Derin Öğrenme Algoritmalarında Eğitim Örneklerini Derecelendirmenin Etkisinin Araştırılması

Kaan KARAKÖSE

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME ALGORİTMALARINDA EĞİTİM ÖRNEKLERİNİ DERECELENDİRMENİN ETKİSİNİN ARAŞTIRILMASI

Kaan KARAKÖSE
0000-0002-4136-7592

Doç. Dr. Metin BİLGİN
(Danışman)

YÜKSEK LİSANS
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2022
Her Hakkı Saklıdır.

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME ALGORİTMALARINDA EĞİTİM ÖRNEKLERİNİ DERECELENDİRMENİN ETKİSİNİN ARAŞTIRILMASI

Kaan KARAKÖSE

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Metin BİLGİN

Son yıllarda, derin sinir ağları, özellikle bilgisayarla görme görevleri için hem endüstri hem de akademide başarılı olmuştur. İnsanlar ve hayvanlara bilgiyi rastgele sunmak yerine aşamalı olarak daha fazla kavram ve karmaşık olanları gösteren anlamlı bir düzende sunulduğunda çok daha iyi öğrenebilirler. Bu tür eğitim stratejilerini yapay sinir ağları bağlamında kullanımına "Müfredat Öğrenme" denmektedir.

Bu çalışmada, örneklerin derecelendirilmesi ve yapay sinir ağları ile kullanımından oluşan müfredat öğrenme için yeni bir yaklaşım sunulmuştur. Çalışmamızda CIFAR-10 ve CIFAR-100 eğitim kümesiyle önceden eğitilmiş öğretmen modelinin (Xception modeli) son altı katmanı eğiterek eğitim kümesine ait gizli bilgilerin ağırlıklarda tutulması sağlanmıştır. Model damıtma yöntemleriyle, her bir örnek için oluşacak katman çıktılarını örnek ve örneğe ait etiket bilgilerini bir işleme tabi tutarak her bir örnek için bir zorluk derecesi üretilmiştir. Örnek ve modelin aynı boyutta ayarlanmış sondan bir önceki katman çıktısıyla arasındaki ilgileşim (korelasyon) örnek kaybını, model damıtma kullanarak örneğin etiket bilgisi ve modelin son katman çıktısıyla oluşan çapraz entropisi ile etiket kaybı hesaplanmaktadır. Bu zorluk derecelerini kullanarak oluşturduğumuz 4 farklı sıralama yöntemli ile çok daha küçük bir model eğitilmiştir. Gelişigüzel karıştırılmış verilerle eğitilen öğrenci modelinin doğruluk değeriyle sunduğumuz dört farklı yöntemden ikisi ile oluşturulan sıralama ile eğitildiğinde oluşan doğruluk değerinin her bir devir için daha iyi sonuç verdiği tespit edilmiştir. Öğrenci modeli için doğruluk değerinin anlamlı olabilmesi için her bir durum için on kez çalıştırarak ortalama doğruluk değerleri kullanılmıştır.

Her iki veri kümesi için başarı gösteren sınıf bazlı küçükten büyüğe (SBKB) ve sınıf bazlı büyükten küçüğe (SBBK) yöntemleri ile sistem başarısını arttırmak mümkündür. Küçük cihazlarda (mikroişlemciler, mobil telefon vb.) sunduğumuz bu yöntemler kullanılabilir ve daha az işlem gücüne sahip cihazlar ve küçük modeller için iyileştirmeler gözlenebilir.

Anahtar Kelimeler: Müfredat Öğrenme, Model Damıtma, Derin Öğrenme
2022, vii + 44 sayfa.

ABSTRACT

MSc Thesis

INVESTIGATION OF THE EFFECT OF GRADING EDUCATIONAL EXAMPLES ON DEEP LEARNING ALGORITHMS

Kaan KARAKÖSE

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Metin BİLGİN

In recent years, deep neural networks have been successful in both industry and academia, especially for computer vision tasks. Humans and animals learn much better when gradually presented in a meaningful order that shows more concepts and complex ones rather than randomly presenting the information. The use of such training strategies in the context of artificial neural networks is called "Curriculum Learning".

In this study, a new approach to curriculum learning, which consists of grading examples and their use with artificial neural networks, is presented. With the CIFAR-10 & CIFAR-100 training set, we train the last six layers of the pre-trained teacher model (Xception Model) to keep the training set's hidden information in the model's weight. We produced a difficulty level for each sample with model distillation methods by processing the layer outputs and each sample and its label. We calculate the sample loss from the similarity (correlation) between the sample and the model's penultimate layer output set at the same size and, we calculate label loss with the sample's ground truth information and the final layer output of the model by using cross-entropy loss.We trained a much smaller model called the student model with 4 different sample sequencing methods we created using these difficulty levels. We determined that the accuracy value generated when trained by the 2 of 4 methods we provide with the accuracy value trained with randomly mixed dataset is more than 1% for each epoch. For the accuracy value to be meaningful for the student model, we used the average accuracy running ten times.

It is possible to increase the model’s accuracy with class-based small to large (CBSL) and class-based large to small (CBLS), which are successful for both datasets. These methods we offer in small devices (microprocessors, mobile phones, etc.) can be used, and improvements can be observed for devices with less processing power limits and small models.

Key words: Curriculum Learning, Model Distillation, Deep Learning,
2022, vii + 44 pages.

TEŞEKKÜR

Tez çalışmamın planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini hiçbir zaman esirgemeyen, bilgi ve tecrübelerinden yararlandığım, yönlendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren değerli danışmanım Doç. Dr. Metin BİLGİN'e sonsuz teşekkürlerimi sunarım.

Çalışmalar sırasında fikir alışverişinde bulunarak yaşadığım problemlerin çözülmesine katkı sağlayan değerli meslektaşım Arş. Gör. Ahmet DEMİR'e ve değerli arkadaşım Korhan MUTLUDOĞAN'a teşekkürü bir borç bilirim.

Üniversite yıllarımdan itibaren vermiş olduğu destekler, yol gösterici tavsiyeler sayesinde akademik kariyerime vesile olan değerli arkadaşım ve meslektaşım Samet Süleyman DOĞAN'a teşekkürü bir borç bilirim.

Ve son olarak hayatımın her anında desteğini benden hiçbir zaman esirgemeyen bu sebeple teşekkürü en çok hak eden başta sevgili annem Mürüvet KARAKÖSE'ye, değerli ağabeyim Süleyman KARAKÖSE'ye ve sevgili kardeşim Sudem KARAKÖSE'ye en içten teşekkürlerimi ve sevgilerimi sunarım.

Kaan Karaköse
07/02/2022

İÇİNDEKİLER

Sayfa

ÖZET i
ABSTRACT ii
ÖNSÖZ VE/VEYA TEŞEKKÜR iii
SİMGELER ve KISALTMALAR DİZİNİ v
ŞEKİLLER DİZİNİ vi
ÇİZELGELER DİZİNİ vii
1. GİRİŞ 1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI 3
2.1. İleri Beslemeli Sinir Ağları 3
2.2. Evrişimsel Sinir Ağları 10
2.3. Müfredat Öğrenme 13
2.4. Müfredat Öğrenme İle İlgili Yapılan Çalışmalar 15
2.4.1. Evrişimsel sinir ağlarında müfredat öğrenme kullanımı 15
2.4.2. Tekrarlayan sinir ağlarında müfredat öğrenme kullanımı 16
2.5. Bilgi Damıtma 17
3. MATERYAL VE YÖNTEM 20
3.1. Veri Kümesi 20
3.2. Müfredat Oluşturma 22
3.2.1. Öğretmen modelinin eğitilmesi 22
3.2.2. Katmanlardan bilgi damıtma 25
3.2.3. Zorlukların sıralanması 28
3.2.4. Öğrenci modelinin eğitilmesi 29
4. BULGULAR 31
5. TARTIŞMA ve SONUÇ 39
KAYNAKLAR 41
ÖZGEÇMİŞ 45

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	**Açıklama**
X	Veri kümesi Vektörü
Y	Veri kümesi Etiket Vektörü
w	Ağırlık Değeri
z	Katman Çıktısı
b	Yanlılık Değeri
σ	Aktivasyon Fonksiyonu
W	Ağırlık vektörü
Z	Katman Çıkış Vektörü
B	Yanlılık Vektörü
t	Zaman
$*$	Konvolüsyon İşlemi
T	Sıcaklık Parametresi
K	Katman
N	Örnek Sayısı
$\bar{X}$	X'e ait Ortalama

Kısaltmalar	**Açıklama**
YSA	Yapay Sinir Ağı
ReLu	Rectified Linear Unit
CNN	Evrişimsel Sinir Ağı
RNN	Tekrarlayan Sinir Ağı
NMT	Sinirsel Makine Çevirisi (Neural Machine Translation)
CEL	Çapraz Entropi Kaybı (Cross Entropy Loss)
TL	Transfer Öğrenimi (Transfer Learning)
SBBK	Sınıf Bazlı Büyükten Küçüğe
SBKB	Sınıf Bazlı Küçükten Büyüğe

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Bir nöron hücresinin anatomik yapısı 4
Şekil 2.2. İki girişli bir perseptron modeli 6
Şekil 2.3. Gizli katmanında üç adet perseptron bulunan bir yapay sinir ağı örneği 8
Şekil 2.4. Geri yayılım 9
Şekil 2.5. İki boyutlu dizilerde evrişim işleminin uygulaması 11
Şekil 2.6. En yüksek ortaklama işlemi 12
Şekil 2.7. Derin evrişimsel sinir ağı 13
Şekil 2.8. Bilgi damıtma için genel bir öğretmen-öğrenci çerçevesi 18
Şekil 2.9. Genel yanıta-dayalı bilgi damıtma 19
Şekil 3.1. CIFAR-10 veri kümesine ait örneklerin sınıflarla gösterimi 21
Şekil 3.2. Öğretmen modeli eğilen katmanlar 24
Şekil 3.3. Öğretmen modeli bilgi damıtma katmanları 26
Şekil 3.4. Sınıf bazlı sıralama 28
Şekil 3.5. Öğrenci modeli 29
Şekil 4.1. Sıralanmış CIFAR-10 veri kümesine ait örnekler 31
Şekil 4.2. Sıralanmış CIFAR-100 veri kümesine ait örnekler 32
Şekil 4.3. CIFAR-10 veri kümesi üzerinde öğrenci modeline ait doğruluk grafiği 34
Şekil 4.4. CIFAR-100 veri kümesi üzerinde öğrenci modeline ait doğruluk grafiği 36

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1. Yapay sinir ağlarında sık kullanılan aktivasyon fonksiyonları ve kapsama aralıkları 7
Çizelge 3.1. ImageNet üzerinde eğitilmiş ağlar 23
Çizelge 3.2. Öğretmen Modeline ait eğitim parametreleri..... 25
Çizelge 4.1. CIFAR-10 veri kümesi üzerinde öğrenci modeline ait doğruluk değerleri 33
Çizelge 4.2. CIFAR-100 veri kümesi üzerinde öğrenci modeline ait doğruluk değerleri 35
Çizelge 4.3. CIFAR-10 için sıralama yöntemlerinin karıştırılmış duruma göre farkları 37
Çizelge 4.4. CIFAR-100 için sıralama yöntemlerinin karıştırılmış duruma göre farkları 37

1. GİRİŞ

Son yıllarda, derin sinir ağları, özellikle bilgisayarla görme görevleri için hem endüstride hem de akademide başarılı olmuştur. Derin öğrenmenin büyük başarısı, büyük ölçekli verileri kodlama ve milyarlarca model parametrelerini manipüle etme konusundaki ölçeklenebilirliğinden kaynaklanmaktadır. Gelişen teknolojinin ve bilimin ilham kaynağı doğa ve doğa temelli olgular olmuştur. Yapay sinir ağlarında kullanılan temel düşünce, beyin sinir hücreleri olduğundan süreç içerisinde yapay sinir ağlarına insan ve doğaya dayalı geliştirmeler yapılmış ve yapılmaya devam etmektedir. İnsanlara ve hayvanlara bilgiyi rastgele sunmak yerine aşamalı olarak daha fazla kavram ve karmaşık olanları gösteren anlamlı bir düzende sunulduğunda çok daha iyi öğrenirler. Yapay sinir ağlarında buna benzer eğitim stratejilerinin uygulanmasına "Müfredat Öğrenme" (Curriculum Learning) (Bengio ve ark. 2009) denmektedir. Veri kümesine ait örnekler bir müfredata göre şekillendirildiğinde model daha iyi bir yerel minimum noktası bulur (Bengio ve ark. 2009). Farklı yaklaşımlar da eğitim örneklerinin bir müfredata dönüştürülmesi ve modele ait metrikler tartışılmıştır (Platanios ve ark. 2019, J. Wang ve ark. 2018, Y. Wang ve ark. 2019, Yamashita ve Watasue 2014, Zaremba ve Sutskever 2014). Müfredat Öğrenme sayesinde eğitim aşamasında örnekler için farklı ön işleme süreçlerinden söz etmek mümkün olabilmiştir.

Bu çalışma kapsamında, eğitim örneklerinin sıralanışları üzerinde durulmuştur. Sistem kurgusu için iki farklı evrişimsel sinir ağı modeli kullanılmıştır. Bu modeller sırasıyla "öğretmen" ve "öğrenci" modeli olarak adlandırılmıştır. Öğretmen modeli için fazla sayıda eğitilebilir model parametresi sayısına sahip "Xception" modeli (Chollet 2017) kullanılırken, öğrenci modeli için ise öğretmen modelinden çok daha düşük sayıda model parametresine sahip, basit bir evrişimsel bir sinir ağı tercih edilmiştir. Öğrenci modelinin daha küçük olmasının nedeni, örnek zorluklarının model üzerindeki etkisinin daha açık ve kolay bir şekilde gözlemlenebilmesine olanak sağlamasıdır.

CIFAR-10 (Krizhevsky 2009) ve CIFAR-100 (Krizhevsky 2009) veri kümeleri üzerinde çalışmalar gerçekleştirilmiştir. Bu veri kümeleri farklı örneklere ve yeterli sınıf sayısına sahip olması, örnek boyutlarının küçük olması böylece hesaplama maliyetlerinin daha az

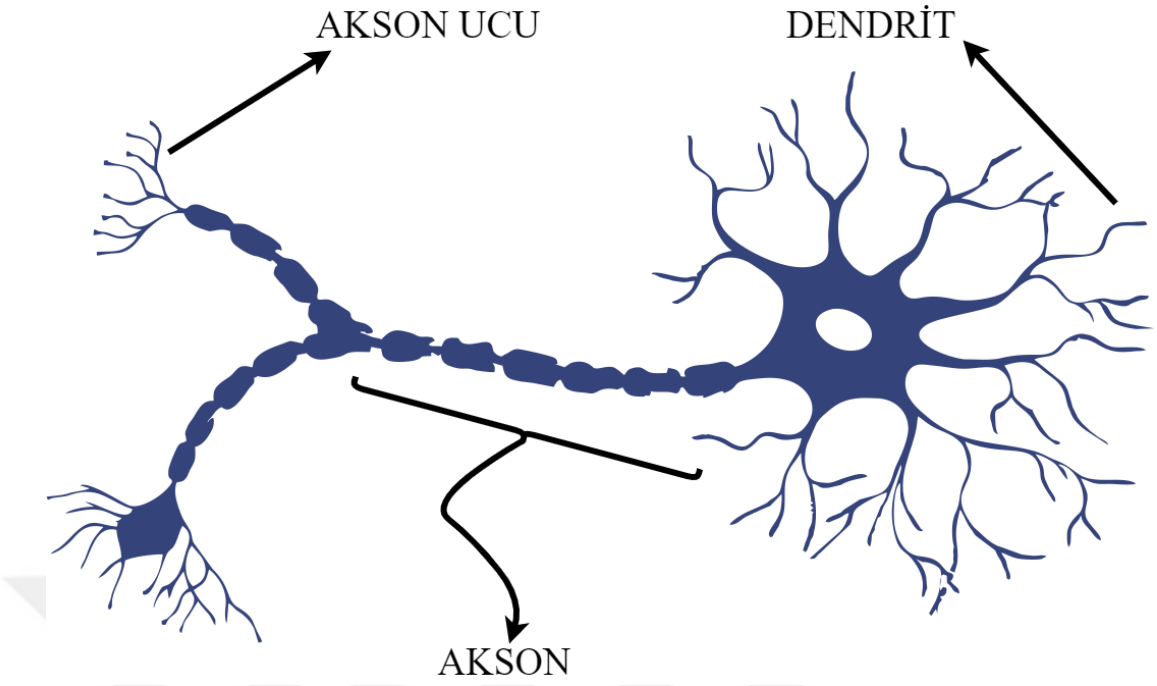
olmasından dolayı seçilmiştir. Ayrıca bu tez kapsamında savunulan hipotezin, literatürde sıklıkla kullanılan bu veri kümelerinin üzerindeki etkisinin incelenmesi bu çalışma kapsamında önemli bir yere sahiptir. Model damıtma (Hinton ve ark. 2015) yöntemi kullanılarak eğitim örneklerinin derecelendirmesi için yeni bir yaklaşım sunulmuştur. Model katmanlarında veri kümesine ait bilgilerin damıtılmasıyla elde edilen bilginin kullanımıyla bir müfredat oluşturma hedeflenmiştir. Bu çalışmada eğitim örnekleri statik olarak farklı sıralamalar elde edilerek, bu sıralamaların model başarısı üzerine olan etkisi incelenmiştir, yöntemler birebiriyle karşılaştırılmış ve iyi sonuç verenler tartışılmıştır.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Yapay sinir ağları biyolojik beyin sinirlerinden esinlenerek tasarlanmış hesaplama sistemleridir. Yapay sinir ağları temelde biyolojik sinir sisteminin çalışma prensibini taklit eder. Yapay sinir ağları bir görevi öğrenme, bilgiyi hafızada tutma ve veriler arasındaki ilişkileri çözümleme gibi kabiliyetlere sahiptir. Yapay sinir ağları, bir insanın düşünmeye ve tecrübeye bağlı gözlemler gerektiren problemlere çözüm üretmeyi amaçlamaktadır. Son yıllarda gelişen teknolojinin her alanında kullanılan yapay sinir ağı dayalı sistemler birçok bilimsel ve endüstriyel probleme çözüm oluşturmuştur, gelinen noktada ise insanın günlük hayatının her alanına girmiştir.

2.1. İleri Beslemeli Sinir Ağları

Gelişmiş biyolojik canlılardaki beyin hücrelerinin temelinin oluşturan nöronlar, yapay sinir ağlarındaki temel hesaplama birimlerine ilham kaynağı olmuştur. Nöronlar beynin temel hesaplama birimidir ve diğer nöronlardan gelen sinyalleri işleyerek bir çıkış sinyali üretip üretemeyeceğine karar verme yeteneğine sahiptirler. Şekil 2.1'de bir sinir hücresi gösterilmiştir.



Şekil 2.1. Bir nöron hücresinin anatomik yapısı
(Anonim2021a'dan değiştirilerek alınmıştır)

Ortalama bir insan beyninde her biri 10 bin farklı nöronla bağlantılı 100 milyar tane nöron hücresi bulunmaktadır (Herculano-Houzel 2009). Nöronlar trilyonlarca sinaptik ağlar sayesinde sinyalleri birbirlerine iletirler (Herculano-Houzel 2009). Bu ağların işlevleri, oluşan bağların kuvveti ve yapısıyla belirlenmektedir. Beyinde bilgi iletişimi elektro-kimyasal bir süreç ile yönetilir. Bir nöron hücresinin çıkışı sürekli bir sinyaldir. Nöronlar aslında hücre içi gerilimleri (Membran potansiyeli), frekans olarak çıktı vererek bir gerilim-frekans dönüştürücü olarak davranmaktadır (Anderson ve Rosenfeld 1993). Nöron hücresine yüksek gerilim geldiğinde çıkış sinyalinin frekansını arttırırken genliğini korur. Nöron, uyarıcı veya engelleyici girdilerinin toplam değerine göre çıktı üretecektir veya üretmeyecektir. Üretilen değer nöronun eşik (threshold) değerinden büyükse, sinyal akson vasıtasıyla yayılır ve akson (sinaptik) terminalleri aracılığıyla diğer nöronların girişlerine iletilir.

Matematiksel sinir modeli ilk kez Nörofizyolog Warren McCulloch ve matematikçi Walter Pitts tarafından tasarlamıştır (McCulloch ve Pitts 1943). Tasarlanan bu model, 2

giriş ve 1 çıkıştan oluşan bir elektrik devresi olarak tasarlanmıştır. Giriş sinyalleri iki durumlu bir sinyal (0 veya 1) ve ağırlık parametresi kullanılmamıştı.

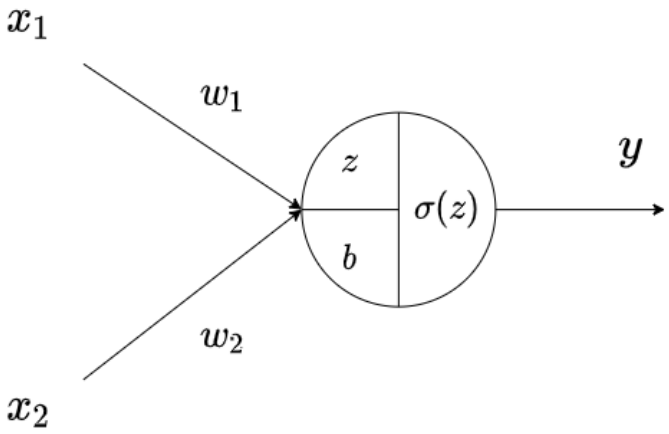
$$f(y) = \begin{cases} 1, & y \geq 0 \\ 0, & y < 0 \end{cases} \tag{2.1}$$

Denklem 2.1'de, bu model için kullanılan aktivasyon fonksiyonu gösterilmiştir. y değerinin sıfırdan büyük ve eşit olduğu değerler için bu fonksiyon 1 değerini üretirken, y'in sıfırdan küçük olduğu değerler için 0 değerini üretmektedir. Burada, y değeri ise girdi işaretlerinin toplamlarını Denklem 2.2'de olduğu şekilde hesaplanmış halidir.

$$y = \sum_{i=1}^{d} x_i \tag{2.2}$$

Bu model ilkel bir model olduğu için Denklem 2.2'de görüleceği üzere bir ağırlıklandırma söz konusu değildir. İlerleyen çalışmalarla birlikte giriş sinyalleri rasgele dağılmış sayılarla çarpılması sonucunda daha karmaşık matematiksel ifadeler elde edilerek, daha iyi sonuçlar elde edilmeye başlanacaktır.

Warren McCulloch ve Walter Pitts'in beynin çalışma prensibinden esinlenerek yapmış oldukları bu çalışmadan sonra yapay sinir ağları bilim dünyasına girmiş, David C. Plaut ve Geoffrey Hinton'nın geri yayılım (Plaut ve Hinton 1987) çalışmasıyla birlikte hız kazanarak gelişimine devam etmiştir. Nöron hücresinin matematiksel modellemesine perseptron (perceptron) denmektedir (Minsky ve Papert 1969).



Şekil 2.2. İki girişli bir perseptron modeli (Anonim2021b'den değiştirilerek alınmıştır)

Şekil 2.2'de bir perseptron gösterilmektedir. Perseptronlar, Denklem 2.3'de gösterilen matematiksel ifade ile girdi sinyallerini genellikle rasgele belirlenmiş katsayılarla çarparak toplar ve bir aktivasyon fonksiyonu sayesinde çıkış üretir. Burada x_1, x_2veri setine ait bir örnekteki noktaları, σ ile gösterilen aktivasyon fonksiyonu, b yanlılığı (bias), w (weight) ise ağırlık vektörünü ifade eder. Çizelge 2.1'de yapay sinir ağlarında kullanılan aktivasyon fonksiyonlarına yer verilmiştir.

$$z = x_1 w_1 + x_2 w_2$$
$$y = \sigma(z) + b \tag{2.3}$$

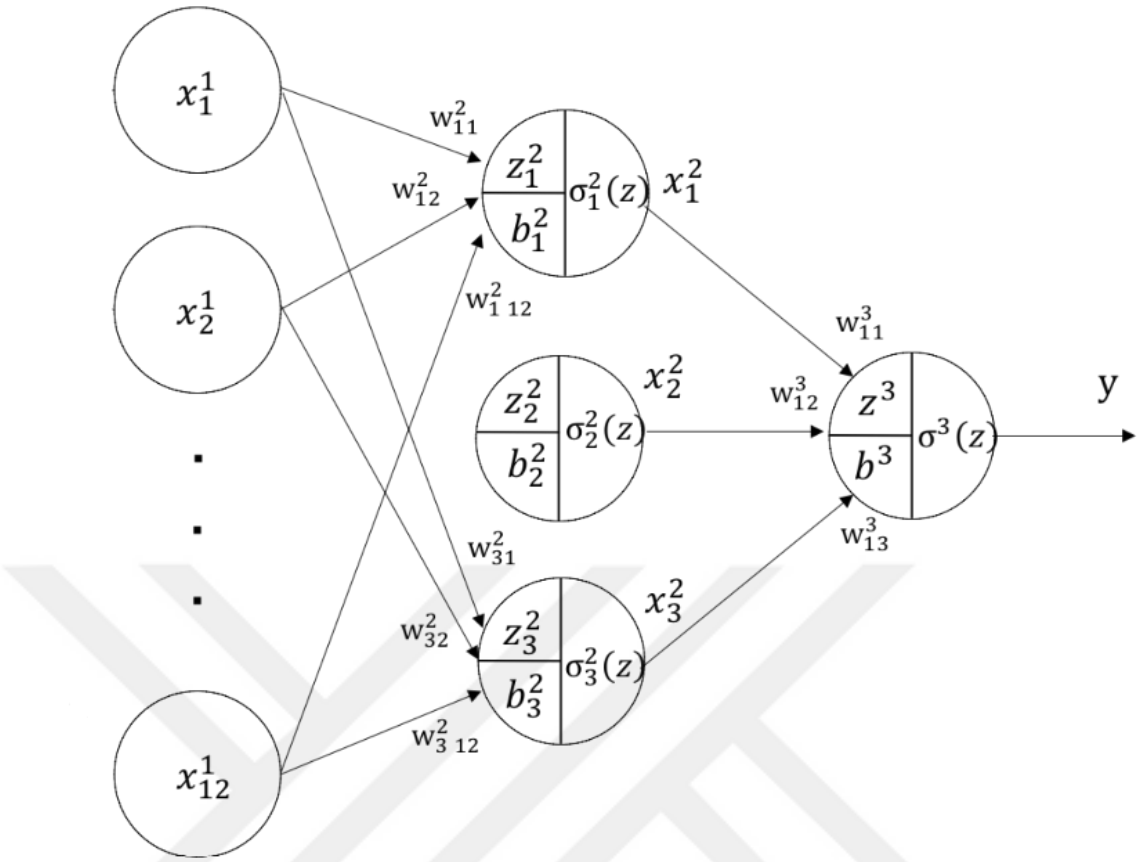
Perseptronların birleşmesiyle oluşan Şekil 2.3'de gösterilen yapay sinir ağlarında, her bir perseptron ağırlık katsayıları ile çarpılıp girdi olarak gelen sinyalleri bir aktivasyon fonksiyonundan geçirerek kendisi için çıkış olan bu değeri diğer katmandaki perseptrona girdi olarak iletir. Bu işlem çıkış katmanına kadar devam eder. Denklem 2.4'de bu işlemler sırasında oluşan ileri yayılım işlemlerinin matematiksel ifadesi Şekil 2.3'de gösterilen yapay sinir ağı için oluşturulmuştur.

Çizelge 2.1. Yapay sinir ağlarında sık kullanılan aktivasyon fonksiyonları ve kapsama aralıkları

Aktivasyon Fonksiyonu	**Denklem**	**Kapsadığı Aralık**
Birim Basamak	$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$	{0,1}
Doğrusal	$f(x) = x$	(-∞,∞)
ReLu (Rectified Linear Unit)	$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$	[0,∞)
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	(0,1)
Hiperbolik Tanjant	$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	(-1,1)

$$
\begin{aligned}
Z^1 &= X.W^1 \\
Z^2 &= \sigma^2(Z^1) + B^1 \\
Z^3 &= Z^2 W^2 \\
y &= \sigma^3(Z^3) + B^2
\end{aligned}
\tag{2.4}
$$

Denklem 2.4'de büyük harf kullanılarak değişkenlerin vektör olduklarını ve üstel kullanılan ifadeler ise ait olduğu katman bilgisini göstermektedir. Tüm bu işlemler sonucunda her bir örnek için bir reel sayı üretilecektir. Yapay sinir ağı modellerindeki kullanılan öğrenme modeli, üretilen sonucun olması gereken sonuçla farkından (hata oranı) geri yayılım yapılarak (Plaut ve Hinton 1987) en uygun şekle sokma işlemidir; genelde rasgele seçilerek oluşturulan ağırlık değerlerinin güncellenmesiyle anlamlı sonuçlar üretilene dek bu süreç devam eder. Denklem 2.5'de parametrelerin yani ağırlıkların güncellemesi için yapılan işlem gösterilmektedir.



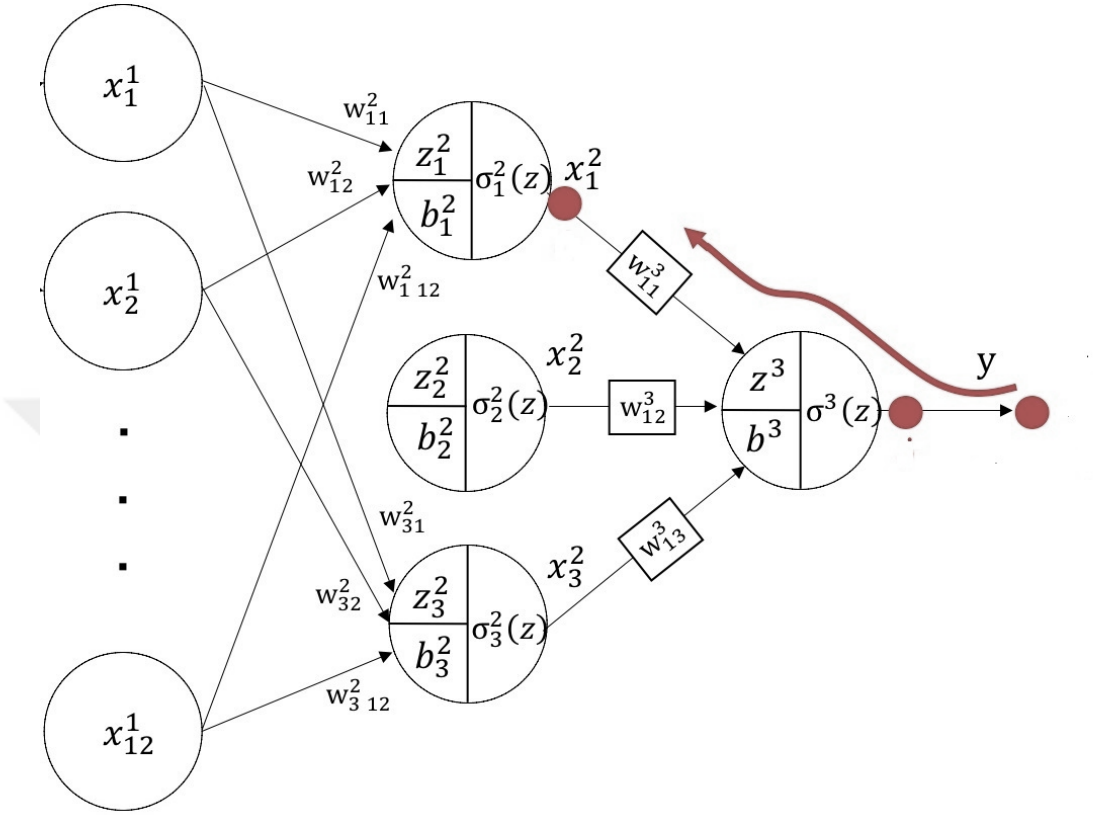
Şekil 2.3. Gizli katmanında üç adet perseptron bulunan bir yapay sinir ağı örneği (Anonim2021b'den değiştirilerek alınmıştır)

$$w_{t+1} = w_t - \eta \frac{\partial L}{\partial w}$$
$$b_{t+1} = b_t - \eta \frac{\partial L}{\partial b} \tag{2.5}$$

Denklem 2.5'de w ağırlığı, t epoğu, η öğrenme katsayısını ve L ise kayıp fonksiyonunu ifade etmektedir. $\frac{\partial L}{\partial w}$ ve $\frac{\partial L}{\partial b}$ ifadeleri gradyan olarak tanımlanmaktadır. Burada L fonksiyonu ile aktivasyon fonksiyonları problem tanımına göre regresyon veya sınıflandırma için seçilebilir.

Şekil 2.4'de gösterilen geri yayılım için, Denklem 2.6'de w_{11}^3 parametresine ait türev ifadesi gösterilmiştir. Diğer parametreler için geri yayılım ifadesi Denklem 2.5 ve zincir kuralı kullanılarak bulunabilmektedir.

$$\frac{\partial L}{\partial w_{11}^{3}} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial Z^{3}}\frac{\partial Z^{3}}{\partial w_{11}^{3}} \tag{2.6}$$



Şekil 2.4. Geri yayılım
(Anonim2021b)'den değiştirilerek alınmıştır)

Geri yayılım işlemiyle bulunan gradyanlar kullanılarak ağ parametreleri güncellenir ve ileri besleme işlemiyle kayıp fonksiyonun yerel minimum noktası bulununcaya kadar devam eder. Yerel minimum noktası bir fonksiyon için türevinin başka bir deyişle gradyan değerinin sıfır olduğu noktadır. Burada türevi olan bir fonksiyonun yerel minimumunu bulmak için, birinci dereceden yinelemeli en uygun şekle sokma algoritması olan gradyan iniş (Curry 1944) kullanılmaktadır. Son 20 yılda bu fikir geliştirilmiş ve iyileştirilmiştir. Gradyan iniş algoritmasının geliştirilmiş ve özelleştirilmiş halleri, Stokastik Gradyan İniş (Stochastic gradient descent, SGD), Momentum Yönetimi, RMSprop ve Adam (Kingma ve Ba 2014) algoritmaları güncel olarak kullanılan algoritmalardandır (Ruder 2016).

Denetimli öğrenmelerde çoğu zaman gradyan inişi yöntemleriyle bulunan nokta, problem için bir çözüm üretse de bazı problemler için yerel minimum noktasına takılı kalarak daha iyi bir çözüm üretememektedir. Kullanılan veri setine göre daha geniş ve özelleştirilmiş yapılardaki yapay sinir ağı modelleri kullanılarak daha iyi çözümler üretilebilir. Evrişimsel Sinir Ağı (Convolutional Neural Network, CNN) genellikler görüntülerde başarılı olan model topolojisi iken, Tekrarlayan Sinir Ağları (Recurrent Neural Network, RNN) zaman serileri için başarılı olan model topolojisidir (De Mulder ve ark. 2015).

2.2. Evrişimsel Sinir Ağları

İleri yapay sinir ağları görsel bilgilerle uğraşırken iyi performans gösteremezler. Bunun nedenlerinden biri bilgisayarla görme uygulamaları, bölgesel değerlerin birbiri ile ilgileşimi (korelasyon) olan, çok boyutlu verilerle uğraşmasıdır. Her bir piksel için bir nöron atamak ise girdi olarak kullanılacak görüntünün boyutlarına bağlı olarak katlanarak artacaktır. Bu ağın eğitilmesini olanaksızlaştırır. Bir diğer zorluk ise ağın girdi içinde farklı noktalardaki aynı yapıyı tanıyamamasıdır. Çünkü her bir nöron birbirinden kabaca bağımsız sadece bir piksel için öğrenme adımlarını gerçekleştirir. Bu zorluklarla başa çıkabilmek için görüntü işleme uygulamalarında sıklıkla kullanılan evrişim (konvolüsyon) fikri yapay sinir ağlarında kullanılmaya başlanmıştır. Huber ve Wiesel'in kedilerin görsel korteksiyle ilgili yaptıkları çalışmaları (Hubel ve Wiesel 1962), evrişimsel sinir ağlarının gelişmesinde büyük ölçüde etkili olduğu söylenebilir. İlk kez evrişimsel sinir ağı fikri Fukushima tarafından ortaya atılmıştır (Fukushima 1975).

Matematiksel terim olarak evrişim, bir fonksiyonun başka bir fonksiyonun çıktısı üzerine tekrar tekrar uygulanmasını temsil eder. Görüntü işleme uygulamalarında ise filtreleme aşamasında kullanılan bir matematiksel operatördür. Denklem 2.7'de evrişim işleminin iki fonksiyon üzerine nasıl uygulanacağına dair matematiksel gösterimine yer verilmektedir. Burada $*$ evrişim işlemini, τ ve t ise fonksiyonlara ait bağımsız değişkenleri ifade eder.

$$(f * g) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.7}$$

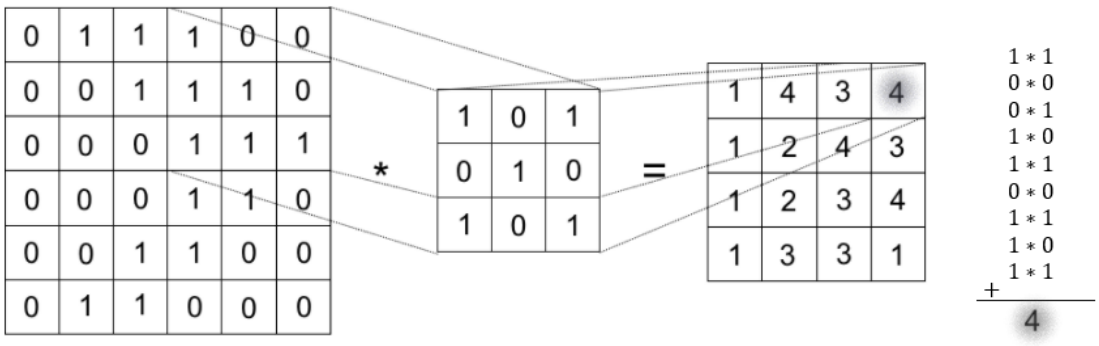
Fakat bu işlem sürekli fonksiyonlar için geçerlidir. Bir imgeyi piksellerden oluşan bir matris olarak düşünecek olursak, sürekli bir yapıda olduğundan söz edemeyiz. Ayrık sinyaller için kullanılan matematiksel ifade Denklem 2.8'de gösterilmiştir.

$$(f * g) = \sum_{\tau} f(\tau)g(t-\tau) \tag{2.8}$$

Evrişimsel sinir ağlarında, konvolüsyon katmanı öğrenilebilir filtre katsayılarından oluşmaktadır. Bunlar filtrenin katsayıları yani ağırlık parametreleri (w) ve yanlılık (b) olarak adlandırılır. Kullanılan her bir filtre giriş işaretiyle evirilerek bir sonraki katmana aktarılır. Denklem 2.9'da bu işlemi gerçekleştiren matematiksel ifade gösterilmiştir.

$$X * F(x,f) = \sum_{m}\sum_{n} X(m,n)F(x-m, f-m) \tag{2.9}$$

Denklem 2.9'da X girdiyi, F filtreyi, x ve f girdi ve filtreye ait bağımlı değişkeni ifade etmektedir. Şekil 2.5'de evrişim işleminin uygulanması gösterilmiştir.



Şekil 2.5. İki boyutlu dizilerde evrişim işleminin uygulaması

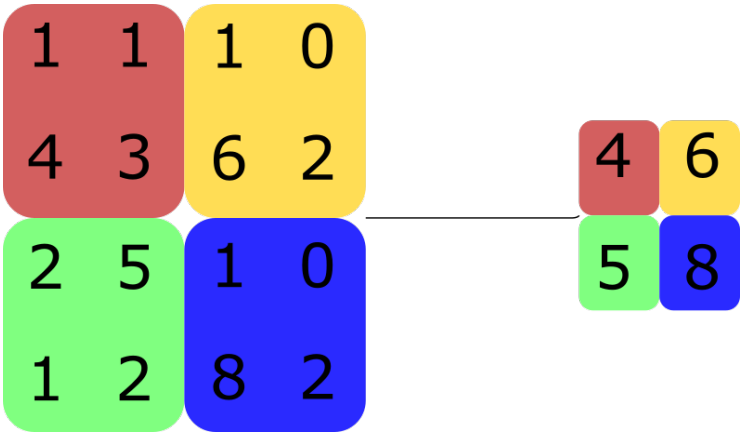
Görüntü işleme alanında yaygın olarak kullanılan filtreler, keskinleştirme, bulanıklaştırma, güçlendirme vb. sıralanabilir. Evrişimli sinir ağlarında kullanılan

filtreler girdinin özellik matrislerini oluşturur ve bu matrisler girdinin karakteristik özelliklerini öğrenir.

Evrişimsel sinir ağlarında belirli bir stride (offset) ve padding işlemleri uygulanabilir. Stride değeri arttırıldığında bir sonraki katmanı temsil edecek çıktının boyutu küçülür. Bu iki parametre çıktının boyutunu doğrudan etkiler. Denklem 2.10'da evrişim işleminde stride ve padding kullanımıyla oluşan matrisin boyutlarının ne olacağını ifade etmektedir. Burada e ve b indisleri, oluşacak olan matrisin en ve boyunu ifade etmektedir.

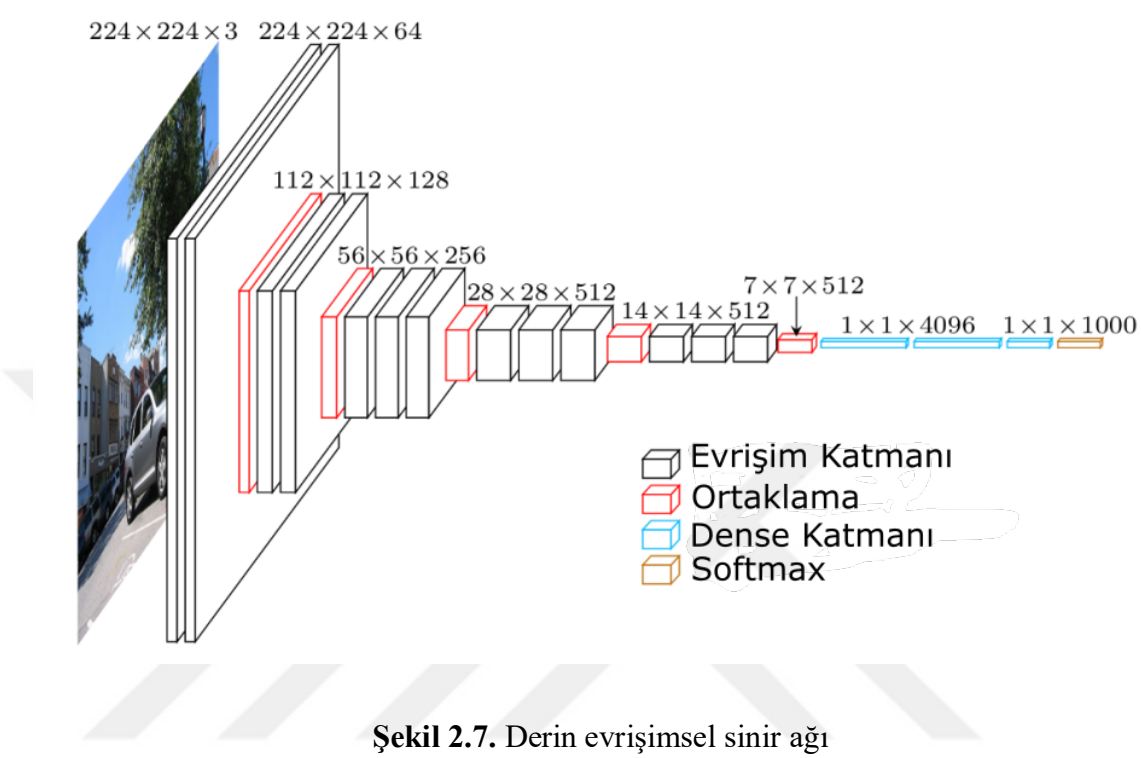
$$\left[\frac{X_e - F_e + P_e + S_e}{S_e}, \frac{X_b - F_b + P_b + S_b}{S_b}\right] \tag{2.10}$$

Evrişimsel sinir ağlarında kullanılan bir diğer katman ise ortaklama (pooling) katmanıdır. Bir önceki katmanın boyutlarını her iki boyutta azaltmak için kullanılır. Sıklıkla kullanılan yöntemler en yüksek ortaklama (max-pooling) ve ortalama ortaklama (avearage pooling) yöntemleridir. En yüksek ortaklama işleminde en büyük değer seçilirken, ortalama ortaklama işleminde pencerenin ortalama değeri alınmaktadır. Bu işlem girdinin boyutlarını azaltırken aynı zamanda yumuşatılmasını da sağlamaktadır. Şekil 2.6'de en yüksek ortaklama işlemi gösterilmektedir.



Şekil 2.6. En yüksek ortaklama işlemi

Genel olarak, evrişimsel sinir ağlarında konvülosyon katmanı, bu katmana ait bir aktivasyon fonksiyonu ve (sub-sampling) katmanlarının, birbiri ardına seri bağlanarak kullanılır. Şekil 2.7'de büyük ölçekli imgeler için derin evrişimsel sinir ağı gösterilmiştir.



Şekil 2.7. Derin evrişimsel sinir ağı
(Anonim2021c'den değiştirilerek alınmıştır.)

Bu ağlarda geri yayılım yoluyla gradyan temelli öğrenmenin kullanımı, Rumelhart, Hinton ve Williams (Rumelhart ve ark. 2013) çalışmasında ve el yazısı veri kümesinde evrişimsel sinir ağı kullanılan çalışmasında (Lecun ve ark. 1990) görülmüştür. Evrişimsel sinir ağlarının tanımlayıcı özellikleri arasında; seyrek bağlanabilirlik, paylaşılan parametreler, ortaklama katmanları, alt örnekleme katmanları olarak sıralanabilir.

2.3. Müfredat Öğrenme

İnsanlar ve hayvanlara öğrenim süreçlerinde bilgiyi karışık sunmak yerine aşamalı olarak daha fazla kavram ve karmaşık olanları gösteren anlamlı bir düzende sunulduğunda çok daha iyi öğrendikleri bilinmektedir. İnsanların, toplum içerisinde tamamen işlevsel yetişkinlik seviyesinde eğitilmeleri için yaklaşık yirmi yıla ihtiyaç vardır. Bu süreç, bir eğitim sistemine ve farklı zamanlarda farklı kavramları tanıtan bir müfredata dayalı

olarak ilerletilmektedir. Bu müfredat önceden öğrenilen kavramları kullanarak yeni soyutlamaları öğrenmeyi kolaylaştırmak için kullanılır. Bu tür eğitim stratejilerini yapay sinir ağları bağlamında kullanımına "Müfredat Öğrenme" (Curriculum Learning) (Bengio ve ark. 2009) denmektedir. Bu fikir makine öğrenmesi ile kesişimden önceki yapılan çalışmalarda (Elman 1993; Rohde ve Plaut 1999; Krueger ve Dayan 2009) araştırmacıların sordukları soru şuydu;

- Makine Öğrenmesi algoritmaları benzer bir eğitim stratejisinden yararlanabilir mi?

Bu soru zaman içerisinde araştırmacılar tarafından irdelenmiştir. Bu soruya cevap niteliği taşıyan çalışma (Bengio ve ark. 2009) ilk kez derin yapay sinir ağlarında müfredat oluşturmayı sezgisel olarak gerçekleştirebilmiştir. Bu çalışmada kullanılan veri kümesindeki örnekler sezgisel olarak bir sırayla modele öğretilmiştir. Kullanılan veri kümesi geometrik şekillere ait görsellerden oluşan bir veri setidir. Araştırmacılar, bu geometrik şekillerin insan için de zor olduğunu düşündükleri eşkenar dörtgen, paralel kenar vb. örnekleri modele daha sonra sunarak müfredatı oluşturmayı denemişlerdir. İlk etapta modele; kare, dörtgen ve üçgen gibi daha kolay şekiller ağa sunulmuştur. Bu strateji ile model daha iyi bir yerel minimum noktası bulmuş ve daha iyi başarım göstermiştir.

Eğitim örneklerinin önceden sıralanmadan eğitim aşamasında şekillendirmeye dayanan bir diğer çalışma (Kumar et al. 2010) kendini planlayan öğrenme (Self-paced learning) şeklinde tanıtılan öğrenme yöntemidir. Bu stratejide kullanılan model, eğitim örnekleriyle bir kaç devir (epoch) eğitilerek "warm-up" olarak ifade edilen bir ısındırma evresi gerçekleştirilir. Bu işlemden sonra örneğin ileri geçişinde oluşacak olan kayba göre ağırlıklar güncellenir. Örneğe ait modelin ürettiği kayıp değeri, ağırlıkların güncellenmesi aşamasında güncelleme miktarının etkisini belirleyecek olan bir "lambda" ile çarpılarak belirlenir. Bu değer bir önceki güncelleme miktarından daha büyük ise bu işlem tamamlanmaz ve bu örnek sonraya atılır ve sıradaki örneğe geçilir. Her bir devirde bu lambda değeri ve gelen örneğin kaybına göre modelin kayıp fonksiyonu şekillenir ve süreç bu şekilde devam eder. Böylece, model örneklere göre kendini şekillendirerek, öğrenme algoritmasının daha iyi bir yerel minimum noktası bulmasını sağlamaya çalışır.

Müfredat öğrenme farklı stratejiler ile kullanılarak modellerde iyileştirme yapması, daha küçük modellerin etkili şekilde kullanılması, problem doğrultusunda en uygun şekle sokma algoritmasının daha iyi bir yerel minimum noktası bulması gibi model başarısını pozitif yönde etkileyerek farklı yapay sinir ağı model topolojilerinde yüksek başarı vermesi söz konusu olmuştur.

2.4. Müfredat Öğrenme İle İlgili Yapılan Çalışmalar

Eğitim örneklerini belirli bir düzene göre sıralanarak yapay sinir ağını besleme fikri, Bengio ve ark. çalışmasından (Bengio ve ark. 2009) sonra hız kazanarak devam etmiştir. Bu çalışmadan sonra yapılan farklı yaklaşımlar bu başlık altında ele alınacaktır. Yapılan çalışmalarda kullanılan ağın topolojisine göre bir başlık sınıflandırması yapılmıştır. Kullanılan yöntemlerin çoğu sezgisel yaklaşımlar olmakla beraber teorik yaklaşımlar da (Weinshall ve ark. 2018 ; Weinshall ve Amir 2020) bu alanda yapılan çalışmalar açısından önemlidir. Bu tez çalışmasında sunulan yöntem sezgisel bir yöntem olduğu için teorik yöntemlerden söz edilmeyecektir.

2.4.1. Evrişimsel sinir ağlarında müfredat öğrenme kullanımı

FER-2013 (Wolfram 2018) veri kümesi üzerinde yapılan çalışma (Goodfellow ve ark. 2018) müfredat öğrenme fikrini Inception Resnet-V2 net modelinde uygulamayı denemiştir. Zorluk derecelerini belirlerken, her bir örneğin birbirlerine olan uzaklıklarından ve bir denetimsiz öğrenmede kullanılan kümele algoritmasının merkez noktası parametresine olan uzaklıklarından faydalanmışlardır. Beş farklı şekilde bu uzaklıkları kullanarak stratejiler belirlenmiş ve modeller birbirleriyle kıyaslanmıştır. En iyi sonuç basit kümelerden karmaşık kümelere göre sıralanması durumunda oluşmuştur.

WebVision, ImageNet, Clothing-1m ve Food-101 veri kümeleri üzerinde yapılan bu çalışmada (Guo et al. 2018) örnek zorluklarını belirlerken, özellik uzayında verinin yoğunluk dağılımını kullanarak belirlemektedir. Yazarlar bu dağılımının ortalama değerine yakın olan örnekleri “clean” olarak tanımlamaktadır. Bu yoğunluk dağılımına göre belirledikleri kategorileri sırasıyla “clean”, “noisy” ve “high noisy”’ olarak belirtmişlerdir.

Pascal Voc 2007 (Everingham ve ark. 2010) veri kümesinde yapılan bir obje tanıma çalışmasında (J. Wang ve ark. 2018) müfredat öğrenme kullanmışlardır. Bu çalışmada zorluklar veri setine ait bir örnekte bulunan obje sayısı ile belirlenmiştir. Yazarlar örnekte bulunan obje sayısına göre "zor" ve "kolay" olarak kategorize etmiş ve başarılı olmuşlardır.

Görsel özellikler için çoklu görev (Multi-Task Classification) sınıflandırma çalışmasında (Sarafianos ve ark. 2017) örnek zorlukları ilgileşim (korelasyon) ile belirlemeye çalışmışlardır. Bir örneğin diğer örneklerle arasında oluşan ilgileşim değerine göre "güçlü" ve "zayıf" olarak ayrılmış ve bu gruplar temel alınarak örnekler ağa sunulmuştur. SobİR (Schumann ve Monari 2014), VIPer (Gray ve Tao 2008) ve PETA (Y. Deng ve ark. 2014) veri setinde başarılı olduklarını belirtmişlerdir. Çoklu görev sınıflandırması için yapılan başka çalışmalar da (Pentina ve ark. 2015; X. Wang ve Xu 2019) mevcuttur.

Müfredat öğrenme farklı yaklaşımlarla (Yamashita ve Watasue 2014; Gong ve ark. 2016; Liu ve ark. 2019; X. Wang ve Xu 2019; Qin ve ark. 2020) örnek sıralamasında yapılan değişikler ile başarıyla kullanılmış ve ağın doğruluğunu arttırdığı görülmüştür.

2.4.2. Tekrarlayan sinir ağlarında müfredat öğrenme kullanımı

Tekrarlayan sinir ağı (RNN), düğümler arasındaki bağlantıların zamansal bir dizi boyunca birbirine yönlendirilmiş çizgelerden oluşan bir yapay sinir ağı türüdür. İleri beslemeli sinir ağlarından türetilen tekrarlayan sinir ağları, değişken uzunluktaki girdi dizilerini işlemek için iç durumlarını (states) kullanmaktadır. Bu çalışmada, tekrarlayan sinir ağı kullanılmadığı için temel kavramlar bölümünde değinilmemiştir. Fakat literatürde müfredat öğrenmeyi tekrarlayan sinir ağları üzerinde kullanan çalışmalar olduğu için bu çalışmalara değinilmiştir.

Metin verileri üzerinde yapılan bir çalışmada (Han ve Myaeng 2017) yazarlar örneklerin zorlukları için üç farklı yöntem önermişlerdir. Bu yöntemler sırasıyla ilk devirden sonra artan cümle uzunluklarıyla ağa örneklerin sunulması, "sorted", ortalama cümle uzunluklarına göre seçilen örneklerin ağa sunulması ve "one-pass" en kısa cümle

uzunluğuna sahip örneklerden başlayarak ağa sunulması "baby-steps" olarak sıralanmaktadır. Ayrıca her yöntemde aşırı öğrenmeyi engellemek için erken-durma (early-stopping) yöntemi kullanıldığı belirtilmiştir.

Literatürdeki başka bir çalışma (Platanios ve ark. 2019) ise Han ve Myaeng'in çalışmasına (Han ve Myaeng 2017) benzer bir şekilde cümle uzunluklarından ve cümlelerdeki az kullanılan kelimeleri bir zorluk ölçütü olarak kullanmışlardır. Bir cümle ne kadar uzun ve ne kadar az kullanılan kelimeler kümesine ait kelimelerden oluşuyor ise bu örnekler daha "zor" olarak belirlenmekte ve bu stratejiye göre ağa sunulmaktadır. Bu çalışma sinirsel makine çevirisi (Neural Machine Translation, NMT) üzerine bir çalışmadır. Kullanılan yöntem sezgisel bir yaklaşımdır.

Girdi olarak kullanılacak metin belgesinin uzunluğundan zorluğu ile ilgili çıkarımlar yaparak ağa sunan çalışmalar (Zaremba ve Sutskever 2014; Shi ve ark. 2015) müfredat öğreniminin tekrarlayan sinir ağlarında farklı stratejilerle kullanımını göstermektedir.

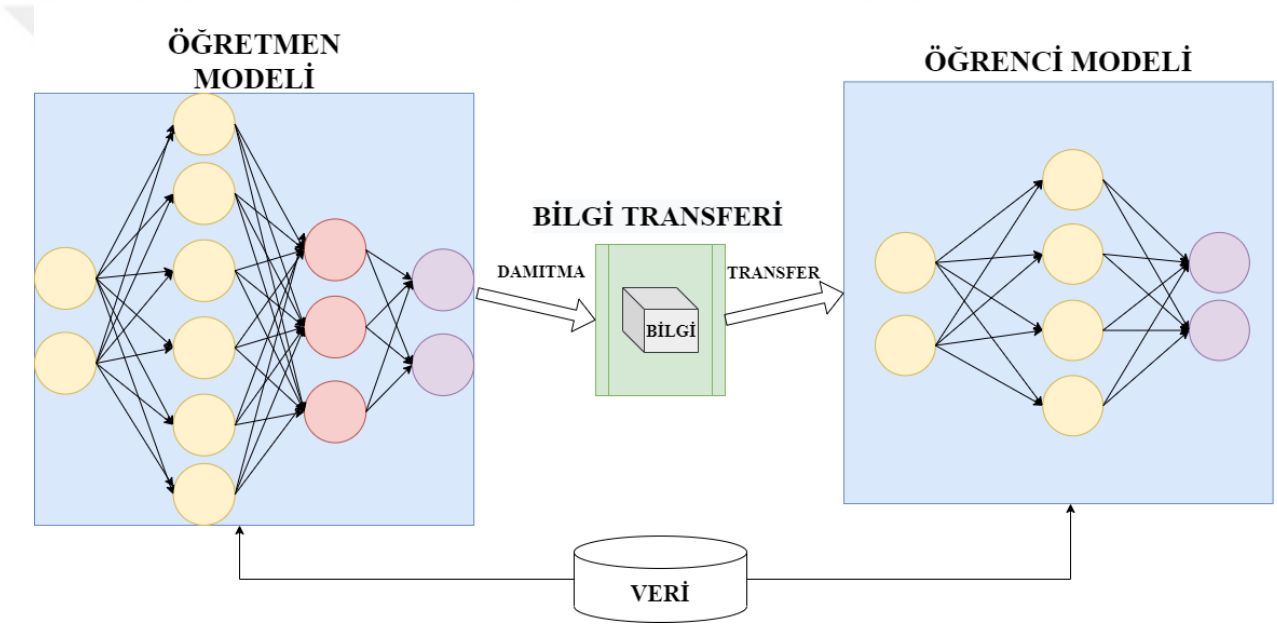
2.5. Bilgi Damıtma

Makine öğreniminde bilgi damıtma, bilginin büyük bir modelden daha küçük bir modele aktarılması sürecidir. Büyük modeller (çok derin sinir ağları veya birçok modelin toplulukları vb.) küçük modellerden daha yüksek bilgi kapasitesine sahipken, bu kapasite tam olarak kullanılamayabilir. Bilgi kapasitesinin çok azını kullansa bile bir modeli değerlendirmek hesaplama açısından maliyetli olabilmektedir. Bilgi damıtma, bilgiyi geçerlilik kaybı olmaksızın büyük bir modelden daha küçük bir modele aktarma sürecidir. Küçük modellerin değerlendirilmesi ve eğitilmesi daha az maliyetli olduğundan, daha düşük kaynaklara sahip donanımlara (mobil cihaz vb.) yerleştirilebilirler (Hinton ve ark. 2015).

Bilginin damıtılmasında, bilgi türleri, damıtma stratejileri ve öğretmen-öğrenci mimarileri öğrenci modelinin öğrenmesinde çok önemli bir rol oynamaktadır. Bilgi damıtma yöntemleri üç farklı şekilde sıralanabilir (Gou ve ark. 2020). Bu stratejiler yanıta-dayalı (response-based knowledge), özellik-dayalı (feature-based knowledge) ve ilişki-dayalı (relation-based knowledge) bilgi damıtma yöntemleridir. Bu tez

çalışmasında, yanıta-dayalı bilgi damıtma stratejisi kullanıldığı için sadece yanıta-dayalı bilgi damıtmadan söz edilecektir.

Derin öğrenme modelleri, sınırlı kaynaklara sahip cihazlarda, örneğin cep telefonları ve gömülü sistemler üzerinde kullanmak, yalnızca yüksek hesaplama karmaşıklığı nedeniyle değil, aynı zamanda büyük depolama gereksinimleri nedeniyle de büyük bir zorluk oluşturur. Bu amaçla, çeşitli model sıkıştırma ve hızlandırma teknikleri geliştirilmiştir. Temsili bir model sıkıştırma ve hızlandırma türü olarak bilgi damıtma stratejisinde, küçük bir öğrenci modeli, büyük bir öğretmen modelinden etkili bir şekilde çıkarım ve bilgi sentezleyebilmektedir.



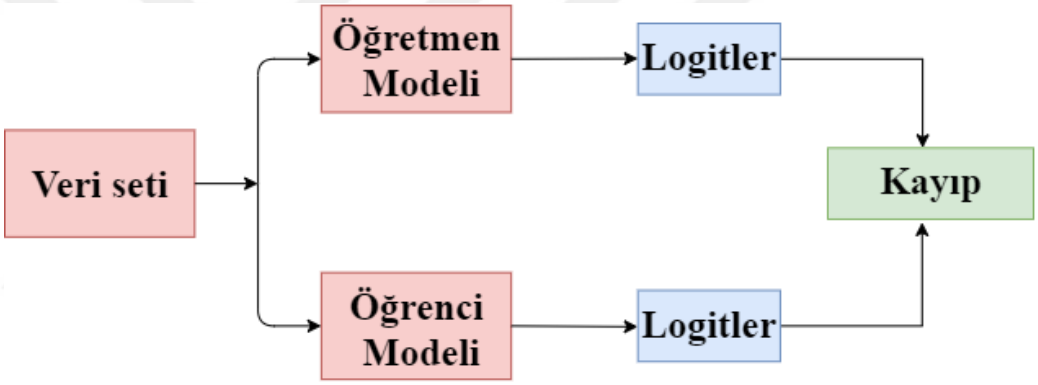
Şekil 2.8. Bilgi damıtma için genel bir öğretmen-öğrenci çerçevesi (Gou ve ark. 2020 'den değiştirilerek alınmıştır)

Şekil 2.8'de bilgi damıtma işleminin diyagramı gösterilmektedir. Burada Öğretmen modelinden bilgi çıkarımı yapılırken genellikle en son katman kullanılmaktadır. Bu katmanın boyutu kullanılan veri setinin sınıf sayısına göre değişmektedir. Öğretmen modeli kullanılan veri seti için bir eğitim işlemi (warm-up) uygulanır. Eğitim işlemi sona erdikten sonra her bir örneğe ait çıktı vektörleri (logits) özelleştirilmiş bir eşiksiz en büyük işlev fonksiyonu (softmax fonksiyonu) olan Denklem 2.11'de gösterilen sıcaklık

eşiksiz en büyük işlevi (temperatura softmax) fonksiyonu tarafından yumuşatılır veya sertleştirilir. Burada T yumuşatma için kullanılan sıcaklık parametresi y_i ve x_i her bir örneğe ait ilgili vektörün indis elemanı, N ise sınıf sayısıdır.

$$\sigma(x_i, T) = \frac{e^{\frac{y_i}{T}}}{\sum_j^N e^{\frac{y_j}{T}}} \tag{2.11}$$

Buradan elde edilen dağılım genellikle Kullback- Leibler sapma kaybı (Kullback-Leibler divergence loss) kullanılarak öğrenci modeli eğitiminde kullanılır. Şekil 2.9'de bu uygulama gösterilmektedir. Ayrıca Kullback-Leibler sapma kaybı $T = 1$ için çarpaz entropi kaybına (cross entropy loss) eşittir.



Şekil 2.9. Genel yanıta-dayalı bilgi damıtma

Yanıta-dayalı bilgi damıtma fikri, özellikle damıtılmış bilgi kullanımı bağlamında açık ve anlaşılması ve uygulaması kolay bir yöntemdir.

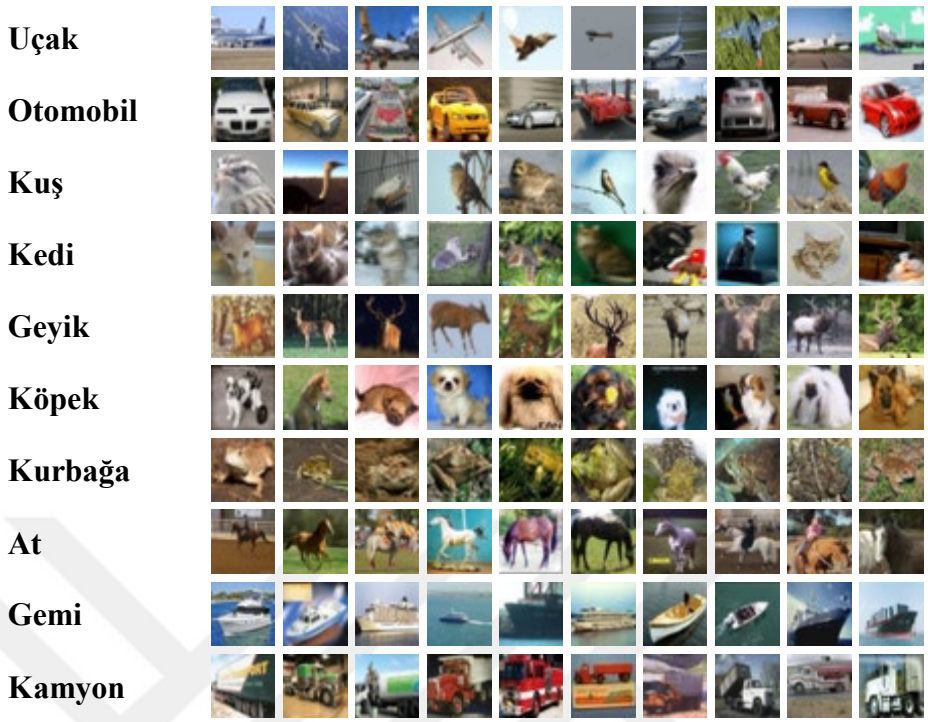
3. MATERYAL VE YÖNTEM

Bu bölümde, deneysel çalışmalarda kullanılan veri kümesi ve özelliklerine, müfredat oluşturulması için kullanılan yöntemlerin işleyişlerine yer verilmiştir.

3.1. Veri Kümesi

Bu tez çalışmasında CIFAR-10 & CIFAR-100 (Krizhevsky 2009) veri kümeleri kullanılmıştır. Bu veri setleri bir bilgisayara nesneleri nasıl tanıyacağını öğretmek için kullanılabilen bir dizi görüntülerden oluşmaktadır. CIFAR-10 ve CIFAR-100'deki görüntüler düşük çözünürlüklü olduğundan hesaplama maliyeti açısından elverişlidir. Bu nedenle, bu veri seti araştırmacılara, yaklaşımların işe yarayıp yaramadığını görmek için farklı algoritmaları hızlı bir şekilde denemesine izin vermektedir. Veri setine ait sınıf sayısı ve örnek sayılarının yeterli ve hesaplama maliyetinin uygun olduğu için bu tez çalışmasında kullanılmıştır.

CIFAR-10 veri kümesi, sınıf başına 6000 görüntü ile 10 sınıfta 60000 32x32 boyutunda renkli görüntüden oluşur. 50000 eğitim görüntüsü ve 10000 test görüntüsü vardır. Veri kümesi, her biri 10000 görüntü içeren beş eğitim grubuna ve bir test grubuna bölünmüştür. Test grubu, her sınıftan tam olarak rastgele seçilmiş 1000 görüntü içermektedir. Eğitim grupları kalan görüntüleri rastgele sırayla içerir, ancak bazı eğitim grupları bir sınıftan diğerine göre daha fazla görüntü içerebilir. Eğitim grupları her sınıftan tam olarak 5000 görüntü içermektedir. Şekil 3.1'de CIFAR-10 veri kümesinden alınan rasgele örnekler verilmiştir.



Şekil 3.1. CIFAR-10 veri kümesine ait örneklerin sınıflarla gösterimi

Sınıflar tamamen birbirinden bağımsızdır. Örneğin; Otomobil sınıfı ile Kamyon sınıfı arasında hiçbir örtüşme bulunmamaktadır. "Otomobil" sedan otomobilleri, arazi araçlarını ve buna benzer araçları içerir. "Kamyon" yalnızca büyük kamyonları içerirken küçük kamyon türlerini içermemektedir.

CIFAR-100 veri kümesi, her biri 600 görüntü içeren 100 sınıfı olması dışında CIFAR-10 ile aynı yapıdadır. Sınıf başına 500 eğitim görüntüsü ve 100 test görüntüsü vardır. CIFAR-100'deki 100 sınıf (Yunus, Yassı Balık, Gelincik, Kâse, Bilgisayar Klavyesi vb.), 20 üst sınıfa (super-class) (Suda Yaşayan Memeliler, Balıklar, Çiçekler, Yemek Kapları vs.) ayrılmıştır. Bu tez çalışmasında üst sınıflarla ilgilenilmemiştir.

3.2. Müfredat Oluşturma

Bu tez çalışmasında veri kümesine ait örneklerin doğrudan sıralanması ve oluşturulan düzene göre modelin eğitilmesi öngörülmüştür. Bu süreçte izlenen adımlar sırasıyla şu şekildedir;

- Daha önce eğitilmiş (pre-trained) bir ağın yalnızca son altı katmanını veri kümesine ait örnekler için eğitilmesi ve öğretmen modelinin oluşturulması,
- Eğitilen ağın sonuncu ve sondan bir önceki katmanından, eğitim kümesindeki her bir örnek için çıktı vektörlerinin oluşturulması,
- Her bir örneğe ait katman çıkış vektörleri ile örneğin kendisi ve etiket vektörleri ile yapılan işlemler sonuncunda reel bir sayı (örneğin zorluk değeri olarak adlandırılacaktır) üretilmesi,
- Oluşan bu sayılar,-skorlar- ile örneklerin küme içerisinde sıralamasının yapılması,
- Öğretmen modelinden çok daha küçük olan öğrenci modelini bu skorlarla oluşturulacak farklı sıralama yöntemlerine göre eğitilmesi ve karıştırılmış örneklerle beslenen durum kıyaslanması,

Bu çalışmada, yukarıda belirtilen adımlar sırasıyla incelenerek sunduğumuz yöntem ile ilgili detaylar verilecektir.

3.2.1. Öğretmen modelinin eğitilmesi

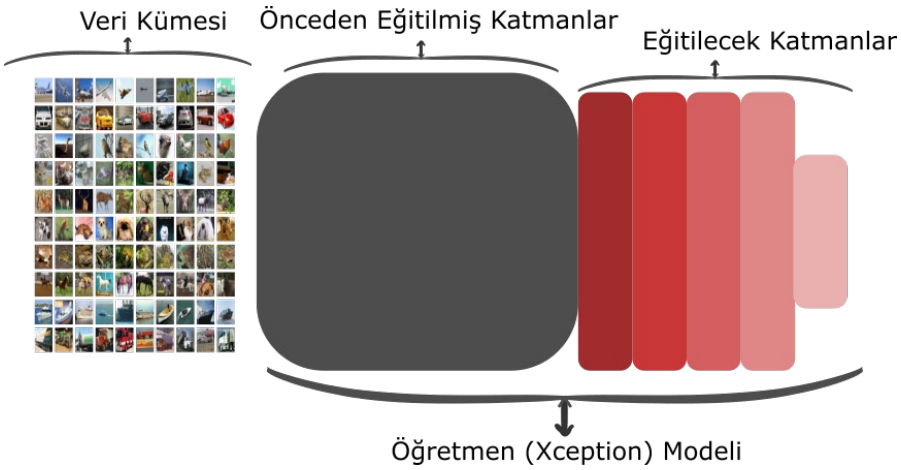
Gelişen teknoloji sayesinde bilgisayar bilimcileri birçok derin yapay sinir ağı önermişlerdir. Çizelge 3.1'de bu alanda sıklıkla kullanılan derin yapay sinir ağlarının, ImageNet (J. Deng ve ark. 2009) veri seti üzerinde başarılarını ve parametre sayıları gösterilmektedir. Öğretmen modeli için 22 910 480 parametreye sahip ve katman derinliği 126 olan Xception modeli (Chollet 2017) tercih edilmiştir. Diğer modeller de bu çalışma için kullanılabilirdi fakat bu model hesaplama maliyetleri, model boyutları ve hipotezimiz için en iyisi olduğu düşünüldüğünden dolayı tercih edilmiştir. Bu çalışma kapsamında Xception (Chollet 2017) modeli, "Öğretmen Modeli" olarak kullanılmıştır.

Çizelge 3.1. ImageNet üzerinde eğitilmiş ağlar
(Anonim2021d 'den değiştirilerek alınmıştır)

Model adı	**Boyutu(MB)**	**Doğruluk(%)**	**Parametre**	**Derinlik**
Xception	88	79	22 910 480	126
VGG16	528	71,3	138 357 544	23
VGG16	549	71,3	143 667 240	26
ResNet50	99	74,9	25 636 712	168
InceptionV3	92	77,9	23 851 784	159
Inception ResNetV2	215	80,3	55 873 736	572
MobileNet	16	70,4	4 253 864	88
MobileNetV2	14	71,3	3 538 984	88
DenseNet121	33	75	8 062 504	121
DenseNet169	57	76,2	14 307 880	169
NASNet Mobile	23	74,4	5 326 716	-

Şekil 3.2'de, öğretmen modelinin eğitilen ve eğitilmeyen katmanları gösterilmektedir. Buradaki ana fikir eğitim kümesinde bulunan verilere ait temel bilgilerin öğrenilmesidir. Bu tez çalışmasında öğretmen modelinin ilk katmanları hiçbir şekilde eğitilmeden olduğu gibi kullanılmıştır. Öğretmen modelinin diğer katmanları CIFAR-10 ve CIFAR-100 için ayrı ayrı eğilmiştir. Eğitim için kullanılan parametrelerden sadece devir sayısı farklıdır. Temelde bu fark veri setlerinin sahip olduğu farklı sayılardaki sınıf sayılarından kaynaklanmaktadır.

Bu tez çalışmasında öğretmen modelinin eğitilmesinde transfer öğreniminden (Transfer Learning, TL) (Pan ve Yang 2010 ; Zhuang ve ark. 2021) kullanılmıştır. TL bir görev üzerinde eğitilmiş bir modelin ikinci bir görev için yeniden tasarlandığı bir makine öğrenimi tekniğidir. Bu teknik, ikinci görevi modellerken hızlı ilerleme veya gelişmiş performans sağlayan bir en uygun şekle getirme yaklaşımıdır.



Şekil 3.2. Öğretmen modeli eğilen katmanlar

Çizelge 3.2'de Öğretmen Modeli eğitimi sonucunda oluşan model sonuçlarına yer verilmiştir. Burada görüleceğe üzere eğitilen ve eğitilmeyen parametreler sabit iken her bir kümeye ait doğruluk ve kayıp değerleri farklı sonuçlanmıştır. Test kümesi doğruluğu bu tez aşamasında önemsizdir. Çünkü öğretmen ağı test kümesine ait skorları üretmek için kullanılmayacaktır. CIFAR-100 veri kümesi diğerine göre daha fazla devir sayısı ile eğitilmiştir. Burada beklenen, modelin veri kümelerine ait gizli bilgileri öğrenmesidir. Ayrıca öğretmen modelinin son katmanı sınıf sayısına göre değişiklik gösterirken, sondan bir önceki katmanı da veri kümesine ait örneğin boyutlarına göre şekillendirilmektedir. Bunun nedeni, örneğin kendisi ile bu katman arasında bir sonraki bölümde hesaplanacak işlemler için gerekli olmasıdır. Deneysel çalışmalarda, bu modele ait eğitim parametreleri deneme yanılma yöntemi kullanılarak bulunmuştur. Bulunan en iyi devir sayıları CIFAR-100 için 100 devir iken CIFAR-10 veri kümesi için 80'dir. Doğruluk değerleri yaklaşık %5 kadar farklı olmasının nedeni veri kümelerine ait sınıfların farklı sayılarda olmasıdır.

Çizelge 3.2. Öğretmen Modeline ait eğitim parametreleri

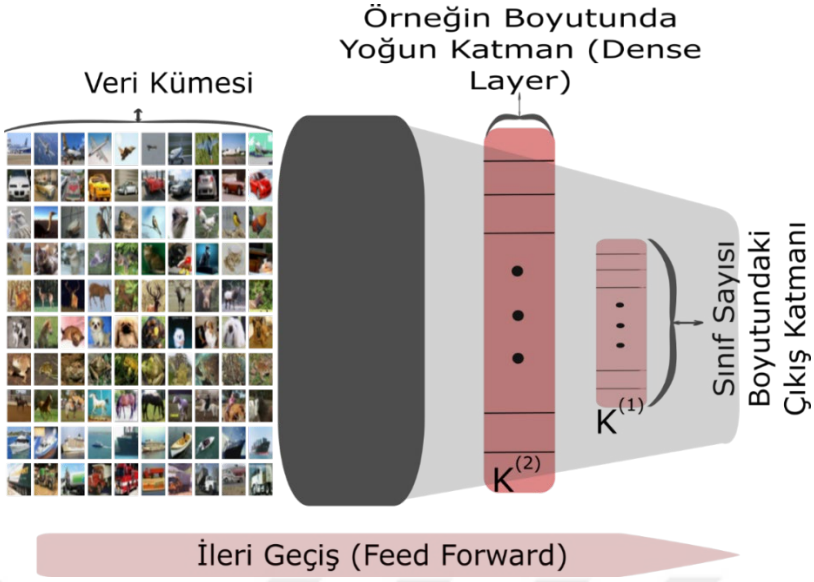
Veri Kümesi	Eğitilen Parametre Sayısı	Eğitilmeyen Parametre Sayısı	Toplam Parametre Sayısı	Eğitim Kümesi Doğruluk	Test Kümesi Doğruluk	Kayıp (Loss)	Toplam Devir Sayısı
CIFAR-100	11,147,876	16,110,632	27,258,508	0.9421	0.3744	1.2936	100
CIFAR-10	11,147,876	16,110,632	27,258,508	0.9833	0.6603	0.201	80

Bu aşama sonucunda öğretmen modeli skorları oluşturmak için hazır durumdadır.

3.2.2. Katmanlardan bilgi damıtma

Yapay sinir ağlarında katmanlardaki parametreler eğitim kümesine ait olan birçok özelliği öğrenmeye çalışırlar. Katmanların taşıdığı bilgiler farklı yöntemlerle gözler önüne serilebilmektedir. Bu konuda yapılmış bir çalışmada (Yosinski ve ark. 2015) derin sinir ağlarındaki katmanların anlamlandırdığı bilgiler görselleştirmiştir. Böylece karmaşık yapıdaki bir ağa ait parametrelerin belirli bilgileri öğrendiği gösterilmiştir. Bu durum transfer öğrenmenin de temelinde yatan sinir ağına ait ağırlıkların girdinin özelliklerine göre değiştiğidir. Bir modelin içerisinde bulunan farklı evrişimsel katmanlar belirli bir özelliği anlamak için uzmanlaşmaya çalışırlar. Bu özellikler çoğu zaman kafa, göz, köşeler vb. gibi anlaşılabilir ve görselleştirebilir (Yosinski ve ark. 2015) olsa da diğer tüm katmanların girişteki hangi özellik için bir cevap ürettiği halen tam olarak bilinmemektedir.

Bu tez çalışmasında, bir küme için eğitilmiş büyük bir modelden, bir müfredat oluşturmak için katmanlardaki bilgilerin kullanılması öngörülmüştür. Bu çalışmada, bir örnek için iki farklı tanımlama yapılmıştır. Bunlar, örneğin sınıf bilgisine ait etiket kaybı (label loss) ve örneğin kendisiyle sondan bir önceki katman ile arasında oluşması öngörülen örnek kaybı (sample loss)'dır.



Şekil 3.3. Öğretmen modeli bilgi damıtma katmanları

Şekilde 3.3'de eğitilen öğretmen modeline ait $K^{(1)}$ ve $K^{(2)}$ şeklinde adlandırılan katmanların boyutsal özelliği gösterilmektedir. Burada $K^{(1)}$ son katmanı $K^{(2)}$ ise sondan bir önceki katmanı ifade eder.

Her bir örnek ağa sunulur ve bir sonuç üretmesi beklenir. Bu işlem sırasında her katmanın değerleri eğitilen model parametreleriyle işlenip, aktivasyon fonksiyonun üreteceği değerler ile farklı sonuçlar üreterek son katmana kadar devam ettirilir. Bu tez çalışmasında bilgi damıtma yöntemi (Hinton ve ark. 2015) uygulanarak Denklem 3.1'deki çapraz entropi hesaplanmaktadır. Bu denklemdeki N_s sınıf sayısını, y_i örneğin etiket değer vektöründeki i. elemanını, $K_i^{(1)}$ ise Denklem 2.11'ile elde edilen örneğe ait katman vektörünün i. elemanını temsil etmektedir. Denklem 2.11'deki sıcaklık parametresi bir hiper-parametredir (hyperparameter) ve deneysel çalışma esnasında denenerek en uygun olanı bulunmuştur.

$$Etiket\ Kaybı = -\frac{1}{N_s}\sum_{i=1}^{N_s}\left[y_i \log K_i^{(1)} + (1-y_i)\log(1-K_i^{(1)})\right] \qquad (3.1)$$

Bu aşamaya benzer şekilde örneğe ait kayıplarda hesaplanabilir. $K^{(2)}$ katmanındaki perseptron sayısını örneğin boyutuna göre seçilmiştir. Örneğin, bu çalışmada kullanılan CIFAR-10 ve CIFAR-100 veri kümesine ait örneklerin boyutları 32×32 olduğu için $K^{(2)}$ katmanında bulunan birim sayısı 32*32 =1024 olacaktır. Bu şart sağlandıktan sonra örnek ile $K^{(2)}$ katmanı arasında oluşacak ilgileşim (korelasyon) hesaplanabilir. Benzer şekilde son katmana ait çıktılar örnek ile değiştiği gibi $K^{(2)}$ katmanın değerleri de girdi boyutu ile değişecektir. Bu aşamada ön işlem olarak $K^{(2)}$katmanın boyutu örneğin boyutuna dönüştürülerek hesaplamalara hazır hale getirilmiştir.

Örnek zorluklarını bulabilmek için örnek kaybı tanımlanmıştır ve Denklem 3.1'de gösterilen çapraz-benzerlik (cross-correlation) olarak hesaplanmıştır. Denklem 3.2'de bu aşamada yapılan işlemler gösterilmektedir. Burada M ve N örneğin ve katmanın örnek boyutuna çevrilmiş olan boyutlarını temsil etmektedir. Bu işlem sonunca üretilen değer 1 ile -1 değeri arasında olacaktır.

$$\text{Örnek Kaybı} = \frac{\sum_m^M \sum_n^N (X_{m,n} - \overline{X})(K_{m,n}^{(2)} - \overline{K^{(2)}})}{\sqrt{\left(\sum_n^N (X_{m,n} - \overline{X})^2\right)\left(\sum_n^N \left(K_{m,n}^{(2)} - \overline{K^{(2)}}\right)^2\right)}}. \tag{3.2}$$

$$\overline{X} = \frac{1}{M+N}\sum_m \sum_n X_{m,n}$$

$$\overline{K^{(2)}} = \frac{1}{M+N}\sum_m \sum_n K_{m,n}^{(2)}$$

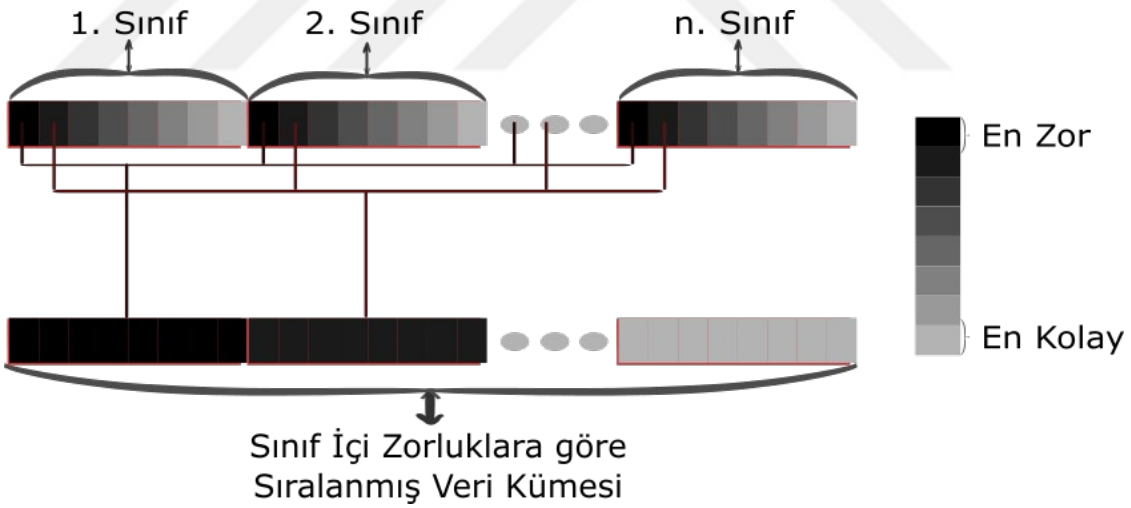
Buradaki temel fikir, modelin üreteceği, problem için istenen son çıktının bir önceki katmanla ilişkisinde örneğe ait bilgilerin en fazla temsil edildiği hipotezidir.

Tanıtılan örnek kaybı ve etiket kaybı değerleri her bir örnek için farklı sonuçlanacaktır. Her bir örnek için Denklem 3.3'de gösterilen şekilde değerlerin toplamı bu tez çalışmasında örneğe ait "zorluk" derecesi olarak tanımlanacak ve müfredat oluşturmak için bu zorluklardan faydalanılacaktır.

$$Zorluk = \text{Örnek Kaybı} + Etiket\ Kaybı \tag{3.3}$$

3.2.3. Zorlukların sıralanması

Veri kümesine ait örneklerin zorlukları elde edildiğinde farklı yöntemlerle sıralanmaları mümkündür. Bu tez çalışmasında sıralama için dört farklı sıralama yöntemi sunulmuştur. Bu sıralamalar örnek zorluğuna göre büyükten küçüğe, örnek zorluğuna göre küçükten büyüğe, sınıf bazlı büyükten küçüğe (SBBK) ve sınıf bazlı küçükten büyüğedir (SBKB). İlk iki sıralama isimlerinden anlaşılacağı üzere zorluk değerlerinin büyükten küçüğe veya küçükten büyüğe olarak standart sıralama şeklidir. Sınıf bazlı sıralamalarda ise öncelikle her sınıfa ait örnekler kendi içerisinde sıralanmıştır. Bu sıralamalar, modele ve çalıştırılma ortamına uygun olan yığın boyutuna (batch size) göre sınıf bazlı sıralanmış verilerden örnekler alınarak oluşturulmuştur. Örneğin; 600 örnek, 10 sınıflı ve eşit dağılmış sayıda örnekten oluşan bir veri kümesi, yığın boyutu 32 olan bir durumda her bir yığında her bir sınıfın ilk üç örneği alınarak ilk yığın parçasını oluşturacaktır. Diğer yığınlar için de aynı şekilde 2'nci için üçer örnek olarak devam edilecektir. Toplamda 18 (600/32) adet yığın oluşacak ve her bir yığında 32 örnek bulunacaktır. Şekil 3.4'de bu sıralama yöntemi gösterilmektedir.



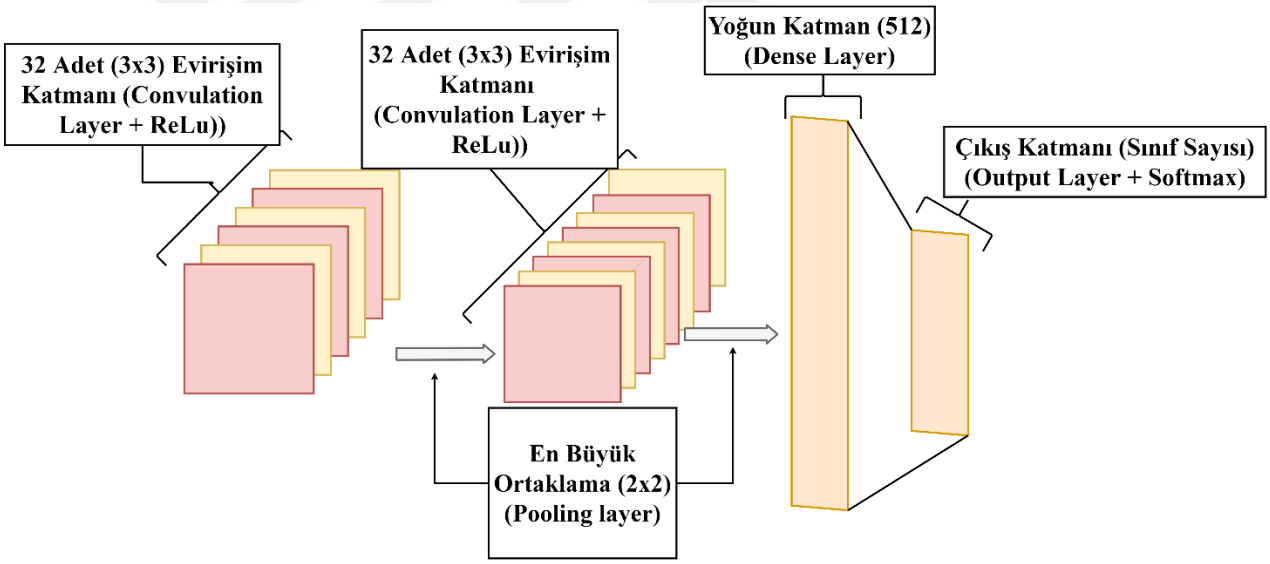
Şekil 3.4. Sınıf bazlı sıralama

Sınıf bazlı sıralamada Şekil 3.4'de gösterilenin tersine küçükten büyüğe olacak şekilde sıralanarak diğer sıralama yöntemi elde edilebilmektedir. Buradaki temel düşünce her bir yığında her sınıfa ait en zor veya en kolay örneklerden eşit sayıda bulunması sağlanması ve modele sunulmasının müfredat öğrenme ile ilgili yapılan sezgisel yöntemlerle daha

tutarlı oluşudur. Kullanılan sıralama yöntemleri ve karışık sıralanmış (shuffle) haliyle CIFAR-10 ve CIFAR-100 için bir öğrenci modeli eğitilmiştir ve oluşan sonuçlar tartışılmıştır.

3.2.4. Öğrenci modelinin eğitilmesi

Veri kümesine ait örnekler belirli bir yöntem ile sıralandığında, başka bir deyişle yapay sinir ağı için müfredat oluşturulduğunda bu sıralamalar kullanılarak öğrenci modeli eğitilebilir. Öğrenci modeli olarak öğretmen modeline göre çok daha az parametreye sahip özelleştirilmemiş (vanilla) bir evrişimsel sinir ağı kullanılmıştır. Öğrenci modeli iki adet 3×3 boyutunda 32 ve 64 adet filtreye sahip evrişim katmanı, iki adet 2×2 boyutunda en yüksek ortaklama katmanı ve 512 adet perseptrona sahip yoğun katmandan oluşur ve 1,250,852 adet eğitilebilir parametreden meydana gelmektedir. Şekil 3.5'de öğrenci modeli gösterilmektedir.



Şekil 3.5. Öğrenci modeli

Öğrenci modelinin küçük olmasının temel nedeni kullanılan veri kümesine uygunluğu ve savunulan fikir için örnek zorluklarının etkisini incelemenin daha kolay olması gerçeğidir. Çünkü modelin eğitilebilir parametreleri ve karmaşıklığı arttığında model için örneklere ait farklılıkların, "zorlukların" etkisi ortadan kalkabilmektedir. Çizelge 3.1'de bahsedilen büyük ve karmaşık modeller için sunulan örneklere ait farklılıkların etkisi

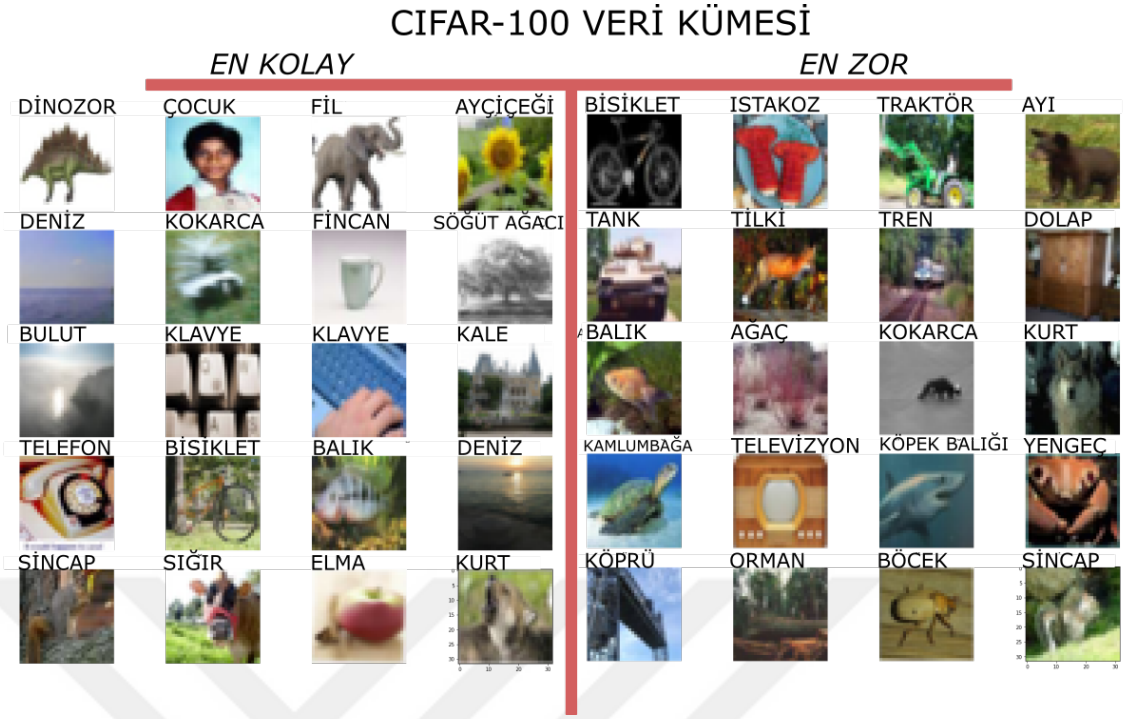
model için azalacak ve modelin en uygun minimum noktasını bulması kolaylaşacaktır. Başka bir deyişle, bir öğrenci normalden daha zeki ise öğreneceği bilgilerin zorlukları, ortalama bir öğrenciye göre öğrenme düzeyine etkisi daha da az olması beklenmektedir. Bu nedenle bu tez çalışmasında öğretmen modelinin eğitilebilir parametre sayısına göre yaklaşık 11 kat daha küçük ve basit bir öğrenci modeli kullanılmıştır.

4. BULGULAR

Bu bölümde, dört farklı sıralama yöntemin iki farklı veri kümesi üzerinde gerçekleştirilmesine ait bilgiler ve elde edilen sonuçlar sunulmuştur. CIFAR-10 ve CIFAR-100 eğitim kümeleri eğitim ve test verisi olarak iki kısma ayrılmıştır. Toplam 60000 görüntüden oluşan bu veri setleri 5000 eğitim kümesinde 1000 test kümesinde olacak şekilde düzenlenmiştir. Öğretmen modelinden elde edilen zorluk bilgisi her bir örnek için kaydedilmiştir. Müfredat oluşturabilmek için kaydedilen bu zorluk bilgilerine göre eğitim kümesindeki veriler farklı yöntemlerle sıralanmıştır. Öğretmen modelinden damıtılan bilgiyle küçükten büyüğe göre sıralanmış CIFAR-10 ve CIFAR-100 veri kümesine ait örnekler Şekil 4.1 ve Şekil 4.2'de verilmiştir. Şekilde gösterilen örnekler zorluklara göre her iki veri seti için en zor ilk on örnek ve en kolay ilk on örnek olarak gösterilmiştir.



Şekil 4.1. Sıralanmış CIFAR-10 veri kümesine ait örnekler



Şekil 4.2. Sıralanmış CIFAR-100 veri kümesine ait örnekler

Öğretmen ve Öğrenci Modellerinin eğitiminde programlama dili olarak Python (van Rossum 1995) kullanılmıştır. Kullanılan her iki modelin eğitiminde TensorFlow (Abadi ve ark. 2016) kütüphanesi, matematiksel işlemler ve operasyonlar için NumPy (Harris ve ark. 2020) ve scikit-learn (Pedregosa ve ark. 2011) kütüphaneleri kullanılmıştır.

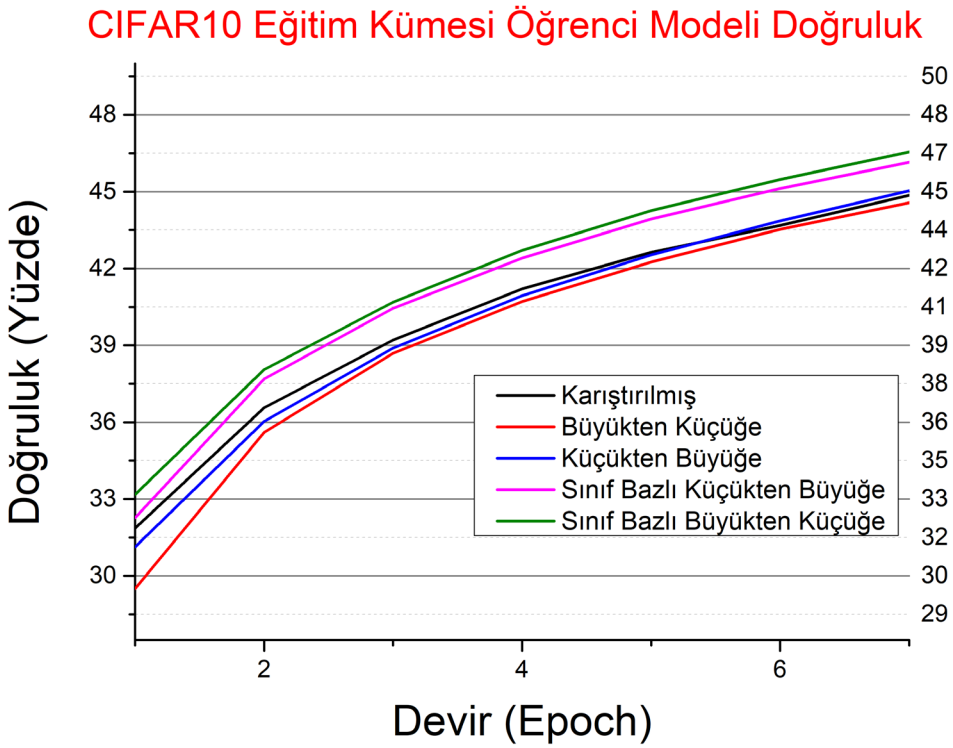
Öğrenci modeli küçük olduğu için yedi devir eğitilmiş ve veri kümesinin oluşturulan zorluklarla sıralanmasının karıştırılmış duruma göre değişimi incelenmiştir. Yedi devir eğitilmesinin nedeni, modelin küçük olması ve devir sayısı arttığında doyuma ulaşmasıdır. Modelin başarısını daha iyi analiz edebilmek için her bir sıralama şekli ile eğitilen model on kez tekrar edilmiş ve devir başına oluşan doğrulukların ortalaması alınmıştır. Test verisi her bir durum ve model için üzerinde herhangi bir değişiklik yapılmadan kullanılmıştır. Bu çalışmada öğretmen modelinden etiket kaybı hesaplanırken Denklem 2.11 kullanılmıştır ve buradaki sıcaklık parametresi 0,6 olarak ayarlanmıştır. Bu parametrenin seçimi için deneme yanılma yöntemi kullanılmıştır ve her iki veri kümesi içinde aynı değer kullanılmıştır.

Çizelge 4.1'de sunulan sıralama yöntemleriyle sıralanmış veriler ile eğitilmiş öğrenci modelinin her bir devirdeki ortalama doğruluğu gösterilmiştir. Burada görüleceğe üzere

karıştırılmış veri kümesiyle eğitilmiş model aynı test verisi üzerinde bu tez çalışmasında sunulan zorluk tespiti ve sıralaması yöntemiyle her bir devirde farklı sonuçlanmıştır. Sunduğumuz zorlukları yalnızca büyükten küçüğe veya tersi olacak şekilde sıraladığımızda model karıştırılmış durum ile neredeyse aynı sonuçları üretirken; SBKB ve SBBK sıralamaları ile oluşan doğruluklar her bir devirde daha iyi sonuç üretmiştir.

Çizelge 4.1. CIFAR-10 veri kümesi üzerinde öğrenci modeline ait doğruluk değerleri

	SIRALAMA YÖNTEMİ				
Devir (Epoch)	Karıştırılmış	Büyükten Küçüğe	Küçükten Büyüğe	Sınıf Bazlı Büyükten Küçüğe	Sınıf Bazlı Küçükten Büyüğe
1.	31,865%	29,499%	31,133%	**32,265%**	**33,177%**
2.	36,567%	35,600%	36,026%	**37,682%**	**38,053%**
3.	39,211%	38,693%	38,885%	**40,442%**	**40,683%**
4.	41,196%	40,713%	40,942%	**42,397%**	**42,706%**
5.	42,628%	42,256%	42,532%	**43,935%**	**44,258%**
6.	43,682%	43,532%	43,853%	**45,117%**	**45,469%**
7.	44,855%	44,569%	45,028%	**46,148%**	**46,546%**



Şekil 4.3. CIFAR-10 veri kümesi üzerinde öğrenci modeline ait doğruluk grafiği

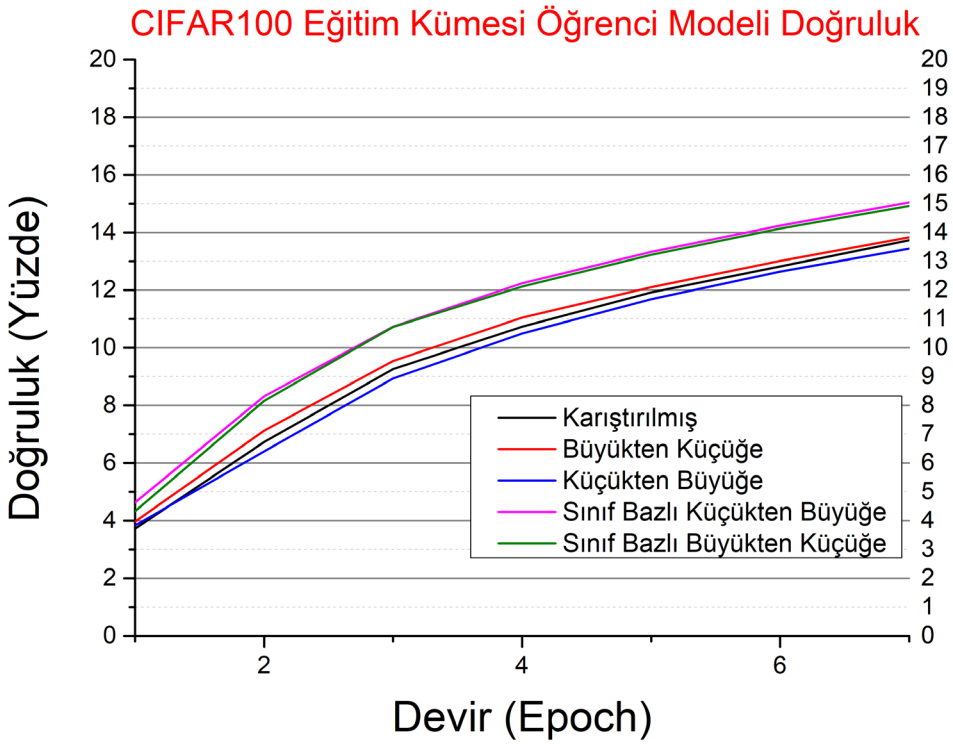
Şekil 4.3'de görüldüğü üzere sunulan iki yöntem karıştırılmış duruma göre her bir devirde daha iyi sonuç üreterek modelin başarısını arttırmıştır. Burada başlangıçta oluşan doğruluk yüzdesi ile oluşan farklar her bir devirde korunarak devam etmektedir. Büyükten küçüğe ve tersi durumundaki sıralanışların karıştırılmış durumdan daha kötü olmasının nedeni ise sıralamada oluşan aynı sınıfa ait verilerin sıklıkla bir araya gelmesidir. Bu durumda model her bir yığının içine denk gelmiş tek bir sınıfa ait örnekleri diğer sınıflardan daha fazla öğrendiğinden, daha kötü bir yerel minimum noktasına sıkışmaya başlayacaktır. Bu durum sistemin genel başarısını olumsuz yönde etkilemeye devam edecektir. Bu durumdan kurtulmak için sunulan diğer iki sıralama yöntemiyle bu sorun ortadan kalkmış ve modelin başarısı artmıştır.

Çizelge 4.2'de sunulan yeni stratejilerin CIFAR-100 veri kümesi üzerinde doğruluk değerleri gösterilmiştir. Burada oluşan metrikler, aynı şekilde SBBK ve SBKB durumu için CIFAR-10 sonuçlarına benzer şekilde sonuçlanmıştır. Kullanılan modelin doğruluk değerleri CIFAR-10 için kullanılan model ile aynı olduğu için çok alt seviyedir. Çünkü

kullanılan öğrenci modeli CIFAR-100 için çok küçük ve basit kalmıştır. Fakat bu sonuçlar, on kez çalıştırılıp ortalama doğruluk değeri kullanıldığı için bu tez çalışması kapsamında anlamlıdır.

Çizelge 4.2. CIFAR-100 veri kümesi üzerinde öğrenci modeline ait doğruluk değerleri

	SIRALAMA YÖNTEMİ				
Devir (Epoch)	Karıştırılmış	Büyükten Küçüğe	Küçükten Büyüğe	Sınıf Bazlı Büyükten Küçüğe	Sınıf Bazlı Küçükten Büyüğe
1.	3,718%	3,964%	3,503%	**4,627%**	**4,330%**
2.	6,735%	7,120%	6,246%	**8,314%**	**8,156%**
3.	9,261%	9,540%	9,073%	**10,735%**	**10,719%**
4.	10,726%	11,041%	10,603%	**12,234%**	**12,119%**
5.	11,921%	12,106%	11,763%	**13,324%**	**13,228%**
6.	12,822%	13,010%	12,760%	**14,235%**	**14,138%**
7.	13,726%	13,833%	13,606%	**15,048%**	**14,921%**



Şekil 4.4. CIFAR-100 veri kümesi üzerinde öğrenci modeline ait doğruluk grafiği

Şekil 4.4'de CIFAR-100 veri kümesi üzerinde sunduğumuz yöntemlerin başarıları grafiksel olarak verilmiştir. CIFAR-10 için oluşan benzer durum burada da oluşmuştur. Başlangıçta oluşan fark diğer devirler için korunarak devam etmiştir. Burada oluşan farklılık ise büyükten küçüğe ve tersi sıralamaların karıştırılmış sıralamaya ait değerlerin altında ve üstünde olmasıdır. Modelin bu veri kümesi için çok küçük olması bu durumu destekler niteliktedir. Fakat bu tez çalışmasında savunulan hipotez eğitim örneklerinin farklı sıralamalarının model başarısını nasıl etkileyeceği üzerinedir. Bir diğer farklılık ise kazanan sıralama yöntemlerinin birbirlerine çok yakın sonuçlar üretmesidir.

Çizelge 4.3'te CIFAR-10 veri kümesi için oluşan sonuçların karıştırılmış duruma göre her bir devirde oluşturduğu yüzdelik başarı farklarını göstermektedir. SBBK ve SBKB yöntemleri her bir devir sonunda oluşan model doğruluğu, karıştırılmış duruma göre daha iyi sonuçlanmıştır.

Çizelge 4.3. CIFAR-10 için sıralama yöntemlerinin karıştırılmış duruma göre farkları

	Karıştırılmış Duruma Göre Farkları			
Devir (Epoch)	Büyükten Küçüğe	Küçükten Büyüğe	Sınıf Bazlı Büyükten Küçüğe	Sınıf Bazlı Küçükten Büyüğe
1.	-2,366%	-0,732%	**0,400%**	**1,312%**
2.	-0,967%	-0,541%	**1,115%**	**1,486%**
3.	-0,518%	-0,326%	**1,231%**	**1,472%**
4.	-0,483%	-0,254%	**1,201%**	**1,510%**
5.	-0,372%	-0,096%	**1,307%**	**1,630%**
6.	-0,150%	0,171%	**1,435%**	**1,787%**
7.	-0,286%	0,173%	**1,293%**	**1,691%**

Çizelge 4.4'de CIFAR-100 veri kümesi üzerinde elde edilen sonuçların karıştırılmış ile arasında oluşan devire dayalı farklar gösterilmektedir. Her bir devir için sunulan sınıf bazlı sıralama yöntemleri, karıştırılmış duruma göre genel olarak yüzde 1 ile 2 arasında daha iyi sonuçlanmıştır.

Çizelge 4.4. CIFAR-100 için sıralama yöntemlerinin karıştırılmış duruma göre farkları

	Karıştırılmış Duruma Göre Farkları			
Devir (Epoch)	Büyükten Küçüğe	Küçükten Büyüğe	Sınıf Bazlı Büyükten Küçüğe	Sınıf Bazlı Küçükten Büyüğe
1.	0,246%	-0,215%	**0,909%**	**0,612%**
2.	0,385%	-0,489%	**1,579%**	**1,421%**
3.	0,279%	-0,188%	**1,474%**	**1,458%**
4.	0,315%	-0,123%	**1,508%**	**1,393%**
5.	0,185%	-0,158%	**1,403%**	**1,307%**
6.	0,188%	-0,062%	**1,413%**	**1,316%**
7.	0,107%	-0,120%	**1,322%**	**1,195%**

Bu bölümde öğretmen modelinden edinilen bilgiler neticesinde sadece sıralaması değiştirilmiş veri kümelerinin başarıları tartışılmıştır. Sınıf bazlı sıralamalar her iki veri kümesi için de karışık sıralanmış duruma göre daha iyi sonuç üretmiştir.

5. TARTIŞMA ve SONUÇ

Bu çalışmada, veri kümesindeki örneklerin yerlerini belirli bir düzenle değiştirip model başarısının nasıl değiştiği gözlemlenmiştir. Ortaya koyulan yöntemler, CIFAR-10 ve CIFAR-100 veri kümelerindeki başarıları araştırılmıştır. Büyük bir öğretmen modelinde veri kümesine ait bilgilerin öğrenilmesiyle oluşturulan zorluklar sıralanarak model başarısına nasıl etki ettiği gösterilmiştir. Bu çalışmadaki temel savunulan nokta örneklerin yerleri değiştirildiğinde modelin öğrenme becerisini artırıcı yönde olacağı fikridir. Öğretmen modeline göre çok küçük olan öğrenci modeli, eğitim örneklerinin karıştırılmış sıra ile ağa sunulması durumu ile sunduğumuz yöntemlerle sunulması arasında her bir devirde yüzde 1 ile 2 arasında fark oluşmuştur. Sunulan iki sıralama yöntemi karıştırılmış duruma göre daha iyi sonuç üretmiştir.

Örnek zorluklarının doğrudan büyükten-küçüğe veya tersi sıralaması durumlarının karıştırılmış duruma göre kötü çıkmasının nedeni, oluşan sıralamada aynı sınıfa ait verilerin birden fazla tekrara denk geliyor olması olarak ifade edilebilir. Modelin oluşturmaya çalıştığı matematiksel model için genel bir yerel minimum noktası bulması aynı sınıfların sıklıklarla art arda gelmesi ile zorlaşmaktadır. Bu zorluk, sunduğumuz SBKB ver SBBK sıralama yöntemleri ile giderilmiştir. Bu yöntemler ile aynı sınıfa ait örneklerin sıklıkla art arda gelmesi engellenerek, modelin daha genel ve karıştırılmış olarak sunulan duruma göre daha iyi bir yerel minimum noktası bulması sağlanmıştır. Modelin bu yerel minimum noktasına yaklaşması, her bir devirde sistemin başarısını pozitif yönde etkileyerek daha iyi sonuç üretmesine olanak sağlamıştır.

Üretilen zorluk ölçütü, önceden eğitilmiş bir model tarafından veriye ait bilgilerin sentezlenmesi sonucunda otomatik olarak oluşturulmuştur. Burada öğretmen modelinin bir örnek için oluşturduğu kayıp ve katman ile örneğin arasında oluşan ilgileşimlerin fazla, az veya negatif olması, modelin örnek için belirlediği bir öğrenme zorluğu parametresidir. Gelecekteki çalışmalarda bu tez kapsamında sunulan ilgileşimin, zorlukları belirlemede nasıl ve neden etki ettiği araştırılabilir. Öğretmen modelinin örnek için oluşturduğu kayıp modelin yerel minimum noktasına olan uzaklığı ile

açıklanabilirken kullandığımız ilgileşim benzer şekilde açıklanması güçtür. Her bir örnek için oluşturduğumuz bu örnek kaybının etkisi araştırılması gereken bir noktadır.

Öte yandan, örneklerin veri kümesi içerisinde yerlerini değiştirerek modelin performansını arttırmak başlı başına ilginç bir noktadır. Örneğin, herhangi bir n adet örnek içeren veri kümesi $n!$ farklı şekilde sıralanabilir. Kullandığımız veri kümesinin 50000 adet örnek içermektedir. Bu durumda, oluşacak farklı sıralamaların sayısı $50000! \cong 3 \times 10^{213236}$'dır. Sunduğumuz yöntem bu sıralama kombinasyonları içerisinden sistemin doğruluğunu pozitif yönde etkileyen iki farklı sıralamanın tespit edebiliyor olması açısından çok değerlidir ve bu konu ile daha detaylı çalışma isteğini araştırmacılarda uyandırabilir.

Her iki veri kümesi için de başarı gösteren SBKB ve SBBK yöntemleri ile sistem başarısını arttırmak mümkün gözükmektedir. Küçük cihazlarda (mikroişlemciler, mobil telefon vb.) sunduğumuz bu yöntemler kullanılabilir ve daha az işlem gücüne sahip cihazlar ve küçük modeller için iyileştirmeler gözlenebilir.

KAYNAKLAR

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X. 2016. TensorFlow: A system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, 265–283.

Anderson, J. A., Rosenfeld, E. 1993. Neurocomputing, MIT press.

Anonim2021a https://www.pngwing.com/en/free-png-nuogo Erişim Tarihi: 14.04.2021. .

Anonim2021b https://www.renom.jp/notebooks/tutorial/beginners_guide/backpropagation_example/notebook.html Erişim Tarihi:10.03.2021. .

Anonim2021c https://www.jeremyjordan.me/convnet-architectures/ Erişim Tarihi: 18.03.2021. .

Anonim2021d https://keras.io/api/applications/ Erişim Tarihi: 18.03.2021. .

Bengio, Y., Louradour, J., Collobert, R., Weston, J. 2009. Curriculum learning. Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09 : Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09, New York, New York, USA, : ACM Press: , 1–8. https://doi.org/10.1145/1553374.1553380

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua:, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

Curry, H. B. 1944. The method of steepest descent for non-linear minimization problems. Quarterly of Applied Mathematics, 2(3):, 258–261. https://doi.org/10.1090/qam/10667

De Mulder, W., Bethard, S., Moens, M.-F. 2015. A survey on the application of recurrent neural networks to statistical language modeling. Computer Speech & Language, 30(1):, 61–98. https://doi.org/10.1016/j.csl.2014.09.005

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, Li Fei-Fei 2009. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition : 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE: , 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Deng, Y., Luo, P., Loy, C. C., Tang, X. 2014. Pedestrian attribute recognition at far distance. MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia, 789–792. https://doi.org/10.1145/2647868.2654966

Elman, J. L. 1993. Learning and development in neural networks: the importance of starting small. Cognition, 48(1):, 71–99. https://doi.org/10.1016/0010-0277(93)90058-4

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2):, 303–338. https://doi.org/10.1007/s11263-009-0275-4

Fukushima, K. 1975. Cognitron: A self-organizing multilayered neural network.

Biological Cybernetics, 20(3–4):, 121–136. https://doi.org/10.1007/BF00342633
Gong, C., Tao, D., Maybank, S. J., Liu, W., Kang, G., Yang, J. 2016. Multi-Modal Curriculum Learning for Semi-Supervised Image Classification. IEEE Transactions on Image Processing, 25(7):, 3249–3260. https://doi.org/10.1109/TIP.2016.2563981
Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Groze, C., Bengio, Y. 2018. Wolfram Research, "FER-2013" from the Wolfram Data Repository (2018). Wolfram Research. Retrieved from https://datarepository.wolframcloud.com/resources/FER-2013
Gou, J., Yu, B., Maybank, S. J., Tao, D. 2020. Knowledge Distillation: A Survey. arXiv, 1–36. https://doi.org/10.1007/s11263-021-01453-z
Gray, D., Tao, H. 2008. Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. , , 262–275. https://doi.org/10.1007/978-3-540-88682-2_21
Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M. R., Huang, D. 2018. CurriculumNet: Weakly supervised learning from large-scale web images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11214 LNCS:, 139–154. https://doi.org/10.1007/978-3-030-01249-6_9
Han, S., Myaeng, S. H. 2017. Tree-structured curriculum learning based on semantic similarity of text. Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, 2017-Decem:, 971–976. https://doi.org/10.1109/ICMLA.2017.00-27
Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T. E. 2020. Array programming with NumPy. Nature, 585(7825):, 357–362. https://doi.org/10.1038/s41586-020-2649-2
Herculano-Houzel, S. 2009. The human brain in numbers: a linearly scaled-up primate brain. Frontiers in Human Neuroscience, 3:. https://doi.org/10.3389/neuro.09.031.2009
Hinton, G., Vinyals, O., Dean, J. 2015. Distilling the Knowledge in a Neural Network. , 1–9. Retrieved from http://arxiv.org/abs/1503.02531
Hubel, D. H., Wiesel, T. N. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology, 160(1):, 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837
Kingma, D. P., Ba, J. 2014. Adam: A Method for Stochastic Optimization. . Retrieved from http://arxiv.org/abs/1412.6980
Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. .
Krueger, K. A., Dayan, P. 2009. Flexible shaping: How learning in small steps helps. Cognition, 110(3):, 380–394. https://doi.org/10.1016/j.cognition.2008.11.014

Kumar, M., Packer, B., Koller, D. 2010. Self-Paced Learning for Latent Variable Models. Advances in Neural Information Processing Systems : Advances in Neural Information Processing Systems, Ed.: J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (Eds.) (Vol. 23), Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2010/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf

Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. 1990. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO : Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO, Ed.: D. Touretzky (Ed.) (Vol. 2), Morgan Kaufmann.

Liu, X., Zhou, F., Shen, D., Wang, S. 2019. Deep Convolutional Neural Networks with Curriculum Learning for Facial Expression Recognition. Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019, 5925–5932. https://doi.org/10.1109/CCDC.2019.8832982

McCulloch, W. S., Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4):, 115–133. https://doi.org/10.1007/BF02478259

Minsky, M., Papert, S. 1969. Perceptrons., Oxford, England, : M.I.T. Press.

Pan, S. J., Yang, Q. 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):, 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É. 2011. Scikit-Learn: Machine Learning in Python. J. Mach. Learn. Res., 12(null):, 2825–2830.

Pentina, A., Sharmanska, V., Lampert, C. H. 2015. Curriculum learning of multiple tasks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June:, 5492–5500. https://doi.org/10.1109/CVPR.2015.7299188

Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., Mitchell, T. M. 2019. Competence-based curriculum learning for neural machine translation. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1:, 1162–1172. https://doi.org/10.18653/v1/n19-1119

Plaut, D. C., Hinton, G. E. 1987. Learning sets of filters using back-propagation. Computer Speech and Language, 2(1):, 35–61. https://doi.org/10.1016/0885-2308(87)90026-X

Qin, W., Hu, Z., Liu, X., Fu, W., He, J., Hong, R. 2020. The Balanced Loss Curriculum Learning. IEEE Access, 8:, 25990–26001. https://doi.org/10.1109/ACCESS.2020.2970726

Rohde, D. L. T., Plaut, D. C. 1999. Language acquisition in the absence of explicit negative evidence: How important is starting small?. Cognition, 72(1):, 67–109. https://doi.org/10.1016/S0010-0277(99)00031-1

Ruder, S. 2016. An overview of gradient descent optimization algorithms. , 1–14. Retrieved from http://arxiv.org/abs/1609.04747

Rumelhart, D. E., Hinton, G. E., Williams, R. J. 2013. Learning Internal Representations by Error Propagation. Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence, (V):, 399–421. https://doi.org/10.1016/B978-1-4832-1446-7.50035-2

Sarafianos, N., Giannakopoulos, T., Nikou, C., Kakadiaris, I. A. 2017. Curriculum learning for multi-task classification of visual attributes. Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017, 2018-Janua:, 2608–2615. https://doi.org/10.1109/ICCVW.2017.306

Schumann, A., Monari, E. 2014. A soft-biometrics dataset for person tracking and re-identification. 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) : 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE: , 193–198. https://doi.org/10.1109/AVSS.2014.6918667

Shi, Y., Larson, M., Jonker, C. M. 2015. Recurrent neural network language model adaptation with curriculum learning. Computer Speech & Language, 33(1):, 136–154. https://doi.org/10.1016/j.csl.2014.11.004

van Rossum, G. 1995. Python tutorial, May 1995. CWI Report CS-R9526, (CS-R9526):, 1–65. Retrieved from http://oai.cwi.nl/oai/asset/5007/05007D.pdf

Wang, J., Wang, X., Liu, W. 2018. Weakly- and Semi-supervised Faster R-CNN with Curriculum Learning. Proceedings - International Conference on Pattern Recognition, 2018-Augus:, 2416–2421. https://doi.org/10.1109/ICPR.2018.8546088

Wang, X., Xu, Z. 2019. Automatic sleep staging based on curriculum learning approach. ACM International Conference Proceeding Series, 1–6. https://doi.org/10.1145/3354031.3354033

Wang, Y., Gan, W., Yang, J., Wu, W., Yan, J. 2019. Dynamic curriculum learning for imbalanced data classification. Proceedings of the IEEE International Conference on Computer Vision, 2019-Octob(2):, 5016–5025. https://doi.org/10.1109/ICCV.2019.00512

Weinshall, D., Amir, D. 2020. Theory of curriculum learning, with convex loss functions. Journal of Machine Learning Research, 21:, 1–18.

Weinshall, D., Cohen, G., Amir, D. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. 35th International Conference on Machine Learning, ICML 2018, 12:, 8331–8339.

Yamashita, T., Watasue, T. 2014. Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning. 2014 IEEE International Conference on Image Processing, ICIP 2014, 853–857. https://doi.org/10.1109/ICIP.2014.7025171

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H. 2015. Understanding Neural Networks Through Deep Visualization. . Retrieved from http://arxiv.org/abs/1506.06579

Zaremba, W., Sutskever, I. 2014. Learning to Execute. , 1–25. Retrieved from http://arxiv.org/abs/1410.4615

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q. 2021. A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE, 109(1):, 43–76. https://doi.org/10.1109/JPROC.2020.3004555