

## Yapay Sinir Ağları

### \*Yapay Sinir Ağları

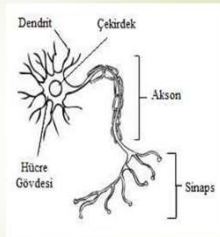
- Sinir ağı, beynin belirli bir işi veya fonksiyonu yerine getirme yöntemidir.
- Yapay sinir ağı, Sinir ağını modelleyen bir makinedir.

1

2

### \*Biyolojik Sinir Hücresi

- Biyolojik bir sinir hücresi: bir gövde, bir akson, çok sayıda sinir ucu (dendrit) ve akson ile diğer sinir hücresinin sinir ucu arasında kalan ince uzantılar (sinaps) olmak üzere dört bölümden oluşmaktadır.
- Dendritler, gelen sinyalleri çekirdeğe iletir.
- Çekirdek dendritten gelen sinyalleri bir araya toplar ve aksone iletir. Toplanan bu sinyaller, akson tarafından işlenerek sinapslara gönderilir. Sinapslar da yeni üretilen sinyalleri diğer sinir hücrelerine iletir.
- Yapay sinir hücreleri, gerçek sinir hücrelerinin simüle edilmesiyle gerçekleştirilir.



### Biyolojik ve Yapay Sinir Hücresi

Yapay sinir ağlarının oluşturulmasında da sinir sistemi model alınmış ve benzer tanımlamalar yapılmıştır.

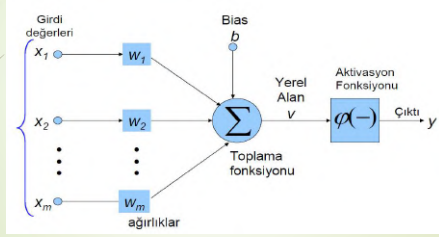
Biyolojik sinir hücresi	Yapay sinir hücresi
Akson	Çıktı
Dendrit	Toplama Fonksiyonu
Çekirdek	Aktivasyon fonksiyonu
Sinaps	Ağırlıklar

3

4

## Biyolojik ve Yapay Sinir Hücresi

### Temel nöron yapısı



## \*YSA Tanım

- Yapay Sinir Ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir.

## Tarihçe

- 1943 yılı YSA'nın gelişiminin başlangıç yılı olarak kabul edilmektedir. Bu tarihte **McCulloch ve Pitts**, **ilk sinir hücresi** modelini geliştirmişlerdir. McCulloch psikiyatrist ve nöroanatomist idi. Sinir sisteminin modellenmesi üzerine 20 kışur yılını harcamıştır. Pitts ise bir matematik dahisi idi ve 1942'de McCulloch ile çalışmaya başladı. (Haykin, 1999).
- İkinci önemli gelişme 1949'da Hebb ile geldi. Bu bilim adamı hücre bağlantılarını ayarlamak için kendi adıyla anılan ilk öğrenme kuralını geliştirdi.
- 1958'de Rosenblatt, algılayıcı (perceptron) modelini ve öğrenme kuralını geliştirerek bugün kullanılan kuralların temelini koydu.
- 1960 yılında Widrow ve Hoff tarafından LMS (Least Mean Squares) tanımlandı ve ADALINE (ADaptive Lineer Element)'in formüle edilmesinde kullanıldı.
- 1969 yılında, Minsky ve Papert, algılayıcının kesin analizini yaptı ve algılayıcının karmaşık lojik fonksiyonlar için kullanılamayacağını ispatladılar. Bunun üzerine, yapay sinir ağları üzerine yapılan çalışmalar durma noktasına gelmiştir.

- 1982 yılında Hopfield YSA'nın birçok problemi çözebilecek kabiliyeti olduğunu göstermiştir. Optimizasyon gibi teknik problemleri çözmek için doğrusal olmayan Hopfield ağını geliştirmiştir.
- 1982 yılında Kohonen öz düzenlemeli harita (selforganizing map SOM) yı tanımlamıştır. Kendi adıyla anılan eğitimcisz öğrenen bir ağı geliştirmiştir.
- 1983'de Barto, Sutton ve Anderson *reinforcement learning'i* geliştirdiler.
- 1986 yılında Rumelhart, Hinton ve Williams geriye yayılım algoritmasını geliştirdiler.
- 1988 yılında Broomhead ve Lowe Radyal RBF'i, Chua ve Yang hücrel sinir ağlarını geliştirdiler.
- 1990 ve sonrasında sayısız çalışma ve uygulamalar geliştirilmiştir.

### \*Uygulama alanları

- Sınıflandırma
- Örnek tamamlama
- Optimizasyon
- Örnek eşleştirme
- Gürültü uzaklaştırma
- Finansman ve Yatırım
- Üretim
- Kontrol
- Tıp
- Bilim ve Mühendislik

9

### YSA'nın özellikleri

- Kendi kendini organize etme ve öğrenebilme yetenekleri vardır.
- Eksik bilgi ile çalışabilmektedirler.
- Hata toleransına sahiptirler.
- Belirsiz ve tam olmayan bilgileri işleyebilmektedirler.
- YSA, ani bozulma göstermezler.
- Sadece nümerik bilgiler ile çalışabilmektedirler.
- YSA, normal yollarla çözülmesi zor olan problemleri çözmek için tasarlanmıştır.

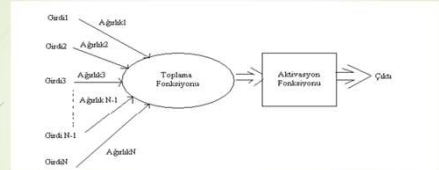
10

### \*Yapay Sinir Ağlarının Özellikleri

- Paralellik
- Öğrenebilirlik
- Hata toleransı
- Uyarlanabilirlik
- Genelleme
- Yerel bilgi işleme
- Gerçekleme kolaylığı
- Donanım ve hız

11

### \*Yapay Sinir Hücresi (YSH)



- Dış ortamdan veya diğer hücrelerden alınan **girdiler**, **ağırlıklar** yardımıyla hücreye bağlanır. **Toplama fonksiyonu** ile net girdi hesaplanır. Net girdinin **aktivasyon fonksiyonundan** geçirilmesiyle net çıktı hesaplanır.
- Bu işlem aynı zamanda hücrenin çıkışını verir.

12

## YSH Elemanları

- **Girdiler**
  - Yapay sinir ağına dış dünyadan veya diğer bir hücreden gelen bilgilerdir.
- **Ağırlıklar**
  - Hücreler arasındaki bağlantıların sayısal değerini ifade etmektedir. Bir hücreye gelen bilginin değerini ve hücre üzerindeki etkisini gösterir.
- **Toplama Fonksiyonu**
  - Hücreye gelen girdileri ağırlıklarla çarpıp toplayarak o hücrenin net girdisinin hesaplanmasını sağlar.
- **Aktivasyon Fonksiyonu**
  - Hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktının belirlenmesini sağlar.
- **Çıktılar**
  - Aktivasyon fonksiyonları tarafından belirlenen çıktı değerleridir. Üretilen çıktı ya dış dünyaya, başka bir hücreye ya da kendisine girdi olarak gönderilebilir.

13

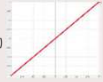

## Bazı Toplama Fonksiyonları

Toplam $Net = \sum_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak Net girdi hesaplanır.
Çarpım $Net = \prod_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleriyle çarpılarak Net Girdi Hesaplanır.
Maksimum $Net = \max(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en büyüğü Net girdi olarak kabul edilir.
Minimum $Net = \min(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en küçüğü Net girdi olarak kabul edilir.
Çoğunluk $Net = \sum_{i=1}^N Sgn(X_i * W_i)$	n adet girdi içinden girdilerle ağırlıklar çarpıldıktan sonra pozitif ile negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
Kümülatif Toplam $Net = Net(eski) + \sum_{i=1}^N X_i * W_i$	Hücreye gelen bilgiler ağırlıklı olarak toplanır. Daha önce hücreye gelen bilgilere yeni hesaplanan girdi değerleri eklenerek hücrenin net girdisi hesaplanır.

14

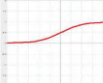

## Aktivasyon fonksiyonu

Bu fonksiyon hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. Yapay sinir ağlarının bir özelliği olan "doğrusal olmama", aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Aktivasyon fonksiyonu seçilmeden önce dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığı için hesaplamaların yavaşlamaması için türevi kolay hesaplanabilir bir fonksiyon seçilir. Günümüzde en yaygın olarak kullanılan "Çok katmanlı algılayıcı" modelinde genel olarak aktivasyon fonksiyonu olarak "Sigmoid fonksiyonu" kullanılır.

Doğrusal (Linear)		$F(Net) = A * Net$ (A sabit bir sayı)	Doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon olarak seçilebilir. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım (Step)		$F(Net) = \begin{cases} 1 & \text{if } Net > \text{Eşik Değer} \\ 0 & \text{if } Net \leq \text{Eşik Değer} \end{cases}$	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.

15

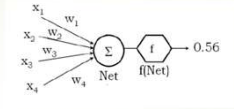
## Aktivasyon fonksiyonu

Sigmoid		$F(Net) = \frac{1}{1 + e^{-Net}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.
Tanjant Hiperbolik		$F(Net) = \frac{e^{Net} - e^{-Net}}{e^{Net} + e^{-Net}}$	Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.
Eşik Değer		$F(Net) = \begin{cases} 0 & \text{if } Net \leq 0 \\ Net & \text{if } 0 < Net < 1 \\ 1 & \text{if } Net \geq 1 \end{cases}$	Gelen bilgilerin 0 dan küçük-eşit olduğunda 0 çıktısı, 1 den büyük-eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretilebilir.

16

## Örnek

Girdiler	Ağırlıklar
X1=0.5	W1=-0.2
X2=0.6	W2=0.6
X3=0.2	W3=0.2
X4=0.7	W4=-0.1



Hücrenin net girdisi;

$$\text{Net} = \sum X_i W_i \quad i=1 \dots 4$$

$$\text{Net} = 0.5 * (-0.2) + 0.6 * 0.6 + 0.2 * 0.2 + 0.7 * (-0.1)$$

$$\text{Net} = 0.23$$

Sigmoid aktivasyon fonksiyonuna göre hücrenin çıkışı;

$$f(\text{Net}) = 1 / (1 + e^{-0.23})$$

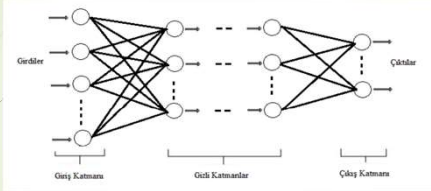
$$f(\text{Net}) = 0.56$$

17

- YSA, biyolojik sinir ağlarının diğer bir ifadeyle insan beyninin işleyişinin taklit edilerek bilgisayar ortamına aktarılmasıyla oluşturulan sistemlerdir.
- Mühendislik açısından bakıldığında YSA, "kara kutu" gibi görülebilir.
- Kara kutu, dışarıdan bilgileri alıp dışarıya ürettiği çıktıları vermektedir. İçeride ne olduğu ise bilinmemektedir. Diğer bir deyişle YSA'nın sonuçları nasıl oluştuğunu açıklama yeteneği yoktur. Bu durum ağa olan güveni sarsmakla birlikte başarılı uygulamalar yapay sinir ağlarına olan ilgiyi sürekli artırmaktadır.

18

## YSA'nın Yapısı



YSA, yapay sinir hücrelerinin birbirlerine bağlanması sonucu oluşan yapılardır. Ancak sinir hücrelerinin bir araya gelmesi rasgele olmaz.

19

## YSA'nın Yapısı

- Giriş Katmanı**
  - Yapay sinir ağına dış dünyadan girdilerin geldiği katmandır. Bu katmanda, girdi sayısı kadar hücre bulunmaktadır ve girdiler herhangi bir işleme uğramadan gizli katmana iletilirler.
- Gizli Katmanlar**
  - Giriş katmanından aldığı bilgiyi işleyerek bir sonraki katmana iletir. Gizli katman sayısı ve gizli katmandaki hücre sayısı ağdan ağa değişebilir.
  - Gizli katmanlardaki hücre sayıları, girdi ve çıktı sayılarından bağımsızdır.
- Çıkış Katmanı**
  - Gizli katmanlardan gelen bilgiyi işler ve giriş katmanına gelen girdiye uygun olarak üretilen çıktıyı dış dünyaya gönderir. Çıkış katmanındaki hücre sayısı birden büyük olabilir. Her bir çıkış hücresinin bir adet çıkışı vardır. Her bir hücre bir önceki katmandaki bütün hücrelere bağlıdır.

20

## YAPILARINA GÖRE YAPAY SİNİR AĞLARI

Yapay sinir ağları içerdiği nöronların birbirine bağlantı şekline göre ileri ve geri beslemeli olarak ikiye ayrılır.

**İleri Beslemeli Ağlar:** İleri beslemeli ağlarda nöronlar girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katmanlara bağ bulunmaktadır. Yapay sinir ağına gelen bilgiler giriş katmanına daha sonra sırasıyla ara katmanlardan ve çıkış katmanından işlenerek geçer ve daha sonra dış dünyaya çıkar.

**Geri Beslemeli Ağlar:** Geri beslemeli yapay sinir ağlarında ileri beslemeli olanların aksine bir hücrenin çıktısı sadece kendinden sonra gelen hücrenin katmanına girdi olarak verilmaz. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir hücreye de girdi olarak bağlanabilir.

## Yapay Sinir Ağ Modelleri

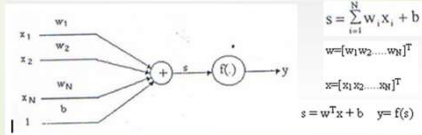
- Hücrelerin bağlantı şekilleri, öğrenme kuralları ve aktivasyon fonksiyonlarına göre yapay sinir ağları farklılaşmaktadır.

21

22

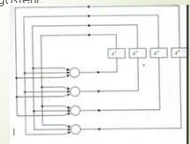
## \*İleri Beslemeli YSA

- Hücreler, girişten çıkışa doğru düzenli katmanlar şeklindedir. Ağa gelen bilgiler giriş katmanına daha sonra sırasıyla gizli katmanlardan ve çıkış katmanından işlenerek geçer ve sonra dış dünyaya çıkar.



## \*Geri Beslemeli YSA

- Geri beslemeli yapay sinir ağlarında bir katmandaki hücre girdi olarak sadece kendinden önceki hücrenin çıktısını değil aynı zamanda o katmanın veya daha sonraki katmanların çıktılarını da alabilir.
- Bu çeşit sinir ağlarında dinamik hafıza özelliği bulunmaktadır ve bir andaki çıkış hem o anın hem de önceki girişlerin sonucudur. Bu özelliği ile geri beslemeli yapay sinir ağları tahmin modellerinde sık kullanılır.
- Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir.



23

24

## Öğrenme Zamanına Göre Yapay Sinir Ağları

Yapay sinir ağları öğrenme zamanına göre de statik ve dinamik öğrenme olarak ikiye ayrılır.

**Statik Öğrenme:** Statik öğrenme kuralıyla çalışan yapay sinir ağları kullanmadan önce eğitilmektedir. Eğitim tamamlandıktan sonra ağı istenilen şekilde kullanılabilir. Ancak bu kullanım sırasında ağı üzerindeki ağırlıklarda herhangi bir değişiklik olmaz.

**Dinamik Öğrenme:** Dinamik öğrenme kuralı ise yapay sinir ağlarının çalıştığı süre boyunca öğrenmesini öngörerek tasarlanmıştır. Yapay sinir eğitim aşaması bittikten sonra da daha sonraki kullanımlarında çıkışların onaylanmasına göre ağırlıklarını değiştirerek çalışmaya devam eder.

## Ağırlık uzayı

Bir çok YSA öğrenme işlemi, işlem elemanlarının ağırlığı değiştirilerek sağlanır. Böylece tanımlanan ağırlık değiştirilerek öğrenmede iyi bir model kullanılıp, ağırlıkların bu modele göre değiştirilmesi esastır.

Örneğin;  $x_1=10$ ,  $x_2=18$ ,  $x_3=24$ ,  $x_4=6$  olarak belirtilen 4 girdili bir hücrenin başlangıç ağırlıkları şöyle varsayalım;  $w_1=w_2=w_3=w_4=0,4$

Bir başka girdi kombinasyonu da şöyle verilsin;  $x_1=11$ ,  $x_2=18$ ,  $x_3=20$ ,  $x_4=2$

Eğer hücre iki örüntüye de 1 cevabi vermiş ise ağırlıklar örneğin şu şekilde değişime uğratacağıdır,  $x_1$  yükseldiği için  $w_1=0,45$ ;  $x_2$  değişmediği için  $w_2=0,4$ ;  $x_3$  düştüğü için  $w_3=0,35$ ;  $x_4$  düştüğü için  $w_4=0,35$ . Böylece hücre her iki girdi ile karşılaştığında 1 cevabi verecek şekilde ağırlıklar (korelasyon matrisi değerleri) ayarlandı. Diğer bir ifadeyle iki girdi örüntüsünü öğrendi.

25

26

## AĞIRLIK UZAYI

Basit bir matematiksel model olarak her bir işlem elemanının "n" adet gerçek ağırlığı olduğu düşünülerek ve N adet işlem elemanı göz önüne alınırsa;

$$W = (W_{11}, W_{12}, \dots, W_{1n}, W_{21}, W_{22}, \dots, W_{2n}, W_{N1}, W_{N2}, \dots, W_{Nn})^T$$

$$W = (W_1^T, W_2^T, W_3^T, \dots, W_N^T)^T$$

$W_1, W_2, W_3, \dots, W_N$  işlem elemanlarının ağırlık vektörleridir.

YSA ağırlık vektörü N, n boyutlu uzaya yayılır. YSA'nın enfomasyon işleme performansı, ağırlık vektörünün belirli bir değeri ile bulunacaktır.

Hata değişimini inceleyen iki çeşit kural vardır.

- 1- Hata düzeltme kuralları
- 2- Gradyen kuralları.

**Hata Düzeltme Kuralları:** Her bir iş örüntüsünde ağırlıklar yeniden ayarlayarak çıktı hatalarını en aza indirmeye çalışılır.

**Gradyen Kuralları:** Ağırlıklar yeniden ayarlanarak ortalama karesel hatayı (MSE - Mean Squared Error) en aza indirmeye çalışılır.

$$F(w) = \int_A |f(x) - G(x, w)|^2 \rho(x) dv(x)$$

Amaç F'i küçültmeye çalışmaktır.

$y=G(x,W)$ : sistemin giriş çıkış fonksiyonu

$y$ : çıkış işaret vektörü,  $x$ : giriş işaret vektörü

$W$ : Ağırlık vektörü

$p(x)$ : Olasılık yoğunluk fonksiyonu

27

28



## Öğrenme (Yapay Sinir Ağlarının Eğitilmesi)

- Yapay sinir ağlarının öğrenme sürecinde de dış ortamdan girişler alınır, aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir. Bu çıkış yine tecrübeyle verilen çıkışla karşılaştırılarak hata bulunur.
- Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır. Bu çalışma süresince yenilenen yapay sinir ağının ağırlıklardır. Ağırlıklar her bir çevrimde yenilenerek amaca ulaşmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışardan verilen bir değerdir. Eğer **yapay sinir ağı verilen giriş-çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır.**
- Ağırlıkların sürekli yenilenerek istenilen sonuca ulaşılan kadar geçen zamana **öğrenme** adı verilir. Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilir, sinir ağı çıkışıyla gerçek çıkış yaklaşımlıncelenir. Eğer yeni verilen örneklere de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş, ezberlemiş demektir. Genelde eldeki örneklerin yaklaşık %70-%80'i ağına verilir ağı eğitilir. Daha sonra kalan %20-%30'lık kısım verilip ağıın davranışı incelenir. Böylece ağıın testi yapılmış olur.

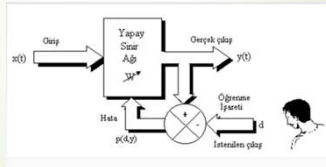
## Yapay Sinir Ağlarında Öğrenme

29

30

## \*Danışmanlı Öğrenme - Gözetimli Öğrenme

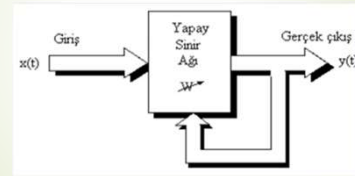
- YSA, kullanılmaya başlanmadan önce eğitilir. Eğitim sırasında, hem girdi değerleri hem de o girdi değerleri için istenen çıktı değerleri sisteme verilir. İstenen çıktı ile ağıın ürettiği çıktı karşılaştırılarak hata hesabı yapılır, ağırlıklar güncellenir.



31

## \*Danışmasız Öğrenme – Gözetimsiz Öğrenme

- Sistemin öğrenmesine yardımcı olan herhangi bir danışman yoktur. Sisteme sadece girdiler verilir, örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendine öğrenmesi beklenir.

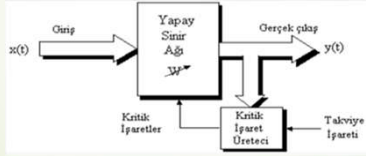


32



### \*Pekiştirmeli Öğrenme – Takviyeli Öğrenme

- Giriş verileri ağı uygulanır ve sonucun danışman tarafından değerlendirilmesi istenir. Ödüllendirme ve cezalandırma yöntemiyle ağı ağırlıkları güncellenir.



33

### Öğrenme yöntemine göre uygulamalar

#### ■ Danışmanlı öğrenme:

- Regresyon (Fitting) problemi (Matlab ağı: **fitnet**)

Veriler sayısal, Çıktılar önceden verilmiş (Etiketli), Çıktı tahmini

- Sınıflandırma problemleri (Matlab ağı: **patternnet**)

Veriler sayısal veya kategorik, Sınıflar etiketlenmiş, Sınıf tahmini

34

### Öğrenme yöntemine göre uygulamalar

#### ■ Danışmansız öğrenme:

- Kümeleme problemi (Matlab ağı: **selforgmap**)

Veriler sayısal, Veriler etiketsiz (Sınıflar bilinmiyor)

35

### Öğrenme Algoritmaları

- Geri Yayılım Algoritması
- Esnek Yayılım Algoritması
- Delta Bar Delta Algoritması
- Geliştirilmiş Delta Bar Delta Algoritması
- Hızlı Yayılım Algoritması
- Levenberg-Marquardt Metodu
- Eşleştirmeli Eğitim Algoritmaları
- Genetik Algoritma
- .....
- <https://www.mathworks.com/help/deeplearning/ug/choose-a-multilayer-neural-network-training-function.html>

36

## Veri normalizasyonu

YSA için kullanılacak eğitim ve test verileri ham hali ile işleme girdikleri zaman elde edilen sonuçların beklenen değerlerden farklı çıktığı ve eğitim süresinin uzadığı gözlemlenmiştir. Bu sebeple eğitim verilerinin olabildiğince normalize edildikten sonra eğitime başlanması tercih edilmektedir.

Yaygın olarak kullanılan normalizasyon tipleri şunlardır;

1. Min-Max
2. Standard Score

Ayrıca;

Sigmoidal (Softmax), Energy, Principal Components olarak farklı normalizasyon türleri de bulunmaktadır.

37

## Min-Max Normalization

Bu yöntemde, bir grup verinin içerisindeki en büyük ve en küçük değerler ele alınır. Diğer bütün veriler, bu değerlere göre normalize edilir. Buradaki amaç en küçük değeri 0 ve en büyük değeri 1 olacak şekilde normalleştirmek ve diğer bütün verileri bu 0-1 aralığına yaymaktır.

$$X_{normal} = \frac{X - X_{Asgari}}{X_{Azami} - X_{Asgari}}$$

Örnek: 5, 8, 9, 11, 20, 22, 24, 25, 27, 29 sayılarının normalize edilmiş hali;

5	8	9	11	20	22	24	25	27	29
0	0.125	0.146667	0.25	0.625	0.708333	0.791667	0.833333	0.916667	1

38

## Standard Score

Bu da normalleştirme yöntemidir. Bir önceki yöntemde, sayılar en yüksek ve en düşük değerlere göre normalleştirilmişti. Bu yöntemde ise ortalama değer (mean value) ve standart sapma (standard deviation) değerleri göz önüne alınır. Sistemde kullanılan standart sapmaya affen, standart skor (standard score) olarak da isimlendirilir. Oldukça popüler normalleştirme yöntemlerinden birisidir. Formülü aşağıdaki şekilde yazılabilir:

$$X_{normal} = \frac{X - \mu}{\sigma_x}$$

Örnek: 5, 8, 9, 11, 20, 22, 24, 25, 27, 29 sayılarının normalize edilmiş hali;

5	8	9	11	20	22	24	25	27	29
-1.54718	-1.19014	-1.07112	-0.8331	0.238028	0.474056	0.714083	0.833097	1.071125	1.309153

39

## YSA KULLANIMINDA UYGULANACAK ADIMLAR

1. **Örneklerin toplanması:** Örnek veri kümesi hazırlanır.
2. **Ağın topolojik yapısının belirlenmesi:** Probleme göre ağ türü belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane proses elemanı kaç tane çıktı elemanı olması gerektiği bu adımda belirlenmektedir.
3. **Öğrenme parametrelerinin belirlenmesi:** Ağın topolojik yapısına göre gerekli eğitim toplama ve aktivasyon fonksiyonları, momentum katsayısı vs. girilir. Katman ve hücre sayıları belirlenir.
4. **Ağırlıkların başlangıç değerlerinin atanması:** Başlangıçta rastgele değerler atanır. Ağ öğrendikçe ağırlıklar kendisi değişir.
5. **Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi:** Ağ öğrenmeye başlaması ve Öğrenme kuralına uygun olarak ağırlıkların değişmesi için ağa örnekler (%70-%80) belirli bir düzeneğe göre gösterilir.
6. **Öğrenme sırasında iteri hesaplamaların yapılması:** Sunulan girdi için ağın çıktı değerleri hesaplanır.
7. **Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması:** Ağın ürettiği hata değerleri bu adımda hesaplanır.
8. **Ağırlıkların değiştirilmesi:** Üretilen hatanın azalması için parametrelerin değiştirilmesi.
9. **Öğrenmenin tamamlanması:** Sinir ağı öğrenmeyi tamamlayıncaya, yani gerçekleşen ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar eğitim devam eder.

40

## Yapay sinir ağlarının kullanıldığı alanlar

Yapay sinir ağları başlıca; Sınıflandırma, Modelleme ve Tahmin uygulamaları olmak üzere, pek çok alanda kullanılmaktadır. Başarılı uygulamalar incelendiğinde, YSA'nın çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek sensör verilerinin olması ve problemi çözmek için matematiksel modelin ve algoritmaların bulunmadığı, sadece örneklerin var olduğu durumlarda yaygın olarak kullanıldıkları görülmektedir.

- **Uzay:** Uçuş simülasyonları, otomatik pilot uygulamaları, komponentlerin hata denetimleri vs.
- **Otomotiv:** Otomatik yol izleme, rehber, garanti aktivite analizi, yol koşullarına göre sürüş analizi vs.
- **Bankacılık:** Kredi uygulamaları geliştirilmesi, müşteri analizi ve kredi müracaat değerlendirilmesi, bütçe yatırım tahminleri vs.
- **Savunma:** Silah yönlendirme, hedef seçme, radar, sensör sonar sistemleri, sinyal işleme, görüntü işleme vs.
- **Elektronik:** Kod sırası öngörüsü, çip bozulma analizi, non-lineer modelleme vs.

41

## Yapay sinir ağlarının kullanıldığı alanlar

- **Eğence:** Animasyonlar, özel efektler, pazarlama öngörüsü vs.
- **Finans:** Kıymet biçme, pazar performans analizi, bütçe kestirimi, hedef belirleme vs.
- **Sigortacılık:** ürün optimizasyonu, uygulama politikası geliştirme vs.
- **Üretim:** üretim işlem kontrolü, ürün dizaynı, makina yıpranmalarının tespiti, dayanıklılık analizi, kalite kontrolü, iş çizelgeleri hazırlanması vs.
- **Sağlık:** göğüs kanseri erken teşhis ve tedavisi, EEG, ECG, MR, kalite artırımı, ilaç etkileri analizi, kan analizi sınıflandırma, kalp krizi erken teşhis ve tedavisi vs.
- **Petro kimya:** arama, verim analizi vs.
- **Robotik:** yürünge kontrol, forklift robotları, görsel sistemler, uzaktan kumandalı sistemler, optimum rota belirleme vs.
- **Dil/sözcük tanıma:** yazı ve konuşma çevrimi, dil tercüme vs.
- **Telekomünikasyon:** görüntü ve data karşılaştırma, filtreleme, eko ve gürültü azaltılması, ses ve görüntü işleme, trafik yoğunluğunun kontrolü ve anahtarlama vs.
- **Güvenlik:** parmak izi tanıma, kredi kartı hileleri saptama, retina tarama, yüz eşleştirme vs.

42

## YSA AVANTAJLARI

- YSA'lar **makine öğrenmesi** gerçekleştirebilirler. Olayları öğrenerek benzer olaylar karşısında **mantıklı kararlar** verebilirler.
- Örnekleri kullanarak öğrenirler.
- Daha önce görülmemiş örnekler hakkında bilgi üretebilirler.
- Örüntü (pattern) ilişkilendirme ve sınıflandırma yapabilirler.
- Kendi kendine öğrenebilme ve organize etme yetenekleri vardır.
- Eksik bilgi ile çalışabilmektedirler.

43

## Avantajları

- Doğrusal olmayan çok boyutlu, gürültülü, eksik bilgili ve özellikle problemin çözümünde kesin bir matematiksel modelin veya algoritmanın bulunmadığı durumlarda başarılıdır.
- İstisnai ve anormal veri sayısı çok olan konularda iyi sonuçlar verir.
- Adaptasyon yeteneği vardır.
- Bilgiler ağıın tamamında saklanır.

44

## YSA DEZAVANTAJLARI

- Donanım Bağımlıdır. Paralel işlem yapabilme yeteneği, paralel çalışan işlemciler ile performans gösterir.
- Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlandığı anlamına gelmektedir.
- Ağın davranışları açıklanamamaktadır. Bu sorun YSA'nın en önemli sorunudur. YSA bir probleme çözüm ürettiği zaman, bunun neden ve nasıl olduğuna ilişkin bir ipucu vermez. Bu durum ağa olan güveni azaltıcı bir unsurdur.

45

## Dezavantajları

- Uygun ağ yapısının belirlenmesinde belli bir kural yoktur.
- Ağın parametre değerlerinin belirlenmesinde belli bir kural yoktur.
- Eğitim örnekleri seçiminde genel bir kural yoktur.
- Öğrenilecek problemin ağa gösterimi önemli bir problemdir. Veriler nümerik ve belli kurallara göre ağa gösterilmelidir.
- Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur.

46

## Örnek: Super Mario

- [https://www.youtube.com/watch?v=Cl3FRsSAa\\_U](https://www.youtube.com/watch?v=Cl3FRsSAa_U)
- <https://youtu.be/qv6UVOQ0F44>



47

## Örnekler:

- <https://www.youtube.com/watch?v=wL7tSgUpy8w>
- <https://www.youtube.com/watch?v=zlBYwckuTk>

48