

SÜRÜ ZEKASI YÖNTEMLERİYLE AŞIRI ÖĞRENME MAKİNESİ'NİN ÖĞRENME PARAMETRELERİ OPTİMİZASYONU

Musa DOĞAN, İlker Ali ÖZKAN

Selçuk Üniversitesi, Bilgisayar Mühendisliği Bölümü, Konya Türkiye

musa.dogan@selcuk.edu.tr (ORCID 0000-0001-8757-2869)

ilkerozkan@selcuk.edu.tr (ORCID 0000-0002-5715-1040)

Özet

Sinir ağları algoritmalarından olan Aşırı Öğrenme Makinesi (AÖM)'de giriş ağırlığı ve gizli eşik değeri parametrelerinin rastgele seçilmekte ve çıktı katman ağırlıkları analitik olarak hesaplanmaktadır. Bundan dolayı ağırlık öğrenme işlemi hızlı bir şekilde gerçekleşmektedir. Ayrıca AÖM'nin gradyan temelli algoritmalara göre gizli katmanda ihtiyaç duyduğu nöron sayısı daha fazla olmaktadır. Bu nedenle giriş ağırlıkları ve gizli nöron eşik değerlerinin optimum değerlerinin bulunması AÖM'nin performansına etki etmektedir. Bu çalışmada bu optimum değerlerin belirlenmesinde sürü zekası algoritmalarından Parçacık Sürü Optimizasyonu (PSO) ve Rekabetçi Sürü İyileştirici (RSİ) kullanılmıştır. Optimum giriş ağırlıkları ve gizli eşik değerlerinin belirlenerek çıkış ağırlıkları Moore-Penrose genelleştirilmiş tersiyle analitik olarak hesaplanmıştır. AÖM, RSİ-AÖM ve PSO-AÖM modellerinin çok sınıflı tiroid veri setine uyarlanarak öğrenme parametrelerinin optimizasyonu ile en iyi doğruluk oranları sırasıyla %94.74, %94.86, %95.42 olarak elde edilmiştir. Optimizasyon metotlarının AÖM modellerinin sınıflandırma performansını artırdığı görülmüştür.

Anahtar Kelimeler: Aşırı Öğrenme Makinesi (AÖM), Metasezgisel, Parçacık Sürü Optimizasyonu (PSO), Rekabetçi Sürü İyileştirici (RSİ)

OPTIMIZATION OF LEARNING PARAMETERS OF EXTREME LEARNING MACHINE WITH SWARM INTELLIGENCE METHODS

Abstract

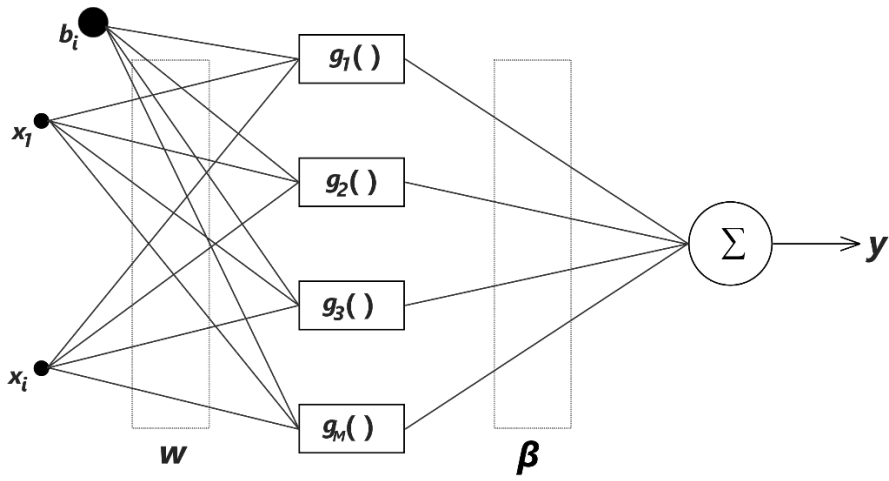
In the Extreme Learning Machine (ELM), which is one of the neural networks algorithms, the input weight and hidden bias value parameters are randomly selected and the output layer weights are calculated analytically. Therefore, the learning process of the network takes place quickly. In addition, the number of neurons needed by the hidden layer is higher than the gradient-based algorithms. Finding optimum values of entry weights and hidden neuron bias values affects the performance of the ELM. In this study, Particle Swarm Optimization (PSO) and Competitive Swarm Optimizer (CSO) were used to determine these optimum values. By determining the optimum input weights and hidden bias values, the output weights were analytically calculated by Moore-Penrose generalized inverse. By adapting the multi-class thyroid data set of ELM, CSO-ELM and PSO-ELM models, the best accuracy rates were obtained as 94.74%, 94.86%, 95.42% respectively. It has been seen that optimization methods increase the classification performance of the ELM models.

Keywords: Extreme Learning Machine (ELM), Metaheuristic, Particle Swarm Optimization (PSO), Competitive Swarm Optimizer (CSO)

1. Giriş

İleri beslemeli YSA'lar belirgin özellikleri nedeniyle bir çok alanda fonksiyon yaklaşımı için yaygın olarak kullanılmaktadır [1], [2]. Hem doğrudan girdi örneklerinden karmaşık doğrusal olmayan eşlemeleri yaklaşık olarak göstermek amacıyla hem de klasik parametrik tekniklerin kullanımı zor olan çok sayıda doğal ve yapay olay için model üretmesinde kullanılmaktadır [3]. Bir sınıflandırıcı olarak YSA'yı değerlendirirken, gizli katmandaki nöron sayısı, giriş katmanı ile gizli katman arasındaki ve gizli katman ile çıktı katman arasındaki ağırlıkların değerleri ve öğrenme algoritmasının seçimi önemli rol oynamaktadır. YSA'nın en büyük dezavantajı ise, eğitim sürecinde ağırlıkları ve parametreleri ayarlamak için modelin performansını zaman alıcı hale getiren ve hesaplama yükünü artıran gradyan iniş algoritmasının kullanılmasıdır [4]. Huang ve ark. (2006), tek

gizli katmana sahip aşırı öğrenme makinesi (AÖM) ismini verdikleri yeni bir ileri beslemeli bir YSA algoritması sunmuşlardır [5]. AÖM'de geleneksel ileri beslemeli YSA'lardan farklı olarak giriş ağırlıkları ve gizli eşik değeri rastgele verilmekte, çıkış katmanındaki nöronların ağırlıkları hesaplanmaktadır. Önerdikleri bu yaklaşımda eğitim sırasında yineleme olmadan tüm işlemlerin yapılması çoğu uygulama için öğrenme aşamasını çok kısa bir süre içerisinde tamamlanmasını mümkün kılmaktadır. Ayrıca yapılan çalışmalarla AÖM'nin çoğu gradyan temelli öğrenmeden daha iyi bir genelleme performansına sahip olduğu yapılan çalışmalarla gösterilmiştir [5]. Tek gizli katmana sahip AÖM modeli Şekil 1'de gösterilmiştir.



Şekil 1. Tek gizli katmanlı AÖM modeli

Standart ileri beslemeli bir sinir ağında gizli katmanda M tane nöron bulunan bir modelin matematiksel gösterimi Denklem (1)'de verilmiştir.

$$\sum_{i=1}^M \beta_i g(w_i \cdot x_j + b_i), j = 1, \dots, N. \quad (1)$$

burada β_i çıkış ağırlıkları, g aktivasyon fonksiyonu, w_i giriş ağırlıkları ve b_i 'de eşik değeridir. Yukarıdaki eşitlikte gizli katman çıkış matrisi $H = g(w_i \cdot x_j + b_i)$ olarak ifade edilirse sade bir şekilde Denklem (2) elde edilir.

$$H\beta = y \quad (2)$$

H matrisi Denklem (3)'te, β ve y değerleri Denklem (4)'te verilmiştir.

$$\begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_M \cdot x_1 + b_M) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_M \cdot x_N + b_M) \end{bmatrix}_{N \times M}, \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_M \end{bmatrix}_{M \times 1} \quad \text{ve} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1} \quad (4)$$

H matrisi'nin i . sütunu, x_1, x_2, \dots, x_N girişlerine göre i . gizli nöron çıkışıdır. AÖM yaklaşımında w_i ve b_i değerleri rastgele atanır ve çıkış ağırlıkları Denklem (5)' te gösterilen denklem ile hesaplanır.

$$\hat{\beta} = H^+ y \quad (5)$$

Denklem (5)'te verilen denklemde $\hat{\beta}$ yaklaşık çıkış ağırlığı ve H^+ , H matrisinin genelleştirilmiş ters Moore-Penrose matrisi olur. Bu matris aynı zamanda pseudoinverse matrisi olarak adlandırılır. Moore-Penrose ters matrisi küçük normda en küçük kareler çözümü sağlar. Denklem (5)'i ifade etmenin bir başka yolu Denklem (6) ise şu şekildedir:

$$\|H\hat{\beta} - y = 0\| \quad (6)$$

AÖM algoritması aşağıda verilen 3 adımda özetlenebilir:

1. Adım: Gizli katman nöron sayısını tanımlama, giriş parametrelerine (ağırlık w_i ve eşik değeri b_i) rastgele atama yapma
2. Adım: H Gizli katman çıktı matrisini hesaplama
3. Adım: Eşitlik (5)'de göre $\hat{\beta}$ Çıktı ağırlığını hesaplama

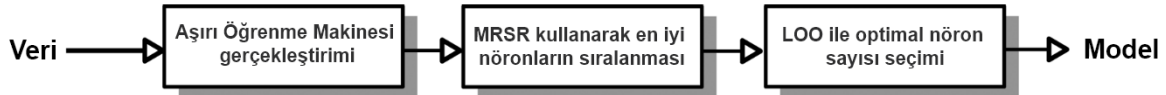
AÖM'nin diğer yapay zeka uygulamalarına nazaran eğitimi çok kısa bir sürede bitirmesi en büyük avantajlarından birisidir. Bunun yanı sıra geleneksel klasik gradyan temelli öğrenme algoritmaları hatanın yerel bir noktaya takılması, düzensiz öğrenme oranı ve aşırı uyma gibi sorunlarla uğraşırken AÖM bu sorunlarla uğraşmadan sonuca ulaştığı belirtilmiş, ayrıca standart UCI veri setleri üzerinde yapılan çok sayıda deneyde, geleneksel öğrenme algoritmalarından daha iyi bir genelleme performansına sahip olduğu yapılan çalışmalarla gösterilmiştir [5], [6], [7]

AÖM'nin dikkate değer eğitim hızı ve genelleştirme performansından dolayı araştırmacılar, sunulan geleneksel AÖM modelini geliştirerek çeşitli modeller ortaya koymuşlardır. Artırımlı AÖM (A-AÖM) adı verilen modelde gizli katmandaki düğümler tek tek rastgele eklenir, yeni bir gizli düğüm eklenmesi sırasında mevcut gizli düğümlerin çıkış

ağırlıkları dondurulur. Yapılan bu çalışmayla teorik olarak rastgele oluşturulmuş gizli düğümlere sahip ileri beslemeli sinir ağlarının evrensel tahminçiler olduğu ortaya koyulmuştur [8].

Kompakt ağlardaki başarıyı artırmak için evrimsel AÖM (E-AÖM) olarak adlandırılan bir öğrenme algoritması önerilmiştir. Önerilen algoritmada AÖM adımlarına ilaveten girdi ağırlıklarını ve gizli eşik değerlerini optimize etmek için diferansiyel evrim (DE) nin modifiye edilmiş bir formu kullanılmıştır. Deneysel sonuçlarla E-AÖM'nin geri yayımlı sinir ağı, GALS ve geleneksel AÖM'den genellikle daha iyi genelleştirme performansına ulaştığını göstermişlerdir [9].

AÖM algoritmasında birbirinden alakasız veya ilişkisi olmayan verilerle karşılaşıldığı zaman bazı sorunların olabileceği görülmüştür. Bu sebeple orjinal AÖM algoritmasına dayalı optimum budanmış AÖM (OB-AÖM) metodolojisi önerilmiştir. Şekil 2'de görüldüğü gibi bu metodolojide önce çok yanıtı seyrek regresyon (MRSR) ile gizli katmanda bulunan kullanışsız nöronlardan kurtulmak için nöronlar sıralamaya sokulmuş ardından leave-one-out (LOO) ile optimal nöron seçilmiştir [10].



Şekil 2. OB-AÖM Modeli

Kendinden Uyarlamalı Evrimsel AÖM (KUE-AÖM) algoritması ile gizli katmandaki öğrenme parametreleri kendinden uyarlamalı diferansiyel evrim algoritması ile optimize edilmiştir. Test vektörü oluşturma stratejileri ve bağımlı kontrol parametreleri önceki yinelemelerden öğrenilerek bir strateji havuzunda kendiliğinden uyarlanan ve ağ çıktı ağırlığı geliştirilmiş ters Moore-Penrose matrisi kullanılarak hesaplanır. E-AÖM [9] ve Diferansiyel Evrim-Levenberg Marquardt (DE-LM) [11] algoritmalarındaki test vektörü oluşturma stratejileri ve kontrol parametrelerinin manuel seçiminin ortaya çıkarttığı sıkıntıyı gidermekle beraber genelleştirme performansını da artırmıştır [12].

Gradyan tabanlı yöntemlerin karşılaştığı bazı sorunlarla başa çıkabilen AÖM, gerçek dünya problemlerine uygulandığında hızlı öğrenme hızı ve dikkate değer genelleme performansı nedeniyle araştırılmaya değer bulunmuştur. Bununla birlikte, AÖM'de giriş

ağırlıkları ve gizli eşik değerleri rastgele seçildiğinde gereksiz gizli nöronlar eklenmesiyle kötü sonuçlar verebilir [13].

Bu çalışmada literatürde belirtilen giriş ağırlıkları ve gizli eşik değerlerinin rastgele seçilmesiyle oluşan sınıflandırma performansındaki olumsuz etkisini gidermek amacıyla AÖM'nin öğrenme parametreleri olan giriş ağırlığı ve eşik değerlerinin sürü zekâsı algoritmalarından Parçacık Sürü Optimizasyonu (PSO) ve Rekabetçi Sürü İyileştirici (RSİ) ile model oluşturularak optimize edilmesi sağlanmıştır. Oluşturulan modellerin performanslarının ölçülmesi için Tiroit veri seti kullanılmıştır.

2. Materyal Metot

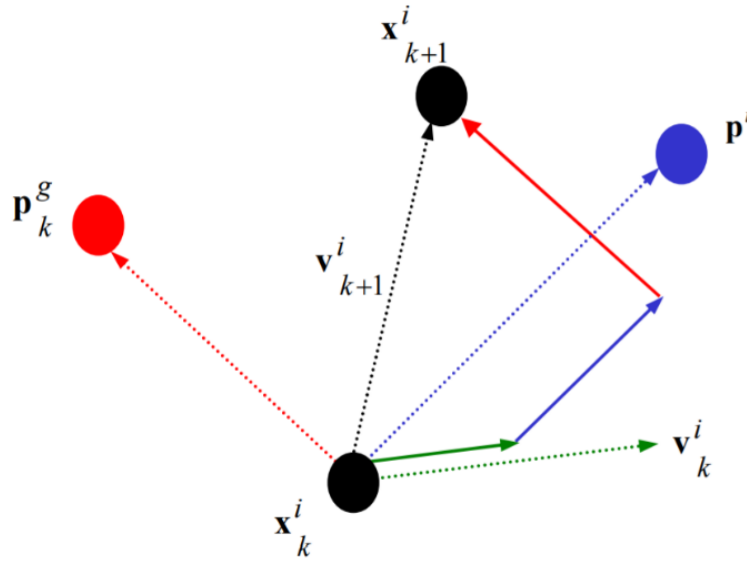
2.1. Parçacık Sürü Optimizasyonu (PSO)

PSO, Eberhart ve Kennedy tarafından geliştirilen popülasyon tabanlı optimizasyon algoritmasıdır. Sürü halinde hareket eden Kuş, Balık ve Karınca gibi hayvanların davranışlarından ilham alınarak geliştirilmiştir. PSO her bir kuşun “parçacık” olarak adlandırıldığı arama uzayında rastgele bir kuş sürüsünün ilk değer ataması yapılarak başlatılır. Bu parçacıklar parametre olarak verilen maksimum yineleme sayısına ulaşıncaya değin uyarlanabilir bir hızda V_i belirli bir konuma X_i uçarlar. Yine bu parçacıklar her yinelemede bulunan en iyi konuma (pbest) göre hızlarını ayarlar ve yineleme sonlandığında küresel en iyi pozisyon (gbest) değeri bulunmuş olur [14]. Şekil 3'te PSO algoritmasında parçacıkların hareketi gösterilmiştir.

Hız güncelleme işlemleri boyunca w, c_1 ve c_2 parametreleri dışarıdan girilmesi gereken parametrelerdir. Parçacıkların PSO'da hız ve konumlarının güncelleme işlemi sırasıyla Denklem (7) ve Denklem (8)'de gösterilmiştir.

$$v_i = wv_x + 2 * rand_{c_1} * (pbest_i - x_i) + 2 * rand_{c_2} * (gbestx - x_i) \quad (7)$$

$$x_i = x_i + v_x \quad (8)$$



Şekil 3. PSO’da parçacıkların hız ve konum güncellemeleri [15]

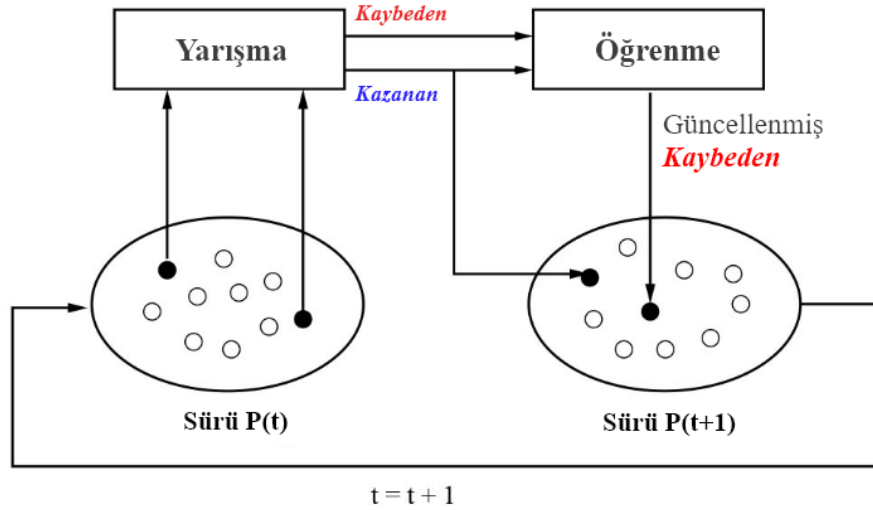
2.2. Rekabetçi Sürü İyileştirici (RSİ)

Cheng ve Jin tarafından önerilen RSİ algoritması fikir olarak PSO’dan esinlense de işleyiş olarak farklı bir algoritmadır. Bu algortmada PSO’dan farklı olarak, PSO algoritmasında bulunan her parçacığın kendine ait en iyi değeri (pbest) ve küresel en iyi pozisyonu (gbest) değerleri, parçacıkların güncellenmesi işlemlerinde kullanılmamaktadır. Bunun yerine Şekil 4’de belirtildiği gibi ikili gruplar halinde daha iyi uygunluğa sahip parçacık “kazanan” diğer parçacık ise “kaybeden” olarak nitelendirilerek parçacıkların güncellenmesi sağlanmaktadır. Sürülerden oluşturulan rastgele vektörlerden (R_1^t, R_2^t, R_3^t) 2/3’ünün yükseltme (upgrade) işlemine katılarak daha yüksek bir yakınsama oranı elde edilmiştir. Sürünün geri kalan 1/3’ü ise bir sonraki jenerasyona doğrudan aktarılarak çeşitlilik sağlanmış olur. Daha net bir ifadeyle sürüdeki çeşitlilik sürdürülmüş olur [16].

İkili yarışma yapıldıktan sonra “kaybeden” olarak nitelendirilen parçacıkların, “kazanan” parçacıklardan öğrenmesi sonucunda hız vektörleri ve konumların değişimi sırasıyla Denklem (9) ve Denklem (10)’da gösterilmiştir. Yarışma sonunda x_w kazanan parçacığın konumunu, x_l kaybeden parçacığın konumunu, v_l kaybeden parçacığın hızını ve v_w ise kazanan parçacığın hızını belirtmektedir. RSİ’de yinelemedeki ilgili parçacıkların ortalama konum değerinin etki kontrolü için ayarlanması gerek tek parametre Denklem (9)’da yer alan phi (φ) parametresidir [17].

$$v_i^{t+1} = R_1^t v_i^t + R_2^t (x_w^t - x_i^t) + \varphi R_3^t (x^{-t} - x_i^t) \quad (9)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (10)$$



Şekil 4. RSI’de parçacıkların güncelleme işlemleri [16]

2.3. Tiroit Veri Seti

Yapılan bu çalışmada UCI Machine Learning Repository’den (<http://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>) alınan tiroit veri seti kullanılmıştır [18]. Veri seti hastalara ait 15 ikili [0-1], 6 reel sayı olmak üzere 21 girdiden ve teşhis sonucuna göre 3 sınıftan oluşmaktadır:

- Sınıf 1: Hipertiroid bulunan bireyler {1}
- Sınıf 2: Hipotiroid bulunan bireyler {2}
- Sınıf 3: Sağlıklı bireyler (normal kabul edilen) {3}

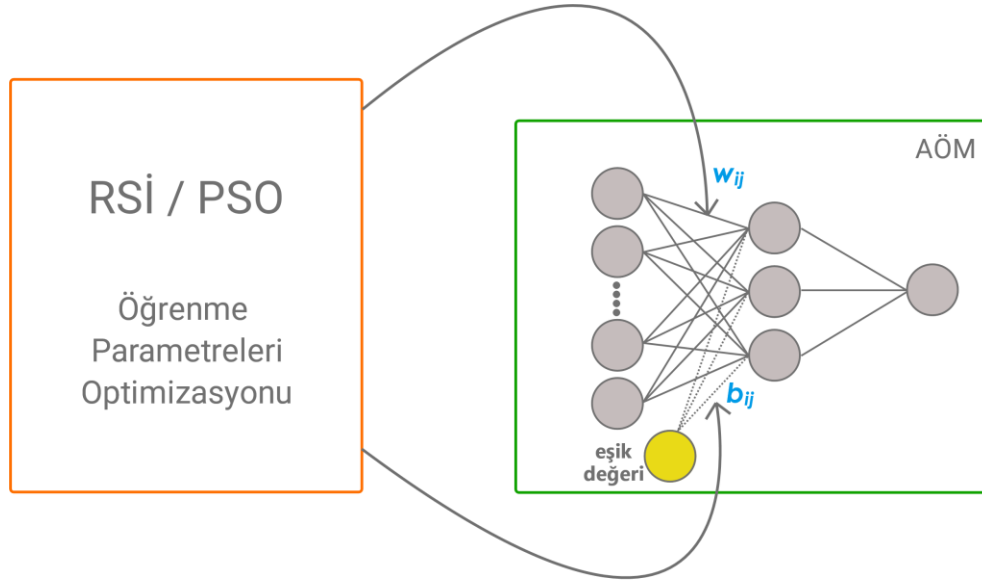
Veri seti toplam 7200 örnek içermektedir. Tablo 1’de veri setine ait girdilerin değer aralıkları ve ortalama değerleri verilmiştir.

Tablo 1. Tiroit Veri Setindeki Özelliklerin Değer Aralıkları ve Ortalama Değeri

Özellik Adı	Aralık	Ortalama Değer
Yaş	[0.01, 0.097]	0.5205
Cinsiyet	[0, 1]	0.3043
Tiroksin	[0, 1]	0.1306
Tiroksin'e ilişkin sorgulama	[0, 1]	0.0154
Anti tiroit ilacı	[0, 1]	0.0128
Hasta	[0, 1]	0.0383
Hamile	[0, 1]	0.0108
Tiroit Cerrahisi	[0, 1]	0.014
I131 tedavi	[0, 1]	0.0168
Hipotiroidizm	[0, 1]	0.0656
Hipertiroidizm	[0, 1]	0.0688
Lityum	[0, 1]	0.0126
Guatr	[0, 1]	0.0082
Tümör	[0, 1]	0.0256
Hipopitüiter	[0, 1]	0.0001
Psikolojik	[0, 1]	0.0489
TSH	[0.0, 0.53]	0.0049
T3	[0.00005, 0.18]	0.02
TT4	[0.002, 0.6]	0.1094
T4U	[0.017, 0.233]	0.0978
FTI	[0.002, 0.642]	0.1132

2.4. Önerilen Modeller

Bu çalışmada RSİ-AÖM ve PSO-AÖM modelleri kullanılarak giriş ağırlıkları ve gizli eşik değerleri optimize edilmiştir. Bu modellerin öğrenme işleminin grafiksel gösterimi Şekil 5'te verilmiştir. Bu optimizasyon modellerinin gizli katmandaki az sayıda nöron sayısı daha iyi sonuç verme durumunun araştırmak amacıyla nöron sayısı 5'den 50'ye kadar her seferinde kademeli olarak 5'er artırılmıştır. Optimize edilmemiş geleneksel AÖM için yüksek nöron sayılarında başarımlarını gözlemlemek amacıyla 50'den 200'e kadar 50'şer artırılmıştır. AÖM'de Sigmoid, Sin, Triangular, Hard Limit ve Radial Basis fonksiyonları aktivasyon fonksiyonu olarak kullanılmaktadır. Bu çalışmada kullanılan üç model (AÖM, RSİ-AÖM ve PSO-AÖM) için aktivasyon fonksiyonu olarak doğrusal olmayan, sinir ağlarında sıklıkla kullanılan sigmoid fonksiyon $[f(x) = 1 / (1 + \exp(-x))]$ kullanılmıştır.



Şekil 5. PSO-AÖM ve RSİ-AÖM modelleriyle parametre optimizasyonunun gösterimi

RSİ ve PSO’da maksimum yinleme sayısı ve popülasyon sayısı yapılan diğer çalışmalara uygun şekilde 100 olarak alınmıştır [19], [20], [21]. Bu iki algoritmaya ait parametreler ise Tablo 2’de verilmiştir.

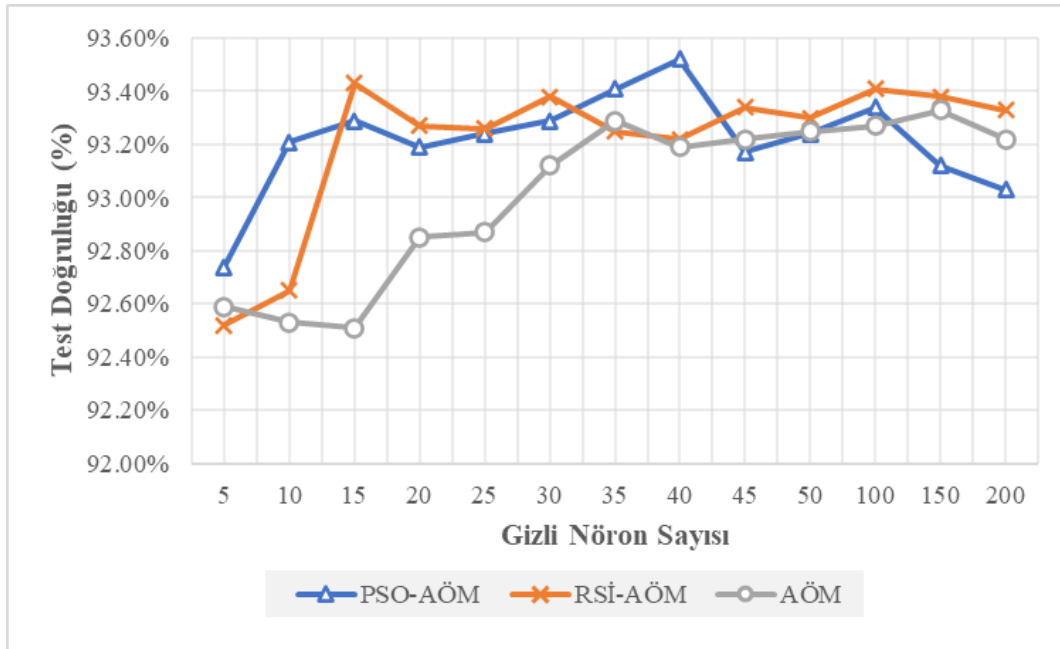
Tablo 2. PSO ve RSİ algoritmalarının parametreleri

Algoritma	Parametre	Değer
PSO	c1	2
	c2	2
	$W_{min-max}$	0.3-0.9
RSİ	phi	0.03, 0.05
PSO ve RSİ	Maks. Yinleme	100

Gerçekleştirilen her denemede modellerin eğitimi ve testi için kullanılan veriler rastgele olarak eğitim (%80), test (%10) ve doğrulama (%10) için ayrılmıştır. En iyi, en kötü, ortalama doğruluk değeri ile standart sapma ve eğitim hızları hesaplanmıştır.

3. Sonuçlar ve Analiz

Bu bölümde sürü zekasına dayalı yöntemler olan PSO-AÖM ve RSİ-AÖM ile geleneksel AÖM'nin Tiroit veri setindeki sınıflandırma performansları karşılaştırılmıştır. Çalışmaya ait tüm denemeler i7 6700HQ işlemcili bilgisayar üzerinde MATLAB yazılımı ile yapılmıştır. Aşırı Öğrenme Makinesi'nin gerçek dünya probleminde çoklu sınıflandırma işlemindeki başarımının karşılaştırılması amacıyla oluşturulan PSO-AÖM ve RSİ-AÖM modelleri tiroit veri seti üzerinde denenmiştir. RSİ-AÖM ve PSO-AÖM modellerinde ağırlık aşırı uyma'sının önüne geçmek için veri setinin %10'u doğrulama için kullanılmıştır. Her 50 denemeden sonra gizli katmandaki nöron sayısını artırılarak alınan maksimum doğruluğa dayalı sonuçlar Şekil 6'da verilmiştir.



Şekil 6. PSO-AÖM, RSİ-AÖM ve AÖM'nin Gizli Katman Nöron Sayılarına Göre Doğruluk Oranlarının Değişimi

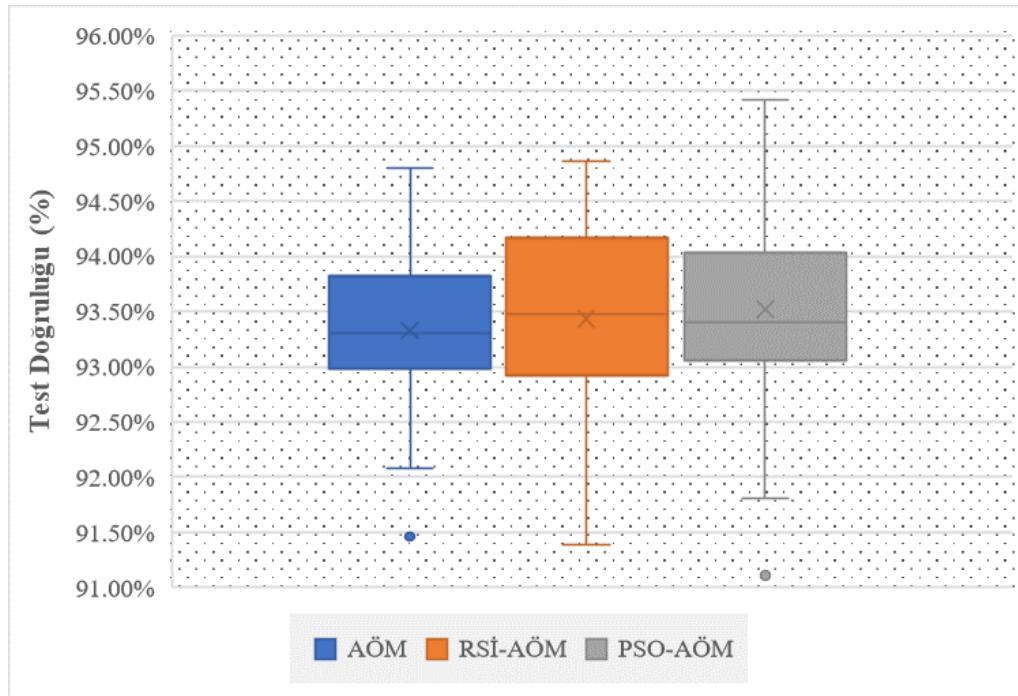
En iyi, en kötü ve ortalama doğruluk değerleriyle, standart sapma, eğitim süresi ve en iyi doğruluk değeri veren gizli katmandaki nöron sayısı Tablo 3'te verilmiştir. Çalışmada kullanılan tüm modellerde denemelerin doğruluk ve eğitim süresi ortalaması en iyi test doğruluk sonucu veren gizli nörondaki sonuç ile beraber verilmiştir. Tablo 3'te görüldüğü

üzere PSO-AÖM modeli ile elde edilen %93.52 ortalama doğruluk değeri ile RSİ-AÖM ve AÖM'ye üstünlük sağlanmıştır. Aynı zamanda PSO-AÖM modeli almış olduğu %95.42 en iyi doğruluk değeri ile, RSİ-AÖM modelinin almış olduğu %94.86 en iyi doğruluk değeri ve AÖM modelinin almış olduğu %94.74 en iyi doğruluk değerine kıyasla daha iyi bir sonuç vermiştir. Gizli katmandaki bulunan nöron sayısı incelendiğinde ise RSİ-AÖM'nin diğerlerine göre daha az gizli nöron sayısında elde ettiği iyi sonuç görülmektedir.

Tablo 3. Tiroit veri seti için AÖM, PSO-AÖM ve RSİ-AÖM'den alınan deneysel sonuçlar

Birim		AÖM	RSİ-AÖM	PSO-AÖM
Test	Ortalama \pm SS (%)	93.33 \pm 0.62	93.43 \pm 0.89	93.52 \pm 0.84
Doğruluk Oranı	En İyi (%)	94.74	94.86	95.42
	En Kötü (%)	91.46	91.39	91.11
Gizli Katman Nöron Sayısı		150	15	40
Eğitim Süresi \pm SS (s)		0.1869 \pm 0.0351	17.7497 \pm 0.3274	108.4743 \pm 0.5710

Şekil 7'de en iyi sonuç alınan gizli katmandaki test sonuçlarının gösterildiği kutu çizim grafiği verilmiştir.



Şekil 7 . PSO-AÖM, RSİ-AÖM ve AÖM'nin Kutu Çizim Grafiği

Literatürde Tiroit veri seti üzerinde denenmiş bir çok çalışma mevcuttur. Bunlardan, Yurtay ve ark. (2013), Levenberg Marquardt algoritmasıyla çok katmanlı sinir ağı çalışmalarında %93.19 doğruluk oranı elde etmişlerdir [22]. Berglund ve ark. (2018), AÖM'nin performansını iyileştirmek için Karınca Koloni Optimizasyonu (KKO) ile öznelik seçimi yapmışlardır. KKO-AÖM adını verdiği model ile çapraz doğrulama kullanarak 100 deneme sonucunda %92.59 ortalama doğruluk değerine ulaşılmıştır [23]. Yapılan bir başka çalışma da ise Abusnaina ve ark. (2018) Salp Sürü Algoritması (SSA) kullanarak gradyan temelli sinir ağı modelinin ağırlık katsayılarını optimize etmişlerdir. Çalışma sonucunda en iyi %94.86, ortalama %93.06 doğruluk değeri elde edilmiştir [24]. Literatürde yer alan diğer çalışmalar göz önüne alındığında, bu çalışmada önerilen PSO-AÖM modelinin %93.52 ortalama doğruluk oranı yapılan diğer çalışmalardan daha iyi bir sonuç vermektedir.

Alınan bu sonuçlarla sürü zekası algoritmalarından olan PSO ve RSİ'nin AÖM'nin rastgele verilen gizli eşik değeri ve giriş ağırlıklarının optimal değerini bulunmasında etkili bir yol olduğu gösterilmiştir. Genelleştirme performansı açısından baktığımızda PSO-AÖM, RSİ-AÖM ve AÖM'nin medikal veri setlerinden olan tiroit veri setinde iyi birer sınıflandırıcı oldukları yapılan bu çalışmayla gösterilmiştir. Parametrelerin optimizasyonu işlemiyle PSO-AÖM ve RSİ-AÖM modellerinin sınıflandırma doğruluğunu artırdığı ve gizli katmandaki nöron sayısını azalttığı görülmekte fakat eğitim süresi açısından incelendiğinde geleneksel AÖM modelinin diğer modellere kıyasla eğitimi çok daha kısa sürede gerçekleştirdiği görülmektedir. Optimizasyonda yapılacak iyileştirmelerle önemli performans ölçütlerinden olan test doğruluğunu yüksek seviyede tutarak, eğitimde geçen süre daha da aşağı çekilebilir.

Kaynaklar

- [1] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, 1991, doi: 10.1016/0893-6080(91)90009-T.
- [2] Y. Ito, "Approximation of continuous functions on R^d by linear combinations of shifted rotations of a sigmoid function with and without scaling," *Neural Networks*, 1992, doi: 10.1016/S0893-6080(05)80009-7.
- [3] S. Ding, X. Xu, and R. Nie, "Extreme learning machine and its applications," *Neural*

- Computing and Applications*, vol. 25, no. 3–4. Springer, pp. 549–556, 12-Dec-2014, doi: 10.1007/s00521-013-1522-8.
- [4] W. Sun, C. Wang, and C. Zhang, “Factor analysis and forecasting of CO2 emissions in Hebei, using extreme learning machine based on particle swarm optimization,” *J. Clean. Prod.*, vol. 162, pp. 1095–1101, Sep. 2017, doi: 10.1016/j.jclepro.2017.06.016.
- [5] G.-B. Bin Huang *et al.*, “Extreme learning Machine: Theory and Applications,” *Neurocomputing*, vol. 70, pp. 489–501, 2006, doi: 10.1016/j.neucom.2005.12.126.
- [6] C.-U. Yeom and K.-C. Kwak, “Short-Term Electricity-Load Forecasting Using a TSK-Based Extreme Learning Machine with Knowledge Representation,” *Energies*, vol. 10, no. 10, p. 1613, Oct. 2017, doi: 10.3390/en10101613.
- [7] W. Deng, Q. Zheng, and L. Chen, “Real-Time Collaborative Filtering Using Extreme Learning Machine,” in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009, pp. 466–473, doi: 10.1109/WI-IAT.2009.80.
- [8] G. Bin Huang, L. Chen, and C. K. Siew, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, Jul. 2006, doi: 10.1109/TNN.2006.875977.
- [9] Q. Y. Zhu, A. K. Qin, P. N. Suganthan, and G. Bin Huang, “Evolutionary extreme learning machine,” *Pattern Recognit.*, vol. 38, no. 10, pp. 1759–1763, Oct. 2005, doi: 10.1016/j.patcog.2005.03.028.
- [10] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, “OP-ELM: Optimally pruned extreme learning machine,” *IEEE Trans. Neural Networks*, vol. 21, no. 1, pp. 158–162, Jan. 2010, doi: 10.1109/TNN.2009.2036259.
- [11] B. Subudhi and D. Jena, “Differential evolution and levenberg marquardt trained neural network scheme for nonlinear system identification,” *Neural Process. Lett.*, 2008, doi: 10.1007/s11063-008-9077-x.
- [12] J. Cao, Z. Lin, and G. Bin Huang, “Self-adaptive evolutionary extreme learning machine,” *Neural Process. Lett.*, vol. 36, no. 3, pp. 285–305, Dec. 2012, doi: 10.1007/s11063-012-9236-y.
- [13] N. Zeng, H. Zhang, W. Liu, J. Liang, and F. E. Alsaadi, “A switching delayed PSO

- optimized extreme learning machine for short-term load forecasting,” *Neurocomputing*, 2017, doi: 10.1016/j.neucom.2017.01.090.
- [14] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *IEEE International Conference on Neural Networks - Conference Proceedings*, 1995, vol. 4, pp. 1942–1948.
- [15] R. Hassan, B. Cohanım, O. De Weck, and G. Venter, “A comparison of particle swarm optimization and the genetic algorithm,” in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2005, doi: 10.2514/6.2005-1897.
- [16] R. Cheng and Y. Jin, “A competitive swarm optimizer for large scale optimization,” *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015, doi: 10.1109/TCYB.2014.2322602.
- [17] S. Gu, R. Cheng, and Y. Jin, “Feature selection for high-dimensional classification using a competitive swarm optimizer,” *Soft Comput.*, vol. 22, no. 3, pp. 811–822, Feb. 2018, doi: 10.1007/s00500-016-2385-6.
- [18] Quinlan R, 1987, Thyroid Disease Data Set [online].
<https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease> [Ziyaret Tarihi: 21 Mayıs 2020].
- [19] T. Matias, R. Araújo, C. H. Antunes, and D. Gabriel, “Genetically optimized extreme learning machine,” in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2013, doi: 10.1109/ETFA.2013.6647975.
- [20] J. Sánchez-Monedero, C. Hervás-Martinez, P. A. Gutiérrez, M. C. Ruz, M. C. R. Moreno, and M. Cruz-Ramirez, “Evaluating the performance of evolutionary extreme learning machines by a combination of sensitivity and accuracy measures,” *Neural Netw. World*, vol. 20, no. 7, p. 899, 2010.
- [21] M. Eshtay, H. Faris, and N. Obeid, “Improving Extreme Learning Machine by Competitive Swarm Optimization and its application for medical diagnosis problems,” *Expert Syst. Appl.*, vol. 104, pp. 134–152, Aug. 2018, doi: 10.1016/j.eswa.2018.03.024.
- [22] G. A. Yüksel Yurtay and N. Z. Bacinoğlu, “Tıbbi Karar Destek Sistemlerinin

Yöntemsel Olarak Değerlendirilmesi Üzerine Bir Çalışma,” *Int. Symp. Innov. Technol. Eng. Sci.*

- [23] R. Berglund and S. Belciug, “Improving extreme learning machine performance using ant colony optimization feature selection. Application to automated medical diagnosis,” *Ann. Univ. Craiova, Math. Comput. Sci. Ser.*, 2018.
- [24] A. A. Abusnaina, S. Ahmad, R. Jarrar, and M. Mafarja, “Training neural networks using Salp Swarm Algorithm for pattern classification,” in *ACM International Conference Proceeding Series*, 2018, doi: 10.1145/3231053.3231070.