

DERİN AĞLAR İÇİN YENİ BİR BİRİMDİK DÜZGÜNLEŞTİRME YAKLAŞIMI

Kazım FIRILDAK^{1*}, Gaffari ÇELİK², Muhammed Fatih TALU³

¹Fırat Üniversitesi, Kovancılar Mektek Yüksek Okulu, Bilgisayar Teknolojileri Bölümü, Elazığ, 23100, Türkiye

²Ağrı İbrahim Çeçen Üniversitesi, Meslek Yüksek Okulu, Bilgisayar Teknolojileri Bölümü, Ağrı, 04100, Türkiye

³İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya, 44200, Türkiye

Geliş Tarihi/Received Date: 14.11.2023 Kabul Tarihi/Accepted Date: 24.04.2024 DOI: 10.54365/adyumbd.1390894

ÖZET

Ortogonal Düzgünleştirme (OD), derin ağların aşırı öğrenme (overfitting), gradyan patlamansı/kaybolmasını engellemek için kullanılmaktadır. Literatürde derin öğrenme için geliştirilen OD yöntemlerinin çoğunda ağ ağırlıklarını birim dik vektörler olarak öğrenme amaçlanmaktadır. Bu makalede ağ ağırlıklarını ikili olarak gruplayarak birim dik öğrenmeye zorlayan fonksiyon, maliyet fonksiyonuna eklenmektedir. Bu yöntem yapay sinir ağlarında ve konvülsiyonel sinir ağlarında çeşitli veri kümelerinde (yapay veri ve gerçek veri) test edilmektedir. Ayrıca önerilen yöntem, literatürde öne çıkan Yumuşak Ortogonal (SO), Çift Yumuşak Ortogonal (DSO), Karşılıklı Tutarlılık (MC) ve Spektral Sınırlı İzometri Özellikli (SRIP) gibi yöntemler ile doğruluk, yürütülme zamanı, hata oranı metriklerinde karşılaştırılmaktadır. Karşılaştırma sonucunda doğruluk metriğinde farklı veri kümelerinin kullanan ağlarda %1-%5 arasında iyileşme sağlanmaktadır. Önerilen yöntem, Cifar10 veri kümesinde Resnet 110 ağında 92,96 dan %93,90'a ve Resnet 28-10 %95,84'den %96,78'a test başarısını yükseltmektedir.

Anahtar Kelimeler: Ortogonal Düzgünleştirme, Sinir Ağları, Evrimsel Sinir Ağları

A NEW ORTOGONAL REGULARIZATION APPROACH FOR DEEP NETWORKS

ABSTRACT

Orthogonal Regularization (OR) is used to prevent overfitting, gradient explosion and Vanishing Gradient in deep networks. OR methods developed for deep learning in the literature aim to learn network weights as unit orthogonal vectors. In this article, the function that enables unit orthogonal learning of network weights for binary groups is added to the cost function. This method is tested on various data sets (artificial data and real data) in artificial neural networks and convolutional neural networks. In addition, the proposed method is compared with methods such as Soft Orthogonal (SO), Double Soft Orthogonal (DSO), Mutual Coherence (MC) and Spectral Restricted Isometry Property (SRIP), which are prominent in the literature, in terms of accuracy, execution time and error rate metrics. As a result of the comparison, an improvement of 1% to 5% is achieved in the accuracy metric in networks using different data sets. The proposed method increases the test success from 92.96 to 93.90% in the Resnet 110 network and from 95.84% to 96.78% in the Resnet 28-10 dataset on the Cifar10 dataset.

Keywords: Orthogonal Regularization, Neural Networks, Convolutional Neural Networks

*e-posta¹: kfirildak@firat.edu.tr ORCID ID: <https://orcid.org/my-orcid?orcid=0000-0002-1958-3627> (Sorumlu Yazar)

e-posta²: gcelik@agri.edu.tr ORCID ID: <https://orcid.org/my-orcid?orcid=0000-0001-5658-9529>

e-posta³: fatihaltu@gmail.com ORCID ID: <https://orcid.org/my-orcid?orcid=0000-0003-1166-8404>

1. Giriş

Yapay sinir ağları (YSA), önemli bir yapay öğrenme mimarisidir ve derin öğrenme mimarilerinin temelini oluşturmaktadır [1–5]. Derin mimariler imge sınıflama [6–9], doğal dil işleme [10] ve nesne tespiti [11] gibi birçok uygulama alanında kullanılmaktadır. Yapay öğrenme araştırmacıları, derin mimarilerin optimizasyonu üzerinde son yıllarda yoğun bir şekilde çalışmaktadırlar. Bu çalışmalar, yeni derin mimarilerin önerilmesi [1,12] optimizasyon algoritmalarının geliştirilmesi [13,14] düzgünleştirme yöntemleri [15,16] olarak özetlenebilir.

YSA, eğitim maliyetinin azaltılması, doğruluk metriğinin yükseltilmesi derin ağların optimizasyonu için büyük önem taşımaktadır. YSA optimizasyonunun 2 temel zorluğu vardır. Bunların ilki ara katmanların ağırlık sayısının ne olacağı ve öğrenme hızının nasıl artırılacağıdır [17]. Bunun yanında Ann'lerin de bir parçası olduğu derin öğrenme mimarilerinin gradyan patlaması/kaybolması çözülmesi gereken bir diğer problemdir [18]. Bu durum genellikle bir problemin çözümünde daha derin sinir ağlarının tercih edilmesinden kaynaklanmaktadır. Problemin çözümü için artık derin ağ önerilmiştir [19,20]. Bu ağlar probleme kısmi bir çözüm üretse de ağ derinliği artması durumunda tam bir çözüm sağlayamamıştır. Bunun yanında ağ ağırlıklarının, ortagol başlatılması ve ya eğitiminde ortogonal düzenlenmesi gibi yöntemler problemin daha etkili bir çözümü olarak kullanılmaktadır [21–23].

Bu makalede YSA ve evrişimsel sinir ağları (ESA), ağırlık vektörlerini ortogonal tabanlar olarak öğrenmeye dayalı bir regularisasyon metodu önerilmektedir. Önerilen yöntem ile ağın temel yapısı değiştirilmeden etkili bir hata eklenti metodu sunulmaktadır. Yöntem ağın ara katmanındaki ağırlıklarını ikili ve atlamalı olarak dikleştirmeye zorlanmaktadır. Böylece YSA ve ESA, ortogonal ağırlıkları etkili bir şekilde öğrenmesi sağlanmaktadır. Önerilen metod katman sayısı düşük YSA ve ara katman sayısı daha fazla olan ESA nasıl davrandığı analiz edilmektedir. Analiz aşamasından sentetik veriler ve gerçek veriler üzerinde deneyler yapılmaktadır. Ağlara önce gürültüsüz veriler daha sonra ise verilere gaus dağılımlı gürültü eklenerek deneyler gerçekleştirilmektedir. Daha sonra önerilen yöntemin derin ağlarda performansı incelenmektedir. Önerilen yöntem, Resnet ağının sınıflama katmanında kullanılarak ağın hızlı yakınsadığı ve daha iyi test doğruluğu sağladığı gözlemlenmektedir. Önerilen yöntem YSA ve ESA toplam doğruluğu ve öğrenme hızında iyileşme sağlanmaktadır. Son olarak literatürde sunulan diğer ortogonal düzgünleştirme yöntemleri ile kıyaslanmaktadır.

Bu çalışmanın 2. bölümünde literatürde sunulan ilgili çalışmalar sunulmaktadır. 3 bölümde ortogonalite ve literatürdeki ortogonal yöntemler açıklanmaktadır. Daha sonra 4. Bölümde önerilen ortogonal ağırlık öğrenen yöntem tanımlanmaktadır. 5 bölümden önerilen yöntemin deneysel sonuçları sunulmaktadır. Son olarak elde edilen sonuçlar tartışılarak ileriki çalışmalar için düşünce ve öneriler belirtilmektedir.

2. İlgili Çalışmalar

Literatürde ortogonal ağırlıkların yapay öğrenmede kullanımı ile ilgili birçok çalışma geliştirilmiştir. Mishkin ve diğ. [21] tarafından derin sinir ağları (DSA) için Katman sıralı birim varyansı başlangıç metodu önerilmiştir. Bu yaklaşımda ilk olarak ağ ağırlık matrisi ortanormal olarak başlatılmıştır. Sonraki aşamada ise ağın her bir katmanı normalleştirmek için çıkış varyansını 1'e eşitlenmiştir. Böylece DSA toplam test doğruluğu arttırılmıştır. Huang ve diğ. [14] Orthogonal ağırlık matrisi normalisasyonu için çoklu stiefel manifolds optimizasyon çözümünü kullanmışlardır. Böylece yöntem karesel ve karesel olmayan ağırlık matrisi için geliştirilmiş bir ortogonal çözüm sağlanmıştır. Metot Cifar100 veri kümesinde geniş residual ağın hatasını yaklaşık %2 oranında düşürmektedir. Jia ve diğ. [24] DSA'nın, veri dağılımlarının local izometri olduklarını önermektedir. Daha sonra local izometri kullanarak her bir ağırlık matrisinin tekil değer spectrumuna ve aralığına duyarlı bir hata fonksiyonu geliştirmişlerdir. Bu metod modern derin ağlara kullanarak çeşitli metriklerde

başarısı kanıtlanmıştır. Zhang ve diğ tarafından [25] ise DSA ağ ağırlıkları arasındaki açılarının ortalamasının ve varyansının 0 dan 90 doğru iten kendi kendine düzgünleştirme yapma metodu kullanmışlardır. Böylece ağ eğitiminin daha hızlı yakınsaması sağlanmıştır. Bansal ve diğ. [26] tarafından karşılıklı tutarlılık ve kısıtlı izometri özelliği gibi çeşitli liner cebir metotları kullanarak derin ESA'ların eğitimi konusunda yeni bir ortagonallik yöntemi geliştirilmiştir. Daha sonra yöntem Resnet ve WideResnet ağlarında kullanılarak Cifar10,Cifar100 ve Imagenet veri kümeleri üzerinde test edilmiştir. Xie ve diğ. [27] DSA'lerin gradyan patlamasını/kaybolmasını engellemek için farklı filtre bankaları arasındaki ortanormallik kullanan bir düzgünleştirme yöntemi önermişlerdir. Bu yöntem iki parametrik katman arasındaki kuasizometri varsayım geliştirmiştir.

Ortogonal yaklaşımların kullanıldığı birçok yöntem güncel hayat problemlerini çözmüştür. Hava kirliliği tahmini gerçekleştirmek için derin Spatio-Temporal Ortogonal Düzgünleştirme kullanan artık ESA geliştirilmiştir [28]. Bu çalışmada ESA doğruluğunun artırılması için ortogonal düzgünleştirme yöntemi olan spektral Sınırlı İzometri özellikli kullanılmıştır. Sonuç olarak bu yönteminin nerdeyse bütün durumlarda ağırlık tahmin doğruluğunu artırmıştır. Bir diğer çalışmada EEG sinyallerini kullanarak uyku analizi yapmak için ortogonal tekrarlamalı yapay sinir ağı (OTNN) önerilmiştir [29]. OTNN, literatürdeki TNN yapılarıyla karşılaştırıldığında F1-score, toplam doğruluk ve AUC değerinde daha iyi performans sağlamıştır. Benzer olarak tek kanallı EEG sinyali kullanarak uyku aşaması sınıflandırmak için ortogonal konvülsiyonel yapay sinir ağı (OESA) yapısı geliştirilmiştir [30]. Bu yöntemde, EEG sinyalleri Hilbert-Huang dönüşümü aracılığıyla bir zaman frekans domainine dönüştürülmüştür. Daha sonra OESA ağırlıkları ortogonal olarak başlatılmıştır. OESA eğitim sırasında ortogonal düzgünleştirme kullanarak daha etkili bir sınıflama gerçekleştirmiştir. Bunun yanında ortogonal derin bir model kara kutu saldırılarına karşı koruma önlemi olarak önerilmiştir. Ortogonal model yardımıyla, gradyan kontrolü sağlanarak kara kutu saldırılarından daha güvenli bir sistem tasarımı sağlanmıştır [31].

3. Ortogonal Düzgünleştirme

DSA, performansını etkileyen en önemli etkenlerden biri gradyan patlaması/kaybolmasıdır. Ağların derinliğindeki artış bu problemleri ortaya çıkarmaktadır. DSA'da, ileriye doğru yayılımın kararlılığı yığın normalleştirme (YN) yöntemi ile sağlanmaktadır. Fakat geri yayılımda YN hatanın düzenli dağılımını garanti etmemektedir [28]. Bu nedenle gradyanların patlaması/kaybolmasını önlemek için OD kullanılması önerilmektedir [26–28]. OD, geri yayılım hatalarını azalttığı, ara katman enerji korunumu sağladığı ve ağların doğruluk metriğini arttırdığı literatürde sunulmuştur [26,27,32]. Bu çalışmalardan, OD genellikle ağırlık zayıflama metotlarının yerine kullanılmaktadır.

Ortogonalite, x ve y iki vektör olmak üzere $x, y \in R^n$ için $x \cdot y = 0$ ise $x \perp y$ denir. Bunun yanında dik olan x ve y vektörlerinin uzunlukları $\|x\| = 1$ ve $\|y\| = 1$ ise bu vektörler ortanormal olarak ifade edilmektedir. YSA'da ortogonalite ve ortanormallik kavramı, norm-korunumu olarak adlandırılan denklem 1' de gösterilen hipotez ile açıklanmaktadır. Norm-korunumunda, x ve y vektörü liner bir dönüşüm ile W ortogonal matrisi kullanarak eşleştirilir. W^T , transpoz matrisi olmak üzere $y = W^T x$ denkleminde $\|x\| = \|y\|$ ise bu norm-korunumu olarak tanımlanmaktadır [27].

$$\|y\| = \sqrt{y^T y} = \sqrt{x^T W W^T x} = \sqrt{x^T x} = \|x\| \text{ if } W W^T = I \quad (1)$$

Literatürdeki norm-korunumu kullanan OD yöntemleri Bansal ve diğ tarafından dört grupta incelenmektedir. Bunlar yumuşak ortogonal düzgünleştirme, çift yumuşak ortogonal düzgünleştirme, karşılıklı tutarlılık düzgünleştirme ve spektral sınırlı izometri özellikli düzgünleştirme yöntemleridir.

3.1. Yumuşak Ortogonal Düzgünleştirme

Literatürde birçok çalışma, aşırı öğrenme, gradyan patlaması/ kaybolması probleminin çözüm olarak ağırlık matrisinin gram matrisine yakın olmasını öne sürmektedir. Bu çalışmalarda ağırlık matrisinin ortogonal olması için denklem 2’deki düzgünleştirme yaklaşımı kullanılmışlardır [27,33].

$$\lambda \|W^T W - I\|_F^2 \quad (2)$$

Burada λ , düzgünleştirme katsatısı, W , ağırlık matrisi ve I ise birim matristir. $W^T W$ karesel matrisinin birim matrisden farkının Frobenius norm değeri λ ile çarpılarak ağ hata fonksiyonuna eklenmektedir. W , satır ve sütun sayısı sırasıyla m , n dir. Eğer $m \geq n$ ise m rank olan W matrisi, undercompletedir. Bu durumda W matrisi sütunları arasında ortogonality tanımlanabilir. Fakat $n \geq m$ ise W matrisi m rank olduğundan overcompletedir. Bu durumda gram matrisi $W^T W \in R^{n \times n}$ tanımlanamayabilir. Bu eksiklikleri giderebilmek için ağırlık matrisini stiefel manifold, jakobi gibi alt uzaylara bölen yaklaşımlar sunulmuştur [14].

Denelerimizde denklem 2 deki temel soft ortogonallik yaklaşımı kullanılarak YSA ve ESA regularize edilmektedir. Denklem 2 deki formülün gradyanı $\lambda \|W^T W - I\|_F^2$ kullanarak hata geri yayılımı gerçekleştirilmiştir.

3.2. Çift Yumuşak Ortogonal Düzgünleştirme

Çift yumuşak ortogonal düzgünleştirme (DSO), SO metodundan geliştirilmektedir. SO yaklaşımındaki aşırı öğrenme ve aşırı uyum durumlarına daha iyi adapte olmak için Denklem 3’te verilen düzgünleştirme önerilmektedir [26].

$$\lambda (\|W^T W - I\|_F^2 + \|W W^T - I\|_F^2) \quad (3)$$

Burada W , ağırlık matrisi için m satır ve n sütun sayısıdır. Yöntem, $m > n$ için $\lambda \|W^T W - I\|_F^2$; $m \leq n$ için ise $\lambda \|W W^T - I\|_F^2$ denklemine göre düzgünleştirme kaybı hesaplanmaktadır. Daha sonra düzgünleştirme kaybı ağ hata fonksiyonuna eklenmektedir. DSO düzgünleştirme yönteminin ağ hatası, otomatik türev kullanılarak geriye doğru yayılmaktadır.

3.3. Karşılıklı Tutarlılık Düzgünleştirme

W , ağırlık matrisinin aşırı öğrenme ve aşırı uyum durumlarına çözüm üretmek için karşılıklı tutarlılık yaklaşımı önerilmektedir. Bu yöntem denklem 4’te gösterilmektedir. [26,34].

$$\mu_w = \max_{i \neq j} \frac{|(w_i, w_j)|}{\|w_i\| + \|w_j\|} \quad (4)$$

Burada, w_i , W matrisinin i . sütun vektörüdür. MC, $[0,1]$ aralığında bir değerdir. Denklem 4’teki μ_w değerinin düşük olması W matrisinin, ortogonal veya yakın-ortogonal olmasına neden olmaktadır. Bu nedenle μ_w değeri OD için kullanılmaktadır. Denklem 5’te MC yöntemi gösterilmektedir.

$$\lambda \|W^t W - I\|_\infty \quad (5)$$

Denklem 5’teki denklemin gradyanını hesaplamak için l_∞ norm kullanılmaktadır [35]. Vektörel uzunluk hesabının iteratif olarak gerçekleşmesinden dolayı, MC yönteminin gradyanı otomatik-türev ile hesaplanmaktadır.

3.4. Spektral Sınırlı İzometri Özellikli Düzgünleştirme

Sınırlı izometrik özellikler (RIP) kullanılarak geliştirilen düzgünleştirme yöntemleri zamansal karmaşıklığı ve doğruluk metriklerinde diğer yöntemlere göre daha iyi performans gösterdiği literatürde gösterilmektedir [25,28]. RIP, matematiksel modeli denklem 6’te verilmektedir.

$$(1 - S_w) \leq \frac{\|Wz\|^2}{\|z\|^2} < (1 + S_w) \quad (6)$$

Burada, $\forall z \in R^n$ vektörleri için k-sparse’ da $S_w \in (0,1)$ sabit değeri vardır. Denklem 6’ da $k=n$ için W ağırlık matrisi ortagonallığe zorlanmaktadır. Denklem 7’de, $k = n$ için RIP tekrar düzenlenerek gösterilmektedir.

$$\left| \frac{\|Wz\|^2}{\|z\|^2} - 1 \right| \leq S_w, \forall z \in R^n \quad (7)$$

Denklem 7’ye göre W ağırlık matrisinin ortogonalitesi, S_w değerine bağlıdır. Öyleyse S_w , minimize edilmesi, W matrisinin ortagonelliğe yaklaştırmaktadır. Öyleyse $W^T W - I$ spectral normunun minimize edilmesi gerekmektedir. Buna spektral sınırlı izometrik özellik (SRIP) denilmektedir. Denklem 8’ de SRIP yönteminin matematiksel gösterimi sunulmaktadır [28,36].

$$\lambda \cdot \sigma(W^T W - I) \quad (8)$$

Burada, λ düzgünleştirme katsayısıdır. $\sigma(W^T W - I)$, $W^T W - I$ matrisinin spektral normudur. Spektral norm denklem 8’ in gradyan hesaplamasını kolaylaştırmak için iteratif bir düzenleme denklem 9’ da verilmektedir [26].

$$u \leftarrow (W^T W - I)v, v \leftarrow (W^T W - I)u, \sigma(W^T W - I) \leftarrow \frac{\|v\|}{\|u\|} \quad (9)$$

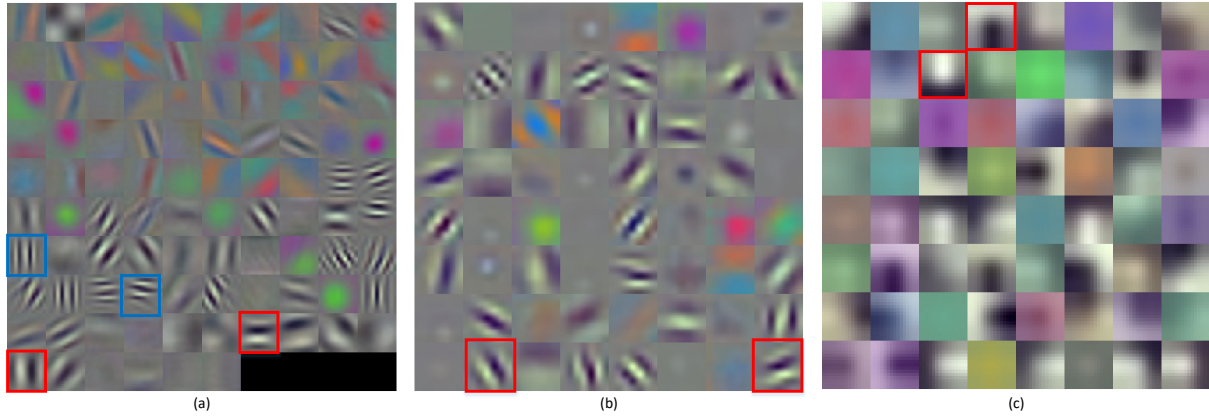
Burada $v \in R^n$ için rastgele bir değerdir. İteratif olarak gradyan yaklaşık bir değerde hesaplanmaktadır. Gradyan hesaplanmasında otomatik türev kullanılmaktadır.

4. Önerilen Ortogonal Düzgünleştirme Yöntemi

Literatürdeki OD algoritmalarının çoğu, ağırlık matrisini gram matrisine yakınlaştırmayı amaçlamaktadır. Bu durum aşırı öğrenme problemini ortaya çıkarmaktadır. Bu çalışmada aşırı öğrenme, probleminden kaçınmak için YSA’ların ağırlıkları 2’li vektör çiftleri olarak düzgünleştirilmesi amaçlanmaktadır. Bu amaçla ön eğitim modelleri verilen Alexnet [2], Vgg16 [37], Resnet50 [19] derin öğrenme ağlarının ağırlıkları incelenmektedir. Bu ağırlıklar incelendiğinde ESA’nın filtrelerin ve Ann tam bağlı ağırlıklarının bir bölümü dik vektörler şeklinde öğrenildiği gözlemlenmektedir [25]. Şekil 1’de, Alexnet, Resnet50, Vgg16 ağlarının ön eğitim modelinden rastgele alınmış evrişim katman filtreleri gösterilmektedir.

Şekil 1’de AlexNET, Resnet50 ve Vgg16 ön eğitim ağırlık çiftlerinin bazıları (Kırmızı ve mavi kutular ile gösterilen) ortagonal ve ya ortagonelliğe yakın filtrelerdir. Yani, bir evrişim katmanındaki ağırlıkların sadece belirli çiftleri birbirine diktir. Öyleyse ağırlık matrisinin tamamı yerine belirli ikililerin ortagonelliğinin hesaplanması aşırı öğrenme probleminin çözümü için etkili bir yöntem olabilir. Denklem 10’ da 2’li vektörlerin dikliğini öğrenen önerilen yeni bir düzgünleştirme yönteminin denklemi sunulmuştur.

$$\lambda \sum_{j=1}^E \sum_{i \in \{1,3,5,\dots\}}^D \|w_{j,i}^T w_{j,i+1} - I\|^2 \quad (10)$$



Şekil 1. ESA ağlarının bazı evrişim filtreleri (a) AlexNET evrişim filtreleri, (b) Resnet50 evrişim filtreleri, (c) Vgg16 evrişim filtreleri

Burada, λ düzgünleştirme katsayısıdır. D, j . katmandaki toplam evrişim filtre sayısıdır ve tam bağlı sinir ağlar da ise ara katmandaki eleman sayısıdır. E ise YSA ve ESA katman sayısıdır. $i, i \in \{1,3,5, \dots, n\}$ ESA ve tam bağlı katmanlarının sütunları ifade eder. w , ağırlık matrisidir, I ise birim matrisi ifade eder. $w_{j,i}$, j ninci katmandaki i ninci ağırlık vektörüdür. ESA için evrişim filtresi $W \times H \times C$ giriş (W ağırlık, H ağırlık ve C ise giriş kanal sayısıdır), çıkış ise Öklit normu sonucudur. Böylece bütün ikili vektör çiftlerinin ortagonallik değerleri J_{orh} hesaplanmaktadır ve denklem 11 deki gibi ağırlık maliyet fonksiyonu $J_{similarity}$ 'e $1 - \alpha$ katsayısıyla çarpılarak eklenmektedir. λ düzgünleştirme katsayısı $1 - \alpha$ değeridir.

$$J(\theta) = \alpha * J_{similarity}(w, x, y_{gold}) + (1 - \alpha)J_{orh}(w) \quad (11)$$

Burada, $J(\theta)$, ağırlık maliyet fonksiyonudur. YSA eğitimi için $J(\theta)$ 'nin türevi geriye doğru yayılmaktadır. Bu tür döngü ve şart deyimleri içeren denklemlerin türevinin alınması için otomatik türev kullanılmaktadır. Otomatik türev, zincir kuralını kullanarak fonksiyonların türevini hesaplamaktadır [38,39]. Bilgisayar programlarının türevi alınırken programı oluşturan fonksiyonların çıktılarının girdilerine göre türevleri alınarak zincir kuralı ile birleştirilmektedir. Önerilen yöntemin $J(\theta)$ türevi, otomatik türev ile hesaplanmaktadır.

5. Deneysel Sonuçlar

Önerilen düzgünleştirme yönteminin uygulamaları Nvidia p4000 32 gb ekran kartına sahip bilgisayarda yürütülmektedir. Sadece Resnet110 ve Resnet28-10 deneyleri Nvidia K80 ekran kartına sahip Google colab platformunda sonuçlar elde edilmektedir. Python programlama dili kullanılarak gerçekleştirilmektedir. Uygulamaların otomatik türevinin alınması için YSA'da autograd, ESA ise keras, tensorflow kütüphaneleri kullanılmaktadır. Otomatik türev, yazılım fonksiyonlarının türevinin alınıp geriye yayılması için geliştirilmiştir. Bir veri noktasından giriş alan fonksiyonun çıkışına kadar olan işlemleri bağımlı değişken üzerinden türevi hesaplanır.

Deneylerde ilk olarak önerilen yöntem, Xor, Moons, Spiral, Circle, Mnist, Fashion Mnist ve Cifar 10 veri setlerinde; denklem (12-15)' de sırasıyla verilen Doğruluk (*Doğ*), Duyarlılık (*Duy*), Keskinlik (*Kes*) ve *F skor* metrik performansları incelenmektedir. Daha sonra literatürde öne çıkan SO, DSO, MC, SRIP kullanan algoritmalar ile doğruluk metriğine göre karşılaştırılarak sonuçlar açıklanmaktadır [40].

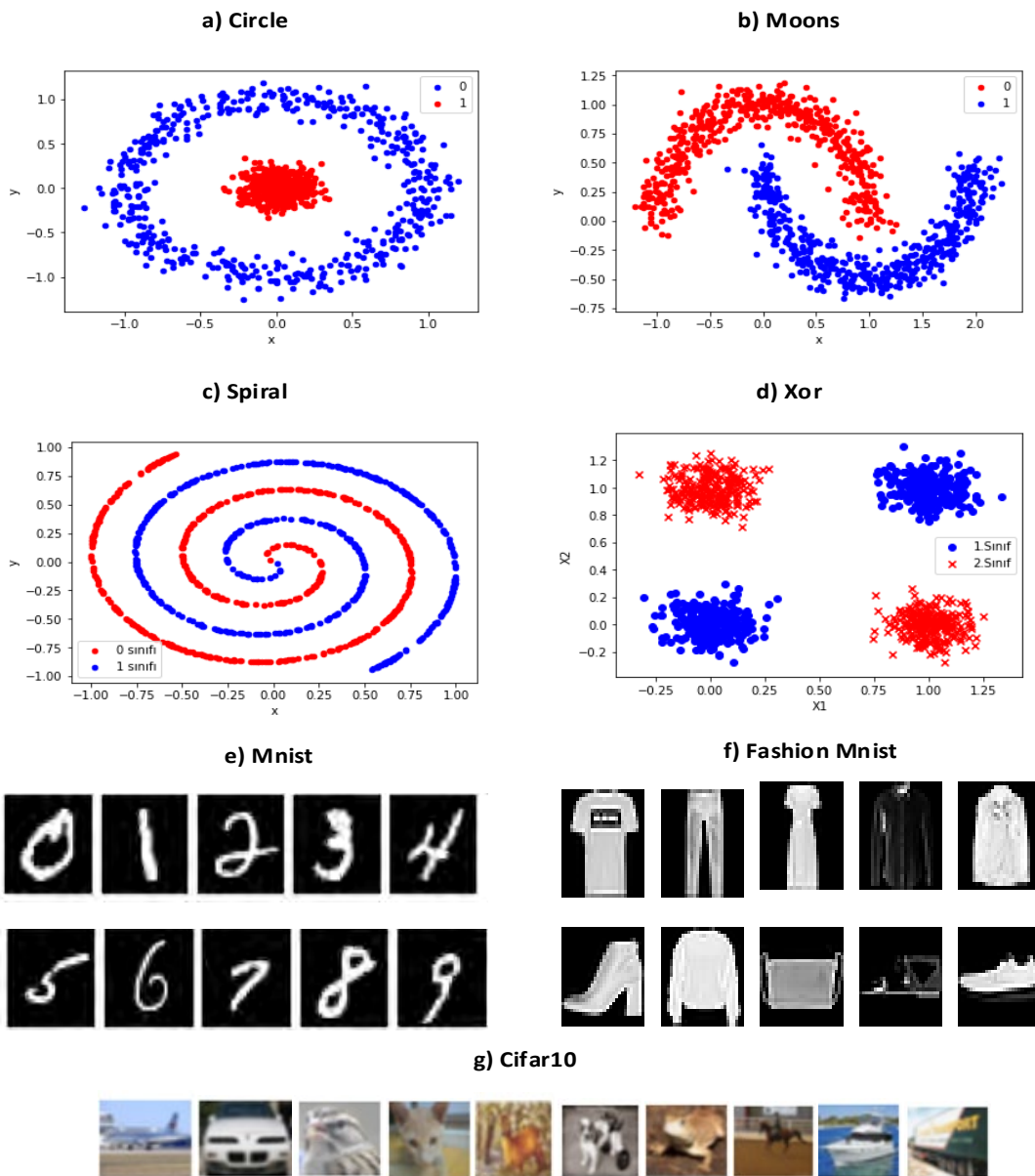
$$Doğ = \frac{TP+TN}{(TP+FN)+(FP+TN)} \times 100 \quad (12)$$

$$Duy = \frac{TP}{TP+FN} \times 100 \quad (13)$$

$$Kes = \frac{TP}{TP+FP} \times 100 \quad (14)$$

$$F\ skor = \frac{2 \times TP}{(2 \times TP + FP + FN)} \times 100 \quad (15)$$

Metrik ölçütlerinde iki sınıflı (A ve B sınıfı için) bir veri kümesi için TP , A sınıfına ait doğru tahmin edilen eleman sayısıdır. TN , B sınıfı için doğru tahmin edilen eleman sayısını göstermektedir. FP , A sınıfına ait olup yanlış sınıflandırılan veri sayısıdır. FN , B sınıfına ait olup A sınıfında sınıflanan veri sayısıdır. $Doğ$, veri kümelerinde doğru tahmin edilen test verisi sayısını göstermektedir. Duy , bir sınıfa ait duyarlılıktır. Kes ise keskinlik oranıdır. $F\ skor$ ise Duy ve Kes değerlerinin harmonik ortalamasıdır.



Şekil 2. Veri kümeleri

5.1. Kullanılan Yapay Sinir Ağları ve Veri Kümeleri

Önerilen yöntemin değerlendirilmesi için 7 farklı veri kümesi kullanılmaktadır. Bu veri kümelerini iki farklı gruba ayırabiliriz. İlki matematiksel denklemler ile üretilen sentetik veri kümeleridir. Xor, Moons, Spiral ve Circle sentetik veri kümeleri önerilen yöntemin deneysel çalışmalarında kullanılmaktadırlar. Denklem 16' da Xor, Moons ve Circle veri kümelerinin giriş, çıkış boyutu gösterilmektedir. Spiral veri kümesinde ise giriş 7 tanedir ve denklem 17' de gösterilmektedir. Bunun nedeni YSA'daki eğitim ve test performansının artırılmasıdır. Veri kümelerindeki bazı elemanları Şekil 2' de gösterilmektedir.

$$\text{Giriş} = \{x_1, x_2\}, \text{Çıkış} = \{y\} \quad (16)$$

$$\text{Giriş} = \{x_1, x_2, x_1^2, x_2^2, x_1x_2, \sin(x_1), \sin(x_2)\}, \text{Çıkış} = \{y\} \quad (17)$$

İkinci grup veri kümesi literatürde sunulan Mnist [41] Fashion Mnist [42] ve Cifar10 veri kümeleridir. Mnist, el yazısı rakamları, Fashion Mnist günlük kıyafetleri ve Cifar ise taşıt, hayvan imgelerini içermektedir. Veri kümelerindeki test ve eğitim veri sayısı, sınıflara eşit oranda dağıtılmaktadır. Çizge 1'de veri kümelerinin detayları gösterilmektedir.

Çizelge 1. Veri kümelerinin özellikleri.

Veri Kümesi	Eğitim Veri Sayısı	Test Veri Sayısı	Boyut	Sınıf Sayısı	Veri Türü
Cifar10	50000	10000	32x32x3	10	RGB imge
Mnist	60000	10000	28x28x1	10	Gri imge
Fashion Mnist	60000	10000	28x28x1	10	Gri imge
Xor	10000	1000	1x2	2	2 Boyutlu veri
Spiral	10000	1000	1x2	2	2 Boyutlu veri
Moons	10000	1000	1x2	2	2 Boyutlu veri
Circle	10000	1000	1x2	2	2 Boyutlu veri

YSA insan beyninden esinlenerek geliştirilmiştir. YSA'nın tek ve çok katmanlı modelleri bulunmaktadır. Katman girişi a , katman ağırlığı (başlangıçta rastgele belirlenen) w ve her katman için bias b olmak üzere katman çıkışı $aw + b$ olarak tanımlanmaktadır. Katman çıkışları Sigmoid, Tanh gibi doğrusal olmayan fonksiyonlar (aktivasyon fonksiyonları) uygulanmaktadır. YSA çıkışlarında hesaplanan hata geri yayılım algoritması ile ağı yayılarak eğitilmektedir. Önerilen yöntem bu prensiple çalışan 3 farklı YSA yapısı için düzenleyici olarak uygulanmaktadır. Çalışmamızda kullanılan YSA'ların katman sayıları, aktivasyon fonksiyonu, hata fonksiyonu, devir ve optimisasyon yöntemi Çizelge 2' de sunulmaktadır.

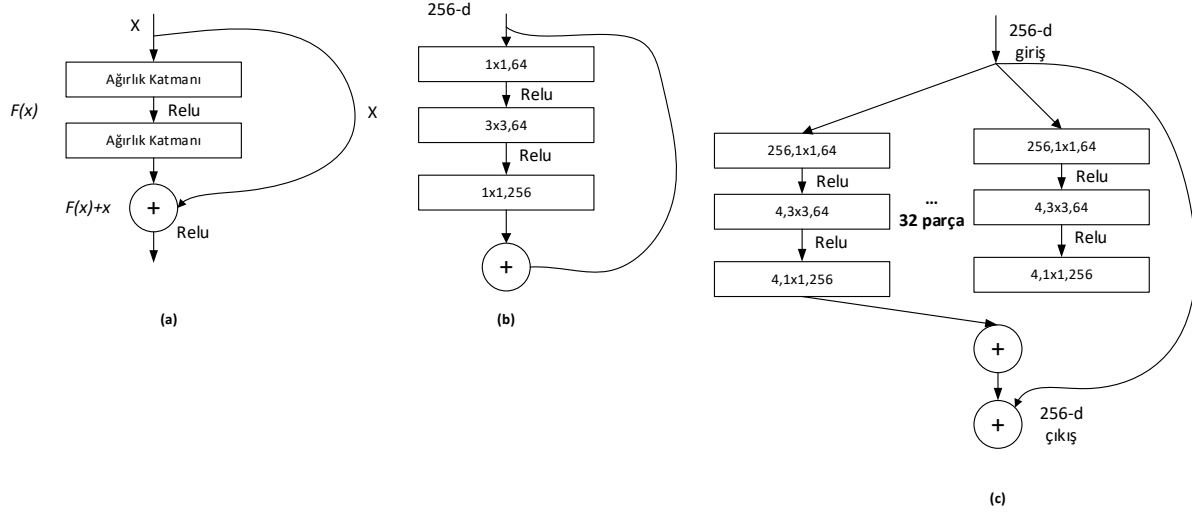
Çizelge 2. Kullanılan YSA'lar.

Ağ	Giriş katmanı	Ara katman	Çıkış katmanı	Aktivasyon	Optimisasyon	Batch size	Devir	Öğrenme Oranı
YSA1	2	4	2	tanh	Adam	100	100	0.01
YSA2	7	4	2	tanh	Adam	100	100	0.01
YSA3	728	-	10	tanh	Adam	64	50	0.01

ESA, konvülyasyon, havuzlama, aktivasyon ve sınıflandırıcı katmanlardan oluşan YSA'dır. Artık sinir ağları (Resnet), evrişimsel sinir ağlarındaki gradyan patlaması probleminin çözmek için önerilmiştir [19]. Şekil 3' te Resnet Mimarisi ve artık katman yapısı (residual blok) gösterilmektedir.

Artık katmanlar, ağıncı önceki katman değerini ileriye aktararak ağıncı öğrenme hatası optimize etmektedir. Şekil 3 (a)' da, x girişi için beklenen öğrenme işlemi $f(x)$ 'dir. Artık bağdaştırıcısından gelen x ile birleştğinde artık katmanı çıkışı $f(x) + x$ olur. Literatürde farklı derinliklerde ve

genişliklerde Resnet mimarileri bulunmaktadır. Önerilen yöntem Resnet 110 [19] ve Resnet 28-10 [43] test edilmektedir. Resnet110, Şekil 3 (b)'deki artık katmanlardan meydana gelmektedir. Toplam 18 tane konvülyasyon katmanı bulunan ağ toplam 1.7 milyon parametreden oluşmaktadır. Resnet 28-10 ağı şekil 3 (c)'deki katman yapısından meydana gelmektedir. Resnet 28-10'de, artık blokların derinliği 28 genişliği ise 10'dur.



Şekil 3. Artık Katmanlar. (a) Temel artık katman (b) ResNET 110 katmanı (c) Resnet 28-10 katmanı

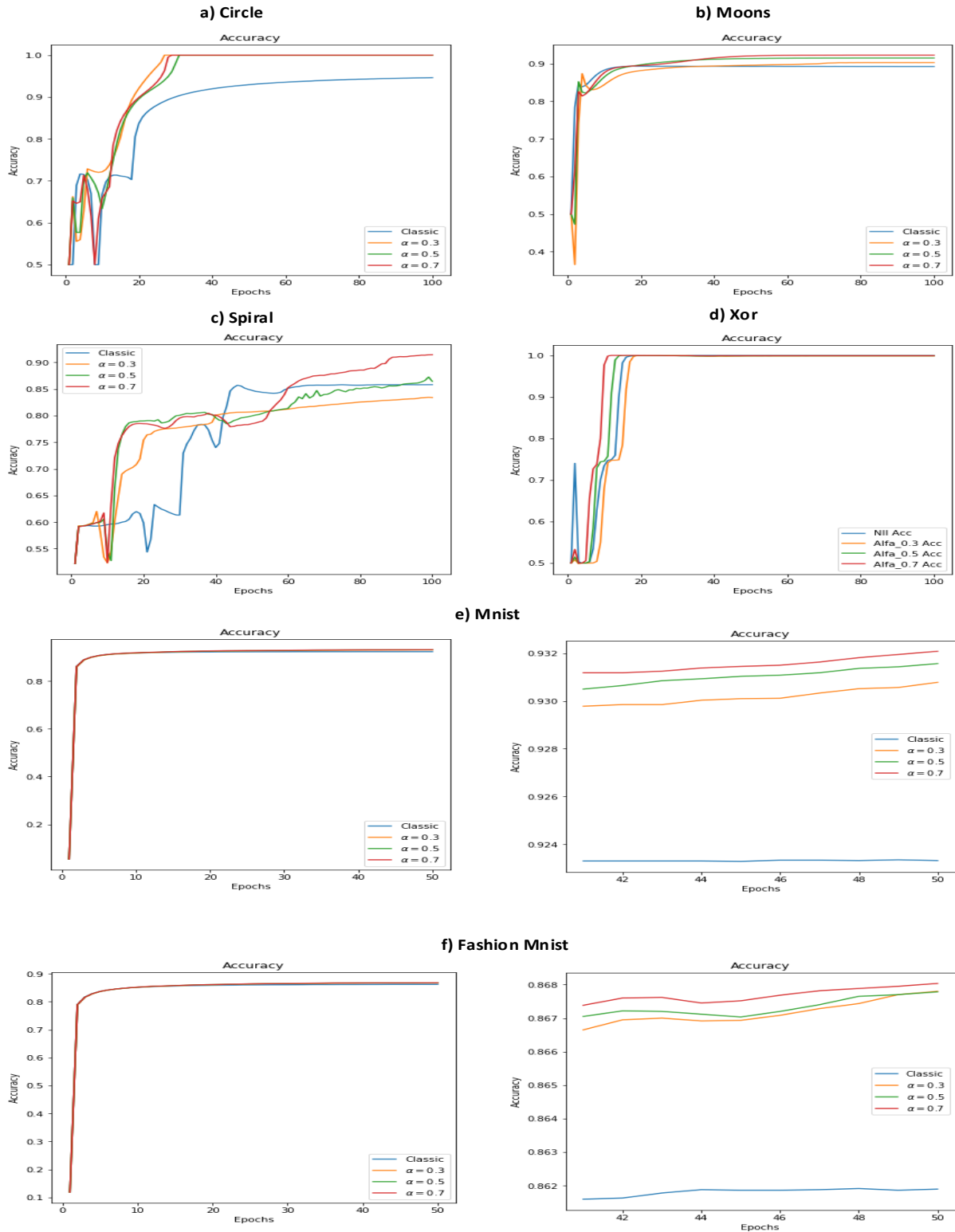
5.2. Önerilen Yöntemin Deneysel Sonuçları

Deneylerin ilkinde Xor, Moons, Circle için YSA1, spiral için YSA2 ve mnist, fashion mnist veri setleri için YSA3 kullanılmaktadır. Önerilen yöntemin, bu ağlarda Doğ ve yürütülme zamanı (t) metriklerinde öğrenme etkisi incelenmektedir.

YSA1,YSA2 ve YSA3'nın, ağırlık değerleri $[-1,1]$ aralığında başlangıçta rastgele seçilerek Çizelge 2' deki eğitim parametrelerine göre eğitilmektedirler. İlk önce Ağlar düzgünleştirme yöntemi olmadan daha sonra önerilen düzgünleştirme yöntemi $\alpha = \{0.3,0.5,0.7\}$ değerleri ile eğitilmektedir. Doğ ve t metriği sonuçları Çizelge 3' te gösterilmektedir.

Çizelge 3. Önerilen yöntemin, $\alpha = \{0.3,0.5,0.7\}$ için Doğ ve t sonuçları.

	Önerilen Yöntem						Düğünlüştürmesiz	
	$\alpha = 0.3$		$\alpha = 0.5$		$\alpha = 0.7$		-	
	Doğ	t	Doğ	t	Doğ	t	Doğ	t
Circle	99	59	99	59	99	59	94,6	24
Moon	90,3	56	91,5	55	92,3	55	89,3	22
Spiral	83,4	87	87,2	87	91,4	87	85,8	30
Xor	99	55	99	55	99	55	99	22
Mnist	93	92	93,1	92	93,2	92	92,3	78
F Mnist	86,8	96	86,7	96	86,8	96	86,1	78



Şekil 4. Önerilen Yöntemin kullanılan veri kümelerindeki Doğ-Devir Grafikleri.

Circle, Moons, Mnist, Fashion Mnist veri setleri için önerilen yöntem Doğ metriğinde düzgünleştirme içermeyen ağa göre daha iyi test sonuçları vermektedir. Sadece Xor veri setinde Doğ metriğinde artış sağlamamaktadır. Bunun nedeni ağın önerilen düzgünleştirme yöntemi kullanılmadan modeli öğrenmesidir. Spiral veri setinde ise düzgünleştirmesize göre α katsayısı 0.3 iken Doğ

metriğinde azalmaktadır, 0.5,0.7 değerlerinde ise artmaktadır. Mnist ve Fashion MNIST veri kümelerinde önerilen yöntemin Doğ değerinde en iyi sonuç $\alpha = 0.7$ değerinde sağlamaktadır. Çizelge 3'deki yürütülme zamanı (t) ağ eğitimi için geçen süre olarak tanımlanmaktadır. Bütün veri setlerinde, önerilen yöntem yürütülme zamanının arttırdığı gözlemlenmektedir. Sonuç olarak önerilen yöntemin YSA ağlarında Doğ değerinde önemli artışlar sağlamaktadır. Ayrıca α katsayısının 0.3, 0.5, 0.7 değerindeki ağ eğitimindeki etkisi değerlendirildiğinde en iyi sonucun $\alpha = 0.7$ ' de alındığı gözlemlenmektedir. Bu nedenle deneylerimizde $\alpha = 0.7$ olarak sabitlenmektedir. Önerilen yöntemin ilgi ağlardaki Doğ – devir (Accuracy-Epoch) grafikleri Şekil 4'te gösterilmektedir.

Şekil 4'de Xor ve Circle veri seti %99 başarıyla YSA ağları tarafından öğrenilmiştir. Önerilen yöntem ilgi ağlarda öğrenme eğrisinde daha hızlı yakınsama sağlanmaktadır. Moons ve spiral veri setinde ise önerilen yöntem ağın öğrenme eğrisinde kısmen yakınsama sağladığı gözlemlenmektedir. Mnist ve Fashion mnist veri setinde ise öğrenme eğrisinde kayda değer bir yakınsama gözlemlenmemektedir. Sonuç olarak önerilen yöntem öğrenme eğrileri üzerinden hızlı yakınsama sağlayabilir. Önerilen yöntemin $\alpha = 0.7$ katsayısıyla Xor, Circle, Moons, Spiral veri kümelerinde Duy, Kes ve F skor metriklerindeki sonuçları çizge 4'te gösterilmektedir. Önerilen yöntem tüm metriklerde düzgünleştirmesiz YSA göre üstünlük sağlamaktadır.

Çizge 4. Önerilen yöntemin Duy, Kes ve F skor metriklerindeki sonuçları

	Önerilen Yöntem			Düzgünleştirmesiz		
	$\alpha = 0.7$			-		
	Duy	Kes	F skor	Duy	Kes	F skor
Circle	99,4	98,6	99	94,7	94,4	94,6
Moon	92,9	91,6	92,2	90	88,4	89,2
Spiral	90,9	92	91,4	86,6	84,6	85,6
Xor	98,8	99,2	99	98,4	99,6	99

5.3. Önerilen Yöntemin Gürültü Eklenmiş Veri Setlerindeki Deneysel Sonuçları

Gürültülü veri kümeleri yapay öğrenme yöntemlerinin performansını olumsuz etkilemektedir. Kullanılan veri kümeleri denklem 18' deki gaus gürültüsü sırasıyla 0.1, 0.2, 0.3 standart sapması ile uygulanmaktadır. Gürültü veriler için önerilen yöntemin Doğ metriğindeki sonuçları çizelge 5'te gösterilmektedir.

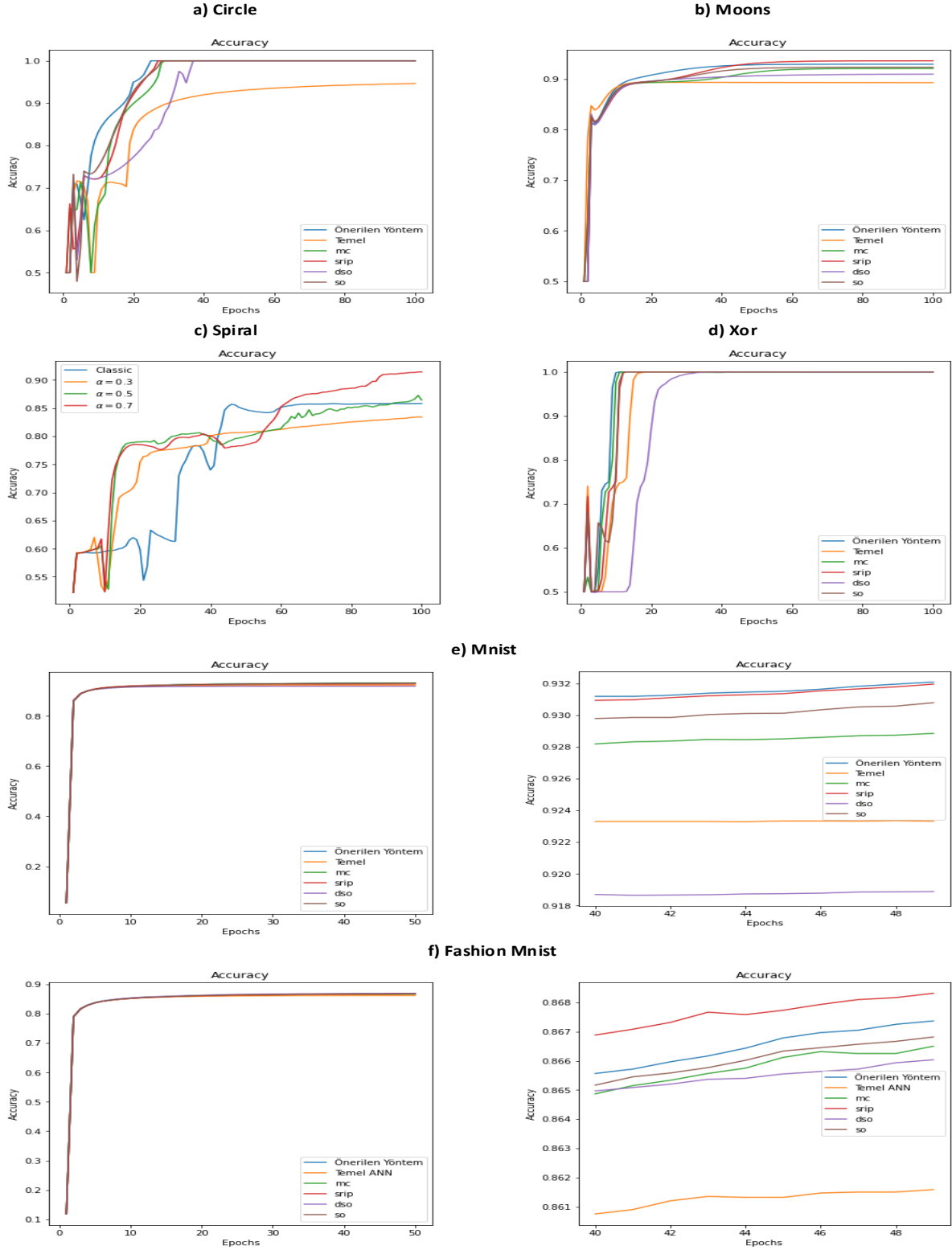
$$G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}} \quad (18)$$

Burada z gri ton seviyesi, \bar{z} gaus gürültüsünün ortalama değeri ve σ ise gaus gürültüsü standart sapmasıdır. Önerilen yöntem $\alpha = 0.7$ katsayısı ile eklenmiş YSA ve temel YSA farklı $\sigma = 0.1, 0.2, 0.3$ standart sapmalı gaus gürültüsü içeren veri kümelerine uygulanmaktadır.

Çizelge 5. Önerilen yöntemin gürültü veriler için sonuçları.

	Önerilen Yöntem			Düzgünleştirmesiz		
	$\sigma = .1$	$\sigma = .2$	$\sigma = .3$	$\sigma = .1$	$\sigma = .2$	$\sigma = .3$
	Doğruluk					
Circle	99	97,8	89,8	90,4	98,09	77,5
Moon	92,4	90,6	87,6	88,2	87,38	85,1
Spiral	95,3	93,3	89,2	85,7	83,19	87,9
Xor	97,3	90,5	80,7	98	89,4	80,6
Mnist	93	91,9	90,5	92,2	91,6	90,7
F Mnist	86,4	85	83,5	85,7	84,6	83,3

Önerilen yöntem, gürültülü Circle, Xor, Mnist veri kümelerinin bazı durumlarında hassasiyet gösterirken genel olarak gürültüye karşı duyarlılığı yüksektir. Bunun temel nedeni yöntemin, ağırlıklarını regülerize ederek ağın aşırı öğrenmesini engellemesidir. Sonuç olarak ağın test verilerini daha iyi sınıflandırması sağlanmaktadır.



Şekil 5. Önerilen Yöntemin literatürdeki yöntemlerle Doğ-Devir Grafikleri

5.4. Önerilen Yöntemin Literatürdeki Yöntemlerle Kıyaslanması

OD, en temel problemi aşırı öğrenme ve aşırı uyumdur. SO, DSO, MC ve SRIP bu problemi için çeşitli çözümler önermektedir. Önerilen yöntemin, Doğ ve t metriği ile literatürdeki yöntemlerle karşılaştırılması çizelge 5'te gösterilmektedir. Bunun yanında Şekil 5'da eğitim doğruluğu devir grafikleri (Accuracy-Epoch) veri kümeleri için verilmektedir.

Çizelge 5. Literatürdeki Diğer Yöntemlerle Kıyaslanması.

	Düzgünleştirmesiz		SO		DSO		MC		SRIP		Ön. Yöntem	
	Doğ	t	Doğ	t	Doğ	t	Doğ	t	Doğ	t	acc	t
Circle	94,6	24,3	99	42	99	47	99	60	99	54	99	59
Moon	89,3	22	92	40	91,6	42	90,3	57	93,5	56	92,3	55
Spiral	85,8	30	90	48	90,7	54	91	86	90,	83	91,4	87
Xor	99	22	99	39	99	44	99	58	99	46	99	55
Mnist	92,3	78,8	89	80	93	86	92	95	90	89	93,2	92
Fmnist	86,1	78	82,3	86	85,5	90	86	98	87,7	95	86,8	96

Circle, Spiral, Xor ve Mnist veri kümesinde önerilen yöntem, Moons ve Fashion Mnist veri kümesinde ise SRIP literatürdeki yöntemlere göre daha yüksek Doğ sağlamaktadır. Önerilen yöntem, iki vektör çiftleri üzerinde hesaplama yaptığından dolayı SO, DSO, MC ve SRIP yöntemlerine göre t metriğinde kısmen daha uzun sürede yürütülmektedir. Bu metrikte literatürde SO ve DSO algoritması diğer ortogonal yöntemlere göre daha iyi t sunmaktadır. Düzgünleştirme kullanmayan YSA'ları daha iyi t metrik değerine sahiptir. Bunun nedeni SO, DSO, MC, SRIP ve önerilen yöntem ağırlık hesaplama maliyetini artırmasıdır.

5.5. Resnet Sonuçları

OD, derin ağlarda gradyan patlamasını/kaybolmasını engellemek için kullanılmaktadır. Resnet101 ve Resnet28-10 ağında önerilen yöntem test edilmektedir ve bunun yanında literatürdeki yöntemler ile kıyaslanmaktadır.

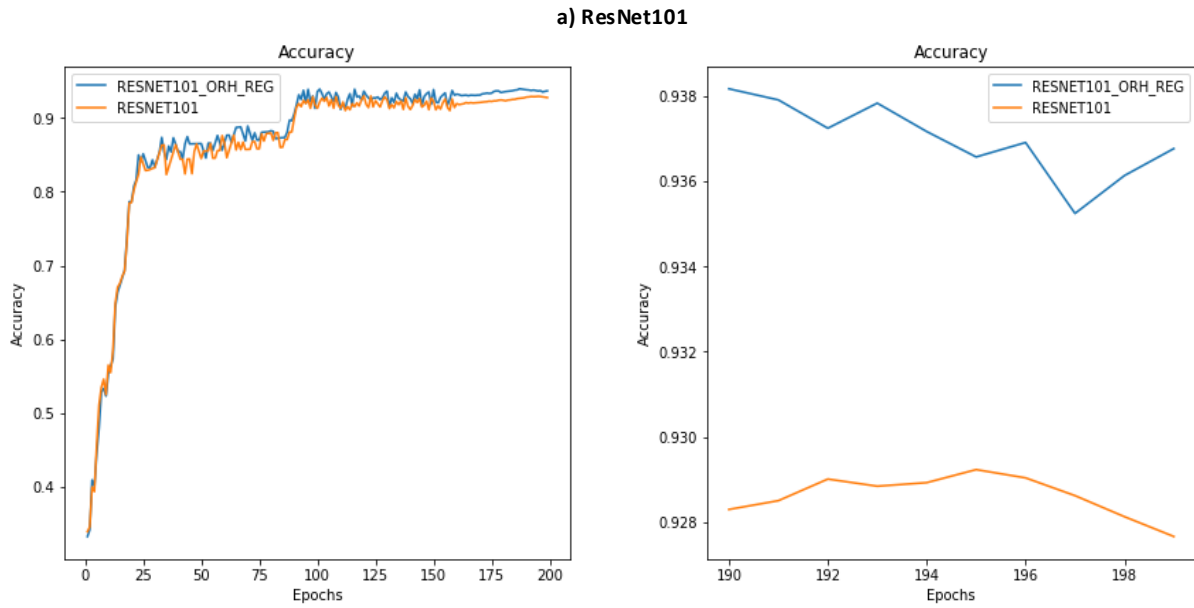
Çizelge 6. Önerilen yöntemin Cifar10 görüntü verileri için sonuçları.

Model-Optimizasyon	Yöntem	Doğ
Resnet-110	Düzgüleştirmesiz	92,96
	SO	93,22
	DSO	92,96
	MC	93,03
	SRIP	93,45
	Önerilen Yöntem	93,90
Resnet28-10	Düzgüleştirmesiz	95,84
	SO	96,24
	DSO	96,14
	MC	96,32
	SRIP	96,4
	Önerilen Yöntem	96,78

Cifar10 veri kümesi Resnet ağlarına giriş olarak verilmektedir. Resnet110 ağı 200 devir (epoch) Adam [44] optimizasyon algoritması ile eğitilmektedir. Öğrenme oranı 10^{-2} ile başlamaktadır. Daha sonra sırayla 10^{-3} , 10^{-5} ve 10^{-6} olarak 80,120 ve 160 devirlerde sırasıyla azaltılmıştır. Resnet28-10

ağında ise SGD optimizasyon algoritması kullanılarak, momentum değeri 0.9 ve 200 epoch eğitilmektedir. Öğrenme oranı 0.1 ile başlayarak 60, 120, 160 devirlerde %20 azaltılmaktadır. Çizelge 6'da önerilen yöntemin Resnet 110 ve Resnet 28-10 ağındaki doğruluk metriğindeki sonuçları gösterilmektedir.

Resnet mimarileri, ağ hata fonksiyonunu indirgemek için kullanılmaktadır. OD kullanılarak iki farklı Resnet mimarisinin hatası yaklaşık %1 indirgenmektedir. Önerilen yöntem literatürdeki klasik ağırlık zayıflatma yöntemine göre eğitim ve test doğruluğunu iyileşmesini sağlamaktadır. Bu durum Şekil 6' daki Resnet 110 mimarisinde eğitim işleminin Doğ-devir (Accuracy-Epoch) grafiğinde daha açık gösterilmektedir.



Şekil 6. Temel Resnet110 ve Önerilen yöntem ile düzenlenen Resnet110 Doğ devir grafikleri.

Şekil 6'da ortogonalite uygulanmamış temel Resnet110 mimarisi ile Önerilen ortogonal yöntemin $a = 0.7$ sabiti ile Doğ devir (accuracy epoch) grafikleri kıyaslanmaktadır. İlk 60 iterasyon ağı eğitiminde önerilen yöntemin herhangi bir düzenleme sağlamadığı görülmektedir. Bunun yanısıra 60 iterasyondan sonra ağı öğrenme hızı kademeli olarak artış sağlamaktadır. 200 devir eğitilen Resnet110 ağı hatasını azaltmakta ve test doğruluğunu %92,96 dan %93,9 artırmaktadır. Bunun yanında Önerilen yöntem Resnet 28-10 ağında Çizge 6'da görüldüğü üzere yaklaşık %2 oranında test doğruluğunu artırmaktadır.

6. Sonuçlar

Bu çalışmada ikili vektörlerin dikliğini kullanan yeni bir ortogonal düzgülendirme yöntemi önerilmektedir. Yöntem ile ESA ve YSA'ların aşırı uyum, patlayan/kaybolan gradyan problemine çözüm sağlanması amaçlanmaktadır. Bu amaçla önerilen yöntemi kullanan YSA'lar, sentetik veri kümeleri Xor, Moons, Spiral, Circle ve literatürde sunulan Mnist, Fashion Mnist veri kümelerinin gürültülü ve gürültüsüz senaryolarında Doğ, Duy, Kes ve F skor ve yürütülme zamanı (t) metriklerinde düzgülendirmesiz YSA ile karşılaştırılmaktadır. Bu deneyde yöntem YSA'lara ek bir hesaplama maliyeti getirdiğinden t zamanını artırmaktadır. Fakat diğer tüm metriklerde %1-%5 arası iyileşme sağladığı gözlemlenmektedir. Bunun yanında yöntem, literatürde öne çıkan SO, DSO, MC ve Srip algoritmalarıyla kıyaslanmaktadır. SO, DSO'da Doğ, Mc algoritmasında Doğ, t ve Srip algoritmasında ise Doğ metriğinde önemli iyileşmeler sağlanmaktadır. Ayrıca Önerilen Yöntem Gradyan

patlaması/kaybolması ve aşırı uyum problemlerinde alternatif bir düzgünleştirme aracı olarak Resnet 110 ve Resnet 28-10 ağlarında, Cifar10 veri setinde test edilmekte ve SO, DSO, MC VE SRIP algoritmalarıyla hata oranı ve *Doğ* metriğinde karşılaştırılmaktadır. *Doğ* metriğinde Resnet 110 da 93,90 ve Resnet 28-10 %96,78 skorla önemli bir başarımla sağlanmaktadır.

İleriki çalışmalarda önerilen yöntem farklı ESA mimarilerine uygulanarak, medikal imgelerin sınıflama iyileştirmesi sağlanması amaçlanmaktadır. Özellikle son yıllarda virüslerin ve bakterilerin neden olduğu zatüre hastalığının x-ray görüntülerinden tespitinin iyileştirilmesi için kullanılması planlanmaktadır.

Çıkar Çatışması Beyanı

Makale yazarları herhangi bir kurum, kuruluş, kişi ile kişisel ve finansal çıkar çatışması olmadığını beyan etmektedirler.

Kaynaklar

- [1] Szegedy C, Wei Liu, Yangqing Jia, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. 2015 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE; 2015, p. 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>.
- [2] Krizhevsky A, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. *Adv Neural Inf Process Syst* 2012;1907–1105:1–9.
- [3] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.*, 2014.
- [4] Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks 2015.
- [5] Çelik G, Talu MF. Resizing and cleaning of histopathological images using generative adversarial networks. *Phys A Stat Mech Its Appl* 2019;122652. <https://doi.org/10.1016/j.physa.2019.122652>.
- [6] Iizuka O, Kanavati F, Kato K, Rambeau M, Arihiro K, Tsuneki M. Deep Learning Models for Histopathological Classification of Gastric and Colonic Epithelial Tumours. *Sci Rep* 2020; 10:1504. <https://doi.org/10.1038/s41598-020-58467-9>.
- [7] Akram T, Lodhi HMJ, Naqvi SR, Naeem S, Alhaisoni M, Ali M, et al. A multilevel features selection framework for skin lesion classification. *Human-Centric Comput Inf Sci* 2020;10:12. <https://doi.org/10.1186/s13673-020-00216-y>.
- [8] Xu W, Keshmiri S, Wang G. Adversarially Approximated Autoencoder for Image Generation and Manipulation. *IEEE Trans Multimed* 2019;21:2387–96. <https://doi.org/10.1109/TMM.2019.2898777>.
- [9] Turkoglu M, Hanbay D, Sengur A. Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. *J Ambient Intell Humaniz Comput* 2019. <https://doi.org/10.1007/s12652-019-01591-w>.
- [10] Griebhaber D, Vu NT, Maucher J. Low-resource text classification using domain-adversarial learning. *Comput Speech Lang* 2020;62:101056. <https://doi.org/10.1016/j.csl.2019.101056>.
- [11] Yin Y, Li H, Fu W. Faster-YOLO: An accurate and faster object detection method. *Digit Signal Process* 2020;102:102756. <https://doi.org/10.1016/j.dsp.2020.102756>.
- [12] Wu Y, Zhang Z, Wang G. Unsupervised Deep Feature Transfer for Low Resolution Image Classification. 2019 IEEE/CVF Int. Conf. Comput. Vis. Work., IEEE; 2019, p. 1065–9. <https://doi.org/10.1109/ICCVW.2019.00136>.
- [13] Salimans T, Kingma DP. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Adv. Neural Inf. Process. Syst.*, 2016.
- [14] Huang L, Liu X, Lang B, Yu AW, Wang Y, Li B. Orthogonal Weight Normalization: Solution to Optimization over Multiple Dependent Stiefel Manifolds in Deep Neural Networks. 32nd AAAI Conf Artif Intell AAAI 2018 2017.

- [15] Martin CH, Mahoney MW. Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning 2018.
- [16] Xu X, Wang G, Sullivan A, Zhang Z. Towards Learning Affine-Invariant Representations via Data-Efficient CNNs 2019.
- [17] Wang J, Zeng Z, Hou ZG. Advances in neural networks. vol. 149. 2015. <https://doi.org/10.1016/j.neucom.2014.07.026>.
- [18] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterton M, editors. Proc. Thirteen. Int. Conf. Artif. Intell. Stat., vol. 9, Chia Laguna Resort, Sardinia, Italy: PMLR; 2010, p. 249–56.
- [19] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2016 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE; 2016, p. 770–8. <https://doi.org/10.1109/CVPR.2016.90>.
- [20] Veit A, Wilber M, Belongie S. Residual Networks Behave Like Ensembles of Relatively Shallow Networks 2016.
- [21] Mishkin D, Matas J. All you need is a good init 2015.
- [22] Cui L, Zhai H, Lin H. A Novel Orthogonal Extreme Learning Machine for Regression and Classification Problems. Symmetry (Basel) 2019;11:1284. <https://doi.org/10.3390/sym11101284>.
- [23] Shin J, Koo B, Kim Y, Paik J. Deep Binary Classification via Multi-Resolution Network and Stochastic Orthogonality for Subcompact Vehicle Recognition. Sensors 2020;20:2715. <https://doi.org/10.3390/s20092715>.
- [24] Jia K, Li S, Wen Y, Liu T, Tao D. Orthogonal Deep Neural Networks. IEEE Trans Pattern Anal Mach Intell 2020;1–1. <https://doi.org/10.1109/TPAMI.2019.2948352>.
- [25] Zhang Z, Ma W, Wu Y, Wang G. Self-Orthogonality Module: A Network Architecture Plug-in for Learning Orthogonal Filters. 2020 IEEE Winter Conf. Appl. Comput. Vis., IEEE; 2020, p. 1044–8. <https://doi.org/10.1109/WACV45572.2020.9093466>.
- [26] Bansal N, Chen X, Wang Z. Can we gain more from orthogonality regularizations in training deep CNNs? Adv Neural Inf Process Syst 2018;2018-Decem:4261–71.
- [27] Xie D, Xiong J, Pu S. All You Need is Beyond a Good Init: Exploring Better Solution for Training Extremely Deep Convolutional Neural Networks with Orthonormality and Modulation. 2017 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE; 2017, p. 5075–84. <https://doi.org/10.1109/CVPR.2017.539>.
- [28] Zhang L, Li D, Guo Q. Deep Learning From Spatio-Temporal Data Using Orthogonal Regularization Residual CNN for Air Prediction. IEEE Access 2020;8:66037–47. <https://doi.org/10.1109/ACCESS.2020.2985657>.
- [29] Zhu F, Liang Q. OCRNN: An orthogonal constrained recurrent neural network for sleep analysis based on EEG data. Ad Hoc Networks 2020;104:102178. <https://doi.org/10.1016/j.adhoc.2020.102178>.
- [30] Zhang J, Yao R, Ge W, Gao J. Orthogonal convolutional neural networks for automatic sleep stage classification based on single-channel EEG. Comput Methods Programs Biomed 2020;183:105089. <https://doi.org/10.1016/j.cmpb.2019.105089>.
- [31] Jalwana MAAK, Akhtar N, Bennamoun M, Mian A. Orthogonal Deep Models as Defense Against Black-Box Attacks. IEEE Access 2020;8:119744–57. <https://doi.org/10.1109/ACCESS.2020.3005961>.
- [32] Rodríguez P, González J, Cucurull G, Goufous JM, Roca X. Regularizing CNNs with Locally Constrained Decorrelations 2016.
- [33] Balestriero R, richard baraniuk. A Spline Theory of Deep Learning. In: Dy J, Krause A, editors. Proc. 35th Int. Conf. Mach. Learn., vol. 80, PMLR; 2018, p. 374–83.
- [34] Donoho DL. Compressed sensing. IEEE Trans Inf Theory 2006;52:1289–306. <https://doi.org/10.1109/TIT.2006.871582>.
- [35] Lu C, Li H, Lin Z. Optimized Projections for Compressed Sensing via Direct Mutual Coherence Minimization 2015.
- [36] Yoshida Y, Miyato T. Spectral Norm Regularization for Improving the Generalizability of Deep Learning 2017.

- [37] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition 2014.
- [38] Griewank A, Walther A. Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM; 2008.
- [39] Tuncer İH, Kaya H, Tiftikci H. OTOMATİK TÜREV ARAÇLARI İLE AYRIK ADJOINT ÇÖZÜCÜ GELİŞTİRİLMESİ. 7. Havacılık ve Uzay Konf., Samsun/Türkiye: 2018, p. 1.
- [40] D. M. W. Powers. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation 2011. <https://doi.org/10.9735/2229-3981>.
- [41] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86:2278–324. <https://doi.org/10.1109/5.726791>.
- [42] Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms 2017.
- [43] Zagoruyko S, Komodakis N. Wide Residual Networks 2016.
- [44] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization 2014.