

GERİ-YAYILMALI ÖĞRENME ALGORİTMASINDAKİ ÖĞRENME PARAMETRELERİNİN GENETİK ALGORİTMA İLE BELİRLENMESİ

Oğuz ÜSTÜN *, İlker YILDIZ **

Özet : Bu çalışmada, ileri beslemeli bir sinir ağının eğitiminde kullanılan geri-yayılımlı öğrenme algoritmasındaki öğrenme parametreleri genetik algoritmalar kullanılarak belirlenmiştir. Öğrenme parametreleri öğrenme ve momentum katsayıları olarak bilinmektedir. Öğrenme parametreleri ağın öğrenme hızının artırılması, öğrenme esnasında oluşabilecek osilasyonların giderilmesi ve lokal minimumlardan kaçılması gibi özellikleri belirlemektedirler. Dolayısıyla bu parametrelerin uygun biçimde seçilmesi ağın daha etkin olarak eğitilmesinde oldukça önemlidir. Öğrenme parametrelerinin genetik algoritma ile belirlenmesi için, dört katmanlı ileri beslemeli bir ağ tasarlanmıştır. Tasarlanan ağdaki üç öğrenme ve üç momentum katsayısı, genetik bir kromozom ile ifade edilmiştir. Çalışmanın amacı; en uygun kromozomun seçilmesidir. Ortaya konulan yöntemin test edilmesinde özel tanımlı iki boyutlu regresyon problemlerinden yararlanılmıştır. Yapılan test çalışması ortaya konulan yöntemin geleneksel sabit parametrelili öğrenme algoritmasına göre daha etkin olduğunu göstermiştir.

Anahtar kelimeler: Geri-yayılımlı öğrenme algoritması, öğrenme parametreleri, ileri beslemeli sinir ağları, genetik algoritmalar, öğrenme katsayısı, momentum katsayısı.

DETERMINATION OF THE LEARNING PARAMETERS IN BACKPROPAGATION LEARNING ALGORITHM BY GENETIC ALGORITHM

Abstract : In this study, the learning parameters in backpropagation learning algorithm, which is used for training of a feedforward neural network are determined by using genetic algorithms. Learning parameters are known as learning rate and momentum rate. The learning parameters identify the features such as improving of the learning speed, eliminating the oscillations and avoiding from the local minima during learning of the network. Therefore, a selection of these parameters is quite important for training of the network more efficiently. A feedforward neural network with four layers is designed to define learning parameters by genetic algorithm. Three learning rates and three momentum rates in the designed network have been denoted by single genetic chromosome. The aim of the study is to choose the ideal chromosome. In order to test the proposed method, a specific described two dimensional regression problems are used. Test results show that the suggested method is more efficient than the conventional learning algorithm with fixed parameter.

Key Words: Backpropagation learning algorithm, learning parameters, feedforward neural networks, genetic algorithms, learning rate, momentum rate.

* Abant İzzet Baysal Üniversitesi Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Bolu. E-posta: oguzustun@ibu.edu.tr

** Abant İzzet Baysal Üniversitesi Bolu Meslek Yüksekokulu, Teknik Programlar Bölümü, Bolu. E-posta: yildiz_i2@ibu.edu.tr

1. Giriş

İleri beslemeli sinir ağları, yapılarının esnek olması, iyi bir gösterim kapasitesi ve çok sayıda öğrenme algoritmasına sahip olması nedeniyle en popüler mimari haline gelmiştir (Kwok ve Yeung, 1997). Geriyayılmalı danişmalı öğrenme yöntemi ileri beslemeli ağların eğitiminde en çok tercih edilen öğrenme yöntemidir. Etkinlik fonksiyonları yanında öğrenme ve momentum katsayılarının uygun biçimde belirlenmesi de ağın performansı açısından önemli bir faktördür. Öğrenme katsayısının düşük seçilmesi öğrenme hızını yavaşlatırken yüksek seçilmesi de öğrenme esnasında büyük osilasyonlara sebep olup en nihayetinde öğrenme işleminin gerçekleşmemesine neden olmaktadır. Uygulamalar göstermiştir ki momentum katsayısının küçük tutulması osilasyonları azaltırken öğrenme hızında düşüslere sebep olmaktadır. Büyük değerli olması ise kararsızlıklara ve hedeften uzaklaşılmasına sebep olmaktadır. Bu nedenle momentum katsayısının belirlenmesi kendi başına önemli bir sorundur (Kandi1 vd.,1193).

Öğrenme parametrelerin belirlenmesinde literatürde dikkat çekici çalışmalar gerçekleştirilmiştir. Yu vd. (1993) geri-yayılmalı öğrenme algoritmasının hızını artırmak için öğrenme ve momentum katsayılarını nümerik yöntemler kullanarak dinamik olarak ayarlamışlardır. Sheell vd. (2007) ağı istenen ve çıkışı arasındaki hatanın işaretine bağlı olarak heuristik bir yaklaşımla öğrenme katsayılarını ayarlamışlar. Yu ve Liu (2002) ağı istenen ve gerçek çıkışı arasındaki hata ve hatanın işaretine bağlı olarak öğrenme ve momentum katsayılarını üssel fonksiyon olarak ifade etmişler. Böylece hataya bağlı adaptif bir yaklaşım ortaya koymuşlar. Yu ve Chen (1997) hata ölçütüne göre öğrenme ve momentum katsayılarının birinci ve ikinci dereceden kısmi türevlerinden yararlanarak dinamik olarak öğrenme parametrelerini elde etmişlerdir. Zaghwl ve Dong (1994) öğrenme ve momentum katsayılarını belirlemede eşlenik eğim metodunu (conjugate gradient method) kullanmışlardır. Kamiyama vd. (1992) farklı öğrenme ve momentum katsayılarına bağlı verilere dayanarak momentum ve öğrenme katsayısını lineer bir eşitlik halinde tanımlamışlardır. Bu çalışmalarda ağ için sadece birer öğrenme ve momentum katsayısı kullanılmıştır. Buna rağmen iki parametrenin dinamik olarak hesaplanması için oldukça yüksek bir hesaplama yükü olduğu görülmektedir. Daha çok sayıda örneğin katman sayısına göre öğrenme ve momentum katsayılarının hesaplanması durumunda bu hesaplama yükü ile parametrelerin belirlenmesi algoritmanın çalışma hızını oldukça çok düşürecektir. Bu nedenle gerçek zamanlı sistemlerde bu yaklaşımların kullanılması oldukça zordur.

Evrim algoritmaları, doğal evrim mantık ve prensiplerine göre geliştirilen popülasyon tabanlı stokastik araştırma algoritmalarıdır. EA'lar özellikle birçok lokal optimumun üretildiği büyük kompleks problemlerin çözümünde faydalıdır. Klasik gradient tabanlı araştırma algoritmalarına göre lokal minimuma takılma olasılıkları daha azdır. EA'lar gradient bilgisine bağlı değildirler ve bu nedenle bu tip bilginin elde edilmesinin çok zor olduğu problemlerin çözülmesi için oldukça elverişlidirler. EA'lar tam bir objektif fonksiyonun olup olmamasına bakmaksızın da çözüm üretebilirler (Yao, 1999). Bu özelliklerinden dolayı genetik algoritmalar sinir ağlarının optimize edilmesinde sıkça kullanılmaktadır. Ustun (2009a) ileri beslemeli sinir ağlarındaki etkinlik fonksiyonlarının belirlenmesinde genetik algoritmaları kullanmıştır. Blanco vd. (2001) recurrent sinir ağlarının eğitilmesinde kullanılan eğim algoritmaları yerine genetik programlama kullanarak ağ eğitimini gerçekleştirmişlerdir. Marwala (2007), Markov zinciri tabanlı Monte Carlo tekniğini genetik programlamada kullanarak bayesian sinir ağının eğitilmesini gerçekleştirmiştir. Sexton ve Gupta (2000) sinir ağlarının eğitilmesinde genetik algoritmaları kullanılmışlardır. Leung vd. (2003) ileri beslemeli sinir ağlarındaki bağlantı sayılarının optimize edilmesinde genetik algoritmalar kullanmışlardır. Angeline vd. (1994) recurrent sinir ağının yapısal düzenlemesi ve bağlantı

ağırlıklarının hesaplanmasında genetik algoritmaları kullanmışlar. Pedrajas vd. (2006) recurrent sinir ağları için benzer bir çalışma gerçekleştirmişler. Arifovica ve Gencay (2001) ileri beslemeli ağların düğüm sayısı katman sayısı gibi yapısına yönelik parametrelerin belirlenmesinde genetik algoritmaları kullanmışlardır.

Bu makale, genetik algoritmalar kullanılarak ileri beslemeli yapay sinir ağlarındaki öğrenme ve momentum katsayılarının belirlenmesinde yeni bir yöntem ortaya koyar. Ortaya konulan bu çalışmayla, aynı örnek sayısında ağın en kısa zamanda ve en iyi doğrulukta sonuç vermesi sağlanmaktadır. Öğrenme parametreleri öğrenme ve momentum katsayıları olarak bilinmektedir. Öğrenme parametreleri ağın öğrenme hızının artırılması, öğrenme esnasında oluşabilecek osilasyonların giderilmesi ve lokal minimumlardan kaçılması gibi özellikleri belirlemektedirler. Dolayısıyla bu parametrelerin uygun biçimde seçilmesi ağın daha etkin biçimde eğitilmesinde oldukça önemlidir. Bu çalışmada 4 katmanlı ileri beslemeli bir ağdaki 3 öğrenme ve 3 momentum katsayısı bir genetik kromozom ile ifade edilmiştir. En uygun kromozom, buna bağlı olarak en uygun öğrenme parametreleri, genetik algoritma ile belirlenmiştir. Ortaya konulan yöntemin test edilmesinde özel tanımlı 2 boyutlu regresyon problemlerinden yararlanılmıştır. Yapılan test çalışması ortaya konulan yöntemin geleneksel sabit parametrelili öğrenme algoritmasına göre daha etkin olduğunu göstermiştir.

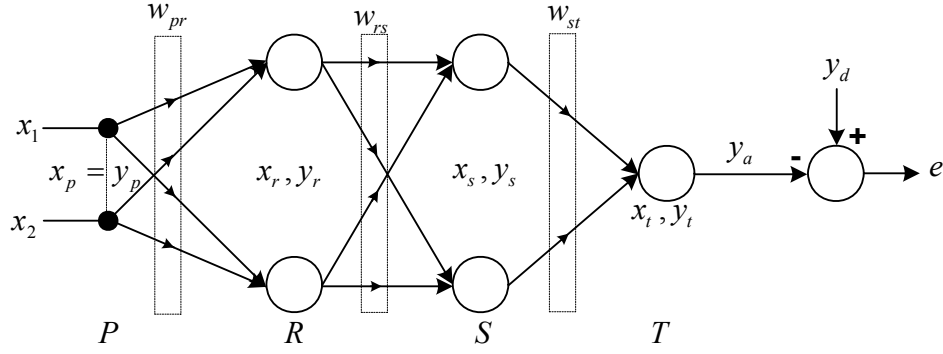
Çalışmanın bundan sonraki bölümü şu şekilde düzenlenmiştir. İkinci bölümde genetik tabanlı sinir ağ modeli ve algoritması anlatılmış, üçüncü bölümde test çalışması gerçekleştirilmiş ve son bölümde genel bir değerlendirme yapılmıştır.

2. Genetik Algoritmanın Oluşturulması

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

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 dört katmanlı (P, R, S, T) klasik bir ileri beslemeli ağ yapısı tercih edilmiştir. Benzer ağ modelleri ayrıntılı olarak kaynak (Ustun, 2009b; Ustun, 2009c)'de görülebilir. Tasarlanan ağ yapısı Şekil 1'de verilmiştir. Ağ iki girişli ve tek çıkışlıdır. 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.

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

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

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

$$x_t = \sum_{s=0}^S y_s \cdot w_{st} \text{ ve } y_t = y_n(x_t) \quad (4)$$

Burada ağın gerçek çıkış değeri $y_a = y_t$ Eşitlik 4 ile bulunur.

2.1.2. Geri-yayılım algoritması

Danışmanlı eğitim tabanlı geri yayımlı öğrenme algoritması için Eşitlik 5’de verilen enerji fonksiyonuna bağlı olarak, hata değerleri geriye yayılır ve Δw ağırlık değişimleri aşağıda olduğu gibi bulunur. Burada η öğrenme ve α momentum katsayısıdır.

$$E(k) = \frac{1}{2} e^2(k), \quad E_{ort} = \frac{1}{K} \sum_{k=1}^K E(k) \quad (5)$$

$$e(k) = y_d(k) - y_a(k) \quad (6)$$

$$\Delta w_{st}(k) = \alpha_{st} \Delta w_{st}(k-1) + \eta_{st} \left(-\frac{\partial E}{\partial w_{st}} \right) \quad (7)$$

$$\Delta w_{rs}(k) = \alpha_{rs} \Delta w_{rs}(k-1) + \eta_{rs} \left(-\frac{\partial E}{\partial w_{rs}} \right) \quad (8)$$

$$\Delta w_{pr}(k) = \alpha_{pr} \Delta w_{pr}(k-1) + \eta_{pr} \left(-\frac{\partial E}{\partial w_{pr}} \right) \quad (9)$$

$$w_{st}(k+1) = w_{st}(k) + \Delta w_{st}(k) \quad (10)$$

$$w_{rs}(k+1) = w_{rs}(k) + \Delta w_{rs}(k) \quad (11)$$

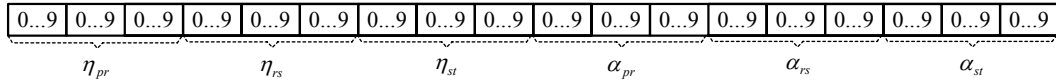
$$w_{pr}(k+1) = w_{pr}(k) + \Delta w_{pr}(k) \quad (12)$$

2.2. Genetik yapının belirlenmesi

Genetik algoritmaların yapısını belirleyen özellikler 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 (Ustun, 2009a).

2.2.1. Kromozom yapısının belirlenmesi:

Popülasyon içerisinde bulunan bir kromozomun görünümü Şekil 2’de verilmiştir. Kromozom sinir ağındaki 3 öğrenme ve 3 momentum katsayısının her biri için 3 gen olmak üzere toplam 18 genden meydana gelmiştir. 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

2.2.2. Popülasyonunun belirlenmesi

Popülasyonun belirlenmesi iki önemli nokta vardır. Bunlardan birincisi hangi büyüklükte olacağı ikincisi ise başlangıç değerlerinin nasıl seçileceğidir. Genetik algoritmalar tasarlanırken popülasyon büyük seçilirse, genetik algoritmanın istenen çözüme ulaşması daha uzun zaman alır. Tersine popülasyon çok küçük seçilirse bu durumda popülasyon içerisindeki bireylerin çeşitliliği düşeceği için lokal minimumda kalmasına neden olacaktır. Bu nedenle algoritma çevrim süresine göre 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 (Ustun, 2009a).

2.2.3. Değerlendirme

Uygunluk fonksiyonu her bir kromozomun bir sonraki jenerasyonda 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 öğrenme ve momentum katsayılarını 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 (Ustun, 2009a).

$$f = \frac{1}{1 + \sum_{k=1}^K |e(k)|} \quad (13)$$

2.2.4. Aile seçimi

Bir sonraki jenerasyonun üretilmesi için popülasyon içerisindeki bireyler arasından uygunluk değerlerine bağlı olarak aileler belirlenir. Bu çalışmada öncelikle uygunluk değerine göre popülasyon içerisindeki bireyler en yüksekte aşağıya doğru sıralanmıştır. En yüksek uygunluk değerine sahip birey hem aile olarak seçilmiş hem de doğrudan bir sonraki jenerasyona dahil edilmiştir. Böylece en iyi performansı sağlayan bireyin algoritma tamamlanana kadar korunması sağlanmıştır (Ustun, 2009a).

2.2.5. Üreme yöntemi

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

a. Çaprazlama

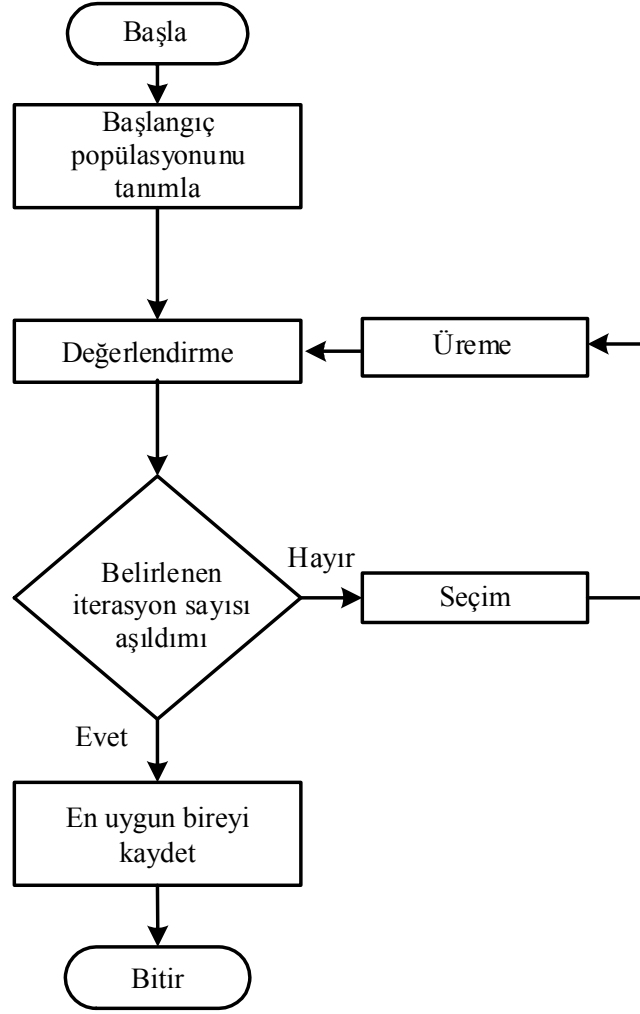
Çaprazlama işlemi bir sonraki neslin üretilmesi için kullanılan bir genetik işlemdir. Aile seçiminden elde edilen aileler kullanılarak yeni döller meydana getirilir. Burada dikkat edilmesi gereken husus popülasyon içerisindeki çeşitliliğin kaybolmaması ve bunun tersine uygun bireylerin çaprazlamada tamamen 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 (Ustun, 2009a).

b. Mutasyon

Çaprazlama sonucu elde edilen bireylere popülasyondaki çeşitliliği artırmak için mutasyon uygulanarak yeni popülasyonun oluşumu tamamlanır. Mutasyon oranı büyük seçilirse popülasyondaki bireylerin çeşitliliği artarken uygun bireylerin tamamen kaybedilmesine de yol açabilir. Her bir bireyin mutasyon olasılığı 0.1 olarak seçilmiştir (Ustun, 2009a).

2.3. Genetik algoritma

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. Akıştan da görüleceği gibi amaç problemin çözümüne yönelik en uygun kromozomun seçilmesidir. Gösterilen akışa ait algoritma işlemleri aşağıda ayrıntılı olarak verilmiştir (Ustun, 2009a).



Şekil 3. Geliştirilen genetik algoritma akış diyagramı

1. Başlat
2. Başlangıç popülasyonunu üret.
3. Her bir birey için sinir ağı algoritmasını tüm örnekler için eğitimini gerçekleştirerek uygunluk değerlerini hesapla. Uygunluk değerlerini sıralayarak en uygun olan bireyi belirle.
4. İstenen uygunluk değerinin yakalanıp yakalanmamasına bak. Eğer yakalanmışsa 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 jenerasyonla eskileri yer değiştir. Üçüncü adımdan devam et.
8. Dur

3. Simülasyon Sonuçları

Bu bölümde, önerilen algoritma ve geleneksel algoritmanın performansı test edilmektedir. Test aşamasında öğrenme parametrelerinin, genetik algoritmalarla ayarlanmasının tasarlanan ağ üzerindeki etkileri gözlenecektir.

3.1. Test düzeneği

Test aşamasında ilk olarak Hwang vd. (1994) tarafından ortaya konulan ve daha sonra literatürde (Kwok ve Yeung, 1997; Ustun, 2009a) yer alan test fonksiyonları kullanılmıştır. Kullanılan non-linear test fonksiyonları 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 $r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$ dir.

Harmonik fonksiyon:

$$f^{(3)}(x_1, x_2) = 42.659(0.1 + \tilde{x}_1(0.05 + \tilde{x}_1^4 - 10\tilde{x}_1^2\tilde{x}_2^4 + 5\tilde{x}_2^4))$$

Burada $\tilde{x}_1 = x_1 - 0.5$, $\tilde{x}_2 = x_2 - 0.5$ dir.

Eklemeli fonksiyon:

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

Karmaşık etkileşimli fonksiyon:

$$f^{(5)}(x_1, x_2) = 1.9(1.35(1 - x_1) + e^{x_1} \sin(13(x_1 - 0.6)^2) + e^{-x_2} \sin(7x_2))$$

Geliştirilen genetik tabanlı sinir ağının performansının test edilmesinde test ve eğitim için $[0,1]$ aralığında düzgün dağılımlı 225 örnek çifti (x_{k1}, x_{k2}) kullanılmıştır. 5 fonksiyonun eğitiminde de aynı veri kümesi kullanılmış ve örnek kümenin tanımlaması aşağıda verilmiştir,

$$y_k^{(j)} = f^{(j)}(x_{k1}, x_{k2}) \quad (14)$$

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.

$$y_k^{(j)} = f^{(j)}(x_{k1}, x_{k2}) \quad (15)$$

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

Eşitlik 15’deki bağımsız örnek kümesi kullanılarak ağın testi yapılırken (friction of variance unexplained - FVU) (Hwang vd., 1994; Kwok ve Yeung, 1997; Ustun, 2009a) FVU performans ölçütü kullanılmıştır. FVU aşağıda olduğu gibi tanımlanır,

$$FVU = \frac{\sum_{k=1}^K (y_d(k) - y_a(k))^2}{\sum_{k=1}^K ((y_d(k) - \bar{y})^2)} \quad (16)$$

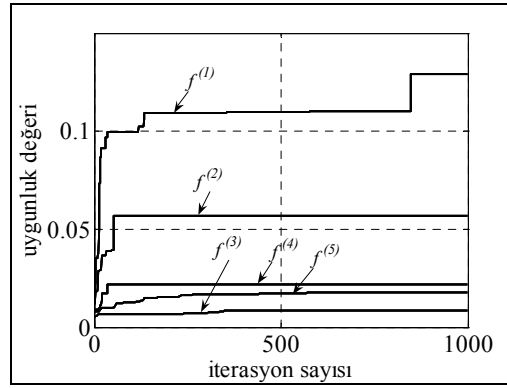
burada, $\bar{y} = \frac{1}{K} \sum_{k=1}^K y_d(k)$ dır.

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

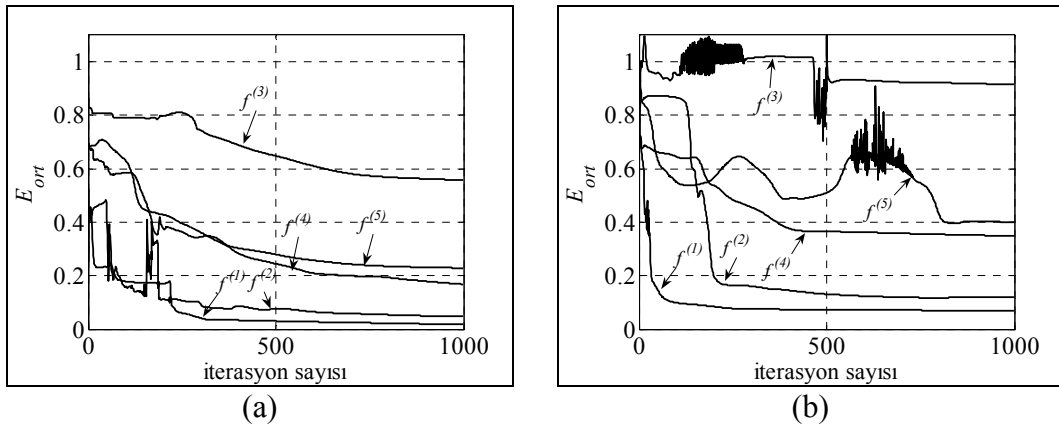
3.2. Test sonuçları

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

Şekil 4’te genetik algoritma eğitimi esnasında 5 fonksiyon için uygunluk değerinin değişim grafiği verilmiştir. Uygunluk değerleri $f^{(1)}$, $f^{(2)}$, $f^{(4)}$, $f^{(5)}$ ve $f^{(3)}$ şeklinde elde edilmiştir. Şekil 5.a’ 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. Şekil 6.b’ de geleneksel algoritma kullanılarak gerçekleştirilen ağ eğitimi için hata ölçütünün ortalamasının değişimi görülmektedir.



Şekil 4. Genetik algoritma eğitiminde iterasyona bağlı uygunluk değerleri



Şekil 5. İterasyona göre hata değişimleri (a) Genetik algoritma (b) Geleneksel algoritma

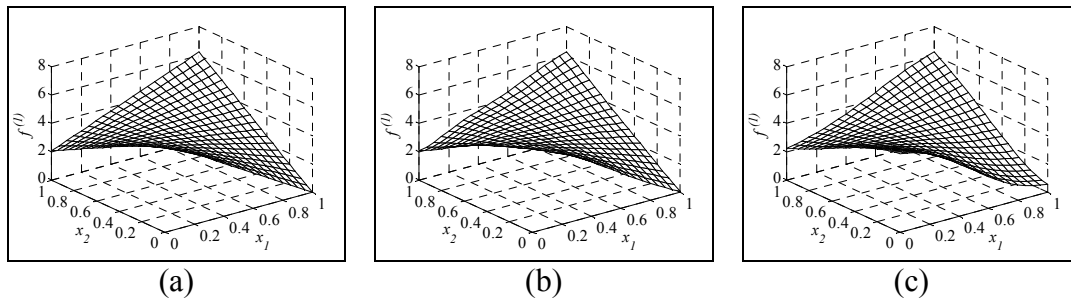
Beş test fonksiyonuna göre, önerilen genetik algoritma ve geleneksel algoritma için elde edilen sonuçlar Tablo 1’de verilmiştir. Tablo 1’de her bir öğrenme parametresi için kromozom yapısı ve FVU değerleri verilmiştir.

Tablo 1: Genetik ve geleneksel algoritma için test sonuçları

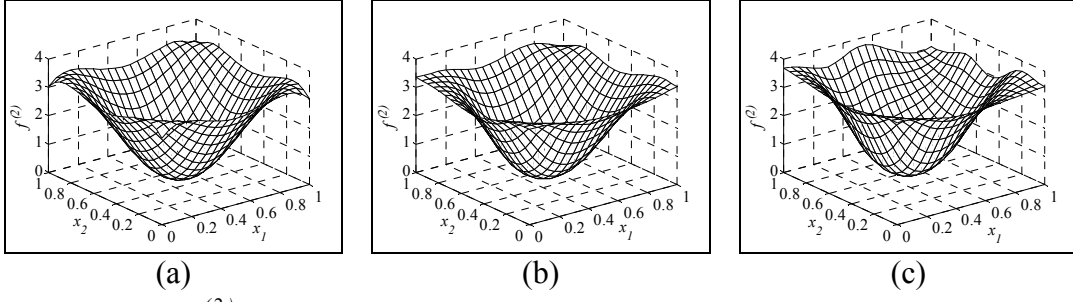
Test fonksiyonları	Genetik algoritma ile belirlenen öğrenme parametreleri				Geleneksel algoritma ile belirlenen öğrenme parametreleri						
	kromozom yapısı				FVU	kromozom yapısı				FVU	
η_{pr}	η_{rs}	η_{st}	α_{pr}	α_{rs}		α_{st}	η_{pr}	η_{rs}	η_{st}		α_{pr}
$f^{(1)}$	0,1,2,0,7,0,5,6,7,3,1,7,8,5,9,1,1,0	0.001293	0,5,0,0,0,2,0,5,2,0,0,4,0,5,5,0,2,7	0.025294							
$f^{(2)}$	9,2,1,5,9,7,5,1,6,4,1,0,2,9,8,8,0,1	0.007534	0,5,0,0,0,2,0,5,2,0,0,4,0,5,5,0,2,7	0.084936							
$f^{(3)}$	2,0,0,6,1,9,5,1,0,7,9,8,2,5,2,2,3,6	0.483642	0,5,0,0,0,2,0,5,2,0,0,4,0,5,5,0,2,7	26.27465							
$f^{(4)}$	1,1,0,9,7,7,4,3,0,1,1,9,3,1,3,9,5,6	0.057192	0,5,0,0,0,2,0,5,2,0,0,4,0,5,5,0,2,7	0.965102							
$f^{(5)}$	0,2,0,7,1,2,9,5,0,0,1,6,8,7,3,2,7,0	0.093339	0,5,0,0,0,2,0,5,2,0,0,4,0,5,5,0,2,7	3.399665							

Örneğin, Tablo1’de $f^{(1)}$ için genetik tabanlı sinir ağı ile elde edilen kromozom yapısında **0,1,2,0,7,0,5,6,7,3,1,7,8,5,9,1,1,0** değerleri görülmektedir. Şekil 2’de gösterildiği gibi her üç bit bir öğrenme parametresine karşılık gelmektedir. Buna göre, $\eta_{pr} = 210/1000 = 0.21$ dir. Diğer parametreler de benzer şekilde hesaplanır.

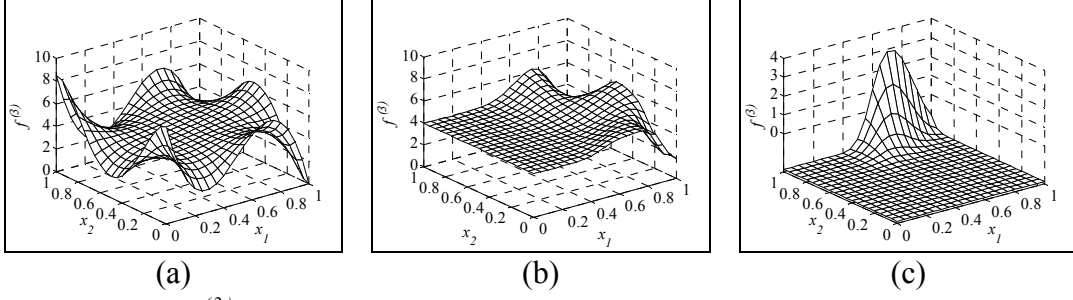
Her bir fonksiyon için ağ eğitimi gerçekleştirilerek uygun bağlantı ağırlık değerleri elde edilmiştir. Bağımsız test verileri kullanılarak da her bir fonksiyon için FVU değerleri hesaplanmıştır. Şekil 4, 5’de eğitim aşamasında verilen hata yaklaşım eğrileri ile Tablo1’de verilen FVU değerleri arasında doğrudan ilişki olduğu görülmektedir. Buna göre yine FVU sıralaması $f^{(1)}$, $f^{(2)}$, $f^{(4)}$, $f^{(5)}$ ve $f^{(3)}$ şeklinde olmaktadır. Yine sabit yapıli klasik sinir ağında FVU ölçütü $f^{(1)}$, $f^{(2)}$, $f^{(4)}$, $f^{(5)}$ ve $f^{(3)}$ şeklinde gerçekleşmiştir. FVU oranlarına bakıldığında klasik yapı ile önerilen yapı arasında oldukça büyük farkların olduğu görülmektedir.



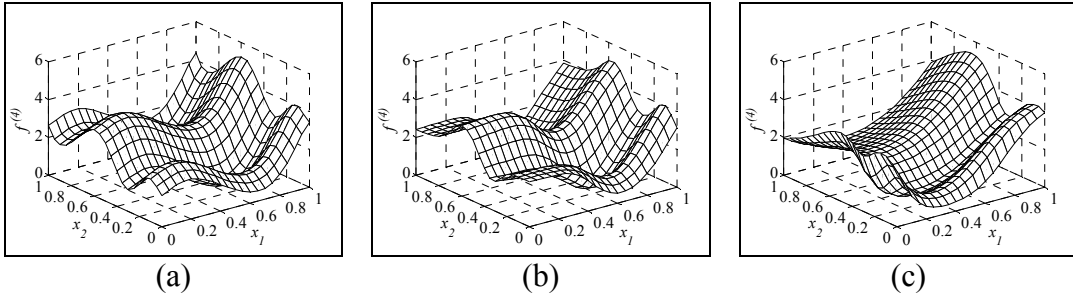
Şekil 6. $f^{(1)}$ fonksiyonu için sonuçlar (a) ideal (b) genetik (c) geleneksel



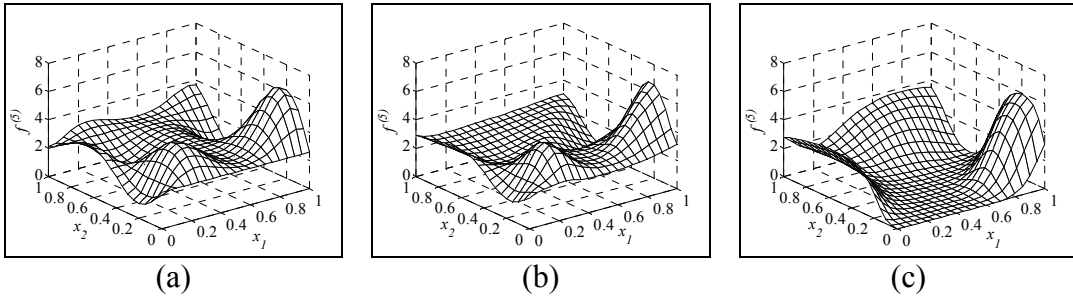
Şekil 7. $f^{(2)}$ fonksiyonu için sonuçlar (a) ideal (b) genetik (c) geleneksel



Şekil 8. $f^{(3)}$ fonksiyonu için sonuçlar (a) ideal (b) genetik (c) geleneksel



Şekil 9. $f^{(4)}$ fonksiyonu için sonuçlar (a) ideal (b) genetik (c) geleneksel



Şekil 10. $f^{(5)}$ fonksiyonu için sonuçlar (a) ideal (b) genetik (c) geleneksel

Örneğin $f^{(1)}$ fonksiyonunu ele alacak olursak FVU oranı önerilen algortmada 0.001293 iken aynı veriler için klasik yapıda 0.025294 değerlerinde olduğu görülmektedir. Yaklaşık olarak 20 kat bir iyileşmenin olduğu görülmektedir. Diğer dört fonksiyon içinde benzer sonuçların olduğu görülmektedir. Şekil 6-10'da beş fonksiyon için ideal, genetik algoritma ve geleneksel yaklaşımla elde edilen grafikler verilmiştir. Bu grafikler incelendiğinde tüm fonksiyonlar için genetik tabanlı olarak öğrenme parametreleri belirlendiğinde ağ yapısı sabit olsa da ağ performansında oldukça önemli bir iyileşme sağladığı açıkça görülmektedir. Tablo 1 ve Şekil

6-10' da verilen grafiklerden de görüleceği gibi ortaya konulan yaklaşımla oldukça yüksek bir doğrulukta fonksiyonlara yaklaşıldığı görülmektedir.

4. Sonuçlar

Bu çalışmada genetik algoritma kullanılarak ileri beslemeli bir sinir ağının eğitiminde kullanılan geri yayımlı öğrenme parametreleri belirlenmiştir. Önerilen genetik tabanlı yaklaşım ve geleneksel yaklaşım ile gerçekleştirilen öğrenme algoritmaları kullanılarak test fonksiyonlarının her biri için ağ eğitimleri gerçekleştirilmiştir. Elde edilen sonuçlar, ortaya konulan genetik tabanlı yaklaşımın geleneksel yaklaşıma göre fonksiyonlara yaklaşımda çok daha yüksek bir performans sağladığını göstermiştir. Sonuç olarak, öğrenme parametrelerinin deneme yanılma yoluyla belirlenmesi yerine genetik tabanlı algoritmalarla belirlenmesi geliştirilecek sinir ağlarının daha etkin olmasını sağlayacaktır.

Kaynaklar

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

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

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.

Kamiyama, N., Iijima, N., Taguchi, A., Mitsui, H., Yoshida, Y., Sone, M. (1992). Tuning of learning rate and momentum on back-propagation. Singapore Iccs/Isita '92, Communications on the Move, 16-20 November 1992, Westin Stamford, Singapore, 528-532.

Kandi1, N., Khorasani, K., Patel, R. V., Sood, V. K. (1993). Optimum learning rate for backpropagation neural Networks. *Canadian Conference on Electrical and Computer Engineering* (pp.465-468), Vol.1.

Kwok, T. Y., 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., 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.

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

Pedrajas, N. G., Boyer, D. O., 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., Gupta, J. N. D. (2000). Comparative evaluation of genetic algorithm and backpropagation for training neural Networks. *Information Sciences*, 129(1-4), 45-59.

Sheell, S. , Varshney, T., Varshney, R. (2007). Accelerated learning in MLP using adaptive learning rate with momentum coefficient, Second International Conference on Industrial and Information Systems. ICIIS 2007, 8 – 11 August 2007, Sri Lanka, (307-310).

Ustun,O. (2009a). Genetik algoritma kullanılarak ileri beslemeli bir sinir ağında etkinlik fonksiyonlarının belirlenmesi. Pamukkale Üniversitesi. Mühendislik Bilimleri Dergisi (Basımda).

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

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

Yao, X. (1999). Evolving Artificial Neural Networks. *Proceedings of the IEEE*, 9(87), 1423–1447.

Yu, C. C., Liu, B. D. (2002). A back propagation algorithm with adaptive learning rate and momentum coefficient. *Proceedings of the 2002 International Joint Conference on Neural Networks (1218–1223)*, Vol. 2.

Yu, X.H., Chen G.A., Cheng S.X. (1993). Acceleration of backpropagation learning using optimised learning rate and momentum, *Electronics*. 29(14), 1288-1290.

Yu, X. H., Chen, G. A. (1997). Efficient backpropagation learning using optimal learning rate and momentum. *Neural Networks*. 10(3), 517-527.

Zaghloul, A., Dong, W. M. (1994). An automated approach for selecting the learning rate and momentum in back-propagation Networks. *IEEE International Conference on Neural Networks*. 27 Jun-2 Jul 1994, 464-469 Vol.1, Orlando. FL. USA.