

T.C.
ANKARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME
ANABİLİM DALI

YAPAY SİNİR AĞLARI İLE FİNANSAL TAHMİN

Yüksek Lisans Tezi

Barış Bilen VURAL

Ankara-2007

T.C.
ANKARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI

YAPAY SİNİR AĞLARI İLE FİNANSAL TAHMİN

Yüksek Lisans Tezi

Barış Bilen VURAL

Tez Danışmanı
Doç.Dr. Yalçın KARATEPE

Ankara-2007

T.C.
ANKARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI

YAPAY SİNİR AĞLARI İLE FİNANSAL TAHMİN

Yüksek Lisans Tezi

Tez Danışmanı :

Tez Jürisi Üyeleri

Adı ve Soyadı

İmzası

Doç.Dr. Yalçın Karatepe

Doç.Dr. Ali Argun KARACABEY

Doç.Dr.Hasan ŞAHİN

.....

.....

.....

.....

.....

.....

Tez Sınavı Tarihi : 09.02.2007

İÇİNDEKİLER

1. YAPAY SINİR AĞLARININ GENEL YAPISI	9
1.1 Yapay Sinir Ağlarının Tanımı ve Genel Yapısı.....	9
1.2 Yapay Sinir Hücresi.....	11
1.3 Yapay Sinir Ağının Yapısı.....	16
1.4 İlk Yapay Sinir Ağları	17
1.4.1 Tek Katmanlı Algılayıcılar.....	17
1.4.1.1 Basit Algılayıcılar	18
1.4.1.2 ADALINE/MADALINE Modeli.....	19
1.4.2 Çok Katmanlı Yapılar	22
1.4.2.1 İleri Beslemeli Ağlar	22
1.4.2.2 Geri Dönüşümlü Ağlar	24
1.5 Yapay Sinir Ağının Öğrenmesi.....	27
1.5.1 Öğretmenli Öğrenme	27
1.5.2 Öğretmensiz Öğrenme.....	29
1.6 Yapay Sinir Ağlarının Temel Özellikleri.....	30
1.7 Yapay Sinir Ağlarının Önemli Avantajları	32
2. YAPAY SINİR AĞLARI TARİHÇESİ VE	35
FİNANSAL ALANDAKİ KULLANIM ALANLARI	35
2.1 Yapay Sinir Ağlarının Tarihçesi.....	35
2.2 Yapay Sinir Ağlarının Finansal Alanlardaki Kullanım Alanları.....	38
3. ŞİMDİYE KADAR YAPILMIŞ ÇALIŞMALAR.....	41
3.1 Önceki Çalışmaların Detaylı Analizi.....	41
3.1.1. Finans ve Yatırımda Yapay Sinir Ağlarının Uygulandığı	
Problemlerin Sınıflandırılması	41
3.2 Hisse Senedi Tahminlerinde Kullanılan Modeller ve Değişkenler	45
3.3 Önceki Araştırmada Kullanılan Yapay Sinir Ağı Metodolojisi Üzerine	
Karşılaştırmalı Analiz.....	52
3.3.1 Önceki Araştırmada Kullanılan Yapay Sinir Ağı Mimarileri.....	53
3.3.2 Önceki Araştırmada Kullanılan Yapay Sinir Ağı Yapı ve	
Parametrelerinin Optimizasyon Teknikleri	56
3.3.3 Önceki Araştırmanın Sonuçları ve Değerlendirme Ölçümleri.....	57
3.4. Önceki Araştırmaların Yararları, Sınırlamaları ve Sonuçları.....	61
3.4.1 . Önceki Araştırmaların Yararları	61
3.4.2 Önceki Araştırmalardan Ortaya Çıkan Sınırlamalar	62

3.4.3 Önceki Araştırmalarla İlgili Sonuçlar	63
4.YAPAY SİNİR AĞININ OLUŞTURULMASI.....	66
5.Yapay Sinir Ağı Programı	73
5.1 Veri Yapıları ve Fonksiyonlar	74
5.1.1 Sinir Hücresi.....	74
5.1.2 Sinir Ağı Katmanı	77
5.1.3 Sinir Ağı	79
6.Testler ve Sonuçlar	81
6.1 Genel Test ve Eğitim Bilgileri	81
6.2 Değişken Seçimleri	82
6.2.1 Sinir Ağı Yapısı	82
6.2.2 Aktivasyon Fonksiyonu	85
6.2.3 Sinir Hücrelerinin Ağırlıklarının Atanması.....	86
6.2.4 Öğrenim Algoritması Değerleri	91
6.3 Hisse Senedi ve Endeks Grafikleri	95
6.3.1 Hisse Senedi Grafikleri.....	95
6.3.1.1 Garanti Bankası	96
6.3.1.2 İşBank(C).....	96
6.3.1.3 Koç Holding	97
6.3.1.4 Akbank	97
6.3.1.5 Yapı ve Kredi Bankası.....	98
6.3.1.6 Tüpraş	98
6.3.1.7 Migros	99
6.3.1.8 Beko	99
6.3.1.9 Sabancı	100
6.3.1.10 Türkcell.....	100
6.3.2 Endeks Grafikleri.....	101
6.3.2.1 IMKB Ulusal-100	102
6.3.2.2 IMKB Ulusal-30	102
6.3.2.3 IMKB Ulusal-Tüm	103
6.3.2.4 IMKB Ulusal-Sınai	103
6.3.2.5 IMKB Ulusal-Hizmet	104
6.3.2.5 IMKB Ulusal-Mali	104
7.Sonraki Çalışmalar	105
8.ÖZET	106
1. Abstract	109
2. Ekler.....	112
10.1 Program Kod Dokümantasyonu.....	112

AdUzayı Listesi.....	112
AdUzayı : NeuralNetwork	112
NeuralNetwork Sınıf Listesi.....	112
Interfaces.....	112
Sınıfı.....	113
NeuralNetwork Interfaces	114
ActivationFunction İnterface	114
Özet	114
Sınıf Üyeleri	114
NeuralNetwork Sınıfı.....	115
SigmoidActivationFunction Sınıfı	115
Özet	115
Yapılandırıcı Üyeler.....	115
Lokal Alan Üyeleri.....	116
Global Alan Üyeleri	116
Sınıf Üyeleri	116
LinearActivationFunction Sınıfı.....	117
Özet	117
Yapılandırıcı Üyeler.....	117
Lokal Alan Üyeleri.....	117
Global Alan Üyeleri	118
Sınıf Üyeleri	118
HeavisideActivationFunction Sınıfı.....	118
Özet	118
Yapılandırıcı Üyeler.....	119
Global Alan Üyeleri	119
Sınıf Üyeleri	119
GaussianActivationFunction Sınıfı.....	120
Özet	120
Yapılandırıcı Üyeler.....	120
Lokal Alan Üyeleri.....	120
Global Alan Üyeleri	121
Sınıf Üyeleri	121
Layer Sınıfı	122
Özet	122
Yapılandırıcı Üyeler.....	122
Lokal Alan Üyeleri.....	122
Global Alan Üyeleri	123
Sınıf Üyeleri	123

LearningAlgorithm Sınıfı	124
Özet	124
Yapılandırıcı Üyeler.....	124
Lokal Alan Üyeleri.....	125
Global Alan Üyeleri	125
Sınıf Üyeleri	126
BackPropagationLearningAlgorithm Sınıfı	126
Özet	126
Yapılandırıcı Üyeler.....	127
Lokal Alan Üyeleri.....	127
Global Alan Üyeleri	128
Sınıf Üyeleri	129
GeneticLearningAlgorithm Sınıfı	130
Özet	130
Yapılandırıcı Üyeler.....	130
Lokal Alan Üyeleri.....	130
Global Alan Üyeleri	132
Sınıf Üyeleri	134
GeneticLearningAlgorithm.GeneticNeuralNetwork Sınıfı	134
Özet	134
Yapılandırıcı Üyeler.....	135
Lokal Alan Üyeleri.....	135
Global Alan Üyeleri	135
Sınıf Üyeleri	136
NeuralNetwork Sınıfı	136
Özet	136
Yapılandırıcı Üyeler.....	136
Lokal Alan Üyeleri.....	137
Global Alan Üyeleri	137
Sınıf Üyeleri	138
Neuron Sınıfı	139
Özet	139
Yapılandırıcı Üyeler.....	139
Lokal Alan Üyeleri.....	140
Global Alan Üyeleri	141
Sınıf Üyeleri	142
10.2 Örnek Veriler	144
3. Referanslar	152

1. YAPAY SİNİR AĞLARININ GENEL YAPISI

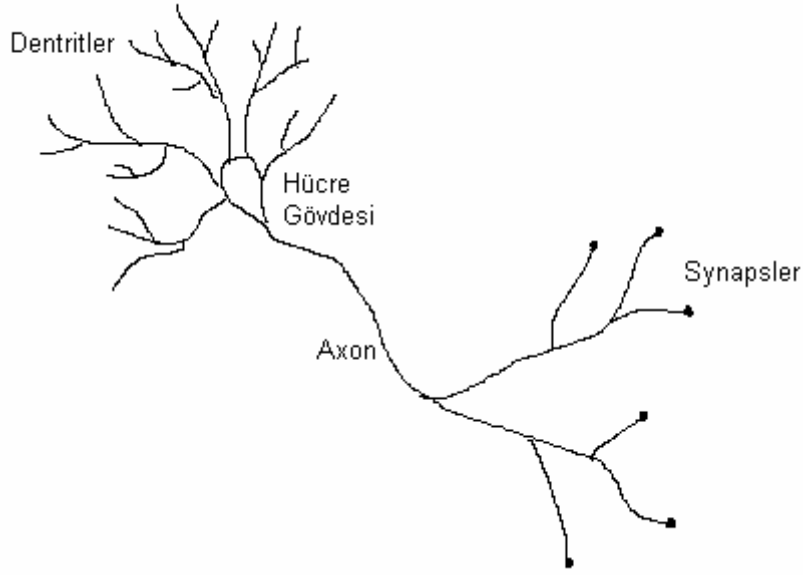
İnsanlığın başlangıcından beri insanoğlunun en temel hedefi hayatı kolaylaştırmak olmuştur. Günümüzde, bilgisayarlar ve bilgisayar sistemleri yaşamı kolaylaştırarak insanların en temel ihtiyaçlarından biri haline gelmişlerdir. Teknoloji geliştikçe, bilgisayarlar karmaşık denklemleri çözenin yanında olaylar hakkında karar verebilme ve olaylar arasındaki ilişkileri öğrenebilme özelliği kazanmışlardır. Yapay zekâ çalışmaları 1950'li yıllardan beri devam etmektedir. Yapay zekâ, bir bilgisayarın veya bilgisayar denetimli bir makinenin, insana özgü nitelikler olduğu varsayılan akıl yürütme, anlam çıkarma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçlere ilişkin görevleri yerine getirme olarak tanımlanabilir (Halıcı ve Üçoluk, 1993). Araştırmacıların ve ticari firmaların ilgisini çeken yapay zekâ çalışmaları günümüzde birçok olaya ve probleme çözüm üretmektedir. Karşılaşılan problemlerin çokluğu ve araştırmacıların problemlere olan farklı bakış açıları; yapay sinir ağları ve genetik algoritmalar gibi birçok yapay zekâ teknolojilerinin yaşantımıza girmesini sağlamıştır.

1.1 Yapay Sinir Ağlarının Tanımı ve Genel Yapısı

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi işlemleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir (Öztemel, 2003).

Genel anlamda yapay sinir ađları, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. Yapay sinir ađları, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak gerçekleştirilebilir. Beynin bilgi işleme yöntemine uygun olarak yapay sinir ađları, bir öğrenme sürecinden sonra bilgiyi toplar, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklar ve genelleme yeteneğine sahip paralel dağılmış bir işlemci olarak çalışır. Bunun yanında öğrenme süreci, arzu edilen amaca ulaşmak için yapay sinir ađlarının ağırlıklarının yenilenmesini içeren öğrenme algoritmasını içerir.

Yapay sinir ađları, biyolojik sinir ađlarının işlevlerinden esinlenerek oluşturulmuş matematiksel modellerdir. Biyolojik sinir ađları, sinir hücrelerinden meydana gelmektedir ve insan beyninde yaklaşık on milyar sinir hücresi ve bunlarında altı yüz trilyondan fazla bağlantısının olduğu düşünülmektedir. Bir biyolojik sinir hücresinin yapısı Şekil 1. de verilmektedir ve görüldüğü gibi temel bir sinir hücresi; dendritler, hücre gövdesi, akson ve sinapslardan oluşmaktadır. Çevreden alınan uyarıcılar elektrik sinyallerine dönüştürülerek hücre gövdesine ulaşır ve burada işlem uygulanarak başka bir sinyal oluşturulup akson aracılığıyla dendrite gönderilir.

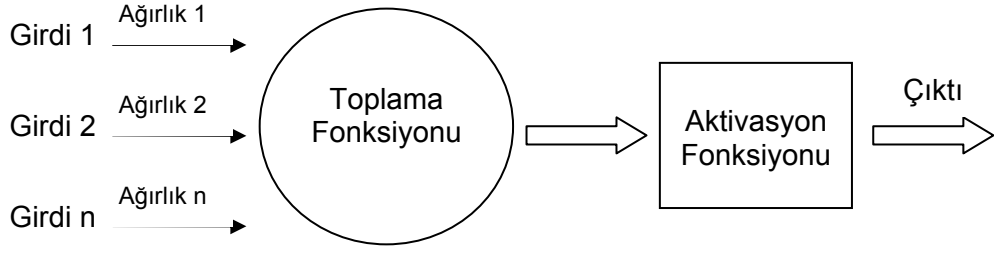


Şekil 1. Biyolojik Sinir Hücresinin Yapısı

Dendritler bu bilgileri sinapslara ileterek diğer hücelere gönderirler. Milyarlarca sinir hücresi bir araya gelerek sinir sistemini oluşturmakta ve bilim adamları biyolojik hücrelerin yapısal özelliklerinden yararlanarak yapay sinir ağlarını geliştirmektedirler.

1.2 Yapay Sinir Hücresi

Yapay sinir ağlarının yapay sinir hücreleri bulunmakta ve bu hücreler süreç elemanı olarak adlandırılmaktadır. Genel özellikleri ile bir yapay sinir hücresinin yapısı Şekil 2. de verilmektedir.



Şekil 2. Yapay Sinir Hücresi

Yapay sinir hücresine dışarıdan verilen bilgiler girdi olarak adlandırılır. Ağırlık değerleri ise; hücreye girdi olarak verilen bilgilerin önemini ve hücre üzerindeki etkisini gösterir. Toplama fonksiyonu, hücreye gelen net girdiyi hesaplayan fonksiyondur ve genellikle gelen girdilerin kendi ağırlıklarıyla çarpımlarının toplamıdır, Toplama fonksiyonu Denklem 1. deki gibi ifade edilebilir (Öztemel, 2003),

$$NET = \sum_i^n G_i A_i \quad (1)$$

Burada G girdileri, A ise ağırlıkları, n ise bir hücreye gelen toplam girdi sayısını göstermektedir. Literatürde farklı toplama fonksiyonları kullanılmıştır, bu fonksiyonlardan bazıları Tablo 1. de verilmektedir. Bir problem için en uygun toplama fonksiyonunu belirlemek için bulunmuş bir formül yoktur. Genellikle deneme yanılma yöntemi ile belirlenmektedir.

Tablo 1. Toplama Fonksiyonları

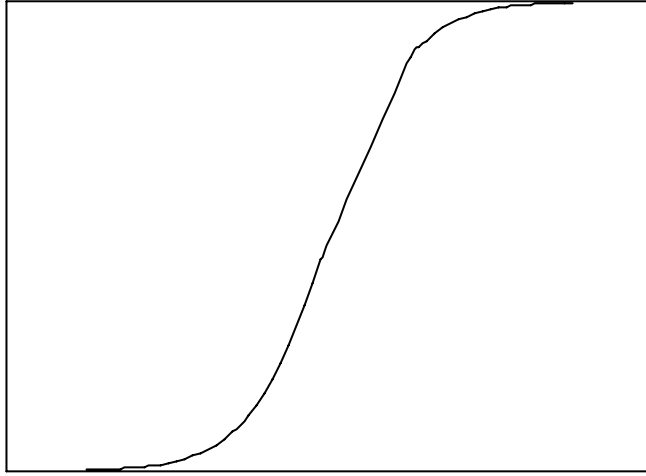
Net Giriş	Açıklama
<p>Çarpım</p> $NET = \prod_i G_i A_i$	<p>Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak NET girdi hesaplanır.</p>
<p>Maksimum</p> $NET = \text{Max}(G_i A_i), i = 1 \dots N$	<p>N tane girdi içinden ağırlıklar ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin NET girdisi olarak kabul edilir.</p>
<p>Minimum</p> $NET = \text{Min}(G_i A_i), i = 1 \dots N$	<p>N tane girdi içinden ağırlıklar ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin NET girdisi olarak kabul edilir.</p>
<p>Çoğunluk</p> $NET = \sum_i \text{sng}(G_i A_i)$	<p>N tane girdi içinden ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücrenin NET girdisi olarak kabul edilir.</p>
<p>Tarihli Toplam</p> $NET = \text{Net}(\text{eski}) + \sum (G_i A_i)$	<p>Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi bulunur.</p>

Aktivasyon fonksiyonu ise, toplama fonksiyonundan gelen net girdiyi işlemde geçirerek hücrenin çıktısını üreten ve genellikle doğrusal olmayan fonksiyondur. Kullanılan hücre modelini çeşidine göre değişik aktivasyon fonksiyonları kullanılmaktadır.

Tablo 2. Aktivasyon Fonksiyonları

Aktivasyon Fonksiyonu	Açıklama
Doğrusal fonksiyon $F(NE T) = NE T$	Gelen girdiler olduğu gibi hücrenin çıktısı olarak kabul edilir.
Step fonksiyonu $F(NE T) = \begin{cases} 1 & \text{if } NE T > ED \\ 0 & \text{if } NE T \leq ED \end{cases}$	Gelen NET girdi değerinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sinüs fonksiyonu $F(NE T) = \text{Sin}(NE T)$	Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır.
Eşik değer (ED) fonksiyonu $F(NE T) = \begin{cases} 0 & \text{if } NE T \leq 0 \\ NE T & \text{if } 0 < NE T < 1 \\ 1 & \text{if } NE T = 1 \end{cases}$	Gelen bilgiler 0 veya 1'den büyük veya küçük olmasına göre bir değer alırlar. 0 ve 1 arasında değerler alabilir. Bunların dışında değerler alamaz.
Hiperbolik tanjant fonksiyonu $F(NE T) = \frac{e^{NE T} + e^{-NE T}}{e^{NE T} - e^{-NE T}}$	Gelen NET girdi değerinin tanjant fonksiyonundan geçirilmesi ile hesaplanır.

Tablo 2. de çeşitli aktivasyon fonksiyonları verilmiştir. Problemlerin çözümünde genellikle aktivasyon fonksiyonu olarak **Sigmoid Fonksiyonu** kullanılmaktadır. Türevi alınabilir, sürekli ve doğrusal olmayan bir fonksiyon olması dolayısıyla, doğrusal olmayan problemlerin çözümünde yaygın olarak kullanılır. Fonksiyonun matematiksel tanımını Denklem 2. de verilmektedir.



Şekil 3. Sigmoid Aktivasyon Fonksiyonu

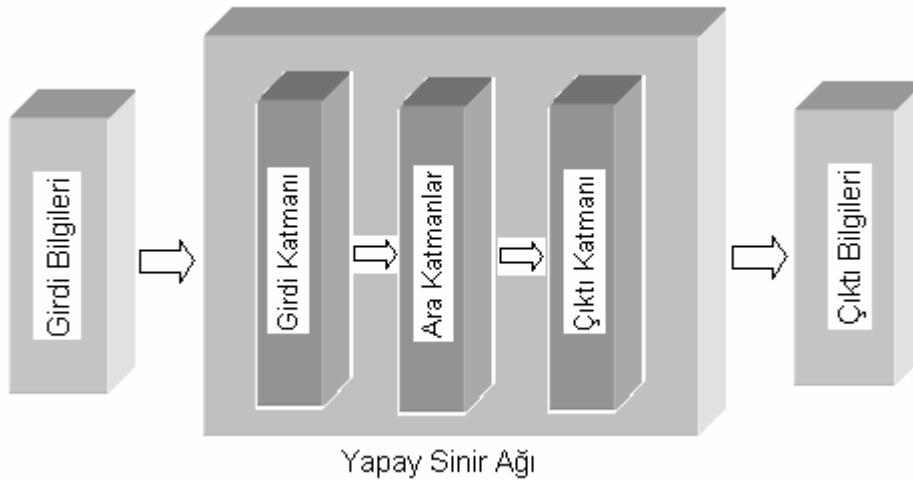
$$F(NET) = \frac{1}{(1 + e^{-NET})} \quad (2)$$

Sigmoid fonksiyonun (tanh) grafiği ise Şekil 3. de gösterilmektedir.

Yapay sinir hücresinin çıktısı, aktivasyon fonksiyonu kullanılarak hesaplanan çıktı değeridir. Üretilen çıktı dış dünyaya veya diğer bir hücreye gönderilir. Bunun yanında hücre kendi çıktısını kendine girdi olarak da gönderebilir.

1.3 Yapay Sinir Ađının Yapısı

Yapay sinir ađları, yapay sinir hücrelerinin bir araya gelmeleriyle oluşmaktadır. Hücreler birbirleriyle bağlanarak; üç veya daha fazla katman halinde ve her katman içinde paralel olarak bir araya gelerek ađ oluştururlar. Yapay sinir ađının katmanları Şekil 4. de verilmektedir (Öztemel, 2003). Girdi katmanındaki süreç elemanları dış dünyadan bilgileri alarak ara katmanlara transfer etmekten sorumludur. Ara katmanlar kullanılarak girdi katmanından gelen bilgiler işlenip çıktı katmanına gönderilir. Bilgiler ara katmanda işlenir. Çıktı katmanı süreç elemanları, ara katmandan gelen bilgileri işleyip ađın girdi katmanından sunulan girdi seti için üretilmesi gereken çıktıyı üretirler.



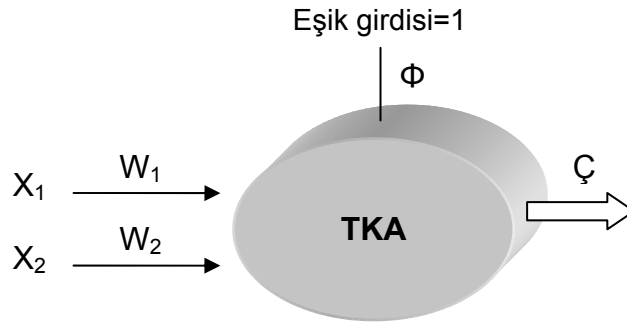
Şekil 4. Yapay Sinir Ađı Katmanları

1.4 İlk Yapay Sinir Ağları

Yapay sinir ağları ile ilgili geliştirilen ilk modeller bu bölümde verilmektedir. Bu modeller; tek katmanlı algılayıcılar (TKA), perceptron ve ADALINE/ MADALINE' dir.

1.4.1 Tek Katmanlı Algılayıcılar

Tek katmanlı yapay sinir ağları sadece girdi ve çıktı (Ç) katmanlarından oluşur. Çıktı üniteleri bütün girdi ünitelerine (X) bağlanmaktadır ve her bağlantının bir ağırlığı (W) vardır. İki girdi ve bir çıktıdan oluşan tek katmanlı bir yapay sinir ağı Şekil 5. de verilmiştir.



Şekil 5. Tek Katmanlı Algılayıcı Modeli

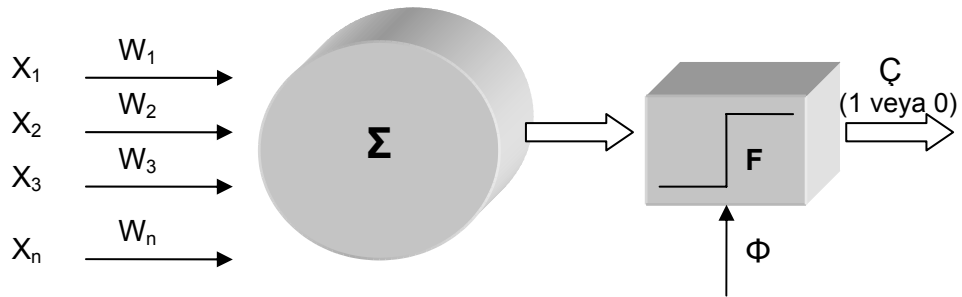
Bu ağlarda süreç elemanlarının değerlerinin ve dolayısıyla ağın çıktısının sıfır olmasını önleyen bir de eşik değeri (Φ) vardır ve değeri daima 1'dir. Ağın çıktısı ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu bulunur. Bu girdi ile bir aktivasyon fonksiyonundan geçirilerek ağın çıktısı hesaplanır. Tek

katmanlı algılayıcılarda çıktı fonksiyonu doğrusal bir fonksiyondur ve 1 veya -1 değerlerini almaktadır. Eğer çıktı 1 ise birinci sınıfa -1 ise ikinci sınıfa kabul edilmektedir.

1.4.1.1 Basit Algılayıcılar

Basit algılayıcılar modeli 1950'li yılların sonlarında Rosenblatt tarafından ortaya atılmıştır. Esas olarak, basit algılayıcılar sınıflandırma amacıyla kullanılan, McCulloch ve Pitts tarafından geliştirilmiş daha karmaşık bir modeldir. Bir araya gelmiş sinir hücrelerinin miktarına bağlı olarak, basit algılayıcılar ile değişik sayıda sınıflandırma problemi çözebilir. Doğru bir sınıflandırma için sınıfların düzlemsel olarak ayrılması gerektiği gösterilmiştir (Nygren, 2004, Minsky ve Papert,1969)).

Basit algılayıcılar bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanmaktadır. Ağın çıkışı bir veya sıfırdan oluşan mantıksal değerdir. Çıktının hesaplanmasında eşik değer fonksiyonu kullanılır. Ağın yapısı Şekil 6. da verilmektedir.



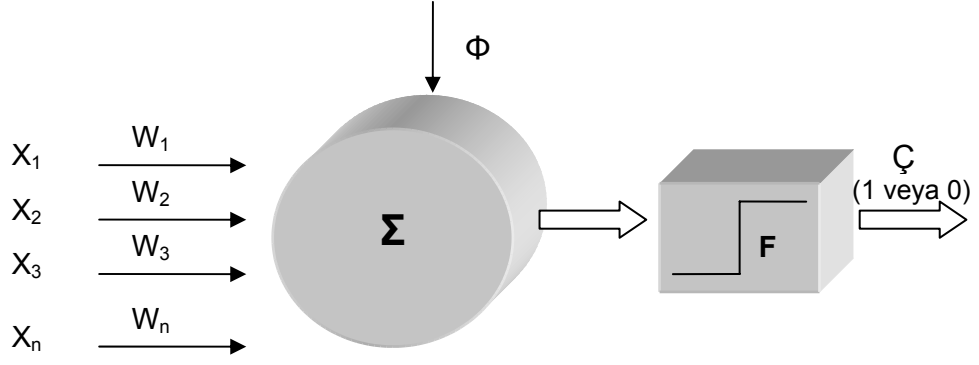
Şekil 6. Basit Algılayıcı Modeli

Basit algılayıcılar eğitilebilir tek bir yapay sinir hücresinden oluşur. Eğitilebilirden kasıt ağırlıkların değiştirilebilir olmasıdır. Girdiler işlem elemanına gösterilirler ve her girdi setine karşılık gelen çıktı değerleri de ağa gösterilir. Daha sonra öğrenme kuralına göre ağın çıktı değerleri hesaplanır. Eğer ağın çıktısı olması gereken çıktıdan farklı ise ağırlıklar ve eşik değerleri değiştirilir. Değişikliğin nasıl yapılacağını ise öğrenme kuralı belirler ve girdilere karşılık gelen çıktı değerleri bir veya sıfırdan oluşur (Öztemel, 2003).

1.4.1.2 ADALINE/MADALINE Modeli

Bu modeller, 1959 yılında Stanford Üniversitesinden Bernard Widrow ve Marcian Hoff tarafından geliştirilmiştir. ADALINE uyabilen doğrusal elemanlar (ADaptive LINear Elements), MADALINE ise çoklu uyabilen doğrusal elemanlar (Multiple ADaptive LINear Elements) olarak tanımlanır (Anderson ve McNeill ,1992).

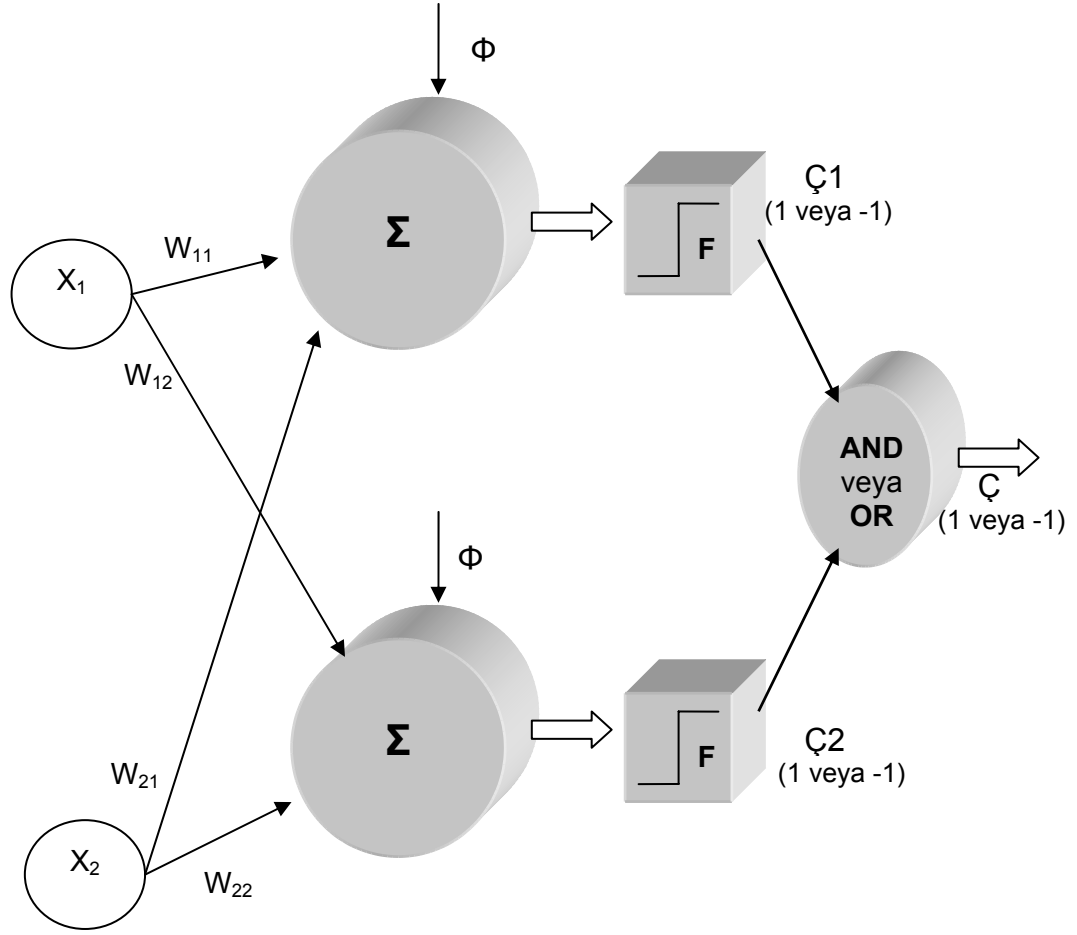
ADALINE ağı en küçük ortalamaların karesi yöntemine dayanmaktadır. Öğrenme kuralına delta kuralı da denilmektedir. Öğrenme kuralı, ağın çıktısının beklenen çıktı değerine göre hatasını en aza indirecek şekilde ağın ağırlıklarının değiştirilmesi prensibine dayanır. ADALINE ünitesinin yapısı Şekil 7. de verilmektedir.



Şekil 7. ADALINE Modeli

ADALINE modeli tek katmanlı algılayıcıya benzemektedir ve aralarındaki fark ise öğrenme kuralında görülmektedir (Öztemel, 2003).

MADALINE ağları birçok ADALINE ünitesinin bir araya gelmesiyle oluşur. MADALINE genel olarak iki katmandan oluşmaktadır ve her katmanda değişik sayıda ADALINE ünitesi bulunmaktadır. Ağın çıktısı 1 ve -1 değerleri ile gösterilmektedir ve her biri bir sınıfı temsil etmektedir. Şekil 8. iki ADALINE ünitesinden oluşan bir MADALINE ağını göstermektedir.



Şekil 8. İki ADALINE Ağından Oluşan MADALINE Ağı

MADALINE'nin öğrenme kuralı ADALINE üniteleri için kullanılan öğrenim kuralı ile aynıdır. Burada en sonda bulunan AND veya OR sonlandırıcısı önemlidir. Klasik mantık teorisinde olduğu gibi AND sonlandırıcısı olması durumunda bütün ADALINE ünitelerinin 1 değerini üretmesi sonucu MADALINE ağının çıktısı 1 olur. Aksi halde -1 (veya 0) değerini alır. OR sonlandırıcısı olması durumunda ADALINE ünitelerinin birisinin 1 üretmesi MADALINE ağının çıktısının 1 olması için yeterlidir (Öztemel, 2003).

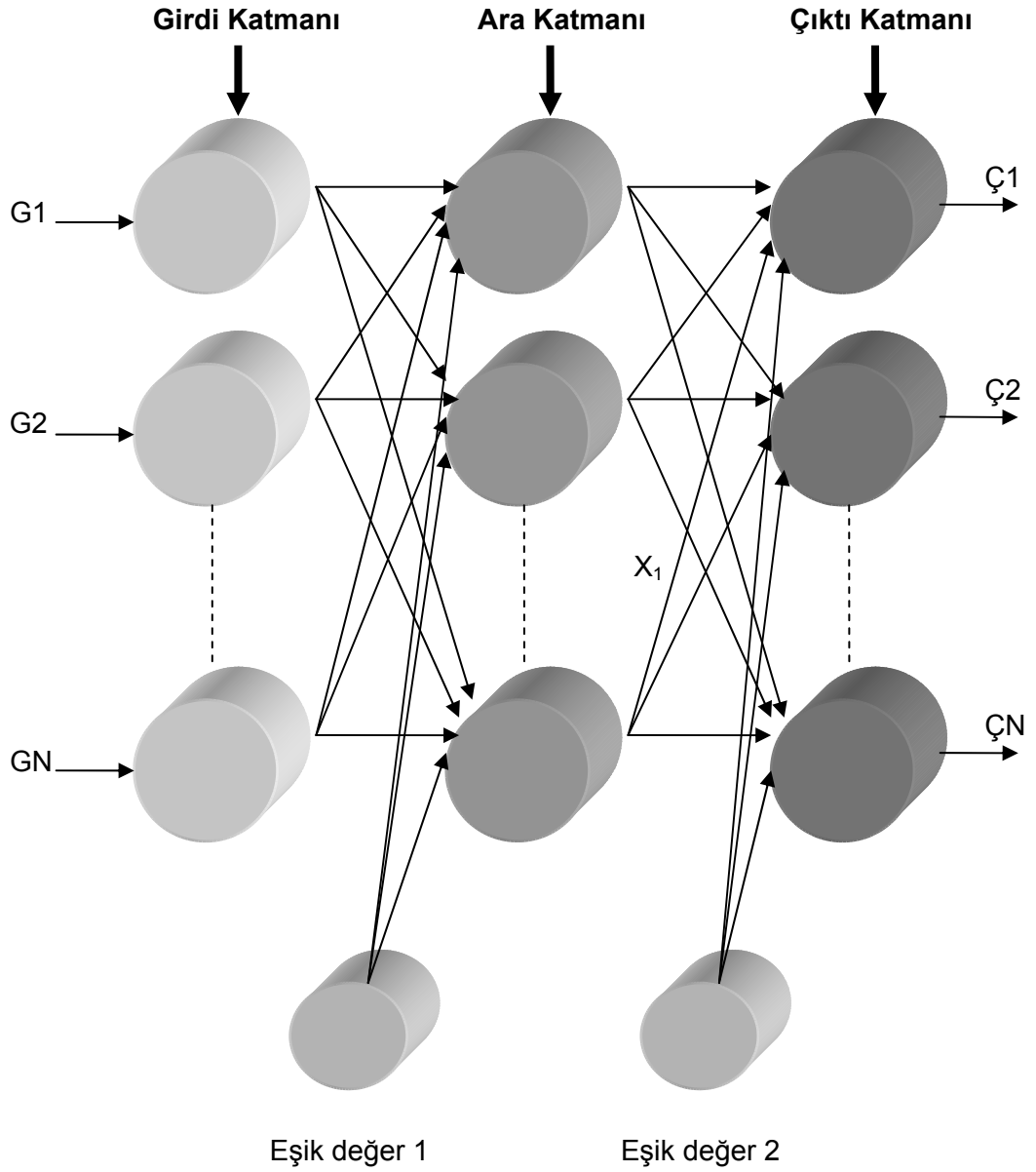
1.4.2 Çok Katmanlı Yapılar

Yapay sinir ağı, sinir hücresi olarak adlandırılan birbirleriyle bağlanmış işlem ünitelerinin katmanlar halinde düzenlenmeleriyle oluşur. Üniteler arasındaki her bağlantı, beyindeki biyolojik sinapsların etkisine benzeyen gerçek ağırlık değerlerine sahiptir (Lisi and Schiavo, 1999). Öğrenme sürecinde ağ, ağırlık değerlerini ayarlar böylelikle ağa verilen girdi setine karşılık gelen doğru şekilde tahmin edilmiş veya sınıflandırılmış çıktı değerleri elde edilir (Enke ve Thawornwong, 2004). Geliştirilen birçok yapay sinir ağı, genel olarak İleri beslemeli ve geri beslemeli ağlar olarak ikiye ayrılabilir.

1.4.2.1 İleri Beslemeli Ağlar

Yapay sinir ağı katmanlar içerir. Tek katmanlı bir ağ girdi ve çıktı katmanlarından oluşur. Çok katmanlı bir ağda ise bunlara ek olarak bir veya daha fazla ara (gizli) katman bulunur. Ara katman sayısı arttıkça yapay sinir ağının girdi verilerinden yüksek mertebede istatistikler elde etme yeteneği artmaktadır. Bu, özellikle geniş girdi katmanlarının olduğu durumlarda önemli bir nitelik kazanmaktadır (Nygren, 2004).

İlk yapay sinir ağları ile doğrusal olmayan ilişkiler öğrenilememekteydi. Bu sorunu çözmek için çok katmanlı ağlar geliştirilmiş ve 1986 yılında XOR probleminin çözümü Rumelhart ve arkadaşları tarafından yapılarak ileri beslemeli çok katmanlı ağlar geliştirilmiştir. Şekil 9.'da ileri beslemeli üç katmanlı (girdi, ara, çıktı katmanları) bir yapay sinir ağı göstermektedir (Öztemel, 2003).



Şekil 9. İleri Beslemeli Çok Katmanlı Yapay Sinir Ağı Modeli

İleri beslemeli yapay sinir ağında, 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 ara (gizli) katmandaki hücrelere iletir. Bilgi, ara ve çıkış katmanında işlenerek ağ

çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı yapay sinir ağının, orta katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen algoritma olan geriye yayılım öğrenme algoritması, bu tip yapay sinir ağların eğitiminde etkin olarak kullanılmaktadır. Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar (beklenen çıktılar) verilmektedir. Ağ kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir (Öztemel, 2003).

Herhangi bir problemi çözmek amacıyla kullanılan yapay sinir ağında, katman sayısı ve orta katmandaki hücre sayısı kesin belirlenememiş olmasına rağmen sinir ağı tahmin yapabilir. Özellikle nesne tanıma ve sinyal işleme gibi alanlarda yaygın olarak kullanılmaktadır.

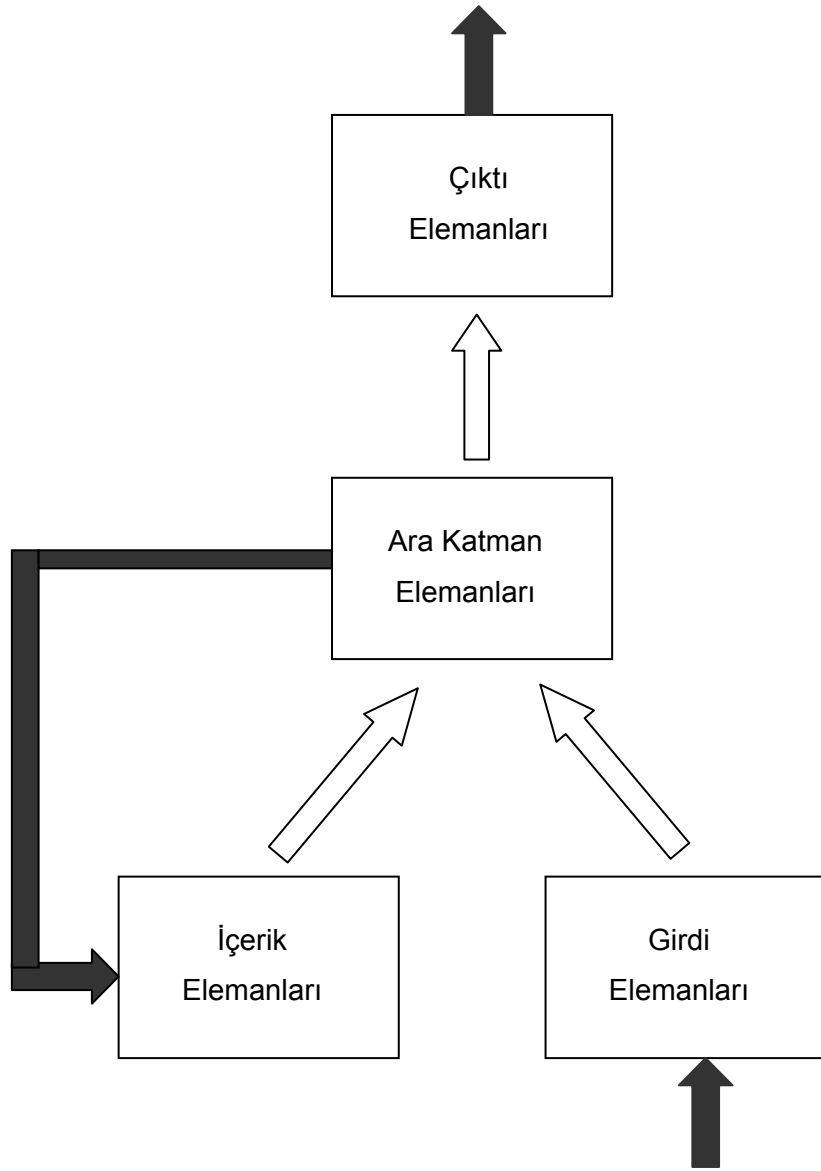
1.4.2.2 Geri Dönüşümlü Ağlar

Geri dönüşümlü yapay sinir ağları, süreç elemanlarının çıktılarını ağa ya da diğer süreç elemanlarına tekrar girdi şeklinde gönderir. Geri dönüşümlü ağlar yüksek performans vermelerine karşın yapılarının belirlenmesi oldukça zor olabilir ve çok dinamik oldukları için durumları denge noktasına ulaşıncaya kadar sürekli olarak yeni girdiler vermek gerekebilir. Denge noktasında girdi değişinceye kadar kalırlar ve daha sonra yeni bir denge bulunur. Dinamik sistemlerin modellenmesinde zaman gecikmesinin göz önüne alınması gereklidir. Geri beslemeli yapay sinir ağları,

özellikle bozucu ve ölçülemeyen girişleri olan dinamik sistemleri modellemek amacıyla kullanılmış ve başarılı sonuçlar alınmıştır.

Geri besleme, geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli yapay sinir ağları, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli yapay sinir ağları yapıları elde edilebilir.

Geri dönüşümlü ağlar arasında en basit yapıya sahip olan ve kullanılması en kolay olan Elman ağıdır. Elman ağının yapısı Şekil 10.'da verilmektedir. Görüldüğü gibi, Elman ağının dört çeşit işlem elemanı vardır. Bunlar; girdi elemanları, ara katman elemanları, çıktı elemanları ve içerik elemanlarıdır. Girdi ve çıktı elemanları dış dünya ile etkileşim halindedir. Girdi elemanları dış dünyadan alınan bilgileri ara katmanlara iletirler. Çok katmanlı algılayıcılarda olduğu gibi, Elman ağında da girdi elemanlarının bilgi işleme özelliği yoktur ve sadece girdileri ara katmana iletirler. Çıktı ünitelerinin bilgi işleme fonksiyonları doğrusaldır, sadece kendilerine gelen bilgileri toplarlar. Ara katman elemanları ise hem doğrusal hem de doğrusal olmayan aktivasyon fonksiyonlarına sahip olabilirler.



Şekil 10. Elman Ağı Yapısı

İÇerik elemanları ara katman elemanlarının önceki aktivite değerlerini hatırlamak için kullanılmaktadır. Bu elemanlar bir adım gecikmeyi içermektedir. Bir önceki döngüdeki aktivasyon değerlerini bir sonraki döngüye girdi olarak taşırlar. Şekil 10.'daki ileri beslemeli bağlantıların ağırlıkları (boş oklar) eğitim sırasında

değiştirilir. Geri dönüşümlerin (içerik elemanlarının) bağlantı ağırlıkları (dolu oklar) ise sabittir (Öztemel, 2003).

1.5 Yapay Sinir Ağının Öğrenmesi

Yapay sinir ağlarında, kullanılan mimarinin yanı sıra, ağırlık değerlerinin belirlenmesi işlemi de ağın performansını etkileyen önemli bir unsurdur. Ağın ağırlık değerlerinin belirlenmesi işlemi, ağın eğitilmesi olarak tanımlanır ve ağın problem ile ilgili doğru sonuçlar veren ağırlık değerlerine ulaşması işlemine de ağın öğrenmesi denir. Gruplama açısından bakıldığında, öğrenme modelleri ikiye ayrılabilir; öğretmenli ve öğretmensiz öğrenme modelleri.

1.5.1 Öğretmenli Öğrenme

Yapay sinir ağlarının eğitilmesinde kullanılan en yaygın yöntemdir. Bu modelde yapay sinir ağından elde edilen gerçek çıktı değeri ile elde edilmesi istenilen çıktı değeri karşılaştırılır. Başlangıçta ağ tarafından rasgele atanan ağırlık değerleri, ağ tarafından tekrar düzenlenir, böylelikle bir sonraki döngüde istenilen çıktı ile gerçek çıktı arasında daha yakın değerler üretir. Bu öğrenme metodu, bütün işlem elementlerinin var olan bütün hatalarını minimize etmeye çalışır ve hataların azaltılması işlemi, girdilerin ağırlık değerlerinin sürekli olarak değiştirilerek kabul edilebilir bir ağ performansına ulaşıncaya kadar devam eder.

Ağın öğrenmeye başlamadan önce eğitilmesi gerekmektedir. Eğitim, girdi ve çıktı verilerinin ağa sunulmasını içermektedir. Bu veriler eğitim seti olarak tanımlanır ve sisteme sunulan her girdiye karşılık istenilen çıktı değerleri de ağa sunulur. Eğitim

aşaması çok fazla zaman alabilir ve yetersiz işlem gücüne sahip bir sistemin öğrenmesi de haftalar alabilir. Yapay sinir ağının eğitimi, ağ kullanıcının tanımladığı performans seviyesine ulaştığında sonlandırılabilir. Bu seviye, ağın verilen girdi değerlerine karşılık istenilen istatistiksel kesinlikteki çıktı değerlerini üretebildiği nokta olarak tanımlanır. Eğer daha ileri seviyede bir öğrenmeye ihtiyaç yoksa elde edilen ağırlık değerleri uygulamalarda kullanılır. Bazı yapay sinir ağları, çalışma sırasında da düşük hızlarda öğrenmeye devam etmekte, bu da ağın ileri seviyelerdeki değişim koşullarına adapte olmasını sağlamaktadır.

Eğer ağdan önemli özellikleri ve ilişkileri öğrenmesi isteniyorsa, eğitim setinin de ihtiyaç duyulan bütün bilgileri sağlayacak derecede oldukça büyük olması gerekmektedir. Ağ sadece bir örnekle eğitiliyorsa, bu örnek için çok dikkatli bir şekilde belirlenen bütün ağırlıklar, diğer örneği öğrenme aşamasında değiştirilebilir. Önceki örnek yeni bir şey öğrenildiğinde unutulabilir. Sonuç olarak sistem her şeyi birlikte öğrenmeli ve bütün örnekler için en iyi ağırlık değerlerini belirlemelidir.

Bir ağı başarılı bir şekilde eğitmek için, girdi ve çıktı verilerinin ağa nasıl sunulacağı çok önemlidir. Yapay sinir ağları sadece sayısal girdi verileriyle çalışabilmekte, bu nedenle dış dünyadan alınan sembolik verilerin dönüştürülmesi gerekmektedir. Ayrıca, bu verilerin ölçeklendirilmesi veya ağın algılayabileceği şekilde normalize edilmesi sağlanmalıdır.

Öğrenen ağ eğitim verileri üzerinde iyi performans sergiledikten sonra, daha önce hiç görmediği verilerle ne yapacağını görmek çok önemlidir. Eğer bir sistem test seti

için mantıklı çıktılar veremiyorsa, eğitim aşaması tam olarak gerçekleşmemiştir. Gerçekten de, test aşaması ağı verilen girdileri ezberlemediği, bir uygulama içindeki genel örnekleri öğrendiğini göstermesi açısından önemlidir (Anderson ve McNeill, 1992).

Öğretmenli öğrenme algoritmalarına örnek olarak; Widrow ve Hoff tarafından geliştirilen **delta kuralı** ve Rumelhart ve McClelland tarafından geliştirilen **genelleştirilmiş delta kuralı** algoritması verilebilir.

1.5.2 Öğretmensiz Öğrenme

Öğretmensiz öğrenme gelecekte ümit verici gelişmeler sağlayacak bir metottur. Bu sayede, bilgisayarlar bir gün kendi kendilerine öğrenebileceklerdir. Günümüzde, öğretmensiz öğrenme metodunu uygulandığı ağların kullanımı çok yaygın olmayıp, sadece temel akademik çalışmalar yürütülmektedir. Bu metot kendi kendine öğrenme olarak da adlandırılabilir.

Ağlar, girdi ağırlıklarını belirlemek için dışarıdan bir etkiye ihtiyaç duymayıp, bunun yerine performanslarını içeriden gözlemektedirler. Bu ağlar girdi sinyallerinde düzen aramaktadırlar ve ağı fonksiyonuna göre adaptasyon yapmaktadırlar. Ağa bir verinin doğru veya yanlış olup olmadığı belirtilmeden, ağ onu nasıl organize edebileceği hakkında bazı bilgilere sahip olmaktadır. Bu bilgi, ağ topolojisinin ve öğrenme kurallarının içine yerleştirilmiştir.

Öğretmensiz öğrenme algoritması, süreç elemanları grupları arasındaki işbirliğine önem vermektedir. Eğer grup içine dışarıdan bazı girdiler aktivite edilirse, grubun etkinliği artabilir. Aynı şekilde, eğer gruptaki dışarıdan girdiler azaltılırsa, bütün grup üzerinde engelleyici etki yaratabilir.

İşlem elemanları arasındaki rekabet, öğrenme esaslı da olabilir. Rekabetçi grupların eğitilmesi, özel grupların özel uyarıcılara olan cevaplarını kuvvetlendirebilir. Böylelikle, gruplar kendi aralarında özel uygun bir cevap ile birleşirler. Öğrenme için rekabet etki altında olduğunda, sadece kazanan süreç elemanına ait ağırlıklar güncellenir.

Günümüzde, öğretmensiz öğrenme ile ilgili araştırmalar genellikle hükümetlerin ilgi duyduğu alandır; çünkü askeri durumlar herhangi bir çatışma çıkıncaya kadar, ağı eğitmek için gerekli veri setine sahip olunamamaktadır (Anderson ve McNeill, 1992).

1.6 Yapay Sinir Ağlarının Temel Özellikleri

Yapay sinir ağlarının hesaplama ve bilgi işleme gücünü, paralel dağılmış yapısından, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme, eğitim ya da öğrenme sürecinde karşılaşılmayan girişler için de yapay sinir ağlarının uygun tepkileri üretmesi olarak tanımlanır. Bu üstün özellikleri, yapay sinir ağlarının karmaşık problemleri çözebilme yeteneğini gösterir. Günümüzde birçok bilim alanında yapay sinir ağları, aşağıdaki özellikleri nedeniyle etkin olmuş ve uygulama yeri bulmuştur (Ergezer, 2003).

Doğrusal Olmama

Yapay sinir ağlarının temel işlem elemanı olan hücre, doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen yapay sinir ağları da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile yapay sinir ağları, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

Öğrenme

Yapay sinir ağlarının arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. Yapay sinir ağlarının karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle yapay sinir ağları, istenen davranışı gösterecek şekilde ilgilendiği problemde aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

Genelleme

Yapay sinir ağları, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir yapay sinir ağı, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş yapay sinir ağları modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

Uyarlanabilirlik

Yapay sinir ađları, ilgilendiđi problemdeki deđişikliklere göre ađlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eđitilen yapay sinir ađları, problemdeki deđişimlere göre tekrar eđitilebilir ve deđişimler devamlı ise gerçek zamanda da eđitime devam edilebilir. Bu özelliđi ile yapay sinir ađları, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır.

Hata Toleransı

Yapay sinir ađları, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuđu için paralel dağılmış bir yapıya sahiptir ve ađın sahip olduđu bilgi, ađdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eđitilmiş bir yapay sinir ađlarının bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ađın doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

1.7 Yapay Sinir Ađlarının Önemli Avantajları

Biyolojik sinir ađlarının çalışma prensibinden yararlanarak oluşturulan yapay sinir ađlarının, geleneksel sistemlere göre birçok üstünlüđu bulunmaktadır. Bu üstünlüklerden bazıları tarafından aşağıdaki şekilde açıklanmıştır (Öztemel,2003).

Görülmemiş örnekler hakkında bilgi üretebilirler; ađ kendisine gösterilen örneklerden genellemeler yaparak görmediđi örnekler hakkında bilgiler üretebilir.

Algılamaya yönelik olaylarda kullanılabilirler; ağların algılamaya yönelik bilgileri işlemede başarılı olduğu yapılan uygulamalarda görülmüştür. Bilgiye dayalı çözümlerde uzman sistemler kullanılmaktadır. Bazı durumlarda yapay sinir ağı ve uzman sistemleri birleştirmek daha başarılı sistemler oluşturmaya neden olmaktadır.

Şekil ilişkilendirme ve sınıflandırma yapabilirler; genel olarak ağların çoğunun amacı kendisine örnekler halinde verilen şekillerin kendisi ve diğerleriyle ilişkilendirilmesidir. Diğer bir amaç ise, sınıflandırma yapmaktır. Verilen örneklerin kümelenendirilmesi ve belirli sınıflara ayrıştırılarak daha sonra gelen bir örneğin hangi sınıfa gireceğine karar vermesi hedeflenmektedir. Şekil tamamlama gerçekleştirebilirler; bazı durumlarda ağa eksik bilgiler içeren örüntü veya şekil verilir ve ağın bu eksik bilgileri bulması istenir. Örneğin yırtık bir resmin kime ait olduğunu belirlemesi ve tam resmi vermesi gibi bir sorumluluk ağdan istenebilmektedir.

Kendi kendini organize etme ve öğretebilme yetenekleri vardır; yapay sinir ağlarının örnekleri ile kendisine gösterilen yeni durumlara adapte olması ve sürekli yeni olayları öğrenebilmesi mümkündür.

Eksik bilgi ile çalışabilmektedirler; yapay sinir ağları kendileri eğitildikten sonra eksik bilgiler çalışabilir ve gelen yeni örneklerde eksik bilgi olmasına rağmen sonuç üretebilirler. Hâlbuki geleneksel sistemler eksik bilgi olmadan çalışmazlar. Yapay sinir ağının eksik bilgi ile çalışması performansının düşük olacağı anlamına gelmez. Performansın düşmesi eksik olan bilginin önemine bağlıdır. Hangi bilginin önemli

olduđunu ađ kendisi eđitim sırasında renmektedir. Ađın performansı dşuk olunca kayıp olan bilginin nemli olduđu kararına varılır.

2. YAPAY SİNİR AĞLARI TARİHÇESİ VE FİNANSAL ALANDAKİ KULLANIM ALANLARI

2.1 Yapay Sinir Ağlarının Tarihçesi

İnsan beyniyle ilgili çalışmalar binlerce yıldır devam etmektedir ancak modern elektroniğin gelişmesiyle, düşünme süreçlerinin anlaşılması da hızlanmıştır. Yapay sinir ağları ile ilgili ilk çalışmalar; 1943 yılında neurophysiologist, Warren McCulloch ve genç matematikçi Walter Pitts'in sinirlerin nasıl çalışabileceğine dair yayınladıkları bir makale ile başlamıştır. Bu çalışmalarında elektrik devrelerinden basit bir sinir ağı modellemişlerdir. Sinirler ve nasıl çalıştıklarına dair bir kitap da 1949 yılında Donald Hebb tarafından yazılmıştır. Kitapta, sinir yollarının her kullanımda daha da kuvvetlendiği belirtilmiştir. 1950'li yıllarda bilgisayar teknolojisinin gelişmeye başlamasıyla, insan düşüncesiyle ilgili teorilerin modellenmeye başlaması mümkün olmuştur. IBM araştırma laboratuvarlarında görev yapan Nathaniel Rochester, yapay bir sinir oluşturmak için çaba göstermiş fakat ilk çalışma başarıya ulaşamamıştır. Daha sonraki çalışmalar başarılı olmuş ve böylelikle yapay sinir ağları ile ilgili yeni bir dönem çağı başlamıştır. Bunun yanında yine bu dönemde makine öğrenmesinin savunmaları devam etmiştir. 1956 yılında, yapay zekâ konusunda yapılan Dartmouth Summer araştırma projesi, yapay zekâ ve yapay sinir ağları ile ilgili ümit verici gelişmeler sağlamıştır. Dartmouth projesini izleyen yıllarda John von Neumann, telgraf kablolarını veya temiz tüpleri kullanarak örnek alan basit sinir fonksiyonlarını önermiştir. Ayrıca, nörobiyoloji uzmanı Frank Rosenblatt algılayıcılar üzerine çalışmaya başlamış ve uçan bir böceğin gözünün çalışması ilgi çekmiştir. Bir böceğe uçmasını anlatan işlemlerin birçoğu gözünün

içindedir ve bu çalışmanın sonucunda donanım olarak inşa edilen algılayıcı, günümüzde hala kullanılan en eski yapay sinir ağı olarak kabul edilmektedir. Tek tabaka algılayıcıdan, iki sınıftan birine yerleştirilmesi gereken sürekli-değer girdilerinin sınıflandırılmasında yararlanılmıştır. Bu algılayıcı ağırlıklı girdi toplamlarını hesaplayıp, bir eşik değeri çıkartmakta ve mümkün olan iki değerden birini geçirerek sonuç sunmaktadır.

1959 yılında, Bernard Widrow ve Marcian Hoff ADALINE ve MADALINE olarak tanımlanan modelleri geliştirdiler. Bu modeller kullanımlarına göre Multiple ADaptive LINear Elements olarak adlandırıldılar. MADALINE, gerçek dünya problemlerine uygulanan ilk yapay sinir ağıdır. MADALINE telefon hatlarındaki ekoları ortadan kaldıran bir filtre yapılmasında kullanılmıştır ve hala ticari olarak kullanımdadır.

Bu erken başarılar, insanların yapay sinir ağlarının potansiyelini abartmasına neden olmuş; özellikle eldeki kısıtlı elektronik sistemlerden insanlar büyük başarılar beklentisi doğmuştur. Akademik ve teknik dünyadaki konuyla ilgili aşırı ilerleme, dönemin genel edebiyatını da etkilemiştir. Düşünen makinelerin insanoğlunu nasıl etkileyeceği konusunda korkular oluşmuş ve olayın ahlaki boyutu tartışılmaya başlanmıştır.

Yapay zekâ biliminin önemli isimlerinden, Marvin Minsky ve Seymour Papert'ın 1969 yılında yayınladıkları Algılayıcılar isimli kitapla, yapay sinir ağları ile ilgili

çalışmaları kısıtlamışlardır. Bu kitapta, algılayıcıların doğrusal olmayan problemlere çözüm üretemeyeceği sonucu ortaya atılmıştır.

Ancak 1982 yılındaki gelişmeler, yeni ilgi alanları oluşturmuştur. John Hopfield, ulusal Bilim Akademisine bir makale sunmuş ve bu makalede Hopfield'ın yaklaşımı basit bir şekilde beyni modellemek değil, faydalı cihazlar yapmak olmuştur. Matematiksel analizlerle böyle bir ağın nasıl çalışabileceğini ve neler yapabileceğini göstermiştir. Aynı tarihte Japonya'nın Kyoto şehrinde bir konferans düzenlenmiştir. Bu konferans, Amerika-Japonya ortak katılımlı Yardımcı/Rekabetçi Yapay Sinir Ağları üzerine yapılmış,1985 yılında Amerikan Fizik Enstitüsü yapay sinir ağları üzerine bir toplantı düzenlemiş ve 1987 yılında, Elektrik ve Elektronik Mühendisleri Enstitüsü 1. Uluslar arası yapay zekâ konferansını organize etmiştir.

Günümüzde, yapay sinir ağları ile ilgili tartışmalar her yerde devam etmektedir. Gelecekte, donanım alanındaki gelişmelerin bu alandaki yeniliklerde kilit rol oynayacağı düşünülmektedir. Şirketler bugün, dijital, analog ve optik olmak üzere üç çeşit sinir yongası üzerinde çalışmaktadırlar ve birçok kısıtlamadan dolayı yapay sinir ağı çalışmaları çok uzun zaman gerektirmektedir. Silikon derleyiciler yaratarak yapay sinir ağı üretme çalışmaları da bazı firmalar tarafından yürütülmektedirler (Anderson ve McNeill, 1992).

2.2 Yapay Sinir Ağlarının Finansal Alanlardaki Kullanım Alanları

Bugün yapay sinir ağları, akademik çevrenin çalışmalarının yanı sıra günlük işlerini kolayca halletmek isteyen kullanıcılarında ilgi alanına girmiştir. Ancak ağların birçoğu çok iyi tasarlanmalarına rağmen, bilgisayar kullanıcılarının problemlerine kesin çözümler üretememektedirler. Ağların verdiği sonuçların kesinliği %85-%90 civarında olmaktadır ve ne yazık ki gerçek yaşamda çok az uygulama bu hata derecesine tolerans gösterebilmektedir. Araştırmacılar, tasarlanan ağlardan üretilen sonuçların kesinliğini arttırmak için çalışmalara devam ederken, bazıları da günümüz teknolojisi için yeni kullanım alanları yaratmaktadır. Günümüzde kullanılan yapay sinir ağları, makinenin insan sesini algılayıp söylenenleri yapmasını sağlayabilmektedir. Bu gelişmeler bir gün, video oynatıcılar, güvenlik sistemleri ve CD çalarların sesle kolayca komutları alıp çalışmaya başlayacaklarının bir işaretidir. Ancak günümüzde yapay sinir ağları, şimdilik, istatistiksel kesinliğin kabul edilmediği piyasalara girmiş durumdadırlar. Özellikle finansal kurumlarda %90 kesinlik, kabul edilebilir bir sonuçtur ve bazı bankalar, yapay sinir ağları tarafından onaylanan borçlanmadaki başarısızlık hızının, geçmişte kullanılan en iyi geleneksel yöntemler tarafından onaylanandan daha düşük olduğunu kanıtlamışlardır. Ayrıca, bazı kredi kartı firmaları yapay sinir ağlarını koruma işlemi uygulamalarında kullanmaktadırlar. Örneklere bakıldığında yapay sinir ağlarının, doğrusal olmayan, karmaşık, eksik bilgiye sahip, hata olasılığı yüksek, gürültülü verilerin olduğu ve problemin algoritmasının bulunmadığı durumlarda kullanıldığı görülmektedir. Aşağıda yapay sinir ağlarının kullanıldığı bazı alanlarla ilgili bilgi verilmektedir.

Bankalar, kredi kartı firmaları ve borç veren kurumlar sıklıkla kesin sonuçları belli olmayan kararlar vermek zorunda kalırlar ve genellikle istatistiksel verilerden yararlanırlar. Borç verme onay işlemi, borç veren kurumun karar vermesini kolaylaştıran bir form doldurma işlemini içerir. Günümüzde bu formdaki bilgiler, geçmişte alınan kararların verileriyle eğitilen yapay sinir ağları tarafından kullanılmaktadır. Ayrıca kredi kartı firmaları da benzer geri-beslemeli yapay sinir ağlarını, kredi risklerini ve limitlerini saptamak için kullanmaktadırlar.

Yapay sinir ağları, bütün finansal piyasalarda, borsada, bono işlemlerinde, uluslar arası nakit para akışında ve ticaret mallarında kullanılmaktadır. Birçok kullanıcı bu sistemleri para getiren sistemler olarak tanımlamakta ve gerçekten de Japonya finans piyasasında yapay sinir ağlarının çok başarılı olduğu bildirilmektedir. Daiichi Kangyo Bankasının hazine bonosu işlemleri ile ilgili raporuna göre, yapay sinir ağları hazine bonosunun hit hızını %60'dan %75'e değerine yükseltmişler. Ayrıca, Daiwa Araştırma Enstitüsü, bir yapay sinir ağının Nikkei ortalamadan %20 daha iyi sonuç verdiğini belirtirken; Daiwa Güvenlik'in borsa tahmin sisteminin, şirketlerin hit hızını %70'den %80 değerlerine yükselttiğini açıklamışlardır (Anderson, McNeill, 1992).

Bunun yanında aşağıdaki finansal alanlarda ayrıca yapay sinir ağları kullanılmaktadır

- Makro ekonomik tahminler,
- Borsa benzetim çalışmaları endekslerinin tahmin edilmesi,
- Kredi kartı hilelerinin tespiti,
- Kredi kartı kurumlarında iflas tahminleri,

- Banka kredilerinin deęerlendirilmesi,
- Emlak kredilerinin ynetilmesi,
- Dviz kuru tahminleri,
- Risk analizleri.

Yapay sinir aęları son dnemlerde regresyon analizi, sınıflandırma, rnt tanıma ve zaman serilerinin analizi gibi istatistiksel alanlarda da kullanılmaktadır. Finans ve para yatırımı yapılan alanlarda yapay sinir aęlarının uygulamalarının arttıęı alanlardır. İflas tahminleri, ipoteęin risk vergi takdiri ve dięer borçlanmalar, hisse senedi piyasası tahminleri (hisse senedi, bono ve satma veya satın alma fiyatları, sermaye karları gibi nemli problemlerin zmnde yapay sinir aęları kullanılmaktadır. Chase Manhattan Bankası (Marose, 1990), Peat Marwick, American Express (Zahedi, 1993) finansal problemlerinin zmnde yapay sinir aęlarını etkili bir Őekilde kullanan birok firmadan sadece birkaçıdır. Yapılan birok araŐtırma ve uygulama, yapay sinir aęlarının, yapay zekâ iermeyen klasik metotlara gre stnlkleri olduęunu kanıtlamıŐtır.

Yapay sinir aęları finans sektrnde en fazla, hisse senedi performanslarının tahmininde ve hisse senedi piyasasında iŐlem grecek hisse senetlerinin Őeiminde kullanılmaktadır.

3. ŞİMDİYE KADAR YAPILMIŞ ÇALIŞMALAR

3.1 Önceki Çalışmaların Detaylı Analizi

Hisse senedi piyasasındaki gelecek tahminlerinde geliştirme yapabilmek için, yapay sinir ağı uygulamalarıyla ilgili önceki çalışmalarda kullanılan problemleri ve modelleri sınıflandırmak ve önceki çalışmaların temel yararlarını ve kısıtlamalarını belirlemek çok önemlidir. Önceki çalışmaların karşılaştırmalı analizini elde etmek için, üç veritabanı indeksi araştırılmıştır. “neural+network+stock” anahtar kelimeleri kullanılarak, INSPEC, Applied Tech & Science Index ve ABI/Inform kaynaklarına başvurulmuştur. Araştırma sonuçlarında çıkan makaleler referans olarak kabul edilmiştir. Referans kabul edilen makalelerin tamamı 1990 yılından sonra yayımlanan makalelerden oluşmaktadır.

3.1.1. Finans ve Yatırımda Yapay Sinir Ağlarının Uygulandığı Problemlerin Sınıflandırılması

Finansal yatırım, yapay sinir ağı uygulamalarının en sık uygulandığı alanlardandır. Yapay sinir ağları kullanılarak çözülen en tipik problemlerden bazıları, iflas tahminleri, ipotek miktarı ve ipotek riski, borçlanmalar, hisse senedi piyasası tahminleri (hisse senetleri, bonolar ve opsiyonlar, anapara geri dönüşleri) ve finansal tahminlerdir. Chase Manhattan Bank (Marose, 1990), Peat Marwick, American Express (Zahedi, 1993), finansal yatırım problemlerinin çözümünde yapay sinir ağlarını etkili bir şekilde kullanan birçok şirketten sadece birkaçıdır.

İş dünyasında yapay sinir ağlarıyla ilgili yapılan birçok araştırma ve uygulama, yapay zekâ içermeyen klasik metotlara karşı yapay sinir ağlarının avantajını kanıtlamıştır. Geçmişteki 10 yılda yapay zekâ uygulamalarının sıklıkla kullanıldığı alanlar; %53,5 üretim/işletim ve %25,4 finansdır (Wong, 1997). Yapay sinir ağlarının finansta en çok kullanıldığı uygulamalar, hisse senedi performansının tahmini ve hisse senedi piyasasında ticaret yapmak için hisse senetlerinin seçimidir. Hisse senedi piyasasında yapay sinir ağı uygulamalarıyla ilgili birçok makale, özel bir problem tanımı yapıldığında, kesin bir yapay sinir ağı algoritmasının ve hiçbir standartlaştırılmış bilginin bulunmadığını söylemektedir (Li, 1994). Buna rağmen birçok araştırmacı, finanstaki sayısız problemin yapay sinir ağları tarafından çözülebildiğini vurgulamaktadır (Zahedi,1997).

Yapay sinir ağlarının en çok kullanıldığı problem grupları aşağıdaki gibidir:

1. Borçlanma ile ilgili tahminler:

- Gayrimenkule dayalı çok uzun vadeli borçlanma (Collins, 1988/93),
- Hisse senedi risklerinin; iyi, kritik veya çıkış olarak tahmin edilmesi (Marose,1990/93),
- Sigorta poliçeli borçlanmalar ve kredilerde risk tahmini (Malhotra 1994),

2. Hisse senedi ve bono piyasası ile ilgili tahminler:

- Alım Satım Zamanının Belirlenmesi:

- 1) S&P 500 de alım satım zamanına karar verme (Trippi, De Sieno, 1992),
- 2) Hisse senetlerinin ne zaman alınıp satılacağına karar verme (Kimoto, Asakawa, Yoda, ve Takeoka, 1990),
- 3) Alış ve satış düzenleri (Bergerson and Wunsch, 1992),
- 4) Hisse senedi değişiminde hisse senedi fiyatlarının ortaya çıkartılması (Kamijo and Tanigawa, 1993),
- 5) Canlı büyükbaş hayvan eşya vadeli işlem piyasalarının alış ve satış modellerinin fark edilmesi (Collard, 1991),

- Riskin değerlendirilmesi ve sınıflandırması

1. Bono değerlendirilmesi (Dutta ve Shekhar, 1992),
2. Bono sınıflandırması (Surkan ve Singleton, 1995),
3. Düşük veya yüksek performanstaki hisse senedi gelirlerinin sınıflandırması (Yoon ve Swales, 1992, 1996),

3. Piyasada bulunan finansal yatırım araçlarının getiri miktarını veya indeks değerlerini tahmin etmek

- Hisse senedi değerlerinin tahmini (Schoeneburg, 1990),
- Vadeli işlemler piyasa değerlerinin tahmin edilmesi (Grudnitski ve Osborn, 1993),
- Aylık hisse senedi hareketlerinin tahmini (E.Y.Li, 1994),

- Hisse senedi piyasası tahmini (Kimoto ve Asakawa, 1990),
- Piyasa etkinliğinin test edilmesi – hisse senedi gelirlerinin tahmini (White,1997),
- Problem: Hisse senedi gelirlerinin tahmini (Wong ve Long, 1995).

Hisse senedi piyasasındaki analize dikkat edildiğinde, yapay sinir ağı uygulamalarının sıklıkla kullanıldığı üç ana problem gurubu olduğu görülür. Bu gruplardan ilki; hisse senetlerinin pozitif gelir getiren ve negatif gelir getiren hisse senetleri (Kryzanowski , 1993; Swales and Yoon, 1992, 1996; Trippi ve DeSieno, 1992) veya iyi, nötr ve kötü performans gösteren hisse senetleri olarak sınıflandırarak, hisse senedi performansı tahminini içeren gruptur. Bu gruba dâhil edilen yapay sinir ağı uygulamaları, yatırım kararlarının alınmasında büyük yarar sağlar; fakat beklenen değerin veya karın miktarını belirtmez. Daha fazla bilgi yaygın bir uygulama olan diğer grup tarafından verilir; bunlar hisse senedi değerlerinin sayısal tahmini için yapay sinir ağlarını kullanan modellerdir. (Schoeneburg, 1990; Grudnitzky ve Osburn, 1993). Bu tür sistemler, önceki hisse senedi değerlerini ve ilgili finansal oranları temel alarak, bir veya daha fazla gün sonraki hisse senedi değerlerini tahmin ederler. Hisse senedi piyasalarında yapay sinir ağı uygulamalarının üçüncü önemli grubu ise, hisse senedi performansının ve tahminlerinin sadece modellenmesiyle ilgilenir (Refenes, 1994; Yoon, 1994). Bu tür uygulamalar sadece gelecek değerlerinin tahminiyle ilgilenmeyip; ayrıca sonucu etkileyebilecek girdiler arasındaki ilişki ve bağımlılık miktarları üzerinde de durmaktadır. Uygulamaların son grubu, çoğunlukla problem alanlarında değil, yapay

sinir ağı araştırılmalarında teorik olarak yer almaktadır. Bu son grup ile ilgili olan çalışmalar bu tez konusunun dışında yer almaktadır.

Yapay sinir ağları finans ve yatırımda çok yaygın olarak kullanılmasına rağmen, ileriki bölümlerde de açıklanacağı gibi, yukarıdaki problem alanlarından gelen araştırmalar için hala bazı açık sorular vardır. Yapay sinir ağları hala açıklanmamış belli noktalar ve sorular bulunmaktadır. Bizim araştırmamız, bir yatırım problemi olan özel hisse senedi gelirlerinin tahmini üzerine yoğunlaşacak ve bu sorular üzerinde duracaktır.

3.2 Hisse Senedi Tahminlerinde Kullanılan Modeller ve Değişkenler

Hisse senedi piyasasında daha önce yapılan yapay sinir ağı uygulamalarına bakıldığında, hemen hemen bütün çalışmaların farklı veri modelleri kullanılarak gerçekleştirildiği görülmektedir. Çünkü bir yapay sinir ağı için veri modelinin tasarımı, girdi ve çıktı değişkenlerinin seçimiyle tanımlanır (Zahedi, 1993), veri modelinin dört özelliği belirtilmiştir: girdi değişkenlerinin sayısı, girdi değişkenlerinin adı, çıktı değişkenlerinin sayısı ve çıktı değişkenlerinin adı. Tablo 4.'de literatürde kullanılan modellerin karşılaştırması verilmektedir.

Tablo 4. Veri Modellerine göre Yapay sinir ağı algoritmaları

		VERİ MODELLERİ
--	--	----------------

Yapay sinir ağı uygulaması	Yapay sinir ağı algoritması	Girdi Değişkenleri Sayısı	Girdi Değişkenleri	Çıktı Değişkenleri
Hisse senedi değeri tahmini (Schoeneburg, 1990)	Geriye yayılım, Basit algılayıcılar, ADALINE / MADALINE	40 (Geriye yayılım için 10)	<ul style="list-style-type: none"> - Bugünkü hisse senedi değeri, - Değerin bir önceki güne göre tam değişimi, - Değişimin yönü, - İki gün öncesinden değişimin yönü, - Önceki güne ilişkin büyük değişiklik, - Son 10 günün değerleri (Geriye yayılım için) 	Sonraki günün hisse senedi değeri
Ticaret yapmak için tavsiye (Trippi ve	Çeşitli yapay sinir ağları + kurallar seti	3	<ul style="list-style-type: none"> - S&P 500 hisse senedi indeksinin açık değeri, - S&P 500 hisse 	Ticaret tavsiyesi: <ul style="list-style-type: none"> - Uzun - Kısa

DeSieno, 1992)			senedi indeksinin düşük değeri, - S&P 500 hisse senedi indeksinin kapalı değeri,	
Hisse senetlerinin sınıflandırılm ası (Kryzanowsk i, Galler ve Wright, 1993)	Boltzman makinesi	88	- 14 firmanın finansal oranları, - Firmaların finansal oranlarının birbilerine göre nispi oranı, - Ayrı ayrı endüstri ölçümlere karşı 5 finansal oranın bağıl performansının 20 özelligi, - Her bir makroekonomik faktör için, 35 yıl ve üzerinde % değişim	Hisse senedi gelirleri: - Pozitif - Nötr - Negatif
S&P's 500 Hisse senedi	Geriye yayılım	24	- Piyasadaki toplam arzın aylık büyüme hızı, M-1	Tahmin yapılan ay için, aylık ortalama

İndeksinin değer değişiminin tahmin edilmesi (Grudnitzky ve Osburne, 1993)			<ul style="list-style-type: none"> - S&P ve Altın vadeli işlem değerlerinin değişimi ve iniş-çıkış eğiliminin ölçümü, - Ayın sonunda, büyük spekülâtorlerin, borsa oyuncularının ve küçük tüccarların net % taahhütleri 	değerin değişimi
Hisse senedi değerlerinin performansın tahmini (Yoon, 1994)	Hibrid Yaklaşım (geriye yayılım + uzman Sistem)	4	<p>4 finansal oran:</p> <ul style="list-style-type: none"> - Bugünkü cari Oran - Özsermayedeki gelir, - Değer/özsermaye oranı, - Değer/satış oranı 	<p>Hisse senedi performansı:</p> <ul style="list-style-type: none"> - İyi - Kötü
Hisse senedi performansın tahmini (Swales ve Yoon,	Geriye yayılım	9	Başkanın mektubundan hissedarlara yinelenen temalar (nitel veriler)	<p>Hisse senedi performansı:</p> <ul style="list-style-type: none"> - İyi - Kötü

1992, 1996)				
Hisse senedi gelirinin tahmini (Chenoweth ve Obradovic, 1997)	Hibrid yaklaşımı (2 Geriye yayılım+uzman sistem)	6 günlük, 8 aylık veriler	Günlük veriler: - 30 yıllık Devlet Tahvili geliri, gecikmeli US Hazine bonosunun gelirindeki oran değişimi, - S&P Bileşik İndeksindeki gelir, - S&P Bileşik İndeksindeki gecikmeli gelir, Aylık veriler: - 30 yıllık Devlet Tahvili geliri, Devlet Tahvili gelirinin gecikme oranının değişimi, - US Hazine bonosu indeksi, - US Hazine bonosunun gecikme	Günlük ve aylık gelirlerin oranı (gelirin yıllık oranı, sistemin performansını ölçmek için sonra hesaplanır)

			oranı, - CPI, - S&P Bileşik İndeksi	
TOPIX İndeksinin tahmini (Kohara et al., 1997)	Geriye yayılım, Geri döngü	5	- Tokyo hisse senedi değişim değeri indeks (kapanış) - Dolardan yine değişim oranı - Faiz oranı - Ham petrolün değeri (dolar/varil) - Dow Jones ortalaması	1 – TOPIX Diğer gün için tahmin
İndeksdeki gelir değişkenliğin in tahmini (Donaldson, Kamstra, 1997)	Geriye yayılım	2	- MA sapma denklem model tahmini - GARCH model tahmini	1 – Piyasa indeksindeki değişkenliğinin kazancı

Hisse deęişkenlięinin tahmini (Refenes , 1997)	Geriye yayılım	6	- Paralı olma, - Yer deęiştirmek - Vade etkisi - Hisse senedi iniş çıkış eğiliminin ölçümü lag1, lag2, lag3	1 – kastedilen hisse senedinin iniş ve çıkış eğiliminin ölçümü
Hisse senedi deęeri tahmini (Saad et al., 1998)	Zaman gecikmesi, Geri Döngü ve Çok katmanlı Yapay sinir aęı	7	- Bugünün hisse senedi deęeri - Geri kalmış hisse senedi deęeri	1 – dięer günün hisse senedi deęeri

Tablo 4.'de görüldüğü gibi, araştırmacılar çok farklı modeller kullanmışlardır ancak modellerden birinin dięerlerine göre üstünlüğü gözlenmemektedir. Bu durum, yapay sinir aęları verimlilięi üzerine belirgin bir sistem oluşturmaya zorlaştırmaktadır. Girdi deęişkenlerinin sayısı 3'den (Trippi ve DeSieno, 1992) 88'e (Kryzanowski, Galler ve Wright, 1993) kadar deęişmektedir. Bununla beraber, girdilerin deęişkenlerin büyük çoğunluęu hisse senedi deęerleri ve finansal oranlardır. Swales

ve Yoon (1992, 1996) ve Kohara (1997) dışındaki bütün arařtırmacılar, genellikle aynı kaynaktan aldıkları nicel verileri kullanmışlardır. Bu kaynaklar: hisse senedi piyasası indeksleri (S&P, TOPIX, Dow Jones, v.b. gibi) veya Fortune 500 ve Business Week Top 1000'dir (Yoon, 1994). Yapay sinir uygulamalarında nitel verilerin ortaya sunulması ileride yapılacak arařtırmalar için fırsatlar yaratmaktadır. Donaldson ve Kamstra (1997) tarafından yapay sinir aęları ve istatistiksel metotlar birleřtirilerek yine farklı bir açılım saęlanmıştır.

3.3 Önceki Arařtırmada Kullanılan Yapay Sinir Aęı Metodolojisi Üzerine Karşılařtırma Analiz

Yöntemsel bir başlangıç noktası belirlemek için, yapay sinir aęı oluřturma yöntemleri üç şekilde gruplanabilir:

- (a) Kullanılan yapay sinir aęı mimarisine göre,
- (b) kullanılan yapay sinir aęı parametrelerinin ve yapısının optimizasyon teknięine göre,
- (c) Elde edilen ölçümlerin ve sonuçların deęerlendirmesine göre.

İř sektöründe yapay sinir aęı uygulamalarına bakıldığında, kullanım alanlarına (Li, 1994; Zahedi, 1997; Wong, 1997) ve yonteme (Zahedi, 1997; Zekic, 1998) yoğunlařan, bazı makaleler öne çıkmaktadır. Bu makaleler; iř, finans ve yatırım alanlarındaki kesin bir problem alanında kullanılacak özel bir yapay sinir aęı

yöntemi için, belirlenmiş bir kural grubunun yada kesin çözümün hala geliştirilmediğini göstermektedir. Çok sayıda araştırmacıya göre, yapay sinir ağı yaratma yönteminin içeriğinde, yapay sinir ağı mimarisinin tasarımı, eğitim metotları, test durumları, değerlendirmeler ve ağı uygulamaya geçirmek bulunmaktadır (Zahedi, 1993).

3.3.1 Önceki Araştırmada Kullanılan Yapay Sinir Ağı Mimarileri

Aşağıdaki tablo, problem alanına göre farklı uygulamalarda kullanılmış yapay sinir ağı algoritmalarını ve yapılarını göstermektedir. Tablodan da görülebileceği gibi, Geriye yayılım algoritması en yaygın kullanılan yapay sinir ağı mimarisidir ve diğer algoritmalar çok nadir uygulamalarda kullanılmıştır. Schoeneburg (1990) ve Refenes (1994) dışındaki çoğu araştırmacıya göre, üç katmanlı yapı daha etkin görünmektedir.

Tablo 5. Problem alanına göre yapay sinir ağı algoritmaları ve yapısı

Hisse Senedi Performansının Pozitif yada Negatif Olacağıın Tahmini	
Yapay Sinir Ağı Mimarisi	
Algoritma	Yapay Sinir Ağı Yapısı
1. Geriye yayılım (Swales ve Yoon, 1992, 1996)	1. 2,3 ve 4 katmanlı (9-3-3-2) ¹ (Swales ve Yoon, 1992, 1996)
2. Geriye yayılım (Trippi, DeSieno, 1992)	2. 6 ileri beslemeli yapay sinir ağı (Trippi ve DeSieno, 1992)

3. Boltzman makinesi (Kryzanowski, Galler, Wright, 1993)	3. 2 katmanlı sinir ağı(88-1) (Kryzanowski, Galler ve Wright, 1993)
4. Geriye yayılım (Yoon , 1994)	4. 1 katmanlı (4 girdi, 2 çıktı, 0'dan 9'a gizli değişken)
5. Geriye yayılım (Refenes, 1997)	5. 3 katmanlı (6 girdi, 1 çıktı, 2'den 5'e gizli değişken)

1.Parantezdeki sayılar, her bir katmandaki yapay sinir hücresinin sayısını göstermektedir, örneğin 9-3-3-2; birinci katmanda 9, ikinci katmanda 3, üçüncü katmanda 3, dördüncü katmanda 2 yapay sinir hücresinin bulunduğunu belirtmektedir.

Hisse Senedi Değeri Sayısal Tahminleri	
Yapay Sinir Ağı Mimarisi	
Algoritma	Yapay Sinir Ağı Yapısı
1. Geriye yayılım (Grudnitzky ve Osborne, 1993)	1. 3 katmanlı (24-24-1) (Grudnitzky ve Osborne, 1993)
2. Geriye yayılım (Schoeneburg, 1990)	2. 4 katmanlı (10-10-10-1) (Schoeneburg, 1990)
3. Basit algılayıcılar (Schoeneburg, 1990)	3. 2 katmanlı (40-1) (Schoeneburg, 1990)
4. ADALINE / MADALINE (Schoeneburg,	4. 2 katmanlı (40-1) (Schoeneburg, 1990)

<p>1990)</p> <p>5. Hybrid (2 Geriye yayılım yapay sinir ağı + uzman sistem)</p> <p>6. Geriye yayılım (Kohara, 1997)</p> <p>7. Geri beslemeli (Kohara, 1997)</p> <p>8. Geriye yayılım (Donaldson ve Kamstra, 1997)</p> <p>9. Zaman Gecikmesi, Geri beslemeli, Olasılıklı (Saad, 1998)</p>	<p>5. günlük veriler-3 katmanlı (6-4-1), aylık veriler-3 katmanlı (8-31)</p> <p>6. 3 katmanlı (5-5-1,6-6-1)</p> <p>7. 3 katmanlı (25-20-1, 26-20-1)</p> <p>8. 3 katmanlı (2-3-1), optimize edilmiş</p> <p>9. 3 katmanlı, min. 7-3-1, max. 10-7-1</p>
<p>1. Hybrid yaklaşımı (Geriye yayılım yapay sinir ağı + uzman sistem) (Yoon, Guimaraes ve Swales, 1994)</p>	<p>1. 3 katmanlı (4-7-2) (Yoon, Guimaraes ve Swales, 1994)</p>

Yukarıdaki çalışmalardan farklı olarak istatistiksel metotların (MAV ve GARCH) çıktılarının, yapay sinir ağı içinde girdi değişkenleri olarak bütünleştirilmesinin

uygulandığı bir hibrid yaklaşım biçiminde uygulamıştır (Donaldson ve Kamstra,1997).

Yapay sinir adı yönteminin diğer özelliği transfer fonksiyonudur. Yapay sinir ağı uygulamalarının büyük çoğunluğunda, deneylerde optimizasyonu yapılan α ve η değiştirilebilen öğrenme parametreleri ile sigmoid transfer fonksiyonunun kullanıldığı belirlenmiştir. Uygulamalardaki genel eğilim, iki veya daha fazla yapay sinir ağının tek yapay sinir ağı sisteminde birleştirilmesi veya uzman sistemler, genetik algoritmalar ve doğal dil işlemi gibi yapay zekâ metotlarının yapay sinir ağı sistemine dâhil edilmesidir. Kohonen, Hopfield ve diğer algoritmaların sayısı, hisse senedi piyasasındaki yapay sinir ağı uygulamalarında oldukça düşüktür. Bazı araştırmacılar, diğer algoritmaları hisse senedi piyasası uygulamalarında gelecek araştırmalar için rehber olarak önermesine rağmen, (Schoeneburg, 1990; Yoon, 1994) bu mümkün olmamıştır. Bunun nedeni o yapay sinir ağı algoritmalarının tahmin yapmaktan çok sınıflandırma için uygun olmasından ileri gelmektedir (Zahedi, 1993)

3.3.2 Önceki Araştırmada Kullanılan Yapay Sinir Ağı Yapı ve Parametrelerinin Optimizasyon Teknikleri

Tablo 5. de adı geçen birçok araştırmacı, yapay sinir ağlarındaki gizli yapay sinir hücrelerinin sayısını deneysel olarak belirlerler. Bazıları, sadece bir yapıyı test ederken (Grudnitzky ve Osburne, 1993), diğerleri gizli yapay sinir hücrelerinin ve gizli katmanların sayısını değiştirmektedir (Schoeneburg, 1990; Swales and Yoon, 1992, 1996; Trippi ve DeSieno, 1992) ve yapı için gerçek zamanlı optimizasyon

prosedürü olmadan bütün kombinasyonları test etmektedir. Bazı arařtırmacılar ise karşılıklı onaylama prosedürünü uygulamaktadır, bunlarda (Donaldson ve Kamstra ,1997), standart on katmanlı karşı onaylama seçim yapısını kullanmaktadır. Bu yapının işleyişı aşığıdaki gibidir:

1. Model, örnek içindeki verilerin (örnek içindeki eğitim verileri) bölümleri üzerinden tahmin edilir,
2. Belirlenen model, kalan örnek verilerin tahmininde kullanılır,
3. Ağ örnek dışındaki veri setleriyle test edilir,
4. Örnek içindeki verilerin altkümümesi dışarıda bırakılır ve örnek dışındaki verilerin altkümümesi dâhil edilir,
5. 1'den 4'e kadar olan basamaklar, örnek dışındaki tahminler, başlangıçtaki örnek içindeki seti oluşturuncaya kadar tekrar edilir.

Öğrenme hızı ve Delta kuralındaki momentum gibi ağ öğrenme parametrelerini ve diğer parametreleri optimize eden arařtırmacıların sayısı daha azdır.

İleri yapı ve parametre optimizasyon tekniklerinin, hisse senedi piyasasındaki tahminlerde kullanılan yapay sinir ağı uygulamaları için yeterli şekilde arařtırılmamış olunduğı sonucuna varılabilir.

3.3.3 Önceki Arařtırmanın Sonuçları ve Değerlendirme Ölçümleri

Önceki makalelerin yazarları, ağlarında değişik değerlendirme ölçümleri kullanmış olsalar da doğruluk oranı en sık kullanılanıdır. En fazla analiz edilen uygulamalarda yapay sinir ağı, çoklu lineer regresyon analizi (Refenes,1994), sayısal analiz (Swales ve Yoon, 1992, 1996) ve diğer yöntemler daha iyi istatistiksel metotlarla sonuçlanmaktadır. Yapay sinir ağı sisteminin doğruluk oranı %58 ile %90 arasında değişmektedir (Schoeneburg, 1990). Tablo 6. analiz edilen makalelerdeki yapay sinir ağı değerlendirme ölçümlerini ve sistemlerin sonuçlarını göstermektedir.

Tablo 6. Analiz edilen yapay sinir ağı uygulamalarındaki yapay sinir ağı değerlendirme ölçümleri ve sonuçları

Yapay Sinir Ağı Uygulamaları	Değerlendirme Ölçümleri (Performans Kriteri)	Sonuçlar
1. Hisse senedi performans modeli (Refenes, , 1994)	1. RMS 2. RMSE (ortalama karekök hata)	1. Yapay sinir ağının RMS =0.066, regresyonun RMS =0.128
2. Hisse senedi gelir Değişkenliği İndeksinin tahmini (Donaldson ve Kamstra, 1997)	ve RMAE (ortalama kesin kök hata)	2. RMSE: 0,00135, 0.00117, 0.00124, 0.00067 RMAE: 0.00862, 0.00746, 0.00930, 0.00665
3. Hisse senedi değeri tahmini	3. Kesinlik oranı 4. Kesinlik oranı	3. BASF için %90, Mercedes için %58

<p>(Schoeneburg, 1990)</p> <p>4. Hisse senedi performansının tahmini (Swales ve Yoon, 1992, 1996)</p> <p>5. Hisse senetlerinin sınıflandırılması (Kryzanowski, Galler ve Wright, 1993)</p> <p>6. Hisse senedi performansının tahmini (Yoon, 1994)</p>	<p>5. Kesinlik oranı</p> <p>6. Kesinlik oranı</p>	<p>4. Yapay sinir ağı %77, ayırıştırma çözümlemesi %65</p> <p>5. %72</p> <p>6. Yapay sinir ağı %76, ayırıştırma çözümlemesi %63</p>
<p>7. Kastedilen hisse senedi değişkenliğinin tahmini (Refenes , 1997)</p>	<p>7. Korelasyon ve kesinlik oranı</p>	<p>7. Korelasyon: yapay sinir ağı 0,676, Box-Jenkins 0,50, MLR 0,63, kesinlik oranı: yapay sinir ağı %74, Box-Jenkins %65,3, MLR %73,46</p>
<p>8. TOPIX İndeksinin tahmini (Kohara, 1997)</p>	<p>8. Ticaret karı</p>	<p>8. En yüksek kar = %40</p>

9.	Hisse senedi gelir tahmini (Chenoweth, Obradovic, 1997)	9.	Gelirin yıllık oranı (Chenoweth, Obradovic, 1997)	9.	En iyi sonuç: günlük veride %13,35, aylık veride %16,39
10.	Hisse senedi değeri tahmini (Saad, 1998)	10.	Hata alarının yüzdesi	10.	En iyi %0, en kötü %13,6

RMS hata ve değışikleri (MSE,MAE) en sık kullanılan hata fonksiyonlarıdır (Refenes ,1994). Birçok yazar, gelir oranı, toplam gelir, ortalama gelir veya kar gibi hisse senedi tahminleri için daha uygun olan diđer ölçümlerinde kullanılabileceğini söylemektedir. Bazı makaleler (Chenoweth ve Obradovic, 1996; Refenes, 1997; Saad, 1998), gelirin yıllık oranı veya korelasyon gibi doğrulukların dışında metrikler kullanılmaktadır. Piyasadaki yanlış alarmları minimize etmek için, doğruluk yüzdesinin bir biçimini kullanan arařtırmacılar da vardır (Saad,1998). Doğruluk en fazla %70'den % 80 'e kadar ayarlanabilmekte ve yapay sinir ağlarını kullanmanın riski hala yüksek olmasına rağmen, yapay sinir ağları, oranlarda %5–20 daha yüksek doğruluk için istatistiksel metotlarla birleřtirilebilmektedir (Refenes,1997; Swales ve Yoon, 1992). Ayrıca, Geriye yayılım algoritmasının diđer yapay sinir ađı algoritmalarından daha yüksek doğruluk oranına sahip olduđu ve basit algılayıcıların

en az hata yapan algoritma olduğu hemen tüm arařtırmacılar tarafından belirtilmektedir.

Hibrid yaklařımlarının sonuçları çok ümit vericidir. İstatistiksel metotların çıktılarını yapay sinir ağıının girdileri olarak kullanarak, çok düşük RMS ve MAE elde eden arařtırmacılar mevcuttur (Donaldson ve Kamstra,1997). Test edilen özel istatistiksel metotlarla karşılaştırıldığında (MAV ve GARCH), yapay sinir ağıları önemli ölçüde iyi performans göstermişlerdir. Ayrıca yapay sinir ağıları, hisse senedi iniş ve çıkış değerlerinin tahmininde Box-Jenkins ve çoklu regresyondan daha iyi performans göstermişlerdir. (Refenes,1997).

Buna rağmen, yapay sinir ağılarını uzman sistemlerle birleřtiren arařtırmacılar yapay sinir ağıının doğruluk yüzdesini belirtmemişlerdir (Yoon, 1994). Bu nedenle, arařtırmacıların yapay sinir ağılarının uzman sistemlerle birleřtirilmeleri durumunda, birleřtirilmeden önceki durumlarına kıyasla daha yüksek doğruluk oranları elde ettiklerini belirtmelerine rağmen bu uygulamalar diğerleriyle karşılaştırılamamıştır. (Yoon, 1994).

3.4. Önceki Arařtırmaların Yararları, Sınırlamaları ve Sonuçları

3.4.1 . Önceki Arařtırmaların Yararları

Makalelerde, problem alanı ve kullanılan yapay sinir ağı metodolojisiyle ilgili birçok faydalı bilgi verilmiştir. Yapay sinir ağı uygulamalarının genel yardımı, sağlam ve kesin olmayan verilerle iş yapma yeteneğidir. Bu nedenle, yapay sinir ağıları hisse senedi piyasasında, ya hisse senedi değerinin ya da hisse senedi gelirlerinin

tahmininde etkili şekilde kullanılabilir. Karşılaştırmalı analizden de görülebileceği gibi; geriye yayılım algoritması, diğer yapay sinir ağı algoritmalarından her durumda değil ama genelde daha yüksek doğrulukta tahmin yapma yeteneğine sahiptir. Mimarilerin test edildiği makalede, geri dönüşümlü yapay sinir ağı, geriye yayılımdan daha kötü performans göstermiştir (Donaldson ve Kamstra, 1997).

Makalelerdeki veri modellerinin farklılığı, bazı durumlarda (bazı veriler elde edilemediğinde) yapay sinir ağının esnekliğini ve etkinliğini gösterdiğinden yarar olarak düşünülebilir. Yapay sinir ağının klasik tahmin metotlarından ve çoklu regresyon analizi (Trippi ve DeSieno, 1992, Refenes, 1997) ve ayırıştırma analizi gibi istatistiksel metotlardan daha iyi performans gösterdiği kanıtlanmıştır. Bununla birlikte yapay sinir ağları istatistiksel yöntemlerle birleştirildiğinde, çeşitli sinir ağları değerleri çok doğru şekilde tahmin edebilmektedirler, çünkü çıktı hesaplamak için gerekli farklı özellikteki veri setleri üzerine yoğunlaşabilmektedirler.

Analizler, yapay sinir ağı metodolojilerinin, uzman sistem gibi metotlarla farklı birleşim olasılıklarını da göstermektedir. Farklı problem alanlarında, karar vermek için kuralların işleme geçirilmesi ve sonuçların açıklanması için, yapay sinir ağının hesaplama yeteneğinin uzman sistemin hesaplama yeteneği ile kombinasyonu çok etkili bir zekâ desteği olabilir.

3.4.2 Önceki Araştırmalardan Ortaya Çıkan Sınırlamalar

Hisse senedi piyasasında yapay sinir ağı uygulamalarıyla ilgili yapılan çalışmalarda, yapay sinir ağlarının dikkat edilmesi gereken belli noktaları ortaya çıkmıştır:

- (1) Yapay sinir ađları, önceki durumlara ait çok sayıda veri gerektirmektedir(Yoon, 1994).
- (2) En iyi ađ mimarisi hala bilinmemektedir (Schoeneburg, 1990);
- (3) Çok karmaşık ađlar, sonuçların güvenilirliği azalabilmektedir (Yoon, 1994)
- (4) Sonuçların istatistiksel olarak anlamlı olması gerekmektedir(Schoeneburg, 1990)
- (5) Çok dikkatli veri tasarımı gereklidir (Refenes, 1994).

İlk sınırlandırma, verilerin kullanılabilirliği ile bağlantılıdır ve bazı araştırmacılar, etkili hisse senedi tahminleri için çok sayıda veri seti toplanması gerektiğini ispatlamışlardır, örneğin; Schoeneburg, girdi için 2000–3000 tane veri seti kullanmıştır (Schoeneburg, 1990). Bu sınırlandırma, çok fazla eski verisi bulunmayan tahmin özneleri, örneğin yeni kurulmuş şirketler, için problem oluşturmaktadır. Etkin bilinen bir mimari olmaması sınırlandırmasının yakın gelecekte bir çözümü hala bulunmamaktadır. Araştırmacıların çabaları, farklı topolojilerin ve veri modellerinin testleri üzerine yoğunlaşsa da, sonuçlar hala özel durumlara bağlıdır. Sonuçların güvenilirliği ile ilgili üçüncü sınırlamayı gidermek için, değişik ađ mimarilerinin test edildiđi deneyler gereklidir. Sonuç olarak, veri modellerinin çeşitliliđi, veri tasarımının sistematik olarak analiz edilemediđini göstermektedir ve hemen hemen her yazar farklı veri modeli kullanmaktadır.

3.4.3 Önceki Araştırmalarla İlgili Sonuçlar

Önceki araştırmalardan aşağıdaki sonuçlara varılabilir:

- (1)Yapay sinir ađları, hisse senedi piyasası tahminlerinin etkili bir metodudur, fakat problemler için kesin uygulama yolları öneren bir reçete yoktur.
- (2)Yapay sinir ađı uygulamalarının ümit verici problem alanı, hisse senedinin her açıdan modellemesi olmasına rağmen, yapay sinir ađları en fazla hisse senedi deđerlerinin ve gelirlerinin tahmininde kullanılmıştır.
- (3)Geriye yayılım algoritması en yaygın uygulamadır, fakat bazı arařtırmacılar tarafından yapay sinir ađı ile diđer yapay zekâ metotlarının birleřtirilmesinin önemi belirtilmiştir.
- (4)Hisse senedi piyasası uygulamalarındaki yapay sinir ađı yapısı, ileri teknikler kullanılarak çok nadir olarak optimize edilir.
- (5)Hisse senedi gelirleri söz konusu olduđunda, **dođruluk oranı** en sık deđerlendirilen ölçümdür.
- (6)Yapay sinir ađının yararları, kesin olmayan verilerle ve diđer metotlarla birleřimlerinde bile, dođru bir şekilde tahmin yapma yeteneğindedir.
- (7)Kesin bir problem alanında en iyi topolojiyi tanımlamak için, eksik testlerle, eksik veri tasarımıyla ve diđer yetersizliklerle tanımlama yapmak zorunluluđu bulunmaktadır.

Birçok arařtırmacı, daha fazla finansal oran ve nitel verilerin model içerisinde yer almasının önemini belirtmişlerdir. Buna ek olarak, deđişik zaman aralıklarının kullanımı sıklıkla tavsiye edilmektedir. Bazı arařtırmacılar haftalık ve aylık verileri kullanmasına rağmen, hisse senetleri, genellikle günlük veriler temel alınarak tahmin edilir (Grudnitzky ve Osburne, 1993). Ayrıca, gelecekteki arařtırmaların Hopfield's, Kohonen gibi pek kullanılmayan ađ çeřitlerinin incelenmesi üzerine yoğunlařacağı

belirlenmektedir. Son olarak hemen hemen bütün arařtırmacılar; sınırlandırmaları kaldırmak için, yapay sinir ađlarının diđer yapay zekâ metotlarıyla birleřtirilmesi gerektiđini vurgulamaktadırlar.

4.YAPAY SİNİR AĞININ OLUŞTURULMASI

Bundan önceki bölümlerde sinir ağının genel yapısı, özelliklerini belirtilmiş, ikinci bölümde ise tahmin yapabilmek için gerekli olan temel ekonomik teoremlerden bahsedilmiştir. Son olarak da bugüne kadar yapılmış çalışmalar incelenerek bulunan sonuçlar analiz edilmiştir.

Bugüne kadar sinir ağlarının hisse fiyatı tahmini için kullanılması ile ilgili birçok çalışma yapılmıştır. Bu çalışmalar sonucunda belli ortak adımlar üzerinde durulmaktadır.

Bir sinir ağının finansal ağlar için tahmin yapabilmesi için gerekli temel adımlar genel olarak şu şekilde sıralamak mümkündür:

1. Tahmin yapılacak mali unsurun belirlenmesi
2. Tahmin yapılacak sürenin belirlenmesi
3. Tahmin yapılacak hisse ile ilgili girdilerin belirlenmesi, gerekli bilgilerin toplanması ve bu girdiler arasındaki ilişkilerin belirlenmesi
4. Girdilerin işlenmesi ve ayrıştırılması: Girdilerin ağın yapısına uygun hale getirilmesi
5. Ağ yapısının seçilmesi (Ağ topolojisi, katman sayısı, ağ parametreleri, öğrenme algoritmaları)
6. Ağın öğrenmesi ve test edilmesi. Farklı değişkenler ile ağın tekrar tekrar testi

Tahmin yapılacak mali unsurun belirlenmesi

İlk olarak tahmin edilecek finansal varlığın tipi belirlenmelidir. Bir hisse senedinin günlük kapanış fiyatı veya gün içindeki maksimum ya da minimum değeri tahmin edilebileceği gibi, bir indeksin günlük veya haftalık tahmininde mümkündür. Bunun yanında genel kabul fiyat tahmininin yanında fiyatların yönünün belirlenmesinde mümkün olduğu yönündedir.

Tez konusu olarak hisse senetleri seçilmiştir. Tahmin yöntemleri kullanılırken hisse senedi tahmini üzerine odaklanılmıştır.

Tahmin yapılacak sürenin belirlenmesi

Tahmin yapılacak sürenin belirlenmesi oldukça kritik bir öneme sahiptir. Uzun vadede makro ekonomik etkilerin fazlalığı nedeniyle tahmin yapmak kolaylaşmaktadır ancak genelde yatırımcılar kısa süreli sonuçlara odaklanmaktadır. Yapılan çalışmalarda genel olarak bir haftalık ya da daha kısa süreli yatırım sonuçlarına yönelik tahminlere daha çok ilgi gösterilmektedir. Bu nedenle tez süresince yapılan çalışmalarda günlük ve haftalık tahmin yapılmasına odaklanılmıştır.

Tahmin yapılacak hisse ile ilgili girdilerin toplanması ve bu girdiler arasındaki ilişkilerin belirlenmesi

Hisse senedi tahmininde daha öncede belirtildiği gibi üç ile seksen sekiz arasında farklı girdi kullanılabilir. Ancak daha önce belirttiği zaman serileri diğer

karmaşık yapay sinir ağı yöntemleri kadar verimli çalışmaktadır. Bu nedenle zaman serileri kullanılacağı için sadece hisse senedinin bir önceki dönemlerdeki kapanış değerleri yapay sinir ağına girdi olarak verilecektir. Çalışma boyunca 2003 yılından itibaren tüm hisse senetlerinin kapanış değerleri girdi olarak kullanılmıştır.

Girdilerin işlenmesi ve ayrıştırılması: Girdilerin ağına yapısına uygun hale getirilmesi

Girdi değerleri olarak kullandığımız hisse senedi kapanış değerleri direk olarak ağına verilememektedir. Bu nedenle tez konusu dahilinde geliştirilmiş olan program kendi içerisinde almış olduğu verileri sinir ağına anlayabileceği şekilde korelasyonunu normalize etmekte ve ölçeklendirmektedir.

Ağımız tarafından kullanılan ölçeklendirme işlemi, verileri girdi dosyasından yüklerken, minimum değer, maksimum değer, ortalama değer, değerlerin sayısı ve standart sapma belirlemektedir. İlk önce, ortalama değer hesaplanmakta: ortalama olarak okunana her bir değer eklenmekte ve değerlerin sayısına bölünmektedir. Eğer N veri setinin boyutu ise, standart sapma belirtilen formülle hesaplanır:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Yapay sinir hücreleri, çok etkili olmak için girdi verilerini kesin girdi setleri içinden alırlar. 3 ve 110 Aralığında değişen, Ahold'un hisse senedi değerlerine benzeyen girdi setleri; gizli katmandaki sinir hücrelerinin geniş girdi verilerini ya 0 ya da +1'e

sıkıştırılan sigmoid aktivasyon fonksiyonuna sahip olmasından dolayı, faydalı olmayacaktır. Bu nedenle, tez konumuzda verileri, yapay sinir hücrelerini öğrenmesini engellemeyen veya ezmeyen bir aralıkta seçilmiştir. Girdileri -1'den -1'e seçmek en uygun çözüm olarak kabul edilmiştir. Girdileri belirtilen fonksiyona göre ölçeklendirebiliriz:

$$x' = \frac{x - \bar{x}}{sd}$$

Ortalamayı çıkarmak ve standart sapmaya bölmek içinse aşağıdaki formüller kullanılmıştır. Bunun yanında çıktıların ölçeklendirilmesi, için aşağıdaki formül yardımıyla program içinden yapılmıştır:

$$x' = \frac{x - \min}{\max - \min}$$

Ölçeklendirmeden sonra, girdiler sıfır çevresinde yerleşecekler ve çıktılar sıfır ile bir arasında yer alacaklardır. Bu noktaya ulaşmak istememizin bir diğer nedeni de, birimlerin aktivasyonu için kullanılan sigmoid fonksiyonunun sıfır ile bir arasında sınırlandırılmış olmasıdır. Çıktı üretildikten sonra, geriye gerçek değere ölçeklendirilecektir (gerçek tahmin değerine):

$$x = (x'(\max - \min)) + \min$$

Ağ yapısının seçilmesi (Ağ topolojisi, katman sayısı, ağ parametreleri, öğrenme algoritmaları)

Daha önceki çalışmalardan edinilen bilgiler bize birçok ağ yapısının bulunmasına karşın, kullanım açısından en doğru sonuçlara ulaştırmanın geriye doğru yayılım algoritmasının olduğunu göstermiştir. Yine daha önceki çalışmaların ışığında 3. katmanlı yapının daha az ya da daha fazla katmanlı yapılara oranla daha doğru sonuçlar ürettiği görülmüştür.

Bunun yanında ağın her bir sinir hücresinin başlangıç ağırlıkları rasgele bir şekilde verilmiştir.

En iyi parametreleri belirlemenin en yaygın yolu deneme ve yanılmadır. Problem için en uygunu belirlemenin yolu farklı seçenekleri araştırmaktır ancak yukarıda belirtilen bilgiler ışığında tez çalışmamızın ilk başlangıç parametreleri aşağıdaki gibi olmuştur:

Girdi birimlerinin sayısı	5
Gizli birimlerin sayısı	3
Eşik	0,1 veya daha az
Eta	1,0
Alfa	0,9
Rasgele belirlenmiş başlangıç ağırlıkları	[-0,01, 0,01]

Son MSE	0,01
---------	------

Ağın eğitilmesi ve test edilmesi

Ağ eğitiminde, ağ topolojisi ile başlayıp ve ağı eğitim verilerini kullanarak eğitmek gerekmektedir. Minimum eğitim hatasına ulaşıldığında, eğitilmiş ağa test verilerini uygulayıp ve hatayı not edilmelidir. İşlemi, aynı ağ topolojisinde başlangıç ağırlık değerlerinin farklı bir seti için tekrar başlatmak ve eğitim ve test setleri ile daha iyi bir hata değerine ulaşıp ulaşılamadığını görmek gerekmektedir. Bunun nedeni, ağın ilk denemede bir yerel minimum noktasında kalabilecek olması ve rasgele atanmış ağırlık değerlerinin farklılaşmasıyla belki de daha iyi sonuçlar alınabilecek olmasıdır.

Bunun yanında bir girdiyi çıkararak ve bu işlemi tekrarlayarak, eğitim ve test setleriyle elde edilebilecek en iyi hatayı not edip girdilerin sayısını azaltılmalıdır.

Test setini, optimize edilmiş ağın nasıl performans gösterdiğini görmek için kullanılmalı, eğer elde edilen hata tatmin edici beklenenden değerden yüksek ise, ağ yeniden tasarlanmalı ve yeniden test edilmelidir.

Bunun yanında yeni bilgiler ağın performansını arttırabilecekse, ağ tekrar eğitilmelidir. Eğer günlük temelde değilse, hisse senedi tahmini problemleri için ağ en azından haftada bir kez tekrar eğitilebilir. Eğer ağ yeni bilgilerle iyi genelleme yapmak için daha uzun görünmüyorsa, tasarım aşaması tekrar girilmelidir.

Test verilerimiz YTL kabulünden sonraki tüm hisse senedi verilerini kapsamaktadır. Burada test verilerinin yüzde 10'luk bir kısmı test verisi olarak ayrıştırılmış, geri kalanlar eğitim verisi olarak kullanılmıştır.

Test verisi hazırlanırken beş günlük arda arda kapanış değerleri bir test verisi olarak sinir ağına verilmiştir. İkinci test verisi oluşturulurken de gün bir iş günü ileriye alınarak bir sonraki gün kapanış değeri geri kalan dört kapanış değerine eklenerek yeni bir test verisi elde edilmiştir.

Ağın eğitilmesi için 2000'e yakın eğitim ve test verisi oluşturulmuştur.

5.Yapay Sinir Ađı Programı

Yapay Sinir ađları programı kullanmanın bir yolu hali hazırda var olan sinir ađı uygulaması için hazırlanmış olan programları alıp bunlara var olan bilgileri girmektir. Ancak bu birçok sıkıntıyı beraberinde getirmektedir. Örneđin böyle bir programı dağıtmanın her hangi bir yolu yoktur. Yazılmış olan program herhangi bir şekilde izin alınmaksızın kullanılamaz. Diğer bir sıkıntı ise bu kullanım tarzında kullanıcıya internet üzerinden hizmet verme şansı yoktur. Bunun yanında programcının kendi yazdığı kodlar üzerinde deđişiklik yapması, program geliştiriciye büyük bir avantaj sağlar çünkü bu sayede geliştirici program üzerinde her türlü deđişikliđi yapabilmektedir.

Yukarıdaki ifade edilen nedenlerden dolayı bu tez çalışmasında kullanılan program sıfırdan geliştirilmiştir. Programın kodlarına Ekler bölümünde ulaşılabilir.

Programın geliştirildiđi ortam Microsoft .NET ortamıdır,programlama dili c# olarak seçilmiştir.Programın çekirdek modülü olan Hücre,katmanlar ve temel ađ yapısı internet ortamı veya masa üstü ortam fark etmeksizin çalıştırılabilmektedir.Bu tez çalışmasında kullanıcıların rahat kullanılabilmesi açısından masaüstü uygulama tercih edilmiştir.Ancak temel fonksiyonlar kolay bir şekilde internet ortamından kullanılacak bir yapay sinir ađı programına temel oluşturabilir.Bunun yanında program Microsoft Windows platformlarının tümünde çalışmaktadır.

Programın temel yapısının ileriki çalışmalarda uzman sistemler ile birleştirilebileceği düşünülerek arayüz tanımlamaları açık bir şekilde tanımlanmıştır.

Programda sinir hücreleri katmanları, katmanlarsa sinir ağını meydana getirmiştir. Programın içerisinde kullanılan önemli temel yapılar ve yaklaşımlar aşağıdaki gibidir

5.1 Veri Yapıları ve Fonksiyonlar

Bilindiği üzere hücre ve hücrenin oluşturduğu katmanlar ve bu katmanlar arasındaki bağlantılar yapay sinir ağını oluşturmaktadır.

5.1.1 Sinir Hücresi

Program içerisinde bir hücre aşağıdaki veri yapısı ile gösterilmiştir:

Değişken Adları
A : <i>Single</i>
F : <i>ActivationFunction</i>
Item : <i>Single</i>
Last Threshold : <i>Single</i>
Last W : <i>Single[]</i>
N Inputs : <i>Int32</i>
Output : <i>Single</i>
OutputPrime : <i>Single</i>
Randomization Max : <i>Single</i>
Randomization Min : <i>Single</i>

Threshold : <i>Single</i>
WS : <i>Single</i>

“N_Inputs” deęişkeni sinir hüresine gelen girdi sayısını, “Item” deęişkeni ise her bir sinir dendritinin aęırlık deęerini tutar. Daha sonra bu aęırlık deęerleri test sırasında girdiler ile çarpılarak sinir hüresi üzerindeki toplam fonksiyonu ile sinir hüresi çıktısı hesaplanır. Bu toplam “WS” deęişkeni içerisinde tutulmaktadır. F deęişkeni sinir hüresinin hangi aktivasyon fonksiyonu ile çalışacağını bilgisini taşır. Bu deęişken Aktivasyon fonksiyonunda kullanılır ve “WS” deęişkeninde tutulan deęeri girdi olarak alan Aktivasyon Fonksiyonu “Output” deęişkeninde bu bilgiyi tutar. “Output” deęişkeninin hesaplanmasında kullanılan eşik deęeri ise “Threshold” deęişkeni içerisinde tutulur. Hem sinir hüresinin girdi aęırlıkları hemde eşik deęeri bilgilerinin tutulduęu deęişkenler sinir aęı oluşturulurken rastgele deęerlere atanırlar. “Randomization_Max” ve “Randomization_Min” deęerleri yukarıdaki iki rastgele deęer atama işlem, sırasında oluşturulan deęerlerin alt ve üst sınırlarını belirler.

“Output Prime” deęişkeni yukarıdaki mekanizmanın dışında olup, geriye yayılım algoritması içerisinde kullanılan fonksiyonun türevi görevini görür ve “Output” deęişkenin taşıdığı sinir hüresi çıktı bilgisinin türev bilgisini taşır.

Yukarıdaki veri yapısında “A”, “Last_Threshold”, “Last_W” deęişkenleri sistemin çalışmasını sağlayan, geriye yayılım algoritmasında deęer taşıyıcılığı yapan deęişkenlerdir.

Metod Adları:
ComputeOutput() : <i>Single</i>
randomizeAll() : <i>Void</i>
randomizeThreshold() : <i>Void</i>
randomizeWeight() : <i>Void</i>

“randomizeWeight” metodu sinir hücreninin girdilerinin ağırlıklarını rastgele bir şekilde atarken, “randomizeThreshold” metodu sinir hücreninin eşik değerinin rastgele bir şekilde atanmasını sağlar. Bu iki metodu birlikte çağıran metod ise “RandomizeAll” metodudur. Her üç metoddada sinir ağı oluşturulurken çağrılmaktadır.

Sinir hücreninin en önemli metodu “ComputeOutput” metodudur. Bu metod sayesinde sinir hücreninin girdiler sonunda ürettiği sonuç bulunmaktadır.

5.1.2 Sinir Ağı Katmanı

Program içerisinde yer alan hücrelerin birleşimi bir katman oluşturmaktadır. Bunlar girdi katmanı, çıktı katmanı ya da ara katmandan herhangi biri olabilmektedir. Katmanlarla ilgili bilginin tutulduğu yapı aşağıdaki gibidir:

Değişken Adları :
Item : <i>Neuron</i>
Last_Output : <i>Single[]</i>
N_Inputs : <i>Int32</i>
N_Neurons : <i>Int32</i>

“N_Neurons” değişkeni bulunulan katmanda kaç tane sinir hücresi olduğu bilgisini taşırken, “N_Inputs” değişkeni ise bu katmanda yer alan hücrelerin kaç tane girdisi olduğu bilgisini taşır. Buradaki önemli nokta aynı katmanda bulunan her bir hücrenin aynı sayıda girdi taşımasıdır.”Last_Output” değişkeni katmanda bulunan her bir sinir hücresinin çıktıların tutulduğu değişkendir.”Item” değişkeni ise katmanda bulunan her bir sinir hücresinin ayrı ayrı bilgilerinin tutulduğu değişkendir. Burada bahsedilen sinir hücrelerinin bilgisi bir önceki bölümde bahsedilen sinir hücreleridir. Bir bakıma birden sonlu bir doğal sayıya kadar olan hücre miktarı birleşerek bir ağ katmanı oluşturmaktadır.

Metod Adları:
Output() : <i>Single[]</i>
randomizeAll() : <i>Void</i>
randomizeThreshold() : <i>Void</i>
randomizeWeight() : <i>Void</i>
setActivationFunction() : <i>Void</i>
setRandomizationInterval() : <i>Void</i>

Yukarıda ki metotlardan “RandomizeThreshold”, ”randomizeWeight” bu katmandaki her bir sinir hücresi için sinir hücreleri içerisinde bulunan “RandomizeThreshold”, ”randomizeWeight” metodlarını çağırılmaktadır. Bu sayede katmandaki bütün eşik değerleri ve girdi ağırlık değerleri rastgele bir değere atanmaktadır.”RandomizeAll” metodu ise her iki metodun arka arkaya çağrılmış halidir.”setActivationFunction” katmandaki her bir hücrenin aktivasyon fonksiyonunu değiştirirken, “setRandomizationInterval” metodu sinir hücrelerinde rastgele atama yapılırken kullanılan alt ve üst sınırları değiştirir.

Sinir Ağı katmanının en önemli metodu “Output” metodudur ve bu metod sayesinde katmanın ürettiği sonuç öğrenilebilir.

5.1.3 Sinir Ađı

Sinir Ađı sinir hücreleri tarafından oluşturulmuş olan sinir ađı katmanlarının birleşmesinden oluşur. Temel veri yapıları ve metodları aşağıdaki gibidir.

Deđişken Adları :
Item : <i>Layer</i>
LearningAlg : <i>LearningAlgorithm</i>
N_Inputs : <i>Int32</i>
N_Layers : <i>Int32</i>
N_Outputs : <i>Int32</i>

Sinir Ađının kaç tane girdisi olduđu “N_Inputs”, kaç tane çıktısı olduđu “N_Output”, kaç tane katmanı olduđu “N_Layers” deđişkenlerinde tutulur. Sinir Ađının kullandığı öğrenim algoritması “LearningAlgorithm” içerisinde saklanır. “Item” deđişkeni ise sinir ađının sahip olduđu her bir katmana ait bilgileri içerir. Bu katman bilgileri bir önceki bölümde anlatılmış olan bilgilerdir.

Metod Adları:
load() : <i>NeuralNetwork</i>
Output() : <i>Single[]</i>
randomizeAll() : <i>Void</i>
randomizeThreshold() : <i>Void</i>
randomizeWeight() : <i>Void</i>
save() : <i>Void</i>
setActivationFunction() : <i>Void</i>
setRandomizationInterval() : <i>Void</i>

Daha önceki bölümlerden aşına olduğumuz “SetActivationFunction” ve “SetRandomizationInterval” , sinir ağının aktivasyon fonksiyonunu ve rastgele atanan değerlerin alt ve üst sınırlarını belirler. ”RandomizeThreshold” metodu ise sinir ağının sahip olduğu her bir hücre için eşik değerlerini değiştirirken, “RandomizeWeight” metodu sinir ağındaki her bir hücrenin girdi değerleri ağırlıklarını rastgele değerlere atar.”RandomizeAll” ise bu iki metodu kullanır ve bu metodların yaptığı işi tek başına yapar. ”Save” metodu bir sinir ağını daha sonraki kullanımlar için saklamaya yararırken, “Load” metodu tersi bir şekilde saklanmış olan sinir ağının yüklenmesini sağlar. En son metod olan “Output” ise tüm sinir ağının sonucunun öğrenilmesini sağlar.

6.Testler ve Sonular

6.1 Genel Test ve Eđitim Bilgileri

Yapay sinir ađı programının eđitimi sonucunda oluřan tahminler genel olarak řimdiye kadar iki farklı řekilde deđerlendirilmiřtir. Bunlardan ilki ve en ok kullanılanı, hisse senedi ya da endeks deđerinin dūřmesi ya da artması hakkındaki sadece yn gsteren tahminler, diđerisi ise hisse senedi ya da endeks deđerinin sayısal bir deđer olarak direk olarak tahminidir. Bu alıřma boyunca hem sinir ađı performansını gerekten gstermesi hem de gnlk yařama daha uyarlanabilir sonular retmesi aısından ikinci yaklařım benimsenmiřtir. Bu nedenle sinir ađı eđitimi sonrasında oluřan deđerler nmeriktir ve yzde cinsinden hata oranı bu sayede belirtilebilmektedir.

Sinir ađı eđitimi sırasında kullanılan veriler 01 Ocak 2005 tarihinden sonra oluřan borsa kapanıř deđerleridir. Bunlar hisse senedi (ISCTR,GARAN,SAHOL vs) bilgileri olabildiđi gibi, endeks kapanıř deđerleri de veri olarak alınarak sinir ađının eđitiminde kullanılmıřtır. Bu nedenle yapay sinir ađı programı sadece hisse senetlerinin deđil aynı zamanda endeks deđerlerinin gelecek tahmini iinde kullanılabilir. Bu nedenle yapay sinir ađı programı sadece hisse senetlerinin deđil aynı zamanda endeks deđerlerinin gelecek tahmini iinde kullanılabilir.

01 Ocak 2005 tarihinden itibaren elde edilen veri miktarına bakıldıđında, 10 Ekim 2006 tarihi itibariyle borsanın aık olduđu 438 gn iin test ve eđitim verisi oluřturulmuřtur. Bu veriler genel kabul gren 80–20 kuralıyla %80’lik blm

eđitim,%20'lik b6l6m6 ise test verisi olarak ayrıştırılmıştır. Bunun sonucunda 350 eđitim verisi, 88 test verisi elde edilmiştir.

Bilindiđi üzere ađ eđitiminde her bir 6rnek bir6ok defa ađa verilmektedir. Ancak her bir 6rneđin binlerce kez ađa verilmesi ađın eđitim s6resini 6ok arttırmaktadır. Ađ eđitimi sırasında ađın 6đrenebilme miktarı kadar ađın eđitim s6reside g6z 6n6nde bulundurulmuştur ve ađ eđitim s6resi 45 sn ile sınırlandırılmıştır. Bu s6re i6erisinde sinir ađı 350 test verisiyle 10.000 kez eđitilebilmektedir.

Veriler normalize edilirken aktivasyon fonksiyonunun kullanılabilmesi i6in [0,1] aralıđı kullanılmıştır. En y6ksek test girdisi 0.9 olacak Őekilde veriler normalize edilmiştir. Bunun nedeni 1 deđerine yakın deđerlerin normalden 6ok fazla sapma g6stermesidir.

6.2 DeđiŐken Se6imleri

Yapay sinir ađı programına temel olan sinir ađının deđiŐkenleri sinir ađının tahmin yeteneđini dođrudan etkilemektedir. Bu deđiŐkenlerin en 6nemlileri sinir ađı yapısı (katman sayısı, ara katmandaki sinir h6creti sayısı),aktivasyon fonksiyonu, sinir h6creti ađırlıklarının atanması ve 6đrenim katsayılarıdır.

6.2.1 Sinir Ađı Yapısı

Sinir ađında bulunan ara katmandaki sinir h6creti sayısını deđiŐtirmenin nasıl bir sonu6 verdiđini g6rebilmek amacıyla ISCTR hisse senedi kullanılarak aŐađdaki sinir ađı yapıları kullanılmıştır.

No	Sinir Ağı Yapısı	Yüzde Olarak Hata	Eğitim Süresi
	Girdi Katmanı – Ara Katman – Çıktı Katmanı		
1	4 – 1 – 1	3,110332	23 saniye
2	4 – 2 – 1	3,692801	28 saniye
3	4 – 3 – 1	3,286979	34 saniye
4	4 – 4 – 1	3,019778	38 saniye
5	4 – 5 – 1	2,97751	44 saniye
6	4 – 6 – 1	2,98888	48 saniye
7	4 – 7 – 1	3,017169	54 saniye
8	4 – 8 – 1	3,054991	60 saniye
9	4 – 10 – 1	3,130544	1 dakika 8 saniye
10	4 – 25 – 1	3,79247	2 dakika 22 saniye
11	4 – 100 – 1	4,92556	8 dakika 32 saniye

Tablodan görüldüğü üzere en iyi sonucu 4 – 5 – 1 sinir ağı yapısı sağlamıştır. Hücre sayısını arttırmak sinir ağının tahminine bir katkıda bulunmamış, üstelik karmaşıklaşan ara katman ve bağlantılardan dolayı sinir ağının tahmin edebilme yeteneği azalmıştır. Bunun yanında ara katmandaki hücre sayısının çok az olması sinir ağının genelleştirme yapmasına engel olmaktadır, bu nedenle çok az hücreye sahip ara katmanların öğrenim için yeterli kapasiteye sahip olmadığı düşünülebilir.

Diğer bir dikkat çekici durumsa sinir ağının eğitim süresine aittir. Ara katmandaki hücre sayısı arttıkça sinir ağının eğitilme zamanında lineer olarak artmaktadır.

Yukarıdaki testimizde katman sayısını sabit tutarak ara katmandaki hücre sayısını değiştirdik. Aşağıdaki tabloda ise ara katman sayısı değişmektedir.

No	Sinir Ağı Yapısı	Yüzde Olarak Hata	Eğitim Süresi
1	Girdi Katmanı – Ara Katman – Çıktı Katmanı 4 – 5 – 1	%2,97751	44 saniye
2	4 – 5 – 5 – 1	%3,212026	1 dakika 09 saniye
3	4 – 3 – 5 – 1	%3,213319	53 saniye
4	4 – 5 – 3 – 1	%3,29551	57 saniye
5	4 – 3 – 2 – 1	%3,354999	41 saniye
6	4 – 2 – 3 – 1	%3,136895	38 saniye
7	4 – 5 – 5 – 5 – 1	%3,453903	1 dakika 34 saniye
8	4 – 5 – 5 – 5 – 5 – 1	%3,405211	1 dakika 58 saniye

Tablodan anlaşıldığı üzere sinir ağına birden daha fazla katman eklemek sinir ağının performansına herhangi bir katkıda bulunmamaktadır. Bunun nedenini problemin çözümü için gerekli olan katman miktarının üç ile sınırlı olmasında bulmak mümkündür. Sinir ağını gereğinden daha karmaşık hale getirmek sinir ağının tahmin keskinliğine bir katkıda bulunmamaktadır.

6.2.2 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonları, toplam fonksiyonu sonucunda çıkan değeri, sinir ağı hücresinin çıktısı haline getiren fonksiyonlardır. Daha önce belirlediğimiz sinir ağı yapısına (4-5-1) aşağıdaki aktivasyon fonksiyonlarını uyguladık.

Aktivasyon Fonksiyonu	Yüzdesele Hata
Sigmoid	%3,156549
Linear	%80,55606
Heaviside	%100
Gaussian	%81,72775

Tablodan anlaşıldığı gibi Sigmoid Aktivasyon fonksiyonu problemimiz için en uygun değerdir. En basit aktivasyon fonksiyonu olan Heaviside türevlenememe özelliği yüzünden ağı hemen hiç eğitememiştir. Yine basit bir aktivasyon fonksiyonu olan Linear aktivasyon fonksiyonu uygun bir şekilde türevlenememe nedeniyle geriye yayılım algoritması ile tam olarak uyumlu bir sonuç vermemektedir. Gaussian aktivasyon fonksiyonu birçok problemde uygun çözüm olmasına karşın bu problemde uygun aktivasyon fonksiyonu olarak seçilmemiştir. Sigmoid aktivasyon fonksiyonunun diğer aktivasyon fonksiyonlarına belirgin bir üstünlüğü bulunmaktadır.

6.2.3 Sinir Hücrelerinin Ağırlıklarının Atanması

Sinir ağı oluşturulurken sinir hücrelerinin girdilerinin her biri rastgele olarak atanmaktadır. Daha önce kullanmış olduğumuz (4-5-1) sinir ağı yapısının hata oranları bu rastgele atama işleminden etkilenmektedir. Aşağıda (4-5-1) ağı yapısına ait bazı hata oranları görülmektedir.

Aşağıdaki tabloda ISCTR hisse senedi için sinir ağının yüzdesel olarak hatasını gösteren binin üzerinde test değeri bulunmaktadır. Bu test değerlerinin aynı öğrenim ve test verileriyle, aynı sinir ağı yapısında, aynı öğrenim metodu ile farklı sonuçlar vermesinin nedeni sinir ağının ilk oluşturma anında farklı ağırlıklar verilmesinden kaynaklanmaktadır. Bu farklılık en iyi koşulda sinir ağının %2,766854 hata ile tahmin yapmasını sağlarken, en kötü koşulda %4,159798 hata ile tahmin yapmasını sağlamıştır. Ancak bunlar en alt ve en üst sınırdır, aşağıdaki tablodaki değerlerin aritmetik ortalaması %3,084816, standart sapması ise 0,173023 olarak hesaplanmıştır. Bu sonuç bize yapay sinir ağının %90 oranında %2,911793 ile %3,257839 arasında hata ile tahmin yapabileceği bilgisine götürmektedir.

En iyi tahmin yapan sinir ağı daha sonraki çalışmalarda kullanılmak üzere kaydedilmiştir.

3,023917	2,927009	2,891405	3,096278	3,088274	2,883657	2,974378	3,019671
3,022034	3,121605	3,277803	2,891572	2,979354	3,045019	3,1879	3,008701
3,08786	2,971467	2,816304	3,277505	2,944762	3,294983	3,05842	3,053496
2,932497	3,05974	3,089284	3,030166	3,288555	3,018685	2,96274	3,037604
3,305841	3,04585	3,071052	2,941756	3,138531	3,088422	3,04564	3,076432

2,860093	2,98535	3,037817	3,00064	3,127339	3,153937	3,07814	3,058075
3,096915	3,022344	3,019264	3,026759	3,084959	3,126182	3,011326	3,074675
2,968331	3,057368	2,873219	3,383137	3,098064	3,163723	3,001336	2,988679
3,097021	3,240594	3,074271	3,056564	3,204017	2,916552	2,932677	2,965079
3,072971	3,151346	3,25817	3,16482	3,080647	2,955008	2,958258	3,167333
3,019877	2,953939	2,997365	3,056663	3,051409	3,021771	3,008999	2,921583
2,92032	3,016524	3,060739	2,983177	3,703886	3,07344	3,114211	3,018946
3,144681	3,629642	3,281655	3,149198	3,076221	2,958935	3,164094	3,048372
3,091789	2,98218	3,016535	2,842944	3,069855	2,873869	3,015547	3,02047
3,407881	3,401834	2,92381	2,953316	2,947807	2,802445	3,102863	3,110142
3,453941	3,160441	3,14061	2,954981	2,987264	3,421305	3,189604	2,884755
2,972835	3,126157	2,936117	3,008761	3,137121	3,037745	3,030833	3,909496
2,97711	2,918907	3,074272	3,338442	3,304906	3,084091	3,073856	2,88789
2,956564	3,014826	3,132687	2,971018	3,058877	3,172198	2,917748	2,922577
2,992375	3,01854	2,996818	3,190521	2,983577	3,811496	3,080894	3,111227
3,067837	2,952184	3,249134	3,022793	3,014442	3,082523	3,321995	3,401184
2,957142	3,070631	3,031655	3,13325	2,90063	3,536153	3,079454	3,27828
3,146951	3,039885	3,101791	3,135942	2,877762	3,076289	4,159798	3,090598
2,885612	3,096198	3,009592	3,114538	2,920213	3,130306	2,981667	3,147879
3,64723	3,060354	2,998842	2,974857	3,102214	3,008499	2,946055	3,081628
3,047657	3,067403	3,081668	3,068238	3,029733	2,936942	3,017237	2,922111
2,982819	3,119378	3,158561	3,437886	2,92124	2,926963	2,889282	2,972658
3,10379	3,447445	3,035232	3,111216	2,976322	3,082572	2,918902	3,094759
2,957755	2,949948	3,23872	3,001221	3,348447	3,090861	2,945233	2,937823
2,917276	3,310737	3,083295	2,979439	3,185325	2,802045	3,211642	3,211054
3,006836	2,906933	3,177809	3,047003	3,047322	3,490172	3,004142	3,04491
3,217197	2,992577	3,109228	3,257527	2,90533	3,675671	3,143588	3,024255
3,357184	3,017554	2,982986	3,059976	3,110486	2,946331	3,150805	3,068025

2,966278	3,231025	3,09453	3,260882	3,061656	2,982831	2,91249	3,048753
3,873633	3,213215	3,13559	2,982839	3,103364	2,982774	3,056125	3,361737
3,110871	3,12357	2,984903	3,13746	2,88405	2,889296	2,95065	2,977096
3,0018	3,184553	3,073109	3,245712	2,918806	2,945505	3,031637	3,101384
3,06298	3,065106	2,89658	3,051003	3,238396	2,94474	3,536368	3,097147
2,919546	2,868987	2,950382	3,177948	3,080848	3,03162	2,981912	3,036115
2,975311	2,968884	3,026192	3,0297	3,247543	3,019614	2,795849	3,077038
2,927311	2,928517	3,308865	3,785	3,04569	3,060111	3,207619	2,890482
3,142844	3,139246	2,854284	3,07819	3,026387	2,960874	2,933281	2,915114
2,985592	3,408588	2,960585	3,204418	2,985956	3,269765	2,989111	3,157274
3,128116	3,134307	3,571356	3,040837	3,363776	3,121391	2,962069	3,153534
3,266584	3,73849	3,34399	3,084814	3,077606	2,938234	2,97281	3,011056
3,032644	3,071274	2,934358	2,939519	2,936143	3,370681	3,000932	2,941864
3,277258	2,959039	3,0154	3,251852	3,114077	3,017953	2,992029	3,105787
2,992417	2,86289	3,151278	3,098161	2,970476	3,026095	3,075269	2,984379
3,056152	3,001589	2,979177	3,068629	2,960716	2,968594	3,061985	2,838364
3,054229	3,130714	3,063785	3,096924	2,938964	3,300313	2,996762	3,08556
3,0289	2,932559	2,98368	2,836899	2,847874	3,088667	3,022211	2,868325
2,998045	3,119342	3,035963	2,962841	3,004617	2,895833	3,646982	3,007824
2,994617	2,989996	2,944571	2,971564	3,204761	2,982796	3,151038	2,900354
3,128821	2,837016	3,073819	3,062917	3,017822	3,067766	2,979959	3,041818
3,007414	3,253275	3,044495	2,994605	2,983146	2,975387	3,282406	3,027455
3,024838	2,962137	3,244911	3,057508	2,900315	3,078977	3,03265	2,882007
3,773281	3,075711	2,880014	3,098421	2,974452	3,103407	3,032332	3,106062
3,189533	2,990358	3,13042	2,97487	3,750447	3,164509	2,905495	3,230304
3,060063	3,229352	3,22604	2,998054	3,263333	3,01594	3,105155	3,130982
2,993456	2,918288	2,984782	3,328019	2,911268	2,970166	3,262696	3,051852
3,066954	2,971765	3,176797	3,092502	3,023394	3,026534	2,894022	3,262942

3,014087	3,066391	3,023019	2,927021	3,267276	3,300902	3,416228	2,863582
2,974548	3,225345	2,909265	3,083736	3,092973	3,058373	2,952751	3,299014
3,432339	3,139595	3,165977	2,933296	3,104233	3,056239	3,580062	2,982524
2,766854	3,131854	2,933482	2,996552	2,993159	3,061424	2,954987	3,037118
3,007677	2,915058	2,960333	2,903625	3,003291	3,101448	3,055607	2,976186
3,401273	2,905192	3,162493	3,060636	2,873436	3,108173	3,056886	3,015489
3,089301	3,025053	3,399828	2,834453	2,904859	3,30511	2,894996	2,970593
2,994622	2,97429	3,038927	3,044775	3,1825	3,373167	3,036215	3,179385
2,946957	3,009093	2,941612	3,134728	2,985012	3,013879	2,999607	3,179163
2,976187	3,050007	3,348385	2,980776	3,24059	2,903712	3,683767	2,955884
3,432576	3,091244	2,939838	3,130614	3,155753	2,980767	3,356214	2,947743
3,223274	3,104965	3,154892	3,123967	3,088745	3,096714	3,007322	3,149542
3,015635	3,101488	2,932108	2,812463	3,712717	3,152232	2,954046	3,012637
2,97839	3,036833	2,939811	3,164119	3,114795	3,768131	3,11556	3,417646
3,001304	3,021086	3,06519	3,223926	2,996351	3,148482	3,09984	3,043481
3,053536	3,200354	3,320066	3,03628	2,884782	3,150985	3,136759	3,041315
2,999979	3,138175	3,05858	3,148766	2,94926	2,968848	3,064801	2,994841
3,581694	3,461677	3,109789	2,952754	2,949036	3,144377	2,975749	2,942719
3,339609	3,096929	3,051264	2,880329	3,019086	3,328465	3,117405	3,178587
3,094516	3,127266	3,34227	3,027194	2,974209	3,058836	3,154495	2,99088
3,25174	3,107352	3,369409	3,142161	2,894186	3,051272	3,601357	3,113698
3,037297	3,37364	3,00402	3,093919	3,205555	2,935551	3,115378	3,001306
2,960291	3,037037	3,01909	3,103548	3,076476	3,044667	3,125898	3,073275
3,232068	3,083844	3,419104	3,158734	3,055499	3,114513	3,021925	2,998825
3,615978	3,128816	2,920873	2,988756	2,993088	2,913473	3,162045	3,206821
3,022156	2,905312	3,505094	3,017402	3,5556	3,122156	2,962612	2,98774
3,202647	2,997356	2,895011	2,976372	3,087356	3,051594	3,112049	3,077557
3,070076	3,057783	3,058307	2,941519	3,036447	3,040142	3,057534	2,966197

3,211144	3,159583	3,019741	3,019856	3,060083	2,938719	3,437078	3,276141
3,509779	3,420766	2,950362	3,094086	2,925185	3,208621	3,280368	3,070042
3,02778	2,950043	3,120548	3,278427	3,160706	2,946275	2,84494	3,033053
2,99178	2,941503	2,904849	3,119831	3,11821	3,087039	3,051305	3,221689
3,075291	2,832929	3,147937	3,578186	3,089282	2,961459	3,047141	3,138824
3,003959	3,164822	3,424383	2,956138	2,899365	3,170455	3,015799	3,157065
3,113881	2,854227	3,074776	3,025452	3,355354	3,014374	3,207137	3,025007
3,07745	2,928674	3,079182	3,108299	3,103766	3,115371	2,821528	3,118896
2,997658	2,980172	2,964505	2,905237	3,085542	3,176563	3,041685	3,048159
3,124204	3,269929	3,050249	3,043423	3,066433	3,097004	2,8855	3,052796
2,903291	3,467535	2,940917	3,099383	3,229427	2,898396	3,16525	3,779459
3,094094	2,92141	3,054379	3,058173	3,150279	3,037651	3,07142	3,180183
2,930222	2,892896	3,011409	3,126147	2,959694	3,320142	3,191967	2,941289
2,965034	2,868029	3,258683	3,037064	3,091341	3,02445	2,97052	3,543998
2,887498	3,198921	3,105671	2,901438	3,155193	3,088533	3,145981	2,988439
3,597683	3,117595	3,830272	3,009131	3,169659	3,170376	2,994006	2,913297
3,119608	2,990722	2,91756	3,058997	2,921052	2,993284	3,087143	2,977075
3,044413	3,08184	3,093677	2,95774	2,988073	2,911656	2,953799	3,02138
3,380836	2,910021	2,958164	3,098833	3,017751	3,01819	2,999467	2,935182
2,994832	3,063312	3,0246	3,305174	2,960003	2,905863	3,000027	3,038807
2,979292	2,975119	2,979595	3,09798	3,12905	2,967201	3,07052	3,089647
3,187791	2,949374	2,952741	3,022213	3,028095	3,279725	2,881311	3,203953
2,948084	2,937281	2,951028	3,43996	2,980062	3,024348	3,029046	2,990551
3,162817	3,384052	3,726774	3,451681	3,794369	3,022147	2,983818	2,975544
2,91914	3,118426	3,101077	3,124156	2,995811	2,980213	3,086769	3,054639
3,130112	3,133081	2,99659	3,186953	2,93325	4,008408	3,033125	3,228366
2,958119	3,581904	3,150587	3,011895	3,167788	3,020531	2,993058	2,983146
2,968226	3,003535	3,019291	3,085638	2,909012	3,135005	2,941684	3,026197

3,073019	3,15202	3,060528	3,314254	3,198368	2,944609	2,972765	3,031896
2,867833	3,235529	2,868284	2,963064	3,261771	3,063907	3,233963	3,065274
3,172384	3,155857	3,07618	3,083021	2,912979	3,137167	3,323601	3,028259
3,089627	2,92697	3,01336	3,202438	3,087862	3,024632	3,046309	3,025841
3,145048	3,104231	3,158226	3,071071	3,033945	3,014884	3,094139	3,219402
2,923245	3,010101	2,892485	3,386546	3,169443	3,166929	2,927536	2,877757
2,951395	3,326582	3,104353	2,979454	3,35641	2,820274	3,036473	2,919306
3,100439	2,979617	3,439912	3,260475	2,809958	3,118175	2,872984	2,88241
3,032581	2,990026	3,121027	3,074131	2,90983	3,13507	3,059644	3,582308
2,955883	3,106439	3,064055	3,031222				

6.2.4 Öğrenim Algoritması Değerleri

Sinir ağları eğitiminde kullanılan öğrenim algoritmalarının en bilineni geriye doğru yayılım algoritmasıdır. Bu algoritmanın alpha ve gamma olmak üzere temel iki değişkeni vardır. Bu değişkenler 0 ile 1 arasında değerler almaktadır. Bir sonraki sayfada bulunan tabloda hem 4–5–1 sinir ağı yapısı hemde 4–1–1 sinir ağı yapısı için alpha ve beta değerleri verilmiştir. Tablodan da anlaşılacağı üzere en iyi test sonuçları alpha ve beta değerleri için 0,1 değeri atandığında ortaya çıkmaktadır.

0,5 den küçük alpha değerleri için yüzdesel hata oranları:

4–5–1 Sinir Ağı için Alpha

	0	0,1	0,2	0,3	0,4	0,5
0	29,63879	2,424745	2,868432	2,951122	3,218765	2,936432
0,1	72,91676	2,356953	2,75332	2,899086	2,932038	3,053292
0,2	34,44098	2,38052	2,875065	2,854392	2,725242	3,079086
0,3	49,43277	2,52955	3,069461	2,793275	3,589133	3,16229
Gamma 0,4	86,70334	2,651704	2,701874	2,761094	3,02056	3,177106
0,5	79,75966	2,538007	2,68153	2,718645	2,87208	3,163772
0,6	67,17587	3,204424	3,094376	3,241174	3,004139	2,906981
0,7	46,64452	2,84564	3,380486	2,697425	2,947512	3,423623
0,8	80,53938	3,45202	3,301025	4,459786	3,156242	3,600021
0,9	34,05585	3,39279	4,370225	11,13906	9,427395	5,447317
Total	581,3079	27,77635	31,09579	38,51506	36,89311	33,94992

4–1–1 Sinir Ağı için Alpha

	0	0,1	0,2	0,3	0,4	0,5
0	16,88641	2,315814	2,544713	2,997154	2,849729	2,978655
0,1	30,95383	2,330517	2,553137	2,82555	2,707614	2,926291
0,2	44,54593	2,336453	2,49792	2,782192	2,754015	2,777071
0,3	42,9332	2,355584	2,571773	2,748616	2,705207	3,059216
Gamma 0,4	29,36969	2,407588	2,441743	2,617188	2,660302	2,665146
0,5	10,17111	2,367872	2,661394	2,577705	2,625749	2,723641
0,6	5,041243	2,50504	2,57099	2,701395	2,610444	3,429338
0,7	53,50805	2,43391	2,444831	2,510283	2,853338	32,69022
0,8	44,48816	2,543103	2,697049	2,856399	4,344858	2,612365
0,9	5,369196	2,311411	2,481565	5,894192	7,508139	9,631993
Total	283,2668	23,90729	25,46512	30,51067	33,6194	65,49394

0,5 den büyük alpha değerleri için yüzdesel hata oranları:

4-5-1 Sinir Ağı için

		Alpha				
		0,6	0,7	0,8	0,9	Total
Gamma	0	3,217373	3,256275	3,479757	3,60418	57,59587
	0,1	3,169514	3,239467	3,947012	3,565572	100,833
	0,2	3,235761	3,067705	3,359728	3,487239	61,50572
	0,3	3,057826	3,218696	2,992771	4,697396	78,54317
	0,4	3,228623	3,15593	4,162829	3,387223	114,9503
	0,5	3,108001	3,421253	4,917408	3,643349	108,8237
	0,6	3,284319	3,377286	3,507162	3,826108	96,62184
	0,7	3,512698	3,524883	4,093591	4,142294	77,21267
	0,8	4,193957	4,511877	3,696922	3,289708	114,2009
	0,9	7,884533	6,341465	4,328143	3,525625	89,9124
Total		37,89261	37,11484	38,48532	37,16869	

4-1-1 Sinir Ağı için

		Alpha				
		0,6	0,7	0,8	0,9	Total
Gamma	0	3,393463	3,487382	3,72745	3,868345	45,04912
	0,1	3,104442	3,028218	3,122272	3,799247	57,35112
	0,2	3,012411	3,048776	3,628235	3,184591	70,56759
	0,3	2,835891	2,912826	2,943253	4,073825	69,13939
	0,4	2,774794	3,640597	3,826816	2,978143	55,38201
	0,5	3,418156	2,832794	2,841094	2,850891	35,07041
	0,6	2,73363	2,808812	4,079778	4,973444	33,45411
	0,7	4,403349	2,840928	2,748273	3,134685	109,5679
	0,8	2,638805	4,542809	2,942945	3,251304	72,9178
	0,9	3,499806	33,90682	3,292457	3,192473	77,08805

Total	31,81475	63,04996	33,15257	35,30695
--------------	----------	----------	----------	----------

6.3 Hisse Senedi ve Endeks Grafikleri

6.3.1 Hisse Senedi Grafikleri

Grafiđi izilecek olan hisse senetleri seilirken en byk iřlem hacmine sahip hisseler seilmiřtir.

Garanti Bankası hisse senedi gnlk %2,077507 hata oranı ile tahmin edilmiřtir.

İř Bankası hisse senedi gnlk % 2,380625 hata oranı ile tahmin edilmiřtir.

Ko Holding hisse senedi gnlk %3,091623 hata oranı ile tahmin edilmiřtir.

Akbank hisse senedi gnlk % 2,987652 hata oranı ile tahmin edilmiřtir.

Yapı Kredi Bankası hisse senedi gnlk % 3,174435 hata oranı ile tahmin edilmiřtir.

Tprař hisse senedi gnlk %1,798049 hata oranı ile tahmin edilmiřtir.

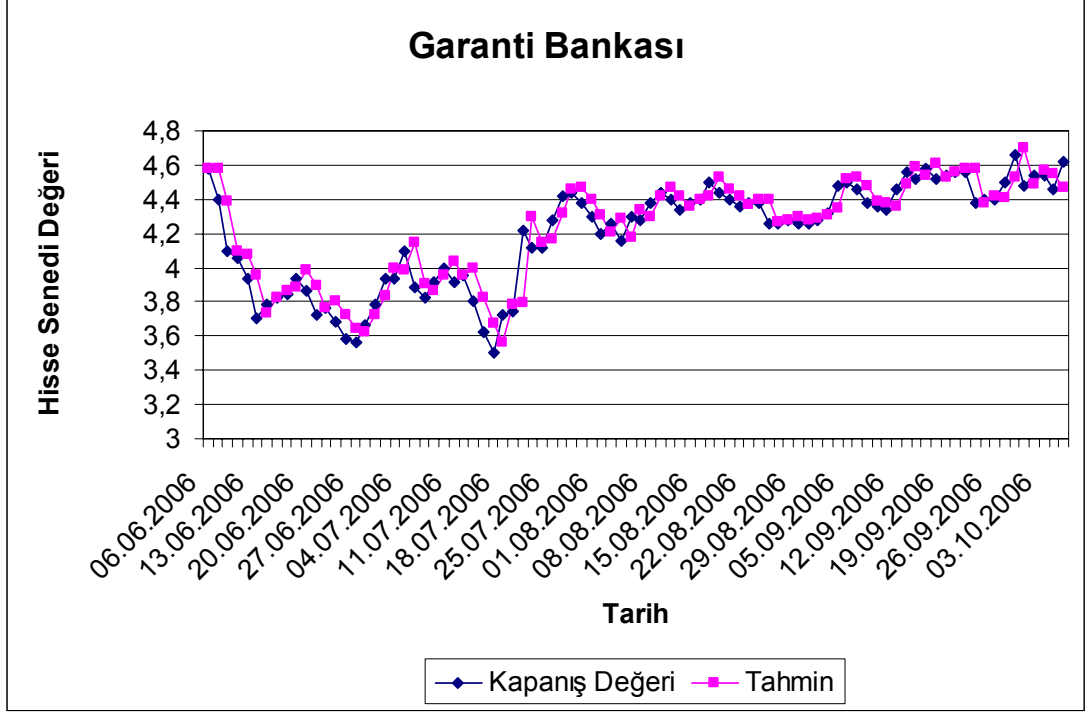
Migros hisse senedi gnlk % 2,331222 hata oranı ile tahmin edilmiřtir.

Beko hisse senedi gnlk % 4,843626 hata oranı ile tahmin edilmiřtir.

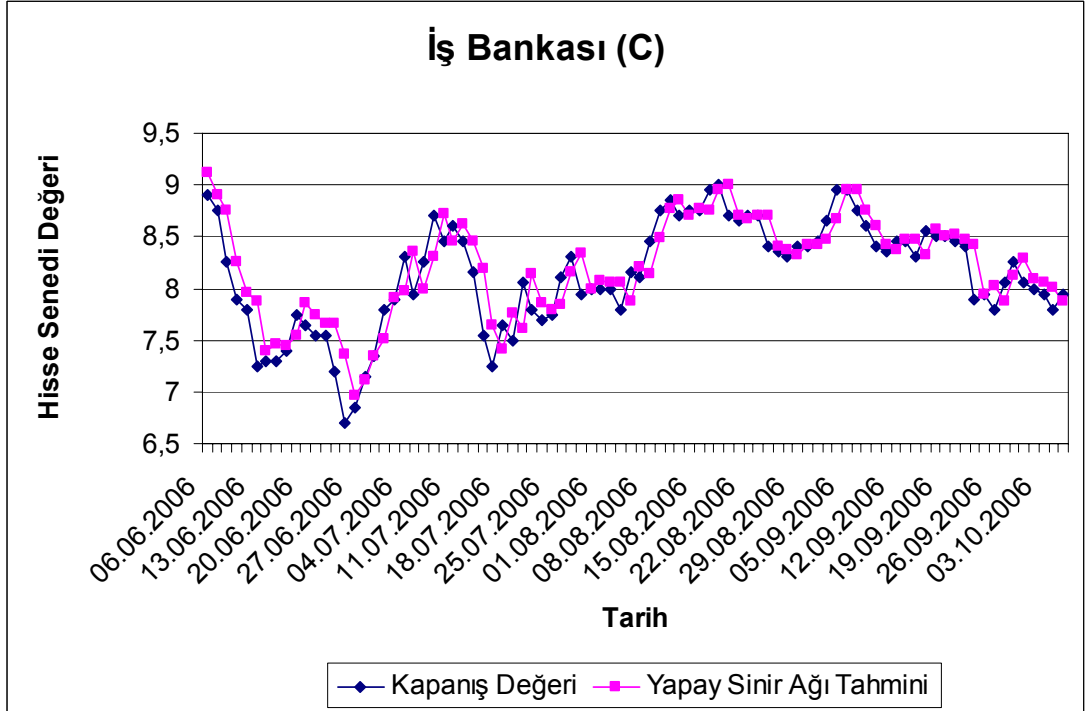
Sabancı Holding hisse senedi gnlk % 2,771967 hata oranı ile tahmin edilmiřtir.

Trkcell hisse senedi gnlk % 2,567028 hata oranı ile tahmin edilmiřtir.

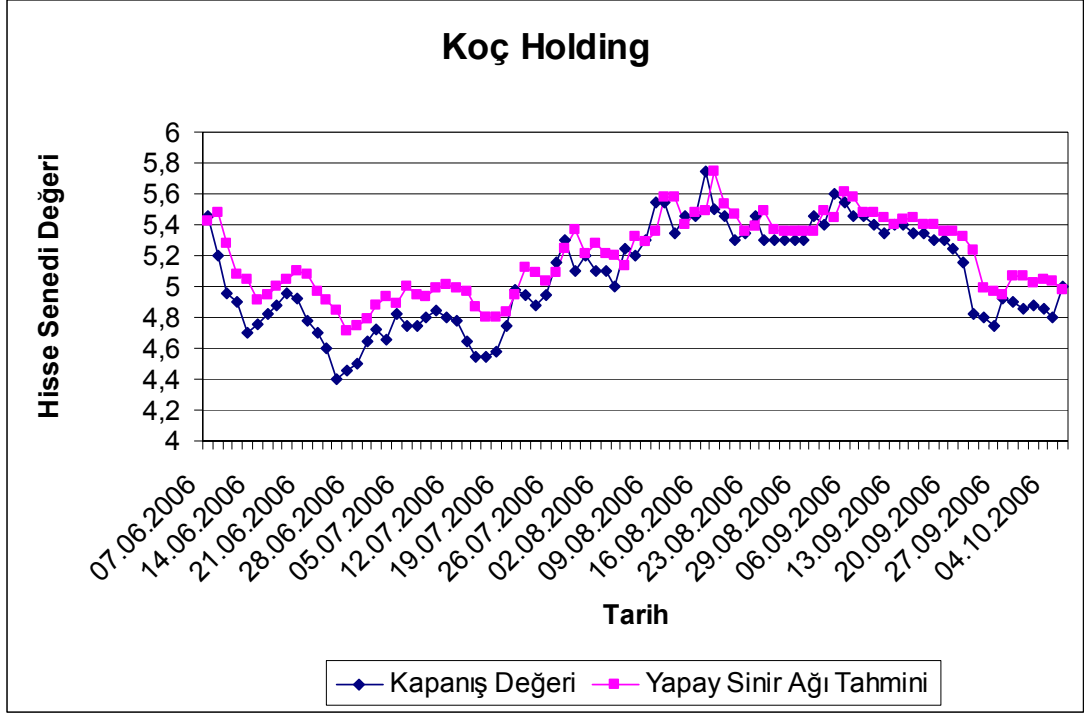
6.3.1.1 Garanti Bankası



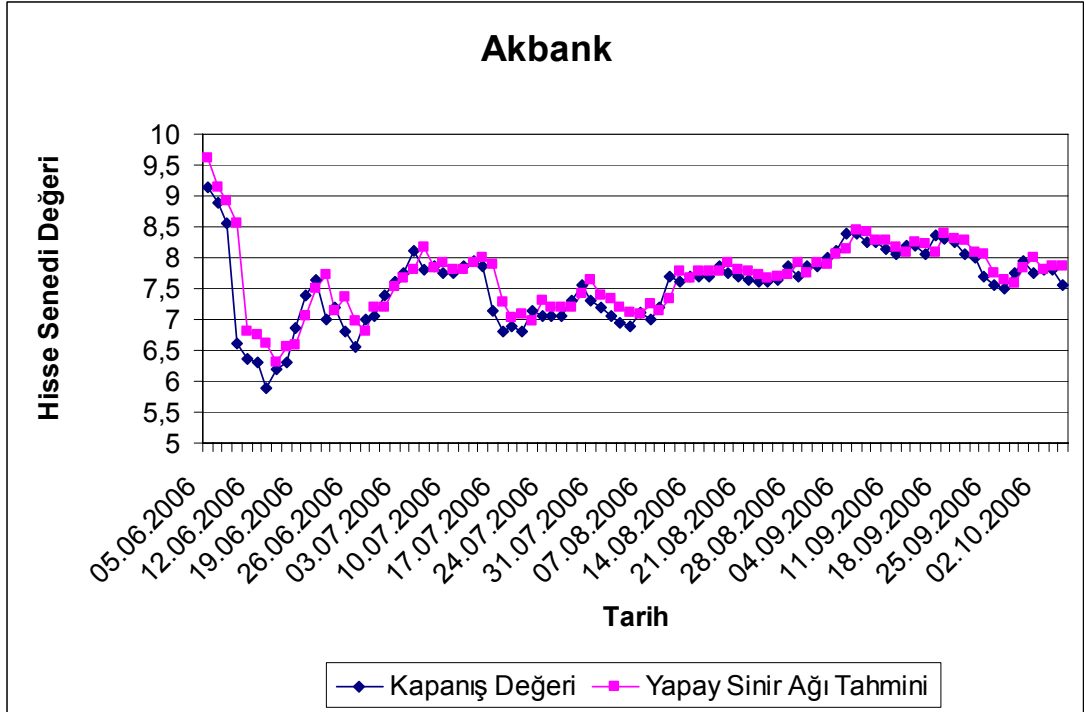
6.3.1.2 İşBank(C)



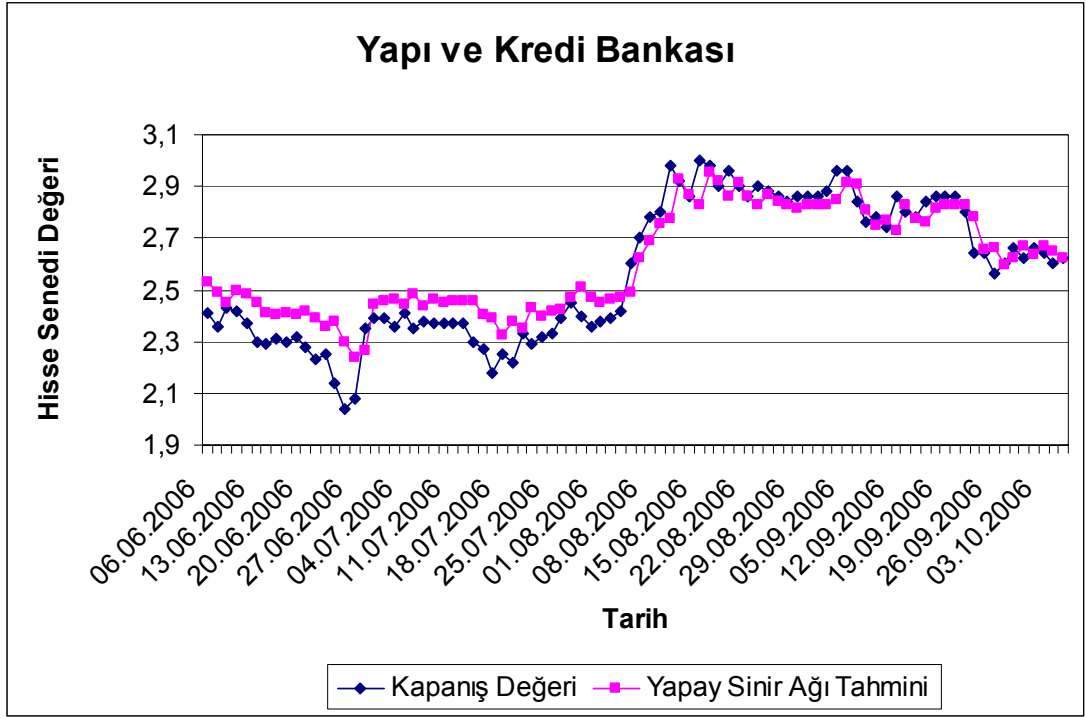
6.3.1.3 Koç Holding



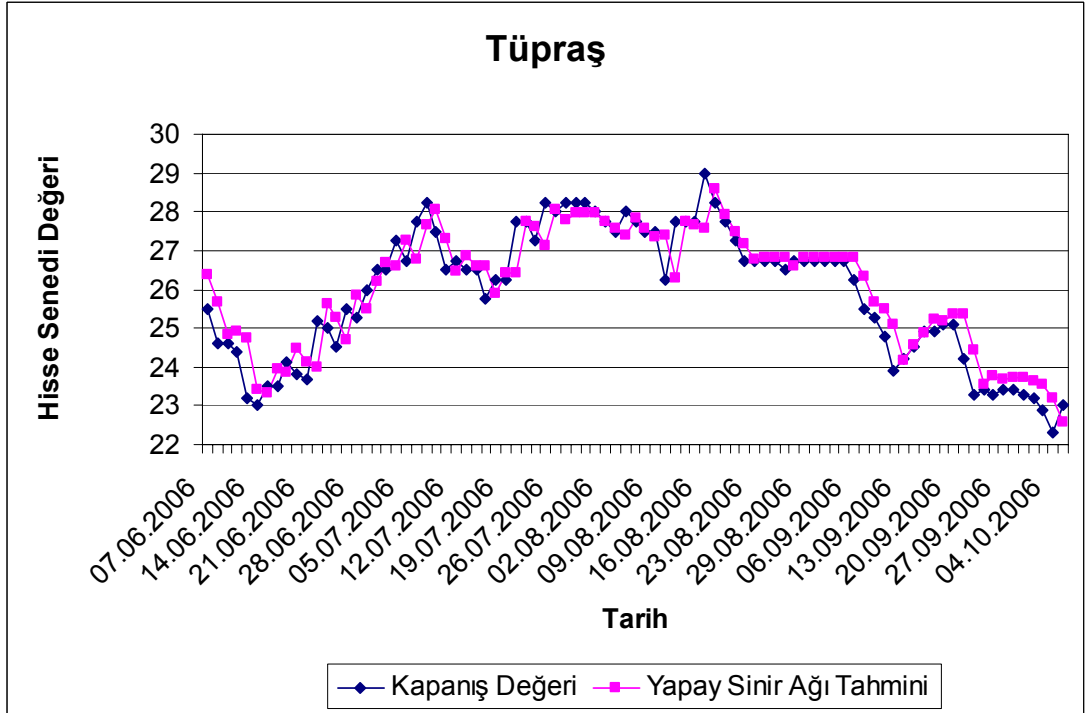
6.3.1.4 Akbank



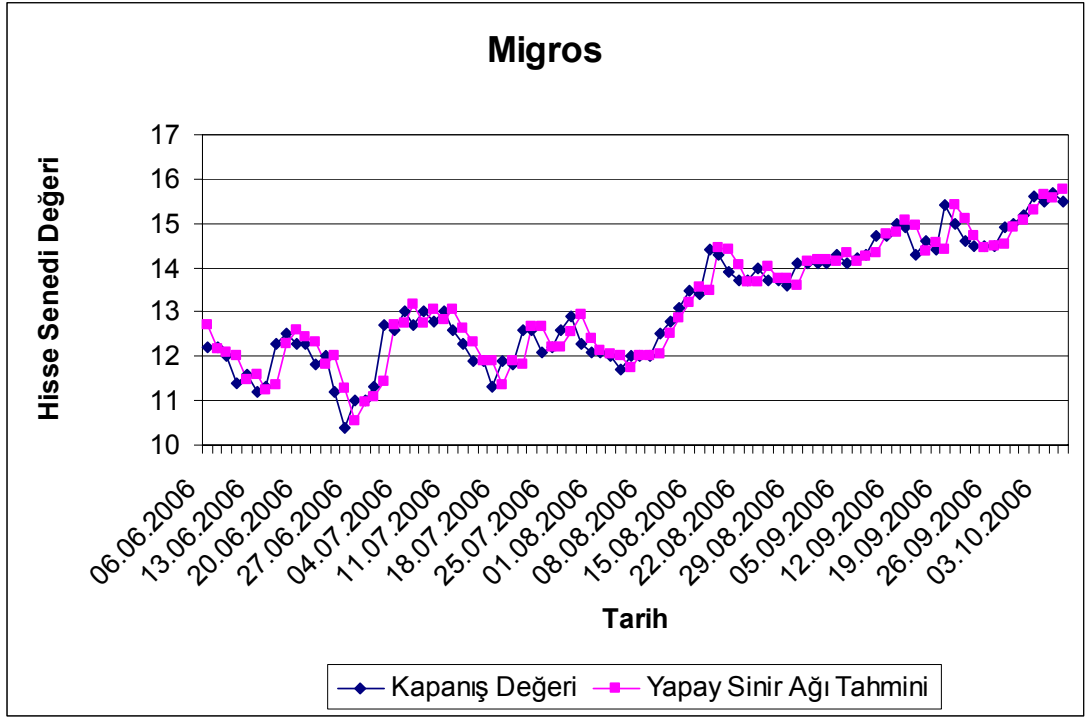
6.3.1.5 Yapı ve Kredi Bankası



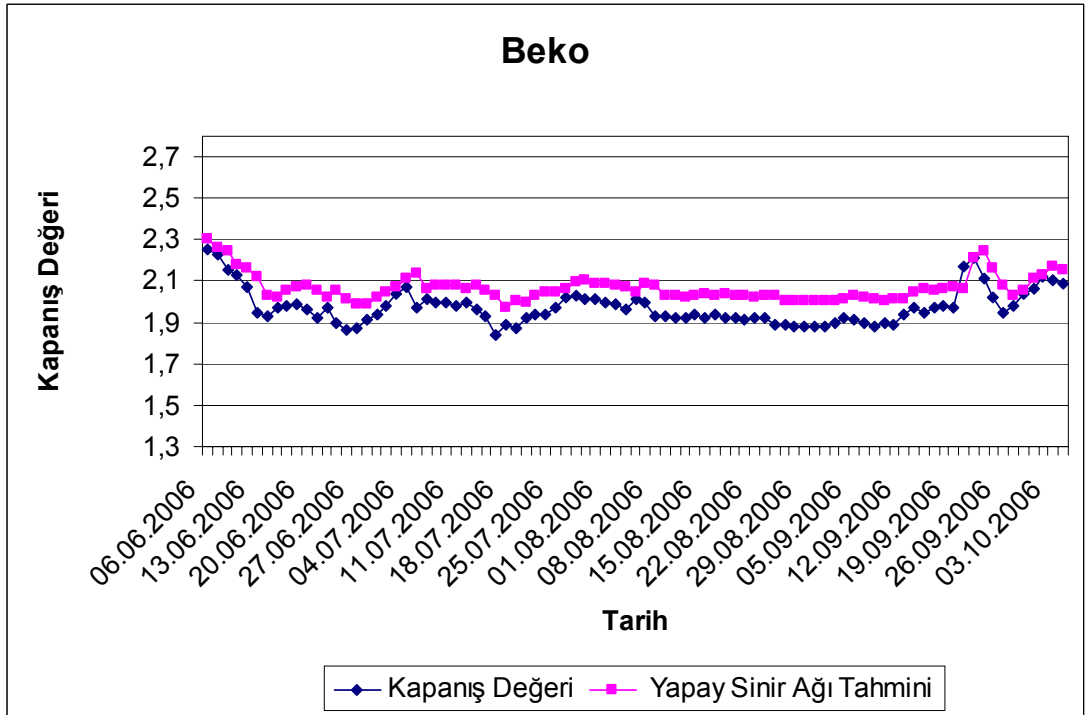
6.3.1.6 Tüpraş



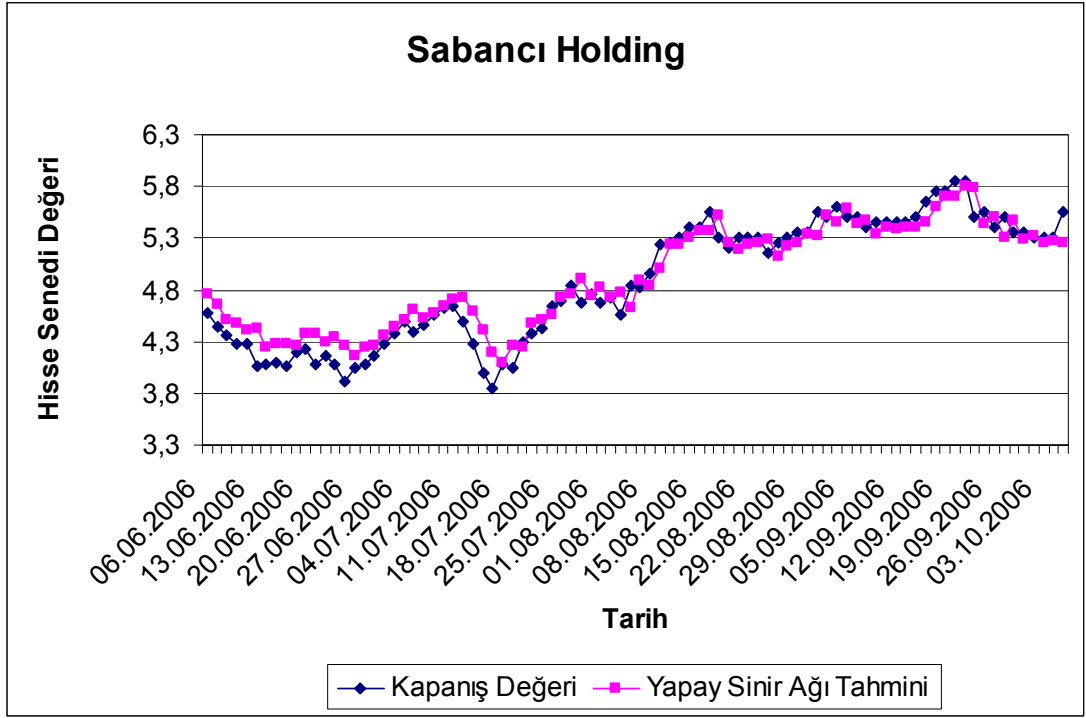
6.3.1.7 Migros



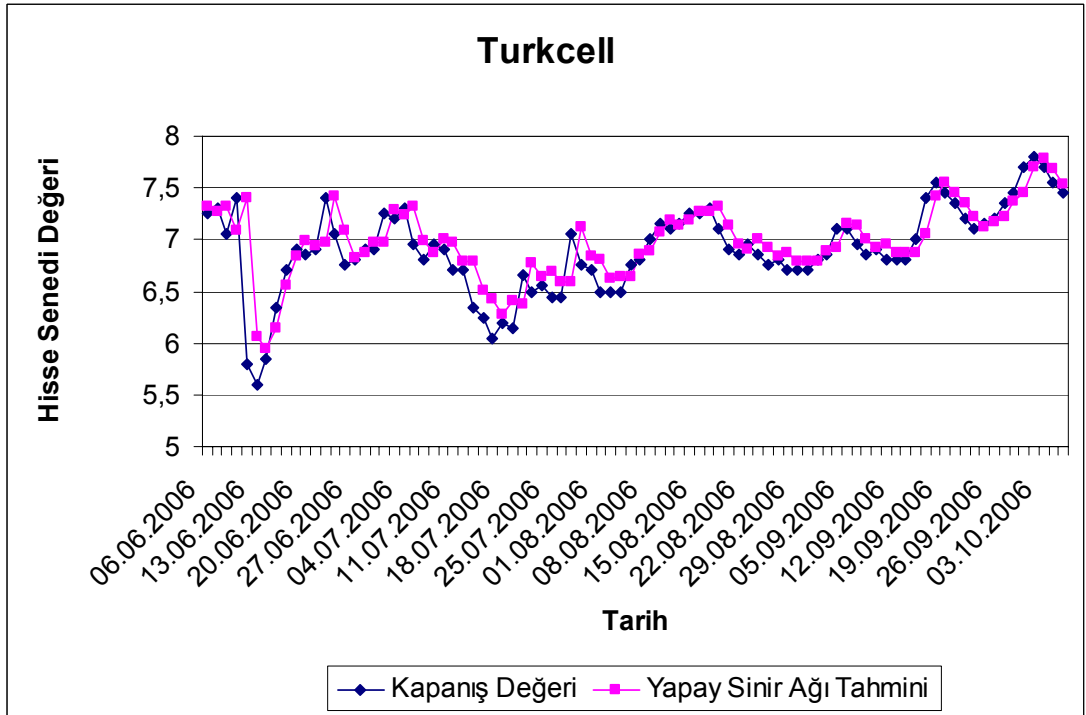
6.3.1.8 Beko



6.3.1.9 Sabancı



6.3.1.10 Türkcell



6.3.2 Endeks Grafikleri

IMKB Ulusal 100 Endeksi gnlk % 1,424335 hata oranı ile tahmin edilmiřtir.

IMKB Ulusal 30 Endeksi gnlk % 1,406907 hata oranı ile tahmin edilmiřtir.

IMKB Ulusal Hizmet Endeksi gnlk % 1,41406 hata oranı ile tahmin edilmiřtir.

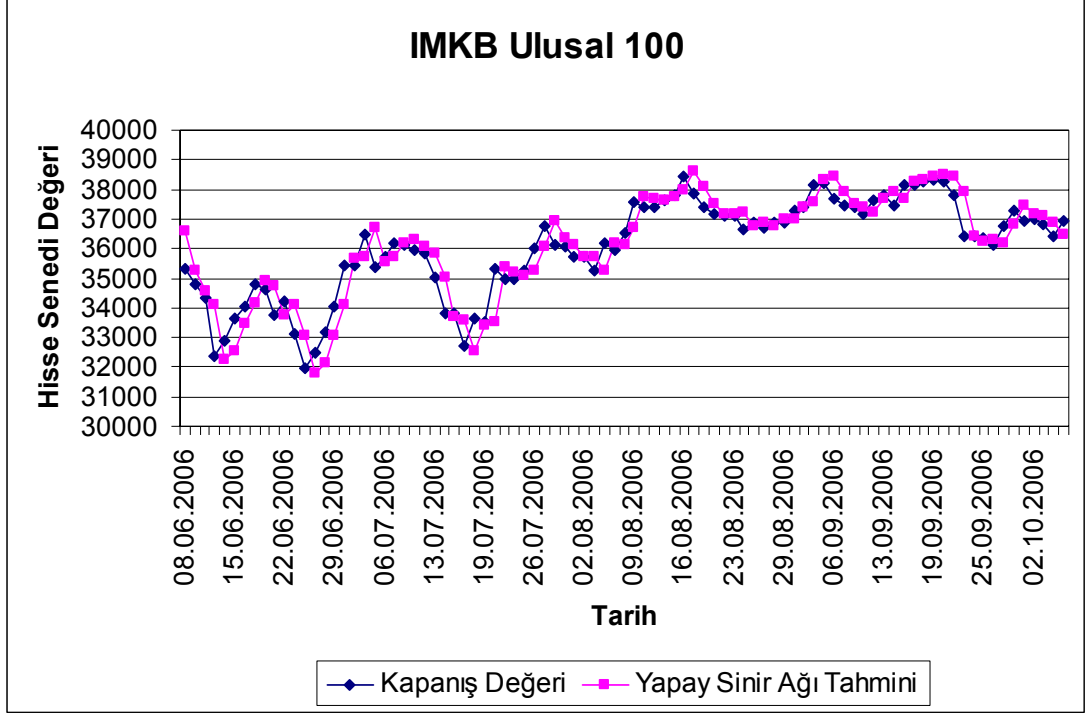
IMKB Ulusal Mali Endeksi gnlk % 1,468639 hata oranı ile tahmin edilmiřtir.

IMKB Ulusal Sınaf Endeksi gnlk % 1,331163 hata oranı ile tahmin edilmiřtir.

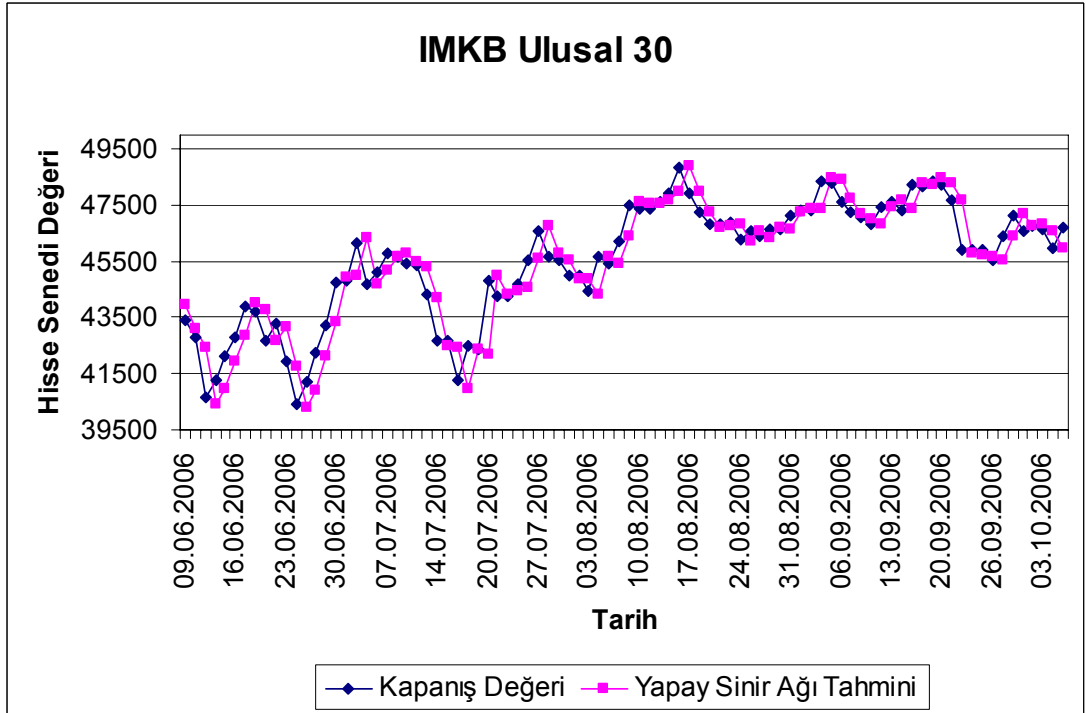
IMKB Ulusal Tm Endeksi gnlk %1,298801 hata oranı ile tahmin edilmiřtir.

Ařađıdaki tablolardan Gerek kapanıř deđerleri ile birlikte sinir ađı tahmin deđerleri birlikte verilmiřtir.

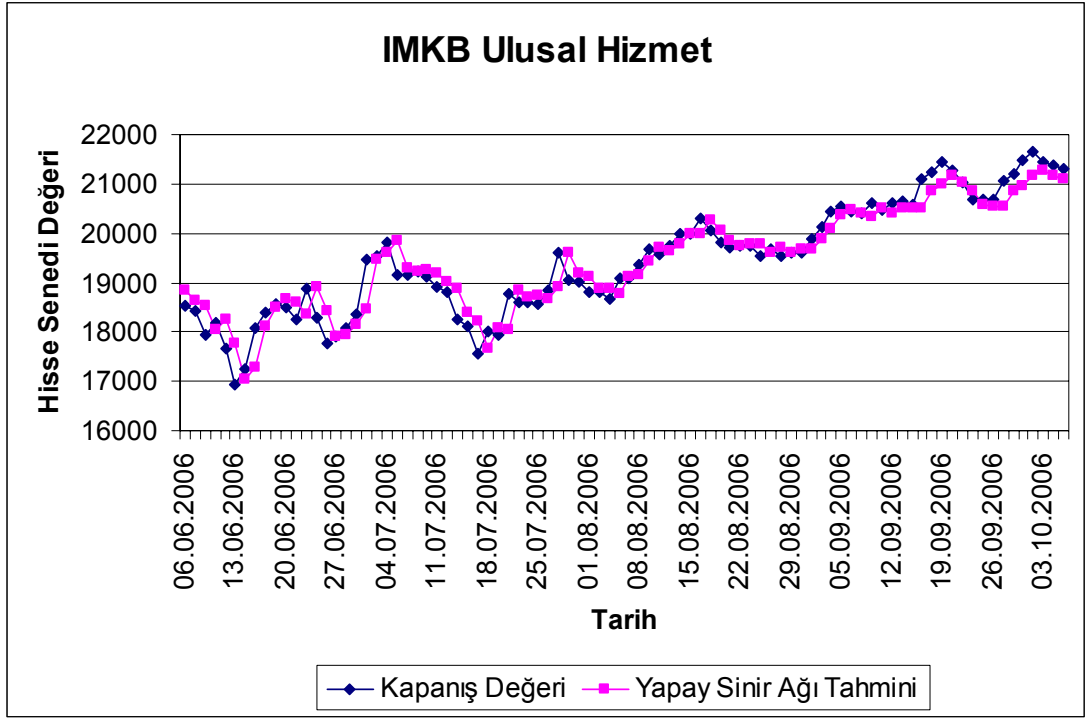
6.3.2.1 IMKB Ulusal-100



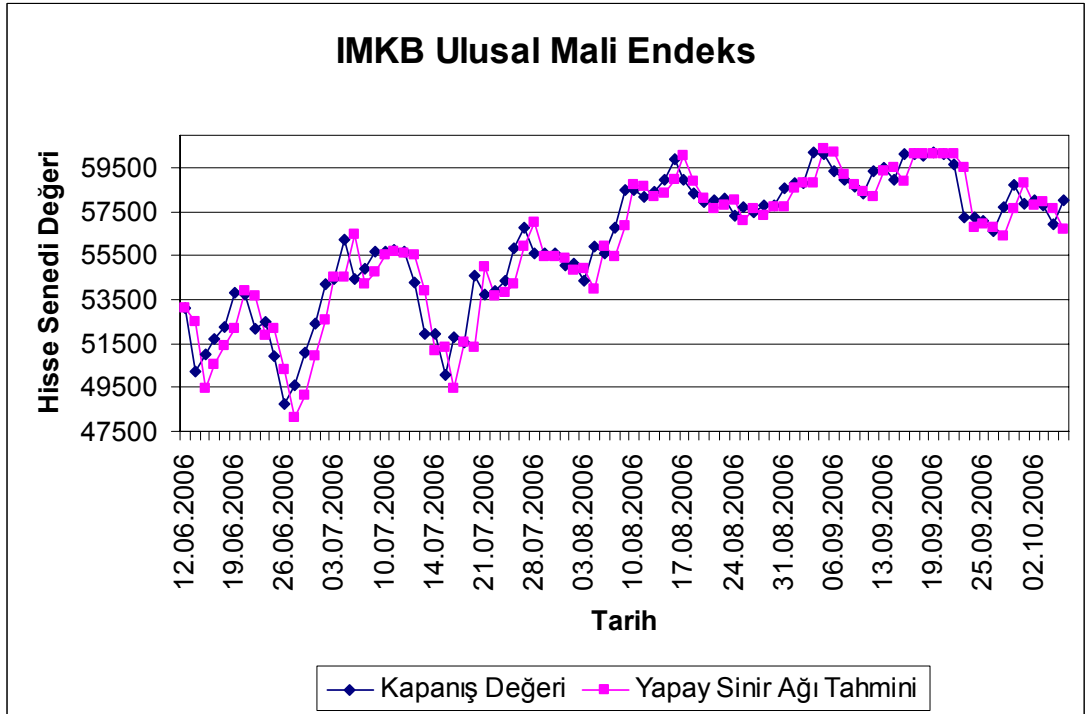
6.3.2.2 IMKB Ulusal-30



6.3.2.5 IMKB Ulusal-Hizmet



6.3.2.5 IMKB Ulusal-Mali



7.Sonraki Çalışmalar

Bu tez çalışmasında Türk Finansal Piyasalarındaki gelişmelerin hisse senedi bazındaki değerlendirilmesi yapılmaya çalışılmıştır. Bu sırada Yapay Sinir ağları temel hareket noktası olarak kabul edilerek, çözüme bu yoldan ulaşılmaya çalışılmıştır. Birçok araştırmacı yapay sinir ağlarının performansını yeterli görmeye beraber yapay sinir ağlarının uzman sistemlerle birleştirilmesinin çok daha başarılı sonuçlar doğuracağını ifade etmektedir. Ancak bu konudaki çalışmalar henüz teorik düzeydedir. Bu açıdan bakıldığında uzman sistemler ile yapay sinir ağlarını birleştirmek yeni bir çalışma konusu olarak değerlendirilebilir. Yine Genetik Algoritmaları kullanan sistemlerin yapay sinir ağları ile birleştirilmesi konusunda yeni çalışmalar yapılabilir.

Sadece hisse senedini değil, aynı zamanda diğer türev piyasaları elemanlarının değerlerini tahmin edebilen (Hazine Bonosu, Tahvil, Altın Fiyatı gibi) yapay sinir ağları da geliştirebilmek mümkündür. Ancak bu tür sinir ağlarının girdilerinin belirlenmesi, bunların elenmesi ve verilerin normalize edilerek Sinir Ağının anlayabileceği formata getirilmesi farklı bir çalışma konusu olarak değerlendirilebilir.

Bir farklı çalışma konusu da aynı sinir ağının farklı ülkeler için eğitilmesi sonucu ortaya çıkan farklılıklar neticesinde herhangi bir ülke için genel geçer bir ağ yapısı, aktivasyon fonksiyonu, eğitim algoritmasının olup olmayacağını tartışmasıdır.

8.ÖZET

Bu tez çalışmasında Türk Finansal Piyasalarındaki gelişmelerin hisse senedi bazındaki değerlendirilmesi yapılmaya çalışılmıştır. Bu sırada Yapay Sinir ağları temel hareket noktası olarak kabul edilerek, çözüme bu yoldan ulaşılmaya çalışılmıştır.

Temel olarak bakıldığında yapay sinir ağları Türk finansal piyasalarında çok başarılı sonuçlar vermektedir ve bu tez çalışmasında Türk finansal piyasaları için gerekli olan ağ yapısı ve parametreleri tespit edilmiştir. Bundan sonraki araştırmacılar bu çıkarımları referans olarak kabul edebilirler. Çalışmamızda 2005 yılı kapanış verileri için 4-5-1 sinir ağının çok başarılı performans gösterdiği görülmüştür. Üç katmanlı yapıdan daha fazla katman kullanmak sinir ağının performansını düşürmüştür. Yine aynı şekilde orta katmandaki sinir hücreleri sayısını arttırmak herhangi bir fayda sağlamamıştır. Sinir ağının öğrenim algoritması olarak en iyi sonucu verdiği görülen geriye doğru yayılım algoritması seçilmiş, en önemli öğrenim parametrelerinden olan alpha ve beta değerlerinin her ikisinde de 0.1 değerini aldığı sinir ağının performansının en iyi sonuca ulaştığı görülmüştür. Öğrenim fonksiyonları arasında ise geriye doğru türevlenebilirliği açısından diğer öğrenim fonksiyonlarından ayrılan “sigmoid” fonksiyonu üstün bir performans göstermiştir. Girdi değerlerinin normalizasyonu yapılırken de borsa kapanış değerlerinin 0 ile 0,9 arasına çekilmesi gerektiği tespit edilmiştir. Yine ağın performansının artması için eldeki verilen girdi seti ile test seti arasında 80'e 20 oranında paylaşılması gerektiği sonucuna varılmıştır. Bir diğer önemli bulguda ağın ilk yaratıldığı anda ağın ağırlıklarının

sonuca etkisi ile ilgilidir. Bu noktada ađın rastgele deđerleri yüzde bir oranında sonuç üzerinde bir etki yaratmaktadır.

En son nokta olarak yapılan bütün iyileştirme sonuçlarında yapay sinir ađı günlük kapanış deđerlerini ortalama yüzde üç, endeks deđerlerini de yüzde bir civarında hata ile tahmin edebilmektedir. Bu oldukça başarılı olan bir orandır.

Tez çalışmasının ilk bölümünde yapay sinir ađları ile ilgili olarak temel bilgiler verilmiştir. Yapay sinir hücresinin temel yapısı, girdiler, girdi ađlıkları, toplam fonksiyonu ve aktivasyon fonksiyonu ile ilgili bilgiler bu bölümde verilmiştir. Daha sonra sinir hücrelerinin birleşimiyle oluşan sinir ađ yapıları ile ilgili bilgiler verilmiş ve sinir ađlarının öğrenim metodolojileri anlatılmıştır.

İkinci bölümde oldukça kısa olan yapay sinir ađları tarihçesinden bahsedilmiş, yapay sinir ađlarının finansal alandaki kullanım alanları hakkında bilgi verilmiştir.

Üçüncü bölüm sinir ađları hakkında daha önce yapılan detaylı bir çalışma olup, Black-Scholes ve CAPM gibi teoriler detaylı bir şekilde incelenmiştir. Bunun yanında şimdiye kadar yapılmış olan çalışmalarda ne gibi sonuçlara ulaşıldığı anlatılmıştır.

Dördüncü bölüm bir sinir ađının nasıl oluşturulacağını bilgisini içermektedir. Bu bölümde tahmin yapılacak mali unsurun belirlenmesi, tahmin yapılacak sürenin belirlenmesi, tahmin yapılacak hisse ile ilgili girdilerin belirlenmesi, gerekli

bilgilerin toplanması ve bu girdiler arasındaki ilişkilerin belirlenmesi hakkında detaylı bilgi verdikten sonra, girdilerin işlenmesi ve ayrıştırılması, ağ yapısının seçilmesi (Ağ topolojisi, katman sayısı, ağ parametreleri, öğrenme algoritmaları),

Ağın öğrenmesi ve test edilmesi konularını detaylı bir şekilde açıklar.

Beşinci bölümde sinir ağı içerisinde geçen belli başlı yapıların nasıl programlama mantığına çevrildiği anlatılmaktadır.

Altıncı bölümde hisse senedi ve endeks değerlerinin sonuçları bulunmaktadır. İlk bölümde ağ parametre değişkenlerinin ağ sonucuna etkisi ifade edilmiştir. En önemli bölüm olan ikinci bölümde de sinir ağının ürettiği sonuçlar detaylı bir şekilde grafiksel olarak gösterilmiştir.

1. Abstract

In this particular work, Turkish Financial Area assessed in terms of stock prices with Artificial Neural Networks (ANN). Results are based on Artificial Neural Networks prediction results.

Fundamentally, artificial neural networks make it good guesses on the Turkish financial market and this master thesis point the best effective neural network structure and parameters. After this works, new neural network developers can take this master thesis as a reference. After the analysis the 4-5-1 structure is found to be the best artificial neural networks structure for the data which is the close values after the 2005. Incrementing numbers of layer is not an effective solution and decrement the performance .In addition to this, incrementing the number of neuron in the middle layer also does not help to make good guess. After detailed analysis, back propagation algorithm is found as the best learning algorithm and alpha and beta ,one of the most important parameters of this algorithms, taken the value of 0,1 for the best result. Moreover, “Sigmoid” learning function is far better then any other learning algorithms and it’s best fit with the back propagation algorithm in all over the other learning functions. Normalization of input values is another important point and experiments shows us input values should be normalized between the values 0 and 0,9. The founded ratio in the tests between learning set and test set is 80 -20 for the best performance. Although the effect of giving randomized input values for a neuron during the creation of neural network is very small, most probably less then

one percentage, the neural network structure designer should be keep in mind that is another important point during the design of neural net work.

At the final word, after the all improvement the financial neural network predict stock price with the three percent error ratio and also predict all the financial index values with the one and half percent error ratio and this is very successful prediction

The first part includes Artificial Neural Networks fundamental information. Fundamental structure is explained using inputs, weight factors, total function and activation function. At the final part, layers and network structure explained as well as their learning algorithms.

At the second part very short Artificial Neural Networks history is introduced and also Artificial Neural Networks working areas is explained.

The third part includes a detailed view of historical works, infact state of the art of Artificial Neural Networks, Black-Scholes and CAPM theories are researched in detail. At the final part results of this works are explained.

The answer of “How to construct a Artificial Neural Network” is included at part four. Financial aspects, the subject for prediction, the time interval for prediction, collecting and reorganizing the data are explained in this section.

At the fifth section explains how to convert fundamental Artificial Neural Networks structures to programming model.

Sixth part is the most important part of our work and includes stock and index prediction values and their graphical representation.

2. Ekler

10.1 Program Kod Dokümantasyonu

AdUzayı Listesi

The AdUzayıs specified in this document are:

AdUzayı	Assembly
NeuralNetwork	NeuralNetwork

AdUzayı : NeuralNetwork

NeuralNetwork Sınıf Listesi

Interfaces

Sınıf	Özet
ActivationFunction	

Sınıfı

Sınıf	Özet
BackPropagationLearningAlgorithm	
GaussianActivationFunction	Initializes a new instance of the class.
GeneticLearningAlgorithm	
GeneticLearningAlgorithm.GeneticNeuralNetwork	Gets the of the current instance.
HeavisideActivationFunction	Initializes a new instance of the class.
Layer	Gets the of the current instance.
LearningAlgorithm	Gets the of the current instance.

LinearActivationFunction	Initializes a new instance of the class.
NeuralNetwork	Gets the of the current instance.
Neuron	Gets the of the current instance.
SigmoidActivationFunction	Initializes a new instance of the class.

NeuralNetwork Interfaces

ActivationFunction Interface

Özet

```
public interface ActivationFunction
```

Sınıf Üyeleri

Name	Access	Özet
-------------	---------------	-------------

Output() : <i>Single</i>	public	
OutputPrime() : <i>Single</i>	public	

NeuralNetwork Sınıfı

SigmoidActivationFunction Sınıfı

Özet

```
public class SigmoidActivationFunction :
NeuralNetwork.ActivationFunction
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
SigmoidActivationFunction()	public	Initializes a new instance of the class.

Lokal Alan Üyeleri

Name	Access	Özet
beta : <i>Single</i>	protected	

Global Alan Üyeleri

Name	Access	Özet
Beta : <i>Single</i>	public	
Name : <i>String</i>	public	

Sınıf Üyeleri

Name	Access	Özet
Output() : <i>Single</i>	public	
OutputPrime() : <i>Single</i>	public	

LinearActivationFunction Sınıfı

Özet

```
public class LinearActivationFunction :  
NeuralNetwork.ActivationFunction
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
LinearActivationFunction()	public	Initializes a new instance of the class.

Lokal Alan Üyeleri

Name	Access	Özet
a : <i>Single</i>	protected	
threshold : <i>Single</i>	protected	

Global Alan Üyeleri

Name	Access	Özet
<i>A : Single</i>	public	
<i>Name : String</i>	public	

Sınıf Üyeleri

Name	Access	Özet
<i>Output() : Single</i>	public	
<i>OutputPrime() : Single</i>	public	

HeavisideActivationFunction Sınıfı

Özet

```
public class HeavisideActivationFunction :  
    NeuralNetwork.ActivationFunction
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
HeavisideActivationFunction()	public	Initializes a new instance of the class.

Global Alan Üyeleri

Name	Access	Özet
Name : <i>String</i>	public	

Sınıf Üyeleri

Name	Access	Özet
Output() : <i>Single</i>	public	
OutputPrime() : <i>Single</i>	public	

GaussianActivationFunction Sınıfı

Özet

```
public class GaussianActivationFunction :  
    NeuralNetwork.ActivationFunction
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
GaussianActivationFunction()	public	Initializes a new instance of the class.

Lokal Alan Üyeleri

Name	Access	Özet
<i>C : Single</i>	protected	
<i>K : Single</i>	protected	
<i>mu : Single</i>	protected	

sigma : <i>Single</i>	protected	
-----------------------	-----------	--

Global Alan Üyeleri

Name	Access	Özet
Mu : <i>Single</i>	public	
Name : <i>String</i>	public	
Sigma : <i>Single</i>	public	

Sınıf Üyeleri

Name	Access	Özet
computeCK() : <i>Void</i>	protected	
Output() : <i>Single</i>	public	
OutputPrime() : <i>Single</i>	public	

Layer Sınıfı

Özet

```
public class Layer
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
Layer()	public	
Layer()	public	

Lokal Alan Üyeleri

Name	Access	Özet
neurons : <i>Neuron</i> []	protected	
ni : <i>Int32</i>	protected	
nn : <i>Int32</i>	protected	

output : <i>Single</i> []	protected	
---------------------------	-----------	--

Global Alan Üyeleri

Name	Access	Özet
Item : <i>Neuron</i>	public	
Last_Output : <i>Single</i> []	public	
N_Inputs : <i>Int32</i>	public	
N_Neurons : <i>Int32</i>	public	

Sınıf Üyeleri

Name	Access	Özet
Output() : <i>Single</i> []	public	
randomizeAll() : <i>Void</i>	public	

randomizeThreshold() : <i>Void</i>	public	
randomizeWeight() : <i>Void</i>	public	
setActivationFunction() : <i>Void</i>	public	
setRandomizationInterval() : <i>Void</i>	public	

LearningAlgorithm Sınıfı

Özet

```
public abstract class LearningAlgorithm
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
LearningAlgorithm()	public	

Lokal Alan Üyeleri

Name	Access	Özet
error : <i>Single</i>	protected	
ERROR_THRESHOLD : <i>Single</i>	protected	
ins : <i>Single</i> [][]	protected	
iter : <i>Int32</i>	protected	
MAX_ITER : <i>Int32</i>	protected	
nn : <i>NeuralNetwork</i>	protected	
outs : <i>Single</i> [][]	protected	

Global Alan Üyeleri

Name	Access	Özet
------	--------	------

Error : <i>Single</i>	public	
ErrorTreshold : <i>Single</i>	public	
Iteration : <i>Int32</i>	public	
MaxIteration : <i>Int32</i>	public	
N_Network : <i>NeuralNetwork</i>	public	

Sınıf Üyeleri

Name	Access	Özet
Learn() : <i>Void</i>	public	

BackPropagationLearningAlgorithm Sınıfı

Özet

```
public class BackPropagationLearningAlgorithm :
LearningAlgorithm
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
BackPropagationLearningAlgorithm()	public	

Lokal Alan Üyeleri

Name	Access	Özet
alpha : <i>Single</i>	protected	
e : <i>Single[]</i>	protected	
error : <i>Single</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
ERROR_THRESHOLD : <i>Single</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
gamma : <i>Single</i>	protected	

<i>ins : Single[][]</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<i>iter : Int32</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<i>MAX_ITER : Int32</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<i>nn : NeuralNetwork</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<i>outs : Single[][]</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)

Global Alan Üyeleri

Name	Access	Özet
<i>Alpha : Single</i>	public	
<i>Error : Single</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>)

ErrorTreshold : <i>Single</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>)
Gamma : <i>Single</i>	public	
Iteration : <i>Int32</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>)
MaxIteration : <i>Int32</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>)
N_Network : <i>NeuralNetwork</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>)

Sınıf Üyeleri

Name	Access	Özet
ComputeA() : <i>Void</i>	protected	
Learn() : <i>Void</i>	public	
setWeight() : <i>Void</i>	protected	

GeneticLearningAlgorithm Sınıfı

Özet

```
public class GeneticLearningAlgorithm :  
LearningAlgorithm
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
GeneticLearningAlgorithm()	public	

Lokal Alan Üyeleri

Name	Access	Özet
error : <i>Single</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
ERROR_THRESHOLD : <i>Single</i>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)

<code>ins : Single[][]</code>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<code>iter : Int32</code>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<code>MAX_ITER : Int32</code>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<code>MAX_MUTATION_AMP :</code> <i>Single</i>	protected	
<code>MUTATION_RATIO : Int32</code>	protected	
<code>nn : NeuralNetwork</code>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<code>outs : Single[][]</code>	protected	(from <i>NeuralNetwork.LearningAlgorithm</i>)
<code>POPULATION_SIZE : Int32</code>	protected	

population : <i>ArrayListesi</i>	protected	
rand : <i>Random</i>	protected	
SELECTION_RATIO : <i>Int32</i>	protected	

Global Alan Üyeleri

Name	Access	Özet
Error : <i>Single</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>
ErrorTreshold : <i>Single</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>
Iteration : <i>Int32</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>
MaxIteration : <i>Int32</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>

MaxMutationAmplitude : <i>Single</i>	public	
MutationRatio : <i>Int32</i>	public	
MutationValue : <i>Single</i>	protected	
Mute : <i>Boolean</i>	protected	
Muted_NeuralNetwork : <i>GeneticLearningAlgorithm#GeneticNeuralNetwork</i>	protected	
N_Network : <i>NeuralNetwork</i>	public	(from <i>NeuralNetwork.LearningAlgorithm</i>
PopulationSize : <i>Int32</i>	public	
RandSelectionIndex : <i>Int32</i>	protected	
SelectionRatio : <i>Int32</i>	public	

Sınıf Üyeleri

Name	Access	Özet
ComputeErrors() : <i>Void</i>	protected	
CrossOver() : <i>GeneticLearningAlgorithm#GeneticNeuralNetwork</i>	protected	
Learn() : <i>Void</i>	public	
makeNewGeneration() : <i>Void</i>	protected	
RandomizePopulation() : <i>Void</i>	public	

GeneticLearningAlgorithm.GeneticNeuralNetwork Sınıfı

Özet

nestedFamily class

GeneticLearningAlgorithm.GeneticNeuralNetwork :

System.IComparable

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
GeneticLearningAlgorithm.GeneticNeuralNetwork()	public	

Lokal Alan Üyeleri

Name	Access	Özet
genes : <i>Single</i> []	protected	
nn : <i>NeuralNetwork</i>	protected	
sq_err : <i>Single</i>	protected	

Global Alan Üyeleri

Name	Access	Özet
Error : <i>Single</i>	public	
Item : <i>Single</i>	public	

N_Genes : <i>Int32</i>	public	
------------------------	--------	--

Sınıf Üyeleri

Name	Access	Özet
CompareTo() : <i>Int32</i>	public	
Init() : <i>Void</i>	public	
setWeights() : <i>Void</i>	public	

NeuralNetwork Sınıfı

Özet

```
public class NeuralNetwork
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
------	--------	------

NeuralNetwork()	public	
NeuralNetwork()	public	
NeuralNetwork()	public	

Lokal Alan Üyeleri

Name	Access	Özet
<i>la : LearningAlgorithm</i>	protected	
<i>layers : Layer[]</i>	protected	
<i>ni : Int32</i>	protected	

Global Alan Üyeleri

Name	Access	Özet
<i>Item : Layer</i>	public	

LearningAlg : <i>LearningAlgorithm</i>	public	
N_Inputs : <i>Int32</i>	public	
N_Layers : <i>Int32</i>	public	
N_Outputs : <i>Int32</i>	public	

Sınıf Üyeleri

Name	Access	Özet
load() : <i>NeuralNetwork</i>	public	
Output() : <i>Single[]</i>	public	
randomizeAll() : <i>Void</i>	public	
randomizeThreshold() : <i>Void</i>	public	
randomizeWeight() : <i>Void</i>	public	

save() : Void	public	
setActivationFunction() : Void	public	
setRandomizationInterval() : Void	public	

Neuron Sınıfı

Özet

```
public class Neuron
```

This Sınıf has the following attributes: System.SerializableAttribute

Yapılandırıcı Üyeler

Name	Access	Özet
Neuron()	public	
Neuron()	public	

Lokal Alan Üyeleri

Name	Access	Özet
<i>a : Single</i>	protected	
<i>f : ActivationFunction</i>	protected	
<i>last_threshold : Single</i>	protected	
<i>last_w : Single[]</i>	protected	
<i>o : Single</i>	protected	
<i>R_MAX : Single</i>	protected	
<i>R_MIN : Single</i>	protected	
<i>rand : Random</i>	protected	
<i>threshold : Single</i>	protected	

<i>w : Single[]</i>	protected	
<i>ws : Single</i>	protected	

Global Alan Üyeleri

Name	Access	Özet
<i>A : Single</i>	public	
<i>F : ActivationFunction</i>	public	
<i>Item : Single</i>	public	
<i>Last_Threshold : Single</i>	public	
<i>Last_W : Single[]</i>	public	
<i>N_Inputs : Int32</i>	public	
<i>Output : Single</i>	public	

OutputPrime : <i>Single</i>	public	
Randomization_Max : <i>Single</i>	public	
Randomization_Min : <i>Single</i>	public	
Threshold : <i>Single</i>	public	
WS : <i>Single</i>	public	

Sınıf Üyeleri

Name	Access	Özet
ComputeOutput() : <i>Single</i>	public	
randomizeAll() : <i>Void</i>	public	
randomizeThreshold() : <i>Void</i>	public	
randomizeWeight() : <i>Void</i>	public	

10.2 Örnek Veriler

Programımız tüm hisseler için tahmin yapabilecek şekilde tasarlanmıştır. Aşağıda sadece AKBANK için kullanılan örnek veriler verilmiştir.

Hisse				Kapanış	İşlem
Adı	Tarih	En Yüksek	En Düşük	Değeri	Hacmi
AKBNK	03.01.2005	8.5	7.55	8.4	999202
AKBNK	04.01.2005	8.45	8.05	8.05	2148428
AKBNK	05.01.2005	8.1	7.65	7.7	3349148
AKBNK	06.01.2005	7.85	7.6	7.8	2455843
AKBNK	07.01.2005	8.2	7.7	8.2	3207959
AKBNK	10.01.2005	8.45	8.1	8.25	4907930
AKBNK	11.01.2005	8.35	8	8.15	1651714
AKBNK	12.01.2005	8.45	8.05	8.3	4435033
AKBNK	13.01.2005	8.5	8.3	8.45	2685584
AKBNK	14.01.2005	8.5	8.3	8.4	5479228
AKBNK	17.01.2005	8.5	8.3	8.5	752370
AKBNK	18.01.2005	8.6	8.15	8.35	2806821
AKBNK	19.01.2005	8.8	8.2	8.8	1277650
AKBNK	24.01.2005	8.65	8.15	8.25	4361606
AKBNK	25.01.2005	8.35	8.15	8.3	2134296
AKBNK	26.01.2005	8.45	8.2	8.4	4175731
AKBNK	27.01.2005	8.55	8.4	8.45	1608941
AKBNK	28.01.2005	8.4	8.2	8.25	2901621
AKBNK	31.01.2005	8.35	8.15	8.35	2530830
AKBNK	01.02.2005	8.5	8.3	8.3	4034149

AKBNK	02.02.2005	8.45	8.3	8.35	4951374
AKBNK	03.02.2005	8.4	8.1	8.15	2376200
AKBNK	04.02.2005	8.3	7.95	8.1	4076276
AKBNK	07.02.2005	8.25	8.05	8.15	4196126
AKBNK	08.02.2005	8.3	8.15	8.15	5049430
AKBNK	09.02.2005	8.35	8.05	8.1	5350205
AKBNK	10.02.2005	8.2	7.75	7.95	3556160
AKBNK	11.02.2005	8.1	7.75	7.95	5049469
AKBNK	14.02.2005	8.1	7.95	8.1	1983314
AKBNK	15.02.2005	8.2	7.95	8	3721485
AKBNK	16.02.2005	8.05	7.7	7.85	5967026
AKBNK	17.02.2005	8	7.7	7.75	3655458
AKBNK	18.02.2005	7.85	7.45	7.55	8351504
AKBNK	21.02.2005	7.6	7.15	7.25	8111133
AKBNK	22.02.2005	7.4	7.2	7.35	9954010
AKBNK	23.02.2005	7.45	7.25	7.45	3951499
AKBNK	24.02.2005	7.7	7.45	7.7	6519832
AKBNK	25.02.2005	7.95	7.65	7.8	8252506
AKBNK	28.02.2005	8.05	7.65	8	3662954
AKBNK	01.03.2005	8.05	7.8	7.85	2887580
AKBNK	02.03.2005	7.9	7.5	7.5	3463762
AKBNK	03.03.2005	7.6	7.45	7.5	6159804
AKBNK	04.03.2005	7.6	7.45	7.6	4230732
AKBNK	07.03.2005	7.65	7.5	7.5	2399952
AKBNK	08.03.2005	7.55	7.35	7.35	2473352
AKBNK	09.03.2005	7.45	7.2	7.3	3790826
AKBNK	10.03.2005	7.3	7.05	7.05	4682176
AKBNK	11.03.2005	7.35	7.05	7.3	2937716

AKBNK	14.03.2005	7.3	7.15	7.25	5990516
AKBNK	15.03.2005	7.35	7.1	7.15	6400984
AKBNK	16.03.2005	7.15	6.8	6.9	7642059
AKBNK	17.03.2005	7.05	6.35	6.7	9699049
AKBNK	18.03.2005	7.15	6.75	6.9	8482211
AKBNK	21.03.2005	6.9	6.45	6.65	5770534
AKBNK	22.03.2005	6.75	6.45	6.75	5260068
AKBNK	23.03.2005	6.75	6.45	6.6	7200799
AKBNK	24.03.2005	6.85	6.65	6.75	5596518
AKBNK	25.03.2005	6.85	6.65	6.8	1028694
AKBNK	28.03.2005	6.8	6.35	6.4	3112516
AKBNK	29.03.2005	6.45	6.2	6.25	4360136
AKBNK	30.03.2005	6.35	6.1	6.2	4135282
AKBNK	31.03.2005	6.5	6.25	6.45	4252195
AKBNK	01.04.2005	6.75	6.5	6.7	2951281
AKBNK	04.04.2005	6.75	6.6	6.65	2113672
AKBNK	05.04.2005	6.75	6.55	6.75	1963652
AKBNK	06.04.2005	6.9	6.85	6.9	692695
AKBNK	07.04.2005	7.1	6.9	6.95	1754372
AKBNK	08.04.2005	7.05	6.9	6.95	2665042
AKBNK	11.04.2005	7	6.75	6.8	2375598
AKBNK	12.04.2005	7	6.85	6.9	3652451
AKBNK	13.04.2005	6.85	6.65	6.8	3435740
AKBNK	14.04.2005	6.85	6.5	6.6	1218269
AKBNK	15.04.2005	6.45	6.25	6.3	2056002
AKBNK	18.04.2005	6.3	6.15	6.25	3064656
AKBNK	19.04.2005	6.5	6.15	6.5	4310802
AKBNK	20.04.2005	6.65	6.45	6.6	4157666

AKBNK	21.04.2005	6.7	6.4	6.6	1834393
AKBNK	22.04.2005	6.8	6.5	6.6	2460883
AKBNK	25.04.2005	6.75	6.4	6.7	1587134
AKBNK	26.04.2005	6.75	6.55	6.7	2256829
AKBNK	27.04.2005	6.65	6.55	6.6	2102816
AKBNK	28.04.2005	6.7	6.5	6.55	3948425
AKBNK	29.04.2005	6.85	6.45	6.65	5770823
AKBNK	02.05.2005	7.2	6.65	7.05	3182444
AKBNK	03.05.2005	7.1	6.8	6.9	3731182
AKBNK	04.05.2005	7.3	6.95	7.2	4942871
AKBNK	05.05.2005	7.5	7.25	7.45	3525546
AKBNK	06.05.2005	7.5	7.25	7.25	2555319
AKBNK	09.05.2005	7.3	7.1	7.25	1782110
AKBNK	10.05.2005	7.4	7.2	7.25	2983352
AKBNK	11.05.2005	7.2	7.1	7.2	2371234
AKBNK	12.05.2005	7.4	7.15	7.25	5087744
AKBNK	13.05.2005	7.35	7.2	7.3	2677459
AKBNK	16.05.2005	7.4	7.25	7.3	2414428
AKBNK	17.05.2005	7.4	7.15	7.2	1470700
AKBNK	18.05.2005	7.25	7.15	7.2	1428329
AKBNK	20.05.2005	7.45	7.25	7.4	2148978
AKBNK	23.05.2005	7.4	6.8	6.85	4563800
AKBNK	24.05.2005	7	6.6	6.65	3232213
AKBNK	25.05.2005	6.75	6.5	6.6	3150130
AKBNK	26.05.2005	6.7	6.5	6.65	2710724
AKBNK	27.05.2005	6.85	6.65	6.8	2505238
AKBNK	30.05.2005	7.05	6.7	7.05	2363397
AKBNK	31.05.2005	7.3	6.95	7.3	4309904

AKBNK	01.06.2005	7.4	7.15	7.35	4025285
AKBNK	02.06.2005	7.7	7.35	7.45	3649999
AKBNK	03.06.2005	7.55	7.4	7.45	2839851
AKBNK	06.06.2005	7.4	7.1	7.15	4282935
AKBNK	07.06.2005	7.25	7.1	7.15	2907253
AKBNK	08.06.2005	7.3	7.15	7.2	4725091
AKBNK	09.06.2005	7.25	7	7.1	3235137
AKBNK	10.06.2005	7.15	7	7.15	3141137
AKBNK	13.06.2005	7.15	7	7.05	1479634
AKBNK	14.06.2005	7.15	7.05	7.1	1637347
AKBNK	15.06.2005	7.35	7.05	7.2	7599413
AKBNK	16.06.2005	7.35	7.2	7.25	4018068
AKBNK	17.06.2005	7.4	7.2	7.2	4538628
AKBNK	20.06.2005	7.35	7.2	7.3	1915270
AKBNK	21.06.2005	7.4	7.25	7.3	3399439
AKBNK	22.06.2005	7.45	7.25	7.45	3447864
AKBNK	23.06.2005	7.65	7.4	7.5	3415664
AKBNK	24.06.2005	7.65	7.4	7.6	2362298
AKBNK	27.06.2005	7.6	7.45	7.45	1174371
AKBNK	28.06.2005	7.55	7.35	7.45	1438673
AKBNK	29.06.2005	7.7	7.45	7.65	4330290
AKBNK	30.06.2005	7.9	7.65	7.7	3552050
AKBNK	01.07.2005	6.85	6.55	6.75	5771463
AKBNK	04.07.2005	6.95	6.75	6.8	3465955
AKBNK	05.07.2005	6.85	6.65	6.7	1866120
AKBNK	06.07.2005	6.95	6.7	6.9	2508539
AKBNK	07.07.2005	6.9	6.65	6.85	3655795
AKBNK	08.07.2005	6.9	6.75	6.75	2896762

AKBNK	11.07.2005	6.85	6.65	6.7	2638238
AKBNK	12.07.2005	6.75	6.55	6.6	2700802
AKBNK	13.07.2005	6.75	6.6	6.65	3284813
AKBNK	14.07.2005	6.8	6.65	6.75	4522550
AKBNK	15.07.2005	6.85	6.7	6.7	6842924
AKBNK	18.07.2005	6.75	6.6	6.6	2200721
AKBNK	19.07.2005	6.8	6.6	6.8	3457062
AKBNK	20.07.2005	7	6.75	6.8	13368099
AKBNK	21.07.2005	6.9	6.75	6.75	8857585
AKBNK	22.07.2005	6.95	6.75	6.75	9628263
AKBNK	25.07.2005	7.15	6.75	7.15	10491668
AKBNK	26.07.2005	7.25	6.9	6.9	7463203
AKBNK	27.07.2005	7.15	6.95	7.15	3929311
AKBNK	28.07.2005	7.45	7.1	7.45	8256377
AKBNK	29.07.2005	7.55	7.3	7.35	5275757
AKBNK	01.08.2005	7.65	7.4	7.45	6190853
AKBNK	02.08.2005	7.55	7.4	7.5	3855950
AKBNK	03.08.2005	7.65	7.35	7.55	3604726
AKBNK	04.08.2005	7.8	7.5	7.7	4312429
AKBNK	05.08.2005	7.75	7.5	7.7	2323597
AKBNK	08.08.2005	8	7.65	7.95	1959653
AKBNK	09.08.2005	7.95	7.75	7.85	2662004
AKBNK	10.08.2005	7.95	7.5	7.6	5320409
AKBNK	11.08.2005	7.6	7.4	7.45	5102659
AKBNK	12.08.2005	7.55	7.05	7.05	8873757
AKBNK	15.08.2005	7.1	6.9	7.1	4519466
AKBNK	16.08.2005	7.35	7.1	7.2	6114655
AKBNK	17.08.2005	7.2	7	7	4734215

AKBNK	18.08.2005	7.3	6.95	7.05	4577031
AKBNK	19.08.2005	7.15	6.95	7.1	4769835
AKBNK	22.08.2005	7.25	7.05	7.2	2861477
AKBNK	23.08.2005	7.3	7.1	7.25	2878310
AKBNK	24.08.2005	7.55	7.2	7.5	5034819
AKBNK	25.08.2005	7.75	7.45	7.7	3524681
AKBNK	26.08.2005	7.8	7.55	7.6	2089800
AKBNK	29.08.2005	7.6	7.45	7.55	2062822
AKBNK	31.08.2005	7.95	7.5	7.95	5692730
AKBNK	01.09.2005	8.65	7.95	8.65	8810409
AKBNK	02.09.2005	8.55	8.2	8.3	4872975
AKBNK	05.09.2005	8.5	8.15	8.45	2213939
AKBNK	06.09.2005	8.45	8.25	8.3	1897756
AKBNK	07.09.2005	8.4	8.15	8.15	3456920
AKBNK	08.09.2005	8.25	8.05	8.1	3332909
AKBNK	09.09.2005	8.35	8.15	8.3	2993982
AKBNK	12.09.2005	8.5	8.35	8.4	5858360
AKBNK	13.09.2005	8.8	8.15	8.25	6160720
AKBNK	14.09.2005	8.3	8	8.25	4944014
AKBNK	15.09.2005	8.65	8.2	8.65	3188819
AKBNK	16.09.2005	9	8.55	8.9	4701368
AKBNK	26.09.2005	8.5	8.5	8.6	14046
AKBNK	27.09.2005	8.95	8.7	8.75	2504852
AKBNK	28.09.2005	8.85	8.45	8.55	5371841
AKBNK	29.09.2005	8.85	8.55	8.75	6503197
AKBNK	30.09.2005	9.05	8.7	8.95	6628531
AKBNK	03.10.2005	9.3	8.65	9.25	6346267
AKBNK	04.10.2005	9.9	9.4	9.7	15702989

AKBNK	05.10.2005	9.75	9.45	9.55	829397
AKBNK	06.10.2005	9.25	8.7	8.75	5394812
AKBNK	07.10.2005	8.85	8.55	8.65	11010847
AKBNK	10.10.2005	8.8	8.45	8.7	6250635
AKBNK	11.10.2005	9.15	8.6	9.1	5944347
AKBNK	12.10.2005	9.1	8.65	8.7	4750746
AKBNK	13.10.2005	8.75	8.25	8.35	6309763
AKBNK	14.10.2005	8.45	7.8	8	10477967
AKBNK	17.10.2005	8.2	7.95	8.2	7232344
AKBNK	18.10.2005	8.4	8.1	8.2	4248637

3. Referanslar

Anderson, D., McNeill, G., **Artificial Neural Networks Technology**, A DACS State-of-the- Art Report, Contact Number F30602-89-C-0082, New York, 1992.

Bergerson, K. y Wunsch, D.C. (1991): "**A Commodity Trading Model Based on a Neural Network-Expert System Hybrid**", Proceedings of the IEEE International Conference on Neural Networks, pag 1289-1293, Seattle y publicado en Neural Networks in Finance and Investing. Ed Trippi y Turban. Probus Publishing Company, 1992, Chicago

Chenoweth, T., Obradovic, Z., **A Multi-Component Nonlinear Prediction System for the S&P 500 Index**, Neurocomputing, Vol. 10, 1996, pp. 275-290.

Coakley, J.R., Brown, C.E., "Artificial Neural Networks In Accounting And Finance: Modeling Issues", **International Journal Of Intelligent Systems In Accounting, Finance & Management**, 2000, Vol.9, pp. 119-144.

Collard, J. E. (1991) A B-P ANN Commodity Trader, **Advances in Neural Information Processing System**, vol. 3, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, ed., Morgan Kaufmann, San Mateo, CA.

Donaldson, R.G., Kamstra, M., **Forecast Combining with Neural Networks**, **Journal of Forecasting**, Vol. 15, 1997., pp. 49-61.

Enke, D., Thawornwong, S., “The Use Of Data Mining And Neural Networks For Forecasting Stock Market Returns”, **Expert Systems With Applications**, 2003, Vol21, pp. 1-14.

Ergezer, H., Dikmen, M., Özdemir, E., **Yapay Sinir Ağları ve Tanıma Sistemleri**, Pivolka 6, ELYAD-DAL Araştırma Laboratuarları, Başkent Üniversitesi, Ankara, 2003.

Fieldsend, J.E., “Pareto Evolutionary Neural Networks”, 2001, Vol. 227, pp. 2137-2157 (8)

Grudnitzky, G., Osburn, L., **Forecasting S&P and Gold Futures Prises: An Application of Neural Networks**, **Journal of Futures Markets**, September 1993, vol. 13, No. 6, pp.631-643.

Halıcı, U., Üçoluk, G., **Toplumsal, Felsefi ve Hukuksal Boyutları ile Yapay Zekâ**, Orta Doğu Teknik Üniversitesi, Ankara, 1993.

Hamida, S.A., Iqbal, Z., “Using Neural Networks For Forecasting Volatility Of S&P 500 Index Futures Prices”, **Journal of Business Research**, 2004, Vol. 57, pp. 1116– 1125.

Hill, T., O'Conner, M., Remus, W., "Neural Network Models For Time Series Forecasts", **Management Science**, 1996, Vol. 42, No.7, pp. 1082-1092.

Kaastra, I., Boyd, M., "Designing A Neural Network For Forecasting Financial and Economic Time Series", **Neurocomputing**, 1996, Vol. 10, pp. 215-236.

Kalyvas, E., **Using Neural Networks And Genetic Algorithms To Predict Stock Market Returns**, Master Thesis, University Of Manchester, 2001.

Kamijo, K. and Tanigawa, T. (1993). «**Stock Price Pattern Recognition: A Recurrent Neural Network Approach**». *Neural Networks in Finance and Investing*, Trippi, R.R. et al. (eds.), 357-370.

Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M., **Stock Market Prediction System With Modular Neural Networks**, in Trippi, Turban,...1990, pp. 497-510

Kohara, K., Ishikawa, T., Fukuhara, Y., Nakamura, Y., **Stock Price Prediction Using Prior Knowledge and Neural Networks**, *Intelligent Systems in Accounting, Finance and Management*, Vol. 6, pp. 11-12, 1997.

Kryzanowski, L., Galler, M., Wright, D.W., **Using Artificial Networks to Pick Stocks**, *Financial Analyst's Journal*, August 1993, pp. 21-27.

Kutsurelis, J.A., Terasava, K., Gates, W.R., **Forecasting Financial Markets Using Neural Networks: An Analysis Of Methods And Accuracy**, Master Thesis, Naval Postgraduate School, Monterey, California, 1998.

Lendasse, A., Lee, J., de Bodt, E., Wertz, V., Verleysen, M., "Input Data Reduction For The Prediction Of Financial Time Series", **ESANN'2001 Proceedings – European Symposium On Artificial Neural Networks Bruges (Belgium)**, April 25-27, 2001, pp. 237-244.

Li, E.Y., **Artificial Neural Networks and Their Business Applications**, **Information & Management**, vol. 27, 1994, pp. 303-313.

Lin, F., Yu, X.H., Gregor, S., Irons, R., "Time Series Forecasting With Neural Networks", **Complexity International**, 1995, Vol. 02, pp.61-99

Lisi, F., Schiavo, R.A., "A Comparison Between Neural Networks And Chaotic Models For Exchange Rate Prediction", **Computational Statistics & Data Analysis**, 1999, Vol. 30, pp. 87-102.

Man-Chung, C., Chi-Chung, W., Chi-Chung, L., "**Financial Time Series Forecasting By Neural Network Using Conjugate Gradient Learning Algorithm And Multiple Linear Regression Weight Initialization**", 1998, Vol. 53, pp. 182-228.

Marose, R. A.: 1990, '**A Financial Neural Network Application**'. *AI Expert* May, 50–53.

Minsky, M., Papert, S., **Perceptrons**, The MIT Press, 1969.

Nygren, K., **Stock Prediction: A Neural Network Approach**, Master Thesis, Royal Institute Of Technology, KTH, 2004.

Op't Landt, F.W., **Stock Price Prediction Using Neural Networks**, Master Thesis, Leiden University, 1997.

Öztemel, E. (2003), **Yapay Sinir Ağları**, Papatya Yayıncılık, İstanbul.

Refenes, A.N., Zapranis, A., Francis, G., **Stock Performance Modeling Using Neural Networks: A Comparative Study with Regression Models**, *Neural Networks*, vol. 7, No.2, 1994, pp. 375-388.

Refenes, A.N., **Neural Networks in the Capital Markets**, John Wiley & Sons, 1995.

Refenes, A.N., Skapura, D., **Building Neural Networks**, Addison Wesley Publishing Co., 1996.

Refenes, A.N., Abu-Mostafa, Y., Moody, J., Weigend, A., Eds., **Neural Networks in Financial Engineering, Proceedings of the Third International Conference on Neural Networks in Capital Markets**, World Scientific, 1996.

Refenes, A.N., Barr, D.S., Mani, G., **Using Neural Nets to Manage Investments**, *AI Expert*, February, 1997, pp. 16-21.

Rosenblatt, F., "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain", **Psychoanalytic Review**, 1958, Vol. 65, pp. 386-408.

Saad, E.W., Prokhorov, D.V., Wunsch, D.C., **Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks**, *IEEE Transactions on Neural Networks*, Vol. 9, 1998, pp. 1456-1470.

Singh, S., "Noise Impact On Time – Series Forecasting Using An Intelligent Pattern Matching Technique", **Pattern Recognition**, 1999, Vol. 32, Issue 8, pp. 1389-1398.

Singh, S., Fieldsend, J., **“Financial Time Series Forecasts Using Fuzzy And Long Memory Pattern Recognition Systems”**, 2001, Vol. 13, Issue 21, pp. 31-61.

Schoeneburg, E., **Stock Price Prediction Using Neural Networks: A Project Report**, Neurocomputing, vol. 2, 1990, pp. 17-27.

Singleton, J.C, Surkan, A.J, **Bond rating with neural networks**, in: A. Refenes (Ed.), Neural Networks in the Capital Markets, Wiley, Chichester, 1995, pp. 301 -- 307.

Soumitra Dutta, Shashi Shekhar, W. Y. Wong: **Decision Support in Non-Conservative Domains: Generalization with Neural Networks**. IFIP Congress (3) 1992: 259-265

Swales, G.S.Jr., Yoon, Y., **Applying Artificial Neural Networks to Investment Analysis**, Financial Analysts Journal, September-October, 1992, pp. 78-80.

Tan, C.N.W., **An Artificial Neural Networks Primer With Financial Applications Examples in Financial Distress Predictions And Foreign Exchange Hybrid Trading System**, PhD Thesis, School Of Information Technology, Bond University, Australia, 1997.

Trippi, R.R., DeSieno, D., **Trading Equity Index Futures With a Neural Network**, The Journal of Portfolio Management, Fall 1992, pp. 27-33.

Verleysen, M., de Bodt, E., Lendasse, A., "Forecasting Financial Time Series Through Intrinsic Dimension Estimation And Non-linear Data Projection", **Proceedings of IWANN'99 – International Work – Conference On Artificial And Natural Neural Networks, Alicante (Spain)**, June 2-4, 1999, pp. II-596 – II-605.

White, H., **Economics Prediction Using NN; The Case of IBM Daily Stock Returns**, in Trippi, R.R., Turban, E., Neural Networks in Finance and Investing, Irwin Professional Publishing, 1997., pp. 469-481.

Widrow, B., Hoff, M.E., "Adaptive Switching Circuits", 1960, **WEST-CON Convention**, Record Part IV, pp. 96-104.

Wong, F.S., Long, P.Z., Goh, T.H., Quek, B.K., **Fuzzy Neural Systems for Stock Selection**, Financial Analyst Journal, January-February 1992, pp. 47-53.

Wong, B.K., Bodnovich, T.A., Selvi, Y., **Neural Network Applications in Business: A Review and Analysis of the literature (1988-95)**, Decision Support Systems, vol. 19,1997, pp. 301-320.

Yoon, Y., Guimaraes, T., Swales, G., **Integrating Artificial Neural Networks With Rule-Based Expert Systems**, **Decision Support Systems**, vol. 11, 1994, pp. 497-507.

Yoon, Y., Swales, G., **Predicting Stock Price Performance: A NN Approach**, in Trippi,R.R., Turban, E., **Neural Networks in Finance and Investing**, Irwin ProfessionalPublishing, 1996., pp.483-495.

Zahedi, F., **Intelligent Systems for Business, Expert Systems With Neural Networks**,Wodsworth Publishing Inc., 1993.

Zahedi, F., **A Meta-Analysis of Financial Applications of Neural Networks**, **International Journal of Computational Intelligence and Organizations**, Vol. 1, No. 3, 1997, pp. 164-178.

Zekić, M., **Structure Optimization of Neural Networks in Relation to Underlying Data**,Proceedings of the 7th International Conference on Operational Research KOI '98,Eds. Aganović, I., Hunjak, T., Scitovski, R., Rovinj, Croatia, September 30-October 2,1998., p.135-144.

Zorin, **Alexey, Stock Price Prediction Kohonen Versus Back Propagation**, Technical University Of Riga.

