YAPAY ZEKA VE MAKİNE ÖĞRENMESİNE GİRİŞ

1. Yapay Zekaya Giriş ve Temel Kavramlar

1.1. Temel Tanımlar

- ***Yapay Zeka (Artificial Intelligence - AI):*** Bilgisayar sistemlerinin öğrenme, akıl yürütme ve problem çözme gibi insan zekasına özgü yeteneklerini taklit etmesini sağlayan bir teknolojidir.
- ***Makine Öğrenmesi (Machine Learning - ML):*** Bilgisayarların açıkça programlanmadan verilerden ve deneyimlerden yararlanarak öğrenmesini ve zamanla kendini geliştirmesini sağlayan bir yapay zeka dalıdır.
- ***Derin Öğrenme (Deep Learning - DL):*** Bilgisayarların insan beynindeki sinir ağlarından esinlenen yapılar kullanarak büyük verilerdeki kalıpları kendi kendine öğrenmesini sağlayan bir makine öğrenmesi alt dalıdır.
- ***Yapay Sinir Ağları (YSA / Neural Networks):*** İnsan beyninin bilgi işleme yöntemini taklit eden bir yapay zeka teknolojisidir.

1.2. Alanların Hiyerarşik Yapısı (Venn Şeması)

Defter notlarında belirtilen ve yapay zeka disiplinleri arasındaki ilişkiyi gösteren iç içe küme hiyerarşisi şu şekildedir:

> 📌 ***Yapay Zeka Disiplinleri Hiyerarşisi***
>
> [En Geniş Çerçeve] Yapay Zeka (Artificial Intelligence)
> └── [Alt Dal] Makine Öğrenmesi (Machine Learning)
> └── [Alt Dal] Derin Öğrenme (Deep Learning)
> └── [Alt Yapı/Teknoloji] Yapay Sinir Ağları (Neural Networks)
> Not: Bu disiplinlerin tamamı birbiriyle hiyerarşik bağ içinde olan ancak odaklandıkları yöntemler açısından birbirinden farklılaşan alt dallardır.

1.3. Bir Problemi Çözme ve İş Süreci Adımları

Ders notlarında problem çözme ve algoritma geliştirme süreci 4 temel aşamada özetlenmiştir:

1. ***İş Verildi:*** Problemin tanımlanması ve görevin belirlenmesi.
2. ***Belirsizliklerin Tespit Edilmesi:*** Problemdeki bilinmeyenlerin, eksik verilerin veya belirsiz dinamiklerin ortaya çıkarılması.
3. ***Belirsizliklerin Giderilmesi:*** Veri toplama, analiz, modelleme veya kural belirleme yoluyla belirsizliklerin çözülmesi.
4. ***İşin Yapılması:*** Modelin eğitilmesi veya nihai çözümün hayata geçirilmesi.

1.4. Öğrenme Teorisi ve Beyinde Öğrenme Mekanizması

- ***Öğrenme Tanımı:*** Bireyin yaşantıları ve deneyimleri sonucunda davranışlarında ya da zihninde meydana gelen nispeten kalıcı izli değişim sürecidir.
- ***Öğrenme Döngüsü:*** Denemek ➔ Hata Yapmak ➔ Hatadan Ders Çıkarıp Bir Sonraki Adımı Daha İyi Yapmak.
- ***Beyinde Biyolojik Öğrenme:*** Beyinde yeni bir bilgi öğrenilirken haberciler (nöronlar) ilk kez bir araya gelir ve aralarında zayıf bir bağlantı (sinaps) oluşur. Eğer o bilgi ne kadar çok tekrar edilirse nöronlar arasındaki bağlar o kadar güçlenir ve kalınlaşır. Artık o bilgi düşünmeden, son derece hızlı bir şekilde hatırlanır ve uygulanır. Ancak o bilgi kullanılmazsa nöronlar arasındaki bağlantılar zayıflar ve zamanla kaybolur.

1.5. Yapay Sinir Ağlarının Çalışma Prensibi

- ***Nöronlar (Temsil):*** Biyolojik sinir hücrelerine benzer şekilde yapay sinir hücreleri (nöronlar), verileri alır ve işler.
- ***Katmanlar (Yapı):*** Nöronlar üç ana katmanda organize edilir: (1) Verileri alan Girdi Katmanı (Input Layer), (2) Hesaplama yapan Gizli Katmanlar (Hidden Layers) ve (3) Sonucu veren Çıkış Katmanı (Output Layer).
- ***Öğrenme Süreci:*** Sistem, örnek veriler üzerinden tahminler yürütüp hata yaparak ve bu hatalara göre ağırlıkları (weights) güncelleyerek zamanla daha doğru kararlar vermeyi öğrenir.

1.6. Sınav Sorusu: Kara Kutu (Black Box) Modeli

> 📌 ***Sınav Sorusu İncelemesi: Kara Kutu Algoritması***
>
> Soru/Problem Mantığı: Giriş ve çıkış değerleri bilinen bir sistemde iç mekanizmanın modellenmesi.
>
> Girdiler: [3, 5, 7] ➔ Giriş Düğümü ➔ [KARA KUTU: İç Katman Kutucukları] ➔ Çıkış Düğümü ➔ Çıktılar: [6, 8, 9]
>
> Ders İçi Notlar ve Çözüm Yaklaşımı:
>
> • 'Bunun algoritması nasıl yazılır?'
>
> • 'Girdi ve çıktı tek bir hat gibi görünse de araya birkaç kutucuk (katman/düğüm) ekleyebiliriz; ancak başlangıçta hangisinin hangisine girip çıkacağını doğrudan bilemeyiz.'
>
> • 'Temel ilke: Girdiyi çıktıya göre uyduracaksın (yani ağırlıkları çıktıdaki hedefi tutturacak şekilde optimize edeceksin).'

1.7. Temel Kullanım Alanları

- ***Görüntü İşleme:*** Yüz tanıma, otonom araçlarda nesne tespiti, tıbbi görüntü analizi.
- ***Doğal Dil İşleme (NLP):*** Metin analizi, makine çevirisi, özet çıkarma, duygu analizi.
- ***Tahmin ve Sınıflandırma:*** Finansal öngörüler, risk analizi, sağlık sektöründe hastalık teşhisi.

2. Makine Öğrenmesi Görevleri: Regresyon ve Sınıflandırma

2.1. Temel Değişken Kavramları

- ***Bağımlı Değişken (y):*** Değeri tahmin edilmeye veya açıklanmaya çalışılan ana etkendir (hedef/sonuç).
- ***Bağımsız Değişken (x):*** Bağımlı değişkeni etkileyen veya onun değişimini açıklayan sebeptir (girdi/etken).
- ***Matematiksel Model:*** Değişkenler arasındaki bağı kuran matematiksel denklemdir:

> 📌 ***Matematiksel Formül:***
>
> $y = b_o + b_1x$
>
> Burada:
> • y: Bağımlı değişken (tahmin edilmek istenen sonuç)
> • x: Bağımsız değişken (tahminde kullanılan girdi/etken)
> • b_o: Sabit terim (doğrunun y eksenini kestiği nokta / bias)
> • b_1: Regresyon katsayısı veya eğim (x'teki 1 birimlik değişimin y üzerindeki etkisi)

2.2. Regresyon (Sürekli Değer Tahmini)

Regresyon Ne İşe Yarar?

- ***Geleceği Tahmin Etme:*** Geçmişteki verilere bakarak gelecekteki sayısal durumları öngörmeyi sağlar.
- ***İlişkiyi Ölçme:*** Değişkenler arasında bir bağ olup olmadığını, varsa bu bağın ne kadar güçlü olduğunu gösterir.
- ***Eğilimleri Belirleme:*** Verilerdeki kalıpları ve yönelimleri ortaya çıkarır.

Başlıca Regresyon Türleri:

- ***Basit Doğrusal Regresyon:*** Tek bir bağımsız değişkenin (x), tek bir bağımlı değişken (y) üzerindeki doğrusal etkisini inceler.
- ***Çoklu Regresyon:*** Birden fazla bağımsız değişkenin (x_1, x_2, ..., x_n), bir bağımlı değişken (y) üzerindeki ortak etkisini analiz eder.

2.3. Sınıflandırma (Classification)

Sınıflandırma, bilgisayar sistemlerinin verileri belirli özelliklerine bakarak önceden tanımlanmış kategorilere veya etiketlere ayırma sürecidir. Genellikle Denetimli Öğrenme (Supervised Learning) dalında kullanılır. Model etiketli verilerle eğitilir ve yeni gelen bir verinin hangi sınıfa ait olduğunu tahmin eder.

Sınıflandırma Türleri:

- ***İkili (Binary) Sınıflandırma:*** Verinin yalnızca iki farklı kategoriden birine ait olabileceği durumlardır. Örnek: Bir e-postanın 'Spam' veya 'Spam Değil' şeklinde ayrılması; bir tümörün 'İyi Huylu' veya 'Kötü Huylu' olarak teşhis edilmesi.
- ***Çok Sınıflı (Multi-Class) Sınıflandırma:*** Verinin üç veya daha fazla kategoriden birine atandığı durumlardır. Örnek: Bir fotoğraftaki hayvanın 'Kedi', 'Köpek' veya 'Kuş' olarak sınıflandırılması; el yazısı rakamların (0-9) tanınması.

Yaygın Kullanım Alanları:

- ***E-posta Filtreleme:*** Gelen kutusu iletilerini Birincil, Sosyal veya Spam olarak ayrıştırmak.
- ***Sağlık Sektörü:*** Tıbbi görüntüleri (MR, Röntgen) inceleyerek patolojik bulguların iyi/kötü huylu olduğunu saptamak.
- ***Görsel Tanıma:*** Fotoğraflardaki nesneleri veya yüzleri tespit etmek.
- ***Müşteri Analizi:*** Müşterileri harcama alışkanlıklarına göre segmentlere ayırmak.

3. İstatistiksel Model Başarım Ölçütleri: R ve R-Kare

Yapay zeka ve makine öğrenmesi bağlamında R (korelasyon katsayısı) ve R-kare (R^2 / belirleme katsayısı), özellikle regresyon problemlerinde modelin tahmin gücünü ve veriye ne kadar iyi uyum sağladığını ölçmek için kullanılan iki temel istatistiksel metriktir.

3.1. R (Pearson Korelasyon Katsayısı)

İki değişken arasındaki doğrusal ilişkinin gücünü ve yönünü ölçen katsayıdır. Model kurulmadan önce veya veri analizi (EDA) aşamasında, hangi girdilerin hedef çıktıyla ilişkili olduğunu anlamak için kullanılır.

- ***Değer Aralığı:*** -1 ile +1 arasındadır.
- ***+1:*** Mükemmel pozitif bir doğrusal ilişki vardır (biri artarken diğeri de doğrusal olarak artar).
- ***-1:*** Mükemmel negatif bir doğrusal ilişki vardır (biri artarken diğeri doğrusal olarak azalır).
- ***0:*** İki değişken arasında doğrusal bir ilişki yoktur.

> 📌 ***Pearson Korelasyon Katsayısı (r) Formülü***
>
> $r = [\, n(\Sigma xy) - (\Sigma x)(\Sigma y) \,] \,/\, \sqrt{[\, (n\Sigma x^2 - (\Sigma x)^2) \cdot (n\Sigma y^2 - (\Sigma y)^2) \,]}$
>
> Formüldeki Değişkenlerin Anlamı:
>
> • r: Pearson korelasyon katsayısı
>
> • n: Toplam gözlem (veri noktası) sayısı
>
> • Σxy: x ve y değişkenlerinin karşılıklı çarpımlarının toplamı
>
> • Σx, Σy: x ve y değerlerinin ayrı ayrı toplamları
>
> • Σx^2, Σy^2: x ve y değerlerinin karelerinin toplamı
>
> Örnek değişken ilişkileri: Yaş, Boy, Kilo, Maaş, Araba Değeri.

3.2. R-Kare (R^2 / Belirleme Katsayısı)

R-kare, R'nin karesinin alınmasıyla elde edilir ($R \times R = R^2$) ve bir regresyon modelindeki bağımlı değişkendeki varyansın (değişkenliğin), modele dahil edilen bağımsız değişkenler tarafından ne oranda açıklandığını gösterir.

- ***Değer Aralığı:*** 0 ile 1 (veya %0 ile %100) arasındadır.
- ***1 (%100):*** Model verilere mükemmel uyum sağlıyor; tüm tahminler gerçek değerlerle birebir örtüşüyor.
- ***0 (%0):*** Model verideki değişkenliği hiç açıklayamıyor; rastgele tahmin yapmaktan farksızdır.

> 📌 ***R^2 Değerinin Pratikte Yorumlanması (Ev Fiyatı Örneği)***
>
> Örnek Senaryo: Ev fiyatlarını tahmin eden bir yapay zeka modelinin R^2 değeri 0.85 olarak hesaplanmışsa:
>
> Bu değer şu anlama gelir: 'Ev fiyatlarındaki değişimin %85'i, modele verdiğimiz bağımsız değişkenler (oda sayısı, metrekare, konum vb.) tarafından açıklanabilmektedir. Kalan %15 ise model dışı faktörlerden veya rastgele gürültüden kaynaklanmaktadır.'
>
> 📌 ***R-Kare (R^2) Matematiksel Hesaplama Formülü***
>
> $R^2 = 1 - (RSS / TSS)$
>
> Formül Bileşenleri:
>
> • TSS (Total Sum of Squares / Toplam Kareler Toplamı): Bağımlı değişkendeki toplam varyasyonu/değişimi ifade eder.
>
> • RSS (Residual Sum of Squares / Artık Kareler Toplamı veya SSE): Modelin tahmin ettiği değerler ile gerçek değerler arasındaki hataların kareler toplamıdır; modelin açıklayamadığı kısmı gösterir.
>
> (Not: Bazı kaynaklarda RSS yerine SSE veya SSR sembolleri kullanılabilir).

3.3. R ve R-Kare Özet Karşılaştırma Tablosu

Özellik	R (Korelasyon Katsayısı)	R-Kare (R^2 - Belirleme Katsayısı)
Neyi ölçer?	İki değişken arasındaki doğrusal ilişkinin yönünü ve gücünü	Modelin verideki değişkenliği açıklama yüzdesini
Değer aralığı	-1 ile +1 arası	0 ile 1 (veya %0 - %100) arası
Kullanım amacı	Değişkenler arası ilişkiyi ve korelasyonu görmek	Regresyon modelinin başarımını ve veriye uyumunu değerlendirmek

4. Yapay Sinir Ağlarının Temeli: Perceptron Modeli

Perceptron, yapay sinir ağlarının temelini oluşturan en basit tek katmanlı yapay sinir hücresi modelidir. Biyolojik nöronun matematiksel bir soyutlamasıdır.

4.1. Perceptron'un Temel Bileşenleri

- ***Girdiler (Inputs - x_1, x_2):*** Dış ortamdan veya bir önceki katmandan alınan veri değerleridir.
- ***Ağırlıklar (Weights - w_1, w_2):*** Girdilerin önem derecesini ve modele etkisini belirleyen katsayılardır. Öğrenme sürecinde güncellenen parametrelerdir.
- ***Yanlılık (Bias - b):*** Karar sınırını koordinat düzleminde kaydırmaya yarayan ve tüm girdiler sıfır olduğunda dahi modelin esnek çalışmasını sağlayan ek parametredir.
- ***Ağırlıklı Toplam (Weighted Sum - Σ):*** Girdiler ile ilgili ağırlıkların çarpımlarının bias ile toplanması işlemidir.
- ***Aktivasyon Fonksiyonu:*** Ağırlıklı toplamı işleyerek çıktının '0' veya '1' (ikili karar) olmasını sağlayan eşik değer fonksiyonudur.

> 📌 ***Perceptron Matematiksel Mimarisi***
>
> Ağırlıklı Toplam: $\Sigma = x_1w_1 + x_2w_2 + b$
>
> Basamak Fonksiyonu (Step Function):
> $f(x) = \{$
> 0 , eğer $x < 0$ ise
> 1 , eğer $x \geq 0$ ise
> $\}$
> Bu mekanizmada elde edilen net toplam (Σ), aktivasyon fonksiyonuna girdi olarak verilir ve model ikili (binary) bir çıktı (0 veya 1) üretir.

4.2. Perceptron'un Çalışma Mantığı ve Sınırlamaları (XOR Problemi)

Perceptron doğrusal olarak ayrılabilen (linearly separable) problemleri başarıyla çözebilirken, doğrusal olmayan karmaşık örüntülerde yetersiz kalır:

- ***Çözebildiği Problemler:*** AND (Ve) ve OR (Veya) gibi girdilerin bir doğru parçası ile iki sınıfa ayrılabildiği doğrusal problemleri başarıyla çözer.
- ***Sınırlaması (XOR Problemi):*** XOR (Özel Veya) gibi doğrusal olarak tek bir çizgiyle ayrılamayan problemleri tek katmanlı Perceptron TEK BAŞINA ÇÖZEMEZ. Bu durum yapay zeka tarihindeki ilk kış dönemine (AI Winter) yol açmıştır.
- ***Çözüm:*** XOR sınırının aşılması ancak çok katmanlı yapıların (MLP - Multi-Layer Perceptron / Çok Katmanlı Algılayıcı) ve doğrusal olmayan aktivasyon fonksiyonlarının kullanılmasıyla mümkün olmuştur.

5. Yapay Sinir Ağlarında Öğrenme ve Optimizasyon: Geri Yayılım

Backpropagation (Geri Yayılım), yapay sinir ağlarının öğrenme sürecinde kullanılan temel ve en kritik optimizasyon algoritmasıdır.

5.1. Temel Mantığı

Sinir ağının ürettiği tahmin ile gerçek değer arasındaki hata (kayıp) miktarını hesaplar ve bu hatayı ağın çıkışından girişine doğru (geriye dönük olarak) her bir katmandaki ağırlıklara (weights) dağıtır. Amaç, ağın sonraki tahminlerinde daha az hata yapmasını sağlamak için parametreleri güncellemektir.

5.2. Adım Adım Çalışma Aşamaları

Geri yayılım süreci genellikle bir döngü (epoch döngüsü) halinde şu 4 adımda gerçekleşir:

1. ***İleri Yayılım (Forward Propagation):*** Girdi verileri sinir ağının katmanlarından geçer, başlangıçta rastgele veya mevcut ağırlıklarla işlenir ve bir tahmin/çıktı üretilir.
2. ***Hata Hesabı (Loss Calculation):*** Üretilen tahmin ile gerçek hedef değer karşılaştırılarak bir kayıp fonksiyonu (loss function) yardımıyla hata oranı bulunur.
3. ***Geri Yayılım (Backpropagation):*** Hata değeri sondan başa doğru türev yardımıyla (zincir kuralı - chain rule kullanarak) geriye taşınır ve her bir ağırlığın bu hatadaki payı (gradyanlar / gradients) hesaplanır.
4. ***Ağırlık Güncellemesi (Weight Update):*** Genellikle Stokastik Gradyan İnişi (SGD - Stochastic Gradient Descent) veya Adam gibi optimizasyon algoritmaları kullanılarak, ağırlıklar hatayı azaltacak yönde küçük adımlarla (öğrenme katsayısı / learning rate ile) güncellenir.

6. Kayıp Fonksiyonları (Loss Functions)

Yapay zeka ve makine öğrenmesinde kayıp fonksiyonu (loss function), bir modelin tahminleri ile gerçek veriler arasındaki hatayı veya farkı sayısal bir değer olarak ölçen temel matematiksel araçtır. Eğitim sürecinde amaç, bu kayıp değerini mümkün olan en düşük seviyeye indirmektir. Kayıp fonksiyonları, çözülmek istenen problemin türüne göre iki ana gruba ayrılır:

6.1. Regresyon (Sürekli Değer Tahmini) Kayıp Fonksiyonları

Sayısal değerler (ev fiyatı, sıcaklık, teslimat süresi, borsa tahmini vb.) tahmin edilirken kullanılır:

- ***Ortalama Kare Hata (MSE - Mean Squared Error / L2 Kaybı):*** Gerçek değer ile tahmin arasındaki farkın karesini alır ve ortalamasını hesaplar. Büyük hatalara karesi alındığı için yüksek ceza verir; ancak veri setindeki aykırı değerlerden (outliers) aşırı derecede etkilenir.
- ***Ortalama Mutlak Hata (MAE - Mean Absolute Error / L1 Kaybı):*** Gerçek ile tahmin arasındaki mutlak farkların ortalamasını alır. Tüm hatalara doğrusal davranır ve aykırı değerlere (outliers) karşı MSE'ye göre çok daha dayanıklıdır (robust).
- ***Huber Kaybı (Huber Loss):*** MSE ile MAE'nin güçlü yönlerini birleştiren hibrit bir fonksiyondur. Küçük hatalarda MSE gibi parabolik davranarak pürüzsüz türevlenebilirlik

sağlar; büyük (aykırı) hatalarda ise MAE gibi lineer bir ceza vererek modelin aşırı hassasiyetini ve sapmasını önler.

6.2. Sınıflandırma (Kategorik Tahmin) Kayıp Fonksiyonları

Verileri belirli sınıflara veya etiketlere ayırmak için kullanılır:

- ***İkili Çapraz Entropi (Binary Cross-Entropy / Log Loss):*** Çıktının sadece iki sınıf olduğu durumlarda (örneğin kedi/köpek, spam/normal, hasta/sağlıklı) kullanılır. Modelin olasılık çıktıları ile gerçek etiketler arasındaki bilgi farkını logaritmik olarak ölçer.
- ***Kategorik Çapraz Entropi (Categorical Cross-Entropy):*** İkiden fazla sınıfın bulunduğu senaryolarda (örneğin el yazısı rakam tanıma 0-9, nesne tespiti, dil modelleme) tercih edilir. Softmax çıktıları ile one-hot etiketler arasındaki farkı cezalandırır.
- ***Hinge Loss (Menteşe Kaybı):*** Genellikle Destek Vektör Makineleri (SVM) algoritmalarında, sınıflar arasında güvenli ve maksimum bir karar sınırı (margin) oluşturmak için kullanılır.

6.3. Kayıp Fonksiyonları Neden Hayati Öneme Sahiptir?

> 📌 ***Kayıp Fonksiyonunun Rolü***
>
> ***1. Geriye Yayılım (Backpropagation) Yönlendiricisi:*** Sinir ağları, kayıp fonksiyonunun parametrelere göre türevini (gradyanını) alarak modelin ağırlıklarını hangi yönde ve ne büyüklükte güncelleyeceğini tayin eder.
> ***2. Optimizasyon Başarımı:*** Doğru seçilmiş bir kayıp fonksiyonu, modelin hızlı, kararlı ve doğru öğrenmesini sağlar. Yanlış seçilen bir kayıp fonksiyonu ise modelin mantıksız tahminlerde takılıp kalmasına veya hiç öğrenememesine yol açar.

7. Aktivasyon Fonksiyonları (Activation Functions)

Aktivasyon fonksiyonu, bir yapay sinir ağındaki nöronun kendisine gelen girdilerin ağırlıklı toplamını işleyerek bu nöronun bir sonraki katmana çıktı üretip üretmeyeceğini (veya çıktının büyüklüğünü) belirleyen temel matematiksel bileşendir. Aktivasyon fonksiyonları, sinir ağlarının sadece düz (lineer) hesaplamalar yapmasının önüne geçer; modelin karmaşık ve doğrusal olmayan (non-linear) örüntüleri öğrenmesini sağlar.

7.1. Neden Aktivasyon Fonksiyonu Kullanılır?

- ***Doğrusal Olmama (Non-Linearity) Kazandırma:*** Aktivasyon fonksiyonu olmadan, kaç katman üst üste eklenirse eklensin (100 katman dahi olsa), tüm ağ matematiksel olarak tek bir doğrusal regresyon modeline indirgenir. Karmaşık problemleri (görüntü, ses, dil) çözmek için doğrusal olmayan fonksiyonlar zorunludur.
- ***Geri Yayılım (Backpropagation) Uyumluluğu:*** Modelin gradyan inişi ile öğrenebilmesi için fonksiyonun türevlenebilir olması şarttır. Türev, ağırlıkların hata oranına göre nasıl güncelleneceğini belirler.
- ***Sinyal Kontrolü:*** Nöronun biyolojik bir hücre gibi aktif olup olmayacağını bir kapı gibi denetler; aşırı büyük değerlerin patlamasını sınırlar.

7.2. En Yaygın Aktivasyon Fonksiyonları

a) Sigmoid (Lojistik Fonksiyon)

- ***Çıktı Aralığı:*** [0, 1]
- ***Kullanım Yeri:*** Genellikle ikili (binary) sınıflandırma problemlerinde çıkış katmanında olasılık üretmek için tercih edilir.
- ***Dezavantajı:*** Girdiler çok büyüdüğünde veya çok küçüldüğünde türevi sıfıra yaklaşır. Bu durum derin ağlarda gradyan sönmesi (vanishing gradient) problemine yol açar ve öğrenmeyi neredeyse tamamen durdurur. Ayrıca hesaplama açısından yavaştır.

b) Tanh (Hiperbolik Tanjant)

- ***Çıktı Aralığı:*** [-1, 1]
- ***Kullanım Yeri:*** Sigmoide benzer ancak orijin etrafında simetriktir (sıfır merkezli / zero-centered). Bu özellik gizli katmanlarda öğrenmenin yönü açısından Sigmoide göre avantaj sağlar.
- ***Dezavantajı:*** Uç noktalarda Sigmoid gibi yine gradyan sönmesi (vanishing gradient) sorunundan muzdariptir.

c) ReLU (Rectified Linear Unit)

- ***Çalışma Prensibi:*** f(x) = max(0, x). Negatif girdileri 0 yapar, pozitif girdileri olduğu gibi (eğim = 1) geçirir.

- ***Kullanım Yeri:*** Günümüzde derin öğrenmede ve gizli katmanlarda (özellikle Evrişimli Sinir Ağları - CNN) açık ara en çok tercih edilen fonksiyondur. Türevi çok kolay hesaplanır ve hesaplama hızını inanılmaz artırır.
- ***Dezavantajı:*** Negatif tarafta türevi mutlak sıfır olduğu için nöron bir kez negatif bölgeye kilitlenirse bir daha asla güncellenemez; buna 'ölü nöron' (dying ReLU) sorunu denir.

d) Leaky ReLU

- ***Çalışma Prensibi:*** ReLU'nun negatif bölgede tamamen sıfır olma sorununu çözmek için negatif tarafa çok küçük bir eğim verir (örneğin f(x) = 0.01x).
- ***Kullanım Yeri:*** Ölü nöronların önüne geçmek için standart ReLU'nun alternatifi olarak kullanılır.

e) Softmax

- ***Çıktı Aralığı:*** [0, 1] (tüm sınıfların olasılık çıktıları toplamı kesinlikle 1 eder).
- ***Kullanım Yeri***: Çok sınıflı (multi-class) sınıflandırma problemlerinin en dış (çıkış) katmanında sınıflara ait normalize edilmiş bir olasılık dağılımı elde etmek için standart olarak kullanılır.

7.3. Aktivasyon Fonksiyonları Karşılaştırma Özeti

Fonksiyon	Çıktı Aralığı	Başlıca Kullanım Yeri	Temel Özellik / Dezavantaj
Sigmoid	[0, 1]	İkili sınıflandırma çıkış katmanı	Olasılık verir; gradyan sönmesi dezavantajıdır
Tanh	[-1, 1]	Gizli katmanlar (sıfır merkezli)	Sigmoidden hızlıdır; gradyan sönmesi yaşar
ReLU	[0, ∞)	Derin öğrenme gizli katmanları (CNN)	Çok hızlıdır, standarttır; dying ReLU riski vardır
Leaky ReLU	(-∞, ∞)	Gizli katmanlar (ölü nöron alternatifi)	Negatif tarafta küçük eğim (0.01x) ile nöronu canlı tutar
Softmax	[0, 1] (Toplam=1)	Çok sınıflı sınıflandırma çıkış katmanı	Çoklu sınıflar için olasılık dağılımı üretir