

Derin Ağlarda Katmanların Kapatılmasının Sonuca Etkisi



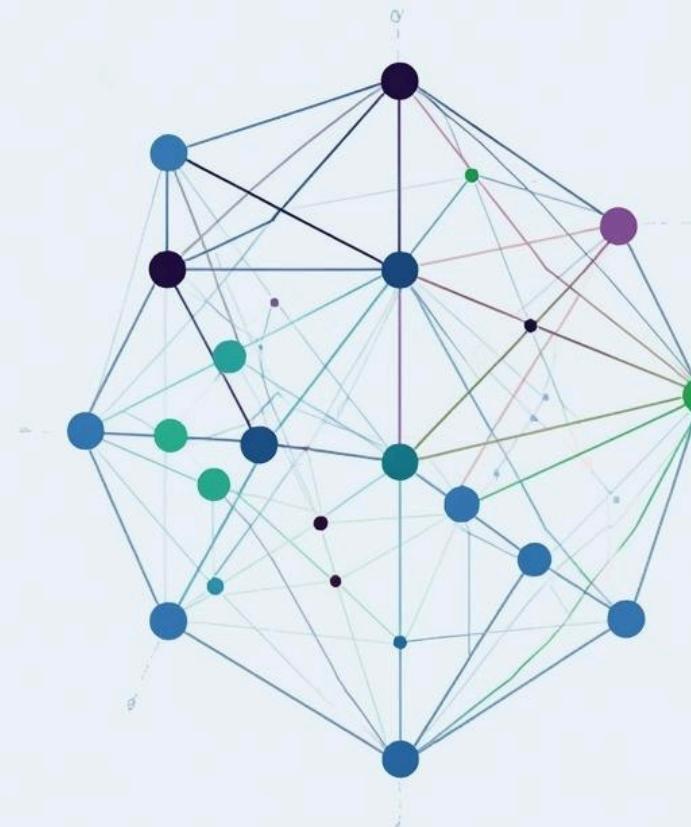
Yusuf Diler - 032290015

Eren Yılmaz - 032290114

Barış Kabacaoğlu - 032290027

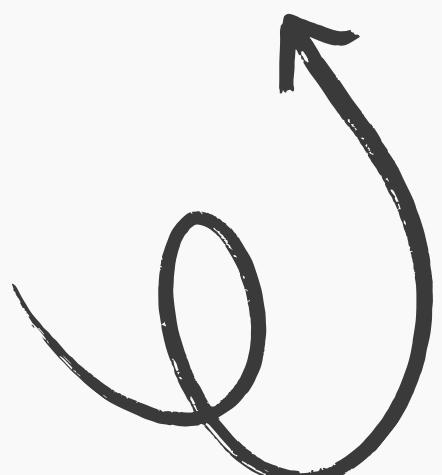
Zeynep Buse Can - 032290055

Mehmet Küsgül - 032290075



Sunum AKİŞI

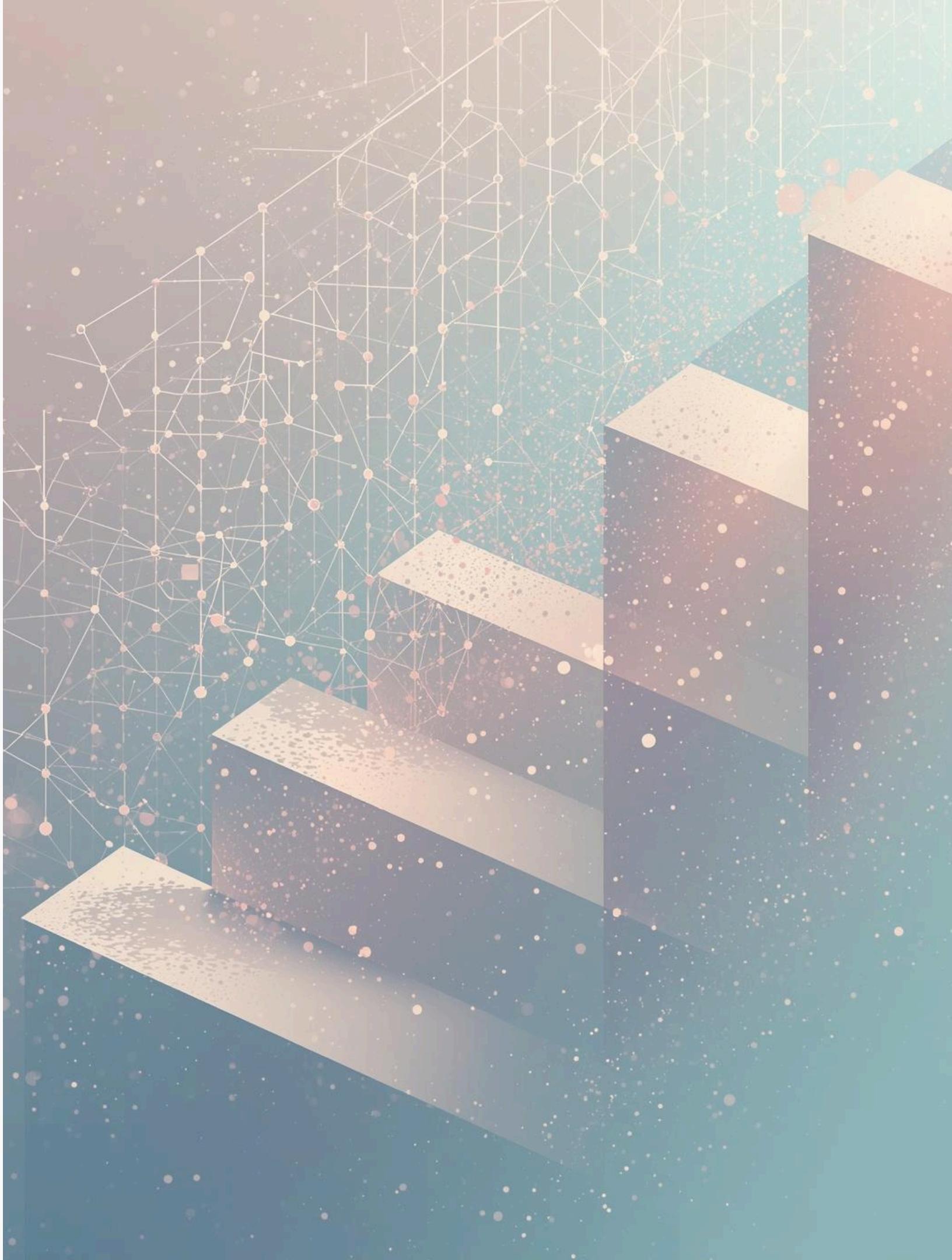
- 1. Giriş ve Problem Tanımı**
- 2. Çözüm Yaklaşımları**
- 3. Metodoloji ve Matematiksel Model**
- 4. Literatür Analizi**
- 5. Sonuç ve Değerlendirme**



Giriş ve Problem Tanımı

Konunun Önemi ve Mevcut Durum

Son yıllarda Derin Sinir Ağları (DNN) ve özellikle ResNet (Residual Networks) mimarileri, bilgisayarlı görüp (computer vision) alanında devrim niteliğinde başarılar elde etmiştir. Görüntü sınıflandırma, nesne tespiti ve yüz tanıma gibi karmaşık görevlerde, ağ derinliğinin artırılması (katman sayısının çoğaltıılması) ile modelin öğrenme kapasitesi arasında doğrudan bir korelasyon olduğu kanıtlanmıştır. Ancak, literatürdeki "daha derin daha iyidir" yaklaşımı, pratik uygulamalarda ve eğitim süreçlerinde ciddi darboğazlara yol açmaktadır.



Giriş ve Problem Tanımı

Eğitim Kararlılığı (Training Stability)

Derinlik arttıkça, hata fonksiyonunun (loss function) yüzeyi daha karmaşık ve "non-convex" (dışbükey olmayan) bir hale gelir.

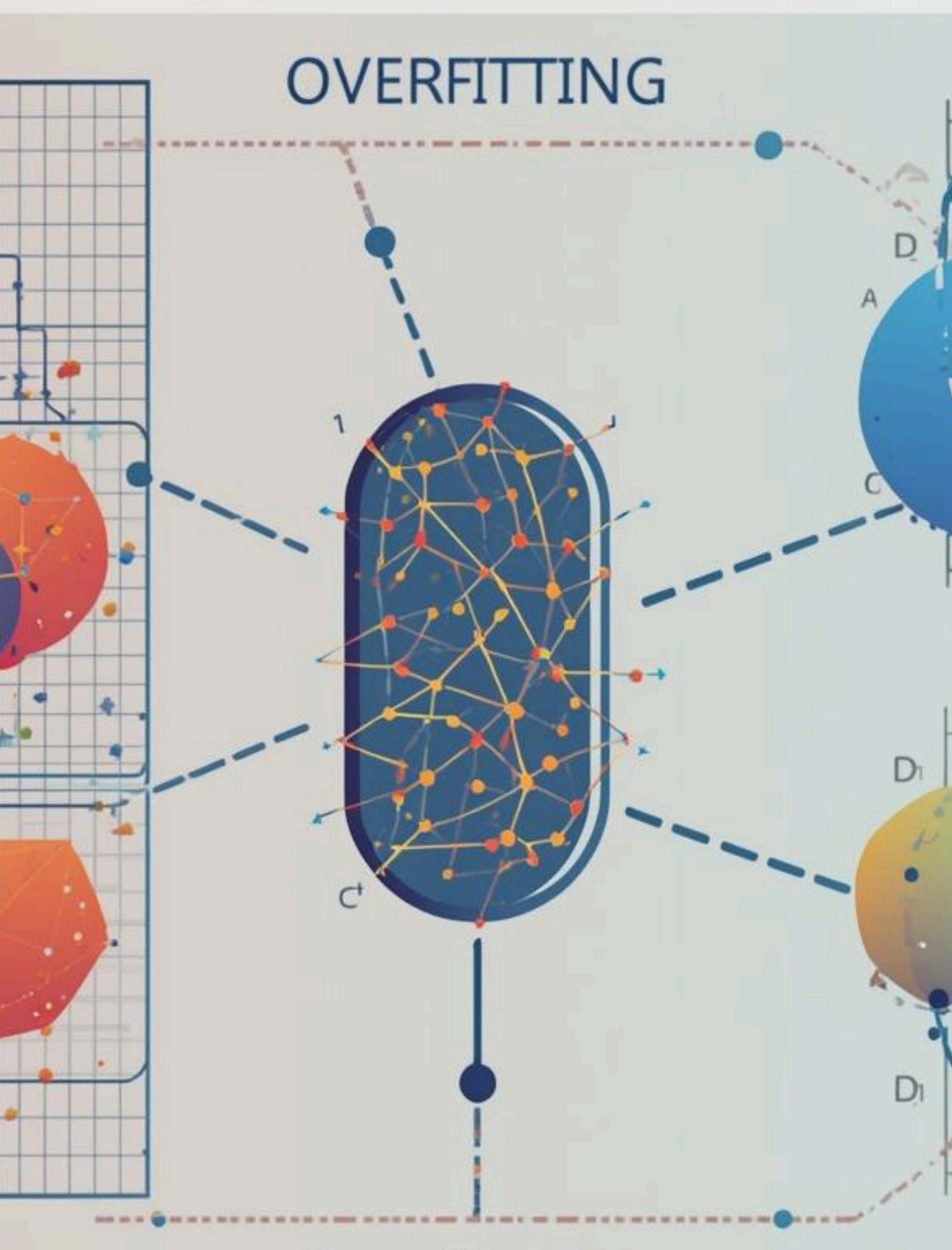
ResNet mimarisi "skip connection" yapısıyla Kaybolan Gradyan (Vanishing Gradient) problemini hafifletse de, yüzlerce katmana sahip ağlarda bilgi akışı ve geri yayılım (backpropagation) sırasında gradyanların kararsızlaşması sorunu devam etmektedir. Bu durum, modelin yakınsama (convergence) süresini uzatmakta veya eğitimin tamamen başarısız olmasına neden olabilmektedir.



Giriş ve Problem Tanımı

Aşırı Öğrenme ve Genelleme Sorunu

Milyonlarca parametreye sahip derin ağlar, "aşırı parametreleşmiş" (over-parameterized) yapılar olarak kabul edilir. Eğer eğitim verisi yeterince büyük ve çeşitli değilse, bu devasa kapasite, verideki genel örüntüleri öğrenmek yerine eğitim setini "ezberlemeye" (memorization) yönlendir. Bu rapor, ağın karmaşıklığını dinamik olarak yöneterek, ezberlemenin önüne geçmeyi ve modelin hiç görümediği veriler üzerindeki başarısını (genelleme yeteneğini) artırmayı hedefleyen yöntemleri incelemektedir.



Giriş ve Problem Tanımı

İşlem Maliyeti ve Kaynak Kısıtları

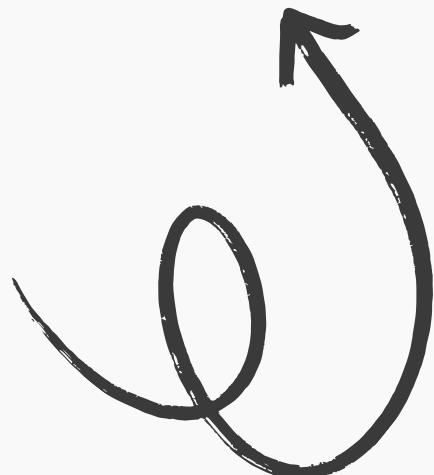
Belki de en kritik problem, bu modellerin Hesaplama Yükü (FLOPs) ve Bellek Tüketimidir. Standart bir ResNet-152 modelinin eğitimi için gereken GPU gücü ve enerji miktarı çok yüksektir. Daha da önemlisi, eğitilen bu devasa modellerin mobil cihazlar, gömülü sistemler veya IoT cihazları gibi sınırlı kaynağa sahip donanımlarda çalıştırılması (inference) neredeyse imkansız hale gelmektedir.



Çözüm Yaklaşımları

Metotlar

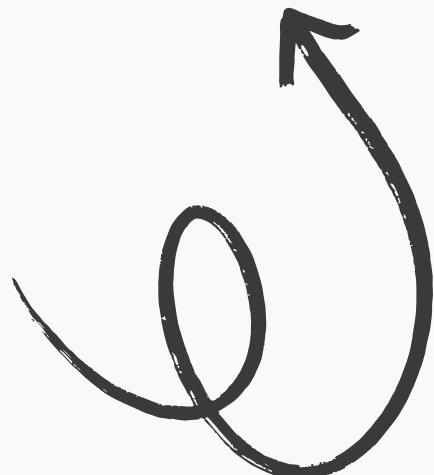
Katman Kapatma (Layer Dropping), Stokastik Derinlik (Stochastic Depth) ve Budama (Pruning) tekniklerini birer çözüm aracı olarak konumlandırmaktadır. Amaç, ResNet mimarisinin güçlü temsil yeteneğinden ödün vermeden; gereksiz (redundant) katmanların tespit edilip elendiği, eğitimin daha kararlı olduğu ve işlem yükünün optimize edildiği hibrit bir mimari yaklaşımı sunmaktır. Bu çalışma, statik bir ağ yapısı yerine, eğitimin gidişatına veya katmanların önem derecesine (duyarlılık analizi) göre şekillenen dinamik bir ağ yapısını savunmaktadır.



Çözüm Yaklaşımları

Çalışmanın Amacı

Çalışmamızın ana ekseni, derin yapay sinir ağlarında (Deep Neural Networks) eğitimin verimliliğini ve başarısını artırmak için katmanların veya blokların stratejik olarak devre dışı bırakılmasıdır. ResNet (Residual Networks) mimarisi, sahip olduğu "skip connection" (atlamlı bağlantı) yapısı sayesinde, bilginin bir katman kapatıldığında bile akmaya devam etmesini sağladığı için bu çalışma için en uygun matematiksel zemini oluşturur.

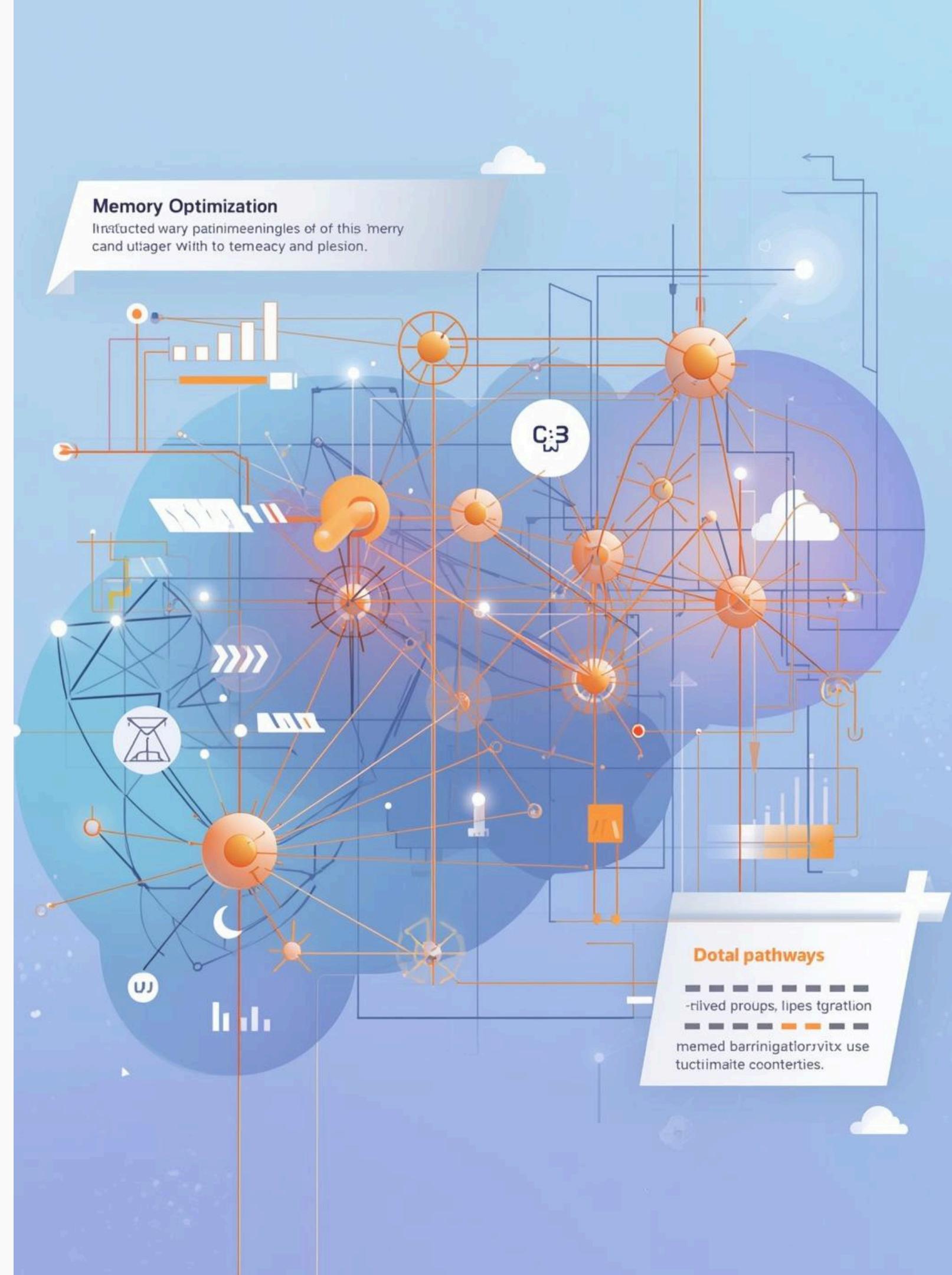


Metodoloji ve Matematiksel Model

İki Ana Yaklaşım

1. Eğitim Sırasında Kapatma (Stochastic Depth): Ağın eğitimi esnasında rastgele katmanların atlanması (bypass) yoluyla modelin "daha kısa" yollar öğrenmesini sağlamak. Bu işlem, ağın bir topluluk (ensemble) gibi davranışmasını sağlayarak güçlü bir düzenlileştirme (regularization) etkisi yaratır.

2. Çıkarım Sırasında Kapatma (Pruning/Inference Acceleration): Eğitilmiş modelin hızlanması için gereksiz katmanların çıkarım (test) aşamasında tamamen devre dışı bırakılmasıdır.



Metodoloji ve Matematiksel Model

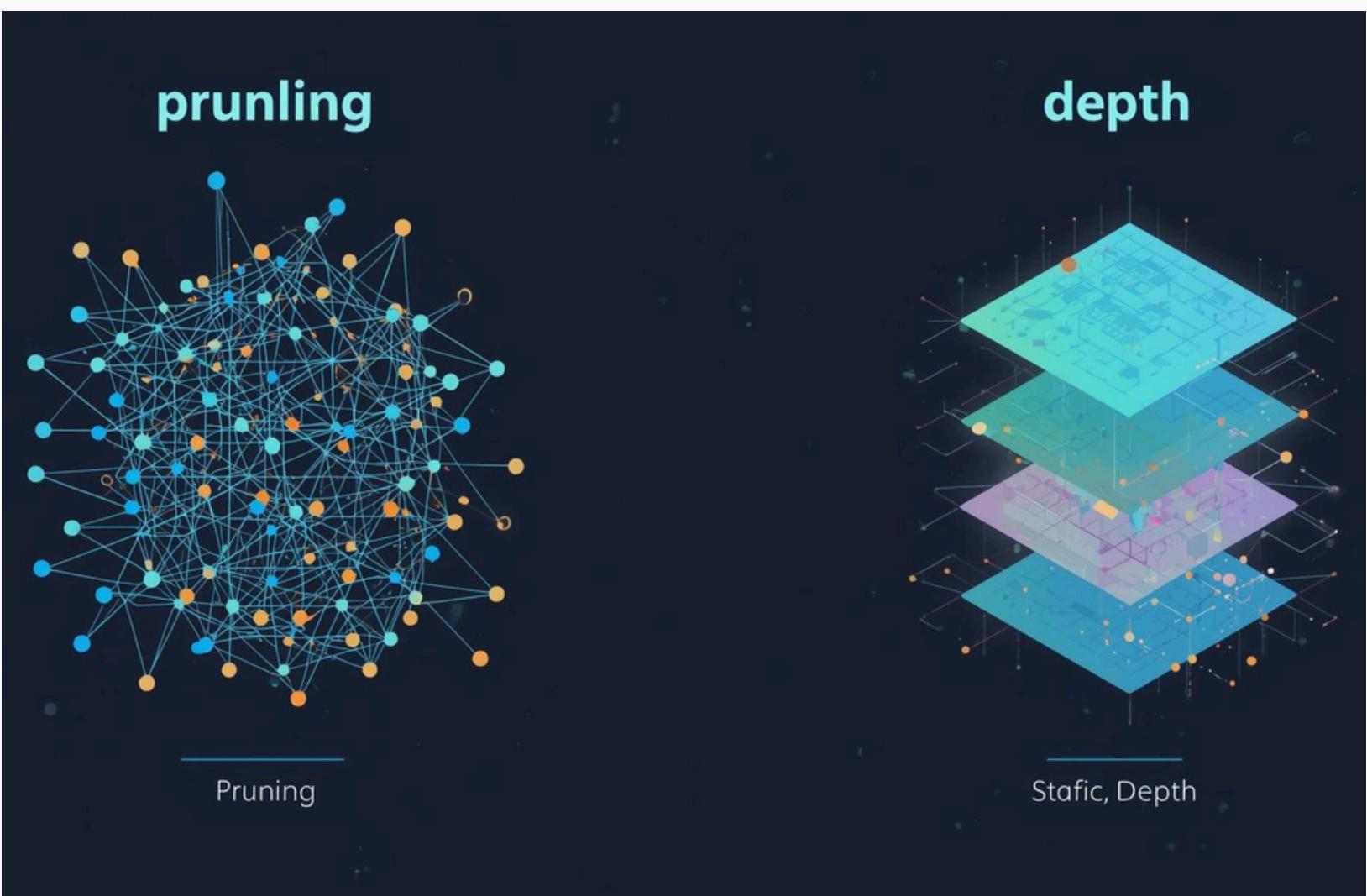
Özellik	Stochastic Depth (Rastgele Derinlik)	Pruning (Budama)
İşlem Tipi	Geçici / Anlık	Kalıcı / Sonsuza dek
Amaç	Modeli zorlayarak güçlendirmek (Eğitim)	Modeli hafifleteerek hızlandırmak (Sıkıştırma)
Metafor	"Antrenmanda tek kolunu bağlamak"	"Gereksiz eşyaları çöpe atmak"
Sonuç	Daha zeki ve dayanıklı bir beyin	Daha küçük ve hızlı bir beyin

Metodoloji ve Matematiksel Model

Neden Pruning Yapılır?

Derin öğrenme modelleri genellikle "aşırı parametreli" (over-parameterized) yapılardır; yani bir işi öğrenmek için gerekenden çok daha fazla nörona sahiptirler. Pruning şu avantajları sağlar:

- Daha Az Hafıza: Modelin dosya boyutu küçülür, mobil cihazlara veya IoT cihazlarına sığması kolaylaşır.
- Daha Hızlı Çıkarım (Inference): Daha az işlem (FLOPs) gerektiği için model veriyi daha hızlı işler.
- Enerji Verimliliği: Özellikle pil ile çalışan cihazlarda güç tüketimini azaltır.



Metodoloji ve Matematiksel Model

Pruning Süreci (Pipeline)

Pruning genellikle 3 aşamalı bir döngü ile yapılır:

- 1.Eğitim (Training): Büyük ve karmaşık bir model eğitilir.
- 2.Budama (Pruning): Önemsiz ağırlıklar belirlenir ve silinir (veya maskelenir).
 - Kriter: Genellikle "Magnitude-based" (Büyüklük tabanlı) yaklaşım kullanılır; "Mutlak değeri en küçük olan ağırlıklar en önemsizdir" varsayıımı yapılır.
- 3.İnce Ayar (Fine-tuning): (En Kritik Adım) Bağlantılar kesildiğinde modelin doğruluğu (accuracy) düşer. Bu düşüşü toparlamak için model, kalan ağırlıklarla tekrar kısa bir süre eğitilir.

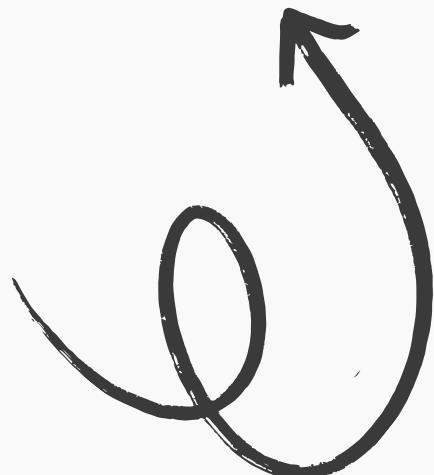


Literatür Analizi

Literatürün Genel Panoraması

Bu çalışma kapsamında incelenen 20 temel kaynak, derin sinir ağlarının verimliliğini artırmak için üç ana eksende toplanmıştır.

- Teorik Temeller: Rastgeleliğin ve katman yapısının matematiksel analizi.
- Verimlilik (Efficiency): İşlem yükünü azaltan dinamik yaklaşımlar.
- Yapısal Analiz: Modellerin neden çökmediğine dair yorumlanabilirlik çalışmaları.

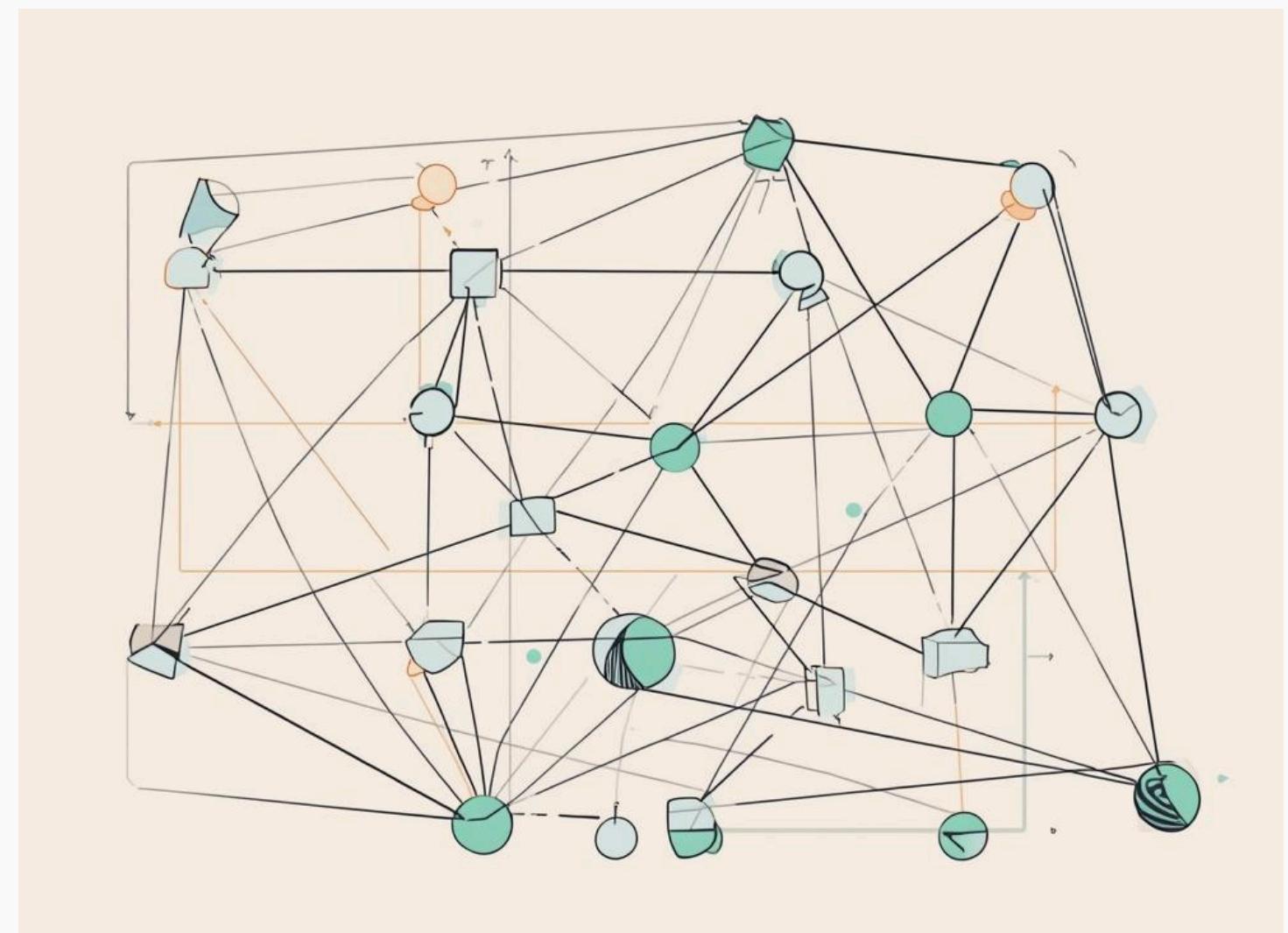


Literatür Analizi

Huang et al. (2016) - Deep Networks with Stochastic Depth

Bu yöntemin temel felsefesi, derin ağların eğitim süresini kısaltmak ve aşırı öğrenmeyi (overfitting) engellemektir. Eğitim katmanları rastgele kapatılarak hem süreyi kısaltmış hem de hatayı azaltmıştır.

- Eğitim sırasında katmanlar rastgele devre dışı bırakılır (drop). Bu sayede ağ, aslında farklı derinliklere sahip çok sayıda "alt ağın" birleşimi (ensemble) gibi davranmaya zorlanır.
- Bu rastgelelik, katmanların birbirine aşırı uyum sağlamaşını (co-adaptation) engeller. Yani bir katman, "bir önceki katman nasılsa hatayı düzeltir" diyemez; kendisi ayırt edici özellikler (discriminative features) öğrenmek zorundadır.



Literatür Analizi

Her bir katman için eğitimin o anki adımda katmanın çalışıp çalışmayağına karar veren bir b değişkeni tanımlanır:

$b_l \in \{0, 1\}$: Eğer değer **1** ise katman çalışır, **0** ise katman devre dışı kalır.

Modelin temel çalışma prensibi şu formülle ifade edilir:

$$H_l = \text{ReLU}(b_l \cdot f_l(H_{l-1}) + \text{id}(H_{l-1}))$$

- $f_l(H_{l-1})$: Katmanın yaptığı asıl hesaplama (konvolüsyon vb.).
- $\text{id}(H_{l-1})$: "Identity" yani girişin olduğu gibi aktarılması.

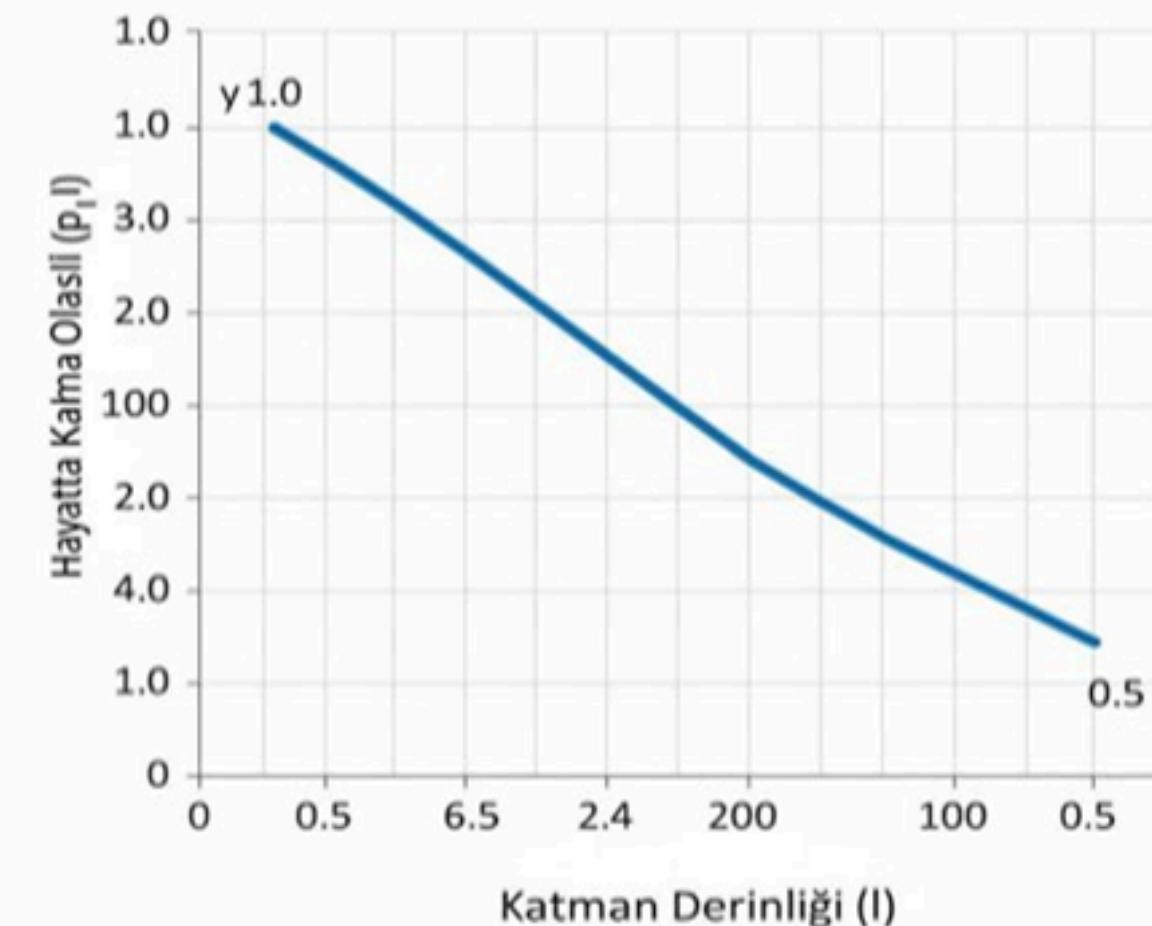
Literatür Analizi

Ağdaki tüm katmanlar aynı olasılıkla kapatılmaz.
Lineer Azalma Kuralı (Linear Decay Rule)
uygulanır:

$$p_l = 1 - \frac{l}{L} \cdot (1 - pL)$$

Girişe yakın katmanlar daha kritiktir ve hayatı
kalma olasılıkları daha yüksektir.

Lineer Azalma Kuralı



Sonuç ve Değerlendirme

- Stochastic Depth: Bir sıkıştırma değil, eğitim regülarizasyonudur. Rastgele kapatma ile ağıın bir "topluluk" (ensemble) gibi davranışmasını sağlar ve katmanlar arası eş-uyumu (co-adaptation) bozar.
- Pruning (Budama): Zayıf katmanların tespit edilip kalıcı olarak silinmesidir; amacı öğrenme kapasitesini değil, hızı artırmaktır.
- Bizim Yaklaşımı: Bu iki yöntemi birleştirerek hem eğitim gürbüzlüğünü (robustness) hem de çıkışım hızını optimize etmeyi hedefler.

