YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FUZZY LOJİK PROBLEMLERİNDE ÜYELİK FONKSİYONUNUN BELİRLENMESİNDE DENEYSEL VERİLERE DAYANARAK BİR YÖNTEM GELİŞTİRİLMESİ

Doç. Dr. İsmail Hakkı Armutlulu

Doç. Dr. Ayşe Kurmuzüm

Salih KARANFİL

F.B.E. Matematik Anabilim Dalında
hazırlanan

DOKTORA TEZİ

Tez Danışmanı : Prof. Yavuz AKSOY

İSTANBUL, 1997

İÇİNDEKİLER

TEŞEKKÜR

Tüm çalışmalarımda ve tezimin hazırlanması sırasında yardımlarını hiçbir zaman esirgemeyen çok değerli hocam Sayın Prof. Yavuz AKSOY'a sonsuz teşekkürlerimi sunarım.

Ayrıca çalışmalarımda ve araştırmalarımda bilgi, fikir ve öneri katkılarında bulunan hocalarım Prof. Dr. Galip CANSEVER, Doç. Dr. İsmail Hakkı ARMUTLULU, Araştırma görevlisi arkadaşlarım Kayhan GÜLEZ, Gürsel HACIBEKİROĞLU, Hamid TORPİ, Kamil DİMİLİLER ve şu ana kadar bana emeği geçmiş, beni yetiştiren tüm hocalarıma teşekkürlerimi belirtmek isterim.

Problemin çözümünde kullandığımız pascal programının problemimize uygun duruma getirilmesinde yardımcı olan fedakar arkadaşım bigisayar programcısı Halil SAYARSOY'a ve tüm yardımlarından dolayı eşim Arş. Gör. Gaye Türker KARANFİL'e sonsuz teşekkürler.

Salih KARANFİL

Haziran, 1997

ÖZET

"Fuzzy Kümeler Teorisi" belirsizliğin temsili için 1965 yılında Lotfi A. Zadeh tarafından geliştirilmiş ve üzerinde dünyanın her yerinde birçok insan tarafından çalışmalar yapılmıştır. Daha sonraları fuzzy sayılar ve fuzzy mantık teorisi geliştirilmiş ve günümüze kadar büyük gelişmeler kaydedilmiştir.

Tezin giriş bölümünde fuzzy kümeler ve fuzzy mantığın tarihsel gelişimine yer verilmiştir.

Bölüm II, fuzzy küme teorisinin temellerini, fuzzy kümeler üzerindeki işlemleri ve fuzzy kümelerin uygulama alanlarını içermektedir.

Üçüncü bölümde ise, üyelik fonksiyonları, üyelik fonksiyonu şekilleri, üyelik fonksiyonu özellikleri ve üyelik fonksiyonu atamaları üzerinde durulmuştur.

Bölüm IV, yapay sinir ağlarının gelişimini, yapısını ve işleyişini içermektedir.

Beşinci bölümde, yedinci bölümde ele aldığımız problemimize yardımcı olması bakımından İstanbul Menkul Kıymetler Borsası (İMKB) hakkında temel ve açıklayıcı bilgiler verilmiştir.

Altıncı bölümde de problemimizin çözümünde bize yardımcı olacak kümeleme yöntemi konusunda temel bilgiler verilmiştir.

Son bölümde ise, problemimizi oluşturan, yapay sinir ağlarını kullanarak, deneysel verilere dayalı olarak İMKB'deki en düşük ve kapanış değerleri arasında bir ilişki kuran fuzzy üyelik fonksiyonunun geliştirilmesi üzerinde durulmuştur. Bununla ilgili pascal programlama dilinde bir program hazırlanmış ve problemimize uygulanmıştır.

SUMMARY

The theory of fuzzy sets for the representation of uncertainty was introduced in by Lotfi A. Zadeh and has been studied and applied by many people in all parts of the world. More recently the theory of fuzzy numbers and fuzzy logics has been introduced and up to now a great development was provided.

Historical development of fuzzy sets and fuzzy logics are given in the introduction of the thesis.

Chapter II covers the fundamentals of fuzzy set theory, operations on fuzzy sets and application areas of fuzzy sets.

Chapter III covers membership functions, their shapes and properties and the methods of membership value assignments .

Neural Network's development, structure and working are mentioned in fourth chapter.

In chapter V, the basic informations about İstanbul Stocks Exchange (İMKB) are given. We are going to use this information in chapter seven while we are solving our problem.

Again the basic informations about clustering are given in chapter six, which we are going to use while we are solving our problem.

Finally, in the last chapter, for solving our problem depending on data, by using neural network, the relationship between the minimum price and the closing price in İMKB is developed by a fuzzy membership function. A pascal program is developed and applied to our problem.

1.0. GİRİŞ

1965 yılının başlarında Lotfi A. Zadeh (Berkeley - California), belirsizliğin temsili için araç olarak "Fuzzy Kümeler" ve "Fuzzy Mantık" teorisini geliştirmiş ve bu bilim dünyasında bir çığır açmıştır. Onu takiben Rescher, Dubois, Prade, Lakeoff, Yager, Kandel ve diğerleri fuzzy kümeler ve fuzzy mantığın gelişmesine katkıda bulunmuşlardır.

Türkçe karşılığı "bulanık", "puslu" olan fuzzy kelimesi civciv tüyü anlamına gelen İngilizce "fuzz" kelimesinden türetilmiştir. Geçmişte, belirsizlik ifade eden terimler ve kavramlar rastgele bir ayırıma tabi tutulmuşlar ve iki değerli kümeler kuramı ile tanımlanmışlardır. Bunun yanı sıra son otuziki yılda gelişen "Fuzzy Kümeler Kuramı" ise, belirsizlik ifade eden terimler ve kavramların rastgele bir ayırıma tabi tutulmaksızın, belirsizliğe belirlilik derecesi atayarak çok değerli kümeler kuramı ve kapsamı içinde tanımlanmalarına yol açmıştır.

Fuzzy mantık ise temelde çok değerli mantık, olasılık kuramı ve yapay zeka alanları üzerine oturtulmuştur.

Fuzzy mantık için matematiğin gerçek dünyaya uyarlanması diyebiliriz. Çünkü "Fuzzy Mantık"; "Keskin (Crisp) Mantığın" 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 niteleyicilere belli üyelik dereceleri atayarak gerçek dünyamıza yansıtmayı ve gerçek dünyayı daha yaklaşık temsil eden sistemler oluşturmayı başarmıştır.

Fuzzy mantığın teorik yapısından doğan çıkarımlar fuzzy denetimin uygulama alanlarına yansımıştır. Fuzzy denetimin, Kuzey Japonya'nın Sendai kentindeki metro sisteminde çok başarılı bir biçimde kullanılması fuzzy denetim uygulamalarına büyük bir ivme kazandırmıştır. 1987'de başlayan bu ivme 1990'da zirveye ulaşarak fuzzy denetimin çok büyük bir alan içerisinde kullanılmasıyla sonuçlanmıştır.

Fuzzy kümeler yeni matematiksel kavramların doğmasına, araştırma konularının ortaya çıkmasına, mühendislik uygulamalarının tasarımlanmasına yol açmıştır. Özellikle yapay zeka alanlarındaki ilginç uygulamaların ilk örnekleri günlük yaşamımıza girmeye başladı bile (Şekercioğlu, 1994).

Günümüzde otuzdan fazla ülkede "Fuzzy Kümeler" ve "Fuzzy Mantık" konusunda araştırmalar yapılmaktadır. Çin'de bu alanda uğraşan bilim adamı sayısı on binin üzerinde olup, Onun en yakın takipçisi Japonya ise uygulama açısından belirgin bir şekilde öndedir.

2.0. FUZZY KÜMELER

2.1. Temel Kavramlar

Bu bölümde Zadeh tarafindan geliştirilen "Fuzzy Kümeler Teorisi" temel alınmıştır. Fuzzy kümeler kuramının amacı belirsizlik ifade eden, tanımlanması güç veya anlamı zor kavramlara üyelik derecesi atayarak onlara belirlilik getirmektir. Belirlilik getirme yaklaşımı iki değerli kümeler kuramının, çokdeğerli kümeler kuramına dönüşümünden doğar (Türkşen, 1985).

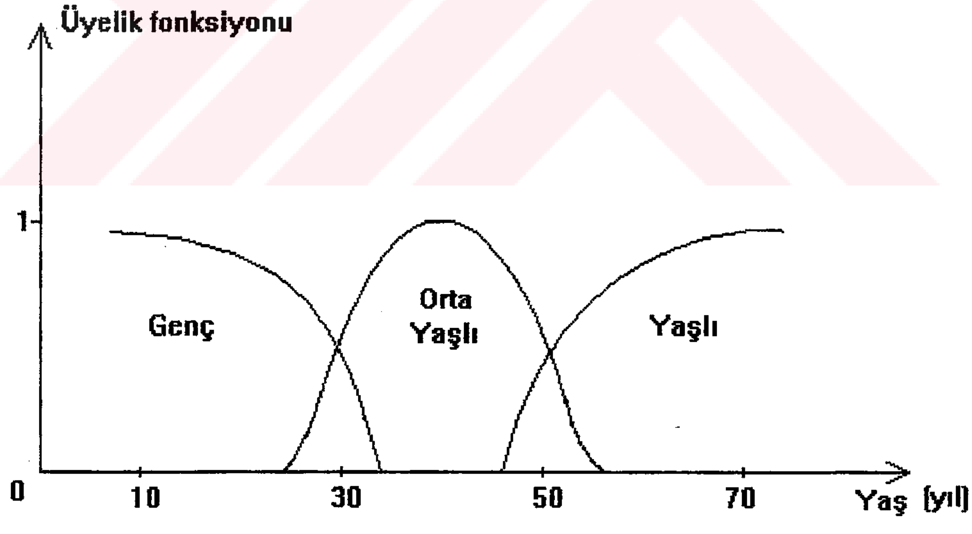
Modern mantıkta bir kümeyi oluşturan elemanlar, keskin elemanlar olup bir eleman bir kümenin ya elemanıdır ya da değildir. Bu tür kümelere de "Keskin (Crisp) Kümeler" denir. Fuzzy kümeler, belirlilik derecesi ya hep ya hiç kavramının ötesinde bir görüşten ortaya çıkar. Çoğunlukla günlük hayatta kesin sayılar ve ifadeler yerine, sınırları "bulanık" sayılar, ifadeler ve nesne sınıfları kullanırız. Keskin kümelerle, fuzzy kümelerin farkını ayırt edebilmek için aşağıdaki örneği ele alalım.

Eğer 40 yaşı orta yaş olarak ele alırsak, keskin kümelerde 30 yaşın altındaki kişiler "genç", 30-50 arası "orta yaşlı", 50 yaşın üstü de "yaşlı" kümelerine alınabilir. Buna göre 29.5 yaşındaki biri "genç" sayılırken 30.5 yaşındaki bir diğer kişi de "orta yaşlı" sayılacaktır. Aynı zamanda 35 yaşındaki bir insana pek "orta yaşlı" denemeyeceği gibi pek de "genç" sayılmaz, duruma göre belki genç belki de orta yaşlı tanımı uygun düşecektir. Belki de bu kişiyi hem "genç" hem de "orta yaşlı" olarak düşünmek isteyeceğiz. İşte fuzzy kümeler bu düşünceyi olanaklı kılar. Kümelerin keskin çizgilerle ayrılmamış olması, aralarında belirli bir örtüşüm olması, 35 yaş için istediğimizi düşünebilmemize olanak tanır.

Zadeh, küme elemanlarının üyelik derecelerini göstermek için $[0\ ,1]$ aralığındaki gerçel sayıların kullanılmasını önermiştir. Eğer bir elemanın üyelik derecesi 1 ise tümüyle o kümenin içinde olduğu, 0 ise hiç bir şekilde o kümenin bir elemanı olmadığı söylenebilir. Ya da $[0\ ,1]$ aralığında bir değer ise o üyelik derecesinde aittir diyebiliriz. Keskin kümeleri tanımlamak için kullandığımız

$$\mu_A(x) = \begin{cases} 1, & \text{ancak ve ancak } x \in A \\ 0, & \text{ancak ve ancak } x \notin A \end{cases}$$

karakteritik fonksiyonuna benzer olarak fuzzy kümeleri ifade etmek için de üyelik fonksiyonları (membership functions) kullanılır. Şekil 2.1. *yaş* için bir üyelik fonksiyonu ifade eder.



Şekil 2.1. Fuzzy Kümelerde Üyelik Fonksiyonu

2.2. Fuzzy Kümeler

X bir küme olsun. X'in bir A fuzzy altkümesi kısmi üyeliğe sahip bir alt kümedir. Bu kısmi üyelik kesin olmayan sınırlara ait kavramları anlamamıza yardımcı olur. A fuzzy altkümesi μ_A üyelik fonksiyonu ile bağlantılıdır. Öyle ki her bir $x \in X$ elemanı için $\mu_A(x) \in [0\,,1]$ x'in A'ya ait olma derecesini gösterir.

Fuzzy küme teorisi, keskin küme teorisinin bir genelleştirmesidir Başka bir deyişle, fuzzy küme teorisindeki tanımlar, teoremler ve ispatlar fuzzy olmayan kümeler için de daima doğrudur. Özetle X 'deki bir A fuzzy kümesi, $A = [(x, \mu_A(x)], x \in X$ sıralı ikililerinin bir kümesidir. $\mu_A(x)$ $[0\,,1]$ aralığında bir sayıdır.

Tanım 2.2.1.

$X = \{x_1, x_2, \cdots, x_n\}$ sonlu bir küme olsun. X'in bir fuzzy altkümesi aşağıdaki gibi ifade edilir. (Klir et al, 1988):

$$\begin{aligned} A &= \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_n)/x_n \\ &= \sum_{i=1}^{n} \mu_A(x_i)/x_i \end{aligned}$$

Tanım 2.2.2.

Eğer X sonlu bir küme değilse, buna ait fuzzy küme de şu şekilde ifade edilir. (Klir et al, 1988):

$$A = \int_X \mu_A(x)/x$$

2.3. Fuzzy Küme İşlemleri

A ve B kümeleri X'in iki fuzzy altkümesi olsun. Şimdi fuzzy kümeler için geçerli olan bazı temel işlemleri gözden geçirelim.

1. Ancak ve ancak

$$\int_X \mu_A(x)/x = \int_X \mu_B(x)/x$$

ise $A=B$ dir.

2. Ancak ve ancak

$$\int_X \mu_A(x)/x \leq \int_X \mu_B(x)/x$$

ise $A \subseteq B$ dir.

3. $\vee$ maksimum işareti olmak üzere

$$A \cup B = \int_X \left(\mu_A(x) \vee \mu_B(x)\right)/x$$

ile tanımlanır.

4. $\wedge$ minimum işareti olmak üzere

$$A \cap B = \int_X \left(\mu_A(x) \wedge \mu_B(x)\right)/x$$

ile tanımlanır.

5. *A*'nın tümleyeni

$$\overline{A} = \int_X \left(1 - \mu_A(x)\right)/x$$

dir.

6. $AB = \int_X \mu_A(x)\mu_B(x)/x$,

α herhangi bir pozitif sayı olmak üzere

$$A^{\alpha} = \int_X \left(\mu_A(x)\right)^{\alpha}/x$$

ve $\alpha\ \mathrm{Sup}_X \mu_A(x) \leq 1$ olacak şekilde α herhangi negatif olmayan bir reel sayı ise

$$\alpha A = \int_X \alpha\mu_A(x)/x$$

dir. Özel olarak, yoğunlaşma işlemi

$$\mathrm{CON(A)} = \mathrm{A}^2$$

ile tanımlanırken, genişleme işlemi

$$\mathrm{DIL(A)} = \mathrm{A}^{0.5}$$

ile ifade edilir.

7. $\mathrm{A} \oplus \mathrm{B} = \int_X 1 \wedge (\mu_A(x) + \mu_B(x))/x$ sınırlı toplamı,

$A \Theta B = \int_X 0 \vee (\mu_A(x) - \mu_B(x))/x$ sınırlı farkı tanımlar.

8. $\mathrm{V} = \{x^\alpha | x \in \mathrm{X}\}$ olmak üzere

$${}^\alpha\mathrm{A} = \int_V \mu_A(x)/x^\alpha$$

dır.

9. $\mathrm{A}_1, \cdots, \mathrm{A}_k$ sırasıyla $\mathrm{X}_1, \cdots, \mathrm{X}_k$ ların fuzzy alt kümeleri iseler, $\mathrm{A}_1, \cdots, \mathrm{A}_k$ ların Kartezyen çarpımı

$$\mathrm{A}_1 \times \cdots \times \mathrm{A}_k = \int_{\mathrm{X}_1 \times \cdots \times \mathrm{X}_k} (\mu_{A_1}(x_1) \wedge \cdots \wedge \mu_{A_k}(x_k)) / (x_1, \cdots, x_k)$$

ile gösterilir. Bu ise $\mathrm{A}_1 \times \cdots \times \mathrm{A}_k$ nın, üyelik fonksiyonu

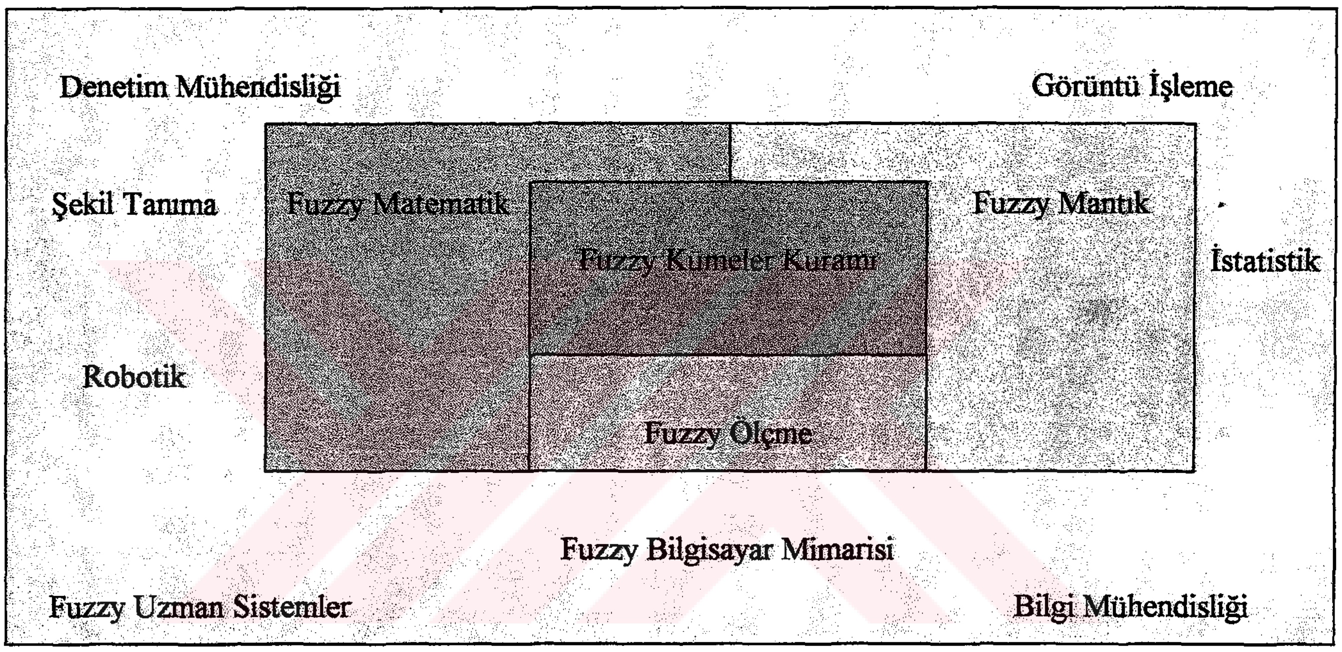
$$\mu_{A_1 \times \cdots \times A_k}(x_1, \cdots, x_k) = \mu_{A_1}(x_1) \wedge \cdots \wedge \mu_{A_k}(x_k)$$

ile ifade edilen $\mathrm{X}_1 \times \cdots \times \mathrm{X}_k$ nın fuzzy alt kümesi olduğunu gösterir.

2.4. Fuzzy Kümelerin Uygulama Alanları

Tablo2.1'de Fuzzy kümelerin kullanıldığı uygulama alanlarından bazıları verilmiştir.

Tablo 2.1. Fuzzy Kümelerin Uygulama Alanları



3.0. ÜYELİK FONKSİYONLARI

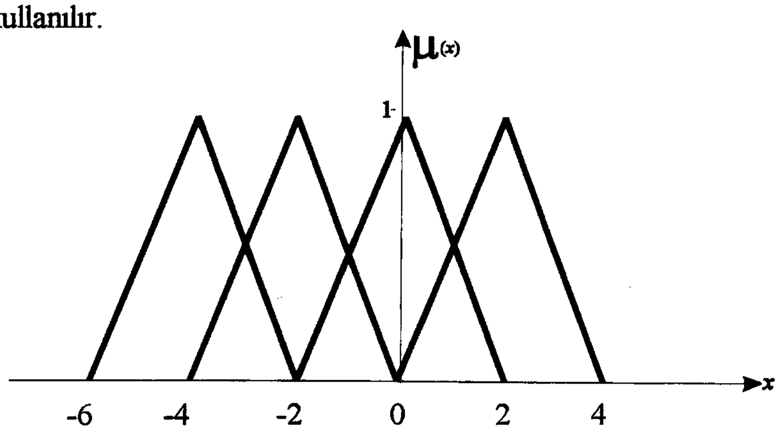
Üyelik fonksiyonları bir fuzzy kümeyi ifade ettiklerinden, tanımlanmaları fuzzy küme teorisi içinde önemli bir yer tutar. Üyelik fonksiyonlarının şekilleri önemli olduğundan bu fonksiyonların geliştirilmesi üzerine pek çok çalışmalar yapılmıştır. Üyelik fonksiyonları değişken parametreleri olan bir fonksiyon veya bir tablo olarak ifade edilebilir. Bu bölümde üyelik fonksiyonlarının şekilleri, özellikleri ve oluşturulmalarında kullanılan yöntemler üzerinde durulacaktır.

3.1. Üyelik Fonksiyonlarının Şekilleri

Üyelik fonksiyonları denetlenen sürecin özelliklerine göre üçgen, çan, yamuk, monolitik ve monotonik şeklindedirler.

3.1.1. Üçgen Üyelik Fonksiyonu:

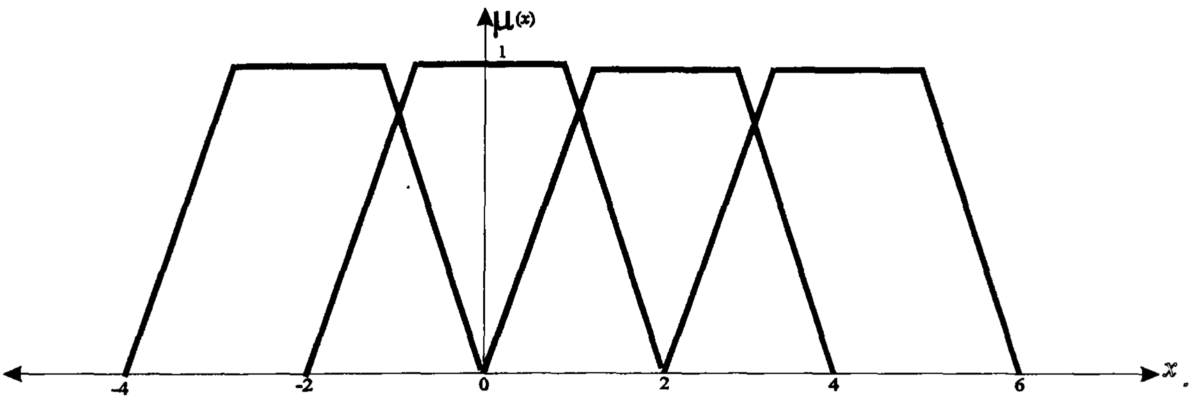
Üçgen şeklindeki üyelik fonksiyonu hem giriş hem de çıkış büyüklüğünü tanımlamak için kullanılır.



Şekil 3.1. Üçgen üyelik fonksiyonu

3.1.2. Yamuk Üyelik Fonksiyonu:

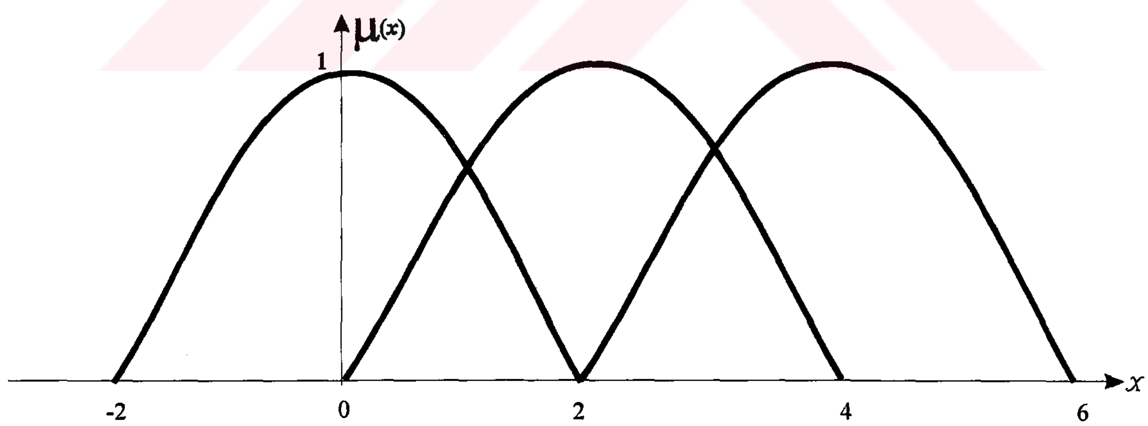
Yamuk şeklindeki üyelik fonksiyonu da yine üçgen üyelik fonksiyonu gibi giriş ve çıkış büyüklüğünü tanımlamak için kullanılır.



Şekil 3.2. Yamuk üyelik fonksiyonu

3.1.3. Çan Üyelik Fonksiyonu:

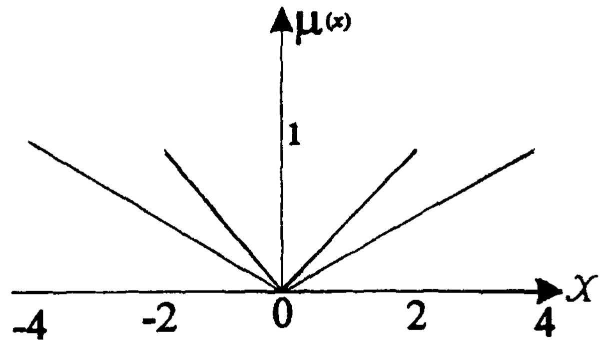
Çan şeklindeki üyelik fonksiyonu giriş ve çıkış büyüklüklerini tanımlamada kullanılan, en çok tercih edilen üyelik fonksiyonudur.



Şekil 3.3. Çan üyelik fonksiyonu

3.1.4. Monolitik Üyelik Fonksiyonu:

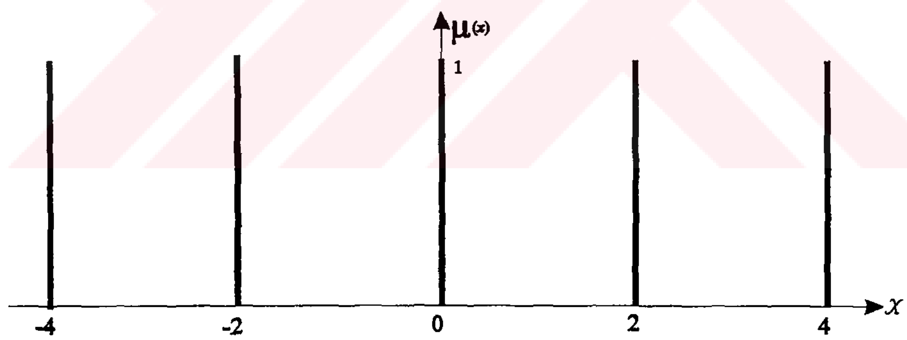
Yine giriş ve çıkış büyüklüklerini tanımlamada kullanılır. Her üyelik fonksiyonu doğru parçaları şeklinde tanımlanmıştır.



Şekil 3.4. Monolitik üyelik fonksiyonu

3.1.5. Monotonik Üyelik Fonksiyonu:

Monotonik üyelik fonksiyonu yalnızca çıkış değişkenlerini tanımlamada kullanılır. Her üyelik fonksiyonu doğru parçaları şeklinde tanımlanmıştır. Üyelik fonksiyonları birbirlerini örtmezler. Yani üyelik fonksiyonlarının merkez noktaları haricindeki değişkenlerin üyeliği sıfırdır. Bu yüzden çıkış değişkenlerinin tanımlanmasında kullanılırlar.



Şekil 3.5. Monotonik Üyelik Fonksiyonu

3.2. Üyelik Fonksiyonlarının Özellikleri

3.2.1. Üyelik Fonksiyonunun Çekirdeği:

A fuzzy kümesinde tam üyeliğe sahip elemanların oluşturduğu bölgeye üyelik fonksiyonunun çekirdeği (core) denir. Başka bir deyişle $\mu_A(x) = 1$ eşitliğini sağlayan x elemanlarının olşturduğu bölge A fuzzy altkümesinin üyelik fonksiyonunun çekirdeğidir.

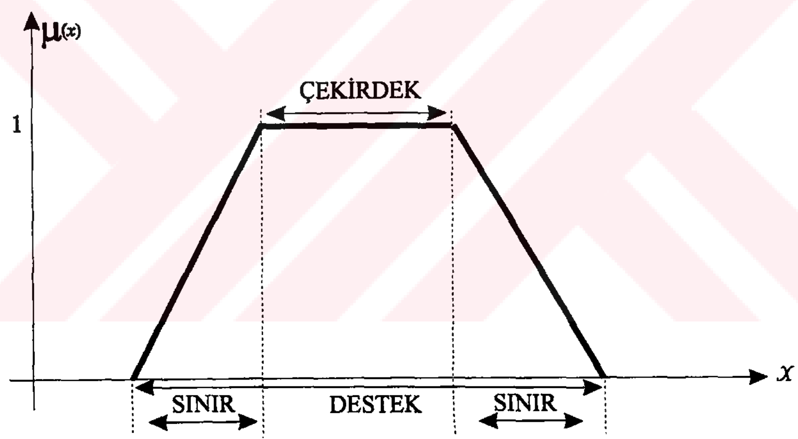
3.2.2. Üyelik Fonksiyonunun Desteği:

A fuzzy kümesinde sıfır olmayan üyeliğe sahip elemanların oluşturduğu bölgeye *A* fuzzy kümesinin üyelik fonksiyonunun desteği (support) denir. *X* deki $\mu_A(x) > 0$ noktalarının oluşturduğu bu küme *X*'in bir altkümesi olup $SuppA = \{x | x \in X,\ \mu_A(x) > 0\}$ ile gösterilir.

3.2.3. Üyelik Fonksiyonunun Sınırları:

A fuzzy kümesinde $0 < \mu_A(x) < 1$ şartını sağlayan yani ne tam ne de sıfır üyeliğe sahip olan *x* elemanlarının oluşturduğu bölgeye de, *A* fuzzy kümesinin üyelik fonksiyonunun sınırları (boundary) adı verilir.

Şekil 3.6. da bir fuzzy kümenin üyelik fonksiyonunun çekirdeği, desteği ve sınırları gösterilmektedir.



Şekil 3.6. Bir fuzzy kümenin çekirdeği, desteği ve sınırları

3.3. Fuzzy Kümelerin Temel Özellikleri

3.3.1. Bir Fuzzy Kümenin Yüksekliği:

A fuzzy kümesinin yüksekliği, *X* üzerindeki $\mu_A(x)$' in en küçük üst sınırıdır. Diğer bir ifadeyle *A*'nın yüksekliği, *A*'nın üyeliği en üst olan noktasını tanımlar.

3.3.2. Çapraz Geçiş Noktası:

A'nın çapraz geçiş noktası (crossover point) ise A'daki üyelik derecesi 0.5 olan X'in bir noktasıdır.

3.3.3. Normal ve Normal Olmayan Fuzzy Kümeler:

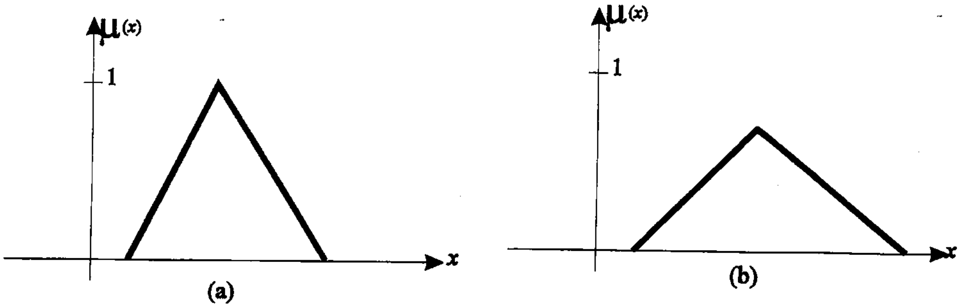
X'in en az bir elemanı için A kümesi 1 üyelik değerini alıyorsa bu kümeye normal fuzzy küme denir. Eğer fuzzy kümenin yüksekliği 1'den küçük ise bu kümeye normal olmayan fuzzy küme adı verilir. Şekil 3.7. de normal ve normal olmayan fuzzy kümelere birer örnek verilmiştir

3.3.4. Konveks Fuzzy Küme:

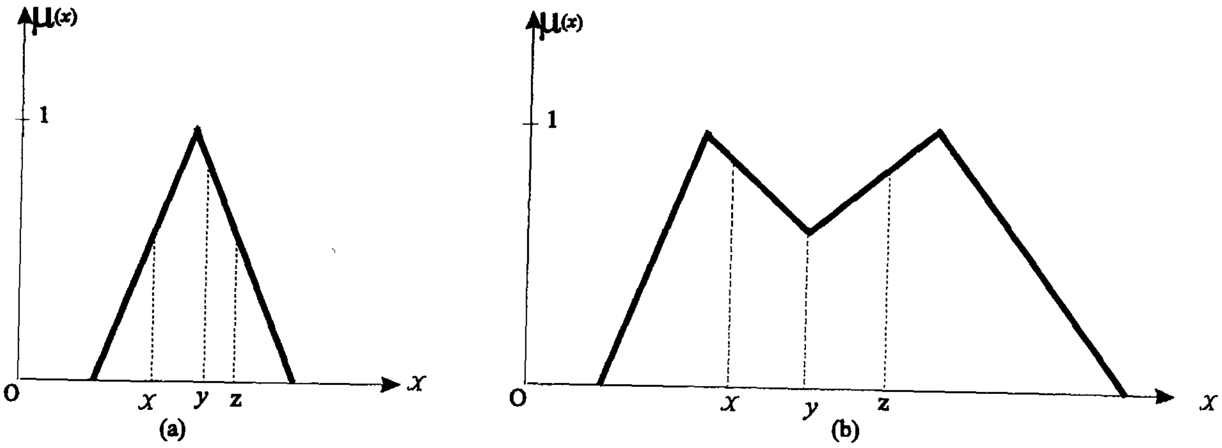
$x < y < z$ bağıntısını sağlayan A fuzzy kümesinin x, y, z elemanları için $\mu_A(y) \geq \min[\mu_A(x),\ \mu_A(z)]$ oluyorsa A kümesi bir konveks fuzzy kümedir (Zadeh, 1965). Başka bir ifadeyle A'nın artan değerleri için, üyelik değerleri monoton artan veya monoton azalan veya monoton artıp sonra monoton azalan oluyorsa A konveks fuzzy küme adını alır. Şekil 3.8. de konveks ve konveks olmayan iki fuzzy küme verilmiştir. Konveks fuzzy kümelerin bir özelliği de A ve B gibi iki konveks kümenin kesişiminin de konveks fuzzy küme olmasıdır.

3.3.5. Fuzzy Sayı:

Konveks ve normal olan bir fuzzy küme fuzzy sayı olarak adlandırılır.



Şekil 3.7. Normal (a) ve normal olmayan (b) fuzzy kümeler



Şekil 3.8. Konveks-normal fuzzy küme (a) ve konveks olmayan-normal fuzzy küme (b)

3.4. Üyelik Değeri Atamaları

Fuzzy değişkenlere üyelik değerlerinin ya da fonksiyonlarının atanabilmesi için literatürde aşağıdaki yöntemler gibi pek çok yöntem vardır (Ross, 1995).

- Sezgisel (Intuition)
- Çıkarım (Inference)
- Kademe sıralama (Rank ordering)
- Açısal fuzzy kümeler (Angular fuzzy kümeler)
- Yapay sinir ağları (Neural networks)
- Genetik algoritması (Genetic algorithms)
- İnduktif çıkarım (Inductive reasoning)
- Parçalama (Soft partitioning)
- Değişim kuralları (Meta rules)
- Fuzzy istatistik (Fuzzy statistics)

4.0. YAPAY SİNİR AĞLARI

4.1. Giriş

Yapay sinir ağları, insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır (Yamakawa, 1992). En genel anlamda bir yapay sinir ağı insan beynindeki birçok nöronun, ya da yapay olarak basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanması ile oluşan karmaşık bir sistem olarak düşünülebilir. İlk olarak insan beynindeki nöronların matematiksel modellenmesi çabalarıyla başlayan çalışmalar son on yıl içerisinde disiplinli bir şekil almıştır. Yapay sinir ağları bugün matematik, fizik, elektirik ve bilgisayar mühendisliği gibi birçok bilim dalının araştırma konusu haline gelmiştir. Pratikteki kullanımı farklı yapılarda ve formlarda bulunabilen informasyon verilerini tanımlama ve algılama üzerinedir. Mühendislik uygulamalarında ise geniş bir şekilde kullanılmasının en önemli nedenlerinden biri klasik yöntemlerle çözülemeyen problemlere etkin bir alternatif oluşturmasıdır.

4.2. Gelişimi

1940'larda Mc. Culloch ve Pitts nöronun, mantık fonksiyonlarını sağlayan basit bir eşik cihazı olarak modellenebileceğini gösterdi (Culloch, 1943). 1949'da Donald Hebb'in beyinin öğrenme mekanizması üzerine ortaya koyduğu biyolojik öğrenme kuralına göre; bir nörondan dendrit yoluyla gelen bir aksonal giriş bir darbe üretilmesine sebep olur ve böylece sonraki aksonal girişlerin darbe üretmesi olasılığı artar. Sonuç olarak da yapılan davranışın mükafatı ortaya çıkar.

Basit nöronlara dayalı bir hesaplama modeli 1958'de Rosenblatt tarafından ortaya atılmıştır. 1982 yılında artık yapay sinir ağlarının teorisi tamamlanmıştı. 1982'de J.J. Hopfield tarafından yayınlanan "Neural Networks and Physical Systems" adlı çalışması ile yapay sinir ağları alanında bir devir başlamış oluyordu.

1986'da Rumelhart ve arkadaşlarının "Parallel Distributed Processing" grubu ileri beslemeli yeni öğrenme modeli olan hatanın geriye yayılma algoritmasını (error back propagation algorithm) geliştirerek, bu konuda bazı bilim adamlarınca ortaya atılan çıkması muhtemel aksaklıkların aşılabileceğini göstermişlerdir (Rumelhart et al, 1986). Bugün yapay sinir ağı uygulamalarında bu öğrenme yöntemi ve bunun değişik varyasyonları kullanılmaktadır. Bu algoritma çok kullanılan, öğrenilmesi kolay bir ağdır.

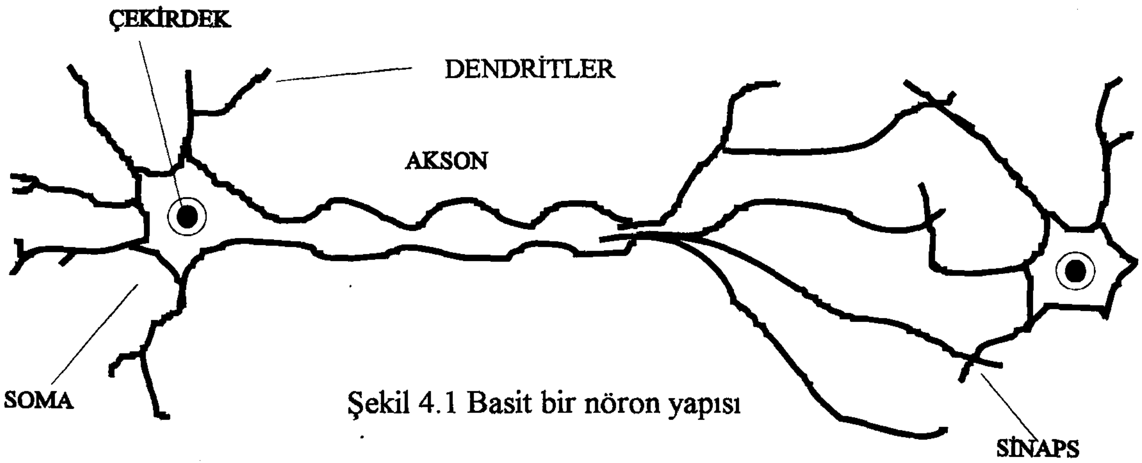
4.3. Yapay Sinir Ağlarının Tanımı ve Modeli

4.3.1. Yapay Sinir Ağının Tanımı

Yapay sinir ağı paralel dağılmış bir bilgi işleme sistemidir. Yapay sinir ağlarının temelinde, zeka gerektiren işlemlerden oluşan bir bilgi işleme işlevi vardır. Bu sistem tek yönlü işaret kanalları ile birbirine bağlanan işlem elemanlarından oluşur. Çıkış işareti bir tane olup isteğe göre çoğaltılabilir. Yapay sinir ağlarının temel düşüncesiyle, insan beyninin fonksiyonları arasında benzerlik vardır. Bu yüzden yapay sinir ağına insan beyninin modeli denilebilir. Yapay sinir ağları çevre şartlarına göre davranışlarını şekilleyebilir. Girişler ve istenen çıkışların sisteme verilmesi ile kendisini farklı cevaplar verebilecek şekilde ayarlayabilir. Son derece de karmaşık bir içyapısı vardır. Bu yüzden bugüne kadar gerçekleştirilen yapay sinir ağları, biyolojik fonksiyonların temel nöronlarını örnek alarak yerine getiren kompoze elemanlardır (Gülez, 1995).

4.3.2. Nöronun Biyolojik Yapısı ve Nöron Modeli

Bilgi işleme insanda beyinde gerçekleşir. En karmaşık sinir ağı Cerebral Cortex denilen beyin'dir. Sinir sisteminin en basit yapısı nöronlardır. Beyinde yaklaşık olarak 10^{10} sinir hücresi vardır. Yine hücre başına bağlantı sayısı ise 10^4 mertebesindedir. Vücudun



Şekil 4.1 Basit bir nöron yapısı

değişik yerleri ile bilgi alışverişi yapan nöron hücresidir. Basit bir nöron yapısı Şekil 4.1 de görülmektedir. Nöron, soma adı verilen hücre gövdesi, dendrit denilen kıvrımlı uzantılar ve somanın diğer dalları sayesinde nöronu dallarına bağlayan tek sinir fiberli aksondan oluşur. Dendritler hücreye gelen girişleri toplarlar. Dendrit tarafından alınan işaretler hücrede birleştirilerek bir çıkış darbesi üretilip üretilemeyeceğine karar verilir. Eğer bir iş yapılacaksa üretilen çıkış darbesi aksonlar tarafından taşınarak diğer nöronlarla olan bağlantılara veya terminal organlara iletilir. Beyindeki korteksde her nöronun bir karşılığı vardır. Bir nöronun çıkışı ona bağlı tüm nöronlara iletilir. Fakat korteks, işin yapılabilmesi için hangi nöron harekete geçirilecekse ona komut gönderir.

Nöronlar arasındaki bağlantılar hücre gövdesinde veya sinaps adı verilen dendritlerdeki geçişlerde olur. Sinir sistemi milyarlarca nöron ile tek bir nörondan çıkan aksonun 10000 kadar diğer nöronu bağlayan bir ağdır. Sinapslarla düzeltilen işaretleri taşıyan aksonlar ve dendritlerle içiçe geçmiş önnöronlar bir sinir ağı oluştururlar (Gülez, 1995).

Şekil 4.2 'de bir başlangıç elemanı olarak bir nöron modeli verilmektedir. $x_i, x_2, \cdots, x_n$ değişkenleri başlangıç elemanının n girişidir. $w_1, w_2, \cdots, w_n$ değişkenleri ise girişlere bağlı ağırlıklardır. w_i pozitif olduğu zaman x_i girişi arttırıcı giriş vazifesi görür. w_i negatif olduğu zaman x_i girişi azaltıcı giriş görevi yapar.

Başlangıç elemanı bu girişlerle onlara bağlı ağırlıkların çarpımlarını toplar. Bu toplam değerini verilen başlangıç değeri ile karşılaştırır. Eğer toplam başlangıç değerinden büyükse doğrusal olmayan (F) fonksiyonunu kullanarak bir çıkış hesaplar. y

çıkış işareti bu toplam ile başlangıç değeri arasındaki farkın doğrusal olmayan (F) fonksiyonudur. Bu çıkış,

x_i : giriş işareti

w_i : x_i ye bağlı ağırlık

t : başlangıç değeri

(F) : doğrusal olmayan fonksiyon; örneğin $F(s) = \frac{1}{1+e^{-s}}$ sigmoid fonksiyonu

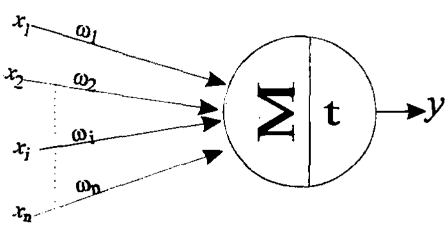
olmak üzere

$$y = F\left(\sum w_i x_i - t\right)$$

olarak ifade edilir. Doğrusal olmayan F fonksiyonu, bir modelleme seçimi ve yapay sinir ağı modelinde istenen çıkış işareti cinsinden bir fonksiyondur. Bu fonksiyon için en fazla kullanılan seçimler ise sigmoid, basamak ve rampa fonksiyonudur (Ross ,1995).

4.4. Yapay Sinir Ağlarının Yapısı

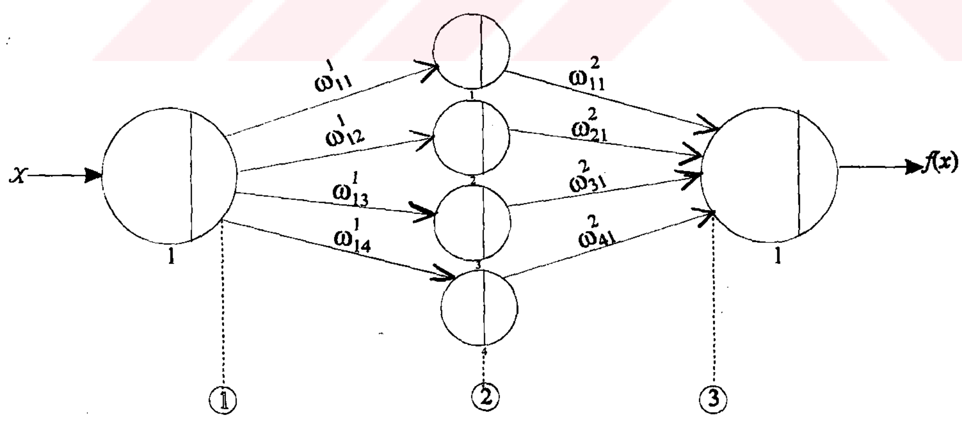
Yapay sinir ağındaki her bir düğüm hücre denilen n. dereceden doğrusal olmayan bir devredir. Düğümler işlem elemanı olarak tanımlanır. Düğümler arasında bağlantılar vardır. Her işlem elemanı istenildiği kadar giriş bağlantısı ve tek bir çıkış bağlantısı alabilir. Fakat bu tek çıkış birçok hücreyi besleyebilir. İşlem elemanının çıkışı istenilen matemetiksel tipte olabilir. Giriş işaretleri yapay sinir ağına bilgi taşır. Sonuç ise çıkış işaretlerinden alınabilir (Gülez, 1995).



Şekil 4.2. Nöron modeli

Yapay sinir ağı birtakım kümelere ayrılabilir. Bu alt kümelerdeki elemanların transfer fonksiyonları aynıdır. Bu küçük gruplara katman (layer) adı verilir. Yapay sinir ağı katmanların birbirlerine hiyerarşik bir şekilde bağlanmasından oluşmuştur. Dışardan alınan bilgi giriş katmanı ile taşınır. Transfer fonksiyonları yoktur. Yapay sinir ağı transfer fonksiyonu öğrenme kuralı ile giriş çıkış arasındaki bağıntıya göre ayarlanır. Özetle yapay sinir ağlarından beklenen gerçek dünyadaki nesneler ile biyolojik sinir ağının yaptığı işlevi, benzer bir yolla yerine getirmesidir.

Şekil 4.3. x tek-giriş işaretli ve y tek-çıkış işaretli sistem için bir basit yapay sinir ağını gösterir. Birinci katmanda bir tek eleman vardır. Bu ise bir tek giriş içerir, fakat çıkışlarını ikinci katmandaki diğer dört elemana gönderir. İkinci katmandaki elemanların tümü tek-giriş ve tek-çıkış elemanlarıdır. Üçüncü katman dört giriş içeren tek bir elemandan oluşur ve sistem için çıkışı hesaplar. Bu $(1\times 4\times 1)$ Yapay Sinir Ağı olarak adlandırılır. Sayılar her katmandaki elemanların sayısını gösterirler. Giriş ve çıkış katmanları dışındaki katmanlar gizli katmanlar adını alırlar. Üçten fazla katman içeren bir sistem birden fazla gizli katmana sahip olur (Ross, 1995).



Şekil 4.3. $1\times 4\times 1$ Yapay sinir ağı (Burada w^i_{jk}, *i.* katmanın *j.* elemanını *(i+1).* katmanın *k.* elemanına taşıyan ağırlığı gösterir.)

4.5. Yapay Sinir Ağının İşleyişi

Nöron sistemler, problemleri aldıkları verilerin (işaretlerin) yapısına uygulayarak çözerler. Bunu yaparken (x,y) giriş ve çıkış verilerinin eğitme veri kümesini ve kontrol veri kümesini kullanmak gerekir. Farklı katmanlardaki elemanları birbirlerine bağlayan yolara ait w^i_{jk} ağırlıklara rastgele atanarak problemin çözümüne başlanılır. Daha sonra eğitme veri kümesinden x girişi yapay sinir ağına gönderilir. Yapay sinir ağı $f(x)$ çıkış değerini hesaplar. Bu değer gerçek y değeri ile karşılaştırılır. δ hata ölçümü, bu iki çıkış değer kullanılarak,

$$\delta = y - f(x)$$

olarak hesaplanır. Hata ölçümü yapay sinir ağının son katmanıyla ilgilidir. Daha sonra bu hatayı gizli katmanlardaki elemanlara dağıtmak için geriye yayılma (back-propagation) yöntemi kullanılır.

Gizli katmanlardaki elemanların hata ölçümü ise şu şekilde gerçekliştirilir: δ_j j. ci elemanla ilgili hata olsun. w_{nj} n. inci elemandan j. inci elemana bağlanan yol boyunca olan ağırlık olsun. I da n birimine bir giriş olsun. n elemanı için hata, $F(I) = 1/(1+e^{-I})$ sigmoid fonksiyonu için

$$F'(I) = F(I)(1 - F(I))$$

olmak üzere

$$\delta_n = F'(I)\, w_{nj}\delta_j$$

formülüyle hesaplanır. Bundan sonra yapılacak iş son çıkışa daha yakın bir sonuç elde edebilmek için farklı elemanları birbirine bağlayan w^i_{jk} ağırlıkları değiştirilir. Bunu yaparken de hata ölçümlerinden yararlanılır. Birleşik ağırlıklar,

α : öğrenme sabiti

δ : hata ölçümü

x_i : giriş işareti

olmak üzere

$$w_i(yeni) = w_i(eski) + \alpha\,\delta\,x_i$$

formülü kullanılarak düzeltilirler.

Düzeltilmiş ağırlıklar alınarak x_i giriş değeri yapay sinir ağından tekrar geçirilir. Varsa hata tekrar hesaplatılır. Bu yöntem son çıkışın hata değeri kullanıcının istediği limitleri alıncaya kadar tekrarlanır. Bundan sonra yapay sinir ağı bir sonraki giriş-çıkış verilerini kullanır. Bu yöntem eğitme veri kümesindeki tüm veriler için sürdürülür. Bu yöntem yapay sinir ağının, giriş-çıkış veri kümeleri arasındaki doğrusal olmayan bağıntıyı temsil etmesine yardımcı olur. Son olarak da kontrol veri kümesi yapay sinir ağının doğrusal olmayan bağıntıyı ne kadar gerçeklediğini görmek için kullanılır (Ross, 1995).

4.6. Yapay Sinir Ağlarının Kullanımının Nedenleri

1. Yapay sinir ağları verilerden hareketle, bilinmeyen ilişkileri akıllıca hemen ortaya koyabilmektedir. Bu özellik ise uygulama açısından son derece önemlidir. Ayrıca veri toplama için bir ön sorgulama ya da açıklama gerekmemektedir.
2. Bir örnekten hareketle diğer örneklerdeki benzerlikleri doğru olarak anlayabilirler. Günlük hayattaki verilerde sürekli olarak gürültü ve bozucu etkiler mevcut olduğundan genelleştirebilme yapabilmesi de diğer bir önemli özelliğidir.
3. Yapay sinir ağları son derece paralelliğe sahiptir. Aynı anda birbirinden bağımsız işlemleri çok hızlı yürütebilirler.
4. Yapay sinir ağları doğrusal değildir. Bu özellikleri dolayısıyle problemleri doğrusal tekniklerden daha doğru çözerler (Gülez, 1995).

5.0. İSTANBUL MENKUL KIYMETLER BORSASI (İMKB)

5.1. Borsa'nın Tanım ve Görevleri

5.1.1. Borsa'nın Tanımı

Yönetmeliğe göre, borsalar, borsada işlem görmesi kabul edilen menkul kıymetlerin alım-satımının Kanun Hükmünde Kararnamede yazılı esaslar dairesinde, belli kurallara göre düzen içinde yapılmasını sağlayan, oluşan fiyatların ilanına yetkili, tüzel kişiliği haiz kamu kurumlarıdır. Borsalar Sermaye Piyasa Kurulu'nun teklifi üzerine Bakanlığın izni ile kurulur.

Kanuna göre, sermaye piyasası araçların işlem göreceği borsalar, menkul kiymetlerin ve diğer sermaye piyasası araçlarının güven ve istikrar içinde, serbest rekabet şartları altında kolayca alınıp satılabilmesini sağlamak ve oluşan fiyatları tespit ve ilan etmekle yetkili olarak kurulan, tüzel kişiliğe haiz kuruluşlardır.

5.1.2. İstanbul Menkul Kıymetler Borsası (İMKB)

Menkul Kıymetler Borsaları Hakkında 91 sayılı Kanun Hükmünde Kararnamede öngörülen görevleri yerine getirmek üzere kurulmuş, yetkilerini kendi sorumluluğu altında bağımsız olarak kullanan ve Sermaye Piyasası Kurulu'nun gözetim ve denetimi altında olan tüzel kişiliği haiz bir kamu kurumudur.

İMKB, resmi çalışma günlerinde faaliyette bulunur. İMKB aynı zamanda bir meslek kuruluşudur. Her meslek kuruluşu gibi üyeleri vardır.

5.1.3. Borsa'nın Üyeleri

Borsa'ya üye olabilecek, yani Borsa'da işlem yapabilecekler, Sermaye Piyasası Kurulundan faaliyetleri için yetki belgesi almış olan:

1. Yatırım ve kakınma bankaları,
2. Ticari bankalar,
3. Aracı kurumlardır.

Borsanın üst karar organı üyelerden oluşan genel kuruldur. Genel kurul, Yönetim Kurul'unu seçer, Borsa Başkanı aynı zamanda yönetim kurulu başkanıdır ve hükümetçe müşterek kararname ile tayin edilir.

5.1.4. İMKB'nin Görev ve Yetkileri

İMKB'nin başlıca görev ve yetkileri aşağıda gösterilmiştir.

- Menkul kıymetlerin Borsa kotuna alınması ile ilgili başvuruları, İç Yönetmelik'te belirtilen esaslar dahilinde incelemek, ek bilgi ve belgeler istemek, başvuruları değerlendirmek ve karara bağlamak.
- Kanuni gerekler yerine getirilerek para, kambiyo ve kıymetli maden ve taşlar ile vadeli işlemlerle ilgili piyasalar açmak.
- Borsa'da işlem görebilecek menkul kıymetler için türlerine göre menkul kıymetler pazarları oluşturmak, bu pazarlarda işlem görecek menkul kıymetleri belirlemek ve Borsa bülteninde yayınlamak, pazarlara Borsa binasında yer tahsis etmek.
- Borsa pazarlarında yapılan işlemler sonucunda oluşan fiyatları ve bu fiyatlardan yapılan toplam işlem miktarlarını seans bitiminde ilan etmek.
- Borsa'da yapılan alım, satım işlemlerini güven ve istikrar içinde serbest rekabet şartları altında kolayca ve düzenli bir şekilde yürütülmesini sağlamak, bu kuralların dışına çıkan Borsa üyelerine İç Yönetmelik'te belirtilen müeyyideleri uygulamak.
- Borsa'da olağan dışı menfi gelişmelerin meydana gelmesi halinde, mevzuatın verdiği yetkiler içinde gerekli önlemleri almak.(İMKB, 1997)

5.2. Endeksler

5.2.1. Giriş

İstatitikte bir gösterge olarak kullanılan endeks, bir veya birkaç değişkenin zaman, mekan veya diğer özelliklere göre gösterdiği değişmelerin ölçüsü olarak tanımlanabilir. Endeksler, karmaşık olayların tek bir sayıya indirgenmesini sağlayan, olaylar ve sonuçları hakkında yaklaşık bilgi verebilen araçlardır.

Endeksler, zaman içerisinde süreklilik, dolayısıyle karşılaştırabilme imkanı sağlarlar. Böylece endekse konu olan değişken veya değişkenlerin yönü, değişimi veya gidişi belirlenebilir. Bilimsel araştırmalarda geniş bir kullanım alanı bulunan endekslere en yaygın olarak iktisat ve işletme ile ilgili konularda başvurulmaktadır. Alternatif yatırım araçlarının getirilerinin ölçülmesinde ve karşılaştırılmasında endekslerden yararlı bilgiler sağlanabilmektedir.

Hisse senedi piyasasının genel bir göstergesi olan hisse senedi fiyat endeksleri endeks kapsamındaki hisse senetlerinin fiyatlarını baz alarak piyasa performansı hakkında genel bir bilgi verir. Hisse senedi fiyat endeksleri, genellikle piyasanın anlık durumunu yansıtır.

Hisse senetleri fiyatlarındaki değişmeleri ölçen endeksler genel olarak iki çeşittir. Bir kısmı endeks kapsamındaki hisse senetlerinin nisbi önemini dikkate almadan, genel fiyat düzeyini, bütün hisse fiyatlarına eşit ağırlık veren aritmetik veya geometrik ortalama esaslı basit endekslerdir. Bir kısmı ise, hisse senetleri fiyatlarının tedavüldeki hisse sayısıyla ağırlıklandırıldığı, çok sayıda hisse senedini ihtiva eden karmaşık endekslerdir. İMKB endeksi de karmaşık endeksler kategorisindedir.

İMKB Endeksi, 100 başlangıç değeri ile birinci pazarda işlem gören 40 şirketin 1986 yılı Ocak ayı ortalama piyasa değerleri baz alınarak oluşturulmuştur. 26 Ekim 1987 tarihine kadar haftalık hesaplanan endeks bu tarihten itibaren günlük olarak hesaplanmaya başlanmıştır. Endeks 1989 yılı sonuna kadar şirketlerin piyasa değerlerinin baz döneme göre olan değişimlerinin aritmetik ortalaması alınarak hesaplanmıştır. İMKB Bileşik Endeksi, 1990 yılında şirketlerin piyasa değerleri üzerinden piyasa değeri ağırlıklı olarak hesaplanmaya, 1991 yılından itibaren de şirketlerin halka açık bölümlerinin piyasa değerleri dikkate alınarak, piyasa değeri ağırlıklı olarak hesaplanmaya başlanmıştır. İşlem gören şirket sayısındaki artışa bağlı olarak endeks kapsamına alınan şirket sayısı 100 şirkete ulaşmış ve bileşik endeks İMKB 100 olarak tanımlanmıştır.

5.2.2. İMKB Endeksi'nin Özellikleri

1. İMKB Endeksi piyasa değeri (şirketlerin halka açık bölümlerinin) ağırlıklıdır.
2. İMKB Endeksi süreklidir.
3. İMKB Endeksinin başlangıç (baz) değeri 100'dür.
4. Endeks hesabında tescil edilen fiyatlar dikkate alınmaktadır.
5. İMKB Bileşik Endeksi geniş kapsamlıdır ve piyasanın büyük bir bölümünü temsil etmektedir.
6. Bileşik Endeks 1. seans (10:00 - 12:00) ve 2. seans (14:00 - 16:00) süresince her 15 saniyede bir yeniden hesaplanarak minimum, maksimum ve o anki endeks değeri Reuters, Telerate vb. veri dağıtım kuruluşları ve TRT-Teleteks sistemi ile yayınlanmaktadır (İMKB, 1997).

5.3. Bazı Borsa Terimleri ve Açıklamaları

Ağırlıklı Ortalama Fiyat : Bir sonraki seansa ait baz fiyatın hesaplanmasına esas oluşturan hisse senedinin miktar ağırlıklı ve küsüratsız fiyatıdır.

Baz Fiyat : Bir hisse senedinin seans içinde işlem görebileceği üst ve alt fiyat limitlerini ve fiyat adımlarının belirlenmesine esas oluşturan fiyattır.

Bileşik Endeks : Endeks kapsamındaki hisse senetlerinin fiyatları ve halka açıklık oranları baz alınarak hesaplanan ve hisse senedi piyasasının genel bir göstergesi olan ölçümdür.

Fiyat Adımı : Her hisse senedi fiyatı için bir defada gerçekleşebilecek en küçük fiyat değişimidir.

Fiyat Aralığı : Bir hisse senedi için, seans içinde önerilebilecek en düşük ve en yüksek fiyatlar o hisse senedi için fiyat marjını oluşturur. Mevcut uygulamada bu limitler baz fiyatının %10 altı ve üstüdür.

Hisse Senedi : Anonim ortaklıklar tarafından çıkarılan ve anonim ortaklığın sermayesine belirli bir katılma payını temsil eden, yasal şekil koşullarına uygun olarak düzenlenmiş, kıymetli evraktır.

İşlem Hacmi : Tüm hisse senetleri için gerçekleşen işlemlerdeki her emrin içerdiği hisse senedi sayısı ile işlem fiyatının çarpılarak elde edilen toplam.

Kapanış Fiyatı : Bir seansta Borsa kaydına alınan (tescil edilen) en son işlemin fiyatıdır.

Seans : Borsada işlemlerin başlaması ve bitmesi arasında geçen süre.

Taban Fiyatı : Hisse senetlerinin bir seans içinde işlem görebileceği en düşük fiyattır. Her hisse senedi için fiyat ve fiyat adımı gözönüne alınarak ayrı olarak hesaplanır.

Tavan Fiyatı : Hisse senetlerinin bir seans içinde işlem görebileceği en yüksek fiyattır. Yine her hisse senedi için fiyat ve fiyat adımı gözönüne alınarak ayrı olarak hesaplanır. (İMKB, 1995)

6.0. KÜMELEME

6.1. Giriş

Deneysel verilerle çalışırken giriş değerleri ve çıkış değerleri arasında ilişkiler geliştiririz. Yaptığımız deney sayısı arttıkça ilişkinin sınıflandırılabilecek bir yapı oluşturduğunu görebiliriz. Bu yapıyı tespit ettikten sonra verileri benzer özelliklerine, modellerine veya diğer özelliklerine göre sınıflandırabiliriz. Buna kümeleme (clustering) adı verilir.

6.2. Keskin Kümelerde Kümeleme

Bu bölümde keskin kümelerde kullanılan c-ortalama kümeleme tekniğini gözönüne alacağız. Bu yöntem n boyutlu Euclid uzayında tanımlı veri noktalarınının geometrik kapanışını ifade edip onları çeşitli kümelere atamak ve kümeler arasındaki uzaklığı ifade etmede kullanılır.

X in c-bölüntüsünün $\{A_i, i = 1,2,\cdots,c\}$ küme ailesini gözönüne alalım. c, kümelemek istediğimiz verilerin, kümelenecek küme sayısı, $X = \{x_1, x_2, \cdots, x_n\}$ veri örnekleri evreninin sonlu bir kümesi olmak üzere bu bölüntülere aşağıdaki küme işlemleri uygulanacaktır.

$$\bigcup_{i=1}^{c} A_i = X \tag{6.1}$$

$$\forall\ i \neq j \text{ için } A_i \cap A_j = \varnothing \tag{6.2}$$

$$\forall\ i \text{ için } \varnothing \subset A_i \subset X \tag{6.3}$$

Burada $2 \leq c < n$ dir. $c = 1$ ise tüm verilerin aynı kümede olacağı aşikardır. Böylece kümelemeye ihtiyaç duyulmamaktadır. (6.1) denklemi tüm küme elemanlarının evreni oluşturduğunu ifade eder. (6.2) ise bir verinin birden fazla kümeye ait olamayacağını gösterir. (6.3) ifadesi de bir kümenin ne boş küme ne de evren olabileceğini ifade eder.

$$\mu_{A_Y}(x_k) = \begin{Bmatrix} 1 & x_k \in A_i \\ 0 & x_k \notin A_i \end{Bmatrix} \quad (6.4)$$

karekteristik fonksiyonu verildikten sonra yukarıda verilen (6.1), (6.2) ve (6.3) denklikleri

$$\forall k \text{ için } \quad \bigvee_{i=1}^{c} \mu_{A_i}(x_k) = 1 \quad (6.5)$$

$$\forall k \text{ için } \quad \mu_{A_i}(x_k) \wedge \mu_{A_j}(x_k) = 0 \quad (6.6)$$

$$\forall i \text{ için } \quad 0 < \sum_{k=1}^{n} \mu_{A_i}(x_k) < n \quad (6.7)$$

olarak ifade edilebilecektir. (6.5) ve (6.6) x_k 'nın c sınıflarından sadece birine ait olduğunu ifade ederken (6.7) hiçbir kümenin boş küme ve evren (X) olamayacağını ifade eder.

i. ci kümedeki j. ci veri noktalarının üyeliğini kolaylık olması bakımından $\mu_{ij} \equiv \mu_{A_i}(x_j)$ olarak gözönüne alacağız. μ_{ij} $(i = 1,2,\cdots,c;\ j = 1,2,\cdots,n)$ elemanlarından oluşan $(c \times n)$ boyutlu U matrisini tanımlayalım. X için c- bölüntü uzayını aşağıdaki matrisle tanımlayabiliriz.

$$M_c = \left\{ U \middle| \mu_{ij} \in \{0,1\}, \sum_{i=1}^{c} \mu_{ik} = 1,\ 0 < \sum_{k=1}^{n} \mu_{ik} < n \right\} \quad (6.8)$$

$U \in M_c$ yi sağlayan herhangi bir U matrisi c-bölüntüsüdür. Herhangi bir c-bölüntüsünün M_c kardinalitesi

$$\eta_{M_c} = \left(\frac{1}{c!}\right)\left[\sum_{i=1}^{c}\binom{c}{i}(-1)^{c-i}\cdot i^n\right] \tag{6.9}$$

formülüyle hesaplanır. Buradan birden fazla matris bulunacağı görülmektedir. Bu durumda optimal U matrisini bulabilmek için c-ortalama kümeleme (clustering) yöntemini uygulamak gerekir. Uzaklığı ifade etmek için Euclid normu kullanılır. U bölüntü matrisi, v de kümeleme merkezlerinin vektörü olmak üzere

$$J(U,v) = \sum_{k=1}^{n}\sum_{i=1}^{c}\mu_{ik}(d_{ik})^2 \tag{6.10}$$

ile ifade edilen algoritma kullanılır. Burada d_{ik} Euclid uzaklık ölçüsü olmak üzere (*m*-boyutlu uzayda)

$$d_{ik} = d(x_k - v_i) = \|x_k - v_i\| = \left[\sum_{j=1}^{m}(x_{kj} - v_{ij})^2\right]^{1/2} \tag{6.11}$$

formülünden hesaplanmaktadır. Her bir veri örneği R^m-uzayında *m* koordinat ile ifade edildiğinden kümeleme merkezlerinin de aynı uzayda ifade edilebilmesi için *m* koordinat gerekir. Bundan dolayı, i.ci kümeleme merkezi *m* uzunluklu bir vektördür. Bu vektör, $v_i = \{v_{i1}, v_{i2}, \cdots, v_{im}\}$ olmak üzere vektörün j.ci kordinatı

$$v_{ij} = \frac{\sum_{k=1}^{n}\mu_{ik}\cdot x_{kj}}{\sum_{k=1}^{n}\mu_{ik}} \tag{6.12}$$

formülüyle hesaplanır. Geriye J fonksiyonu için minimum olan Optimal U^* bölüntüsünün bulunması kalır. Bu ise

$$J(U^*,v^*) = \min_{U \in M_c} J(U,v) \tag{6.13}$$

olarak ifade edilir (Ross, 1995). U^* matrisini bulunması pratikte oldukça zor bir problemdir. Bunun için başlangıç olarak olabilir bir U matrisi gözönüne alınır. Bu matristen kümelerin merkezleri hesaplanır. Bu kümelerden de kümelerdeki verilerin yeni üyelik değerleri hesaplanır. Bu değerler kabul edilmiş ilk değerlerle karşılaştırılır. Bu işlemler istediğimiz sonucu elde edene kadar sürdürülür. Ross'un eserinde kullandığı Bezdek tarafından geliştirilen bu yöntemin daha anlaşılır bir algoritması kısım 6.3'de verildiği şekildedir. Bunla ilgili geliştirdiğimiz pascal programlama dilinde yapılan program ise EK 1'de verilmiştir.

6.3. Algoritma

1. $U^{(0)} \in M_c$ başlangıç matrisini belirle.
2. Verileri gir.
3. $v_i^{(r)}$, c merkezi vektörlerini bul.
4. $\mu_i^{(r+1)} = \begin{cases} 1 & d_{ik}^{(r)} = \min\{d_{jk}^{(r)}\} \ (\forall j \in c) \\ 0 & \text{aksi takdirde} \end{cases}$ $(\forall\, i, k$ için) olacak şekilde değişmiş $U^{(r)}$ matrisini bul.
5. $\|U^{(r+1)} - U^{(r)}\| = 0$ olunca $U^{(*)}$ 'ı yaz ve dur, aksi takdirde $r = r+1$ al ve 3.cü basamağa geri dön.

7.0. İSTANBUL MENKUL KIYMETLER BORSASI VERİLERİNE BAĞLI OLARAK FUZZY ÜYELİK FONKSİYONUNUN BULUNMASI

İstanbul Menkul Kıymetler Borsası verilerini kullanarak herhangi bir seans sonunda o seansa ait verileri yapay sinir ağını kullanarak, eğitip herhangi bir hisse senedi için bir sonraki seanstaki kapanış fiyatının kestirimini hesaplamaya çalıştık. Bunun için de yapay sinir ağlarını kullanarak, fuzzy üyelik fonksiyonu oluşturup bunun kestirimini yapmaya çalıştık. Problemimize geçmeden önce yapay sinir ağında fuzzy mantığı kullanmaya neden gereksinim duyduğumuzu ve yapay sinir ağlarını kullanarak üyelik fonksiyonlarının nasıl elde edilebileceği üzerinde duralım.

7.1. Giriş

Girişlerin ve bunlara karşılık gelen çıkışların veri kümelerinin ve giriş ve çıkış arasındaki ilişki doğrusal olmayan ya da tam olarak bilinmeyen sistemlerde fuzzy mantık kullanılarak, giriş ve çıkış veri kümelerini farklı fuzzy sınıflarına ayırabiliriz. Bundan başka, doğada dinamik olan sistemler için fuzzy üyelik fonksiyonları tekrar tekrar güncelleştirilmelidir. Bu tip sistemlerde, değişmeleri sağlamak için kendi kendine değişim yaptığından, yapay sinir ağını kullanmak en uygun yöntem olacaktır.

7.2. Yapay Sinir Ağını Kullanarak Fuzzy Üyelik Fonksiyonlarının Elde Edilişi

Burada bir giriş veri kümesinin fuzzy sınıfları için fuzzy üyelik fonksiyonlarının elde edilişi üzerinde duracağız. Belli sayıda giriş veri değeri seçilir ve bunlar eğitme ve kontrol olmak üzere iki ayrı veri kümesine ayrılır. Eğitme veri kümesi yapay sinir ağını eğitmek için kullanılır. Kontrol veri kümesi de eğitilen yapay sinir ağını test etmede kullanılır. Veri noktaları ilk olarak kümeleme (clustering) yöntemiyle farklı sınıflara ayrılır (Bölüm 6).

Veri noktalarının R_1 ve R_2 gibi iki sınıfa ayrıldığını gözönüne alırsak R_1 sınıfına ait bir veriye R_1 sınıfında 1 üyelik değerini R_2 sınıfında ise 0 üyelik değerini veririz. Bütün veriler için bu işlem tekrarlanır ve veri noktalarına başlangıçta ait oldukları sınıflar için 1 üyelik değerleri atanır. Veri koordinatları ve üyelik fonksiyonu arasındaki ilişkiyi eğitmek için bu veri noktalarını ve bunlara karşılık gelen üyelik fonksiyonlarını kullanan yapay sinir ağı oluşturulur. Sırasıyla tüm veriler yapay sinir ağından geçirilerek eğitilirler. Daha sonra kontrol veri kümesi kullanılarak yapay sinir ağının performansı ölçülür. Daha sonra istenilen herhangi bir veri değeri için eğitilmiş yapay sinir ağı değerleri kullanılarak bu verilerin yukarıda verdiğimiz sınıflardaki gerçek üyelik değerlerini elde ederiz (Ross, 1995).

7.3. İstanbul Menkul Kıymetler Borsasında Yapay Sinir Ağlarını Kullanarak Düşük ve Kapanış Değerleri Arasındaki İlişkiyi Bulduran Üyelik Fonksiyonunun Elde Edilişi

Problemimizin çözümü için İstanbul Menkul Kıymetler Borsasından seans seans ve haftalık veriler aldık. Bir seans bitiminde aldığımız verilerden rastgele olarak seçtiğimiz otuz tanesini yukarıdaki yöntemeden biraz farklı bir yolla yapay sinir ağında eğittik. Amacımız aracı kuruluşların, müşterileri için, bu eğitilmiş yapay sinir ağı değerlerini kullanarak istedikleri herhangi bir hisse senedinin bir sonraki seanstaki kapanışının düşük değerden mi yoksa yüksek değerden mi olacağının tahminini yapabilmesi ve kapanış anında ona göre hareket etmelerini sağlamaktı. Yaptığımız çalışmalarda çok yüksek bir performans elde ettik.

İlk önce rastgele seçtiğimiz 30 hisse senedinin düşük, yüksek ve kapanış değerlerini fuzzy kümelerde çalışacağımızdan normalize ettik. Daha sonra bu değerleri altıncı bölümde ele aldığımız kümeleme (clustering) yöntemiyle (EK1'de verilen programı kullanarak) iki sınıfa ayırdık. Bunlardan birincisini "düşük" ikincisini ise "yüksek" sınıfı olarak aldık. Bununla ilgili veriler ise Tablo 7.1. de verilmiştir.

İstanbul Menkul Kıymetler Borsası'nın 9 Mayıs 1997 Cuma günü 1. Seansına ait, yapay sinir ağını eğitmede ve performansını test etmede kullandığımız veriler Tablo 7.2. de verilmiştir. Aynı gün ikinci seansta istenilen bir hisse senedinin kapanışının düşük değerden mi yoksa yüksek değerden mi olacağını, birinci seanstaki kapanış değeriyle ikinci seans içerisinde aracı kuruluşun o hisse senedinin seyrine göre aldığı en düşük değer giriş verisi olarak alınarak, eğitilen yapay sinir ağından geçirilerek bulunur. Bunlarla ilgili verilerimiz de Tablo 7.3. de verilmiştir. Tahminlerimizin ne kadarının doğru olduğu görebilmek için de, üzerinde araştırma yaptığımız senetlerin ikinci seanstaki değerleri Tablo 7.4. de verilmiştir.

Tablo 7.1. Kümeleme Tekniği ile Eğitilen 30 verinin Düşük ve Yüksek Sınıflarına Ayrılışı

Senet No:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Düşük	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
Yüksek	1	1	0	1	1	1	0	1	0	1	1	1	0	1	1

Senet No:	16	17	18	19	20	21	22	23	24	25	25	27	28	29	30
Düşük	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0
Yüksek	1	0	1	1	0	0	1	1	1	1	1	0	1	1	1

Tablo 7.2. 9 Mayıs 1997 Cuma Günü 1.ci Seansa Ait Normalize Edilmiş Veriler

Hisse Senedi Numarası	En Düşük Fiyat	En Yüksek Fiyat	Kapanış Fiyatı
1	0.01275	0.01325	0.013
2	0.0975	0.105	0.1
3	0.44	0.45	0.445
4	0.2275	0.2325	0.2325
5	0.27	0.28	0.275
6	0.1375	0.14	0.14
7	0.5	0.51	0.51
8	0.2125	0.225	0.2175
9	0.57	0.58	0.58
10	0.11	0.1125	0.1125
11	0.105	0.1075	0.1075
12	0.1075	0.1125	0.1075
13	0.32	0.32	0.32
14	0.1425	0.1475	0.1425
15	0.145	0.15	0.1475
16	0.043	0.046	0.044
17	0.65	0.65	0.65
18	0.091	0.093	0.091
19	0.265	0.275	0.265
20	0.365	0.365	0.365
21	0.5	0.53	0.53
22	0.26	0.275	0.265
23	0.225	0.2325	0.2275
24	0.2125	0.2175	0.2125
25	0.12	0.125	0.125
26	0.01925	0.0195	0.01925
27	0.49	0.53	0.49
28	0.038	0.039	0.038
29	0.1375	0.14	0.14
30	0.2025	0.2025	0.2025

Tablo 7.3. Bazı Hisse Senetlerinin İkinci Seanstaki Kapanışı İçin Tahminler (Veriler Normalize Edilmiştir)

Hisse Senedi Adı	2.ci Seans Sırasında Aldığı En Düşük Değer	1.ci Seans Sonundaki Kapanış Fiyatı	Düşük Oranı	Yüksek Oranı
Alarko Holding	0.1825	0.1725	0	1
Borusan	0.069	0.07	0	1
Brisa	0.51	0.51	1	0
Ege Biracılık	0.365	0.36	0.998	0.002
Ereğli D. Çelik	0.1465	0.1425	0	1
İst. Motor Pis.	0.1475	0.15	0	1
Koniteks	0.82	0.88	1	0
Raks Elektronik	0.3	0.3	0.632	0.368
Good-Year	0.65	0.65	1	0
Derimod	0.1225	0.125	0	1

Tablo 7.4. 9 Mayıs 1997 2.ci Seans Sonunda Tablo 7.3.'deki Hisse Senetlerinin Aldığı Değerler (Veriler Normalize Edilmiştir)

Hisse Senedi Adı	En Düşük Fiyat	En Yüksek Fiyat	Kapanış Fiyatı
Alarko Holding	0.18	0.185	0.1825
Borusan	0.068	0.07	0.07
Brisa	0.51	0.51	0.51
Ege Biracılık	0.36	0.365	0.365
Ereğli D. Çelik	0.1425	0.1475	0.1475
İst. Motor Pis.	0.1475	0.15	0.1475
Koniteks	0.8	0.89	0.82
Raks Elektronik	0.295	0.305	0.305
Good-Year	0.65	0.65	0.65
Derimod	0.1225	0.1275	0.125

Problemimizin çözümünde $(2 \times 2 \times 1)$-üç katmanlı yapay sinir ağı modelini kullandık. Bu model Şekil 7.1. de verilmiştir. Rastgele ağırlıklar seçerek ilk veriyi yapay sinir ağından geçirdik. Bunu yaparken de düğümlerdeki çıkışları hesaplamak için aşağıdaki formülü kullandık. t başlangıç değeri de rastgele seçilecektir.

O : Sigmoid fonksiyonu kullanılarak başlangıç değeri için hesaplanan çıkış (out)

x_i : Başlangıç değerine bağlı giriş değeri

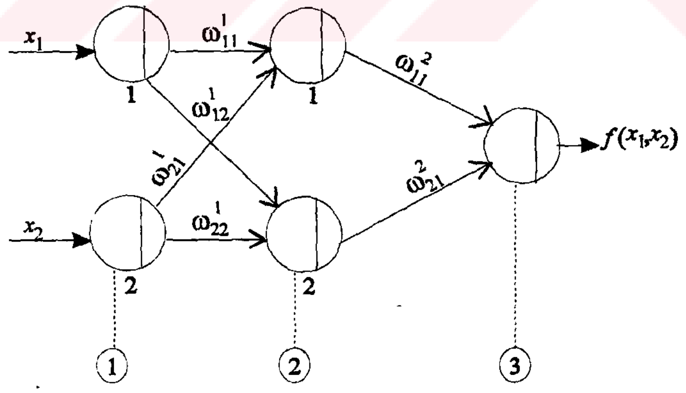
w_i : x_i ye bağlı ağırlık

t : başlangıç değeri

olmak üzere çıkışlar

$$O = \frac{1}{1 + \exp - \left(\sum x_i w_i - t\right)} \tag{7.1}$$

formülüyle hesaplanırlar.



Şekil 7.1. Problemimizdeki $(2 \times 2 \times 1)$ Yapay Sinir Ağı Modeli

Son katmandaki çıkışımız tek olduğundan ve biz de bir tek "düşük" değeriyle ilgilendiğimizden yukarıda anlattığımız yöntemden farklı olarak tek çıkışı ele alarak problemi tek boyutta düşüneceğiz. Bunun için de Tablo 7.1. deki sadece düşük verilerini kullanıp

$$D{:}E_1^3 = O_1^3 - O_1^3(\text{gerçek}) \quad . \tag{7.2}$$

($O_1^3(\text{gerçek})$ değeri Tablo 7.4. deki düşüğe karşılık gelen değerdir) değerini bulacağız. Daha sonra bulduğumuz bu hatayı geriye yayarak ağırlıkları değiştireceğiz. Bunu yaparken de

$$E_n^i = O_n^i\left(1 - O_n^i\right)\sum_j w_{nj}^i E_j^{i+1} \tag{7.3}$$

formülünü kullanırız.

Daha sonra,

w_{jk}^i : *i.* katmanın *j.* elemanını *(i+1).* katmanın *k.* elemanına taşıyan ağırlık

α : öğrenme sabiti

E_k^{i+1} : *(i+1).* katmanın *k.* elemanının hatası

O_j^i : *i.* katmanın *j.* elemanından *(i+1).* katmanın *k.* elemanına olan çıkış

olmak üzere,

$$w_{jk}^i(son) = w_{jk}^i(ilk) + \alpha\ E_k^{i+1} O_j^i \tag{7.4}$$

formülüyle ağırlıklar değiştirilir. Ağırlıklar değiştirilirken hata da gizli katmanlara dağıtılmış olur. Daha iyi sonuç alabilmek için de α epsilon gibi bir katsayıyla ve çıkış değeriyle çarpıldı. Ayrıca ağırlıklar değiştirilirken t başlangıç değeri de ilk t değerine -epsilon*alpha*(son katmandaki çıkış değeri) çarpımı eklenerek değiştirilir. Çözümümüz

Ross'un verdiği yukarıda girişte anlattığımız yöntemden farklı olarak tekrar birinci veriyi alarak değil de ikinci veriyi alarak bu işlem otuzuncu veriye kadar sürdürülür. Bu bizim için 1.ci iterasyonun sonudur ve biz yapay sinir ağını iyi eğitebilmek için, iterasyonun 10000 kez ardışık olarak tekrarlanmasını sağladık. Ele aldığımız problemde Ross'un verdiği şekilde tek tek herbir veri için sistemi eğittikten sonra elde ettiğimiz sonuç bizi çözüme götürmedi. Bu işlemleri yapmak için üzerinde değiştiklikler yapıp kullandığımız pascal programı EK2 'de verilmiştir. Buna göre yapay sinir ağını eğitmeye başlayacağımız ilk verimiz olan (0.01275, 0.013) için rastgele seçilen ağırlıklar, elde edeceğimiz çıkış değerleri ve yeni ağırlıklar aşağıdaki gibidir.

Rastgele Atanan İlk Ağırlıklar :

$\omega_{11}^{1} = 0.06$

$\omega_{12}^{1} = 0.08$

$\omega_{21}^{1} = 0.2$

$\omega_{22}^{1} = 0.09$

$\omega_{11}^{2} = 0.04$

$\omega_{21}^{2} = 0.11$

İlk Çıkış Değerleri :

$O_1^2 = 0.05004512$

$O_2^2 = 0.500929$

$O_1^3 = 0.45639$

Değiştirilmiş Ağırlıklar :

$\omega_{11}^{1} = 0.059997$

$\omega_{12}^{1} = 0.07997$

$\omega_{21}^{1} = 0.199992$

$\omega_{22}^{1} = 0.089991$

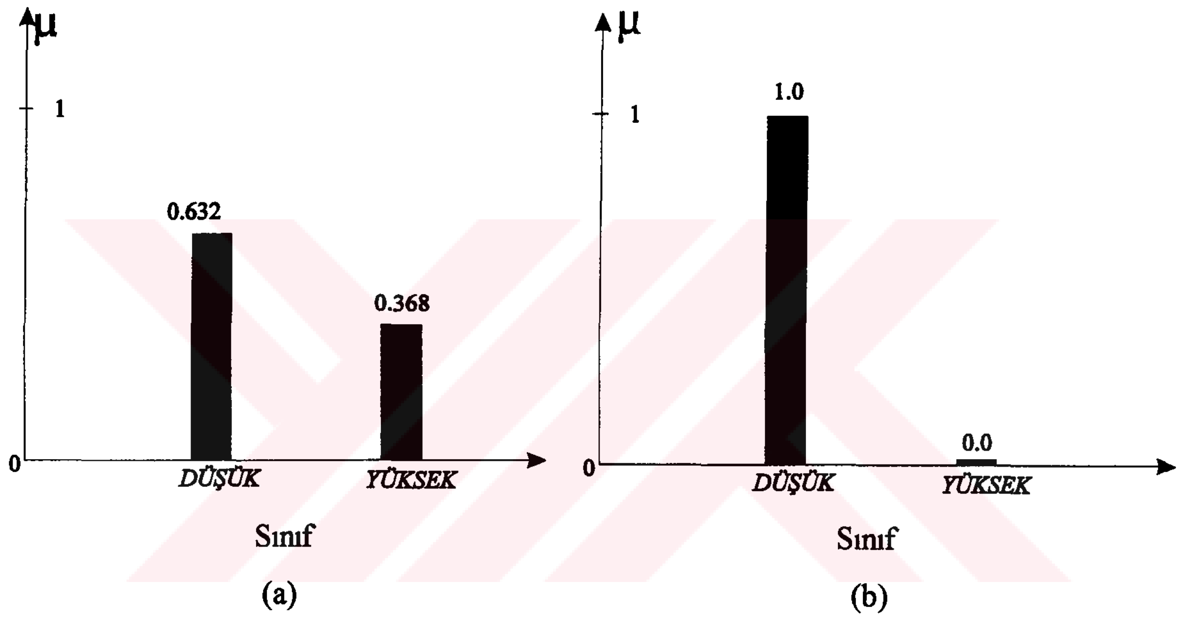
$\omega_{11}^{2} = 0.02866$

$\omega_{21}^{2} = 0.098655$

Yine Ross'dan farklı olarak eğitilen yapay sinir ağının performansını test etmek için ikinci bir veri kümesi yerine eğittiğimiz veri kümesini kullanıyoruz. Problemimiz için 30 verinin eğitilen değerleri ve istenen değerleri Tablo E.1' de verilmiştir.

Tablo 7.3. deki sonuçlarımızı haklı çıkaracak sonradan aracı kuruluşlardan temin ettiğimiz MetaStock grafikleri de Şekil E.1' de verilmiştir.

Tablo 7.3. deki düşük ve yüksek değerleri, bu tablodaki verilerin üyelik değerlerini vermektedir. Şekil 7.2 (a) Raks Elektronik için (b) ise Brisa için elde ettiğimiz monotonik üyelik fonksiyonlarını göstermektedir.



Şekil 7.2. Raks Elektronik (a) ve Brisa (b) için Üyelik Fonksiyonu Grafikleri

SONUÇ

Tezimde, ilk olarak, ülkemizde son yıllarda üzerinde çalışmaların oldukça hızlandığı fuzzy mantığa temel oluşturması bakımından, gerekli tanım ve açıklamaları vermeye çalıştım.

L. A. Zadeh tarafindan geliştirilen fuzzy kümeler, zaman içerisinde, fuzzy mantığın oluşmasına öncülük etmiştir.

Tezimde, fuzzy üyelik fonksiyonu elde etmek için Ross'un Takagi ve Hayashi'nin eserlerinde yola çıkarak kullandığı yöntemlerden biri olan yapay sinir ağı yöntemini ele aldım. Ross, doğadaki dinamik sistemler için fuzzy üyelik fonksiyonlarının devamlı güncelleştirilmesi gerektiğini, bu yüzden kendi kendine değişim yapabilen yapay sinir ağının kullanılmasının uygun bir yöntem olacağını savunmuştur.

Biz de, Ross'un kullandığı yöntem üzerinde bazı değişiklikler yaparak, fuzzy mantık probleminde deneysel verilere dayanarak fuzzy üyelik fonksiyonu belirlemeye çalıştık. Buna ek olarak yapay sinir ağlarını eğitmek için kullanılan pascal programlarından birini, kendi problemimiz doğrultusunda değiştirip, problemimize uyguladık.

Problem olarak da, dinamik bir sistem olan İMKB verilerine dayanarak, bir seanstaki en düşük ve kapanış verilerini, yapay sinir ağını kullanarak eğittik ve test ettik. Buna göre bir sonraki seans sırasında herhangi bir hisse senedinin fiyat seyrine göre aldığı en düşük fiyat ile bir önceki seanstaki kapanış değerini kullanarak o hisse senedinin ikinci seanstaki kapanışının en düşük fiyattan mı yoksa en yüksek fiyattan mı olacağının kestirimini yapmaya çalıştık. Bu sonuca göre de aracı kuruluşlar, müşterileri için, kapanış sırasında bu üyelik değerlerini kullanarak, alış ya da satış yapmaya karar vereceklerdir.

Çalışmalarımız sırasında gördük ki, yaptığımız kestirimler bir sonraki seansta %90 ve üzerinde olumlu sonuçlar vermiştir.

İMKB oldukça dinamik bir sistem olduğundan yaptığımız çalışmalar, seanslık ve günlük olarak uygun sonuçlar verecektir. Türkiye'de her olay borsayı çok rahat dalgalandırabileceğinden çalışmamızın, haftalık ve aylık olarak güvenilir sonuçlar verebileceğini söyleyemiyoruz. En güvenilir sonuçlar ise, bir önceki seansın verilerinin eğitilerek bir sonraki seanstan kestirimlerin yapıldığı durumlarda alınmaktadır.

Ross'un kullandığı gibi birden fazla çıkışlar düşünülerek de bu gibi problemlere farklı yorum ve yaklaşımlar getirilebilinir.

Fuzzy kümelerle çalışırken en zor işlerden biri de üyelik fonksiyonunun bulunması işlemidir. Bizim, burada kullandığımız yapay sinir ağı yönteminin, deneysel verilere dayalı tüm problemlerin çözümlerinde yardımcı olabileceğine ve giderek bir model oluşturmuş bulunduğuna inanıyoruz.

KAYNAKLAR

Alkan Ş., 1994. Y.T.Ü. Yükşek Lisans Tezi - Yüksek Hızlı Fuzzy Lojik Kontrolcu Kullanarak Bir Servo Motorun Adaptif Kontrolu: p.p. 13-19.

Baş E., 1995. Y.T.Ü. Yüksek Lisans Tezi - Bulanık Analiz ve Modelleme ile Ekonomi Sistemi İçinde Borsa Bileşik Endeksi Çıkaran Uzman Sistem Tasarımı.

Başbuğ A., Şubat 1994. Byte Türkiye, Bulanık Teknoloji: p.p.147-151.

Black M., 1937. Vagueness : An Exercise in Logical Analysis, Philosophy of Science, Cilt 4; No. 4: p.p. 427 - 455.

Gülez K., 1995. Y.T.Ü. Yüksek Lisans Tezi - Motor Hata Tspit ve Analizi için Yapay Sinir Ağlarının Tasarımı. p.p. 38-60.

İMKB Yayınları, Aralık 1995. Borsa Terimleri Sözlüğü.

İMKB Yayınları, Ocak 1997. Sermaye Piyasası ve Borsa Temel Bilgiler Kılavuzu, Eğitim Yayınları No:1, 11. Basım: p.p. 158-159, 323-335.

Kandel A., 1986. Fuzzy Mathematical Techniques with Applications: p.p.1-13, 123-135.

Karanfil S., 1994. Y.T.Ü. Yüksek Lisans Tezi - Bulanık Kümeler ve Bulanık Mantığın Temelleri.

Kaufmann A., and Gupta M.M., 1991. Introduction to Fuzzy Arithmetic, Theory and Applications: p.p. 1-35.

Kaynak O., 1992. Şişe Cam Semineri-Bulanık Denetim ve Endüstriyel Uygulamaları: p.p. 1-14.

Klir G.J., and Folger T.A., 1988. Fuzzy Sets, Uncertainty and Information: p.p.1-33.

Kosko B., and Isaka S., Eylül 1993. Bilim Dergisi - Puslu Mantık : p.p. 56 - 61.

Ross T.J., 1995. Fuzzy Logic With Engineering Applications, McGraw-Hill, Inc. p.p. 1-126, 371-402.

Rumbelhart E., Hinton G.E., and Williams R.J., 1986. Learning Representations by Back-Propagation Errors, Nature 323: 533-536.

Şekercioğlu A., 1994. Bulanık Kümeler. Matematik Dünyası, Cilt:4, Sayı:3: p.p. 14-18.

Tekin N., and Karanfil S., 1995. Fuzzy Mantık- Temel Felsefesi, Kuralları, Keskin Mantıkla Karşılaştırılması ve Uygulama Alanları. Öneri- Sayı:2, Cilt:1: p.p. 3-15.

Terano T., Asai K., and Sugeno M., 1992. Fuzzy Systems Theory And Its Applications, Academic Press, Inc. p.p. 1-39.

Türkşen İ.B., 1985. Yöneylem Araştırma Dergisi, Cilt:4, Sayı:1, Bulanık Kümeler Kuramı ve Uygulamaları: p.p.1-15.

Wildberger A.M., 1994. ESS' 94 European Simulation Symposium, Introduction to Fuzzy Sets and Fuzzy Logic.

Zadeh L.A., 1965. Fuzzy Sets , Information and Control, Academic Press, Vol. 8: p.p. 338-353.

Zadeh L.A., 1987 Fuzzy Sets and Applications : Selected Papers by L.A. ZADEH, der. YAGER R.R., vd., A Wiley - Interscience Publication.

Zadeh L.A., 1992. Fuzzy Logic : Advanced Concepts and Structures. I.E.E.E. Educational Activities.

TABLO E.1

Senet No	girisler		egitilen	istenen
0	0.013	0.013	0.000	0.000
1	0.098	0.100	0.000	0.000
2	0.440	0.445	1.000	1.000
3	0.228	0.233	0.000	0.000
4	0.270	0.275	0.041	0.000
5	0.138	0.140	0.000	0.000
6	0.500	0.510	1.000	1.000
7	0.212	0.217	0.000	0.000
8	0.570	0.580	1.000	1.000
9	0.110	0.113	0.000	0.000
10	0.105	0.107	0.000	0.000
11	0.107	0.107	0.000	0.000
12	0.320	0.320	0.952	1.000
13	0.142	0.142	0.000	0.000
14	0.145	0.147	0.000	0.000
15	0.043	0.044	0.000	0.000
16	0.650	0.650	1.000	1.000
17	0.091	0.091	0.000	0.000
18	0.265	0.265	0.016	0.000
19	0.365	0.365	0.999	1.000
20	0.500	0.530	1.000	1.000
21	0.260	0.265	0.012	0.000
22	0.225	0.228	0.000	0.000
23	0.212	0.212	0.000	0.000
24	0.120	0.125	0.000	0.000
25	0.019	0.019	0.000	0.000
26	0.490	0.490	1.000	1.000
27	0.038	0.038	0.000	0.000
28	0.138	0.140	0.000	0.000
29	0.203	0.203	0.000	0.000

iterasyon= 10000   eps = 0.023456   error_1 = 0.002201

egitme ve test ayni zamanda bitirildi

EK 1:

```
program Salihcluster;
uses crt;
const
m=2;n=30;
type mat=array[1..m,1..n] of real;
     mat1=array[1..n,1..m] of real;
     mat2=array[1..m,1..m] of real;
Var
   a,a1,d:mat;
        b:mat1;
        v:mat2;
   i,j,k,p:integer;
   min,pay,payda:real;
   ter,t:real;
   tp,s:real;

procedure sifirla;
begin
     for i:=1 to m do
     for j:=1 to n do
         begin
              a[i,j]:=0;
              a1[i,j]:=0;
              d[i,j]:=0;
         end;
     for i:=1 to m do
     for j:=1 to m do
         begin
              v[i,j]:=0;
         end;
     for i:=1 to n do
     for j:=1 to m do
         begin
              b[i,j]:=0;
         end;
end;

procedure matoku;
begin
     clrscr;
     for i:=1 to m do
         begin
```

```
                for j:=1 to n do
                    begin
                      gotoxy(j*5,i+5);readln(a[i,j]);
                    end;
           end;
end;

procedure giris;
begin
     clrscr;
     for i:=1 to n do
         begin
              for j:=1 to m do
                   begin
                      gotoxy(j*5,i+5);readln(b[i,j]);
         end;          end;
end;

procedure vektor;
begin
     for i:=1 to m do
         begin
              for j:=1 to m do
                  begin
                      pay:=0;payda:=0;
                      for k:=1 to n do
                           begin
                              pay:=pay+a[i,k]*b[k,j];
                              payda:=payda+a[i,k];
                           end;
                      v[i,j]:=pay/payda;
                  end;
         end;
end;

procedure uzaklik;
begin
     for i:=1 to m do
         begin
              for k:=1 to n do
                  begin
                      t:=0;
                      for j:=1 to m do
                           begin
                              ter:=sqr(b[k,j]-v[i,j]);
                              t:=t+ter;
                           end;
```

```
                         d[i,k]:=sqrt(t);
                    end;
          end;
end;
procedure minimum;
begin
     clrscr;
     for k:=1 to n do
         begin
              min:=d[1,k];i:=2;
              if min>d[2,k] then
                 begin
                      a1[i-1,k]:=0;a1[i,k]:=1;
                 end
                 else begin
                      a1[i-1,k]:=1;a1[i,k]:=0;
                      end;
         end;
end;

procedure matyaz;
begin
     clrscr;
     writeln('a[i,j]:');
     for i:=1 to m do
         begin
              for j:=1 to n do write(a[i,j]:9:3);
              writeln;
         end;
end;

procedure matata;
begin
     for i:=1 to m do
         begin
              for j:=1 to n do
                   a[i,j]:=a1[i,j];
         end;
end;


BEGIN
sifirla;
matoku;
giris;
p:=1;s:=0;
```

```
REPEAT
      clrscr;
      vektor;
      uzaklik;minimum;
      for i:=1 to m do
         begin
              s:=0;
              for j:=1 to n do
                  begin
                     tp:=a1[i,j]-a[i,j];
                     s:=s+tp;
                  end;
         end;
      if s=0 then
         begin
              matyaz;
              readln;
              p:=0;
         end
            else
                begin
                     matata;
                end;
until p=0;
clrscr;
writeln('v[i,j]:');
for i:=1 to m do
    begin
       for j:=1 to m do write(v[i,j]:9:3);
       writeln;
    end;
readln;

END.
```

EK 2:

```
Program Fuzzy;
{$N+ E+}
CONST
     Layers_Max=3;     {maksimum katman sayisi}
     Nodes_Max=2;      {maksimum dugum sayisi}
     Outputs_Max=30;  {egitilen maksimum veri sayisi}
     Input_Nodes=2;   {giris dugum sayisi}
     Output_Nodes=1;  {cikis dugum sayisi}
     Hidden_Nodes=2;  {gizli katmanlarin dugum sayisi}
     Epsilon=0.2;
     Alpha=0.9;
     Iterative_Max=10000;
     Every=1000;

VAR
   i,ii,j,count,isylla,num:integer;
   eps,alp,err,error,terr,terror,total_rate:double;

   Layers,Inputs,Outputs:integer;
   Node:array [0..Layers_Max-1] of integer;
   matrix:array [0..Outputs_Max-1,0..(Input_Nodes+
          Output_Nodes-1)] of double;
   matrix_tmp:array [0..trunc(Iterative_Max/Every),0..
        Outputs_Max-1,0..(Input_Nodes+Output_Nodes-1)] of
double;

   des:double;
   rate:array[0..Outputs_Max-1] of double;
   out:array[0..Layers_Max-1,0..Nodes_Max-1] of double;
   der: array[0..Layers_Max-1,0..Nodes_Max-1] of double;
   theta:array[0..Layers_Max-1,0..Nodes_Max-1] of double;
   dtheta:array[0..Layers_Max-1,0..Nodes_Max-1] of double;
   weight:array[0..Layers_Max-1,0..Nodes_Max-1,
        0..Nodes_Max-1] of double;
   dweight:array[0..Layers_Max-1,0..Nodes_Max-1,
        0..Nodes_Max-1] of double;
   dess:array[0..Outputs_Max-1] of double;

   st1:string[5];
   st2,st3,st4:string[10];
   outfile:text;
```

```
procedure Wrand;
var
li,ni,ni_end,no,no_end:integer;
drand48,power:double;
begin
      power:=1.0;
      for li:=1 to 31 do
          power:=power*2;
      power:=power-1;
      for li:=1 to Layers_Max-1 do
      begin
         ni_end:=node[li];
         for ni:=0 to ni_end -1 do
         begin
            drand48:=Random(30)/100;
            theta[li,ni]:=drand48;
            dtheta[li,ni]:=0;
         end;
       end;
       for li:=1 to Layers_Max-1 do
       begin
         ni_end:=node[li];
         no_end:=node[li-1];
         for ni:=0 to ni_end-1 do
         for no:=0 to no_end-1 do
         begin
         drand48:=Random(30)/100;
         weight[li,ni,no]:=drand48;
         dweight[li,ni,no]:=0;
         end;
       end;
end;

procedure initial;
var
  l,n,n_end:integer;
begin
  for l:=0 to Layers-1 do
  begin
     n_end:=node[l];
     for n:=0 to n_end-1 do
     out[l,n]:=0;
  end;
end;
procedure yread(isylla:integer );
{30 veriyi okur}
begin
```

```
  case isylla of
     0:begin out[0,0]:=0.01275;out[0,1]:=0.013 ;end;
     1:begin out[0,0]:=0.0975 ;out[0,1]:=0.1   ;end;
     2:begin out[0,0]:=0.44   ;out[0,1]:=0.445 ;end;
     3:begin out[0,0]:=0.2275 ;out[0,1]:=0.2325;end;
     4:begin out[0,0]:=0.27   ;out[0,1]:=0.275 ;end;
     5:begin out[0,0]:=0.1375 ;out[0,1]:=0.14  ;end;
     6:begin out[0,0]:=0.5    ;out[0,1]:=0.51  ;end;
     7:begin out[0,0]:=0.2125 ;out[0,1]:=0.2175;end;
     8:begin out[0,0]:=0.57   ;out[0,1]:=0.58  ;end;
     9:begin out[0,0]:=0.11   ;out[0,1]:=0.1125;end;
    10:begin out[0,0]:=0.105  ;out[0,1]:=0.1075;end;
    11:begin out[0,0]:=0.1075 ;out[0,1]:=0.1075;end;
    12:begin out[0,0]:=0.32   ;out[0,1]:=0.32  ;end;
    13:begin out[0,0]:=0.1425 ;out[0,1]:=0.1425;end;
    14:begin out[0,0]:=0.145  ;out[0,1]:=0.1475;end;
    15:begin out[0,0]:=0.043  ;out[0,1]:=0.044 ;end;
    16:begin out[0,0]:=0.65   ;out[0,1]:=0.65  ;end;
    17:begin out[0,0]:=0.091  ;out[0,1]:=0.091 ;end;
    18:begin out[0,0]:=0.265  ;out[0,1]:=0.265 ;end;
    19:begin out[0,0]:=0.365  ;out[0,1]:=0.365 ;end;
    20:begin out[0,0]:=0.5    ;out[0,1]:=0.53  ;end;
    21:begin out[0,0]:=0.26   ;out[0,1]:=0.265 ;end;
    22:begin out[0,0]:=0.225  ;out[0,1]:=0.2275;end;
    23:begin out[0,0]:=0.2125 ;out[0,1]:=0.2125;end;
    24:begin out[0,0]:=0.12   ;out[0,1]:=0.125 ;end;
    25:begin out[0,0]:=0.01925;out[0,1]:=0.01925;end;
    26:begin out[0,0]:=0.49   ;out[0,1]:=0.49  ;end;
    27:begin out[0,0]:=0.038  ;out[0,1]:=0.038 ;end;
    28:begin out[0,0]:=0.1375 ;out[0,1]:=0.14  ;end;
    29:begin out[0,0]:=0.2025 ;out[0,1]:=0.2025;end;


  end;
end;


function sigmoid(x:double):double;
var
y:double;
begin
  y:=1/(1+exp(-x));
  sigmoid:=y;
end;

procedure wforward;
var
```

```
li,ni,ni_end,no,no_end:integer;
x,th:double;
begin
  for li:=1 to Layers-1 do
  begin
    ni_end:=node[li];
    no_end:=node[li-1];
    for ni:=0 to ni_end-1 do
    begin
      x:=0;
      for no:=0 to no_end-1 do
      x:=x+weight[li,ni,no]*out[li-1,no];
      out[li,ni]:=sigmoid(x-theta[li,ni]);
    end;
  end;
end;
function back:double;
var
li,ni,ni_end,no,no_end:integer;
e,err,x,d,w,y:double;
begin
   err:=0;
   for ni:=0 to outputs-1 do
   begin
      y:=out[2,ni];
      e:=des-y;
      der[2,ni]:=e*y*(1-y);
      err:=err+e*e;
   end;
   for li:=2 downto 1 do
   begin
      no_end:=node[li-1];
      ni_end:=node[li];
      for no:=0 to no_end-1 do
      begin
         x:=0;
         y:=out[li-1,no];
         for ni:=0 to ni_end-1 do
         begin
            d:=der[li,ni];
            w:=weight[li,ni,no];
            x:=x+d*w;
         end;
         der[li-1,no]:=y*(1-y)*x;
      end;
   end;
   back:=0.5*err;
```

```
end;

procedure learning(alp,eps:double);
{t baslangic degerini ve agirliklari degistirir}
var
li,lo,ni,ni_end,no,no_end:integer;
di,yo,ew,et:double;
begin
for li:=1 to Layers-1 do
begin
ni_end:=node[li];
for ni:=0 to ni_end-1 do
begin
  et:=der[li,ni];
  dtheta[li,ni]:=-eps*et+alp*dtheta[li,ni];
  theta[li,ni]:=theta[li,ni]+dtheta[li,ni];
end;
end;
for li:=1 to Layers-1 do
begin
lo:=li-1;
ni_end:=node[li];
no_end:=node[lo];
for ni:=0 to ni_end-1 do
begin
di:=der[li,ni];
for no:=0 to no_end-1 do
begin
yo:=out[lo,no];
ew:=di*yo;
dweight[li,ni,no]:=eps*ew+alp*dweight[li,ni,no];
weight[li,ni,no]:=weight[li,ni,no]+dweight[li,ni,no];
end;
end;
end;
end;{procedure LEARNING}

procedure make_matrix(it,isylla:integer);
var
i,j:integer;
begin
     for i:=0 to inputs-1 do
     matrix_tmp[it,isylla,i]:=out[0,i];
   for i:=0 to outputs-1 do
     matrix_tmp[it,isylla,inputs+i]:=out[Layers_Max-1,i];
     end;{procedure make_matrix}
```

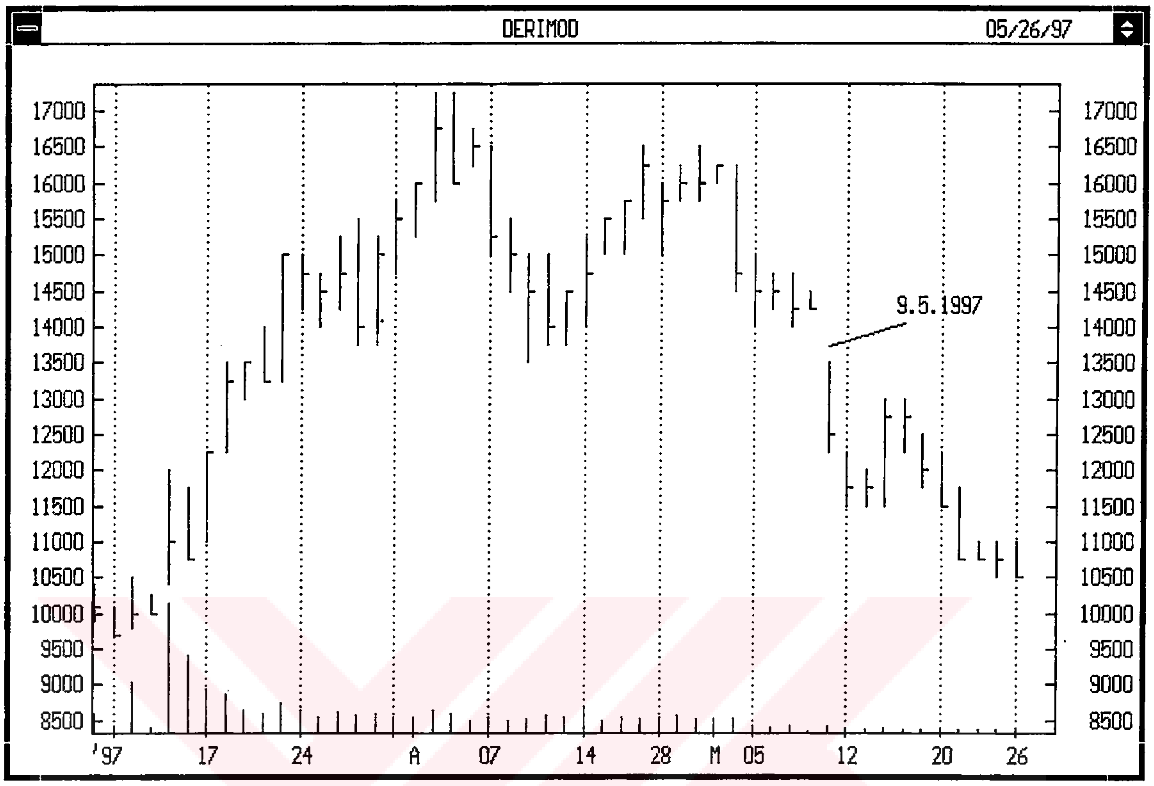
```
begin{main}

FOR I:=0 TO 0 DO WRITELN('Islem Yapiliyor. Bekleyiniz.');
node[2]:=Output_Nodes;
Outputs:=Output_Nodes;
node[1]:=Hidden_Nodes;
node[0]:=Input_Nodes;
Inputs:=Input_Nodes;
Layers:=Layers_Max;
eps:=Epsilon;
alp:=Alpha;
num:=Outputs_Max;


{ ISTENEN CİKISLARIN OKUNMASI}

dess[0]:=0;
dess[1]:=0;
dess[2]:=1;
dess[3]:=0;
dess[4]:=0;
dess[5]:=0;
dess[6]:=1;
dess[7]:=0;
dess[8]:=1;
dess[9]:=0;
dess[10]:=0;
dess[11]:=0;
dess[12]:=1;
dess[13]:=0;
dess[14]:=0;
dess[15]:=0;
dess[16]:=1;
dess[17]:=0;
dess[18]:=0;
dess[19]:=1;
dess[20]:=1;
dess[21]:=0;
dess[22]:=0;
dess[23]:=0;
dess[24]:=0;
dess[25]:=0;
dess[26]:=1;
dess[27]:=0;
dess[28]:=0;
dess[29]:=0;
```

```
{ Gecici Matrisin Sifirlanmasi}
 for i:=0 to num-1 do
     for j:=0 to (inputs+Outputs-1) do
      matrix[i,j]:=0;
 for ii:=0 to (Trunc(Iterative_Max/Every)-1) do
     for i:=0 to num-1 do
         for j:=0 to (inputs+Outputs-1) do
             matrix_tmp[ii,i,j]:=0;

{Ogrenme basliyor...}

Wrand;      {baslangic degeri ve agirliklar ataniyor};
j:=0;
assign(outfile,'c:\OUTFILE.TXT');
rewrite(outfile);

{Egitme Asamasi}

while (j<Iterative_Max ) do
begin
     for count:=0 to Every-1 do
     begin
         error:=0;
         for isylla:=0 to num-1 do
         begin
           initial;
           yread(isylla);
           des:=dess[isylla];{ istenen degeri okur}
           wforward;
           err:=back;
           learning(alp,eps);
           error:=error+err;
         end;
         error:=error/outputs;
         eps:=0.5*sqrt(error);
         Inc(j);
       end;

{***    TEST (kontrol) ASAMASI ***}
end;
       terror:=0;
       for isylla:=0 to num-1 do
       begin
          yread(isylla);
          wforward;
          des:=dess[isylla];
          terror:=terror+terr;
```

```
          make_matrix(trunc(j/Every)-1,isylla);
       end;

writeln(outfile,' girisler     egitilen   istenen ');
writeln(outfile,'-----------   --------  ---------');
for i:=0 to num-1 do
begin
     str(i:5,st1);
     write(outfile,st1+'        ');
     for ii:=0 to inputs-1 do
     begin
       str(matrix_tmp[trunc(j/Every)-1,i,ii]:4:3,st1 );
       write(outfile,st1+'     ');
     end;
     write(outfile,'     ');
     for ii:=0 to outputs-1 do
     begin
    str(matrix_tmp[trunc(j/Every)-1,i,inputs+ii]:4:3,st1);
     write(outfile,st1+'     ');
     end;
     str(dess[i]:4:3,st1);
     writeln(outfile,st1);
end;
   terror:=terror/outputs;
   str(j:5,st1);str(eps:9:6,st2);
   str(error:9:6,st3);str(terror:9:6,st4);
   writeln(outfile,'iteration= ',st1,'   eps = ', st2,
   '    error_1 =', st3,'    error_t= ',st4);
   writeln(outfile);
   total_rate:=0;
for isylla:=0 to outputs-1 do
begin
   rate[isylla]:=trunc(matrix_tmp[trunc(j/Every)-1,isylla,
isylla]+0.5)*100;
   total_rate:=total_rate +rate[isylla];
end;
total_rate:=total_rate/outputs;
{}
if (j=Iterative_Max) then
begin
  writeln(outfile,'egitme ve test ayni zamanda bitirildi')
;
end;
out[0,0]:=1;
out[0,1]:=1;
while  ((out[0,0]<>0) and (out[0,1]<>0)) do
begin
```

```
  write(' 1. deger ');readln(out[0,0]);
  write(' 2. deger ');readln(out[0,1]);

  Wforward;
  WRITELN('****');
  WRITELN(' ',OUT[2,0]:3:3);

end;

close(outfile);

end.
```

Şekil E.1 Bazı Hisse Senetlerinin MetaStock'dan Alınan Grafikleri


DERIMOD
05/26/97
17000
16500
16000
15500
15000
14500
14000
13500
13000
12500
12000
11500
11000
10500
10000
9500
9000
8500
9.5.1997
'97
17
24
A
07
14
28
M
05
12
20
26


RAKS ELEKT
05/26/97
42000
41000
40000
39000
38000
37000
36000
35000
34000
33000
32000
31000
30000
29000
28000
27000
26000
9.5.1997
'97
10
17
24
A
07
14
28
M
05
12
20
26

ÖZGEÇMİŞ

Adı Soyadı	: Salih Karanfil
Doğum Tarihi	: 26 Haziran 1970
Doğum Yeri	: Gökağaç/KIBRIS
İlk Öğrenimi	: 1982 yılında Yeşilyurt İlkokulundan mezun oldu.
Orta Öğrenimi	: 1988 yılında Lefkoşa Türk Maarif Kolejinden mezun oldu.
Lisans Öğrenimi	: 1992 yılında Yıldız Üniversitesi Fen-Edebiyat Matematik Bölümünden mezun oldu.
Yüksek Lisans Öğrenimi	: 1994 yılında Yıldız T. Üniversitesi, Fen-Bilimleri Enstitüsü Matematik Anabilim Dalından mezun oldu.
Doktora Öğrenimi	: Y.T.Ü. Fen-Bilimleri Enstitüsü Matematik Anabilim Dalı (1994 -).
Görevi	: Yıldız Teknik Üniversitesi Fen-Edebiyat Fakültesi, Matematik Bölümü, Matematiğin Temelleri ve Matematik Lojik Anabilim Dalı Araştırma Görevlisi.