Yapay Sinir Ağları İçin Net Platformunda Görsel Bir Eğitim Yazılımının Geliştirilmesi

Kerim Kürşat ÇEVİK[1], Emre DANDIL[2]

[1] Bor Meslek Yüksekokulu, Niğde Üniversitesi, Niğde, Türkiye
[2] Meslek Yüksekokulu, Bilecik Üniversitesi, Bilecik, Türkiye
kcevik@nigde.edu.tr, emre.dandil@bilecik.edu.tr
(Geliş/Received: 08.12.2011; Kabul/Accepted: 28.01.2012)

Özet— Bu çalışmada, yapay sinir ağlarının eğitiminde kullanılmak üzere C#.NET programlama dili ile görsel arayüze sahip olan bir eğitim yazılımı geliştirilmiştir. Gerçekleştirilen yapay sinir ağı sınıflandırma yazılımı rakam görüntülerinin sınıflandırılması örneğine uygulanmış ve başarılı sonuçlar alınmıştır. Ayrıca çalışmada, yapay sinir ağları konusunun önemi anlatılmış ve öğrenebilen yazılımların bilgisayar teknolojilerindeki avantajlarına değinilmiştir. Yapay sinir ağlarının eğitilmesi aşamasında kullanılan mevcut yazılımların kullanım karmaşıklığının ve zorluğunun önüne geçmek için geliştirilen yazılımda rakam örneklerinin tanınmasında geriye yayılım algoritmasından yararlanılmıştır. Yazılım ile oluşturulan eğitim ortamı üzerinden sonuçlar için başarı analizi yapılabilmekte ve performansı ölçülebilmektedir. Yazılım yapay sinir ağları ile çözülebilecek uygulamalarda ve akademik çalışmalarda kullanılacak yapıda tasarlanmıştır.

Anahtar Kelimeler— Yapay sinir ağları, sınıflandırma, c#.net, arayüz, karakter tanıma

Development of Visual Educational Software for Artificial Neural Networks on .Net Platform

Abstract— In this study, educational software which is visual interface with C#. NET programming language was developed for use in training artificial neural networks. Performed an artificial neural network classification software has been implemented and successful results in the instance of the number of classification. Also, in this paper, the importance of the issue artificial neural networks is described and software that can learn the benefits of computer technology is mentioned. Existing software which used during the training of artificial neural networks to avoid the complexity and difficulty of access in developed software the back-propagation algorithm was used identification of samples of numbers. Comments can be made to the results generated by the software via the medium training and analyze of the performance achievement can be measured. The software is structure designed to be used in applications that can be solved with artificial neural networks and academic studies.

Keywords— Artificial neural network, classification, c#.net, user interface, character recognition

1. GİRİŞ

Günlük hayatımızda vazgeçilmez bir yere sahip olan bilgisayarın, artık insanlar gibi hem karar verebilmesi hem de öğrenmeyi gerçekleştirmesi kullanım alanlarını oldukça genişletmiştir. Matematiksel olarak bir formülle gösterilemeyen ve insan tarafından çözülmesi zor olan problemlerin yapay zekâ yöntemleri ile bilgisayarlar tarafından çözülebilmesi, uygulamalar açısından vazgeçilmez bir durum haline gelmiştir. Yapay zekâ uygulamaları zeki sistemler olarak da adlandırılırlar. Zeki sistemlerin en temel özellikleri, olaylara ve problemlere çözümler üretirken veya çalışırken bilgiye dayalı olarak karar verebilme özelliklerinin olmasıdır [1]. Bu mekanizmaların taklit edilmesi, Yapay Sinir Ağları (YSA) gibi farklı teknolojilerin doğmasına neden olmuştur [2].

Yapay sinir ağları, insan beyninin çalışma yapısı taklit edilerek bilgisayarlar üzerinde modellenmesi sonucu ortaya çıkmıştır. İnsan beyninde olduğu gibi sinirlerin birbirine bağlı olduğu bir sinir ağı üzerinde; öğrenme işlemi, öğrenilen bilgilerin saklanması ve bu bilgiler sayesinde çıkarsama yapılması sağlanmaktadır [3]. Gerçek zamanda öğrenme, uyarlama ve genelleme yeteneği ile birlikte yapısı gereği çok girişli, çok çıkışlı sistemlere de uygulanabilmesi nedeniyle kontrol alanında tercih edilmektedir. Ayrıca paralel yapısı nedeniyle hatayı azaltma ve hızlı bilgi işleme yeteneği de arzu edilen bir özelliktir [4]. Günümüzde sistemlerin modellenmesi ve

kontrolü için YSA uygulamaları üzerine birçok araştırma yapılmaktadır [5].

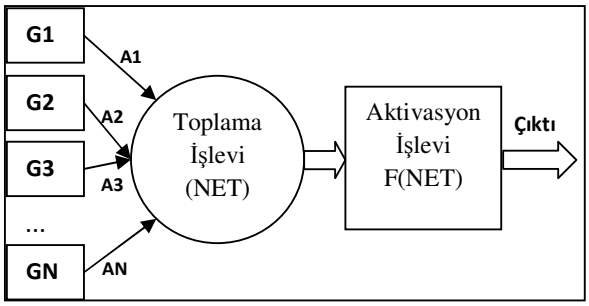
YSA alanında en çok tercih edilen yazılımlardan birisi Mathworks Inc. tarafından geliştirilen MATLAB ve bunun YSA araçlarıdır. Bunun dışında SNNS (Stuttgart Neural Network Simulator), FANN (Fast Artificial Neural Network Library), JOONE (Java Object Oriented Neural Engine) gibi başka alternatif yazılımlar da vardır. Bu gibi yazılımlar YSA'nın geliştirilmesi ve değerlendirilmesi için bir ortam sağlamaktadır. Ancak bu yazılımların kullanıcı tabanlı geliştirmeye kapalı olması, kullanımlarının zor ve karmaşıklığı gibi dezavantajları vardır. Fakat bir eğitim yazılımından istenen animasyon ve grafiklerle simülasyon ortamı oluşturmak, öğrenme işlemi esnasında grafiksel ve görsel geri dönütler sağlamak, değişik kavramlarla kavramaya yardımcı olmak gibi temel noktalardır.

Bu çalışmada YSA'nın öğrenme, genelleme, özelliklerinin kolayca görülebileceği bir eğitim yazılımı geliştirilmiştir. Yazılımda C#.NET programlama dili kullanılarak bir kullanıcı arayüzü tasarlanmıştır. Geliştirilen eğitim yazılımı ile rakam karakterlerinden alınmış görüntü örneklerini kullanılarak YSA'nın Geriye Yayılım algoritması ile eğitim gerçekleştirilmekte ve diğer rakam görüntüleriyle test edilmektedir. Bunun yanında, yazılım ekranından sisteminin performansı, değişiklikler ve sonuçlar grafiksel olarak gözlenebilmektedir. Ayrıca yazılım yapay sinir ağlarının öğrenme ve karar verme özellikleri; gelişmekte olan yazılım platformlarından C#.NET teknolojisi ile her türlü sınıflandırma probleminde uygulanabilecek düzeyde tasarlanmıştır.

2. YAPAY SİNİR AĞLARI

Yapay zekâ (YZ), bir bilgisayarın ya da bilgisayar denetimli bir makinenin, genellikle insana özgü nitelikler olduğu varsayılan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak tanımlanmaktadır [6].

Sınıflandırma problemlerinde YZ tekniklerinden genel olarak YSA kullanılır. Yapay Sinir Ağları, insan beyninin işleyişini taklit ederek yeni sistem oluşturmaya çalışan yaklaşımlardır [7]. Beynimizdeki biyolojik sinir hücrelerinin yapısı temel alınarak YSA yapısı oluşturulur. YSA'nda aynen beynimizde olduğu gibi öğrenme ve öğrenilen bilgilere göre karar verme mekanizmaları bulunur [8,9,10]. YSA, birbirine hiyerarşik olarak bağlı, Şekil 1.' de gösterilen yapay hücrelerden (nöron) meydana gelmektedir. Yapay sinir hücreleri proses elemanı olarak adlandırılmaktadır [10]. Her proses elemanının Şekil 1.' deki gibi 5 farklı birimi vardır.



Şekil 1. Bir Yapay Sinir Hücresi Örneği

G1, G2, G3, ..., GN ile gösterilen *Girdiler* bir YSA' nın girişleri olarak bilinir. A1, A2, A3,....AN *Ağırlıklar* olarak tanımlanır ve yapay sinir hücresine gelen bilginin etkisini gösterir. *Toplama İşlevi Fonksiyonu (NET)* ise bir sinir hücresine gelen net bilgiyi hesaplar. Bu net değeri bulmak için değişik fonksiyonlardan yararlanılır ancak en çok kullanılanı Denklem(1)' de gösterilen toplam ağırlığı bulan ifadedir. Burada G_i i. giriş değerini, A_i ise bu giriş değerinin ağırlığını ve *NET* ise fonksiyonun toplam değerini gösterir.

$$NET = \sum_{i}^{n} G_i \, A_i \qquad (1)$$

Yapay sinir hücresindeki *Aktivasyon İşlevi Fonksiyonu(FNET)* da hücreye gelen net girdileri hesaplayarak üretilecek olan çıktı değerini belirler. Aktivasyon fonksiyonlarının lineer fonksiyon, step fonksiyon, sinüs fonksiyonu, eşik değer fonksiyonu, sigmoid fonksiyonu ve hiperbolik tanjant fonksiyonu olmak üzere değişik gösterimleri mevcuttur. Uygulamaların çoğu Çok Katmanlı Algılayıcı biçiminde tasarlandığı için genelde olarak bu çalışmalarda Sigmoid Fonksiyonu tercih edilmektedir [10]. Bu fonksiyon Denklem(2)' deki formülle gösterilir. Bu denklemde *F(NET)* aktivasyon fonksiyonunu temsil eder. Bu çalışmada geliştirilen yazılım, problemlerin çözümünde Sigmoid fonksiyonundan yararlanacak biçimde tasarlanmıştır.

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

Proses elemanındaki *Çıktı* değeri ise aktivasyon fonksiyonu tarafından belirlenen çıktı değeridir. Üretilen çıktı ya başka hücreye ya da kendisine tekrar gönderilerek değerlendirilir.

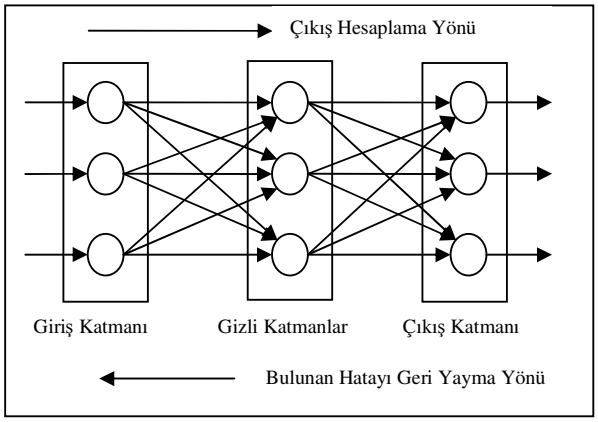
Temel olarak bir YSA'nın görevi, kendisine gösterilen giriş setine karşılık bir çıkış seti belirlemektir. Bunu gerçekleştirebilmek için ağ, ilgili problemin örnekleri ile eğitilerek (öğrenme), o problemle ilgili istenenleri çözebilme yeteneğine kavuşturulur [11].

YSA sinir hücrelerinin bir araya gelmesiyle oluşur. Bu işlem rastgele olarak yapılmamaktadır. Genelde hücreler,

3 katman halindedir ve her katman kendi içinde paralel halde ağı oluştururlar. Bu katmanlar;

- *Giriş Katmanı:* Bu katmandaki hücreler giriş bilgilerini gizli katmana ulaştırmakla görevlidir.
- *Gizli Katman:* Giriş katmanından gelen bilgiler işlenerek çıkış katmanına gönderilir. Bir ağda problemin durumuna göre, birden fazla gizli katman olabilmektedir.
- *Çıkış Katmanı:* Bu katmandaki hücreler gizli katmandan gelen bilgiyi çıkış katmanına gönderirler. Bu kısımda üretilen çıkışlar problemin çözümünü içermektedir [12].

Bir YSA' nın giriş katmanı, gizli katman ve çıkış katmanından oluşan genel ağ yapısı Şekil 2.' de gösterilmiştir.



Şekil 2. YSA'nın genel ağ yapısı

Genelde YSA ağ yapısında, verilen bir girdi setine karşılık çıktı değerleri verilerek belirtilen öğrenme kuralına göre ağırlık değerleri otomatik olarak değiştirilmektedir. Eğitim verisinin tamamlanmasından sonra eğitilmiş olan ağ, ağırlık değerlerinin son durumuna göre, verilen herhangi bir veri setinin sonucunu tahmin edebilmektedir. Yapay sinir ağı bir dizi sinir hücresinin ileri sürümlü ve geri beslemeli bağlantı şekilleri ile birbirine bağlanması ile oluşur [12,13].

YSA' da hücre eleman bağlantılarının ağırlık değerlerinin belirlenmesi işlemine *ağın eğitilmesi* denir. Başlangıçta bu değerler rastgele belirlenir. Ancak daha sonra çıktı değerlerine göre bu ağırlık değerleri tekrar tekrar değiştirilerek gerçek durumuna erişir. Buna da *ağın öğrenmesi* denilir. Ağın öğrenmesine referans olarak gösterilen farklı öğrenme modelleri vardır. En çok kullanılanları;

- Algılayıcılar, Perceptron ve Adaline
- Çok Katmanlı Algılayıcı Modelleri (Hatayı geriye yayma modelleri-backpropagation)
- Vektör Kuantizasyon Modelleri (LVQ)
- Kendini organize eden model (SOM)
- Adaptif Rezonans Teorisi (ART)
- Hopfield Ağları
- Counterpropagation Ağı
- Neocognitron Ağı
- Bolztman Makinesi
- Probabilistic Ağlar
- Elman Ağı
- Radyal Temelli Ağlar

şeklinde açıklanabilir [14].

Bu alandaki çalışmaların çoğunda yapay sinir ağlarının, doğrusal olmayan, çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek verilerin olması ve problemin çözümü için özellikle bir matematik modelin ve algoritmanın bulunmaması hallerinde yaygın olarak kullanıldıkları görülmektedir.

Bu amaçla geliştirilmiş ağlar genel olarak örüntü tanıma, sınıflandırma, doğrusal olmayan sinyal işleme, ilişkilendirme veya örüntü eşleştirme, doğrusal olmayan sistem modelleme, zaman serileri analizleri, sinyal filtreleme, zeki ve doğrusal olmayan kontrol, veri sıkıştırma gibi fonksiyonları yerine getirirler.

Bunların ötesinde günlük hayatta kullanılan finansal uygulamalarından mühendisliğe ve tıp bilimine kadar birçok alandan bazıları ise şöyle sıralanabilir. Bunlar veri madenciliği, optik karakter tanıma ve çek okuma, bankalardan kredi isteyen müracaatları değerlendirme, ürünün pazardaki performansını tahmin etme, kredi kartı hilelerini saptama, zeki araçlar ve robotlar için en uygun değer rota belirleme, güvenlik sistemlerinde konuşma ve parmak izi tanıma, borsa ve para yönetimi, kalite kontrolü, bilgisayar oyunlarındaki zeki karakterlerin yaratılması, radar ve sonar sinyalleri sınıflandırma, kan hücreleri reaksiyonları ve kan analizlerini sınıflandırma, kanserin saptanması ve kalp krizlerinin tedavisi olarak sınıflandırılabilir [15].

Bu çalışmada geliştirilen yazılımın doğrulanması için gerçekleştirilen rakam tanıma uygulamasında aktivasyon fonksiyonlarından Sigmoid Fonksiyonu'ndan, öğrenme modellerinden ise Hatayı Geriye Yayma Modeli'nden yararlanılmıştır.

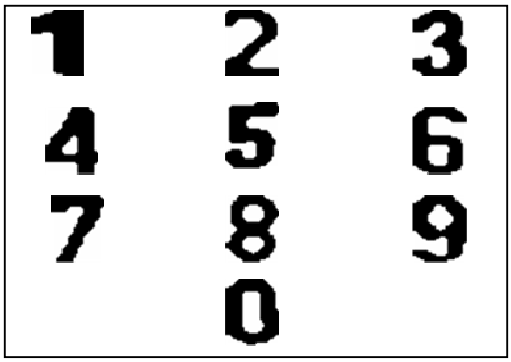
3. GELİŞTİRİLEN YAPAY SİNİR AĞLARI GÖRSEL EĞİTİM YAZILIMI İÇİN RAKAM TANIMA UYGULAMASI

Karmaşık sistemler içerisindeki karakterlerin tanınması ve bu karakterlerden sistem hakkında değerlendirmeler yapılması günümüz teknolojilerinde oldukça sık rastlanan konulardan birisi olmuştur. Karakter tanıma alanında yapılan çalışmalar sonucunda, elle veya dijital baskı ile oluşturulmuş karakterlerin otomatik tanınması konusunda birçok teknik geliştirilmiştir [16,17]. Bu tekniklerden en çok tercih edilenlerden bir tanesi de YSA' dır.

Karakter veya rakam tanıma klasik bilgisayar yöntemleri ile de yapılabilmesine karşın YSA yönteminin klasik yöntemlere göre oldukça üstün tarafları bulunmaktadır. Klasik yöntemlerle karakter tanıma işlemlerinde giriş ile çıkış arasında tam anlamıyla bir uyum olmalıdır. Gürültülerden doğacak sorunlar sonucu olumsuz

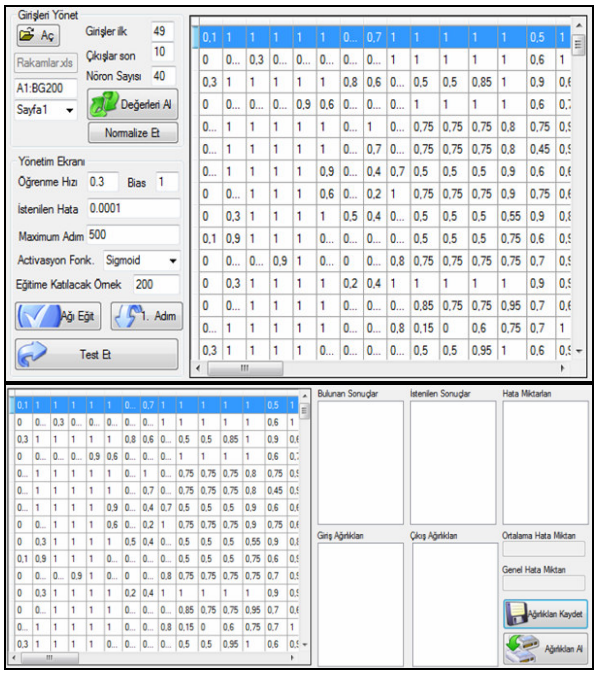
etkileyecektir [18]. Bu gibi sonucu etkileyecek sorunlardan kurtulabilmenin yolu, gürültüdeki bütün pikselleri inceleyebilecek ve bu gürültüleri ihmal edecek yeni bir yöntemdir. Bu yöntem, bilinen karakterlere benzetilebilen fakat gürültü nedeniyle bozuk olan şekilleri önceden öğrendiği şekillerden birine benzetebilen ve gürültüyü ihmal edebilen YSA'lardır [19] Ayrıca işlemlerin hızlı yapılması ve sonuçlardaki başarım oranında YSA' nın ön plana çıkan üstünlükleri görülmektedir [20].

Çalışmalarda birden fazla farklı ağın gruplandırılarak eğitilmesi sıranda genel olarak geri beslemeli bir ağ yapısı tercih edilir [21]. Bunun nedeni giriş ve çıkışlar arasında bir bağlantı kurularak girişi etkileyen bir durumun çıkışa aktarılmak istenmesidir. Bu durum yeni girişleri değiştirdiği için öğrenmenin daha başarılı olduğu gözlenmektedir [22]. Bu çalışmada yazılımın başarımını göstermek için bir rakam tanıma uygulaması gerçekleştirilmiştir. Bu amaç için geliştirilen YSA eğitim yazılımında kullanılan 0 ile 9 arasındaki rakamlara ait her birisinden birer örnek Şekil 3.' te gösterilmiştir.



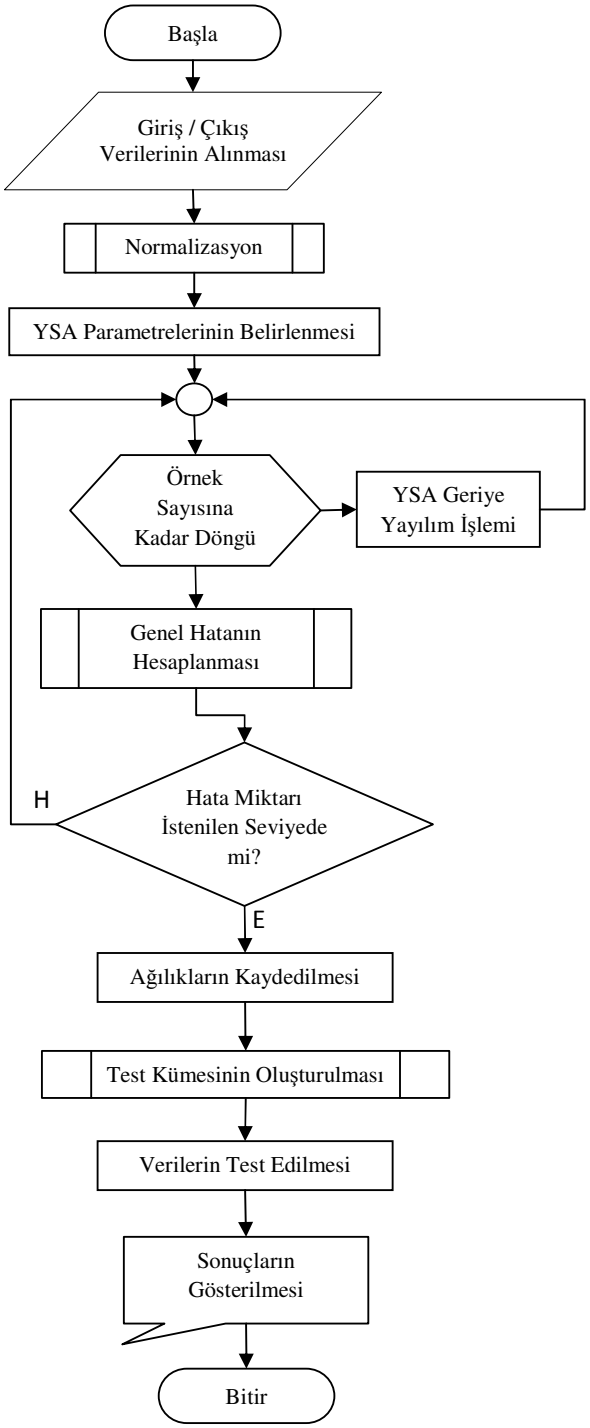
Şekil 3. Tanınması istenen rakamlar sınıfına ait örüntüler

Çalışmadaki rakam tanıma uygulaması için geliştirilen YSA tek gizli katmanlı, ileri beslemeli, hatayı geriye yayma modeline sahip, Sigmoid aktivasyon fonksiyonunu kullanacak şekilde çok katmanlı algılayıcı biçiminde tasarlanmıştır. Eğitim ve test aşamaları olmak üzere iki modülden oluşan yazılımın kullanıcı arayüzü Şekil 4.' de gösterilmiştir.



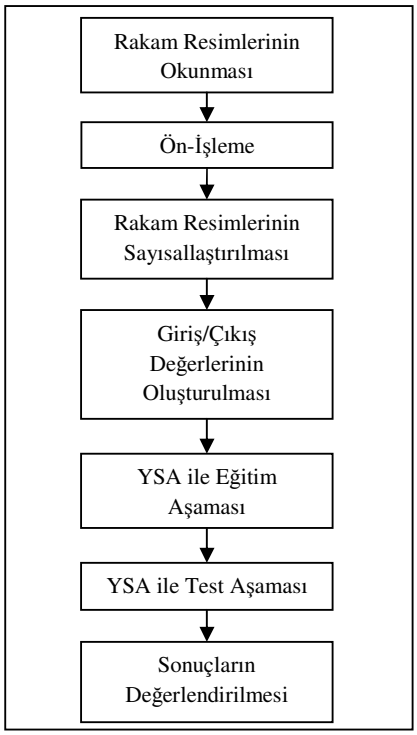
Şekil 4. Geliştirilen YSA Eğitim Yazılımın Arayüzü

Geliştirilen eğitim yazılımı, YSA'nın kullanım alanlarının genişliği göz önüne alınarak, kullanıcıların YSA'nın öğrenme kabiliyetini her türlü probleme uygulayabilecek şekilde tasarlanmıştır. YSA' nın eğitim ve test aşamasındaki temel işlemler bu arayüz üzerinden yönetilmektedir. Geliştirilen yazılımın genel akış şeması Şekil 5.' te gösterilmiştir.



Şekil 5. Uygulama yazılımının akış şeması

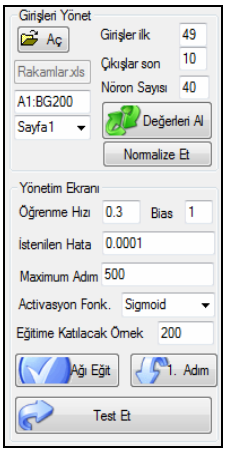
Tasarlanan bu yazılım akademik çalışmalarda da kullanılabilecek ve sonuçlar üzerinde değerlendirmeler, başarı analizi, performans ölçümü yapılabilecek yapıdadır. Bu özelikler bir YSA eğitim yazılımından istenen temel özelliklerdir. Ayrıca YSA eğitim yazılımı geliştirilebilecek ve gerektiğinde güncellenebilecek şekilde tasarlanmıştır. Yazılım üzerinde uygulama yapılırken işlem sistemin adımlarını, çalışma yapısını gösterir blok diyagram Şekil 6.' da verilmiştir.



Şekil 6. Sistemin Blok Diyagramı

3.1. Giriş ve Çıkış Değerlerinin Belirlenmesi

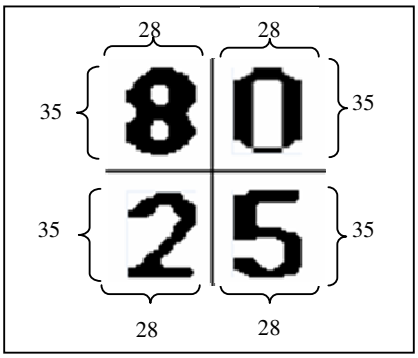
Çalışmada öncelikle tanınması istenilen her bir rakam sinir ağına öğretilmelidir. Bunun için arayüz üzerinde giriş çıkış değerleri ve parametreleri Şekil 7' de de gösterildiği gibi belirlenir.



Şekil 7. Giriş çıkış parametrelerinin belirlenmesi

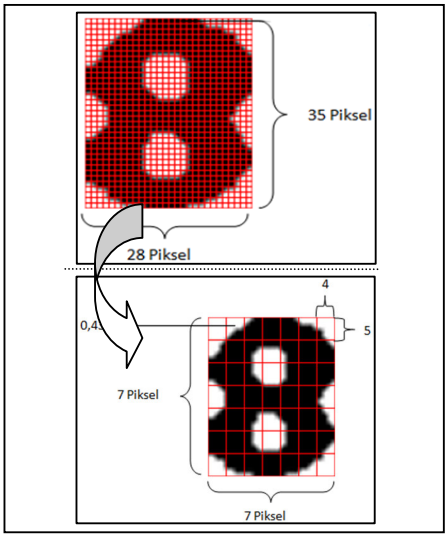
Yazılımda rakam karakterlerinin tanınması için her birisi 28x35 pikselden oluşan rakamlar bu çalışmaya özgü olarak Şekil 8. üzerinde gösterildiği gibi giriş değerlerini oluşturmak üzere resim dosyası halinde kaydedilirler. Resim karakterlerinin boyutu her çalışma için istenilen şekilde belirlenebilir. Diğer aşamada, rakam karakterlerinden oluşan resimler, rakamların daha kolay bir şekilde sınıflandırılabilmesi için Siyah=1 ve Beyaz=0

renk değerlerinden oluşacak şekilde(binary) bir ön-işlemden geçirilirler.



Şekil 8**.** 28x35 boyutunda örnek rakam resimleri

Uygulama için her rakam resminin boyutu 28x35 piksel olarak alındığında 28x35=980 tane giriş değerinin ağa verilmesi gerekmektedir. Ancak bu şekilde çok fazla giriş değerinin olması istenen bir durum değildir. Bunun yerine ağın daha iyi performans için Şekil 9.' da gösterildiği gibi her bir resim 4x5'lik bölümlere ayrılmıştır.



Şekil 9. Rakam resimlerinden giriş değerlerinin oluşturulması

4x5 boyutlarındaki resim bölümlerinde bulunan piksellerin ortalama renk değeri ağa giriş değeri olarak verilmiştir. Her bölümde ortalama renk değeri bulunurken Denklem (3)' teki formül kullanılır. Her bir bölümde 20 piksel vardır.

$$ORD = \frac{\sum_{i=1}^{20} R_i}{20} \qquad (3)$$

Bu denklemde *ORD*, ortalama renk değeri; *Ri* ise *i.* pikselin renk değeridir. Rakam karakterlerini oluşturan resimler 4x5 boyutlarından oluşan bölümlere ayrıldıktan sonra 28x35 boyutundaki her bir rakam resminde 7x7=49 giriş değeri oluşacaktır. Bu girişler YSA' nın giriş değerlerini oluşturur. 4x5' lik bölümlerdeki renk ortalamaları alınarak oluşturulan 49 girişe karşılık, 0-9 arasındaki 10 farklı rakama ait 0, 1, 2, 2, 4, 5, 6, 7, 8, 9 çıkış değerlerini gösteren örnek veriler Tablo 1.' de gösterilmiştir. Bu çıkışlar ise YSA çıkış değerleridir ve her birisi bir sınıfı temsil eder.

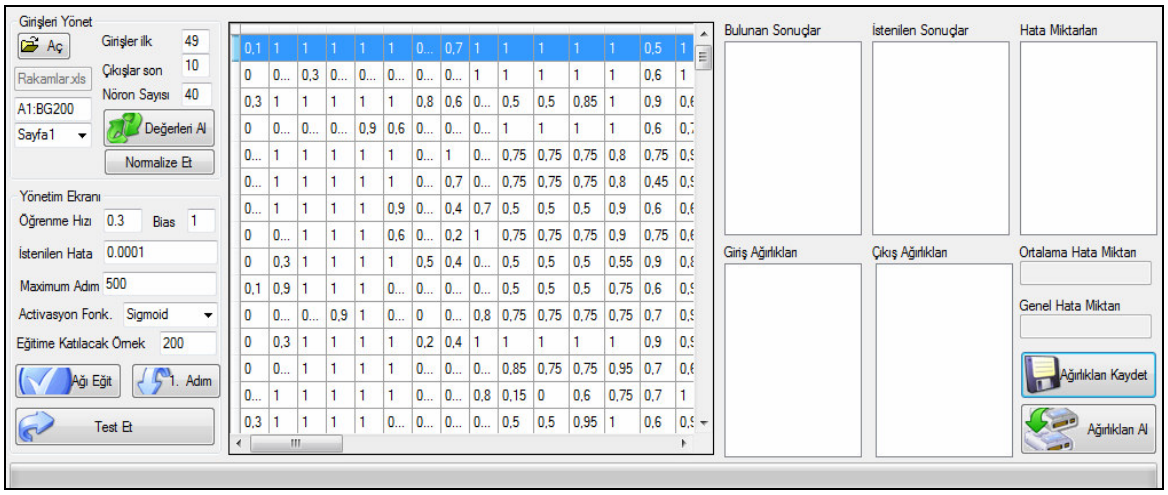
Tablo 1. YSA'nın Giriş ve Çıkış verileri

GİRİŞLER							**ÇIKIŞLAR**										**Rakam**
G1	**G2**	**G3**	**...**	**G47**	**G48**	**G49**	**Ç0**	**Ç1**	**Ç2**	**Ç3**	**Ç4**	**Ç5**	**Ç6**	**Ç7**	**Ç8**	**Ç9**	
1	0.4	0.2	**...**	0.9	0.8	0.3	**1**	0	0	0	0	0	0	0	0	0	**0**
1	0.6	0.6	**...**	1	1	0.4	0	**1**	0	0	0	0	0	0	0	0	**1**
0.1	1	1	**...**	1	1	0.1	0	0	**1**	0	0	0	0	0	0	0	**2**
0	0.2	0	**...**	0	0	0.35	0	0	0	**1**	0	0	0	0	0	0	**3**
1	0.1	0.1	**...**	0.6	0.8	0.3	0	0	0	0	**1**	0	0	0	0	0	**4**
1	0.5	0.1	**...**	1	1	0.4	0	0	0	0	0	**1**	0	0	0	0	**5**
0.1	0.2	0.3	**...**	1	1	0.1	0	0	0	0	0	0	**1**	0	0	0	**6**
0	0.2	0	**...**	0	0	0.2	0	0	0	0	0	0	0	**1**	0	0	**7**
1	1	0.2	**...**	0.6	0.8	0.7	0	0	0	0	0	0	0	0	**1**	0	**8**
0.1	0.8	1	**...**	0.9	1	0.1	0	0	0	0	0	0	0	0	0	**1**	**9**

(...)*tabloya yazılamayan ancak bu aralıkta giriş değerlerinin olduğunu göstermektedir*

3.2. Eğitim Aşaması

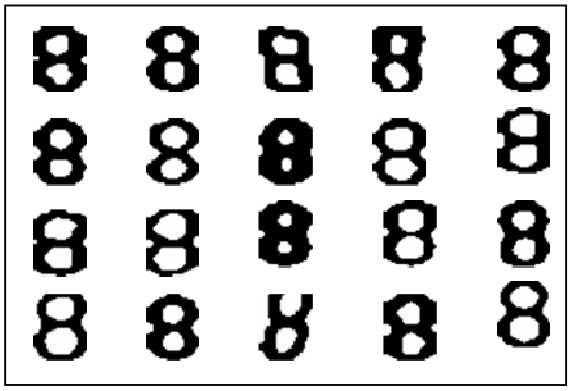
Giriş çıkış değerleri belirlendikten sonra eğitimin gerçekleştirilmesi için YSA eğitim yazılımına elimizdeki problemin giriş ve çıkış değerlerinin sayısal olarak kaydedilmiş olduğu harici dosyanın programa yüklenmesi gerekmektedir. Bu aşamadan sonra yazılım üzerinde YSA eğitim parametre değerlerinin Şekil 10.' daki gibi uygun bir şekilde belirlenmesi gerekmektedir.



Şekil 10**.** Tasarlanan YSA Eğitim Programı ve eğitim verilerinin girilmesi

Yazılıma her rakam için 20 adet örnek, toplamda 10 farklı rakam için 200 adet örnek eğitim seti için hazırlanmış ve bir harici dosyaya düzenli bir şekilde kaydedilmiştir. Bu veri setleri bu probleme için özeldir. YSA eğitimi için geliştirilen bu yazılım başka çalışmalar içinde kullanılabilecek şekilde tasarlanmıştır. Sayısal verileri bulunduran dosya içerisinde başka çalışmalar için gerekli veriler girildiğinde yazılım bu çalışmalar içinde kullanılabilir.

Çalışmada her karakter için farklı yapıda resimler kullanılmıştır. Örnek olarak 8 rakamı için kullanılan 20 farklı yapıda ve biçimde resim örüntüleri Şekil 11.' de gösterilmektedir. Diğer rakamlar içinde birbirinden farklı resim bulunmaktadır.



Şekil 11. 8 rakamına ait 20 farklı resim örneği

Yazılımın eğitim aşamasında ana form ekranındaki butonların hangi amaçlar için kullanıldığı ise aşağıda açıklanmıştır.

Nöron Sayısı: Ara katmandaki nöronların kaç tane olacağını kullanıcı programa girebilir. En ideal nöron sayısı genelde deneme yanılma yöntemiyle bulunur.

Normalize Et: YSA çalışma mantığı verilerin 0-1 aralığında olmasına dayalıdır. Eğer giriş ve çıkış değerlerimiz 0-1 aralığında değilse bunun normalizasyonu yapılmaktadır.

Öğrenme Hızı: Eğitimde kullanılan geriye yayılım algoritmasının öğrenme katsayısını (□) belirtmektedir.

Bias: Giriş değerlerinin yanında tüm nöronlara destek sağlayacak bir eşik girişinin verilmesi gerekmektedir. Standart değeri 1'dir. Fakat kullanıcı isterse bu değeri değiştirebilmektedir.

İstenilen Hata: İstenilen hata değerine ulaşıldığında eğitimin durdurulma ölçütünü belirler.

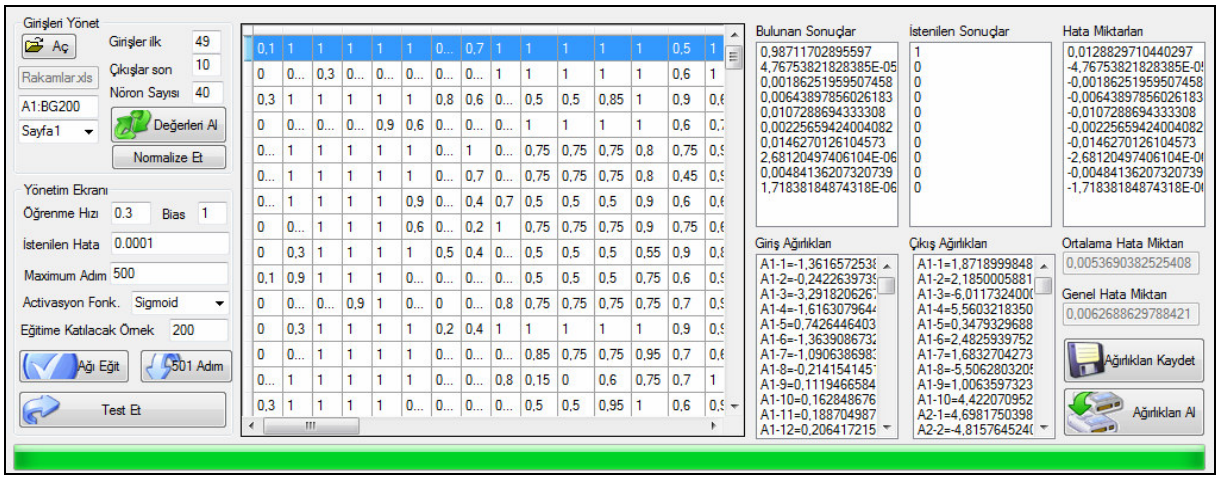
Maksimum Adım Sayısı: Sistemin eğitim işlemini kaç defa tekrarlayacağını kullanıcı belirler.

Aktivasyon Fonksiyonu: Her bir nöron için kullanılacak aktivasyon fonksiyonu (Sigmoid,TanH) kullanıcının tercihine bırakılmıştır. Problemin yapısına göre bu aktivasyon fonksiyonları çözümün bulunmasını kolaylaştırabilir.

Eğitime Katılacak Örnek: Harici dosyadan aldığımız verilerin istenilen kadarını eğitime dâhil edebiliriz. Eğitim tamamlandıktan sonra kalan örnekleri de test için kullanabiliriz.

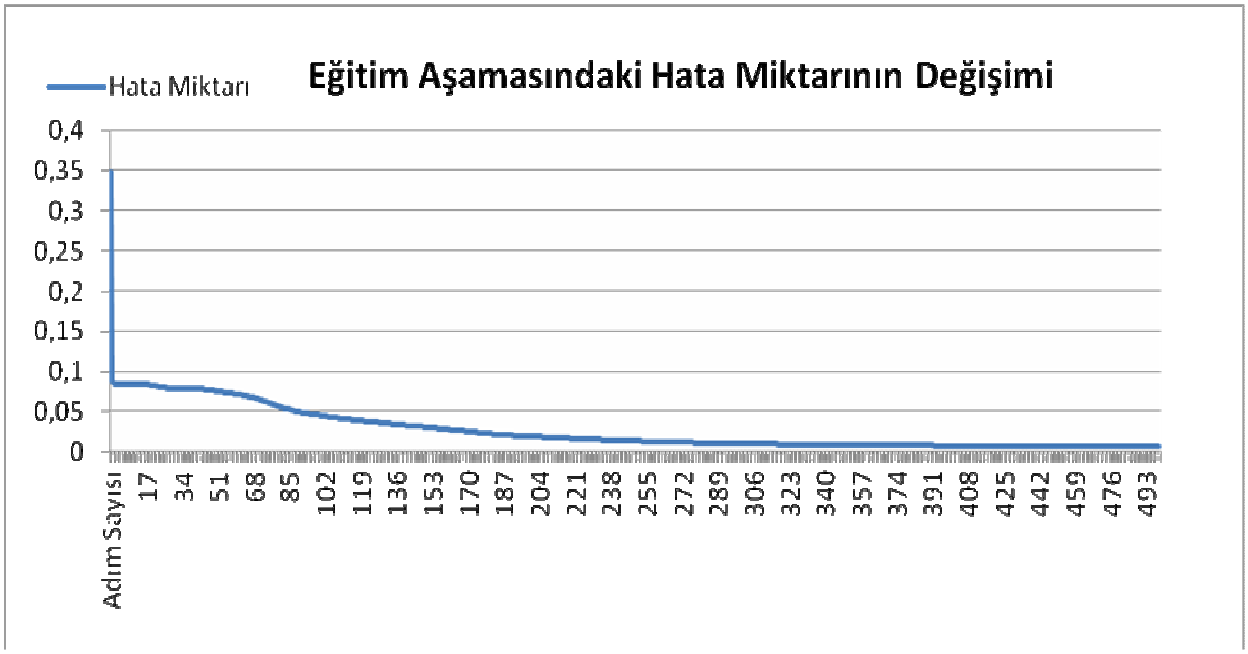
Ağırlıkları Kaydet: Eğitim aşaması bittikten sonra her sinir hücresinde oluşan son ağırlık değerlerinin kaydedilmesini sağlar.
Ağırlıkları Al: Daha önce kaydedilmiş ağırlık değerlerinin yazılıma yeniden yüklenmesi için kullanılır.

Programda giriş / çıkış değerleri ve ağ yapısının parametreleri belirlendikten sonra yazılım arayüzündeki "Ağı Eğit" butonuna basılırsa; sistem istenilen adım sayısı kadar eğitime devam eder ve önceden rastgele belirlenmiş olan ağırlık değerlerini değiştirerek problemin çözümü için en uygun ağ parametrelerini belirler. Bu süre esnasında istenilen hata miktarına ulaşırsa hesaplama işlemini durdurur. Programda rakam karakterlerinin tanınması için Şekil 12.' deki 500 adım sonrası rakam görüntüleri problemi giriş ve çıkış ağırlıklarını belirlemiş ve 0,006 genel hata oranıyla problemi çözülmüştür. Program eğitim aşamasını yaklaşık 3 dakika gibi kısa bir sürede tamamlayarak sonuçlara ulaşmıştır.



Şekil 12. 500 adım sonrası programın genel görüntüsü

Gerçekleştirilen yazılımda YSA' nın eğitim aşaması sonucunda rakam karakterlerinin tanıma işlemi sırasında oluşan hata miktarının adım sayısına(Epoch) göre değişimi Şekil 13.'de gösterilmektedir. Şekil üzerinde de görüleceği üzere yaklaşık 500 adım sonrasında hata miktarı sıfıra yaklaşmaktadır. Bu YSA yazılımları için istenilen bir durumdur. Çünkü bunun gibi yazılımların kısa süre içerisinde hatayı en aza indirgemesi gerekmektedir.



Şekil 13. Eğitim sırasında genel hatanın adım sayısına göre değişimi

3.3. Test Aşaması

Bu aşamada test verilerinin bulunduğu dosya yazılıma yüklendikten sonra "Test Et" butonu kullanılarak rakamların tanınması işlemi gerçekleştirilir. Tablo 2.' de görüldüğü gibi her bir rakam için 20 adet veri eğitim aşamasında kullanılmıştır. Bu çalışma için test aşamasında ise yine her rakam için 20 adet örnek veri sınıflandırma işlemi için kullanılmıştır.

Tablo 2. YSA eğitim yazılımının eğitim ve test aşaması sonuçları

Rakam Karakterleri	Eğitim Aşaması		Test Aşaması		Başarı(%) Sınıflandırma
	Örnek Ad.	Ort. Hata	Örnek Ad.	Ort. Hata	
0	20	0.007	20	0.013	100
1	20	0.006	20	0.012	100
2	20	0.005	20	0.006	100
3	20	0.005	20	0.012	95
4	20	0.006	20	0.009	100
5	20	0.007	20	0.005	100
6	20	0.006	20	0.018	95
7	20	0.005	20	0.013	100
8	20	0.005	20	0.023	90
9	20	0.006	20	0.011	100

Tablo 2. incelendiğinde gerçekleştirilen eğitim yazılımının her bir rakamdan 20 adet olmak üzere toplamda 200 adet test verisini, ortalama hatanın yaklaşık olarak 0.01 ve daha düşük değerlerde %98 oranında başarılı olarak sınıflandırdığı görülmüştür. Yazılımda 6 ve 3 rakamları için başarılı sınıflandırma oranı %95, diğer tüm rakam örüntüleri için ise %100 olarak hesaplanmıştır. Hem eğitim hem de test aşamalarında örnek sayısı artırılarak YSA eğitim aşaması gerçekleştirildiğinde sonuçlar daha da başarılı olacaktır.

Bu durumlardan sonra geliştirilen yeni bir YSA yazılımının doğrulaması başarılı olarak gerçekleştirilmiştir ve yazılımın doğru çalıştığı test edilmiştir.

4. SONUÇLAR

Bu çalışmada, yapay sinir ağlarının eğitilmesi aşamasında kullanılan mevcut yazılımların kullanım karmaşıklığının ve zorluğunun önüne geçmek için .NET platformu kullanılarak bir yapay sinir ağı sınıflandırıcı yazılım geliştirilmiştir. Bu yazılım .NET platformunun en güçlü dillerinden olan C# ile kodlanmıştır. Gerçekleştirilen yapay sinir ağı sınıflandırma eğitim yazılımı rakam görüntülerinin sınıflandırılması örneğine uygulanmış ve başarılı sonuçlar elde edilmiştir.

Yazılım gerekli giriş ve çıkış değerleri doğru olarak belirlendiği zaman tüm çalışmalarda başarılı olarak kullanılabilir. Bu durumda yazılımın önemli bir artısını ön plana çıkarmaktadır.

Eğitim yazılımı kullanıcının YSA tasarımı ve tanıma parametrelerinin değiştirilmesine imkan veren esnek bir yapıya ve bir ara yüze sahiptir. Eğitim yazılımı geriye yayılım algoritması ile YSA'nın eğitimini gerçekleştirmektedir Karakter tanıma değişkenleri ve diğer sonuçlar izlenebilmektedir.

YSA eğitim yazılımı 0-9 arasında rakamları YSA ile tanımak ve sınıflandırmak amacıyla denenmiştir. Yazılımın çalışmasını ve performansını gösteren simülasyon sonuçları kaydedilmiştir. Elde edilen sonuçlar, yazılımın rakamların tanınması bakımından kendisinden beklenen başarımı sağladığı görülmüştür.

Geliştirilen YSA eğitim yazılımı yapay sinir ağları ile çözülebilecek uygulamalarda ve akademik çalışmalarda kullanılacak yapıda tasarlanmıştır. Yazılıma diğer yapay zeka yöntemlerinin de entegre edilerek farklı algoritmalar ile sonuçların karşılaştırılması ve değerlendirilebilecek yapıda olması olumlu yazılımın yönlerini göstermektedir.

KAYNAKLAR

[1] Ç. Elmas, **Yapay Zeka Uygulamaları**, Seçkin Yayınevi, Ankara, 2007.

[2] Ö. Yeloğlu, A. Uğur, "Modern Programlama Platformlarında Yapay Sinir Ağı Yazılımlarının Geliştirilmesi", **Bilgi Teknolojileri Kongresi III (Bilgitek 2004)**, Denizli, 1-8, 2004.

[3] L. Fauset, **Fundamentals of Neural Networks**, Prentice Hall Inc., USA, 1994.
[4] A. Tekin, M. Gökbulut, "Yapay Sinir Ağları ile Asenkron Motorların Hız Kontrolü İçin Bir Eğitim Yazılımın Geliştirilmesi", *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 20(3), 449-453, 2008.
[5] K. S. Narendra, K. Parthasarthy, "Identification and Control of dynamical system using neural Networks", *IEEE Trans. Neural Network*, 1(1), 4-7, 1990.
[6] V. Nabiyev, **Yapay Zeka Problemler**, **Yöntemler, Algoritmalar**, Seçkin Yayıncılık, Ankara, 2005.
[7] C. Öz, R. Köker, S. Çakar, "Yapay Sinir Ağları ile Karakter Tabanlı Plaka Tanıma", **Elektrik Elektronik ve Bilgisayar Mühendisliği Sempozyumu (ELECO'2002)**, Bursa, 1-6, 2002.
[8] N. Allahverdi, **Uzman Sistemler Bir Yapay Zeka Uygulaması**, Atlas Yayın Dağıtım, İstanbul, 2002 .
[9] Ş. Sağıroğlu, E. Beşdok, M. Erler, **Mühendislikte Yapay Zeka Uygulamaları-1 Yapay Sinir Ağları**, Ufuk Yayıncılık, Kayseri, 2003.
[10] E. Öztemel, **Yapay Sinir Ağları**, Papatya Yayınevi, İstanbul, 2003.
[11] I. Kaastra, M. Boyd, "Designing a Neural Network for Forecasting Financial and Economic Time Series", *Neurocomputing*, 10(3), 215-236, 1995.
[12] A. Uğur, A. C. Kınacı, "Yapay Zeka Teknikleri ve Yapay Sinir Ağları Kullanılarak Web Sayfalarının Sınıflandırılması", **XI. Türkiye' de İnternet Konferansı (inet-tr'06)**, Ankara, 1-4, 2006.
[13] S. Haykin, **Neural Networks: A Comprehensive Foundation**, Pearson Prentice Hall Inc., New Jersey, 1999.
[14] S. Haykin, **Neural Networks and Learning Machines (Third Edition)**, Pearson Prentice Hall Inc., New Jersey, 2009.
[15] J. A. Anderson, **An Introduction to Neural Networks**, Massachusetts Institute of Technology Press, USA, 1995.
[16] A. Uğur, E. Binici, "Java ile Yapay Zekâ Yazılımları Geliştirme", **II. Ulusal Meslek Yüksekokulları Sempozyumu**, İzmir, 22, 2003.
[17] Ç. Elmas, **Yapay Sinir Ağları**, Seçkin Yayınevi, Ankara, 2003.
[18] A. M. Sarhan, O. I. Al Helalat, "Arabic Character Recognition using Artificial Neural Networks and Statistical Analysis", *World Academy of Science, Engineering and Technology*, 27(1), 32-36, 2007.
[19] O. Ayhan Erdem, E. Uzun, "Yapay Sinir Ağları ile Türkçe Times New Roman, Arial ve El Yazısı Karakterleri Tanıma", *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 20(1), 13-19, 2005.
[20] M. Po-Hsien Wu, **Handwritten Character Recognition**, Thesis for the degree of Bachelor of Engineering(Honors), The University of Queensland, The School of Information Technology and Electrical Engineering, 2003.
[21] H. B. Kılıç, **Yapay Sinir Ağları ile Karakter Algılama**, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 1998.
[22] B. Kır, C. Öz, A. Gülbağ, "Yapay Sinir Ağlarında Negative Correlation Learning Metodu Kullanarak Optik Karakter Tanıma", **Elektrik-Elektronik Bilgisayar Sempozyumu (FEEB 2011)**, Elazığ, 105-109, 2001.