paternlerin Yapay Sinir Ağı ile Sınıflandırılması

1. Biyolojik Sinirin Yapısı

Bilgi işleme açısından bir sinir üç kısımdan oluşur; Şekil 1.

- Eklemler:** Geçmiş deneyimlerin (bilgi tabanı) saklandığı alandır. Diğer sinirlerden bilgileri alır ve onları ağırlıklandırarak somaya iletir.
- Soma:** Sinirin gövdesidir. Eklemlerden gelen ağırlıklandırılmış bilgileri alır, onları toplar, eşiklendirir ve genelde doğrusal olmayan bir aktivasyon fonksiyonundan geçirir. Eklemlerden gelen ağırlıklandırılmış girişlerin toplamı belirli bir eşik değerini aşarsa soma bir çıkış işareti üretir.
- Akson:** Somanın ürettiği işareti diğer sinirlere ileten bir fiber yoldur.

Sinirler, çözülmüş kimyasal iyonlar içeren akışkanlarla doldurulmuş ve kuşatılmıştır. Bu kimyasal iyonlar sodyum (Na^+), kalsiyum (Ca^{++}), potasyum (K^+) ve klördür (Cl^-). K^+ iyonları sinir hücresinin içinde yoğun olarak bulunurken, Na^+ iyonları ise hücre membranının dışında yoğunlaşmıştır. Na^+ ve K^+ iyonları sinir darbesi adı verilen aktif sinir cevabının üretilmesinden sorumludur.

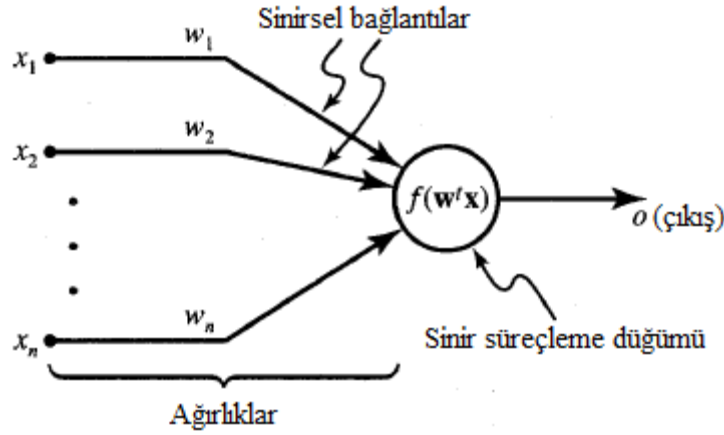


Şekil 1. Biyolojik sinirin yapısı

İnsan beyni sinir adı verilen yaklaşık 10^{11} hesaplama elemanından oluşur. Bu sinirlerin her biri 10^4 tane sinir eklemine sahiptir. Merkezi sinir sistemi reseptörlerden aldığı bilgileri kendi içinde işler ve efektörleri kumanda ederek çeşitli aksiyonlar biçiminde cevaplar üretir.

2. Yapay Sinir Modeli

Her sinir modeli çok sayıda sinirsel giriş bağlantıları ve tek bir çıkışı olan bir süreçleme elemanından ibarettir; Şekil 2.



Şekil 2. Girişler, ağırlıklandırıcılar, ve süreçleyiciden oluşan bir sinirin genel sembolü.

Sinirin çıkış işareti aşağıdaki bağıntı ile verilir.

$$o = f(\mathbf{w}'\mathbf{x}) \quad (1)$$

veya

$$o = f\left(\sum_{i=1}^n w_i x_i\right) \quad (2)$$

Burada \mathbf{w} ağırlık vektörüdür ve

$$\mathbf{w} \triangleq [w_1 \ w_2 \ \dots \ w_n]^t$$

ile tanımlanır ve \mathbf{x} ise giriş vektörü olup aşağıdaki gibi verilir.

$$\mathbf{x} \triangleq [x_1 \ x_2 \ \dots \ x_n]^t \quad (3)$$

Ağırlıklarla giriş vektörünün skaler çarpımı net olarak adlandırılır.

$$\text{net} \triangleq \mathbf{w}'\mathbf{x} \quad (4)$$

$f(\mathbf{w}'\mathbf{x})$ fonksiyonu aktifleme fonksiyonu olarak bilinir ve net argümenti biyolojik sinirin membran potansiyeline benzer.

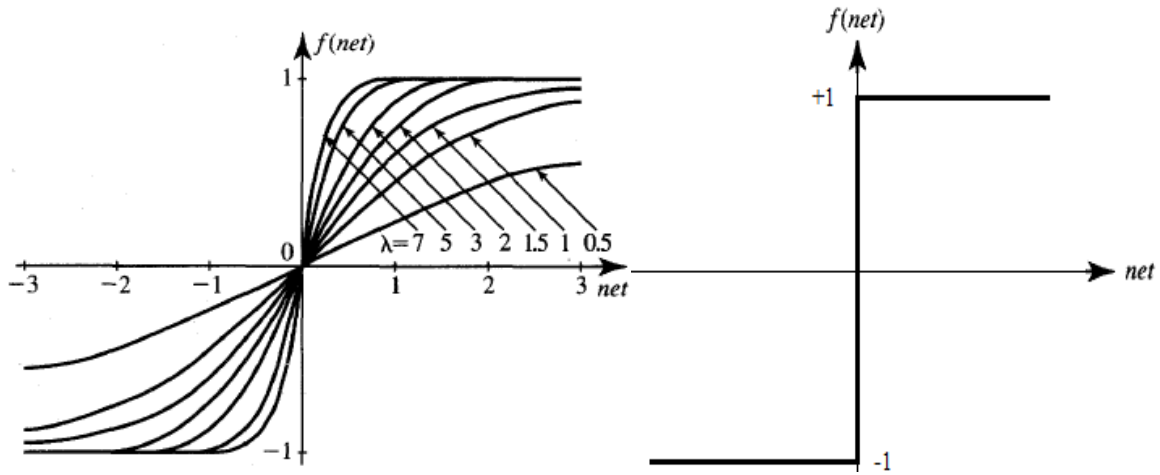
Sinir, süreçleme düğümü olarak, kendi ağırlıklandırılmış girişlerinin toplanması işlemi veya net'i elde etmek için skaler çarpma işlemi gerçekleştirir. Ardından, kendi aktivasyon fonksiyonu sayesinde $f(\text{net})$ lineer olmayan işlemi yapar. Kullanılan aktivasyon fonksiyonları

$$f(\text{net}) \triangleq \frac{2}{1 + \exp(-\lambda \text{net})} - 1 \quad (5)$$

$$f(\text{net}) \triangleq \text{sgn}(\text{net}) = \begin{cases} +1, & \text{net} > 0 \\ -1, & \text{net} < 0 \end{cases} \quad (6)$$

$\lambda > 0$ katsayısı $\text{net} = 0$ yakınında $f(\text{net})$ sürekli aktivasyon fonksiyonunun derinliğini belirleyen sinir kazancı ile orantılıdır. $\lambda \rightarrow \infty$ için sürekli aktivasyonun limiti (6)

bağıntısında tanımlanan $\text{sgn}(\text{net})$ fonksiyonu haline gelir. (5) ve (6) bağıntıları sürekli ve ayrık bipolar aktivasyon fonksiyonları gösterir; şekil 3.



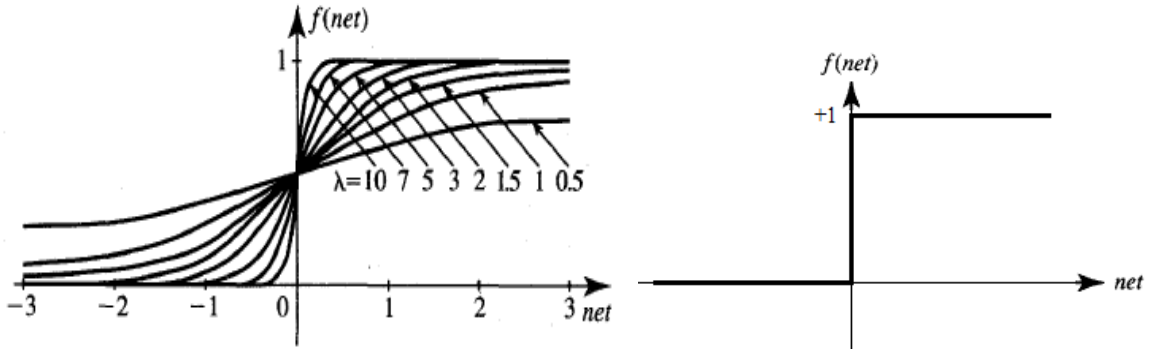
Şekil 3. Sürekli ve ayrık bipolar aktivasyon fonksiyonları

Aktivasyon fonksiyonları tek kutuplu olabilir, bu durumda sinirlerin çıkışları sadece pozitif değerler alır.

$$f(\text{net}) \triangleq \frac{1}{1 + \exp(-\lambda \text{net})} \quad (7)$$

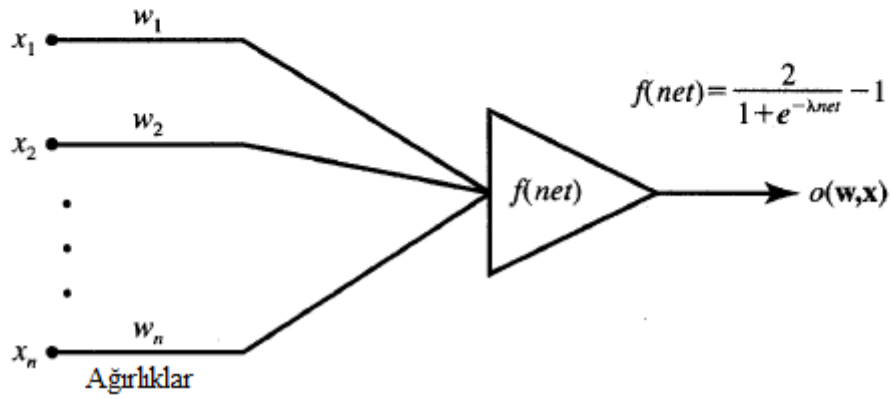
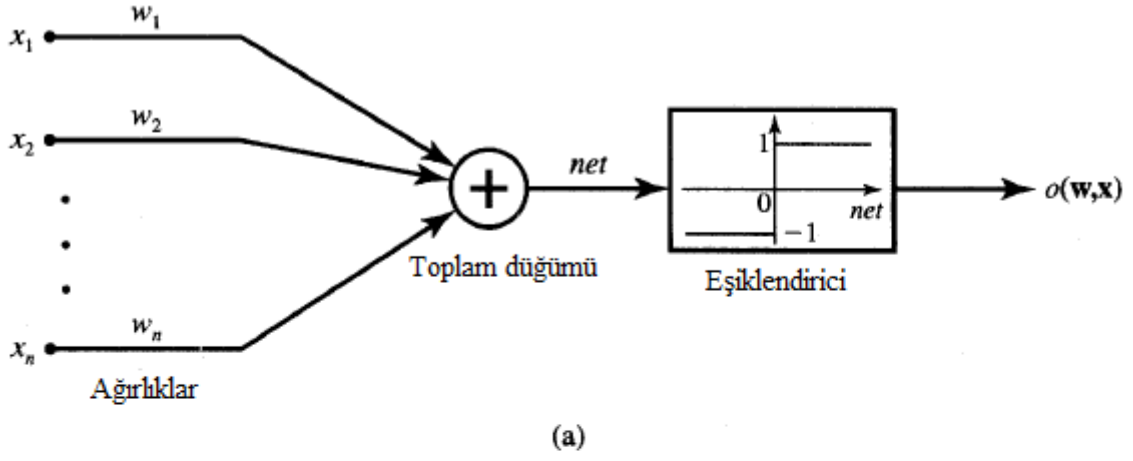
$$f(\text{net}) \triangleq \begin{cases} 1, & \text{net} > 0 \\ 0, & \text{net} < 0 \end{cases} \quad (8)$$

(7) ve (8) bağıntılarına karşı düşen aktivasyon fonksiyonları şekil 4'te gösterilmiştir



Şekil 4. Tek kutuplu sürekli ve ayrık aktivasyon fonksiyonları

Ayrık ve sürekli aktivasyon fonksiyonuna sahip sinirlerin sembolik gösterimi şekil 5'te gösterilmiştir.



Şekil 5. Ayrık ve sürekli aktivasyon fonksiyonuna sahip sinirlerin sembolik gösterimi

3. Yapay Sinir Ağ Modelleri

Yapay sinir ağı, sinirlerin kendi aralarında bağlantısı olarak tanımlanabilir. Sinir çıkışları, ağırlıklar üzerinden, kendileri de dâhil olmak üzere tüm diğer sinirlere hem gecikmesiz hem de gecikmeli bağlantılara izin verecek şekilde bağlanarak bir ağ oluşturulabilir. Ağ mimarileri genelde ileri beslemeli ağlar ve geri beslemeli ağlar olmak üzere iki sınıfa ayrılabilir.

3.1. İleri Beslemeli Ağ

n girişli m sinirin ilkel ileri beslemeli mimarisi şekil 6'da gösterilmiştir. Çıkış ve giriş vektörleri sırasıyla aşağıdaki gibidir.

$$\mathbf{o} = [o_1 \ o_2 \ \dots \ o_m]^t \quad (9)$$

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^t$$

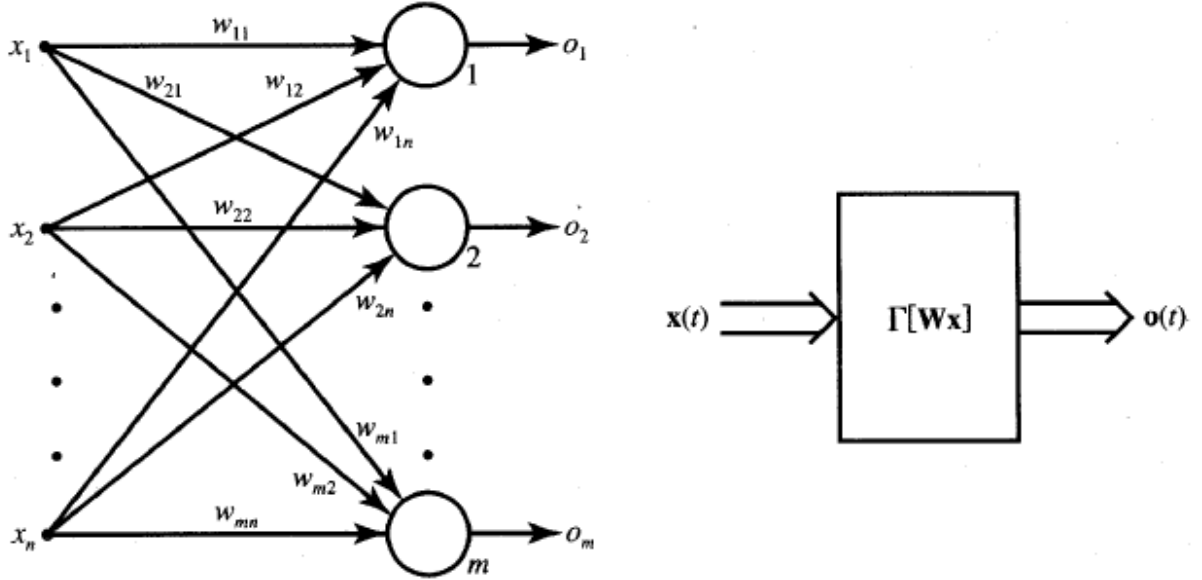
w_{ij} ağırlığı i . siniri j . girişe bağlar, yani çift indislerden birincisi hedefi ikincisi de kaynak düğümünü işaret eder. O zaman i . sinirin aktivasyon değeri aşağıdaki gibi yazılabilir

$$net_i = \sum_{j=1}^n w_{ij}x_j, \quad \text{for } i = 1, 2, \dots, m \quad (10)$$

Ağdaki m sinirin her biri tarafından gerçekleştirilen transformasyon aşağıdaki gibi ifade edilen güçlü bir doğrusal olmayan iz düşümdür

$$o_i = f(\mathbf{w}_i^t \mathbf{x}), \quad \text{for } i = 1, 2, \dots, m \quad (11)$$

Tek katmanlı ileri beslemeli ağın ara bağlantı şeması Şekil 6'da verilmiştir.



Şekil 6. İleri beslemeli bir ağ modeli ve blok diyagramı

3.2. Geri Beslemeli Ağ

Geri beslemeli ağ, sinirlerin çıkışlarını onların girişlerine bağlayarak Şekil 6'da gösterilen ağdan elde edilebilir. İleri beslemeli ağlar için takdim edilen notasyon kullanılarak $\mathbf{o}(t)$ 'nin $\mathbf{o}(t + \Delta)$ 'ya izdüşürülmesi aşağıdaki gibi yazılabilir.

$$\mathbf{o}(t + \Delta) = \Gamma[\mathbf{W}\mathbf{o}(t)] \quad (12)$$

Bu bağıntı bir dizi yuvalı çözümden ibarettir. Bu ağ tekrarlamalı ağ olarak da bilinir. Bu deneyin amacına uygun olmadığından bu ağ üzerinde daha fazla durulmayacaktır.

4. Sinir Ağı Öğrenme Kuralları

Yapay sinir ağı, girişleri ile çıkışları arasındaki ilişkisi bilinen her sistemin yerini alabilir. Eğer bir sistemin girişi ile çıkışı arasındaki ilişkisi matematiksel olarak modellenilebiliyorsa, o zaman yapay sinir ağından ziyade matematiksel izdüşüm bağıntısının hesaplanması daha uygun olabilir. Yapay sinir ağı daha ziyade matematiksel olarak ifade edilemeyen ama girişlerine karşılık çıkışları bilinen sistemlere eğitim yoluyla istenilen giriş-çıkış izdüşüm yeteneğinin kazandırılması için kullanılır. Birçok farklı eğitim algoritması vardır. Burada karakter sınıflama işlemi yapılacağı için sınıflamaya yönelik bir ağın seçimi ve eğitimi anlatılacaktır.

4.1. Perceptron Öğrenme Kuralı

Bu deneyde iki karakterin sınıflamasını yapan bir ağın tasarımı ve eğitimi anlatılacaktır. Bipolar ayrık aktivasyon fonksiyonuna sahip bir sinir +1 ve -1 olmak üzere iki çıkış değeri verdiği için bir sınırlı ve tek katmanlı bir ağ iki farklı karakterin sınıflandırılması için uygun bir ağdır.

Tek katmanlı böyle bir ağın danışmanlı eğitimi için en uygun eğitim kuralı Perceptron Öğrenme kuralıdır, Şekil 7. Bu kural için r öğrenme işareti, arzulanan ile gerçek sinir cevabı arasındaki farktır. Bu yüzden öğrenme danışmanlıdır ve öğrenme işareti

$$r \triangleq d_i - o_i \quad (13)$$

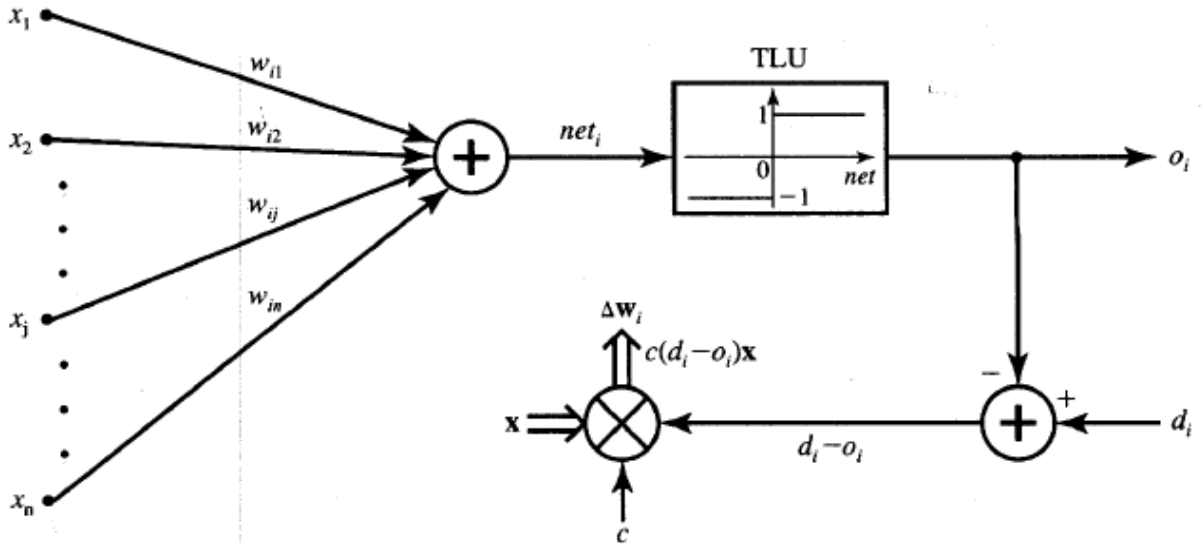
değerine eşittir. Burada $o_i = \text{sgn}(\mathbf{w}_i^t \mathbf{x})$ ve d_i ise arzulanan cevaptır. Şekil 7'den $\Delta \mathbf{w}_i$ için

$$\begin{aligned} \Delta \mathbf{w}_i &= c [d_i - \text{sgn}(\mathbf{w}_i^t \mathbf{x})] \mathbf{x} \\ \Delta w_{ij} &= c [d_i - \text{sgn}(\mathbf{w}_i^t \mathbf{x})] x_j, \quad \text{for } j = 1, 2, \dots, n \end{aligned} \quad (14)$$

yazılabilir. Bu kuralda, yalnız o_i cevabı doğru değilse ağırlık ayarlaması yapılır. Arzulanan cevap ya +1 veya -1 olduğundan ağırlık ayarı

$$\Delta \mathbf{w}_i = \pm 2c \mathbf{x} \quad (15)$$

bağıntısına indirgenir. Arzulanan ve gerçek cevaplar aynı ise ağırlık ayarı yapılmaz. Bu yöntemde ağırlıklar herhangi bir değere ilklendirilebilir.



Şekil 7. Perceptron öğrenme kuralı

5. Tek Ayrık Perceptron Eğitim Algoritmasının Özeti(SDPTA)

P eğitim çifti veriliyor.

$$\{ \mathbf{x}_1, d_1, \mathbf{x}_2, d_2, \dots, \mathbf{x}_P, d_P \} \text{ burada}$$

\mathbf{x}_i vektörü ($n \times 1$), d_i ise (1×1) $i=1, 2, \dots, P$ boyutlardır. İlaveli giriş vektörlerinin kullanıldığına dikkat ediniz.

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}, \quad i=1, 2, \dots, P \text{ için}$$

Aşağıda k, eğitim adımını ve p ise eğitim dönemi içindeki adım sayıcısını gösterir.

- Adım 1: $c > 0$ seçilir.
- Adım 2: Ağırlıklar, \mathbf{w} vektöründe küçük rasgele değer ile başlatılır. \mathbf{w} 'nin boyutu $(n+1) \times 1$ 'dir
- $$k \leftarrow 1, p \leftarrow 1, E \leftarrow 0$$
- Adım 3 : Eğitim adımı buradan başlar. Giriş sunulur ve çıkış hesaplanır.
- $$\mathbf{y} \leftarrow \mathbf{y}_p, d \leftarrow d_p$$
- $$o \leftarrow \text{sgn}(\mathbf{w}^t \mathbf{y})$$
- Adım 4 : Ağırlıklar güncelleştirilir.
- $$\mathbf{w} \leftarrow \mathbf{w} + \frac{1}{2} c (d-o) \mathbf{y}$$
- Adım 5 : Dönem (cycle) hatası hesaplanır.
- $$E \leftarrow \frac{1}{2} (d-o)^2 + E$$
- Adım 6: Eğer $p < P$ ise o zaman $p \leftarrow p+1, k \leftarrow k+1$ ve adım 3'e git, aksi halde adım 7'ye git.
- Adım 7: Eğitim dönemi bitmiştir. $E=0$ için, eğitim oturumunu sonlandır. Ağırlıkları ve k 'yı ver. Eğer $E > 0$ ise, o zaman $E \leftarrow 0, p \leftarrow 1$, ve adım 3'e giderek yeni eğitim dönemine gir.

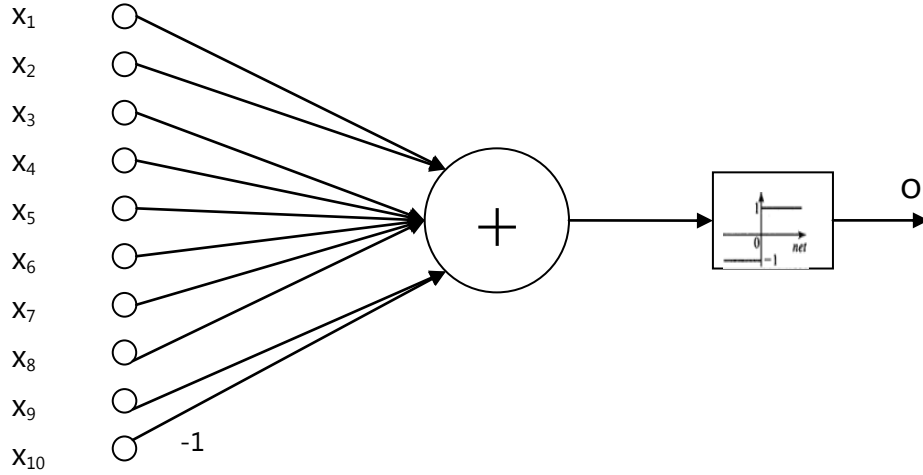
6. Tek-Katmanlı Bipolar Ayrık Perceptronun Patern Sınıflayıcı Olarak Kullanılması

Şekil 8'de gösterilen L ve I karakterleri için bir sınıflayıcının tasarlanması ve eğitilmesi istenmektedir. Siyah pikseller 1 ve beyaz pikseller 0 (sıfır) ile temsil edilmektedir. L karakteri için $o=1$ ve I karakteri için de $o=-1$ çıkışı üretecek eşik ile birlikte 10 girişe sahip ayrık bipolar perceptrondan oluşan bu sınıflayıcının ağırlıklarını eğitimle belirleyelim.

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

Şekil 8. L ve I karakterleri (3x3 boyutunda)



$$y_L = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ -1]^t$$

$$y_I = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ -1]^t$$

$$w^1 = w + \frac{c}{2} (d - o) y \quad w^1 = w \pm y, \quad o = \text{sgn}(w^t y) \quad w = 0$$

$$w^1 = w_{\text{initial}} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1] \text{ olsun.}$$

Adım 1: y_L girilir, $d_L = +1$ varsayılıyor

$$o = \text{sgn}(w^{1t} y_L) = +1 \text{ olduğundan } w^2 = w^1 \text{ olur.}$$

Adım 2: y_I girilir ve istenen çıkış $d_I = -1$ dir.

$$o = \text{sgn}(w^{2t} y_I) = \text{sgn}(1) = +1 \text{ olduğundan } w^3 = w^2 - y_I = [0 \ -1 \ 0 \ 0 \ -1 \ 0 \ 0 \ -1 \ 0 \ 0]$$

Adım 3: y_L girilir.

$$o = \text{sgn}(w^{3t} y_L) = \text{sgn}(-1) = -1 \quad w^4 = w^3 + y_L = [1 \ -1 \ 0 \ 1 \ -1 \ 0 \ 1 \ 0 \ 1 \ -1]$$

Adım 4: y_I girilir.

$$o = \text{sgn}(w^{4t} y_I) = \text{sgn}(-1) = -1 \quad w^5 = w^4 \text{ olur.}$$

Eğitim bitmiştir. L karakteri $o = +1$ çıkışına ve I karakteri de $o = -1$ çıkışına izdüşürülmüştür.

$$W_{\text{son}} = [1 \ -1 \ 0 \ 1 \ -1 \ 0 \ 1 \ 0 \ 1 \ -1] \text{ olur.}$$

7. Pattern Sınıflandırma Programının Ekran Görüntüsü

Form1

Eğitim Verileri--pattern tanımlama

sınıf A-Giriş

sınıf B-Giriş

bias -1 $yA=[1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ -1]$

bias -1 $yB=[0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ -1]$

sınıf A-Çıkış

İstenen Çıkış: 1

sınıf B-Çıkış

İstenen Çıkış: -1

Tek Aynık Perceptron--başlangıç değerleri

Ağırlıklar ve Bias

$w_0=0$ $w_1=0$ $w_2=0$ $w_3=0$ $w_4=0$

$w_5=0$ $w_6=0$ $w_7=0$ $w_8=0$ bias -1

$W=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -1]$

Eğitim Süreci

Eğitimi sonlandırma kriteri

Adım Sayısı Adım Sayısı= 4

Değişim olmayış

Eğitime Başla

Eğitim Adımları

Adım	W Ağırlık Matrisi, ve bias	İstenen Çıkış	Aktüel Çıkış
1	0 0 0 0 0 0 0 0 0 -1	1	1
2	0 -1 0 0 -1 0 0 -1 0 0	-1	1
3	1 -1 0 1 -1 0 1 0 1 -1	1	-1
4	1 -1 0 1 -1 0 1 0 1 -1	-1	-1

$W_{son}=[1\ -1\ 0\ 1\ -1\ 0\ 1\ 0\ 1\ -1]$

Perceptron Ağ Modeli

$W = ?$

$OUTPUT = 0 = sgn(w'y)$

Test

$testData=[1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ -1]$

Sınıflandırma Sonucu

A