

EKG Vurularının Sürekli Zaman Karınca Koloni Optimizasyonu Yardımıyla Sınıflandırılması

ECG Beat Classification Using Ant Colony Optimization for Continuous Domains

Berat Doğan¹, Mehmet Korürek¹

¹Elektronik ve Haberleşme Mühendisliği Bölümü
İstanbul Teknik Üniversitesi
bdogan@itu.edu.tr, korurek@itu.edu.tr

Özet

Bu çalışmada altı farklı EKG vurusunu sınıflamak için oldukça yeni bir optimizasyon algoritması olan sürekli zaman karınca koloni optimizasyonu (KKO_R) kullanılmıştır. EKG vuruları MIT-BIH aritmi veri tabanından derlenmiştir. Bu vurular sırasıyla, Normal (N), Premature Ventriküler Kontraksiyon (PVC), Ventriküler ve Normal Vuru Füzyonu (F), Atrial Prematüre Vuru (A), Sağ Yan Dal Bloğu (R) ve Yapay ve Normal Vuru Füzyonu'ndan (f) oluşmaktadır. Eğitim aşamasında her vuru tipinden 50'şer adet olmak üzere toplam 300 öznitelik vektöründen oluşan eğitim kümesi radyal tabanlı fonksiyon yapay sinir ağı (RTFA) girişine uygulanmıştır. Ağın parametreleri sürekli zaman karınca koloni optimizasyonu ile optimize edilmiştir. Test kümesi üzerinde yapılan denemeler sonucunda, önerilen yöntemin klasik RTFA eğitim yöntemleri olan K-Ortalamlar ve Ortogonal En Küçük Kare algoritmalarına kıyasla, sınıflama başarısından ödün vermeden çok daha küçük bir ağ yapısı ile EKG vurularını sınıfladığı gözlemlenmiştir.

Abstract

In this study, a naturally inspired optimization algorithm, Ant Colony Optimization for Continuous Domains (ACO_R), is used to classify six types of ECG beats including, Normal Beat (N), Premature Ventricular Contraction (PVC), Fusion of Ventricular and Normal Beat (F), Atrial Premature Beat (A), Right Bundle Branch Block Beat (R) and Fusion of Paced and Normal Beat (f). A radial basis function neural network is evolved for classification with the training set obtained from MIT-BIH arrhythmia database by using Ant Colony Optimization for Continuous Domains. Training set includes 50 feature vectors for each class. The results are then compared with the classical radial basis function training methods such as Orthogonal Least Square Algorithm and the K-Means algorithm. It is observed that the proposed method can classify ECG beats with a smaller size of network without making any concession on classification performance when compared to the classical methods.

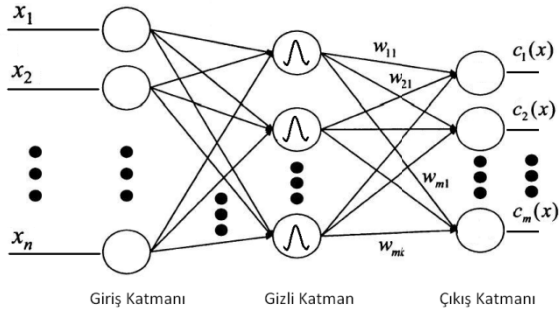
1. Giriş

Kalbin elektriksel aktivitesi hakkında bilgi veren EKG, kalp hastalıklarının erken teşhis ve tedavisinde hayati bir önem arz etmektedir. Klinik ortamda bu kadar öneme sahip olan EKG kayıtlarının analizi çoğu zaman herhangi bir kardiyolog için oldukça zaman alıcı olabilmektedir. Bu zaman kaybı beraberinde EKG kayıtlarını farklı algoritmalar yardımıyla analiz etme ihtiyacını doğurmuştur. Literatürde EKG vurularını analiz etmede kullanılan birçok yöntem mevcuttur [1]. Bu yöntemlerin temel prensibi örüntü tanımaya dayanmaktadır [2]. Örüntü tanımada yapay sinir ağlarının kullanımı oldukça yaygın bir yöntemdir.

Radyal tabanlı fonksiyon yapay sinir ağı (RTFA) ilk kez Broomhead ve Lowe (1988) tarafından ortaya atılmıştır [3]. RTFA, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşan doğrusal olmayan hibrid bir ağ modelidir. RTFA'nın gizli katmanında bulunan nöronlar, radyal tabanlı fonksiyon (RTF) olarak adlandırılan aktivasyon fonksiyonlarına sahiptirler.

RTFA'ların eğitimini başlıca iki ana kısımda ele almak mümkündür. Bunlar; nöron merkezleri ve her bir nöronun yayılım parametresinin bulunması ve gizli katman ile çıkış katmanı arasındaki ağırlıkların bulunması olarak sıralanabilir. Nöron merkezlerinin tespiti aslında çok bilinen öbek merkezini belirleme problemiyle benzeşmektedir. Bu şüphesiz ağı nöronlarının öznitelik uzayında, öznitelik vektörleri arasında bir yere konuşlanmış olduğu kabulünden hareketle ortaya atılan bir önermedir. Ancak bu önerme her zaman doğru olmayabilir. Bir başka ifadeyle, nöron merkezleri öznitelik vektörleri içerisinde konuşlanmış olmayabilir.

Literatürde kullanılan öbikleştirme algoritmaları yardımıyla nöron merkezlerini bulmak mümkündür. Bu çalışmada da çok bilinen bir öbikleştirme algoritması olan K-Ortalamlar algoritması kullanılarak öbek merkezleri bulunmuştur. RTFA'nın eğitimi için kullanılan bir diğer yöntem ise Ortogonal En Küçük Kare (OLS) yöntemidir.



Şekil 1: RTFA ağının genel yapısı.

Yapay sinir ağlarının eğitiminde kullanılan oldukça etkili bir yöntem optimizasyon algoritmaları yardımıyla ağların eğitilmesidir. Optimizasyon algoritmaları yardımıyla kimi zaman herhangi bir ağı sadece ağırlıkları optimize edilirken, bazen de bütün ağ yapısı veya ağ eğitim algoritmaları optimize edilebilmektedir [4].

Son yıllarda sürü zekasına dayalı optimizasyon algoritmaları birçok mühendislik probleminin çözümünde olduğu gibi yapay sinir ağlarının eğitiminde de yaygın olarak kullanılmaktadır. Bu algoritmaların en çok bilinenleri, parçacık sürü optimizasyonu (PSO), karınca koloni optimizasyonu (KKO) ve arı koloni optimizasyonu (AKO) 'dur [5] – [7].

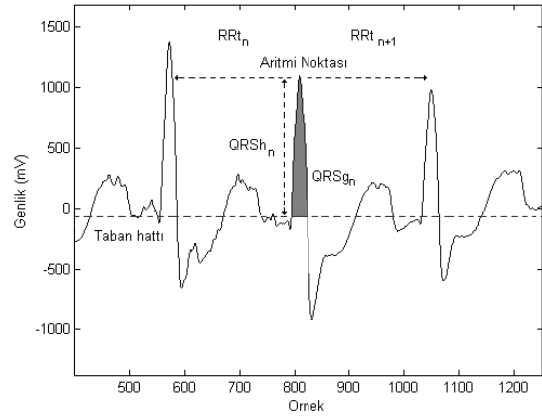
Karınca koloni optimizasyonu (KKO) ilk kez Dorigo ve Di Caro tarafından kombinasyonel optimizasyon problemlerinin çözümü için öne sürülmüştür [6]. Gerçek dünyadaki birçok problem kombinasyonel problemlere örnek olarak verilebilir. Örneğin, zaman çizelgesi oluşturma, araç yönlendirme, ders programı oluşturma vb. gibi problemler kombinasyonel optimizasyon problemleridir. Fakat bunun yanında sürekli değişkenlere ihtiyaç duyulan problemler de mevcuttur. Socha ve Dorigo, KKO'dan esinlenerek bu tür problemler için sürekli zaman karınca koloni optimizasyonunu (KKO_R) önermişlerdir [8]. Burša ve Lhotská (KKO_R) 'yi kullanarak RTFA ağını eğitmişler ve UCI veri tabanında bulunan bazı veri kümelerini sınıflandırmışlardır [9].

Literatürde birkaç çalışmaya rastlanmasına rağmen EKG vurularının RTFA yardımıyla sınıflandırılması, diğer yöntemlere kıyasla son zamanlarda kullanılan bir yöntem olarak kabul edilebilir. Azemi A. ve ark. RTFA, çok katmanlı ağ (ÇKA) ve k-en yakın komşuya sınıflayıcı kullanarak EKG vurularını sınıflamışlardır. Öznitelik çıkartma yöntemi olarak bağımsız bileşenler analizi ve dalgacık dönüşümünü kullanmışlardır [10]. Chudacek V. ve ark. EKG vurularını sınıflamak için on üç şekilsel öznitelik elemanı kullanarak, RTFA'nın da içinde bulunduğu yedi farklı yöntemi denemişlerdir [11]. Doğan B. ve Korürek M., dört farklı şekilsel öznitelik elemanı kullanarak RTFA'nın EKG vurularını sınıflamadaki performansını değerlendirmişlerdir [12]. Bu çalışmalarda kullanılan RTFA eğitim yöntemleri klasik yöntemlerdir.

EKG vurularını sınıflamada sürü zekasından esinlenerek geliştirilen optimizasyon algoritmalarının kullanılması oldukça yeni bir yöntemdir. Melgani F. ve Bazi Y., EKG vurularını sınıflamak için parçacık sürü optimizasyonu (PSO) yardımıyla destek vektör makinalarının (DVM) parametrelerini optimize etmiştir. Sınıflama sonrasında

önerdikleri yöntemi sekiz farklı yöntemle kıyaslamışlar ve önerilen yöntemin en yüksek başarıyı verdiğini gözlemlemişlerdir [13]. Korürek M. ve Nizam A., karınca koloni optimizasyonunu (KKO) kullanarak EKG vurularını sınıflamışlardır [14]. Bu çalışmada kullanılan algoritma Dorigo ve Di Caro tarafından önerilen klasik KKO algoritmasıdır [6]. Son olarak Korürek M. ve Doğan B., parçacık sürü optimizasyonu (PSO) yardımıyla RTFA ağına parametrelerini optimize etmiş ve altı farklı EKG vurusunu sınıflamışlardır [15].

Bu çalışmada da oldukça yeni bir optimizasyon algoritması olan sürekli zaman karınca koloni optimizasyonu (KKO_R) algoritması yardımıyla RTFA'nın nöron merkezleri ve her bir nöronun yayılım parametresi optimize edilmiş ve altı farklı EKG vurusu sınıflandırılmıştır.



Şekil 2: Şekilsel öznitelikler.

2. Yöntem

MIT-BIH veri tabanından elde edilen kayıtlar filtreleme ve taban hattı düzeltme gibi ön işlemlerden geçirildikten sonra R tepesi deteksiyonu yapılmıştır. R tepeleri sıklıkla kullanılan Pan-Tompkins algoritması ile bulunmuştur. Kayıtlardaki vuruların R tepeleri bulunduktan sonra, R tepelerinin her iki yanından 128 örnek alınmak suretiyle altı farklı vuru tipine ait EKG vuruları elde edilmiştir. Bu vurular sırasıyla, Normal (N), Premature Ventriküler Kontraksiyon (PVC), Ventriküler ve Normal Vuru Füzyonu (F), Atrial Prematüre Vuru (A), Sağ Yan Dal Bloğu (R) ve Yapay ve Normal Vuru Füzyonu'ndan (f) oluşmaktadır. Daha sonra bu vurulara ait dört adet şekilsel öznitelik elemanı çıkartılmıştır. Bu elemanlar sırasıyla, ilgili vuru ile bir önceki vuru arasındaki mesafe RR_t , ilgili vuru ile bir sonraki vuru arasındaki mesafe RR_{t+1} , taban hattından itibaren QRS yüksekliği QRS_h ve QRS genişliği QRS_g 'den oluşmaktadır. Elde edilen öznitelikler ilgili kayıttaki sekiz adet normal vurunun öznitelik değerleri baz alınarak normalize edilmiştir. Eğitim kümesi her vuru tipinden 50'şer adet olmak üzere toplam 300 öznitelik vektöründen oluşmaktadır. Şekil – 2'de kullanılan öznitelik elemanları görülmektedir. Daha sonra eğitim kümesi RTFA'nın girişine uygulanmış ve sürekli zaman karınca koloni optimizasyonu algoritması yardımıyla ağı parametreleri optimize edilmiştir. Bulunan optimum ağ yardımıyla da test kümesi sınıflandırılmış ve sonuçlar klasik RTFA eğitim yöntemleri ile kıyaslanmıştır.

2.1. Radyal Tabanlı Fonksiyon Yapay Sinir Ağı (RTFA)

Radyal tabanlı fonksiyon yapay sinir ağı, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşan doğrusal olmayan hibrid bir ağıdır. Şekil – 1’den RTFA’nın genel yapısı görülebilmektedir. RTFA çıkışının matematiksel ifadesi en genel haliyle Eşitlik – 1 ile verilebilir.

$$c_j(x) = \sum_{i=1}^k \phi(\omega_{ji} \|x - \mu_i\|; \sigma_i) \quad (1)$$

Burada $c_j(x)$, j . sınıfa ait çıkışı, $\phi()$ ise nöron merkezi μ_i ve ilgili nörona ait yayılım parametresi σ_i ’den oluşan radyal tabanlı fonksiyonu (RTF) temsil etmektedir. ω_{ji} ise i . nöron merkezi ile j . sınıf çıkışı arasındaki ağırlığı temsil etmektedir.

Radyal tabanlı fonksiyon yapay sinir ağının nasıl çalıştığını anlamak açısından literatürde “Cover’s Theorem” olarak bilinen teoreme kısaca değinmekte yarar vardır. Bu teorem şunu ifade etmektedir: şayet düşük boyutlu bir öznelik uzayını doğrusal olarak ayıramıyorsanız, bu uzayı doğrusal olmayan bir şekilde daha yüksek boyutlu bir uzaya taşırsınız ve bu yüksek boyutlu uzay doğrusal olarak ayrılabilir bir uzaydır. Radyal tabanlı fonksiyon yapay sinir ağının yaptığı şey de tam olarak budur. Girişe uygulanan herhangi bir öznelik vektörünün her bir nöron merkezine ne kadar ait olduğunun bir ölçüsü olarak çıkışta yeni bir öznelik vektörü oluşur. Genelde nöron sayısı öznelik eleman boyutundan fazla olduğu için oluşan aitlik vektörünün boyutu öznelik vektörünün boyutundan fazladır. Oluşan aitlik vektör kümesi doğrusal olarak ayrılabilen bir kümedir. Böylece aitlik vektörleri doğrusal olarak ağırlıklandırılıp toplanarak ilgili giriş için ağ çıkışı kolaylıkla üretilebilmektedir. Örüntü tanıma problemlerinde yüksek boyutlu uzaya doğrusal olmadan taşıma işlemi genelde Gaussian fonksiyonu yardımıyla yapılır. Böylece Eşitlik -1 ile verilen ifade yeniden düzenlenecek olursa,

$$c_j(x) = \sum_{i=1}^k \omega_{ji} \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma_i^2}\right) \quad (2)$$

elde edilir. Dikkat edilecek olursa ağ çıkışı, k nöron sayısına, μ_i nöron merkezine, σ_i her bir nöronun yayılım parametresine ve ω_{ji} ağırlıklarına bağlıdır. Bu çalışmada her bir nöronun μ_i nöron merkezi ve σ_i yayılım parametresi sürekli zaman karınca koloni optimizasyonu algoritması ile optimum olarak bulunmuştur. ω_{ji} ağırlıkları ise sözde-ters (pseudo-inverse) yöntemiyle bulunmuştur.

2.2. Sürekli Zaman Karınca Koloni Optimizasyonu

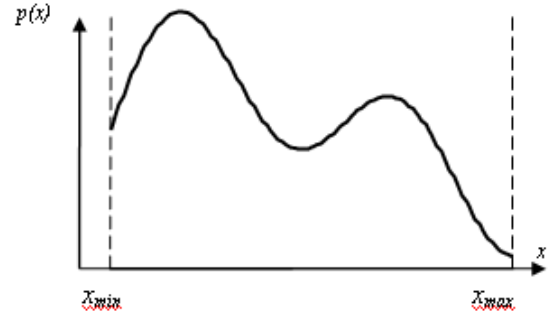
Kombinasyonel optimizasyon algoritmalarında olduğu gibi sürekli zaman optimizasyon algoritmaları için de bir model tanımlanabilir [9]. $Q=(S,\Omega,f)$ bir model olmak üzere,

- Sürekli karar değişkenlerinin sonlu kümesi üzerinde tanımlı ve değişkenler üzerindeki kısıtlamaları ihtiva eden Ω kümesini içeren bir S arama uzayı,
- $f: S \rightarrow R_0^+$ minimize edilmesi gereken bir uyumluluk fonksiyonu

biçiminde tanımlanabilir.

S arama uzayı, X_i , $i=1,\dots,n$ sürekli değişken kümelerinden oluşur. Bir $s \in S$ çözümü, herhangi bir sürekli zaman optimizasyon probleminin Ω kümesindeki bütün kısıtlamaları sağlayan ve bütün değişkenleri içeren bir çözümdür. Bir $s^* \in S$ çözümü ancak ve ancak $f(s^*) \leq f(s)$ şartı sağlanıyorsa küresel (global) bir çözüm olarak kabul edilir. Sürekli bir optimizasyon probleminin çözümü en az bir $s^* \in S^*$ çözümü içermelidir.

Sürekli zaman karınca koloni optimizasyonunda klasik karınca koloni optimizasyonu algoritmasından farklı olarak sürekli bir olasılık yoğunluk fonksiyonu (PDF) kullanılmaktadır. Bu olasılık yoğunluk fonksiyonu Şekil – 3’teki gibi verilebilir.



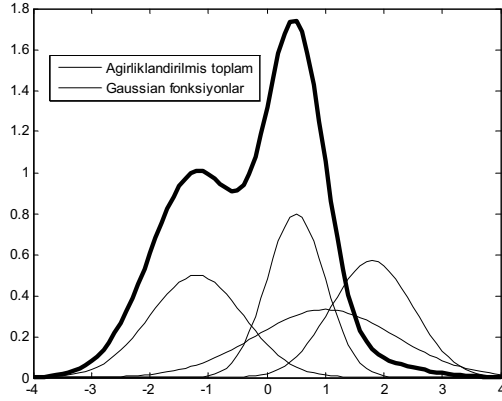
Şekil 3: Olasılık yoğunluk fonksiyonu.

Socha ve Dorigo birden fazla Gaussian tabanlı olasılık yoğunluk fonksiyonunun ağırlıklandırılmış toplamından oluşan ve Eşitlik – 3 ile verilen olasılık yoğunluk fonksiyonunu tanımlamışlardır [8].

$$G^i(x) = \sum_{l=1}^k w_l g_l^i(x) = \sum_{l=1}^k w_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^i{}^2}} \quad (3)$$

Burada w her bir Gaussian fonksiyonunun ağırlığını, μ^i ortalama vektörünü ve σ^i standart sapma vektörünü temsil etmektedir. Şekil – 4’te Eşitlik – 3 ile verilen birden fazla Gaussian fonksiyonunun ağırlıklandırılmış toplamından oluşan Gaussian çekirdek fonksiyonu ve her bir Gaussian fonksiyonu görülebilir.

KKO_R’de feromen bilgisi bir tablo içerisinde saklanır. Bu tablo, mevcut problemin aday çözümleri ve bu çözümlerin uyumluluk fonksiyonları $f(s_i)$ ’leri içerir. Tablonun yapısı Şekil – 5’te verilmiştir. Tabloda çözümler uyumluluk fonksiyonları baz alınarak ($f(s_1) \leq f(s_2) \leq f(s_3) \leq \dots \leq f(s_k)$) sıralanmıştır. Ayrıca tabloda, çekirdek Gaussian fonksiyonu oluşturan k adet Gaussian fonksiyon bulunur.



Şekil 4: Gaussian fonksiyonlar ve fonksiyonların ağırlıklandırılmış toplamı.

G^1, G^2, \dots, G^n 'ler Gaussian çekirdek fonksiyonlarıdır. Burada n problemin boyutunu temsil etmektedir. Gaussian çekirdek fonksiyonun ortalaması μ , ilgili çözüm için Eşitlik - 4'teki gibi tanımlanır.

$$\mu^i = \{\mu_1^i, \dots, \mu_k^i\} = \{s_1^i, \dots, s_2^i\} \quad (4)$$

Çözümler çözüm tablosuna eklendikten sonra uygunluk değerlerine göre sıralanır. Daha sonra Eşitlik - 5 ile verilen ağırlık vektörü ω tanımlanır.

$$\omega_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \quad (5)$$

Ağırlık vektörü l argümanlı, 1.0 ortalamalı ve qk standart sapmalı bir Gaussian fonksiyondur. q değeri küçüldükçe iyi çözümlerin seçilme olasılığı artar.

s_1	s_1^1	...	s_1^i	...	s_1^n	$f(s_1)$	ω_1
s_2	s_2^1	...	s_2^i	...	s_2^n	$f(s_2)$	ω_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
s_l	s_l^1	...	s_l^i	...	s_l^n	$f(s_l)$	ω_l
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
s_k	s_k^1	...	s_k^i	...	s_k^n	$f(s_k)$	ω_k
	G^1		G^i		G^n		

Şekil 5: Çözüm tablosu.

Çözüm tablosundaki bir çözümü seçip güncelleyebilmek için, çözüm tablosundaki her bir satırın (Gaussian fonksiyonların) seçilebilme olasılığının ilgili satırın ω ağırlık vektörü göz önüne alınarak hesaplanması gerekir. Eşitlik - 6'da her bir satırın seçilebilme olasılığının matematiksel ifadesi verilmektedir.

$$p_l = \frac{\omega_l}{\sum_{r=1}^k \omega_r} \quad (6)$$

Herhangi bir çözüm olasılık değeri göz önünde bulundurularak seçildikten sonra, ilgili çözümün parametreleri Eşitlik - 7 ile verilen bir σ standart sapma değeri baz alınarak güncellenir.

$$\sigma_j^i = \xi \sum_{e=1}^k \frac{|s_e^i - s_j^i|}{k-1} \quad (7)$$

σ standart sapma değeri, seçilen çözümün çözüm tablosundaki diğer çözümlere olan ortalama uzaklığı olarak tanımlanabilir. $\xi > 0$ değeri klasik karınca koloni optimizasyonundaki feromon buharlaşma katsayısına benzer bir işleve sahiptir. ξ değeri büyüdükçe algoritmanın yakınsama hızı azalır. Tablo - 1'de KKO_R algoritması daha açık bir anlatımla verilmiştir.

Tablo 1: KKO_R algoritması

```

Çözüm tablosunu rasgele oluştur
while sonlandırma şartı sağlanmadıkça do
  Çözümlerin uygunluk değerini hesapla
  Çözümleri uygunluk değerine göre
    büyükten küçüğe doğru sırala
   $\omega$  ağırlık vektörünü hesapla (Eşitlik - 5)
  Olasılık değerlerini baz alarak (Eşitlik - 6)  $t^*$  adet
    Gaussian fonksiyonu (satır) seç
  Seçilen çözümlerin  $\sigma$  değerlerini hesapla (Eşitlik - 7)
   $\sigma$  değerlerini kullanarak seçilen çözümlerden
    yeni çözümler üret
  Yeni çözümleri çözüm tablosunun son  $t$  satırı
    ile değiştir
end while

```

t değeri çözüm tablosuna eklenecek aday çözüm sayısını temsil eder.

2.3. Radyal Tabanlı Fonksiyon Yapay Sinir Ağının KKO_R Algoritması ile Optimizasyonu

Radyal tabanlı fonksiyon yapay sinir ağının başlıca üç parametresi vardır. Bunlar, gizli katmandaki nöronların merkezleri, her bir nöronun yayılım parametreleri ve gizli katman ile çıkış katmanı arasındaki ağırlıklardır. Daha önce çalışmada gizli katman ile çıkış katmanı arasındaki ağırlıkların sözde-ters (pseudo-inverse) yöntemi ile bulunduğu değinilmişti. Bu durumda optimizasyon aşamasında sadece gizli katmandaki nöron merkezleri ve her bir nörona ait yayılım parametresi optimize edilecektir. Böylece çözüm tablosuna eklenecek her bir çözüm Eşitlik - 8 ile verilen aday bir ağı temsil edecek olursa,

$$net = [(c_1, \sigma_1), (c_2, \sigma_2), \dots, (c_n, \sigma_n)] \quad (8)$$

elde edilir. Burada net çözüm tablosunda $\{s_1^1, \dots, s_1^n\}$ çözümüne karşılık düşer. c ve σ sırasıyla nöron merkezlerini ve her bir nöronun yayılım parametresini temsil etmektedir. Uyumluluk fonksiyonu $f(s)$ Eşitlik - 9 ile verilmektedir.

$$E = N_t \times \log \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \|y_i - o_i\|^2 \right) \quad (9)$$

Burada N_t eğitim kümesindeki toplam öznitelik vektörü sayısını, y_i arzu edilen çıkışı, o_i ise ağ çıkışını temsil etmektedir. Algoritma Tablo - 1'deki gibi başlatılır ve optimum ağ bulunur.

3. Sonuçlar

Tablo-2’de farklı farklı sınıflayıcıların test kümesi üzerindeki sınıflama başarısı verilmiştir. Bütün sınıflayıcılar için maksimum sınıflama başarısı dikkate alınmıştır. Testler MATLAB ortamında 1.86 GHz işlemcili ve 2GB RAM’e sahip bir bilgisayarda yapılmıştır. Dikkat edilecek olursa önerilen yöntem, sınıflama başarısından ödün vermeden çok daha küçük bir ağ yapısı kullanılarak EKG vurularını sınıflamaktadır. Toplam sınıf sayısına eşit sayıda nöron kullanıldığı durumda bile önerilen yöntemin sınıflama başarısı kabul edilebilir bir düzeydedir. Yöntemin bu denli başarılı bir sonuç vermesinin altında yatan en büyük etkenlerden biri, önerilen yöntemle her bir nöronun σ yayılım parametresinin de ayriyeten optimize ediliyor olmasıdır. Klasik RTFA eğitim algoritmalarında σ değeri bütün nöronlar için aynı kabul edilir ve Eşitlik – 10 ile hesaplanır.

$$\sigma = \frac{d_{\max}}{\sqrt{2k}} \quad (10)$$

Burada d_{\max} , iki nöron arasındaki maksimum mesafeyi, k ise toplam nöron sayısını temsil etmektedir.

Tablo 2: Farklı sınıflayıcıların sınıflama başarısı

Sınıflayıcı	Nöron sayısı	Duyarlılık %	Eğitim süresi	Test süresi
K-Ort. - RTFA	25	95.232	0.346 sn.	0.49 sn.
OLS - RTFA (Matlab newrb) ($\sigma = 0.4$)	25	95.377	3.806 sn.	0.52 sn.
KKO _R - RTFA	10	95.595	52.69 sn.	0.24 sn.
KKO _R -RTFA	6	92.425	32.75 sn.	0.19 sn.

Tablo-3’te önerilen yöntemin 10 nöronla test kümesi üzerindeki sınıflama başarısı verilmektedir.

Tablo 3: Önerilen yöntemin 10 nöronla sınıflama başarısı

	N	PVC	F	A	R	f
N	822	0	3	2	9	0
PVC	1	591	9	3	1	0
F	14	2	229	0	2	1
A	4	4	2	155	0	0
R	4	0	0	1	145	0
f	0	2	5	0	0	123
Duyarlılık (Se) = $100 * TP / (TP + FN) = 94.69$						
Belirlilik (Sp) = $100 * TN / (TN + FP) = 99.49$						

İleriki çalışmalarda RTFA’nın nöron merkezlerini daha optimum bir şekilde belirleyebilmek için farklı yöntemler üzerinde çalışılacaktır. Önerilen yöntemde RTFA öznelik uzayında hiperküreler çizmektedir. Halbuki öznelik elemanlarının uzaydaki dağılımı çoğunlukla hiperküre şeklinde değildir. Bu ise optimizasyon algoritmasının yerel minimum noktalara takılmasına sebebiyet verebilir. Önerilen

yöntem iyi sonuçlar vermesine rağmen, farklı yöntemlerle RTFA’nın gizli katmanlarındaki nöronların, özneliklerin uzaydaki dağılım formunu alabileceği hibrid algoritmalar geliştirilerek daha etkin bir ağ yapısı elde edilebilir.

4. Kaynaklar

- Kohler B U, Hennig, C. and Orglmeister, R., "The Principles of Software QRS Detection" *IEEE Eng. Med. Biol. Mag.*, 21 42-57, 2002
- Devine, B., and Macfarlane, P.W., "Detection of Electrocardiographic left ventricular strain using neural nets", *Med. Biol. Eng. Comput.*, 31 pp. 343-348, 1993
- Broomhead, D. S. and Lowe, D., "Multivariable functional interpolation and adaptive networks," *Complex Systems*, pp. 321-355, 1988.
- Kennedy, J., & Eberhart, R., *Swarm intelligence*. 1-55860-595-9. Academic Press., 2001
- Kennedy, J., & Eberhart, R. C., "Particle swarm optimization." *In Proceedings of IEEE international conference on neural Networks*, pp. 1942–1948, 1995
- Dorigo M. & G. Di Caro, "Ant Colony Optimization: A New Meta-Heuristic.", *Proceedings of the 1999 Congress on Evolutionary Computation*, 1470–1477, 1999.
- Karaboga, D., Basturk, B., "A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", *Journal of Global Optimization*, 39-3, pp. 459-171, 2007.
- Socha K., Dorigo M., "Ant colony optimization for continuous domains". *European Journal of Operational Research* 185 1155--1173, 2008
- Burša, M., Lhotská L., "Nature Inspired Methods in the Radial Basis Function Network Learning Process." *Artificial Neural Networks ICANN 2008*, 829--838. 2008
- Azemi, A., Sabzevari, V. R., Khademi, M., Gholizade, H., Kiani, A., Dastgheib, Z. S., "Intelligent arrhythmia detection and classification using ICA", *Engineering in Medicine and Biology Society*, 2006
- Chudacek, V., Petrik, M., Georgoulas, G., Cepek, M., Lhotska, L., & Stylios, C., "Comparison of seven approaches for holter ECG clustering and classification." *Engineering in Medicine and Biology Society*, 3844–3847, 2007.
- Doğan B., & Korürek M., "EKG vurularını sınıflamada radyal tabanlı fonksiyon yapay sinir ağının performans değerlendirilmesi", *BIYOMUT 2009, Dokuz Eylül Üniv. İzmir*, 2009
- Melgani, F., & Bazi, Y., "Classification of electrocardiogram signals with support vector machines and particle swarm optimization. ", *Information Technology in Biomedicine IEEE Transactions on*, 12(5), 667–677, 2008.
- Korurek, M., & Nizam, A., "A new arrhythmia clustering technique based on ant colony optimization.", *Journal of Biomedical Informatics*, 41(6), 874–881, 2008.
- Korürek, M., & Doğan, B. "ECG beat classification using particle swarm optimization and radial basis function neural network. ", *Expert Systems with Applications*, 37 12, pp. 7563- 7569, 2010.