

Genetik Algoritma Kullanılarak İleri Beslemeli Bir Sinir Ağında Etkinlik Fonksiyonlarının Belirlenmesi

Determination of Activation Functions in A Feedforward Neural Network by using Genetic Algorithm

Oğuz ÜSTÜN

Abant İzzet Baysal Üniversitesi, Müh.-Mim. Fakültesi, Elk-Elk. Mühendisliği Bölümü, 14280, Bolu

Geliş Tarihi/Received : 16.07.2009, Kabul Tarihi/Accepted : 02.09.2009

ÖZET

Bu çalışmada çok katmanlı ileri besleneli bir sinir ağının tüm katmanlarındaki üyelik fonksiyonları genetik algoritma kullanarak belirlenmiştir. Bir sinir ağının etkinliğini gösteren temel ölçüt aynı sayıda düğüm ve bağlantı ağırlığı ile istenen sonuca daha iyi yaklaşabilmektir. Bu performansı belirleyen en önemli parametrelerden birisi uygun etkinlik fonksiyonlarının seçilmesidir. Klasik sinir ağı tasarımında genellikle bilinen etkinlik fonksiyonlarından birisi seçilerek ağı tasarımı gerçekleştirilmektedir. Bu çalışmada etkinlik fonksiyonları için bir tablo oluşturulmuş ve her bir düğüm için en uygun etkinlik fonksiyonu genetik algoritma ile bu tablodan seçilmiştir. Klasik sabit yapılı bir sinir ağı ile önerilen genetik tabanlı sinir ağının performansının karşılaştırılması için 2 boyutlu regresyon problem kümesi kullanılmıştır. Test sonuçları ortaya konulan yöntemin oldukça yüksek bir yaklaşım kapasitesine sahip olduğunu göstermiştir.

Anahtar Kelimeler : İleribeslemeli sinir ağları, Genetik algoritmalar, Etkinlik fonksiyonları.

ABSTRACT

In this study, activation functions of all layers of the multilayered feedforward neural network have been determined by using genetic algorithm. The main criteria that show the efficiency of the neural network is to approximate to the desired output with the same number nodes and connection weights. One of the important parameter to determine this performance is to choose a proper activation function. In the classical neural network designing, a network is designed by choosing one of the generally known activation function. In the presented study, a table has been generated for the activation functions. The ideal activation function for each node has been chosen from this table by using the genetic algorithm. Two dimensional regression problem clusters has been used to compare the performance of the classical static neural network and the genetic algorithm based neural network. Test results reveal that the proposed method has a high level approximation capacity.

Keywords : Feedforward neural networks, Genetic algorithms, Activation functions

1. GİRİŞ

Son yıllarda yapay sinir ağları örnek tanıma, fonksiyon yaklaşımı, görüntü işleme, regresyon problemleri ve daha önemlisi gerçek zamanlı endüstriyel uygulamalarında büyük bir ilgi toplamaktadır. Bunların arasında ileri beslemeli sinir ağları, yapılarının esnek olması, iyi bir gösterim kapasitesi ve çok sayıda öğrenme

algoritmasına sahip olması nedeniyle en popüler mimari haline gelmiştir (Ma ve Khorasani, 2005). İleri beslemeli bir sinir ağında temel değişkenler bağlantı ağırlıkları, etkinlik fonksiyonları, katman sayıları, düğüm sayıları, öğrenme ve momentum katsayılarıdır. Dolayısıyla etkin bir ağı tasarlanması için bu parametrelerin çok iyi düzenlenmesi gerekir. Özellikle gerçek zamanlı sinir ağı uygulamalarında (Ustun, 2009a;b) sinir

* Yazışılan yazar/Corresponding author. E-posta adresi/E-mail address : oguzustun@ibu.edu.tr (O. Üstün)

ağındaki bağlantı sayılarının ve çarpımların artmasından dolayı yazılım ve donanım işlemi güçleşir.

Genel olarak ağ yapılarının oluşturulmasındaki yöntemler Kwok ve Yeung tarafından ayrıntılı olarak verilmiştir (Kwok and Yeung, 1997). Buna göre ağların oluşturulmasında budama (pruning), inşa (constructive) ve düzenleme (regularisation) ana üç yaklaşım vardır. Bu genel yapının dışında ağdaki hesaplama yükünü ve ağın performansını belirleyen en önemli faktörlerden biriside etkinlik fonksiyonlarıdır. Etkinlik fonksiyonları ne kadar iyi seçilirse ağın yaklaşım hataları da o kadar az olacağı bilinmektedir. Etkinlik fonksiyonlarının belirlenmesi içinde dikkat çekici çalışmalar gerçekleştirilmiştir. Ma ve Khorasani tek gizli katmanlı yapısal bir ağın her bir düğümünde birbirinden farklı polinom fonksiyonlarının nasıl kullanılabileceğini göstermişlerdir (Ma ve Khorasani, 2005). Etkinlik fonksiyonu olarak Hermite fonksiyonlar kullanılmıştır. Wang ve arkadaşları çok katmanlı bir ağın eğitimi esnasında oluşan hata değerine bağlı olarak belirledikleri kurallara göre ağ katmanları için en uygun etkinlik fonksiyonlarını belirlemişlerdir (Wang v.d., 1991). Guarnieri ve arkadaşları çok katmanlı sinir ağlarının optimizasyonu için etkinlik fonksiyonu olarak Catmull-Rom cubic spline kullanmışlardır (Guarnieri v.d., 1999). Mayer ve Schwaiger çok katmanlı sinir ağlarının tasarlanmasında netGEN olarak adlandırdıkları bir simülatör kullanarak çeşitli tipteki cubic- spline etkinlik fonksiyonlarını test etmişlerdir (Mayer ve Schwaiger, 2002). Xu ve Zhang ileri beslemeli sinir ağları için deneysel olarak belirledikleri 6 serbest parametrelili uyarlanabilir bir etkinlik fonksiyonu kullanmışlardır (Xu ve Zhang, 2001). Daqi ve Genxing yapmış oldukları çalışmalarda etkinlik fonksiyonu olarak sigmoid ve gaussian fonksiyonlarının parametrelerini değiştirerek ileri beslemeli ağlardaki öğrenme etkisini incelemişlerdir (Daqi ve Genxing, 2003). Wong ve arkadaşları bu çalışmada çeşitli problemlerin çözümü için farklı etkinlik fonksiyonlarına sahip çok katmanlı ileri beslemeli sinir ağlarda geleneksel sigmoid fonksiyon, sinüzoidal fonksiyon ve bu iki fonksiyonun birleşimi olan periyodik fonksiyonu olmak üzere üç farklı etkinlik fonksiyonunu ağa uyarlamışlardır (Wong v.d., 2002).

Genetik algoritmalar, doğal evrim mantık ve

prensiplerine göre geliştirilen popülasyon tabanlı stokastik araştırma algoritmalarıdır. Genetik algoritmalar özellikle birçok lokal optimumun üretildiği büyük kompleks problemlerin çözümünde faydalıdır. Klasik eğim (gradient) tabanlı araştırma algoritmalarına göre lokal minimuma takılma olasılıkları daha azdır (Yao, 1999). Bu özelliklerinden dolayı birçok sinir ağı tasarımında genetik algoritmalar tercih edilmiştir. Blanco ve arkadaşları recurrent sinir ağlarının eğitilmesinde genetik programlama kullanarak ağ eğitimi gerçekleştirmişlerdir (Blanco, 2001). Marwala, Markov zinciri tabanlı Monte Carlo tekniğini genetik programlamada kullanarak bayesian sinir ağının eğitilmesini gerçekleştirmiştir (Marwala, 2007). Sexton ve Gupta sinir ağlarının eğitilmesinde genetik algoritmaları kullanılmışlardır (Sexton ve Gupta, 2000). Leung ve arkadaşları ileri beslemeli sinir ağlarındaki bağlantı sayılarının optimize edilmesinde (Leung v.d., 2003) genetik algoritmaları kullanılmışlardır. Angeline ve arkadaşları recurrent sinir ağının yapısal düzenlemesi ve bağlantı ağırlıklarının hesaplanmasında genetik algoritmaları kullanmışlardır (Angeline v.d., 1994). Pedrajas, ve arkadaşları recurrent sinir ağları için benzer bir çalışma gerçekleştirmişlerdir (Pedrajas v.d., 2006). Arifovica ve Gencay ileri beslemeli ağların düğüm sayısı katman sayısı gibi ağ yapısına yönelik parametrelerin belirlenmesinde genetik algoritmaları kullanmışlardır (Arifovica ve Gencay, 2001). Oha ve Pedrycz polinomial sinir ağlarında bulunan polinom parametrelerinin genetik algoritmalarla belirlendiği bir çalışma ortaya koymuşlardır (Oha ve Pedrycz, 2006). Ferentinos bağlantı ağırlıkları ve lojistik ve hiperbolik tanjant fonksiyonların giriş ve gizli katmanlardaki düğümlerde belirlenmesinde genetik algoritmalarla yararlanmışlardır (Ferentinos, 2005).

Bu makale, genetik algoritmalar kullanılarak ileri beslemeli çok katmanlı yapay sinir ağlarındaki maksimum düğüm sayısına göre etkinlik fonksiyonlarının belirlenmesinde yeni bir yöntem ortaya koyulmuştur. Çalışmanın amacı aynı örnek sayısında ağın en kısa zamanda ve en iyi doğrulukta sonuç vermesini sağlamaktır. Özellikle gerçek zamanlı uygulamalarda ağın hesaplama yükünü belirleyen en önemli etken düğüm sayılarıdır. Bu uygulamalarda çalışma

zamanı çok sınırlı olduğu için ağlarda kullanılan düğüm sayılarının sınırlı olması gerekmektedir. Dolayısıyla aynı düğüm sayısında en uygun ağın tasarlanması oldukça önemlidir. Bu çalışmada çok katmanlı ileri beslemeli bir sinir ağının tüm katmanlarındaki üyelik fonksiyonları genetik algoritma kullanarak belirlenmiştir. Her bir düğüm fonksiyonu genetik algoritmadaki bir kromozom içerisinde bir gen olarak ifade edilmiştir. Bir 2 boyutlu regresyon problem kümesi içerisinde ortaya konulan yöntemin etkinliği ve kullanılabilirliği test edilmiştir. Elde edilen sonuçlar ortaya koyulan genetik tabanlı sinir ağının performansının klasik yapıdaki sinir ağına göre oldukça yüksek olduğunu göstermiştir.

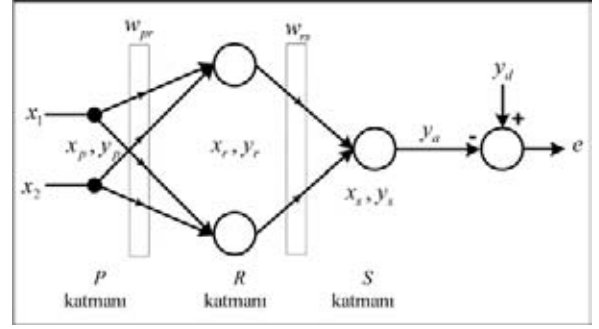
Çalışmanın bundan sonraki bölümü şu şekilde düzenlenmiştir. İkinci bölümde genetik tabanlı sinir ağ modeli ve algoritması anlatılmış, üçüncü bölümde test çalışması gerçekleştirilmiş ve son bölümde genel bir değerlendirme yapılmıştır.

2. GENETİK SİNİR AĞININ TASARIMI

Genetik algoritmaların araştırmadaki başarısı kullanılarak yapay sinir ağlarının optimizasyonu gerçekleştirilmiştir. Bu bölümde ilk olarak optimizasyonu gerçekleştirecek olan sinir ağı tanıtılacak, ikinci olarak genetik algoritmadaki yapılar ve özellikleri ortaya koyulacak ve son olarak bu iki yapının birleştirilmesinde kullanılan algoritma ayrıntılı olarak anlatılacaktır.

2. 1. Sinir Ağının Tanımlanması

Çok katmanlı ileri beslemeli sinir ağlarında her bir düğüm için en uygun etkinlik fonksiyonunun ve öğrenme katsayılarının belirlenmesinde genetik algoritma seçim performansının gösterilmesi için üç katmanlı klasik bir ileri beslemeli ağ yapısı tasarlanmıştır. Benzer ağ modelleri ayrıntılı olarak kaynak (Ustun, 2009a;b)'de görülebilir. Tasarlanan ağ yapısı Şekil 1'de verilmiştir. Ağ iki girişli ve tek çıkışlıdır. P toplam giriş sayısı, R gizli katmandaki toplam düğüm sayısı, S çıkış sayısını göstermektedir. Burada ağ girişleri x_1, x_2 ve ağ çıkışı y_a 'dır. w katmanlar arasındaki bağlantı ağırlıklarını, x,y çiftleri sırasıyla buldukları katmandaki düğümlerin ağırlıklandırılmış giriş toplamını ve çıkışını temsil etmektedir. y_d istenen çıkış değerini ve e istenen ile gerçek çıkış arasındaki hatayı göstermektedir.



Şekil 1. Tasarlanan sinir ağının mimarisi.

2. 1. 1. İleri Besleme Algoritması

İleri besleme algoritmasında girişten çıkışa doğru bir akış vardır. Eşitlik 1'de verilen her bir giriş çifti için ağırlık değerleri ve etkinlik fonksiyonlarına bağlı olarak aşağıda verilen akış kullanılarak gerçek çıkış değeri y_a hesaplanır.

$$x_p = \{x_1, x_2\}, y_p = x_p \quad (1)$$

$$x_r = \sum_{p=0}^1 y_p \cdot w_{pr} \text{ ve } y_r = y_n(x_r) \quad (2)$$

Burada $r=0 \dots 9$ ve $n=r$.

$$x_s = \sum_{r=0}^9 y_r \cdot w_{rs} \text{ ve } y_s = x_s \quad (3)$$

Burada; ağın gerçek çıkış değeri $y_a = y_s$ Eşitlik 3 ile bulunur.

2. 1. 2. Geri Yayılım Algoritması

Danışmanlı eğitim tabanlı geri yayımlı öğrenme algoritması için Eşitlik 4'de verilen enerji fonksiyonuna bağlı olarak, hata değerleri geriye yayılır ve w ağırlık değişimleri aşağıda olduğu gibi bulunur. Burada öğrenme katsayısıdır.

$$E(k) = \frac{1}{2} e^2(k) \text{ ve } E_{ort} = \frac{1}{K} \sum_{k=1}^K E(k) \quad (4)$$

$$e(k) = y_d(k) - y_a(k) \quad (5)$$

$$\Delta w_{rs} = \eta_{rs} \left(-\frac{\partial E}{\partial w_{rs}} \right) \quad (6)$$

$$\Delta w_{pr} = \eta_{pr} \left(-\frac{\partial E}{\partial w_{pr}} \right) \quad (7)$$

$$w_{rs}(k+1) = w_{rs}(k) + \Delta w_{rs} \quad (8)$$

$$w_{pr}(k+1) = w_{pr}(k) + \Delta w_{pr} \quad (9)$$

2. 2. Etkinlik Fonksiyon Tablosunun Belirlenmesi

Bu çalışmada literatürde sıkça kullanılan klasik etkinlik fonksiyonlarının hemen tamamı kullanılmaya çalışılmıştır. Maalesef fonksiyonların sınıflandırılmalarında tam bir birlik söz konusu değildir. Etkinlik fonksiyonları yapısal olarak doğrusal, non-lineer, doyumlu, doyumsuz, monotonik, non-monotonik, sürekli süreksiz, türevlenebilen, türevlenemeyen olmak üzere birçok gruba ayrılmaktadır. Bu kadar farklı sayıda isimlendirmenin sebebi yapay sinir ağlarının örnek tanıma, görüntü işleme, regresyon gibi çok farklı problemlerin çözümlerinde kullanılmaları ve de çok farklı sayıda sinir ağı tipinin olmasından kaynaklanmaktadır. Tablo 1'de verilen etkinlik fonksiyon şablonunda pozitif doğrusal fonksiyon, doğrusal, doyumlu doğrusal, işaret, kare, kübik, lojistik sigmoid, hiperbolik tanjant, radial tabanlı ve sinüs fonksiyon olmak üzere toplam 10 fonksiyon seçilmiştir. Tabloda görülen numaralandırma genetik algoritmadaki gerçek kodlama da aynı biçimiyle 0-9 aralığında kullanılmıştır. Burada her bir etkinlik fonksiyonu genetik algoritmadaki her bir kromozomda bir gen olarak temsil edilecektir. Kullanılan etkinlik fonksiyonlarına ait matematiksel tanımlamalar Tablo 1'de verilmiştir.

Tablo 1. Etkinlik fonksiyon tablosu.

No	İsim	Tanımlama
0	Pozitif doğrusal fonksiyon	$f(x) = \begin{cases} 0 & \text{eğer } x \leq 0 \\ x & \text{eğer } x > 0 \end{cases}$
1	Doğrusal fonksiyon	$f(x) = x$
2	Doyumlu doğrusal fonksiyon	$f(x) = \begin{cases} -1 & \text{eğer } x \leq -1 \\ x & \text{eğer } -1 < x < 1 \\ 1 & \text{eğer } x \geq 1 \end{cases}$
3	İşaret fonksiyonu	$f(x) = \begin{cases} -1 & \text{eğer } x \leq 0 \\ 1 & \text{eğer } x > 0 \end{cases}$
4	Kare fonksiyon	$f(x) = x^2$
5	Kübik fonksiyon	$f(x) = x^3$
6	Lojistik sigmoid fonksiyon	$f(x) = \frac{1}{1 + e^{-x}}$
7	Hiperbolik tanjant fonksiyon	$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$
8	Radial tabanlı fonksiyon	$f(x) = e^{-x^2}$
9	Sinüs fonksiyon	$f(x) = \sin(x)$

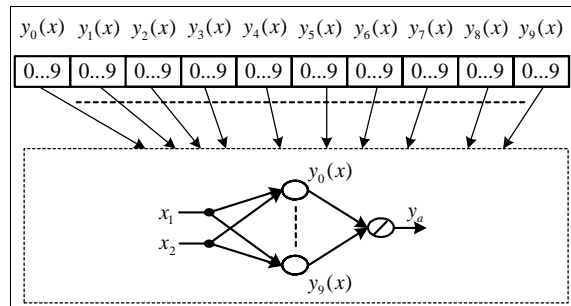
Tanımlamalarda değeri ağırlandırılmış düğüm girişlerini göstermektedir. ise etkinlik fonksiyonu çıkışı temsil etmektedir. Seçilen etkinlik fonksiyonlarının herhangi bir parametreye bağlı olmadıkları görülebilir. Çünkü fonksiyonlar üzerinde bir değişiklik yapılmadan en uygun kombinasyonlarını belirleyerek en optimum ağı yapısının elde edilmesi amaçlanmaktadır.

2. 3. Genetik Yapının Belirlenmesi

Genetik algoritmaların yapısı sırasıyla kromozom yapısının teşkil edilmesi, kromozomu oluştururken kullanılacak kodlama tekniğinin belirlenmesi, popülasyon büyüklüğünün belirlenmesi, uygunluk fonksiyonunun seçimi, ailelerin seçim yöntemi, genetik operatör tiplerinin belirlenmesidir. Bu yapısal ayarlamalar belirlendikten sonra genetik yapı oluşturulur. Bu yapı ve yöntemler kullanılarak algoritma çıkartılır.

2. 3. 1. Kromozom Yapısının Belirlenmesi

Popülasyon içerisinde bulunan bir kromozomun görünümü Şekil 2'de verilmiştir. Kromozom 2-10-1 olarak Şekil 1'de gösterilen sinir ağının gizli katmandaki etkinlik fonksiyonlarını temsil eden $y_0(x), \dots, y_9(x)$ toplam 10 gen den oluşturulmuştur. Klasik genetik algoritmalarda ikili kodlama tekniği kullanılmaktadır. Bu çalışmada genetik algoritma işlem süresinin düşürülmesi için gerçek sayılarla kodlama tercih edilmiştir. Kromozomdaki her bir genin tanımlanmasında 0 ile 9 arasında tam sayılar kullanılmıştır.



Şekil 2. Kromozom yapısı.

2. 3. 2. Popülasyonunun Belirlenmesi

Popülasyonun belirlenmesinde önemli olan popülasyonun kaç bireyden oluşturulacağı ve başlangıç değerlerinin nasıl seçileceğidir. Genetik algoritmalar tasarlanırken popülasyon büyük seçilirse, genetik algoritmanın istenen çözüme ulaşması daha uzun zaman alır. Tersine popüla-

syon çok küçük seçilirse bu durumda popülasyon içerisindeki bireylerin çeşitliliği düşeceği için lokal minimumda kalmasına neden olacaktır. Bu nedenle algoritma çevrim süresi dikkate alınarak mümkün olduğunca popülasyonun büyük seçilmesi önerilir. Bu çalışmada 20 bireyden oluşan bir popülasyon kullanılmıştır. Başlangıç değerleri bir problem konusunda geçmiş tecrübeler göre belirlenebileceği gibi rastgele olarak da atanabilir. Bu çalışmada popülasyonda ki her bir bireye ait gen değerleri 0 ile 9 arasında rastgele seçilmiştir.

2. 3. 3. Değerlendirme İşlemi

Uygunluk fonksiyonu hangi kromozomun bir sonraki nesilde bulunup bulunmayacağını gösteren parametredir. Bir kromozoma ait uygunluk değeri ne kadar yüksek olursa o kromozomun aile olarak seçimi daha güçlü bir olasılıktır. Bu çalışmada genetik algoritmanın kullanım amacı sinir ağındaki etkinlik fonksiyonlarını temsil eden en uygun kromozomun elde edilmesidir. Buna göre her bir kromozom dizisi için sinir ağı eğitilir ve K sayıda örnek için ileri beslemeli olarak çalıştırılarak hata değerleri elde edilir elde edilen hata değerlerinin toplamına bağlı olarak aşağıda verilen uygunluk değeri hesaplanır.

$$f = 1 / (1 + \sum_{k=1}^K |e(k)|) \quad (10)$$

Popülasyon içerisindeki tüm bireyler için uygunluk değerleri bulunur. Genetik algoritma ya daha önce belirlenen uygunluk değeri yakalanırsa sonlandırılır yada bu çalışmada olduğu gibi belirlenen iterasyon sayısı ile sonlandırılır. Her iki durumda da uygunluk değeri en yüksek olan birey saklanır böylece en uygun etkinlik fonksiyonları elde edilmiş olur.

2. 3. 4. Seçim İşlemi

Bir sonraki neslin üretilmesi için popülasyon içerisindeki bireyler arasından uygunluk değerlerine bağlı olarak aileler belirlenir. Bu çalışmada ailelerin seçiminde özel bir yöntem uygulanmıştır. Rank-based ve elitizm yöntemleri kullanılmıştır. Öncelikle uygunluk değerine göre popülasyon içerisindeki bireyler en yüksekten aşağıya doğru sıralanmıştır. En yüksek uygunluk değerine sahip birey hem aile olarak seçilmiş hem de doğrudan bir sonraki nesile dahil edilmiştir. Böylece en iyi performansı sağlayan birey algoritma tamamlanana kadar

korunması sağlanmıştır.

2. 3. 5. Üreme İşlemi

Üreme işleminde çaprazlama ve mutasyon olarak adlandırılan iki genetik operatör kullanılır.

- Çaprazlama: Çaprazlama işlemi bir sonraki neslin üretilmesi için kullanılan bir genetik işlemidir. Aile seçiminden elde edilen aileler kullanılarak yeni döller meydana getirilir. Burada dikkat edilmesi gereken husus popülasyon içerisindeki çeşitliliğin kaybolmaması ve bunun yanında uygun bireylerin çaprazlamada tamamen kaybedilmemesidir. Bu çalışmada bir sonraki nesil için gerekli bireylerin oluşturulması için çok noktadan çaprazlama işlemi gerçekleştirilmiştir. Çaprazlamaya girecek olan bireyler rastgele seçilmiş ve 0.75 olasılık değeri kullanılmıştır. Çaprazlama noktaları yine rastgele seçim ile belirlenmiştir.
- Mutasyon: Çaprazlama sonucu elde edilen bireylere popülasyondaki çeşitliliği artırmak için mutasyon uygulanarak yeni popülasyonun oluşumu tamamlanır. Bu genetik operatörde de mutasyon oranlarına dikkat edilmesi gerekir. Mutasyon oranı büyük seçilirse popülasyondaki bireylerin çeşitliliği artarken uygun bireylerin tamamen kaybedilmesine de yol açabilir. Her bir bireyin mutasyon olasılığı 0.1 olarak seçilmiştir.

2. 4. Genetik Sinir Ağı Algoritması

Genetik tabanlı sinir ağı algoritmasına ait akış diyagramı Şekil 3'de verilmiştir. Burada işlemler temel olarak başlangıç popülasyonunun tanımlanması, değerlendirme işlemi, seçim işlemi ve üreme işlemi olarak verilebilir.

Gösterilen akışa ait algoritma işlemleri aşağıda ayrıntılı olarak verilmiştir.

- Başlat,
- Başlangıç popülasyonunu üret,
- Her bir birey için sinir ağı algoritmasını tüm örnekler için eğitimini gerçekleştirerek uygunluk değerlerini hesapla. Uygunluk değerlerini sıralayarak en uygun olan bireyi belirle,

4. Belirlenen iterasyona aşıldığına bak. Eğer aşılmışsa en uygun bireyi saklayarak algoritmayı sonlandır. Değilse bir sonraki adımdan devam et,
5. Üreme için ailelerin seçimini gerçekleştir,
6. Çaprazlama ve mutasyon işlemleri ile yeni bireyleri üret,
7. Yeni nesil ile eskisini yer değiştir. Üçüncü adımdan devam et.

3. SİMÜLASYON SONUÇLARI

Bu bölümde, genetik tabanlı sinir ağı algoritmasının performansı test edilmektedir. Test aşamasında etkinlik fonksiyonların, genetik algoritmalarla ayarlanmasının tasarlanan ağ üzerindeki etkileri gözlenecektir.

3. 1. Test Düzenegi

Tasarlanan genetik tabanlı sinir ağının test edilmesi için ilk olarak Hwang ve arkadaşları (Hwang v.d., 1994) tarafından ortaya konulan ve daha sonra literatürde (Kwok ve Yeung, 1997; Ma ve Khorasani, 2005) kullanılan test fonksiyonları kullanılmıştır. Test aşamasında kullanılan non-linear test fonksiyonları 3 boyutlu olarak Şekil 4'de verilmiştir ve aşağıda olduğu gibi tanımlanmıştır,

Basit etkileşimli fonksiyon:

$$f^{(1)}(x_1, x_2) = 10.391((x_1 - 0.4)(x_2 + 0.6) + 0.36)$$

Radial fonksiyon:

$$f^{(2)}(x_1, x_2) = 24.234(r^2(0.75 - r^2))$$

Burada; $r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$

Eklemeli fonksiyon:

$$f^{(3)}(x_1, x_2) = 1.3356(1.5(1 - x_1) + e^{2x_1 - 1} \sin(3\pi(x_1 - 0.6)^2) + e^{3(x_2 - 0.5)} \sin(4\pi(x_2 - 0.9)^2))$$

Geliştirilen genetik tabanlı sinir ağının eğitimi için [0,1] aralığında düzgün dağılımlı 225 örnek çifti (x_{k1}, x_{k2}) kullanılmıştır. 3 fonksiyonun eğitiminde de aynı veri kümesi kullanılmış ve örnek kümenin tanımlaması aşağıda verilmiştir ,

$$y_k^{(j)} = f^{(j)}(x_{k1}, x_{k2}) \quad (11)$$

Burada; $k = 1, 2, \dots, 225$ ve $j = 1, 2, 3$.

Belirlenen veri kümesi ile genetik sinir ağının Bölüm 2'de belirtilen algoritmalar kullanılarak eğitiminden sonra elde edilen ağın test edilmesinde düzgün dağılımlı 10000 örnek çiftinden oluşan bir veri kümesi kullanılmıştır.

$$y_k^{(j)} = f^{(j)}(x_{k1}, x_{k2}) \quad (12)$$

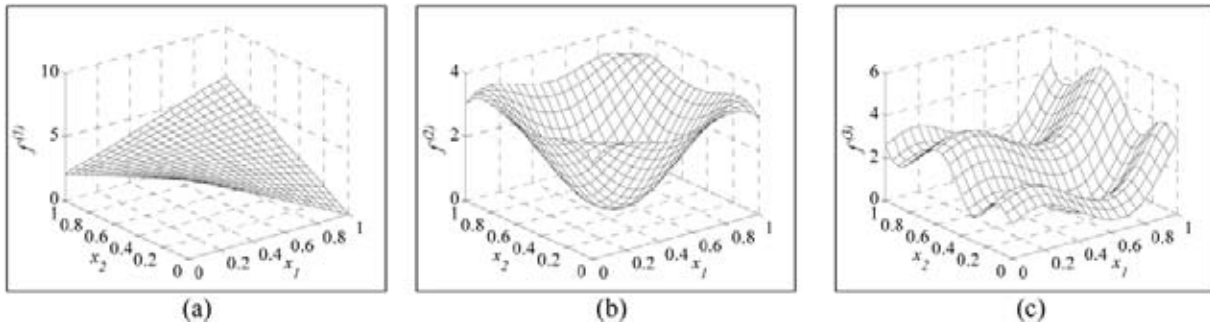
Burada; $k = 1, 2, \dots, 10000$ ve $j = 1, 2, 3$.

Eşitlik 12'deki bağımsız örnek kümesi kullanılarak ağın testi yapılırken (friction of variance unexplained-FVU) (Hwang v.d., 1994; Kwok ve Yeung, 1997; Ma ve Khorasani, 2005) FVU performansı ölçütü kullanılmıştır. FVU aşağıda olduğu gibi tanımlanır,

$$FVU = \frac{\sum_{k=1}^K (y_d(k) - y_o(k))^2}{\sum_{k=1}^K ((y_d(k) - \bar{y})^2)} \quad (13)$$

Burada, $\bar{y} = \frac{1}{K} \sum_{k=1}^K y_d(k)$.

FVU genel olarak kullanılan hataların kareleri ortalamasının oranı olarak ifade edilir.

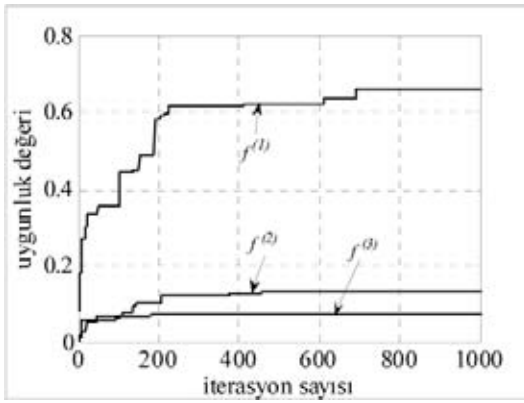


Şekil 4. Test fonksiyonlarının grafikleri (a) $f^{(1)}$ (b) $f^{(2)}$, (c) $f^{(3)}$.

3. 2. Test Sonuçları

Bölüm 2'de ayrıntılı olarak verilen genetik tabanlı sinir ağı algoritması ve yukarıda verilen test düzeneği kullanılarak simülasyon çalışmaları gerçekleştirilmiştir. Simülasyon çalışmaları C/C++ programı kullanılarak gerçekleştirilmiştir. Hem genetik algoritma hem de geri yayımlı sinir ağı algoritması 1000 iterasyon ile gerçekleştirilmiştir.

Şekil 5 genetik algoritma eğitimi esnasında 3 fonksiyon için uygunluk değerinin değişim grafiği verilmiştir. Uygunluk değerleri $f^{(1)}, f^{(2)}, f^{(3)}$ ve $f^{(3)}$ şeklinde elde edilmiştir.

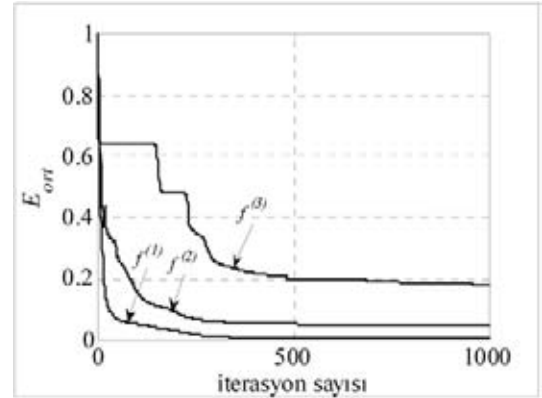


Şekil 5. Genetik algoritma eğitiminde iterasyona bağlı uygunluk değerleri.

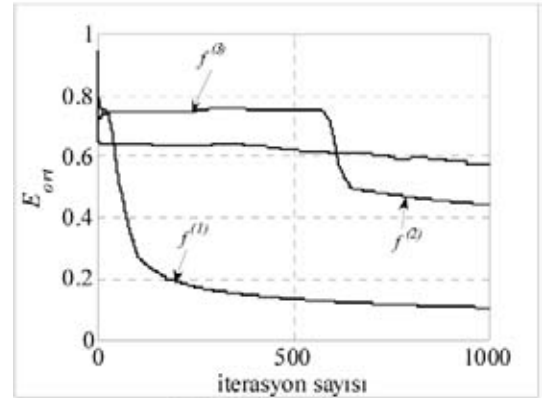
Şekil 6a'da genetik algoritma ile elde edilen en uygun kromozom kullanılarak gerçekleştirilen ağ eğitimleri için hata ölçütünün ortalamasının değişimi görülmektedir. Yine genetik algoritma için elde edilen sonuçlarla paralel olduğu açıkça görülmektedir. Ortaya konulan yöntemin klasik yöntemlerle kıyaslanabilmesi için ayrıca gizli katman düğümlerinde geleneksel etkinlik fonksiyonu olarak bilinen sigmoid fonksiyon kullanılmıştır. Şekil 6b'de sabit etkinlik fonksiyonları kullanılarak gerçekleştirilen ağ eğitimi için hata ölçütünün değişimi görülmektedir. Üç test fonksiyonuna göre, önerilen genetik sinir ağı ve klasik sinir ağı için elde edilen sonuçlar Tablo 2'de verilmiştir. Tablo 2'de gizli katmandaki her bir fonksiyon için katmanındaki düğüm fonksiyonlarına ait numaralar kromozom yapısı ve FVU değerleri verilmiştir. Örneğin, Tablo 1'de $f^{(1)}$ için genetik tabanlı sinir ağı ile elde edilen kromozom yapısında 3, 6, 6, 0, 2, 0, 5, 5, 2, 4 değerleri görülmektedir.

Bölüm 2'de verilen Tablo 1'e bakılırsa belirlenen kromozomdaki genler sırasıyla 3(işaret), 6(sig-

moid), 6(sigmoid), 0(pozitif doğrusal), 2(doyumlu doğrusal), 0(pozitif doğrusal), 5(küçük), 5(küçük), 2(doyumlu doğrusal), 4(kare) fonksiyonlarından oluşmaktadır. Görüldüğü gibi ağ altı farklı etkinlik fonksiyonundan oluşmaktadır. FVU değeri de 0.000014 olarak elde edilmiştir. Klasik sistemde ise tüm etkinlik fonksiyonları 6 (sigmoid) olduğu görülmektedir. Klasik yöntemde kromozom söz konusu olmasa da kıyaslama için kromozom şeklinde verilmiştir. Her bir fonksiyon için ağ eğitimi gerçekleştirilerek uygun bağlantı ağırlık değerleri elde edilmiştir. Bağımsız test verileri kullanılarak da her bir fonksiyon için FVU değerleri hesaplanmıştır.



(a) Genetik tabanlı sinir ağı.



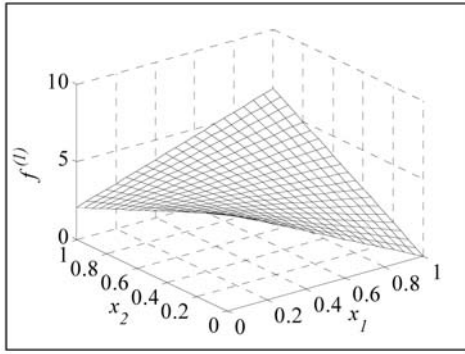
(b) Klasik sinir ağı.

Şekil 6. İterasyona göre hata değişimleri.

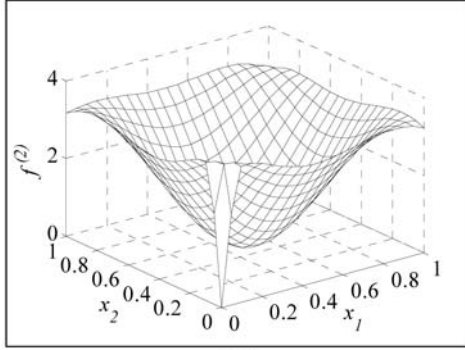
Tablo 2. Klasik ve önerilen genetik sinir ağı için test sonuçları.

Test fonk.	Genetik tabanlı sinir ağı		Klasik sinir ağı	
	Kromozom yapısı	FVU	Kromozom yapısı	FVU
$f^{(1)}$	3,6,6,0,2,0,5,5,2,4	0.000014	6,6,6,6,6,6,6,6,6,6	0.085572
$f^{(2)}$	4,9,9,9,5,2,5,5,9,7	0.010651	6,6,6,6,6,6,6,6,6,6	1.461816
$f^{(3)}$	9,1,0,0,1,7,1,9,9,9	0.057261	6,6,6,6,6,6,6,6,6,6	1.358987

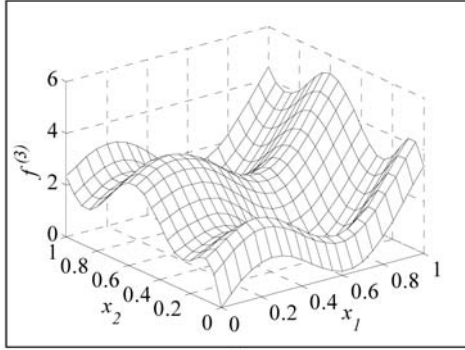
Şekil 5 ve 6'da eğitim aşamasında verilen hata yaklaşım eğrileri ile Tablo 2'de verilen FVU değerleri arasında doğrudan ilişki olduğu görülmektedir. Buna göre, yine FVU sıralaması $f^{(1)}$, $f^{(2)}$, ve $f^{(3)}$ şeklinde olmaktadır. Yine sabit yapıli klasik sinir ağında FVU ölçütü $f^{(1)}$, $f^{(2)}$, ve $f^{(3)}$ şeklinde gerçekleşmiştir. FVU oranlarına bakıldığında klasik yapı ile önerilen yapı arasında oldukça büyük farkların olduğu görülmektedir. Örneğin $f^{(1)}$ fonksiyonunu ele alacak olursak FVU oranı aynı veriler için yaklaşık sıfır iken klasik yapıda 0.085872 değerlerinde olduğu görülmektedir. Diğer iki fonksiyon içinde benzer sonuçların olduğu elde görülmektedir.



(a)



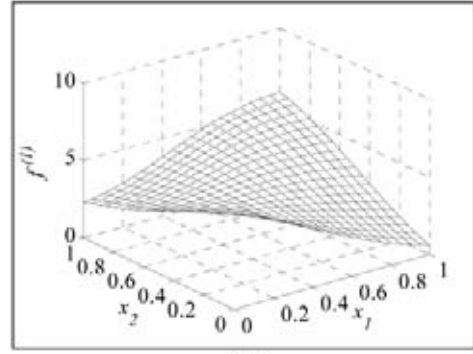
(b)



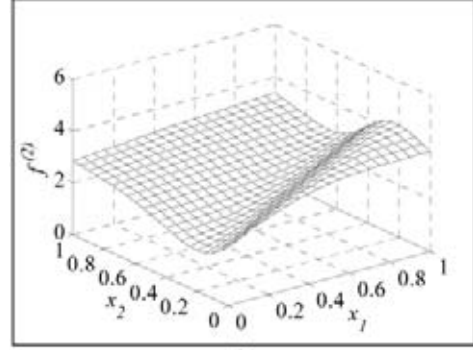
(c)

Şekil 7. Genetik tabanlı sinir ağı ile elde edilen sonuçlar (a) $f^{(1)}$ (b) $f^{(2)}$ (c) $f^{(3)}$.

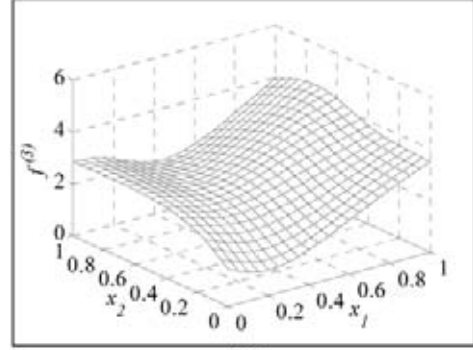
Tablo 2 ve Şekil 7-8'de verilen grafiklerden de görüleceği gibi ortaya konulan ağda parametre değişimine sahip (Hermite, Gauss gibi) özel tanımlı bir fonksiyon kullanılmamasına rağmen oldukça yüksek bir doğrulukta fonksiyonlara yaklaşıldığı görülmektedir. Bu tip çalışmaların başlangıçta tasarımı ve ilk modeli belki klasik sinir ağı tasarımına göre daha zahmetli olabilir. Fakat ağdan elde edilecek performans düşünüldüğünde bu tip bir ağ tasarımının vazgeçilemez olduğu da görülmektedir.



(a)



(b)



(c)

Şekil 8. Klasik sinir ağı ile elde edilen sonuçlar (a) $f^{(1)}$ (b) $f^{(2)}$ (c) $f^{(3)}$.

4. SONUÇLAR

Bu çalışmada genetik algoritma kullanılarak ileri beslemeli bir sinir ağının gizli katmanındaki etkinlik fonksiyonları belirlenmiştir. Etkinlik fonksiyonları olarak literatürde sıkça kullanılan ve parametre değişimi olmayan basit fonksiyonlar tercih edilmiştir. Geliştirilen genetik tabanlı sinir ağı ve klasik sabit yapıdaki sinir ağı için test fonksiyonları kullanılarak testler

gerçekleştirilmiştir. Elde edilen sonuçlar ortaya konulan genetik tabanlı sinir ağının klasik ileri beslemeli sinir ağına göre fonksiyonlara yaklaşımda çok daha yüksek bir performans sağladığını göstermektedir. Sonuç olarak, hemen hemen aynı matematiksel yüke sahip olan iki ağdan daha iyi performans elde edilebilmesi için etkinlik fonksiyonlarının uygun bir biçimde belirlenmesi ağın etkinliğinde önemli bir faktör olduğu burada rahatlıkla söylenebilir.

KAYNAKLAR

Angeline, P. J., Saunders, G. M. and Pollack, J. B. 1994. An Evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*. 5 (1), 54-65.

Arifovica, J. and Gencay, R. 2001. Using genetic algorithms to select architecture of a feedforward artificial neural network. *Physica A*. 289 (3-4), 574-594.

Blanco, A., Delgado, M. and Pegalajar, M. C. 2001. A real-coded genetic algorithm for training recurrent neural networks. *Neural Networks*. 14 (1), 93-105.

Daqi, G. and Genxing, Y. 2003. Influences of variable scales and activation functions on the performances of multi-layer feedforward neural networks. *Pattern Recognition*. 36 (4), 869 - 878.

Ferentinos, K. P. 2005. Biological engineering applications of feedforward neural networks designed and parameterized by genetic algorithms. *Neural Networks*. 18 (7), 934-950.

Guarnieri, S., Piazza, F. and Uncini, A. 1999. Multilayer feedforward networks with adaptive spline activation function. *IEEE Transactions on Neural Networks*. 10 (3), 672-683.

Hwang, J. N. S., Lay, R. Maechler, M., Martin, R. D. & Schimert, J. 1994. Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks*. 5 (3), 342-353.

Kwok, T. Y. and Yeung, D. Y. 1997. Objective functions for training new hidden units in constructive neural networks. *IEEE Transactions on Neural Networks*. 8 (5), 1131-1148.

Leung, F. H., Lam, F. H., Ling, K. S. H. and Tam, P. K. S. 2003. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*. 14 (1), 79-88.

Ma, L. and Khorasani, K. 2005. Constructive feedforward neural networks using hermite polynomial activation functions. *IEEE Transactions on Neural Networks*. 16 (4), 821-833.

Marwala, T. 2007. Bayesian training of neural networks using genetic programming. *Pattern Recognition Letters*. 28 (12), 1452-1458.

Mayer, H. A. and Schwaiger, R. 2002. Differentiation of neuron types by evolving activation function templates for artificial neural Networks. In *Proceedings of the International Joint Conference on Neural Networks 12-17 May 2002*. Honolulu, Hawaii. Vol. 2, 1773-1778.

Oha, S. K. and Pedrycz, W. 2006. Multi-layer self-organizing polynomial neural networks and their development with the use of genetic algorithms. *Journal of the Franklin Institute*. 343 (2), 125-136.

Pedrajas, N. G., Boyer, D. O. and Martinez, C. H. 2006. An alternative approach for neural network evolution with a genetic algorithm: crossover by combinatorial optimization. *Neural Networks*. 19 (4), 514-528.

Sexton, R. S. and Gupta, J. N. D. 2000. Comparative evaluation of genetic algorithm and backpropagation for training neural networks. *Information Sciences*, 129 (1-4), 45-59.

Ustun, O. 2009a. A Nonlinear full model of switched reluctance motor with artificial neural network. *Energy Conversion and Management*. (50), 2413-2421.

Ustun, O. 2009b. Measurement and real-time modeling of inductance and flux linkage in switched reluctance motors. *IEEE Transactions on Magnetics*. (Accepted paper).

Wang, C., Qin, S.Y. and Wan, B.W. 1991. A novel neural network structure with fast convergence based on optimizing combination of different activation function. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 31 Oct-3 Nov 1991. Orlando, Florida, Vol. 13 (3), 1399-1400.

Wong, K. W., Leung, C.S. and Chang, S. J. 2002. Use of periodic and monotonic activation functions in multilayer feedforward neural networks trained by extended kalman filter algorithm. *IEE Proc. Vis. Image Signal Process*. 149 (4), 217-224.

Xu, S. and Zhang, M. 2001. A novel adaptive activation function. In *Proceedings of the International Joint Conference on Neural Networks*, 15-19 July 2001. Washington, DC, Vol. 4, 2779-2782.

Yao, X. 1999. Evolving artificial neural networks. *Proceedings of the IEEE*. 9 (87), 1423-1447.