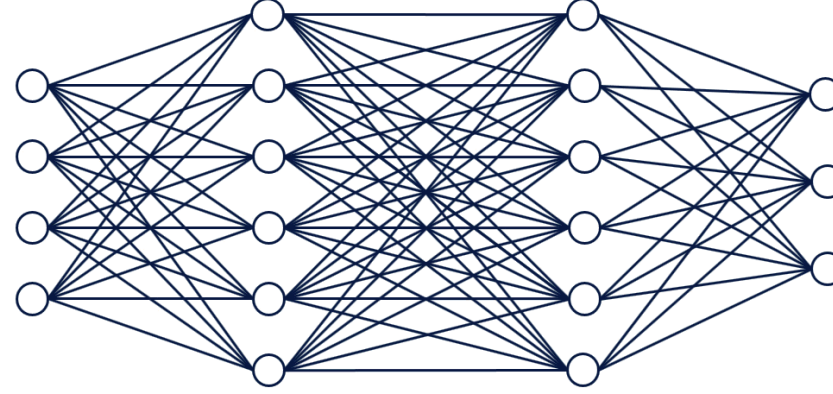


Yapay Sinir Ağları

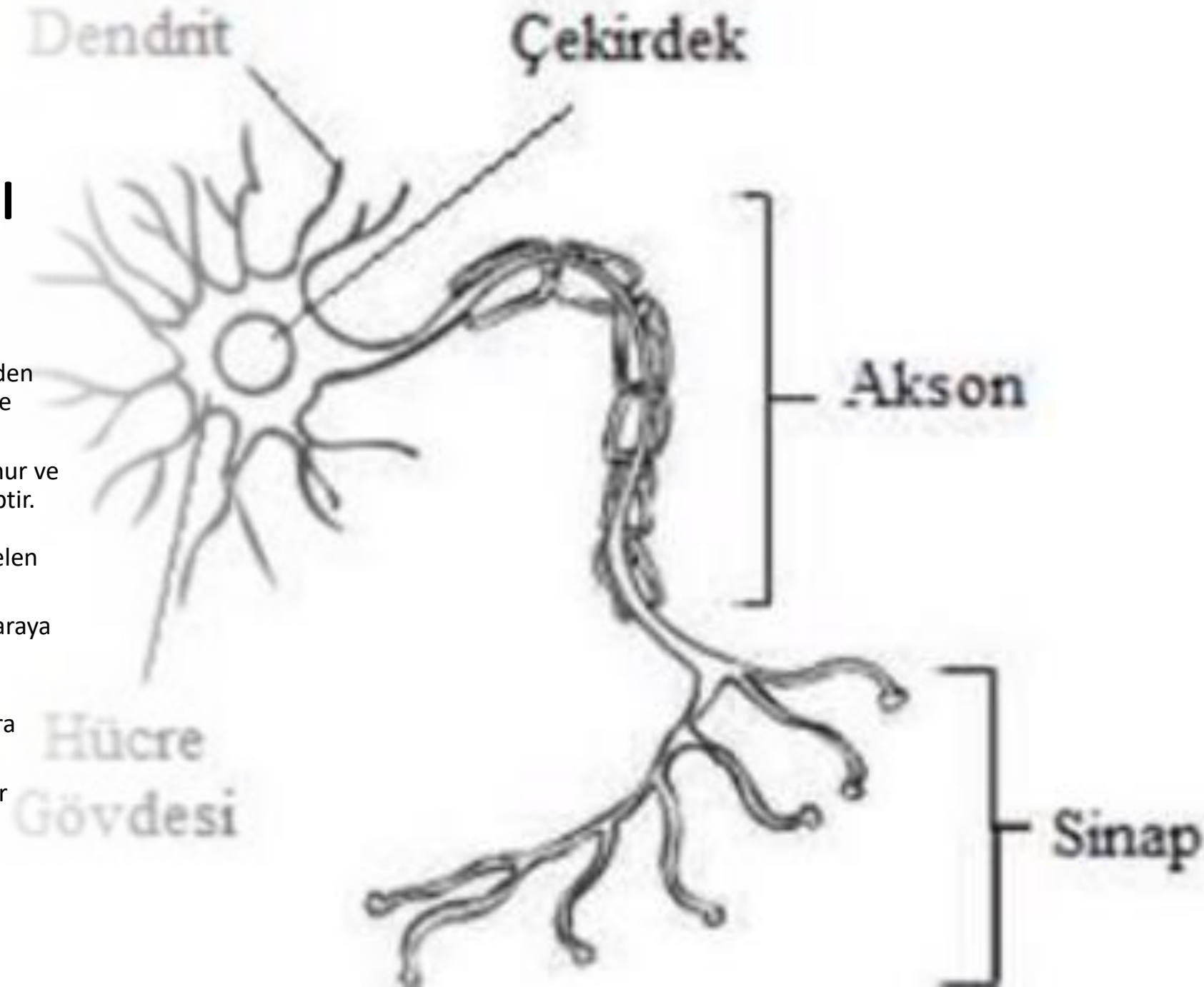


- Yapay sinir ağıları (YSA), insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir.
- Yapay sinir ağıları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır.
- Bu nedendir ki, bu konu üzerindeki çalışmalar ilk olarak beyni oluşturan biyolojik üniteler olan nöronların modellenmesi ve bilgisayar sistemlerinde uygulanması ile başlamış, daha sonraları bilgisayar sistemlerinin gelişimine de paralel olarak bir çok alanda kullanılır hale gelmiştir.



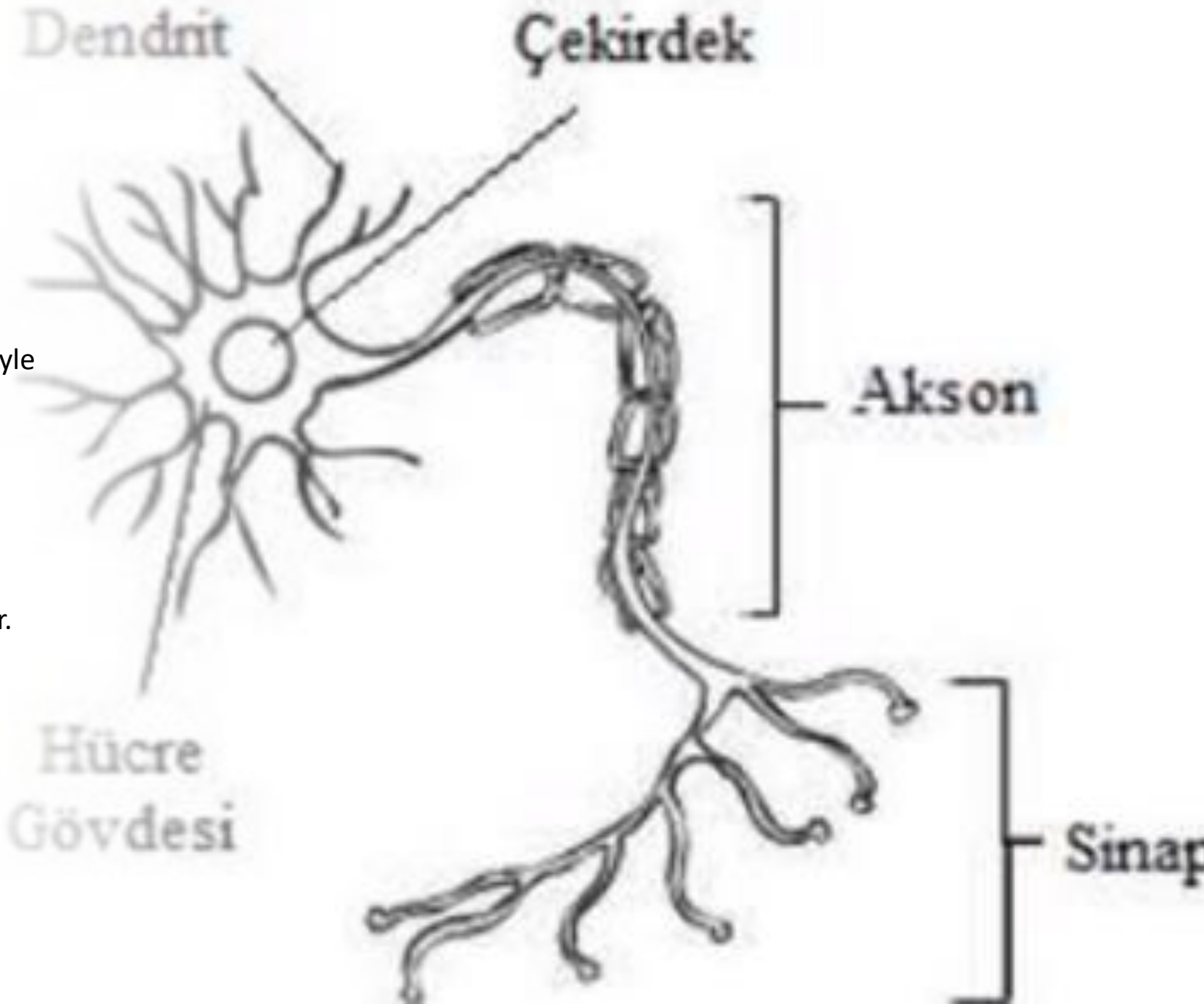
Biyolojik Sinir Hücresinin Yapısı

- Biyolojik sinir sisteminin temel yapı taşı nöronlardır ve nöronlar dört ana bölümden oluşmaktadır; dendrit, akson, çekirdek ve sinaps.
- **Dendritler** sinir hücresinin ucunda bulunur ve ağaç kökü görünümünde bir yapıya sahiptir. Dendritlerin görevi bağlı olduğu diğer nöronlardan veya duyu organlarından gelen sinyalleri çekirdeğe iletmektir.
- **Çekirdek** dendritten gelen sinyalleri bir araya toplar ve aksona iletir.
- **Akson** toplanan bu sinyalleri işleyerek nöronun diğer ucunda bulunan sinapslara gönderir.
- **Sinapslar** ise yeni üretilen sinyalleri diğer nöronlara iletir.



Biyolojik Sinir Hücresinin Yapısı

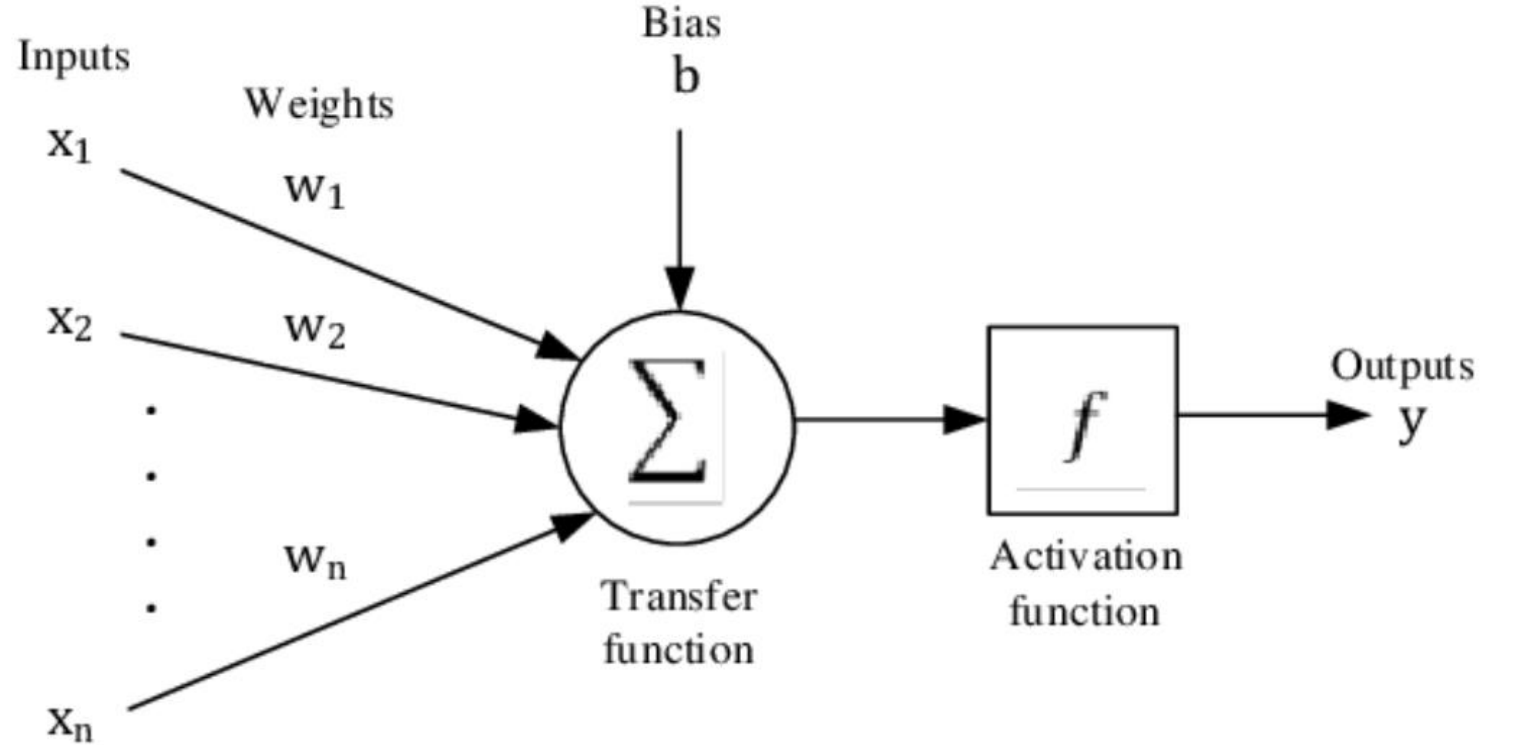
- Bir insanın beyinde yaklaşık olarak 10 milyar sinir hücresi olduğu ve bu nöronların birbirleriyle yaptığı bağlantı sayısının ise 60 trilyon olduğu tahmin edilmektedir.
- Bu sinirler girdi bilgilerini duyu organlarından alırlar.
- Daha sonra alıcı (taşıyıcı) sinirler bu sinyalleri işleyip bir sonraki sinire aktararak sinyalin merkezi sinir sistemine kadar ulaşmasını sağlar.
- Merkezi sinir sistemi bu sinyalleri alıp yorumladıktan sonra tepki sinyallerini üretir.
- Bu sinyaller de tepkilerin oluşacağı organlara tepki sinirleri vasıtasıyla iletilir.
- Bu sayede duyu organlarından gelen bilgilere karşı tepki organlarına uygun işaretler sinir sistemi vasıtasıyla yollar.



Yapay Sinir Hücresinin Yapısı

- Yapay sinir hücreleri de biyolojik sinir hücrelerine benzer yapıdadır.
- Yapay nöronlar da aralarında bağ kurarak yapay sinir ağlarını oluştururlar.
- Aynı biyolojik nöronlarda olduğu gibi yapay nöronların da giriş sinyallerini aldıkları, bu sinyalleri toplayıp işledikleri ve çıktıları ilettikleri bölümleri bulunmaktadır.
- Bir yapay sinir hücresi beş bölümden oluşmaktadır; Girdiler, Ağırlıklar, Toplama Fonksiyonu (Birleştirme Fonksiyonu), Aktivasyon fonksiyonu, Çıktılar

YSA yapısında bilgi nöronlarda tutulur, nöron sayısı belleği temsil eder.

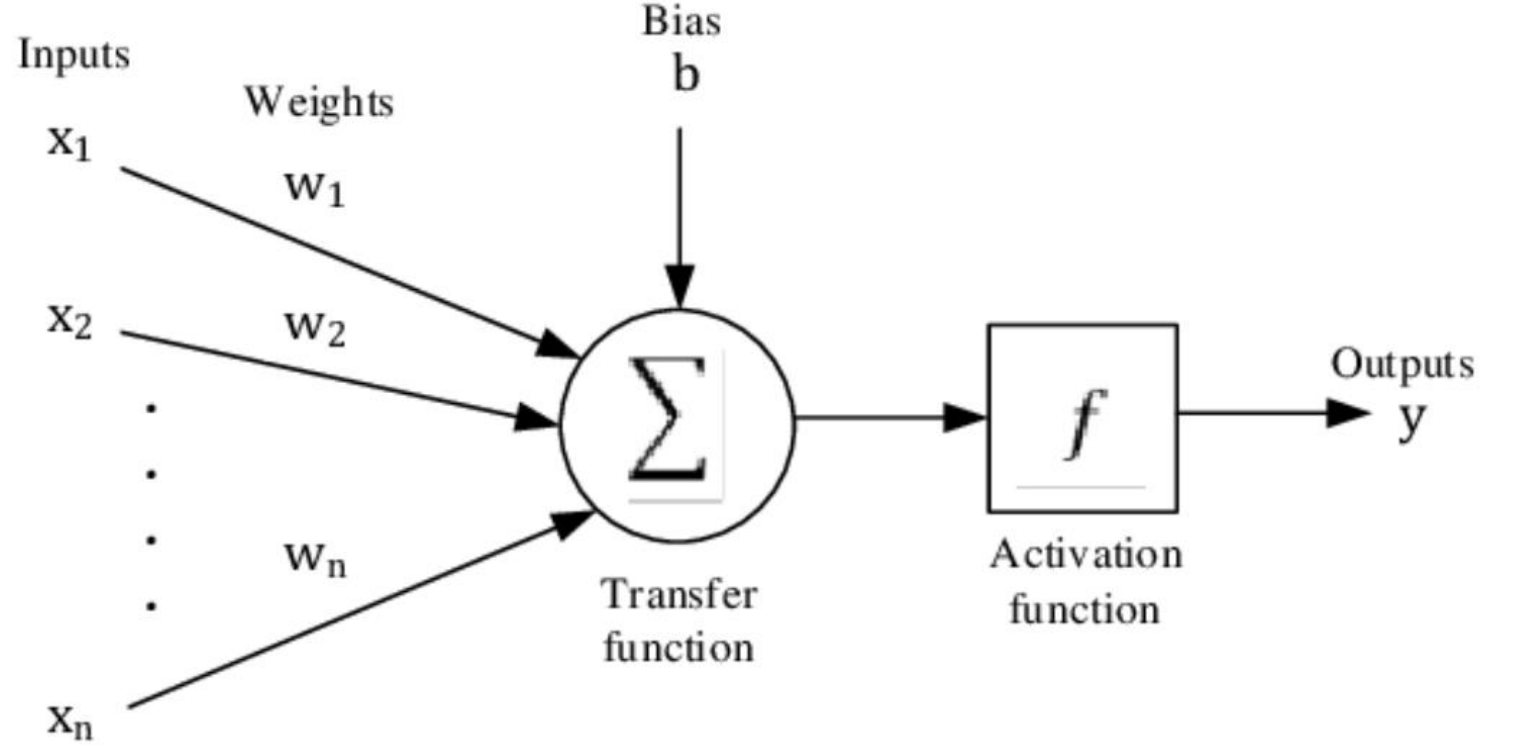


Yapay Sinir Hücresinin Yapısı

Girdiler: Girdiler nöronlara gelen verilerdir. Girdiler yapay sinir hücreğine bir diğer hücreden gelebileceği gibi direkt olarak dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir.

Ağırlıklar: Yapay sinir hücreğine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan girdilerin çıktı üzerinde herhangi bir etkisi olmamaktadır.

YSA yapısında bilgi nöronlarda tutulur, nöron sayısı belleği temsil eder.



Yapay Sinir Hücresinin Yapısı

Toplama Fonksiyonu (Birleştirme Fonksiyonu):

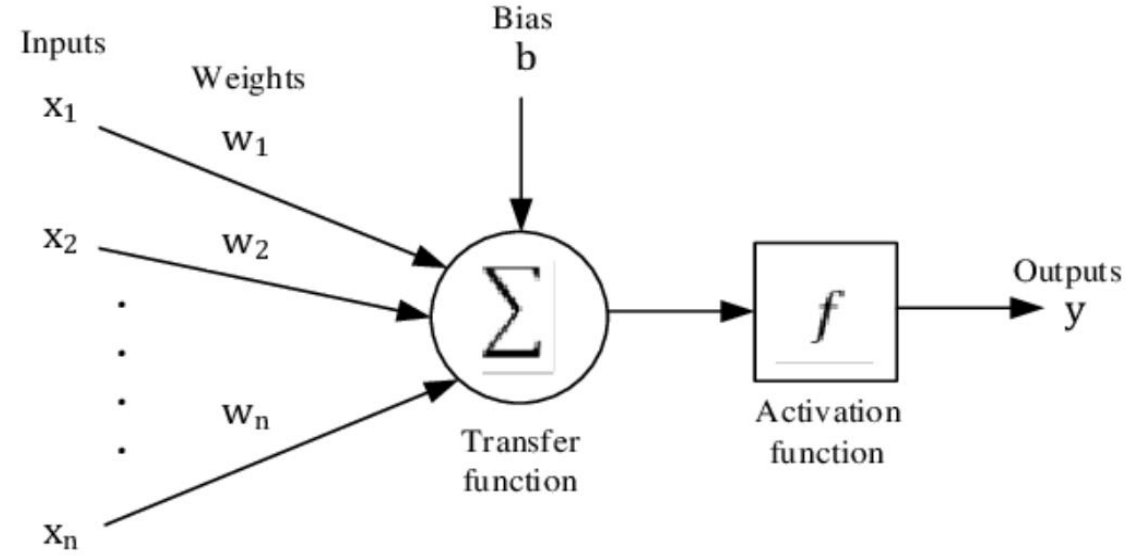
Toplama fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur.

Bazı durumlarda gelen girdilerin değeri dikkate alınırken bazı durumlarda ise gelen girdilerin sayısı önemli olabilmektedir.

Bir problem için en uygun toplama fonksiyonu belirlenirken geliştirilmiş bir yöntem yoktur. Genellikle deneme yanılma yoluyla toplama fonksiyonu belirlenmektedir.

Bazen her hücrenin toplama fonksiyonunun aynı olması gerekmez.

Bu konulara karar vermek tasarımcıya aittir.



Toplam $Net = \sum_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak Net girdi hesaplanır.
Çarpım $Net = \prod_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleriyle çarpılarak Net Girdi Hesaplanır.
Maksimum $Net = \text{Max}(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en büyüğü Net girdi olarak kabul edilir.
Minimum $Net = \text{Min}(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en küçüğü Net girdi olarak kabul edilir.
Çoğunluk $Net = \sum_{i=1}^N \text{Sgn}(X_i * W_i)$	n adet girdi içinden girdilerle ağırlıklar çarpıldıktan sonra pozitif ile negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
Kümülatif Toplam $Net = \text{Net}(\text{eski}) + \sum_{i=1}^N X_i * W_i$	Hücreye gelen bilgiler ağırlıklı olarak toplanır. Daha önce hücreye gelen bilgilere yeni hesaplanan girdi değerleri eklenerek hücrenin net girdisi hesaplanır.

Yapay Sinir Hücresinin Yapısı

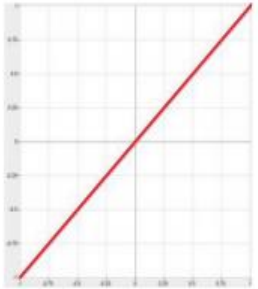
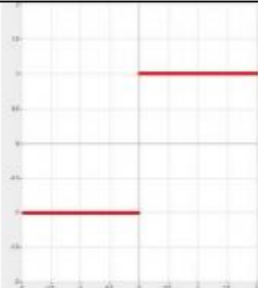


Aktivasyon Fonksiyonu:

Bu fonksiyon hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler.

Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir.

Yapay sinir ağlarının bir özelliği olan “doğrusal olmama” aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir.

Aktivasyon fonksiyonu seçilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığı için hesaplamamanın yavaşlamaması için türevi kolay hesaplanır bir fonksiyon seçilir.

Doğrusal (Lineer) Aktivasyon Fonksiyonu		$F(Net) = A * Net$ (A sabit bir sayı)	Doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon olarak seçilebilir. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım (Step) Aktivasyon Fonksiyonu		$F(Net) = \begin{cases} 1 & \text{if } Net > \text{Eşik Değer} \\ 0 & \text{if } Net \leq \text{Eşik Değer} \end{cases}$	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(Net) = \frac{1}{1 + e^{-Net}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.
Tanjant Hiperbolik Aktivasyon Fonksiyonu		$F(Net) = \frac{e^{Net} + e^{-Net}}{e^{Net} - e^{-Net}}$	Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.

Yapay Sinir Hücresinin Yapısı

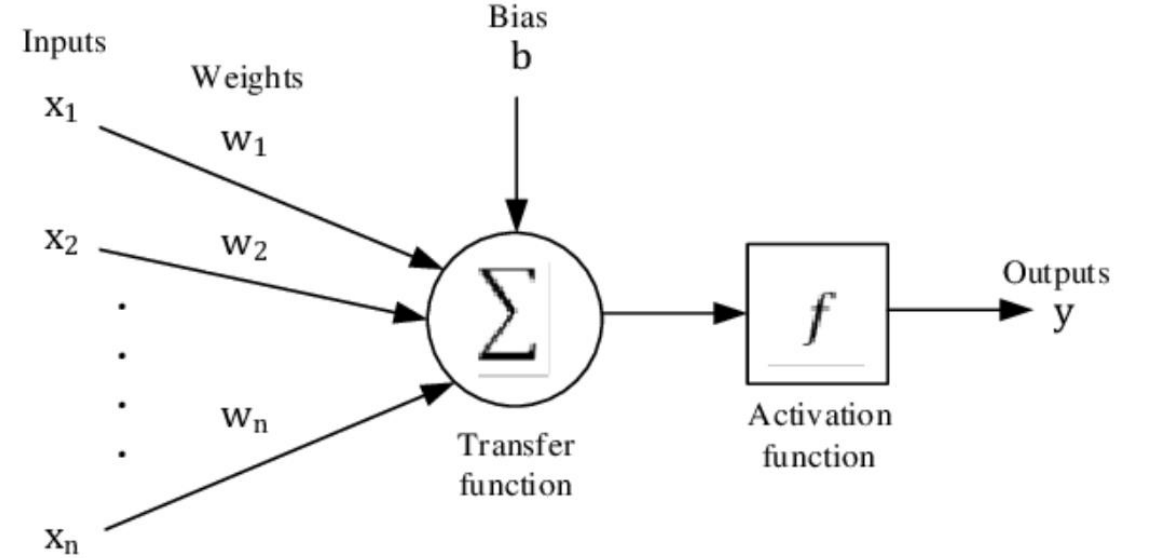
Hücrenin Çıktısı:

- Aktivasyon fonksiyonundan çıkan değer hücrenin çıktı değeri olmaktadır.
- Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilir istenilirse tekrardan ağın içinde kullanılabilir.
- Her hücrenin birden fazla girdisi olmasına rağmen bir tek çıktısı olmaktadır.
- Bu çıktı istenilen sayıda hücreye bağlanabilir.
- Her nöronun çıktısı aşağıda verilmiş olan eşitlik kullanılarak hesaplanır.

$$f(\alpha) = f\left(\sum_{i=1} w_i x_i + b\right)$$

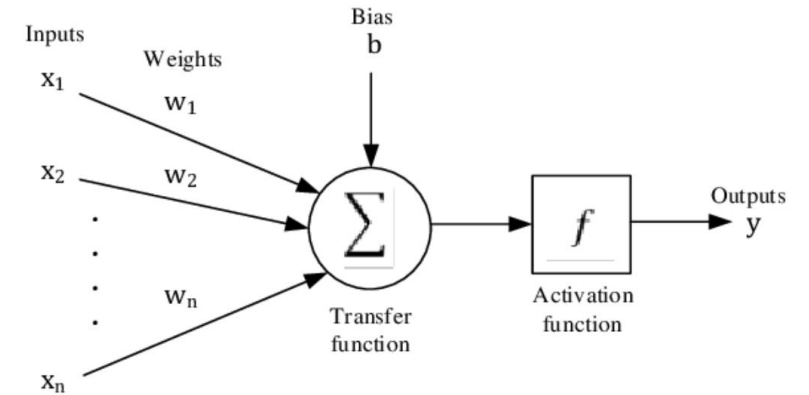
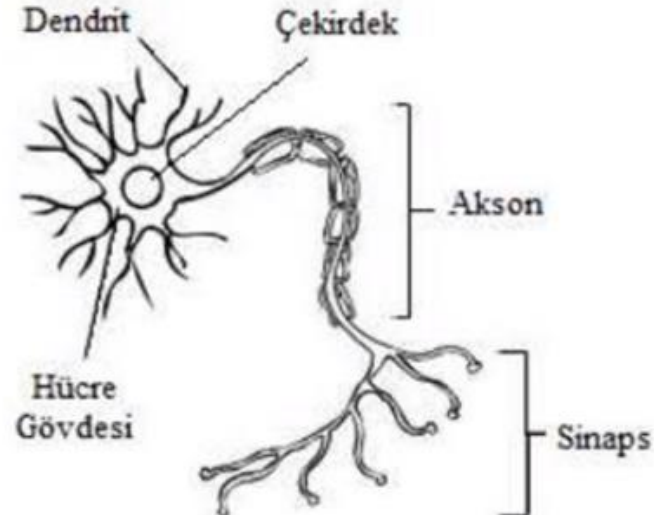
- Denklemden x_i ve w_i sırasıyla nöronun giriş değerlerini ve onların ağırlıklarını temsil eder.
- f ağ boyunca kullanılan doğrusal olmayan aktivasyon fonksiyonudur.
- Bias değeri b , nöronun aktivasyon eşiğini temsil eder.
- Yapay sinir ağlarında amaç, kurulan model için en iyi sonucu üretecek ağırlık ve bias değerlerinin bulunmasıdır. Her epochta ağırlıklar ve bias güncellenir. Bu değerlerin hesaplanma süreci öğrenme olarak adlandırılır.

Bias değeri, aktivasyon fonksiyonunu sağa veya sola ötelenmesini (shift) sağlar.



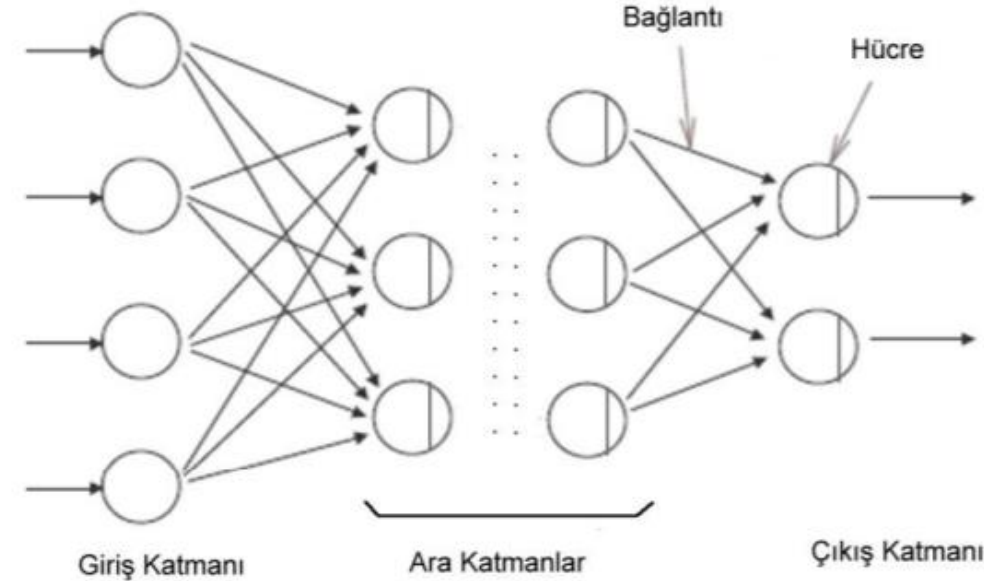
Biyolojik ve Yapay Sinir Hücreleri

Biyolojik Sinir Hücresi	Yapay Sinir Hücresi
Akson	Çıktı
Dendrit	Toplama Fonksiyonu
Çekirdek	Aktivasyon Fonksiyonu
Sinaps	Ağırlıklar



Yapay Sinir Ağı Katmanları

- Yapay sinir ağı üç ana katmanda incelenir; **Giriş Katmanı**, **Ara (Gizli) Katmanlar** ve **Çıkış Katmanı**.
- **Giriş Katmanı**: Yapay sinir ağına dış dünyadan girdilerin geldiği katmandır.
- **Ara (Gizli) Katman(lar)**: Giriş katmanından çıkan bilgiler bu katmana gelir. Ara katman sayısı ağdan ağa değişebilir. Bazı yapay sinir ağlarında ara katman bulunmadığı gibi bazı yapay sinir ağlarında ise birden fazla ara katman bulunmaktadır. Ara katmanlardaki nöron sayıları giriş ve çıkış sayısından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki hücre sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlar.
- **Çıkış Katmanı**: Ara katmanlardan gelen bilgileri işleyerek ağın çıktılarını üreten katmandır. Bu katmanda üretilen çıktılar dış dünyaya gönderilir. Geri beslemeli ağlarda bu katmanda üretilen çıktı kullanılarak ağın yeni ağırlık değerleri hesaplanır.



Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağları işleyiş olarak benzer olmalarına rağmen herhangi bir tasarım ve işleyiş standardı bulunmamaktadır. Nöron dizilimlerine, nöronların ağırlıklarını düzenleme için yapılan hesaplamaların türüne ve zamanına göre yapay sinir ağlarını üç ayrı dalda inceleyebiliriz.

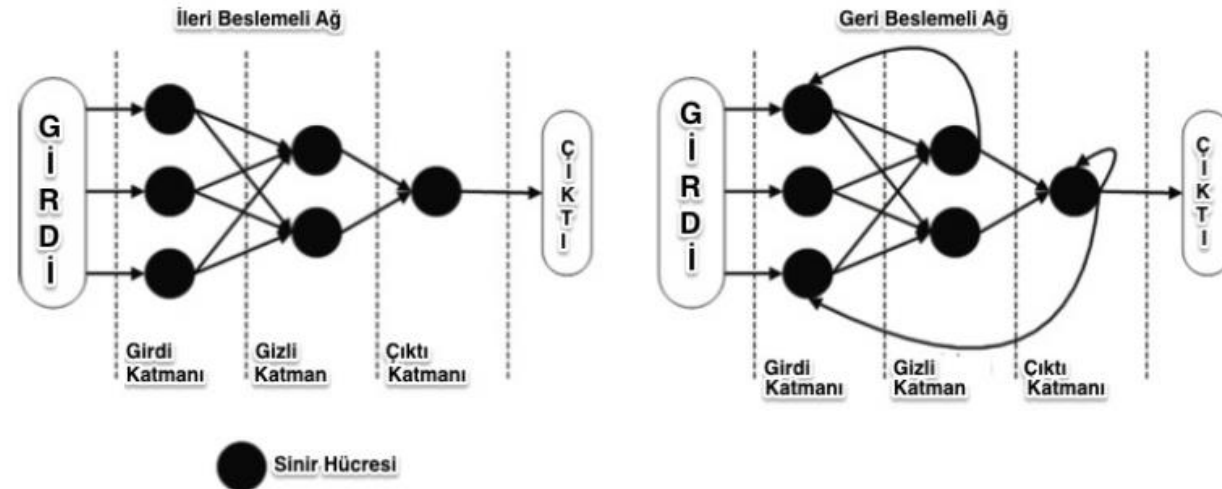
- **Yapılarına Göre Yapay Sinir Ağları**
- **Öğrenme Algoritmalarına Göre Yapay Sinir Ağları**
- **Öğrenme Zamanına Göre Yapay Sinir Ağları**

Yapılarına Göre Yapay Sinir Ağları

Yapay sinir ağları içerdiği nöronların birbirine bağlantı şekline göre ileri ve geri beslemeli olarak ikiye ayrılır.

İleri Beslemeli Ağlar: İleri beslemeli ağlarda nöronlar girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katmanlara bağ bulunmaktadır. Yapay sinir ağına gelen bilgiler giriş katmanına daha sonra sırasıyla ara katmanlardan ve çıkış katmanından işlenerek geçer ve daha sonra dış dünyaya çıkar.

Geri Beslemeli Yapay Sinir Ağları: Geri beslemeli yapay sinir ağlarında ileri beslemeli olanların aksine bir hücrenin çıktısı sadece kendinden sonra gelen hücrenin katmanına girdi olarak verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir hücreye de girdi olarak bağlanabilir.



Öğrenme Algoritmalarına Göre Yapay Sinir Ağları

Yapay sinir ağlarının verilen girdilere göre çıktı üretebilmesinin yolu ağın öğrenebilmesidir.

Bu öğrenme işleminin de birden fazla yöntemi vardır.

Yapay sinir ağları öğrenme algoritmalarına göre danışmanlı (denetimli), danışmansız (denetimsiz) ve takviyeli öğrenme olarak üçe ayrılır.

Danışmanlı Öğrenme:

Danışmanlı öğrenme sırasında ağa verilen giriş değerleri için çıktı değerleri de verilir.

Ağ verilen girdiler için istenen çıkışları oluşturabilmek için kendi ağırlıklarını günceller.

Ağın çıktıları ile beklenen çıktılar arasındaki hata hesaplanarak ağın yeni ağırlıkları bu hata payına göre düzenlenir.

Hata payı hesaplanırken ağın bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her hücreye düşen hata payı bulunur.

Daha sonra her hücrenin kendine gelen ağırlıkları günceller.

Danışmansız Öğrenme:

Danışmansız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez.

Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur.

Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar

Destekleyici Öğrenme:

Bu öğrenme yaklaşımında ağın her iterasyonu sonucunda elde ettiği sonucun iyi veya kötü olup olmadığına dair bir bilgi verilir.

Ağ bu bilgilere göre kendini yeniden düzenler.

Bu sayede ağ herhangi bir girdi dizisiyle hem öğrenerek hem de sonuç çıkararak işlemeye devam eder.

Örneğin satranç oynayan bir yapay sinir ağı yaptığı hamlenin iyi veya kötü olduğunu anlık olarak ayırt edememesine rağmen yine de hamleyi yapar. Eğer oyun sonuna geldiğinde program oyunu kazandıysa yaptığı hamlelerin iyi olduğunu varsayacaktır ve bundan sonraki oyunlarında benzer hamleleri iyi olarak değerlendirerek oynayacaktır.

Öğrenme Zamanına Göre Yapay Sinir Ağları

Yapay sinir ağları öğrenme zamanına göre de statik ve dinamik öğrenme olarak ikiye ayrılır.

Statik Öğrenme: Statik öğrenme kuralıyla çalışan yapay sinir ağları kullanılmadan önce eğitilmektedir. Eğitim tamamlandıktan sonra ağ istenilen şekilde kullanılabilir. Ancak bu kullanım sırasında ağın üzerindeki ağırlıklarda herhangi bir değişiklik olmaz.

Dinamik Öğrenme: Dinamik öğrenme kuralı ise yapay sinir ağlarının çalıştığı süre boyunca öğrenmesini öngörerek tasarlanmıştır. Yapay sinir eğitim aşaması bittikten sonra da daha sonraki kullanımlarında çıkışların onaylanmasına göre ağırlıklarını değiştirerek çalışmaya devam eder.

Yapay Sinir Ağlarının Eğitilmesi

- İnsan yaşadıkça beyin gelişir ve tecrübelenir, olaylar karşısında nasıl tepki göstereceğini öğrenir. Fakat hiç karşılaşmadığı bir olay karşısında yine tecrübesiz kalabilir.
- Yapay sinir ağlarının öğrenme sürecinde de dış ortamdan girişler alınır, aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir.
- Bu çıkış yine tecrübeyle verilen çıkışla karşılaştırılarak hata bulunur. Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır.
- Bu çalışma süresince yenilenen yapay sinir ağının ağırlıklarıdır. Ağırlıklar her bir çevrimde yenilenerek amaca ulaşılmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir.
- Eğer yapay sinir ağı verilen giriş-çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır.
- Ağırlıkların sürekli yenilenerek istenilen sonuca ulaşılan kadar geçen zamana öğrenme adı verilir.
- Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkış yaklaşımı incelenir.
- Eğer yeni verilen örneklerle de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir.
- Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş, *ezberlemiş* demektir.
- Genelde eldeki örneklerin %80'i ağı verilip ağı eğitilir. Daha sonra kalan %20'lik kısım verilip ağın davranışı incelenir. Böylece ağı testi yapılmış olur.

Yapay Sinir Ağlarının Eğitilmesi

1. Örneklerin toplanması: Ağın öğrenmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağın eğitilmesi için örnekler toplandığı gibi (eğitim seti) ağın test edilmesi için de örneklerin (test seti) toplanması gerekmektedir.

Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağın performansı ölçülür.

Hiç görmediği örnekler karşısındaki başarısı ağın iyi öğrenip öğrenmediğini ortaya koyar.

2. Ağın topolojik yapısının belirlenmesi: Öğrenilmesi istenen olay için oluşturulacak olan ağın topolojik yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane proses eleman kaç tane çıktı eleman olması gerektiği bu adımda belirlenmektedir.

3. Öğrenme parametrelerinin belirlenmesi: Ağın öğrenme katsayısı, proses elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.

Yapay Sinir Ağlarının Eğitilmesi

4. Ağırlıkların başlangıç değerlerinin atanması: Proses elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin atanması yapılır. Başlangıç genellikle rasgele değerler atanır. Daha sonra ağ uygun değerleri öğrenme sırasında kendisi belirler.

5. Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi: Ağın öğrenmeye başlaması ve Öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler belirli bir düzeneğe göre gösterilir.

6. Öğrenme sırasında ileri hesaplamaların yapılması: Sunulan girdi için ağın çıktı değerleri hesaplanır.

7. Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması: Ağın ürettiği hata değerleri bu adımda hesaplanır.

8. Ağırlıkların değiştirilmesi: Geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.

9. Öğrenmenin tamamlanması: İleri beslemeli sinir ağı öğrenmeyi tamamlayıncaya, yani gerçekleşen ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar devam eder.

Yapay Sinir Ağlarının Eğitilmesi

- Ağın kendisine gösterilen girdi örneği için beklenen çıktıyı üretmesini sağlayacak ağırlık değerleri başlangıçta rastgele atanmakta ve ağa örnekler gösterildikçe ağırlıklar değiştirilerek istenen değerlere ulaşması sağlanmaktadır.
- İstenen ağırlık değerlerinin ne olduğu bilinmemektedir. Bu nedenle YSA'nın davranışlarını yorumlamak ve açıklamak mümkün olmaz.
- Bazı durumlarda ağın takıldığı yer hata düzeyinin üstünde kalabilir. Bu durumda ağın olayı öğrenmesi için bazı değişiklikler yapılarak yeniden eğitilmesi gerekir.
 - ☐ Başlangıç değerlerinde değişiklik yapılabilir.
 - ☐ Ağ topolojisinde değişiklik yapılabilir.
 - ☐ Ağın parametrelerinde değişiklik yapılabilir.
 - ☐ Ağa sunulan verilerin gösterimi ve örneklerin formülasyonu değiştirilerek yeni örnek seti oluşturulabilir.
 - ☐ Öğrenme setindeki örneklerin sayısı artırılabilir veya azaltılabilir.

Yapay Sinir Ağlarının Eğitilmesi

- İleri beslemeli sinir ağının yerel sonuçlara takılıp kalmaması için momentum katsayısı geliştirilmiştir.
- Ağların eğitilmesinde diğer önemli bir sorun ise öğrenme süresinin çok uzun olmasıdır.
- Ağırlık değerleri başlangıçta büyük değerler olması durumunda ağın lokal sonuçlara düşmesi ve bir lokal sonuçtan diğerine sıçramasına sebep olmaktadır.
- Eğer ağırlıklar küçük aralıkta seçilirse o zaman da ağırlıkların doğru değerleri bulması uzun zamanlar almaktadır.
- Bazı problemlerin çözümü sadece 200 iterasyon sürerken bazıları 5-10 milyon iterasyon sürmektedir.
- Ağın öğrenmesinin gösterilmesinin en güzel yolu hata grafiğini çizmektir. Her iterasyonda oluşan hatanın grafiği çizilirse hatanın zaman içinde düştüğü gözlenebilir. Belirli bir iterasyondan sonra hatanın daha fazla azalmayacağı görülür. Bu ağın öğrenmesinin durduğu ve daha iyi bir sonuç bulunamayacağı anlamına gelir. Eğer elde edilen çözüm kabul edilemez ise o zaman ağ yerel bir çözüme takılmış demektir.

Katsayıların Belirlenmesi

- İleri beslemeli sinir ağında bağlantıların ağırlık değerlerinin başlangıçta belirlenmesi ağı performansı yakından ilgilendirmektedir.
- Genel olarak ağırlıklar belirli aralıklarda atanmaktadır. Bu aralık eğer büyük tutulursa yerel çözümler arasında sürekli dolaştığı küçük olması durumunda ise öğrenmenin geç gerçekleştiği görülmüştür.
- Bu değerlerin atanması için henüz standart bir yöntem yoktur. Tecrübeler 0.1 ile 1.0 arasındaki değerlerin başarılı sonuçlar ürettiğini göstermektedir. Fakat bu tamamen öğrenilmesi istenen problemin niteliğine bağlıdır.



Katsayıların Belirlenmesi

- Başlangıç değerleri kadar öğrenme ve momentum katsayılarının belirlenmesi de ağıın öğrenme performansı ile yakından ilgilidir.
- Öğrenme katsayısı ağırlıkların değışim miktarını belirlemektedir. Eğer büyük değler seçilirse o zaman yerel çözümler arasında ağıın dolaşması ve osilasyon yaşaması söz konusu olmaktadır.
- Küçük değler seçilmesi ise öğrenme zamanını artırmaktadır.
- Tecrübeler genellikle 0.2-0.4 arasındaki değlerin kullanıldığını göstermiştir. Bazı uygulamalar öğrenme katsayısının 0.6 değerini aldığı zaman en başarılı sonuçları verdiği göstermiştir. Bu durum tamamen probleme bağıdır.



Katsayıların Belirlenmesi

- Benzer şekilde momentum katsayısı da öğrenmenin performansını etkiler.
- Momentum katsayısı bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesidir.
- Bu özellikle yerel çözümlere takılan ağırların sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacıyla ile önerilmiştir.
- Bu değerlerin küçük olması yerel çözümlerden kurtulmayı zorlaştırabilir.
- Çok büyük değerler ise çözüme ulaşmada sorunlar yaşatabilir.
- Tecrübeler momentum katsayısının 0.6-0.8 arasında seçilmesinin uygun olacağını göstermiştir. Problemin niteliğine göre kullanıcının almasında fayda vardır.

