Cilt 9, Sayı 1, Sayfa 107-113, 2020 Volume 9, Issue 1, Page 107-113, 2020

Araştırma Makalesi DOI: **10.46810/tdfd.707200** Research Article

Türk Doğa ve Fen Dergisi
Turkish Journal of Nature and Science
www.dergipark.gov.tr/tdfd

Boyut İndirgeme Yöntemlerinin Karşılaştırmalı Analizi

Mücahit ÇALIŞAN [1*], Muhammed Fatih TALU [2]

[1]Bingöl Üniversitesi, Enformatik Bölümü, Bingöl, Türkiye
[2]İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye
Mücahit ÇALIŞAN ORCID No: 0000-0003-2651-5937
Muhammed Fatih TALU ORCID No: 0000-0003-1166-8404

**Sorumlu yazar: mcalisan@bingol.edu.tr*

(Alınış: 21.03.2020, Kabul: 31.05.2020, Online Yayınlanma: 18.06.2020)

Anahtar Kelimeler
Boyut indirgeme,
PCA,
LDA,
AE

Öz: Günümüz veritabanları hızlı bir şekilde büyümektedir. Örneğin Youtube'a her dakikada ortalama 300 saatlik video yüklenmektedir. Veri boyutuyla orantılı bir şekilde, işleme, depolama ve transfer maliyetleri artmaktadır. Buna karşılık, özellikle video ve imge gibi yüksek boyutlu veri içeriklerinin büyük oranda benzer olduğu bilinmektedir. Bu tür yüksek boyutlu ham verilerin, düşük boyutlara indirgenmesi, imge sınıflandırma, algılama ve anlamlı bilgi çıkarım prosesleri için hayati öneme sahiptir.

Veri boyutunu indirgeyen çok sayıda teknik mevcuttur. Klasik yapay öğrenme tekniklerinden; PCA (Temel Bileşenler Analizi) ve LDA (Doğrusal Ayıraç Analizi), probleme matematiksel bir çözüm zemini kazandırdıkları için ön plana çıkarken, doğrusal olmayan tekniklerden, derin öğrenme yaklaşımlarından olan Oto-Kodlayıcı (Auto-Encoding), büyük verilerin indirgenmesine izin vermesi bakımından araştırmacıların ilgisini çekmektedir.

Bu çalışmada, gerçek ve sentetik veriler (doğrusal ve doğrusal olmayan) kullanılarak PCA, LDA ve Auto-Encoding (AE) yöntemlerinin boyut indirgeme performansları incelenmiştir. Belirli kıstaslarda (harcanan zaman, yeniden inşa etme doğruluğu vb.) alınan sonuçlar karşılaştırmalı bir şekilde sunulmuştur.

Comparative Analysis of Dimension Reduction Methods

Keywords
Dimension reduction,
PCA,
LDA,
AE

Abstract: Today's databases are growing rapidly. For example, Youtube uploads an average of 300 hours of video every minute. In proportion to the size of the data, processing, storage and transfer costs are increasing. On the other hand, it is known that high-dimensional data contents such as video and image are largely similar. Such high-dimensional raw data has a vital proposition for the reduction of images to low dimensions, image classification, detection and meaningful information extraction processes.

There are many techniques available to reduce data size. From classical artificial learning; PCA (Principal Components Analysis) and LDA (Linear Discriminant Analysis), while probing is at the forefront of gaining a mathematical solution, Autoencoder, which is one of the non-linear techniques and deep learning approaches, attracts researchers to allow the reduction of large data.

In this study, dimensional reduction performances of PCA, LDA and Auto-Encoding (AE) methods using real and synthetic data (linear and nonlinear) were investigated. The results obtained on certain criteria (time spent, correctness of reconstruction, etc.) are presented comparatively.

1. GİRİŞ

Yüksek boyutlu verilerin (sosyal medya verileri, konuşma sinyalleri, dijital fotoğraflar vb.) analizi birçok alanı kapsamaktadır. Veri boyutunun indirgenmesi analiz kolaylığını arttırmaktadır. Boyut indirgeme yöntemlerinin temel amacı, veri içeriğindeki en asgari bir kayba karşılık boyutun maksimum küçültülmesidir.

Bunun için verinin asli bileşenlere izdüşümü yaklaşımı kullanılmaktadır. Başka bir deyişle, verinin önemli olmayan bileşenlerini tespit edip, kaldırmak ve boyutun düşmesini sağlamaktır. Boyut indirgeme ile yüksek boyutlu veri de bulunan bilgi içeriği daha az sayıda öz

nitelik ile temsil edilmektedir. Az sayıda öz nitelik ile sınıflandırma performanslarının yükselmesi hedeflenmektedir.

Bu çalışmada, doğrusal boyut azaltma tekniklerinden PCA [1],[2], LDA [3],[4] ve doğrusal olmayan boyut azaltma tekniklerinden Autoencoder yöntemleri incelenmiştir [5],[6]. Bu yöntemlerin performansları, literatürde sık kullanılan MNIST [7] veri kümesi (el yazısıyla yazılmış rakamlar) kullanılarak elde edilmiş, sınıflandırma doğrulukları ve hesaplama süreleri kıyaslanmıştır. Çalışmada kullanılan bu yöntemlerde MNIST veri tabanının tamamı kullanılmamıştır. MNIST veri setinin büyüklüğü düşünüldüğünde işlemlerin uzun sürmemesi için her sınıftan düşük sayıda örnek kullanılmıştır. Sınıflandırma işlemi için yapay sinir ağı yöntemi kullanılmıştır. Ayrıca sentetik bir veri üzerinde de bu tekniklerin gözden geçirilmesi ve sistematik olarak karşılaştırılması sunulmuştur.

Yüksek boyutlu veriler ile çalışmak hesaplama zamanını arttırmaktadır. Bu veriler için öz temsillerin keşfi için makine öğrenmesi ve örüntü tanıma çalışmaları kullanılmaktadır. Bu işlem literatürde boyut indirgeme olarak karşımıza çıkmaktadır.

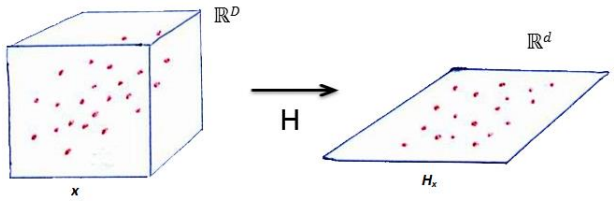
Boyut indirgeme teknikleri zaman serileri analizi [8], makine öğrenmesi [9], veri inceleme [10] ve biyometrik [11] gibi birçok uygulamada yoğun bir şekilde kullanılmaktadır. Zaman serileri ile ilgili veri tabanlarının boyutları genellikle çok büyüktür [8]. Uzay çalışmaları göz önüne alındığında, bu astronomik veri tabanları terabaytlar büyüklüğünde veri içermekte ve gün geçtikçe büyümeye devam etmektedirler. LM Bruce tarafından yapılan çalışmada hiperspektral sensörlerden devamlı ve farklı dalga boylarında alınan görüntülerin işlenmesi için çok fazla zaman ve işlem gücü gerektiği vurgulanmıştır [12]. Bu görüntülerdeki fazlalıkları azaltıp daha düşük boyutta veri kullanılarak sınıflandırma performansını arttırabilmek için boyut indirgeme tekniklerine ihtiyaç duyulmaktadır. Bu teknikler sayesinde öznitelik uzayı iki ya da üç boyuta indirgenip bir monitör ile görüntüleme yapılabilmektedir. Nature dergisinde yayınlanan bir makalede [13] boyut küçültme algoritması kullanılarak tasarlanan derin öğrenme algoritmasının içerisindeki veri yapıları gösterilmiştir.

2. KULLANILAN BOYUT KÜÇÜLTME YÖNTEMLERİ

Makine öğrenimi ve model sınıflandırma uygulamaları için bir ön işlem basamağı olarak boyut azaltma teknikleri kullanılmaktadır. Bu tekniklerden doğrusal olarak PCA ve LDA ön plana çıkarken, doğrusal olmayan ve aynı zamanda bir derin öğrenme algoritması olan Autoencoder yöntemi öne çıkmaktadır.

Boyut azaltma şu şekilde tanımlanabilir; nxD boyutunda bir X matrisi olsun. Burada x_i $(i \epsilon \{1,2,\dots,n\})$. x_i , D boyutundaki X verisinin i. satırını temsil etmektedir. Yüksek boyuta sahip x_i 'nin düşük boyuttaki karşılığı y_i dir. y_i , d boyutundaki Y verisinin i. satırını temsil eder. Şekil 1'de D boyutundaki bir verinin daha düşük boyuta sahip d boyutundaki temsili görülmektedir.

$x_i \epsilon R^D \rightarrow y_i \epsilon R^d$



Şekil 1. Yüksek boyutlu verinin düşük boyuta indirgenmesi

2.1. PCA

Bir verinin temel bileşenleri, verinin normalize edilmesinden sonra Kovaryans matrisinin öz değer ve öz vektörü hesaplanarak elde edilmektedir. Denklem 1'de verinin ortalama değeri, Denklem 2'de Kovaryans matrisi hesabı gösterilmektedir. m verinin boyutunu, C ise Kovaryans matrisini temsil etmektedir.

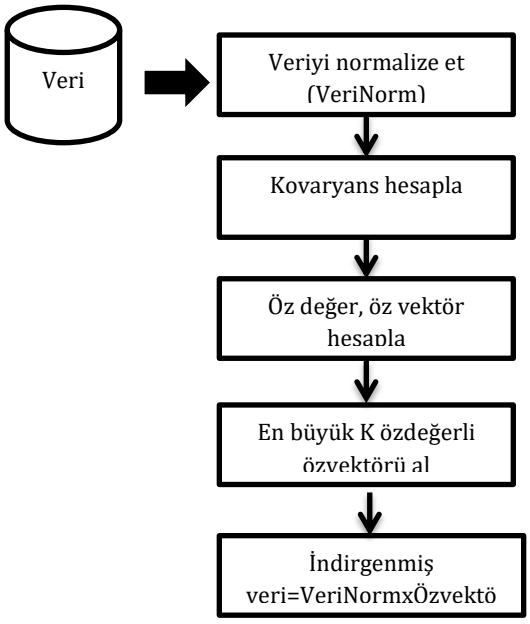
$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i \qquad (1)$$

$$C = \sum_{i=1}^{n}(X - \bar{X})(X - \bar{X})^T \qquad (2)$$

Kovaryans matrisin sırasıyla özdeğerleri (λ) ve özvektörleri (V) Denklem 3'de hesaplanmaktadır.

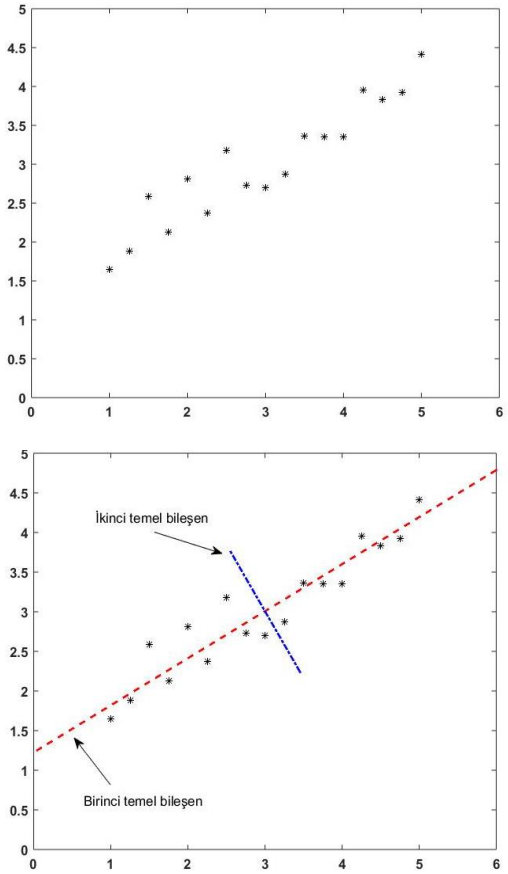
$$\det(\lambda I - C) = 0, \qquad (\lambda_k I - C)xV_k = 0 \qquad (3)$$

Özdeğerler büyükten küçüğe sıralanır ve en büyük özdeğerlere karşı gelen özvektörler bulunur. Normalleştirilmiş verinin K adet özvektör üzerine izdüşümü indirgenmiş veriyi üretir. Şekil 2'de PCA'nın akış diyagramı verilmiştir.



Şekil 2. PCA akış diyagramı

Şekil 3(a)'da üretilen iki boyutlu sentetik bir veri, 3(b)'de ise bu verinin iki temel bileşeni gösterilmektedir.



Şekil 3. PCA örneği (a) sentetik veri, (b) temel bileşenler

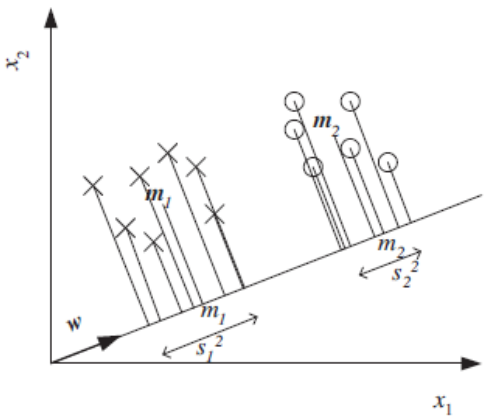
PCA algoritmasındaki K bileşen sayısının optimal değeri için aşağıdaki ilk K öz değer toplamının bütün özdeğer toplamına oranı 0.99 gibi yüksek olması beklenir. Şekil 4'de alınan bir görüntünün farklı K bileşen etkisinin çıktıları gösterilmektedir.

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \geq 0.99$$

Şekil 4. Yeniden inşa etmede K değişkeni (bileşen sayısı) etkisi. Orijinal görüntü $K = 20,10$ ve 5 kullanılarak indirgenmiştir

2.2. LDA

LDA, sınıf bilgileri belirli olan verinin indirgenmesinde kullanılır ve sınıfları en iyi ayrıştıracak vektörleri arar [13]. Yöntemin rahatlıkla anlaşılması için, Şekil 5'de gösterilen iki boyutlu veriyi göz önüne alalım.



Şekil 5. İki boyutlu veri örneği

LDA, sınıflar arası uzaklığı $|m_1 - m_2|$ büyük, sınıf içi uzaklığı $(s_1^2 + s_2^2)$ küçük ister. Bu nedenle Denklem 4'deki maliyet fonksiyonunu maksimize eden w vektörünü bulmaya çalışır:

$$J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{4}$$

m_1 ve m_2 sınıf ortalamalarını, s_1^2 ve s_2^2 sınıf varyanslarını temsil eder ve aşağıdaki gibi hesaplanır:

$$m_1 = \frac{\sum_i w^T x_i y_i}{\sum_i y_i} = w^T M_1$$
$$m_2 = \frac{\sum_i w^T x_i (1 - y_i)}{\sum_i (1 - y_i)} = w^T M_2$$
$$s_1^2 = \sum_i (w^T x_i - m_1)^2 y_i$$
$$s_2^2 = \sum_i (w^T x_i - m_2)^2 (1 - y_i)$$

$(m_1 - m_2)^2$ ve $s_1^2 + s_2^2$ ifadelerinde w ayrışımı yapıldığında $J(w)$ maliyeti aşağıdaki gibi yazılabilir:

$$\begin{aligned}(m_1 - m_2)^2 &= (w^T M_1 - w^T M_2)^2 \\ &= w^T (M_1 - M_2)(M_1 - M_2)^T w \\ &= w^T \Sigma_M w\end{aligned}$$
$$\begin{aligned}s_1^2 = \sum_i (w^T x_i - m_1)^2 y_i &= \sum_i (w^T x_i - w^T M_1)^2 y_i \\ &= \sum_i w^T (x_i - M_1)(x_i - M_1)^T w y_i \\ &= w^T \Sigma_{S1} w\end{aligned}$$
$$\Sigma_{S1} = \sum_i (x_i - M_1)(x_i - M_1)^T y_i$$
$$s_1^2 + s_2^2 = w^T (\Sigma_{S1} + \Sigma_{S2}) w = w^T \Sigma_s w$$

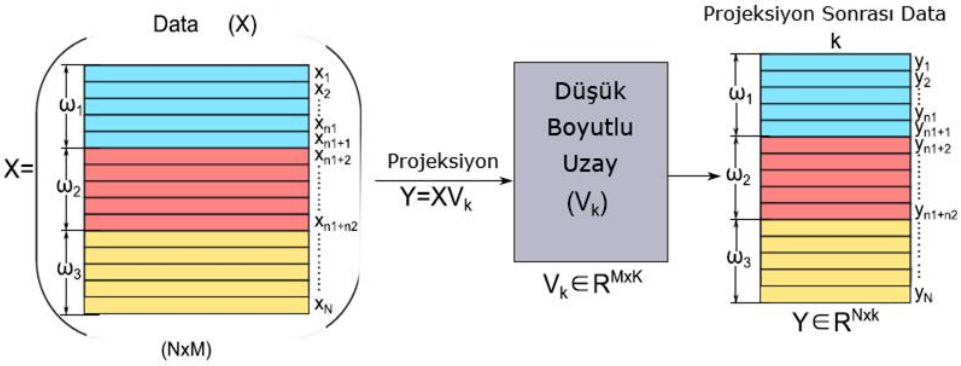
$$J(w) = \frac{w^T \Sigma_M w}{w^T \Sigma_s w} = \frac{|w^T (M_1 - M_2)|^2}{w^T \Sigma_s w}$$

Burada $J(w)$ maliyet fonksiyonunun maksimum noktası w ya göre türevinin sıfıra eşit olduğu noktadır:

$$\frac{\partial J(w)}{\partial w} = 0 \rightarrow w = c \Sigma_s^{-1} (M_1 - M_2)^2$$

c katsayısı yön bilgisi içermediği için önemsiz bir sabittir.

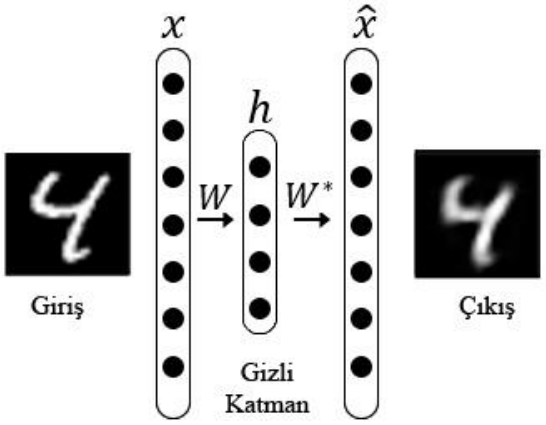
Şekil 6'da, 3 sınıfta kümelenen N adet M boyutlu örüntüye sahip giriş verisinin LDA yöntemiyle k boyuta indirgenişi gösterilmektedir [14].



Şekil 6. Girdi örneklerinin (X) LDA'nın daha düşük boyutlu alanına (V_k) yansıtılması

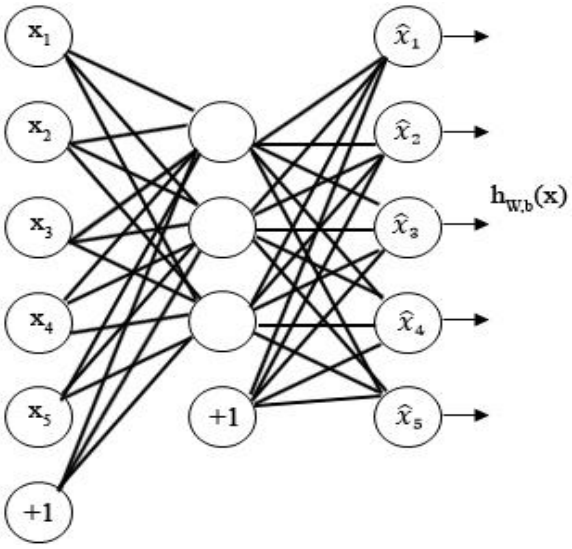
2.3. AE

AE eğiticisiz bir yapay öğrenme tekniğidir. Temel olarak iki kısımdan oluşmaktadır: kodlayıcı ve çözücü. Bu modeli ileri beslemeli yapay sinir ağlarından ayıran en önemli özellik, giriş veri seti (x) ile çıkış veri setinin ($\hat{x}$) benzer olması ve dolayısıyla çıktı katmanındaki nöron sayısının girdi katmanındaki nöron sayısına eşit olmasıdır. Şekil 7'de AE'nin çalışma prensibi gösterilmektedir.



Şekil 7. Oto-kodlayıcının çalışma prensibi

$y^{(i)} = x^{(i)}$ davranışını sergileyen bu eğiticisiz yapay öğrenme mimarisi Şekil 8'de görülmektedir. Bu mimari $h_{w,b}(x) \approx x$ 'i öğrenmeye çalışır. Başka bir deyişle çıkış olan $\hat{x}$, giriş olan x 'e benzetilmeye çalışılır.



Şekil 8. Oto-kodlayıcı mimarisi

$y^{(i)} = x^{(i)}$ davranışını sergileyen bu eğiticisiz yapay öğrenme tekniğinin amaç fonksiyonu Denklem 5'deki gibi tanımlanır;

$$\min_{W,b} J(W,b) = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - y^{(i)})^2 \qquad (5)$$

Optimum W, b paremetrelerinin tespiti için j. gizli ünitenin aktivasyonu ${a_j}^{(2)}$ olarak tanımlanır. Ağa belirli bir x girişi verildiğinde gizli ünitenin ortalama aktivasyonunun açık ifadesi Denklem 6'da gösterilmektedir.

$$\hat{p}_j = \frac{1}{m}\sum_{i=1}^{m}[{a_j}^{(2)}(x^{(i)})] \qquad (6)$$

m örüntü sayısıdır. Ortalama aktivasyon değerinin dışarıdan girilen ve seyreklik parametresi olarak adlandırılan p değerine eşit olması istenir. p genelde 0'a yakın (0.05 gibi) bir değer seçilir. Bunu başarmak için optimizasyon hedefine $\hat{p}$ değerini p den saptırmak için bir ceza terimi (penalty term) eklenir. Ceza terimi için birçok seçenek mevcuttur. Oto-kodlayıcı mimarilerinde

en çok kullanılan ceza terimi Denklem 7'de gösterilmiştir;

$$\sum_{j=1}^{s_2} plog\frac{p}{\hat{p}_j} + (1-p)log\frac{1-p}{1-\hat{p}_j} \quad (7)$$

Denklemdeki s_2, gizli katman nöron sayısını temsil etmektedir. Denklem 7'de yer alan ceza terimi Kullback-Leibler (KL) uzaklığı olarak bilinmektedir. KL uzaklığı birer Bernoulli rastgele değişkeni olan seyreklik parametresi p ve Denklem 6'da hesaplanan ortalama aktivasyon değeri $\hat{p}_j$ arasındaki ilişki olarak tanımlanır. KL uzaklığı iki farklı dağılımın birbirine her noktadaki oranları alınarak bu oranların logaritmalarının alınmasıyla hesaplanır [15]. Böylece KL uzaklığı kullanılarak hesaplanan ceza terimi Denklem 8'de ki gibi yazılmaktadır;

$$\sum_{j=1}^{s_2} KL(p||\hat{p}_j) \quad (8)$$

Optimizasyon için kullanılan oto-kodlayıcı maliyet fonksiyonu iki terimin toplamından oluşur. Bunlar tek seviyeli standart yapay sinir ağı maliyeti ile KL uzaklığından gelen terimlerdir. KL'nin amaç fonksiyonundaki etkinlik değerini ayarlayan ve seyreklik değeri gibi dışarıdan girilen β değeri genelde 1-6 değerleri arasından seçilir. $W^{(1)}, b^{(1)}$ değerleri ise giriş ile gizli katman arasındaki yani kodlayıcı olarak adlandırılan kısımdaki ağın ağırlık parametreleridir. Denklem 9'da oto-kodlayıcının toplam maliyet fonksiyonu gösterilmiştir.

$$J_{sparse}(W,b) = J(W,b) + \beta\sum_{j=1}^{s_2} KL(p||\hat{p}_j) \quad (9)$$

3. DENEYSEL SONUÇLAR

3.1. Kullanılan Veri Tabanları

Çalışmada gerçek ve sentetik verilerden oluşan farklı eğitim kümeleri kullanılmıştır. Gerçek veri kümesi olarak MNIST veri seti (elle yazılmış rakamlar) kullanılmıştır. MNIST veri seti $28x28$ boyutlarında, gri seviyeli, 10 farklı rakama ait toplam 50000 eğitim ve 10000 test örneğinden oluşmaktadır. Bu veri tabanının seçilme sebebi, fazla sayıda ve çeşitte el yazısı rakamı içermesidir.

3.2. Deneysel Sonuçlar

MATLAB ortamında gerçekleştirilen uygulamalarda test sonuçları için birden fazla yineleme yapılmaktadır. Uygulamalarda PCA, LDA ve AE yöntemlerinin veri tabanları üzerindeki performansları sınıflandırma doğrulukları ve çalışma süresi açısından karşılaştırılmıştır. MNIST veri tabanındaki imgeler $28x28 = 784$ uzunluğunda vektörler içermektedir. PCA ve LDA yöntemleri ile öznitelik vektörü boyutu farklı boyutlara indirgenmiş ve sınıflandırma sonuçları Tablo 1'de verilmiştir. AE yönteminin gizli katman sayısı (öznitelik vektör boyutu) ayarlanarak giriş bilgisinin çıkışta inşa edilmesi sağlanmıştır. Her üç yöntem kullanılarak elde edilen öznitelikler sınıflandırıcıya verilmiş ve ağın doğruluk oranları belirlenmiştir. Eğitim kümesindeki örnek sayısının yüksek olması ve eğitim zamanının uzaması nedeniyle, eğitim işlemi için 10000, test işlemi için 2000 örnek kullanılmıştır.

Tablo 1. PCA ve LDA'nın farklı özniteliklerdeki sınıflandırma doğrulukları ve harcanan zaman değerleri

	Doğruluk(%)		**Zaman (sn)**	
Öznitelik Sayısı	PCA	LDA	PCA	LDA
10	**86**	78.75	0.4341	0.7465
20	**92.9**	80.85	0.4452	0.7619
40	**94.25**	79.6	0.4492	0.7689
80	**93.5**	75.35	0.4565	0.7723
200	**92.85**	84.3	0.4784	0.7879

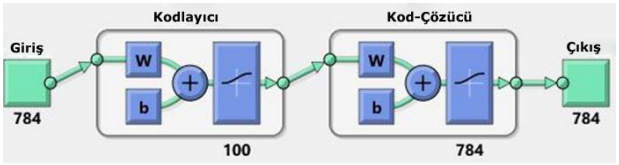
Yöntemlerin özellik sayılarına göre doğruluk ve çalışma süreleri Tablo 1'de verilmiştir. PCA ve LDA gibi doğrusal yöntemler çok hızlı bir sürede boyut indirgeme yapabilirken doğrusal olmayan AE yönteminin ise çok daha uzun sürede boyut indirgemeyi tamamladığı görülmektedir.

Yapılan son deneysel çalışma oto-kodlayıcının MNIST üzerindeki doğruluk ve zaman değerlerinin elde edilmesiyle ilgilidir. Tablo 2'de PCA ve Autoencoder yöntemlerinin doğruluk oranları verilmektedir.

Tablo 2. PCA ve AE'nin farklı özniteliklerdeki sınıflandırma doğrulukları

	Doğruluk(%)	
Öznitelik Sayısı / Ara katman hücre sayısı	PCA	AE
10	86	81.6
20	92.9	93.4
40	94.25	97.5
80	93.5	98.7
200	92.85	97.3

Şekil 9'da kullanılan oto-kodlayıcı mimarisi görülmektedir. Mimaride hücre sayıları sırayla 784, 100, 784 (giriş, orta, çıkış) şeklindedir.



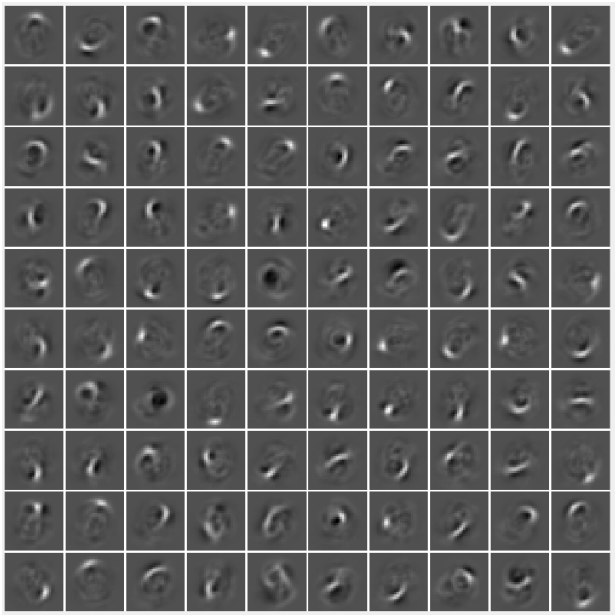
Şekil 9. MNIST için kullanılan AE mimarisi

Değiştirilen seyreklik parametresinin (p) gizli hücre sayılarına göre doğruluk oranına etkisi Tablo 3'de verilmiştir.

Tablo 3. Seyreklik parametresinin gizli hücre sayısına göre sınıflandırma doğrulukları

Orta katman hücre sayısı	*p*	Doğruluk(%)
10	0.05	80
	0.10	78.3
	0.20	82
	0.50	80.9
100	0.05	96
	0.10	96.9
	0.20	98.7
	0.50	97.7

Bir Autoencoderin kodlayıcı parçası tarafından öğrendiği eşleme, verilerden özellikler ayıklamak için kullanılır. Kodlayıcıdaki her bir nöron, belirli bir görsel özelliğe karşılık gelen bir ağırlık vektörüne sahiptir. Orta katman sayısı 100 olan AE'nin orta katman çıkışı görsel olarak Şekil 10'da verilmektedir.



Şekil 10. AE'nin orta katman çıkışları

İkinci bir AE uygulanarak 100 boyutuna indirgenen özellik sayısı 50'ye indirgenmiştir. Bu 50 özellik ise ağa verilerek sınıflandırılmıştır. İkili Autoencoder ile 784 özellikli MNIST veri setinin farklı gizli katman hücre sayılarına göre doğruluk oranları Tablo 4'de verilmektedir.

Tablo 4. İkili AE yönteminin farklı gizli katman hücre sayısına göre sınıflandırma işlemi doğruluk oranları (%)

Giriş Veri Boyutu	AE1	AE2	Doğruluk(%)
784	100	50	99.1
	80	40	98.7
	60	30	97.6
	50	25	97.3
	40	20	96.1

MNIST veri setine 3 tane Autoencoder uygulanıp veri setinin boyutu farklı gizli katman hücre sayılarına indirgenmiştir. İndirgenen bu veri setinin sınıflandırma doğruluk oranları Tablo 5'de verilmiştir. Tablo 5'den de görüldüğü gibi sınıflandırma performanslarında düşüş yaşanmıştır. Bunun temel nedeni gizli katman sayısının artması olarak gözlemlenmiştir. Her bir oto-kodlayıcı çıkışında ister istemez bir miktar önemli sayılabilecek veri kaybı yaşanmaktadır. Üçlü ardışıl yapıda da bu kayıp oranının daha fazla olması olası bir durumdur. Bunun neticesinde doğruluk oranlarında düşüş meydana gelebilmektedir.

Tablo 5. Üçlü AE yönteminin farklı gizli katman hücre sayısına göre sınıflandırma işlemi doğruluk oranları (%)

Giriş Veri Boyutu	AE1	AE2	AE3	Doğruluk(%)
784	392	196	98	78.29
	196	98	49	78.92
	132	66	33	79.01
	98	49	25	78.64
	80	40	20	77.63
	66	33	16	76.23
	56	28	14	73.94

4. TARTIŞMA VE SONUÇ

Bu çalışmada MNIST ve sentetik verilerin kullanılarak doğrusal yöntemlerden olan PCA ve LDA'nın ayrıntılı olarak algoritmaları incelenmiştir. Ayrıca doğrusal olmayan AE yönteminin ise gizli katman hücre sayısına göre sınıflandırma doğruluk oranları incelenmiştir. Yüksek oranda doğru ve hızlı bir sınıflandırma gerçekleştirebilmek için veri tabanları üzerinde boyut indirgemesi gerçekleştirilmiş ve deneysel çalışmalar yürütülmüştür. Literatürde yapılan çalışmalarda ayırt edici özniteliği tespit edilmiş veri tabanları üzerinde doğrusal ve doğrusal olmayan boyut indirgeme tekniklerinin sınıflandırma başarıları incelenmiştir. Sentetik veriler üzerinde temel bileşen analizinin başarısının daha yüksek olduğu görülmüştür. 784 boyutunda özellik vektörüne sahip el yazısı veri seti üzerindeki sınıflandırma doğruluk değerlerine bakıldığında, PCA'nın LDA'dan daha başarılı sonuçlar verdiği gözlemlenmiştir. Ortalama 20 öznitelik ile MNIST veri setinin %90'lar üzerinde temsil edilmesi PCA'nın başarısını göstermektedir. Aynı zamanda bu iki yöntemin çalışma sürelerinin kıyaslamalarına bakılarak PCA'nın daha kısa sürede veri boyutunu indirgediği gözlemlenmiştir.

Doğrusal olmayan derin öğrenme modellerinden oto-kodlayıcı çalışma zamanı doğrusal yöntemlere göre daha yüksektir. Bunun başlıca nedeni oto-kodlayıcı mimarisinde birden fazla gizli katman ağının olması dolayısıyla hesap yükünün fazla olmasıdır. Oto-kodlayıcının gizli hücre sayısı ve seyreklik parametresi gibi hiper parametrelerinin değiştirilmesiyle bu parametrelerin öğrenme başarısı üzerindeki etkileri

gözlemlenmiştir. Elde edilen sonuçlar incelendiğinde boyut indirgemekle birlikte en yüksek başarı oranı oto-kodlayıcı ile elde edilmiştir. Deneysel sonuçlar doğrusal tekniklerin seçilen sentetik veriler üzerinde iyi performans göstermekle birlikte gerçek veriler üzerinde istenilen düzeyde olmadığını göstermektedir. Derin öğrenme yöntemlerinden Autoencoder ise gerçek veri seti üzerinde daha yüksek bir performans sergilemektedir.

KAYNAKLAR

[1] Tharwat A. Principal component analysis - a tutorial. Int J Appl Pattern Recognit. 2016.

[2] Jamal A, Handayani A, Septiandri AA, Ripmiatin E, Effendi Y. Dimensionality Reduction using PCA and K-Means Clustering for Breast Cancer Prediction. Lontar Komput J Ilm Teknol Inf. 2018.

[3] Gu Q, Li Z, Han J. Linear discriminant dimensionality reduction. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2011.

[4] Analysis LD. Introduction to LDA LDA. Cancer Lett. 2005.

[5] Ng A. "Sparse autoencoder." CS294A Lect notes 72. 2011;1(19).

[6] Çalişan M, Talu MF. Examination of the effect of the basic parameters of the auto-encoder on coding performance. In: IDAP 2017 - International Artificial Intelligence and Data Processing Symposium. 2017.

[7] MNIST Dataset [Internet]. [cited 2019 May 12]. Available from: http://yann.lecun.com/exdb/mnist/

[8] Keogh EJ, Pazzani MJ. A simple dimensionality reduction technique for fast similarity search in large time series databases. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2000.

[9] Bishop CM. Pattern Regonization and Macine Learning. Oxidation Communications. 2004.

[10] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. Data Mining: Concepts and Techniques. 2012.

[11] Tantawi MM, Revett K, Salem A, Tolba MF. Fiducial feature reduction analysis for electrocardiogram (ECG) based biometric recognition. J Intell Inf Syst. 2013.

[12] Bruce LM, Koger CH, Li J. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. IEEE Trans Geosci Remote Sens. 2002.

[13] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015.

[14] Vinjamuri R, Patel V, Powell M, Mao ZH, Crone N. Candidates for synergies: Linear Discriminants versus principal components. Comput Intell Neurosci. 2014.

[15] Runnalls AR. Kullback-Leibler approach to Gaussian mixture reduction. IEEE Trans Aerosp Electron Syst. 2007.