

Chebyshev Filtre Parametrelerinin Yapay Sinir Ağları Kullanılarak Hesaplanması

Oğuzhan Yavuz¹

M. Can Bayram²

Tülay Yıldırım³

^{1,2,3}Elektronik ve Haberleşme Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul

¹e-posta: ogyavuz@yildiz.edu.tr

²e-posta: can.bayram@gmail.com

³e-posta: tulay@yildiz.edu.tr

Özetçe

Bu çalışmada, Chebyshev filtre katsayılarının yapay sinir ağları yardımıyla hesaplanması önerilmiştir. Yapay sinir ağları (YSA) kullanarak filtre tasarımcısının matematiksel eşitlikleri kullanmadan filtre kutuplarını elde etmesi amaçlanmıştır. Çok Katmanlı Algılayıcı (ÇKA), Radyal Tabanlı Fonksiyon Ağları (RTFA) ve Genel Regresyon Sinir Ağları (GRSA) kullanılıp test edildikten sonra ortaya çıkan sonuçlar gösterilmiştir. Ayrıca bulunan sonuçlara göre karesel hata değerleri hesaplanmıştır. Genel olarak ileri beslemeli yapay sinir ağlarının fonksiyon yaklaşırma özelliğinden faydalanılmıştır.

1. Giriş

YSA'ların çok sayıda farklı çeşitleri vardır. Bu farklılıkların kaynağı mimarisi, öğrenme yöntemi, bağlantı yapısı vb. olabilmektedir. Bağlantı yapılarına göre YSA'lar ikiye ayrılırlar. Bazı ağlar ileri besleme şeklinde yapılandırılırken, bazı ağlar ise geri beslemeli olarak gerçekleştirilir. İleri beslemeli yapay sinir ağlarında, işlem elemanları arasındaki bağlantılar bir döngü oluşturmazlar ve bu ağlar girilen veriye genellikle hızlı bir şekilde karşılık üretirler. Geri beslemeli ağlarda (Yinelenen Ağlar) ise bağlantılar döngü içerirler; hatta her seferinde yeni veriyi işleyebilmektedirler. Bu ağlar, döngü sebebiyle girilen verinin karşılığını yavaş bir şekilde oluştururlar. Bu yüzden, bu tür ağların eğitime süreci daha uzun olmaktadır [1].

İleri Beslemeli Geriye Yayılma mimarisi 1970'li yıllarda geliştirilmiştir. Bu mimarinin geliştirilmesinde birbirlerinden bağımsız olarak birkaç araştırmacının katkıları olmuştur. Asıl katkı ise 1986 yılında Rumelhart ve arkadaşları tarafından yapılmıştır [2]. Ortaya çıkışından sonra, hem etkili hem de çok kullanışlı olmasından dolayı büyük bir popülerlik kazanmıştır. Hala en çok kullanılan ağ türü olarak bilinmektedir. Statik bilgi işleme, kinematik kontrol, örüntü tanıma ve fonksiyon yaklaşırma gibi çok sayıda farklı uygulama alanında kullanılmaktadır. Sonlu uzunlukta bir veri kümesinde fonksiyon yaklaşırma problemi için ileri beslemeli ağların uygulanması çok elverişlidir. Diğer taraftan da, yinelenen yapay sinir ağları sonsuz uzunlukta veri kümelerine uygulanması daha uygundur [3].

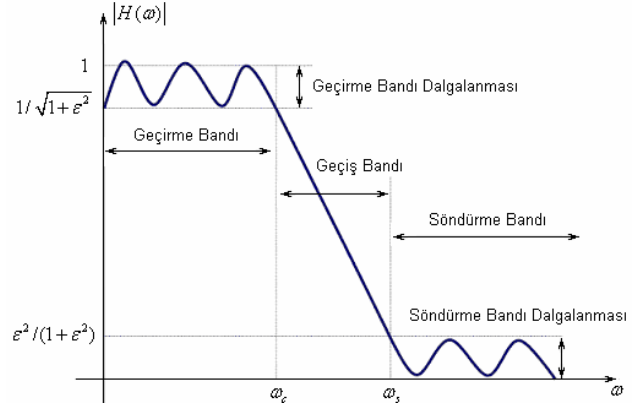
Bu çalışmada, yapay sinir ağları kullanarak farklı derece ve farklı zayıflama değerleri için Chebyshev katsayılarını hesaplayan nöral bir yapı oluşturulması amaçlanmıştır. İleri beslemeli ağ yapılarının fonksiyon yaklaşırma özelliğinden faydalanarak Chebyshev filtre parametrelerinin hesaplanması önerilmiştir. Bu çalışmada sırası ile ÇKA, RTFA ve GRSA ağları kullanılmıştır.

Çalışma toplam olarak 7 kısma ayrılmıştır. Bunlar sırasıyla şu şekildedir: Chebyshev filtre hakkında bilgi, ÇKA, RTFA,

GRSA yapıları hakkında bilgi, yöntem ve simülasyon değerlerinin gösterimi ve sonuçlar.

2. Chebyshev Filtre

Chebyshev filtreleri bir çeşit yüksek-Q filtreleridir. Bu filtreler; söndürme bandında dik iniş istenildiğinde, geçiş bandının düz olmasının gerekli olmadığı durumlarda kullanılır. Bu filtre cevabında, geçiş bandı dalgalanmasına izin verilir. Butterworth cevabına oranla söndürme bandındaki başlangıç inişleri daha keskindir. Bu karşılaştırma Şekil 1'de yer almaktadır. Şekilde eğriler n=3 derecesindeki filtreler içindir. Chebyshev filtresi, geçiş bandında 3 dB'lik dalgalanma yapar. Butterworth filtresinden 10 dB kadar söndürme bandında daha fazla zayıflama yapar [4].



Şekil 1. Chebyshev filtre karakteristiği.

Chebyshev filtresi için zayıflama;

$$\alpha_n = 10 \log [1 + \varepsilon^2 C_n^2(\omega)] \text{ dB} \quad (1)$$

n. dereceden Chebyshev polinomu;

$$C_n(\omega) = \cosh n \cosh^{-1}(\omega) \quad (2)$$

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 C_n^2(\omega)}} \quad (3)$$

Geçirme bandındaki maksimum zayıflama $C_n(\omega) = 1$ olduğunda olur.

$$\alpha_{\max} = 10 \log [1 + \varepsilon^2] \quad (4)$$

$$\varepsilon = \sqrt{10^{0.1\alpha_{\max}} - 1} \quad (5)$$

n = filtrenin derecesi

α_n = dB cinsinden geçiş bandı dalgalanmasıdır.

2.1. Chebyshev Filtre Parametrelerinin Bulunması

Bir Chebyshev filtresinin katsayılarını belirlemek için, tasarlanacak filtrenin derecesi (n), geçiş bandında arzu edilen maksimum zayıflama değeri (α_{\max}) ve geçirme ve durdurma bant frekans değerleri bilinmelidir. Aşağıdaki eşitlikleri kullanarak filtrenin kutup değerleri belirlenebilir. Bu kutup değerlerini tespit ederek tasarım için gerekli olan Chebyshev polinomu elde edilebilir.

$$a = \frac{1}{n} \sinh^{-1} (10^{0.1\alpha_{\max}} - 1)^{-1/2} \quad (6)$$

$$\alpha_k = \pm \sinh a \sin \frac{2k+1}{2n} \pi \quad (7)$$

$$\beta_k = \pm \cosh a \cos \frac{2k+1}{2n} \pi \quad (8)$$

$$p = \sigma + j\beta \quad (9)$$

Kutup bulunduktan sonra filtrenin kesim ya da merkez frekansı (ω_0) ve kalite faktörü (Q) bulunabilir. Filtrenin derecesi büyüdükçe kutup değerlerinin sayısı artacağından polinomun bulunması güçleşir. 2. dereceden filtrenin polinom eşitlikleri aşağıda gösterilmiştir [4].

$$s^2 + 2\alpha s + \alpha^2 + \beta^2 \quad (10)$$

$$\omega_0 = \sqrt{\alpha^2 + \beta^2} \quad (11)$$

$$Q = \sqrt{\alpha^2 + \frac{\beta^2}{2\alpha}} \quad (11)$$

3. Çok Katmanlı Algılayıcılar (ÇKA)

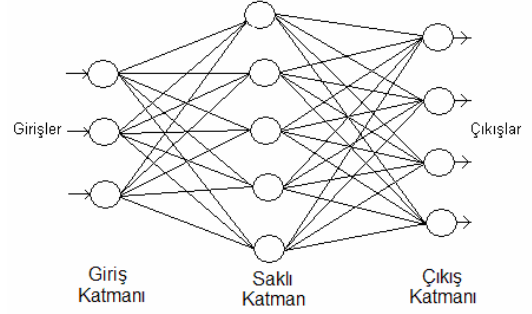
Çok Katmanlı Algılayıcılar; günümüzde birçok mühendislik probleminin çözümünde kullanılabilir ve sınıflandırma amacıyla etkin şekilde sonuçlar üretebilmektedir. ÇKA' da Delta öğrenme kuralı denilen bir öğrenme yöntemi kullanılmaktadır. Bu kuralın amacı; ağız istenen çıktı ile ürettiği çıktı arasındaki hatayı minimum yapmaktır.

ÇKA'lar; 3 katmandan oluşmaktadır. Bunlar; girdi katmanı, gizli katmanlar ve çıktı katmanıdır. Bilgiler girdi katmanından ağa tanıtılır, gizli katmanlardan çıktı katmanına ulaşır ve çıktı katmanından dış dünyaya aktarılır.

ÇKA'larda; eğitici öğrenme yöntemi kullanılmaktadır. Ağa hem örnekler, hem de bu örneklerden oluşturulması gereken çıktılar sunulmaktadır. Ağ, örnekler bakarak problem uzayında bir çözüm üretir; bu genellemeye bağlı olarak gelecek yeni örnekler için de çözüm üretebilmektedir.

ÇKA'nın performansında öğrenme oranı ve momentum parametreleri büyük önem taşımaktadır. Öğrenme oranı küçük seçilirse öğrenme işlemi yavaş olacak, büyük seçilirse değişimler kararsız olacaktır. Aynı zamanda; ÇKA'lar yerel sonuçlara da takılabilmektedir. Bu nedenle, momentum terimi

kullanılmakta ve ele alınan problem için ağız ürettiği çözümler kabul edilebilir düzeye çekilmektedir [4].



Şekil 2. Oluşturulan ÇKA ağının gösterimi, giriş katmanında 3, saklı katmanda 5 ve çıkış katmanında 4 nöron kullanılmıştır.

4. Radyal Tabanlı Fonksiyon Ağları (RTFA)

RTFA'da temel fikir, bir grup radyal taban fonksiyonu istenen fonksiyona yaklaşacak şekilde ağırlıklı olarak toplamaktan ibarettir [5]. RTFA üç katmanlı bir yapıdır. Giriş katmanı giriş vektör uzayı ile çıkış katmanı da örüntü sınıfları ile ilişkilidir. Böylelikle tüm yapı, gizli katmanın yapısı ve gizli katman ile çıkış katmanı arasındaki ağırlıkların belirlenmesine indirgenir. Gizli katmandaki nöronların aktivasyon fonksiyonları bir C_j merkezi ve σ_j bant genişliği ile belirlenir. Aktivasyon fonksiyonu,

$$\varphi_j(X) = \exp\left(-\frac{\|X - C_j\|^2}{2\sigma_j^2}\right) \quad (12)$$

eşitliği ile tanımlanan bir Gauss eğrisidir. Çıkış katmanındaki j . nöronun çıkışı için genel eşitlik ise şu şekildedir:

$$s_j(X) = \sum_{i=1}^K w_{ij} \varphi_i(X) + b_j \quad (13)$$

Burada w_{ij} , j gizli nöron i . ve çıkış nöronu j . arasındaki ağırlık katsayısıdır [5,6].

5. Genel Regresyon Sınır Ağları (GRSA)

GRSA, RTFA'nın merkez ve bant genişliklerinin eğitime verisinin deterministik fonksiyonları olarak belirlendiği özel bir durumdur. Başka bir deyişle, bu tür ağlarda eğitim için iteratif yöntemler kullanılmaz.

GRSA da bir x_i eğitime girişi, ağdaki Gauss çekirdeklerden birinin merkezi olarak atanır. Herhangi bir giriş vektörü x için i . RTFA biriminin çıkışı şu şekilde hesaplanır:

$$\beta_i = \exp\left[-\frac{(x - x_i)^T (x - x_i)}{2\sigma^2}\right] \quad (14)$$

Burada σ kullanıcı tarafından belirlenen yumuşatma parametresidir. Herhangi bir x girişi için ağız çıkışı y , şu şekilde hesaplanır:

$$y = \sum_{i=1}^K \alpha_i y_i \quad (15)$$

Burada α katsayıları şu şekilde hesaplanır:

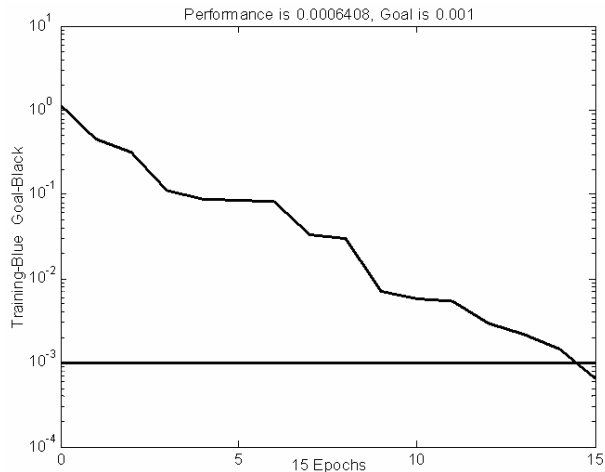
$$\alpha_i = \frac{\beta_i}{\sum_{i=1}^K \beta_i} \quad (16)$$

Eğer giriş vektörü x , herhangi bir x_i eğitime vektörüne yakın ise, x_i 'ye ilişkin α_i en büyük olacak ve istenen çıkış y , x_i 'ye ilişkin y_i çıkışına yaklaşacaktır [6,7].

6. Yöntem ve Simülasyon

Öncelikle belirlenen hedefe uygun bir veri kümesi oluşturulmuştur. Veri kümesi, filtre derecesinin sırasıyla 2 ve 3 değerleri için listelenmiştir. Maksimum zayıflama değeri (α_{\max}) 0,01–10 dB aralığında logaritmik ölçeğe uygun olarak belirlenmiştir. Ayrıca hesaplamalarda kullanmak için gerekli olan a değişkeni Eşitlik 6'dan yararlanarak bulunmuştur. Ağa giriş olarak bu 3 parametre (n , α_{\max} , a) verilmiştir. Bu değerlerin sonucunda hesaplanan kutup değerleri sisteme istenen çıkış olarak verilmiştir. Yapay sinir ağı 3 katmanlı olup, giriş katmanında 3, çıkış katmanında 4 ve gizli katmanda 5 nörondan oluşmaktadır. ÇKA için öğrenme oranı olarak 0,5 ve ortalama karesel hata da 0,001 olarak seçilmiştir. RTFA ve GRSA için yayılma (spread) parametresi 0,05 olarak alınmıştır. Sonuçların elde edilmesinde MATLAB Neural Network Toolbox 7.0 kullanılmıştır.

Oluşturulan veri kümesi toplam 60 adet veriden oluşmaktadır. Bunlardan 54 tanesi eğitim için 6 tanesi de test için kullanılmıştır. Ayrıca eğitim içinde belirlenmeyen farklı ara değerler için de test yapılmıştır. Tablo 2 ve 3 de sadece test kümesi değerleri gösterilmektedir. Ara dereceler için bulunan sonuçlar gösterilmemiştir. Şekil 3'de ÇKA için eğitim hata oranı gösterilmiştir.



Şekil 3. ÇKA kullanılarak eğitilen ağın eğitim hata oranı.

Tablo 1 de her üç ağ yapısı içinde hedef sonuç değerleri verilmiştir. Tablo 2 de ağ yapılarının verdiği çıkış sonuçları ve Tablo 3 de fonksiyon yaklaşırma işleminde kullanılan tüm ağ yapılarına ait karesel hata değerleri yer almaktadır.

Tablo 1. Oluşturulan veri kümesi içerisinde test için kullanılan giriş değerleri ve hedef olarak çıkması gereken sonuçlar.

Giriş Değerleri			Çıkış Değerleri			
α_{\max}	n	a	α_1	β_1	α_2	β_2
0,05	2	1,4626	1,4445	1,6083	0	0
0,5	2	0,8871	0,7128	1,004	0	0
5	2	0,3181	0,2288	0,7432	0	0
0,02	3	1,1278	0,6913	1,4777	1,3826	0
0,5	3	0,5914	0,3132	1,0219	0,6265	0
7	3	0,1602	0,0804	0,8772	0,1609	0

Tablo 2. Aynı veriler ile eğitilmiş olan üç farklı ağ yapısının test çıkışında verdiği sonuçlar.

	α_{\max}'	n'	a'	α_1'	β_1'	α_2'	β_2'
ÇKA Trainlm	0,05	2	1,4626	1,4548	1,6130	-0,0059	0
	0,5	2	0,8871	0,6836	0,9998	-0,0045	0
	5	2	0,3181	0,1892	0,8024	-0,0133	0,0004
	0,02	3	1,1278	0,6452	1,5019	1,3345	0
	0,5	3	0,5914	0,3502	1,0091	0,6479	0
	7	3	0,1602	0,1086	0,8264	0,1764	0,0002
RTFA Newrb	0,05	2	1,4626	1,4271	1,5932	0	0
	0,5	2	0,8871	0,7117	1,0039	0	0
	5	2	0,3181	0,2231	0,7494	0,0086	0
	0,02	3	1,1278	0,6443	1,4132	1,2884	0
	0,5	3	0,5914	0,3142	1,0232	0,6283	0
	7	3	0,1602	0,0891	0,8644	0,1481	0
GRSA	0,05	2	1,4626	1,4511	1,6163	0,0017	0
	0,5	2	0,8871	0,7221	1,0099	-0,0033	0
	5	2	0,3181	0,1971	0,7889	0,0732	0
	0,02	3	1,1278	0,6898	1,4835	1,3956	0
	0,5	3	0,5914	0,3125	1,0222	0,6200	0
	7	3	0,1602	0,1195	0,8144	0,0894	0

Tablo 3. Oluşturulan veri kümesi içerisinde test için kullanılan giriş değerleri için hedeflenen çıkış değerinin, 3 farklı ağ yapısının bulduğu çıkış değerlerine göre karesel hatalarının gösterimi.

	α_{\max}	n	a	$(\alpha_1 - \alpha_1')^2$	$(\beta_1 - \beta_1')^2$	$(\alpha_2 - \alpha_2')^2$	$(\beta_2 - \beta_2')^2$
ÇKA Trainlm	0,05	2	1,4626	0,0001	0	0	0
	0,5	2	0,8871	0,0009	0	0	0
	5	2	0,3181	0,0016	0,0035	0,0002	0
	0,02	3	1,1278	0,0021	0,0006	0,0023	0
	0,5	3	0,5914	0,0014	0,0002	0,0005	0
	7	3	0,1602	0,0008	0,0026	0,0002	0
RTFA Newrb	0,05	2	1,4626	0,0003	0,0002	0	0
	0,5	2	0,8871	0	0	0	0
	5	2	0,3181	0	0	0,0001	0
	0,02	3	1,1278	0,0022	0,0042	0,0089	0
	0,5	3	0,5914	0	0	0	0
	7	3	0,1602	0,0001	0,0002	0,0002	0
GRSA	0,05	2	1,4626	0	0,0001	0	0
	0,5	2	0,8871	0,0001	0	0	0
	5	2	0,3181	0,0010	0,0021	0,0054	0
	0,02	3	1,1278	0	0	0,0002	0
	0,5	3	0,5914	0	0	0	0
	7	3	0,1602	0,0015	0,0039	0,0051	0

7. Sonuçlar

Bu çalışmada, ileri beslemeli ağ yapıların fonksiyon yaklaşırma özelliğinden faydalanarak Chebyshev filtre parametrelerinin hesaplanması önerilmiştir. Bir Chebyshev analog filtre tasarımında derece ve dalgalanma değerine göre transfer fonksiyonunun kutuplarının bulunması çok önemlidir. Kutupların tespiti için yukarıda belirtilen eşitlikler mevcuttur. YSA kullanarak filtre tasarımcısının bu matematiksel eşitlikleri kullanmadan kutupları elde etmesi amaçlanmıştır. Bu çalışmada sırası ile ÇKA, RTFA ve GRSA kullanılmıştır. Sonuçların elde edilmesinde MATLAB Neural Network Toolbox kullanılmıştır. Denenen ağ yapılarının üçü de ileri beslemeli ağ yapısına sahip olduğundan fonksiyon yaklaşırıcı olarak iyi sonuçlar verdiği görülmüştür. Ayrıca ek olarak farklı ileri beslemeli ağ yapılarında ve yinelenen ağlarda denenerek bunların kıyaslaması, RTFA ve GRSA için farklı yayılma parametrelerine de bakılarak incelenmesi düşünülmektedir. Sonuç olarak filtre tasarımcılarının hızlı bir şekilde katsayılarına ulaşabilecekleri bir yapı önerilmiştir.

8. Teşekkür

M. Can BAYRAM'a doktora eğitimi süresince maddi destek verecek olan TÜBİTAK'a bilim insanlarını desteklediği için teşekkür ederiz.

9. Kaynakça

- [1] Yurtoğlu H., Yapay Sinir Ağları Metodolojisi İle Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği , DPT – Uzmanlık Tezleri, ISBN 975 – 19 – 3996 – 9, ŞUBAT 2005.

- [2] D. E. Rumelhart ve J. L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, , Eds. Cambridge, MA: MIT Press, vol. 1, ch. 8, sayfa. 318–362. , 1986.
- [3] Lee T. T. ve Jeng J. T., The Chebyshev-Polynomials-Based Unified Model Neural Networks for Function Approximation, IEEE Transactions On Systems, Man, And Cybernetics Vol. 28, No. 6, sayfa 925-935, Aralık 1998.
- [4] Valkenburg, M.E.V., Analog Filter Design: Cbs College College Publishing Company, New York, 1982.
- [5] Paredes, V., Vidal, E., A Class-Dependent Weighted Dissimilarity Measure for Nearest Neighbor Classification Problems, Pattern Recognition Letters, Vol. 21, sayfa. 1027- 1036, 2000.
- [6] Haykin, S., Neural Networks : A Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.
- [7] Wong, H. S., Wu, M., Joyce, R. A., Guan, L., Kung, S. Y., A Neural Network Approach for Predicting Network Resource Requirements in Video Transmission Systems, Proceedings of IEEE Pacific Rim Conference On Multimedia, 2000.