

Yapay Sinir Ağları

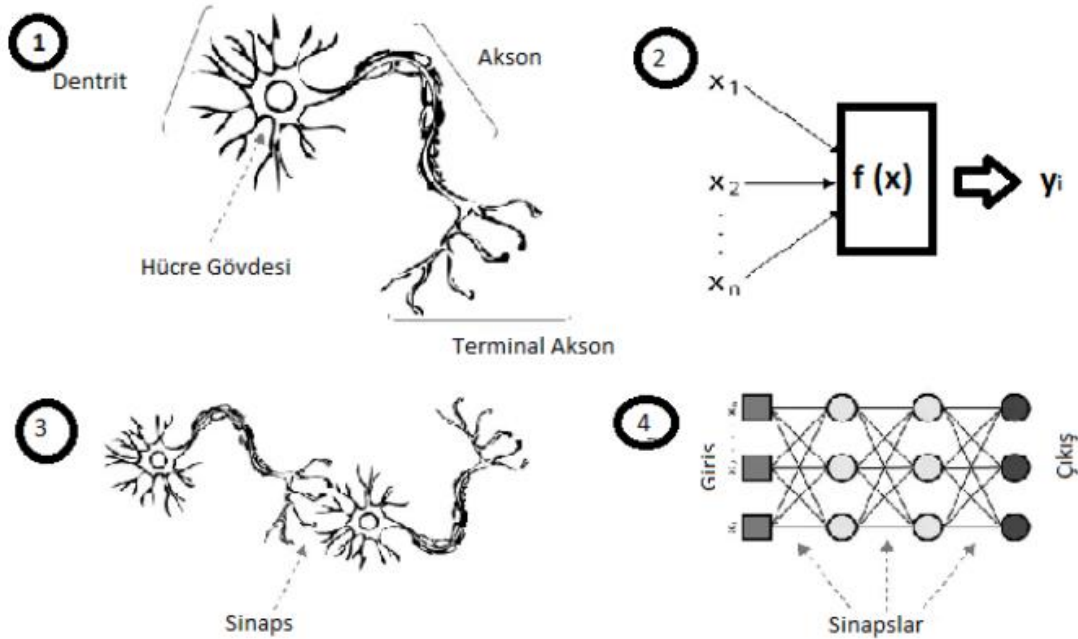
Yapay sinir ağları (YSA), insan beyninin öğrenme yolunu taklit ederek beynin öğrenme, hatırlama, genelleme yapma yolu ile topladığı verilerden yeni veri üretebilme gibi temel işlevlerin gerçekleştirildiği bilgisayar yazılımlarıdır. Yapay sinir ağları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır.

Yapay Sinir Ağlarının Özellikleri:

- Doğrusal Olmama
- Paralel Çalışma
- Öğrenme
- Genelleme
- Hata Toleransı ve Esneklik
- Eksik Verilerle Çalışma
- Çok Sayıda Değişken ve Parametre Kullanma
- Uyarlanabilirlik

Yapay Sinir Ağları Tarihçesi

İlk yapay sinir ağı modeli 1943 yılında bir sinir hekimi olan Warren McCulloch ve bir matematikçi olan Walter Pitts tarafından “Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap (A Logical Calculus of Ideas Immanent in Nervous Activity)” başlıklı makale ile ortaya çıkarılmıştır (URL 1, 2016). YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır.



Şekil 1. Biyolojik sinir hücresi ve yapay sinir ağı (Maltarollo, 2013).

Yapay sinir hücresi

Yapay sinir ağı yapay sinir hücrelerinden oluşur ve bu hücrelere proses elemanı denir. Prosesler beş elemandan oluşur. Girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktısıdır.

Yapay sinir ağı yapısı

Yapay sinir hücreleri bir araya gelerek yapay sinir ağlarını oluştururlar. Ağ yapısı; girdi katmanı, ara katmanlar ve çıktı katmanından oluşur.

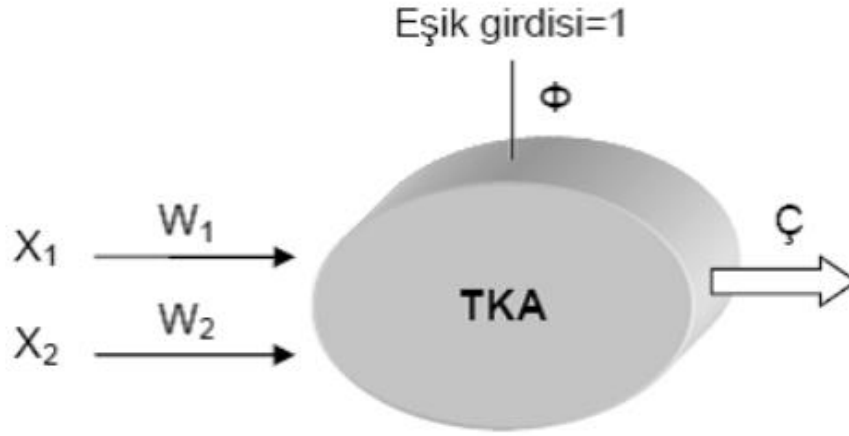
YSA da bir girdi ve bir çıktı seti bulunur. Eğitilmek istenen YSA ya öğretilecek bilgi bir vektör haline çevrilerek verilir. Doğru çıktıyı üretecek şekilde sistem parametreleri ayarlanır.

YSA da öğrenme 2 aşamada yapılır. İlk aşamada sistem eğitilirken üretilen çıktı değerleri verilir. Çıktı değerinin doğruluğuna göre ağırlıklar değiştirilir. Ağırlıklar değiştirildikten sonra sistemin performansını ölçmek için test verisi sisteme sunulur. Test aşamasında ağırlık değerlerinde herhangi bir değişiklik yapılmaz eğitimde kullanılan ağırlıklar kullanılır ve bu karşılaştığı test verileri için çıktılar üretir. Elde edilen çıktıların doğruluk değerlerine göre sistemin performansı değerlendirilir.

1.Tek katmanlı algılayıcılar

Yapay sinir ağlarıyla ilgili geliştirilen tek katmanlı yapılardan biri de perceptrondur (algılayıcı). Bu, şimdiye kadar oluşturulan ilk sinir ağıdır. Giriş katmanında 2 nöron ve çıkış katmanında 1 nöron oluşur. Bu yapı 2 grubu ayırt etmek için basit bir sınıflandırıcı oluşturmaya izin verir. Bir perceptron, ikili sınıflandırıcıların denetimli öğrenimi için bir algoritmadır. Bu algoritma, nöronların eğitim setindeki öğeleri birer birer öğrenmesini ve işlenmesini sağlar.

Tek katmanlı ve çok katmanlı olmak üzere 2 tür algılayıcı vardır. Tek katmanlı algılayıcılar yalnızca doğrusal olarak ayrılabilir desenleri öğrenebilir. Çok katmanlı algılayıcılar iki veya daha fazla katmana sahip ileri beslemeli sinir ağlarıdır ve daha fazla işlem gücüne sahiptir. Perceptron algoritması, doğrusal bir karar sınırı çizmek için giriş sinyallerinin ağırlıklarını öğrenir. Bu, doğrusal olarak ayrılabilir +1 ve -1 sınıfı arasında ayrım yapmanızı sağlar.



Şekil 3. Tek katmanlı algılayıcı modeli (Öztemel, 2006)

Şekilde görüldüğü üzere tek katmanlı algılayıcı yapısında girdi katmanı ve bu girdilerin ağırlıkları, belirlenmiş bir bias değeri ve bunların toplamı da çıktı (y) olarak hesaplanır. Burada y değeri x değerinin ne kadar doğru hesaplandığını gösterir. Çıktı skorunu iyileştirmek için ağırlık (w) ve bias (b) değerleri güncellenerek kullanılır. Derin öğrenme ya da yapay sinir ağlarındaki amaç en iyi skoru verecek olan ağırlık ve bias değerlerini hesaplamaktır.

Tek katmanlı algılayıcılar doğrusal olmayan problemlerin çözümünde yeterli gelmezler ve çok katmanlı algılayıcılar devreye girer.

Temel olarak perceptron yapısı giriş değerleri, giriş değerlerinin ağırlıkları, bias değerleri, net toplam ve aktivasyon fonksiyonlarından oluşmaktadır. Perceptron, tek katmanlı bir sinir ağıdır ve çok katmanlı bir algılayıcı da sinir ağları (Neural Network) olarak adlandırılır.

Perceptron'un çalışma mantığı şu şekildedir.

- Tüm x girdileri, ağırlıkları w ile çarpılır.
- Tüm çarpılan değerleri toplanır ve bunlara ağırlıklı toplam denir.
- Bu ağırlıklı toplama doğru etkinleştirme fonksiyonu uygulanır.

Ağırlıklar, belirli düğümün gücünü gösterir.

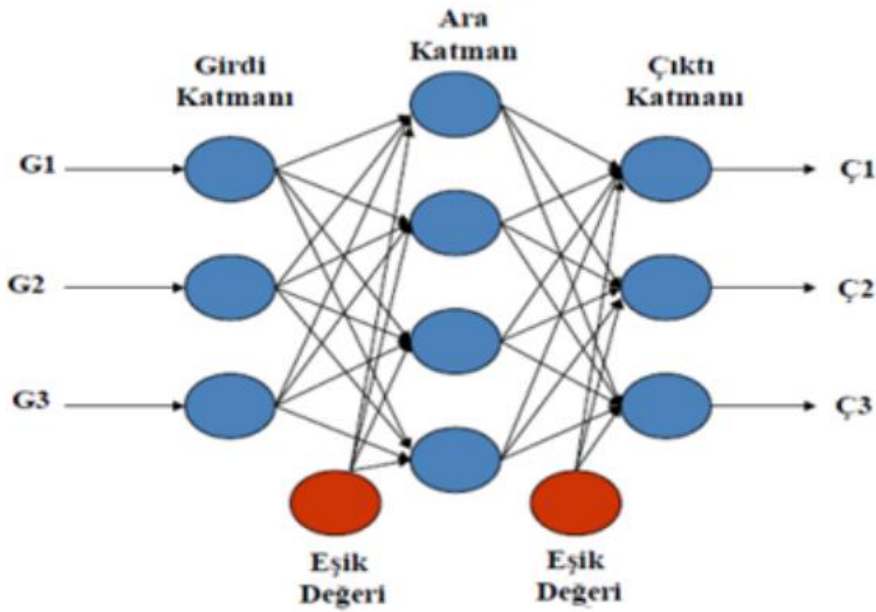
Sapma değeri, etkinleştirme işlevi eğrisini yukarı veya aşağı kaydırmanıza olanak tanır.

Kısaca, aktivasyon fonksiyonları girdi değerlerini (0, 1) veya (-1, 1) gibi gerekli değerler arasında eşlemek için kullanılır. Perceptron genellikle verileri iki bölüme ayırmak için kullanılır. Bu nedenle, doğrusal ikili sınıflandırıcı olarak da bilinir.

Aktivasyon fonksiyonları doğrusal (linear) ve doğrusal olmayan (non-linear) olmak üzere 2 ye ayrılır. Evet veya hayır gibi sinir ağlarının çıktısını belirlemek için kullanılır. Elde edilen değerleri kullanarak aktivasyon fonksiyonuna göre 0 ila 1 veya -1 ila 1 değerleri arasına eşler.

2.Çok katmanlı algılayıcılar

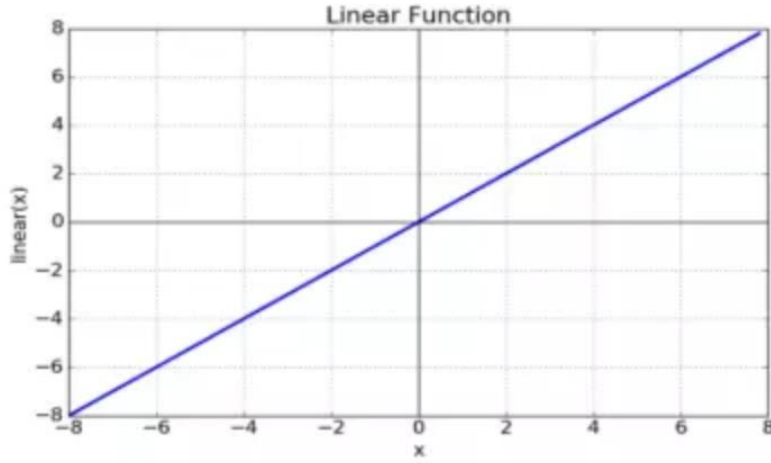
Yapısal olarak doğrusal olmayan aktivasyon fonksiyonu olan birçok nöronun belli bir üstünlük içerisinde bağlandığı yapıya çok katmanlı 31 algılayıcılar denir. Çok katmanlı algılayıcıların ortaya çıkmasında bazı yöntemlerin etkisiz kalmasının rolü vardır.



Şekil 4. Çok katmanlı algılayıcı modeli (Kabalcı, 2014).

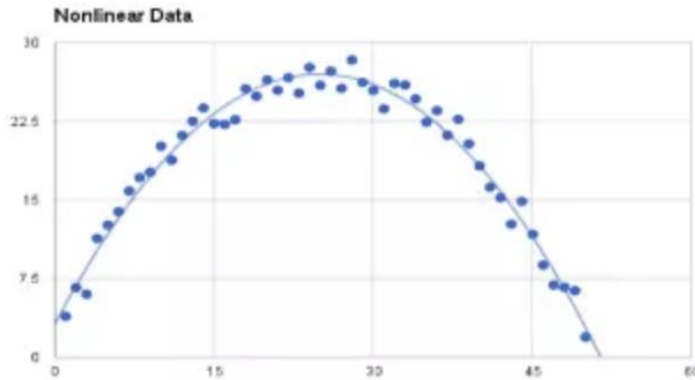
Aktivasyon Fonksiyonları

Doğrusal aktivasyon fonksiyonu



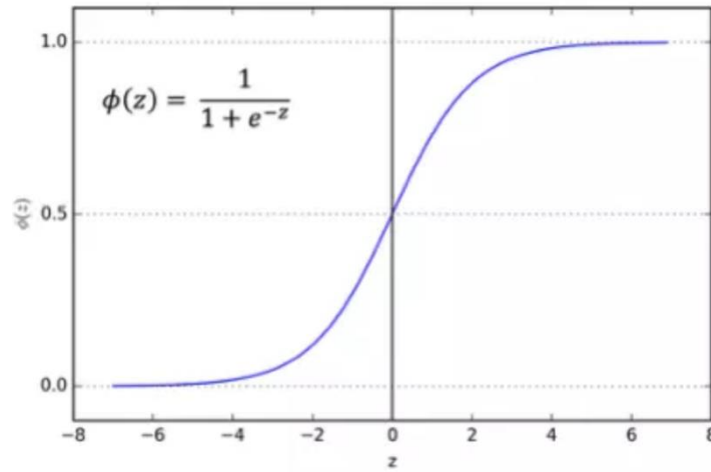
Doğrusal fonksiyon bir çizgi veya doğrusaldır. Bu nedenle, fonksiyonun çıktısı herhangi bir aralık arasında sınırlanmayacaktır. Sinir ağılarıyla beslenen olağan verilerin karmaşıklığında veya çeşitli parametrelerinde kullanılmaz. Türevlenebilir fonksiyon değildir, doğrusal olduğu için ve öğrenme işleminde geriye besleme yapıldığında sabit bir sayı olacağından öğrenme gerçekleşmeyecektir.

Doğrusal olmayan aktivasyon fonksiyonu



Doğrusal olmayan etkinleştirme işlevleri en çok kullanılan etkinleştirme işlevleridir. Doğrusal olmamak, grafiğin Şekildeki gibi görünmesine yardımcı olur. Modelin genelleştirilmesini veya çeşitli verilerle uyum sağlamasını ve çıktılar arasında ayırım yapmasını kolaylaştırır. Doğrusal olmayan aktivasyon fonksiyonları esas olarak aralıklarına veya eğrilerine göre bölünür.

Sigmoid veya lojistik aktivasyon fonksiyonu



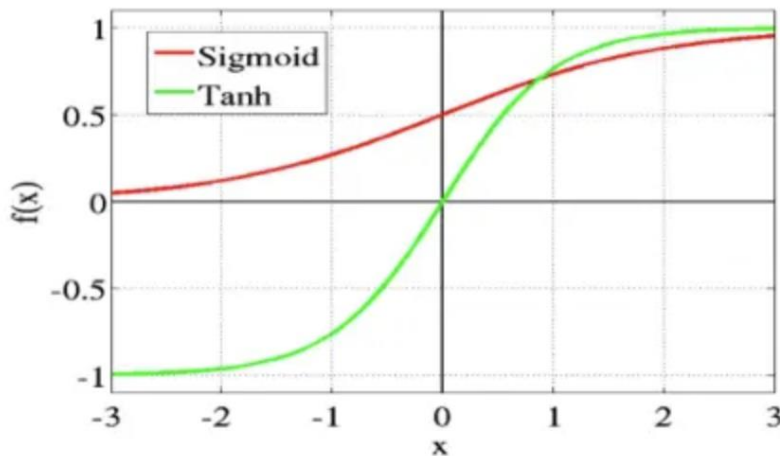
Şekilde görüldüğü gibi sigmoid fonksiyon eğrisi bir S şekline benzer. Sigmoid işlevini kullanmamızın ana nedeni, (0 ile 1) arasında var olmasıdır. Bu nedenle, özellikle olasılığı bir çıktı olarak tahmin etmemiz gereken modeller için kullanılır.

Herhangi bir şeyin olasılığı yalnızca 0 ile 1 aralığı arasında olduğundan, sigmoid doğru bir seçimdir. Fonksiyon türevlenebilir. Yani sigmoid eğrisinin eğimini herhangi iki noktada bulabiliriz.

Lojistik sigmoid işlevi, eğitim sırasında bir sinir ağının sıkışmasına neden olabilir.

Softmax fonksiyonu, çok sınıflı sınıflandırma için kullanılan daha genelleştirilmiş bir lojistik aktivasyon işlevidir.

Tanh or hyperbolic tangent aktivasyon fonksiyonu



Tanh aslında lojistik sigmoid gibidir ancak ondan daha iyidir. Şekil 5'te görüldüğü gibi tanh fonksiyonunun aralığı (-1 ile 1) arasındadır. Bunun

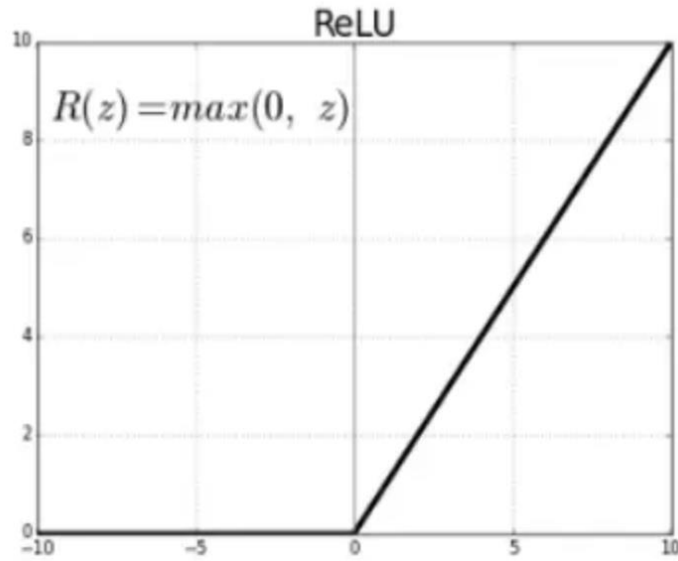
avantajı, negatif girişlerin güçlü bir şekilde negatif olarak eşleştirilmesi ve sıfır girişlerinin tanh grafiğinde sıfıra yakın bir şekilde eşlenmesidir.

Bu fonksiyon da türevlenebilir. Tanh işlevi esas olarak iki sınıf arasında sınıflandırmada kullanılır.

Hem tanh hem de lojistik sigmoid aktivasyon fonksiyonları ileri beslemeli ağlarda kullanılır.

ReLU (rectified linear unit) activation function

ReLU en çok kullanılan aktivasyon fonksiyonudur ve neredeyse tüm evrişimli sinir ağlarında veya derin öğrenmede kullanılmaktadır.



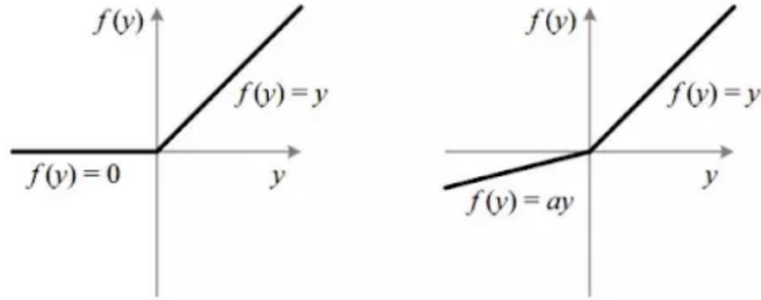
Şekilde üzere z görüldüğü sıfırdan küçük olduğunda $f(z)$ sıfırdır ve z sıfırdan büyük veya eşit olduğunda $f(z)$ z 'ye eşittir. Aralık 0 dan sonsuzadır.

Fonksiyon ve türevi monotondur. Ancak sorun, tüm negatif değerlerin anında sıfırlanmasıdır ve bu da modelin verilere uygun şekilde uyması veya eğitilmesi yeteneğini azaltır. Bu, ReLU aktivasyon fonksiyonuna verilen herhangi bir negatif girdinin, değeri grafikte hemen sıfıra çevirdiği anlamına gelir, bu da negatif değerlerle eğitilmesini engeller.

Leaky ReLU

ReLU deki negatif deęerlerin yok olması sorununu ozmek iin geliřtirilmiřtir. řekil 7’de grldę zere negatif blgelerde trevin 0 olup ęrenmenin devam etmesini saęlar.

Ancak hız bakımından daha hızlı olmasını istiyorsak ReLU daha ok tercih edilmektedir. Genellikle ara katmanlarda ReLU tercih edilir.



Sinir aęları temel yapısı

En yaygın drt sinir aęı:

- **Tam baęlantılı (fully connected):** Tamamen baęlı katmanlar, bir katmandaki her nronu bir sonraki katmandaki her nrona baęlar. Tamamen baęlı katmanlar, standart sinir aęlarından evriřimli sinir aęlarına (CNN) kadar tm farklı sinir aęlarında bulunur.
- **Evriřimli (convolutional)**
- **Ters evriřim (deconvolutional)**
- **Tekrarlayan (recurrent)**