

YSA TABANLI SİSTEMLER İÇİN GÖRSEL BİR ARAYÜZ TASARIMI

***Ramazan BAYINDIR, **Ömer SESVEREN**

*Gazi Üniversitesi, Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümü, Beşevler/Ankara

**Çukurova Elektrik Meslek Lisesi, Kenan Evren Bulvarı, Seyhan/Adana

Geliş Tarihi : 12.11.2007

ÖZET

Teknolojik gelişmelere paralel olarak, klasik kontrol algoritmaları yanında yapay zeka yöntemleri sistemlerin kontrolünde kullanılmaya başlanmıştır. Yapay zeka uygulamalarının artması bu alanda eğitimi zorunlu hale getirmiştir. Bu makalede Yapay Sinir Ağları'nın (YSA) etkin bir şekilde öğrenilmesi, kavranması için hazırlanan bilgisayar tabanlı yapay sinir ağları yazılımı sunulmuştur. Geliştirilen yazılım ile yapay sinir ağının iterasyon sayısı, momentum katsayısı, öğrenme oranı, etkinlik fonksiyonu gibi bileşenleri değiştirilerek, verilere göre yapay sinir ağının eğitimi ve test işlemi yapılabilmektedir. Çalışma sonucunda, yapay sinir ağları yapısının gerçek zamanlı uygulamalara kolaylıkla adapte edilebileceği görsel bir eğitim seti elde edilmiştir.

Anahtar Kelimeler : Yapay sinir ağları, Yapay zeka, Eğitim seti.

DESIGN OF A VISUAL INTERFACE FOR ANN BASED SYSTEMS

ABSTRACT

Artificial intelligence application methods have been used for control of many systems with parallel of technological development besides conventional control techniques. Increasing of artificial intelligence applications have required to education in this area. In this paper, computer based an artificial neural network (ANN) software has been presented to learning and understanding of artificial neural networks. By means of the developed software, the training of the artificial neural network according to the inputs provided and a test action can be performed by changing the components such as iteration number, momentum factor, learning ratio, and efficiency function of the artificial neural networks. As a result of the study a visual education set has been obtained that can easily be adapted to the real time application.

Key Words: Artificial neural network, Artificial intelligence, Educational set.

1. GİRİŞ

Yapay sinir ağları (YSA); deneysel bilgiyi alan, depolayan ve kullanan fiziksel hücreli sistemlerdir (Sağiroğlu, 2003). Yerel bir hafıza yapısı tanımlayan ve birbirleri ile çeşitli şekillerde bağlantılı olan veri işleme elemanlarının dağıtılmış veriyi geliştirebildikleri paralel bir yapıdır (Haykin, 1994).

Klasik yaklaşımlarda; çeşitli sistemler üzerinde çalışırken genellikle matematiksel modeller oluşturulur. Matematiksel modelleri çok zor tanımlanabilen veya tanımlanamayan problemlerin çözümü için YSA kullanılmaktadır. Ayrıca YSA belirsiz, gürültülü ve eksik verili sistemlerin çözümlenmesinde kolaylıkla kullanılabilir. Bu özelliklerinden dolayı YSA günümüzde iş hayatı, finans ve endüstri alanlarında doğrusal olmayan sistemlerin modellenmesinde mevcut yöntemlere

göre daha yoğun bir ilgi görmektedir (Elmas, 2003). Ayrıca, güç sistemlerinin güvenliği, güç sistemleri kararlılığı, reaktif güç kompanzasyonunda ve motor kontrol uygulamalarında da YSA'ların sıkça tercih edildiği rapor edilmektedir (Bayındır, 2007; Çolak, 2003; Sağıroğlu, 2006).

Tasarımı yapılan sistemlerde sistemin davranışını önceden görülmesi ve sistemden beklenen sonuçların oluşup oluşmadığı hakkında bilgi edinilmesi gerçekleştirilecek olan sistemin istenilen çıkış verip vermeyeceğinin bir göstergesidir. Simülatörler, gerçek sistemlerin çalışma şartlarını ve sistem tepkilerini bilgisayar ortamında taklit eden yazılımlardır. Gerçek sistemlerin kurulmasının, çalıştırılmasının zorlukları ve maliyetini ortadan kaldırarak sistemler üzerinde çalışmaya imkan sağlar. YSA'nın bir kontrol yöntemi olarak kullanılmasında kontrolü yapılacak sistemin incelenmesi, giriş verilerinin değiştirilerek YSA denetleyicilerinin üreteceği kontrol sinyallerinin hesaplanması gibi işlemler YSA simülatörü ile yapılabilmektedir. Farklı yapay sinir ağı metotları ile yağış-akış ilişkisinin modellenmesi (Alp, 2004), bulanık mantık denetleyicili bir iklimlendirme sistemi (Elmas, 2000), bulanık mantık denetleyicili fırçasız doğru akım motor hız kontrolü (Elmas, 2000a) gibi farklı uygulama alanları için hazırlanmış simülatörler ile yapay sinir ağları eğitim seti (Güvenç, 2007) gibi eğitim uygulamaları için YSA yazılımları hazırlanmıştır.

Bu çalışmalarda tasarımı düşünülen sistemlerin, prototip olarak fiziksel sistem elemanlarından oluşturulmadan önce bilgisayar ortamında analiz edilmesi, tasarımı yapılan sistemin davranışının önceden görülmesine ve sistemden beklenen sonuçların oluşup oluşmadığı hakkında bilgi edinilmesine yardımcı olmuştur. Ayrıca YSA eğitimi yapılabilen MATLAB, Neural Ware gibi paket programlar mevcuttur. Bu programlarda YSA yapısının nasıl oluşturulduğu, algoritma yapısının nasıl oluşturulduğu, YSA eğitiminin nasıl gerçekleştirildiği gibi YSA'nın öğretilmesinde kullanılacak detaylar verilmemektedir (Güvenç, 2007).

Bu çalışmada, YSA denetleyicinin Artırmalı Back propagation (BP), Artırmalı BP momentum, Grup BP, Grup BP momentum algoritmaları ile eğitilerek daha hızlı öğrenme, daha basit bir yapı elde edebilme ve yüksek başarımlı ve esnek bir eğitim aracı sağlanması amacıyla bilgisayar tabanlı bir YSA yazılımı sunulmuştur. Yazılımda, öğrenme algoritması olarak danışmanlı öğrenme, YSA yapısı olarak ta çok katlı perceptron (MLP) öğrenme algoritması tercih edilmiştir. YSA yazılımının hazırlanmasındaki amaç; YSA'nın yapısının,

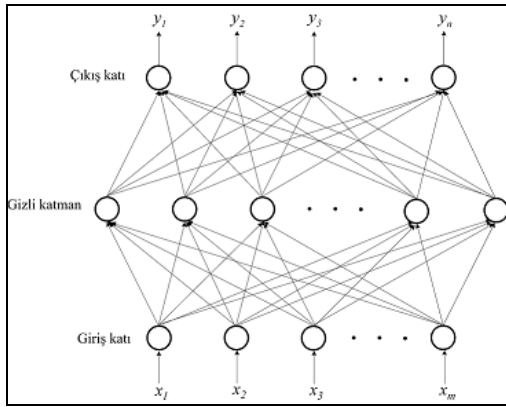
bileşenlerinin, YSA'nın eğitim ve test aşamalarının etkili bir şekilde öğrenilmesini sağlamaktır. Yazılımda YSA yapısı, giriş-çıkış katmanları, ara katman sayısı ve ara katmandaki düğüm sayıları kullanıcı tarafından tanımlanmaktadır. YSA yapısı oluşturulurken YSA şekli çizilerek yapının kolaylıkla anlaşılması sağlanmaktadır. YSA'nın çalışma parametreleri olan öğrenme sayısı, öğrenme oranı ve momentum katsayısı kullanıcı tarafından değiştirilebilmektedir. YSA'nın performansı RMS değerine bakarak değerlendirilebilmektedir. Eğitim, azami öğrenme sayısına (iterasyon sayısı) veya istenen RMS değerine kadar sürdürülmektedir. RMS değerinin grafiği ile eğitim süreci hakkında yorum yapılabilmektedir. YSA yapısının seçiminde kabul edilmiş herhangi bir kural yoktur. Bu nedenle YSA yapısındaki katman sayısı ve katmanlardaki düğüm sayısı (neuron sayısı) deneme-yanılma yoluyla bulunur. YSA'lar hangi yapıda olursa olsun eğitilirken ilk olarak rasgele bir ağırlık değeri ile çalışmaya başlar. Aynı yapıda olmasına rağmen iki ayrı YSA aynı sonuçları üretmeyebilir. Eğitim bittikten sonra YSA yapısına ait ağırlık ve bias değerleri bir dosyaya kaydedilerek test işlemlerinde kullanılmak üzere saklanır. Eğitilen YSA'lar test edilerek sonuçlar kaydedilebilmektedir. YSA'nın test sonuçlarına göre en iyi sonuçları üreten YSA yapısına karar verilebilir. Bu nedenle hazırlanan simülasyonda kullanıcının uygun YSA yapısına karar vermesini kolaylaştırmak için her eğitim ve test işleminin ürettiği çıktılar kaydedilebilmektedir.

2. YAPAY SİNİR AĞLARI

YSA, öğrenme yeteneği, kolayca farklı problemlere uyarlanabilirliği, genelleme yapabilmesi, paralel yapılarından dolayı hızlı çalışabilme yeteneği ve kullanıcının giriş ile çıkış arasındaki ilişkiyi tarif etme mecburiyetinin olmayışı gibi üstünlüklerden dolayı, pek çok uygulamada kullanılmaktadır (Sağıroğlu, 2003; Haykin, 1994). YSA, bir sisteme ilişkin çeşitli parametrelere bağlı olarak tanımlanan girişler ve çıkışlar arasında ilişki kurabilme yeteneğine sahiptir. Bu ilişkinin doğrusal bir formda olması zorunlu değildir. Ayrıca YSA'lar, çıkış değerleri bilinmeyen tanımlanmış sistem girişlerine de uygun çıkışlar üretebilmekte, böylece çok karmaşık problemlere bile iyi çözüm olabilmektedirler (Haykin, 1994).

Literatürde birçok YSA yapısı mevcuttur (Sağıroğlu., 2003; Haykin, 1994). Sunulan çalışmada öğrenme algoritmasının bu ağı eğitimde kullanılabilir olması, oldukça basit, bu sebeple de yaygın olarak kullanılır olması sebebiyle Çok Katlı Perceptron (ÇKP) modeli tercih edilmiştir. Ayrıca,

ÇKP birçok alana uygulanmış olan bir YSA yapısıdır (Haykin, 1994). Şekil 1’de de verildiği gibi bir ÇKP modeli, bir giriş, bir veya daha fazla ara ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Giriş katındaki nöronlar tampon gibi davranırlar ve giriş sinyalini ara kattaki nöronlara dağıtırlar. Ara kattaki her bir nöronun çıkışı, kendine gelen bütün giriş sinyallerini takip eden bağlantı ağırlıkları ile çarpımlarının toplanması ile elde edilir. Elde edilen bu toplam, çıkışın toplam bir fonksiyonu olarak hesaplanabilir.



Şekil 1. Bir ÇKP-YSA modeli.

Buradaki, fonksiyon, basit bir eşik fonksiyonu, bir sigmoid veya hiperbolik tanjant fonksiyonu olabilir. Diğer katlardaki nöronların çıkışları da aynı şekilde hesaplanır. Kullanılan eğitime algoritmasına göre, ağırlık çıkışı ile arzu edilen çıkış arasındaki hata tekrar geriye doğru yayılarak hata minimuma düşüncüye kadar YSA’da ağırlıklar değiştirilir (Sağıröglü, 2003). Bu çalışmada ağırlık çıkışı ile arzu edilen çıkışlar arasındaki hata tüm giriş seti için bulunduktan sonra ağırlıklar değiştirilmektedir.

Yapay sinir ağlarında kullanılan çok sayıda öğrenme algoritması bulunmaktadır. Sunulan çalışmada geri yayılım (back propagation) algoritması uygulanmıştır. Geri yayılım algoritmasının gradyent azaltma algoritması ile uygulanabilecek; Artırmalı (incremental) ve Grup (batch) öğrenme modlarında standart geri yayılım ve momentumlu geri yayılım algoritmaları öğrenmede uygulanabilmektedir. Artırmalı Mod Geri Yayılım (IBP), Artırmalı Mod Momentumlu Geri Yayılım (IBPM), Grup Mod Geri Yayılım (BBP), Grup Mod Momentumlu Geri Yayılım (BBPM) olmak üzere dört öğrenme algoritması ile çalışılabilmektedir. Bu algoritmalarda Grup Mod’unda eğitim verileri tamamen işleme alınırken, Artırmalı Mod da eğitim verileri teker-teker işleme alınmaktadır. Momentum katsayısının eklendiği ve eklenmediği eğitim işlemlerinin seçilebildiği iki algoritma oluşturulmuştur. Geri

yayılım algoritması eğitici öğrenmede kullanılan en genel algoritmadır. Basit olması ve iyi bir öğrenme kapasitesine sahip olması birçok alana uygulanmasını sağlamıştır.

Geril yayılım algoritması, Yayılma (propagate) ve uyum gösterme (adapt) olmak üzere iki aşamada işlemleri gerçekleştiren katmanlar arasında tam bir bağlantının bulunduğu çok katmanlı, ileri beslemeli ve öğreticili olarak eğitilen bir YSA modelidir. Girişlerle çıkışlar arasındaki hata sinyali bulunarak, ağırlıklar bu hata sinyaliyle güncellenmektedir. Hata yani $e_{(k)}$, arzu edilen çıkış (gerçek çıkış - $y_{(k)}$) ile sinir ağının çıkışı ($O_{(k)}$) arasındaki farktır.

$$e_{(k)} = y_{(k)} - O_{(k)} \quad (1)$$

Hesaplanan hata sinyalleri, her çıktı düğümüne karşı gelen ara katmandaki düğümlere aktarılır. Böylece ara katmandaki düğümlerin her biri toplam hatanın sadece hesaplanan bir kısmını içerir. Bu süreç her katmandaki düğümler toplam hatanın belirli bir kısmını içerecek şekilde giriş katmanına kadar tekrarlanır. Elde edilen hata sinyalleri temel alınarak, bağlantı ağırlıkları her düğümde yeniden düzenlenir. Bu düzenleme tüm verilerin kodlanabileceği bir duruma ağırlık yakınsamasını sağlar.

İleri besleme safhasında, giriş katmanındaki nöronlar veri değerlerini doğrudan gizli katmana iletir. Gizli katmandaki her bir nöron, kendi giriş değerlerini ağırlıklandırarak toplam değeri hesap eder ve bunları bir aktivasyon fonksiyonu ile işleyerek bir sonraki katmana veya doğrudan çıkış katmanına iletir. Katmanlar arasındaki ağırlıklar başlangıçta rasgele küçük rakamlardan seçilir.

Çıkış katmanındaki, her bir nöron ağırlıklandırılmış değeri hesaplandıktan sonra, bu değer yine aktivasyon fonksiyonu ile karşılaştırılarak mevcut hata minimize edilmeye çalışılır. Hata değeri belli bir mertebeye ininceye kadar iterasyon işlemine devam edilir ve böylece ağırlık eğitim aşaması tamamlanmış olur. Katmanlar arasındaki bağlantılardaki ağırlık değerleri eğitimi tamamlamış ağırlıklardan test aşamasında kullanılmak üzere saklanır (Hagan, 1996).

Geril yayılım algoritmasının gradyent azaltma algoritması ile çalıştırılmasında eğitim setinin hesaplamaya dahil edilmesinde kullanılan iki öğrenme yöntemi vardır. Bu yöntemler Artırmalı Mod ve Grup Modudur. Artırmalı modda gradyentin hesaplanarak ağırlıkların güncelleştirilmesi işlemi her bir girişin ağırlık uygulanması ile gerçekleştirilir. Böylece, eğitimde gelişmenin veya ilerlemenin olup-olmadığı görülebilir ve istenilen doğruluk için amaç

fonksiyonun minimum değeri hesaplanabilir. Grup modda ise eğitim veri setinin tamamı ağa uygulanır ve elde edilen sonuçlara göre ağırlıkların güncelleştirilmesi yapılır (Hagan, 1996).

Eğitim sürelerini azaltmak için sezgisel yaklaşım yöntemleri kullanılmaktadır. Sezgisel yöntemler hata oranını daha hızlı bir şekilde azaltmak üzere kullanılır. Hatayı düzeltme ve ağırlıkların değiştirilmesinde bir toparlanma sağlaması için toparlanma oranını hızlandırmak için en çok kullanılan yöntemlerden biri momentum katsayısı (γ) kullanılmaktadır (Çetin, 2006). Bu katsayı; ağırlık daha hızlı toparlanmasına yardım eder. Temel olarak daha önceki değişimin bir kısmını işlem gören değişime eklemeye dayanır. Momentum katsayısı ağırlık yerel gradientleri aşmasını sağladığı gibi aynı zamanda hatanın düşmesine de yardımcı olur. Momentum olmaksızın ağırlık yerel minimum değerine takılarak salınımlı olabilir. Momentum katsayısı kullanılmak, öğrenme esnasında ağırlık salınımlarını engeller. Momentum değeri 0 ile 1 arasındadır. Momentum kullanıldığında, ağırlıklara göre eğitim ardışık iki iterasyonda aynı işaretli ise ağırlıklara uygulanacak düzeltme artar, aksi halde azalır. Eğer momentum sıfır ise ağırlık değişimi tamamen gradiente bağlıdır. Momentum değeri bir ise ağırlık değişimi gradiyentten bağımsızdır ve bir önceki ağırlık değerine eşit olur (Sağıroğlu, 2003).

Öğrenme oranı YSA'nın ağırlıklarının değişiminde kullanılan bir sabitedir. Her zaman birden küçük ve pozitif değerler almalıdır. Pozitif değerler hesaplamalarda, hesaplanan değerlerin istenen değerlere daha hızlı yaklaşmasını sağlarken, 1'den küçük değerler sistemin kararlı olmasını sağlar. Öğrenme oranı ağırlık performansı üzerinde önemli bir etkiye sahiptir. Küçük öğrenme oranı değerleri için eğitim işlemi uzun zaman alırken bu değerlerin büyütülmesi ile eğitim işlemi daha kısa zamanda gerçekleşmektedir. Öğrenme oranının artırılması durumunda öğrenme işlemi için gerekli adım sayısında (öğrenme sayısı-iterasyon) azalma meydana gelmektedir. Öğrenme oranının artırılması ağırlık toplam hatası üzerinde bir iyileştirme meydana getirmektedir. Fakat öğrenme oranına çok büyük değerlerin verilmesi iyi bir yakınsama oluşturmamaktadır (Elmas, 2003).

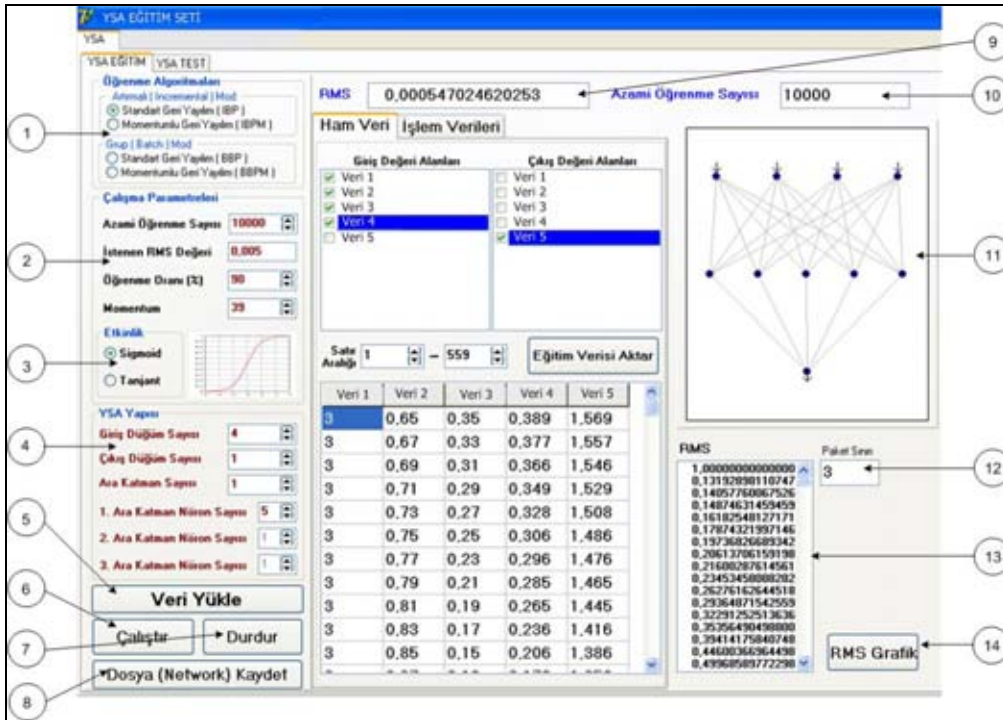
3. GELİŞTİRİLEN YSA YAZILIMININ İNCELENMESİ

YSA; yapay zeka uygulamalarının en çok tercih edilen ve uygulama alanı bulan yöntemlerinden biridir. Bir sistemin çalışmasının ve tepkilerinin

incelenmesi veya kontrolü için YSA tercih edilebilir. Bu çalışmadaki YSA programı; birçok farklı alanda YSA çalışması yapılabilmesi için esnek yapıda ve görsel ara yüze sahip olarak hazırlanmıştır. Kullanıcı YSA hakkında temel bir bilgi seviyesi ve uygun verinin hazırlanması ile kolaylıkla YSA çalışmasını yapabilir. Hazırlanan YSA programında kullanıcıyı yönlendirmek amacı ile işlem sırasına göre butonların pasif-aktif olması sağlanmaktadır. Ayrıca parametrelerin değiştirilmesi ile oluşan farklılıklar kolaylıkla takip edilebilmekte, eğitim sırasında performans kriteri olarak RMS hesaplanmaktadır. RMS değerleri bir grafik ile gösterilerek eğitimin performansının YSA yapısı değiştirildikçe gözlenmesi sağlanmıştır. Şekil 2'de YSA eğitim paneli tam ekran görüntüsü verilmiştir. Şekil 2'deki YSA Eğitim Paneli ekran görüntüsü üzerinde menülerin görevleri aşağıdaki gibi özetlenebilir.

1. YSA Öğretme Algoritması Seçim Paneli: Yazılımda IBP, IBPM, BBP, BBPM olmak üzere 4 öğrenme algoritması seçimi yapılmaktadır.
2. YSA Çalışma Parametrelerinin Ayar Paneli: YSA'nın eğitiminde kullanılan çalışma parametreleri bu panelden ayarlanır. Çalışma parametreleri şunlardır:
 - Azami Öğrenme Sayısı: YSA eğitimi süresince öğrenme sayısının ulaşacağı en yüksek değerdir. İterasyon sayısı da denir. Bu değere ulaşıncaya kadar eğitim durdurulur.
 - İstenen RMS Değeri: YSA eğitilirken performans kriteri olarak RMS değeri hesaplanmaktadır. RMS değerinin istenen bir değere ulaşılması halinde eğitimin başarılı olduğu kabul edilerek eğitim durdurulur.
 - Öğrenme Oranı: YSA'nın ağırlık değerlerini değiştirirken kullanılan öğrenme sabitidir. Her zaman pozitif ve birden büyük değerler alınmalıdır. Yazılımda öğrenme oranı yüzdelik bir ifade ile tanımlanarak hesaplamalarda kullanılmaktadır.
 - Momentum : Hatayı azaltmak ve ağırlıkların değiştirilmesinde bir toparlanma sağlamak için 0 ile 1 arasında bir sayıdır. Yazılımda bu değer 0 ile 100 arasında atandıktan sonra yüzdesi alınarak hesaplamalarda kullanılmaktadır.
3. Etkinlik Fonksiyonunun Seçim Paneli: Etkinlik; hesaplamalarda kullanılan aktivasyon fonksiyonlarıdır. Sigmoid ve tanjant fonksiyonlarından biri seçilerek kullanılmaktadır.
4. YSA Yapısının Belirlendiği Panel: YSA'nın giriş, çıkış ve ara katman sayısı ile ara katmandaki nöron sayısı atanmaktadır.
5. Eğitimde Kullanılacak Veri Yükleme Butonu: Eğitim veri setinin bulunduğu dosyanın yüklenmesi için kullanılır.
6. Eğitim İşlemine Başlatma Butonu: Tüm atamalar yapıldıktan sonra aktif olan ve eğitimi başlatan butondur.

7. Eğitimi Durdurma Butonu: Eğitimin herhangi bir anında çalışmayı durdurur.
8. Eğitim Sonrasında YSA'ya Ait Bilgileri Dosyaya Kaydetme Butonu: Eğitim tamamlandıktan sonra YSA yapısının, azami öğrenme sayısının, hesaplanan ağırlık değerlerinin, momentum değerinin, öğrenme oranı değerinin .NET uzantılı bir dosyaya kaydedilmesinin sağlar.
9. Eğitim Sırasında Hesaplanan RMS Değeri: Eğitim performansının görülmesinin sağlayan RMS değerinin anlık değeridir.
10. Azami Öğrenme Sayısı : Eğitim sırasında işlem yapılan öğrenme sayısı.
11. YSA Yapısının Çizimi : YSA yapısı belirlendikçe çizilen YSA'nın grafiksel gösterimidir.
12. Hesaplanan RMS Değerinin RMS Alanına Yazılma Sıklığı: Anlık RMS değerinin istenen bir sıra ile kaydedilmesinin sağlayan değerdir. 5 olarak atanırsa 1-6-11. RMS değerleri kaydedilir.
13. RMS: Hesaplanan RMS değerleri.
14. Hesaplanan RMS Değerlerine Ait Grafik Çizdirme Butonu: Hesaplanan RMS değerleri alanındaki değerlere göre öğrenme sayısına bağlı olarak RMS grafiği çizdirir.



Şekil 2. YSA eğitim paneli tam ekran görüntüsü.

3. 1. Yazılıma Veri Yükleme

YSA bir sisteme ait giriş ve çıkış verileri arasında bir bağıntı belirleyerek farklı giriş verilerinden yeni çıkış verisi üretir. Bu giriş ile çıkış arasındaki bağıntının üretilmesi sürecine YSA'nın Eğitimi denir. Bu nedenle eğitim için kullanılacak verinin sistemin çalışmasını örnekleyecek şekilde düzenlenmesi gerekir. Bu çalışmadaki YSA yazılımında veriler eğitim ve test işlemlerinde aynı şekilde çağrılıp işlenmektedir. Kullanılacak veriler MS Excel dosyası (*.xls), sekmeli metin dosyası (*.txt) ve virgüle ayrılmış metin dosyası (*.csv) formatında hazırlanmalıdır. Yazılımda "VERİ YÜKLE" butonu kullanılarak öğrenim yazılıma yüklenmesi sağlanır. Veri yükle butonuna basılınca bir "YSA Veri Dosyası Aç" diyalog penceresi açılır ve dosya türü filtreli olarak veri dosyasına ait dosya

yolu tanımlanır. Veri yükleme tamamlandıktan sonra yazılımın ana ekranında "Ham Veri" sekmesinde veriler alan adları ile görüntülenir. Alan Adı veri dosyasının birinci satırıdır. Ham veri alanındaki verilerden tamamını veya istenen miktarını eğitim ve test işleminde kullanmak mümkündür. Ham verilerden işlenecek seçilen veriler yazılımda "İşlem Verileri" alanında görüntülenmektedir. Ham verilerden İşlem verilerinin atanmasında "Giriş Değeri Alanları" ve "Çıkış Değeri Alanları" onay listesi kullanılır. Onay listesindeki alanlar ham verilerin alan adlarıdır. Giriş değeri alanı ile çıkış değeri alanı aynı alan adlarını listeler. Onay alanındaki bu esneklik aynı verilerin giriş ve çıkış değerleri olarak atanabilmesini sağlar. Ham verilerin satır sayısı da "Satır Sayısı" aralığından belirlenir. İşlem verilerine ait aralık tanımlandıkça eğitim işlemi yapılırken "Eğitim Verisi Aktar", test işlemi

yapılırken “Test Verisi Aktar” butonuna basılır. İşlem verileri sekmesi aktif olur ve seçilen veriler görüntülenir. YSA eğitilirken kullanılan her alanın minimum ve maksimum değerleri bulunarak işlem verilerinin ilgili sütunu altında listelenir. YSA yapısına karar verilirken YSA Yapısı menüsündeki ‘Giriş Düğüm Sayısı’ ve ‘Çıkış Düğüm Sayısı’ değerleri Giriş ve Çıkış Değerleri Alanı’nda yapılan işaretlemelerin sayısından farklı olamaz. Farklı olması durumunda ekrana uyarı mesajı gelir. Örneğin giriş veri alanında giriş için 4 alan işaretlenmişse YSA yapısında da Giriş düğüm Sayısı 4 olarak belirlenmelidir.

4. YSA’NIN EĞİTİMİ

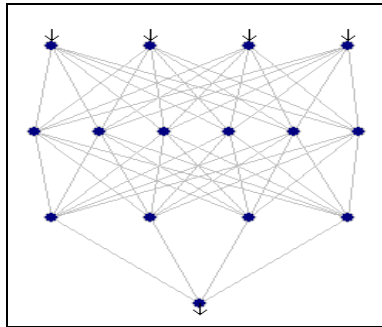
YSA’nın giriş değerlerine göre çıkış değerleri arasındaki bağıntının bulunması sürecine YSA’nın

eğitimi denir. Eğitim yapılırken sırası ile YSA’nın yapısı belirlenmeli, çalışma parametreleri atanmalı ve eğitim başlatıldıktan sonra elde edilen YSA kaydedilmelidir.

4. 1. YSA Yapısını Belirleme

Yazılımın Şekil 3a’da verilen YSA yapısı menüsünden YSA’nın giriş düğüm sayısı ve çıkış düğüm sayısı ile ara katman sayısı belirlenmektedir. YSA’nın giriş düğüm sayısı 10, çıkış düğüm sayısı 7 ve her ara katman 20 nörona kadar artırılabilir. Her ara katman için düğüm sayısının belirleneceği alanlar seçim yapıldıkça aktif hale gelir. Giriş düğüm sayısı ile çıkış düğüm sayısı, veri yükleme alanında belirlenen giriş ve çıkış alan sayısı ile aynı olmalıdır. YSA yapısı değiştirildikçe Şekil 3b’de verilen YSA yapısının grafiği de değişir. Kullanıcı yaptığı değişimi grafiksel olarak da görebilmektedir.

(a) YSA yapısı alanı.



(b) YSA grafiği.

(c) Çalışma parametreleri alanı.

Şekil 3. YSA eğitim işlemi ekran görüntüleri.

4. 2. YSA Eğitim

YSA yapısı, çalışma parametreleri belirlendikten ve veriler yüklenip işlem verileri atandıktan sonra YSA'nın eğitimine başlanabilir. Bu işlemler tamamlandıktan sonra eğitim işlemi başlatacak ÇALIŞTIR butonu aktif hale gelir. Bu yazılımda eğitimin performansı RMS ile belirlenir. YSA eğitimine başlandığında her bir öğrenme sayısı (iterasyon sayısı) için RMS değeri hesaplanır. RMS değerleri anlık olarak ekranda görüntülediği gibi tüm eğitim boyunca hesaplanan RMS değerleri de RMS alanında listelenir. Eğitim çalışma parametrelerinde belirlenen öğrenme sayısına kadar veya istenen RMS değerine kadar sürdürülür. RMS

alanında listelenen RMS değerlerinin grafiği RMS Grafik butonu ile Şekil.4'te verildiği gibi çizdirilerek eğitimin başarısı hakkında yorum yapılabilir.

4. 3. Eğitilen YSA'nın Kayıt İşlemi

YSA eğitimi tamamlandıktan sonra YSA'nın test işlemlerinde, simülasyonlarda veya daha başka uygulamalarda kullanılması için YSA yapısı, çalışma parametreleri ile hesaplanan ağırlık değerleri bir dosyaya kayıt yapılır. Dosya kaydı için "DOSYA(NETWORK) KAYDET" butonu kullanılır. Dosyalar ".NET" uzantısı ile kayıt yapılmaktadır.

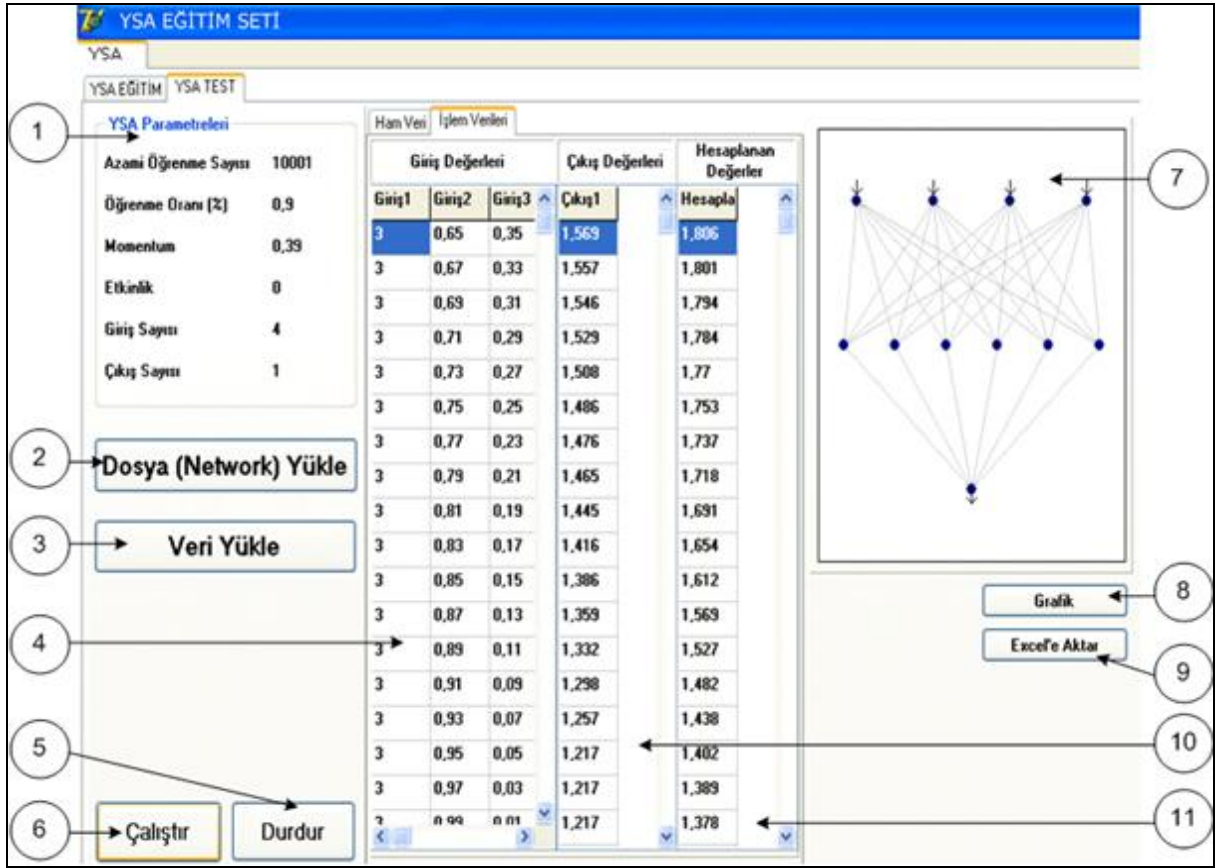


Şekil 4. RMS grafiği.

4. 4. YSA'nın Test İşlemi

YSA eğitilip hesaplanan değerler kaydedildikten sonra YSA'nın testi yapılmalıdır. YSA'nın test işlemi Şekil.5'te verilen ekran üzerinden gerçekleştirilir. Test işlemi eğitimde kullanılan değerlerden farklı olarak uygulanan giriş değerlerine bağlı olarak çıkış değerleri üretilmesidir. Uygulanan sistem için kullanılacak YSA'nın uygun olması hesaplanan çıkış değerleri ile istenen çıkış değerleri arasındaki farkın en az değerde olmasıdır. Bu hazırlanan yazılımda YSA test işleminde veri yüklendikten sonra .NET uzantılı kaydedilmiş YSA dosyası yüklenir. Bu YSA dosyasında bulunan YSA yapısı ve çalışma parametreleri ekranda görüntülenir. Okunan dosyadaki YSA yapısına uygun olarak YSA'nın grafiği de çizilir. Veri yükleme işlemi yapıldıktan sonra ham verilerden

işlem yapılacak veriler, giriş-çıkış değerleri alanları seçilerek ve satır sayısı belirlenerek atanır. Bu veriler işlem verileri alanında tablolar halinde listelenir. Giriş ve Çıkış sayısı yüklenen YSA'nın giriş-çıkış sayısı ile aynı sayıda olmalıdır. giriş-çıkış sayıları aynı olmazsa ekrana uyarı mesajı gelir. Veri yüklemesi ve YSA dosyası yükleme işlemi yapıldıktan sonra test işlemi "ÇALIŞTIR" butonu ile başlatılır. Ekranda çıkış değerleri ile hesaplanan değerler liste halinde görüntülenir. Bu değerlerin grafiği de "GRAFİK" butonu ile çizdirilir. Çıkış değerleri ve hesaplanan değerler MS Excel de kullanılmak üzere "EXCELE AKTAR" butonu ile yeni bir Ms Excel dosyasına aktarılabilir. Şekil 5'deki YSA test ekran görüntüsü üzerindeki menülerin görevleri aşağıdaki gibi özetlenebilir.



Şekil 5. YSA test ekranı.

1. YSA Parametreleri: Testi yapılacak YSA yüklenince eğitimde kullanılan parametreler görüntülenir. YSA yapısına ait giriş-çıkış sayısı, momentum katsayısı, öğrenme oranı ve öğrenme sayısı dosyadan okunur.
2. Dosya (Network) Yükle Butonu: Kaydedilen YSA dosyası yükleme butonudur. Dosya Aç penceresiyle .NET uzantılı YSA dosyaları seçilip yüklenir.
3. Veri Yükle Butonu: Test işleminde kullanılacak veri seti yükleme butonudur.
4. Test Verisi Giriş Değerleri Tablosu: Veri setinden işlenecek verilerin görüntülediği tablodur.
5. Test İşlemini Durdurma Butonu: Test işlemini herhangi bir anda durduran butondur.
6. Test İşlemi Başlatma Butonu: YSA ve veri seti yüklendikten sonra test işlemini başlatan butondur.
7. Test Edilecek YSA'nın Grafiği: YSA parametrelerine göre çizilen YSA grafiğidir.
8. Grafik Butonu: Çıkış değerleri ile Hesaplanan Değerlerin grafiğini çizdirme butonudur.
9. Excele Aktar Butonu: Çıkış Değerleri ile Hesaplanan Değerler Tablolarındaki değerleri MS Excel e aktarma butonudur.
10. Çıkış Değerleri Tablosu : Veri setindeki çıkış değerlerinin görüntülediği tablodur.
11. Hesaplanan Değerler Tablosu: Test işleminde giriş değerlerine göre YSA'nın hesapladığı değerlerin görüntülediği tablodur.

5. SONUÇ VE DEĞERLENDİRME

Bu çalışmada bir YSA yazılımı hazırlanmıştır. Bu yazılım ile yapay sinir ağları ile yapılabilecek her türlü analiz, kontrol çalışması için eğitim ve test aşamaları adım adım izlenebilmektedir. YSA yapısına karar verilirken yapıyı tanımlayacak grafiklerin çizilmesi, eğitim sırasında hesaplanan RMS değerlerinin grafiğinin çizilmesi, test sırasında çıkış değerleri ile hesaplanan değerlerin grafiklerinin çizilmesi YSA'nın anlaşılmasını ve YSA'nın bileşenlerinin değiştirilmesi ile oluşan etkilerin kavranmasını kolaylaştırmıştır. Bu yazılım ile YSA'nın çok farklı alanlara uygulanmasına ait bir çok çalışma ve simülasyon hazırlamak mümkündür. Öğrencinin hazırlanan bu eğitim setini kullanarak YSA denetleyici ile yapılacak bir kontrol için gerekli olan eğitim şartları deneyerek, en uygun YSA

yapısının tespit edilmesi ve bu YSA yapısının simülátördeki YSA denetleyiciye yüklenerek çalışmanın izlenebilmesi, YSA ile yapılacak kontrol çalışmalarının anlaşılmasını kolaylaştıran esnek ve işlevsel bir yapıya sahip olduğu görülmüştür.

Bu çalışmanın diğér bir özgülüğü, öğrencilerin eğitimini güncel teknolojiler kullanılarak desteklemenin yanında, birçok alanda kullanılan yapay zekanın farklı alanlarda uygulanabilirliğine bir örnek olması açısından teknik eğitim ve mühendislik eğitiminde yeni bir yaklaşım hayata geçirilmesi amaçlanmıştır. Bu amaçla yapay sinir ağıları konusunda yeterliliğe sahip 10 Yüksek Lisans öğrencisi üzerinde yapılan anketteki sonuçlar değerlendirildiğinde, öğrencilerin yeni bir yaklaşım sunan çalışmaya ılımlı baktıkları, simülátörün öğrenmeyi kolaylaştırdığını ve yapay zeka öğretiminde bu çalışmada olduğu gibi bu tür simülátörlerin kullanılmasının faydalı olacağını belirtmişlerdir. Geliştirilen simülátörde bir kontrol için gerekli olan eğitim şartları denenerek en uygun YSA yapısının tespit edilmesi, bu yapının veya elde edilen herhangi bir yapının simülátöre yüklenerek çalışmanın izlenebilmesi kontrol çalışmasının anlaşılmasını kolaylaştıran önemli bir etkidir. Değerlendirmeye katılan öğrenciler çalışmanın izlenebilmesi konunun kavranmasında son derece önemli olduğu görüşündedir.

Burada karşılaşılan güçlükler ise, uygun bir YSA yapısının, fonksiyon tipinin ve öğrenme algoritmasının belirlenmesi, simülátörün birbiri ardınca eğitime tabi tutulmasında eğitim performansının düşmesi, aynı YSA yapısının eğitime tabi tutulmasında bile başlangıç ağırlık değerlerinin rasgele atanmasından kaynaklanan benzer sonuçların üretilememesi olarak sıralanabilir.

6. KAYNAKLAR

Alp, M. ve Cıgızođlu, H. K. 2004. Farklı Yapay Sinir Ağı Metotları ile Yađış-Akış İlişkinin Modellenmesi, İTÜ Dergisi 3, 80-88.

Bayındır, R., Sađırođlu, Ş. ve Çolak, İ. 2007. Yapay Sinir Ağıları Tabanlı Reaktif Güç Kompanzasyonu,

Gazi Üniversitesi Teknik Eğitim Fakültesi, Politeknik Dergisi 10, 129-135.

Çetin, M., Uğur, A. ve Bayzan, Ş. 2006. "İleri Beslemeli Yapay Sinir Ağlarında Backpropagation (Geriye Yayılım) Algoritmasının Sezgisel Yaklaşımı", Akademik Bilişim Kongresi, Pamukkale Üniversitesi, Denizli, Şubat 2006.

Çolak, İ., Bayındır, R. ve Bay, Ö. F. 2003. Reactive Power Compensation Using A Fuzzy Logic Controlled Synchronous Motor, Energy Conversion and Management 44, 2059-2215.

Elmas, Ç. 2003. Yapay Sinir Ağları, Seçkin Yayınevi, Ankara.

Elmas, Ç. 2000. Bulanık Mantık Denetleyicili Bir İklimlendirme Sistemi, Gazi Üniv., Politeknik Dergisi 3, 15-22.

Elmas, Ç. 2000a. Bulanık Mantık Denetleyicili Fırçasız Doğru Akım Motor Hız Kontrolü, Gazi Üniv. Politeknik Dergisi 3, 7-14.

Güvenç, U., Birođul, S. ve Sönmez, Y. 2007. Yapay Sinir Ağları Eğitim Seti, The Proceedings of 7th International Educational Technology Conference North Cyprus, May 2007, 709-713.

Haykin, S. 1994. Neural Networks: A Comprehensive Foundation, New York, Macmillan College Publishing Company, 1994, ISBN 0-02-352761-7.

Hagan, M.T., Demuth, H.B. and Beale, M. 1996. Neural Network Design, Boston: PWS Publishing co.

Sađırođlu, Ş., Beşdok, E. ve Erler, M. 2003. Mühendislikte Yapay Zeka Uygulamaları-I Yapay Sinir Ağları, Ufuk Yayınevi, Kayseri.

Sađırođlu, Ş., Çolak, İ. ve Bayındır, R. 2006. Power Factor Correction Technique Based on Artificial Neural Networks, Energy Conversion and Management, Energy Conversion and Management 47/(18-19), 3204-3215.