İçindekiler

[Giriş 3](#_Toc41820525)

[Tarihçesi 3](#_Toc41820526)

[Yapay Sinir Ağları (Artificial Neural Networks) 5](#_Toc41820527)

[1.1 Yapay Zekâ Nedir? 5](#_Toc41820528)

[1.2 Yapay Zekâ Teknikleri 5](#_Toc41820529)

[1.3 Yapay Sinir Ağları Nedir? 6](#_Toc41820530)

[1.3.1 YSA’ların Avantajları 6](#_Toc41820531)

[1.3.2 YSA’ların Dezavantajları 7](#_Toc41820532)

[1.4 Geleneksel Algoritmalar ile YSA'ların Karşılaştırılması 8](#_Toc41820533)

[1.5 Yapay Sinir Ağları Kullanım Alanları 8](#_Toc41820534)

[1.6 Yapay Sinir Ağlarının Sınıflandırılması 10](#_Toc41820535)

[1.6.1 Yapay Sinir Ağlarının Yapıları 10](#_Toc41820536)

[1.6.2 Öğrenme Yöntemlerine Göre Yapay Sinir Ağları 11](#_Toc41820537)

[1.7 Yapay Sinir Ağlarının Yapısı 11](#_Toc41820538)

[1.7.1 Biyolojik Sinir Hücresinin Yapısı 11](#_Toc41820539)

[1.7.2 Yapay Sinir Hücresinin Yapısı 12](#_Toc41820540)

[1.8 Yapay Sinir Ağlarının Eğitilmesi 17](#_Toc41820541)

[Bir Örnek Üzerinde Matematiksel Açıklama 18](#_Toc41820542)

[Kompleks Örnek 22](#_Toc41820543)

[Kaynakça 25](#_Toc41820544)

# Giriş

Bilgisayarın icadından beri, insanlar bilgisayarların asla yapamayacak oldukları şeyler hakkında konuşmuşlardır. İster satrançta grandmaster yenmek olsun, ister bir yarışma programını kazanmak olsun, bu tahminler genellikle yanlış çıkmıştı. Eğer bilgisayarların nasıl çalıştığını bilseydiniz, fiilen erişilmesi imkânsız olacak hedefler bulunduğunu bilirdiniz. İnsanların duygularını yüz ifadelerinden tanımak. Geniş bir el yazısı çeşitliliğini okumak. Konuşulan dildeki kelimeleri düzgün şekilde belirlemek. Yoğun trafikte yarı bağımsız olarak araç sürmek.

Bilgisayarlar şimdi tüm bu şeyleri ve bir hayli daha fazlasını yapabilmeye başlıyorlar. Olumsuz düşünen insanlar, dijital bilgisayarların gerçek yetenekleri hakkında gerçekten sadece çok mu küçümserlerdi? Bilim insanları bu heybetli zorlukları çözmek için, beynin yapısına dayalı olan, tamamen yeni bir bilgisayar türü ile çıkagelmek zorunda kaldılar. Bu yapay sinir ağları (YSA) sadece sıradan bir dijital bilgisayar üzerinde çalışan bir taklit olarak bulunuyor, fakat bu taklidin içinde gerçekleşen şey, klasik bilişimden temel olarak çok farklı. [1]

Yapay zekâ, makinelere insanlar gibi düşünme, karar verme, karşılaştırma, analiz etme gibi birtakım fonksiyonların kazandırılmasıdır. Fikir babası ise, “Makineler düşünebilir mi?” sorusunu ortaya atan ünlü İngiliz matematikçi ve bilgisayar bilimci Alan Mathison Turing’dir. Meşhur Turing Testi ’nin sahibi, aynı zamanda 2. Dünya Savaşında sırasında Arthur Scherbius’un Enigma’sını kıran kişidir. [2]

# Tarihçesi

İnsanoğlu yüzyıllardır cansız varlıkları canlandırmayı ve harekete geçirmeyi düşünmüştür. 4000 yıl önceden başlayan veri depolama isteği ile birlikte, yapay zekâ ve yapay sinir ağlarında asıl gelişme son elli yılda gerçekleşmiştir.

Modern yapay zekânın başlangıcının izlerini klasik filozofların insan düşünce sistematiğini simgesel sistem olarak tanımlama girişiminde görebiliriz. Fakat yapay zekâ alanı 1956’ya kadar resmi olarak oluşturulamamıştır. 1956’da Hannover, New Hampshire, Dartmouth College ’da yapılan bir konferansta ‘’yapay zekâ’’ terimi ilk defa ortaya atıldı [3].

Yapay zekâ varlığının oluşturulmasını başarmak çok da kolay değildi. 1974-1980 arasında ‘’AI winter’’ yani ‘’yapay zekâ kışı’’ olarak bilinen dönemde, yapay zekâ geliştirme sürecini eleştiren birçok rapor yayımlandı. Bu raporlardan sonra, yapay zekâ alanına devletin 24 desteği ve ilgisi azaldı. 1980’lerde yapay zekâ alanı, İngiliz devletinin Japonlarla yarışmak için bu alanı tekrar fonlamasıyla canlandı. [3]

‘Yapay zekâ’ kavramının ortaya çıkışını ve dönem dönem ayrıca kronolojik olarak tarihini ele alalım:

Tarih Öncesi Dönem: Binlerce yıl önce Yunan mitolojisinde rüzgâr tanrısı zannedilen Daedelus ’un “yapay-insan” teşebbüsü ilktir. Tarih öncesi dönemden son yüzyıla kadar robotlarla ilgili farklı çalışmalara rastlanabilmektedir. Örneğin Osmanlı sarayı için geliştirilen Otomatlardan biri de 1769 Yılında Baron Von Kempelen tarafından yapılan satranç oynayan adamdı. Bu otomat Viyana ve Moskova fuarlarında sergilenmişti.

Karanlık Dönem (1965-1970): Bu dönemde çok az bir gelişim elde edilebilmiştir. Bilgisayar uzmanları düşünen bir mekanizma geliştirerek, sadece verileri yükleyerek akıllı bilgisayarlar yapmayı umdular. Sonuç olarak bir bekleme dönemi oldu.

Rönesans Dönemi (1970-1975): Çok hızla artacak gelişmelerin önünün açıldığı dönem olmuştur. Yapay zekâcılar hastalık teşhisi gibi sistemler geliştirdiler. Bugünkü açılımların temelleri oluştu.

Ortaklık Dönemi (1975-1980): Yapay zekâ araştırmacıları, dil ve psikoloji gibi diğer bilim dallarından da yararlanmaya başladılar.

Girişimcilik Dönemi (1980-?): Yapay zekâ laboratuvarlarının dışına çıkarılarak, gerçek dünyanın ihtiyaçlarına göre çok daha karmaşık uygulamalarla düşünülmüştür. Halen devam eden bir dönemdir. [4][3]

# Yapay Sinir Ağları (Artificial Neural Networks)

## 1.1 Yapay Zekâ Nedir?

Yapay zekâ, bir bilgisayarın ya da bilgisayar destekli bir makinenin, genellikle insana özgü nitelikler, çözüm yolu bulma, anlama, bir mana çıkartma, genelleme ve geçmişteki deneyimlerinden öğrenme gibi yüksek mantık süreçlere ilişkin görevleri yerine getirme yeteneği olarak bilim dünyasında tanımlanmıştır [5].

Slage'ye göre yapay zekâ; sezgisel programlama temelinde olan bir yaklaşımdır [6]. Popov’a göre yapay zekâ; insanların yaptıklarını bilgisayarlara yaptırabilme çalışmasıdır [7]. Axe göre ise yapay zekâ; akıllı programları hedefleyen bir bilimdir [8]. Genesereth ve Nilsson'a göre yapay zekâ akıllı davranış üzerine bir çalışmadır. Ana hedefi, doğadaki varlıkların akıllı davranışlarını yapay olarak üretmeyi amaçlayan bir kuramın oluşturulmasıdır. [6]

Görüldüğü gibi her cevap kendi içinde doğru olmasına rağmen, farklılıklarda içermektedir. Şu ana kadar yapılmayanı yapmak demek yeni kavramların ortaya çıkması ve zekânın kullanımı ile mümkün olabilir, bunun için öncelikle zekâ nedir onu bilmemiz gerekir.

Akıl kelimesi toplumda genellikle insanların zekâ seviyelerini belirtmek için kullanılan bir terimdir. Sıklıkla akıl ve zekâ kelimeleri birbiriyle karıştırılmaktadır. Hâlbuki akıl, düşünme kavrama, idrak etme, karar verme, önlem alma yeteneğidir. Zekâ ise gerçekleri algılama yargılama ve bir sonuca varma yeteneklerinin tümüdür.

Bu arada zekânın ne anlama geldiği ve ne kadarının ölçülebildiği konusunda görüş birliği sağlanamamıştır. Yapılan tanımlamaların ortak bir cümlesi olarak, zekâyı beynin bilgiyi alıp, hızlı ve doğru analiz etmesi olarak tarif edebiliriz [4]

## 1.2 Yapay Zekâ Teknikleri

* Uzman Sistemler (Expert Systems)
* Bulanık Mantık (Fuzzy Logic),
* Yapay Sinir Ağları (Artificial Neural Networks),
* Genetik Algoritmalar (Genetic Algorithms).

Kısaca teknikleri açıklayacak olursak;

* **Uzman Sistemler**, uzmanların belli bir konudaki bilgi ve deneyimlerinin bilgisayara aktarılmasını amaçlayan sistemlerdir.
* **Bulanık Mantık,** 1965 yılında Prof. Dr. Lütfi Aliasker Zade tarafından tanımlanmış olup, bulanık küme teorisine dayanmaktadır. Hava sıcaklığını “Sıcak ve Soğuk” olarak değil de Sıcak-Ilık-Az Soğuk-Çok Soğuk gibi ara değerlere göre çalışmaktadır.
* **Genetik Algoritmalar** ise karmaşık optimizasyon problemlerinin çözülmesinde kullanılan bir teknolojidir ve evrimsel hesaplama tekniğinin bir parçasını oluşturur.
* Peki, **Yapay Sinir Ağları**? [2]

## 1.3 Yapay Sinir Ağları Nedir?

Yapay sinir ağları (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 ağları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır. Bu nedenledir 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 birçok alanda kullanılır hale gelmiştir. İnsan beyninin çalışma prensibini taklit ederek çalışan bu sistemler, her ne kadar bilgisayar teknolojisi hızlı bir gelişim göstermiş, işlem hızları nano saniyeler mertebesine inmiş olsa da bırakalım insan beynini, ilkel bir canlı beyninin fonksiyonları dahi baz alındığında, böyle bir organizmanın yanında çok ilkel kalmaktadır. Nano saniyeler bazındaki işlem hızları ile YSA'lar, mili saniyeler mertebesindeki işlen hızları ile işlem yapan insan beyninin işlevselliğinin henüz çok uzağındadır. Burada kısa bir hatırlatma yapmak gerekirse; insan beyninde yaklaşık 10¹¹ sinir hücresinin varlığından bahsedilmekle birlikte, bu sayının bilgisayar ortamında modellenmesi şu an için mümkün görünmemektedir. Fakat karar hızı açısından insan beyni ile henüz yarışamasalar bile, YSA'lar yapısallıkları ve hassas eşleştirmelerin başarı ile gerçekleştirebilmeleri ile gün geçtikçe daha fazla uygulama alanı bulmaktadır. [10]

### 1.3.1 YSA’ların Avantajları

1. YSA’lar birçok hücreden meydana gelir ve bu hücreler eş zamanlı olarak çalışarak karmaşık işlevleri yerine getirir. Diğer bir değişle karmaşık işlevler birçok hücrenin eş zamanlı çalışması ile meydana getirilir. Süreç içerisinde bu hücrelerden herhangi biri işlevini yitirse dahi sistem güvenli bir şekilde çalışmasına devam edebilir.
2. Eğitim esnasında kullanılan nümerik bilgilerden, problemin genel özellikleri elde etmesi ve böylelikle eğitim sırasında kullanılmayan girdiler için de anlamlı yanıtlar üretebilmesidir.
3. Yapı üzerinde dağılmış belli tipteki non-lineer alt birimler, non lineer problemlerin de çözümünü mümkün kılmaktadır.
4. YSA'lar makina öğrenmesi gerçekleştirebilirler. Yapay sinir ağlarının temel işlevi zaten bilgisayarın öğrenmesini sağlamaktır. Olayları öğrenerek benzer olaylar karşısında mantıklı kararlar verebilirler.
5. Bilgi işleme yöntemleri geleneksel programlamadan farklıdır. Bu nedenle geleneksel programlamanın getirdiği birçok olumsuzluk ortadan kaldırılabilir.
6. Bilgiler ağın tamamında saklanır. Geleneksel programlamada olduğu gibi bilgiler veri tabanları ya da dosyalarda belli bir düzende tutulmaz, ağın tamamına yayılarak değerler ile ölçülen ağ bağlantılarında saklanmaktadır. Hücrelerden bazılarının işlevini yitirmesi, anlamlı bilginin kaybolmasına neden olmaz.
7. Dağıtık belleğe sahiptirler. YSA ’larda bilgi ağa dağılmış bir şekilde tutulur. Hücrelerin bağlantı ve ağırlık dereceleri, ağın bilgisini gösterir. Bu nedenle tek bir bağlantının kendi başına anlamı yoktur
8. Örnekleri kullanarak öğrenirler. YSA ’nın öğrenebilmesi için örneklerin belirlenmesi, bu örneklerin ağa gösterilerek istenen çıktılara göre ağın eğitilmesi gerekmektedir. Ağın başarısı, seçilen örnekler ile doğru orantılıdır, ağa olay bütün yönleri ile gösterilemezse ağ yanlış çıktılar üretebilir.
9. Daha önce görülmemiş örnekler hakkında bilgi üretebilirler. YSA'lar eğitimleri sırasında kendilerine verilen örneklerden genellemeler çıkarırlar ve bu genellemeler ile yeni örnekler hakkında bilgi üretebilirler.
10. Algılamaya yönelik olaylarda kullanılabilirler. YSA ’ların en başarılı oldukları alanlar, algılamaya yönelik uygulama alanlarıdır. Bu alanlarda başarıları kanıtlanmıştır.
11. Örüntü (pattern) ilişkilendirme ve sınıflandırma yapabilirler. YSA ’lar kendilerine örnekler halinde verilen örüntüleri kendisi veya diğerleri ile ilişkilendirebilir. Ayrıca kendisine verilen örneklerin kümelenmesi ile, bir sonraki verinin hangi kümeye dahil olacağının karar verilmesi konusunda kullanılabilirler.
12. Örüntü tamamlama yapabilirler. Ağa eksik bilgileri içeren örüntüler verildiğinde eksik bilgilerin tamamlanması konusunda başarılıdırlar.
13. Kendi kendine öğrenebilme ve organize etme yetenekleri vardır. YSA ’lar online olarak öğrenebilirler ve kendi kendilerini eğitebilirler.
14. Eksik bilgi ile çalışabilmektedirler. Geleneksel sistemlerin aksine YSA ’lar eğitildikten sonra veriler eksik bilgi içerse dahi, çıktı üretebilirler. Bu durum bir performans kaybı yaratmaz, performans kaybı eksik bilginin önemine bağlıdır. Burada bilgilerin önem dereceleri eğitim sırasında öğrenilir.
15. Hata töleransına sahiptirler. YSA ’ların eksik bilgilerle çalışabilmeleri ve bazı hücreleri bozulsa dahi çalışabilmeleri, onları hatalara karşı töleranslı yapar.
16. Dereceli bozulma (Graceful degradation) gösterirler. Bir ağ, zaman içerisinde yavaş ve göreceli bir bozulmaya uğrar. Ağlar problemin ortaya çıktığı anda hemen bozulmazlar.

[10]

### 1.3.2 YSA’ların Dezavantajları

YSA'ların, pek çok avantajın yanında bazı dezavantajları da vardır. Belli başlı dezavantajları;

1. Donanım bağımlıdır. YSA'ların en önemli sorunu donanım bağımlı olmalarıdır. YSA'ların en önemli özellikleri ve var oluş nedenlerinden birisi olan paralel işlem yapabilme yeteneği, paralel çalışan işlemciler ile performans gösterir.
2. Uygun ağ yapısının belirlenmesinde belli bir kural yoktur. YSA ’larda probleme uygun ağ yapısının belirlenmesi için geliştirilmiş bir kural yoktur. Uygun ağ yapısı deneyim ve deneme yanılma yolu ile belirlenmektedir.
3. Ağın parametre değerlerinin belirlenmesinde de belli bir kural yoktur. YSA ’larda öğrenme katsayısı, hücre sayısı, katman sayısı gibi parametrelerin belirlenmesinde belirli bir kural yoktur. Bu değerlerin belirlenmesi için belirli bir standart olmamakla birlikte her problem için farklı bir yaklaşım söz konusu olabilmektedir.
4. Öğrenilecek problemin ağa gösterimi önemli bir problemdir. YSA ’lar nümerik bilgiler ile çalışabilmektedirler. Problemler YSA ’lara tanıtılmadan önce nümerik değerlere çevrilmek zorundadırlar. Burada belirlenecek gösterim mekanizması ağın performansını doğrudan etkileyecektir. Bu da kullanıcının yeteneğine bağlıdır.
5. Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlandığı anlamına gelmektedir. Burada optimum neticeler veren bir mekanizma henüz yoktur ve YSA ile ilgili araştırmaların önemli bir kolunu oluşturmaktadır.
6. Ağın davranışları açıklanamamaktadır. Bu sorun YSA ’ların en önemli sorunudur. YSA bir probleme çözüm ürettiği zaman, bunun neden ve nasıl olduğuna ilişkin bir ipucu vermez. Bu durum ağa olan güveni azaltıcı bir unsurdur.

[10]

## 1.4 Geleneksel Algoritmalar ile YSA'ların Karşılaştırılması

|  |  |
| --- | --- |
| **Geleneksel Algoritmalar** | **Yapay Sinir Ağları** |
| Çıkışlar, koyulan kurallara girişlerin uygulanması ile elde edilir. | Öğrenme esnasında giriş çıkış bilgileri verilerek, kurallar koyulur. |
| Hesaplama; merkezi, eş zamanlı ve ardışıldır. | Hesaplama; toplu, eş zamansız ve öğrenmeden sonra paraleldir. |
| Bellek paketlenmiş ve hazır bilgi depolanmıştır. | Bellek ayrılmış ve ağa yayılmıştır. |
| Hata toleransı yoktur. | Hata toleransı vardır. |
| Nispeten hızlıdır. | Yavaş ve donanıma bağımlıdır. |
| Bilgiler ve algoritmalar kesindir. | Deneyimden yararlanır. |

[10]

## 1.5 Yapay Sinir Ağları Kullanım Alanları

YSA'lar pek çok sektörde değişik uygulama alanları bulmuştur. Bunlardan bazıları;

Uzay: Uçuş simülasyonları, otomatik pilot uygulamaları, komponentlerin hata denetimleri vs.

Otomotiv: Otomatik yol izleme, rehber, garanti aktivite analizi, yol koşullarına göre sürüş analizi vs.

Bankacılık: Kredi uygulamaları geliştirilmesi, müşteri analizi ve kredi müracaat değerlendirilmesi, bütçe yatırım tahminleri vs.

Savunma: Silah yönlendirme, hedef seçme, radar, sensor sonar sistemleri, sinyal işleme, görüntü işleme vs.

Elektronik: Kod sırası öngörüsü, çip bozulma analizi, non-lineer modelleme vs.

Eğlence: Animasyonlar, özel efektler, pazarlama öngörüsü vs.

Finans: Kıymet biçme, pazar performans analizi, bütçe kestirimi, hedef belirleme vs.

Sigortacılık: ürün optimizasyonu, uygulama politikası geliştirme vs.

Üretim: üretim işlem kontrolü, ürün dizaynı, makina yıpranmalarının tespiti, dayanıklılık analizi, kalite kontrolü, iş çizelgeleri hazırlanması vs.

Sağlık: göğüs kanseri erken teşhis ve tedavisi, EEG, ECG, MR, kalite artırımı, ilaç etkileri analizi, kan analizi sınıflandırma, kalp krizi erken teşhis ve tedavisi vs.

Petro kimya: arama, verim analizi vs.

Robotik: yörünge kontrol, forklift robotları, görsel sistemler, uzaktan kumandalı sistemler, optimum rota belirleme vs.

Dil: sözcük tanıma, yazı ve konuşma çevrimi, dil tercüme vs.

Telekomünikasyon: görüntü ve data karşılaştırma, filtreleme, eko ve gürültü sönümlendirilmesi, ses ve görüntü işleme, trafik yoğunluğunun kontrolü ve anahtarlama vs.

Güvenlik: parmak izi tanıma, kredi kartı hileleri saptama, retina tarama, yüz eşleştirme vs.

NESTOR adlı şirket; kredilerin iyi ve kötü olarak sınıflandırılmasında, ipotek sigorta kararlarının finansal risklerinin belirlenmesinde kullanmıştır. [11]

YSA aynı zamanda; metni konuşmaya çevirmede kullanılır. Diğer bir alan karakter tanıma ve el yazısı tanımadır. Bu alan bankacılıkta; kredi kart işlemlerinde belgelerin üzerindeki el yazısını tanımada kullanılır. Sinirsel ağların görüntü tanıma kabiliyetleri çek işlemlerinde el yazısına ve sisteme bir insan sistem çek tarafından girilmiş olan miktarı okumak için kullanılmaktadır. Bu işlemi otomatik hale getiren işlemlerindeki hataları önemli derecede azaltacaktır. Böyle bir sistem Bank Tec için HWC tarafından geliştirilmiştir. [11]

En iyi bilinen uygulamalardan biri ABD hava alanında kurulan bomba detektörleridir. Bu araç SNOOPE olarak adlandırılır ve kimyasal bileşiklerin içindeki birtakım maddelerin varlığını belirlemek için kullanılır. [11]

İngiltere’de ki High Street Banks, Visa,Master Cards, American Express, Europen Banks, USA Banks, ASIA Pasific İnstitution, UK Brokers sinir ağları teknolojisine yatırım yapan büyük şirketlerden birkaçıdır. [11]

Ses tanıma teknolojisi ile ilgili yurt dışında, özellikle de Amerika'da bu konuda çalışan pek çok firma var. Ses tanıma ve doğal dil işleme, Microsoft'un hesaplarına göre Dos'tan Windows'a geçişten sonraki en büyük atılım olacak.

YSA tıbbın ilgi alanlarına bugünlerde fazlaca giriyor olduğunu söyleyebiliriz. Önümüzdeki birkaç yıl içinde YSA’nın biyomedikal sistemlerde yoğun bir şekilde uygulanacağına inanılmaktadır. Şu anda ki çalışmalar insan vücudunun parçalarını modelleme üzerine yoğunlaşmıştır ve çeşitli taramalarla hastalık teşhisi üzerine yoğunlaşmıştır.

YSA’nın, finans dünyasında birçok uygulamaları vardır. Bankacılıkta, kredi kartı şirketleri ve faiz kurumları net olmayan kararlarla uğraşır. Bu alanlar öğrenme ve istatistiki eğilim gerektirir. Borç izin işlemi borç memuruna karar verdirebilecek form doldurulmasını gerektirir. Bu formlardaki veriler günümüzde geçmişte alınan kararlarla alakalı verilerde çalışmış sinir ağları tarafından kullanılmaktadır. Kredi kardı şirketleri de kredi riski ve kredi limiti inşasında yardımcı olmak için benzer geri yayma ağları kullanıyor. Genel olarak; YSA bütün finansal borsalarda, bonolarda kullanılıyor.

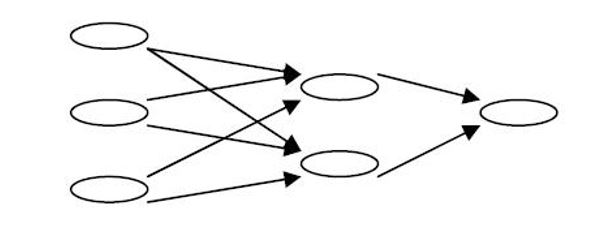
Örnekler çoğaltılabilir. YSA'lar günlük hayatımızda pek çok alanda kullanılmaktadır.

## 1.6 Yapay Sinir Ağlarının Sınıflandırılması

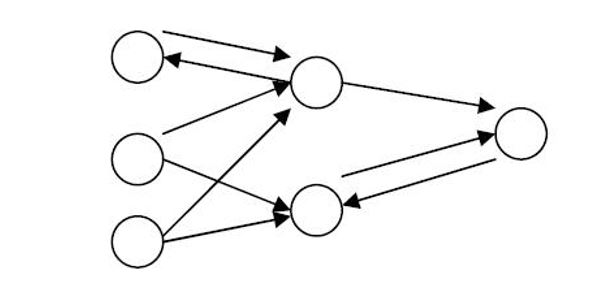
### 1.6.1 Yapay Sinir Ağlarının Yapıları

Yapay sinir ağları içerdiği nöronların birbirine bağlanış 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.



**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. Bu yapısı ile geri beslemeli yapay sinir ağları doğrusal olmayan dinamik bir davranış göstermektedir. Geri besleme özelliğini kazandıran bağlantıların bağlanış şekline göre geri aynı yapay sinir ağıyla farklı davranışta olan yapay sinir ağları elde edilebilir.



### 1.6.2 Öğrenme Yöntemlerine Göre Yapay Sinir Ağları

**Eğiticili Öğrenme:** Eğiticili 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. Güncelleme durumuna ‘backpropogation’ geri besleme denir. [10]

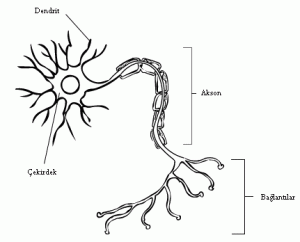
**Yardımsız Öğrenme:** Eğiticili Öğrenme ’nin aksine ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini her veri girdisinde tekrarlar. [10]

**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. [10]

## 1.7 Yapay Sinir Ağlarının Yapısı

### 1.7.1 **Biyolojik Sinir Hücresinin Yapısı**

Biyolojik sinir sisteminin temel yapı taşı olan nöronların yapısı dört ana bölümden oluşmaktadır; dendrit, akson, çekirdek ve bağlantılar. Dendritlerin sinir hücresinin ucunda bulunan ve ağaç kökü görünümüne sahip bir yapıya sahiptir. Dendritlerin görevi bağlı olduğu diğer nöronlardan veya duyu organlarından gelen sinyalleri çekirdeğe iletmektir. Çekirdek dendrit tarafından gelen sinyalleri bir araya toplayarak ve aksona iletir. Toplanan bu sinyaller akson tarafından işlenerek nöronun diğer ucunda bulunan bağlantılara gönderilir. Bağlantılar ise yeni üretilen sinyalleri diğer nöronlara iletir.

 [12]

### 1.7.2 **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
* Birleştirme fonksiyonu
* Aktivasyon fonksiyonu
* Çıktılar

#### Girdiler

Girdiler nöronlara gelen verilerdir. Girdiler yapay sinir hücresine bir diğer hücreden gelebileceği gibi direk olarak dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nörona gönderilir.

#### Ağırlıklar

Yapay sinir hücresine 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.

#### Birleştirme Fonksiyonu

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

Bazı diğer birleştirme fonksiyonları

|  |  |
| --- | --- |
| Çarpım | Ağırlık değerleri girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak Net girdi hesaplanır. |
| Maksimum | n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en büyüğü Net girdi olarak kabul edilir. |
| Minimum | n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en küçüğü Net girdi olarak kabul edilir. |
| Çoğunluk | 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 | Hücreye gelen bilgiler ağırlıklı olarak toplanır. Daha önce hücreye gelen bilgilere yeni hesaplanan girdi değerleri eklenerek hücrenin net girdisi hesaplanır. |

[10]

#### Aktivasyon Fonksiyonu

Birleştirme (toplama) fonksiyonundan çıkan NET toplam hücrenin çıktısını oluşturmak üzere aktivasyon fonksiyonuna iletilir. 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 hesaplamanın yavaşlamaması için türevi kolay hesaplanır bir fonksiyon seçilir.

##### Doğrusal Aktivasyon Fonksiyonu

Doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon da seçilebilir. Doğrusal aktivasyon fonksiyonları matematiksel olarak olarak genellenebilir. Bu formülde A sabit bir katsayıdır. A değerinin değişimi şekilde gösterilen doğrunun çıkış ekseniyle yaptığı açıyı değiştirmektedir.

##### Adım Aktivasyon Fonksiyonu

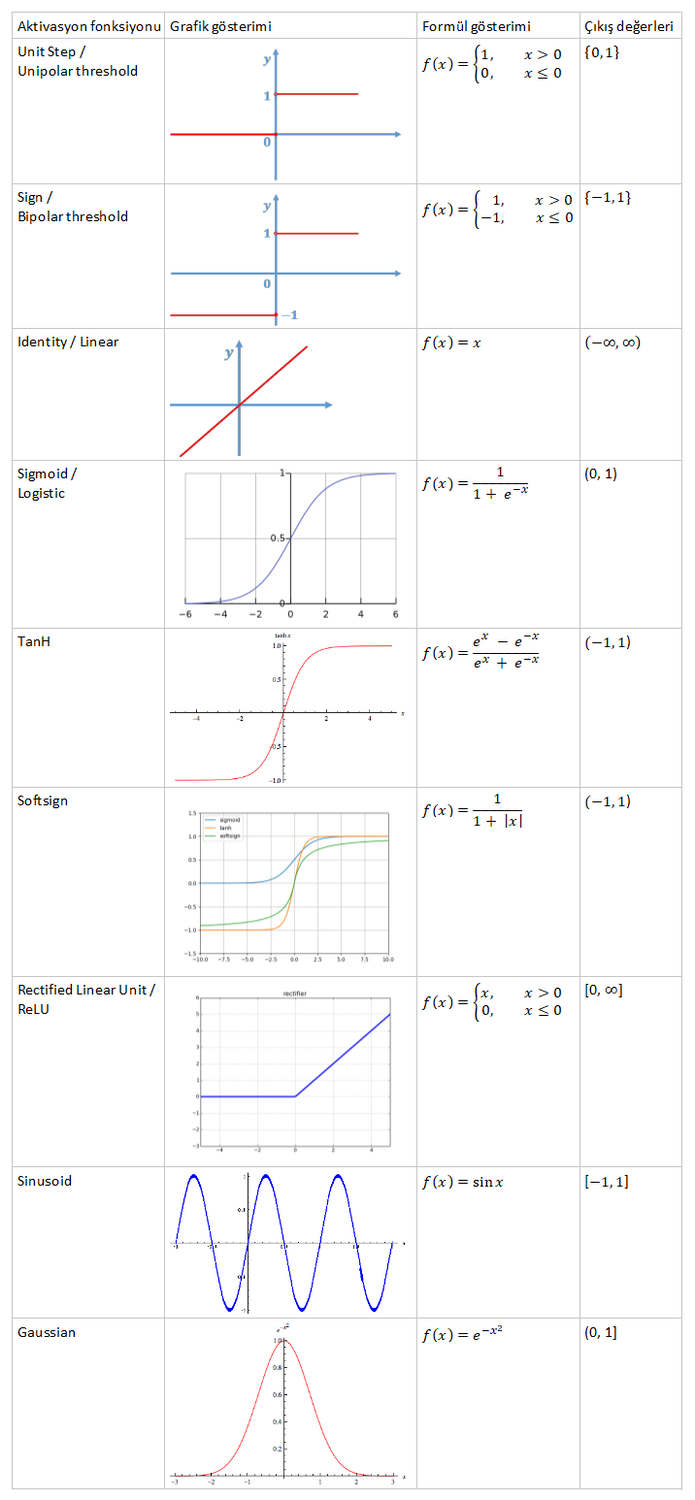
Girdilerin sıfırdan büyük olup olmamasına göre -1 veya 1 çıktısı veren fonksiyondur. Sadece iki çeşit çıktı vermektedir.

##### Sigmoid Aktivasyon Fonksiyonu

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 sıfır ile bir arasında bir değer üretir. Sigmoid fonksiyonunun matematiksel ifadesi

##### Tanjant Hiperbolik Aktivasyon Fonksiyonu

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. Matematiksel ifadesi:

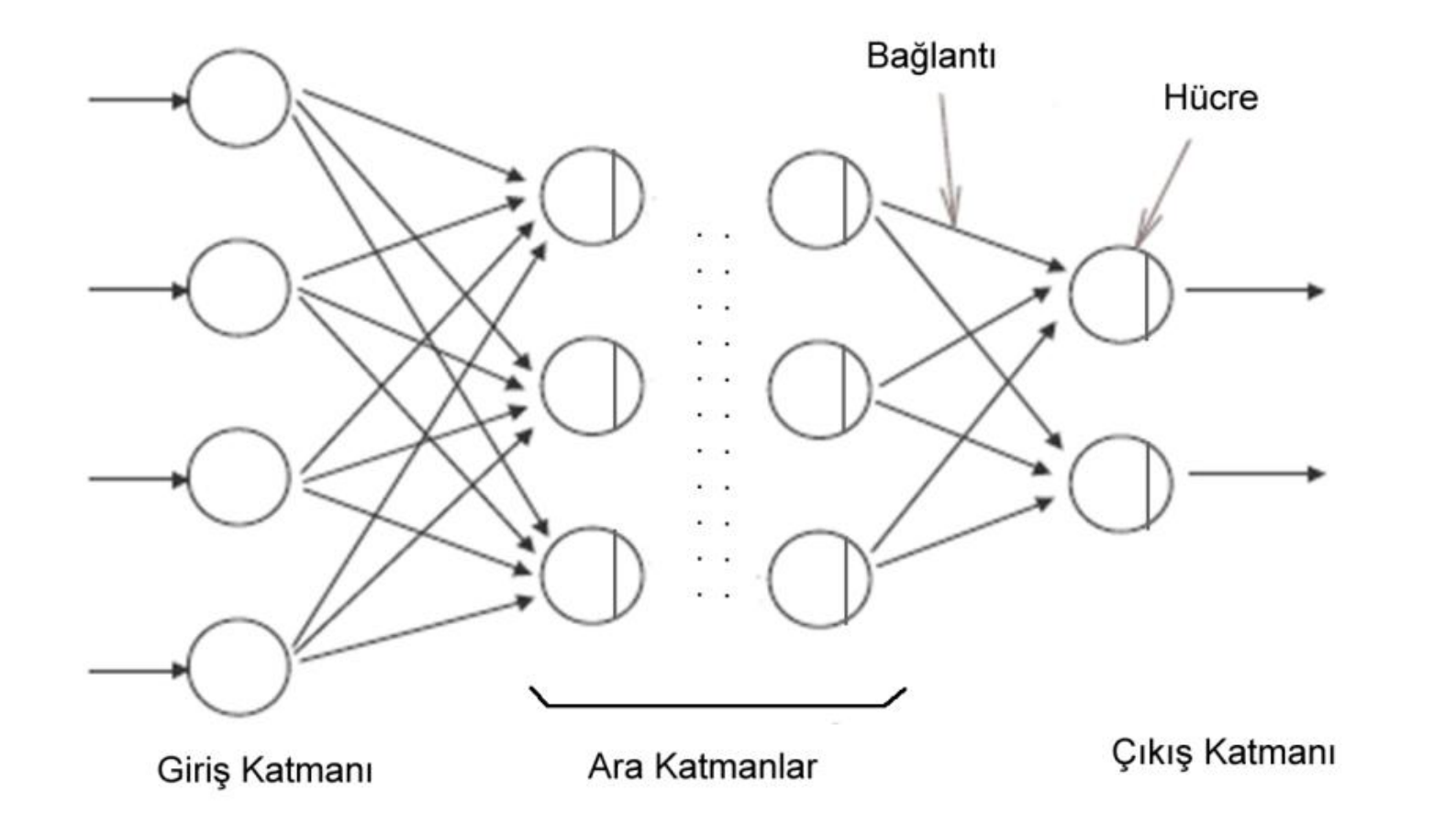
 [13]

#### Çıktılar

Aktivasyon fonksiyonundan çıkan değer nöronun çıktı değeri olmaktadır. Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilir ister tekrardan ağın içinde kullanılabilir. Nöronun bir çıktısı olmasına rağmen bu çıktı istenilen sayıda nörona bağlı olabilir.

#### Tek Katmanlı YSA Modeli

#### Çok Katmanlı YSA Modeli

 [10]

**Giriş Katmanı:** Yapay sinir ağına dış dünyadan girdilerin geldiği katmandır. Bu katmanda dış dünyadan gelecek giriş sayısı kadar hücrenin bulunmasına rağmen genelde girdiler herhangi bir işleme uğramadan alt katmanlara iletilmektedir.

**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.

## 1.8 Yapay Sinir Ağlarının Eğitilmesi

İnsan doğumdan sonra gelişme sürecinde çevresinden duyu organlarıyla algıladığı davranışları yorumlar ve bunları davranışlarında kullanır. Olaylar karşısında tecrübelenmiş olunur. Benzer bir olay karşısında vereceğimiz tepki bellidir. Buna benzer şekilde 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 düzeltmesi yapılır.

Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek sonuca yaklaşılır. Bu çalışma süresince yenilenen yapay sinir ağının ağırlıklarıdır. Ağırlıklar her bir girdide yenilenerek doğru sonuca yaklaşılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir.

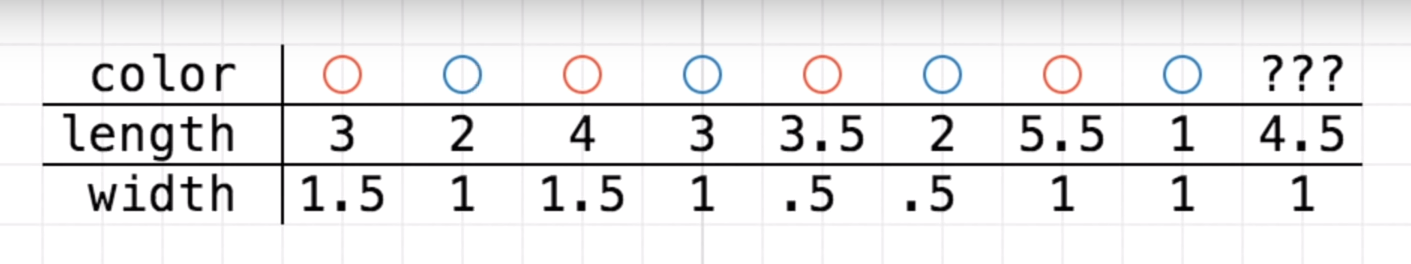
Ağırlıkların sürekli yenilenerek istenilen sonuca ulaşılana kadar geçen zamana öğrenme adı verilir. Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkış yaklaşımı incelenir. Eğer yeni verilen örneklere de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş, ezberlemiş demektir. Genelde eldeki örneklerin büyük bir kısmı öğrenme grubu olarak alınır geri kalan kısmı ile elde edilmiş ağda test edilmek için kullanılır.

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.
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.

[10]

# Bir Örnek Üzerinde Matematiksel Açıklama

Örneğin aşağıda verilmiş tablodaki yaprak uzunluğu ve genişliği verilmiş 2 renkte bulunabilen bir çiçeğin yapay sinir ağları ile tahmin edilmesini düşünelim;

 [14]

Bu örnekte iki adet girdi nöronu bulunmaktadır.

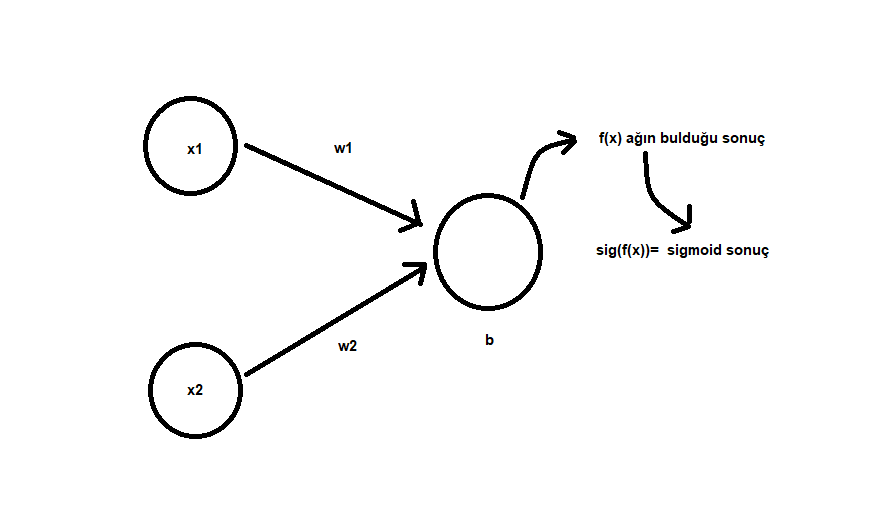
Tahmini çıkış olarak bir çıktı nöronu yeterlidir.

Birleştirici fonksiyon olarak Toplama Net fonksiyonunu kullanalım,

Aktivasyon fonksiyonu olarak sigmoid fonksiyonunu seçelim

Sigmoid fonksiyon seçtiğimiz için sonuç değerlerimiz 0 ile 1 arasında olacaktır. Bu yüzden kırmızı çiçeği 1 mavi çiçeği 0 olarak numaralandırıyoruz.

Sonuçta yapay sinir ağımızın topolojisi aşağıdaki şekilde olmuştur.



Burada , ve başlangıç değeri olarak rastgele değerler almaktadırlar.

değerleri için tablodaki birinci sütuna toplam ve sigmoid fonksiyonumuzu çalıştırırsak;

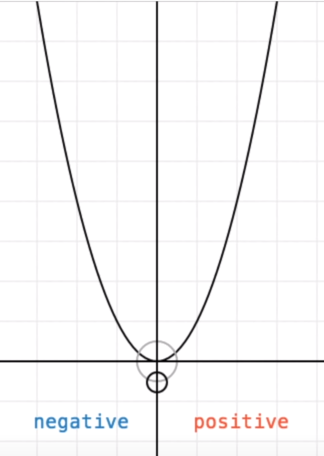
sonuçlarını elde ederiz.

Başlangıçta kırmızı çiçeği 1 seçtik ve sonuç 1’e yakın çıktı yapay sinir ağımız bu girdiler ile doğru tahmin yaptı. Böylece , ve katsayılarımızı düzeltmemize gerek kalmadı.

Aynı , ve değerleri için tablodaki ikinci sütuna toplam ve sigmoid fonksiyonumuzu çalıştırırsak;

Bu sefer tahmin etmeye çalıştığımız çiçek maviydi fakat bize 0 değerine yakın bir değer beklerken yine 1’e yakın bir sonuç vererek yanlış tahminde bulunduğunu gösterdi. , ve katsayılarımızı güncellememiz gerekiyor.

Hata fonksiyonu olarak karesel hata fonksiyonunu seçelim.

[15]

Bu hata ifadesi parabolik bir ifade olduğu için hata fonksiyonundaki değerinin beklenen değere yakın olması istenmektedir. Dediğimiz gibi parabolik bir ifade olduğu için bu hatanın eğimi bize hata payını ve hatanın negatif yönde mi pozitif yönde mi olduğunu göstermektedir.

Hatanın eğimi ise bize hata noktasındaki türevi göstermektedir yani

h ihmal edilebilir.

#### Lineer Regresyon Yöntemi

Elimizdeki verileri kullanıp yapay sinir ağını eğitmek için lineer regresyon ile her bir bağlantı katsayısı ve sapma değeri katsayısı için gradient vektörlerini bulup her bir katsayıyı bulduğumuz gradient vektörleri ile güncellemeliyiz.

Elimizde n tane veri olmak üzere (Bu örnekte 8 veri var);

açık olarak;

.

Sırasıyla

Şimdi gibi bir öğrenme katsayısı alalım. Bu katsayı ve bulduğumuz gradient vektörü ile , bağlantı katsayılarını ve sapma katsayısını güncelleyelim.

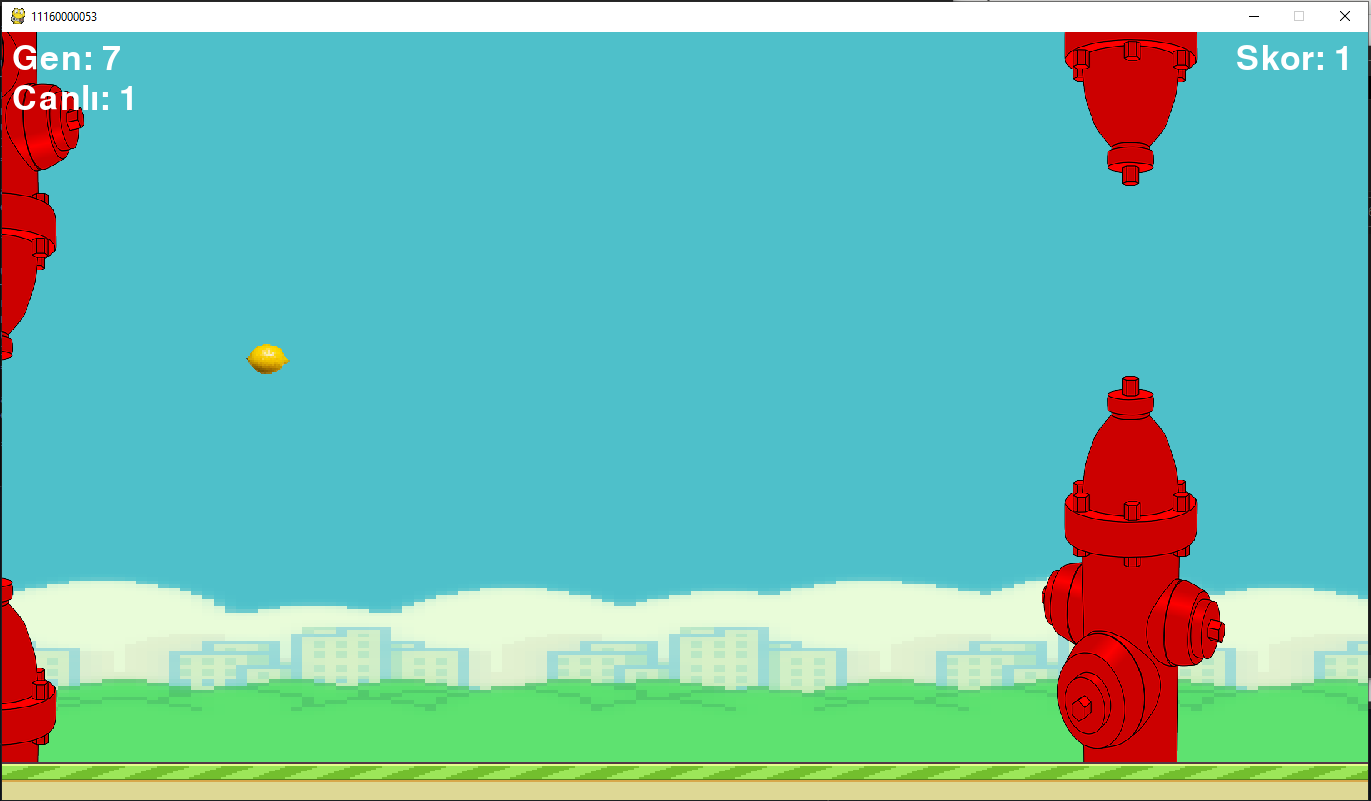
Katsayılarımızı güncellediğimize göre elimizde yapay sinir ağı modelinin verileri tanıyıp tanımadığını kontrol edebiliriz.

Burada uyguladığımız lineer regresyon tekniğini seçtiğimiz öğrenme katsayısının büyüklüğüne bağlı olarak yeteri sayıda iterasyon ile tekrarladığımızda modelimizi eğitmiş yani modelimiz karşılaşacağı benzer veriler karşısında yüzdeli bir oranda doğru şekilde tahmin yapacaktır.

# Kompleks Örnek

[16]

**Bu örnek ek olarak 11160000053 numarası ile ekteki dosyadadır.**



Günlük hayatta karşılaştığımız, çözülmesini istediğimiz veya otomasyonunu yapmak istediğimiz işler için yapay sinir ağları bizim için biçilmiş kaftandır. Fakat her modelde gerektiği gibi bazı ayarlamaları yapmak, matematiksel işlemleri yapmak modelin çözümünü belirlemek uğraştırıcı ve zor olabilir. Bu durum için yardımcı kaynakları kullanarak iş yükünü hafifletebilir, çözüme daha hızlı bir şekilde ulaşabiliriz.

Yapay sinir ağ modelini oluştururken bize gerekli olan parametreleri ve modelin ana parçalarını belirlemememiz gerekir;

* Girdi sayısı
* Çıktı sayısı
* Ara katmanların sayısı (model topolojisi)
* Model yapısı (ileri beslemeli, geri beslemeli, statik vs.)
* Aktivasyon fonksiyonu
* Öğrenme katsayıları
* Uyum başarısı
* Elitizm katsayısı vs.

Yukarıdaki ve yukarıdaki gibi birçok değeri her model için belirlemek zaman kaybı ve uğraştırıcı olacaktır. Bize yardımcı olması için python programla dilini ve aşağıda belirtilen kütüphaneleri kullanmaktayız.

#### Pygame

[17]

Pygame kütüphanesi bize modelimizin görsel bir çıktısını elde edebilmek için yardımcı olur.

#### Random

[18]

Random kütüphanesi/modülü bize bazı katsayılar için gerekli rastgele değerleri üretmemizde yardımcı olur.

#### Os

[19]

Os kütüphanesi/modülü bize sistem dosyalarına erişim sağlayabilmek için kullandığımız bir araçtır.

#### Time

[20]

Zaman ile ilgili işlemleri yapmamızı sağlayan kütüphanenin/modülünün adı Time’dır.

#### Neat

[21]

Neat kütüphanesi yapay sinir ağlarıyla ilgili tüm işimizi otomatik veya yarı otomatik halde yapan araçtır. En büyük özelliği gerekli ayarları yapıp, girdileri ve çıktıları belirledikten sonra modelin topolojisini kendi belirleyip gerekiyorsa iterasyon anında ekleme ve çıkarma gibi düzeltmeler ile daha iyi bir modele evrimleştirmesidir. Girdileri ve çıktıları gönderdiğimizde gerekli matematiksel hesapları yapıp modeli oluşturup yeni girdiler için veri bekler.

#### Pickle

[22]

Bu kütüphane nesne tabanlı programlamada kullanılan nesnelerin düzleştirme işlemini yapar. Böylece objelerin düz yazı gibi kullanılması sağlanır bir yerden bir yere aktarılması kolaylaşır.

#### Örneğin Açıklaması

Bu bir konseptin görselleştirilerek yapay sinir ağlarına modellenmesine bir örnektir.

Program çalıştırıldığında bir pencere açılır. Başlangıç için bir yapay sinir ağı modeli oluşturulur. Bu pencerede neat kütüphanesi ile ayarlanmış özelliklerdeki kadar obje yaratılır ve ekranın ortasına konur. Bu objeler ileri doğru hareket etmeye başlar ilerlemek için zıplamak zorundadırlar. Hareket ettikçe karşılarına arasından geçmekte zorunlu oldukları belli mesafede aralığı olan iki boru çıkar. Nesne yaşamak istiyorsa bu iki borunun arasından geçmek zorundadır, tavana veya yere değerse de ölür.

Her gen için ekrana 10 adet nesne konulur konsept olarak bu nesnelerden en çok hayatta kalan en iyisidir ve bir sonraki kuşağa özellikleri aktarılmalıdır. Nesneler ileri doğru hareket ettikçe yani hayatta kaldıkça ve boruların arasından geçtikçe belirli puanlar kazanır böylece o kuşaktaki en iyi nesne böyle seçilmiş olur.

Bu yapay sinir ağı modelinde girdiler 3 adettir ve sırasıyla nesnenin pozisyonu, yukardaki borunun pozisyonu ve aşağıdaki borunun pozisyonudur. Her bir nesne için bu girdiler model üzerinde çalıştırılır ve nesnenin zıplayıp zıplamayacağı da çıktıdır.

Her bir nesne yaptığı zıplama doğru ise yani zıplama sonucunda hayatta kaldı ise evrilir. Bu işlem kuşakta hiçbir canlı nesne kalmayana kadar devam eder. Kuşakta hiçbir canlı nesne kalmadıysa bir önceki kuşaktaki en iyi nesne seçilir ve bir sonraki kuşağın buna benzer şekilde yaratılmasına yardımcı olur. Yani yeni yaratılan kuşak bir önceki yapay sinir ağının bağlantı katsayıları ’leri ve sapma katsayılarını kalıtımsal olarak almış olur.

Yeni kuşağın yaratılma süreci programda belirlenen sınır uyum başarısına kadar devam eder. Belirli bir süre sonra modeldeki nesne her türlü aralıktan geçmeye elverişli olacaktır böylece bu iş için bir model üretilmiş otomasyon sağlanmış olacaktır.

# Kaynakça

|  |  |
| --- | --- |
| [1] | «Artificial neural networks are changing the world. What are they?,» [Çevrimiçi]. Available: http://www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are-they. |
| [2] | «What is the Artificial Neural Networks (ANN) ?,» [Çevrimiçi]. Available: https://kod5.org/yapay-sinir-aglari-ysa-nedir/. |
| [3] | L. T., A Brief History of Artificial Intelligence, 2014. |
| [4] | T. H., Yapay Zekâ, Journal of Yasar University, Yasar University, 2006. |
| [5] | V. V. Nabiyev, Yapay Zekâ: İnsan-Bilgisayar Etkileşimi, Seçkin Yayıncılık, 2012. |
| [6] | A. M. Andrew, Artificial Intelligence, AddisonWesley Company, 1991. |
| [7] | E. V. Popov, Uzman Sitemler ve Doğal Dil İşleme, Radio i Svyaz, 1990. |
| [8] | J. Copeland, Artificial Intelligence: A Philosophical, Oxford, 1993. |
| [9] | E. &. m. D. Charniak, Introduction to Artifical, Addison-Wesley Company, 1985. |
| [10] | İ. ÇAYIROĞLU, «İLERİ ALGORİTMA ANALİZİ-5,» [Çevrimiçi]. Available: http://www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi/ilerialgoritmaanalizi-5.hafta-yapaysiniraglari.pdf. [Erişildi: 26 Mayıs 2020]. |
| [11] | «YSA Uygulama Alanları,» [Çevrimiçi]. Available: https://tektasi.net/wp-content/uploads/2014/01/ysa-uygulama-alanlari.pdf. |
| [12] | *https://i0.wp.com/yapayzeka.ai/wp-content/uploads/2017/07/noron.gif?resize=300%2C242&ssl=1.* [Art]. |
| [13] | *https://www.linkedin.com/pulse/yapay-sinir-a%C4%9Flar%C4%B1-ve-tek-katmanl%C4%B1-a%C4%9Flarda-%C3%B6%C4%9Frenme-tanju-do%C4%9Fan.* [Art]. |
| [14] | giant\_neural\_network, «Youtube,» [Çevrimiçi]. Available: https://www.youtube.com/watch?v=JNlxvwc-Hho&list=PLxt59R\_fWVzT9bDxA76AHm3ig0Gg9S3So&index=2. |
| [15] | giant\_neural\_network. [Çevrimiçi]. Available: https://www.youtube.com/watch?v=c6NBkkKNZXw&list=PLxt59R\_fWVzT9bDxA76AHm3ig0Gg9S3So&index=5. |
| [16] | T. W. Tim, «Youtube,» [Çevrimiçi]. Available: https://www.youtube.com/playlist?list=PLzMcBGfZo4-lwGZWXz5Qgta\_YNX3\_vLS2. |
| [17] | *https://www.pygame.org/.* |
| [18] | https://docs.python.org/3/library/random.html. |
| [19] | https://docs.python.org/3/library/os.html?highlight=os#module-os. |
| [20] | https://docs.python.org/3/library/time.html?highlight=time#module-time. |
| [21] | K. O. Stanley, https://neat-python.readthedocs.io/. |
| [22] | https://docs.python.org/3/library/pickle.html. |