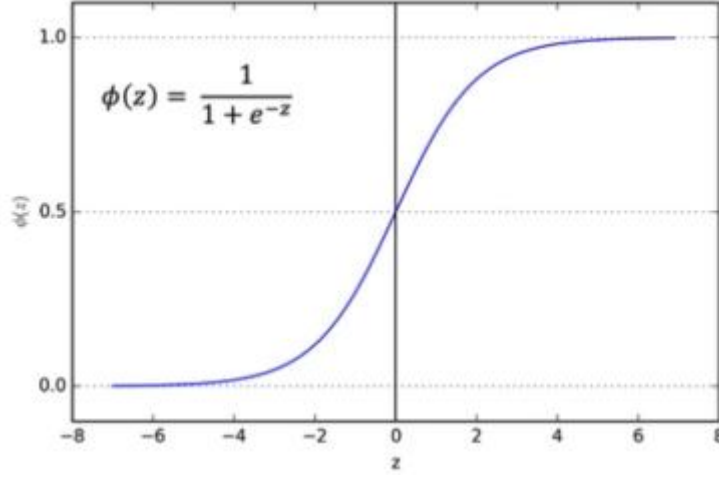


SIGMOID FONKSİYONU

Özellikle yapay sinir ağlarında sıkça kullanılan bir **aktivasyon fonksiyonudur**. Bu fonksiyon, bir girdiyi



(örneğin, bir nöronun ağırlıklı toplamını) **0 ile 1** arasında bir değere dönüştürür.

Özellikleri:

1. **Çıkış Aralığı:** 0 ile 1 arasındadır.
 - Negatif girişler 0'a, pozitif girişler 1'e yaklaşır.
2. **S-eğrisi Şeklinde:** Çıktıları düzgün bir şekilde sıkıştırır.
3. **Küçük Girişler için Yavaş Değişim:** Girdi değeri çok büyük veya çok küçük olduğunda, fonksiyonun çıktısı değişmeye direnç gösterir (gradyan problemi oluşabilir).

Avantajları:

- İkili sınıflandırma problemlerinde **olasılık** gibi yorumlanabilir bir çıktı sağlar. Örneğin, çıktı 0.8 ise "Bu durumun gerçekleşme olasılığı %80" diyebiliriz.

Dezavantajları:

- Büyük ağlarda **gradyan sönmesi** problemi yaşayabilir. (Bu, girdiler büyüdükçe türevin sıfıra yaklaşmasından kaynaklanır.)
- Çıktı aralığı 0'dan farklı olduğu için bazı algoritmalar (örneğin, ağırlık güncelleme) zorlaşabilir.

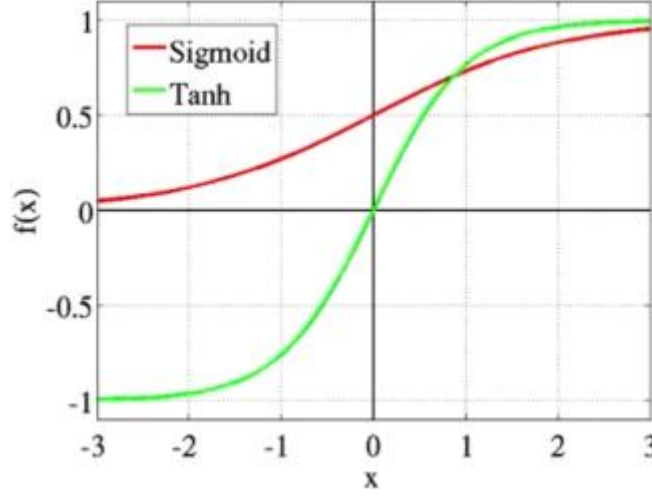
Nerede Kullanılır?

- İkili sınıflandırma problemleri. (Örneğin, bir e-posta **spam** mı, değil mi?)
- Sinir ağlarında gizli katmanların aktivasyon fonksiyonu olarak.

Kısaca, sigmoid, "**bir şeyi 0 ile 1 arasında olasılık olarak ifade etmek**" istediğimizde harika bir araçtır.

TANH (HİPERBOLİK TANJANT) FONKSİYONU

Sigmoid fonksiyonuna benzeyen, ancak biraz daha farklı özelliklere sahip bir **aktivasyon fonksiyonudur**. Girdiyi **-1 ile 1** arasında bir değere dönüştürür.



Özellikleri:

1. **Çıkış Aralığı:** -1 ile 1 arasındadır.
 - o Negatif girdiler -1'e, pozitif girdiler 1'e yaklaşır.
2. **S-eğrisi Şeklinde:** Girdi ile düzgün bir şekilde ilişkilidir, ancak simetrik bir yapıdadır.

Avantajları:

1. Girdi değerlerinin pozitif ve negatif yönlerini daha iyi ayırt eder.
2. Sıfır merkezli olduğu için gradyan inişi (gradient descent) algoritmasının daha hızlı çalışmasını sağlar.

Dezavantajları:

1. **Gradyan Sönmesi:** Sigmoid gibi, büyük veya küçük giriş değerlerinde türevin sıfıra yaklaşması nedeniyle öğrenme hızı yavaşlayabilir.
2. Büyük derin sinir ağlarında etkili olmayabilir; bu durumlarda ReLU (Rectified Linear Unit) gibi diğer aktivasyon fonksiyonları tercih edilir.

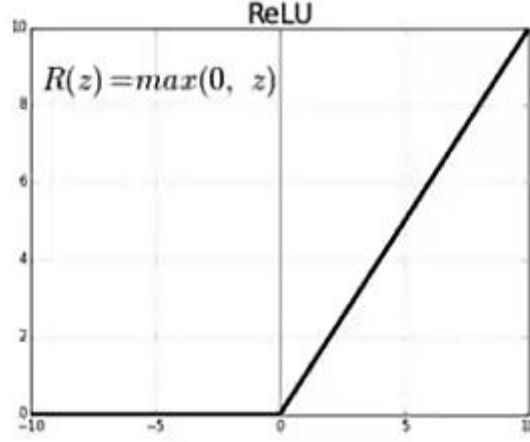
Nerede Kullanılır?

- Sıfır merkezli verilerle çalışılan problemlerde.
- RNN (Recurrent Neural Network) gibi bazı derin öğrenme modellerinde.

Kısaca, **tanh fonksiyonu**, giriş değerlerini -1 ile 1 arasında normalize ederek öğrenmeyi hızlandırabilir ve veri merkezinin sıfır olmasını sağladığı için sigmoid'e göre genellikle daha etkilidir.

RELU (RECTIFIED LINEAR UNIT)

Günümüzde derin öğrenme modellerinde en yaygın kullanılan **aktivasyon fonksiyonudur**. Basitliği ve performansı sayesinde özellikle derin sinir ağlarında (Deep Neural Networks) tercih edilir.

**Matematiksel Tanımı:**

$$\text{ReLU}(x) = \max(0, x)$$

- Eğer $x > 0$ ise: Çıkış olur.
- Eğer $x \leq 0$ ise: Çıkış 0 olur.

Özellikleri:

1. **Çıkış Aralığı:** 0 ile $+\infty$ arasındadır.
2. **Doğrusal Olmayan Doğa:** Fonksiyon doğrusal değildir, bu da ağın karmaşık ilişkileri öğrenmesine olanak tanır.

Avantajları:

1. **Basit ve Hesaplaması Hızlıdır:** Sigmoid ve tanh gibi fonksiyonlarda üstel hesaplamalar varken, ReLU sadece maksimum değer alır. Bu, hesaplama yükünü azaltır.
2. **Gradyan Sönmesi Sorununa Dayanıklıdır:** Büyük giriş değerlerinde gradyan sıfıra yaklaşmadığı için öğrenme süreci hızlanır.
3. **Seyrek Aktivasyon (Sparse Activation):** Negatif değerleri sıfıra eşitleyerek yalnızca belirli nöronları aktif hale getirir. Bu, ağın daha verimli çalışmasını sağlar.

Dezavantajları:

1. **Ölü Nöron Sorunu (Dead Neurons):** Eğer bir nöronun çıktısı sürekli olarak negatifse, bu nöron her zaman sıfır döner ve öğrenme sürecine katkı yapamaz. Bu sorun için Leaky ReLU veya Parametric ReLU gibi türevler geliştirilmiştir.

2. **Patlayan Aktivasyonlar:** Çok büyük giriş değerlerinde aktivasyonlar aşırı büyük olabilir, bu da öğrenme sırasında kararsızlığa yol açabilir.
-

Türev ve Gradyan:

- $x > 0$: Türev 1'dir, bu nedenle gradyan geri yayılım etkili bir şekilde çalışır.
 - $x \leq 0$: Türev 0'dır, bu durum ölü nöron problemini doğurabilir.
-

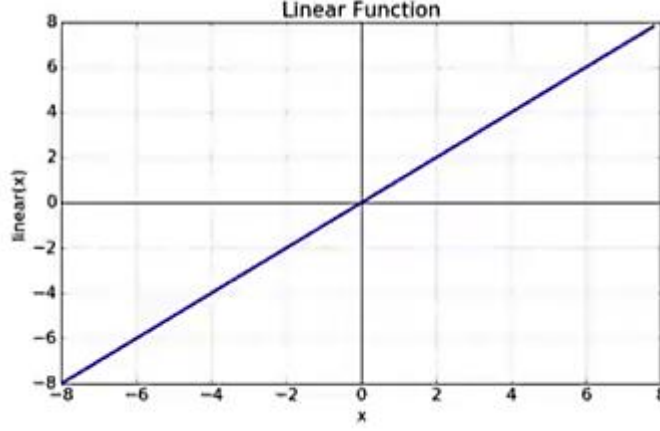
Kullanım Alanları:

- Görüntü işleme uygulamaları (CNN'lerde yaygın).
 - Büyük ve derin sinir ağları.
 - Genellikle varsayılan aktivasyon fonksiyonu olarak kullanılır.
-

Kısaca, **ReLU**, hızı ve etkili öğrenme kapasitesi sayesinde modern derin öğrenme modellerinde **standart** haline gelmiştir. Daha iyi performans ve daha az gradyan sönmesi sorunu sağlar.

LINEER FONKSİYONLAR

Girdiyi herhangi bir dönüşüme uğratmadan doğrusal bir şekilde çıktı üreten en basit aktivasyon fonksiyonlarıdır.



Matematiksel formülü:

$$f(x)=ax+b$$

Burada a ve b sabit değerlerdir ve fonksiyon doğrusal bir grafiğe sahiptir.

Özellikleri:

1. **Çıkış Aralığı:** Fonksiyonun çıktısı $-\infty$ ile $+\infty$ arasında olabilir.
2. **Doğrusallık:** Herhangi bir karmaşıklık veya doğrusal olmayan dönüşüm içermez.

Avantajları:

1. **Basit ve Anlaşılır:** Hesaplaması çok kolaydır.
2. **Bazı Basit Modellerde Kullanışlıdır:** Örneğin, regresyon gibi doğrusal problemlerde.

Dezavantajları:

1. **Doğrusal Karmaşıklık Yetersizliği:** Sinir ağlarında kullanıldığında, ağ ne kadar derin olursa olsun çıktı doğrusal kalır. Bu, ağın doğrusal olmayan ilişkileri öğrenmesini imkânsız hale getirir.
 - Örneğin, birden fazla lineer fonksiyon katmanını ardışık olarak kullansak bile, sonuç yine bir doğrusal fonksiyon olacaktır.
 2. **Aktivasyon Çeşitliliği Yoktur:** Derin öğrenme ağlarının temel amacı, girdiler arasındaki karmaşık, doğrusal olmayan ilişkileri öğrenmektir. Bu nedenle lineer fonksiyonlar bu amaca hizmet etmez.
-

Neden Derin Öğrenmede Kullanılmaz?

Sinir ağlarının temel gücü, **doğrusal olmayan ilişkileri öğrenme yeteneğinden** gelir. Lineer aktivasyon fonksiyonları, ağı bu doğrusal olmayan ilişkilerden mahrum bırakır, bu yüzden genellikle aktivasyon fonksiyonu olarak tercih edilmez. Ancak son katmanda, özellikle regresyon problemlerinde çıktı almak için kullanılabilir.

Nerede Kullanılır?

1. **Son Katman:** Regresyon problemlerinde, sürekli bir çıktı üretmek için son katmanda lineer aktivasyon kullanılır.
 2. **Basit Modeller:** Doğrusal ilişki içeren basit veri kümelerinde.
-

Kısacası, lineer fonksiyonlar temel düzeyde kullanışlı olsa da, derin öğrenmenin **karmaşık ilişkileri öğrenme gücünü** ortaya çıkaramaz. Bu yüzden genellikle yalnızca **son katman** için tercih edilir.