

# BMÜ-430

# NÖRAL BULANIK KONTROL

Prof. Dr. Erhan AKIN

Fırat Üniversitesi Bilgisayar Mühendisliği

# Giriş

- İngilizce fuzzy kelimesinin sözlük anlamı bulanık, hayal mayal'dir.
- İlk defa Prof. Zadeh tarafından kullanılan bu terim temelde çok değerli mantık, olasılık kuramı, yapay zekâ ve yapay sinir ağları alanları üzerine oturtulmuş olayların oluşum olasılığından çok oluşum derecesiyle ilgilenen bir kavramı tanımlar.
- Olasılık ve bulanıklık kavramları arasındaki en önemli farklılık bulanıklığın bir deterministik belirsizlik olmasıdır.

Bu derste temel 2 konu ele alacağız. Bunlar;

- Bulanık Mantık
- Yapay Sinir Ağları

# Giriş

- Geleneksel mantık da bir kümeyi oluşturan elemanlar kesin elemanlar olup bir eleman kümenin ya elemanıdır tada değildir(var veya yok, 0 veya 1).
- Bu tür kümelere kesin kümeler denilir.
- Bir örnek olarak uzun boylu kavramını alalım.
- Eğer 1,75 boyu uzun boy olarak kabul edecek olursak, geleneksel kümelendirmede 1,75 boyun altındaki kişiler şekilde gösterildiği üzere uzun kabul edilmeyecektir.
- Bulanık kümelemede ise 1,75 boy %50 uzun olarak kabul edilecektir.

# Giriş

- Bir endüstriyel denetleyici için bu durum ele alınsın.
- Eğer bu denetleyicide fiziksel büyüklüklerin dâhil olduğu kümeler birbirlerinden böyle kesin çizgilerle ayrılmışlarsa denetim çıktısının ani değişiklikler göstermesi kaçınılmaz olacaktır.
- Örneğin soğuk /sıcak sınırının  $25^{\circ}$  olduğu bir sayısal açık/kapalı denetleyicide  $24.5^{\circ}$  soğuk olarak algılanacak, buna karşın  $25.5^{\circ}$  sıcak olarak ele alınarak denetim çıktısı ani olarak değiştirilebilecek, örneğin buhar vanası ani olarak kapatılabilecektir.

# Giriş

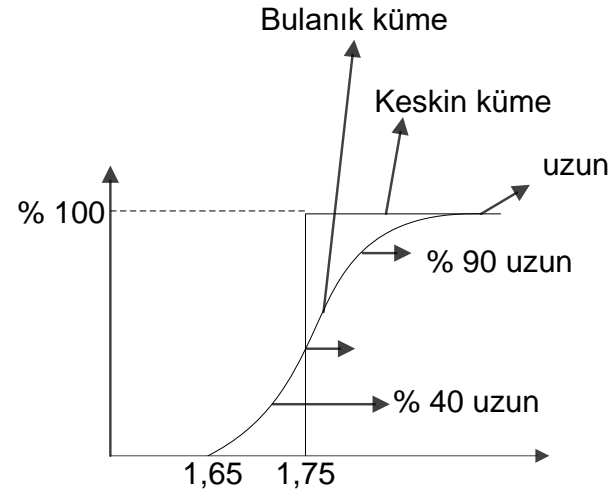
- Normal bir PID kontrol edici yapımıyla Bulanık mantıkla bir PID kontrol edici gerçekleştirilmede adam ve gün sayısı olarak çok fark vardır
- Bulanık mantık özellikle matematiksel modeli fazla yapılamayan matematiksel modeli oldukça karmaşık olan sistemlerde kullanılır
- Bulanık mantıkla matematiksel modeli olan bir sistem de çözülebilir.

# Giriş

- Bazen bulanık mantıkla çözüm daha fazla yarar sağlarken bazen de yarar sağlamayabilirler.
- Bizim öncelikli tercihimiz matematiksel modeli olan sistemlerin bulanık mantık kullanmadan çözülmesidir.
- Bulanık denetim kuramı temelde insan düşünüş tarzını örnek alır.
- Oldukça kapsamlı ve ayrıntılı bir matematiksel temeli varsa da, ana özellikleri aşağıdaki şekilde açıklanabilir.

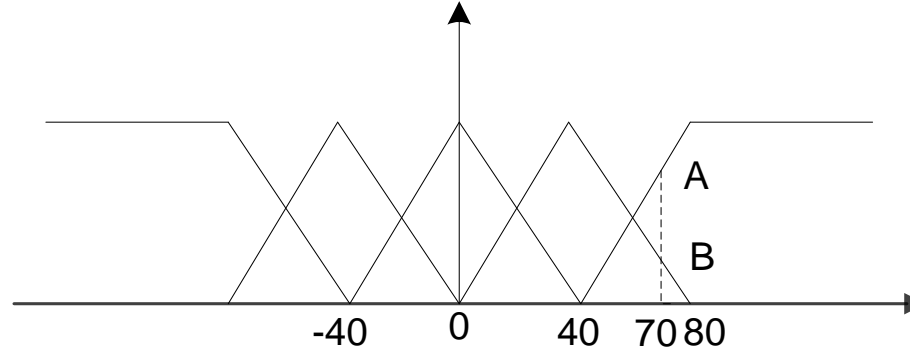
# Giriş

- Yukarıda açıklananlara karşıt olarak bulanık mantık, keskin mantığı açık /kapalı, soğuk/sıcak, hızlı/yavaş gibi ikili denetim değişkenlerinden oluşan keskin dünyayı, az açık/az kapalı, serin/ılık, biraz hızlı/biraz yavaş gibi gevşek denetleyicilerle yumuşatarak gerçek dünyamıza benzetir.



# Bulanık Mantık

- Bulanık mantıkta üyelik fonksiyonu adında bir kavram vardır.
- Üyelik işlevlerinde genel olarak küçük, orta ve büyük olmak üzere 3 veya küçük, orta derecede küçük, orta, orta derecede büyük ve büyük olmak üzere 5 veya 0 dolay ve sıfırın etrafında, pozitif ve negatif küçük, orta ve büyük olmak üzere 7 ayrı etiket tanımlanır.
- En yaygın olanı 7 etiket kullanılmasıdır.



(Burada 70 değerinin hangi kümeye ait olduğunu bulmaya bulanıklaştırma denir )



# Bulanık Mantık

- Bulanık mantıkta bir değişkeni ele aldığımızda bu değişkenleri üyelik fonksiyonuna dâhil ediyoruz.
- Bulanık mantıkla oluşturduğumuz kümeye “bulanık küme” adı verilir. Var-yok bilgileriyle oluşturulan kümeye “keskin küme” adı verilir.
- Üyelik fonksiyonu ile dilsel ifadeler matematiksel ifadeye dönüştürülmüş olunur.

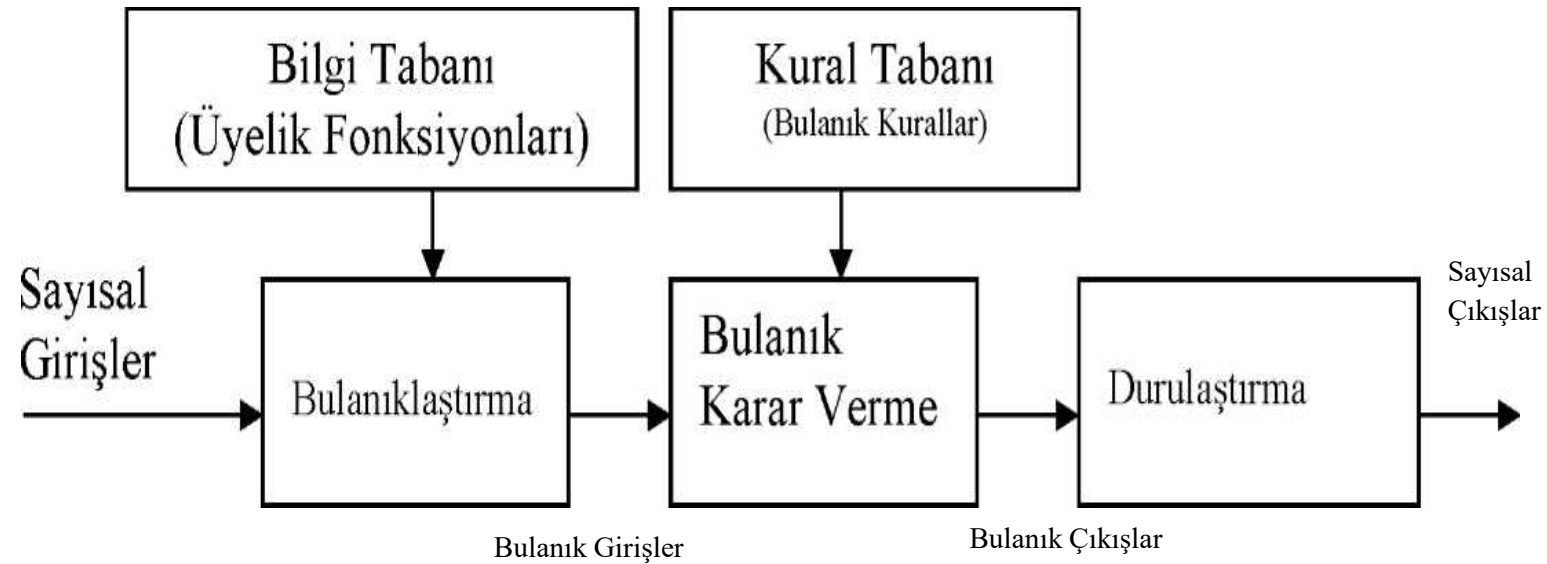
## Bir bulanık mantık denetleyicinin yapısı

- BULANIKLAŞTIRMA: Bu bölüm giriş değişkenlerini (gerçek sayıları)ölçer, onlar üzerinde birçok değişikliği yapar ve bulanık kümelere dönüştürür. Yani onlara birer etiket vererek dilsel nicelik kazandırır.
- BULANIK KARAR VERME: Burada kurallar üzerinde bulanık mantık yürütülür, yani insan beyninin düşünüş şeklinin bir benzetimi yapılmaya çalışılır.

## Bir bulanık mantık denetleyicinin yapısı

- BİLGİ TABANI: Çıkarım motoru, kural tabanında kullanılan bulanık kümelerin üyelik işlevlerini bu bölümden alır.
- KURAL TABANI: Denetim amaçlarına uygun dilsel denetim kuralları burada bulunur ve çıkarım motoruna buradan verilir.
- DURULAŞTIRMA: Çıkarım motorunun bulanık küme çıkışları(birden fazla çıkış olabilir) üzerinde gerekli ölçek değişikliklerini yapar ve bunları gerçek sayılara dönüştürür

# Bir bulanık mantık denetleyicinin yapısı



# MATLAB FUZZY LOGIC TOOLBOX KULLANIMI

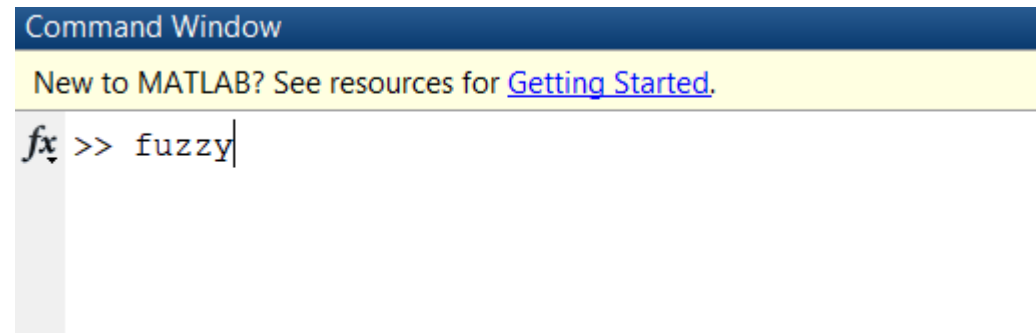
Prof. Dr. ERHAN AKIN

# Matlab Fuzzy Logic Toolboxını kullanma

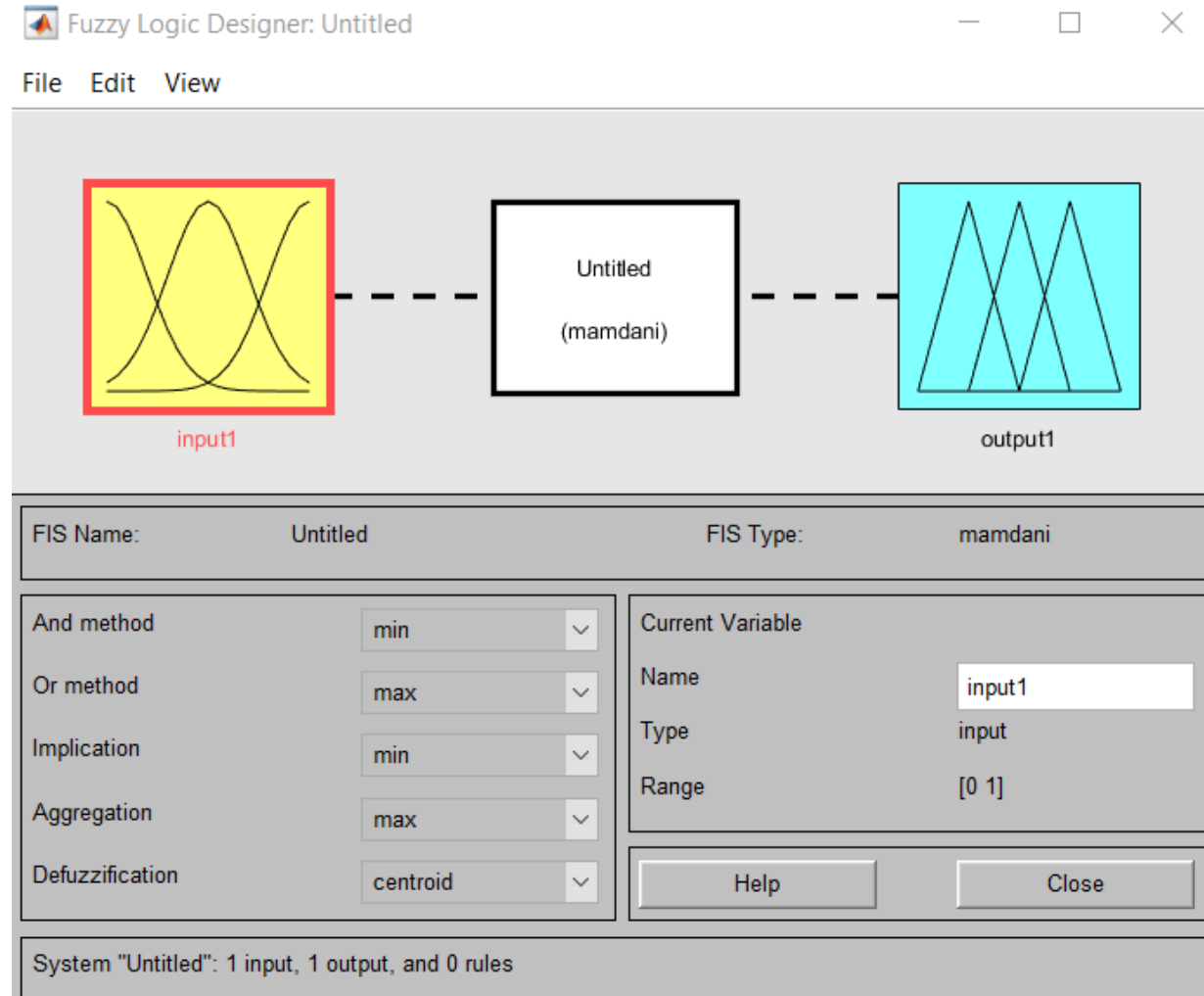
- ❑ Matlab ortamında bulanık mantık uygulaması yapabilmek ve bulanık mantık ile çözebileceğimiz problemlerin daha hızlı çözebilmek için imkan sunmaktadır.
- ❑ Çözülmesi zor problemlerin ve karmaşık olup matematiksel modeli oluşturulmayan problemlerin fuzzy logic toolbox kullanımı ile çözümlenmesi hem daha kolay hem de daha doğruya yakın sonuçlar üretmektedir.

# Matlab Fuzzy Logic Toolboxını kullanma

- ❑ Matlab bulanık mantık toolbox'ını kullanmak için Matlab programının açılarak Command Window bölümüne 'fuzzy' yazılarak enterlenmesi ile arayüz ekranına ulaşılacaktır.



# Matlab Fuzzy Logic Toolboxını kullanma



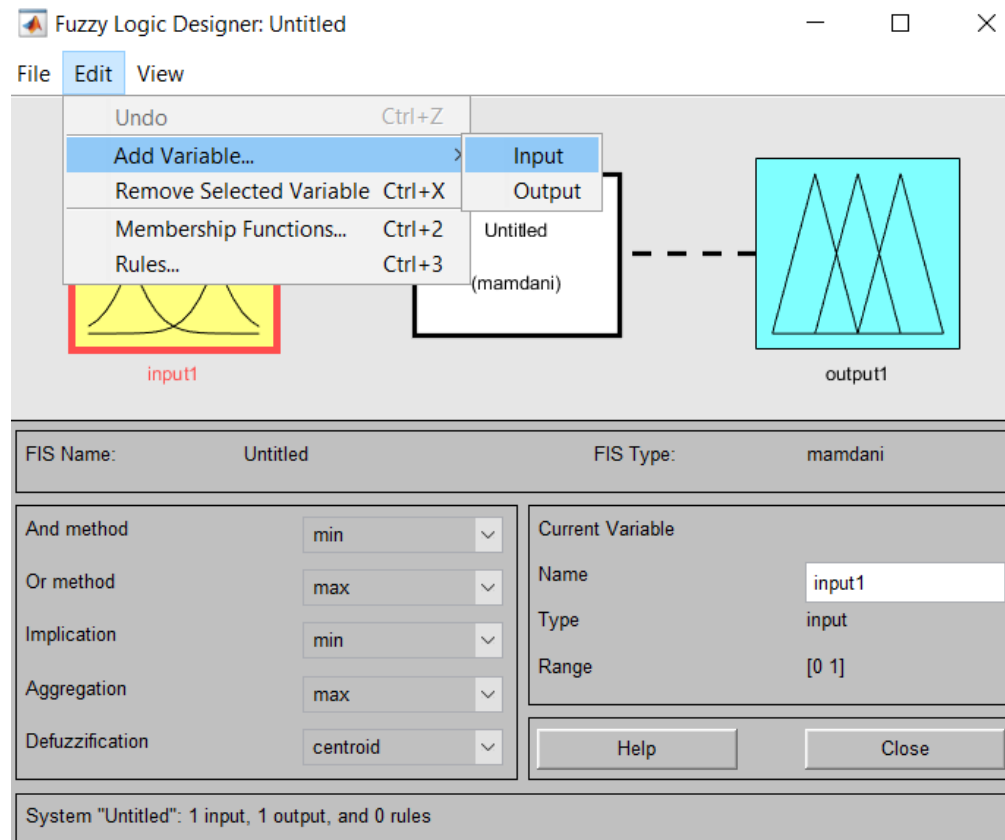


# Matlab Fuzzy Logic Toolboxını kullanma

- ❑ Arayüzde yapılacak işlemler aşağıdaki gibi sıralanmaktadır.
  - ❑ Öncelikle oluşturulacak sistemin giriş ve çıkış sayıları belirlenmelidir.
  - ❑ Giriş ve çıkışlar için başlangıç bitiş değerleri belirlenmelidir.
  - ❑ Giriş ve çıkış için bulanık üyelik fonksiyonları oluşturulmalıdır.
  - ❑ Ardından kullanılacak kural tabanı oluşturulmalıdır.
  - ❑ En son adım olarak sistemin test edilmesi; belirli giriş değerlerine karşılık üretilecek çıkış değeri kontrol edilmelidir.

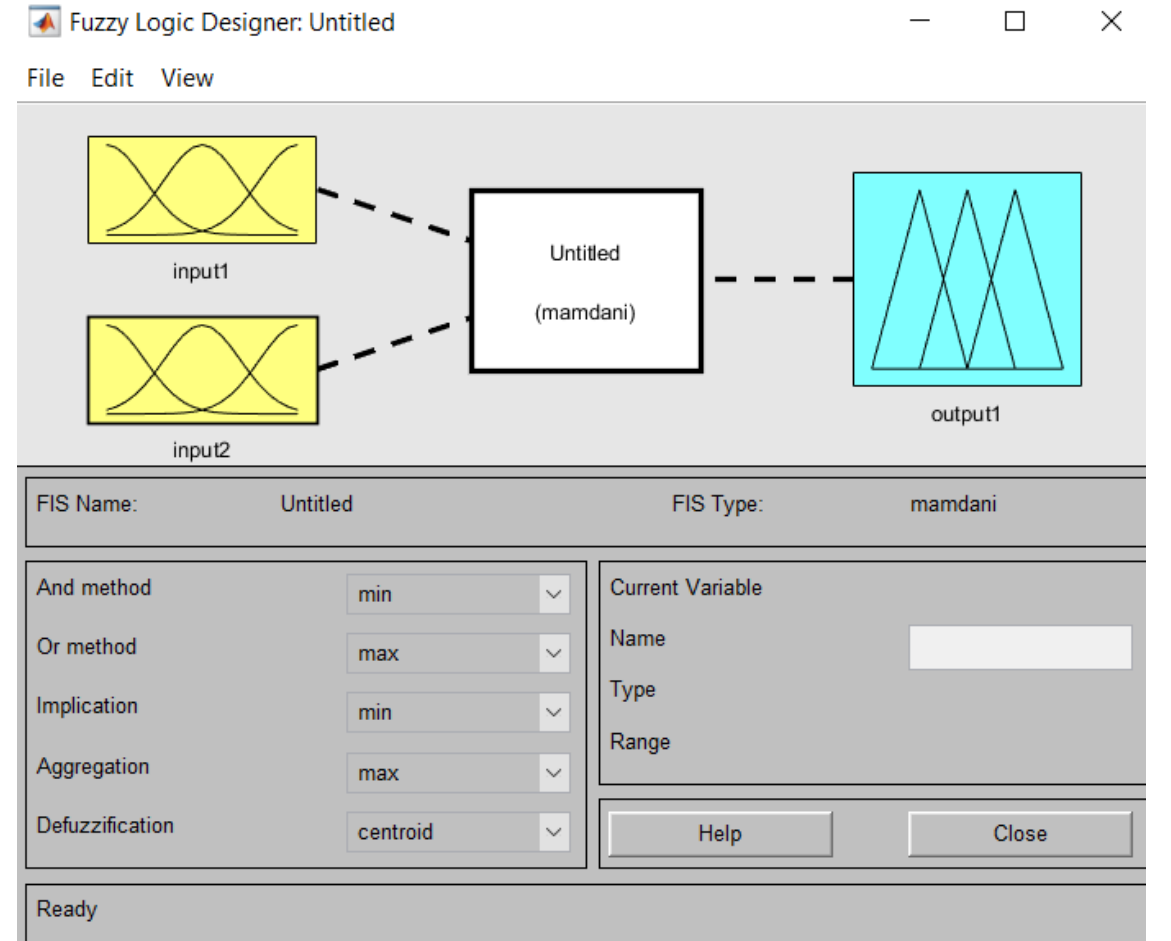
# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

❏ Edit > Add Variable > Input



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

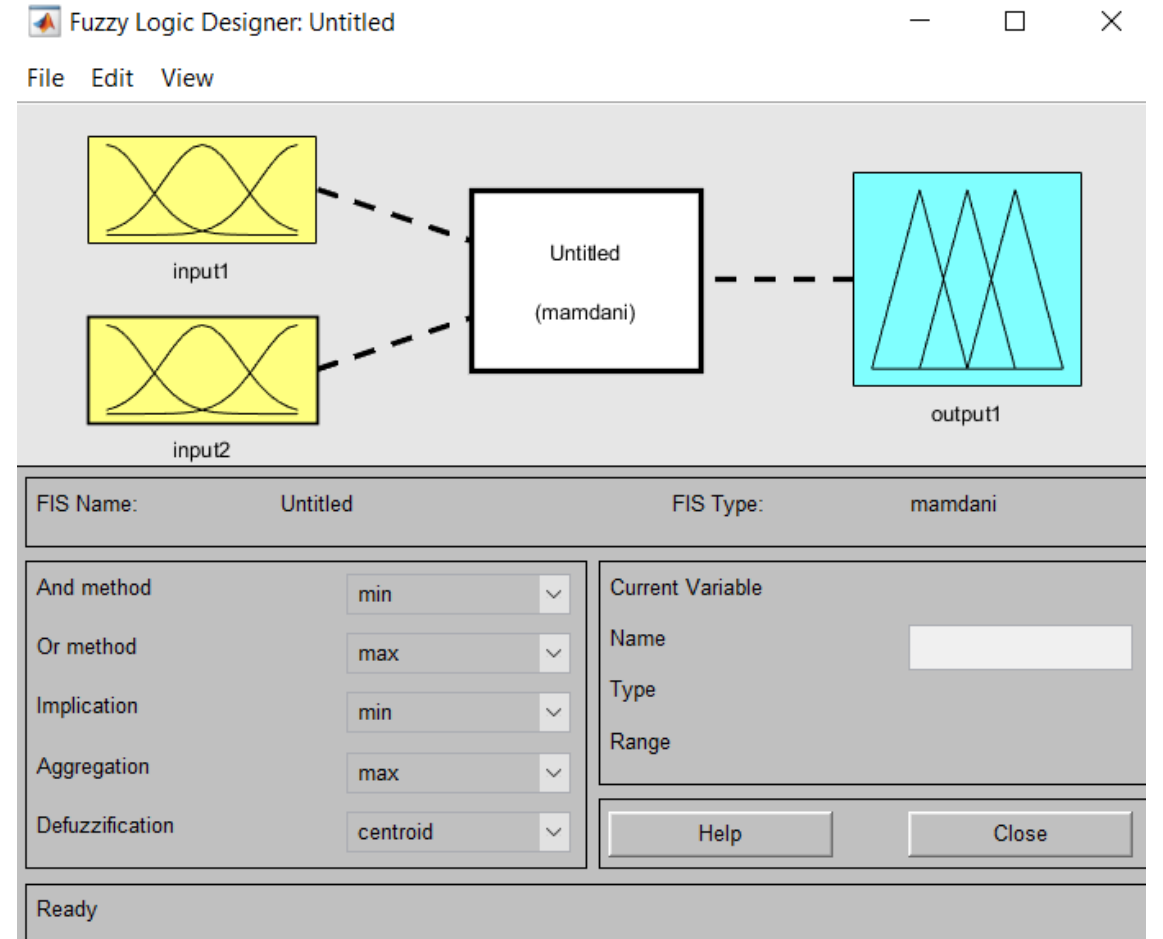
- ❑ Bu arayüzde input ve output sayısı arttırılıp azaltılabilir.
- ❑ Ekleme için Edit > Add Variable > Input ya da Output seçilebilir.
- ❑ Silme işlemi için ise silmek istediğimiz bileşene tıklayarak 'Del' tuşunu kullanabilir ya da Edit> Remove Selected Variable yolu kullanılabilir.



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

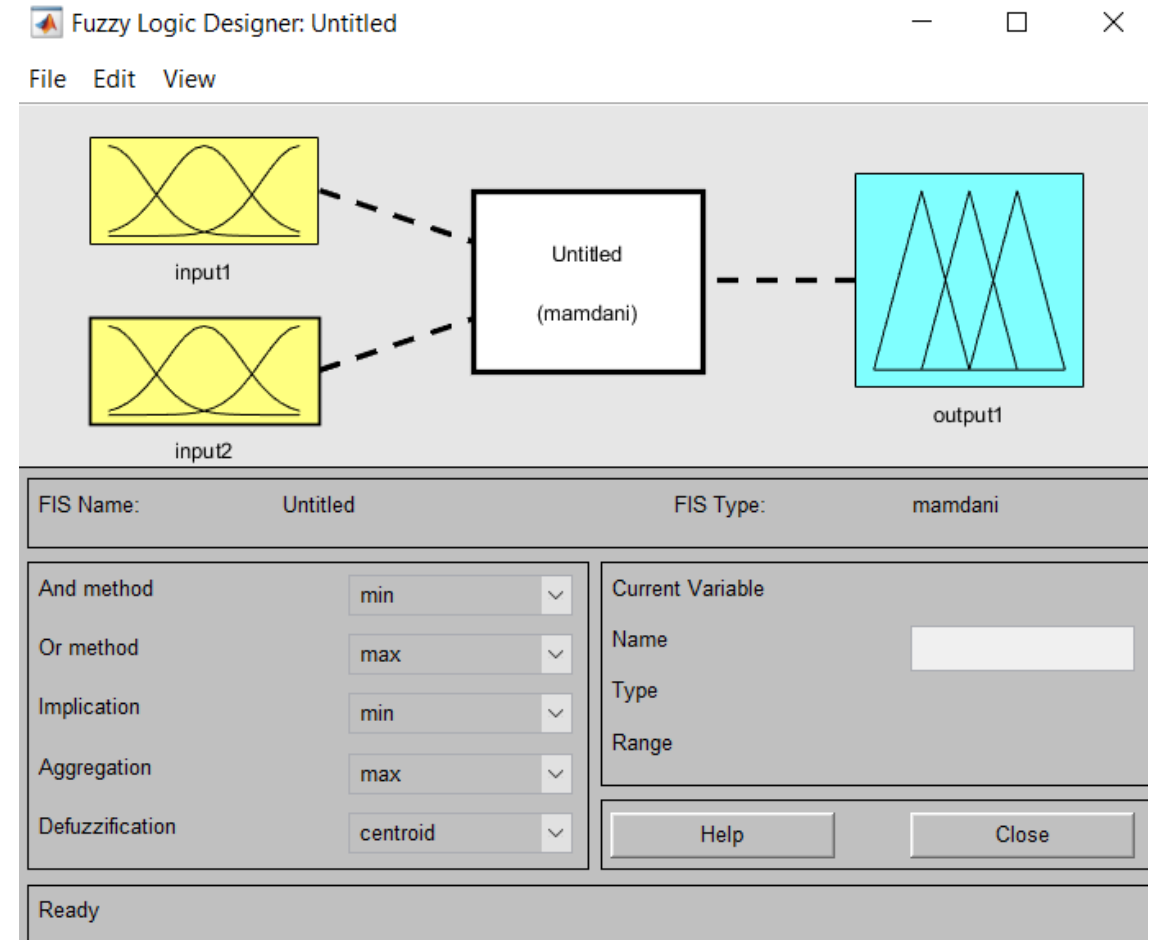
❑ Ayrıca bu arayüzde kural tabanında kullanılacak model olan MAMDANI veya SUGENO yöntemleri de seçilebilir.

❑ Biz bu ders için örnek olması açısından MAMDANI modelini kullanacağız.



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

- Çıkarım mekanizması için Min-Max yöntemi
- Durulaştırma adımı için ise Centroid metodunun kullanılması hedeflenmektedir.



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

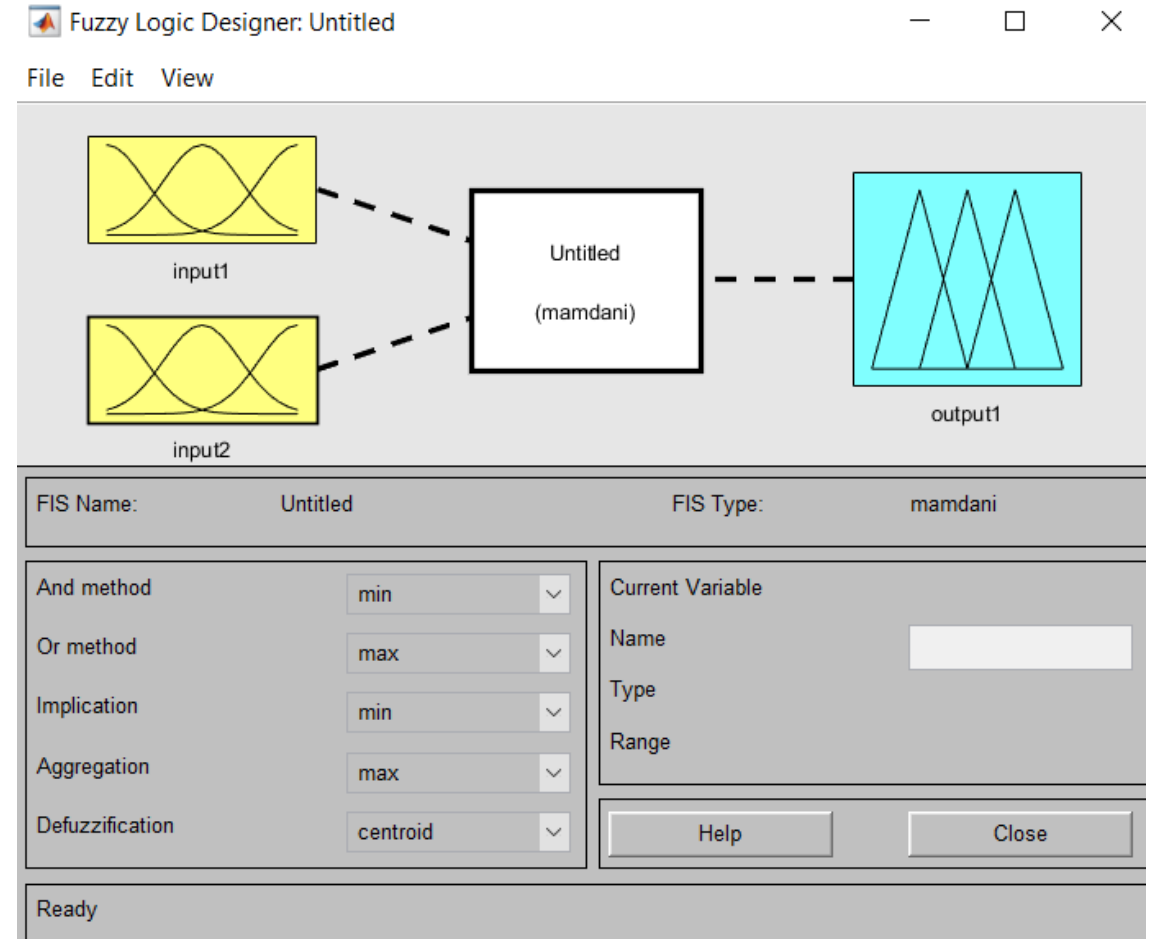
Yapacağımız örnekte;

Bir klimanın çalışma durumunu simüle eden uygulamada iki giriş ve bir çıkış kullanılması planlanmaktadır.

Girişler: Sıcaklık (-20, +50 derece)

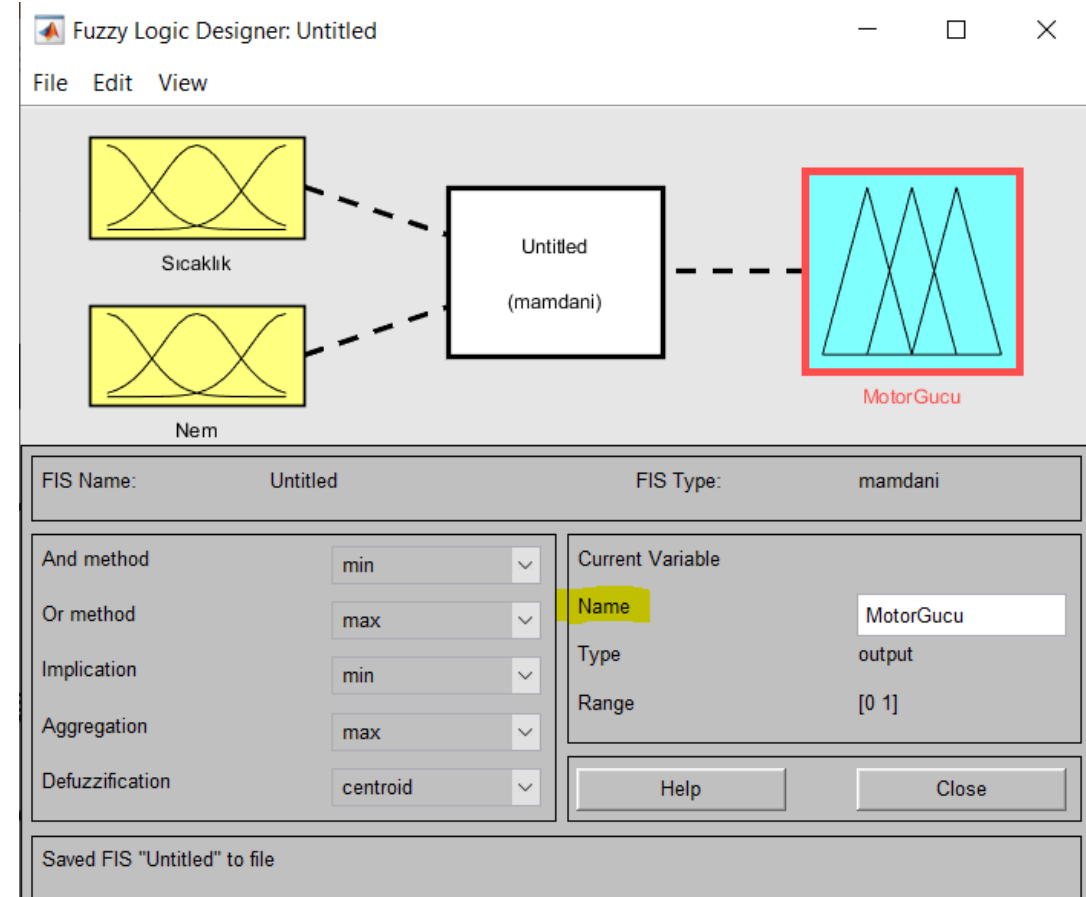
Nem (0, 100)

Çıkış : Klimanın motor gücü (-100,100)



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

❑ Giriş ve çıkış üyelik fonksiyonlarının isimlerini yeniden yanda gösterildiği gibi düzenlenebilmektedir.



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

## ❑ Sıcaklık fonksiyonu;

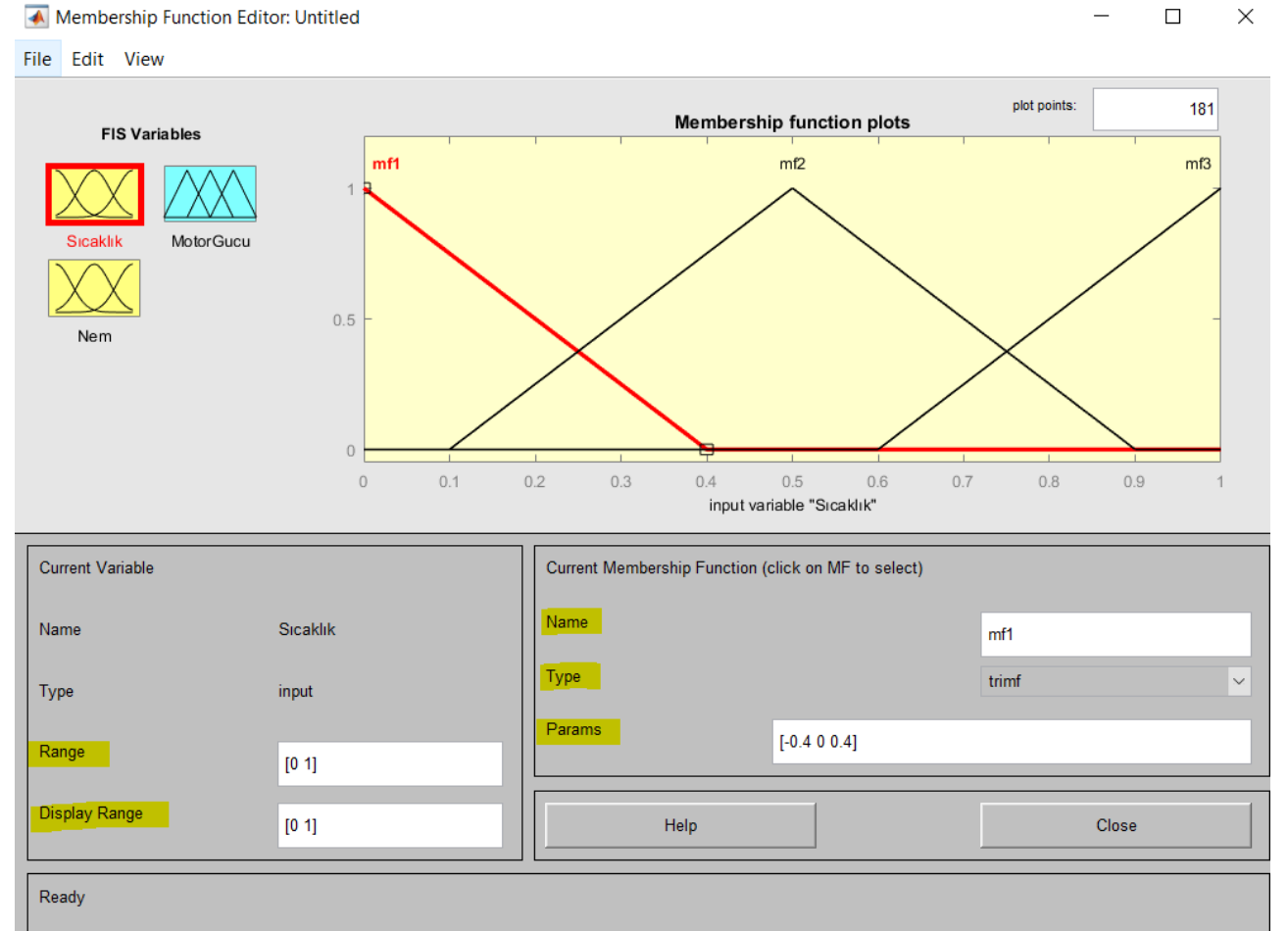
❑ Değer aralığı

❑ Gösterilen değer aralığı

❑ Üyelik fonksiyonun adı

❑ Üyelik Fonksiyon türü (Örn; üçgen)

❑ Ve Üyelik fonksiyonunun başlangıç bitiş değer aralıkları yazılmaktadır.





# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

- ❑ Üyelik fonksiyonunda fonksiyonun türü olarak birçok seçenek bulunmaktadır.
- ❑ Trimf fonksiyonu ile üçgen üyelik fonksiyonu belirlenmektedir.
- ❑ Üçgenin başlangıç tepe ve bitiş noktası Params bölümüyle ayarlanmaktadır.

Current Membership Function (click on MF to select)

Name	Cok
Type	trimf
Params	[50 100 100]

Help

trimf  
trapmf  
gbellmf  
gaussmf  
gauss2mf  
sigmf  
dsigmf  
psigmf  
pimf  
smf  
zmf

# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

## ■ Sıcaklık fonksiyonu;

■ Cso: Çok soğuk

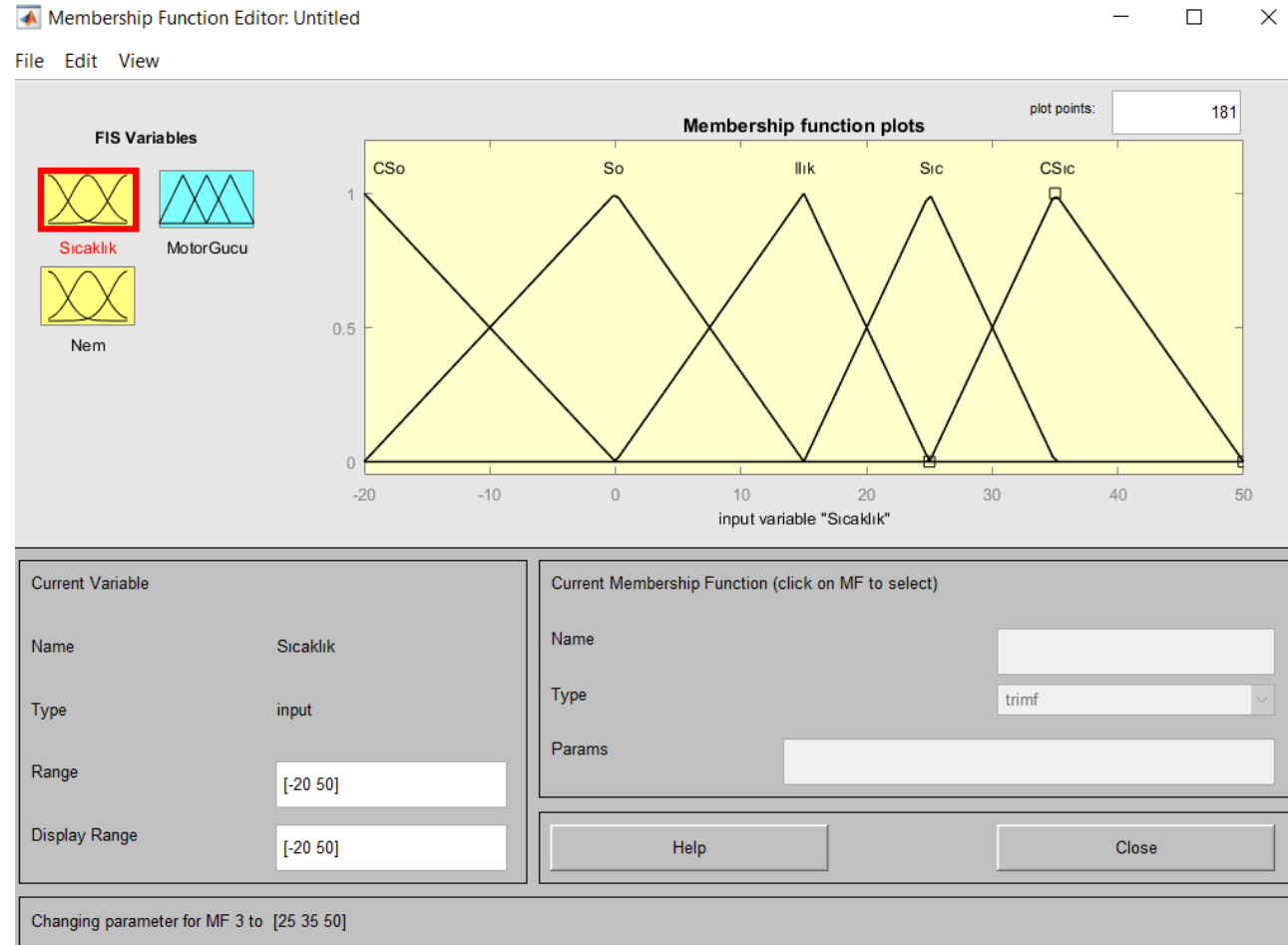
■ So: Soğuk

■ Ilık: ılık

■ Sıc: Sıcak

■ Csıc: Çok sıcak

olarak etiketlenmiştir.



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

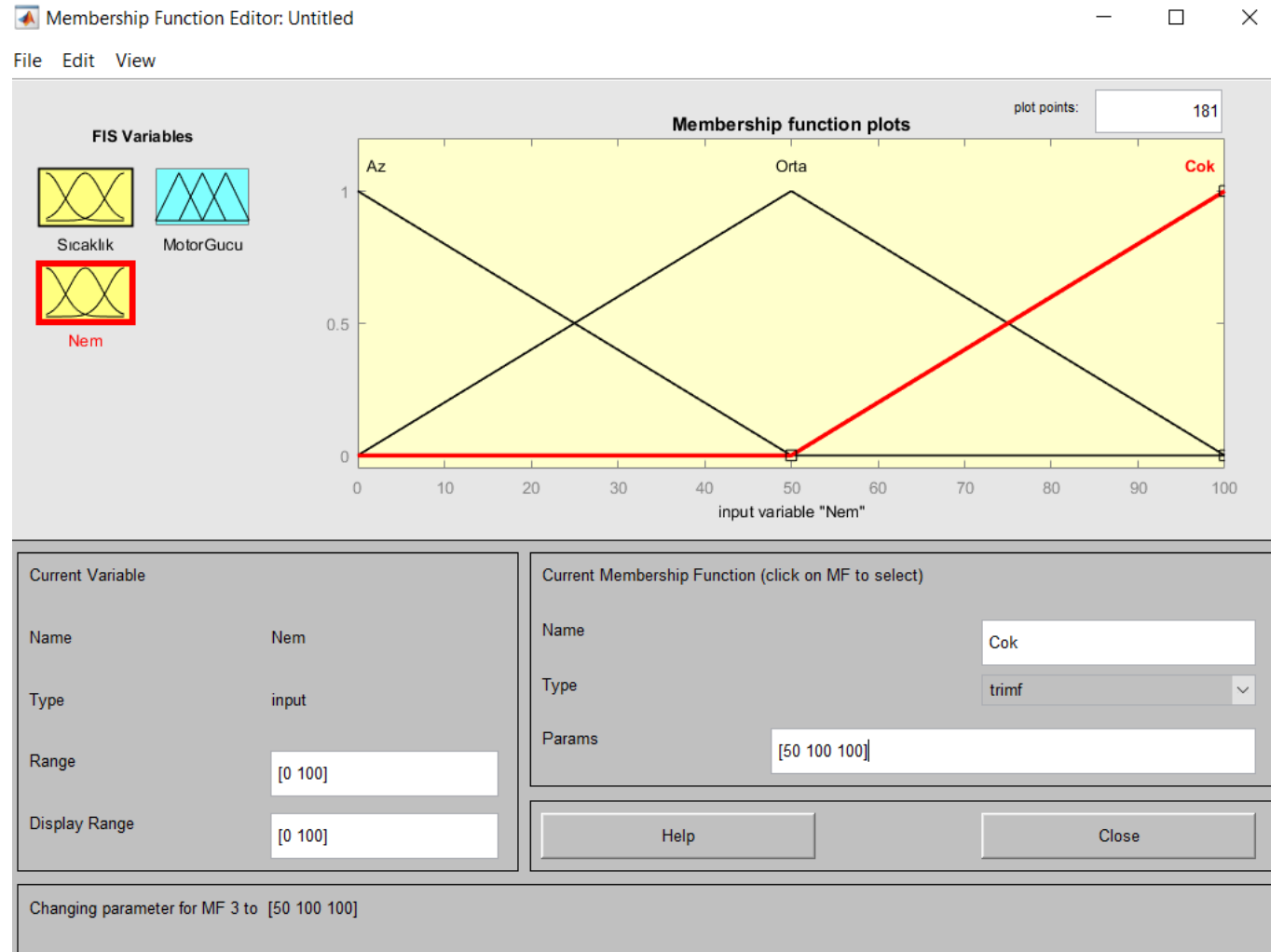
■ Nem fonksiyonu;

■ Az: az nem

■ Orta: orta nem

■ Cok: çok nem

olarak etiketlenmiştir.



# Bulanık giriş ve çıkış üyelik fonksiyonu oluşturma

☐ Klimanın motor gücü fonksiyonu;

☐ NB: negatif büyük

☐ NO: negatif orta

☐ NK: negatif küçük

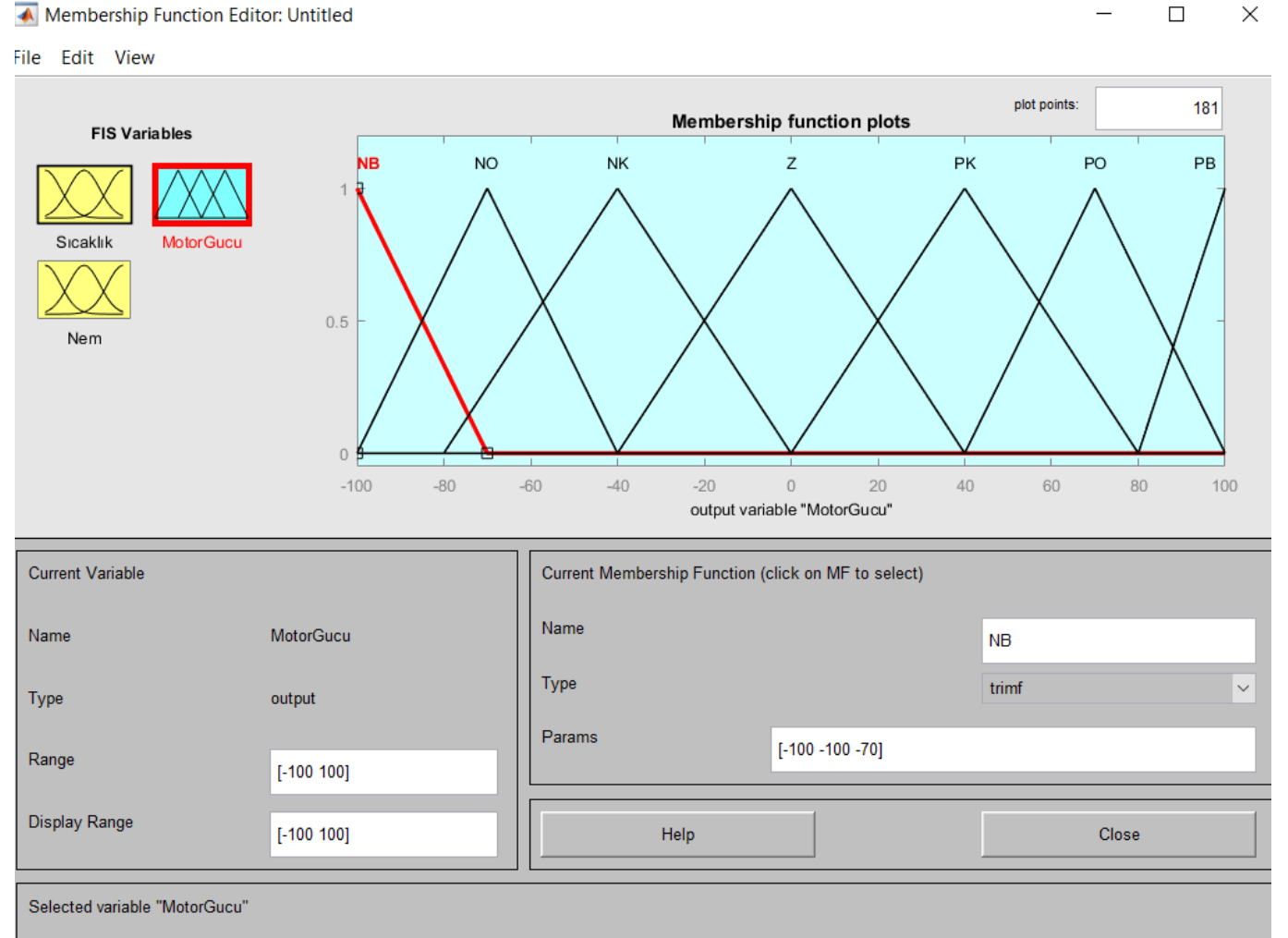
☐ Z: zero(sıfır)

☐ PK: pozitif küçük

☐ PO: pozitif orta

☐ PB: pozitif büyük

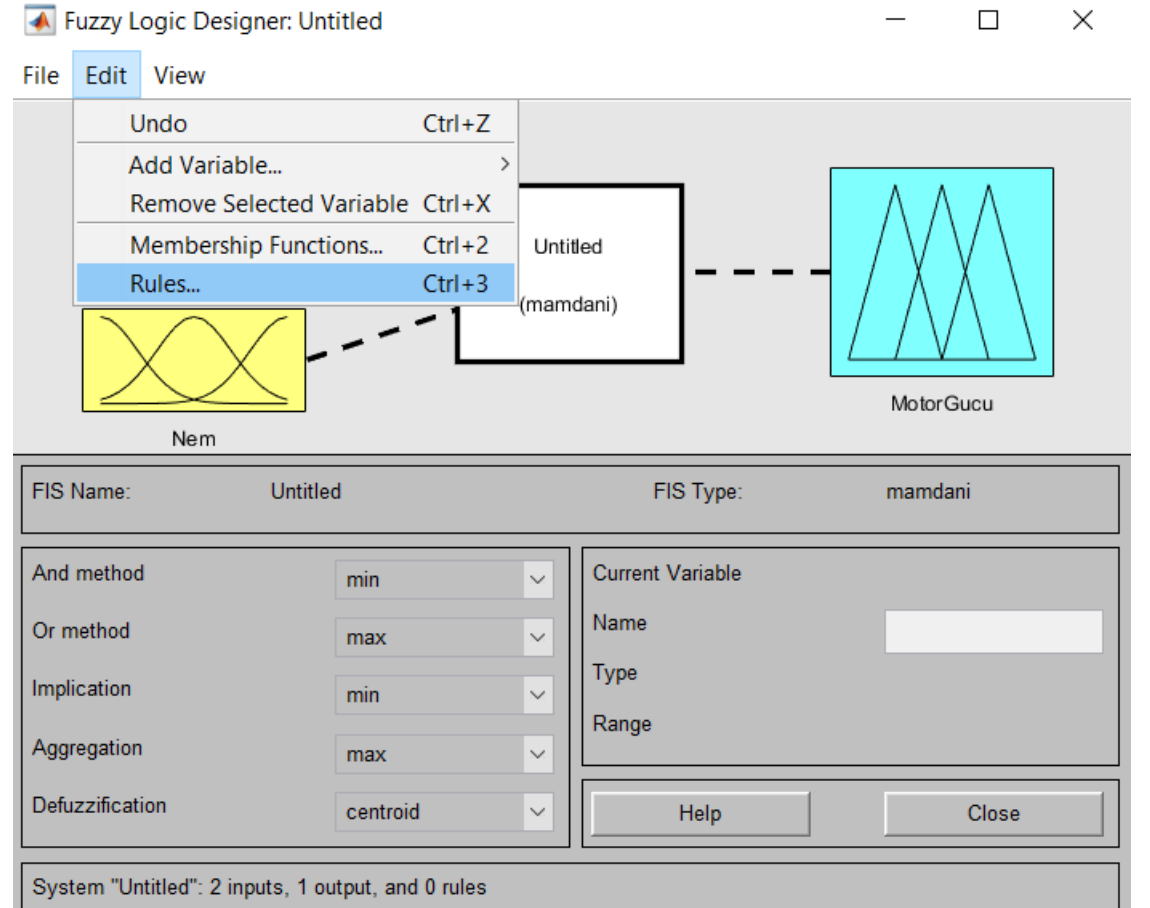
olarak etiketlenmiştir.



# Bulanık Kural tabanı oluřturma

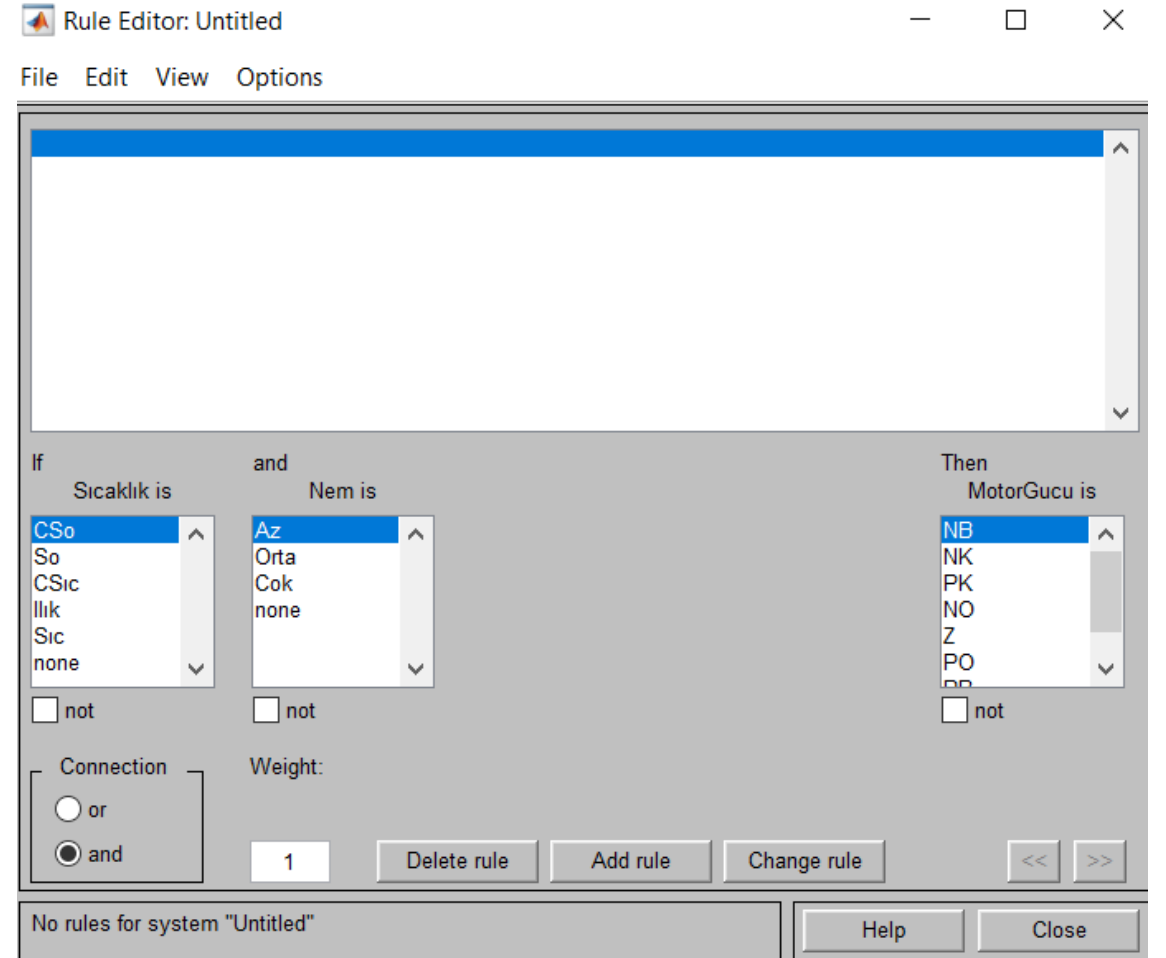
□ Klimanın alıřmasını modelleyen bu sistemde kural tabanı oluřturma adımıında;

□ Edit > Rules yolu izlenerek kural tabanı sekmesi aılmaktadır.



# Bulanık Kural tabanı oluřturma

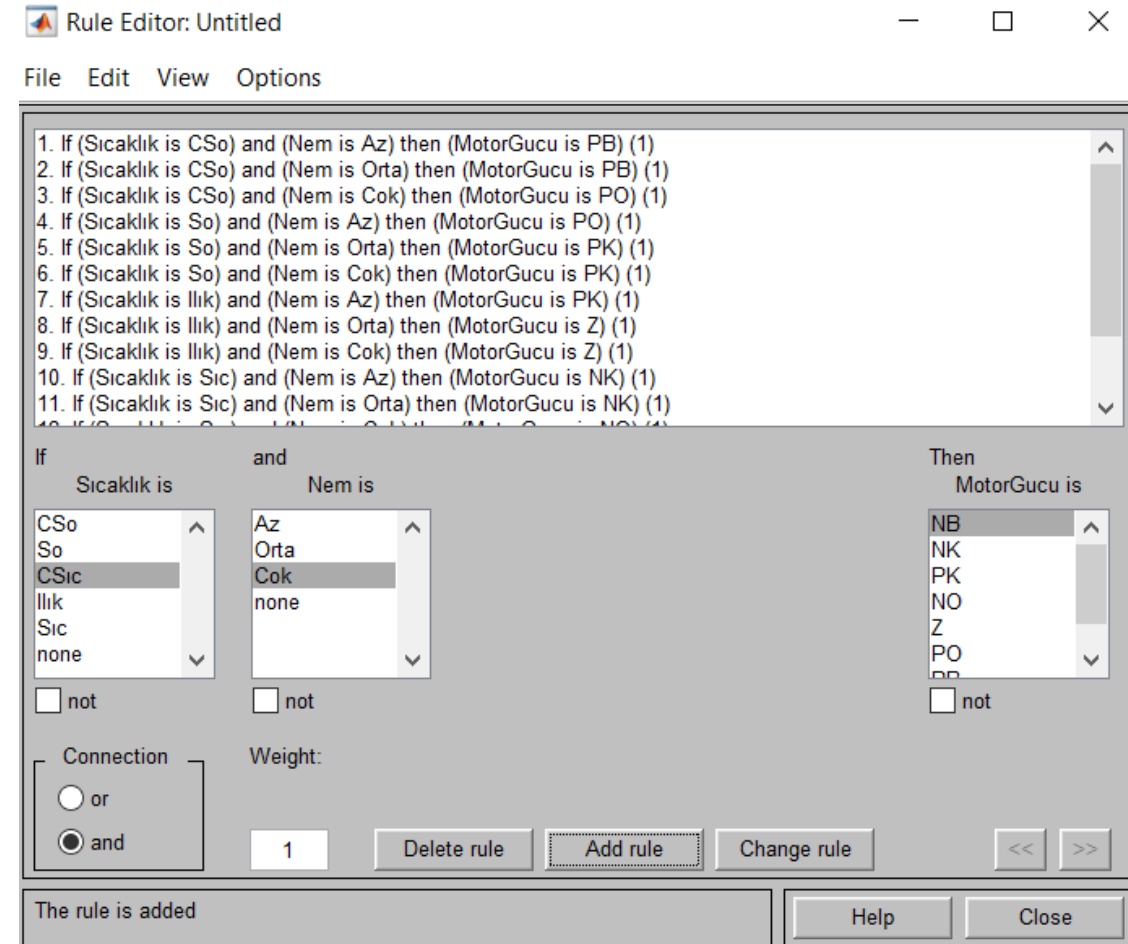
- Bulanık mantık mamdani modelinde kural tabanı; If- Then yaklařımıyla alıřmaktadır.
- Örn; ‘Sıcaklık Cso ve Nem Az ise Klimanın Motor gücü PB olacaktır’ gibi



# Bulanık Kural tabanı oluşturma

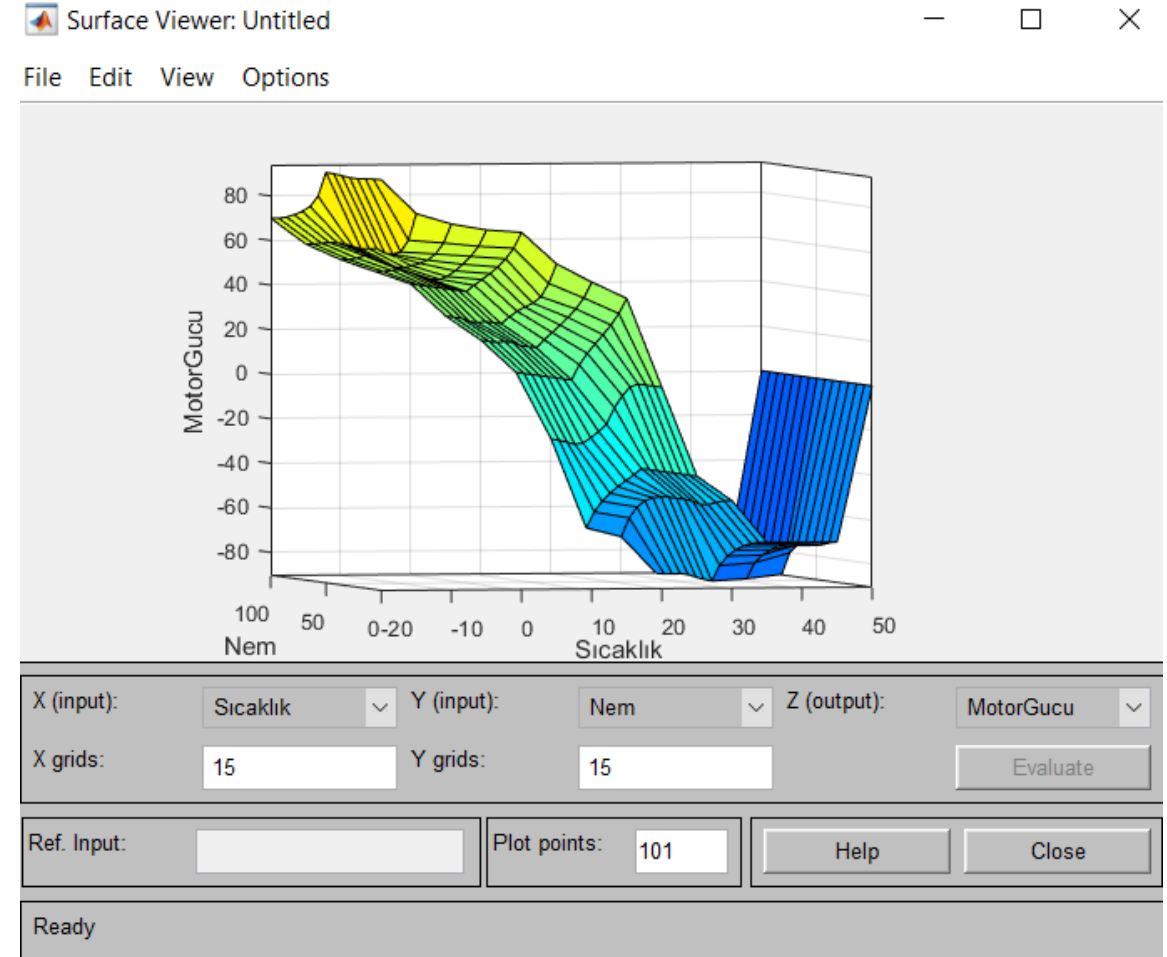
❑ Kural tabanı yan tarafta gösterildiği gibi oluşturulmuştur.

❑ Connection sekmesinde IF-THEN kuralındaki IF bölümünde hangi bağlantı şekliyle bağlanacağı seçilmektedir.



# Bulanık Kural tabanı oluřturma

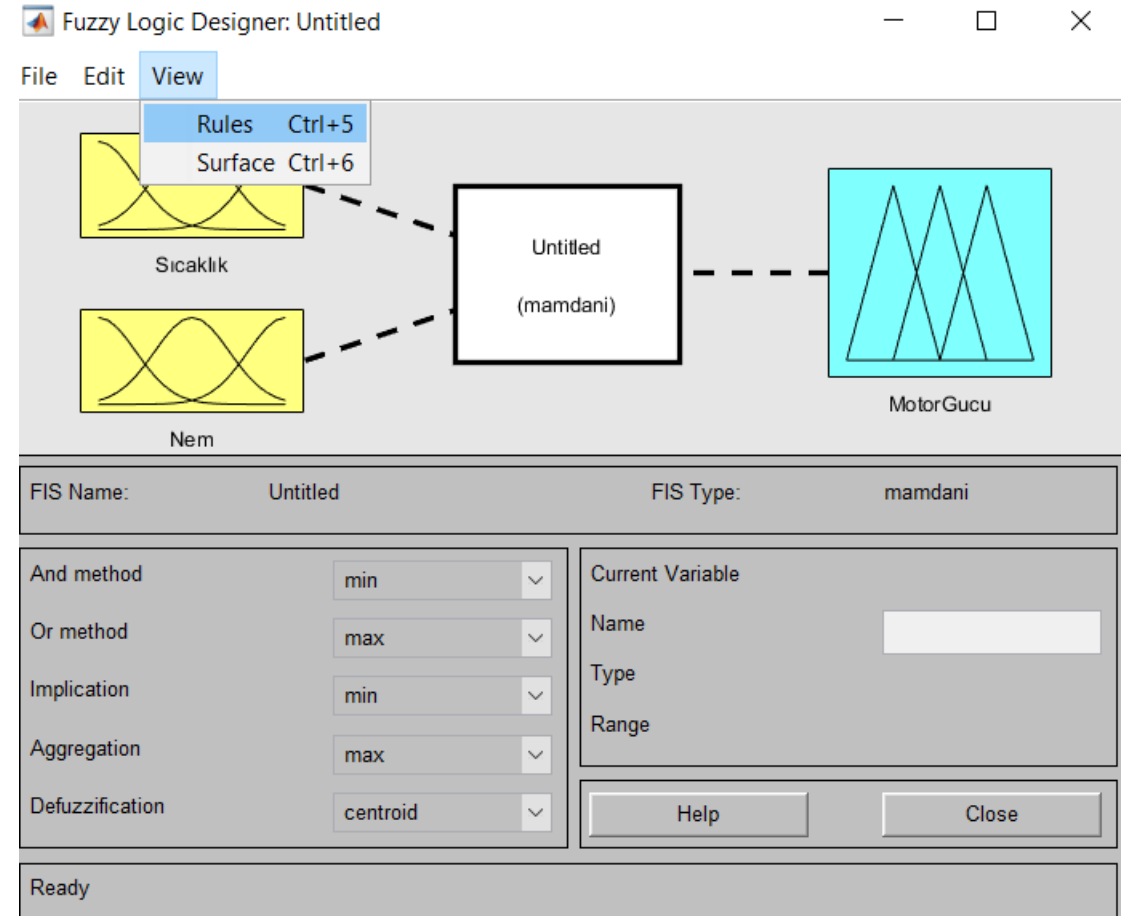
□ View>Surface sekmesi ile oluřturulan modele genel bir bakıř ağısı saęlanabilmektedir.





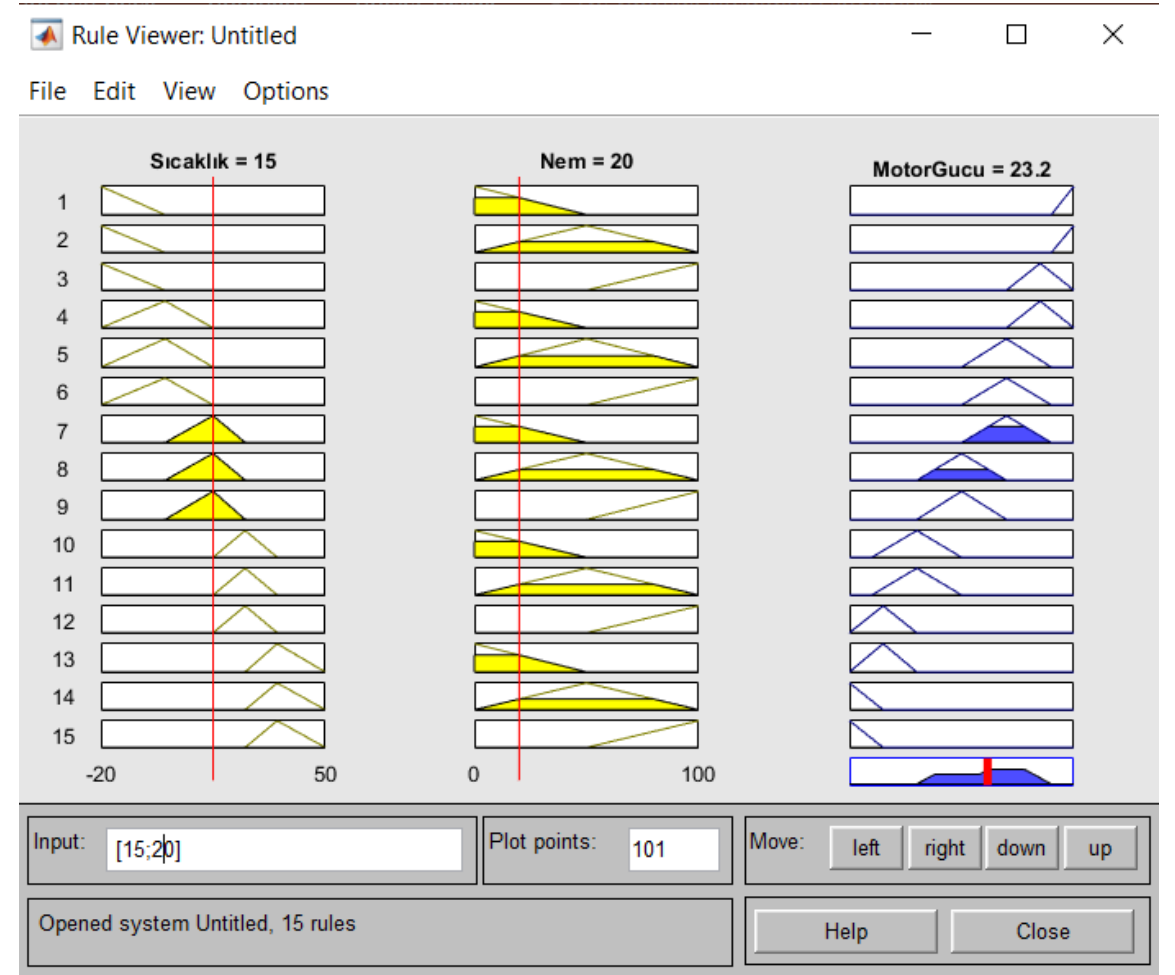
# Bulanık Mantık Test aşaması

❑ Geliştirilen modelin test edilmesi için View> Rules yolu izlenebilmektedir.



# Bulanık Mantık Test aşaması

- ❑ Sıcaklık: 15 ve Nem: 20 iken Klimanın motor gücü bulanık mantığa göre 23.2 olarak bulunmuştur.



# Bulanık Mantık Operatörleri

PROF.DR.ERHAN AKIN

# T-Norm ve S-Norm

- T-norm ve S-norm iki veya daha fazla sayıdaki bulanık kümenin birleştirilmesi işleminde kullanılan operatörlerdir.
- S- norm aynı zamanda T- conorm olarak da adlandırılmaktadır.
- T-normlar bulanık mantık için klasik mantık tarafından incelenen iki-değerli mantıksal birleşimin bir genellemesidir.

# Bulanık Kural Tabanlı Sistemler

- **Üçgen (Triangular) Normlar :**
- Üçgen normlar (t-norm) Schweizer ve Sklar tarafından öne sürülmüştür.
- Herhangi bir  $a \in [0,1]$  aralığı için t-norm  $T(a, 1) = a$  şeklinde tanımlanır ve aşağıdaki özellikleri sağlamaktadır.
  - **Simetri** :  $T(x, y) = T(y, x)$   $x, y \in [0, 1]$
  - **Birleşim** :  $T(x, T(y, z)) = T(T(x, y), z)$   $x, y, z \in [0, 1]$
  - **Monotonluk** :  $T(x, y) \leq T(x', y')$   $x \leq x'$  ve  $y \leq y'$
  - **Nötr Eleman 1**:  $T(x, 1) = x$   $x \in [0, 1]$

# Bulanık Kümelerin Kesişimi için Kullanılan T-normlar

- **1.Minimum** :  $\min(a, b) = \min\{a, b\}$
- **2.Lukasiewicz** :  $T(a, b) = \max(a + b - 1, 0)$
- **3.Product** :  $T(a, b) = a.b$
- **4.Weak** :  $T(a, b) = \text{if } \max(a, b) = 1 \text{ then } \min(a, b) \text{ else } 0$
- **5.Hamacher** :  $T(a, b) = a.b / (a+b-a.b) \quad \mu \in (0, 1)$
- **6.Dubois-Prade** :  $T(a, b) = a.b / \max\{a, b, \alpha\}$
- **7.Yager** :  $T(a, b) = 1 - \min\{1, p\sqrt[p]{(1-a)p+(1-b)p}\} \quad p > 0$

# Bulanık Kümelerin Kesişimi için Kullanılan T-normlar

- Bütün normlar argüman sayısına göre genişletilmektedir.
- Örnek:
- $\min(a_1, a_2, \dots, a_n) = \min\{a_1, a_2, \dots, a_n\}$

# Bulanık Kural Tabanlı Sistemler

- **Üçgen (Triangular) Conormlar :**
- Üçgen conormlar (t-conorm veya s-norm), herhangi bir  $a \in [0,1]$  aralığı için s-norm  $S(a, 0) = a$  şeklinde tanımlanır ve aşağıdaki özellikleri sağlar;
  - **Simetri** :  $S(x, y) = S(y, x)$   $x, y \in [0, 1]$
  - **Birleşim** :  $S(x, S(y, z)) = S(S(x, y), z)$   $x, y, z \in [0, 1]$
  - **Monotonluk** :  $S(x, y) \leq S(x', y')$   $x \leq x'$  ve  $y \leq y'$
  - **Nötr Eleman 0**:  $S(x, 0) = x$   $x \in [0, 1]$



# Bulanık Kümelerin Birleşimi için Kullanılan T-conormlar

1. **Maksimum** :  $\max(a, b) = \max\{a, b\}$
2. **Lukasiewicz** :  $S(a, b) = \min(a + b, 1)$
3. **Probabilistic** :  $S(a, b) = a + b - a.b$
4. **Strong** :  $S(a, b) = \text{if } \min(a, b) = 0 \text{ then } \max(a, b) \text{ else } 1$
5. **Hamacher** :  $S(a, b) = (a+b-2.a.b/(1-a.b))$
6. **Yager** :  $S(a, b) = \min\{1, p\sqrt[p]{a+b}\}$   $p > 0$

# Bulanık Kümelerin Birleşimi için Kullanılan T-conormlar

- Bütün conormlar argüman sayısına göre genişletilmektedir.
- Örnek:
- $\max(a_1, a_2, \dots, a_n) = \max\{a_1, a_2, \dots, a_n\}$

# Bulanık Kural Tabanlı Sistemler

- Örnek : t-norm
- A ve B evrensel küme X'te tanımlanmış iki bulanık alt küme
- olsun.
- $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$
- $A = 0.0/x_1 + 0.3/x_2 + 0.6/x_3 + 1.0/x_4 + 0.6/x_5 + 0.3/x_6 + 0.0/x_7$
- $B = 0.1/x_1 + 0.3/x_2 + 0.9/x_3 + 1.0/x_4 + 1.0/x_5 + 0.3/x_6 + 0.2/x_7$
- $T(x, y) = \max\{x+y-1, 0\}$  Lukasiewicz t-norm'a göre;

# Bulanık Kural Tabanlı Sistemler

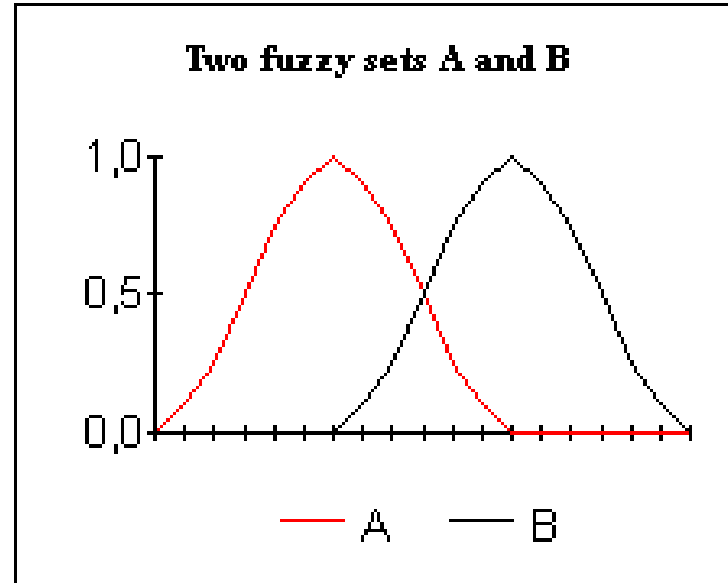
- $(A \cap B)(t) = \max\{A(t)+B(t)-1, 0\}$   $t \in X$  aşağıdaki gibi elde edilir;
- $A \cap B = 0.0/x_1 + 0.0/x_2 + 0.5/x_3 + 1.0/x_4 + 0.6/x_5 + 0.0/x_6 + 0.0/x_7$

# Bulanık Kural Tabanlı Sistemler

- Örnek : t-conorm
- $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$
- $A = 0.0/x_1 + 0.3/x_2 + 0.6/x_3 + 1.0/x_4 + 0.6/x_5 + 0.3/x_6 + 0.0/x_7$
- $B = 0.1/x_1 + 0.3/x_2 + 0.9/x_3 + 1.0/x_4 + 1.0/x_5 + 0.3/x_6 + 0.2/x_7$
- $S(x, y) = \min\{x+y, 1\}$  Lukasiewicz t-conorm'a göre
- $(A \cup B)(t) = \min\{A(t)+B(t), 1\}$   $t \in X$  aşağıdaki gibi elde edilir;
- $A \cup B = 0.1/x_1 + 0.6/x_2 + 1.0/x_3 + 1.0/x_4 + 1.0/x_5 + 0.6/x_6 + 0.2/x_7$

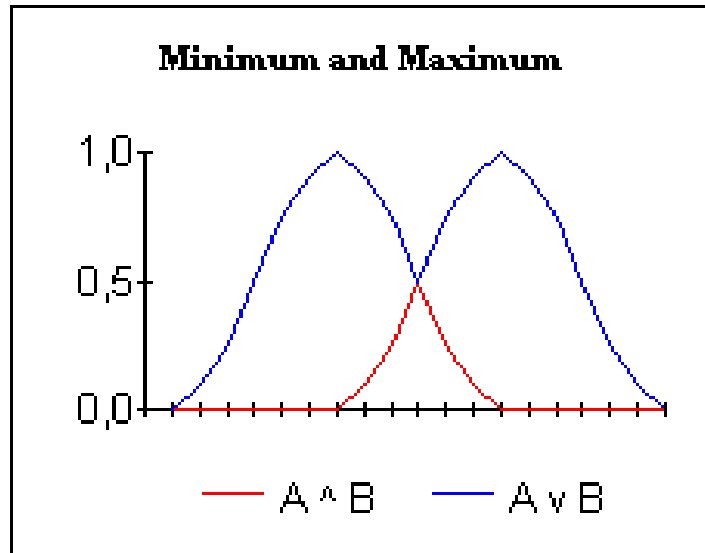
# Bulanık Kural Tabanlı Sistemler

- Örnek : iki bulanık küme



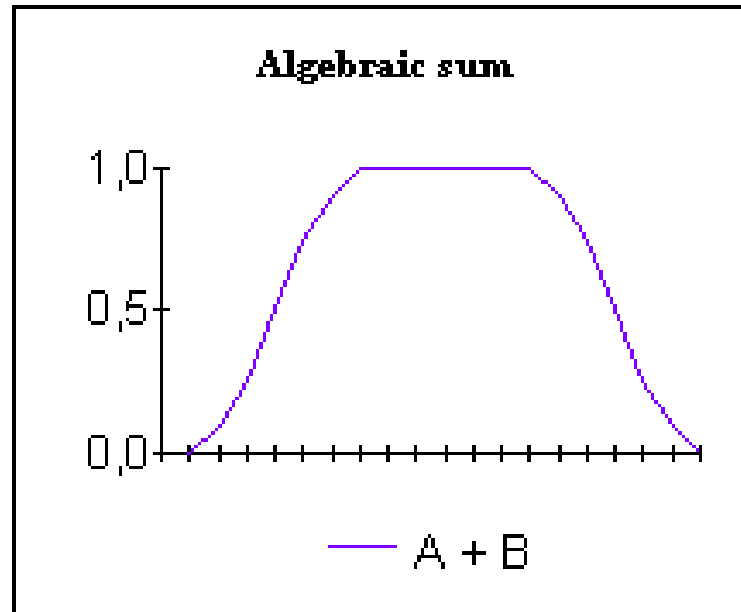
# Bulanık Kural Tabanlı Sistemler

- Örnek : iki bulanık küme
- T-norm ve S-norm kullanarak birleştirme ve kesişme işlemlerini yapınız.



# Bulanık Kural Tabanlı Sistemler

- Örnek : iki bulanık küme
- A ve B nin cebirsel birleşimi:





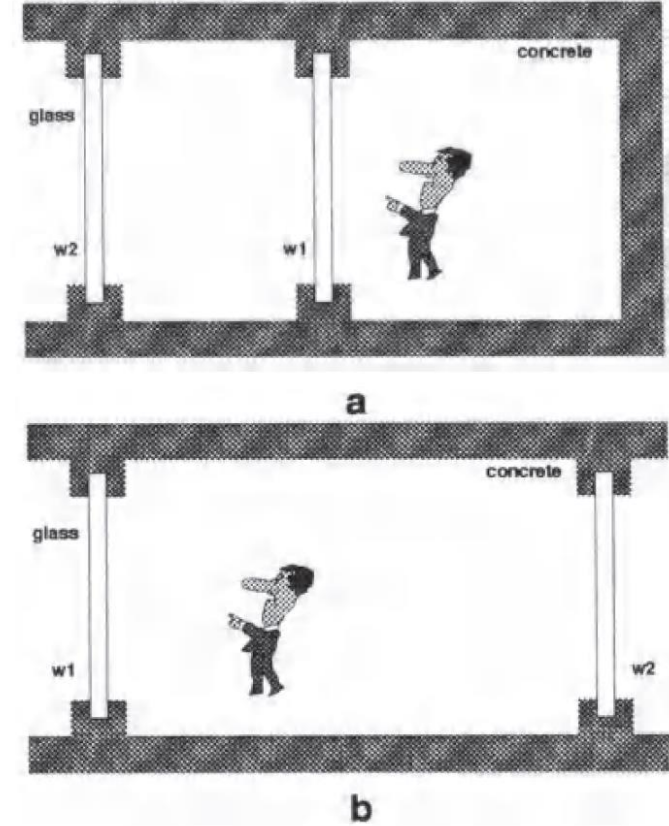
# Bulanık Kural Tabanlı Sistemler

- T-norm ve S-norm mantıksal eşleniklerdir ve birbirinden aşağıdaki gibi hesaplanır:

$$\mathbf{t(A,B) = 1 - s(1-A, 1-B)}$$

# Örnek

- Bu örnekte resimde gösterilen kişinin hapisane hücresinden kaçmasının ve güneş ışığının hapisane hücresine girmesinin ne kadar kolay olduğu incelenmektedir.
- Güneş ışığının içeri girdiği pencereden (W1) kişinin kaçmasının zor olduğu (0.1 oranında kolay), W2 penceresinden kaçmanın daha kolay olduğu (0.3 oranında kolay) ve
- W1 penceresinin kirli ve kalın olduğu böylece güneş ışığının W2 penceresi temizken içeri girmesinin o kadar kolay olmayacağını varsayalım.



İki farklı hapisane hücresi. Durum (a) 've' işlemini, durum (b) 'veya' işlemini temsil eder.

# Örnek

- İlk cevaptaki soru asıl  $(0.1, 0.3) = 0.1$  olacaktır, buradaki gerçeği dikkate almayız. Kişinin hızlı olması ya da yorulması gerektiğinden, sonuç yani kişinin hapishaneden kaçma ihtimali muhtemelen 0.1'den az olacaktır.
- W1 ve W2 pencereleri ve bir kişinin bu pencerelerden kaçması ve güneş ışığının pencerelerden içeri girmesi için kolaylık dereceleri aşağıdaki gibidir.

Pencere	Kaçış	Güneş ışığının içeri girmesi
W1	0.1	0.6
W2	0.3	0.9

# Bulanık Mantık

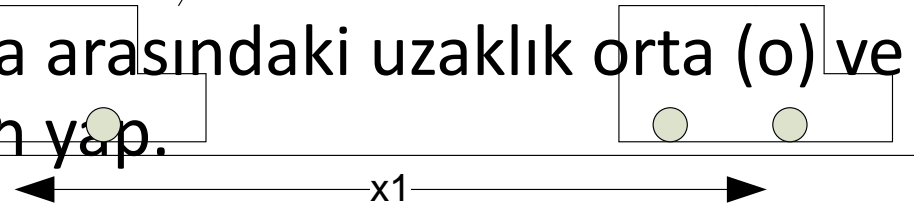
PROF.DR.ERHAN AKIN

# Örnek 1:

- Bir arabayla bir başka arabayı izlemek isteyelim şekilde görüldüğü gibi öndeki arabayla sizin aranızdaki mesafe  $x_1$  sizin hızınız  $x_2$  olsun. İlk olarak yıllardır araba kullanıyor olmamızla elde ettiğimiz deneyimlerimizi ve ustalığımızı birer dilsel kurallar şeklinde açıklayalım.

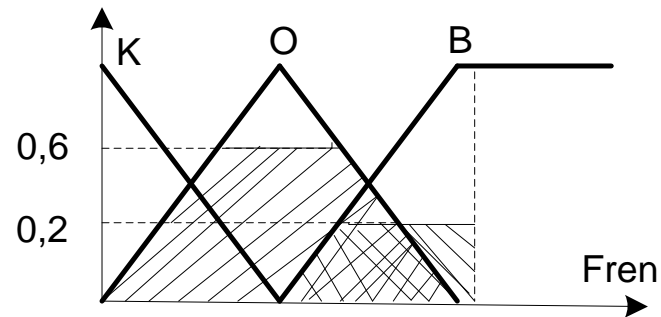
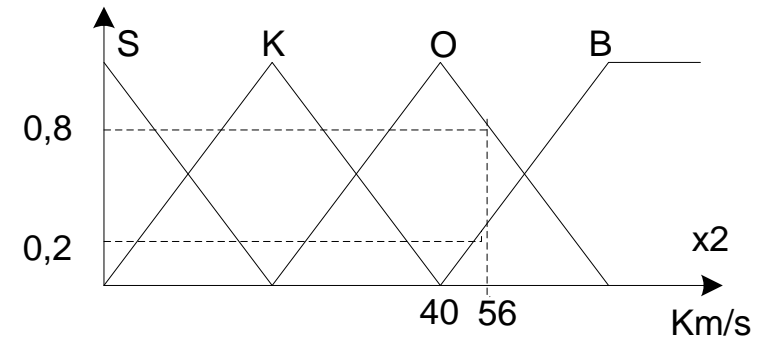
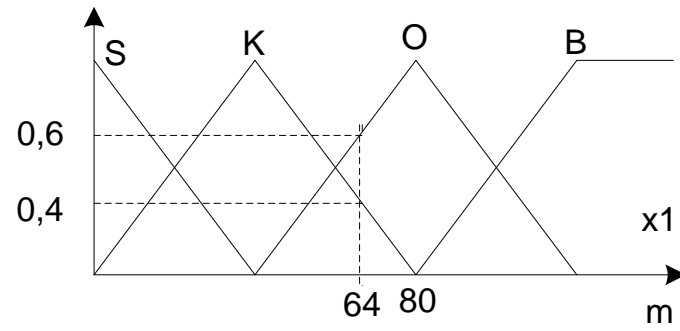
**Kural 1:** Eğer 2 araba arasındaki uzaklık az (küçük-K) ve hızınız orta ise (O) sert (büyük-B) fren yap. <sup>Hız  $x_2$</sup>

**Kural 2:** Eğer 2 araba arasındaki uzaklık orta (o) ve hızınız yüksek ise (B) orta sertlikte fren yap.



# Örnek 1:

- Üyelik fonksiyonunu oluşturalım:



# Örnek 1:

**K1:**Eğer  $x_1=K$  ve  $x_2=O$  ise o halde  $Y=B$

**K2:**Eğer  $x_1=0$  ve  $x_2=B$  ise o halde  $Y=O$



Üyelik fonksiyonu: Dilsel ifadelerin matematiksel karşılığı.  
Örneğimizde  $x_1=64$  m ve  $x_2=56$  km/s olarak verilmiş.

**Kural 1 için:**

**K1:**Eğer  $x_1=0,4$  ve  $x_2=0,8$  ise  $Y=B$

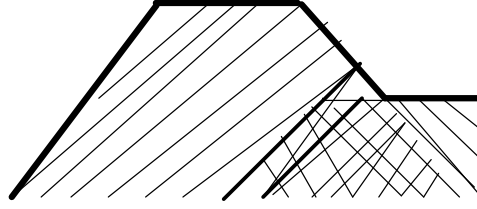
Frenin üyelik fonksiyonunda 0,2 değerine karşılık gelen B durumunu elde ederiz.

**Kural 2 için:**

**K2:** Eğer  $x_1=0,6$  ve  $x_2=0,2$  ise  $Y=O$

# Örnek 1:

- Frenin üyelik fonksiyonunda 0,6 değeriine karşılık gelen O durumunu elde ederiz.
- Kural 1 ve kural 2'yi uyguladıktan sonra fren durumu şu şekilde olur:



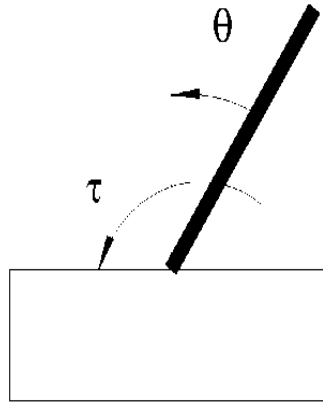


# Örnek 2: Klasik Ters Sarkaç Problemi

- Klasik ters sarkaç sistem görülmektedir. Uzun yıllardır kontrol teorileri için ilginç bir araştırma konusu olmuştur. Sistemin diferansiyel denklemi şu şekilde ifade edilir;

$$-ml^2 d^2\theta / dt^2 + (mgl) \sin(\theta) = \tau = u(t)$$

- Burada  $m$  ters sarkaç üs uzunluğu,  $\theta$  saat yönünde ( $u(t)$  = kontrol hareketi),  $t$



$m$  ağırlık kütlesi,  $l$  ters sarkaç ,  $\tau = u(t)$  dönme momenti imi sabit değeridir.

# Örnek 2: Klasik Ters Sarkaç Problemi

- Durum değişkenleri  $x_1$  ve  $x_2$  'yi  $x_1 = \theta$  ve  $x_2 = d\theta / dt$  kabul edilirse kesikli zamanlı (discrete-time) denklem olarak aşağıdaki ifadeler yazılabilir,

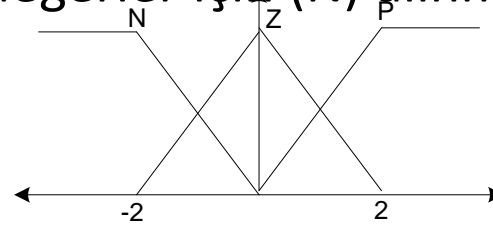
$$x_1(k+1) = x_1(k) + x_2(k) \quad (2)$$

$$x_2(k+1) = x_2(k) - u(k)$$

- Bu problemde bulanık denetleyici giriş değişken olarak alınan  $x_1$  ve  $x_2$  'nin değişim aralıkları,
- $-2^\circ \leq x_1 \leq 2^\circ$  ve  $-5 \text{ dps} \leq x_2 \leq 5 \text{ dps}$  ( $\text{dps}$  = her saniye için derece) olarak belirlenebilir.

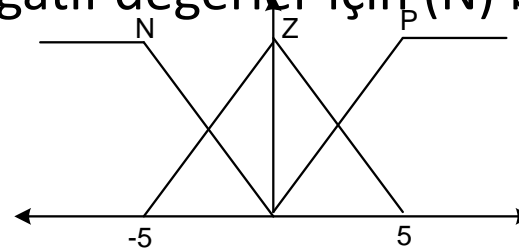
# Adım 1

- Önce  $x_1$  için üç üyelik fonksiyonu oluşturulabilir. Şekilde görüldüğü gibi pozitif değerler için (P), sıfır (Z) ve negatif değerler için (N) alınmıştır.



( $x_1$  girişı için üyelik fonksiyonu)

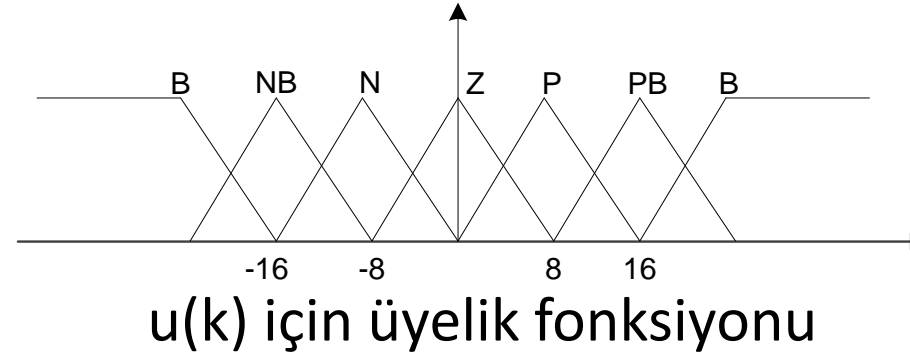
- Daha sonra  $x_2$  için üyelik fonksiyonu oluşturularak şekil de görüldüğü gibi, pozitif değerler için (P), sıfır için (Z) ve negatif değerler için (N) bulanık değerleri verilmiştir.



( $x_2$  girişı için üyelik fonksiyonu)

## Adım 2

- Kontrol aralığı 5 üyelik fonksiyonuyla şekilde görüldüğü gibi oluşturulabilir.  $u(k)$  değerinin değişim aralığı  $-24 \leq u_k \leq +24$  olarak alınmıştır.



# Adım 3

- 3x3 kural tablosunda 9 kural tanımlanabilir. Çizelge ters sarkaç 'ı dengelemek için kullanılan giriş ve çıkış bulanık değerleri içermektedir.

X1 \ X2	P	Z	N
P	PB	P	Z
Z	P	Z	N
N	Z	N	NB

Çizelge 1: Ters sarkaç kural tablosu

# Adım 4

- Çizelge 1' de görülen kurallar kullanılarak bu problem simüle edilirse, önce başlangıç için keskin değerleri,
- $x_1(0) = 1^\circ$  ve  $x_2 = -4 \text{ dps}$  alınabilir. Sistem 4 döngü için simüle edilecektir. ( $0 \leq k \leq 3$ ). Her döngü için sonuç, üyelik fonksiyonlarına girilen iki giriş değişkeni kullanılarak bulunacaktır. Kural tablosundan  $u(k)$  kontrol hareketini tespit ederek, durulaştırma işlemi için ağırlık ortalaması metodu kullanılacaktır. Bulunan değerler  $x_1$  ve  $x_2$ ' nin bir sonraki değerlerini hesaplamak için kullanılacaktır. Şekillerde  $x_1$  ve  $x_2$  için başlangıç şartlarını göstermektedir. Kural tablosu kullanılarak aşağıdaki eşitlikler yazılabilir

# Klasik Ters Sarkaç Problemi

Eğer  $(x_1 = P)$  ve  $(x_2 = Z)$  ise  $(u = P)$   $\min(0.5, 0.2) = 0.2$  (P)

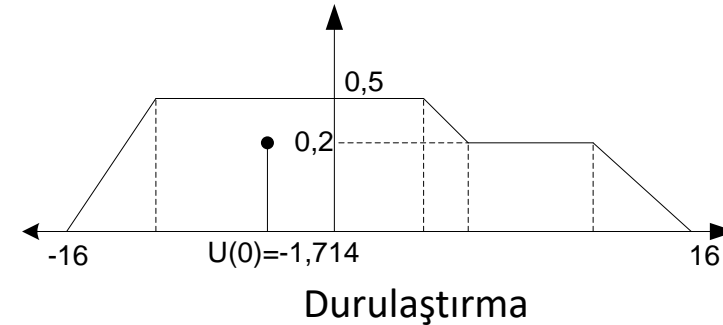
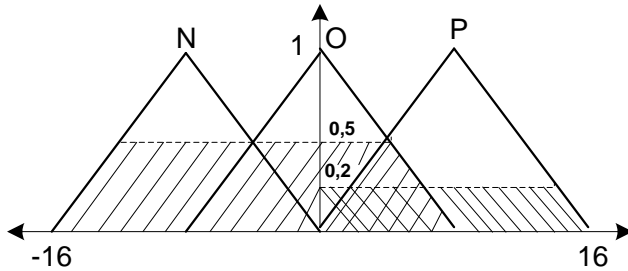
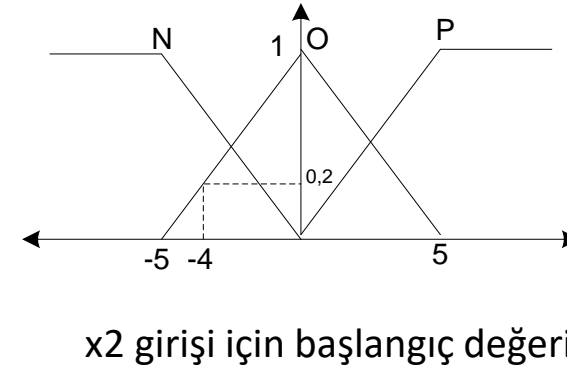
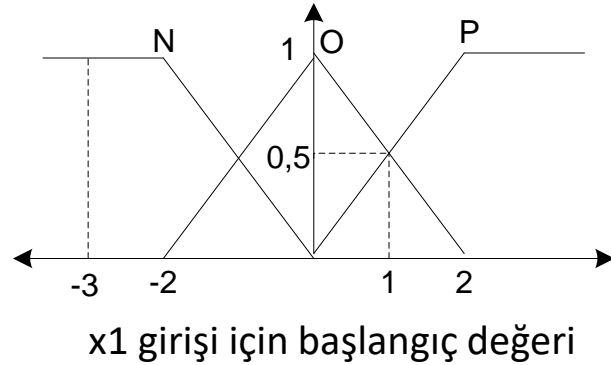
Eğer  $(x_1 = P)$  ve  $(x_2 = N)$  ise  $(u = Z)$   $\min(0.5, 0.8) = 0.5$  (Z)

Eğer  $(x_1 = Z)$  ve  $(x_2 = Z)$  ise  $(u = Z)$   $\min(0.5, 0.2) = 0.2$  (Z)

Eğer  $(x_1 = Z)$  ve  $(x_2 = N)$  ise  $(u = N)$   $\min(0.5, 0.8) = 0.5$  (N).

# Klasik Ters Sarkaç Problemi

- Kontrol değişkeni  $u$  için bulanık çıkışlar ve bunların birleşimi şekilde ve durulaştırılmış kontrol değeri şekilde görülmektedir.





# Klasik Ters Sarkaç Problemi

- Birinci döngü için durulaştırılmış kontrol değişkeni  $u$  için  $-1.714$  değerini bulunmuştur.

$$u = (0.2*8 + 0.5*0 + 0.2*0 + 0.5*(-8))/(0.2+0.5+0.2+0.5) = -1.714$$

- Bir sonraki adım için başlangıç değerleri hesaplanırsa,

$$x1(1) = x1(0) + x2(0) = 1 - 4 = -3$$

$$x2(1) = x1(0) + x2(0) - u(0) = 1 - 4 - (-1.714) = -1.286 \text{ olur.}$$

- İkinci döngü için başlangıç değerleri  $x1(0) = -3$  ve  $x2 = -1.286$  olarak bulunmuştur. Kural tablosu kullanılarak

$$\text{Eğer } (x1 = N) \text{ ve } (x2 = N) \text{ ise } (u = NB) \min(1, 0.2572) = 0.2572 (NB)$$

# Klasik Ters Sarkaç Problemi

- Eğer ( $x_1 = N$ ) ve ( $x_2 = Z$ ) ise ( $u = N$ )  $\min(1, 0.7428) = 0.7428$  (N) bulanık değerleri elde edilir. Bulanık birleşim ve durulaştırılmış değer -10.0576' dır. Üçüncü döngü için başlangıç değerlerini  $u = -10.0576$  değeri kullanılarak hesaplanırsa,

$$x_1(2) = x_1(1) + x_2(1) = -3 - 1.286 = -4.286$$

$$x_2(2) = x_1(1) + x_2(1) - u(1) = -3 - 1.286 - (-10.0576) = 5.7716 \text{ olur.}$$

Bir sonraki adım için başlangıç değerleri  $x_1(2) = -4.286$  ve  $x_2(2) = 5.7716$  bulunur.

Kural tablosu kullanılarak aşağıdaki eşitlik yazılabilir.

- Eğer ( $x_1 = N$ ) ve ( $x_2 = P$ ) ise ( $u = Z$ )  $\min(1,1) = 1$  (Z)

Kontrol değişimi  $u(2)$  için 0 bulunmuştur. Bir sonraki adım için kullanılırsa,

$$x_1(3) = x_1(2) + x_2(2) = -4.286 + 5.7716 = 1.4856$$

# Klasik Ters Sarkaç Problemi

- $x_2(3) = x_1(2) + x_2(2) - u(2) = -4.286 + 5.7716 - 0 = 1.4856$  bulunur. Kural tablosu kullanılarak aşağıdaki bulanık çıkarımlar yapılır.

Eğer  $(x_1 = Z)$  ve  $(x_2 = P)$  ise  $(u = P)$   $\min(0.2572, 0.29712) = 0.2572 (P)$

Eğer  $(x_1 = Z)$  ve  $(x_2 = Z)$  ise  $(u = Z)$   $\min(0.2572, 0.70288) = 0.2572 (Z)$

Eğer  $(x_1 = P)$  ve  $(x_2 = P)$  ise  $(u = PB)$   $\min(0.7428, 0.29712) = 0.29712(PB)$

Eğer  $(x_1 = P)$  ve  $(x_2 = Z)$  ise  $(u = P)$   $\min(0.7428, 0.70288) = 0.70288 (P)$ .

4. adım için  $u = 8.212$  değeri bulunur.

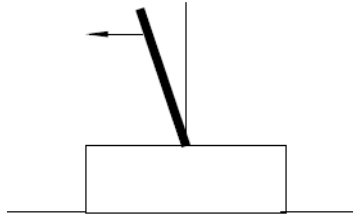
- Bir sonraki adımda  $u = 8.212$  değeri başlangıç değeri için kullanılacaktır. Böylece giriş değerleri aşağıdaki gibi bulunur.

$$x_1(4) = x_1(3) + x_2(3) = 1.4856 + 1.4856 = 2.9712$$

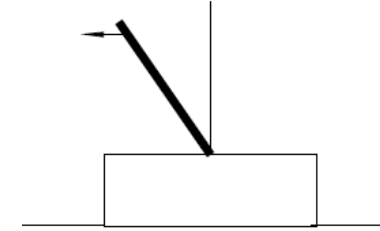
$$x_2(4) = x_1(3) + x_2(3) - u(3) = 1.4856 + 1.4856 - 8.212 = -5.2408$$

# Klasik Ters Sarkaç Problemi

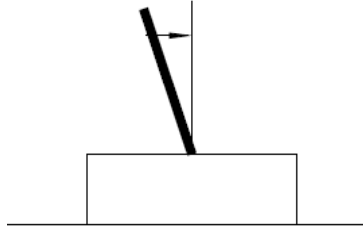
- $x_1$ ,  $x_2$  ve  $u(k)$  için 4 adımlık simülasyonu şekillerde gösterilmektedir.



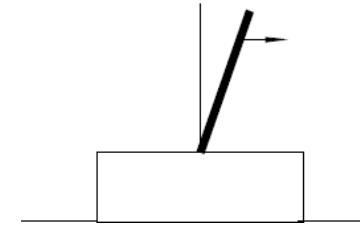
Başlangıç değerleri  $x_1(1) = 1$ ,  $x_2(1) = -4$



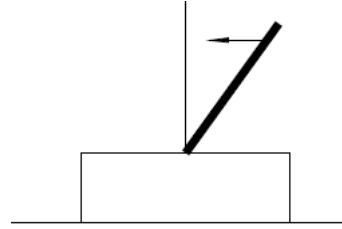
Adım 1:  $x_1(1) = -3$ ,  $x_2(1) = -1.286$



Adım 2:  $x_1(2) = -4.286$ ,  $x_2(2) = 5.7716$



Adım 3:  $x_1(3) = 1.4856$ ,  $x_2(3) = 1.4856$



Adım 4:  $x_1(4) = 2.9712$ ,  $x_2(4) = -5.2408$

# Bulanık Mantık Durulandırma Yöntemleri

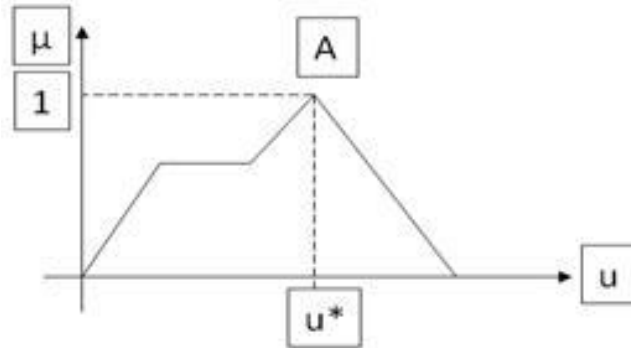
Prof. Dr. Erhan AKIN

# Durulama (Defuzzification) Yöntemleri

Çıkarım biriminin bulanık küme çıkışları üzerinde gerekli ölçekleme değişikliklerini yapar ve bunları gerçek sayılara dönüştürme işlemi gerçekleştirilir. Durulama işleminde çeşitli yöntemler kullanılmaktadır.

- **Üyelik İşlevinin En yüksek Noktası**

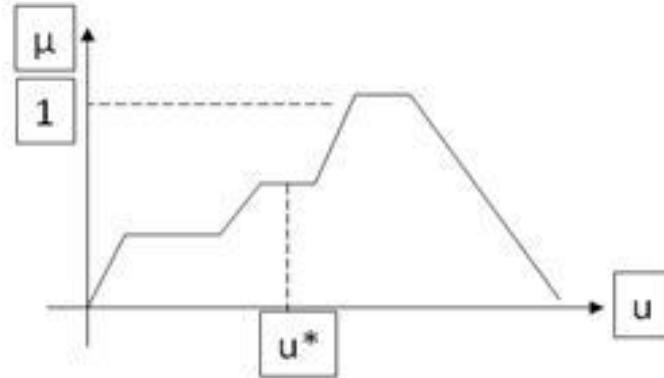
Aktif olan kuralların en büyük üyelik derecesine sahip olan işarete karşılık gelen berraklaştırma değeri dikkate alınır. Burada  $u$  bulanık değer,  $u^*$  ise durulama değeridir.



$$\mu_A(u^*) \geq \mu_A(u)$$

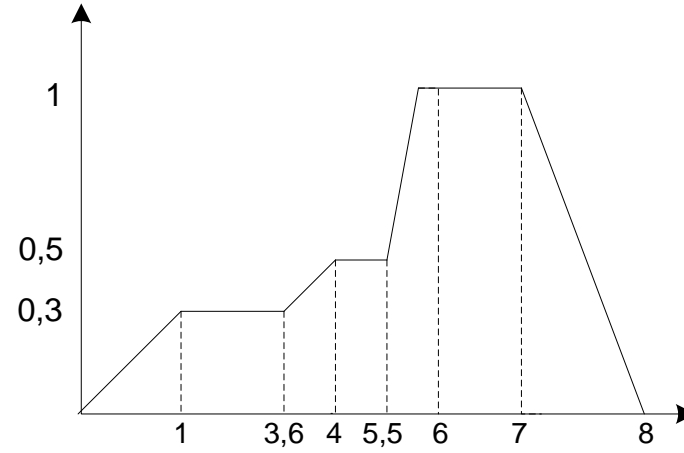
Merkez Yöntemi(Centroid Method), Alan Merkezi(Centre of Area), Ağırlık Merkezi

- Ağırlık merkezi toplanarak yapılan hesaplamaadır.

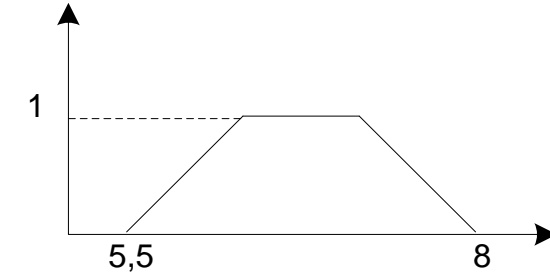
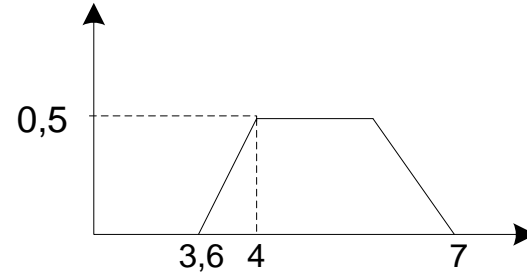
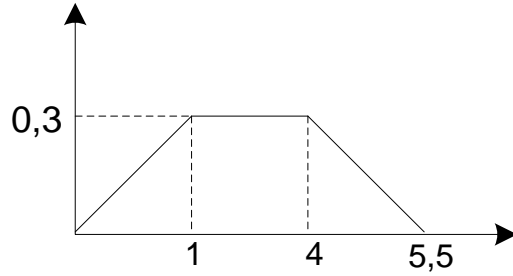


$$u^* = \frac{\int \mu(u).u.du}{\int \mu(u).du}$$

# Örnek



Bu üyelik işlevine kuralları uygularsak şu 3 grafik elde edilir:





# Örnek

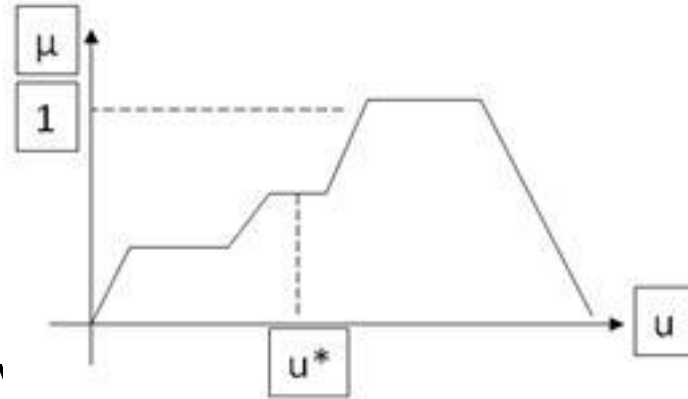
- Formüsel olarak yazacak olursak :

$$u^* = \frac{\int_0^1 (0,3z) \cdot z \, dz + \int_1^{3,6} 0,3 \cdot z \, dz + \int_{3,6}^4 \left(\frac{z-3}{2}\right) \cdot z \, dz + \int_4^{5,5} 0,5 \cdot z \, dz + \int_{5,5}^6 (z-5) \cdot z \, dz + \int_6^7 z \cdot z \, dz + \int_7^8 (8-z) \cdot z \, dz}{\int_0^1 (0,3z) \, dz + \int_1^{3,6} (0,3) \, dz + \int_{3,6}^4 \left(\frac{z-3}{2}\right) \, dz + \int_4^{5,5} (0,5) \, dz + \int_{5,5}^6 (z-5) \, dz + \int_6^7 1 \, dz + \int_7^8 (8-z) \, dz}$$

- = 4,9

# Ortalama Ağırlık Metodu

- Bu yöntem yalnızca simetrik çıkışlı üyelik fonksiyonları için kullanılmaktadır. Her bir simetrik üyelik değerinin tepe noktası değeri belirlenerek ortalamasının alınması yapılmaktadır.



$$u^* = \frac{\sum \mu(u) \cdot u}{\sum \mu(u)}$$

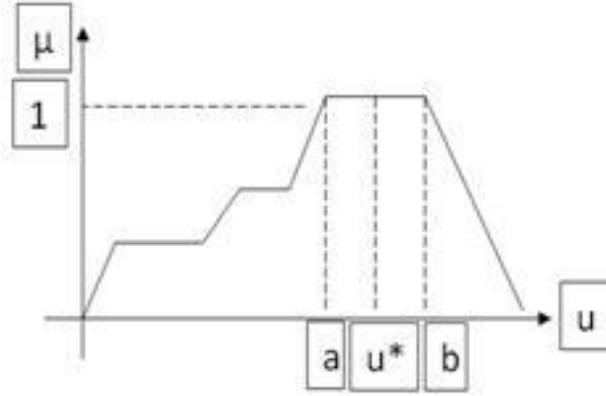
- Bu metoda göre ,

3 grafiğinde ağırlık merkezlerini bulup ortalamalarını alırız. Sonuç şu şekilde olur:

$$u^* = \frac{(0,3 \times 2,5) + (0,5 \times 5) + (1 \times 6,5)}{0,3 + 0,5 + 1} = 5,41 \text{ olur}$$

# Üyelik İşlevinin En Yüksek Noktalarının Alınması

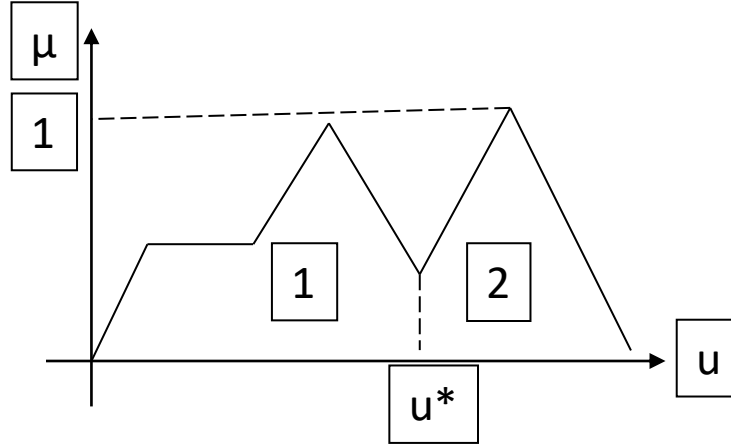
- Üyelik derecesi en yüksek olduğu nokta birden fazladır. Bu durumda ortalaması alınarak değer belirlenir.



$$u^* = \frac{(a+b)}{2}$$

## Geniş Alan Merkezi (Centre of largest area)

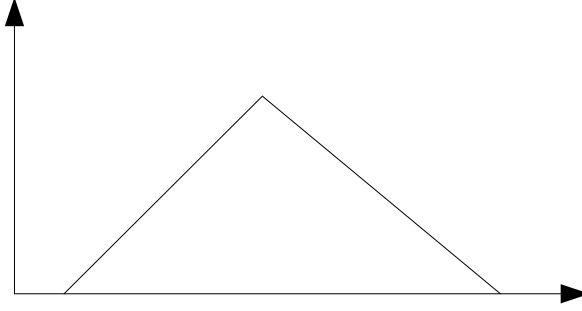
- Bulanık çıkarımlar en az iki tane dışbükey üyelik elemanından oluşuyorsa bu yöntem kullanılabilir. Dışbükeylerden alanı büyük olanın orta noktası alınır.



$$u^* = \frac{\int \mu(u) du}{\int \mu(u) du}$$

1 nolu alan 2 nolu alandan büyük olduğu için 1 nolu alanın orta noktası alınır.

# Larsen Metodu

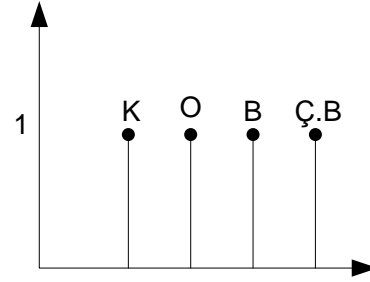


- Bu üyelik fonksiyonundaki üçgenin ağırlık merkezi bulunur ve bu noktaya göre üçgen çizilip bu üçgene göre üçgenin alanı alınır.

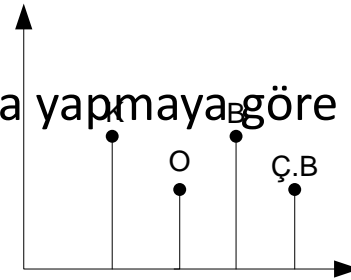


# Singleton Metodu

- Çıkış sürecinde 1 değerine sahip küçük, orta ve büyük değerli çubuklar çizilir. Daha sonra gelen değerlere göre yeni değerler alınır. Alınan yeni değerlere göre ortalama alınır.



- Yeni değerlere göre şöyle olursa:

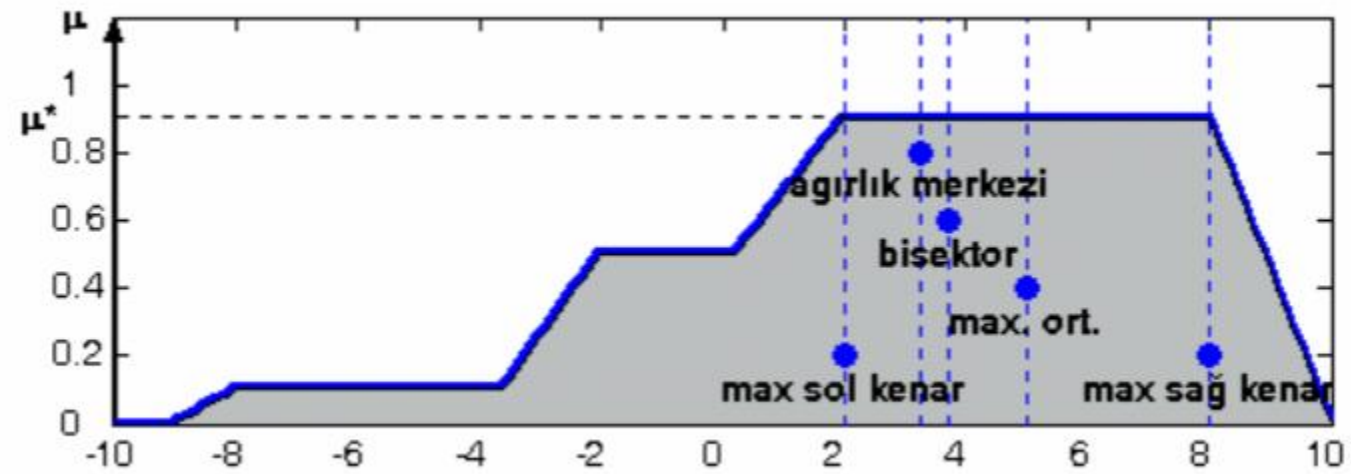


- Bu yöntem üçgenler üzerinde durulama yapmaya göre daha kolaydır.

# Örnek



# Örnek



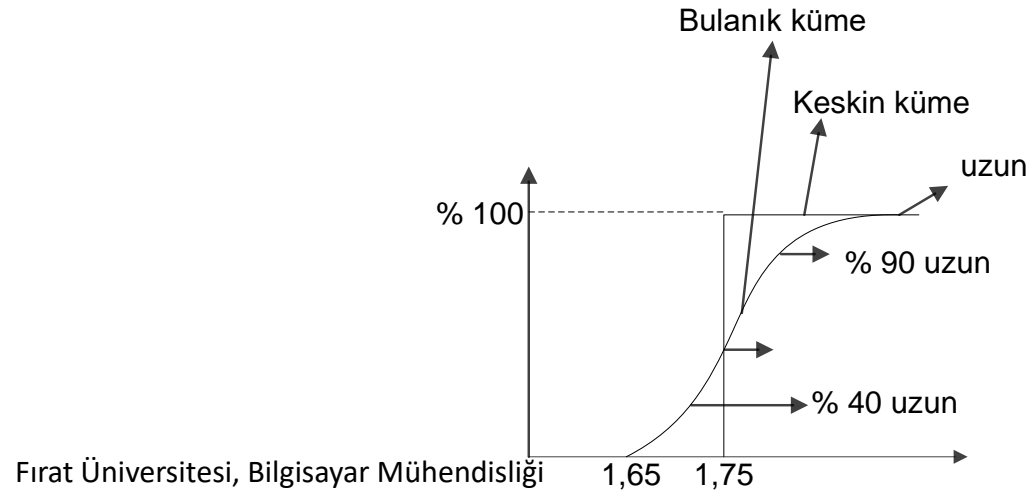


# Bulanık Üyelik fonksiyonları

Prof. Dr. Erhan AKIN

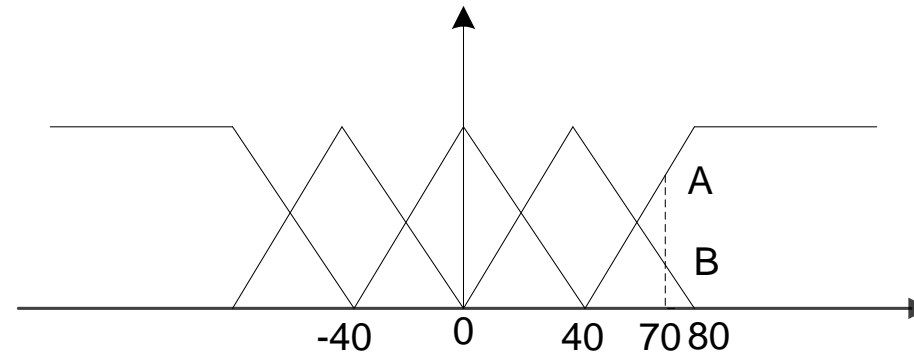
# Bulanık mantık

- Bulanık mantık, keskin mantığı açık /kapalı, soğuk/sıcak, hızlı/yavaş gibi ikili denetim değişkenlerinden oluşan keskin dünyayı, az açık/az kapalı, serin/ılık, biraz hızlı/biraz yavaş gibi gevşek denetleyicilerle yumuşatarak gerçek dünyamıza benzetir.



# Bulanık mantık

- Bulanık mantıkta üyelik fonksiyonu adında bir kavram vardır. Üyelik işlevlerinde genel olarak küçük, orta ve büyük olmak üzere 3 veya küçük, orta derecede küçük, orta, orta derecede büyük ve büyük olmak üzere 5 veya 0 dolayı ve sıfırın etrafında, pozitif ve negatif küçük, orta ve büyük olmak üzere 7 ayrı etiket tanımlanır. En yaygın olanı 7 etiket kullanılmasıdır.



(Burada 70 değerinin hangi kümeye ait olduğunu bulmaya bulanıklaştırma denir )  
Fırat Üniversitesi, Bilgisayar Mühendisliği

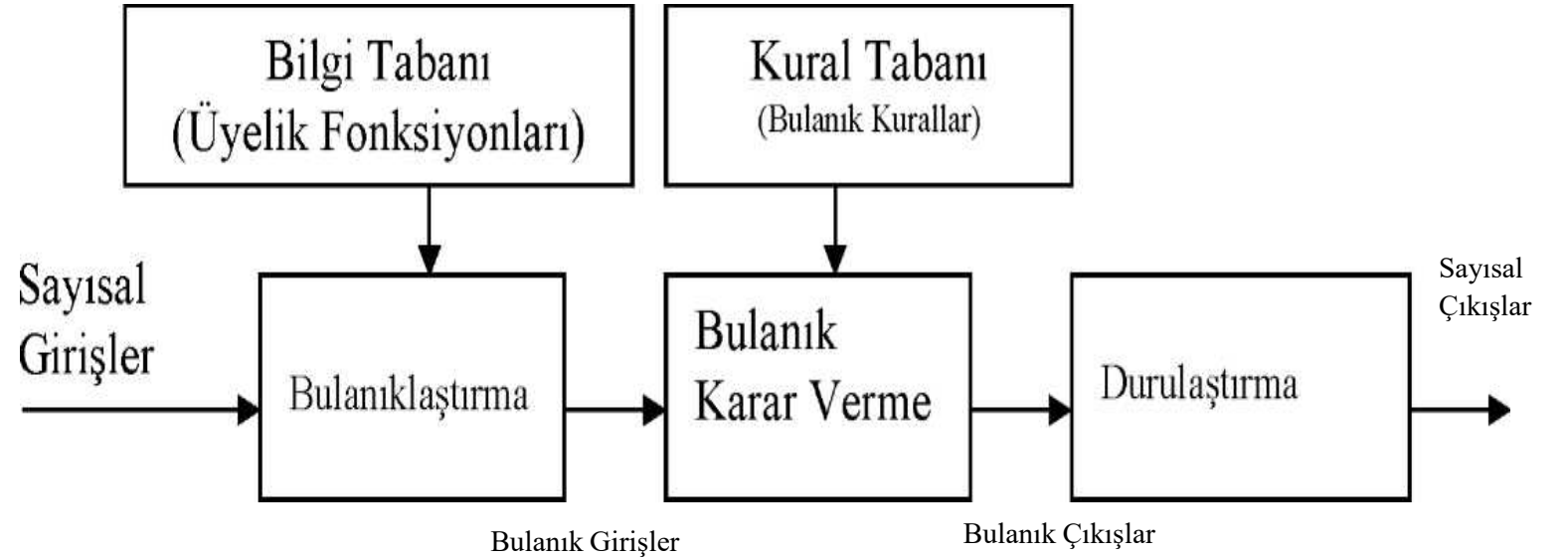
# Bulanık mantık

- Bulanık mantıkta bir değişkeni ele aldığımızda bu değişkenleri üyelik fonksiyonuna dâhil ediyoruz.
- Bulanık mantıkla oluşturduğumuz kümeye “bulanık küme” adı verilir. Var-yok bilgileriyle oluşturulan kümeye “keskin küme” adı verilir.
- Üyelik fonksiyonu ile dilsel ifadeler matematiksel ifadeye dönüştürülmüş olunur.

# Bulanık mantık

- BULANIKLAŞTIRMA: Bu bölüm giriş değişkenlerini (gerçek sayıları)ölçer, onlar üzerinde birçok değişikliği yapar ve bulanık kümelere dönüştürür. Yani onlara birer etiket vererek dilsel nicelik kazandırır.
- BULANIK KARAR VERME: Burada kurallar üzerinde bulanık mantık yürütülür, yani insan beyninin düşünüş şeklinin bir benzetimi yapılmaya çalışılır.
- BİLGİ TABANI: Çıkarım motoru, kural tabanında kullanılan bulanık kümelerin üyelik işlevlerini bu bölümden alır.
- KURAL TABANI: Denetim amaçlarına uygun dilsel denetim kuralları burada bulunur ve çıkarım motoruna buradan verilir.
- DURULAŞTIRMA: Çıkarım motorunun bulanık küme çıkışları(birden fazla çıkış olabilir) üzerinde gerekli ölçek değişikliklerini yapar ve bunları gerçek sayılara dönüştürür

# Bulanık denetleyicinin yapısı



# Mamdani gerçekleştirme

- Mamdani tipi bulanık model çok kolay oluşturulur, insan davranışlarına çok uygundur. Bu nedenle çok yaygın bir kullanıma sahiptir ve diğer bulanık mantık modellerin temelini oluşturur. İlk defa bir buhar motorunun insan tecrübelerinden elde edilen sözel kontrol kuralları yardımıyla kontrolü amacıyla kullanılmıştır (Mamdani ve Assilian, 1975). Bu modelde hem girdi değişkenleri hem de çıktı değişkeni kapalı formdaki üyelik fonksiyonları ile ifade edilir.

# Mamdani gerçekeleştirme

- Mamdani tipi bir bulanık model aşağıdaki 5 adımda oluşturulur;
  - a)Girdilerin bulanıklaştırılması: öncül kısımdaki bütün bulanık ifadeleri kullanarak girdi değişkenlerine ait 0 ile 1 arasında değişen üyelik derecelerinin belirlenmesi.
  - b)Bulanık mantık işlemlerini kullanarak kural ağırlıklarının belirlenmesi
  - c)Bulanık küme mantıksal işlemcilerin (ve, veya uygulanması
  - d)Sonuçların toplanması: her bir kuralın çıktısını temsil eden bulanık kümelerin birleştirilmesi
  - e)Durulaştırma: tek bir sayıya dönüştürülmüş toplam bulanık küme sonuçlarının durulaştırılması



# Mamdani gerçekleştirme

- $x$  ve  $y$  gibi sayısal iki değişkeni içeren iki kurallı bir Mamdani tipi bulanık modelde  $z$  çıkış değerinin bulanık küme fonksiyonlarından nasıl hesaplandığı gösterilmektedir.

Kural 1: Eğer  $x = A1$  VE  $y = B1$  İse  $z = C1$

Kural 2: Eğer  $x = A2$  VE  $y = B2$  İse  $z = C2$

Mamdani tipi bulanık modelin avantajlarını özetlemek gerekirse:

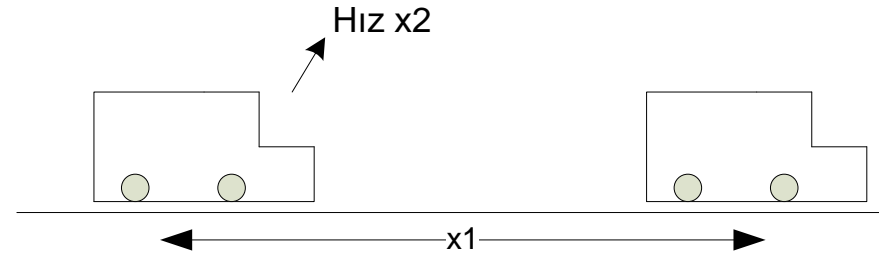
- \*Modelin oluşturulması basittir.
- \*Diğer bulanık mantık modellemenin temelini oluşturur.
- \*İnsan davranış ve duyularına uygundur.

# Örnek

- Bir arabayla bir başka arabayı izlemek isteyelim şekilde görüldüğü gibi öndeki arabayla sizin aranızdaki mesafe  $x_1$  sizin hızınız  $x_2$  olsun. İlk olarak yıllardır araba kullanıyor olmamızla elde ettiğimiz deneyimlerimizi ve ustalığımızı birer dilsel kurallar şeklinde açıklayalım.

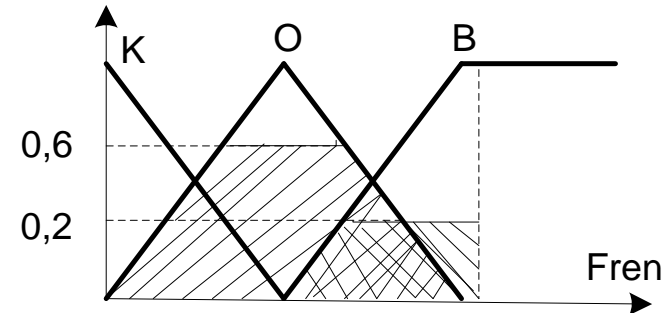
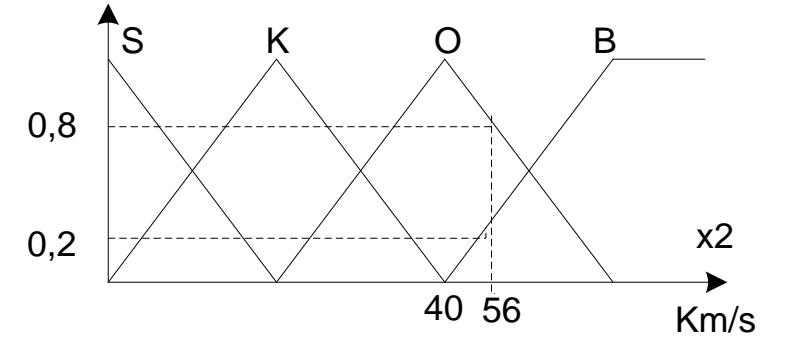
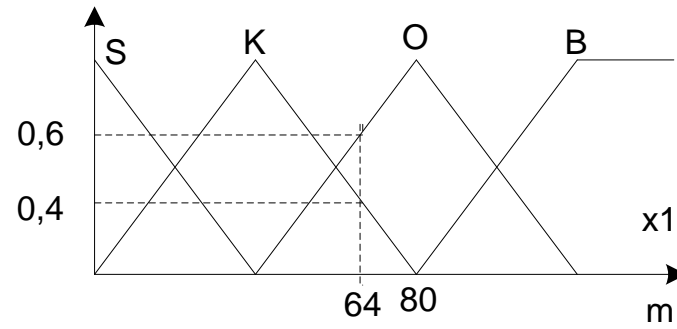
**Kural 1:** Eğer 2 araba arasındaki uzaklık az (küçük-K) ve hızınız orta ise (O) sert (büyük-B) fren yap.

**Kural 2:** Eğer 2 araba arasındaki uzaklık orta (o) ve hızınız yüksek ise (B) orta sertlikte fren yap.



# Örnek

- Üyelik fonksiyonunu oluşturalım:



# Örnek

**K1:**Eğer  $x_1=K$  ve  $x_2=O$  ise o halde  $Y=B$

**K2:**Eğer  $x_1=0$  ve  $x_2=B$  ise o halde  $Y=O$

Üyelik fonksiyonu: Dilsel ifadelerin matematiksel karşılığı.  
Örneğimizde  $x_1=64$  m ve  $x_2=56$  km/s olarak verilmiş.

**Kural 1 için:**

**K1:**Eğer  $x_1=0,4$  ve  $x_2=0,8$  ise  $Y=B$

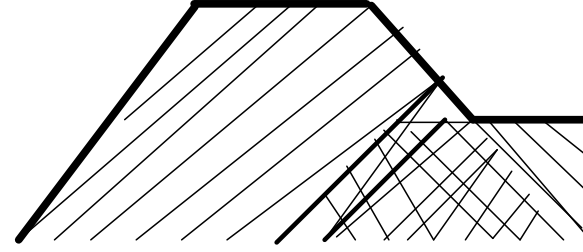
Frenin üyelik fonksiyonunda 0,2 değerine karşılık gelen B durumunu elde ederiz.

**Kural 2 için:**

**K2:** Eğer  $x_1=0,6$  ve  $x_2=0,2$  ise  $Y=O$

# örnek

- Frenin üyelik fonksiyonunda 0,6 değeri karşılık gelen O durumunu elde ederiz.
- Kural 1 ve kural 2'yi uyguladıktan sonra fren durumu şu şekilde olur:

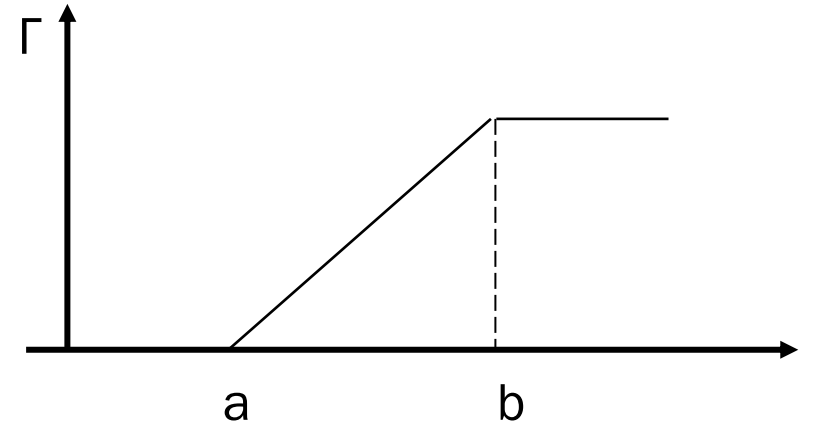


# Üyelik Fonksiyonları

- $\Gamma$  Fonksiyonu

$\Gamma : u [0,1]$

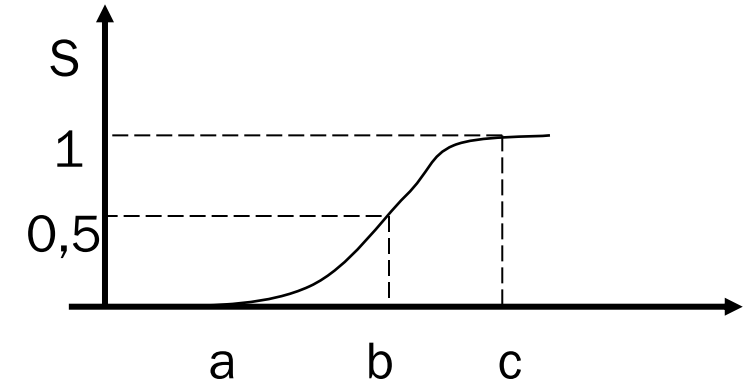
$$(u; a, b) = \begin{cases} 0 & u < a \\ [(u-a)/(b-a)] & a \leq u < b \\ 1 & u > b \end{cases}$$



# Üyelik Fonksiyonları

- Zadeh'in S Fonksiyonu

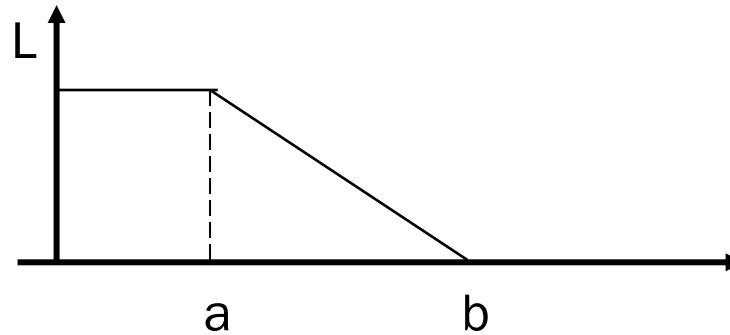
$$S(u; a, b, c) = \begin{cases} 0 & u < a \\ 2[(u - a)/(c - a)]^2 & a \leq u < b \\ 1 - 2[(u - a)/(c - a)]^2 & b \leq u \leq c \\ 1 & u > c \end{cases}$$



# Üyelik Fonksiyonları

- **L Fonksiyonu**

$$L(u; a, b) = \begin{cases} 0 & u < a \\ [(b - u)/(b - a)] & a \leq u \leq b \\ 1 & u > b \end{cases}$$

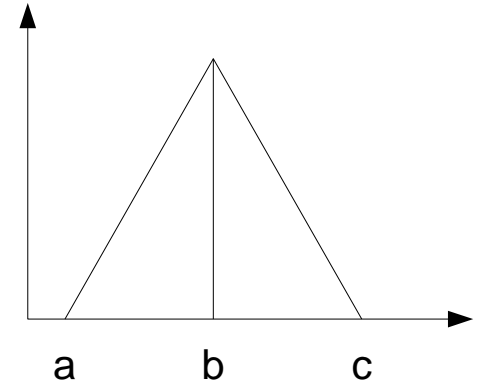




# Üyelik Fonksiyonları

- $\wedge$  Üçgen Fonksiyonu

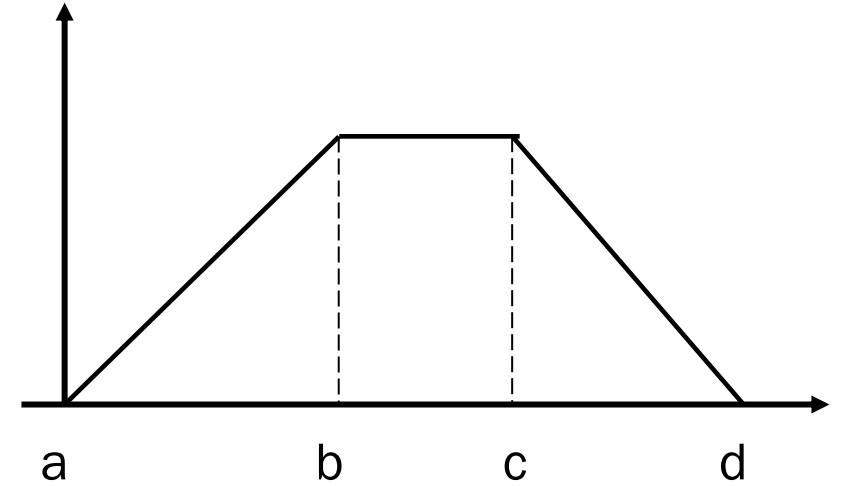
$$\wedge(u; a; b; c) = \begin{cases} \frac{u-a}{b-a}, & a \leq u \leq b \\ \frac{c-u}{c-b}, & b \leq u \leq c \\ 0, & u > c, u < a \end{cases}$$



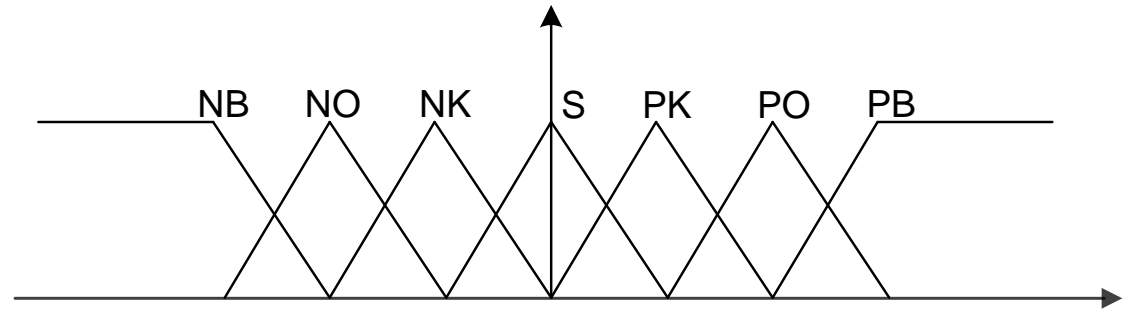
# Üyelik Fonksiyonları

- Yamuk Fonksiyonu

$$\mu(u; a; b; c; d) = \begin{cases} 0, & u < a \\ \frac{u-a}{b-a}, & a \leq u \leq b \\ 1, & b \leq u \leq c \\ \frac{d-u}{d-c}, & c \leq u \leq d \\ 0, & u > d \end{cases}$$



## Bir Değişkenin Fiziksel Değişiminin Elde Edilmesi



- Üyelik fonksiyonlarının sayısını belirlerken kaç tane kuralımızın olacağını da bilmeliyiz. Üyelik fonksiyonlarının taban genişliği önemlidir. Aynı zamanda ikizkenar üçgen olup olmaması da önemlidir.

## Bir Değişkenin Fiziksel Değişiminin Elde Edilmesi

Üyelik fonksiyonu oluştururken şu parametreler önemlidir:

- Taban Genişliği
- Üyelik fonksiyonunun yapısı
- Üyelik fonksiyonlarının sayısı (Kaç tane eylemin ifade edileceği, ne kadar çok eylem (kural) varsa o kadar üyelik fonksiyonu vardır.)

Tanımlanan kurallardan bazıları anlamsızda olabilir. Diğer kuralların altında kalan yani başka kurallar tarafından kapsanan kurallar anlamsızdır ya da geçersizdir.



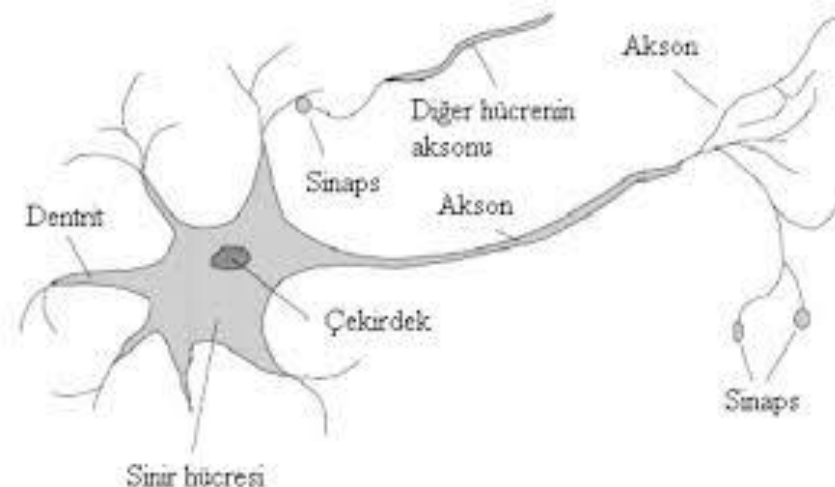
# Sunu içeriği

- Yapay sinir ağları genel tanıtımı
- Yapay sinir ağlarının genel özellikleri
- Yapay sinir ağlarının yapısı
- Yapay sinir ağları elemanları
- Aktivasyon Fonksiyonları
- Biyolojik nöron ve ysa nöron karşılaştırması
- Yapay sinir ağlarının çalışma yapısı
- Yapay sinir ağlarının sınıflandırılması
- Yapay sinir ağlarının avantajları
- Yapay sinir ağlarının dezavantajları
- Geleneksel algoritmalar ile YSA'ların karşılaştırılması
- Yapay sinir ağlarının eğitilme süreci
- Yapay sinir ağlarının uygulama alanları

# Yapay sinir ağıları genel tanıtımı

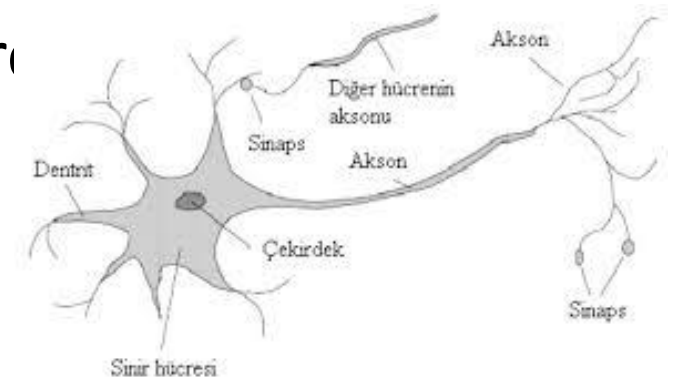
- Öncelikle biyolojik bir sinir hücrelerini anlamakla işe başlamalıyız.
- Biyolojik bir sinir hücresi; bir **gövde**, bir **akson**, çok sayıda sinir ucu(**dentrit**) ve akson ile diğer sinir hücresinin arasında kalan ince uzantılar (**sinaps**) c

uşmaktadır.



# Yapay sinir ağıları genel tanıtımı

- Dendritler, gelen sinyalleri çekirdeğe iletir.
- Çekirdek dendritten gelen sinyalleri bir araya toplar ve aksona iletir.
- Toplanan bu sinyaller, akson tarafından işlenerek sinapslara gönderilir.
- Sinapslar da yeni üretilen sinyalleri diğer sinir hücrelerine iletir.





# Yapay sinir ağlarının genel özellikleri

- Yapay sinir hücreleri, gerçek sinir hücrelerinin simule edilmesiyle gerçekleştirilir.
- Yapay sinir ağları (YSA), insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir.

# Yapay sinir ağlarının genel özellikleri

- YSA, insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır.
- Bu nedendir ki, bu konu üzerindeki çalışmalar **ilk olarak** beyni oluşturan biyolojik üniteler olan **nöronların modellenmesi** ve **bilgisayar sistemlerinde uygulanması ile başlamış**, daha sonraları bilgisayar sistemlerinin gelişimine de paralel olarak bir çok alanda kullanılır hale gelmiştir.

# Yapay sinir ağlarının genel özellikleri

- Kendi kendini organize etme ve öğrenebilme yetenekleri vardır.
- Eksik bilgi ile çalışabilmektedirler.
- Hata toleransına sahiptirler.
- Belirsiz ve tam olmayan bilgileri işleyebilmektedirler.

# Yapay sinir ağlarının genel özellikleri

- İnsan beyninin çalışma prensibini taklit ederek çalışan bu sistemler, her ne kadar bilgisayar teknolojisi hızlı bir gelişim göstermiş, işlem hızları nano saniyeler mertebesine inmiş olsa da, bırakalım insan beynini, **ilkel bir canlı beyninin fonksiyonları dahi baz alındığında, böyle bir organizmanın yanında çok ilkel kalmaktadır.**
- Nano saniyeler bazındaki işlem hızları ile YSA'lar, mili saniyeler mertebesindeki işlem hızları ile işlem yapan insan beyninin işlevselliğinin henüz çok uzağındadır.

# Yapay sinir ağlarının genel özellikleri

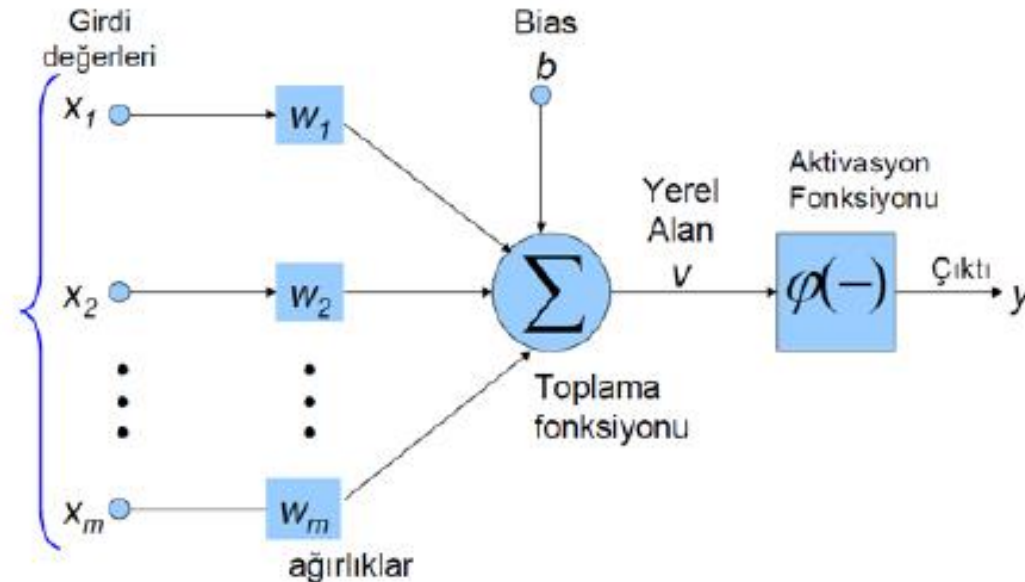
- İnsan beyinde yaklaşık  $10^{11}$  sinir hücresinin varlığından bahsedilmekle birlikte, bu sayının *bilgisayar ortamında modellenmesi şu an için mümkün görünmemektedir.*
- *Fakat karar hızı açısından insan beyni ile henüz yarışamamaları bile, YSA'lar yapısalıkları ve hassas eşleştirmelerin başarı ile gerçekleştirebilmeleri ile gün geçtikçe daha fazla uygulama alanı bulmaktadır.*

# Yapay sinir ağlarının genel 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 tasarlanmışlardır.

# Yapay sinir ağlarının Yapısı

- YSA ların yapısına kısaca yapay nöron ismini verebiliriz.
- Biyolojik nöron örneklerinden sonra basitleştirilmiş bir nöron modeli oluşturulabilir.



# Yapay sinir ağlarının Yapısı

- **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.



# Yapay sinir ağlarının Elemanları

## ■ Girdiler

Yapay sinir ağlarına 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.

# Yapay sinir ağlarının Elemanları

## ■ Aktivasyon Fonksiyonu

Hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktının belirlenmesini sağlar.

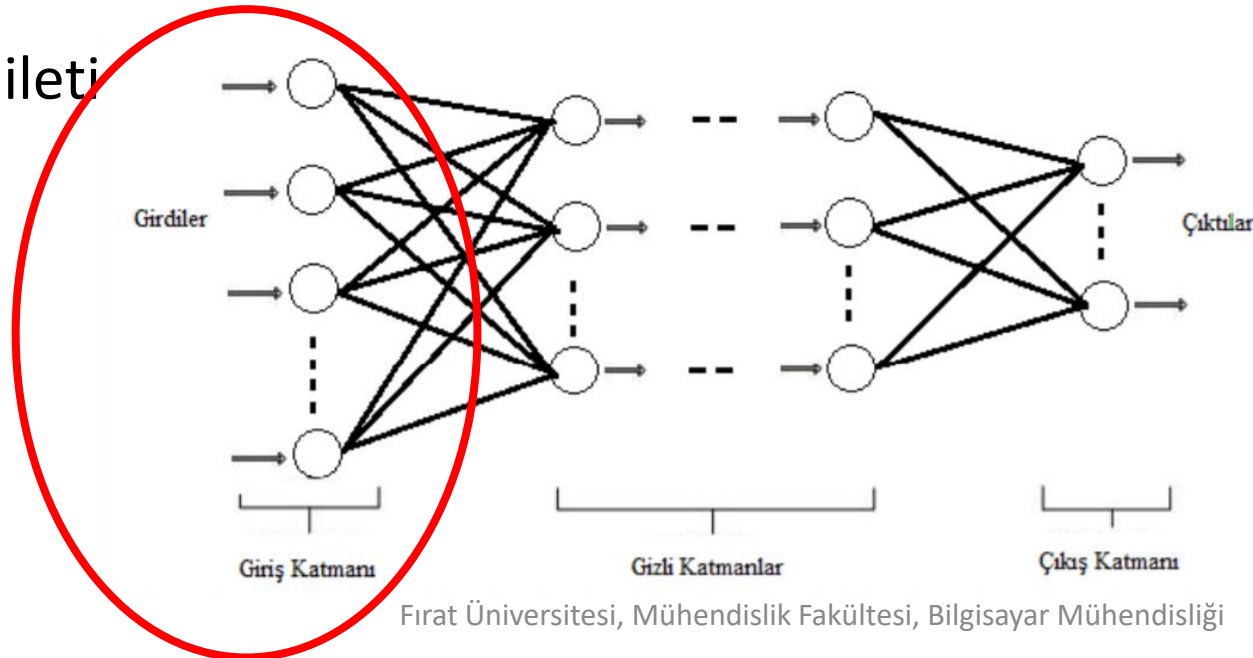
## ■ Çıktılar

Aktivasyon fonksiyonları tarafından belirlenen çıktı değerleridir. Üretilen çıktı ya dış dünyaya, başka bir hücreye ya da kendisine girdi olarak gönderilebilir

# Yapay sinir ağlarının Elemanları

## ■ Giriş Katmanı

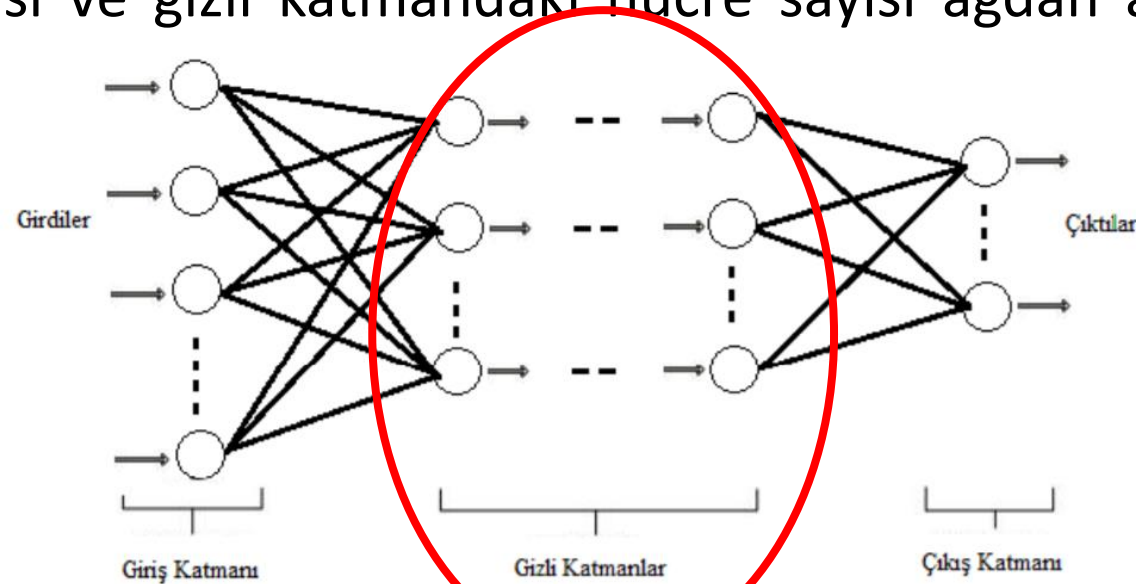
Yapay sinir ağına dış dünyadan girdilerin geldiği katmandır. Bu katmanda, girdi sayısı kadar hücre bulunmaktadır ve girdiler herhangi bir işleme uğramadan gizli katmana iletilir.



# Yapay sinir ağlarının Elemanları

## ■ Gizli Katmanlar

Giriş katmanından aldığı bilgiyi işleyerek bir sonraki katmana iletir. Gizli katman sayısı ve gizli katmandaki hücre sayısı ağdan ağa değişebilir. Gizli katmanlard



şımsızdır.

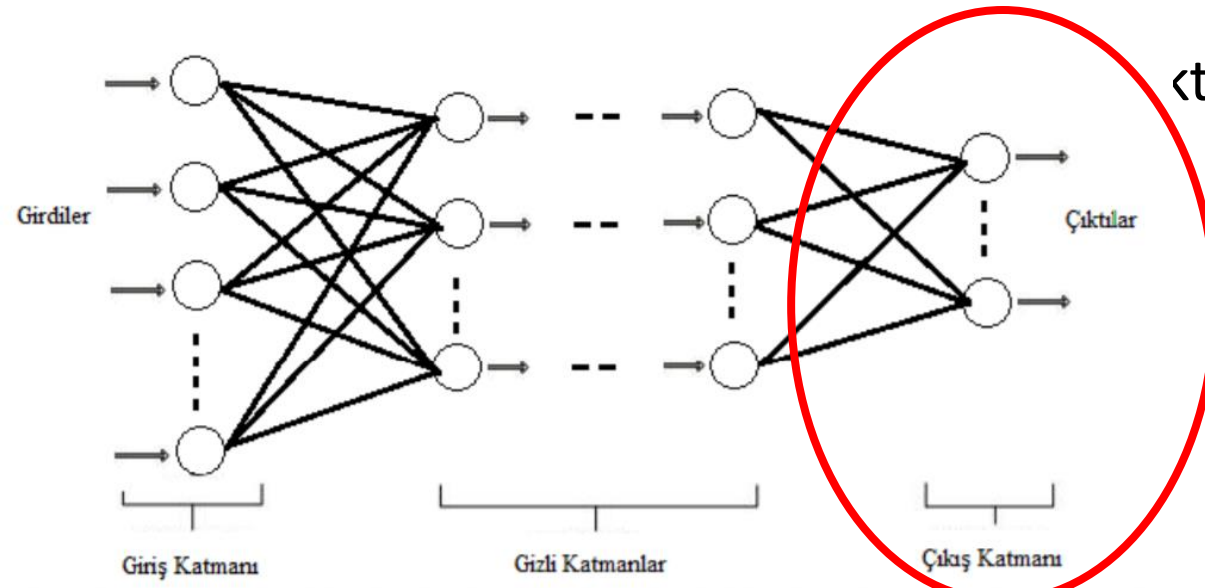
# Yapay sinir ağlarının Elemanları

## ■ Çıkış Katmanı

Gizli katmandan gelen bilgiyi işler ve giriş katmanına gelen girdiye uygun olarak üretilen çıktıyı dış dünyaya gönderir. Çıkış katmanındaki hücre sayısı

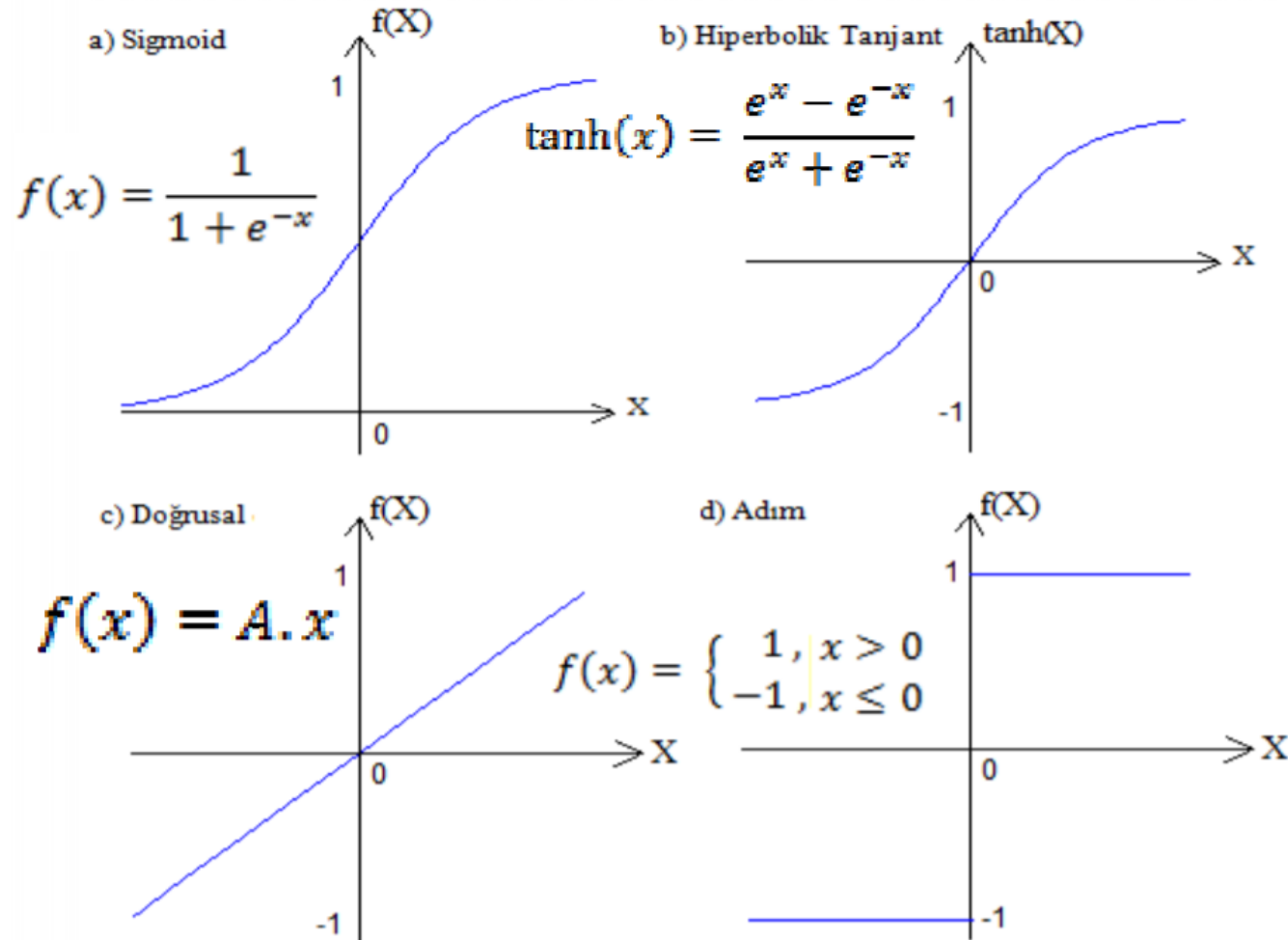
birden büyü

hücre bir ör



ktısı vardır. Her bir

# Aktivasyon fonksiyonları



# Biyolojik Nöron ve YSA Nöron Karşılaştırma

Biyolojik Sinir Hücresi	YSA Sinir Hücresi
Akson	Çıktı
Dentrit	Toplama Fonksiyonu
Çekirdek	Aktivasyon Fonksiyonu
Sinaps	Ağırlıklar

# Yapay sinir ağlarının Çalışma Yapısı

- Örneklerin belirlenmesi
- Ağın topolojisinin belirlenmesi  
Girdi ve çıktı sayısının belirlenmesi
- Ağın öğrenme parametrelerinin belirlenmesi  
öğrenme katsayısı ve sabitlerin belirlenmesi
- Ağın başlangıç değerlerinin atanması
- Epoch sayısı kadar
  - ✓ Eğitim setindeki tüm örnekler için
    - Örnek ağa gösterilir
    - Hatanın hesaplanması
    - Bulunan hataya göre ağırlıkların güncellenmesi
- Sistemin toplam hatası hesaplanır.



# Yapay sinir ağlarının Sınıflandırılması

## ■ Yapılarına Göre

### ■ İleri Beslemeli

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

Bir hücrenin çıktısı sadece kendinden sonra gelen katmana girdi olarak verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir hücreye girdi olarak verilebilir .

# Yapay sinir ağlarının Sınıflandırılması

## ■ Öğrenme Algoritmalarına Göre

### ■ Danışmanlı 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 ağın ürettiği çıktı karşılaştırılarak hata hesabı yapılır, ağırlıklar güncellenir.

### ■ Danışmansız Öğrenme

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

### ■ Pekiştirmeli Öğrenme

Giriş verileri ağa uygulanır ve sonucun danışman tarafından değerlendirilmesi istenir. Ödüllendirme ve cezalandırma yöntemiyle ağın ağırlıkları güncellenir.

# Yapay sinir ağlarının 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 ağın tamamında saklanır.
- Daha önce görülmemiş örnekler hakkında bilgi üretebilir..

# Yapay sinir ağlarının 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.
- Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur.

## Geleneksel algoritmalar ile YSA'ların Karşılaştırılması

Geleneksel Algoritmalar	YSA
Çıkışlar, koyulan kurallara girişlerin uygulanması ile elde edilir.	Öğrenme esnasında giriş çıkış bilgileri verilerek, kurallar koyulur.
Hesaplama; merkezi, eş zamanlı ve ardışıldır.	Hesaplama; toplu, eş zamansız ve öğrenmeden sonra paraleldir.
Bellek paketlenmiş ve hazır bilgi depolanmıştır.	Bellek ayrılmış ve ağa yayılmıştır.
Hata toleransı yoktur.	Hata toleransı vardır.
Nisbeten hızlıdır.	Yavaş ve donanıma bağımlıdır.

# Yapay sinir ağlarının Eğitilme süreci

## ▪ 1. Örneklerin toplanması:

Ağın öğrenmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağın eğitilmesi için örnekler toplandığı gibi (eğitim seti) ağın test edilmesi için de örneklerin (test seti) toplanması gerekmektedir. Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağın performansı ölçülür. Hiç görmediği örnekler karşısındaki başarısı ağın iyi öğrenip öğrenmediğini ortaya koyar.

## ▪ 2. Ağın topolojik yapısının belirlenmesi:

Öğrenilmesi istenen olay için oluşturulacak olan ağın topolojik yapısı 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 öğrenme katsayısı, proses elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.

# Yapay sinir ağlarının Eğitilme süreci

- **4. Ağırlıkların başlangıç değerlerinin atanması:**

Proses elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin atanması yapılır. Başlangıç genellikle rasgele değerler atanır. Daha sonra ağ uygun değerleri öğrenme sırasında kendisi belirler. .

- **5. Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi:**

Ağın öğrenmeye başlaması ve Öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler belirli bir düzeneğe göre gösterilir.

- **6. Öğrenme sırasında ileri hesaplamaların yapılması:**

Sunulan girdi için ağın çıktı değerleri hesaplanır.

# Yapay sinir ağlarının Eğitilme süreci

- **7. Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması:**

Ağın ürettiği hata değerleri bu adımda hesaplanır.

- **8. Ağırlıkların değiştirilmesi:**

Geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.

- **9. Öğrenmenin tamamlanması:**

İleri beslemeli sinir ağı öğrenmeyi tamamlayıncaya, yani gerçekleşen ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar devam eder.



# Yapay sinir ağlarının Uygulama Alanları

- Sınıflandırma
- Karakter, El Yazısı Tanıma
- Kontrol
- Teşhis
- Optimizasyon
- Robotik
- Görüntü işleme
- Veri ilişkilendirme

# Kaynaklar

- Karakuzu, C., 2011, Yapay Sinir Ağları Ders Notları, Bilecik Üniversitesi Fen Bilimleri Enstitüsü, Bilecik.
- Öztemel, E., 2006, Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.
- Elmas, Ç., 2007, Yapay Zeka Uygulamaları, Seçkin Yayıncılık, Ankara.
- Bağış, S., 2009, Yapay Zeka Algoritmaları Kullanılarak Sistem Modelleme, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.
- Yalçın, N., 2020, Yapay Sinir Ağları, Bilecik Üniversitesi, Bilecik.
- Cayiroğlu, İ., 2020, İleri Algoritma Analizi-5, Yapay Sinir Ağları, Karabük Üniversitesi, Mühendislik Fakültesi, Karabük.

# Yapay Sinir Ağları Modelleri

Prof. Dr. Erhan AKIN

# YSA Özellikleri

- YSA modellerinden bahsetmeden önce YSA özelliklerini yeniden hatırlayalım:
  - Doğrusal Olmama
  - Paralel Çalışma
  - Öğrenme
  - Genelleme
  - Hata Toleransı ve Esneklik
  - Eksik Verilerle Çalışma
  - Çok Sayıda Değişken ve Parametre Kullanma
  - Uyarlanabilirlik

# YSA Modelleri

- Yapılarına Göre:
  - İleri beslemeli
  - Geri beslemeli
- Öğrenme algoritmalarına göre
  - Danışmanlı
  - Danışmansız
  - Takviyeli
- Öğrenme zamanına göre
  - Dinamik
  - Statik

# YSA Modelleri

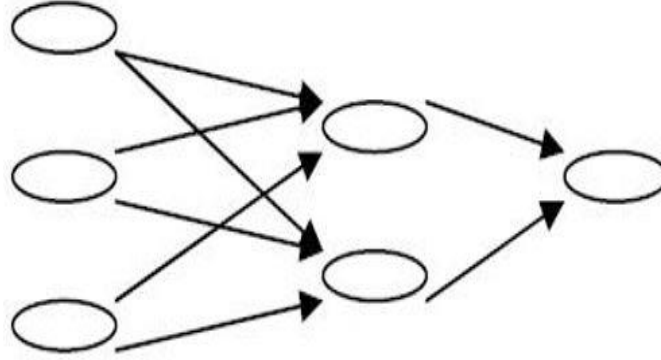
- Hücrelerin bağlantı şekillerine, öğrenme kurallarına ve aktivasyon fonksiyonlarına bağlı olarak çeşitli YSA yapıları geliştirilmiştir.
- YSA modelleri temel olarak iki grupta toplanabilmektedir:
  - İleri beslemeli yapay sinir ağları
  - Geri beslemeli yapay sinir ağları

# İleri Beslemeli YSA

- İleri beslemeli YSA'da, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir.
- Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan orta (gizli) katmandaki hücrelere iletir.

# İleri Beslemeli YSA

- Bilgi, orta ve çıkış katmanında islenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar doğrusal olmayan statik bir işlevi gerçekleştirir.





# İleri Beslemeli YSA

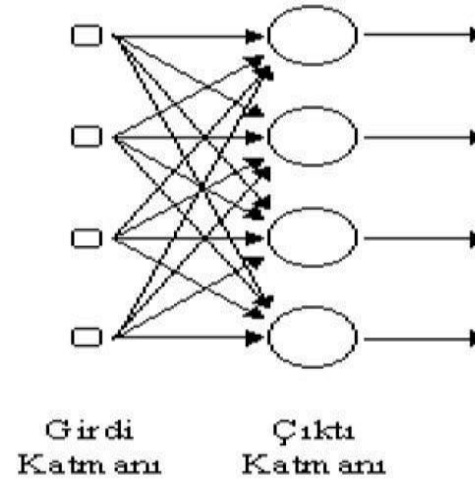
- Tek yönlü bilgi akışı söz konusudur.
- Bu ağ modelinde Girdi tabakasından alınan bilgiler Gizli katmana iletilir.
- Gizli ve Çıktı tabakalarından bilginin işlenmesi ile çıkış değeri belirlenir.

# İleri Beslemeli YSA

- İleri beslemeli YSA'da, gecikmeler yoktur.
- Kendi aralarında :
  - Tek katmanlı ileri beslemeli ağlar
  - Çok katmanlı ileri beslemeli ağlar olarak ayrılırlar.

# Tek Katmanlı İleri Beslemeli YSA

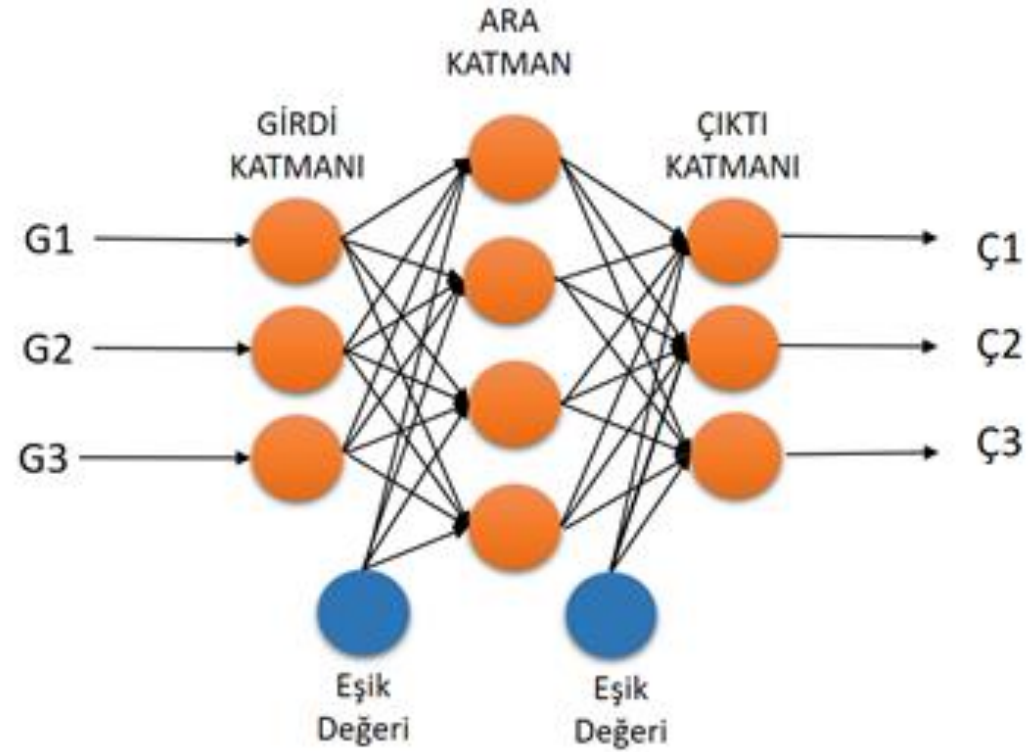
- En basit ağı tipidir.
- Bir çıkış katmanı ve buna bağlı olarak bir girdi katmanından oluşmaktadır.



# Çok Katmanlı İleri Beslemeli YSA

- Girdi katmanı dış ortamdan aldığı bilgileri, hiçbir değişikliğe uğratmadan gizli katmandaki hücrelere iletir.
- Bilgi, orta(gizli) ve çıkış katmanında işlenerek ağ çıkışı belirlenir.
- Çok katmanlı ağlar, tek katmanlı ağlara göre daha karmaşık problemlerin çözümünde kullanılırlar.
- Çok katmanlı ağların eğitilmesi zordur.

# Çok Katmanlı İleri Beslemeli YSA



# İleri Beslemeli YSA – Örnek Soru

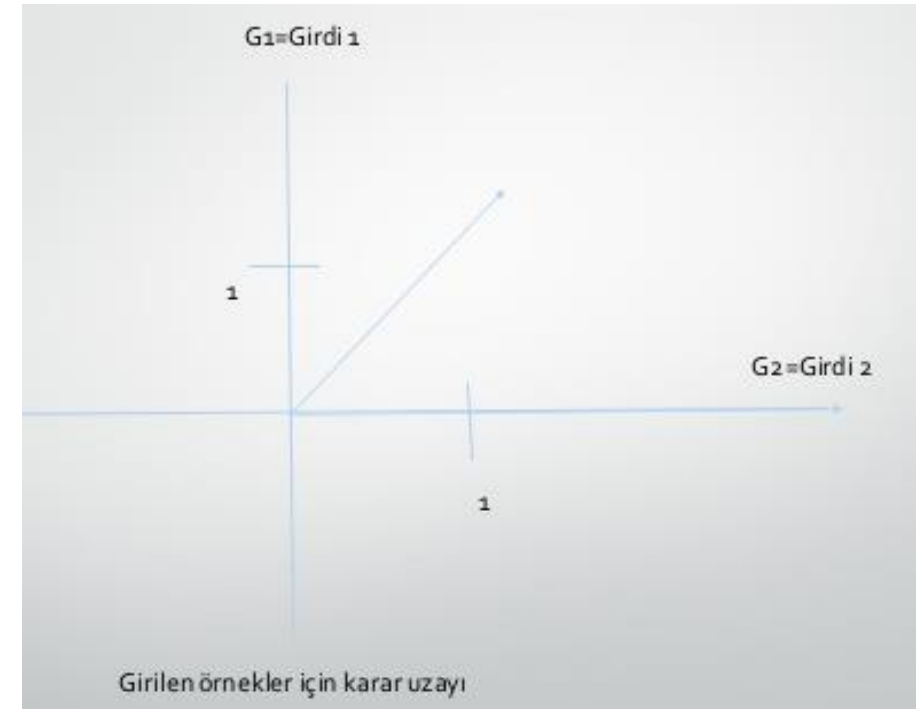
- Xor Problemini YSA ile çözüm adımlarını gösteriniz.
- YSA'da yaşanan bir problem olan XOR problemine göre tek katmanla çözülmesi mümkün değildir.
- Bu durum [doğrusal ayrılabilirlik \(linearly seperable\)](#) ile açıklanabilir.
- XOR probleminin çözümü için çok sayıda katmana ihtiyaç duyulur.
- Problemin çözümü için giriş ve çıkış katmanları dahil olmak üzere 3 katman gerekmektedir.

# İleri Beslemeli YSA – Örnek Soru

*XOR Doğruluk Tablosu*

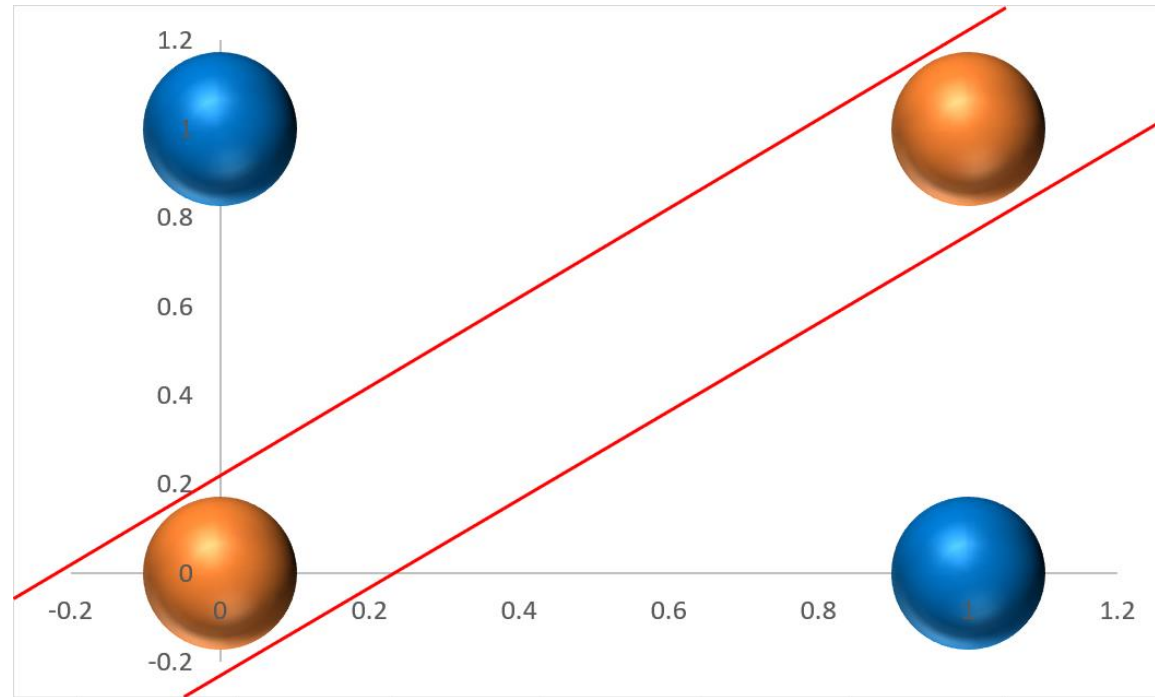
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

A ve B giriş değerlerimiz  
(Girdi1 ve Girdi2) olmak  
üzere



# İleri Beslemeli YSA – Örnek Soru

- 





# İleri Beslemeli YSA – Örnek Soru

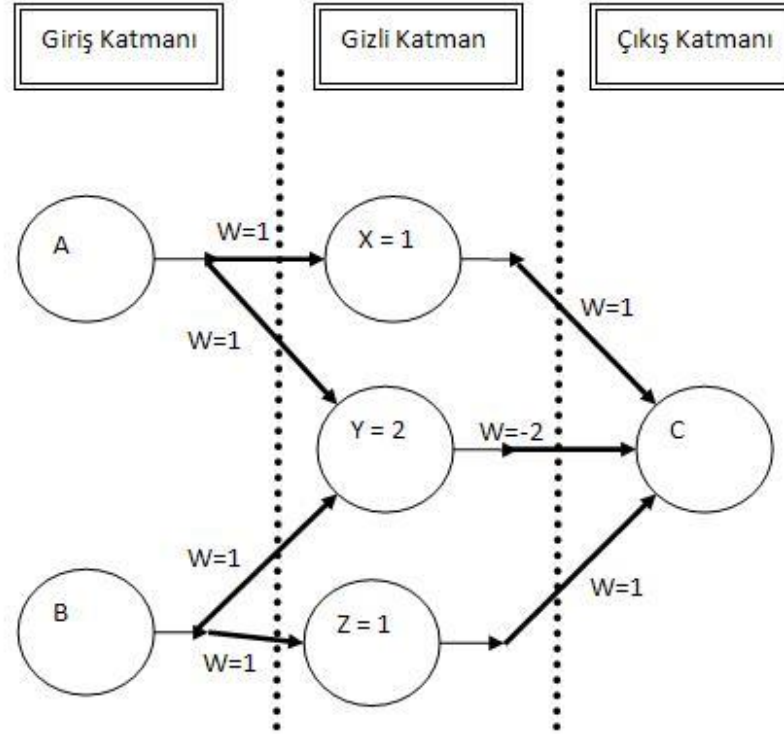
- Şekilde de görüldüğü gibi iki boyutlu uzayda yayılan bu veri kümesini birbirinden ayırmak istiyoruz.
- Burada aslında 0 ve 1'lerden oluşan veri kümelerini birbirinden ayıracağız.
- Mavi renkler 1 çıktılarını ifade etmekte, turuncu renk ise 0 çıktısını temsil etmektedir.

# İleri Beslemeli YSA – Örnek Soru

- Ama burada veri kümelerini doğrusal olarak ayıramamaktayız.
- Bunun sebebi tek bir doğru ile veri kümelerinin birbirinden ayıramıyor oluşudur.
- **Yani XOR, lineer olarak ayrılabilen bir problem değildir.**

# İleri Beslemeli YSA – Örnek Soru

- $C = \text{XOR}(A, B)$  fonksiyonunun çözümü aşağıdaki üç katmanla sağlanmıştır.



Şekildeki **sinapsler** üzerinde verilen **W değerleri** ağırlıkları göstermektedir.

Ayrıca şekilde gizli katmanda bulunan **nöronlar için eşik değeri (threshold)** verilmiştir.

**Eşik değeri = -1** olarak verilmiştir.

**(Aktivasyon Fonk:**

Eşik değerine küçük veya eşit olduğu durumda sonuç 1 aksi durumda sonuç 0 üretmesi sağlanmıştır.)

# İleri Beslemeli YSA – Örnek Soru

- $NET = \sum w_i x_i + f$
- $W_i$  : Ağırlık değeri
- $X_i$ : Giriş değeri
- $F$ : eşik değeri (varsa)
- $NET$ : Nöronun çıkış değeridir.

# İleri Beslemeli YSA – Örnek Soru

- Bu fonksiyonun çalışmasını **0** ve **0** girişi için test edelim:
- $A = 0$  ve  $B = 0$  için nöronlardaki değerler sırasıyla
- $X = 0$  ,  $Y = 0$  ,  $Z = 0$  olacaktır ve çıkış değeri için formülden  
 $0 * 1 + 0 * (-2) + 0 * 1 = \mathbf{0}$  değeri bulunur.

# İleri Beslemeli YSA – Örnek Soru

- Fonksiyonu **0** ve **1** girişi için test edecek olursak:
- $A = 0$  ve  $B = 1$  için nöronlardaki değerler sırasıyla
- $X=0, Y=1, Z=1$  değerleri bulunur ve bu değerler sonucunda

$0*1+1*(-2)+1*1 = -1$  olur ve aktivasyon fonksiyonuna göre sonuç=1 olarak bulunur.

# İleri Beslemeli YSA – Örnek Soru

- Fonksiyonu **1** ve **0** girişi için test edecek olursak:
- $A = 1$  ve  $B = 0$  için nöronlardaki değerler sırasıyla
- $X=1, Y=1, Z=0$  değerleri bulunur ve bu değerler sonucunda

$1*1+1*(-2)+0*1 = -1$  olur ve aktivasyon fonksiyonuna göre sonuç=1 olarak bulunur.

# İleri Beslemeli YSA – Örnek Soru

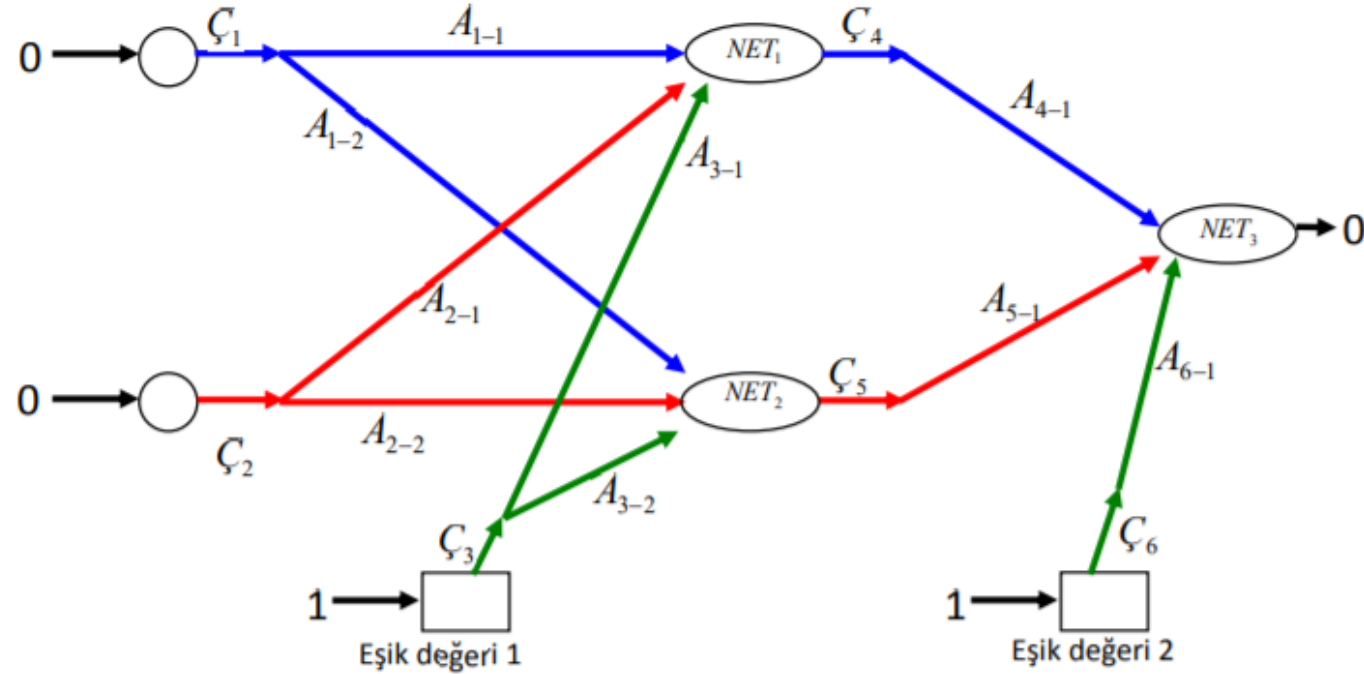
- Fonksiyonu **1** ve **1** girişi için test edecek olursak:
- $A = 1$  ve  $B = 1$  için nöronlardaki değerler sırasıyla
- $X=1$ ,  $Y=2$ ,  $Z=1$  değerleri bulunur ve bu değerler sonucunda  
 $1*1+1*(-2)+1*1 = 0$  olur ve aktivasyon fonksiyonuna göre sonuç=1 olarak bulunur.



# İleri Beslemeli YSA – Örnek Soru 2

Aşağıdaki ileri yönlü YSA' nın çözünüz. NOT: Sigmoid Aktivasyon fonksiyonu tüm katmanlarda

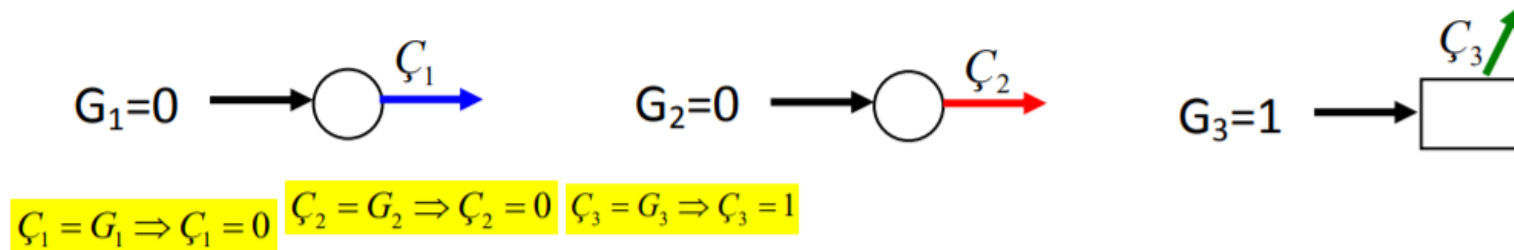
kullanılacaktır.  $1/(1 + e^{-z})$



# İleri Beslemeli YSA – Örnek Soru 2

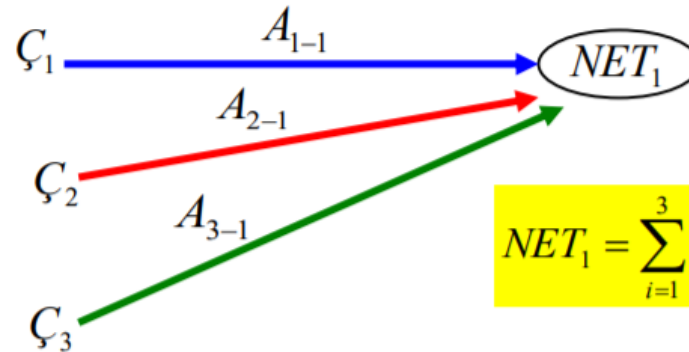
- İleriye doğru hesaplama yöntemi;
  - İlk giriş değerleri, ilk katman olan girdi katmanında çıkış değerlerine eşittir.

Giriş katmanındaki çıkış değerlerini  $\zeta_1$ ,  $\zeta_2$  'i yi ve eşik değeri 1'i hesaplayalım



# İleri Beslemeli YSA – Örnek Soru 2

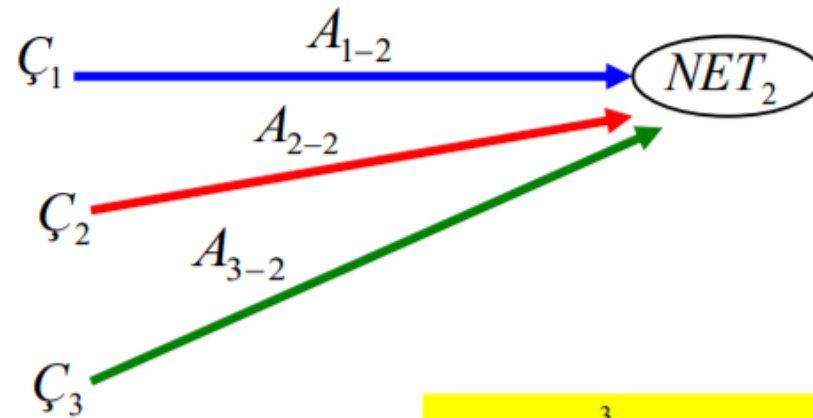
- *İkinci katman olan ara katmana giriş değerlerini hesaplamak için*
  - Giriş katmanından gelen **çıkış değerlerinin** ile **ağırlık değerlerinin çarpımlarını toplayarak** buluruz. Aşağıda **iki ara katmanın giriş değerleri** hesaplanmıştır.



$$NET_1 = \sum_{i=1}^3 C_i * A_i \Rightarrow NET_1 = C_1 * A_1 + C_2 * A_2 + C_3 * A_3$$

$$NET_1 = 0 * 0,129952 + 0 * (-0,923123) + 1 * 0,341232 = 0,341232$$

# İleri Beslemeli YSA – Örnek Soru 2



$$NET_2 = \sum_{i=1}^3 C_i * A_i \Rightarrow NET_2 = C_1 * A_1 + C_2 * A_2 + C_3 * A_3$$

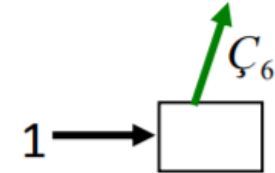
$$NET_2 = 0 * 0,570345 + 0 * (-0,328932) + 1 * (-0,115223) = -0,115223$$

# İleri Beslemeli YSA – Örnek Soru 2

- Ara katmanların çıkış değerleri  $F(\text{NET})$  formülüyle hesaplanmaktadır.
- İki ara nöronun çıkış değerlerini  $\text{Ç4}$ ,  $\text{Ç5}$  'i ve *eşik değeri 2*'i hesaplayalım. (Aktivasyon fonksiyonu uygulanması...)

# İleri Beslemeli YSA – Örnek Soru 2

•



$$C_4 = F(NE_{T_1}) = \frac{1}{1 + e^{-NE_{T_1}}} = \frac{1}{1 + e^{-0,341232}} = 0,584490$$

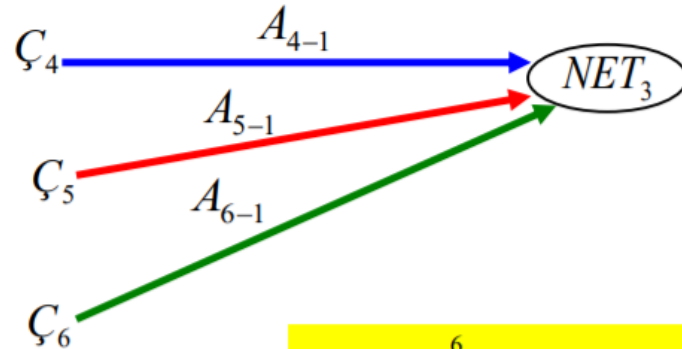
$$C_5 = F(NE_{T_1}) = \frac{1}{1 + e^{-NE_{T_2}}} = \frac{1}{1 + e^{0,115223}} = 0,471226$$

Eşik değeri 2 hesaplanması

$$C_6 = G_6 \Rightarrow C_6 = 1$$

# İleri Beslemeli YSA – Örnek Soru 2

- **Üçüncü katman** olan çıktı katmanının giriş değeri ara katmanın giriş değerinin hesaplanması ile aynıdır. *Çıktı katmanının giriş değerini hesaplayalım*



$$NET_3 = \sum_{i=4}^6 C_i * A_i \Rightarrow NET_3 = C_4 * A_4 + C_5 * A_5 + C_6 * A_6$$

$$NET_3 = 0,584490 * 0,164732 + 0,471226 * 0,752621 + 1 * (-0,993423) = -0,542484$$

# İleri Beslemeli YSA – Örnek Soru 2

- Problemin çıkış değerini  $F(NET)$  formülü ile hesaplayalım;



$$C_7 = F(NET_3) = \frac{1}{1 + e^{-NET_3}} = \frac{1}{1 + e^{542484}} = 0,367610$$



# Kaynaklar

- Cayiroglu, İ., 2020, İleri Algoritma Analizi-5, Yapay Sinir Ağları, Karabük Üniversitesi, Mühendislik Fakültesi, Karabük.
- Öztemel, E., 2006, Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.