



Bir Gizli Katmanlı Yapay Sinir Ağlarında Optimal Nöron Sayısının İncelenmesi

Tayfun ÜNAL^{*1,2}, Ünver ÇİFTÇİ¹, Nurkut Nuray URGAN¹

¹Tekirdağ Namık Kemal Üniversitesi, Fen-Edebiyat Fakültesi, Matematik Bölümü, 59030, Tekirdağ, Türkiye

²Kırklareli Üniversitesi, Bilgi İşlem Daire Başkanlığı, 39000, Kırklareli, Türkiye

*yazışılan yazar e-posta: tayfununal39@gmail.com

(Alınış / Received: 10.01.2022, Kabul / Accepted: 7.08.2022, Yayınlanma / Published: 25.11.2022)

Öz: Bu makalede, bir gizli katmanlı yapay sinir ağları için optimal nöron sayısı araştırılmıştır. Bunun için teorik ve istatistiksel çalışmalar yapılmıştır. Optimal nöron sayısını bulmak için global minimum bulmak gereklidir. Ancak yapay sinir ağlarının eğitimi konveks olmayan bir problem olduğundan optimizasyon algoritmaları ile global minimum bulmak zordur. Bu çalışmada global minimumu dolayısıyla optimum nöron sayısını bulmak için baskı maliyet fonksiyonu önerilmiştir. Baskı maliyet fonksiyonu yardımıyla global minimumu veren yapay sinir ağı modelinin nöron sayısının, optimal nöron sayısını verdiği gösterilmiştir. Ayrıca baskı maliyet fonksiyonu XOR veri kümesi ve daire veri kümesi üzerinde test edilmiş ve XOR veri kümesi üzerinde %99, daire veri kümesi üzerinde ise %97 başarı elde edilmiştir. Bu veri kümeleri için optimal nöron sayısı tespit edilmiştir.

Anahtar kelimeler: Derin Öğrenme, Yapay sinir ağları, Tam bağlı katman, XOR problemi, Nöron sayısı

*Bu çalışmanın bir kısmı, Tekirdağ Namık Kemal Üniversitesi'nde yüksek lisans kapsamında 'Bir ve iki gizli katmanlı yapay sinir ağlarının matematiği' başlıklı tezden üretilmiştir.

Investigation of the Optimal Number of Neuron in One-Hidden-Layer Artificial Neural Networks

Abstract: In this paper, optimal number of neurons in one-hidden-layer artificial neural networks is investigated. Theoretical and statistical studies are carried out for this goal. Finding the global minimum is necessary in order to determine the optimal number of neurons. However, since the training of artificial neural networks is a non-convex problem, it is difficult to find a global minimum with optimization algorithms. In this study, an augmented cost function is proposed to find the global minimum, hence the optimal number of neurons. It is shown that the optimal number of neurons is produced by the artificial neural network model, which gives the global minimum with the aid of the augmented cost function. Additionally, the XOR and circle datasets are used to test the augmented cost function, and 99% success was achieved on the XOR dataset and 97% on the circle dataset. The optimal number of neurons is determined for these datasets.

Keywords: Deep Learning, Artificial Neural Networks, Fully-Connected Layer, XOR Problem, Number of Neurons

*A part of this study was produced from the thesis titled "Mathematics of one and two hidden layer artificial neural networks" within the scope of master's degree at Tekirdağ Namık Kemal University.

1. Giriş

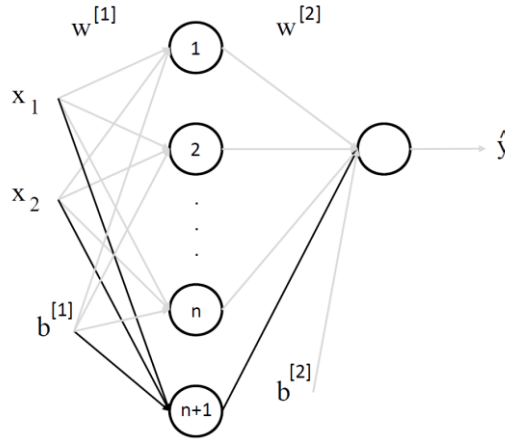
Günümüzde yapay zekâ birçok pratik uygulama ve aktif araştırma konuları ile gelişen bir alandır [1-3]. Bu gelişmede derin öğrenme [1, 4–10] algoritmalarının payı büyüktür. Derin öğrenme algoritmalarının temel yapı taşı nöronlardır. Nöronlar bir araya gelerek katmanları, katmanlar bir araya gelerek yapay sinir ağlarını oluşturmaktadır. Ancak burada kaç tane katman ve bu katmanların her birinde kaç tane nöron kullanılacağı bilinmemektedir. Üstelik oluşturulan bu mimarinin başarısının matematiksel sonucu da tasvir edilememektedir [1, 11]. Bu yüzden ideal mimariyi bulmak için denemeler yapılmalıdır. Bir yapay sinir ağı mimarisine bir nöronun eklenmesi veya çıkarılmasıyla parametreler, hata oranı ve veri kümesi üzerindeki başarı oranı değişebilmektedir. Bu yüzden yapay sinir ağında kullanılacak optimal nöron sayısını bulmak zordur [12, 13]. Çok fazla nöron kullanmak veri kümesi üzerinde fazla uymaya (overfitting) [14] ve bu da modelin genelleştirme gücünün zayıflamasına yol açmaktadır [4]. Aksine modelde gereğinden az nöron kullanıldığında ise model veri kümesini temsil edemeyebilir. Dolayısıyla optimal nöron sayısını bulmak gereklidir.

Bu makalede, bir gizli katmanlı yapay sinir ağlarında optimal nöron sayısı incelenmiştir. Bunun için baskı maliyet fonksiyonu $J_{baskı}(W, b)$ önerilmiştir (Denklem 1).

$$J_{baskı}(W, b) = \frac{1}{m} \sum_{i=1}^m L(y^{(i)}, \hat{y}^{(i)}) + \gamma \Omega(w, b) \quad (1)$$

$$\Omega(w, b) = \left[\sum_{i=1}^2 (w_{n,i}^{[1]})^2 \right] + (b_{n,1}^{[1]})^2 + (w_{1,n}^{[2]})^2 \quad (2)$$

Baskı maliyet fonksiyonu iki terimden oluşmaktadır. İlk terim $L(y^{(i)}, \hat{y}^{(i)})$ hatayı, ikinci terim $\gamma \Omega(w, b)$ ($\gamma \geq 0$) ise son nöronun parametrelerini sıfır yapmak için baskılayan terimi göstermektedir. Bu ceza terimi olarak da adlandırılmaktadır [12]. Optimal nöron sayısını incelemek için gizli katmanda n nöron olan bir mimariyi ve aynı mimariye bir nöron daha ekleyerek elde edilen n+1 nöronlu mimariyi düşünelim (Şekil 1). İki modelin de başlangıç parametreleri ilk n nöron için aynıdır. Sadece ikinci modelin (n+1). nöronunun parametreleri atanmıştır.



Şekil 1. Baskılanan nöron (n+1)

Aynı veri kümesi üzerinde bu iki mimariden n nöronlu olan mimari maliyet fonksiyonu $J(W, b)$ ile $n+1$ nöronlu mimari ise baskı maliyet fonksiyonu $J_{baskı}(W, b)$ ile eğitilsin. Daha sonra buradan elde edilen $J(W, b)$ ve $J_{baskı}(W, b)$ değerleri kıyaslanarak nöron sayıları değerlendirilebilir. Örneğin ϵ istenen maliyet değeri olmak üzere eğer $J_{baskı}(W, b) = J(W, b) < \epsilon$ ise eklenen son nöronun parametreleri sıfır demektir ve bu nöronun gereksiz olduğu açıktır. Dolayısıyla n nöronlu mimari global minimumu vermektedir ve optimal nöron sayısıdır. Eğer $J(W, b) < J_{baskı}(W, b) < \epsilon$ ise bu doğal bir sonuçtur ve istenilen bir sonuçtur. Çünkü baskı maliyet fonksiyonunun doğası gereği maliyet fonksiyonunu baskılayacaktır. Son nöronun parametreleri sıfıra yaklaşmaya çalışacaktır. Böylece maliyet fonksiyonundan büyük olacaktır. Bu durumda optimal nöron sayısı n olarak seçilebilir. Eğer $J_{baskı}(W, b) < J(W, b) < \epsilon$ ise eklenen son nöron yapay sinir ağının gelişmesine katkıda bulunmuştur ve bu nöron gereklidir. Başka bir ifade ile son nöron baskı maliyet fonksiyonu ile sıfıra gitmeye zorlanmasına rağmen sıfır olmamıştır. Bu durumda son nöronun parametreleri sıfır oluncaya kadar ve istenilen durum ($J(W, b) < J_{baskı}(W, b) < \epsilon$) elde edilinceye kadar nöron eklenmeye devam edilmelidir. Bu yüzden bu çalışmada bir gizli katmanlı yapay sinir ağları için 1 nörondan başlayarak, nöron ekleyerek optimal nöron sayısı incelenmiştir. Bu bilgilerden yola çıkarak XOR veri kümesi üzerinde bir gizli katmanlı yapay sinir ağlarının optimal nöron sayısı için teorik ve istatistiksel olarak toplam dört deney yapılmıştır ve yine XOR veri kümesi ve daire veri kümesi üzerinde bir gizli katmanlı yapay sinir ağı mimarisi için optimal nöron sayısı, önerilen yöntem ile test edilmiştir. İlk deneyde 1, 5, 10, ..., 100 nöronlu mimariler için ortalama maliyet değeri ve ortalama test maliyet değeri hesaplanarak modelin fazla uyma yaptığı mimariler atılmıştır. Bunun sonucu olarak ikinci deneyde 1, 2, ..., 20 nöronlu mimariler üzerinde durulmuştur. Burada 1, 2, ..., 20 nöron için ortalama maliyet ve ortalama test maliyet değerleri hesaplanmıştır. Üçüncü deney de ise ikinci deneydeki sonuçlarla kıyaslamak üzere 1, 2, ..., 20, 21 nöronlu mimariler için ortalama baskı maliyet değerleri hesaplanmıştır. Son deney olarak nöron sayısına karşılık gelen maliyet değerlerinin güven aralıkları hesaplanarak istatistiksel bir deney yapılmıştır.

Bu makalenin yenilikçi yönü baskı maliyet fonksiyonunu kullanarak optimal nöron sayısının tespiti için yeni bir yöntem sunmasıdır. Bu yenilikçi yönüyle bir gizli katmanlı yapay sinir ağları için verilen bir veri kümesi için optimal nöron sayısının belirlenmesi konusunda literatüre katkı sağlayacaktır.

Yapay sinir ağlarında optimal nöron sayısını belirlemek için kullanılan iki farklı yaklaşım vardır. İlk yaklaşım minimal bir mimari ile başlayıp, nöron ekleyerek ağı kapasitesini geliştirip optimal nöron sayısını elde etmektir [15–20]. İkinci yaklaşım ise gereğinden fazla nöron bulunan bir mimari ile başlayıp, gereksiz nöronları budayarak optimum nöron sayısını elde etmektir [13, 21–30]. Bu yaklaşımlara ilave olarak her iki yaklaşımı da kullanan hibrit çalışmalar da mevcuttur [31–34]. Bu çalışmada, optimal nöron sayısını bulmak için ilk yaklaşım olan nöron ekleme yöntemi kullanılmıştır. Yapay sinir ağları eğitimi konveks olmayan bir problemdir. Bu yüzden optimizasyon algoritmaları kullanılarak kesin çözüme ulaşılamaz. Ancak optimal nöron sayısı için problem konveks bir yapıya dönüşebilmektedir. Kawaguchi ve Leslie Pack Kaelbling nöron ekleyerek yerel minimumların elenebildiğini ispatlamışlardır [35]. Benjamin D. Haefele ve Ren'e Vidal bir gizli katmanlı yapay sinir ağları için gizli katmanında $n+1$ nöron bulunan mimarinin son nöronu sıfıra yaklaşıyorsa n nöronlu model problemin global minimumunu verdiği ispatlamışlardır [11].

Bu çalışmada da [11] makalesinin bir sonucu olarak son nöronu sıfıra yaklaşmaya zorlayan baskı maliyet fonksiyonu tanımlanmıştır. Baskı maliyet fonksiyonu yardımıyla global minimumu dolayısıyla optimal nöron sayısı incelenmiştir. İkinci bölümde yapay

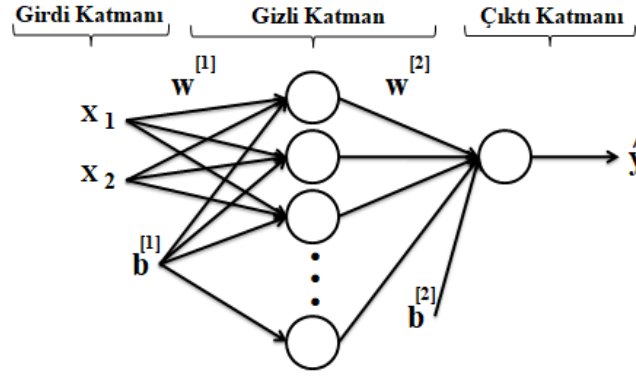
sinir ağlarının algoritmalarından ve kullanılan fonksiyonlardan bahsedilmiştir. Üçüncü bölümde kullanılan XOR veri kümesi ve daire veri kümesi verilmiştir. Dördüncü bölümde yapılan çalışmalar ve bulgular açıklanmıştır. Son olarak beşinci bölümde ise sonuç ve buna bağlı olarak ileride yapılabilecek çalışmalar için açıklamalar yapılmıştır.

2. Materyal ve Metot

Yapay sinir ağlarının başarısını etkileyen faktörler vardır. Bunlar nöron sayısı, başlangıç parametreleri, veri kümesi, aktivasyon fonksiyonları, öğrenme katsayısı (α) ve iterasyon sayısıdır (epoch). Bu çalışmada yapılan deneylerde optimal nöron sayısı incelendiğinden nöron sayısı dışındaki diğer faktörlere kısıtlamalar getirilmiştir. Bu kısıtlamalar modeller arasında kıyaslama yapabilmek için getirilmiştir. Bu bölüm de deneylerde yapılan kısıtlamalardan, kullanılan fonksiyonlardan ve yapay sinir ağlarından bahsedilmiştir.

2.1 Yapay sinir ağı modeli

XOR veri kümesi ve daire veri kümesi (Bölüm 3) ele alındığı için veriler iki boyutludur ve problemimiz bir sınıflandırma problemidir. O halde iki boyutlu veri içeren bir veri kümesi, $i = 1, 2, \dots, 1000$ için $x^{(i)} = (x_1^{(i)}, x_2^{(i)}) \in \mathbb{R}^2$ ve $y^{(i)} \in \{0, 1\}$ olmak üzere Şekil 2’de gizli katmanında n tane nörona sahip bir gizli katmanlı yapay sinir ağı modeli verilmiştir.

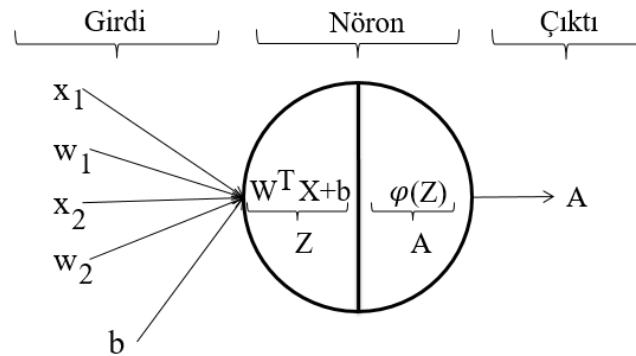


Şekil 2. Bir gizli katmanlı yapay sinir ağı modeli

Şekil 2’de $\hat{y} \in [0,1]$ tahmin değeri ve modelin parametreleri $W^{[1]} \in \mathbb{R}^{n \times 2}$, $b^{[1]} \in \mathbb{R}^{n \times 1}$, $W^{[2]} \in \mathbb{R}^{1 \times n}$ ve $b^{[2]} \in \mathbb{R}$ dir.

2.2 Bir nöronun yapısı

Nöronlar genellikle temsilen bir çember sembolü ile gösterilir. Her bir nöronun girdileri ve ona ait parametreleri vardır. Şekil 3’te bir nöronun yapısı verilmiştir.



Şekil 3. Bir nöronun yapısı

Girdiler Şekil 3'te görüldüğü gibi nörona girer ve Z değeri hesaplanır. Daha sonra, Z değerine bir fonksiyon uygulanır. Bu fonksiyona aktivasyon fonksiyonu denir. Böylece çıktı değeri, yani A elde edilir.

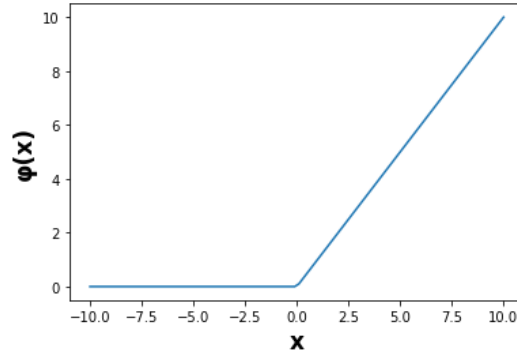
2.3 Aktivasyon fonksiyonları

Nöronlarda kullanılan fonksiyonlara aktivasyon fonksiyonu denir. Modern yapay sinir ağlarında varsayılan öneri ReLU aktivasyon fonksiyonunun kullanılmasıdır [1]. Bu makalede de gizli katmanlarda ReLU aktivasyon fonksiyonu kullanılmıştır (Bölüm 2.3.1). Başka aktivasyon fonksiyonları da kullanılabilir. Ancak yapılan deneylerde modeller arasında kıyaslama yapabilmek için ReLU kısıtlaması getirilmiştir. Çıktı katmanında ise problemimiz ikili sınıflandırma problemi olduğundan sigmoid aktivasyon fonksiyonu kullanılmıştır (Bölüm 2.3.2).

2.3.1 ReLU (Rectified Linear Unit) fonksiyonu

$$\begin{aligned} \varphi &: \mathbb{R} \rightarrow [0, \infty) \\ x &\rightarrow \varphi(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \end{aligned}$$

Fonksiyonuna ReLU aktivasyon fonksiyonu denir ve φ harfi ile gösterilecektir. Grafiği Şekil 4'te verilmiştir.

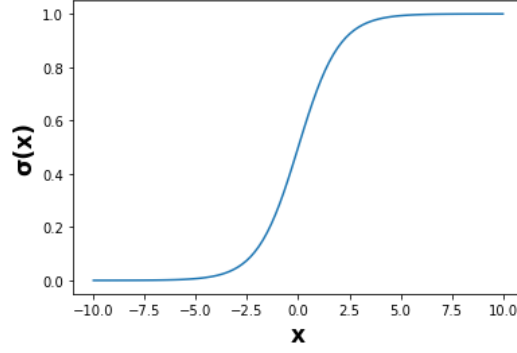


Şekil 4. ReLU aktivasyon fonksiyonu

2.3.2 Sigmoid fonksiyonu

$$\begin{aligned} \sigma &: \mathbb{R} \rightarrow (0,1) \\ x &\rightarrow \sigma(x) = \frac{1}{1 + e^{-x}} \end{aligned}$$

Fonksiyonuna sigmoid aktivasyon fonksiyonu denir ve σ harfi ile gösterilecektir. Grafiği Şekil 5’te verilmiştir.



Şekil 5. Sigmoid aktivasyon fonksiyonu

2.4 Kayıp (loss) fonksiyonu

Bir gizli katmanlı yapay sinir ağı için kayıp fonksiyonu veri kümesi üzerinde yapılan hatayı hesaplamak için kullanılmaktadır. Herhangi bir veri kümesi $(x^{(i)}, y^{(i)})$ olmak üzere bir girdi için tahmin değeri

$$\hat{y}^{(i)} = \sigma(W^{[2]}\varphi(W^{[1]}x^{(i)} + b^{[1]}) + b^{[2]}) \quad (3)$$

Denklem 3’te olduğu gibi elde edilir. Verilen girdiye karşılık tahminde yapılan hatayı bulmak için kayıp fonksiyonu

$$L(y^{(i)}, \hat{y}^{(i)}) = -[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})] \quad (4)$$

olarak tanımlanır. Aynı zamanda Denklem 4’e “cross-entropy” kayıp fonksiyonu da denir. İkili sınıflandırma problemlerinde bu fonksiyon kullanılmaktadır.

2.5 Maliyet fonksiyonu

Kayıp fonksiyonunda hesaplanan her bir veri için bulunan kayıp değerlerinin ortalaması maliyet fonksiyonunu verir. J ile gösterilecektir ve

$$J(X, Y, W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}) = \frac{1}{m} \sum_{i=1}^m L(y^{(i)}, \hat{y}^{(i)}) \quad (5)$$

olarak tanımlanmaktadır. İlerleyen bölümlerde kısaca maliyet fonksiyonu $J(W, b)$ ile gösterilecektir.

2.6 İleri yayılım

İleri yayılım, veri kümesindeki her bir veri (girdi) için tahminin (çıktının) hesaplanması olarak tanımlanmaktadır. Veri kümesindeki veri sayısı m ve modelin gizli katmanındaki nöron sayısı n olmak üzere $X \in \mathbb{R}^{2 \times m}$, $W^{[1]} \in \mathbb{R}^{n \times 2}$, $b^{[1]} \in \mathbb{R}^{n \times 1}$, $W^{[2]} \in \mathbb{R}^{1 \times n}$ ve $b^{[2]} \in \mathbb{R}$ için

$$Z^{[1]} = W^{[1]}X + b^{[1]} \in \mathbb{R}^{n \times m} \quad (6)$$

$$A^{[1]} = \varphi(Z^{[1]}) \quad \in \mathbb{R}^{n \times m} \quad (7)$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]} \quad \in \mathbb{R}^{1 \times m} \quad (8)$$

$$A^{[2]} = \sigma(Z^{[2]}) \quad \in \mathbb{R}^{1 \times m} \quad (9)$$

Denklem 6, 7, 8 ve 9 sırasıyla hesaplanır. Buradan elde edilen $A^{[2]}$ vektörü, belirlenen başlangıç parametrelerine göre her bir veriye karşılık gelen tahminlerdir ve $\hat{Y} = A^{[2]}$ olarak tanımlanır. İleri yayılım kullanılarak tahmin yapıldıktan sonra maliyet fonksiyonu ile yapılan hata hesaplanır. Daha sonra hatayı optimize etmek için meyilli azalım kullanılarak geri yayılım yapılır.

2.7 Geri yayılım

Geri yayılım, zincir kuralı [1] kullanılarak maliyet fonksiyonunun her bir parametreye göre türevlerin hesaplanmasıdır. Maliyet fonksiyonunun parametrelere göre türevlerini bulmak için

$$dZ^{[2]} = \frac{\partial J(W, b)}{\partial Z^{[2]}} \quad , \quad dW^{[2]} = \frac{\partial J(W, b)}{\partial W^{[2]}} \quad , \quad db^{[2]} = \frac{\partial J(W, b)}{\partial b^{[2]}}$$

$$dZ^{[1]} = \frac{\partial J(W, b)}{\partial Z^{[1]}} \quad , \quad dW^{[1]} = \frac{\partial J(W, b)}{\partial W^{[1]}} \quad , \quad db^{[1]} = \frac{\partial J(W, b)}{\partial b^{[1]}}$$

olmak üzere aşağıda bu türevler elde edilmiştir. Burada elde edilen türevler gizli katmanda ReLU aktivasyon fonksiyonu ve çıktı katmanında sigmoid aktivasyon fonksiyonu olduğu varsayılarak hesaplanmıştır.

$$dZ^{[2]} = (A^{[2]} - Y) \quad \in \mathbb{R}^{1 \times m} \quad (10)$$

$$dW^{[2]} = \frac{1}{m} \cdot (dZ^{[2]})(A^{[1]})^T \quad \in \mathbb{R}^{1 \times n} \quad (11)$$

$$db^{[2]} = \frac{1}{m} \sum_{i=1}^m dZ_{1i}^{[2]} \quad \in \mathbb{R} \quad (12)$$

$$dZ^{[1]} = \frac{1}{m} \cdot [(W^{[2]})^T dZ^{[2]}] \cdot \varphi'(Z^{[1]}) \quad \in \mathbb{R}^{n \times m} \quad (13)$$

$$dW^{[1]} = \frac{1}{m} [dZ^{[1]} X^T] \quad \in \mathbb{R}^{n \times m} \quad (14)$$

$$db_i^{[1]} = \frac{1}{m} \sum_{j=1}^m dZ_{ij}^{[1]}, 1 \leq j \leq n, db^{[1]} \quad \in \mathbb{R}^{n \times 1} \quad (15)$$

Denklem 11, 12, 14 ve 15'te elde edilen türevler meyilli azalım da kullanılarak parametreler güncellenip modelin başarısı arttırılmaktadır.

2.8 Meyilli azalım (Gradient descent)

Modelin başarısını arttırmak için maliyet fonksiyonunu minimize eden bir optimizasyon algoritmadır. Geri yayılımda bulunan türevler, burada kullanılmıştır. Modelin parametreleri, öğrenme katsayısı α ve $0 < \alpha < 1$ olmak üzere aşağıdaki gibi güncellenir.

$$W^{[2]} := W^{[2]} - \alpha(dW^{[2]}) \quad (16)$$

$$b^{[2]} := b^{[2]} - \alpha(db^{[2]}) \quad (17)$$

$$W^{[1]} := W^{[1]} - \alpha(dW^{[1]}) \quad (18)$$

$$b^{[1]} := b^{[1]} - \alpha(db^{[1]}) \quad (19)$$

Sonuç olarak Denklem 16, 17, 18 ve 19'da güncellenen parametreler kullanılarak maliyet fonksiyonu minimize edilir. Böylece yapılan her iterasyonda bir güncelleme yapılarak maliyet fonksiyonu minimize edilmiş olur ve modelin temsil gücü artmış olur.

2.9 Yapay sinir ağlarında öğrenme

Bu bölümde yukarıdaki fonksiyon ve algoritmaların çalışma prensibi özetlenmiş ve yapay sinir ağlarında öğrenme [36] algoritması adım adım açıklanmıştır. Öncelikle verilen veri kümesi için başlangıç parametreleri atanır. Daha sonra girdiler ve birinci katmana ait başlangıç parametreleri birlikte gizli katmana gelir. Gizli katmandaki nöronlarda gerekli işlemler yapıldıktan sonra buradaki nöronlar ve ikinci katmana ait başlangıç parametreleri çıktı katmanına iletilir. Çıktı katmanında verilen girdilerin tahminleri yapılır. Yapılan tahminler ve verilerin gerçek çıktıları, maliyet fonksiyonu kullanılarak, yapılan hata hesaplanır. Hata değeri yeterince küçük değil ise veri kümesinin öğrenilmesi için aşağıdaki adımlar takip edilir;

- Geri yayılım
- Meyilli azalım ile parametrelerin güncellenmesi
- İleri yayılım
- Maliyet fonksiyonu (veya baskı maliyet fonksiyonu)

Yukarıdaki adımlar sırasıyla uygulanır. Maliyet fonksiyonu yeterince küçük oluncaya kadar adımlar başa dönülerek tekrar tekrar uygulanır. Öğrenme hata değeri yeterince küçük oluncaya kadar devam eder. Öğrenme tamamlandığında elde edilen parametreler istenilen parametrelerdir. Böylece model bu parametreler ile veri kümesini temsil edebilir.

2.10 Baskı maliyet fonksiyonu

Bir gizli katmanlı yapay sinir ağının sonuncu nöronu n ve bu nörona bağlı olan parametreler w, b olsun. Bu modelin sonuncu nörondaki parametrelerin karelerinin toplamı olacak şekilde $\Omega(w, b)$ Denklem 2'de tanımlanmıştır. Buradan baskı maliyet fonksiyonu $\gamma \geq 0$ için Denklem 1'de verilmiştir. Baskı maliyet fonksiyonu, kayıp fonksiyonuna $\Omega(w, b)$ fonksiyonu eklenerek oluşturulmuştur. Ω fonksiyonu [12]'deki gibi kayıp fonksiyonuna eklenen bir ceza fonksiyonudur. Ancak burada tanımladığımız Ω fonksiyonunun farkı tüm parametreleri almak yerine sadece modele eklenen son nöronun parametrelerini almaktır. Konveks olmayan fonksiyonlarda global minimum bulmak zordur. Baskı maliyet fonksiyonu global minimumu bulmak için önerilmiştir. Gizli katmanda n nöron bulunan bir modeli maliyet fonksiyonu ile eğittiğimizde yerel minimumu verdiğini kabul edelim. Ancak $n+1$ nöronlu modeli baskı maliyet fonksiyonu

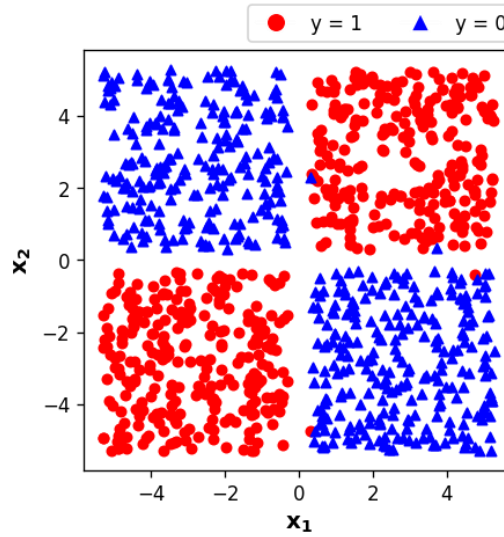
ile eğittiğimizde eklenen son nöron sıfır ve diğer parametreler n nöronlu modelin parametrelerine yaklaşıyorsa eklenen son yani $(n+1)$. nöronun modele bir katkısı olmayacaktır. Dolayısıyla eklenen nöron gereksizdir ve baskı maliyet fonksiyonu ile eğitilen $n+1$ nöronlu model yerel minimum olur. Bu durum, n nöronlu modelin global minimum olduğunu göstermektedir. Bu yüzden n nöronlu model verilen problem için optimal nöron sayısı olacaktır. Bu anlamda [11]'deki makalenin birinci teoremi ile bağdaşmaktadır.

3. Veri Kümeleri

Kullanılan XOR ve daire veri kümeleri “Python” programlama dilinin “plygdata” kütüphanesi kullanılarak üretilmiştir. Her bir veri kümesi 1000 satırdan oluşmaktadır ve veri kümeleri 0,1 gürültü içermektedir.

3.1. XOR veri kümesi

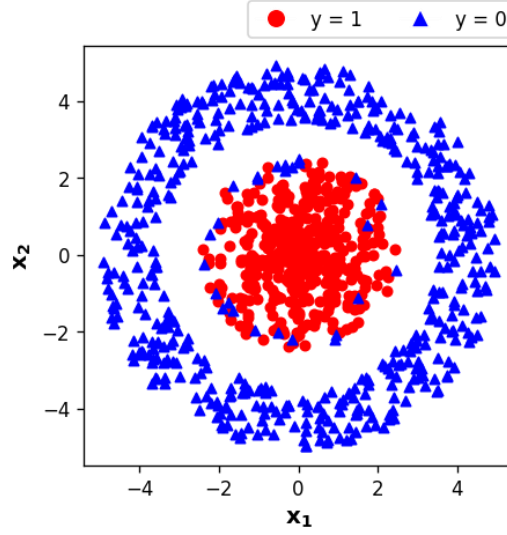
XOR veri kümesi bir sınıflandırma problemidir [10]. Bu çalışmada kullanılan XOR veri kümesinin bir örneği Şekil 6'da gösterilmiştir. XOR veri kümesinin değer kümesi $y = 0$ veya $y = 1$ lerden oluştuğundan veriler etiketli verilerdir. Bu yüzden burada gözetimli (supervised) [1, 4, 7, 37, 38] öğrenme yapılmıştır [1].



Şekil 6. XOR veri kümesinin görselleştirilmesi

3.1. Daire veri kümesi

Daire veri kümesi de bir sınıflandırma problemidir (Şekil 7). Önerilen yöntemi test etmek için kullanılmıştır.



Şekil 7. Daire veri kümesi

4. Bulgular

Bu bölümde bir gizli katmanlı yapay sinir ağlarında optimum nöron sayısını incelemek için baskı maliyet fonksiyonu kullanılarak önerilen yöntem, XOR veri kümesine uygulanmıştır. Toplam 4 deney yapılmıştır. Ayrıca hem XOR veri kümesi hem de daire veri kümesi kullanılarak test edilmiştir. Yapay sinir ağlarının başarısı birçok faktöre bağlı olduğundan yapılan 4 deneyde de Bölüm 2’de tanımlanan fonksiyonlar ve algoritmalar kullanılmıştır. Tablo 1 ve Tablo 2’de kullanılan fonksiyonlar, algoritmalar ve üst değerler verilmiştir.

Tablo 1. Deneylerde kullanılan modellerdeki üst değerler

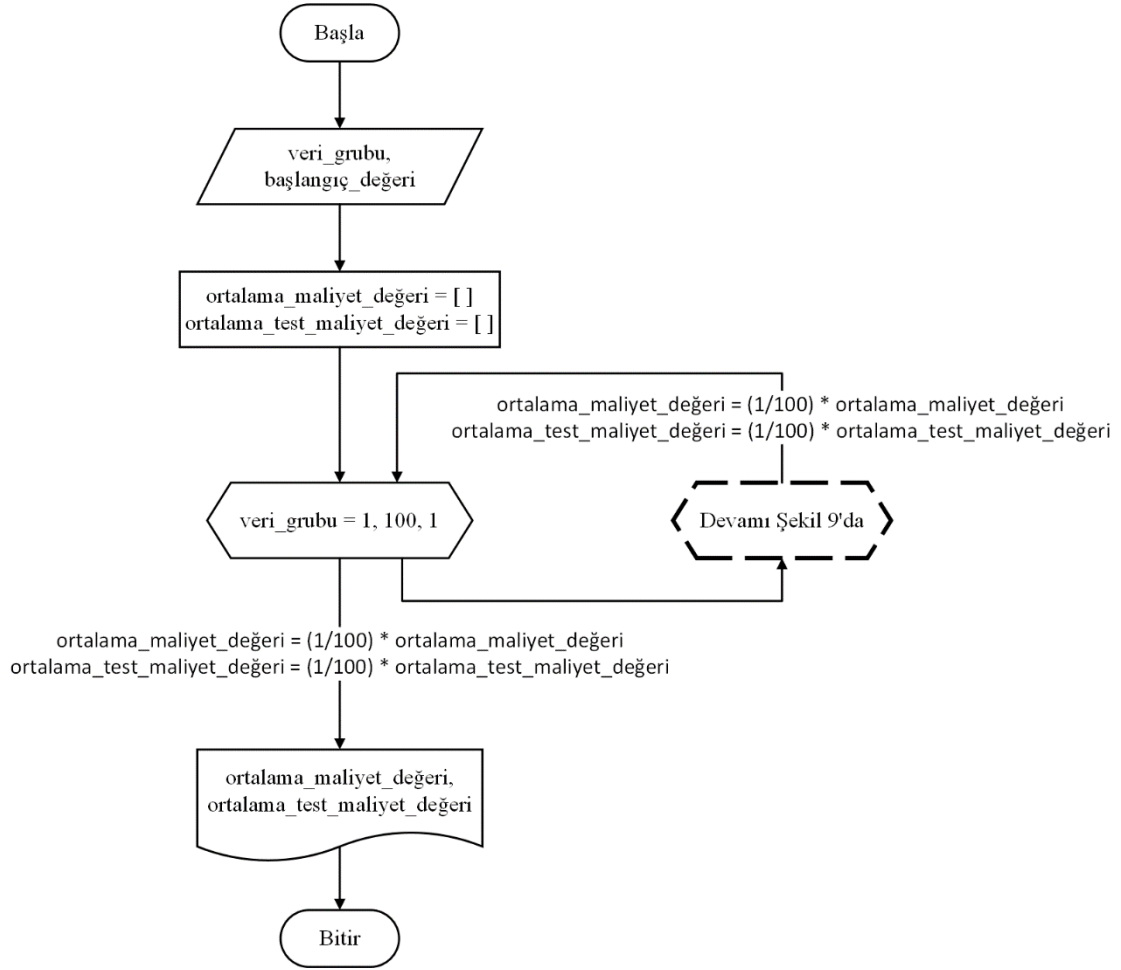
Veri kümelerindeki eleman sayısı	Veri kümesindeki gürültü oranı	İterasyon sayısı	α	γ
1000	0,1	1000	0,6	12

Tablo 2. Deneylerde kullanılan fonksiyon ve algoritmalar

Kayıp fonksiyonu	Optimizasyon fonksiyonu	Gizli katmandaki aktivasyon fonksiyonları	Çıktı katmanındaki aktivasyon fonksiyonu
Cross-entropy	Meyilli azalım	ReLU	Sigmoid

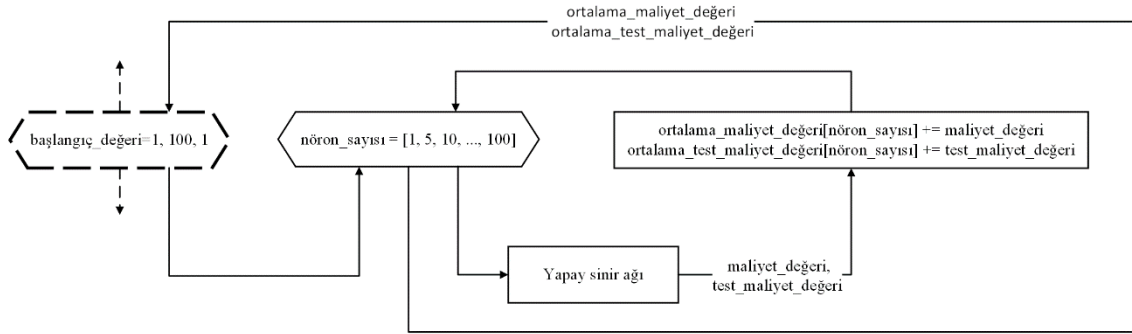
Ayrıca modellerin başarısı, veri kümesi ve başlangıç parametrelerine de bağlıdır. Bu yüzden veri kümesine ve başlangıç parametrelerine olan bağlılığı azaltmak adına deneylerde kullanılmak üzere 100 tane XOR veri kümesi içeren bir veri grubu ve 100 tane rastgele farklı başlangıç parametresi belirlenmiştir. Her bir yapay sinir ağı modelini eğittikten sonra test etmek için XOR veri grubunda olmayan bir test XOR veri kümesi daha üretilmiştir.

İlk deneyin akış şeması Şekil 8 ve Şekil 9’da verilmiştir.



Şekil 8. Deney 1'in akış şeması

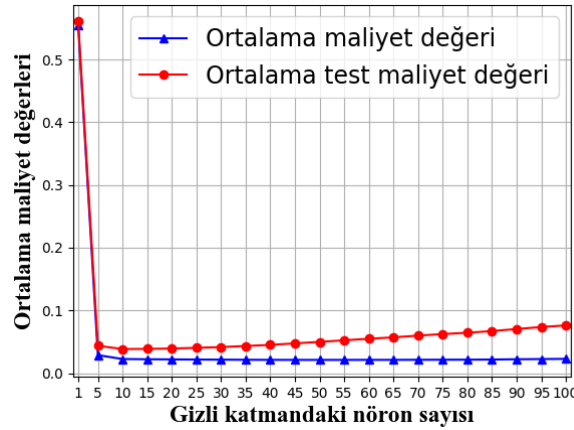
(Akış şeması “başlangıç değerinden” devam etmektedir.)



Şekil 9. Akış şemasının devamı (Deney 1)

İlk deneyde, bir gizli katmanlı yapay sinir ağlarının gizli katmanında kaç nöronun sonra fazla uyamaya sebep olduğu incelenmiştir. Bu modeller optimal nöron içeren modeller olmayacağından atılmıştır. Çünkü fazla uyma olmasını istemiyoruz. Özetle yapay sinir ağı modelinin gizli katmanında maksimum kaç nörona kadar inceleneceği belirlenmiştir. Bunun için XOR veri grubundaki 1. XOR veri kümesi ve 1. başlangıç parametresi için gizli katmanında 1 nöron bulunduran bir gizli katmanlı yapay sinir ağı modeli eğitilmiştir. Maliyet değeri ve test XOR veri kümesi ile test maliyet değeri hesaplanmıştır. Daha sonra gizli katmandaki nöron sayısı 5'e çıkarılmıştır. 1 nöronlu model için kullanılan başlangıç parametreleri sabit tutularak, yeni eklenen nöronlar için başlangıç parametreleri atanıp, model eğitilmiştir. Buradan 5 nöronlu model için bir

maliyet değeri ve eğitilen modelin parametreleri kullanılarak test kümesi üzerinde test maliyet değeri hesaplanmıştır. Benzer şekilde gizli katmanında sırasıyla 10, 15, 20, ..., 95, 100 nöron bulunan modeller eğitilerek, her bir modele karşılık gelecek maliyet değeri ve eğitilen modellerin parametreleri kullanılarak test kümesi üzerinde test maliyet değerleri hesaplanmıştır. Burada, XOR veri grubundaki 1. XOR veri kümesi ve 1. başlangıç parametresi için gizli katmanında sırasıyla 1, 5, 10, ..., 95, 100 nöronlu modellere karşılık gelen maliyet değerleri ve test maliyet değeri elde edilmiştir. Bu işlem 1. XOR veri kümesi için geri kalan 99 farklı başlangıç parametresi ile aynı algoritmalar kullanılarak yapılmıştır. Böylece 100 başlangıç parametresi için sırasıyla 1, 5, 10, ..., 95, 100 nöronlu her bir modele karşılık gelen 100 tane maliyet değerleri ve 100 tane test maliyet değerleri elde edilmiştir. Her bir modelin ayrı ayrı maliyet değerleri ve test maliyet değeri üzerinden ortalamalar alınarak, 1. XOR veri kümesi için 1, 5, 10, ..., 95, 100 nöronlu modellere karşılık gelen ortalama maliyet değerleri ve ortalama test maliyet değeri hesaplanmıştır. Son olarak, 1. XOR veri kümesi için bulunan ortalama maliyet değerleri ve ortalama test maliyet değerleri benzer şekilde veri grubunda geri kalan 99 tane XOR veri kümesi için de hesaplanmıştır. Böylece 100 tane XOR veri kümesine için sırasıyla 1, 5, 10, ..., 95, 100 nöronlu modeller ile her bir modele karşılık gelen 100 tane ortalama maliyet değerleri ve 100 tane ortalama test maliyet değerleri elde edilmiştir. Daha sonra, her bir modelin ayrı ayrı ortalama maliyet değeri ve ortalama test maliyet değeri üzerinden bir kez daha ortalama alınarak, 100 tane XOR veri kümesi ve 100 tane farklı başlangıç parametresi ile eğitilen, gizli katmanında ayrı ayrı 1, 5, 10, ..., 95, 100 nöron olan bir gizli katmanlı yapay sinir ağı modeli için ortalama maliyet değerleri ve ortalama test maliyet değeri hesaplanmıştır. Bu deneyin sonucunda nöron sayısı arttıkça ortalama test maliyet değeri, maliyet değeri azalmasına rağmen artmıştır. Yani nöron sayısı arttıkça fazla uyma olmuştur. Bu deneyde, yapay sinir ağı modelinin gizli katmanındaki nöron sayısına karşılık gelen ortalama maliyet değerleri ve ortalama test maliyet değeri sonuçlarının grafiği Şekil 10'da verilmiştir.



Şekil 10. Sırasıyla gizli katmanında 1, 5, ..., 95, 100 nöron bulunduran bir gizli katmanlı yapay sinir ağı modellerine karşılık gelen ortalama maliyet ve test maliyet değerleri

Tablo 3'te elde edilen sonuçların sayısal değerleri verilmiştir.

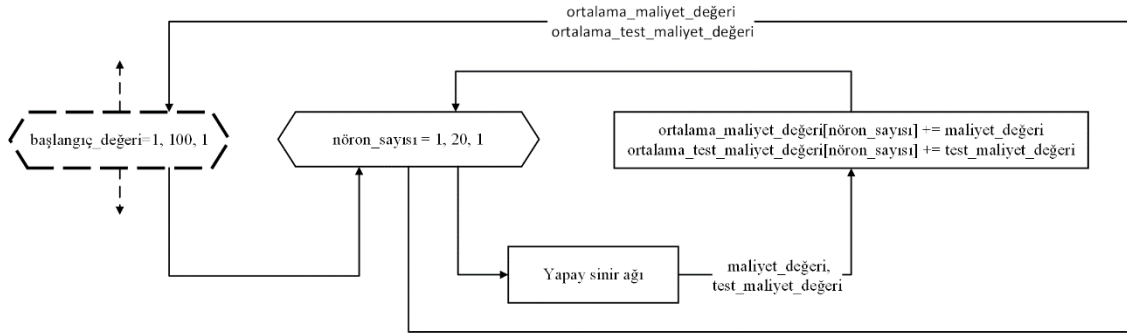
Tablo 3. Ortalama Maliyet Değerleri ve Ortalama Test Maliyet Değerleri

Nöron Sayısı	Ortalama Maliyet Değeri	Ortalama Test Maliyet Değeri
1	0.554539	0.560348
5	0.028810	0.043696
10	0.022603	0.038500
15	0.022235	0.038730

20	0.021971	0.039331
25	0.021762	0.040380
30	0.021571	0.041650
35	0.021397	0.043311
40	0.021267	0.045152
45	0.021196	0.047516
50	0.021181	0.049892
55	0.021186	0.052441
60	0.021242	0.054880
65	0.021238	0.057134
70	0.021360	0.059904
75	0.021468	0.062378
80	0.021552	0.064456
85	0.021745	0.067110
90	0.022220	0.070363
95	0.022528	0.073598
100	0.022845	0.076128

Tablo 3'te ortama maliyet değeri ile ortalama test maliyet değeri arasındaki fark gizli katmanında 20 nöron olan yapay sinir ağı modelinden sonra açık bir şekilde artmıştır. Yani 20 nörondan fazla nöron bulunduran modellerde fazla uyma olmuştur. Bu yüzden ikinci deneyde bir gizli katmanlı yapay sinir ağı modeli için gizli katmanında sırasıyla 1, 2, ..., 20 nöronlu modeller incelenmiştir.

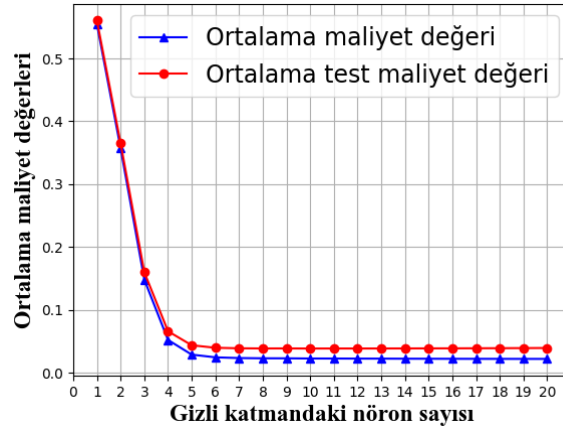
İkinci deneyin akış şeması Şekil 8 ile aynıdır ancak devamı Şekil 9 yerine Şekil 11'de verilmiştir.



Şekil 11. Akış şemasının devamı (Deney 2)

İkinci deneyde, XOR veri grubundaki 1. XOR veri kümesi ve 1. başlangıç parametresi için gizli katmanında 1 nöron bulunduran bir gizli katmanlı yapay sinir ağı modeli eğitilmiştir. Buradan 1 nöronlu model için bir maliyet değeri elde edilmiştir. Eğitilen modelin parametreleri kullanılarak test kümesi üzerinde test maliyet değeri hesaplanmıştır. Daha sonra gizli katmana bir yeni nöron ekleyerek nöron sayısı 2'ye çıkarılmıştır. Gizli katmanında 2 nöron olan bu model için başlangıç parametreleri bir önceki modelin başlangıç parametreleri sabit tutularak, sadece eklenen yeni nöronun parametreleri atanıp, model eğitilmiştir. Buradan 2 nöronlu model için bir maliyet değeri ve eğitilen modelin parametreleri kullanılarak test kümesi üzerinde test maliyet değeri hesaplanmıştır. Benzer şekilde gizli katmanında sırasıyla 3, 4, 5, ..., 19 nöron olan modellerde eğitilerek, her birine karşılık maliyet değerleri ve eğitilen modellerin parametreleri kullanılarak test kümesi üzerinde test maliyet değeri hesaplanmıştır. Son olarak gizli katmanında 19 nöron bulunduran modele bir yeni nöron daha eklenerek nöron sayısı 20'ye çıkarılmıştır. Gizli katmanında 20 nöron olan bu model için başlangıç parametreleri, benzer şekilde bir önceki modelin yani 19 nöronlu modelin başlangıç parametreleri sabit tutularak, sadece eklenen yeni nöronun parametreleri

atanıp, model eğitilmiştir. Buradan 20 nöronlu model için bir maliyet değeri ve eğitilen modelin parametreleri kullanılarak test kümesi üzerinde test maliyet değeri hesaplanmıştır. Sonuç olarak, XOR veri grubundaki 1. XOR veri kümesi ve 1. başlangıç parametresi için gizli katmanında sırasıyla 1, 2, ..., 20 nöronlu modellere karşılık gelen maliyet değerleri ve test maliyet değeri elde edilmiştir. Bu işlem 1. XOR veri kümesi için geri kalan 99 farklı başlangıç parametresi ile aynı algoritmalar kullanılarak yapılmıştır. Böylece 100 başlangıç parametresi için sırasıyla 1, 2, ..., 20 nöronlu her bir modelle karşılık gelen 100 tane maliyet değerleri ve 100 tane test maliyet değeri elde edilmiştir. Her bir modelin ayrı ayrı maliyet değerleri ve test maliyet değeri üzerinden ortalamalar alınarak, 1. XOR veri kümesi gizli katmanında sırasıyla 1, 2, ..., 20 nöronlu modellere karşılık gelen ortalama maliyet değerleri ve ortalama test maliyet değeri hesaplanmıştır. Son olarak, 1. XOR veri kümesi için bulunan ortalama maliyet değerleri ve ortalama test maliyet değeri benzer şekilde geri kalan 99 tane XOR veri kümesi için de hesaplanmıştır. Böylece 100 tane XOR veri kümesi için sırasıyla 1, 2, ..., 20 nöronlu her bir modele karşılık gelen 100 tane ortalama maliyet değerleri ve 100 tane ortalama test maliyet değeri elde edilmiştir. Daha sonra, her bir modelin ayrı ayrı ortalama maliyet değeri ve ortalama test maliyet değeri üzerinden bir kez daha ortalama alınarak, 100 tane XOR veri kümesi ve 100 tane farklı başlangıç parametresi ile eğitilen, gizli katmanında sırasıyla 1, 2, ..., 19 ve 20 nöron olan bir gizli katmanlı yapay sinir ağı modelleri için ortalama maliyet değerleri ve ortalama test maliyet değeri hesaplanmıştır. Sonuçlar Şekil 12'deki grafikte gösterilmiştir. Bu grafikte ortalama maliyet ve ortalama test maliyet değerlerinin sonuçları sırasıyla gizli katmanında 1, 2, ..., 20 nöronlu modeller için gösterilmiştir.



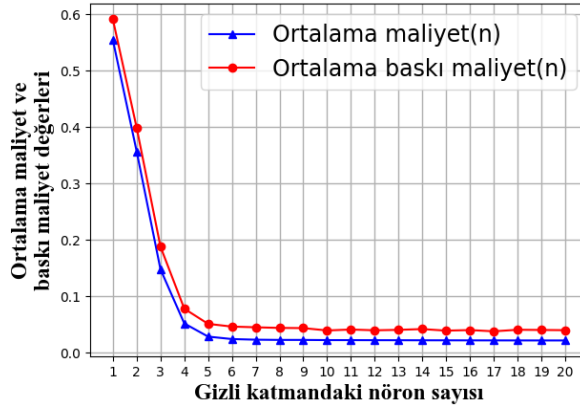
Şekil 12. Sırasıyla gizli katmanında 1, 2, ..., 20 nöron bulunduran bir gizli katmanlı yapay sinir ağı modellerine karşılık gelen ortalama maliyet ve test maliyet değerleri

Tablo 4'te ise bulunan sayısal değerler verilmiştir.

Gizli Katmandaki Nöron Sayısı	Ortalama Maliyet Değerleri	Ortalama Test Maliyet Değerleri
1	0.554539	0.560348
2	0.356986	0.364892
3	0.148057	0.159855
4	0.052143	0.065962
5	0.028810	0.043696
6	0.024475	0.039699
7	0.023342	0.038881
8	0.022937	0.038603
9	0.022913	0.038681

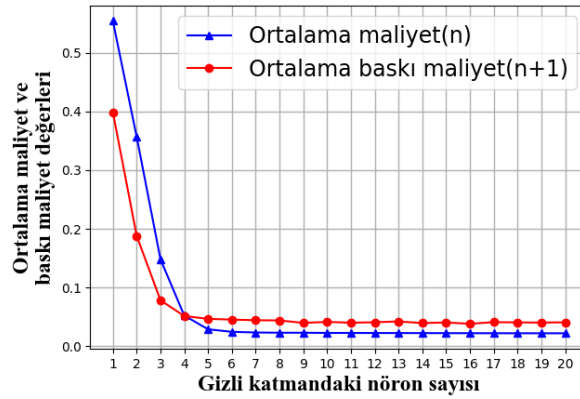
10	0.022603	0.038500
11	0.022559	0.038556
12	0.022461	0.038532
13	0.022402	0.038586
14	0.022285	0.038635
15	0.022235	0.038730
16	0.022137	0.038793
17	0.022089	0.038895
18	0.022038	0.039027
19	0.021994	0.039178
20	0.021971	0.039331

Üçüncü deneyde, ikinci deneyden farklı olarak sadece maliyet fonksiyonu yerine baskı maliyet fonksiyonu kullanılmıştır. XOR veri grubu 1, 2, ..., 20, 21 nöronlu modeller için baskı maliyet fonksiyonu ile eğitilmiştir. Buradan 1, 2, ..., 20, 21 nöronlu modeller için bulunan ortalama baskı maliyet değerleri ile ikinci deneyde 1, 2, ..., 20 nöronlu modeller için bulunan ortalama maliyet değerleri karşılaştırılmıştır. Şekil 13'te aynı nöron sayısı yani $n = 1, 2, \dots, 20$ için hesaplanan ortalama maliyet ve baskı maliyet değerleri karşılaştırılmıştır.



Şekil 13. Aynı nöron sayısı için hesaplanan değerler

Şekil 13'de $J(W, b) \leq J_{baskı}(W, b)$ olduğundan beklenildiği gibi baskı maliyet değeri, maliyet değerinden daha büyük çıkmıştır. Burada n nöron için hesaplanan maliyet değeri ile $n+1$ için hesaplanan baskı maliyet değeri karşılaştırıldığında eklenen nöronun yapay sinir ağının başarısında etkili olup olmadığı gözlemlenmiştir. Şekil 14'te bu karşılaştırmanın grafiği ve Tablo 5'te bulunan sayısal değerler verilmiştir.



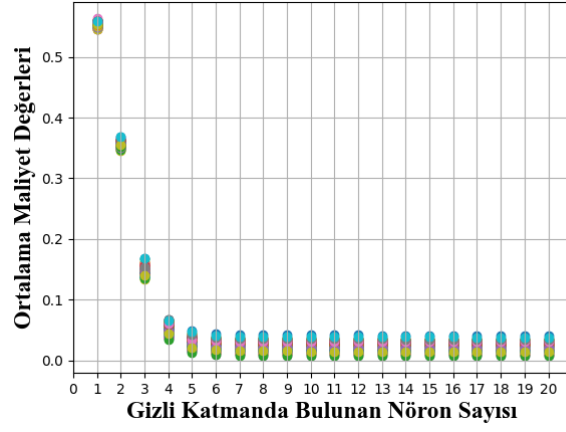
Şekil 14. Nöron sayısı n için hesaplanan ortalama maliyet değeri ile nöron sayısı $n+1$ için hesaplanan ortalama baskı maliyet değerinin grafiği

Tablo 5. Ortalama maliyet değeri ve ortalama baskı maliyet değerlerinin sayısal sonuçları

Gizli Katmandaki Nöron Sayısı	Ortalama Maliyet Değerleri	Ortalama Baskı Maliyet Değerleri
1	0.554539	0.591193
2	0.356986	0.398931
3	0.148057	0.188190
4	0.052143	0.078264
5	0.028810	0.051404
6	0.024475	0.046560
7	0.023342	0.045330
8	0.022937	0.044222
9	0.022913	0.043868
10	0.022603	0.039708
11	0.022559	0.041370
12	0.022461	0.039991
13	0.022402	0.040779
14	0.022285	0.042173
15	0.022235	0.039379
16	0.022137	0.040389
17	0.022089	0.038192
18	0.022038	0.040925
19	0.021994	0.040568
20	0.021971	0.040186
21	-	0.040730

Şekil 14’te 1, 2, 3 ve 4 için baskı maliyet değeri, maliyet değerinden daha küçük çıkmıştır. Ancak normal şartlarda ortalama baskı maliyet değeri, ortalama maliyet değerinden daha büyük çıkmalıdır. Burada gizli katmana 2., 3., ve 4. nöronlar eklendiğinde baskı maliyet değerlerinin küçük çıkması, problemin çözümü için bu nöronların gerekli olduğunun göstergesidir. 5, 6, ..., 20 nöronlu mimarilerde ise durum beklenildiği gibi ortalama baskı maliyet değerleri, ortalama maliyet değerlerinden büyük çıkmıştır. Bu mimariler problemin çözümünde kullanılabilir. Ancak optimum çözüm için 4 nöron gerekli olduğundan 5 nöronlu mimari problemin çözümü için yeterli olacaktır. Üstelik [11]’de verilen birinci teoremin bir sonucu olarak 5 nöronlu modelin global minimum olduğunu ve dolayısıyla gizli katmanında 5 nöron olan modelin optimal nöron sayısı olduğunu söyleyebiliriz.

Son deneyde ise ikinci deneyde bir gizli katmanlı yapay sinir ağı modelleri için veri grubundaki 100 farklı veri ile hesaplanan ortalama maliyet değerleri için nöron sayısı üzerinden güven aralıkları hesaplanmıştır. Veri grubundaki verilerin nöron sayısına karşılık gelen ortalama maliyet değerleri dağılımı Şekil 15’te verilmiştir.



Şekil 15. Nöron sayısına karşılık gelen, 100 farklı veri için hesaplanan ortalama maliyet değerleri dağılımı

Nöron sayısı üzerinden hesaplanan alt güven aralığı sınırı ve üst güven aralığı sınırı Tablo 6'da verilmiştir.

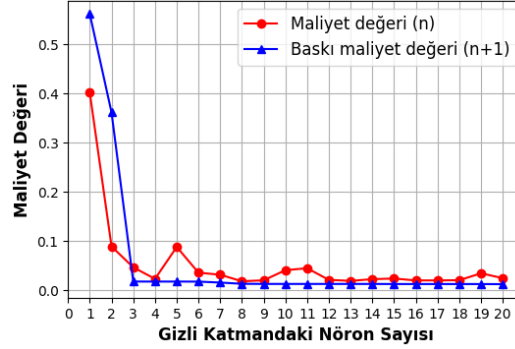
Tablo 6. Nöron sayısı üzerinden hesaplanan alt güven aralığı sınırı ve üst güven aralığı sınırı

Nöron Sayısı	Alt Güven Aralığı Sınırı	Üst Güven Aralığı Sınırı
1	0.544972	0.564106
2	0.345425	0.368547
3	0.129625	0.166490
4	0.031800	0.072486
5	0.009525	0.048095
6	0.005505	0.043446
7	0.004677	0.042007
8	0.004386	0.041487
9	0.004374	0.041452
10	0.004132	0.041073
11	0.004100	0.041017
12	0.004031	0.040890
13	0.004001	0.040803
14	0.003903	0.040668
15	0.003854	0.040616
16	0.003736	0.040539
17	0.003696	0.040482
18	0.003641	0.040435
19	0.003605	0.040383
20	0.003554	0.040387

Şekil 15 ve Tablo 6'dan alt güven aralığı ve üst güven aralıkları nöron sayısı arttıkça orantılı olarak azalmıştır.

4.1. XOR veri kümesi üzerinde test

Şekil 6’da verilen XOR veri kümesi burada eğitim kümesi olarak kullanılmıştır. Bu veri kümesi üzerinde 1, 2, ..., 20 nöronlu modeller eğitilmiştir ve önerilen yöntem test edilmiştir. Tablo 1 ve Tablo 2’deki veriler burada da kullanılmıştır. Ancak eğitimde sadece bir XOR veri kümesi ve bir başlangıç değerine karşılık gelen sonuçlar hesaplanmıştır. Bu sebeple optimum nöron sayısı bu veri kümesine ve kullanılan başlangıç değerine bağlıdır. Önerilen yöntem ile deney 3’te optimum nöron sayısı XOR için 5 nöronlu mimari olurken burada 4 nöronlu mimari önerilmektedir. Bunun nedeni veri kümesine ve başlangıç noktasına olan bağlılığından kaynaklanmaktadır. Şekil 16’da maliyet değerlerinin grafiği ve Tablo 7’de maliyet değerlerinin sayısal değerleri verilmiştir.

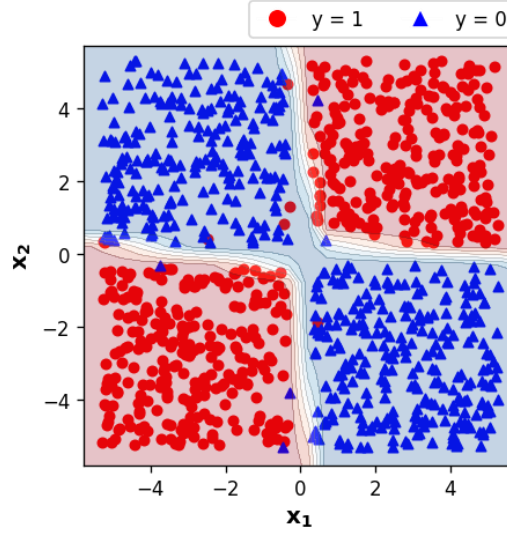


Şekil 16. Nöron sayısına karşılık gelen maliyet (n) ve baskı maliyet değerleri (n+1)

Tablo 7. Maliyet değerleri ve baskı maliyet değerleri

Gizli Katmandaki Nöron Sayısı	Maliyet Değerleri	Baskı Maliyet Değerleri
1	0.560893	-
2	0.361204	0.401004
3	0.017751	0.089012
4	0.017649	0.046597
5	0.017669	0.023210
6	0.017562	0.087965
7	0.015678	0.035753
8	0.013141	0.031550
9	0.012891	0.018297
10	0.012838	0.020212
11	0.012826	0.040783
12	0.012843	0.044798
13	0.012835	0.021141
14	0.012818	0.019065
15	0.012672	0.022561
16	0.012643	0.023850
17	0.012646	0.019966
18	0.012627	0.020055
19	0.012590	0.020463
20	0.012435	0.034497
21	-	0.024653

Gizli katmanında 4 nöron olan yapay sinir ağı modelinin parametreleri kullanılarak Şekil 17’de XOR veri kümesi sınıflandırılmıştır. Bu veri kümesi modeli test etmek için kullanılmıştır ve %99 başarı elde edilmiştir.



Şekil 17. Gizli katmanında 4 nöron olan yapay sinir ağının sınıflandırması

Maliyet fonksiyonu ile eğitilen 4 nöronlu modelin parametreleri

$$W^{[1]} = \begin{bmatrix} 3.262172 & 0.483003 \\ -0.496758 & -3.216276 \\ 1.703715 & -1.981785 \\ 2.437532 & -2.119808 \end{bmatrix}$$

$$W^{[2]} = [3.227747 \quad 3.170006 \quad -1.498600 \quad -2.89502]$$

$$b^{[1]} = \begin{bmatrix} -0.314100 \\ 0.031322 \\ -0.285429 \\ -0.588046 \end{bmatrix}$$

$$b^{[2]} = [-4.532933]$$

olarak elde edilmiştir. Baskı maliyet fonksiyonu kullanılarak 5 nöronlu modelin parametreleri

$$W^{[1]} = \begin{bmatrix} 3.280773 & 0.487982 \\ -0.493251 & -3.185521 \\ 1.769404 & -1.961606 \\ 2.331545 & -1.952088 \\ -0.000388 & -0.002557 \end{bmatrix}$$

$$W^{[2]} = [3.216438 \quad 3.134840 \quad -1.765270 \quad -2.796156 \quad 0.002598]$$

$$b^{[1]} = \begin{bmatrix} -0.330115 \\ 0.047822 \\ -0.289385 \\ -0.592361 \\ 0.000072 \end{bmatrix}$$

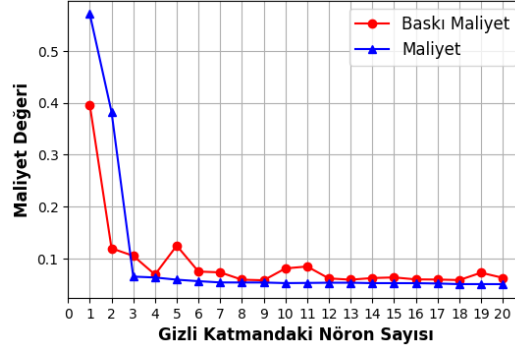
$$b^{[2]} = [-4.510361]$$

elde edilmiştir. Gizli katmanında 5 nöron bulunduran modelin son nöronu baskılanmıştır. Son nörona bağlı olan $W^{[1]}$, $W^{[2]}$ ve $b^{[1]}$ parametreleri sıfıra giderken diğer parametreler gizli katmanında 4 nöron olan modelin parametrelerine yaklaşmıştır.

Buradan 4 nöronlu modele eklenen 5. nöronun gereksiz olduğu açıktır. Modelin temsil gücünün gelişmesine katkıda bulunmamıştır. O halde 4 nöronlu modelin global minimumu verdiği söylenebilir.

4.2. Daire veri kümesi üzerinde test

Şekil 7’de verilen daire veri kümesi burada eğitim kümesi olarak kullanılmıştır. Bu veri kümesi üzerinde 1, 2, ..., 20 modeller eğitilmiştir ve önerilen yöntem test edilmiştir. Tablo 1 ve Tablo 2’deki veriler burada da kullanılmıştır. Daire veri kümesi için yapılan testte de bir veri kümesi ve bir başlangıç değeri ile model eğitilmiştir. Bu sebeple veri kümesine ve başlangıç değerine bağlı olarak optimal nöron tespit edilmiştir. Şekil 18’de maliyet değerlerinin grafiği ve Tablo 8’de maliyet değerlerinin sayısal değerleri verilmiştir.



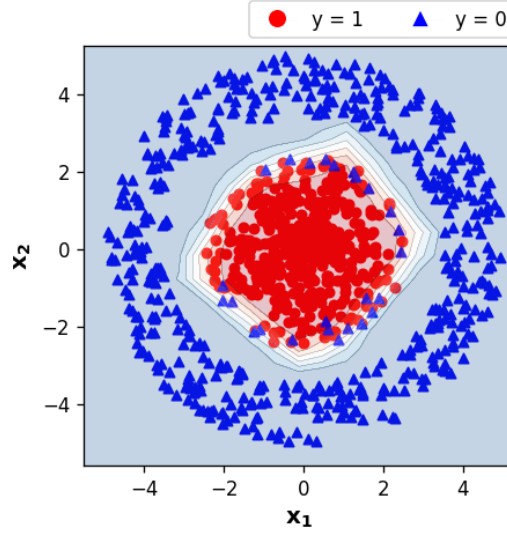
Şekil 18. Nöron sayısına karşılık gelen maliyet (n) ve baskı maliyet değerleri (n+1)

Bu veri kümesi için de gizli katmanında 4 nöron olan model optimum nöron sayısı olarak önerilmektedir.

Tablo 8. Maliyet değerleri ve baskı maliyet değerleri

Gizli Katmandaki Nöron Sayısı	Maliyet Değerleri	Baskı Maliyet Değerleri
1	0.570705	-
2	0.382172	0.395659
3	0.064879	0.119366
4	0.063163	0.104905
5	0.058936	0.068805
6	0.056069	0.124497
7	0.053646	0.074630
8	0.053523	0.073046
9	0.053511	0.058926
10	0.052366	0.057806
11	0.052685	0.080656
12	0.053099	0.084372
13	0.053098	0.061396
14	0.052303	0.058929
15	0.052294	0.062124
16	0.052225	0.063328
17	0.051613	0.059521
18	0.050383	0.059236
19	0.050429	0.058227
20	0.050349	0.072363
21	-	0.062576

Gizli katmanında 4 nöron olan yapay sinir ağı modelinin parametreleri kullanılarak Şekil 19’da daire veri kümesi sınıflandırılmıştır. Bu veri kümesi modeli test etmek için kullanılmıştır ve %97 başarı elde edilmiştir.



Şekil 19. Gizli katmanında 4 nöron olan yapay sinir ağının sınıflandırması

Maliyet fonksiyonu ile eğitilen 4 nöronlu modelin parametreleri

$$W^{[1]} = \begin{bmatrix} 0.387074 & -1.159506 \\ -1.705916 & -0.995456 \\ 0.412283 & -3.152991 \\ 2.443600 & -0.581224 \end{bmatrix}$$

$$W^{[2]} = [3.841179 \quad -1.887189 \quad -2.247470 \quad -2.315851]$$

$$b^{[1]} = \begin{bmatrix} 3.717550 \\ -0.841437 \\ -0.608513 \\ -1.217082 \end{bmatrix}$$

$$b^{[2]} = [-3.578421]$$

olarak elde edilmiştir. Baskı maliyet fonksiyonu kullanılarak 5 nöronlu modelin parametreleri

$$W^{[1]} = \begin{bmatrix} 0.380880 & -1.148764 \\ -1.707935 & -0.952500 \\ 0.410940 & -3.158596 \\ 2.423199 & -0.553552 \\ 0.000208 & -0.000716 \end{bmatrix}$$

$$W^{[2]} = [3.810578 \quad -1.901815 \quad -2.248367 \quad -2.32157 \quad 0.002505]$$

$$b^{[1]} = \begin{bmatrix} 3.687286 \\ -0.916302 \\ -0.631946 \\ -1.281906 \\ 0.002340 \end{bmatrix}$$

$$b^{[2]} = [-3.525128]$$

elde edilmiştir. Gizli katmanında 5 nöron bulunduran modelin son nöronu baskılanmıştır. Son nörona bağlı olan $W^{[1]}$, $W^{[2]}$ ve $b^{[1]}$ parametreleri sıfıra giderken diğer parametreler gizli katmanında 4 nöron olan modelin parametrelerine yaklaşmıştır.

Buradan 4 nöronlu modele eklenen 5. nöronun gereksiz olduğu açıktır. Modelin temsil gücünün gelişmesine katkıda bulunmamıştır. O halde gizli katmanında 4 nöron olan modelin global minimumu verdiğini söyleyebiliriz.

5. Sonuç ve Yorum

Bu çalışmada, bir gizli katmanlı yapay sinir ağları için optimal nöron sayısını veren bir yöntem önerilmiştir. Burada tanımlanan baskı maliyet fonksiyonuyla sistematik bir şekilde optimal nöron sayısı bulunmuştur. XOR veri kümesi ve daire veri kümesi üzerinde test edilmiş ve doğrulanmıştır.

Önerilen yöntemi daha büyük yapay sinir ağlarına, çoklu sınıflandırma problemlerine, CNN veya Transformer gibi farklı mimarilerdeki yapay sinir ağlarına adapte etmek bir sonraki çalışmanın konusu olacaktır. Yine burada kullanılan benzer bir mantıkla konveks olmayan bir fonksiyonun yapay sinir ağları kullanılarak global minimum araştırılacaktır. Örneğin elimizde fonksiyonun $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ olacak şekilde m tane noktası olsun. Bu çalışmada kullanılan yöntem ile verilen noktalar kullanılarak yapay sinir ağları ile konveks olmayan bir fonksiyon için global minimum bulunabilir. Bu anlamda yapay sinir ağlarını global optimizasyon problemine uygulamak ileride yapılacak olan diğer bir çalışma olacaktır.

Araştırmacıların Katkı Oranı Beyanı

Tayfun Ünal: Metodoloji, Yazılım, Araştırma, İnceleme ve Düzenleme, Görselleştirme

Ünver Çiftçi: Metodoloji, Yazılım, Araştırma, Doğrulama, Denetim/Gözlem/Tavsiye

Nurkut Nuray Urgan: Araştırma, İnceleme ve Düzenleme, Doğrulama, Denetim/Gözlem/Tavsiye

Destek ve Teşekkür Beyanı

Bu çalışmanın yazarları olarak herhangi bir destek ve teşekkür beyanımız bulunmadığını bildiririz.

Çatışma Beyanı

Bu çalışmanın yazarları olarak herhangi bir çatışma beyanımız bulunmadığını bildiririz.

Etik Kurul Onayı ve/veya Aydınlatılmış Onam Bilgileri

Bu çalışmanın yazarları olarak herhangi bir etik kurul onayı ve/veya aydınlatılmış onam bilgileri beyanımız bulunmadığını bildiririz.

Kaynakça

- [1] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, Cambridge, 2016.
- [2] M. Şahan ve Y. Okur, “Akdeniz bölgesine ait meteorolojik veriler kullanılarak yapay sinir ağları yardımıyla güneş enerjisinin tahmini,” *Süleyman Demirel Üniversitesi Fen Edeb. Fakültesi Fen Derg.*, 11 (1), 61–71, 2016.
- [3] M. Şahan, “Yapay sinir ağları ve angström-prescott denklemleri kullanılarak Gaziantep, Antakya ve Kahramanmaraş için global güneş radyasyonu tahmini,” *Süleyman Demirel Üniversitesi Fen Edeb. Fakültesi Fen Derg.*, 16 (2), 368–384, 2021.
- [4] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, “Dive into deep learning,” *arXiv*, 2020.
- [5] C. F. Higham and D. J. Higham, “Deep learning: an introduction for applied mathematicians,” *SIAM Rev.*, 61 (4), 860–891, 2019.
- [6] R. Vidal, J. Bruna, R. Giryes and S. Soatto, “Mathematics of deep learning,” *arXiv*, 2017.
- [7] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, Cambridge, 2022.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, New York 2021.
- [9] F. Chollet, *Deep Learning with Python*. 2nd Ed. Manning Publications, 2017.
- [10] A. Géron, *Hands-On Machine Learning with Scikit-Learn*. O'Reilly, USA, 2017.
- [11] B. D. Haeffele and R. Vidal, “Global optimality in neural network training,” *CVPR*, 2 (3), 4390–4398, 2017.
- [12] R. Setiono, “A penalty-function approach for pruning feedforward neural networks,” *Neural Comput.*, 9 (1), 185–204, 1997.
- [13] Z. Zhang and J. Qiao, “A node pruning algorithm for feedforward neural network based on neural

- complexity,” *Proc. 2010 Int. Conf. Intell. Control Inf. Process.*, 1, 406–410, 2010.
- [14] M. M. Bejani and M. Ghatee, “A systematic review on overfitting control in shallow and deep neural networks,” *Artif. Intell. Rev.*, 54 (8), 6391–6438, 2021.
- [15] X. Wu, P. Rózycki and B. M. Wilamowski, “A hybrid constructive algorithm for single-layer feedforward networks learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, 26, 1659–1668, 2015.
- [16] J. Qiao, F. Li, H. Han and W. Li, “Constructive algorithm for fully connected cascade feedforward neural networks,” *Neurocomputing*, 182, 154–164, 2016.
- [17] Y. Bengio, N. L. Roux, P. Vincent, O. Delalleau and P. Marcotte, “Convex neural networks,” *Adv. Neural Inf. Process. Syst.*, 123–130, 2005.
- [18] C. L. P. Chen and Z. Liu, “Broad learning system: an effective and efficient incremental learning system without the need for deep architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, 29 (1), 10–24, 2018.
- [19] W. J. Puma-Villanueva, E. P. dos Santos and F. J. Von Zuben, “A constructive algorithm to synthesize arbitrarily connected feedforward neural networks,” *Neurocomputing*, 75 (1), 14–32, 2012.
- [20] J. L. Subirats, L. Franco and J. M. Jerez, “C-Mantec: a novel constructive neural network algorithm incorporating competition between neurons,” *Neural Netw.*, 26, 130–140, 2012.
- [21] G. M. Augasta and T. Kathirvalavakumar, “A novel pruning algorithm for optimizing feedforward neural network of classification problems,” *Neural Process. Lett.*, 34, 241–258, 2011.
- [22] P. Molchanov, A. Mallya, S. Tyree, I. Frosio and J. Kautz, “Importance estimation for neural network pruning,” *CVPR*, 11256–11264, 2019.
- [23] G. Castellano, A. M. Fanelli and M. Pelillo, “An iterative pruning algorithm for feedforward neural networks,” *IEEE Trans. Neural Netw.*, 8, 519–531, 1997.
- [24] Q. Chang, J. Wang, H. Zhang, L. Shi, J. Wang and N. R. Pal, “Structure optimization of neural networks with l1 regularization on gates,” *Comput. Intell.*, 196–203, 2019.
- [25] J. F. Qiao, Y. Zhang and H. G. Han, “Fast unit pruning algorithm for feedforward neural network design,” *Appl. Math. Comput.*, 205, 622–627, 2008.
- [26] H. Z. Alemu, W. Wu and J. Zhao, “Feedforward neural networks with a hidden layer regularization method,” *Symmetry (Basel)*, 10, 2018.
- [27] A. Bondarenko, A. Borisov and L. Aleksejeva, “Neurons vs weights pruning in artificial neural networks,” *Vide. Tehnol. Resur. - Environ. Technol. Resour.*, 3, 22–28, 2015.
- [28] R. Reed, “Pruning Algorithms—a survey,” *IEEE Trans. Neural Netw.*, 4 (5), 740–747, 1993.
- [29] B. Hassibi, D. G. Stork and G. J. Wolff, “Optimal brain surgeon and general network pruning,” *IEEE International Conference on Neural Networks*, 1, 293–299, 1993.
- [30] X. Xie, H. Zhang, J. Wang, Q. Chang, J. Wang and N. R. Pal, “Learning optimized structure of neural networks by hidden node pruning with l1 regularization,” *IEEE Trans. Cybern.*, 50 (3), 1333–1346, 2020.
- [31] O. Aran, O. T. Yildiz and E. Alpaydin, “An incremental framework based on cross-validation for estimating the architecture of a multilayer perceptron,” *Int. J. Pattern Recognit. Artif. Intell.*, 23 (2), 159–190, 2009.
- [32] H. G. Han, S. Zhang, ve J. F. Qiao, “An adaptive growing and pruning algorithm for designing recurrent neural network,” *Neurocomputing*, 242, 51–62, 2017.
- [33] R. Zemouri, N. Omri, F. Fnaiech, N. Zerhouni, ve N. Fnaiech, “A new growing pruning deep learning neural network algorithm (GP-DLNN),” *Neural Comput. Appl.*, 32, 18143–18159, 2020.
- [34] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang and E. Choi, “MorphNet: fast & simple resource-constrained structure learning of deep networks,” *CVPR*, 1586-1595, 2018.
- [35] K. Kawaguchi and L. P. Kaelbling, “Elimination of all bad local minima in deep learning,” *arXiv*, 2019.
- [36] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 2009.
- [37] R. Reed and R. J. Marks II, *Neural Smothing Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, Cambridge, 2016.
- [38] E. Apaydin, *Introduction to Machine Learning*. 3rd Ed., MIT Press, Cambridge, 2014.