

YSA'ların Voronoi Diyagramıyla Yapılandırılması

Configuring ANNs by means of Voronoi Diagrams

Enver Yağcı¹, Şeref Naci Engin¹, İbrahim İ. Esat²

¹Elektrik Mühendisliği Bölümü, Elektrik-Elektronik Fakültesi
Yıldız Teknik Üniversitesi
enveryagci@yahoo.com, nengin@yildiz.edu.tr

²Makine Mühendisliği Bölümü, Mühendislik ve Tasarım Fakültesi
Brunel Üniversitesi
Ibrahim.Esat@brunel.ac.uk

Özet

Yapay Sinir Ağlarının hesaba dayalı yapılandırılması ve ağırlıklarının doğru belirlenmesi, ağlardan yararlanılan çalışmaların hızlanmasında ve etkinliklerinin artmasında önemli bir etkidir. Bu bildiri, Yapay Sinir Ağı parametrelerinin hızlı ve etkin bir biçimde belirlenmesi konusunda yapılan çalışmaların devamı niteliğindedir. Temel çıkış noktası, Voronoi diyagramı ile Yapay Sinir Ağı yapısı arasında kurulan birebir bağlantıdan kaynaklanmaktadır. Üzerinde odaklanılan esas konu ise bu bağlantının analitik olarak tam çözüm bulmasına karşılık bizi çok büyük ve hantal bir ağ yapısına yönlendirmesi sorunudur. Sunduğumuz algoritma sayesinde Voronoi diyagramı Delaunay üçgenleri yardımıyla sadeleştirilmekte ve bu sayede Yapay Sinir Ağındaki bulunan sinir sayısı ve bağlantı sayısında ciddi bir azalma sağlanmaktadır.

Abstract

Configuring Artificial Neural Networks computationally and determining their weights accurately are significant factors in speeding up the works which benefit from networks and increasing their efficiency. This paper is a continuation of works on determining the parameters of Artificial Neural Networks fast and effectively. Its motivation is stemmed from the one-to-one relationship established between the Artificial Neural Networks and Voronoi Diagrams. The main issue focused is that while an exact solution is found by means of this relationship, it leads us to an unnecessarily large and bulky structure. The algorithm presented here simplifies Voronoi diagram of data set by means of Delaunay triangles, hence reduces the numbers of neurons and connections in the network dramatically.

1. Giriş

Yapay Sinir Ağlarının (YSA) 50 yılı aşkın bir tarihçesi olmasına karşılık, artan hesaplama yöntem ve yetenekleri sayesinde son 20 yıldır çeşitli alanlarda çalışan araştırmacılar tarafından sistem modellemesi, veri sınıflaması ve süreç

denetimi gibi çeşitli uygulama alanları bulmaktadır, [1,2,3]. Öyle ki sadece YSA çalışmalarını yayınlayan dergiler çıkarılmaktadır, örneğin IEEE ve Springer Verlag gibi kuruluşların sadece bu konuda dergileri bulunmaktadır: IEEE Transactions on Neural Networks, Neural Computing and Applications. Yapay Sinir Ağları özellikle matematiksel olarak modellenmesi güç olan problemlerde, sistemle ilgili önceden toplanmış veri kümesi yardımıyla çözüm elde etmemizi sağlayabilmektedir. Ancak, Yapay Sinir Ağlarıyla ilgili çözüm bekleyen önemli sorunlardan bir tanesi kullanılacak Yapay Sinir Ağı'nın mimarisinin belirlenmesidir. Pek çok uygulamada YSA'yı yapılandırmada denemeyanılma ve geçmiş tecrübelerle dayanan sezgisel (heuristic) yöntemlerin kullanılması tercih edilmiştir [4,5]. Bu çalışmada özellikle ulaşmaya çalıştığımız nokta optimum bir ağ tasarımı elde etmek için yöntem geliştirmek olacaktır. Bu konuda daha önce önerilen bir teknik olan Voronoi diyagramı çalışmamızın çıkış noktasını oluşturmuştur. Voronoi diyagramları veri kümesinin oluşturduğu karar uzayını dışbükey (convex) yapılarla böldüklerinden tek gizli katmana sahip bir ağ tasarlamak mümkün olacaktır. Asıl elde edilen yarar ise ağdaki sinir sayısının yanı sıra aralarındaki ağırlıkların da bu yolla belirlenebilecek olmasıdır. Bu sayede YSA tasarımında sezgisel yöntemlere gerek kalmadan, matematiksel dayanağı olan bir modelin geliştirilmesi mümkün olacaktır. Böylece elde edilen yapılandırılmış ve ağırlıkları belirlenmiş bir YSA modeliyle ele alınacak veri sınıflama problemleri daha etkin bir biçimde çözümlenecektir.

Bu bildiri, giriş bölümünü takiben Voronoi diyagramları ve veri sadeleştirmesinde yararlanılan Delaunay üçgenlerinin anlatıldığı Temeller bölümüyle devam edecek, sonrasında geliştirilen algoritma açıklanacak, bu algoritmayla yapılan deneylerden elde edilen sonuçlar sunulacak ve değerlendirme kısmıyla sona erdirilecektir.

2. Temeller

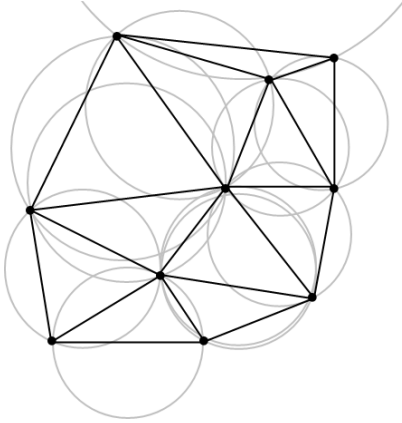
Bu kısımda YSA'nın yapılandırılmasında yararlanılan Voronoi diyagramları ve diyagramların sadeleştirilmesinde kullanılan Delaunay üçgenleri tanıtılacaktır.

2.1. Voronoi Diyagramları

Voronoi diyagramları esas olarak karar uzayını oluşturan verilerin “mozaiklere” bölünmesi olarak sunulan ve adını aldığı Georgy Voronoi tarafından 1903 yılında önerilmiş bir veri parçalama yöntemidir. Euclid uzayında bulunan ayrık S noktalar kümesi için, herhangi bir s noktasına en yakın noktalar kümesinden söz edilebilir. Genel olarak, oluşan her bir kümedeki noktaların tamamı S kümesindeki bir s noktasına en yakın bulunan noktalar kümesi şeklinde tanımlanabilir. İşte bu noktalar kümelerinin aralarındaki sınırı belirleyen yapı Voronoi diyagramı ile elde edilir. Voronoi diyagramı Euclid uzayı dışında Mahalanobis mesafesi gibi farklı bir ölçüm kullanılarak da hesaplanabilir. Ancak bu durumda tek bir Voronoi diyagramının hesaplanabileceğinin garantisiz bulunmamaktadır. Voronoi diyagramından bir çok konuda yararlanmak mümkündür [6,7]. Örneğin, çeşitli nesnelerin bulunduğu bir ortamda en büyük boş daire özelliğini kullanarak, en uzakta nereye konumlandırma yapılabileceğinin hesaplanmasında Voronoi diyagramlarından yararlanılmıştır. Polimer araştırmalarında boş hacim hesaplamada, robot uygulamalarında gidilecek rotanın belirlenmesinde ve diğer benzeri alanlarda aktif olarak kullanılmaktadır [8,9,10]. Voronoi diyagramının hesaplanmasında birçok yöntem önerilmiştir. Bunlara örnek olarak Fortune Algoritması ve Bowyer-Watson algoritması gösterilebilir [11].

2.2. Delaunay Üçgenleri

1934 yılında Boris Delaunay tarafından ortaya atılmıştır. Delaunay üçgenleri Voronoi diyagramının tamamlayıcısıdır. Delaunay üçgenlerinin elde edilmesinde, üç noktadan geçecek ve içerisinde başka herhangi bir nokta bulunmayacak biçimde daireler seçilir; bu dairenin üzerindeki noktalara ait üçgenlerin her biri bir Delaunay üçgenidir, Şekil 1.



Şekil 1: Delaunay üçgenleri.

Delaunay üçgenleri, L^2 normu dışında Chebyshev mesafesi gibi farklı bir ölçüm kullanılarak da hesaplanabilir. Ancak bu durumda üçgenlerin yegâne olduğunun garantisiz bulunmamaktadır. Literatürde “böl ve işgal et” yöntemiyle “tarama çizgisi” (sweepine) tekniği gibi çeşitli algoritmalar Delaunay üçgenlerinin hesaplanmasında kullanılmıştır, [12]. Sonlu elemanlar ve üçboyutlu “rendering” gibi alanlarda Delaunay üçgenleri sıkça kullanılmaktadır [13].

2.3. Voronoi Diyagramı – Yapay Sinir Ağı ilişkisi

Bilindiği gibi tek bir *perceptron* doğrusal olarak ayrılabilen iki veri grubuna ait noktalar kümesini sınıflandırabilir. İki veri grubuna ait noktaların karmaşık dağıldığı ve tek bir doğruyla ayıramadığı durumlarda ise birden fazla *perceptron* bulunan YSA yapısına geçilir. Doğrusal olmayan problemlerde ise birden fazla katman bulunan ve çıkış fonksiyonları doğrusal olmayan sinirlerin oluşturduğu YSA yapıları tercih edilir. N tane ayrık hiper-düzlemden oluşan sistemi N tane sınırdan oluşan katman temsil edebilir. Yani birden fazla hiper-düzlem kullanılarak bölgeler ifade edilebilir. Gizli katman denilen ikinci katmanda ise bu hiper-düzlemlerin oluşturduğu bölge sayısı kadar sinir bulunan bir katman oluşturulur. Son katmanda sınıf sayısı kadar sinir bulundurulur, hangi sinir ateşlenirse girişteki noktanın o sınıfa ait olduğu anlaşılır.

Bizim bu çalışmada örnek göstereceğimiz denemeler iki boyutlu noktalar kümesi olduğundan hiper-düzlem yerine doğrulardan yararlanılacaktır.

3. Algoritma

Geliştirdiğimiz algoritma temel olarak katman sayısı, katmanlardaki sinir sayısı ve sinirler arası ağırlık değerlerinin belirlenmesi de dâhil olmak üzere bir Yapay Sinir Ağı'nın tümüyle tasarlanmasını amaçlamaktadır. Aynı amaçla yapılmış çalışmalardan biri olan [14]'de Voronoi diyagramlarından yararlanarak nasıl bir ağ tasarımı yapılacağı konusunda temeller atılmıştır. Ancak atfedilen çalışmada hesaplanan ağ, veri kümesinin büyüklüğüne bağlı olarak büyüklüğü inanılmaz boyutlara varabilmektedir. Bu nedenle, geniş veri kümelerine ait ağ tasarımı önerilen yöntem yetersiz kalmıştır. Bir başka çalışma [15] ise bu konuyu bir adım öteye götürerek daha düşük sinir sayısına ulaşmayı hedeflemiştir. [15]'de sunulan çalışmayı geliştirmek üzere, bu bildiride sinir sayısı ve bağlantı sayısını düşürmeye yönelik bir çalışma yapılmıştır.

Voronoi diyagramının sadeleştirilebilmesi için esas olarak aralarında hiçbir sınır bağlantısı olmayan noktaların çıkartılması mantığı izlenmiştir. Eğer bir birine komşu hücreler aynı sınıfa ait iseler, bu iki sınıfı bir arada tutmamak için hiç bir sebep bulunmamaktadır. Birbirine komşu hücrelerin aynı sınıftan olup olmadıklarını anlamak amacıyla Voronoi diyagramının tamamlayıcısı olan Delaunay üçgenlerinden yararlanılmıştır. Delaunay üçgenlerini kullanırken amacımız aslında bir bakıma sınır noktalarını tespit etmektir. Bu sayede ilgisiz noktalardan kurtulmak mümkün olabilecektir. Nokta sayısının azalması ise doğrudan YSA'nın büyüklüğünün ve karmaşıklığının azalmasını sağlar.

$S \in N^{m \times 2}$, $A, B \in N^{m \times 2}$ sınıfına ait ögeler içeren bir noktalar kümesi olsun. $d=D(S)$, S noktalar kümesinin Delaunay işlemini ve $v=V(S)$ yine S kümesinin Voronoi diyagramını temsil etsin. Burada $d, v \in N^{m \times 4}$ olmak üzere, aşağıdaki işlem bize sınırdaki bulunan noktalarda Delaunay üçgenlerine ait çizgiler kümesini verecektir.

- Eğer $[v(i,1), v(i,2)] \in A$ ve $[v(i,3), v(i,4)] \in B$ ise bu iki noktanın sınırdaki olduğuna karar verilir.

- Yine aynı şekilde, eğer $[v(i,1), v(i,2)] \in B$ ve $[v(i,3), v(i,4)] \in A$ ise bu iki nokta da sınırdaki noktalar kümesine dahil edilir.
- Diğer durumlarda noktaların sınırdaki olmadığı kabul edilmektedir.

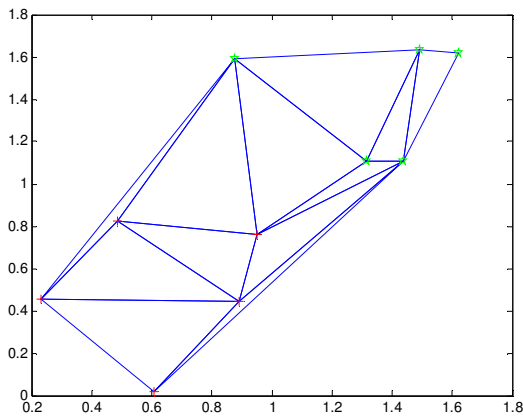
Böylece Delaunay'ın zaten çemberlere dayalı olan en yakın noktalar üzerinde kurduğu üçgenler kullanılarak sınırdaki noktalara karar verilebilmektedir. Diğer bir deyişle, birbirine en yakın noktaların aynı sınıftan olup olmadıklarının tespitine çalışılmaktadır. Zaten çok yakın olan noktalar aynı gruptan iseler, bunları ayrıca Voronoi ile bölütleyip Yapay Sınır Ağımıza daha fazla sayıda sınır ve bağlantı katmak hesap yükümüzün artmasına neden olacaktır.

Delaunay üçgenleri kullanarak noktaların belirlenmesi işleminin tamamlanmasının ardından sınırdaki bulunmayan noktalar atılmaktadır. Zaten sınırdaki bulunmadıklarından sınıflandırmaya direkt bir etkileri bulunmamaktadır. Kalan noktaların Voronoi diyagramını hesaplanarak yeni bir sınıflandırma oluşturulur. Bu sınıflandırmadan yararlanılarak Yapay Sınır Ağının tasarımı için Temeller kısmında anlatılan Voronoi diyagramı ile YSA arasında geçiş için gerekli hesaplamalar yapılarak katsayılar ve sınır sayıları hesaplanır.

4. Sonuçlar

Bu kısımda, üçüncü kısımda anlattığımız adımları takip ederek yaptığımız denemelerde elde ettiğimiz iyileştirmelere ait sonuçlar verilecektir.

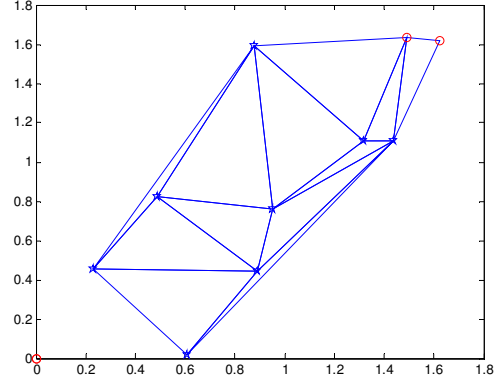
10 noktadan oluşan nispeten küçük bir veri kümesi yaratıp bunun başlangıç Voronoi diyagramını çizdirirsek, Şekil 2'de gösterilen diyagram elde edilecektir.



Şekil 2: 10 nokta üzerinden Voronoi diyagramı.

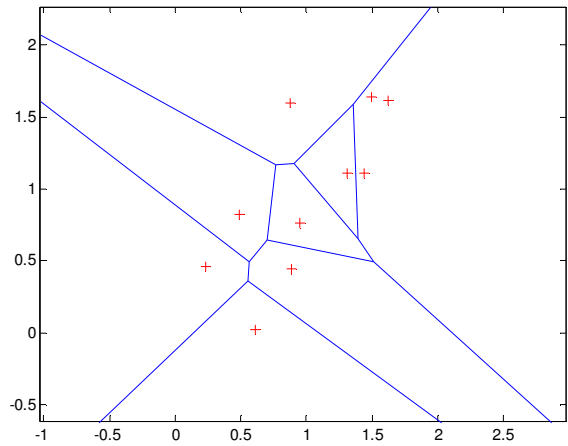
Şekil 2'de görüldüğü gibi kırmızı ile işaretlenmiş A sınıfına ait noktalar kümesi (sol alttaki 5 nokta, "+" işaretli) ve yeşil ile işaretlenmiş B sınıfına ait noktalar kümesi (sağ üstteki 5 nokta, "*" işaretli) bulunmaktadır. İlk anda hiçbir sadeleştirme yapılmadığında karşımıza çıkan Voronoi diyagramı 10 noktanın hepsini kapsayacak şekilde olacaktır.

Bu hesaplama göre tasarımda gizli katmandaki sınır sayısı 10 olacaktır. Ayrıca giriş katmanındaki toplam sınır sayısı ise doğru sayısıyla ilintili olarak 21 olacaktır. Buna göre bu küçük tasarımda 31 adet sınırdan oluşmuş bir yapı önerilmektedir. Ancak Delaunay ile oluşturduğumuz algoritmayı uyguladığımızda Şekil 3'de gösterilen durum ortaya çıkmakta ve kırmızı ile gösterilen noktalar (sağ üstteki 2 nokta, "o" işaretli) dışarıya atılmaktadır.



Şekil 3: 10 nokta içinde sınır olmayan noktalar.

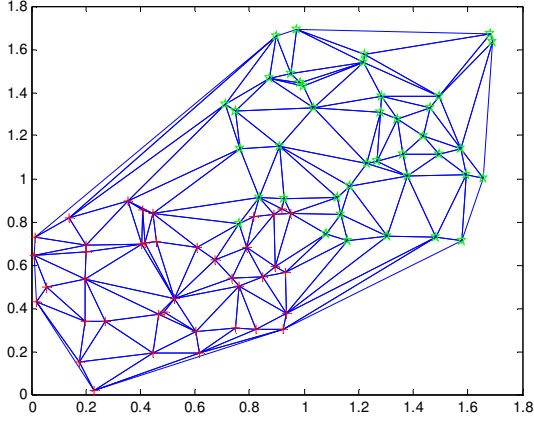
Dikkat edilirse, sınırdaki bulunmayan noktaları seçmeye dayalı yeni Voronoi diyagramımızda kullanacağımız nokta sayısı 10'dan 8'e inmiştir. Ayrıca ilk katmandaki sınır sayısı da 17'ye düşmüştür. 10 nokta gibi düşük sayıda veri barındıran bir uygulama için Yapay Sınır Ağında toplam 6 sınırdan ve ilgili bağlantılarından tasarruf edilmiştir. Sonuç Voronoi diyagramımız ise Şekil 4'de verildiği gibi olmaktadır. Görüldüğü gibi artık Voronoi diyagramının sınıflandırdığı her bir hücreye birden fazla nokta düşebilmektedir.



Şekil 4: 10 nokta için yeni Voronoi diyagramı.

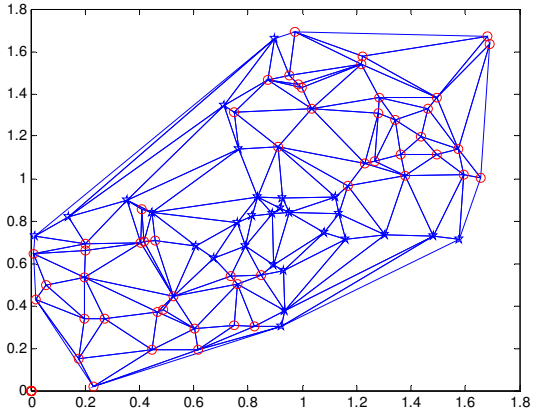
İzlenen yöntemin, veri kümesini oluşturan nokta sayısının az olduğu durumlar için yararı fazla anlaşılacakla birlikte daha fazla nokta sayısına sahip örneklerde tasarruf edilen sınır sayıları daha açık bir şekilde ortaya çıkmaktadır. Yine yeşil ve kırmızı renkleri kullanarak (rahat görünmesi için sırasıyla "+" ve "*" işaretleri ile temsil edilmiştir) ayırdığımız iki sınıftan oluşan noktalar kullanılarak 80 noktalı bir veri kümesini

sınıflandırmak istersek, başlangıçta noktaların dağılımı ve bu noktalara ait Voronoi diyagramı Şekil 5'deki gibi olacaktır.



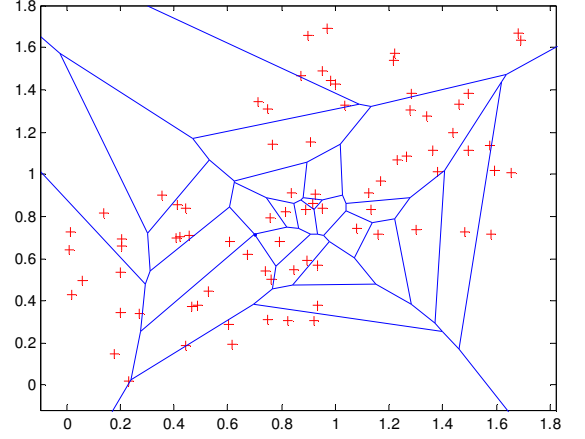
Şekil 5: 10 nokta üzerinden Voronoi diyagramı.

İlk çizilen Voronoi diyagramını takiben yine Delaunay üçgenler kullanılarak sınırdaki noktaların belirlenmesi için algoritmamızı yürüttüğümüzde sınırda olmayan noktalar tespit edilmiş olur. Sınır teşkil etmeyen noktalar ve diğer bütün noktalara ait Delaunay üçgenleri Şekil 6'da gösterilmiştir.



Şekil 6: 60 nokta içinde sınır olmayan noktalar.

Görüldüğü gibi nokta sayısı çok fazla arttığında eksiltilecek nokta sayısı da oldukça yükselmiştir. Objektif olarak bu işlemin örneğimizde ne kadar bir fayda sağladığını anlamak amacıyla Şekil 7'de gösterilen Voronoi diyagramına geçmemiz gerekecektir.



Şekil 7: 60 nokta içinde sınır olmayan noktalar.

Elde ettiğimiz yeni Voronoi diyagramımızda ayrılan her bir hücrenin içinde çok sayıda '+' ile gösterilen nokta göze çarpmaktadır. İşte her bir hücre içerisinde bulunan nokta sayısının çokluğu bize ilk bakışta Yapay Sınır Ağımızı ne kadar küçülttüğümüz ile ilgili bir fikir vermektedir. Nokta sayısının çok olması o bölgeye karşılık gelen sınırlarla daha kompakt bir şekilde bir temsil yapabileceğimizi göstermektedir.

Eğer Voronoi diyagramını hiç nokta sayısı azaltma işlemimizi yapmadan doğrudan hesaplayarak bir Yapay Sınır Ağı oluşturmaya çalışsaydık ortaya çıkacak ağın giriş ve gizli katmanındaki sınır sayıları sırasıyla 226 ve 80 olacaktı. Şekil 7'de gösterilen nokta sayısı azaltılmış şekliyle oluşturulan Voronoi diyagramı kullandığımızda ise ilk katmandaki sınır sayısı 77'ye inmiştir. İkinci katmandaki sınır sayısı ise 28'e düşmüştür. Böylece başlangıçta çıkış katmanındaki iki sinirle birlikte 308 adet sinirden oluşan yapı, 107 adet sinirden oluşan bir yapıya dönüşmüştür.

Sınırlar arası bağlantılar ve bunlara ait kat sayıları da göz önüne alındığında hesap yükünde elde edilen rahatlık daha iyi anlaşılabilir hale gelmektedir. Şu haliyle bu yapıdaki bir veri kümesi için %64'lük bir iyileştirme sağlandığı hesaplanmıştır.

5. Değerlendirme

Bu çalışmada öncelikli amacımız Yapay Sınır Ağının parametrelerinin ve yapısının bir seferde başarılı bir şekilde belirlenebilmesi olmuştur. Bu konuda zaten daha önce yapılmış olan bir çalışma bulunmaktaydı [15]. Ancak yukarıda değindiğimiz gibi elde edilen Yapay Sınır Ağı çok büyük olmaktadır. Bu çalışmada geliştirdiğimiz algoritma sayesinde Yapay Sınır Ağının boyutları önemsenmeyecek miktarda küçülmüştür. Ancak bu konuda yapılabilecek iyileştirmeler bundan ibaret değildir. Hesaplamalarımızla ortaya koyduğumuz küçültmenin ardından Voronoi diyagramına ait doğruların eğimlerinden yararlanılarak ikinci bir küçültmeye gidilebileceği gözükmektedir. Bunun dışında yine bazı bölgelerde daha düz hücre yapısı elde edilerek sınır sayısını azaltmak mümkündür.

Bu çalışmada ortaya çıkan yapay sınır ağı diğer ağlarda yaptığımız gibi belirli bir ilk koşuldan çıkıp eğitilerek

yakınsama yapmak gibi kısmen sezgisel bir metot kullanılması yerine deterministik olarak ağıın çözümünü sunmaktadır. Ancak her veri kümesinde işler bu derece basit olmamaktadır. Çok boyutlu karmaşık noktalar kümesinden oluşan sistemlerdeki sonuçları bizim için çok daha yararlı ve yol gösterici olacaktır. Şu an önerdiğimiz sistemle veriler ne kadar iç içe geçmiş olursa olsun kesin bir çözüm vereceği aşikârdır.

Ayrıca bu algoritma bir adım öteye götürülerek tam bir çözüm bulunması yerine belirli bulanıklıklara sahip ama çok daha küçük bir ağa gidilerek eğitim için bir başlangıç referansı şeklinde kullanılabilir. Böylece klasik olarak yapılan ve farklı sınır sayılarındaki ağıın ne kadar başarılı olduğunun ölçülmesi yerine, direk olarak optimum bir tasarıma veya optimuma çok yakın bir tasarıma ulaşılması mümkündür.

Üzerinde durulmaya değer bir başka nokta ise belirli sınırlar içerisinde Voronoi diyagramını değiştirerek hücrelerin dışbükey olmasını bozmak suretiyle katman sayısının bir fazla yapılabilmesidir. Böylece katman sayısından dolayı bir kayıbmız olurken sınır sayısından dolayı oluşacak kazancımızın çok daha büyük olması beklenmektedir.

6. Kaynaklar

- [1] Pappa, N.; Shanmugam, J., "Neural network based predictor for control of cascaded thermal process", *Proceedings of International Conference on Intelligent Sensing and Information Processing*, vol., no., pp. 289-294, 2004.
- [2] Jiang Chang; Yan Peng, "Movable propeller turbine neural network model and nonlinear simulation", *International Conference on Machine Learning and Cybernetics*, vol.2, no., pp. 1220-1225 Vol. 2, 18-21 Aug. 2005.
- [3] Yi-Shiou Chen; Tah-Hsiung Chu, "A neural network classification tree", *IEEE International Conference on Neural Networks*, vol.1, no., pp.409-413 vol.1, Nov/Dec 1995.
- [4] Chen Senfa; Tang Changbao, "Neural network structure optimization and its application for passenger flow predicting of comprehensive transportation between cities", *IEEE International Conference on Grey Systems and Intelligent Services, 2007. GSIS*, vol., no., pp.1087-1091, 18-20 Nov. 2007.
- [5] Ai-Jun Li; Yun-Hui Liu; Si-Wei Luo, "On the solution of the XOR problem using the decision tree-based neural network", *International Conference on Machine Learning and Cybernetics*, vol.2, no., pp. 1048-1052 Vol.2, 2-5 Nov. 2003
- [6] Maynard, D.N.; Hibbeler, J.D., "Measurement and reduction of critical area using Voronoi diagrams", *Advanced Semiconductor Manufacturing Conference and Workshop, 2005 IEEE/SEMI*, vol., no., pp.243-249, 11-12 April 2005.
- [7] Fan, G.; Zhang, J., "A novel geometric diagram and its applications in wireless networks", *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.1, no., pp.-681, 7-11 March 2004.
- [8] Zhao, G.; Joshi, R.P.; Lakdawala, V.K., "Percolative breakdown model for ceramics based on a random grain-boundary network", *Conference on Electrical Insulation and Dielectric Phenomena. CEIDP '05. 2005 Annual Report*, vol., no., pp. 544-547, 16-19 Oct. 2005.
- [9] Peng Jianliang; Zhu Fan; Sun Xiuxia; Sun Biao, "On Route-Planning of UAV Based on Discrete PSO and Voronoi Diagram", *Control Conference, 2007. CCC 2007. Chinese*, vol., no., pp.804-807, July 26 2007-June 31 2007.
- [10] Yong-Ju Kwon; Jun-Gyu Kim; Jeongyeon Seo; Dong-Ho Lee; Deok-Soo Kim, "A Tabu Search Algorithm using the Voronoi Diagram for the Capacitated Vehicle Routing Problem", *International Conference on Computational Science and its Applications. ICCSA 2007.*, vol., no., pp.480-488, 26-29 Aug. 2007.
- [11] www.diku.dk/hjemmesider/studerende/duff/Fortune
- [12] Guibas, L. and Stolfi, J., "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams", *ACM Transactions on Graphics*, Vol.4, No.2, April 1985, pages 74-123.
- [13] Dittrich, R.; Scholz, W.; Suess, D.; Forster, H.; Tsiantos, V.; Schrefl, T.; Fidler, J., "Finite element simulation of discrete media with granular structure", *IEEE International Magnetics Conference, 2002. INTERMAG Europe 2002. Digest of Technical Papers*. vol., no., pp. FP12-, 2002
- [14] Bose, N.K.; Garga, A.K., "Neural network design using Voronoi diagrams", *IEEE Transactions on Neural Networks*, , vol.4, no.5, pp.778-787, Sep 1993
- [15] Esat, I.I.; "Neural Network Design Based on Decomposition of Decision Space", *ICONIP'99 6th International Conference on Neural Information Processing*. Vol. 1, Page(s):366 – 370, 1999.