

Hücresel Yapay Sinir Ağlarında Kümeleme Yöntemi ile Gürültü Temizleme Uygulaması

¹Orhan Tolluoglu, ²Sedef KENT, ³Serdar Kargin, ⁴Osman N.Ucan

¹Hava Harp Okulu
HUTEN Elektronik ABD
Yeşilyurt, İstanbul

²İstanbul Teknik Üniversitesi
Elektrik-Elektronik Müh.
Maslak, İstanbul

³Hava Harp Okulu
Elektronik Müh.
Yeşilyurt, İstanbul

⁴İstanbul Üniversitesi
Elektrik-Elektronik Müh.
Avcılar, İstanbul

¹orhan_tolluoglu@hotmail.com, ²kent@ehb.itu.edu.tr, ³s.kargin@hho.edu.tr, ⁴uosman@istanbul.edu.tr

Özet

Hücresel Yapay Sinir Ağları (HYSA); hücrelerden meydana gelmiş ve her bir hücre en yakınındaki hücre ile etkileşim ve komşuluk ilişkisi içerisinde bulunan dinamik Yapay Sinir Ağ (YSA) modelidir[1]. Yapıyı oluşturan temel elemanlar hücrelerdir. Görüntü işlemedeki görevi, herhangi bir giriş görüntüsünü istenilen amaca uygun olarak çıkış görüntüsüne dönüştürmektir. Çıkışta piksel değerleri -1 ve +1 olacak şekilde sınırlandırılır. Çok seviyeli gürültülü girişe, HYSA uygulayarak iki seviyeli gürültüden arındırılmış çıkış elde edilebilmektedir. Basit uygulamalarda çıkışın iki seviyeli olması çok önemli değildir. Örneğin köşe ve kenar tanıma gibi uygulamalarda iki renk tonu yeterli olabilmektedir[2]. Ancak gürültülü resimlerin temizlenmesi çalışmalarında resimlere ait ayrıntılar bu iki renk tonunun içine gömülmektedir. Bu nedenle HYSA uygulamalarında daha fazla ayrıntı sağlayabilecek ve çok seviyeli görüntü üretebilecek bir yöntem olan kümeleme metodu geliştirilmiştir.

1. Giriş

HYSA, ilk olarak 1988 yılında Leon Chua ve Lin Yang [3] tarafından ortaya atılmıştır. Dinamik yapıya sahip sinir ağ modelinden oluşmaktadır. İşlem basamakları paralel yapıda oluşturulmuştur. Değişmez bir bağlantı ağı kurma kabiliyeti HYSA'yı klasik anlamdaki YSA'ya göre oldukça avantajlı bir konuma getirmektedir. HYSA, özel sınıf dinamik ağ olup matematiksel temeli en sağlam sinir ağ modellerinden biridir [4]. Tüm dinamik sinir ağlarda olduğu gibi buradaki hücrelerde de toplama ve aktivasyon fonksiyonuna ilaveten bir dinamik birim bulunmaktadır.

HYSA ile görüntü işleme, çoklu gri seviyeye sahip giriş görüntüsünün siyah ve beyaz olarak çıkışta üretilmesi prensibine dayanmaktadır. Görüntü, gürültüden arındırılırken arada bulunan gri seviyeler yok olmaktadır. Bu tip sinir ağ yapısının genel çalışma karakteri içerisinde çoklu seviyeli çıkış üretmek mümkün değildir. Çünkü HYSA'da kararlılık, yani her hücrenin bir denge noktasına varabilmesi, *kısmi doğrusal fonksiyondan* geçen hücre değerlerinin iki saçılma bölgesine (siyah veya beyaz) ayrılması ile mümkündür [5]. Ara değerlerin üretilmesi sistemin henüz dengeye varamadığının, halen *kararsız* olduğunun bir göstergesidir.

HYSA'nın çok seviyeli giriş karşısında çıkışta iki seviyeli sonuç üretmesi gürültü temizleme uygulamalarında ayrıntıları da yok etmektedir. Kenar ve köşe tanıma gibi uygulamalarda bu durum sorun olmazken seviyeli çıkış istenen resimlerde yetersiz kalmaktadır. Bu aksaklığı gidermek üzere çıkışta da çok seviye üretebilecek *kümeleme* yöntemi geliştirilmiştir.

2. Kümeleme Yöntemi

Bu yöntemde, giriş farklı kuanta seviyelerine bölünerek HYSA uygulanır. Ancak çıkış, bu sinir ağ yapısının genel mantığı içerisinde çoklu seviyelere ayırlamadığı için *durum* görüntüsü kullanılarak ara değerlerin bulunması sağlanır. Girişin çeşitli kuanta seviyelerine bölünerek sinir ağ girişine verilmesi sırasında bir tür sınıflandırma yapıldığı için bu yönteme *kümeleme* adı verilmiştir.





Kümeleme yönteminde amaç, HYSA girişine gelen piksellerin seviyelerini düşürmek ve bozulmuş hücreleri ait olduğu küme grubu içerisine öteleyerek gürültüyü temizlemektir. Bu yöntemde giriş seviyesi düşürülürken çıkış

seviyelerinde herhangi bir deęişiklik yapılmamaktadır. Çünkü Recurrent Perceptron Learning Algorithm-RPLA, istenen görüntü ile oluşan görüntüyü karşılaştırırken iki seviyeli çıkışa ihtiyaç duymaktadır [2]. Ancak kümelemede, sonuca ulaşıldığı andaki *durum* bilgisi çok seviyelidir. Bu yöntemde nihai çıkışta durum kullanılarak 256 seviyeli sonuca ulaşılabilmektedir.

3. Kümeleme Yönteminde Seviye Ayrımı ve Uygulama

Giriş kuanta seviyelerine bölerek kümeleme şu şekilde yapılmaktadır. 256 seviye dört kuanta seviyesine düşürüldüğünde {0 .. 256} arasındaki piksel değerleri tablo 1’de görüldüğü gibi dört eşit kümeye bölünür ve her küme için bir renk seviyesi belirlenir. Renk seviyelerinde ilk grup siyahı, son grupta beyazı temsil eder. Aradaki gruplar ise gri tonları gösterir. Bu tabloda sadece dört kuanta seviyesine göre aralıklar oluşturulmuştur.

Tablo 1. Piksellerin dört seviyeli küme tablosu

Küme	Renk aralığı	Küme rengi
1	0 .. 63	
2	64 .. 127	
3	128 .. 191	
4	192 .. 255	

Kuantal aralıklarında bir kısıtlama yoktur ve istenilirse sekiz seviyeli yada on altı seviyeli kümeleme de yapılabilir. Kuantal seviyesi arttıkça renk aralıkları azalır. Örneğin sekiz kuantal seviyesi için birinci kümeye ait renk aralıkları {0.. 31}, on altı kuantal seviyesi için de {0..15} olacaktır.

Tablo 2. Kümeleme örneği için verilmiş giriş değerleri

5	30	56	227	112
78	200	58	202	90
0	16	234	89	170
43	225	67	98	1
69	190	200	12	255

Tablo 3. Giriş değerlerinin 16 kuantal seviyesine göre kümeleneşmesi

1	2	4	15	8
5	13	4	13	6
1	2	15	6	11
3	15	5	7	1
5	12	13	1	16

Kümeleme yöntemi ile HYSYA uygulamasında, birinci adımda kuantal seviyesine karar verilir (4,8,16 gibi). Bu seviyelere göre giriş kümelerine ayrılır. Pikseller artık değerleri ile değil grupları ile temsil edilmektedir. Eğer eğitim yapılacaksa ulaşılması istenen resim de kümelerine ayrılır. Kümelerden oluşan girişe HYSYA uygulanır. Bulunan geçici çıkış istenen çıkışla karşılaştırılır. Buradaki karşılaştırmada pikselin değil, pikseli temsil eden grubun ne kadar sonuca yaklaştığı kontrol edilir. Hata kriterine ulaşıldığında eğitim durdurulur.

Tablo 4. İstenen görüntünün piksel değerleri

51	130	156	27	112
178	201	158	20	190
10	161	34	189	70
43	125	7	90	11
6	199	120	112	55

Tablo 5. İstenen görüntü değerlerinin 16 kuantal seviyesine göre kümeleneşmesi

4	9	10	2	8
12	13	10	2	12
1	11	3	12	5
3	8	1	6	1
1	13	8	8	4

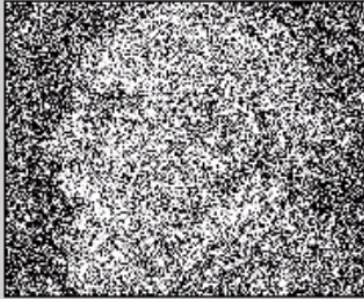
Yukarıda çalışma prensibi anlatılan bu yöntem bir örnekle açıklanacaktır. Tablo 2’de kümeleme örneği için verilmiş *giriş* değerleri görülmektedir. Bu tablodaki değerler yüzlerce pikselden oluşan bir resmin küçük bir parçası olarak kabul edilmiştir. Giriş (gürültü) 5 satır 5 sütundan oluşup 25 piksele sahiptir. Piksel değerleri 0 ile 255 arasına dağılmıştır. Tablo 2’de verilen girişe 16 kuantal seviyesine göre kümeleme yapılarak her piksel kendisine ait gruba dağıtılmıştır. Tablo 3’de piksellerin hangi kümelerde yer aldığı görülmektedir. İstenen görüntünün piksel değerleri tablo 4’de görülmektedir. Bu piksellerin kuantalanmış değerleri de tablo 5’deki gibi olacaktır. Tablo 3 ve tablo 5 karşılaştırıldığında, *giriş* ile *istenenin* başlangıçta aynı kümeyi paylaşan

hücrelerinin $C(1,5)$, $C(2,2)$, $C(3,1)$, $C(4,1)$ ve $C(4,5)$ olduğu görülür. Yani 25 hücreden sadece 5'i istenilen grubun içerisinde. Her konvolüsyon sonucunda yeni oluşan kümelenmiş hücreler istenen hücrelerle karşılaştırılır. Buradaki amaç tüm hücreleri istenen kümenin içine çekebilme. Pratikte bu pek mümkün olmadığından eğitim durdurma kriteri belirlenmelidir. Üç çeşit durdurma kriteri mevcuttur. Bunlar *zaman*, *hata miktarı* veya *hem hata hem de zaman kısıtıdır*.

Yukarıda küçük bir uygulaması yapılan kümeleme yöntemi daha büyük piksel alanına sahip resimde ('dokuz yüz' resminde şekil 1) denenmiştir. Ortalaması 0, sapması 0.8 olan Gaussian gürültüsü ile resim bozularak şekil 2 elde edilmiştir. Kararlılık için altı, eğitim için de beş iterasyon olmak üzere maksimum 30 iterasyon sonucunda şekil 3'deki temizlenmiş görüntü elde edilmiştir. Bu görüntüde siyah ve beyaz dışındaki ara tonların da mevcut olduğu görülmektedir. Bu tonlar çıkış ile değil, çıkışa ulaşıldığındaki durum görüntüsü ile elde edilmiştir. Ayrıca resim incelendiğinde resme ait bir çok ayrıntının ortaya çıkartılabildiği görülmektedir.



Şekil 1. 'Dokuz yüz' resmi



Şekil 2. Gürültü (Gaussian 0,0.8) bindirilmiş 'dokuz yüz' resmi



Şekil 3. Kümeleme yöntemi ile temizlenmiş 'dokuz yüz' resmi

4. Sonuç

Kümeleme yöntemi ile pikseller bir grubu temsil ederek kendi yapılarını terk etmemektedirler. Burada her hücre birbirinden keskin ayrımlarla ayrılır. Kümelemede dikkat çekici nokta birbirine yakın tonlarda olan piksellerin aynı grup içinde temsil edilebilmesi için zaman zaman uzak tonlara ötelenebileceğidir. Ancak bütün bir yapı göz önüne alındığında bazı piksellerde küçük seviyede uzaklaşmalar olurken çok farklı uzaklıklardaki hücrelerin yaklaşması da bu sırada sağlanacaktır. Bu yüzden de giriş piksellerini istenen kümelere yakınlaştırmak bir başarı ölçütü sayılacaktır. Kümeleme işleminde diğer bir konu da eğitimin giriş verisini istenen kümeye yaklaştırmak için yapacağı zorlamadır. Burada eğer büyük bir başarı ile eğitimde kullanılan resmin istenen hale getirilebilmesi farklı bir resim için aynı başarının sağlanmasını engelleyebilir. Bu nedenle eğitimde daha fazla hata kabullenilip şablonların evrensel özellikleri korunmalıdır. Yapılan uygulamalarda eğitim zamanı arttırıldığında bile hata fonksiyonunun belirli bir seviyeden sonra çok zor azaldığı görülmüştür. Aslında kümeleme yönteminde HYSA hücre değişimlerinde radikal değişikliklere izin vermemektedir.

Kaynaklar

- [1]. Uçan, O.N. ve Özmen A., Hücresel Yapay Sinir Ağları ile Bölütlenmiş Görüntülerin ve Hücresel Yapay Sinir Ağ ile Modellenen Kaynak/Kanal Kanal İşaretlerin Rician Ortamında Hata Başarımı, SİU '99, 7. IEEE Sinyal İşleme Uygulamaları Kurultayı s.172- 176, Ankara, 1999.
- [2]. Tolluoğlu, A.O., Bölütlenmiş Hücresel Yapay Sinir Ağları Tasarımı ve Hücresel Ağlarla Lantırın Uçuşlarında Görüntü Kalitesinin Artırılması, Yüksek Lisans Tezi, Hava Harp Okulu, Havacılık ve Uzay Teknolojileri Enstitüsü, İstanbul, 2004.
- [3]. Chua, L.O ve Yang, L., Cellular Neural Networks: Theory. IEEE Transaction on Circuits and Systems, Vol 35, No.10 ; s.1257-1272. 1988.
- [4]. Güzelış, C., Hücresel Yapay Sinir Ağları ile Görüntü İşlenmesi. Tübitak Proje No.EEEAG-103, Rapor, İstanbul, 1993.
- [5]. Chua, L.O ve Yang, L., Cellular Neural Networks: Applications. IEEE Transaction on Circuits and Systems, Vol 35, No.10 ; s.1273-1290. 1988.