

**FİNANS SEKTÖRÜNDE YAPAY SİNİR AĞI
UYGULAMASI**

YÜKSEK LİSANS TEZİ

**End. Müh. Bora ÇEKYAY
(507031106)**

Tezin Enstitüye Verildiği Tarih : 9 Mayıs 2005

Tezin Savunulduğu Tarih : 30 Mayıs 2005

Tez Danışmanı : Yrd. Doç. Dr. Şule Önsel Şahin (İ.T.Ü.)

Diğer Jüri Üyeleri : Prof. Dr. M. Nahit Serarslan (İ.T.Ü.)

Prof. Dr. Ertuğrul Karsak (G.Ü.)

MAYIS 2005

ÖNSÖZ

Sigorta sektörü son yıllarda, ülkemizde hızla gelişen bir sektördür. Bu gelişim, yabancı firmaların pazar payı kapmak için sektöre, yerli firmalarla ortak girişim (joint venture) yaparak girmelerinden görülmektedir. Ancak dünya ile mukayese edildiğinde sigorta sektörümüz hem tecrübe ve birikim açısından hem de yasal düzenlemeler açısından oldukça geridir. Bu açıdan sigorta sektörü ile ilgili akademik çalışmaların yapılmasının ülkedeki bu konuda zayıf kalmış bilinci körüklemesi açısından oldukça önemli olduğunu düşünüyorum. Hazırlamış olduğum bu tezin amacına hizmet edeceğine de yürekten inanıyorum.

Sigortacılık, eskiden beri olasılıkla ve istatistikle anıla gelmiş bir kavramdır. Bunun en temel nedeni, sigortacılığın, olasılık teorisinden gelen güçlü dayanaklarının olmasıdır. Dolayısıyla, prim hesabı da istatistiksel yöntemler kullanılarak yapılmıştır. Ancak, çok sayıda değişkenin dikkate alınmasının gerekliliği, farklı yöntemlerin kullanımını gündeme getirmiştir.

Bu tez kapsamında, sigortacılıkta prim hesabı yapmak için alternatif bir yöntem önerilmiştir. Bu yöntem bütünüyle yapay sinir ağlarını kullanmaktadır. Önerilen modelin uygulanabilirliğini göstermek açısından bir Türk sigorta firmasından alınan gerçek verilerle uygulama yapılmış ve model, firmada uygulanan sistem ile mukayese edilmiştir. Çıkan sonuçlar değerlendirildikten sonra bu modelin firmada bir karar destek sistemi olarak kullanılabileceğine karar verilmiştir.

Çalışmamın her aşamasında benden bilgi ve deneyimini esirgemeyen tez danışmanım Yrd. Doç. Dr. Şule Önsel Şahin'e, güler yüzü, anlayışı ve samimiyeti için Şule Ablam'a sonsuz minnettarlıklarımı sunarım. Ayrıca değerli katkıları için hocam Prof. Dr. M. Nahit Serarslan' a, hayatıma kattıkları anlam ve bana gösterdikleri sabır için sevgili Bahar Orhan (Çekyay)' a ve aileme çok teşekkür ederim.

Mayıs 2005

Bora Çekyay

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER	iii
KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
2. YAPAY SİNİR AĞLARI	3
2.1. Yapay Sinir Hücresi	5
2.2. Algılayıcı ve Çok Katmanlı Ağlara Geçiş	7
2.3. Yapay Sinir Ağlarının Eğitilme Şekilleri	8
2.4. Kohonen'in Özörgütlenmeli Harita Ağı	9
2.4.1. Özörgütlenme Süreci	9
2.4.2. Özörgütlemeli Harita Örneği	11
2.4.3. Kohonen Özörgütlenmeli Harita Ağı'nın (KÖHA) Kullanımında Uygun Parametrelerin Belirlenmesi	14
2.4.3.1. Başlangıçta Rassal Ağırlık Atama	14
2.4.3.2. Girdilerin Ağa Girilme Sırası	14
2.4.3.3. Komşuluk Fonksiyonu	15
2.4.3.4. Durma Kriteri	16
2.5. Çok Katmanlı Ağlar (ÇKA)	16
2.5.1. Hareket Fonksiyonları	17
2.5.1.1. Lojistik Fonksiyonu	18
2.5.1.2. Hiperbolik Tanjant Fonksiyonu	18
2.5.2. İleri Beslemeli Çok Katmanlı Ağların Uygulamasında Dikkat Edilecek Hususlar	19
2.5.2.1. Mimarinin Belirlenmesi	19
2.5.2.2. Ağı Eğitilmesi	20
2.5.2.3. Ağ Çıktısındaki Kusurlar	21
2.5.3. Çok Katmanlı Ağların Eğitilmesi	22
2.5.3.1. Algılayıcı Öğrenme Algoritması	24
2.5.3.2. Geri Yayılmalı Algoritma	25
2.5.3.3. Momentumlu Geri Yayılmalı Algoritma ile Eğitim Örneği	26
2.5.3.4. Değişken Öğrenme Oranı Sezgiseli	31

2.5.3.5. RPROP Algoritması (Resilient Propagation)	32
2.5.3.6. Ölçeklendirilmiş Çekimli Eğitim Algoritması (Scaled Conjugate Gradient Algorithm)	34
3. SİGORTA VE PRİM HESABI	36
3.1. Giriş	36
3.2. Sigortanın Fonksiyonları	37
3.3. Sigortacılıkta Riskin Tahmini ve Prim Hesabı	39
3.3.1. Güçlü Büyük Sayılar Kanunu	39
3.3.2. Sigorta Priminin Hesaplanması	40
4. MATLAB VE YAPAY SİNİR AĞLARI	43
4.1. Yapay Sinir Ağı Arayüzüne Giriş	43
4.2. Ağın Oluşturulması	43
4.3. Ağın Eğitilmesi	46
4.4. Ağ Çıktılarının Komut Satırı Çalışma Ortamına Transfer Edilmesi	48
4.5. Komut Satırı Çalışma Ortamından Yapay Sinir Ağı Modülüne Veri Transferi	49
4.6. Bir Değişkenin Bir Dosyaya Kaydedilmesi, Silinmesi ve Daha Sonra Geri Çağırılması	49
5. ÖNERİLEN MODEL	51
5.1. İstatistiksel Yöntemin Eleştirilmesi	51
5.2. Önerilen Yöntem	52
5.3. Uygulama Aşamaları	53
5.3.1. Veri kümesinin düzenlenmesi	53
5.3.2. Verilerin Analize Hazırlanması	55
5.3.3. KÖHA Aşaması	56
5.3.4. Çok Katmanlı İleri Beslemeli Ağın Oluşturulması Aşaması	59
5.3.5. Modelin Simülasyonu ve Yorumlar	59
6. SONUÇLAR VE GELİŞTİRME ÖNERİLERİ	62
KAYNAKLAR	63
ÖZGEÇMİŞ	66

KISALTMALAR

KÖHA	: Kohonen'in Özörgütlenmeli Harita Ağı
ÇKA	: Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı
SMB	: Sınır Mantık Birim
MİB	: Merkezi İşlem Birimi (CPU)
ÖÇE	: Ölçeklendirilmiş Çekimli Eğitim Algoritması
ÇEG	: Çekimli Eğitim Geri Yayılmalı Algoritması
BFGS	: Broyden, Fletcher, Goldfarb, Shanno Algoritması
SP	: Saf prim
OHS	: Ortalama hasar sıklığı
OHM	: Ortalama hasar miktarı
THS	: Toplam hasar sayısı
TKS	: Toplam kişi sayısı
THM	: Toplam hasar miktarı
KA	: Kullanıcı arayüzü

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Girdi sırasının KÖHA çıktısına etkisi	15
Tablo 2.2. Girdiler ve beklenen değerler	27
Tablo 2.3. XOR problemini öğrendikten sonra ağın ürettiği çözümler ve hata oranları	31
Tablo 3.1. Prim hesabı örneğinin verileri.....	42
Tablo 5.1. KÖHA aşaması sonucunda oluşan kümelerin merkez değerleri ...	57
Tablo 5.2. Gruplara göre hasar olasılıkları.....	58
Tablo 5.3. Önerilen model ile gerçekleşen değerler ve firma sistemi arasındaki korelasyonlar	60

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1. : Biyolojik Sinir Hücresinin Yapısı	4
Şekil 2.2. : Sinaptik Boşluğun Yapısı	4
Şekil 2.3. : İlk Yapay Sinir Hücresi Örneği (Müler vd., 1990)	5
Şekil 2.4. : VE ve VEYA fonksiyonlarının doğrusal olarak ayrılabilmesi ve XOR fonksiyonunun doğrusal olarak ayrılamaması.	6
Şekil 2.5. : Algılayıcı	7
Şekil 2.6. : Kohonen Özörgütlenmeli Harita Ağı	9
Şekil 2.7. : Çıktı katmanındaki sinir hücrelerinin, girdi vektörlerinin oluşturduğu kümelere doğru hareketi.	10
Şekil 2.8. : Komşuluk büyüklükleri	11
Şekil 2.9. : İleri Beslemeli Çok Katmanlı Ağ Mimarisi	17
Şekil 2.10. : Lojistik Fonksiyonunun Grafiği	18
Şekil 2.11. : Hiperbolik Tanjant Fonksiyonunu Grafiği	19
Şekil 2.12. : Eğitim miktarının genelleme yeteneğine etkisi	21
Şekil 2.13. : XOR problemi için tasarlanmış Çok Katmanlı Ağ.....	27
Şekil 2.14. : İleri doğru hesaplama	28
Şekil 2.15. : XOR ağının ağırlıklar değiştirildikten sonraki durumu.....	30
Şekil 2.16. : XOR problemi öğrenildikten sonraki ağırlıklar	31
Şekil 4.1. : Network / Data Manager Arayüzü	44
Şekil 4.2. : Create New Data Ekranı	44
Şekil 4.3. : Create New Network Ekranı	45
Şekil 4.4. : Network: ANDNet Ekranı.....	46
Şekil 4.5. : View of New Network Ekranı	47
Şekil 4.6. : Eğitim Performansı Ekranı	48
Şekil 4.7. : Export or Save Network/Data Manager Ekranı.....	48
Şekil 4.8. : Import or Load Network / Data Manager Ekranı	49
Şekil 5.1. : Önerilen yöntemin ana kademeleri.....	52

SEMBOL LİSTESİ

X_i	: Ağa girilen girdi değişkeni
W_{ij}	: Sinaptik ağırlıklar
θ	: Sınır değeri
T_i	: Ağın eğitiminde kullanılan hedef değerler
d_i	: Öklidyen uzaklık
T	: En büyük yineleme sayısı
$\alpha(t)$	: t . yinelemedeki öğrenme hızı
N_i	: i . sinir hücresinin komşuluğundaki sinir hücreleri kümesi
r_i	: Çıktı katmanındaki i . sinir hücresinin koordinatları
$h(t, r)$	: Zamana ve uzaklığa göre azalan komşuluk fonksiyonu
N	: Çıktı katmanındaki sinir hücresi sayısı
y	: Ağın çıktı değeri
η	: Momentum katsayısı
$\lambda^{+/-}$	: RPROP algoritmasındaki küçültme ve büyütme katsayıları
λ^k	: Lagrange çarpanı
Y_i	: Rassal değişken
μ	: İlgili rassal değişkenin ortalama değeri
$E[Y_i]$	: Y_i rassal değişkeninin beklenen değeri
$\sigma_{Y_i}^2$	: Y_i rassal değişkeninin varyansı
n	: Kümedeki birey sayısı

FİNANS SEKTÖRÜNDE YAPAY SINIR AĞI UYGULAMASI

ÖZET

Günümüz dünyasında ülke ve dünya ekonomileri açısından finans sektörünün önemi, tartışılmazdır. Finans sektörünün, en önemli aktörleri ise bankalar ve tabii ki sigorta firmalarıdır. Sigorta firmaları, piyasadan topladığı likiditeyi, toplumdaki hasarların ve riskin azaltılmasında kullanmaktadır. Bu açıdan verimlilik artırımı için sağlıklı işleyen bir sigorta sektörü hayatidir.

Sigorta sektörünün hayatta kalabilmesi için karşılaşılan riskin tanımlanması ve ölçülmesi gerekmektedir. Çünkü bu riske göre müşterilerden para (prim) alınacaktır ve primler ile müşterilerin hasarları (sağlık sigortasında da oluşan tedavi masrafları “hasar” diye isimlendirilir) karşılanacaktır. En ideal durumda hesaplanan primin gerçekleşecek hasara eşit olması gerekmektedir. Ancak günlük hayattaki belirsizlik gerçekleşecek hasarı tam olarak tahmin etmeyi imkânsız hale getirmektedir. Bu açıdan istatistiksel yasalardan ve tekniklerden yararlanılarak prim hesabı yapılmaktadır. Ancak bu teknikler de dikkate alınması gereken değişken sayısı, arttıkça uygulanması zor hale gelmektedir. Bu nedenle alternatif yöntemlere ihtiyaç duyulmaktadır. Bu tez kapsamında da sağlık sigortasında prim hesabının yapılması için yapay sinir ağlarını kullanan bir alternatif yöntem önerilmiş ve uygulanabilirliği gösterilmiştir. Sağlık sigortasının seçilmesinin sebebi hasar miktarını etkileyen değişken sayısının çok fazla olmasıdır.

Yöntemde iki farklı sinir ağı, iki farklı amaç için kullanılmaktadır. Kullanılan ilk sinir ağı Kohonen’in Özörgütlenmeli Harita Ağı’dır (KÖHA). Bu ağ kullanılarak mevcut müşteriler, 8 farklı kümeye ayrılmıştır. Daha sonra elde edilen her küme için hasar yapma olasılığı hesaplanmıştır. Kullanılan ikinci ağ ise Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı’dır (ÇKA). Bu ağ ile kişilerin demografik bilgileri ile gerçekleşen hasar miktarları arasında var olduğu varsayılan fonksiyona yakınsanmaya çalışıldı. Eğitilen bu iki ağ kullanılarak yeni gelen bir kişinin demografik bilgileri kullanılarak prim hesabı yapılmıştır. KÖHA kullanılarak, kişinin hangi kümeye girdiği ve böylece hasar olasılığı belirlenmiştir. ÇKA kullanılarak da kişinin eğer hasar yaptıysa bu hasarın beklenen değeri hesaplanmıştır. Bu iki değer çarpımı ise prim olarak kabul edilmiştir.

Modelin uygulanabilirliğinin gösterilebilmesi için Türkiye’deki bir sigorta firmasından alınan veriler kullanılmıştır. Analiz sonucunda önerilen yöntemin firma uygulanan skrolama yöntemiyle paralel sonuçlar verdiği görülmüştür. Bu skrolama tekniğinin dezavantajları (güncelleme zorluğu, hasar tahminlerinin dayandığı anlamlı bir temel eksikliği) dikkate alındığında önerilen yöntem başarılı bulunmuş ve firmaya karar destek sistemi olarak önerilmiştir.

ARTIFICIAL NEURAL NETWORK APPLICATION IN FINANCE SECTOR

SUMMARY

In today's world, the importance of finance sector in national and global economy is obvious. The most important actors of finance sector are banks and insurance companies. Insurance firms, use the money which they collect from the policyholders to reduce the loss and risk in the society. Therefore, an insurance sector whose mechanisms work properly is vital to increase the efficiency in a country.

For the survival of insurance sector, measuring and defining the faced risk is needed, because risk premiums taken from customers are determined using these risk measures and claim costs are paid with the premiums. Ideally, calculated premium and claim cost should be equal. On the other hand, uncertainty in life makes it impossible to predict the claim cost with high probability. Thus, to evaluate the premium, statistical techniques have been used. However, the more variables are taken into consideration, the more difficult to apply these statistical techniques. In this respect, alternative methods are required. In this thesis, an alternative method which uses Artificial Neural Networks is introduced and its applicability is shown. The reason why the health insurance is selected in this thesis is the large number of variables that affect the claim cost.

The proposed method integrates two different neural nets which are used for different purposes. One of the neural networks used is Kohonen's Self-organizing Map (SOM). All customers are clustered into 8 segments using this tool. After that, for each set, probability of loss is calculated. The second neural network which is used in the model is Multi-Layer Feedforward Neural Network (MLFNN). This network is used to converge to the function which is assumed to exist between the demographic data and claim costs of people. After training, through use of demographic data, these two networks are used to assess the premium. With SOM, the cluster to which individual belongs and thus the claim probability of him/her is determined. Given that the claim has occurred, the expected value of the claim cost is evaluated by using MLFNN. The premium is taken as the product of these two values.

To demonstrate the applicability of the model, a real-world dataset which belongs to a Turkish insurance firm is used. After the analysis, it is observed that the proposed model and the scoring system in present use in the firm give similar results. Due to the disadvantages of scoring system (upgrade problem, lack of ready explanation of the basis of deriving the predicted claim) the proposed method is more successful and it is proposed as a decision support system for the insurance firm.

1. GİRİŞ

Bu tez kapsamında sigortacılıkta prim hesaplama problemine alternatif bir çözüm üretilmeye çalışılmıştır. Sigortacılık, olasılık teorisindeki "Büyük Sayılar Kanunu" üzerine kurulmuştur. Çok sayıda, aynı riski paylaşan ve bu risk karşısında benzer hasar dağılımına sahip bireylerin bir araya getirilmesi gerekmektedir. Daha sonra bu kişilerin ortalama hasarı büyük bir olasılıkla (veya küçük bir hata ile) tahmin edilebilmektedir. Ancak, özellikle sağlık sigortası gibi hasar olasılığını ve gerçekleşen hasar miktarını birçok değişkenin belirlediği sigorta türlerinde, yukarıda bahsedilen varsayımı gerçeklemek oldukça zordur. Bu nedenle alternatif hesap yöntemleri gündeme gelmiştir. Günümüzde birçok sigorta firması, "skorlama" yöntemini kullanmaktadır. Ancak bu yöntem tahmin edilen prim miktarları için hazır bir açıklama sunamamaktadır. Skorlama da kullanılan parametrelerin belirlenmesi ise güç bir işlemdir. BU tezde, skorlama sistemin yerine daha otomatik çalışabilen ve klasik istatistiksel yöntemlerin kullandığı beklenen değer mantığına sahip bir yöntem önerilmiştir. Önerilen yöntem tamamıyla yapay sinir ağlarını kullanmaktadır. Bu yöntemin uygulanabilirliğini göstermek amacıyla da Türkiye'deki bir sigorta firmasından alınan veriler kullanılmıştır. Yapılan analiz sonucunda elde edilen sonuçlara göre de firmada uygulanan sistem ile karşılaştırmalar yapılmıştır.

Bir sonraki bölümde, öncelikle yapay sinir ağlarına genel bir giriş yapılmıştır. Gözetimli ve gözetimsiz eğitim tipleri ayrı ayrı incelenmiş ve bunlara örnek olarak bu çalışmada kullanılan sinir ağı tipleri (Kohonen'nin Özörgütlenmeli Harita Ağı, Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı) ayrıntıları ile açıklanmıştır. Her iki ağ tipi içinde eğitim algoritmaları açıklanmış ve iki sayısal örnek verilmiştir.

Üçüncü bölümde öncelikle sigortacılığın tanımı ve temel fonksiyonları açıklanmıştır. Bundan sonra da prim hesabının istatistiksel olarak nasıl yapıldığı örnekler üzerinde gösterilmiştir. Bu bölümde açıklanan teorik bölümler, bu tezde önerilen yönteminde temelinde yatan felsefeyi oluşturmaktadır.

Tez kapsamındaki tüm sinir ağı uygulamaları MATLAB programı kullanılarak yapılmıştır. Bu nedenle, dördüncü bölümde MATLAB programının "Neural Networks" modülünün arayüzünün kısa bir kullanma kılavuzu teze eklenmiştir.

Beşinci bölümde, uygulama kapsamında yapılan tüm çalışmalar ayrıntılı bir şekilde açıklanmıştır. Prim hesabında uygulanan yöntemlerin kısa bir eleştirisi yapıldıktan sonra önerilen yöntem adım adım yazılmıştır. Bu açıklamalar ışığında firmadan alınan veriler ile modelin uygulanabilirliği gösterilmiştir. Elde edilen sonuçlar, firmada uygulanan sistemin sonuçları ile karşılaştırılmış ve yorumlanmıştır.

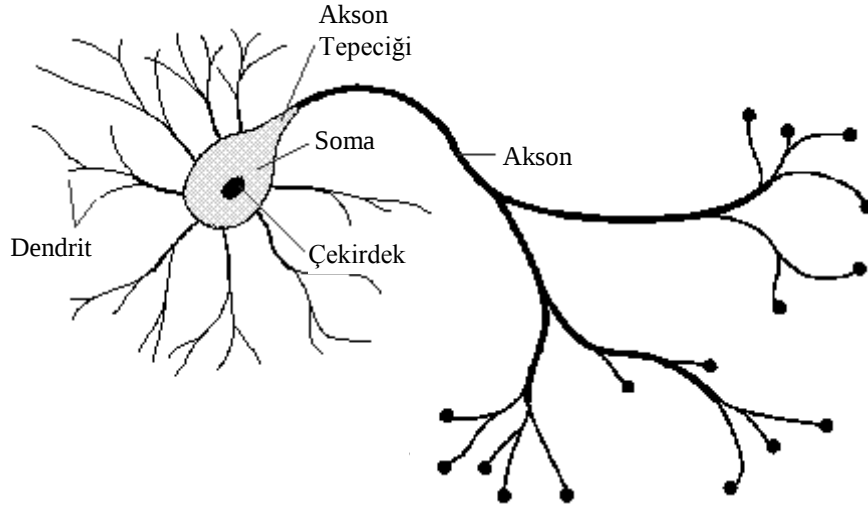
Son bölümde ise uygulama aşamasında elde edilen sonuçlar değerlendirildikten sonra ileriki çalışmalara yön verecek geliştirme önerilerinde bulunulmuştur.

2. YAPAY SİNİR AĞLARI

Doğa, insanoğlu için ne kadar büyük bir öğretmen olduğunu tarihte birçok defa ispatlamıştır. Bilim adamlarına birçok konuda esin kaynağı olmuş ve insanlığın, sahip olduğu bilim çıtasını daha da yukarılara taşımasını sağlamıştır. Doğanın anlaşılması için birçok bilim dalı ortaya çıkmıştır ve yüzyıllardır yapılan çalışmaların en büyük sonucu daha kat etmemiz gereken çok yol olduğu gerçeğidir.

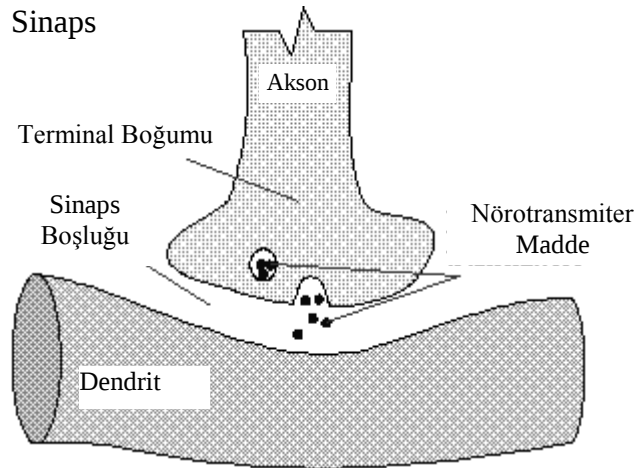
Son yıllarda doğada işleyen mekanizmalar matematiksel modellerin çözümünde de kullanılmaya başlanmıştır. Genetik algoritma, bulanık mantık ve benzetimli tavlama gibi yöntemler çok karmaşık problemlerin çözümünde başarıyla uygulanmaktadır. Bu tez kapsamında da, yine insan sinir sisteminden esinlenilerek ortaya konulmuş Yapay Sinir Ağları temel alınmıştır. Öncelikle insan sinir sisteminin yapısı elemanter seviyede verilecek ve daha sonra yapay sinir ağlarının genel yapısı açıklanacaktır. Bu temel bilgilerden sonra yapay sinir ağlarını eğitme şekilleri tanıtılacak ve daha sonra bu tezde kullanılan sinir ağı çeşitleri ayrıntılı olarak ele alınacaktır.

Şekil 2.1’de basit bir biyolojik sinir hücresi modelinin genel yapısı gösterilmiştir. İnsan sinir sistemi bu tipteki milyonlarca sinir hücresinin birleşiminden oluşmuştur. Her bir sinir hücresine birçok sinir hücresinden dendrit uçları ile girdiler gelmekte, bu girdiler sinir hücresinin soma diye isimlendirilen kısmına geçip içerde işlenmekte ve sinir hücresinin yapısına uygun bir çıktı değeri akson uçları ile diğer sinir hücrelerine aktarılmaktadır. Bu yapı zincirleme olarak sürmektedir. İnsanda öğrenme bu tarzda ki bir sinir ağı ile gerçekleştirildiğine göre bilim adamları bu yapının taklit edilmesiyle benzer bir öğrenmenin (bilgi depolamanın) bilgisayar ortamında gerçekleştirilebileceğini düşünmüşlerdir.



Şekil 2.1. Biyolojik Sinir Hücresinin Yapısı (vv.carleton.ca/~neil/neural/neurona.html)

Şekil 2.1 deki sinir hücresinin çıktısı gelen girdilerin büyüklüğüyle yakından ilgilidir. Kimi girdilerin şiddeti arttıkça sinir hücresinin çıktısı şiddetlenmekte, kimi girdiler için ise bu durumun tersi olmaktadır. Ayrıca her girdinin, sinir hücresinin çıktısına olan etkisi farklı farklıdır. Bunun yanında diğer sinir hücrelerinden gelen girdiler belli bir eşik değerini aşmadığı sürece girdiler sinir hücresinin soma bölümüne ulaşamamaktadır. Bunun nedeni akson-dendrit arasında bir boşluğun olması (Sinaps boşluğu) ve aradaki iletimin nörotransmitter maddelerle yapılması ve belli bir eşik değeri aşılmadan bu nörotransmitter maddelerin salgısının başlamamasıdır. Bu durum Şekil 2.2’ de gösterilmektedir.



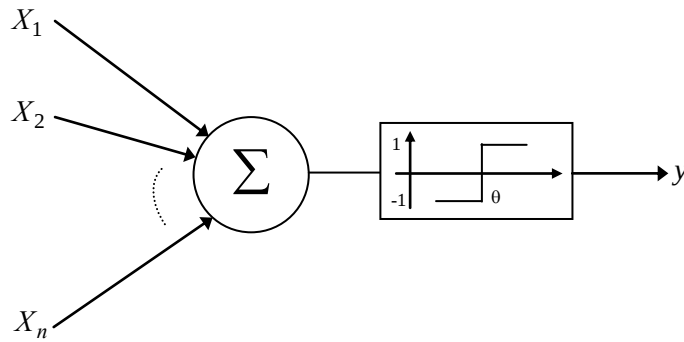
Şekil 2.2. Sinaptik Boşluğun Yapısı (vv.carleton.ca/~neil/neural/neuron-a.html)

Görüldüğü gibi basit bir sinir hücresi, açıklanan oldukça karmaşık işlemleri tek başına yapabilmektedir. Bilim adamlarını büyüleyen bu yapı yapay sinir ağlarının

temelini teşkil eden yapının iskeletini ortaya koymaktadır. Bu iskelet en genel haliyle şu şekilde açıklanabilir. Her bir girdinin sinir hücresinin çıktısına olan etkisi sinir hücreleri arasındaki bağlantılara verilen ağırlıklarla ölçülmektedir. Aşılması gereken eşik miktarı sinir hücresine gelen bir bağlantı (girdi) olarak tanımlanmaktadır, ancak bu bağlantının büyüklüğü sabittir sadece ağırlığı değiştirilmektedir. Sinir hücresi içinde girdilerin birleştirilmesi ağırlıklı toplam ile yapılmaktadır. Sinir hücresinin çıktısı ise ağırlıklı toplamla bulunan değer bir hareket fonksiyonuna girilmesiyle hesaplanmaktadır. Amaç öğrenme veya bilgi depolama olduğuna göre depolanacak bilgi veya öğrenilecek fonksiyon, sinir hücreleri arasındaki uygun ağırlıklara göre tanımlanmaktadır (Hagan vd, 1996). İşte burada yapay sinir ağlarının eğitilmesi yani uygulama konusuna bağlı olarak uygun ağırlık setinin bulunması gerekmektedir. Bu iş için de literatürde birçok algoritma bulunmaktadır. Burada kısaca adı geçen kavramlar daha ileriki bölümlerde ayrıntılı olarak açıklanacaktır.

2.1. Yapay Sinir Hücresi

İlk yapay sinir hücresi örneği McCulloch ve Pitts (1943) tarafından önerilmiştir. Bu model “Sınır Mantık Birimi (SMB) (Threshold Logic Unit)” olarak isimlendirilmektedir. Söz konusu sinir hücresi örneği Şekil 2.3 de gösterilmektedir.



Şekil 2.3 İlk Yapay Sinir Hücresi Örneği (Müler vd., 1990)

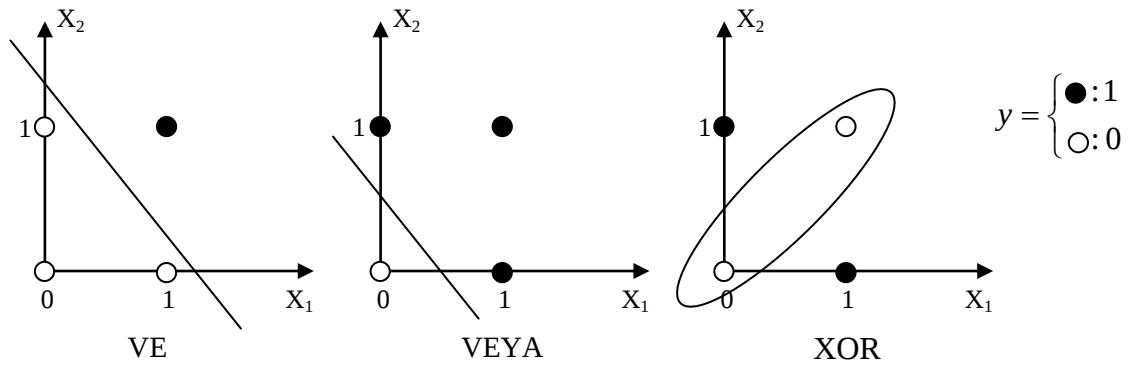
Şekil 2.3’de X_i ler girdi değişkenlerini göstermektedir. W_i ler ise sinaptik ağırlıkları göstermektedir (“Sinaptik” teriminin kullanılmasının sebebi biyolojik sinir hücreleriyle kurulan analogidir). Bunlara genellikle sadece “ağırlık” denilmektedir. SMB, sinir hücresine gelen girdi değişkenlerinin değerlerinin ağırlıklı ortalamasını aldıktan sonra bu toplamı doğrusal olmayan bir sınır operatörüne (adım fonksiyonu)

tabi tutar. Sinir hücresinin çıktısı bu sınır operatörüne göre belirlenir. McCulloch ve Pitts tarafından önerilen modelde çıktı değeri y , eğer ağırlıklı toplam θ değerinden büyükse 1, diğer tüm durumlarda sıfır değerini almaktadır. Bu anlatılan işlem kısaca aşağıdaki formülasyonla özetlenebilir:

$$y(X) = \begin{cases} 1 & , \sum_{i=1}^n W_i X_i \geq \theta \\ 0 & , \sum_{i=1}^n W_i X_i \leq \theta \end{cases} , X = [X_1, X_2, K, X_n]^T \quad (2.1)$$

Yukarıdaki formülden de anlaşılabileceği gibi SMB $\mathcal{R}^n \rightarrow \{0,1\}$ şeklinde fonksiyonlar tanımlamaktadır. Diğer bir ifadeyle n -boyutlu uzaydaki noktaları 2 farklı sınıfa ayırmaktadır. Bu nedenle sadece doğrusal olarak ayrılabilir fonksiyonlar, SMB ile elde edilebilir. Örneğin; VE ve VEYA operatörleri doğrusal olarak ayrılabilirken, XOR fonksiyonu doğrusal olarak ayıramamaktadır. Şekil 2.4 de açıkça görüldüğü gibi VE ve VEYA fonksiyonlarının çıktıları bir doğru ile ikiye bölünebilmektedir. Ancak XOR operatörünün çıktı uzayı ancak doğrusal olmayan bir şekilde ayrılabilir. Buradan çıkacak sonuç, VE ve VEYA fonksiyonları SMB ile elde edilebilirken XOR fonksiyonunun elde edilememesidir.

Bu bölümde incelenen yapay sinir hücreleri kullanılarak oluşturulan ilk yapay sinir ağı örneği Algılayıcı (Perceptron)'dır.



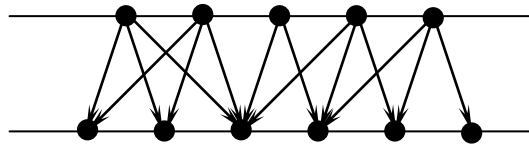
Şekil 2.4. VE ve VEYA fonksiyonlarının doğrusal olarak ayrılabilmesi ve XOR fonksiyonunun doğrusal olarak ayrılamaması.

2.2. Algılayıcı ve Çok Katmanlı Ağlara Geçiş

Bir önceki bölümde açıklanan yapay sinir hücresi örneğinden sonra genel mantığı aynı kalmak şartıyla, özellikle transfer fonksiyonunun değiştirilmesiyle, birçok yeni sinir hücresi ortaya konmuştur. Ancak bir önceki bölümdeki sinir hücresi örneği tarihteki ilk yapay sinir ağında kullanılan sinir hücrelerine temel oluşturması nedeniyle önemlidir. İlk yapay sinir ağı örneği Rosenblatt (1958) tarafından ortaya konulmuştur.

En basit şekliyle algılayıcı, iki sinir hücresi katmanından oluşmaktadır. Bu katmanlardan biri girdi sinir hücrelerini, diğeri ise çıktı sinir hücrelerini kapsamaktadır (Şekil 2.5). Çıktı katmanındaki sinir hücreleri, girdi katmanındaki sinir hücrelerinin çıktıları ile beslenmektedir. Ancak bunun tersi mümkün değildir, yani çıktı katmanındaki sinir hücrelerinden girdi katmanına bir geri besleme bulunmamaktadır (Müller vd., 1990). Ayrıca aynı katmandaki sinir hücreleri arasında da herhangi bir iletişim bulunmamaktadır. Bu özelliklerinden dolayı algılayıcı, ileri beslemeli (feed forward) sinir ağı olarak isimlendirilmektedir. Bu tez kapsamında incelenecek ve kullanılacak olan tüm ağlar, *İleri Beslemeli* olacaktır.

Şekil 2.5 de gösterilen bir sinir ağının istenen işlevi yerine getirebilmesi için sinir hücreleri arasındaki bağlantılara ait olan ağırlıkların uygun değerlerinin belirlenmesi gerekmektedir. Rosenblatt, belli girdi değerlerini istenen çıktılara dönüştürecek ağırlıkların belirlenmesi için iteratif bir yöntem önermiştir. Bu yöntemin ayrıntılarına burada girilmeyecektir. Daha ayrıntılı bilgi için (Rosenblatt, 1958) isimli referans incelenebilir.



Şekil 2.5. Algılayıcı (Müller vd., 1990)

Rosenblatt'ın bu çalışmasının ardından Minsky ve Seymour Papert (1969), Algılayıcı'nın XOR problemini çözemediğini göstermişlerdir. Bu durumun nedeni bir önceki bölümde açıklandığı üzere, problemin doğrusal olmamasıdır. Algılayıcı'nın bu zayıflığının ortaya konulması, konu üzerine olan ilgiyi uzunca bir

süre eksiltmiştir. Daha sonra Rumelhart (1986) yaptığı bir çalışma ile doğrusal olarak ayrılabilirlik şartı aramadan birçok gerçek hayat problemine uygulanabilen Çok Katmanlı Yapay Sinir Ağları için bir eğitim algoritması önermiştir. Çok Katmanlı Yapay Sinir Ağları, Algılayıcı'daki girdi ve çıktı katmanlarına bir veya birden fazla gizli sinir hücresi katmanının eklenmesiyle oluşturulur. Daha ayrıntılı bilgi için 2.5 no'lu bölüm incelenebilir.

2.3. Yapay Sinir Ağlarının Eğitilme Şekilleri

Yapay sinir ağlarında kullanılan başlıca iki tip öğrenme vardır. Bunlar, *danışmanlı* ve *danışmansız* öğrenmedir. Danışmanlı öğrenmede bir öğretmene ihtiyaç vardır. Öğretmen, bir hedef değer kümesi veya ağ sonuçlarının performansını derecelendiren bir gözlemci olabilir. Danışmanlı öğrenmede, danışman, sinirlere öğrenme işaretini gönderir ve bu işaret, sinirler arasındaki bağlantıların ağırlıklarını ayarlama kullanılır (Elmas, 2003). Örneğin; ağa

$$\{X_1, T_1\}, \{X_2, T_2\}, K, \{X_Q, T_Q\}$$

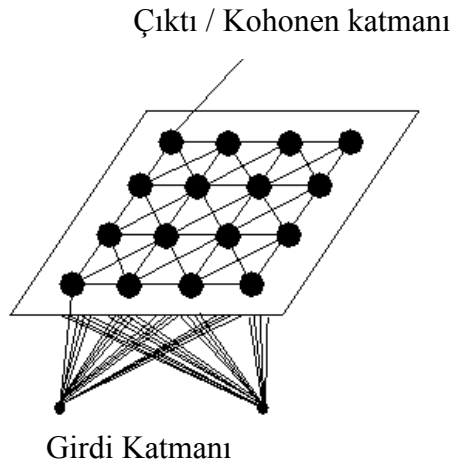
şeklinde bir eğitim kümesi sunulur. Burada X_Q ve T_Q sırasıyla ağın bir girdi değerini ve bu girdi için doğru olan (hedef) çıktı değerini göstermektedir. Girdi değerleri ağa girildikçe, ağın verdiği çıktılar ile hedef değerler karşılaştırılır ve aradaki farkı azaltacak şekilde ağın ağırlıkları güncellenir.

Danışmansız öğrenmede ise bir öğretmen bulunmamaktadır. Ağın ağırlıkları sadece ağa verilen girdi değerlerine göre güncellenir. İlk bakışta, hedef değerler bilinmeden böyle bir güncellenmenin yapılması pek mantıklı görünmeyebilir. Ancak, bu tip öğrenme algoritmaları bir tür kümeleme işlemi yaparlar. Girdi elemanlarını, sonlu sayıdaki sınıflara atamaya çalışırlar (Hagan vd., 1996). Danışmansız öğrenmeye, Hebbian öğrenme, Grossberg öğrenme, Kohonen'in özörgütlenmeli harita ağı örnek olarak verilebilir.

Bu tez kapsamında her iki öğrenme şeklini kullanan ağlar kullanılacaktır. Gözetimli öğrenme ile eğitilen Çok Katmanlı İleri Beslemeli Ağlar, bir fonksiyona yakınsamak için, gözetimsiz öğrenme ile eğitilen Kohonen'in Özörgütlenmeli Harita Ağı ise kümeleme yapmak amacı ile kullanılacaktır.

2.4. Kohonen'in Özörgütlenmeli Harita Ağı

Kohonen tarafından 1982 yılında ilk defa önerilen bu ağ memeli sinir sisteminden esinlenerek ortaya konulmuştur. Nöro-fizyolojik çalışmalar, gelişmiş hayvanların beyinlerinde farklı tiplerde haritaların olduğunu göstermektedir. Bu haritalar (beyin bölümü), beyin korteks bölümünde belli görevlere atanmış (konuşma, görme, duyma vb.) aktivite merkezleri olarak görev yapmaktadırlar. Kohonen'de birbirine yakın nöronların daha benzer girdilere tepki vereceği bir ağ tasarlamıştır (bkz. Şekil 2.6)



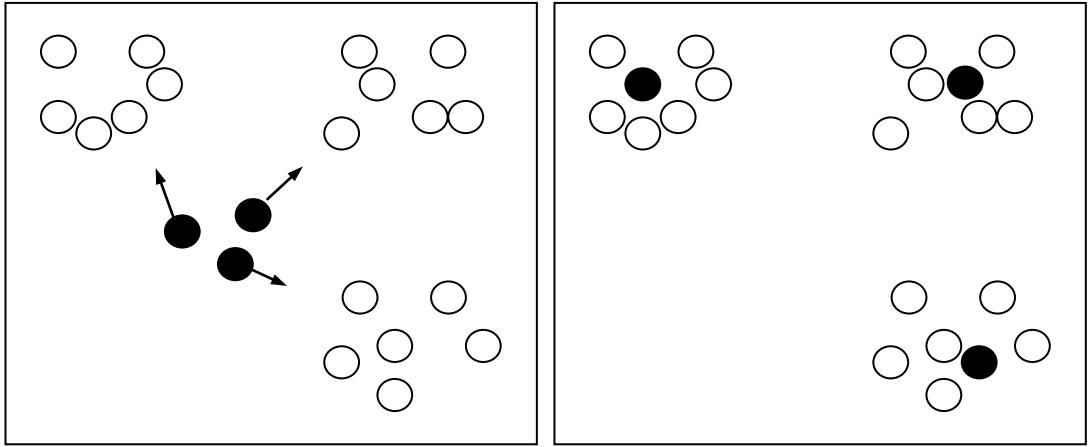
Şekil 2.6. Kohonen Özörgütlenmeli Harita Ağı

Bu ağın genel mimarisi, bir girdi katmanından ve düzlemsel olarak yerleştirilmiş sinir hücrelerinden oluşan bir çıktı katmanından (Kohonen katmanı) oluşmaktadır. Çıktı katmanındaki tüm sinir hücreleri yan yana birbirleriyle bağlantılıdır. Çıktı katmanındaki nöronlar, girdi kümesindeki topolojik yapı korunacak şekilde eğitilmektedirler. Böylece aynı topolojik yapı nöronlar arasında da bulunacaktır ve birbirine yakın nöronlar birbirine benzer girdi elemanlarına tepki verecektir (Nour vd., 1996).

2.4.1. Özörgütlenme Süreci

Özörgütlenme süreci, girdi katmanı ile Kohonen katmanı arasındaki ağırlıkların rasgele atanması ile başlar. Eğitim süreci boyunca girdi vektörleri arka arkaya ağa girilir. Her girdi elemanı ağa girildiğinde çıktı katmanındaki sinir hücreleri bu girdiye tepki göstermek için yarışır. Girdi elemanına en çok benzeyen sinir hücresi

(belli bir uzaklık tipine göre girdi vektörüne en yakın olan) kazanan sinir hücresi olur. Kazanan sinir hücresi ve onun belli komşuluğundaki sinir hücreleri girdi elemanına daha yakın olacak şekilde ayarlanırlar (hareket ederler) (Mangiameli vd., 1996). Bu süreç devam ettikçe Kohonen katmanındaki sinir hücreleri belli girdi kümelerine tepki verecek şekilde (onlara diğer sinir hücrelerinden daha yakın olacak şekilde) özelleşirler (bkz. Şekil 2.7). Böylece girdi vektörleri belli sayıda kümeye ayrılmış olur ve bu kümeler çıktı katmanındaki o küme için özelleşmiş sinir hücresi ile temsil edilirler.



Şekil 2.7. Çıktı katmanındaki sinir hücrelerinin, girdi vektörlerinin oluşturduğu kümelere doğru hareketi.

Özörgütlenmeli harita ağının eğitim algoritması aşağıdaki gibi özetlenebilir (Nour vd., 1996):

Adım 0: $t = 0$ al ve $W_i(0)$ rasgele ata.

Adım 1: Ağa bir X girdi vektörünü gir ve öklid uzaklığını kullanarak d_i değerlerini hesapla

$$d_i = d_i[X(t), W_i(t)] = \|X(t) - W_i(t)\|, \forall i \quad (2.2)$$

Adım 2: Girdi vektörüne en yakın sinir hücresi c 'yi bul

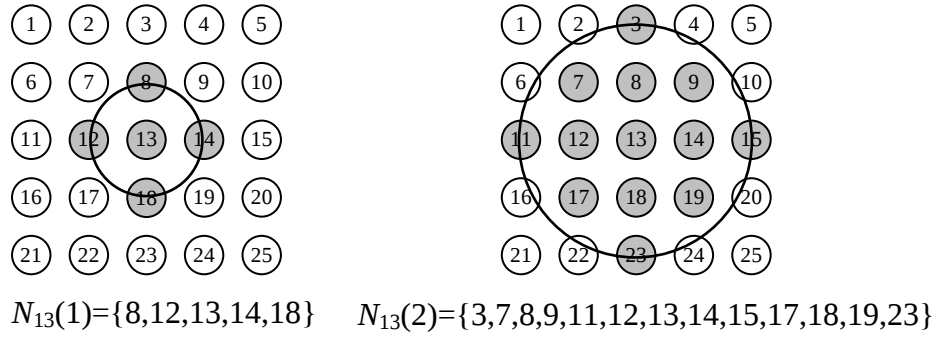
$$d_c = \min\{d_i\}, \forall i \quad (2.3)$$

Adım 3: Ağırlık vektörlerini aşağıdaki gibi güncelle

$$W_i(t+1) = \begin{cases} W_i(t) + \alpha(t)h(t,r)[X(t) - W_i(t)] & i \in N_c \\ W_i(t) & i \notin N_c \end{cases} \quad (2.4)$$

Adım 4: Eğer $t \geq T$ dur, değilse Adım 1'e git.

Bu algoritmada, t kesikli zaman adımını, T önceden tanımlanmış olabilecek en büyük yineleme sayısını, $\alpha(t) \in [0,1]$, zamanla monoton olarak azalan öğrenme hızını, $W_i(t)$, t zamanında çıktı katmanındaki i . sinir hücresi ile girdi katmanı arasındaki ağırlık kümesini, N_i , çıktı katmanındaki i . sinir hücresinin komşuluğundaki sinir hücrelerini içeren kümeyi, $h(t,r)$ ise zamana göre ve $r = \|r_i - r_c\|$ (r_i , çıktı katmanındaki i . sinir hücresinin koordinatını göstermektedir) uzaklığına göre azalan komşuluk fonksiyonunu göstermektedirler. Komşuluk büyüklüğünün daha iyi anlaşılması için Şekil 2.8 incelenebilir.



Şekil 2.8. Komşuluk büyüklükleri (Hagan vd., 1996)

2.4.2. Özörgütlemeli Harita Örneği

$(1,1,0,0)$, $(0,0,0,1)$, $(1,0,0,0)$ ve $(0,0,1,1)$ vektörleri girdi kümesini oluştursunlar. Bu girdileri özörgütlenmeli harita ağını kullanarak iki kümeye ayırmaya çalışalım (Elmas, 2003). $\alpha(1) = 0,6$, $\alpha(t+1) = 0,5\alpha(t)$ olsun ve komşuluk olmadığını varsayalım.

Adım 0: Ağırlıkların aşağıdaki gibi belirleyelim:

$$W(0) = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} = \begin{bmatrix} 0,2 & 0,8 \\ 0,6 & 0,4 \\ 0,5 & 0,7 \\ 0,9 & 0,3 \end{bmatrix}$$

Adım 1 ve 2: (1,1,0,0) vektörü ağı girilip uzaklıklar hesaplanır

$$d_1 = \sqrt{(0,2-1)^2 + (0,6-1)^2 + (0,5-0)^2 + (0,9-0)^2} = 1,36$$
$$d_2 = \sqrt{(0,8-1)^2 + (0,4-1)^2 + (0,7-0)^2 + (0,3-0)^2} = 0,98 \text{ (en az)}$$

Adım 3: Komşuluk olmadığından sadece 2 numaralı sinir hücresinin ağırlıkları güncellenir

$$w_{12}(\text{yeni}) = w_{12}(\text{eski}) + 0,6[x_1 - w_{12}(\text{eski})]$$
$$= 0,8 + 0,6[1 - 0,8]$$
$$= 0,92$$

Benzer şekilde diğer ağırlıklar da güncellenirse

$$W(1) = \begin{bmatrix} 0,2 & 0,92 \\ 0,6 & 0,76 \\ 0,5 & 0,28 \\ 0,9 & 0,12 \end{bmatrix} \text{ olur.}$$

Adım 1 ve 2: (0,0,0,1) vektörü ağı girilip uzaklıklar hesaplanır

$$d_1 = \sqrt{(0,2-0)^2 + (0,6-0)^2 + (0,5-0)^2 + (0,9-1)^2} = 0,66 \text{ (en az)}$$
$$d_2 = \sqrt{(0,92-0)^2 + (0,76-0)^2 + (0,28-0)^2 + (0,12-1)^2} = 2,2768$$

Adım 3: Komşuluk olmadığından sadece 1 numaralı sinir hücresinin ağırlıkları güncellenir

$$W(1) = \begin{bmatrix} 0,08 & 0,92 \\ 0,24 & 0,76 \\ 0,2 & 0,28 \\ 0,96 & 0,12 \end{bmatrix} \text{ olur.}$$

Adım 1 ve 2: (1,0,0,0) vektörü ağı girilip uzaklıklar hesaplanır

$$d_1 = \sqrt{(0,08-1)^2 + (0,24-0)^2 + (0,2-0)^2 + (0,96-0)^2} = 1,8656$$
$$d_2 = \sqrt{(0,92-1)^2 + (0,76-0)^2 + (0,28-0)^2 + (0,12-0)^2} = 0,6768 \text{ (en az)}$$

Adım 3: Komşuluk olmadığından sadece 2 numaralı sinir hücresinin ağırlıkları güncellenir

$$W(1) = \begin{bmatrix} 0,08 & 0,968 \\ 0,24 & 0,304 \\ 0,2 & 0,112 \\ 0,96 & 0,048 \end{bmatrix} \text{ olur.}$$

Adım 1 ve 2: (0,0,1,1) vektörü ağı girilip uzaklıklar hesaplanır

$$d_1 = \sqrt{(0,08-0)^2 + (0,24-0)^2 + (0,2-1)^2 + (0,96-1)^2} = 0,7056(\text{en az})$$

$$d_2 = \sqrt{(0,968-0)^2 + (0,304-0)^2 + (0,12-1)^2 + (0,048-1)^2} = 2,724$$

Adım 3: Komşuluk olmadığından sadece 1 numaralı sinir hücresinin ağırlıkları güncellenir

$$W(1) = \begin{bmatrix} 0,032 & 0,968 \\ 0,0964 & 0,304 \\ 0,680 & 0,112 \\ 0,984 & 0,048 \end{bmatrix} \text{ olur.}$$

Tüm girdi elemanları ağı girildiği için öğrenme oranının azaltılması gerekmektedir. $\alpha(2) = 0,5$ $\alpha(1) = 0,3$ alınır ve daha önceki işlemler yeni ağırlık kümesi ve öğrenme oranı için yenilenir. İşlemler bu şekilde yapılarak öğrenme oranı 100. adımda 0.01 değerine indirilir ve 100. işlem sonucunda aşağıdaki ağırlıklar elde edilir:

$$W(2) = \begin{bmatrix} 0,0053 & 0,99 \\ -0,17 & 0,3 \\ 0,7 & 0,2 \\ 1 & 0,0086 \end{bmatrix}, \dots, \quad W(10) = \begin{bmatrix} 1,5e-7 & 1 \\ 4,6e-7 & 0,37 \\ 0,63 & 5,4e-7 \\ 1 & 2,3e-7 \end{bmatrix}, \dots,$$

$$W(100) = \begin{bmatrix} 6,7e-17 & 1 \\ 4e-16 & 0,49 \\ 0,51 & 2,3e-16 \\ 1 & 1e-16 \end{bmatrix}$$

Sonuç olarak ağırlık matrisi $W = \begin{bmatrix} 0 & 1 \\ 0 & 0,5 \\ 0,5 & 0 \\ 1 & 0 \end{bmatrix}$ şeklinde olur.

Bu ağırlık kümesine göre girdi vektörleri ise aşağıdaki şekilde kümelenir:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 0 & 0,5 \\ 0,5 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1,5 \\ 1 & 0 \\ 0 & 1 \\ 1,5 & 0 \end{bmatrix}$$

Yani, 1. vektör 2. kümeye, 2. vektör 1. kümeye, 3. vektör 2. kümeye ve 4. vektör 1. kümeye yerleştirilmiştir. Vektörler incelendiğinde de 1. ve 3. vektörlerin ve 2. ve 4. vektörlerin birbirlerine oldukça benzedikleri açıkça görülebilmektedir.

2.4.3. Kohonen Özörgütlenmeli Harita Ağı'nın (KÖHA) Kullanımında Uygun Parametrelerin Belirlenmesi

Özörgütlenmeli haritalar, hali hazırda gelişmekte olan ve üzerinde çalışılan bir konudur. Bu nedenle birçok parametre deneme yanılma yoluyla en iyilenmeye çalışılmaktadır. Ancak literatürde bu konu ile ilgili kimi yaklaşımlar bulunmaktadır. Bu bölümde, bu yaklaşımlar hakkında kısa açıklamalar yapılacaktır.

2.4.3.1. Başlangıçta Rassal Ağırlık Atama

Diğer birçok yöntemde olduğu gibi KÖHA uygulamalarında da başlangıç ağırlıklarına bağlılık bulunmaktadır. Diğer bir ifade ile farklı başlangıç ağırlıkları için oluşturulan ağların performansları farklı farklı olabilmektedir (Hsu vd., 1992). Bu problem literatürde çok fazla dikkat çekmemiştir ve üzerine çok fazla çalışma yapılmamıştır. Önerilen birkaç yol da önemli yan etkilere sahiptir (Nour, 1996). Bu sorunun aşılması için farklı başlangıç ağırlık kümeleri ile aynı ağ defalarca eğitilip, en tatmin edici ağ kullanılmaya devam edilebilir.

2.4.3.2. Girdilerin Ağa Girilme Sırası

Hsu vd (1992), KÖHA'nın girdi elemanlarının ağa giriliş sırasının, ağın performansını çok önemli miktarda etkilediğini göstermişlerdir. Örneğin (Nour, 1996); $x \in \{9,10,11,12,23\}$ kümesinin elemanları iki kümeye ayırmak istenilsin ve $W_1 = 6$, $W_2 = 5$ olsun. Girdiler hem verilen şekilde hem de 23, 9, 11, 12 sırasıyla verilsin.

Ayrıca öğrenme oranı 0.5 olsun. Her iki sıralama için de kazanan sinir hücreleri Tablo 4.1’de görülebilmektedir.

Tablo 2.1 Girdi sırasının KÖHA çıktısına etkisi.

Adım	İlk Sıra					İkinci Sıra				
	X(t)	W_1	W_2	Kazanan	İşlem	X(t)	W_1	W_2	Kazanan	İşlem
0	-	6,00	5,00	-	-	-	6,00	5,00	-	-
1	9	7,50	5,00	W_1	$W_1 = W_1 + 0,5 * 3$	23	14,50	5,00	W_1	$w_1 = w_1 + 0,5 * 17$
2	10	8,75	5,00	W_1	$W_1 = W_1 + 0,5 * 2,5$	9	14,50	7,00	W_2	$w_2 = w_2 + 0,5 * 4$
3	11	9,88	5,00	W_1	$W_1 = W_1 + 0,5 * 2,25$	10	14,50	8,50	W_2	$w_2 = w_2 + 0,5 * 3$
4	12	10,94	5,00	W_1	$W_1 = W_1 + 0,5 * 2,12$	11	14,50	9,75	W_2	$w_2 = w_2 + 0,5 * 2,5$
5	23	16,97	5,00	W_1	$W_1 = W_1 + 0,5 * 12,06$	12	14,50	10,88	W_2	$w_2 = w_2 + 0,5 * 2,25$

Tablodaki veriler incelendiğinde farklı sıralar için iki farklı kümeleme sonucu ortaya çıkmıştır. İlk sıra için tüm elemanlar aynı kümeye yerleştirilmiş olmasına rağmen, ikinci sırada bir eleman farklı bir kümeye yerleştirilmiştir.

Yukarıdaki örnekten de anlaşılacağı üzere girdi sırasının harita ağına olan etkisi çok büyüktür. Ancak problemin çözümü üzerinde çalışılan uygulamaya çok bağımlı olduğundan, literatürde tavsiye edilen çözüm pek bulunmamaktadır. Ancak, girdilerin rasgele ağı girilmesi, riski bir miktar azaltan bir çözüm yolu olabilir.

2.4.3.3. Komşuluk Fonksiyonu

Komşuluk fonksiyonu Kohonen tarafından icat edilmiştir. Amacı ise hiçbir zaman kazanamayan, yani bir girdi elemanına tepki veremeyen sinir hücreleri problemini engellemektir (Ahalt vd, 1990).

KÖHA ile birlikte kullanılan komşuluk fonksiyonunun çan eğrisi şeklinde olduğuna daha önce değinilmişti. Bu yaklaşımdan başka literatürde farklı uygulamalar bulunmaktadır. Ancak bu tezin kapsamı gereği daha fazla ayrıntıya girilmeyecektir.

Bu konudaki çalışmaların özlü bir taraması için Nour vd. (1996)'nin çalışması incelenebilir.

Bu çalışma kapsamında belirlenmesi gereken parametrelerden birisi de komşuluk büyüklüğüdür. Bu değer için Lu vd. 1991 yılında, çıktı katmanındaki sinir hücresi sayısına bağlı olan aşağıdaki formülü önermişlerdir.

$$|N_c| = \left\lceil \sqrt{1,2 \cdot N} \right\rceil \quad (N, \text{ çıktı katmanındaki sinir hücresi sayısı.}) \quad (2.5)$$

2.4.3.4. Durma Kriteri

Eğitim sürecinin ne zaman sonlandırılacağına karar veren parametredir. Bu sonlandırma, elbette yakınsama gerçekleştiğinde yapılmalıdır. Bunun içinde durma kriterinin bu yakınsamayı garanti edecek şekilde saptanması gerekmektedir. Ancak literatürde bu tarz, yakınsamayı garanti eden kesin bir öneri bulunmamaktadır. Bunun sebebi ise enküçüklenecek açık bir fonksiyonun bulunmamasıdır. Genel olarak uygulamada aşağıdaki iki yaklaşım sıklıkla kullanılmaktadır:

1. En küçük hata karesi
2. En büyük yineleme sayısı

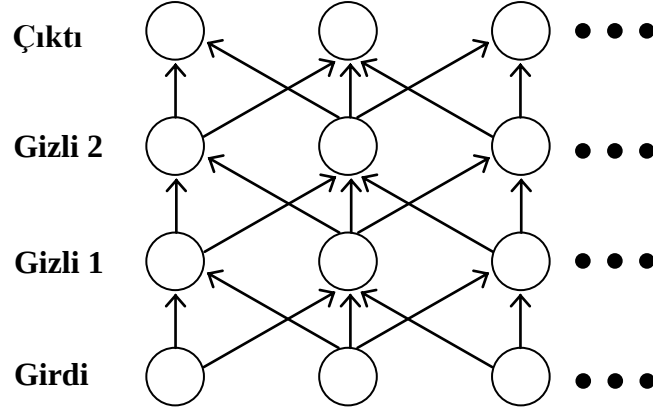
Bu yaklaşımlardan ilki kümeleme problemleri için tutarlı değildir. Çünkü daha ufak en küçük hata karesi değeri, daha ufak kümeleme hatası anlamına gelmemektedir (Nour vd., 1996). İkinci yaklaşım için ise Lu vd. (1991) komşuluk büyüklüğüne bağlı aşağıdaki formülü önermişlerdir.

$$T = 5|N_c| \ln|N_c| \quad (N_c, \text{ komşuluk büyüklüğüdür.}) \quad (2.6)$$

2.5. Çok Katmanlı Ağlar (ÇKA)

İlk yapay sinir ağı örneği olan Algılayıcı'nın bulunmasından 12 yıl sonra, Minsky ve Papert (1969), Algılayıcı'nın yetersizliğini görmüşler ve XOR problemini çözemediğini ispatlamışlardır. Bunun için iki katmanlı ileri beslemeli ağların kullanılabileceğini ileri sürmüşler ve tek katmanlı ağlardaki birçok sınırlamayı ortada kaldırdığını göstermişlerdir (Elmas, 2003). Fakat gizli katmanların ağırlıklarının

nasıl değiştirileceği konusunda bir yöntem önerememişlerdir. Bu soruna Rumelhart v.d. (1986) geri yayılım yöntemi ile bir çözüm getirmişlerdir. Bu tarihten sonra İleri Beslemeli Çok Katmanlı Ağlar oldukça yaygın şekilde kullanılmaya ve geliştirilmeye başlanmıştır.



Şekil 2.9. İleri Beslemeli Çok Katmanlı Ağ Mimarisi

İleri Beslemeli Çok Katmanlı Ağların mimarisi, (Masters, 1993) iki veya daha fazla katmana bölünmüş düğüm (biyolojik sinir hücresi terimi yapay sinir ağları konusunda düğüme karşılık gelmektedir) kümelerinden oluşmaktadırlar (bkz. Şekil 2.9). Her biri en az bir düğüm içeren, bir girdi ve bir de çıktı katmanı vardır. Girdi katmanındaki düğümler herhangi bir işlem yapmadan ağın girdisini aynen çıktı olarak verirler. Girdi ve çıktı katmanlarının arasında “Gizli Katman” lar bulunmaktadır. “İleri Beslemeli” terimi bilgi akışının sadece tek yönde gerçekleştiğini belirtmektedir. Her katmanda bulunan düğümlerin girdileri bir önceki katmandaki düğümlerin çıktılarından gelmektedir. Dolayısıyla ağ içindeki veri akışı girdi katmanından çıktı katmanına doğru, geri besleme olmaksızın devam etmektedir.

Her katmandaki düğümlerde genellikle aynı hareket fonksiyonları kullanılır. Bu hareket fonksiyonları ağın performansını ve karakterini büyük ölçüde belirlerler. Bu nedenle herhangi bir uygulamada kullanılacak hareket fonksiyonunun seçimi önemli bir adımdır.

2.5.1. Hareket Fonksiyonları

İleri Beslemeli Çok Katmanlı Ağlar’ın eğitilme süreci, hareket fonksiyonlarının sınırlı ve türevlenebilir olmasını gerekli kılmaktadır (Elmas, 2003). Bu gereksinimi

karşılayan ve yapay sinir ağı yazılımlarında çoğunlukla kullanılan iki farklı hareket fonksiyonu vardır. Bunlar:

Lojistik Fonksiyon (Logistics Function)

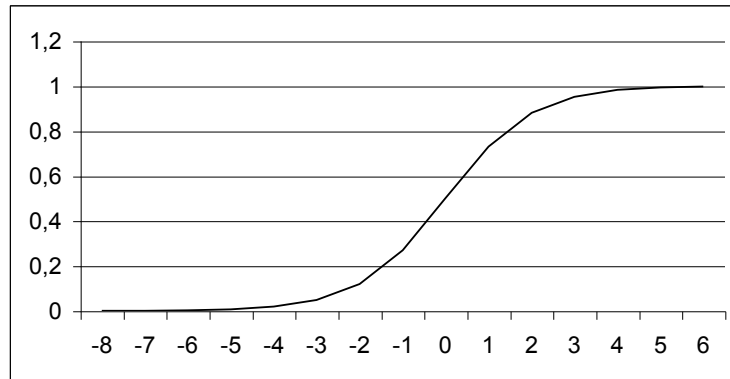
Hiperbolik Tanjant Fonksiyonu

Birçok uygulamada, bu hareket fonksiyonlarının, ağıın sonuç performansında çok etkili olmadıkları ancak eğitim sürecince harcanan süreyi büyük oranda etkiledikleri ortaya çıkmıştır (Masters, 1993). Ancak hangi hareket fonksiyonunun diğerine göre daha üstün olduğuna dair literatürde kesin bir kanıya rastlanmamıştır.

2.5.1.1. Lojistik Fonksiyonu

Bu fonksiyon R 'de tanımlıdır ve aldığı değerler $(0,1)$ aralığıyla sınırlıdır (bkz. Şekil 2.10). Türevi, fonksiyonun kendisinin cinsinden ifade edilebilmektedir. Matematiksel ifadesi ve türevi (2.7)'de gösterilmiştir.

$$f(x) = \frac{1}{1 + e^{-x}} , \quad f'(x) = f(x)(1 - f(x)) \quad (2.7)$$

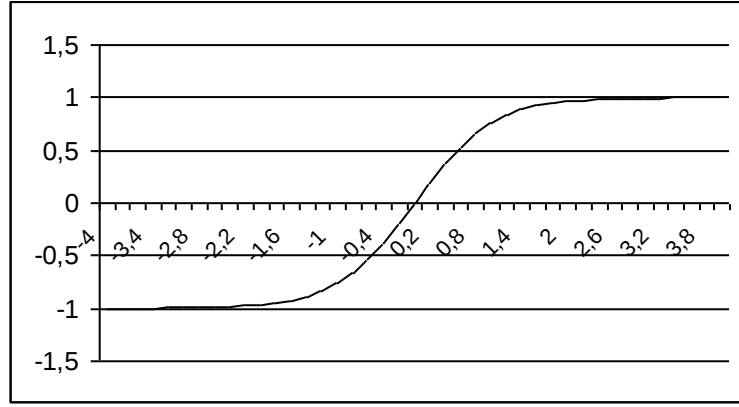


Şekil 2.10. Lojistik Fonksiyonunun Grafiği

2.5.1.2. Hiperbolik Tanjant Fonksiyonu

Bu fonksiyonda R 'de tanımlıdır ve aldığı değerler $(-1,1)$ aralığıyla sınırlıdır. Şekli lojistik fonksiyonuna oldukça benzer (bkz. Şekil 2.11). Bu fonksiyonun türevi de kendi cinsinden ifade edilebilir. Kendisinin ve türevinin matematiksel ifadesi (2.8)'de gösterilmektedir.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} , \quad f'(x) = 1 - f^2(x) \quad (2.8)$$



Şekil 2.11. Hiperbolik Tanjant Fonksiyonunu Grafiği

2.5.2. İleri Beslemeli Çok Katmanlı Ağların Uygulamasında Dikkat Edilecek Hususlar

2.5.2.1. Mimarinin Belirlenmesi

Uygulamada kaç gizli katman kullanılacağı ve bu katmanlarda kaç düğümün bulunacağı önemli bir sorundur. Bu sorulara kesin bir cevap vermek veya tam bir formülasyonunu oluşturmak şu an için pek mümkün görünmemektedir. Ancak bu durum hiçbir kuralın bulunmadığı anlamına gelmez.

Öncelikle yapılan birçok çalışma tek gizli katmanın, içindeki düğüm sayısını değiştirerek hemen hemen bütün problemlerin çözümünde yeterli olduğunu göstermektedir (Hagan, 1996). Hatta Masters (1993) iki veya daha fazla gizli katman kullanmak için herhangi bir teorik neden bulunmadığını söylemektedirler. Dolayısıyla uygulamalarda sadece tek bir gizli katman olacağı varsayılabilir. Bu gizli katmandaki düğüm sayısı da olabilecek en küçük sayıda tutulmalıdır. Gizli katmandaki düğüm sayısının arttırılması simülasyon sırasında hem hafıza hem de merkezi işlem biriminin (MİB, İngilizcesi CPU)'nun yükünü arttırmaktadır. Fakat öğrenme işlemi daha hassas yapılabilmektedir. Bunların yanında düğüm sayısı arttırılırken ağın genelleme yapma yeteneği azalabilmektedir. Ağın genelleme yapma yeteneği, eğitim işlemi tamamlandıktan sonra daha önce denenmemiş verilere ağı doğru yanıtları verme becerisi olarak tanımlanabilir. Eğer düğüm sayısı gerekenden fazla alınırsa, eğitim esnasında ağ eğitim setini çok fazla öğrenir (overfit) (Masters, 1993). Bu durumda ağ eğitim esnasında çok iyi performans göstermesine rağmen yeni verilere uygulandığında doğruluğunu yitirebilir. Bu sakıncaların ortadan

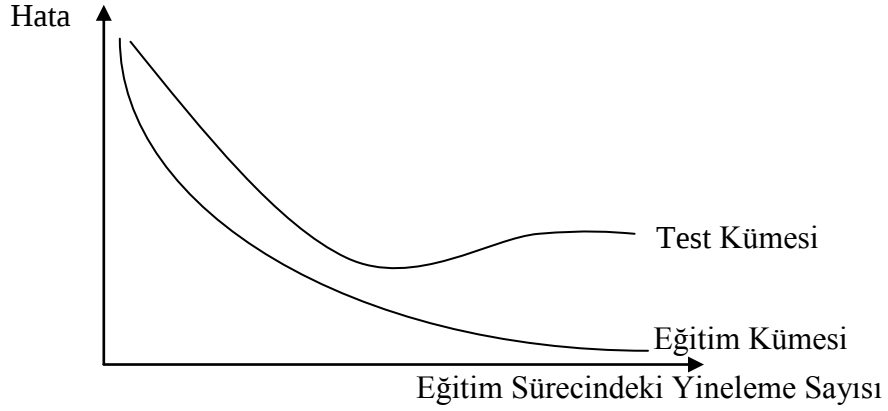
kaldırılması için en azdan başlayarak düğüm sayısı yavaş yavaş artırılır ve her adımda ağıın hata miktarı kontrol edilir. Hata miktarı kabul edilebilir seviyeye geldiğinde düğüm sayısını artırma işlemi durdurulur. Bu şekilde hem ağ iyi bir şekilde eğitilmeye çalışılır, hem de aşırı öğrenme engellenmiş olunur.

2.5.2.2. Ağın Eğitilmesi

Yapay sinir ağları eğitilirken genellikle iki farklı küme ile çalışılır. Bunlardan ilki “eğitim kümesi”, diğeri ise “test kümesi” dir. Test kümesi, ağa hiç uygulanmamış örnekleri içermektedir ve eğitimin ardından ağın genelleme yapma yeteneğini ölçmek için kullanılır. Eğitim esnasında ağın her iki küme için sergilediğı performans çok önemlidir. Ancak test kümesindeki performansı daha önemlidir çünkü, ağın kullanım aşamasında göstereceğı performansı (daha önce hiç denenmemiş verilerle) büyük ölçüde yansıtacaktır. Uygulamalarda genellikle ağın eğitim miktarı arttıkça eğitim kümesindeki performansın arttığı, ancak test kümesindeki performansın azaldığı gözlenmektedir. Buradan da eğitim miktarının belli bir noktada sınırlandırılması gerektiğı gibi bir sonuç çıkmaktadır. Bu durum Şekil.2.12’ de görölmektedir.

Bu şekil incelendiğinde ağın eğitilmesinin belli bir noktada kesilmesi gerektiğı gibi bir kanı oluşmaktadır. Ancak Masters (1993), eğitimdeki yineleme sayısının artmasıyla beraber genelleme yeteneğinin kötüleşmesi sonucunda, eğitimin belli bir noktada (test kümesindeki performans azalmaya başladığında) kesmenin iki sakıncasını ortaya koymuştur:

- a) Bu müdahale, nedenden çok belirti üzerinde odaklanmaktır. Yapay sinir ağlarının eğitim kümeleri oluşturulurken dikkat edilmesi gerekli olan en önemli noktalardan biri oluşturulan eğitim kümesinin tüm popölasyonu temsil edebilmesidir. Eğitim sürecinde ilerledikçe test kümesinde performans kötüleşmesi, test kümesinin popölasyonu temsil etmediğini göstermektedir. Dolayısıyla burada yapılması gereken, eğer mümkünse, eğitim kümesinin genişletilmesidir.
- b) Eğitimi belli bir noktada durdurmanın alternatif maliyeti çok büyük olabilir. Şekil 2.12 incelendiğinde, eğitim sürecinin ilerleyen bölümlerinde test



Şekil 2.12. Eğitim miktarının genelleme yeteneğine etkisi (Masters, 1993)

kümesinin hata miktarı artmaya başlamaktadır. Bu nedenle eğitim işleminin hep iyileşen yönde (genelleme yeteneği açısından) gideceğini varsayamayız. Ancak bu hata artışı ileride daha büyük performans artışlarının olmayacağı anlamına gelmemektedir. Yani, test kümesinin performansı kötüleştikten sonra daha ileriki adımlarda bu performans tekrar iyileşebilir. Dolayısıyla, ilk artmaya başladığı anda eğitimi sonlandırırız, bu andaki ağırlık kümesinin en iyi ağırlık kümesi olduğunu varsayamayız. Diğer bir ifade ile genelleme yeteneğinde belli bir miktar kötüleşmeden sonra çok daha iyi bir performans düzeyi yakalanabilir. Eğer eğitim erken kesilirse, daha iyi performans elde etme olasılığı yitirilebilir.

Sonuç olarak, eğitim ilerledikçe, eğer genelleme yeteneğinde kötüleşme olursa bu durumda eğitim seti genişletilmelidir. Bunun için en uygun yol ise test kümesinin eğitim kümesinin içine dâhil edilmesi ve yeni bir test kümesinin oluşturulmasıdır

2.5.2.3. Ağ Çıktısındaki Küsuratlar

Hareket fonksiyonu olarak kullanılan lojistik ve hiperbolik tanjant fonksiyonları en büyük ve en küçük değerlerini artı ve eksi sonsuzlarda alırlar. Bu sebepten özellikle kategorik değişkenlerde 0.9 ve daha büyük değerler 1 olarak kabul edilmelidir. Benzer şekilde lojistik fonksiyonunda sıfıra yakın olan değerler sıfır olarak ve hiperbolik tanjant fonksiyonunda -1'e yakın olan değerler -1 olarak alınmalıdır.

Bunların yanında ağın çıktılarında daima ufak bir hatanın olduğu akılda çıkarılmamalıdır. Unutulmamalıdır ki bu hata miktarı ağın genelleme yeteneğinin korunması açısından önemlidir ve istenen bir durumdur.

2.5.3. Çok Katmanlı Ağların Eğitilmesi

En genel anlamıyla *öğrenme*, tanımlanmış bir performans kriterini iyileştirmek için ağın ağırlıklarının aşama aşama değiştirildiği bir adaptasyon sürecidir (Tettamanzi, 2001). Yapay sinir ağlarının tipine göre öğrenme iki şekilde olmaktadır: gözetmenli ve gözetmensiz öğrenme. Çok katmanlı ağlar gözetmenli tipteki eğitim algoritmaları ile eğitilirler. Gözetmenli öğrenme de belli büyüklükte bilinen girdi-çıkı ikilerini kapsayan bir *eğitim seti* bulunmaktadır. Öğrenme sürecinde kullanılan performans ölçütü, belli bir girdi değeri için bilinen çıktı değeri ile ağın verdiği çıktı değeri arasındaki farkı temel alır. Her adımda bu performans ölçütü küçültülmeye çalışılır.

Bu bölümde gözetmenli öğrenmenin iki alt kümesi daha incelenecektir. Bu tür öğrenme iki şekilde yapılabilir: Kitle Öğrenme ve Çevrim içi Öğrenme' dir.

- Kitle tipi öğrenmede ağırlıklar, girdi kümesindeki tüm elemanlar ağa girildikten sonra güncellenir. Diğer bir ifadeyle, tam bir *devir* (girdi kümesindeki tüm elemanların ağa girilmesi) tamamlandıktan sonra toplam hata miktarı hesaplanır ve bu hata miktarını azaltıcı yönde ağırlıklar güncellenir.
- Çevrim içi öğrenmede ise her bir girdi elemanı ağa girildikten sonra hata miktarı hesaplanır ve bu hatayı azaltıcı yönde ağırlıklar güncellenir. Kimi kaynaklarda bu kavram Sürekli Öğrenme olarak da geçmektedir.

Kitle tipi ve çevrim içi tipi öğrenme yöntemlerinin hangisinin daha iyi olduğu konusunda literatürde kesin bir kanı bulunmamaktadır. Ancak her ikisinin de belli pozitif ve negatif yönleri bulunmaktadır. Kitle tipi öğrenme daha fazla bilgisayar hafızasına ihtiyaç duyması bakımından eleştirilmektedir. Ancak kitle tipi öğrenmenin daha doğru sonuçlar verdiği ileri sürülmektedir. Bunun sebebi, tam bir devir tamamlandıktan sonra ağırlıkların güncellenmesidir. Böylece hata tüm eğitim seti üzerinde azaltılmaktadır. Çevrim içi tip öğrenmede ise ağırlıklar her girdi elemanına göre güncellendiğinden bütün eğitim seti düşünüldüğünde toplam hata miktarını Kitle tipe göre daha az azaltabilmekte ve hatta arttırabilmektedir. Ancak kimi

arařtırmalar, evrim ii eęitimde ortalama aęırlık deęiřiminin doęru eęimi (yani tm eęitim seti iin aęırlıklar uzayında hatayı azaltma yn) takip ettięini gstermektedir (Wilson vd., 2000). Ayrıca Housson (1995) her iki eęitim tipinin benzer hızlara sahip olduęunu gstermektedir. Bu aıdan bakılınca iki yntem arasında ok nemli bir farklılık olmadığı hatta kitle tipi ęrenmenin biraz daha tercih edilebilir olduęu sylenbilir. Ancak 2000 yılında Wilson vd. tarafından yapılan arařtırma eęitim setinin byk olduęu durumlarda Kitle tipi eęitimin ok da pratik olmadığını ortaya koymuřtur. Kitle tipi ęrenmede daha kk ęrenme oranı kullanılması zorunluluęundan dolayı eęer eęitim seti bykse eęitim sreci evrim ii eęitime gre daha uzun srmektedir (Wilson vd., 2000). Aynı alıřmada evrim ii eęitim srecinde ortalama aęırlık deęiřimlerinin doęru iniři (gradient) takip ettięi bir kez daha belirtilmiřtir. Ancak kesin bir kanıya sahip olabilmek iin daha uzun sreli deneyime ihtiya olduęu da belirtilmektedir. Bu bilgilerin ıřığında kk eęitim setlerinde Kitle tipi eęitim ve byk eęitim setlerinde ise evrim ii Eęitim tavsiye edilebilir. Bu noktada bir soru daha akla gelebilir. Kk ve byk eęitim setlerinden kast edilen nedir? Wilson ve Martinez (2000) alıřmalarında 20000 elemana sahip bir eęitim setini byk olarak kabul etmiřlerdir. Dolayısıyla 15000 ve daha ok eleman ieren eęitim setleri byk olarak kabul edilebilir. Ancak, bu yargının tam bir bilimsel ispatı bulunmamaktadır.

Yapar Sinir Aęları'nın eęitilmesinde en nemli nokta kullanılacak en iyileme yntemidir. Yani belirlenen performans kriterini en iyilemek iin kullanılacak yntemler aęların eęitimi iin ok nemlidir. Yapay Sinir Aęları'nın pratik problemlerde kullanılabilmesi iin uygun eęitim algoritmalarının geliřtirilmesi ve bu yntemlerin pratikte kullanılabilmesi gerekmektedir. Yani kabul edilebilir hız ve kapasite kullanımıyla aęların eęitilebilmesini saęlayan algoritmalara ihtiya vardır. Bu amala Algılayıcı'dan bu yana geliřtirilen her aę iin uygun bir eęitim algoritması geliřtirilmiř ve gnmzde de geliřtirilmeye devam edilmektedir.

Bu blmde anlatılacak olan algoritmalar gzetmenli eęitim tipi iin geliřtirilmiř yntemlerdir. Gzetmenli tipteki eęitimde ama bir hata fonksiyonunun en kklenmesidir. Tanımlanan hata fonksiyonu, eęitim kmesinde her girdi elemanına karřılık gelen hedef deęerler ile aynı girdi deęerleri iin aęın bulduęu ıktı

değerleri arasındaki farkı temel alır. Gözetmenli eğitimde kullanılan algoritmalar bu hata fonksiyonunu azaltmaya çalışır.

2.5.3.1. Algılayıcı Öğrenme Algoritması

Bu algoritma Şekil 2.3 de şekli verilen sinir hücresiden oluşan iki katmanlı ağın eğitilmesi için geliştirilmiştir. Bu sinir hücresinde ikili çıktı söz konusudur ve doğrusal olmayan bir sınır değeri kontrolü ile sağlanmıştır. Algoritmanın adımları kısaca şu şekildedir (Tettamanzi, 2001):

- (a) Ağırlıkların ve sınır değerlerinin başlangıç atamalarını rasgele yap.
- (b) Sinir hücresine bir girdi vektörünü gir.
- (c) Sinir hücresinin çıktısı hesapla.
- (d) Sinir hücresinin hatasını hesapla ve ağırlıkları aşağıdaki formüle göre güncelle:
$$W_i(t+1) = W_i(t) + \alpha(T - y)x_i$$
, burada T, istenen çıktı değerini, y ise ağın çıktı değerini göstermektedir. α ise 0 ile 1 arasında “öğrenme oranı” parametresidir.
- (e) Hata miktarı belli bir değerin altına düşene kadar veya belli sayıda iterasyon için 2. adıma git.

Bu algoritma hakkında daha ayrıntılı bilgi için (Rosenbaltt, 1986) ve (Gurney, 1997) referansları incelenebilir.

Rosenblatt, bu algoritmanın doğrusal olarak ayrılabilen girdi kümeleri için kullanılabileceğini ispatlamıştır. Ancak eğer girdi kümesi doğrusal olarak ayrılabilir değil ise veya ağ gizli katmanlar içeriyorsa bu durumda bu algoritma yetersiz kalmaktadır. Bu sorunların aşılabilmesi için eğim tabanlı algoritmalar daha sonraki senelerde geliştirilmiştir.

2.5.3.2. Geri Yayılımlı Algoritma

Eğim tabanlı algoritmaların ilkidir. Toplam hata fonksiyonunun birinci türevine göre ağırlıklarını günceller. Algoritmanın adı çıktı sinir hücrelerinde meydana gelen hata miktarının en son katmandan önceki katmanlara kademeli olarak dağıtılmasından gelmektedir. En basit haliyle istenen çıktı miktarı ile ağırlık bulduğu çıktı değeri arasındaki fark en son katmandaki çıktı sinir hücrelerinde ortaya çıkmaktadır. Ancak bu hatanın ortaya çıkmasında ağırdaki tüm sinir hücrelerinin katkısı vardır. Dolayısıyla toplam hata miktarının ağırlık tüm sinir hücrelerine dağıtılması, normal çalışma yönünün tersine gerçekleşmektedir. Genel anlamda bu mantığı kullanan algoritmanın adımları aşağıdaki gibi özetlenebilir. Aşağıda hareket fonksiyonu *lojistik* olarak seçilen sinir hücrelerinden oluşmuş bir ağırlık için algoritmanın adımları verilmiştir (Tettamanzi, 2001):

- (a) Ağırlıkların başlangıç atamasını rasgele yap.
- (b) Bir girdi vektörünü ağa gir.
- (c) Ağırlık çıktısını hesapla.
- (d) Tüm çıktı sinir hücreleri için $\delta_j = (y_j - T_j)y_j(1 - y_j)$, değerlerini hesapla.
Burada T_j j. sinir hücresi için istenen çıktı değeridir, y_j ise j. sinir hücresi için ağırlık çıktı değeridir.
- (e) Tüm diğer sinir hücreleri için (en son gizli katmandan ilk katmana kadar)
 $\delta_j = \sum_k W_{jk} \delta_k$ değerlerini hesapla. Burada δ_k bir sonraki katmanın δ_i değeridir.
- (f) Ağırlıkları yandaki formülle güncelle. $W_{ij}(t+1) = W_{ij}(t) - \alpha y_i \delta_j$, burada α öğrenme oranıdır.
- (g) Hata miktarı belli bir değerin altına inene kadar veya belli sayıda yineleme yapıldıkça kadar adım 2'ye git.

Eğimli iniş algoritmasının en temel parametresi öğrenme oranıdır. Öğrenme oranı, herhangi bir adımdaki eğim değerinin eksilisinin kaç katının mevcut ağırlık

parametrelerine ekleneceği, diğer bir deyişle toplam hata fonksiyonunu azaltan yönde ne kadar büyüklükte bir adım atılacağı sorusunun yanıtıdır. Bu nedenle bu parametrenin büyüklüğü oldukça önemlidir. Eğer çok büyük seçilirse algoritma kararsız bir yapıya bürünür. Tersine çok küçük seçilirse bu durumda da algoritmanın yakınsaması çok zaman alabilir.

Standart Geri Yayılmalı algoritmasına yapılan ilk eklemelerden biri momentum katsayısıdır. Standart Geri Yayılmalı algoritmasında her adımda ağırlıklar

$$\Delta W_{ij} = \alpha \delta_j y_i \quad (2.9)$$

formülüne göre değiştirilir. Hata fonksiyonu üzerinde yeterince kaymış ve belli bir hız kazanılmışsa, yerel minimumlar sistemin sadece hızının azalmasına neden olabilir. Bu temele dayandırılarak oluşturulan momentum terimi, bir sabit olup, ağırlık vektörlerinin değişimi ile çarpılır ve böylece bir önceki ağırlık değişim yönü yardımcı ile bir sonraki ağırlık değişim yönü saptanır (Elmas, 2003). Böylece her adımda ağırlıkların güncellenmesi

$$\Delta W_{ij}(t+1) = \alpha \delta_j y_i + \eta \Delta W_{ij}(t) \quad (2.10)$$

formülüne göre yapılır. Burada η , momentum katsayısını göstermektedir. Hesaplamalara momentum teriminin ilave edilmesi ağ performansı üzerinde etkili olduğu gözlenmiştir. Momentum teriminin hesaplamaya katılması adım sayısında ve toplam ağ hatasında bir düşüş meydana getirmektedir. Momentum katsayısı yüksek alındığında ağdaki toplam hatanın sıfıra doğru daha fazla bir eğimle yaklaştığı görülmektedir. Bu parametrenin en uygun değerinin belirlenmesinde kullanılan bir formülasyon mevcut değildir.

2.5.3.3. Momentumlu Geri Yayılmalı Algoritma ile Eğitim Örneği

Bu bölümde momentumlu geri yayılmalı algoritma uygulaması olarak XOR probleminin çözümü verilmiştir (Öztemel, 2003):

1. Adım: Örneklerin Toplanması:

XOR problemi için 4 örnek vardır. Bunlar 1 ve 0 değerlerinden oluşmaktadır. Her örnek için girdiler ve beklenen çıktı değerleri Tablo 2.2' deki gibi olur.

2. Adım: Ağın Topolojik Yapısının Belirlenmesi

XOR probleminde 2 girdi ve 1 de çıktı olduğundan, oluşturulacak olan Çok Katmanlı Ağ'ın da 2 girdi düğümü ve 1 çıktı düğümü olacaktır. 1 gizli katman ve 2 tane gizli kaman düğümünün bu problemi çözeceği varsayılmaktadır. Şekil 2.14' da

Tablo 2.2 Girdiler ve beklenen değerler

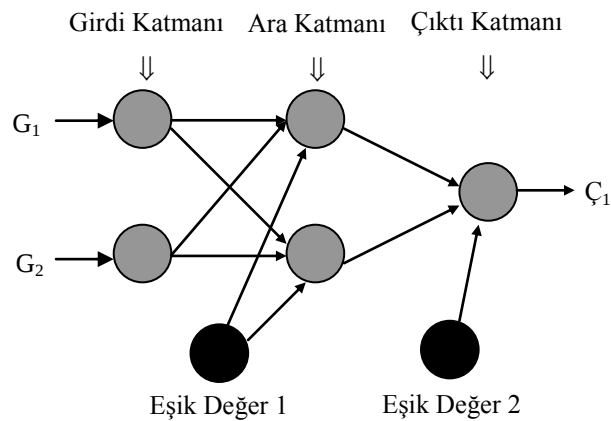
	Girdi 1	Girdi 2	Çıktı
Örnek 1	0	0	0
Örnek 2	0	1	1
Örnek 3	1	0	1
Örnek 4	1	1	0

oluşturulan ağın mimarisi gösterilmektedir. Görüldüğü gibi gizli katman için 1 adet, çıktı katmanını içinde 1 adet eşik değer düğümü bulunmaktadır.

3. Adım: Öğrenme Parametrelerinin Belirlenmesi

Oluşturulan ağ için hareket fonksiyonu olarak *lojistik* fonksiyonunun kullanıldığını, öğrenme oranı (α) ve momentum (η) parametrelerinin ise aşağıdaki gibi belirlendiği varsayılın,

$$\alpha = 0.5, \eta = 0.8$$



Şekil 2.13. XOR problemi için tasarlanmış İleri Beslemeli Çok Katmanlı Ağ

4.Adım: Ağırlıkların Başlangıç Değerlerinin Atanması

Oluşturulan ağ için ağırlık vektörleri ve başlangıç değerleri rassal olarak aşağıdaki gibi belirlenmiş olsun. Girdi Katmanı ile ara katman arasındaki ağırlıklar A^i matrisi ile gösterilsin;

$$A^i = \begin{bmatrix} 0.129952 & 0.570345 \\ -0.923123 & 0.328932 \end{bmatrix}$$

Çıktı katmanı ile ara katman arasındaki ağırlıklar ise A^a ile gösterilsin;

$$A^a = [0.164732 \quad 0.752621]$$

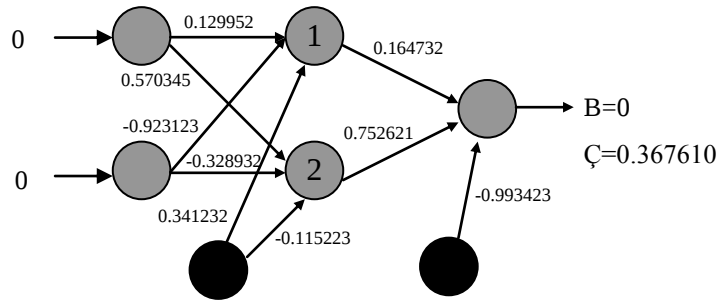
Eşik değeri ağırlıkları rassal olarak aşağıdaki gibi atansın:

$$\beta^a = [0.341332 \quad -0.115223]$$

$$\beta^c = [-0.993423]$$

5.Adım: Örneklerin Ağa Gösterilmesi ve İleri Doğru Hesaplama:

Birinci örnek $G_1=0$, $G_2=0$ ve $B=0$ olarak belirlenmiştir. Bu örnek ağa girilirse, ileri doğru hesaplama Şekil 2.15’de gösterildiği gibi olacaktır.



Şekil 2.14. İleri doğru hesaplama

Ara katman düğümlerinin net girdileri (eşik düğümlerinin ağırlık değerleri eklemiş olarak) şu şekilde hesaplanır.

$$Net1 = (0 * 0.129952) + (0 * -0.923123) + (1 * 0.341332) = 0.341332$$

$$Net2 = (0 * 0.570345) + (0 * -0.328932) + (1 * -0.115223) = -0.115223$$

Ara katman düğümlerinin çıktıları ise şöyle hesaplanır.

$$C_1 = \frac{1}{1 + e^{-0.341232}} = 0.584490$$

$$C_2 = \frac{1}{1 + e^{0.115223}} = 0.471226$$

Çıktı katmanındaki düğümlerin net girdisi hesaplanırsa;

$$Net = (1 * -0.993423) + (0.584490 * 0.164732) + (0.471226 * 0.752621) = -0.542484$$

değeri bulunur. Bu değer ile ağın çıktısı;

$$C = \frac{1}{1 + e^{0.542484}} = 0.367610 \text{ değerini alır.}$$

Beklenen çıktı 0 olduğuna göre ağın hatası: $E = 0 - 0.367610 = -0.367610$ olur. Bu hatanın geriye doğru yayılması sonucu ara katman ile çıktı katmanı arasındaki ağırlıkların değişim miktarları aşağıdaki şekilde hesaplanır.

$$\begin{aligned} \delta_1 &= C_1(1 - C_1) \cdot E_1 \\ \delta_1 &= 0.367610 * (1 - 0.367610) * (-0.367610) \\ \delta_1 &= -0.085459 \end{aligned}$$

$$\begin{aligned} \Delta A_{11}^a(t) &= 0.5 * -0.085459 * 0.584490 + 0.8 * 0 = -0.024875 \\ \Delta A_{21}^a(t) &= -0.020135 \\ \Delta \beta_1^C(t) &= -0.042730 \end{aligned}$$

Ağırlıklardaki bu değişim miktarları ile ara katman ve çıktı katmanı arasındaki ağırlıklar yeniden hesaplanabilir.

$$\begin{aligned} A_{11}^a(t) &= A_{11}^a(t-1) + \Delta A_{11}^a(t) \\ A_{11}^a(t) &= 0.164732 - 0.024975 = 0.139757 \\ A_{21}^a(t) &= 0.752621 - 0.020135 = 0.732486 \\ \beta_1^C(t) &= -0.993423 - 0.042730 = -1.036153 \end{aligned}$$

Benzer şekilde, girdi katmanı ile ara katman arasındaki ağırlıkların değişim miktarları ve yeni ağırlık değerleri hesaplanır. Ara kamandaki hata oranları ve değişim miktarları aşağıdaki şekilde bulunur.

$$\delta_1^i(t) = C_1(1 - C_1)\delta_1 A_{11}^i(t-1)$$

$$\delta_1^i(t) = 0.584490 * (1 - 0.584490) * (0.164732) * (-0.085459)$$

$$\delta_1^i(t) = -0.034190$$

$$\delta_2^i(t) = -0.160263$$

$$\Delta A_{11}^i(t) = 0.5 * -0.034190 * 0 + 0.8 * 0 = 0$$

$$\Delta A_{12}^i(t) = 0.5 * -0.034190 * 0 + 0.8 * 0 = 0$$

$$\Delta A_{21}^i(t) = 0$$

$$\Delta A_{22}^i(t) = 0$$

$$\Delta \beta_1^i(t) = 0.5 * 1 * -0.034190 = -0.017095$$

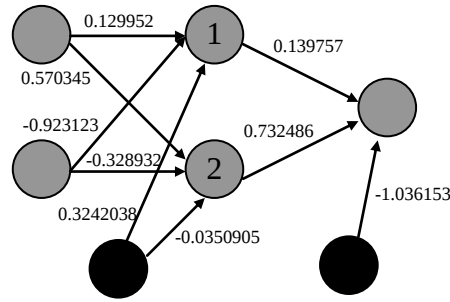
$$\Delta \beta_2^i(t) = 0.5 * 1 * -0.160263 = -0.080132$$

Bu değerler kullanılarak ağırlıklar değiştirilir. Ağırlıklardaki değişim miktarı 0 olduğundan ağırlık değerlerinde herhangi bir değişiklik olmayacak ancak eşik değeri ağırlıklarında değişiklik olacaktır.

$$\beta_1^i(t) = 0.341232 - 0.017095 = 0.3242038$$

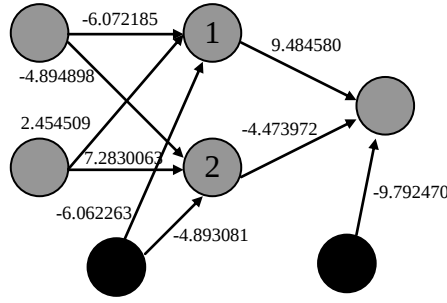
$$\beta_2^i(t) = -0.115223 + 0.081325 = -0.0350905$$

Şekil 2.16'de yeni belirlenen ağırlık değerleri gösterilmektedir.



Şekil 2.15. XOR ağıının ağırlıklar değiştirildikten sonraki durumu

Birinci yineleme bittikten sonra ikinci yineleme başlayacaktır. Bu kez ikinci örnek ağa girilir. $G_1=0$, $G_2=1$ ve $B=1$ olacaktır. Yukarıdaki işlemler aynı şekilde tekrar edilir. Bu yinelemeler bütün çıktılar doğru cevap verinceye kadar devam etmelidir. Şekil 2.17 ağ öğrendikten sonraki ağırlık değerlerini göstermektedir.



Şekil 2.16. XOR problemi öğrenildikten sonraki ağırlıklar

Tablo 2.3 XOR problemini öğrendikten sonra ağıın ürettiği çözümler ve hata oranları

Girdi 1	Girdi 2	Beklenen Çıktı	Ağıın Çıktısı	Hata
0	0	0	0.017622	-0.017
1	0	1	0.981504	0.018
0	1	1	0.981491	0.018
1	1	0	0.022782	-0.020

2.5.3.4. Değişken Öğrenme Oranı Sezgiseli

Literatürde klasik Geri Yayılmalı Algoritma’da kullanılan öğrenme oranının en iyi değerini veren bir formülasyon bulunmamaktadır. Bunun nedenleri, en uygun değer, veri setine çok bağlı olması ve öğrenme süreci içinde sürekli değişmesidir. Bu nedenle öğrenme oranı her uygulamada, kullanıcıdan istenmektedir. Bu durum pratikte birçok soruna ve eleştiriye neden olmaktadır. Değişken Öğrenme Oranı sezgiseli de bu eleştirilere cevap vermek amacıyla ortaya konulmuş bir yöntemdir.

Bu yöntemle daha önce tüm öğrenme süreci boyunca sabit tutulan öğrenme oranı, süreç içinde değiştirilmektedir. Diğer bir ifade ile algoritmayı kararlı durumda tutmaya yetecek en büyük öğrenme oranı her adımda seçilmektedir.

Herhangi bir adımda hata miktarı bir önceki adımdaki hatayı önceden belirlenmiş bir orandan daha fazla aşarsa bu durumda bulunan ağırlık ve eşik değerleri iptal edilmekte ve öğrenme oranı 0 ile 1 arasındaki bir oranla çarpılarak azaltılmaktadır (MatLab yardım dokümanı, 2004). Aksi durumda ağırlıklar ve eşik değerleri korunmaktadır. Eğer yeni hata miktarı eski hata miktarından daha az ise bu durumda öğrenme oranı belli bir oranda arttırılmaktadır.

Bu açıklamaların ışığında şu yorumlar yapılabilir. Eğer artış miktarı algoritmayı kararsız bir yapıya sürüklemiyorsa, prosedür hata miktarının artmasına belli bir oranda izin vermektedir. Eğer daha büyük bir öğrenme oranı algoritmayı daha kararlı yapacaksa bu durumda öğrenme oranı arttırılmaktadır.

Genel olarak bu mantığı kullanan farklı yaklaşımlar mevcuttur. Daha ayrıntılı bilgi için (Jacobs, 1988), (Tollenaere, 1990) ve (Fahlman, 1989) referansları incelenebilir.

Bu yöntemle daha yüksek yakınsama hızları elde edilebilmektedir. Ancak tabii ki bu sezgiselin de zayıf yanları vardır. Bunlardan ilki klasik Geri Yayılmalı Algoritma'da sadece tek bir parametre (öğrenme oranı) isteniyorken, bu yaklaşımda üç ve hatta kimi yaklaşımlarda beş veya altı tane parametre (öğrenme oranı arttırma katsayısı, öğrenme oranı arttırma katsayısı, öğrenme oranı azaltma eşik değeri,...) girilmesi istenebilmektedir. Bu parametreler de algoritmanın performansı üzerinde çok etkilidirler ve üzerinde çalışılan probleme göre değişmektedirler. Diğer bir dezavantajı ise eğimli iniş yöntemini kullanan klasik Geri Yayılmalı Algoritma'nın yakınsadığı kimi problemlerde yakınsamayabilmesidir.

2.5.3.5. RPROP Algoritması (Resilient Propagation)

Bu algoritma Martin Riedmiller ve Heinrich Braun tarafından 1993 yılında geliştirilmiştir. Diğer adaptasyonlu algoritmalar gibi ağıın eğitilmesi sürecinde her adımda, öğrenme oranının, daha hızlı bir öğrenme için, revize edilmesi mantığı üzerine kurulmuştur. Diğer adaptasyonlu yöntemlerden en büyük farkı hata fonksiyonunun büyüklüğünü değil işaretini dikkate almasıdır. Riedmiller vd., (1993) çalışmalarında standart Geri-Yayılmalı Algoritmayı eleştirmişlerdir. Uygun bir öğrenme oranının bulunmasının ve momentum büyüklüğünün en iyi değerinin belirlenmesinin güçlüğünden ve bu sorunun uygulamalarda ortaya çıkardığı problemlerden bahsetmişlerdir. Bunun yanında diğer adaptasyonlu öğrenme algoritmalarının en büyük eksikliğini de büyük çoğunluğunun, direkt olarak öğrenme

oranına (α) odaklanmaları, kısmi türevin $\left(\frac{\partial E}{\partial W_{ij}} \right)$, ağırlıklardaki değişime (ΔW_{ij})

olan etkisini dikkate almamaları olduğunu ileri sürmektedirler. Burada dikkate alınması gereken nokta, kullanılan adaptasyonlu öğrenme yöntemi ne kadar gelişmiş olursa olsun eğer kısmi türev dikkate alınmıyorsa, probleme bağlı olarak kısmi

türevin değerinde gerçekleşebilecek aşırı salınımların öğrenme sırasında performans düşüşüne neden olabilmesidir. Bu problemi ortadan kaldırmak için kısmi türevin sadece işaretini dikkate alan (böylece büyüklükte ki değişimleri ortadan kaldıran) RPROP algoritmasını önermişlerdir. Kullandıkları öğrenme kuralı aşağıdaki şekildedir.

$$\Delta_{ij}^{(t)} = \begin{cases} \lambda^+ * \Delta_{ij}^{(t-1)} , & \frac{\partial E}{\partial W_{ij}}^{(t-1)} * \frac{\partial E}{\partial W_{ij}}^{(t)} > 0 \\ \lambda^- * \Delta_{ij}^{(t-1)} , & \frac{\partial E}{\partial W_{ij}}^{(t-1)} * \frac{\partial E}{\partial W_{ij}}^{(t)} < 0 , \quad 0 < \lambda^- < 1 < \lambda^+ \\ \Delta_{ij}^{(t-1)} , & \text{diğer durumlarda} \end{cases} \quad (2.11)$$

Burada eğer ilgili ağırlık (W_{ij}) işaret değiştirmişse bu durumda bir yerel en iyi noktası atlanmış demektir ve ağırlık değişim değeri (ΔW_{ij}), λ^- değeri ile küçültülmektedir. Eğer türevin işareti değişmemişse bu durumda yakınsamayı kolaylaştırmak için ağırlık değişim değeri arttırılmaktadır. Ağırlık değişim değerlerinin bulunmasından sonra ağırlıkların güncellenmesi aşağıdaki basit kurala göre olmaktadır.

$$\Delta W_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} , & \frac{\partial E}{\partial W_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} , & \frac{\partial E}{\partial W_{ij}}^{(t)} < 0 \\ 0 , & \text{diğer durumlarda} \end{cases}$$

$$\Delta W_{ij}^{(t)} = -\Delta W_{ij}^{(t-1)} , \quad \frac{\partial E}{\partial W_{ij}}^{(t-1)} * \frac{\partial E}{\partial W_{ij}}^{(t)} < 0 \quad \text{ise} \quad (2.12)$$

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \Delta W_{ij}^{(t)}$$

Burada eğer türev pozitifse (artan hata) ağırlık belirlenmiş değişim miktarı kadar azaltılmaktadır ve eğer türev negatifse ağırlık belirlenmiş değişim miktarı kadar arttırılmaktadır. Eğer kısmi türev işaret değiştirmişse bir önceki ağırlık değişimi tersine çevrilmektedir. Tüm bu işlemler yığın eğitimi prosedürü takip edilerek

yapılmaktadır. Daha ayrıntılı bilgi için (Riedmiller vd., 1993) incelenebilir. Bu yöntem uygulamada standart Geri-Yayılım algoritmasına SuperSAB algoritmasına ve Quickprop algoritmasına göre daha hızlı yakınsama sağlamış ve parametreler iyileştirildikçe yöntemin performansı (tutarlılık) daha da artmıştır.

2.5.3.6. Ölçeklendirilmiş Çekimli Eğim (ÖÇE) Algoritması (Scaled Conjugate Gradient Algorithm)

Bu algoritma Çekimli Eğim Algoritmasını kullanan öğrenme yöntemlerinden biridir (Moller, 1990). Bu yöntemin geliştirilmesindeki itici güç “Eğimli iniş” algoritmasını kullanan yöntemlerin zayıf yönleri ile Çekimli Eğim algoritmasını kullanan yöntemlerin (ÇEG-Çekimli Eğim Geri Yayılmalı Algoritması (Johansson vd., 1990) , BFGS-Broyden Fletcher Goldfarb Shanno (Battiti vd., 1990)) zayıf yönlerini gidermektir. “Eğimli iniş” algoritmasını kullanan yöntemlerin en büyük problemi düşük yakınsama hızları ve büyük çaplı problemlerde yaşadıkları ciddi performans problemleridir. Bu problemlerin en büyük nedeni kullanıcı tarafından belirlenmesi gereken öğrenme oranı ve momentum sabiti gibi parametrelere ihtiyaç duymasıdır. Bu yöntem ile kullanıcıdan herhangi bir parametre istenmemektedir. Çekimli Eğim algoritmasını kullanan yöntemlerde ise yaşanan en büyük sorun her öğrenme iterasyonunda doğrusal arama (line search) yapmalarıdır. Çünkü doğrusal arama tekniği çok fazla miktarda hesaplama ihtiyacı duymaktadır ve bu da yöntemleri daha karmaşık bir hale getirmektedir. ÖÇE yönteminde doğrusal arama yapılmamaktadır ve adım büyüklüğünün (step size) belirlenmesinde Levenberg-Marquard yaklaşımını kullanmaktadırlar.

Çekimli Eğim yöntemini kullanan teknikler hata fonksiyonunun ikinci türevini de dikkate aldıklarından Hessian matrisinin pozitif tanımlı (positive definite) olması önemlidir. Eğer Hessian pozitif tanımlı değilse bu sorunu daha önceki teknikler (ÇEG, BFGS) doğrusal arama yaparak aşmışlardır. Ancak ÖÇE tekniğinde bunun yerine Levenberg-Marquard yaklaşımı kullanılmış ve Lagrange çarpanı olarak bilinen λ_k sayısı tanımlanmıştır. Bu çarpan Hessian matrisinin pozitif tanımlı olup olmadığı durumları düzenlemek için kullanılmaktadır. Yöntem ile ilgili daha ayrıntılı bilgi için (Moller, 1990) ve (Fletcher, 1975) incelenebilir.

ÖÇE yöntemi sadece birincil bilgiyi kullanan (Taylor açılımının ilk iki terimi) klasik Geri-Yayılmalı Algoritmaya göre ve ikincil bilgiyi kullanan (Taylor açılımının ilk üç terimi) ÇEG ve BFGS algoritmalarına göre çok daha yüksek bir yakınsama hızına sahiptir. Ayrıca karmaşık problemlerde Geri Yayılmalı Algoritma'nın yakınsamakta problem yaşadığı fonksiyonlara çok daha kolayca yakınsayabilmektedir.

3. SİGORTA VE PRİM HESABI

3.1. Giriş

İnsan, yaşamı boyunca sürekli belirsizliklerle karşılaşır ve bunları aşmaya çalışır. Yatırım yaparız ancak geri dönüşüm oranını tam olarak bilemeyiz, yolculuk yaparız ancak kaza yapma olasılığı sebebiyle gidilecek yere varamama olasılığımız vardır. Bunlar ve bunlar gibi birçok örnek verilebilir. Beklenmedik veya kontrol edilemeyen olayların neden olduğu belirsizliği aşmak için insanoğlu sürekli bir çaba içindedir. İstatistiklerden yararlanarak, yatırım alternatifleri değerlendirilir, belli zamanlarda araçlarımızı kontrol ettiririz, her ihtimale karşı bekçi ile işyerlerini hırsızlığa karşı koruruz. Yapılan bu çalışmalar, kesinlikle kaza olacak veya kesinlikle hırsızlık olacak diye yapılmaz. Böyle bir olasılık olduğu için yapılır. Benzer şekilde insan yaşamı da birçok risklerle doludur. Geleceğimizi sağlıklı bir şekilde geçirmemiz tamamıyla belirsizdir. Bu belirsizliği aşmak içinde insanoğlu sigorta denen kurumları icat etmiştir. Hem ölüm riskine karşı hem de kaza riskine karşı insan hayatı sigorta edilebilmektedir. Böylece zarara uğradığımızda bu zarara bizimle birlikte ortak olan kurumlarla anlaşma yaparız. Bugün ödediğimiz primlerle geleceğimizdeki riskleri azaltmaya çalışırız.

Sigortanın tarihi, çok eski zamanlara kadar dayanmaktadır. Tarihteki ilk sigorta benzeri uygulamalar, M.Ö. 4500 yıllarında Mısır'da görülmektedir. Bazı esnaf grupları, bir sandık meydana getirerek burada para biriktirmekte ve içlerinden ölenlerin ailelerine bu sandıktan para yardımı yapmaktaydılar. Ancak sigorta kavramının gelişmesi ve ilk sigorta poliçesinin kesilmesi 1347 yılını bulmuştur. Bu tarihte İtalya'nın Cenova limanından hareket eden "Santa Clara" adlı geminin taşıdığı yük sigortalıdır. İlerleyen yıllarda sigortacılığı düzenleyen kanunlar birer birer çıkmıştır ve çok değişik alanlarda sigorta yapılmaya başlanmıştır (hayat, sağlık, yangın v.b.) (Acinan, 1998).

Bu tez kapsamında özellikle sađlık sigortaları incelenecek ve yine bu alanda bir uygulama yapılacaktır. Ancak öncelikle sigortanın genel bir tanımının verilmesi yerinde olacaktır.

Sigorta, literatürde çok deđişik şekillerde tanımlanmıştır (Robert vd., 1966, Robert vd., 1963, Denenberg, 1964). Ancak Akmut (1980), çalışmasında verdiği tanımla sigortayı tam olarak içine alan bir çerçeve sunmuştur:

“Sigorta, aynı riskle karşı karşıya bulunan çok sayıda kişinin, bireysel olarak belirsiz olan hasar olasılıđını belirgin duruma getirmek ve bu olasılıđın gerçekleşmesiyle ortaya çıkan zararları birlikte karşılamak amacıyla, risk yönetim ve sorumluluđunu taşıyan bir kiři ya da kurum tarafından bir araya getirilmesidir.”

Bu tanımda öncelikle sigortanın ekonomik işlevi üzerinde durulmaktadır. Bu işlev toplum genelinde riskin azaltılması ya da belirginleştirilmesidir. Bunun için aynı veya benzer tip riskle karşı karşıya olan kişilerin riskleri bir havuzda birleştirilmektedir. Böylece istatistikten (büyük sayılar kanunu) yararlanılarak bu topluluđun riski belirgin duruma getirilmektedir. Tanımda değinilen diđer önemli bir nokta sigorta işlemindeki taraflardır. Bunlardan biri aynı tip riski paylaşan topluluk diđer taraf ise sigortacıdır. Sigortacının yasal statüsü, her ülkede devletin koyduđu kanunlarla belirlenmektedir. Sigortacı, topluluktan topladıđı primleri yatırıma dönüştürür ve gelir elde eder. Hasar olduđu zaman da bu hasarı vaat ettiđi sınırlar içinde kalmak koşulu ile karşılar.

3.2. Sigortanın Fonksiyonları

Sigortanın fonksiyonlarını 3 ana başlıkta toplamak mümkündür (Dinsdale, 1963):

- 1) Ana Fonksiyonu: Tolum içindeki tek bir bireyin finansal kayıpları, birçok kiři üzerine dağıtılmakta ve böylece bu kayıpların oluşumlarına ilişkin belirsizlik azaltılmaktadır. Birçok poliçe sahibi aynı havuzun içine primlerini ödemektedir ve kayba uğrayan kişinin hasarı (claim) bu havuzdan karşılanmaktadır. Burada önemli olan nokta daha fazla (daha az) riske sahip olan kişiler havuza daha fazla (daha az) prim yatırmaktadırlar.

2) İkincil Fonksiyonu:

- (a) Özellikle firma sahipleri, çok farklı risklere karşı korunma amacı ile belli bir miktar parayı riske atmadan (yatırım yapmadan) kenara koymak zorunda kalmazlar. Çünkü ödedikleri (daha az miktardaki) sigorta primleri ile varlıklarını güvence altına alırlar. Böylelikle kendi anaparalarını istedikleri yatırımı yapmak amacı ile kullanabilirler.
- (b) Sigorta, günümüzün büyük ölçekli ticari organizasyonlarını mümkün kılmıştır. Denizler aşırı yapılan ticaret, yangın ve kaza korkusu olmadan sigorta vasıtasıyla mümkün olmuştur. Böylelikle insanlar, daha güvenli bir ortamda daha büyük hacimli yatırımlar yapma olanağı bulmuşlardır.
- (c) Sigorta firmaları, sigortalama yaptıkları alanlarla ilgili kayıpların azaltılmasına dair oldukça kapsamlı bir bilgi hazinesine ve tecrübeye sahip olurlar. Bu şekilde müşterilerine kayıpların, yangınların, hırsızlıkların v.b. gibi birçok alanda önerilerde bulunurlar. Bunun yanında daha az hasara sahip olan firmaları ödüllendirerek, firmaların kayıpları azaltma konusunda daha titiz davranmalarını sağlamada teşvik edici rolünü üstlenirler.

3) Dolaylı Fonksiyonları

- (a) Sigorta firmaları tarafından toplanan büyük miktardaki para yatırım olarak ülke kaynaklarını değerlendirmede kullanılır. Böylelikle sigorta firmaları hükümete ve endüstriye dolaylı olarak yardımcı olurlar.
- (b) Uluslararası seviyede çalışan sigorta firmaları, ülkenin görünmeyen ihracatına önemli derecede katkıda bulunurlar.

3.3. Sigortacılıkta Riskin Tahmini ve Prim Hesabı

Sigortacılıkta prim hesabı, bütünüyle sigortalanan kişinin taşıdığı riskin tahmini ile ilgilidir. Kişinin taşıdığı risk, sigorta firması tarafından değerlendirilir ve riskin para değeri, sigortalıdan prim olarak alınır. Kişinin taşıdığı risk ise kişi hakkındaki temel veriler (yaş, cinsiyet, vb.) dikkate alınarak tahmin edilmeye çalışılır. Yapılan tahminlere güvenerek sigorta işi ile uğraşan firmaların hayatta kalabilecekleri ve kar edebilecekleri, olasılık teorisinin en önemli yasalarından biri olan “Güçlü Büyük Sayılar Kanunu” ile ortaya konulabilir.

3.3.1. Güçlü Büyük Sayılar Kanunu

Olasılık teorisinin en temel sonuçlarından birisi Güçlü Büyük Sayılar Kanunu’dur. Olasılık hakkındaki sezgisel fikirleri haklı çıkarması açısından ve birçok uygulama alanına sahip olması açısından çok önemlidir (Jacod vd., 2003). Bu teori, birbirinin benzeri olayların tek başlarına sahip oldukları belirsizliği, bir araya getirildiklerinde kaybetmeye başladıklarını göstermektedir. Diğer bir ifadeyle olayların ortalamasının kesinliğe yakın bir derecede tahmin edebileceğini göstermektedir. Matematiksel ifadesi aşağıdaki gibidir (Ross, 2002):

$Y_1, Y_2, \dots, Y_n$ bağımsız, aynı dağılıma sahip rassal değişkenler olsunlar. Ayrıca tüm Y_i değişkenleri sonlu $\mu = E[Y_i]$ beklenen değerlerine sahip olsun. Bu takdirde

$$P\left\{\lim_{n \rightarrow \infty} \frac{Y_1 + Y_2 + \dots + Y_n}{n} = \mu\right\} = 1 \text{ olur.}$$

Diğer bir ifadeyle aynı dağılıma sahip çok sayıda değişkenin aldıkları değerlerin ortalama değeri değişkenlerden sadece bir tanesinin beklenen değerine hemen hemen kesinlikle eşittir. Yani değişkenler bir araya getirildikçe belirsizlik azalmaktadır.

Güçlü büyük sayılar kanunu, sigortacılığın temelini oluşturmaktadır. Bu teorem sayesinde, tek bir olayın çıktısı tahmin etmenin imkânsızlığı, yerini, çok sayıda olay düşünülerek tahmin yapabilme yeteneğine bırakmıştır (Mehr vd., 1952). Bu kanunun sigortaya uygulanması sayesinde, her yıl kaç evin yanacağı, kaç ölümün gerçekleşeceği bilinebilmektedir. Benzer şekilde eğer Y_i ler i . kişinin önümüzdeki yıl

belli bir teminattan neden olacağı hasar miktarını gösteriyorsa, eğer kişi sayısı çoksa bu popülasyondaki ortalama hasar miktarı bir kişinin neden olacağı hasar miktarının beklenen değeri kadar olacaktır. Daha genel bir ifade ile toplam hasar, kişi sayısı ile bir kişinin beklenen hasarının çarpımına eşittir. Tabii ki kişilerin hasarlarının birbirlerinden bağımsız olduğu ve aynı ya da benzer dağılımlara sahip oldukları varsayılmaktadır.

Güçlü Büyük Sayılar Kanunu incelenerek günlük sigorta uygulamaları hakkında kimi yorumları yapmak mümkündür. Eğer sigortalı sayısı çok olursa sigorta firması her senenin hasar miktarlarını önceden tahmin ederek bu tahmine göre fiyatlandırma yapabilir. Günlük hayatta sigorta firmaları genel olarak hesaplanan tahminden daha fazla bir miktarı müşterilerinden toplamaktadırlar. Bunun ilk nedeni büyük sayılar kanunu kullanılarak riskin tam olarak yok edilememesidir. Bunun nedeni ise dünyada hiçbir zaman sonsuz sayıda müşterinin sigortalanamamasıdır. Bunlardan başka her kişinin hasarının eş dağılıma sahip olması varsayımı her zaman tutarlı değildir (Sigara içen birisinin, içmeyen insanlara göre kanser riskinin daha fazla olması). Bu nedenlerle prim artırımları veya özendirmeye yönelik hasarsızlık primi uygulamaları sigortacılıkta önemli bir yere sahiptir.

Sigorta firmaları savaş ve doğal afet gibi olaylardan kaynaklanan hasarları karşılamazlar (Akmüt, 1980). Bu durumun temel nedeni de Güçlü Büyük Sayılar Kanununun varsayımlarından birisidir. Yukarıda da belirtildiği gibi Y_i rassal değişkenleri birbirlerinden bağımsız kabul edilmişlerdir. Yani örneğimiz için her kişinin neden olacağı hasar miktarları birbirinden bağımsız olmalıdır. Ancak savaş ve doğal afet gibi herkesi etkileyen olaylarda bu bağımsızlıktan söz etmek mümkün değildir. Dolayısıyla bu tip hasarlar sigortacılığın teorisi ile çelişmekte ve kapsam dışında tutulmaktadırlar.

3.3.2. Sigorta Priminin Hesaplanması

Literatürde mevcut bulunan prim hesaplama yöntemlerine geçmeden önce birkaç terimi açıklamakta fayda görülmektedir. Bunlar *saf prim* ve *yükleme* terimleridir. Dikkat edilecek diğer bir husus ise bu terimlerin riskin tam olarak tanımlandığı ve çok sayıda poliçe sahibinin bu riske maruz kaldığı durumlarda geçerli olduğudur.

Saf prim, çok sayıda poliçe sahibinin neden olacağı hasar miktarının beklenen değeridir. Daha açık söylemek gerekirse, her türlü kardan ve maliyetlerden bağımsız olarak düşünülen bir tahmindir. Yükleme ise saf primin üzerine kar payının ve sigorta firmasının katlandığı her türlü maliyetin eklenmesi işlemidir (Outreville, 1998).

Bu tez kapsamında sadece saf prim hesaplamaları üzerinde durulacaktır. Ayrıca prim sözcüğü her yerde saf prim ile aynı anlamda kullanılacaktır.

Bundan sonraki tüm tanımlar belli bir grup poliçe sahibinin bulunduğu ve sigorta firması tarafından oluşturulmuş bir grup düşünülerek algılanmalıdır. Saf prim (SP), kişi başı ortalama kayıp olarak tanımlanabilir (Outreville, 1998). Bu ifadede kişi başı ortalama hasar sıklığı (OHS) ile ortalama hasar miktarının (OHM) çarpımı olarak ifade edilebilir.

$$SP = OHS \times OHM \quad (3.1)$$

Formülde geçen diğer ifadeler biraz daha açılırsa; OHS, grup içindeki toplam hasar sayısının (THS), toplam kişi sayısına (TKS) bölünmesi ile OHM ise grup içindeki tüm hasarların parasal değerinin (THM) toplam hasar sayısına bölünmesi ile bulunabilir.

$$OHS = \frac{THS}{TKS}, \quad OHM = \frac{THM}{THS} \quad \Rightarrow \quad SP = \frac{THS}{TKS} \cdot \frac{THM}{THS} = \frac{THM}{TKS} \quad (3.2)$$

Hesaplanan SP değeri ortalama bir değerdir. Yani bir kişinin hasar miktarının ortalamasıdır. Bu değer varyansı (σ_{SP}^2), eldeki verilerden hesaplanırsa, bu durumda daha ayrıntılı bir hesap yapma imkânı doğacaktır. Çünkü her kişinin hasar miktarının beklenen değeri (ortalaması) ve varyansı bilindiğinde ve bunlar birbirine eşit olduğunda (aynı grupta bulunma varsayımı), merkezi limit teoremine göre kişilerin ortalama hasarları normal dağılıma uyacaktır ($(SP, \frac{\sigma_{SP}}{\sqrt{n}})$ parametreleri ile).

Buradaki n , gruptaki kişi sayısıdır. Eğer n sonsuza giderse daha önceki bölümlerde söz ettiğimiz Güçlü Büyük Sayılar Kanunu'na göre kişilerin ortalama hasarı, varyans sıfıra yaklaştığından dolayı SP olacaktır.

Gruptaki kişilerin hasarlarının ortalamasının normal dağılıma uyması, sigorta firması açısından belirli bir güven aralığının oluşturulmasına olanak sağlar. Böyle belli bir güven seviyesinde ilgili grubun hasarının ortalama değerinin içinde bulunacağı aralığı hesaplayabilir.

Tablo 3.1’de hasar sıklığının ve hasar büyüklüğünün dağılımları verilmiş olsun.

Tablo 3.1. Prim hesabı örneğinin verileri

<u>Hasar Sıklık Dağılımı</u>		<u>Hasar Büyüklük Dağılımı</u>	
Sıklık (adet)	Olasılık	Büyüklük (YTL)	Olasılık
0	0,7	1000	0,3
1	0,2	2000	0,5
2	0,1	3000	0,2

Tablodaki veriler kullanılarak

$$\mu_{SP} = (0 \cdot 0,7 + 1 \cdot 0,2 + 2 \cdot 0,1) \cdot (1000 \cdot 0,3 + 2000 \cdot 0,5 + 3000 \cdot 0,2) = 760$$

$$E[SP^2] = (0 \cdot 0,7 + 1 \cdot 0,2 + 4 \cdot 0,1) \cdot (10^6 \cdot 0,3 + 4 \cdot 10^6 \cdot 0,5 + 9 \cdot 10^6 \cdot 0,2) = 2460000$$

$$\sigma_{SP} = 1372$$

değerleri hesaplanabilir. İncelenen grupta 10000 kişinin olduğu varsayılırsa, grubun ortalama hasarı $\left(760, \frac{1372}{\sqrt{10000}} = 13,72\right)$ parametrelili normal dağılıma uyar. Güven seviyesini %99 olarak alırsak ($Z=2,33$) ortalama hasar $760 \pm 2,33 \cdot 13,72 = (728,792)$ aralığında olacaktır.

Panjer vd. (1992) kitaplarında benzer bir yaklaşımı açıklamışlardır. Onların yöntemi- nin temel farkı kişilerin belli bir dönemdeki hasar sayısını en fazla bir almalarıdır. Daha farklı bir ifade ile yıl içindeki hasarların tümünü toplayarak tek bir hasarmış gibi değerlendirmektedirler. Dolayısıyla hasar sıklığı terimi yerine hasar yapma olasılığını kullanmaktadırlar. Ancak temelde önemli bir fark yoktur.

4. MATLAB VE YAPAY SİNİR AĞLARI

Bu bölüm MatLab programının İnternet üzerindeki yardım dokümanı temel alınarak hazırlanmıştır. Ayrıntılı bilgi için (www.mathworks.com) sitesi incelenebilir.

4.1. Yapay Sinir Ağı Arayüzüne Giriş

Bu kullanıcı arayüzü (KA) kullanılarak yeni bir ağ oluşturulabilir, oluşturulan bu ağ eğitilip simüle edilebilir, komut satırı çalışma ortamından bu ortama veri transfer edilebilir. İlerleyen bölümlerde arayüzün kullanımı bir örnek aracılığıyla açıklanacaktır. Bu örnek VE işlemini gerçekleştiren bir Algılayıcı'dır.

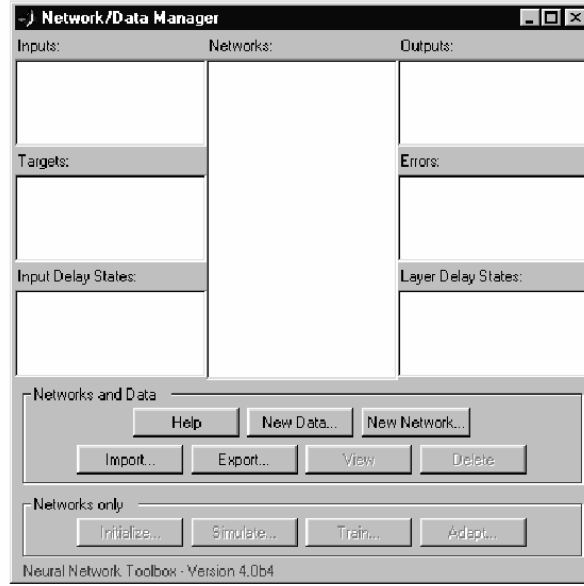
4.2. Ağın Oluşturulması

İlk olarak ağın MatLab programına tanıtılması gerekmektedir. Bu amaçla da KA kullanılarak yeni bir ağ yaratılmalıdır. Ancak hepsinden önce ağın eğitim setinin tanımlanması gerekmektedir. Bu kısımda verilen örnek, VE fonksiyonu olduğundan eğitim setinin girdi vektörü

$X = [0 \ 0 \ 1 \ 1; 0 \ 1 \ 0 \ 1]$ ve çıktı vektörü de $T = [0 \ 0 \ 0 \ 1]$ olmalıdır. Örnek olarak oluşturulan ağ, ANDNet olarak adlandırılmıştır.

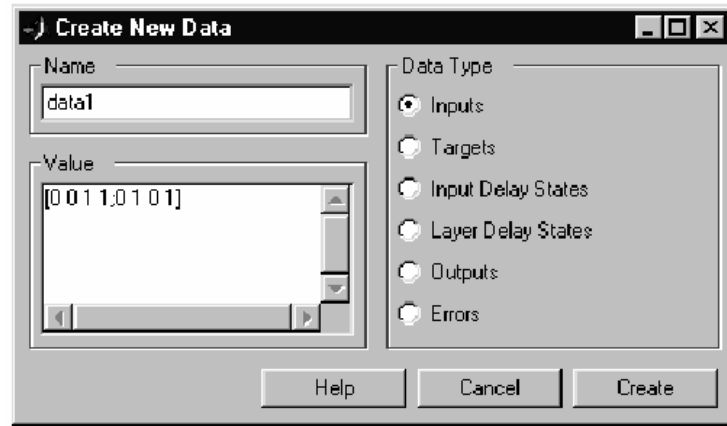
Öncelikle KA'nın aktif hale getirilmesi gerekmektedir. Bunun için komut satırına “nntool” yazmak yeterlidir. Bunun yanında “Start” menüsünden “Toolbox” seçeneği oradan da “Neural Network” seçeneği seçilerek aynı sonuç elde edilebilir. Bu işlemlerden sonra kullanıcının karşısına Şekil 4.1'deki gibi bir ekran gelmektedir.

Bundan sonra girdi ve çıktı vektörlerinin tanımlanması gerekmektedir. Bunun için “New Data” sekmesine tıklanmalıdır. Bu aşamada Şekil 4.2'deki “Create New Data” penceresi gelir.



Şekil 4.1. Network / Data Manager Arayüzü

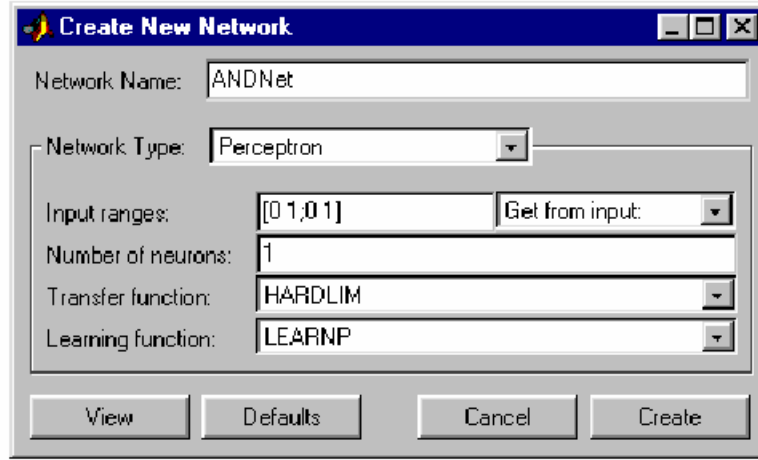
Bu ekranda “Name” kısmına *T* yazılmalı ve “Value” bölümüne [0 0 1 1;0 1 0 1] ifadesi yazılmalıdır. Burada “;” işaretleri sütunları birbirinden ayırmaktadır. Yani noktalı virgüller arasındaki sayılar matrisin sütunlarını oluşturmaktadır. “Data Type” olarak *Input*



Şekil 4.2. Create New Data Ekranı

seçilmelidir ve son olarak da “Create” tuşuna basılmalıdır. Çıktı vektörünün tanımlanması içinde benzer işlemler yapılır ve “Data Type” olarak *Target* seçilir. Bu işlemlerden sonra kullanıcının karşısına tekrar “Network / Data Manager” ekranı gelir. Artık “Inputs” ve “Targets” bölümlerinde tanımlanan *X* ve *T* vektörleri görünecektir.

Yeni bir ağın oluşturulması için “Network / Data Manager” ekranındayken “New Network” tuşuna basılır ve “Create New Network” penceresi açılır (bkz. Şekil4.3).

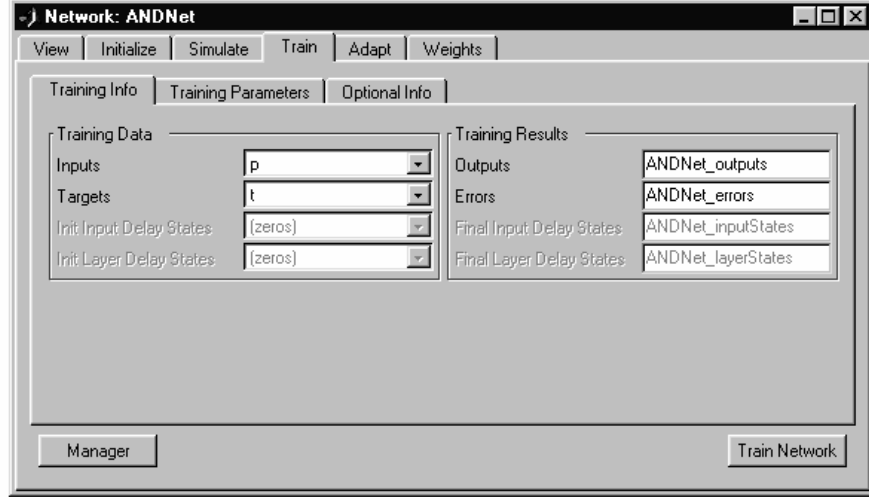


Şekil 4.3. Create New Network Ekranı

Bu ekran yardımı ile yeni bir yapay sinir ağı oluşturulabilir. İlk olarak yeni oluşturulacak ağın ismi girilir. Uygulama örneği için “ANDNet” ismi “Network Name” bölümüne girilir. Daha sonra ağın tipi seçilir. Örnek için “Perceptron” seçilecektir. Bunun dışında diğer tüm ağ tipleri bu bölümden seçilebilir. Seçilen her farklı tip ağ için gerekli parametreler yine “Create New Network” ekranı aracılığıyla program tarafından istenecektir. Bir sonraki adımda girdi vektöründe bulunan verilerin alt ve üst limitlerinin programa bildirilmesi gerekmektedir. Bu işlem için “Get From Input” işlem kutucuğunun üzerine basılıp, istenen girdi vektörü seçilir. Uygulama örneği için X isimli girdi vektörü seçilmelidir. Daha sonra düğüm sayısı girilmelidir. Tek düğümün yeterli olacağı varsayılırsa bu bölüme 1 girilmelidir. Hareket fonksiyonu için “HardLim” (Algılayıcı için kullanılan hareket fonksiyonudur, sınır değeri 0 alınmıştır) fonksiyonunu seçilsin ve öğrenme fonksiyonu için “LEARNP” (Algılayıcı için kullanılan öğrenme fonksiyonudur) fonksiyonu seçilsin. Tüm bu istenenler girildikten sonra “Create” tuşuna basılmalıdır. Böylece tekrar “Network / Data Manager” ekranına geri dönülür ve artık “Network” bölümünde yeni oluşturulan ağın ismi yazmaktadır.

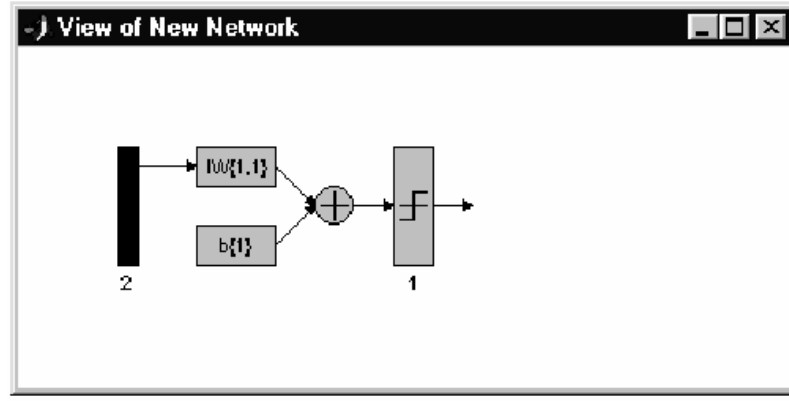
4.3. Ağın Eğitilmesi

Artık oluşturulan “ANDNet” isimli ağ eğitilebilir. Bunun için “Network / Data Manager” ekranındayken ağın ismi seçildikten sonra “Train” tuşuna basılmalıdır. Bu işlemten sonra Şekil 4.4’deki arayüz ekrana gelir.



Şekil 4.4. Network: ANDNet Ekranı

Bu arayüz kullanılarak seçilmiş olan ağ eğitilebilir, görüntülenebilir, başlangıç ağırlıkları tekrar rasgele atanabilir, ağırlıklar görüntülenebilir ve eğitilmiş ağ simüle edilebilir. “View” seçeneği seçilirse şu anda seçili olan ağ için Şekil 4.5’deki arayüz ekrana gelir. Ağı eğitilmesi için “Train” seçeneği seçilmelidir (bkz. Şekil 4.6). Daha sonra eğitimde kullanılacak girdi ve çıktı vektörleri seçilmelidir. “Training Results” bölümüne ağı eğitimi tamamlandıktan sonra ağı çıktılarının ve her girdi elemanı için oluşan hata miktarlarının kaydedilecekleri dosya isimleri girilir. Örnek için “ANDNet_outputs” ve “ANDNet_errors” dosya isimleri girilmiştir. Bu işlemlerden sonra “Train Network” tuşuna basılarak eğitim başlatılır. Eğer eğitim parametreleri değiştirilmek isteniyorsa “Training Parameters” bölümündeki değerler (en büyük devir sayısı, hata hedefi,...) değiştirilebilir.

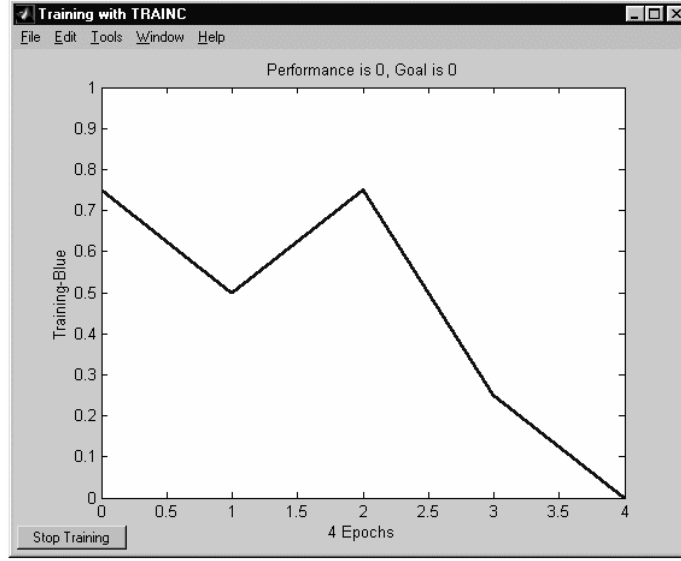


Şekil 4.5. View of New Network Ekranı

“Optional Info” seçeneği kullanılarak eğitim kümesinin yanı sıra ağın genelleme yeteneğini tespit etmek amacı ile eğitim esnasında kullanılmayacak bir test kümesi tanımlanabilir. Eğitim esnasında her devirden sonra test kümesindeki veriler ağa girilir ve ağın daha önce hiç görmediği bu verilere göre ağın performansı (hata miktarı) belirlenebilir.

Eğitim sürecinde Şekil 4.6’da gösterilen grafik ekranda belirir. Bu grafik yardımı ile ağın eğitim ve test performansı kullanıcı tarafından takip edilebilir.

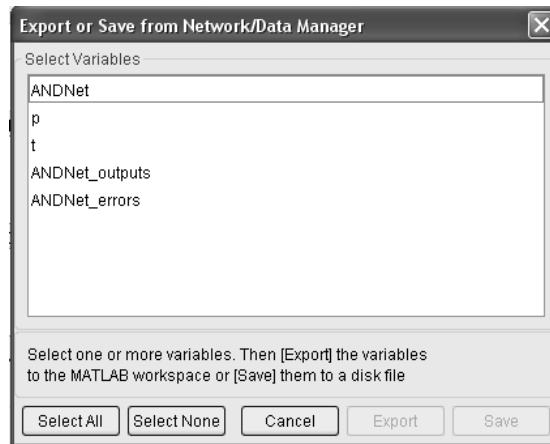
Şekil 4.6’ da gösterilen grafikte ağın eğitimi tamamlanmıştır. 4 devir neticesinde istenen hata miktarına ulaşılmıştır. Bu örnek için hata hedefi 0 olarak belirlenmiştir. Bu değeri kontrol etmek için “Simulate” bölümüne girerek X (eğitim seti) için ağ simüle edilip ağın çıktı değerleri kontrol edilebilir. Bu işlemin gerçekleştirilmesi için “Network / Data Manager” ekranında iken simüle edilecek ağ seçildikten sonra “Simulate” tuşuna basılmalıdır. Ekrana gelen arayüzde “Input” bölümünden X vektörü seçilmelidir. Ağın çıktı değerlerinin kaydedileceği dosya ismi girildikten sonra (örnek için ANDNet_outputsSim girildiği varsayılın) “Simulate Network” tuşuna basılmalıdır. Bu işlemlerden sonra “Network / Data Manager” ekranında “ANDNet_outputsSim” isimli dosya belirecektir. Bu dosyanın üzerine çift tıklandığında ağın çıktıları gösterilecektir. Örnek uygulama için $[0 \ 0 \ 0 \ 1]$ (VE fonksiyonun sırasıyla aldığı değerler) ifadesi görülmelidir. Tam değerlerin çıkmasının sebebi eğitim esnasında sıfır hatanın elde edilmiş olmasıdır.



Şekil 4.6. Eğitim Performansı Ekranı

4.4. Ağ Çıktılarının Komut Satırı Çalışma Ortamına Transfer Edilmesi

Oluşturulmuş olan herhangi bir ağ eğitildikten veya simüle edildikten sonra oluşan çıktı dosyaları Komut Satırı Çalışma Ortamı' na aktarılarak bu dosyalar üzerinde kapsamlı analizler yapılabilir. Bu işlem için “Network / Data Manager” ekranı aktif halde iken “Export” tuşuna basılır ve ekrana Şekil 4.7’deki arayüz gelir. Bu ekranda transfer edilmek istenen nesneler seçildikten sonra export tuşuna basılır. Bu işlemden sonra seçilmiş olan nesneler komut satırı çalışma ortamında kullanılabilirler. “Export” işlemi sadece dosyalara uygulanmaz. Örneğin, bir ağ da komut satırı çalışma ortamına transfer edilebilir ve burada analiz edilebilir.

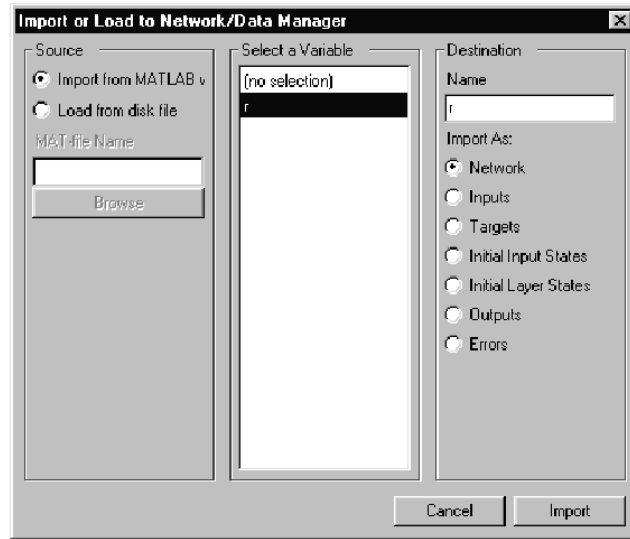


Şekil 4.7. Export or Save Network/Data Manager Ekranı

4.5. Komut Satırı Çalışma Ortamından Yapay Sinir Ağı Modülüne Veri Transferi

Bu işlemi gerçekleştirmek için Komut Satırı Çalışma Ortamı'nda hali hazırda bulunan bir değişken veya ağ bulunmalıdır. Komut Satırı Çalışma Ortamı'nda $r = [0; 1; 2; 3]$ şeklinde tanılanmış bir değişken olduğu varsayılın. Bu değişkenin *girdi değişkeni* olarak transfer edilmek istendiği varsayılın. Bu durumda “Network / Data Manager” arayüzü aktif iken “Import” tuşuna basılır ve Şekil 4.8'daki arayüz ekrana gelir.

İlk olarak “Source” bölümünde “Import from MATLAB” seçeneği seçilir. Bir sonraki adımda istenen değişken (örnek için r değişkeni) seçilir ve değişken tipide belirlendikten sonra (örnek için “Input” seçeneği seçilmelidir) “Import” tuşuna basılır. Bu işlemden sonra “Network / Data Manager” ekranının “Input” bölümünde r değişkeni görülebilecektir.



Şekil 4.8. Import or Load Network / Data Manager Ekranı

4.6. Bir Değişkenin Bir Dosyaya Kaydedilmesi, Silinmesi ve Daha Sonra Geri Çağırılması

Örnek olarak oluşturulmuş ANDNet isimli ağ öncelikle kaydedilecek, daha sonra silinecek ve son olarak da kaydedildiği yerden geri çağırılacaktır. Kaydetme işlemi yapmak için “Network / Data Manager” ekranı aktifken “Export” tuşuna

basılmalıdır. Bu durumda daha öncede söylendiği üzere Şekil 4.7'deki arayüz ekrana gelecektir. Bu ekranda kaydedilmek istenen öğe seçilir (örnek için ANDNet ağı seçilir) ve “Save” tuşuna basılır. Bunun sonucunda “Save To a MAT File” arayüzü ekrana gelir. Bu ekranda dosya ismi verilerek (örnek için mynetfile ismi verilsin) kayıt işlemi gerçekleştirilir. Kaydedilen öğeler “.mat” uzantılı dosyalarda saklanır.

Değişkenlerin ve öğelerin silinmesi kolayca gerçekleştirilebilecek bir işlemdir. “Network / Data Manager” ekranı aktifken öğeyi seçip “Delete” tuşuna basmak yeterlidir. Örnek içinde ANDNet isimli ağın seçildiği ve silindiği düşünölsün.

Daha önceden kaydedilmiş bir öğeyi geri yüklemek için “Import” komutu kullanılır. Diğer bir ifade ile “Network / Data Manager” ekranı aktifken “Import” tuşuna basılmalıdır. Bilindiği gibi bu durumda Şekil 4.8'daki arayüz ekrana gelir. İlk olarak “Source” bölümünde “Load From Disk File” seçeneği seçilir ve “Browse” tuşuna basılır. Açılan ekrandan istenen dosya bulunup seçildikten (örnek için mynetfile.mat dosyası bulunur ve seçilir) sonra “Open” tuşuna basılır ve program kullanıcıyı tekrar “Import or Load to Network / Data Manager” (Şekil 4.8) başlıklı ekranda geri döndürür. Bir sonraki adımda “Import As” listesinden öğenin tipi seçilir (örnek için “Network” seçeneği tıklanmalıdır). Son olarak da “Load” tuşuna basılarak geri yükleme işlemi tamamlanmış olur. Artık “Network / Data Manager” ekranında geri yüklenen öğe görünmeye başlayacaktır (Örnek için bir önceki adımda silinmiş olan ANDNet isimli ağ tekrar “Network” bölümünde görölebilecektir).

5. ÖNERİLEN MODEL

5.1. İstatistiksel Yöntemin Eleştirilmesi

Sigortacılığın teorisi incelendiğinde, sistemin düzgün olarak çalışması için büyük sayılar kanunun çalışması gerektiği görülecektir. Diğer bir ifadeyle bir risk açısından birbirine eşdeğer olan pek çok kişinin sigortalanması gerekmektedir. Daha sonra bu kişilerden hasarlarının beklenen değerlerinin belli bir oranda fazlası alınırsa uzun vadede firmanın hayatta kalabilme olasılığı 1'e yaklaşmaktadır.

Yukarıda açıklanan ifade teoride oldukça başarılıdır. Ancak uygulamaya geçildiğinde, büyük sayılar kanununun varsayımları ile ilgili problemler ortaya çıkmaktadır. Bir araya getirilen kişilerin, ilgili risk açısından aynı dağılıma sahip olması gerekmektedir. Ancak sağlık sigortası uygulamalarında belli bir risk için aynı dağılıma sahip kişilerin bir araya getirilmesi oldukça güç bir iştir. Bunun en temel nedeni insan sağlığına doğrudan etki eden faktörlerin sayısının çok fazla olması ve faktörlerin insandan insana çok değişkenlik göstermesidir. Günümüzde de sigorta firmaları insanları belli değişkenlere göre gruplayıp, bu gruplara göre prim hesabı yapmanın mümkün olduğunu fark etmişlerdir. Bu nedenle sağlık sigortası prim hesaplarında doktorların uzman görüşünü kullanan puanlama sistemini kullanmaktadırlar. Bu sistemde öncelikle bir alt prim hesaplanmaktadır. Bu hesaplama belli değişkenlere göre gruplanmış kişilerin hasarlarının ortalamasıdır. Daha sonra bu alt primler kişinin diğer bilgilerine göre (sigara tüketimi, yaşanan il, vb.) belirlenmiş bir katsayı ile çarpılmaktadır. Bu katsayılar alt primi artırıcı yönde olabileceği gibi azaltıcı yönde de olabilir. Katsayıların belirlenmesi, oldukça zahmetli olabilmektedir. Bu süreç, tarihsel verilerden ve doktor görüşlerinden yararlanılarak yapılmaktadır.

Bu tez kapsamında doktorların verdikleri karara yardımcı olacak ve hatta onun yerine geçecek bir model geliştirilmeye çalışılmıştır. Yapılmak istenen, kişi bazında prim

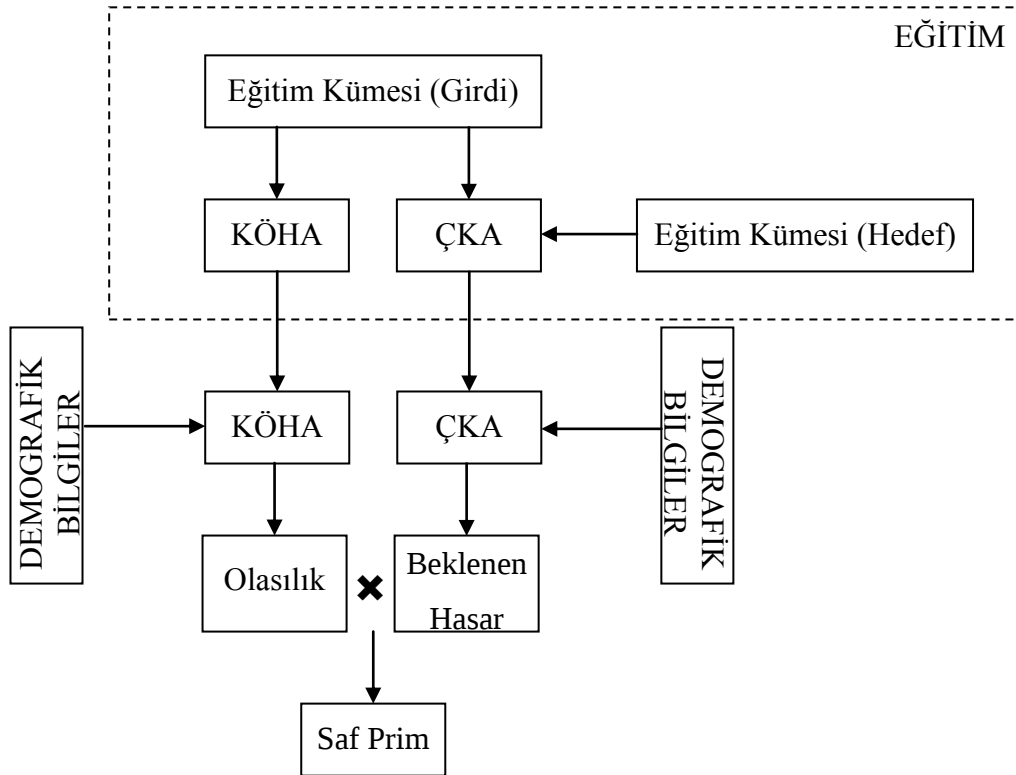
hesabı yapmaya olanak verecek bir yöntem geliştirmektir. Bu amaçla yapay sinir ağları hem kümeleme, hem de kişilerin demografik bilgileri ile beklenen hasar miktarları arasında olduğu varsayılan fonksiyona yakınsama amacı ile kullanılmışlardır.

5.2. Önerilen Yöntem

Daha önce de açıklandığı gibi prim hesabında, her kişinin hasar yapma olasılığı ve eğer hasar olduysa bu hasarın beklenen değeri çarpılmaktadır.

$$\begin{aligned} E[\text{hasar}] &= E[\text{hasar} \mid K = 1] \cdot P\{K = 1\} + E[\text{hasar} \mid K = 0] \cdot P\{K = 0\} \\ &= E[\text{hasar} \mid K = 1] \cdot P\{K = 1\} \end{aligned} \quad (5.1)$$

Burada K rassal değişkeni eğer hasar olduysa 1 değerini aksi halde 0 değerini almaktadır. Ayrıca hasar olmadıysa hasarın beklenen değerinin 0 olacağı açıktır.



Şekil 5.1. Önerilen yöntemin ana kademeleri

Önerilen modelde (5.1) numaralı denklemde çarpılan iki terim için ayrı ayrı sinir ağları geliştirilecektir. Hasar yapma olasılığını hesaplamak için bir KÖHA ağı

kullanılarak eğitim kümesi kümelenecek ve her küme için hasar yapma olasılığı hesaplanacaktır. Hasar gerçekleştiğinde bu hasarın beklenen değerinin hesaplanması için ise çok katmanlı ağlardan yararlanılacaktır. Bu iki ağ oluşturulup eğitildikten sonra yeni kişiler için prim hesabı yapılacağı zaman, kişilerin bilgileri bu iki ağa girilecek ve bulunan değerler çarpılarak prim hesaplanacaktır. Bu adımlar Şekil 5.1’ de özetlenmiştir.

5.3. Uygulama Aşamaları

5.3.1. Veri kümesinin düzenlenmesi

Uygulamada kullanmak amacı ile bir Türk sigorta firmasının verileri alınmıştır. Bilindiği gibi sigorta firmaları, birçok konuda teminat verebilmektedirler. Sadece tek bir risk faktörü ile ilgilenebilmek için sadece tek bir teminata ait veriler kullanılmıştır. Bu amaçla, “Ayakta Tedavi” teminatı seçilmiştir. Firmadan alınan verilerde aşağıdaki değişkenler bulunmaktadır:

- a) Meslek Sınıfı**
 - (i) Elle çalışan
 - (ii) Elle ve bedenen çalışan
 - (iii) Makine ile ve bedenen çalışan
- b) Cinsiyet**
 - (i) Bay
 - (ii) Bayan
- c) Yaş**
- d) Motosiklet kullanımı**
 - (i) Kullanmıyor
 - (ii) Amatör

- (iii) Profesyonel
- e) Alkol kullanımı**
 - (i) Düzenli kullanmaz
 - (ii) Düzenli kullanır, 10 yıldan az
 - (iii) Düzenli kullanmaz, 10 yıldan fazla
- f) Sigara kullanımı (adet/gün)**
- g) Sigara kullanım süresi (yıl)**
- h) Yaşadığı bölge**
 - (i) 1. Bölge: İstanbul
 - (ii) 2. Bölge: Ankara, İzmir
 - (iii) 3. Bölge: Antalya, Diyarbakır, Kocaeli, Gaziantep, Samsun, Kayseri, Erzurum, Eskişehir, Adana, Bursa, Konya
 - (iv) 4. Bölge: Diğer iller
- i) Kilo/Boy indeksi**
- j) Prim**
- k) Hasar**

Yaşadığı bölge değişkeni haricinde diğer tüm değişkenler firma tarafından tutulduğu şekilde kullanılmıştır. Sadece “yaşadığı bölge” değişkeni firma tarafından farklı bir şekilde yaşadığı il olarak tutulmaktadır. Ancak bu değişken modele kategorik değişken olarak dâhil edileceğinden tüm illeri kullanmak hem çok fazla değişkene neden olacağı için hem de iller arasındaki farklılıklar her zaman anlamlı olmadığından modeli gereksiz yere karmaşık ve büyük yapacaktır. Bu amaçla iller genel olarak büyük şehir olup olmama durumuna göre gruplanmıştır. Ancak yine de İstanbul, Ankara ve İzmir diğer illerden ayrı tutulmuştur.

5.3.2. Verilerin Analize Hazırlanması

Öncelikle değişkenlere karar vermek gerekmektedir. Bu amaçla hangi değişkenlerin sayısal, hangi değişkenlerin kategorik (0 ya da 1 değeri alan) olacağı belirlenmelidir. Meslek sınıfı kategorik bir değişken olarak seçilmiştir. 3. tipe giren hiçbir kişi bulunmadığından tek bir kategorik değişken yeterli olacaktır (0 değeri sadece elle çalışanları, 1 değeri ise elle ve bedenen çalışanları temsil edecektir). Cinsiyet için tek bir kategorik değişken yeterlidir. Yaş sayısal bir değişkendir. Motosiklet kullanımı içinde tek bir kategorik değişken yeterlidir, çünkü veri kümesinde profesyonel olarak motosiklet kullanan kimse bulunmamaktadır. Benzer şekilde alkol kullanımı içinde tek bir kategorik değişken yeterlidir, çünkü 10 yıldan fazla düzenli olarak alkol kullanan kimse veri kümesinde bulunmamaktadır. Sigara kullanım miktarı ve süresi, kilo/boy indeksi, prim ve hasar miktarları sayısal değişkendir. Yaşadığı bölgenin temsili için ise 4 farklı kategorik değişkene ihtiyaç vardır. Kişi hangi bölgede yaşıyorsa o bölgenin değişkeni 1 diğer değişkenler 0 değerini alacaktır.

Firmadan alınan veriler içinde tam olarak 5264 adet kayıt bulunmaktadır. Her bir kayıta yukarıda belirtilen değerler yer almaktadır. İlk incelemede bu kayıtların bazılarının eksik kimilerinin ise mantıksız değerlerle dolu olduğu fark edilmiştir. Bu hatalı kayıtlar veri kümesinden elenmiştir. Bu işlemin ardından veri kümesinde aşırı uçlarda bulunan, yani hasar miktarı çok büyük ya da çok küçük olan kayıtlar yine elenmiştir. Bu kayıtlar çok sıradışı durumları göstermektedir ve eğitim aşamasında kullanılmaları durumunda, eğitimin performansını düşürecekleri düşünülmüştür.

Bu eleme işlemlerinin ardından geri kalan kayıtlar rassal olarak ikiye ayrılmıştır. 3402 adet kayıt, ağların eğitiminde kullanılmak için alınmıştır. 896 adet kayıt ise tüm işlemler bittikten sonra modelin uygulanabilirliğini göstermek için kullanılmak üzere ayrılmıştır. Bu test kümesi, sinir ağlarının genelleme yeteneğini test etmek için kullanılan test kümesi ile karıştırılmamalıdır. Sinir ağlarını test etmek için kullanılan veriler yine 3402 adet kayıt içeren ilk kümeden alınmıştır. 3402 kayıtlık ilk veri kümesi tüm popülasyonu temsil edebilecek genişlikte ve çeşitlilikte seçilmeye çalışılmıştır. Bundan sonra 3402 kayıtlı küme birincil küme, 896 kayıtlı küme ise ikincil küme olarak isimlendirilecektir.

5.3.3. KÖHA Aşaması

Bu aşamada birincil kümede bulunan tüm kayıtlar Kohonen'in Özörgütlenmeli Harita Ağı kullanılarak kümelenmiştir. Kümeleme için KÖHA seçilmesinin sebebi Mangiameli vd.'lerinin (1996) bu yöntemi diğer hiyerarşik kümeleme algoritmalarına göre daha üstün bulmalarıdır.

İlk olarak normalizasyon işlemi yapılmıştır. Böylece, daha büyük farklara sahip değişkenlerin kümeleme işleminde baskın olmaları önlenmeye çalışılmıştır. Daha sonra 12 girdi sinir hücresi bulunan bir KÖHA oluşturulmuştur. Daha önceden veri kümesinde kaç kümenin bulunduğu bilinmediğinden, çıktı sinir hücresi sayısı 2'den başlanarak gitgide arttırılmıştır. Bu denemeler sonucunda, en uygun ayrılmanın veri setinin 8 kümeye bölündüğünde ortaya çıktığı görülmüştür. En uygun ayrılmanın tayini gözlem ile yapılmıştır. Ancak her kümenin merkez değerleri açısından birbirinden yeterince farklı olması gerektiği düşünülmüş ve bu bağlamda uygun küme sayısına karar verilmiştir. Kullanılan parametre değerleri, genellikle MatLab programı tarafından önerilen değerlerdir. Farklı olarak yineleme sayıları, komşuluk değeri ve uzaklık fonksiyonu değiştirilmiştir. Uzaklık fonksiyonu olarak Manhattan uzaklık fonksiyonu kullanılmıştır (uygunluğu deneyerek tespit edilmiştir, kriter olarak merkez değerlerin birbirinden anlamlı şekilde farklı olması alınmıştır.). Komşuluk değeri olarak ise 0 alınmıştır. Yani komşuluk tanımlanmamıştır. Bu şekilde kümelerin birbirlerinden bağımsız olarak oluşması sağlanmıştır ve böylece daha iyi bir ayrılmanın gerçekleşeceği var sayılmıştır.

Bu aşamada oluşan 8 kümenin merkez değerleri Tablo 5.1'de verilmiştir.

1. Grup: Bu grupta alkol tüketimi diğer birçok gruba göre oldukça fazladır. Tam bir cinsiyet ayrımına gidilememiştir. Tüm müşteri havuzunda çoğunluk sadece elle çalışmaktadır. Elle ve bedenle çalışan kimi kişiler bu gruba dahil edilmiştir. Bu grubun sigara tüketim miktarı ve süresi diğer gruplardan (2. grup hariç) daha yüksektir. Yaş ve kilo/boy indeksi açılarından ortalama bir gruptur. Genelde İzmir ve Ankara'da yaşamaktadır.

2. Grup: Bu grup büyük şehir olmayan illerde yaşayan ve çok sigara tüketen erkeklerden oluşmaktadır. En çok sigara tüketimi bu grup için geçerlidir. Bu grupta düzenli olarak alkol tüketen bulunmamaktadır. Yaş ortalaması yaklaşık olarak 31 ve

kilo/boy indeksi 0,24'tür. Bu gruptakilerin kilo/boy indeks ortalaması tüm gruplar içinde en büyüktür. Bu gruptakilerin tümü sadece elle çalışmaktadır.

3.Grup: Bu grupta tam bir cinsiyet ayrımına gidilmemiştir. İş tipi genel olarak elle çalışandır, ancak bu grup bedenen çalışanlarında bulunduğu iki gruptan birisidir (diğeri grup1). Sigara tüketimi ilk iki gruba göre oldukça düşüktür (hem miktar hem

Tablo 5.1. KÖHA aşaması sonucunda oluşan kümelerin merkez değerleri

Grup No	Alkol Kul.	Cins.	İş Tipi	Sigara Mik.	Moto. K.	Sigara K. Sür.	Yaş	Kilo / Boy	B.1	B.2	B.3	B.4
1	0,004	1,455	1,006	1,454	0	1,278	27,96	0,234	0	1	0	0
2	0	1	1	1,585	0	2,523	30,91	0,241	0	0	0	1
3	0	1,420	1,008	0,301	0	0,385	27,05	0,232	0	0	1	0
4	0	2	1	0,425	0	0,228	25,03	0,219	0	0	0	1
5	0,001	1	1	0,801	0	1,2	39,31	0,255	1	0	0	0
6	0	2	1	0,013	0	0,005	11,00	0,216	1	0	0	0
7	0	2	1	0,625	0	0,714	36,45	0,220	1	0	0	0
8	0	1	1	0,020	0	0,016	9,73	0,219	1	0	0	0

de süre açısından). Yaş ortalaması yaklaşık olarak 27, kilo/boy indeksi ortalaması ise yaklaşık olarak 0,23'tür. Bu gruptakilerin tamamı üç büyük şehir dışında kalan (İstanbul, İzmir, Ankara) büyük şehir olan illerde yaşamaktadır.

4.Grup: Bu grup tamamı ile sadece elle çalışan, büyük şehir olmayan illerde yaşayan bayanlardan oluşmaktadır. Yaş ortalaması 25, sigara tüketimi ise ilk iki gruba göre çok daha azdır. Bu grubun kilo/boy indeksi tüm gruplar göz önüne alındığında nispeten daha düşüktür.

5.Grup: Bu grupta çoğunluk düzenli olarak alkol kullanmamaktadır. Ancak bu grup, düzenli olarak alkol kullananların bulunduğu iki gruptan birisidir. Bu grup tamamı

ile erkeklerden oluşmaktadır. Bu gruptakiler sigara tüketimi ortalamasında orta sıralardadır, ancak sigara tüketim süresi açısından ilk sıralardadır. Yaş ortalaması ve kilo/boy indeksi ortalaması en yüksek olan gruptur. Tamamı İstanbul'da yaşamaktadır. ,

6.Grup: Bu grupta İstanbul'da yaşayan bayanlar bulunmaktadır. Yaş ortalaması 11 gibi oldukça küçük bir değerdir. Bu gruptakilerin sigara tüketimi oldukça düşüktür ve kilo/boy indeks ortalaması yine tüm gruplar içinde en düşüktür.

7.Grup: Bu grupta yine İstanbul'da yaşayan bayanlar bulunmaktadır. Ancak 6. gruba göre yaş ortalamaları 36 gibi oldukça yüksek bir değerdir. Kilo/boy indeks ortalaması 0,22'dir. Sigara tüketimi yine 5. gruba göre oldukça yüksektir, ancak ilk iki gruba göre yine az kalmaktadır.

8.Grup: Bu grupta yaş ortalaması 9 gibi oldukça küçük bir değerdir ve İstanbul'da yaşayan erkekler bulunmaktadır. Sigara tüketimi 6. grup gibi oldukça düşüktür. Kilo/boy indeksi ortalaması ise 0,216 olarak en küçük ikinci değerdir.

Kümeleme işleminin ardından her kümenin hasar yapma olasılıkları belirlenmiştir. Bu değerler hasarı sıfırdan farklı kayıt sayısını, gruptaki tüm kayıt sayısına bölerek elde edilmiştir. Bulunan olasılıklar Tablo 5.2'de gösterilmiştir.

Tablo 5.2. Gruplara göre hasar olasılıkları

Grup No	Hasar Olasılığı
1	0,70703
2	0,61538
3	0,676
4	0,63636
5	0,64891
6	0,83077
7	0,76834
8	0,80876

5.3.4. Çok Katmanlı İleri Beslemeli Ağın Oluşturulması Aşaması

Bu aşamada birey bazında hasarın gerçekleştiği bilindiğinde, bu hasar miktarının beklenen değerinin belirlenmesinde kullanılacak olan çok katmanlı ağ oluşturulmuştur. Bu işlem için birincil kümede hasarı sıfırdan büyük olan kayıtlar, %80-%20 oranlarında bölünmüştür. %80'lik kısım ağın eğitimi için kullanılmıştır. %20'lik bölüm ise eğitim esnasında ağın genelleme yeteneğini test etmek amacı ile kullanılmıştır. Oluşturulan bu eğitim ve test kümeleri ilk olarak 0-1 arasına normalize edilmiştir (en büyük değere bölerek). Ancak dikkat edilmesi gereken nokta test kümesinin de eğitim kümesinin en büyük değerlerine göre normalize edilmesidir. Hatta benzetim aşamasında ağa girilecek olan değerlerin eğitim kümesinin değerlerine göre normalize edilmesidir. Bunu sağlamak için de eğitim kümesinde her değişkenin alabileceği en büyük ve en küçük değerlerin bulunmasına dikkat edilmiştir..

Normalizasyon işleminin ardından ağın mimarisinin belirlenmesi gerekmektedir. Hareket fonksiyonu olarak tüm ağda lojistik fonksiyonu kullanılmıştır. Sadece tek bir gizli katman kullanılmıştır. Bunun gerekçesi, sinir ağlarının anlatıldığı 2. bölümde verilmiştir. Lojistik fonksiyonunun seçim nedeni ise denemelerde diğer fonksiyonlara göre daha iyi performans sağlamasıdır. Gizli katmandaki gizli sinir hücresi sayısı ise birer birer artırılarak, eğitim ve genelleme performanslarına göre belirlenmiştir. Sonuçta gizli sinir hücresi sayısı 7 olarak belirlenmiştir. Bunun sebebi ise 7 sinir hücreli ağın eğitim performansının tatmin edici olması ve 7 gizli sinir hücresinden sonra (8 ve daha fazla) ağın genelleme yeteneğinin belirgin şekilde azalmasıdır.

5.3.5. Modelin Simülasyonu ve Yorumlar

Modelin uygulanabilirliğinin gösterimi için 896 kayıt içeren ikincil küme kullanılmıştır. Bu kayıtlar, normalize edildikten sonra hem özörgütlenmeli ağa hem de çok katmanlı ileri beslemeli ağa girilmişlerdir. Özörgütlenmeli ağın çıktıları olarak, her kayıttan hangi gruba dâhil olduğu belirlenmiştir ve böylece hasar yapma olasılıkları belirlenmiştir. Çok katmanlı ağın çıktısı olarak da eğer hasar yaptılarsa bu hasarın beklenen değeri belirlenmiştir. Daha sonra bulunan bu iki değer her kayıt için çarpılarak primler hesaplanmıştır.

Modelin başarısının göstergesi olarak Tablo 5.3'deki korelasyon değerleri bulunmuştur. Model hem gerçekleşen değerlerle hem de firma primleri ile mukayese edilmiştir. Önerilen modelle hesaplanan primler ile firmanın önerdiği primler arasında sadece korelasyona bakılabilmektedir. Çünkü firmadan alınan primler içinde belli bir oranda kar bulunmaktadır ve bu oranlar firma tarafından gizli tutulmuştur.

Şurası kesindir ki ideal primler, gerçekleşen hasarlarla korele olmalıdır. Ancak ne bu tezde önerilen model, ne de firmada kullanılan doktorlu puanlama sistemi gerçekleşen hasarlarla yeterli bir korelasyona ulaşamamıştır. Bunun en temel nedeni, kişilerin sağlık masraflarının çok sayıda değişkenden etkilenmesi ve firmada tutulan kayıtların bu değişkenleri içermemesidir. Ancak yine de önerilen model firmadaki sistemden daha yüksek bir korelasyona sahiptir. Ancak esas ilginç olan nokta, modelin önerdiği primlerle, firma tarafından önerilen primler arasında nispeten yüksek bir korelasyonun bulunmasıdır.

Tablo 5.3. Önerilen model ile gerçekleşen değerler ve firma sistemi arasındaki korelasyonlar

Değişken 1	Değişken 2	Korelasyon
Pozitif hedef değerler	Hasar beklenen değerleri	0,1957
Hedef değerler	Modelin verdiği primler	0,1348
Hedef değerler	Firmanın önerdiği primler	0,1214
Modelin verdiği primler	Firmanın önerdiği primler	0,5942

Bu aşamada firmada uygulanan sistemi kısaca açıklamak yerinde olacaktır. Firmada istatistiksel yöntemler kullanılarak sadece iki değişkene göre (yaş ve cinsiyet) prim hesabı yapılmaktadır. Bu değerler, doktorların görüşleri alınarak ve bu tezde önerilen modelde kullanılan değişkenler dikkate alınarak hazırlanan bir puanlama sistemine göre revize edilmektedir. Uygulama aşamasında gerekli görülürse bazı kişilerin muayene raporları doktorlar tarafında kontrol edilmekte ve primler tekrar revize edilmektedir. Diğer bir ifade ile firmada uygulanan sistemde, karar mekanizması içinde doktor görüşü bulunmaktadır.

Bu tezde önerilen model ile firmadaki sistem arasında korelasyonun yüksek çıkması ve gerçekleşen değerlerle nispeten daha iyi bir korelasyonun elde edilmesi, önerilen modelin bir karar destek sistemi olarak firma tarafından uygulanabileceğini göstermektedir.

Değerlendirilmesi gereken bir diğer nokta ise önerilen değerlerin prim olmasıdır. Yani bu sistemin değerlendirilmesinde tüm popülasyon için alınan primlerin oluşan hasarları karşılaması gerekmektedir. Bunu değerlendirmek içinde korelasyonun yanında esas değerlere bakılmalıdır. 896 kayıt incelendiğinde önerilen model ile hesaplanan primlerin standart hatası 683,65'tir ve toplanan prim miktarı, toplam hasar miktarından 175680 YTL daha fazladır. Dolayısıyla, önerilen primler toplam hasar miktarını fazlasıyla karşılamaktadır. Ancak fark yine de büyüktür, bunun nedeni olarak, hedef değerlerle model tarafından önerilen primler arasındaki korelasyonun düşüklüğü gösterilebilir.

6. SONUÇLAR VE GELİŞTİRME ÖNERİLERİ

Bu tez kapsamında, tüm sigorta çeşitlerinde prim hesabı yapmakta kullanılabilecek yeni bir metot önerilmiştir. Metodun iki aşaması mevcuttur. Birinci aşamada Kohonen'in özörgütlenmeli harita ağı ile kümeleme işlemi yapılmaktadır. Diğer aşamada ise çok katmanlı ileri beslemeli yapay sinir ağı kullanılarak, kişilerin demografik bilgileriyle hasar miktarları arasında olduğu varsayılan fonksiyona yakınsanmaya çalışılmıştır. Modelin uygulanabilirliğinin gösterilmesi için ve sonuçlarının test edilmesi açısından bir Türk sigorta firmasından alınan verilerle uygulama yapılmıştır. Uygulamanın sonucunda, firmada kullanılan sistemin sonuçları ile önerilen modelin sonuçları arasındaki yüksek korelasyona bakılarak, önerilen model, firmaya bir karar destek sistemi olarak önerilmiştir.

Modelin şu anda sadece 9 değişkeni kullandığı düşünülürse, daha güvenilir sonuçların elde edilmesi için kişilere ait daha fazla değişken modele eklenebilir. Böylece hedef değerlerle, modelin önerdiği primler arasındaki korelasyon arttırılabilir. Ayrıca kimi durumlarda doktorlar tarafından analiz edilen muayene raporlarının sonuçları sayısallaştırılarak modele eklendiği takdirde, firmanın önerdiği primler ile model tarafından önerilen primler arasındaki korelasyon arttırılabilir.

İleriye dönük olarak, literatürde kullanılan diğer alternatif risk primi hesaplama yöntemleri (Daengdej, vd., 1999) ile bu tezde önerilen model mukayese edilebilir. "Olay bazlı sonuç çıkarma" yöntemini kullanan Daengdej, vd, 1999'da yayınladıkları makalede firmanın kullandıklarını puanlama yönteminin çok gerisinde kaldıklarını açıklamışlardır. Bunun sebebini ellerindeki verinin %90'ının sıfır hasarlı olmasını göstermişlerdir. Bu tezde kullanılan veri kümesinin de %40'ı sıfır hasarlıdır ve firmanın puanlama yöntemi ile 0,6'lık bir korelasyon yakalanmıştır. Daha kesin bir sonuç çıkarabilmek için daha hassas bir analiz yapılması gereklidir. Ancak yine de elde edilen sonuçlar, bu tez de önerilen modelin alternatif risk hesaplama yöntemlerine göre güçlü yönleri olduğu konusunda güçlü ip uçları içermektedir.

KAYNAKLAR

Acinan, H., 1988. Sigortaya Giriş, Can Matbaası, İstanbul.

Ahalt, S.C., Krishnamurthy, A.K., Chen, P. ve Melton, D.E., 1990. Competetive learning algorithms for vector quantization, *Neural Networks*, **3**, 277-290.

Akmut, Ö., 1980. Hayat Sigortası, Ankara Üniversitesi Siyasal Bilgiler Fakültesi Yayınları No: 447, Ankara.

Battiti, R. ve Masulli, F., 1990. BFGS Optimization for faster and automated supervised learning. *INCC 90 Paris, International Neural Network Conference*, **2**, 757-760.

Daengdej, J., Lukose, D. ve Murison, R., 1999. Using statistical models and case-based reasoning in claims prediction: experince from a real-world problem, *Knowledge-Based Systems*, **12**, 239-245.

Denenberg, H., Eilers, R., Hoffman, G., Kline, C., Melone ve J., Snider, H., 1964. Risk and Insurance, Prentice-Hall Inc., New Jersey.

Dinsdale, W.A., 1963. Elements of Insurance, 3rd Ed., Sir Isaac Pitman & Sons, London.

Elmas Ç., 2003. Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama), Seçkin Yayınevi, Ankara.

Fahlman, S.E., 1988. An empirical study of learning speed in back-propagation Networks, Technical report, **CMU-CS-88-162**.

Fletcher, R., 1975. Practical methods of optimization, John Wiley & Sons, New York.

Gurney, K., 1997. An Introduction to Neural Networks, UCL Press, London.

Hagan, M.T., Demuth, H.B. ve Beale, M.H., 1996. Neural Network Design, PWS Publishing, Boston, MA.

Hassoun, M., 1995. Fundamentals of Artificial Neural Networks, MIT Pres, Cambridge, MA.

Hsu, C.-Y. ve Wu, H.-E., 1992. An improved algorithm for Kohonen's self organizing feature maps, *Proc. Int. Joint Conf. on Neural Networks*, Baltimore, MD, 322-335.

- Jacobs, R.**, 1988. Increased rates of convergence through learning rate adaptation, *Neural Networks*, **1**(4), 295-307.
- Jacod, J. ve Protter, P.**, 2003. Probability Essentials, 2nd. ed., Springer-Verlag, Berlin.
- Johansson, E.M., Dowla, F.U. ve Goodman, D.M.**, 1991. Back-propagation learning for multi-layer feed-forward neural networks using the conjugate gradient method, *International Journal of Neural Systems*, **2**(4), 291-302.
- Levenberg, K.**, 1944. A method for the solution of certain non-linear problems in least squares, *Quart. Appl. Math.*, **2**, 164-168.
- Lu, S.-Y., Hernandez, J.E. ve Clak, G.A.**, 1991. Texture segmentation by clustering of Gabor feature vectors, *Proc. Int. Joint Conf. on Neural Networks*, Seattle, WA, Vol. 1, 683-687.
- Mangiameli, P., Chen, S.K. ve West, D.**, 1996. A Comparison of SOM Neural Network and Hierarchical Clustering Methods, *European Journal of Operational Research*, **93**, 402-417.
- Marquardt, D.W.**, 1963. An Algorithm For Least-Squares Estimation of Nonlinear Parameters, *J. Soc. Indust. Appl. Math.*, **2**(2), 431-441.
- Masters. T.**, 1993. Practical Neural Network Recipes in C++, Academic Press, San Diego.
- McCulloch, W.S. ve Pitts, W.H.**, 1943. A logical calculus of ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, **5**, 115-133.
- Mehr, R.I., Cammack, E.**, 1966. Principles of Insurance, 4th ed., Richard D. Irwin Inc., Illinois.
- Mehr, R.I., Hedges, B.A.**, 1963. Risk Management in The Business Enterprise, Richard D. Irwin Inc., Illinois.
- Minsky, M. ve Papert, S.**, 1969. Perceptron, MIT Press, Cambridge.
- Nour, M.A. ve Madey, G.R.**, 1996. Heuristic and Optimization approaches to Extending the Kohonen Self Organizing Algorithm, *European Journal of Operational Research*, **93**, 428-448.
- Outreville, J.F.**, 1998. Theory and Practice of Insurance, Kluwer Academic Publishers, Dordrecht.
- Öztemel, E.**, 2003. Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.
- Panjer, H.H. ve Willmot, G.E.**, 1992. Insurance Risk Models, Society of Actuaries, Schaumburg.

- Riedmiller, M., Braun, H.**, 1993. A Direct Method for Faster Backpropagation Learning: The RPROP Algorithm, *Proceedings of the ICNN*, San Francisco.
- Rosenblatt F.**, 1958. The Perceptron: a probabilistic model for information storage & organization in the brain, *Psychological Review*, **65**(6), 386-408.
- Ross, S.**, 2002. A First Course in Probability, 6th ed., Prentice-Hall Inc., New Jersey.
- Rumelhart, D.E. ve McClelland, J.L.**, 1986. Parallel distributed processing: Explorations in the microstructure of cognition, MIT Press, Cambridge.
- Tettamanzi, A., Tomassini, M. ve Janßen, J.**, 2001. Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems, Springer-Verlag, Berlin.
- Tollenaere, T.**, 1990. Supersab: Fast adaptive backpropagation with good scaling properties, *Neural Networks*, **3**(5), 561-573.
- Wilson, D.R., ve Martinez T.R.**, 2000. The Inefficiency of Batch Training for Large Training Sets, *Proceedings of the International Joint Conference on Neural Networks (IJCNN2000)*, Como, 24-27 Temmuz, 113-117 (İtalya).

ÖZGEÇMİŞ

Bora Çekyay, 25/04/1978 tarihinde İstanbul'da dünyaya geldi. İlkokulu Kazım Karabekir İlköğretim Okulu'nda, ortaokulu, Bakırköy Ortaokulu'nda tamamladı. 1991 yılında, ortaokuldan sonra girdiği meslek liselerine giriş sınavı ile Maçka Anadolu Teknik Lisesi'nin Bilgisayar Bölümü'ne gitmeye hak kazandı. 1995 yılında bu okuldan mezun olduktan sonra, üniversite eğitimi için Trakya Üniversitesi'nin Bilgisayar Mühendisliği Bölümü'ne kayıt oldu. Ancak bir yıl sonra bu okulu bıraktı ve tekrar üniversite sınavına girdi. Bu sefer İstanbul Teknik Üniversitesi'nin Endüstri Mühendisliği Bölümünü kazandı. 2003 yılında bu okulda lisans eğitimini tamamladıktan sonra yine aynı okulda Endüstri Mühendisliği Yüksek Lisansı yapmaya başladı. 1 Mart 2004 tarihinde araştırma görevlisi olarak İTÜ Endüstri Mühendisliği Bölümü'nde akademisyenliğe ilk adımını attı. Halen aynı bölümde araştırma görevlisi olan Bora Çekyay, çok iyi derecede İngilizce bilmektedir.