

Yapay Sinir Ağları Eğitiminde Kullanılan Optimizasyon Yöntemlerinin İncelenmesi ve Kan Nakli Hizmet Merkezi Veri Seti Üzerinden Değerlendirilmesi

Mustafa Kaytan^{1*}, Cemaleddin Yeroğlu², İbrahim Berkan Aydilek³

^{1*}Harran Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Bilgisayar Programcılığı Programı, Şanlıurfa, Türkiye (mkaytan@harran.edu.tr)

²İnönü Üniversitesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye (c.yeroglu@inonu.edu.tr)

³Harran Üniversitesi, Bilgisayar Mühendisliği Bölümü, Şanlıurfa, Türkiye (berkanaydilek@harran.edu.tr)

Received Date : May 12, 2020.

Acceptance Date : Nov. 07, 2020.

Published Date : Dec. 1, 2020

Özetçe— Yapay Sinir Ağları (YSA) makine öğrenmesi gibi birçok uygulama alanında yaygın şekilde kullanılan bir yöntemdir. YSA'da ağı eğitilmesi için ağırlık değerlerinin güncellenmesi ve optimize edilmesi gerekmektedir. Ağırlık değerlerinin optimizasyonunda ağı başarımı yani öğrenme başarısı kullanılan eğitim yöntemi ve öğrenme sürecine bağlıdır. Bu çalışmanın amacı biyolojik sinir ağlarının çalışma şekline göre oluşturulan YSA'ların farklı teknikler ile nasıl optimize edildiğini araştırmaktır. Bununla ilgili olarak çeşitli çalışmalar incelenmiştir. Elde edilen sonuçlar, YSA'ların metasezgisel, karma ve özel yöntemlerle optimize edilebildiğini göstermiştir. Bu alanda disiplinler arası çalışmaların da yapıldığı görülmüştür. Bu çalışma kapsamında sinir ağlarını optimize etmek için kullanılan yöntemlerden birisi olan istatistiksel veya deneysel yöntemler uygulanmıştır. Bunun için sinir ağı parametrelerinden batch boyutu, devir sayısı, öğrenme oranı, momentum, ağırlık başlatma, nöron aktivasyon fonksiyonu, gizli katmandaki nöron sayısı kullanılarak 5 farklı deney yapılmıştır. Yapılan deneylerin sonuçlarında sinir ağı başarımının arttığı görülmüştür.

Anahtar kelimeler: YSA, sinir ağı parametreleri, ağı optimizasyonu.

Analysis of Optimization Methods Used in Artificial Neural Networks Training and Evaluation on Blood Transfusion Service Center Data Set

Abstract— Artificial Neural Networks (ANN) are widely used method in many applications such as in machine learning. In order to train the network of ANN, the weight values need to be updated and optimized. The success of the network in the optimization of weight values and learning success depends on the training method and learning process that is used. The aim of this study is to investigate the optimization techniques of the ANNs, which is generated by the simulation of biological neural

networks. For this purpose, various studies were investigated. The results showed that ANNs have been optimized by metaheuristic, hybrid and special optimization methods. Interdisciplinary studies in this area have also been observed. In this study, statistical or experimental methods which are one of the methods used to optimize neural networks have been applied. For this purpose, 5 different experiments were carried out using batch size, number of epochs, learning rate, momentum, weight initialization, neuron activation function and number of neurons in hidden layer. The results of the experiments showed that neural network performance has been increased.

Keywords : ANN, neural network parameters, network optimization.

1. GİRİŞ

Yapay Sinir Ağı (YSA) bir biyolojik sinir ağı modelidir. Düğüm veya nöron olarak adlandırılan basit işlem birimlerinin ara bağlantılarından oluşur. Giriş katmanı ve çıkış katmanı arasında bir veya daha fazla gizli katman bulunabilir. İki düğüm arasındaki her bağlantı belirli bir ağırlığa sahiptir. YSA, girişleri ve beklenen çıktıları içeren mevcut örnekleri kullanarak eğitilerek öğretilir. Bu durum YSA'nın öğrenme yeteneği ve doğrusal olmama yöntemi sayesinde gerçekleşir [1]. Ortalama web sayfalarının sınıflandırılması [2], veri tabanlarında eksik değerleri tahmin etme [3] gibi çeşitli çalışmalarda ve büyük bir başarı oranıyla farklı uygulamalarda da kullanılmaktadırlar [4]. YSA'lar insanlar gibi örneklerden öğrenmektedirler. Doğal genelleme yeteneğine sahiptirler. Bu şekilde eğitildikleriyle aynı olmayan kalıpları belirleyebilirler [5].

Bu çalışmanın amacı YSA'ların başarımının iyileştirilmesi için kullanılan yöntemlerin ve yaklaşımların bir sınıflandırmasını yapmaktır. Bunun için geniş bir literatür taraması yapılmıştır. Bu çalışmanın hacmini oldukça aşan bir konu olduğu için bu sınıflandırmalar olabildiğince kısa ve öz bir şekilde ele alınmıştır. Ayrıca bu çalışma kapsamında istatistiksel veya deneysel yöntemler ile ilgili olarak sinir ağı optimizasyonu için çeşitli ağ parametreleri kullanılarak 5 deney yapılmıştır.

Çalışmanın geri kalanı şu şekilde düzenlenmiştir. Bölüm 2'de YSA ile ilgili yapılmış inceleme çalışmaları özetlenmiştir. Bölüm 3'te YSA performansının iyileştirilmesi için yapılan araştırmaların sınıflandırması yapılmıştır. Bölüm 4'te YSA performansının iyileştirilmesi için kullanılan optimizasyon yöntemleri anlatılmıştır. Bölüm 5'te uygulama ve vaka çalışması sunulmuştur. Bölüm 6'da ise sonuç ve öneriler verilmiştir.

2. YSA İLE İLGİLİ YAPILMIŞ İNCELEME ÇALIŞMALARI

YSA'lar çok farklı uygulamalarda büyük başarı oranlarıyla kullanım alanı bulduğundan performansını iyileştirmek için çok sayıda çalışma yapılmaktadır. Literatürde bu çalışmaları sınıflandıran ve çeşitli özelliklerini inceleyen oldukça fazla inceleme çalışmaları yer almaktadır. Örneğin:

- Sunulan bir inceleme çalışması [6] ile Evolved Plastic Artificial Neural Network (EPANN) yapılarının alanını tanımlayan çeşitli fikirler bir araya getirilmiş, ana yöntemler ve sonuçlar gözden geçirilmiş ve olası gelişmeler tanıtılmıştır.
- İnceleme [7] ile derin sinir ağlarının (Deep Neural Network (DNN)) analizi için en son geliştirilen yöntemlerin biyolojik sinir ağlarındaki temsilleri anlamadaki faydaları incelenmiştir. DNN'lerin sinirbilim için silico model sistemlerinde kullanımı ve biyolojik sinir ağlarının çalışma prensipleri hakkında çeşitli araştırmalar yapılmıştır.
- Yapılan bir inceleme çalışmasında [8] çeşitli sinir ağı uygulamaları araştırılmıştır. YSA'nın bir çeşit sınıflandırması yapıp çeşitli trendler ile ilgili bilgiler verilmiştir. YSA'nın uygulamadaki zorlukları, sunduğu katkılar, başarımlar karşılaştırmaları sunulmuştur. Bilgisayar, bilim, mühendislik, tıp, çevre, tarım, madencilik, teknoloji, iklim, işletme, sanat ve nanoteknoloji gibi çeşitli disiplinlerdeki çeşitli YSA uygulamaları incelenmiştir. İleri ve geri yayılım YSA gibi modellerin

problemlere uygulanmasındaki yüksek başarımları araştırılmıştır. Doğruluk, işlem hızı, gecikme, hata toleransı, hacim, ölçeklenebilirlik, yakınsama ve başarımlar gibi veri analizi faktörlerine dayanan ileri ve geri yayılım YSA modelleri önerilmiştir. Sadece bir yöntemin uygulanması yerine ağ yapısında uygulanan yöntemlerin birleştirilmesi önerilmiştir.

- İnceleme [9] ile derin iğnecikli sinir ağlarını (Spiking Neural Network (SNN)) eğitmek için denetimli ve denetimsiz yöntemler doğruluk ve hesaplama maliyeti açısından karşılaştırılmıştır. SNN'lerin doğruluk açısından YSA'nın gerisinde kalmasına rağmen aradaki farkın azaldığı görülmüştür. SNN'lerin genel olarak daha az işlem gerektirdiği ve mekansal-zamansal verileri işlemede daha iyi adaylar olduğu görülmüştür.
- Sunulan bir derleme çalışması [10] ile YSA'larda öğrenme için istatistiksel çıkarım uygulamaları gözden geçirilmiştir. Geleneksel istatistiksel yaklaşımlara veya Bayes yaklaşımına dayanan model-seçim yöntemleriyle aşırı uydurma problemine yönelik çözümler tartışılmıştır. YSA'lar için denetimli ve denetimsiz öğrenme algoritmalarının kullanımı gözden geçirilmiştir.
- İnceleme [11] ile derin denetimli öğrenme, denetimsiz öğrenme, takviyeli öğrenme ve evrimsel hesaplama, derin ve geniş ağları kodlayan kısa programlar dolaylı olarak araştırılmıştır.
- Yapılan bir derleme çalışmasında [12] ileri beslemeli sinir ağları karşılaştırılmış, çeşitli uygulama alanlarında tahmin ve sınıflandırma problemleri için kullanılan istatistiksel yöntemler araştırılmış, sinir ağlarının yetenekleri ve farklı uygulamalarda kullanılan istatistiksel yöntemler hakkında bilgiler verilmiştir.
- İnceleme [13] ile beyindeki sinirsel devrelerin YSA'lar tarafından kullanılan hata geri yayılım algoritmasına nasıl yaklaşabileceğine dair yakın zamanda önerilen teoriler özetlenmiştir.
- Derleme çalışması [14] ile sinir ağlarının öğrenmesine yönelik Bayesian yaklaşımı üzerine bir derleme sunulmuştur. Yaklaşımın avantajları üç uygulama üzerinde gösterilmiştir. Yaklaşım, Bayesian modellerde ve klasik hata azaltma yaklaşımlarında araştırılmıştır. Regresyon, sınıflandırma ve inverse problem olmak üzere üç vaka problemi incelenmiştir.
- İnceleme [15] ile tekrarlayan sinir ağlarında (Recurrent Neural Network (RNN)) kullanılan bazı terimler için çeşitli tanımlar sunulmuş, bu tanımlara dayanarak olası tüm RNN mimarileri sınıflandırılmış ve artan karmaşıklık derecesinde dört genel sinir ağı mimarisi tanımlanmıştır. Mevcut RNN mimarilerinin genel RNN mimarisine nasıl dönüştürülebileceği gösterilmiştir. RNN mimarileri ile ilgili bazı konular tartışılmıştır.

Görülebileceği gibi YSA'ların farklı özelliklerini ele alan incelemeler bu konunun önemini ortaya koymaktadır. Bu yayındaki verilen inceleme ve uygulama çalışmasının ise YSA'da kullanılan optimizasyon teknikleri konusuna katkı sağlayacağı söylenebilir.

3. YSA PERFORMANSININ İYİLEŞTİRİLMESİ İÇİN YAPILAN ARAŞTIRMALARIN SINIFLANDIRILMASI

YSA performansının iyileştirilmesi ve bazı problemlerinin giderilmesi için literatürde yapılan çalışmalar şöyle sınıflandırılabilir:

3.1. Veri Problemleri

Genel olarak farklı veri biçimlerinin sinir ağına sunulmadan bazı ön işlemlerden geçirilmesi gerekmektedir. Oluşabilecek temel veri problemleri şöyle sıralanabilir:

Boyutlu Veri: Gerçek dünyadaki uygulamalarda verilerin çok fazla olması ile ilgili bir sorundur. Bu sorunu çözmek için yüksek boyutlu verilerin boyutlarını azaltan sinir ağı modelleri kullanılabilir [16].

Büyük Veri: Günümüzde veri üretiminin katlanarak büyümesi [17] ve çok yüksek miktarda veri alışverişinin gerçekleşmesi ile ilgili bir sorundur. Büyük veri analizi için paralel hesaplama yapan gerçek zamanlı öğrenmeye dayalı sinir ağları ile verimli ve hızlı öğrenme sağlanabilir.

Eksik Veri: Verilerin bazı özelliklerinin veya bazı değişkenlerinin bilinmemesinden kaynaklanan bir sorundur. Bu sorunu çözmek için bilinenle bilinmeyen veriler arasındaki ilişki keşfedilmeye ve eksik değerlere yaklaşılmaya çalışılır. Bunun için sinir ağlarına dayalı [18] veya Multiple Kernel Learning [19] gibi bazı yöntemler kullanılabilir.

Dengesiz Veri: Öğrenme aşamasında görülen bir sorundur. Bir sınıflandırmada bazı sınıfların diğerlerinden daha çok eleman bulundurduğu bir sorundur [20]. Bu sorunu çözmek için sınıflandırıcı düzeyinde veya veri düzeyinde bazı yöntemler kullanılır [21].

Sınırlı Veri: Sınırlı sayıdaki verinin varlığında görülen bir sorundur. Bu sorunu çözmek için veri kümesi eğitim kümesi ve test kümesi şeklinde bölünüp çapraz doğrulama yapılabilir. Ağın değerlendirilmesi ve eğitimi için ağ elemanlarının sayısı ve büyüklüğü değiştirilebilir [22], [23].

3.2. Özel Donanımlar ve Simülasyonlar

Simülasyon veya fiziksel uygulamalar geleneksel bilgisayarlar kullanılarak yapılabileceği gibi bazı özel donanımlar kullanılarak da gerçekleştirilebilir [24]. Beynin nasıl çalıştığını anlamak için biyolojik sinir sistemi üzerine yapılan bazı deneysel çalışmaların büyük ölçekli beyin modelleri sayısal simülasyon ile birleştirilebilir [25].

3.3. Günümüzdeki Bazı Yapı Akımları

Günümüzde yeni yapıların ve çeşitli disiplinlerin geliştirilmesi nedeniyle sinir ağları izole bir alan olarak görülmemektedir. Bu bölümde sinir ağlarının önemli bir yer aldığı bazı önemli yapılara kısaca değinilecektir.

Nöroenformatik [26], sinirbilim alanında üretilen verileri işlemek için gerekli bilgisayar modelleri, veri tabanları ve yazılım araçlarının geliştirilmesi ile ilgili bir alandır.

Hesaplama Zeka geleneksel yaklaşımlarla çözilemeyen bilgi işlem sorunlarını çözmek için biyolojik yapıları referans alan yöntemleri kapsar. Bu yöntemlerin makine öğrenmesinde başarıyla uygulandığı görülmektedir [27].

Doğal Hesaplama, yeni problem çözme tekniklerinin geliştirilmesi için doğadan ilham alan, doğal olayları sentezlemek için bilgisayar kullanan ve hesaplama için doğal malzemeler kullanan yöntemleri kapsar [28].

Sinir Mühendisliği sinir sistemlerinin özelliklerini daha iyi öğrenmek ve kullanmak için nöronlar, sinir ağları ve sinir sisteminin arasındaki ilişkilerden elde edilen bilgileri uygulamaktadır [29].

Nöromorfik Hesaplama sistemleri enerji tasarrufuna sahip gerçek zamanlı yetenekli cihazları gerçekleştirmek amacıyla beyin onarımı için yeni nesil sistemlerin çekirdeği olarak öngörülmektedir [30].

4. YSA PERFORMANSININ İYİLEŞTİRİLMESİ İÇİN KULLANILAN OPTİMİZASYON YÖNTEMLERİ

YSA modellerinin, yapılarının ve algoritmalarının üretimi, gelişimi ve olgunlaşması dört döneme ayrılabilir. 1940 ile 1960 arası dönemde algılayıcı (perceptron) gibi tekil nöron modelleri ve öğrenme kuralları [31] önerilmiştir. 1960 ile 1980 arası dönemde tek katmanlı ağlar için öğrenme kuralları geliştirilmiş, tekrarlayan ağlar için çeşitli yöntemler uygulanmış ve Self-Organizing Maps (SOM) ile

İlgili ilk çalışma sunulmuştur [32]. 1980 ile 2000 arası dönemde çok katmanlı sinir ağları için öğrenme kuralları geliştirilmiş, SOM yeniden incelenmiş [33], [34], Bayes yöntemleri ile Gauss süreçleri uygulanmış ve destek vektör makineleri ortaya çıkmıştır. 2000'den günümüze kadar olan dönemde senkronizasyon denetimi, durum tahmini, kararlılık, istatistiksel denge ve yakınsama analizi gibi konularda araştırmalar yapılmıştır.

Bir veri modelinin parametrelerinin tahmin edilmesi öğrenme olarak adlandırılabilir. Öğrenmede aynı sinir ağı modeli farklı görevler için kullanılabilir. Öğrenmede sinir ağı evrensel yönlendiriciler sınıfı gibi ele alınabilir [35]. Öğrenmede ağ doğrusal olmayan girdi-çıkış eşleşmesi yapabilir. Denetimli öğrenmede girdi çıktı çiftleri etiketli olan bir veri setinde sinir ağına verilen girdilerden çıktılara bir eşleme yapılır. Bu şekilde parametreler ayarlanarak maliyet fonksiyonu minimum yapılır [36]. Denetimsiz öğrenmede verilen bir girdiye karşılık bir çıktı beklentisiyle yeni bilgiler elde edilir. Bu şekilde daha iyi ortak olasılık yoğunluk tahmin edicileri elde edilir [37]. Takviyeli öğrenmede giriş çıkış eşleşmesi için ödül veya ceza sinyalleri kullanılır ve modelin çevresiyle sürekli etkileşimi kurulur. Bu şekilde skaler performans endeksi minimum yapılır [38].

YSA eğitim algoritmalarının çoğunda gradyan tabanlı arama yöntemleri kullanılmaktadır. Ağırlıklar her zaman hatayı en aza indirecek şekilde güncellenmektedir. Bu işlem YSA öğrenme süreci olarak adlandırılır. Bu algoritmaların bazı olumsuz yönleri bulunmaktadır. Örnek olarak; karmaşık ve çok modlu hata alanı, eklenen her ekstra (gizli) katman için büyüklük sırasına göre yavaşlama, ağ felci, öğrenme oranı parametresine bağımlılık verilebilir. Bu tür olumsuzlukların sebebi söz konusu algoritmaların yerel minimumda tutulmasından, tamamen başlangıç durumuna bağımlı kılınmasından ve çeşitli ayarlardan kaynaklanabilmektedir [39].

Metasezgisel global arama yöntemleri ile algoritmaların başarımının ikincil zirve noktasına sıkışıp kalmaması sağlanabilir. Bu şekilde sinir ağı ile eğitim sorununa sağlam ve etkili bir çözüm sunulabilir [40]. Farklı sinir ağları için kullanılabilirler [41].

4.1. YSA'ları Optimize Etmek İçin Kullanılan Yaklaşımlar

Bu kısımda sinir ağlarını optimize etmek için kullanılan yaklaşımlar dört kategori altında sınıflandırılabilir.

Evrimsel Yöntemler: Genetik operatörlerin kullanılması gibi farklı mimarilerin bir amaç fonksiyonu göz önüne alınarak gizli katmanların ve gizli nöronların sayısının değiştirilerek topolojide arama yapan evrimsel yöntemler [42], [43].

Yapıcı ve/veya Budama Algoritmaları: YSA performansının yapılan değişikliklerden hangi şekilde etkilendiğinin gösterilmesi için önceden tanımlanan bir ölçüt yardımıyla nöronları başlangıç mimarisine ekleyen ve/veya başlangıç mimarisinden kaldıran yapıcı ve/veya budama algoritmaları [44], [45].

Karma Yöntemler: YSA'nın gerçek sayılar yerine bulanık sayılar üzerinde çalışabilen veya YSA'nın uyumlu bir bulanık sistem şeklinde düşünülebileceği bulanık çıkarım gibi karma yöntemler [46].

İstatistiksel veya Deneysel Yöntemler: Bir YSA'nın parametrelerinin davranışlarını incelemek ve önerilen modelin performansına dayanarak parametreler için uygun değerleri seçmekte kullanılan istatistiksel veya deneysel yöntemler [47], [48].

4.2. İlgili Yaklaşımları Kullanan Bazı Çalışmalar

Bu bölümde verilen yaklaşımları ve sınıflandırmaları kullanan literatürde oldukça fazla çalışma yer almaktadır. Bu çalışmalardan önemli görülen bazıları şöyle sıralanabilir:

- Nöroformatik ile ilgili bir çalışmada [49] sinirbilim ve sinir hesaplama için bir dizi teori ve matematiksel model sunulmuştur. Sinir kümelerinin ve sinir sistemlerinin nörolojik temelleri incelenmiştir. Sinirsel sinyal üretimi, darbe frekans modülasyonu ve sinirsel sinyal iletimleri için

nörolojik modellerin ve sinirsel sinyal teorilerinin matematiksel bir işleyişi tanımlanmıştır. Uygulanan YSA'ların nörolojik temelini açıklayan biçimsel ilişki modelleri sunulmuştur. Nöroenformatik teorisi ve biçimsel sinir modellerinin mühendislik uygulamaları; bilişsel hesaplama, sinir hesaplama ve sinir ağı analizleri araştırılmıştır.

- Hesaplamalı zeka ile ilgili bir çalışmada [50] verimli ağ saldırı tespiti amacıyla çeşitli ağ trafiği bağlantıları tanımlanıp sınıflandırılmıştır. Bu amaçla hesaplamalı zekanın farklı yöntemlerinin birleşimi olan sinir ağları, bağışıklık sistemleri, sinirsel bulanık sınıflandırıcılar ve destek vektör makineleri kullanılmıştır. Giriş vektörlerinin işlem hızını arttırmak için temel bileşenler yöntemi uygulaması önerilmiştir. Önerilen yaklaşım ile ağ trafiğinin çok seviyeli bir analizi, imza tabanlı bir yöntemle saldırıları tespiti ve hesaplamalı zeka yöntemlerine dayalı bir dizi uyarlanabilir detektörün birleştirilmesi sağlanmıştır. Önerilen mekanizmalara dayalı bir yazılım aracı sunulmuştur. Bilinen ve bilinmeyen saldırıları tespitiyle ilgili hesaplamalı deneyler yapılmıştır.
- Doğal hesaplama ile ilgili yapılan bir çalışmada [51] nöral ağlar, bulanık kümeler ve kaba kümeler gibi bilgi işlem modelleri için doğal hesaplama yöntemleri incelenmiştir. Biyolojik motivasyon, tasarım ilkeleri, uygulama alanları, açık araştırma problemleri ve bu modellerin zorlayıcı konuları araştırılmıştır.
- Sinir mühendisliği alanında yapılan bir çalışmada [52] yeni ve gerçek zamanlı bir YSA modeli önerilmiştir. Önerilen model dinamik bir sinir ağına yerleştirilmiştir. Sunulan model ile ağın gürültülü girdilerde öğrenmesi ve karmaşık, dağınık ve gürültülü arka planda bilinen 2 boyutlu sinirsel aktivite desenlerini tanınması sağlanmıştır. Önerilen model ile güçlü otomatik hedef tanıma ve nöro-hesaplamalı görme sistemlerinin tasarımı için temel sinirsel tasarım ilkeleri ve sinir mühendisliği temelleri sunulmuştur.
- Nöromorfik hesaplama modeli için yapılan bir çalışmada [53] bir yazılım ekosistemi tanımlanmıştır. Tanımlanan ekosistem; simülatör, komutan, evrimsel iyileştirici ve görselleştirici olmak üzere dört parçadan oluşurulmuştur. Önerilen yazılım ekosistem ile uygulama geliştirme kolaylaştırılmış ve daha etkili bir nöromorfik hesaplama modeline dönüşmesi sağlanmıştır. MNIST veri seti kullanılarak basamak sınıflandırması için yazılım bileşenlerinin kullanımı açıklanmıştır.
- Evrimsel yöntemler kullanılarak gerçekleştirilen bir çalışmada [54] genetik algoritmalara dayalı ileri beslemeli ağ modelleri için bir model seçim metodolojisi önerilmiştir. İleri beslemeli ağlar için model seçim literatürüne çeşitli katkılar sağlanmıştır. Bir ileri beslemeli ağ modelinin çıktısı olarak bir fonksiyonun global optimumunu arayabilen bir genetik algoritma oluşturulmuştur. Genetik algoritmanın girdilerin türünü, gizli birimlerin sayısını ve girdiler ile çıktı katmanları arasındaki bağlantı yapısını geliştirmesi sağlanmıştır. Seçim operatörü olarak adlandırılan yerel bir elitist prosedürün uygulanmasının algoritmanın başarımına etkisi incelenmiştir. Çalışılan genetik algoritmanın global yaklaşım özelliklerinin hassasiyetini incelemek için bir Monte Carlo simülasyonu uygulanmıştır. Önerilen metodoloji ile ilgili bir uygulama yapılmıştır.
- Yapıcı ve/veya budama algoritmaları ile ilgili olarak hazırlanan bir çalışmada [55] çoklu gizli katmanlı ileri beslemeli sinir ağı oluşturmak için bir şema önerilmiştir. Önerilen şema ile yeni gizli birimlerin ve yeni gizli katmanların ihtiyaç duyulduğunda modele eklenmesi sağlanmıştır. Regresyon problemlerine uygulanan simülasyon sonuçları ile önerilen şemanın başarımı gösterilmiştir.
- Karma yöntemler kullanan bir çalışmada [56] bulanık IF-THEN kuralları ile temsil edilen uzman bilgisini ve girdiler olarak sayısal verileri kullanan sınıflandırıcılar için yeni bir öğrenme algoritması sunulmuştur. Yapılan çalışma kapsamında gerçekleştirilen simülasyon sonuçları ile önerilen modelin etkinliği gösterilmiştir.

- İstatistiksel veya deneysel yöntemler ile ilgili bir çalışmada [57] belirli bir vaka çalışması için ağ başarımı üzerinde geometri ve parametrelerin etkisi açıklanmıştır. Araştırmada yapılan testlerden elde edilen bilgilerin dikkate alınan probleme özgü olduğu belirtilmiştir. Elde edilen sonuçların öğrenme hızı, momentum, transfer fonksiyonunun kazancı, devir sayısı ve ağ geometrisinin eğitim hızı üzerinde önemli bir etkiye sahip olduğu ifade edilmiştir. Kullanılan transfer ve hata fonksiyonunun genelleştirme kabiliyetinin yanı sıra öğrenme hızı üzerinde de önemli bir etkisi olduğu bulunmuştur.

5. UYGULAMA VE VAKA ÇALIŞMASI

Bu kısımda YSA optimizasyonu için istatistiksel veya deneysel yöntemler ile ilgili bir uygulama yapılmıştır. Bu çalışmada UCI makine öğrenmesi veri setlerinden [58] olan Kan Nakli Hizmet Merkezi Veri Seti (Blood Transfusion Service Center Data Set) [59] kullanılmıştır. Veri setinde 748 örnek, giriş değerlerine ait 4 öznitelik ve çıkış değerine ait 1 öznitelik olmak üzere toplam 5 öznitelik bulunmaktadır. Veri seti ile ilgili olarak yapılan çalışmada [60] RFMTC pazarlama modelinin gösterilmesi için Tayvan'daki Hsin-Chu Şehrindeki Kan Nakli Hizmet Merkezi'nin donör veritabanı kullanılmıştır. Merkez, kan nakli servisi otobüsünü yaklaşık üç ayda bir bağışlanan kanı toplamak için Hsin-Chu şehrindeki bir üniversiteye göndermektedir. Modeli oluşturmak için donör veritabanından rastgele 748 bağışçı seçilmiştir.

5.1. Öznitelik Bilgileri

Tablo 1'de veri setindeki öznitelik bilgileri belirtilmiştir. Veri setinde 4 giriş ve 1 çıkış bulunmaktadır.

Tablo 1. Öznitelik bilgileri.

Giriş	Çıkış
R (Recency): son bağıştan bu yana geçen aylar F (Frequency): toplam bağış sayısı M (Monetary): c.c.'de toplam kan bağışı T (Time): ilk bağıştan bu yana geçen aylar	Mart 2007'de kan bağışı yapıp yapmadığını gösteren ikili bir değişken 1: kan bağışlandı 0: kan bağışlanmadı

5.2. Veri Seti İlk 10 Örnek

Tablo 2'de 4 giriş, 1 çıkış ve 748 örnekten oluşan veri setinden ilk 10 örnek gösterilmiştir.

Tablo 2. Veri seti için ilk 10 örnek.

Giriş				Çıkış
Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)	Donated blood
2	50	12500	98	1
0	13	3250	28	1
1	16	4000	35	1
2	20	5000	45	1
1	24	6000	77	0
4	4	1000	4	0
2	7	1750	14	1
1	12	3000	35	0
2	9	2250	22	1
5	46	11500	98	1

5.3. Çalışma Ortamı

Bu çalışmada donanım ortamı olarak Intel Core i5-3470 CPU @ 3.20GHz, 3201 Mhz işlemcili, 4 GB belleğe sahip bir makine kullanılmıştır. Yazılım ortamı olarak ise Keras 2.3.0, Matplotlib 3.1.1, Numpy 1.17.2, Pandas 0.25.1, Python 3.7.3, Scipy 1.3.1, Sklearn 0.21.3 kullanılmıştır.

5.4. Çalışılan Parametreler

Bu uygulamada sinir ağının aşağıda sıralanan parametrelerinin davranışları incelenmiş ve önerilen modelin performansına dayanarak parametreler için uygun değerler seçilmiştir. Bu parametrelerden bazıları birlikte ele alınarak 5 farklı deney yapılmıştır. Çalışılan parametreler şunlardır:

- batch boyutu ve devir sayısı
- öğrenme oranı ve momentum
- ağırlık başlatma
- nöron aktivasyon fonksiyonu
- gizli katmandaki nöron sayısı

5.5. Deneysel Çalışma

Pratime optimizasyonu sinir ağlarının eğitiminin büyük bir kısmını oluşturmaktadır. Sinir ağlarının yapılandırılması ve ayarlanması gereken birçok parametresi vardır. Modellerin eğitilmesi için uzun süreler gerekebilmektedir. Deneylerde keras derin öğrenme modellerinin parametreleri ayarlamak için scikit-learn python makine öğrenme kütüphanesindeki grid search özelliği kullanılmıştır. Tüm deneylerde 5-katlı çapraz geçerlilik testi uygulanmıştır. Yapılan deneylere ait ortalama başarımlar sonuçları, en iyi başarımlar sonucu ve kullanılan parametreler aşağıdaki tablolarda gösterilmiştir.

Deney 1: batch boyutu ve devir sayısı: Tablo 3'te batch boyutu ve devir sayısı deneyi için oluşturulan ağ modeli parametreleri belirtilmiştir. Batch boyutu yinelemeli eğim iniş (iterative gradient descent)'te ağırlıklar güncellenmeden önce ağa gösterilen desen sayısıdır. Ağın eğitiminde kullanılan bir iyileştirme algoritmasıdır. Bir döngüde okunan desen sayısını bellekte saklamayı tanımlar. Devir sayısı eğitim veri setinin ağa gösterilme sayısıdır.

Tablo 3. 1. Deney için oluşturulan ağ modeli.

Giriş katmanı nöron sayısı	4
Gizli katman nöron sayısı	8
Çıkış katmanı nöron sayısı	1
Gizli katman aktivasyon fonksiyonu	relu
Çıkış katmanı aktivasyon fonksiyonu	sigmoid
Loss fonksiyonu	binary_crossentropy
Optimizasyon algoritması	adam
Seed (rastgele sayı üretici)	5
Grid search parametreleri	batch_size = [5, 10, 15] epochs = [25, 75, 100]

Tablo 4'te batch boyutu 10 ve devir sayısı 100 parametre değerleriyle en iyi sonuç 0.767380 olarak gözlenmiştir.

Deney 2: öğrenme oranı ve momentum: Tablo 5'te öğrenme oranı ve momentum deneyi için oluşturulan ağ modeli parametreleri belirtilmiştir. Öğrenme oranı her batch'in sonunda ağırlık güncelleme miktarını denetler. Momentum önceki ağırlık güncellemesinin mevcut ağırlık güncellemesini hangi derecede etkileyeceğini denetler.

Tablo 4. 1. Deney sonucu.

EN İYİ SONUÇ: 0.767380
PARAMETRE: {'batch_size': 10, 'epochs': 100}
ortalama: parametre:
0.752674 {'batch_size': 5, 'epochs': 25}
0.766043 {'batch_size': 5, 'epochs': 75}
0.647059 {'batch_size': 5, 'epochs': 100}
0.693850 {'batch_size': 10, 'epochs': 25}
0.762032 {'batch_size': 10, 'epochs': 75}
0.767380 {'batch_size': 10, 'epochs': 100}
0.750000 {'batch_size': 15, 'epochs': 25}
0.695187 {'batch_size': 15, 'epochs': 75}
0.750000 {'batch_size': 15, 'epochs': 100}

Tablo 5. 2. Deney için oluşturulan ağ modeli.

Öğrenme oranı	0.02
Momentum	0.0
Giriş katmanı nöron sayısı	4
Gizli katman nöron sayısı	8
Çıkış katmanı nöron sayısı	1
Gizli katman aktivasyon fonksiyonu	relu
Çıkış katmanı aktivasyon fonksiyonu	sigmoid
Loss fonksiyonu	binary_crossentropy
Optimizasyon algoritması	SGD
Epochs	75
Batch_size	5
Seed (rastgele sayı üretici)	5
Grid search parametreleri	learn_rate = [0.000001, 0.00001, 0.0001] momentum = [0.001, 0.01, 0.1]

Tablo 6'da öğrenme oranı 0.0001 ve momentum 0.001 parametre değerleriyle en iyi sonuç 0.762032 olarak gözlenmiştir.

Tablo 6. 2. Deney sonucu.

EN İYİ SONUÇ: 0.762032
PARAMETRE: {'learn_rate': 0.0001, 'momentum': 0.001}
ortalama: parametre:
0.620321 {'learn_rate': 1e-06, 'momentum': 0.001}
0.744652 {'learn_rate': 1e-06, 'momentum': 0.01}
0.577540 {'learn_rate': 1e-06, 'momentum': 0.1}
0.683155 {'learn_rate': 1e-05, 'momentum': 0.001}
0.758021 {'learn_rate': 1e-05, 'momentum': 0.01}
0.594920 {'learn_rate': 1e-05, 'momentum': 0.1}
0.762032 {'learn_rate': 0.0001, 'momentum': 0.001}
0.747326 {'learn_rate': 0.0001, 'momentum': 0.01}
0.760695 {'learn_rate': 0.0001, 'momentum': 0.1}

Deney 3: ağırlık başlatma: Tablo 7'de ağırlık başlatma deneyi için oluşturulan ağ modeli parametreleri belirtilmiştir. Ağ ağırlığı başlatma seçiminin ayarlanması için mevcut tüm yöntemler uygulanmıştır. Her katmanda aynı ağırlık başlatma yöntemi kullanılmıştır. Genel olarak her katmandaki aktivasyon fonksiyonuna göre farklı ağırlık başlatma yöntemlerinin kullanılması daha iyi sonuç verebilir. Ancak bu deneyde gizli katman ve çıktı katmanı için aynı ağırlık başlatma kriteri kullanılmıştır. Gizli katman için relu aktivasyon fonksiyonu kullanılmıştır. Çıktı katmanı için ikili sınıflandırma olduğundan sigmoid aktivasyon fonksiyonu kullanılmıştır.

Tablo 7. 3. Deney için oluşturulan ağ modeli.

Giriş katmanı nöron sayısı	4
Gizli katman nöron sayısı	8
Çıkış katmanı nöron sayısı	1
Gizli katman ağırlık başlatma	kernel_initializer='uniform'
Çıkış katmanı ağırlık başlatma	kernel_initializer='uniform'
Gizli katman aktivasyon fonksiyonu	relu
Çıkış katmanı aktivasyon fonksiyonu	sigmoid
Loss fonksiyonu	binary_crossentropy
Optimizasyon algoritması	adam
Epochs	75
Batch_size	5
Seed (rastgele sayı üretici)	5
Grid search parametreleri	init_mode = ['glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform', 'lecun_uniform', 'normal', 'uniform', 'zero']

Tablo 8’de ağırlık başlatma normal parametre değeriyle en iyi sonuç 0.768717 olarak gözlenmiştir.

Tablo 8. 3. Deney sonucu.

EN İYİ SONUÇ: 0.772841
PARAMETRE: {'init_mode': 'normal'}
ortalama: parametre:
0.536313 {'init_mode': 'glorot_normal'}
0.578353 {'init_mode': 'glorot_uniform'}
0.531230 {'init_mode': 'he_normal'}
0.770157 {'init_mode': 'he_uniform'}
0.516313 {'init_mode': 'lecun_uniform'}
0.772841 {'init_mode': 'normal'}
0.766139 {'init_mode': 'uniform'}
0.762103 {'init_mode': 'zero'}

Deney 4: nöron aktivasyon fonksiyonu: Tablo 9’da nöron aktivasyon fonksiyonu deneyi için oluşturulan ağ modeli parametreleri belirtilmiştir. Aktivasyon fonksiyonu nöronların ateşlenme zamanını ve doğrusal olmama niteliğini denetler. Bu deneyde Keras'ta bulunan aktivasyon fonksiyonları kullanılmıştır. Bu aktivasyon fonksiyonları sadece gizli katmanda kullanılmıştır. Veri seti ikili sınıflandırma problemini ele aldığından çıktı katmanında sigmoid aktivasyon fonksiyonu kullanılmıştır.

Tablo 9. 4. Deney için oluşturulan ağ modeli.

Giriş katmanı nöron sayısı	4
Gizli katman nöron sayısı	8
Çıkış katmanı nöron sayısı	1
Gizli katman ağırlık başlatma	kernel_initializer='uniform'
Çıkış katmanı ağırlık başlatma	kernel_initializer='uniform'
Gizli katman aktivasyon fonksiyonu	relu
Çıkış katmanı aktivasyon fonksiyonu	sigmoid
Loss fonksiyonu	binary_crossentropy
Optimizasyon algoritması	adam
Epochs	75
Batch_size	5
Seed (rastgele sayı üretici)	5
Grid search parametreleri	activation = ['hard_sigmoid', 'linear', 'relu', 'sigmoid', 'softmax', 'softplus', 'softsign', 'tanh']

Tablo 10’da nöron aktivasyon fonksiyonu relu parametre değeriyle en iyi sonuç 0.772727 olarak gözlenmiştir.

Tablo 10. 4. Deney sonucu.

EN İYİ SONUÇ: 0.772823
PARAMETRE: {'activation': 'linear'}
ortalama: parametre:
0.762103 {'activation': 'hard_sigmoid'}
0.772823 {'activation': 'linear'}
0.762103 {'activation': 'relu'}
0.762103 {'activation': 'sigmoid'}
0.762103 {'activation': 'softmax'}
0.763436 {'activation': 'softplus'}
0.762103 {'activation': 'softsign'}
0.762103 {'activation': 'tanh'}

Deney 5: gizli katmandaki nöron sayısı: Tablo 11’de gizli katmandaki nöron sayısı deneyi için oluşturulan ağ modeli parametreleri belirtilmiştir. Optimizasyon için bir katmandaki nöronların sayısı önemli bir parametredir. Genel olarak bir katmandaki nöronların sayısı ağıın kapasitesini etkiler. Bu deneyde gizli katmandaki nöron sayısı ayarlanmıştır. Daha büyük bir ağ daha fazla eğitim gerektirir. Nöron sayısı, batch boyutu ve devir sayısı ile optimize edilmiştir.

Tablo 11. 5. Deney için oluşturulan ağ modeli.

Giriş katmanı nöron sayısı	4
Gizli katman nöron sayısı	1
Gizli katman ağırlık sınırlandırma	kernel_constraint=maxnorm(2)
Dropout (seyreltme) katmanı oranı	0.5
Çıkış katmanı nöron sayısı	1
Gizli katman ağırlık başlatma	kernel_initializer='uniform'
Çıkış katmanı ağırlık başlatma	kernel_initializer='uniform'
Gizli katman aktivasyon fonksiyonu	linear
Çıkış katmanı aktivasyon fonksiyonu	sigmoid
Loss foksiyonu	binary_crossentropy
Optimizasyon algoritması	adam
Epochs	75
Batch_size	5
Seed (rastgele sayı üretici)	5
Grid search parametreleri	neurons = [10, 15, 20, 25, 30, 35, 40, 45, 50]

Tablo 12’de gizli katmandaki nöron sayısı 45 parametre değeriyle en iyi sonuç 0.779412 olarak gözlenmiştir.

Tablo 12. 5. Deney sonucu

EN İYİ SONUÇ: 0.779412
PARAMETRE: {'neurons': 45}
ortalama: parametre:
0.766043 {'neurons': 10}
0.764706 {'neurons': 15}
0.775401 {'neurons': 20}
0.745989 {'neurons': 25}
0.760695 {'neurons': 30}
0.771390 {'neurons': 35}
0.771390 {'neurons': 40}
0.779412 {'neurons': 45}
0.774064 {'neurons': 50}

6. SONUÇ VE ÖNERİLER

Bu çalışmada YSA optimizasyonu ile ilgili olarak bir literatür çalışması yapılmıştır. Yapılan çalışmalar incelendiğinde sinir ağlarının optimizasyonu ile ilgili çok sayıda yöntemin geliştirildiği görülmüştür. Sinir ağları ile ilgili bilim ve teknolojiye geliştirilen yeni donanımsal ve yazılımsal ürünlerle farklı optimizasyon yöntemleri denenebilmektedir. Geliştirilen her yöntemin diğer bir yöntemi tetikleyebilme potansiyeli bulunmaktadır. Disiplinler arası çalışmalarla da sinir ağları optimize edilebilmektedir. Buradaki can alıcı nokta sinir ağlarının belirli bir soruna uygulanmasından ziyade ağın hızını ve doğruluğunu en üst düzeye çıkarmaktır. Bu çalışma ile sinir ağları optimizasyonu ile ilgilenen araştırmacılara farklı bakış açılarıyla optimizasyon yöntemleri denemeleri amaçlanmıştır. Bu çalışma kapsamında ayrıca sinir ağlarının çeşitli parametreleri kullanılarak istatistiksel veya deneysel yöntemlere yönelik 5 deney yapılmıştır.

İncelenen literatüre dayanarak araştırmacılara gelecekte sinir ağları ile ilgili daha fazla araştırma ve geliştirme için bazı iyileştirme alanları önerilebilir. Sinir ağlarının performansı optimize edilirken model geliştirme sürecinde sistemli bir yaklaşıma ihtiyaç vardır. Bu nedenle sunulan bazı öneriler kısa maddeler halinde şu şekilde sıralanabilir:

- Verilerin uygun formata dönüştürülmesi
- Uygun girdi modelinin belirlenmesi
- Sinir ağı geometrisi seçimi
- Değişken belirleme sürecinde sinir ağı özelliği
- Ağırlık alanında farklı adımlarla sürekli eğitim
- Sunulan sinir ağı modelinin doğrulanması

7. Kaynaklar

- [1] Yang S, Chen Y. “An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications”. *Neurocomputing*, 86, 140-149, 2012.
- [2] Kaytan M, Hanbay D. “Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines”. *Journal of Computer Sciences*, 2(1), 15-36, 2017.
- [3] Aydılek İB, Aslan A. “A Novel Hybrid Approach to Estimating Missing Values in Databases Using K-Nearest Neighbors and Neural Networks”. *International Journal of Innovative Computing, Information and Control*, 8(7(A)), 4705-4717, 2012.
- [4] Prieto A, Prieto B, Ortigosa EM, Ros E, Pelayo F, Ortega J, Rojas I. “Neural networks: An overview of early research, current frameworks and new challenges”. *Neurocomputing*, 214, 242-268, 2016.
- [5] Benardos PG, Vosniakos GC. “Optimizing feedforward artificial neural network architecture”. *Engineering Applications of Artificial Intelligence*, 20(3), 365-382, 2007.
- [6] Soltoggio A, Stanley KO, Risi S. “Born to learn: The inspiration, progress, and future of evolved plastic artificial neural networks”. *Neural Networks*, 108, 48-67, 2018.
- [7] Barrett DG, Morcos AS, Macke JH. “Analyzing biological and artificial neural networks: challenges with opportunities for synergy?”. *Current Opinion in Neurobiology*, 55, 55-64, 2019.
- [8] Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H. “State-of-the-art in artificial neural network applications: A survey”. *Heliyon*, 4(11), 1-41, 2018.
- [9] Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A. “Deep learning in spiking neural networks”. *Neural Networks*, 111, 47-63, 2019.

- [10] Yang HH, Murata M, Amari S. “Statistical inference: learning in artificial neural networks”. *Trends in Cognitive Sciences*, 2(1), 4-10, 1998.
- [11] Schmidhuber J. “Deep learning in neural networks: An overview”. *Neural Networks*, 61, 85-117, 2015.
- [12] Paliwal M, Kumar UA. “Neural networks and statistical techniques: A review of applications”. *Expert Systems with Applications*, 36(1), 2-17, 2009.
- [13] Whittington JCR, Bogacz R. “Theories of Error Back-Propagation in the Brain”. *Trends in Cognitive Sciences*, 23(3), 235-250, 2019.
- [14] Lampinen J, Vehtari A. “Bayesian approach for neural networks—review and case studies”. *Neural Networks*, 14(3), 257-274, 2001.
- [15] Tsoi AC, Back A. “Discrete time recurrent neural network architectures: A unifying review”. *Neurocomputing*, 15(3-4), 183-223, 1997.
- [16] Heo S, Lee JH. “Parallel neural networks for improved nonlinear principal component analysis”. *Computers & Chemical Engineering*, 127, 1-10, 2019.
- [17] Bornholt J, Lopez R, Carmean DM, Ceze L, Seelig G, Strauss K. “A DNA-Based Archival Storage System”. *Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, Atlanta, Georgia, USA, 2-6 April 2016.
- [18] Abas AR. “Using incremental general regression neural network for learning mixture models from incomplete data”. *Egyptian Informatics Journal*, 12(3), 185-196, 2011.
- [19] Kumar R, Chen T, Hardt M, Beymer D, Brannon K, Syeda-Mahmood T. “Multiple Kernel Completion and its application to cardiac disease discrimination”. *IEEE 10th International Symposium on Biomedical Imaging: From Nano to Macro*, San Francisco, CA, USA, 7-11 April 2013.
- [20] He H, Garcia EA. “Learning from Imbalanced Data”. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284, 2009.
- [21] Frasca M, Bertoni A, Re M, Valentini G. “A neural network algorithm for semi-supervised node label learning from unbalanced data”. *Neural Networks*, 43, 84-98, 2013.
- [22] Kohavi R. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. *Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada, 20-25 August 1995.
- [23] Bashiri M, Geranmayeh AF. “Tuning the parameters of an artificial neural network using central composite design and genetic algorithm”. *Scientia Iranica*, 18(6), 1600-1608, 2011.
- [24] Benjamin BV, Gao P, McQuinn E, Choudhary S, Chandrasekaran AR, Bussat J, Alvarez-Icaza R, Arthur JV, Merolla PA, Boahen K. “Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations”. *Proceedings of the IEEE*, 102(5), 699-716, 2014.
- [25] Naveros F, Luque NR, Garrido JA, Carrillo RR, Anguita M, Ros E. “A Spiking Neural Simulator Integrating Event-Driven and Time-Driven Computation Schemes Using Parallel CPU-GPU Co-Processing: A Case Study”. *IEEE Transactions on Neural Networks and Learning Systems*, 26(7), 1567-1574, 2015.
- [26] Shepherd GM, Mirsky JS, Healy MD, Singer MS, Skoufos E, Hines MS, Nadkarni PM, Miller PL. “The Human Brain Project: neuroinformatics tools for integrating, searching and modeling multidisciplinary neuroscience data”. *Trends in Neurosciences*, 21(11), 460-468, 1998.

- [27] Sarikan SS, Ozbayoglu AM, Zilci O. “Automated Vehicle Classification with Image Processing and Computational Intelligence”. *Procedia Computer Science*, 114, 515-522, 2017.
- [28] Castro LND. “Fundamentals of natural computing: an overview”. *Physics of Life Reviews*, 4,(1), 1-36, 2007.
- [29] Brunton BW, Beyeler M. “Data-driven models in human neuroscience and neuroengineering”. *Current Opinion in Neurobiology*, 58, 21-29, 2019.
- [30] Buccelli S, Bornat Y, Colombi I, Ambroise M, Martines L, Pasquale V, Bisio M, Tessadori J, Nowak P, Grassia F, Averna A, Tedesco M, Bonifazi P, Difato F, Massobrio P, Levi T, Chiappalone M. “A Neuromorphic Prosthesis to Restore Communication in Neuronal Networks”. *IScience*, 19, 402-414, 2019.
- [31] Rosenblatt F. “The perceptron: A probabilistic model for information storage and organization in the brain”. *Psychological Review*, 65(6), 386-408, 1958.
- [32] Willshaw DJ, Malsburg CVD. “How patterned neural connections can be set up by self-organization”. *Proceedings of the Royal Society of London - Biological Sciences*, 194(1117), 431-445, 1976.
- [33] Amari S. “Topographic organization of nerve fields”. *Bulletin of Mathematical Biology*, 42(3), 339-364, 1980.
- [34] Kohonen T. “Self-Organized Formation of Topologically Correct Feature Maps”. *Biological Cybernetics*, 43(1), 59-69, 1982.
- [35] Hornik K, Stinchcombe M, White H. “Multilayer feedforward networks are universal approximators”. *Neural Networks*, 2(5), 359-366, 1989.
- [36] Jordan MI, Rumelhart DE. “Forward models: Supervised learning with a distal teacher”. *Cognitive Science*, 16(3), 307-354, 1992.
- [37] Wang J, Wang H, Chen Y, Liu C. “A constructive algorithm for unsupervised learning with incremental neural network”. *Journal of Applied Research and Technology*, 13(2), 188-196, 2015.
- [38] Plasencia A, Shichkina Y, Suárez I, Ruiz Z. “Open Source Robotic Simulators Platforms for Teaching Deep Reinforcement Learning Algorithms”. *Procedia Computer Science*, 150, 162-170, 2019.
- [39] Kiranyaz S, Ince T, Yildirim A, Gabbouj M. “Evolutionary artificial neural networks by multi-dimensional particle swarm optimization”. *Neural Networks*, 22(10), 1448-1462, 2009.
- [40] Castellani M, Rowlands H. “Evolutionary Artificial Neural Network Design and Training for wood veneer classification”. *Engineering Applications of Artificial Intelligence*, 22(4-5), 732-741, 2009.
- [41] Rashid TA, Fattah P, Awla DK. “Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms”. *Procedia Computer Science*, 140, 324-333, 2018.
- [42] Castillo PA, Merelo JJ, Prieto A, Rivas V, Romero G. “G-Prop: Global optimization of multilayer perceptrons using GAs”. *Neurocomputing*, 35(1-4), 149-163, 2000.
- [43] Yao X, Liu Y. “Towards designing artificial neural networks by evolution”. *Applied Mathematics and Computation*, 91(1), 83-90, 1998.
- [44] Jiang X, Wah AHKS. “Constructing and training feed-forward neural networks for pattern classification”. *Pattern Recognition*, 36(4), 853-867, 2003.

- [45] Balkin SD, Ord JK. “Automatic neural network modeling for univariate time series”. *International Journal of Forecasting*, 16(4), 509-515, 2000.
- [46] Leski J, Czogola E. “A new artificial neural network based fuzzy inference system with moving consequents in if–then rules and selected applications”. *Fuzzy Sets and Systems*, 108(3), 289-297, 1999.
- [47] Benardos PG, Vosniakos GC. “Prediction of surface roughness in CNC face milling using neural networks and Taguchi's design of experiments”. *Robotics and Computer-Integrated Manufacturing*, 18(5-6), 343-354, 2002.
- [48] Maier HR, Dandy GC. “Understanding the behaviour and optimising the performance of back-propagation neural networks: an empirical study”. *Environmental Modelling & Software*, 13(2), 179-191, 1998.
- [49] Wang Y, Fariello G. “On Neuroinformatics: Mathematical Models of Neuroscience and Neurocomputing”. *Journal of Advanced Mathematics and Applications*, 1(2), 206-217, 2012.
- [50] Branitskiy A, Kotenko I. “Hybridization of computational intelligence methods for attack detection in computer networks”. *Journal of Computational Science*, 23, 145-156, 2017.
- [51] Pal SK, Meher SK. “Title Paper: Natural computing: A problem solving paradigm with granular information processing”. *Applied Soft Computing*, 13(9), 3944-3955, 2013.
- [52] Lozo P. “Selective Attention Adaptive Resonance Theory (SAART) Neural Network for Neuro-engineering of Robust ATR Systems”. *International Conference on Neural Networks*, Perth, WA, Australia, 27 November-1 December 1995.
- [53] Disney A, Reynolds J, Schuman CD, Klibisz A, Young A, Plank JS. “DANNA: A neuromorphic software ecosystem”. *Biologically Inspired Cognitive Architectures*, 17, 49-56, 2016.
- [54] Arifovic J, Gençay R. “Using genetic algorithms to select architecture of a feedforward artificial neural network”. *Physica A: Statistical Mechanics and its Applications*, 289(3-4), 574-594, 2001.
- [55] Ma L, Khorasani K. “A new strategy for adaptively constructing multilayer feedforward neural networks”. *Neurocomputing*, 51, 361-385, 2003.
- [56] Chen J, Chang J. “Fuzzy Perceptron Neural Networks for Classifiers with Numerical Data and Linguistic Rules as Inputs”. *IEEE Transactions on Fuzzy Systems*, 8(6), 730-745, 2000.
- [57] Maier HR, Dandy GC. “The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study”. *Environmental Modelling & Software*, 13(2), 193-209, 1998.
- [58] School of Information and Computer Sciences, University of California, Irvine, CA, USA. “UCI Machine Learning Repository”. <https://archive.ics.uci.edu/ml/index.php>. (16.10.2019).
- [59] School of Information and Computer Sciences, University of California, Irvine, CA, USA. “UCI Machine Learning Repository: Blood Transfusion Service Center Data Set”. <https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center>. (16.10.2019).
- [60] Yeh I, Yang K, Ting T. “Knowledge discovery on RFM model using Bernoulli sequence”. *Expert Systems with Applications*, 36(3), 5866-5871, 2009.