

T.C. TARIM VE ORMAN BAKANLIđI  
SU YÖNETİMİ GENEL MÜDÜRLÜĐÜ  
TAŞKIN VE KURAKLIK YÖNETİMİ DAİRESİ BAŞKANLIđI



# Yapay Sinir Ağları ile Hidrolojik Modelleme

Dr. Gökçen UYSAL  
Eskişehir Teknik Üniversitesi (ESTÜ)

İklim Deđişikliđinin Kar Erimelerine ve  
Akımlarına Etkisinin Belirlenmesi Projesi  
Eđitim Çalıřmaları  
10-04-2019



Tutarlı model çıktısı akım nasıl elde edilir?



# Biyolojik Öğrenme



- Bazı problemler **deneyim tabanlıdır**.
  - el yazısı, yüz ya da konuşma tanıma,
  - sürüş,
  - piyano çalmak gibi...
- Bu problemler yeterli sürede kazanılan bir **öğrenme** ile çözülmektedir.
- Bu problemlerin matematiksel modeller ile tanımlanmaları zordur! ( $ax+b=?$ )
- YSA, bu noktada beynin çalışma prensibinden yola çıkarak fiziksel bir modelin olmadığı, modelin veriden ve öğrenmeden elde edildiği tasarlanmış bir sistemdir.

Tekrarlama

Öğrenme

Taklit  
Etme

# Öğrenme beyinde nasıl gerçekleşir?

- **Beyin:** Sinir hücrelerinin bedeninin bir köşesinde yoğun ağ oluşturduğu bir organdır.
- Tek işlemcili bilgisayarların aksine **beyin karışık bir düğümler kümesidir!**
- **Tek işlemci:** her adım sırası ile çalışır, **Milyar işlemcili beyin:** Her sinir hücresi aynı anda çalışır)
- Hücreler arası **haberleşme** vardır.
- Bu haberleşme ise kimyasal ve elektriksel sinyallerin kullanılması ile mümkündür.
- Öğrenme: **Hücreler arasında bağlantıların** güçlü mü zayıf mı olduğuna göre farklılaşan örüntüler farklı bilgileri kaydetmek için kullanılabilir!

## Basitçe:

- i) Girdi tellerinden (duyu organları veya beyindeki diğer hücreler) gelen sinyalleri ağırlıklı toplamını hesapla,
- ii) Sonuç bir eşik değeri üzerinde ise etkinleşerek bağlı olduğu diğer hücrelere sinyal yolla



# Biyolojik sinir hücresi (nöron)

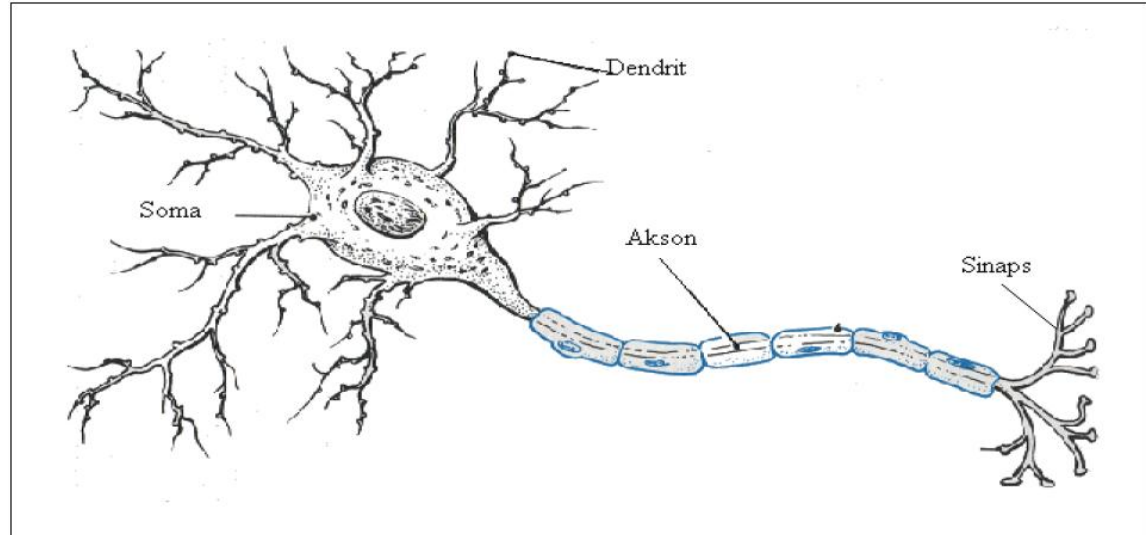
Beyinde 10 trilyon nöron bulur ve bir nöronun birkaç bin ilişkisi vardır

Bir biyolojik nöronun üç temel bileşeni vardır.

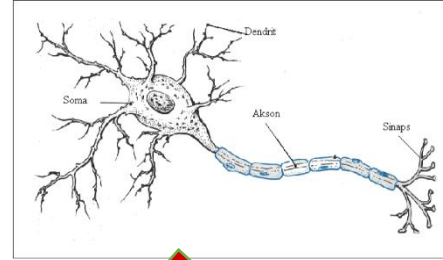
- ❑ *Dentritler*
- ❑ *Soma*
- ❑ *Aksonlar*

*Dentritler* sinyalleri diğer nöronlardan alırlar. *Sinyaller* akson ve dentrit arasındaki *sinaptik boşluklardan* kimyasal işlemlerin yardımıyla iletilen elektrik tepkileridir. Kimyasal vericilerin hareketi giren sinyalleri biraz değiştirir ve bir anlamda sinir ağlarındaki *ağırlıklara* benzemektedir.

*Soma* veya *hücre gövdesi* giren sinyalleri toplar, yeterli girdi alındığında *hücre aktive olur* ve bir çıktı üretilir. Bu nöronun sinyali *aksonlar* yardımıyla başka nöronlara (*sinapsdan*) aktarılır.

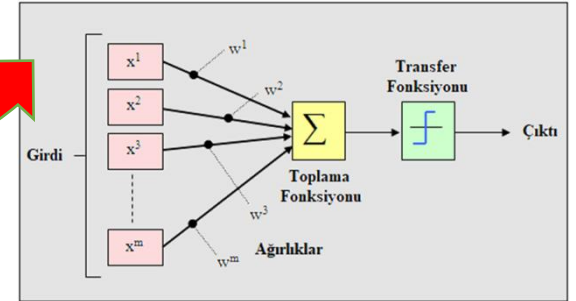
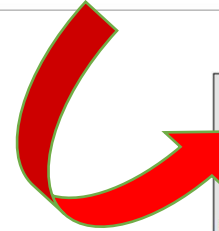


# Beyin örnek alınarak bir model tasarlanabilir mi?



❑ Biyolojik Sinir Ağları

❑ Yapay Sinir Ağları (YSA)

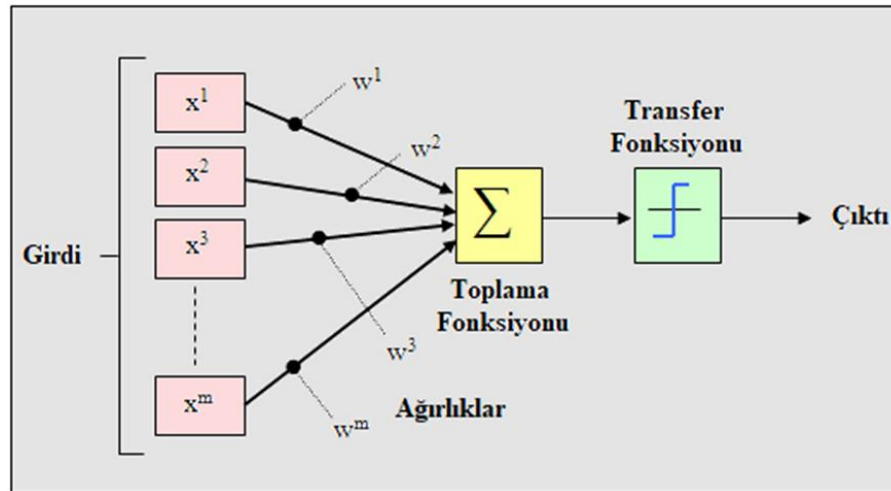


Yapay sinir ağları kavramı, beynin çalışma ilkelerinin bilgisayarlar üzerinde benzetimlerinin yapılması fikri ile ortaya çıkmıştır.

# Genel bir YSA Yapısı

*Bu yapıda da beyin sinir yapısında olduğu gibi 4 ana yapı mevcuttur.*

1. Öncelikle sisteme **girdi** verilmesi gerekmektedir. Bu girdiler belirli ağırlıklarla atanarak toplama işlevine aktarılırlar.
2. Toplama işlevine aktarılan bu veriler daha sonra etkinlik işlevi olan **transfer fonksiyonuna** gönderilir. Burada eğitilen veriler tekrardan toplama fonksiyonuna geri gönderilir.
3. Döngü **öğrenme aşaması (belirli bir hedefe benzetme)** tamamlanana kadar devam ettirilir.
4. Öğrenme sona erdikten sonra **çıktılar** elde edilebilir.



# YSA Modelinde Öğrenme Nasıl Gerçekleşir?

Ağırlık ayarlaması (eğitim) metotları, mimariye ek olarak farklı YSA'ların önemli seçim karakteristiklerinden biridir.

Eğitim metotları iki temel sınıfa ayrılmıştır:

- **Supervised (kontrollü öğrenme)**

*Tahmin, örüntü tanıma vb. uygulamalarda*

- **Unsupervised (kontROLSÜZ öğrenme)**





# YSA yapısının matematiksel avantajı nedir?

- **Yüksek seviyede paralellik** ile, nöron seviyesindeki basit yerel hesaplamalar ağ seviyesinde bir araya getirilerek toplu sonuçların elde edilmesi sağlanır.

- Farklı bir çok **mimari** oluşturulabilir.



- Hata toleransı vardır, hatayı en uygun şekilde azaltır.  
(**Ağırlıklar sürekli güncellenir**)



- Öğrenim ve adaptasyon yeteneği yüksektir.



***YSA çok çeşitli problemlerin çözümünde kullanılabilir.***

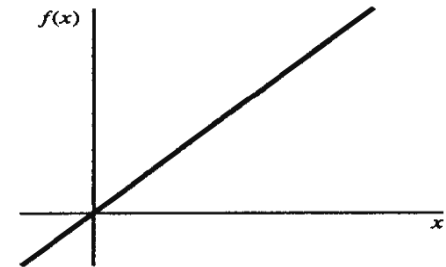
- ❑ Bilgileri ve örnekleri saklama ve tekrar çağırma,
- ❑ Örnekleri (numuneleri) sınıflandırma,
- ❑ Giriş örneklerinden çıkış örneklerinin haritalanması,
- ❑ Benzer örneklerin gruplaştırılması,
- ❑ Bazı optimizasyon problemlerinin çözümlerinin bulunmasında vb. problemlerde

# Tarihçe

- Pitts & McCulloch (1943)
  - Biyolojik nöronların ilk matematiksel modeli
  - Bütün ikili (boolean) işlemler nöron benzeri düğümler ile gerçekleştirilebilir (farklı eşik değerleri ile)
- Hebb (1949)
  - Öğrenim için Hebbian kuralı : her ne zaman  $i$  ve  $j$  aktive edilirse;  $i$  ve  $j$  nöronları arasındaki bağlantı gücü artar.
  - Veya her ne zaman eşzamanlı olarak  $i$  ve  $j$  nöronlarının her ikisi birden ON veya OFF yapılırsa  $i$  ve  $j$  düğümleri arasındaki bağlantı gücü artar.

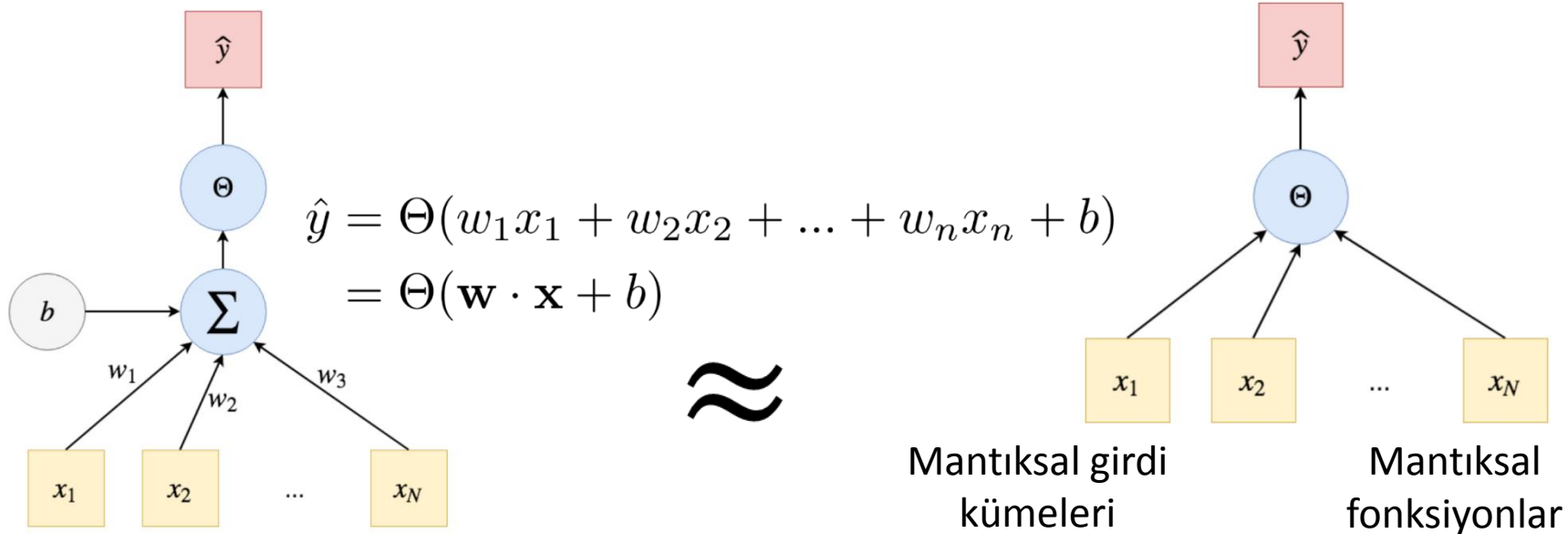
# Aktivasyon Fonksiyonu

- Bir nöronun çıktısını elde edebilmek için o nöronda oluşan çarpımlar toplamının (net) **bir fonksiyondan geçirilmesi ihtiyacı vardır**. İlgili fonksiyona aktivasyon fonksiyonu adı verilir.
- Temel olarak;
  - Identity function :  $f(\text{net}) = \text{net}$  // net giriş toplamı çıkış olarak elde edilir.
  - Constant function :  $f(\text{net}) = c$  // net toplamı ne olursa olsun sonuç hep  $c$  gibi bir değerdir.
  - Step function:  $\text{net} < c$  iken  $f(\text{net}) = a$  ve  $\text{net} > c$  iken  $f(\text{net}) = b$  // net toplamının  $c$  değeriyle ilişkisine göre sonuç elde edilir.
  - Ramp function



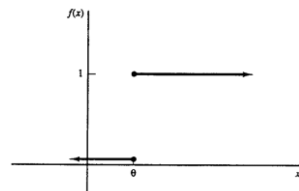
# Perseptron Modeli

- 1950lerin sonlarına doğru Minsky ve Papert tarafından tanıtılmıştır.
- Perceptron, doğrusal olarak ayrılabilir herhangi bir girdi grubunu sınıflandırmada kullanılır.



Model yalnızca eşik değeri üzerinde tepki vermeli!

$$\Theta(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



Eşik fonksiyonu (Heaviside)

A	B	A VE B	A VEYA B	A DEĞİL
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

# Bir perseptron modeli ne yapabilir?

Her model kapasitesi?

- 'NOT' fonksiyonunu (kapısı):

$\hat{y} = \Theta(wx + b)$  için  $w = -1$  ve  $b = 0,5$  seçersek

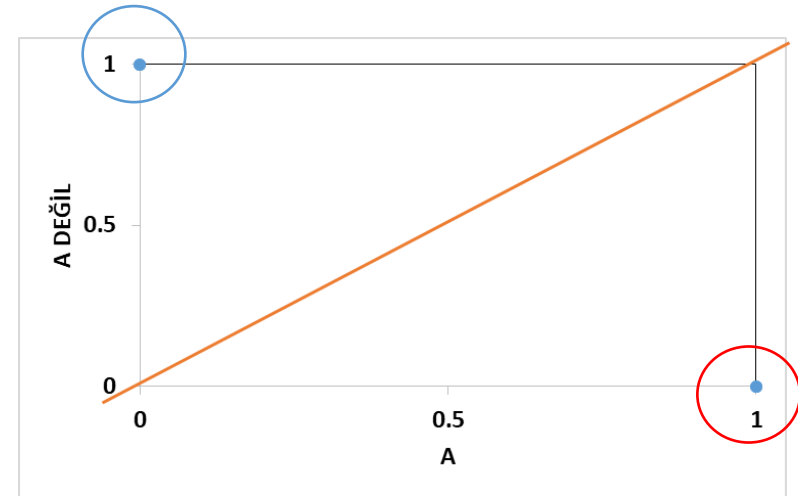
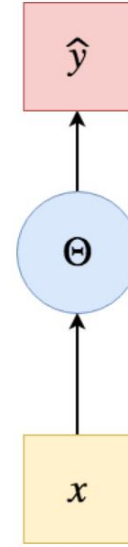
A	A DEĞİL
0	1
1	0

$$\begin{aligned}y(0) &= \Theta(-1 * 0 + 0,5) \\ &= \Theta(0,5) \\ &= 1\end{aligned}$$

$$\begin{aligned}y(1) &= \Theta(-1 * 1 + 0,5) \\ &= \Theta(-0,5) \\ &= 0\end{aligned}$$

$$\Theta(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

SONUÇ;  
DEĞİL(0)=1  
DEĞİL(1)=0



# Bir perseptron modeli ne yapabilir?

- 'VE' fonksiyonunu :

$\hat{y} = \Theta(wx + b)$  için  $w_1 = 1, w_2 = 1$  ve  $b = -1,5$  seçersek

$$\Theta(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

A	B	A VE B
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{aligned} y(0) &= \Theta(\widehat{-1 * 0 + 0,5}) \\ &= \Theta(0,5) \\ &= 1 \end{aligned}$$

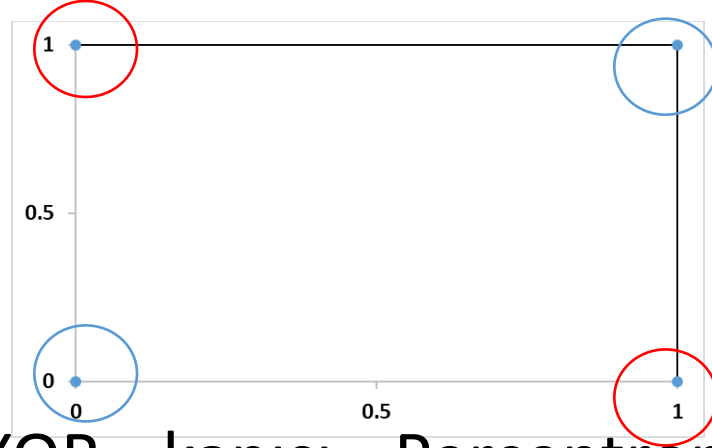
$$\begin{aligned} y(1) &= \Theta(\widehat{-1 * 1 + 0,5}) \\ &= \Theta(-0,5) \\ &= 0 \end{aligned}$$

SONUÇ;  
VE(1,1)=1  
VE(1,0)=0  
VE(0,1)=0  
VE(0,0)=0

# XOR Problemi ve Persptron Ađı

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

$$XOR(x1, x2) = AND(NOT(AND(x1, x2)), OR(x1, x2))$$



- Diđerlerinin aksine XOR kapısı Perceptron ađı yardımıyla **çözölememiştir**.
- Bu durum YSA çalıřmalarını da bir dönem gecikmeye uğratmıřtır.
- XOR kapısının Perceptron ile çözölememesi nedeniyle Çok Katmanlı Perceptron ađı geliřtirilmiřtir.



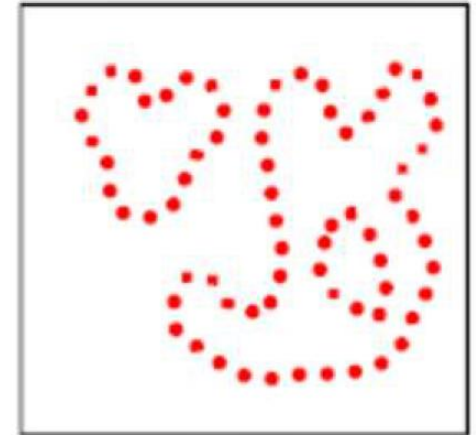
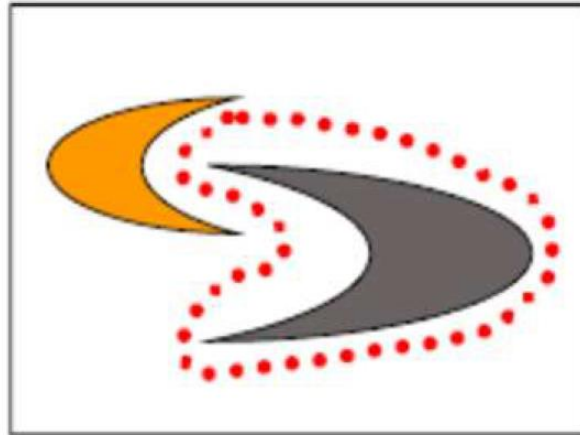
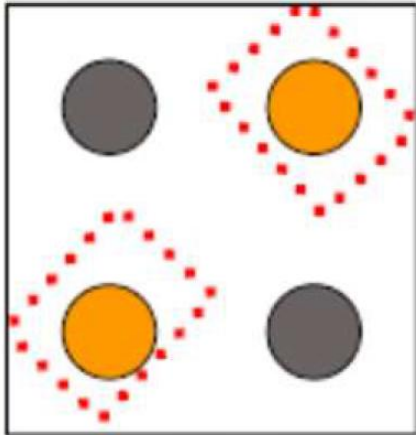
# Perseptron Öğrenmesi

- ❑ Sınıflandırması yapılacak veri için beklenen çıktı  $y$  ve ölçüm yapılan çıktı  $f(x)$  olmak üzere hata aşağıdaki gibi hesap edilir;
  - $E = y - f(x)$
- ❑ Ağırlık güncelleme ( $w_j$  değerleri sistemin ağırlıkları ve  $\alpha$  öğrenim katsayısı olmak üzere)
  - $w_j = w_j + \alpha x_j E$

# Çok Katmanlı Ağlar

## Multilayer Perceptron (MLP)

- Perceptronlar doğrusal ayrılabilir problemler için uygun iken Multilayer perceptron (çok katmanlı algılayıcı) doğrusal türde olmayan sınıflandırmalar için de uygundur.
- Aşağıdaki şekilde görüldüğü gibi veri uzayını konveks alanlar şeklinde sınıflara ayırabilir.



# MLP Ağlar

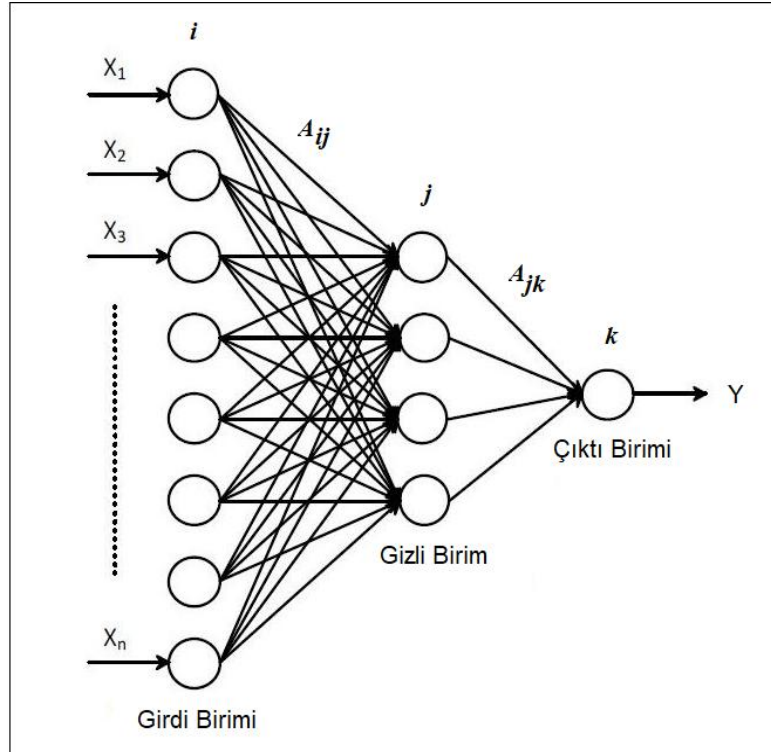
- Gerçek hayatta bir çok problem **doğrusal olmayan yapıdadır**.
- MLP ağları doğrusal olmayan problemlerin çözümünde en sık kullanılan YSA modelidir.
- MLP ağları için en popüler öğrenim yöntemi **Back Propagation (geriye yayılım)** yöntemidir.
- Back Propagation, ilk kez 1974 yılında Werbos tarafından önerilmiştir, şu anda kullanılan versiyon 1986 yılında Rumelhart, Hinton ve Williams tarafından geliştirilmiştir.

# İleri Beslemeli Geriye Yayınım Sinir Ağı

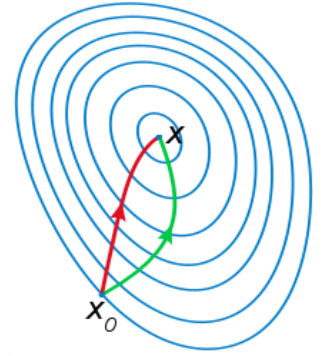
- Çalışmada kullanılan
  - girdi, gizli katman, çıktı birimleri olmak üzere üç farklı birim bulunmaktadır.

Literatürde kullanılan YSA modellerinin çoğu ileri beslemeli ve hatayı geriye yaymalı sinir ağı modellerine dayanmaktadır (Şekil 5.17).

$$V_i = \sum_{i=1}^n w_{ij}x_i + \theta_j$$



# MLP Çalışma Prensipleri ve Back Propagation (BP) öğrenimi



□ Bu eğitim metodu Hataların Geriye Yayılımı veya Genelleştirilmiş Delta Kuralı olarak bilinir.

- Basitçe bu metot, ağ tarafından ağın çıkışında hesaplanan toplam hata karesini azaltmaya yönelik gradyan iniş metodudur (Gradient Descent Metod ).

□ Ağın geriye yayılım ile eğitilmesi 3 aşamada yapılır :

- Girdi eğitim numunelerinin ileri beslemesi
- Birleşmiş hatanın hesaplanması ve geriye yayılması (bu işlem için aktivasyon fonksiyonu türevi kullanılır)
- Ağırlıkların ayarlanması

Eğitimden sonra ağ uygulaması sadece ağın ileri besleme aşamasını gerçekleştirir ve hesaplamaları yapar.

# Ağırlıkların Güncellenmesi (Geriye Yayılım)

- Hesap edilen değer  $o_k$  ve beklenen değer  $t_k$  olmak üzere çıkış katmanındaki hata;

$$E=(t_k - o_k)$$

**Amaç:** Öyle ağırlıklar seçelim ki E minimum olsun!

- Son katmandaki hatadan gizli katman ve giriş katmanındaki ağırlıkları düzeltmek için önce son katman için hata katsayısı olan  $\delta_k$  hesap edilir.

$$\delta_k = (t_k - o_k)F'(Net_k) \text{ (hata*çıkış türevi)}$$

- Gizli katmandan giriş katmanına yayılım için de  $\delta_j$  değerine ihtiyaç vardır.

$$\delta_j = F'(Net_j) \sum (\delta_k w_{jk})$$

- Gizli katmandaki ve çıkış katmanındaki hata katsayıları elde edildikten sonra;

$$w_{ij}(\text{yeni})=w_{ij}(\text{eski})+ \Delta w_{ij}$$

$$\Delta w_{ij}=\eta\delta_j o_j$$

# AKTİVASYON FONKSİYONU

Bir geriye yayılım ađında kullanılacak olan aktivasyon fonksiyonları **řu önemli karakteristikleri sađlamalıdır** :

- Sürekli olmalıdır**
- Türevi alınabilmelidir.**
- Fonksiyon monoton artan olmamalıdır.**
- Fonksiyonun türevi kendi terimleri ile ifade olunabilmelidir.**

En çok kullanılan aktivasyon fonksiyonları řunlardır :

**Binary Sigmoid :**

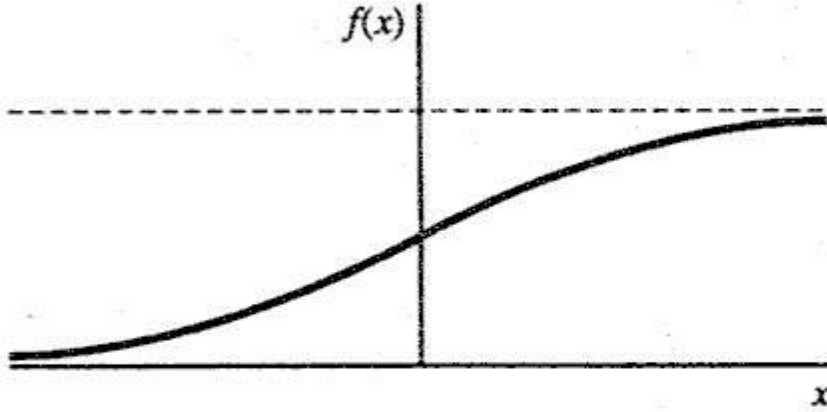


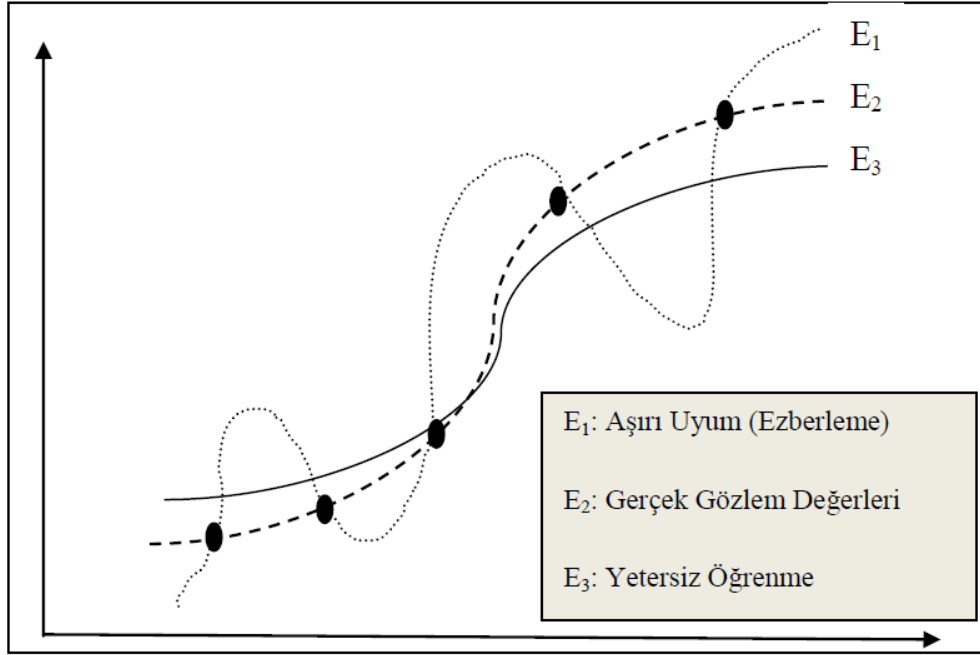
Figure 6.2 Binary sigmoid, range (0, 1).

$$f_1(x) = \frac{1}{1 + e^{-x}}$$

$$f_1'(x) = f_1(x)[1 - f_1(x)]$$

Binary sigmoid (0, 1) arasında deđerler üretir.

# Aşırı Uyum (Over-training) ve Yetersiz Öğrenme



Şekil 3-6 Aşırı Uyum ve Yetersiz Öğrenme Örneği

Aşırı ve yetersiz öğrenmiş bir YSA modeli için eğitim veri kümesi için ağ çıktılarının seyrini göstermektedir.

Buna göre aşırı uyum durumunda (E1) siyah noktalarla gösterilmiş olan eğitim veri kümesi noktaları tam olarak yakalanmış olmakla beraber, serinin gerçek seyri modellenememiştir. Diğer taraftan yetersiz öğrenme durumunda (E3) ise ne eğitim kümesindeki noktalar ne de serinin gerçek seyri yakalanabilmiştir.



# Güçlü Yönler

- Doğrusal olmayan modelleri temsiliyeti yüksektir.
- Yalnızca eğitim örnekleri kümesine ihtiyaç duyar.
- Detaylı çalışma alanı bilgisi gerektirmez.
- Eğitim örneklerindeki gürültü tolere edilebilir.
- İyi genelleştirme yeteneğine sahiptir.

# Zayıf Yönler

- Ağ bir kara kutudur.
- Bazı eğitim süreçleri uzun sürebilir.
- Hatanın minimuma indirilmesi garanti edilmez.
- Başlangıç ağırlıkları eğitim kalitesini etkiler.



Ağ Kullanım Tipi	Sinir Ağları	Sinir Ağının Kullanımı
Tahmin	<ul style="list-style-type: none"> <li>- Geriye Yayılım</li> <li>- Delta-Bar-Delta</li> <li>- Extended delta-bar-delta</li> <li>- Directed random search</li> <li>- Self organising map into back propagation</li> </ul>	Girdi değerlerini bazı çıktıları tahmin etmek için kullanır. (ekonometri, hava tahmini, akım tahmini, insanlardaki kanser risklerini belirleme)
Sınıflandırma	<ul style="list-style-type: none"> <li>- Learning Vector Quantization</li> <li>- Counter Propagation</li> <li>- Probabalistic neural network</li> </ul>	Girdi değerlerini sınıflandırmayı belirlemek için kullanır.(girilen harf A harfi mi, video görüntüsündeki şekil bir uçak mı ? Uçaksa cinsi ne? )
Veri İlişkilendirme	<ul style="list-style-type: none"> <li>- Hopfield</li> <li>- Boltzmann Machine</li> <li>- Hamming Network</li> <li>- Bidirectional Associate Memory</li> <li>- Spatio – temporal pattern recognition</li> </ul>	Sınıflandırma gibidir ama hatalı verileri de tanır (tarayıcıdan taranmış karakterleri tanımakla kalmaz ayrıca scanner tam doğru bir şekilde çalışmasada harfleri tanır)
Veri Kavramsallaştırma	<ul style="list-style-type: none"> <li>- Adaptive Resonance Network</li> <li>- Self Organising Map</li> </ul>	Girdi değerlerini analiz eder ve belirlenmiş olan ilişkileri gruplar. (belirli bir ürünü satın alan kişileri belirlemek)
Veri Filtreleme	<ul style="list-style-type: none"> <li>- Recirculation</li> </ul>	Girdi sinyallerini düzleştirir.( telefon sinyallerindeki gürültüyü kaldırma)

# YSA Tabanlı Hidrolik Modelleme

YSA'nın *doğrusal olmayan ilişkileri açıklayabilme (temsil etme) başarısı* hidrolojide bir çok uygulamada kullanılmasına olanak sağlamıştır.

- ✓ hidrolojik model,
- ✓ akım tahmini,
- ✓ baraj işletme,
- ✓ sediment taşınımı,
- ✓ su kalitesi,
- ✓ yeraltı suyu gibi...

# Hidrolojik Modellemede Girdi-Hedef

- **Girdi verileri** olarak meteorolojik verilerin kullanılması planlanmaktadır.
- Meteorolojik girdi verileri olarak
  - Toplam yağış,
  - Ortalama hava sıcaklığı,
  - Bağıl nem,
  - Radyasyon,
  - Rüzgâr hızı kullanılabilir.
- **Hedef verisi**
  - Akımlar

# YSA Model deęişkenleri:

- Bu alıřmada, **günlük toplam yağış (P)** miktarları ve **günlük ortalama sıcaklık (T)** her bir alt havzada istasyonlar bazında ve Thiessen aęırlıkları atanarak verilmiřtir.
- Bunun dıřında literatürde de uygulanan bir **zamansal düzeltme faktörü** girdi deęişkenlerine eklenerek akımlardaki periyodiklik modele tanıtılmıřtır.
- Bu zamansal deęişken özellikle **kar erime döneminin** modelde ifade edilmesi açısından yararlı olacağı ve model performansına pozitif katkı sağlayacağı düşünülerek farklı denemeler ile de doğruluęu test edilmiřtir.
- Model kurgusu girdi deęişkenleri olarak P ve T alınmıřtır; bu iki deęişkenle gerekli performans kriterlerinin sağlanıldığı görülmüřtür.
- Sadece bir alt havzada (**Ara Havza**) P ve T ile yeterli model performansı elde edilemedięinden bu alt havzada yüksek korelasyonda olmasından ötürü Murat-3 Alt Havzası'nın modellenen YSA akım deęerleri de girdi vektörüne eklenmiřtir.

# Kar Projesi Tasarım

- **Başlangıç ağırlıkları etkisi**
  - Model stabilitesi (100 bağımsız simulasyon)
- **Ağ Yapısı**

Dinamik ağlarda, çıktı sadece mevcut ağ girdisine değil hem mevcut zaman hem de önceki zaman girdi, çıktı ya da ağ durumlarına bağlıdır.

- **Model yapısı (Deneme-Yanılma)**
  - Tek gizli katman ve düğüm sayısı
  - 15 düğüm (nöron)

- **Sigmoid fonksiyon ve veri dönüşümü**

$$z_i = a + \frac{b(x_i - x_{min})}{(x_{max} - x_{min})}$$

- **Girdi – çıktı ilişkisi**

- Öğrenim sonuçlarının kalitesini eğitim örneklerinin kalitesi ve miktarı belirler.
- 9 alt havzada istatistiksel tutarlı çıktı dağılımı
- Çaltı örneği- akım kalitesi?

- **Ağırlıkların optimizasyonu**

- Levenberg-Marquardt (LM) yöntemi
  - LM algoritması aynı zamanda Hessian matrisin hesaplanmasında Jacobi matrisi ile hesaplanması kabulü sayesinde az epok adımı ile işlem süresi açısından da hızlı çözüm sunan bir optimizasyon algoritmasıdır.
- Over-training & durdurma koşulları

$$\Delta w_n = -[H + \lambda I]^{-1} \nabla_{w_n} E *$$

where,

$\Delta w_n$  is changes of connection weights

$H$  is the Hessian matrix, ( $H=J^*J^T$ )

$\lambda$  is learning rate,

$\nabla_{w_n} E$  is equal to the negative of the gradient error

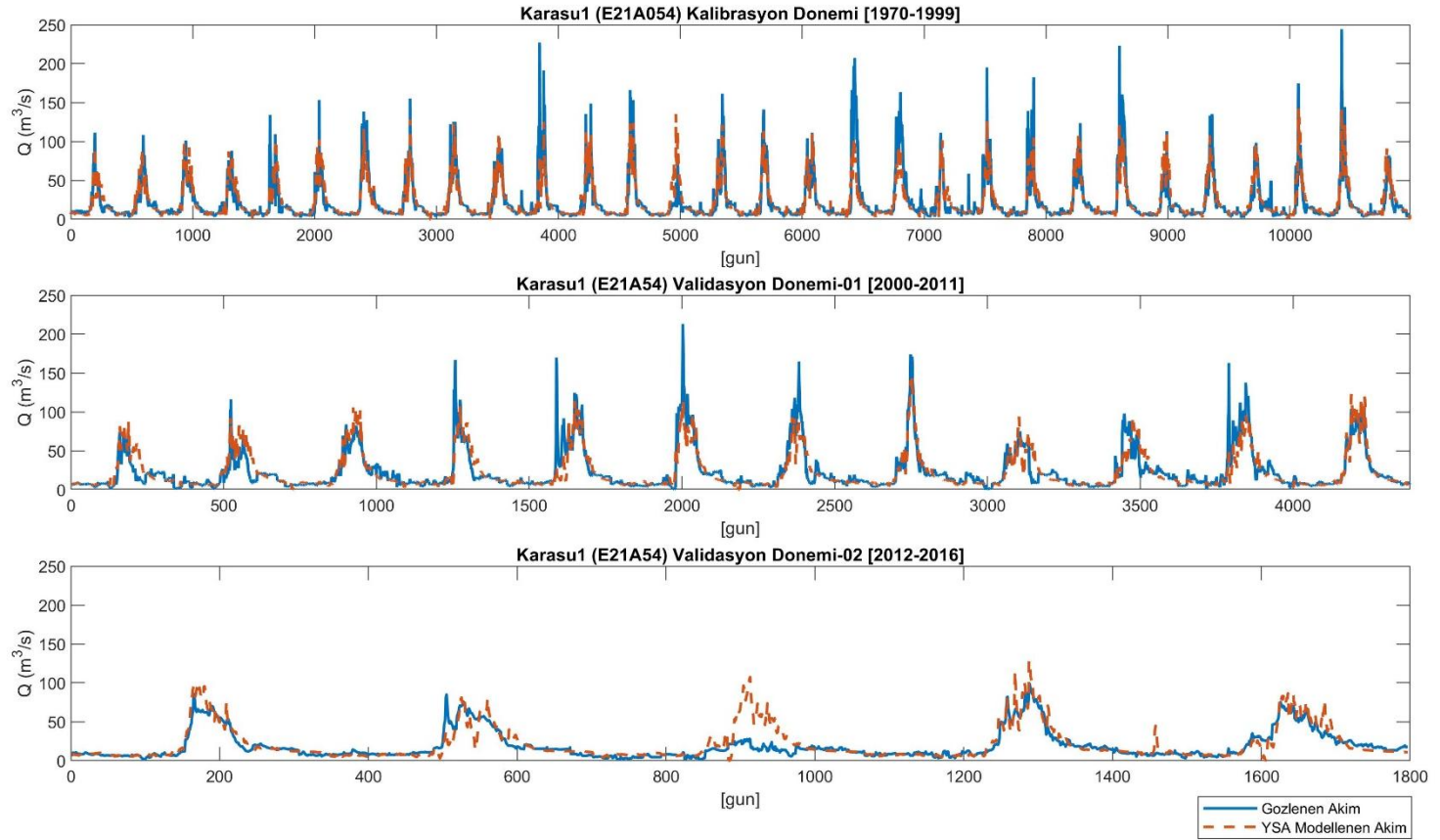
\*mean squared error (btw. output and targets) is used

**Durdurma koşulu (mse)**

$$F = mse \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - \alpha_i)^2$$

# Literatür uygulamalar ve Proje uygulaması

- Literatürdeki diğer YSA tabanlı birçok günlük akım tahmini çalışması, modele girdi olarak meteorolojik veriler yanı sıra **geciktirmeli (1-7 gün öncesine ait) akım gözlem değerlerini** de sağlamaktadır.
- Akım girdilerinin kendi içlerindeki **yüksek oto-korelasyonu** dikkate alındığında girdi ve hedef verisi korelasyonu yüksek bir çalışmada gözlenen akımların kendi içlerindeki hatası kaybolarak yüksek performans elde edilebilmektedir.
- Bu şekilde kurulacak bir model ile daha yüksek performans elde edilmesi bu alt havzalar için de beklenen bir durumdur.
- Fakat, **bu projeyi farklı kılan** ileriye dönük uzun yıllar (**sadece meteorolojik veri girdisi bazlı**) iklim projeksiyonu olduğu göz önünde bulundurularak bu şekilde bir yaklaşımdan kaçınılmıştır.



Örnek bir alt havza sonucu (Karasu1) ve tüm modellere performans

YSA										
ALT HAVZALAR	KARASU-1	KARASU-2	KARASU-3	MUNZUR SUYU	PERİ SUYU	ÇALTI SUYU	MURAT-1	MURAT-2	MURAT-3	ARA HAVZA
<b>KALİBRASYON PERİYODU</b>	(1970-1999)									
Model Başarısı (NSE)	0.790	0.858	0.781	0.858	0.765	0.696	0.776	0.830	0.854	0.846
<b>VALIDASYON PERİYODU</b>	(2000-2011)									
Model Başarısı (NSE)	0.756	0.798	0.750	0.709	0.670	0.400	0.727	0.770	0.762	0.708
<b>VALIDASYON PERİYODU (2)</b>	(2012-2016)									
Model Başarısı (NSE)	0.578	0.312	0.581	0.210	0.472	0.015	0.507	0.553	0.588	-0.329



# Tablo 5.12 Akım Değerlerinin İstatistikleri

Ortalama Akım (m <sup>3</sup> /s)										
	KARASU-1	KARASU-2	KARASU-3	MURAT-1	MURAT-2	MURAT-3	MUNZUR	PERİ	ÇALTI	ARAHAVZA
Kal	22.3	66.4	162.9	48.6	141.2	249.7	104.6	84.7	24.6	78.0
Val-1	23.4	70.8	167.9	51.2	134.2	233.8	112.3	90.0	20.2	79.1
Val-2	18.9	56.8	147.2	39.5	110.2	199.4	91.7	71.8	19.5	65.5
Kalibrasyon dönemine göre ortalama akımdaki değişim (%)										
	KARASU-1	KARASU-2	KARASU-3	MURAT-1	MURAT-2	MURAT-3	MUNZUR	PERİ	ÇALTI	ARAHAVZA
Val-1	4.6	6.5	3.1	5.3	-5.0	-6.4	7.4	6.3	-18.0	1.4
Val-2	-15.2	-14.6	-9.6	-18.7	-22.0	-20.2	-12.3	-15.2	-20.7	-16.0
Akımda Standart Sapma Q (m <sup>3</sup> /s)										
	KARASU-1	KARASU-2	KARASU-3	MURAT-1	MURAT-2	MURAT-3	MUNZUR	PERİ	ÇALTI	ARAHAVZA
Kal	27.2	77.0	119.0	70.8	195.2	344.7	94.4	114.2	38.8	88.1
Val-1	26.0	76.0	117.4	75.0	179.4	299.3	100.6	100.0	28.4	81.9
Val-2	17.9	53.6	85.3	47.0	117.0	225.7	66.5	71.0	29.1	58.3
Pik Akım (m <sup>3</sup> /s)										
	KARASU-1	KARASU-2	KARASU-3	MURAT-1	MURAT-2	MURAT-3	MUNZUR	PERİ	ÇALTI	ARAHAVZA
Kal	244.3	671.8	1137.3	688.0	1898.0	3291.2	1186.7	1139.3	697.1	795.8
Val-1	213.1	758.4	940.4	821.3	1521.5	2209.7	889.7	993.1	435.2	539.9
Val-2	98.3	293.7	509.7	360.3	863.2	1776.1	369.3	383.3	275.9	428.3

# Matlab (Makine Öğrenme)

- Neural Net Clustering (nctool) –kümeleme-
  - Solves clustering problem using Self Organizing Map (SOM) network
- Neural Net Fitting (nftool) –uydurma-
  - Solves fitting problem using two-layer feed-forward network
- Neural Net Pattern Recognition (nprtool) -örüntü tanıma-
  - Solves pattern recognition problem using two-layer feed-forward networks
- Neural Net Time Series (ntstool) -zaman serileri-
  - Solves nonlinear time series problem using dynamic neural networks

# nftool

Neural Fitting (nftool)



## Welcome to the Neural Network Fitting app.



Solve an input-output fitting problem with a two-layer feed-forward neural network.

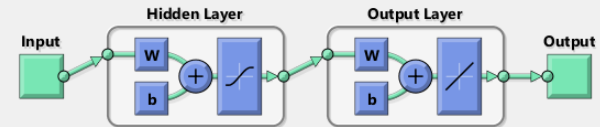
### Introduction

In fitting problems, you want a neural network to map between a data set of numeric inputs and a set of numeric targets.

Examples of this type of problem include estimating engine emission levels based on measurements of fuel consumption and speed (`engine_dataset`) or predicting a patient's bodyfat level based on body measurements (`bodyfat_dataset`).

The Neural Fitting app will help you select data, create and train a network, and evaluate its performance using mean square error and regression analysis.

### Neural Network



A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons (`fitnet`), can fit multi-dimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer.

The network will be trained with Levenberg-Marquardt backpropagation algorithm (`trainlm`), unless there is not enough memory, in which case scaled conjugate gradient backpropagation (`trainscg`) will be used.

➔ To continue, click [Next].

🧠 Neural Network Start

🔊 Welcome

⬅ Back

➡ Next

✖ Cancel

# engine\_dataset

**engine\_dataset** Engine behavior dataset.

Function fitting is the process of training a neural network on a set of inputs in order to produce an associated set of target outputs. Once the neural network has fit the data, it forms a generalization of the input-output relationship and can be used to generate outputs for inputs it was not trained on.

This dataset can be used to train a neural network to estimate an engines torque and emissions from its fuel use and speed.

LOAD [engine\\_dataset](#).MAT loads these two variables:

engineInputs - a 2x1199 matrix defining two attributes of a given engines activity under different conditions:

1. Fuel rate
2. Speed

engineTargets - a 2x1199 matrix of two attributes to be estimated given the inputs:

1. Torque
2. Nitrous oxide emissions

[X,T] = [engine\\_dataset](#) loads the inputs and targets into variables of your own choosing.

For an intro to fitting with the [Neural Fitting app](#) click "Load Example Data Set" in the second panel and pick this dataset.

Here is how to design a fitting neural network with 10 hidden neurons with this data at the command line. See [fitnet](#) for more details.

```
[x,t] = engine\_dataset;  
net = fitnet(10);  
net = train(net,x,t);  
view(net)  
y = net(x);
```

# ntstool

Neural Time Series (ntstool)



## Welcome to the Neural Network Time Series app.

Solve a nonlinear time series problem with a dynamic neural network.

### Introduction

Prediction is a kind of dynamic filtering, in which past values of one or more time series are used to predict future values. Dynamic neural networks, which include tapped delay lines are used for nonlinear filtering and prediction.

There are many applications for prediction. For example, a financial analyst might want to predict the future value of a stock, bond or other financial instrument. An engineer might want to predict the impending failure of a jet engine.

Predictive models are also used for system identification (or dynamic modelling), in which you build dynamic models of physical systems. These dynamic models are important for analysis, simulation, monitoring and control of a variety of systems, including manufacturing systems, chemical processes, robotics and aerospace systems.

This tool allows you to solve three kinds of nonlinear time series problems shown in the right panel. Choose one and click [Next].

### Select a Problem

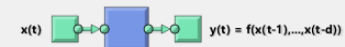
- Nonlinear Autoregressive with External (Exogenous) Input (NARX)  
Predict series  $\mathbf{y}(t)$  given  $\mathbf{d}$  past values of  $\mathbf{y}(t)$  and another series  $\mathbf{x}(t)$ .



- Nonlinear Autoregressive (NAR)  
Predict series  $\mathbf{y}(t)$  given  $\mathbf{d}$  past values of  $\mathbf{y}(t)$ .



- Nonlinear Input-Output  
Predict series  $\mathbf{y}(t)$  given  $\mathbf{d}$  past values of series  $\mathbf{x}(t)$ .



**Important Note:** NARX solutions are more accurate than this solution. Only use this solution if past values of  $\mathbf{y}(t)$  will not be available when deployed.

**!** Choose a problem, then click [Next].

Neural Network Start

Welcome

Back

Next

Cancel

# Kaynakça

- Yapay Sinir Ağları ve Uygulamaları – Dr. Hidayet Takçı
- Matlab
- <https://towardsdatascience.com/perceptrons-logical-functions-and-the-xor-problem-37ca5025790a>