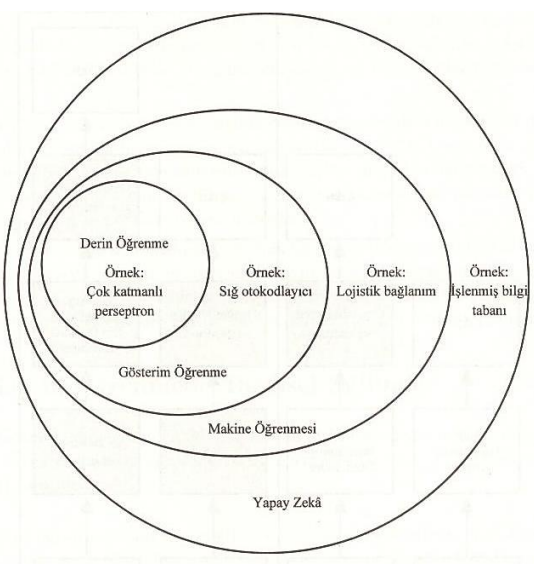
İleri beslemeli derin ağlar ya da çok katmanlı perseptron (MultiLayer Perceptron, MLP) derin öğrenmenin özünü anlatan bir modeldir. Çok katmanlı perseptron, girdileri çıktılara eşleyen matematiksel bir fonksiyondan ibarettir.

Bir modelin derinliğini ölçmenin iki temel yöntemi vardır:

* Sıralı İşlemler Sayısı
* Kavramsal İlişkiler Sayısı



Yapay Zekanın, Uzman Sistemler (US) , Bulanık Mantık (BM), Genetik Algoritma (GA) ve Yapay Sinir Ağları (YSA) gibi alt dalları özellikle son yıllarda, geniş bir araştırma ve uygulama alanı bulmaktadırlar.

**Yapay Sinir Ağları**, genel olarak insan beyninin sinir ağlarını taklit eden bilgisayar programlarıdır. Yapay sinir ağları bir anlamda paralel bilgi işleme sistemi olarak düşünülebilir. Yapay sinir ağlarına bu bilgiler ilgili olaya ait örnekler üzerinde eğitilerek verilir.

**Yapay öğrenme**, makine öğrenmesi olarakta isimlendirilir. Bir işi bilgisayarda gerçekleyebilmek için algoritmasını bilmek gerekir.

Yapay Öğrenme Uygulamalarına Örnekler:

* İlişkilendirme Kuralları
* Sınıflandırma
* Bağlanım
* Gözetimsiz Öğrenme
* Pekiştirmeli Öğrenme

**Örüntü Tanıma**, Aralarında ortak özellik bulunan ve aralarında bir ilişki kurulabilen karmaşık işaret örneklerini veya nesneleri bazı tespit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırmadır.

1. **Veri/İşaret/Görüntü İşleme**: Ön işlem aşamasıdır. İşaret veya görüntünün filtre edildiği, çeşitli dönüşüm ve gösterim teknikleri ile işlendiği, bileşenlerine ayrıldığı veya modellendiği kısımdır.
2. **Özellik Çıkarma/Seçme**: İşaret ve görüntünün veri boyutunun indirgendiği ve tanımlayıcı anahtar özelliklerinin tespit edildiği ve aynı zamanda normalizasyona tabii tutulduğu aşamadır.
3. **Sınıflandırma**: Çıkarılan özellik kümesinin indirgendiği ve formüle edildiği tanımlayıcı karar aşamasıdır.

**Makine öğrenmesi**, basit bir tanımla veriden üretilen yazılımdır. Makine öğrenmesi, belirli bir amaç için programlama yerine örneklerden öğrenir ve bu önceki pratiğe göre önemli bir farklılıktır.

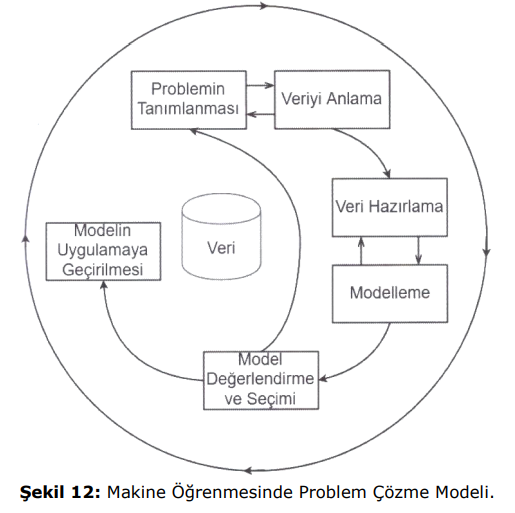
Makine öğrenmesini etkileyen faktörler:

* Makineye deneyim olarak sunulan veri setidir.
* Veri setinde sonuca etkisi olduğu düşünülen değişkenlerin bulundurulmasıdır.
* Seçilen öğrenme stratejisidir.
* Öğrenme için kullanılan algoritma ve varsa algoritmaya ait parametrelerdir. En yüksek performansın elde edildiği modelin kullanılması tercih edilmektedir.

Öğrenme Stratejileri, Üçe ayrılır:

1. Gözetimli Öğrenme
2. Gözetimsiz Öğrenme
3. Pekiştirmeli Öğrenme

**Problem Çözme**: Problemin Tanımlanması, Veriyi Anlama, Veri Hazırlama, Model Kurma, Model Değerlendirme ve Seçimi ve Modelin Uygulamaya Geçirilmesi olmak üzere altı aşamadan oluşmaktadır. Her bir adım ayrıntılı bir biçimde ele alınmıştır.



* **Veriyi Anlama:** enformasyon, bilgi ve bilgelikten oluşan zincirin ilk basamağıdır. Enformasyon, yorumlandığında, bir anlam eklendiğinde bilgi haline gelmektedir; bilgelik ise bilginin kullanılıp geleceğe dair çıkarımlar yapabilme ya da alternatifler içinden karar verebilme gibi zeki davranışları gerçekleştirebilme durumudur.

Veri setini oluşturacak nitelikler ise farklı veri tipine sahip olabilir:

* + Nominal: Öğretmen yerine 1, doktor yerine 2, avukat yerine 3 ve son olarak mühendis yerine de 4 yazılabilir.
  + İkili: Cinsiyet için Kadın-Erkek, medeni durum için Evli-Bekâr, hastanın kanser olup olmaması Evet-Hayır vb. ikili niteliklere ait örneklerdendir.
  + Sıralı: Bir okuldaki öğrenci başarısının "çok düşük/düşük/orta/yüksek/çok yüksek" şeklinde veya bir işletmenin büyüklüğünün "küçük/orta/büyük” biçiminde değerlendirilmesinde olduğu gibi.
  + Tamsayı: Örneğin; kişinin sahip olduğu evcil hayvan sayısı, bir takımın attığı gol sayısı vb.
  + Aralık-Ölçeği: En sık verilen örnek, Selsiyus ve Fahrenayt sıcaklık dereceleri arasındaki dönüşümleridir. Selsiyus olarak ölçülen iki dereceyi birbiriyle büyüktür/küçüktür biçiminde yorumlanabilir.
  + Oran-Ölçeği: Bir hastanın boyu, kilosu, bir malın satış fiyatı vb. gibi örnekler verilebilir.
* **Veri Hazırlama**, Verinin analizlere hazırlanması için bazı durumlarda birtakım işlemlerin veriye uygulanması gerekebilir.

Uç Noktalar: diğerlerinden gözle görülür bir farkla ayrılmış olan gözlemlerdir.

Tekrar Eden Gözlemler: Tekrar eden satırlar bir yandan, analizler sırasında vakit kaybı yaşatarak veri setinin gereksiz biçimde şişmesine neden olacaktır.

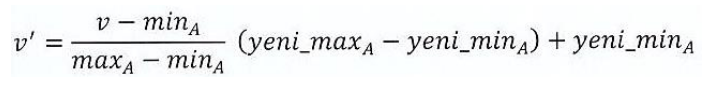
Kayıp Değerlerin Tamamlanması: Regresyon yöntemi ile kayıp değerler tamamlanabilir vb.

**Normalizasyon**: Büyük değerlere sahip nitelikler küçük değerleri domine ederek sonuç üzerindeki etkisi artmaktadır. Min-max, z-score, ondalık ölçekleme bu yöntemlerden bazılarıdır:

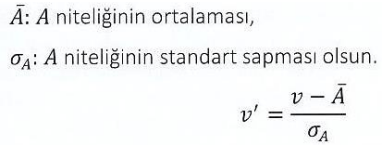
v : A niteliğine ait normalize edilmek istenen değer,

v' : v nin normalize edilen değeri olmak üzere;

* Min-Max:



* Z-Score Yöntemi:



* Ondalık Ölçekleme Yöntemi:
  + - Max(lv’l) < 1 koşulunu sağlayan en küçük tam sayı j olmak üzere;



* **Model kurma**;

Probleme ve öğrenme stratejisine uygun bir algoritmanın yardımı ile girdilerin, istenilen çıktılara dönüştürülmesini ifade etmektedir. Model bir görevin niteliklerle tanımlanan verisinin, çıktılara uygun olarak haritalamadır. Modelleri üretenin öğrenme algoritmaları olup, öğrenme problemleri algoritmalarca çözülmekte ve görevler ise modeller tarafından yönlendirilmektedir.

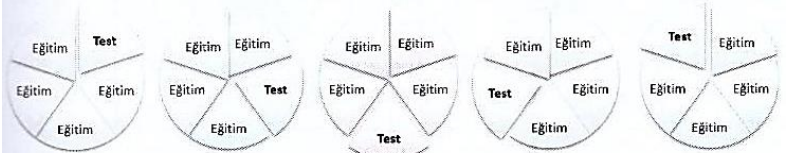
Makine öğrenmesi algoritmasının performasını belirleyen faktörler arasında şunlar sayılabilir:

• Eğitim hatası küçük olmalıdır

• Eğitim hatası ile test hatası arasındaki açıklık küçük olmalı.

Model Performans Değerlendirme Yöntemleri:

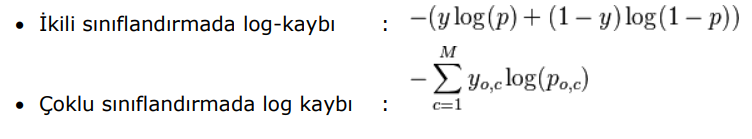
1. **Holdout**: Veri setinin eğitim ve test olmak üzere iki parçaya ayrıldığı yöntemdir. Genel bir yaklaşım olarak ikiye bölmede oranlar, eğitim kümesi için %70-80, test kümesi için ise %20- 30 alınır.
2. **Tekrarlı Holdout**: Kısaca Holdout yönteminin birkaç defa tekrarlanmasıdır.
3. **Tabakalı Örnekleme**: Hedef niteliğin nominal veri tipinde olduğu veri setlerinde, hedef niteliğin bazı kategorilerine ait örnekler az sayıda olabilir ya da hiç olmayabilir. Bu gibi durumlarda, veri seti eğitim ve test olarak ayrılırken hedef niteliğin kategorilerine ait oranlarının korunması istenebilir.
4. **Üçlü Ayırma**: Doğrulama veri setindeki örnekler ile kullanılan algoritmaya ait parametrelerin ince ayarı yapılmaktadır.
5. **Çapraz Geçerleme (cross validation)**: veri seti k eşit parçaya ayrılır. Elde edilen k parçanın her biri bir kez test veri seti, kalan diğer k-1 parça ise eğitim veri seti olarak seçilir. K defa tekrar edilen performans ölçülerinin ortalaması alınır ve nihai performans bulunur.



1. **Bootstrap örnekleme**: veri setinden eğitim veri seti için n defa rastgele örnek seçmektedir. Ancak, seçilen örnek veri setinden çıkarılmaksızın seçim işlemleri sürdürülür.

Model Performans Değerlendirme Ölçütleri:

Bu tarz problemler için kullanılabilecek ölçütlerin bazıları şunlardır:

* **Log-Kaybı**: Tahmin değerinin 0 ve 1 arasında bir olasılık değeri olduğu durumlarda sınıflandırma modelinin performansını ölçer. Mükemmel modelin LogKaybı değeri sıfırdır. 
* **Karmaşıklık/Hata Matrisini**: Pozitif ve Negatif olarak ayrılmış olanlar hedef niteliğin sınıfları biçiminde düşünülebilir.

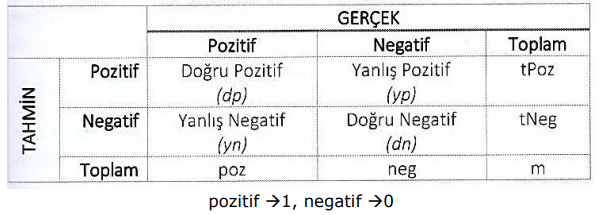
Örneğin; bir kişinin kalp hastalığı var mı / yok mu tahmini yapılırken

dp (TP – true positive): Gerçekte kalp hastası olan hastalardan modelin kalp hastası olarak tahmin ettiği hastaların sayısıdır.

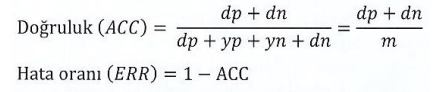
yp (FP – false positive): Gerçekte kalp hastası olmayan hastalardan modelin kalp hastasıdır biçiminde tahmin ettiği hastaların sayısıdır.

yn (FN -false negative): Gerçekte kalp hastası olan hastalardan modelin kalp hastası değildir biçiminde tahmin ettiği hastaların sayısıdır.

dn (TN - true negative): Gerçekte kalp hastası olmayan hastalardan modelin kalp hastası değildir biçiminde tahmin ettiği hastaların sayısıdır.



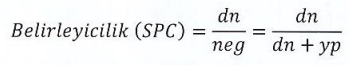
* + - **Doğruluk ve Hata Oranı**: Doğruluk, sınıflandırma problemlerinde doğru tahminlerin bütün tahminlere oranıdır.



* + - **Duyarlılık**: pozitif sınıf etiketini tahmin etmedeki etkililiği.



* + - **Belirleyicilik**: negatif sınıf etiketini tahmin etmedeki etkililiği.



* + - **Yanlış Pozitif Oranı**: Gerçekte negatif olan ancak pozitif olarak sınıflandırılmış örneklerin, tüm negatif etiketli örneklere oranıdır.



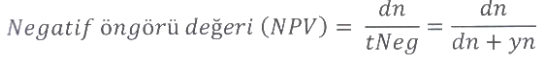
* + - **Yanlış negatif oranı**: Gerçekte pozitif olan ancak negatif olarak sınıflandırılmış örneklerin, tüm pozitif etiketli örneklere oranıdır.



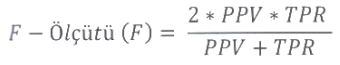
* + - **Kesinlik ya da Pozitif Öngörü Değeri**: Doğru sınıflandırılan pozitif örneklerin toplam pozitif tahmin edilen örneklere oranıdır.



* + - **Negatif Öngörü Değeri**: Doğru sınıflandırılan negatif sınıf etiketine sahip örneklerin toplam negatif tahmin edilen örneklere oranıdır.



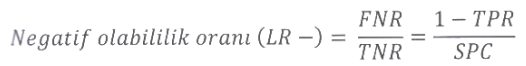
* + **F1-Scoru**: Kesinlik (precision) ve Duyarlılık (recall) performans değerlendirme ölçütlerinin harmonik ortalamasıdır.



* + - **Pozitif Olabilirlik Oranı**: Elde edilen sonuç ne kadar yüksekse, gerçekte pozitif sınıf etiketi almış örnekler o kadar iyi sınıflandırılmaktadır.



* + - **Negatif Olabilirlik Oranı**: Elde edilen değer ne kadar küçükse, gerçekte negatif sınıf etiketine sahip örnekler o kadar iyi ayrılmaktadır.



* + - **Tanısal Üstünlük Oranı**: tahmin edilen pozitif sınıfın üstünlüğünün, negatif sınıfa üstünlüğüne oranıdır.



* + **Alıcı İşlem Karakteristikleri Eğrileri**(ROC): ROC eğrisi altında kalan alan, bize birden fazla sınıflandırıcıyı karşılatırma imkanı veren bir ölçüttür. Bu alanın değeri 1’e ne kadar yakın olursa, o kadar doğru sınıflandırma yapıldığı anlamına gelir.

**Yapay sinir ağları (YSA)**: Bilginin, verilerden öğrenme yolu ile elde edildiği ve işlem elamanlarının bağlantı değerlerinde (ağırlık - bağlantı gücü) saklandığı dolayısıyla dağıtık bir hafızanın söz konusu olduğu da ortak noktaları oluşturmaktadır.

**Yapay Sinir Ağlarının Görevi**; insanlar tarafından gerçekleştirilmiş örnekleri (gerçek beyin fonksiyonlarının ürünü olan örnekleri) kullanarak olayları öğrenebilen, çevreden gelen olaylara karşı nasıl tepkiler üretileceğini belirleyebilmektedir. İnsan beyninin fonksiyonel özelliklerine benzer şekilde;

* Öğrenme
* İlişkilendirme
* Sınıflandırma
* Genelleme
* Özellik Belirleme
* Optimizasyon

Gibi konularda başarılı bir şekilde uygulanmaktadırlar.

Yapay Sinir Ağlarının Genel Özellikleri;

* Yapay sinir ağları makine öğrenmesi gerçekleştirirler.
* Bilginin saklanması
* Yapay sinir ağlarının güvenle çalıştırılabilmesi için önce eğitilmeleri ve performanslarının test edilmesi gerekmektedir.
* Görülmemiş örnekler hakkında bilgi üretebilirler.
* Algılamaya yönelik olaylarda kullanılabilirler.
* Örüntü ilişkilendirme, sınıflandırma ve tamamlama yapabilirler.
* Kendi kendini organize etme ve öğrenebilme yetenekleri vardır.
* Eksik bilgi ile çalışabilmektedirler.
* Hata toleransına sahiptirler.
* Belirsiz, tam olmayan bilgileri işleyebilmektedirler.
* Dağıtık belleğe sahiptirler.
* Sadece nümerik bilgiler ile çalışabilmektedirler.

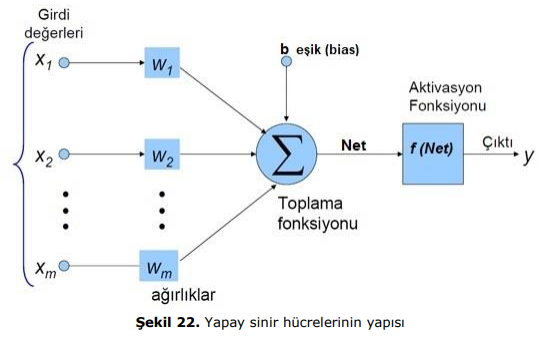
**Yapay Sinir Ağlarının Uygulama Alanları**

Başka çözüm yolları olmasına rağmen yapay sinir ağlarının daha kolay ve daha etkin çözümler üretilebilmesinin sağlanması gerekir.

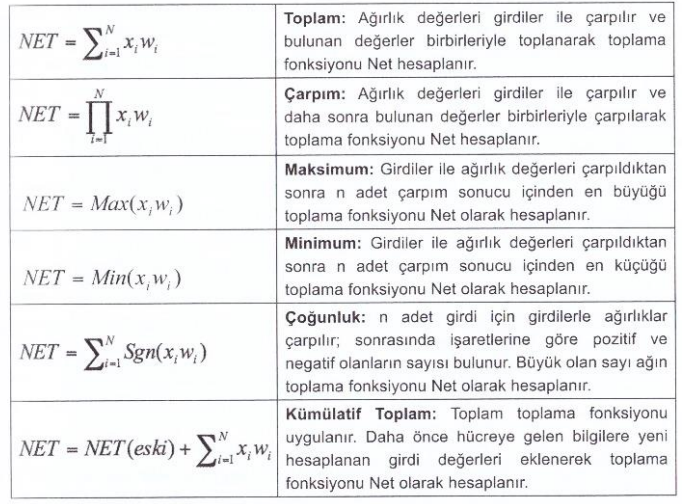
* Olasılıksal fonksiyon kestirimleri
* Sınıflandırma
* İlişkilendirme veya örüntü eşleştirme
* Zaman serileri analizleri
* Sinyal filtreleme
* Veri sıkıştırma
* Örüntü tanıma
* Doğrusal olmayan sinyal işleme
* Doğrusal olmayan sistem modelleme
* Optimizasyon
* Zeki ve doğrusal olmayan kontrol
* Veri madenciliği
* Optik karakter tanıma ve çek okuma
* Bankalardan kredi isteyen müracaatları değerlendirme
* Ürünün pazardaki performansının tahmin etme
* Kredi kartı hilelerini saptama
* Zeki araçlar ve robotlar için optimal rota belirleme
* Güvenlik sistemlerinde konuşma ve parmak izi tanıma
* Robot hareket mekanizmalarının kontrol edilmesi
* Mekanik parçaların ömürlerinin ve kırılmalarının tahmin edilmesi
* Kalite kontrolü
* İş çizelgesi ve iş sıralaması
* İletişim kanallarındaki geçersiz ekoların filtrelenmesi
* İletişim kanallarındaki trafik yoğunluğunu kontrol etme ve anahtarlama
* Radar ve sonar sinyalleri sınıflandırma 64
* Üretim planlama ve çizelgeleme
* Kan hücreleri reaksiyonları ve kan analizlerini sınıflandırma
* Kanserin saptanması ve kalp krizlerinin tedavisi
* Beyin modellenmesi çalışmaları
* Konuşma ve yüz tanıma
* Sözcük tanıma, dil tercüme

**Yapay Sinir Ağlarının Yapısı ve Temel Elemanlar:**

* **Biyolojik Sinir Hücreleri**: Beynimizde 1011 (~100 milyar) adet sinir hücresi (nöron), her bir nöronun yaklaşık 10.000 farklı nörona bağlantısı ve bunlarında toplamda 100 trilyon civarında bağlantının olduğu söylenmektedir
* **Yapay Sinir Hücresi**: Yapay sinir hücreleri mühendislik biliminde işlem elamanları olarak da adlandırılmaktadır. Her işlem elemanının 5 temel bileşeni vardır.

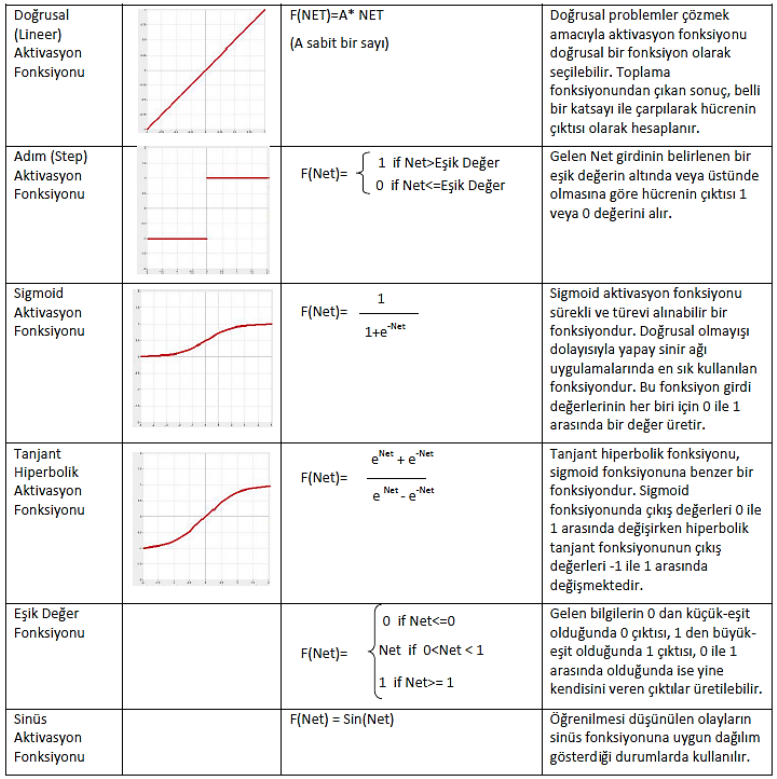


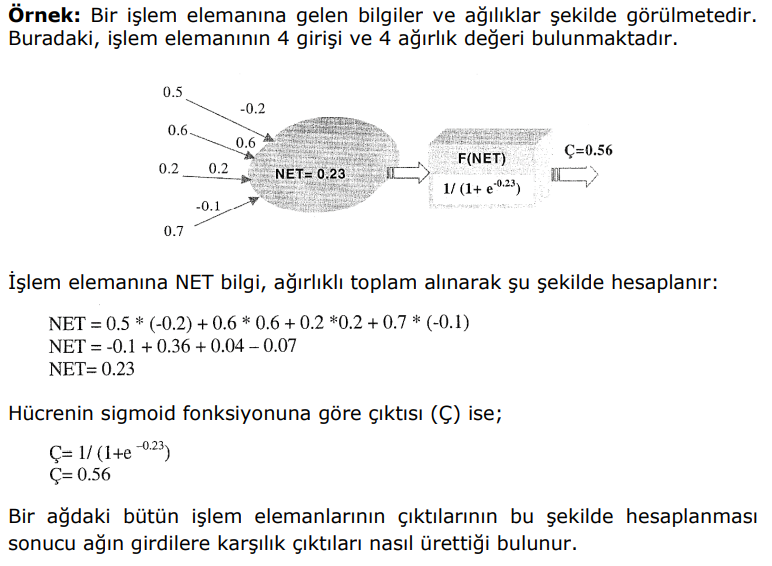
* + **Girdiler**: Bir yapay sinir hücresine dış dünyadan gelen bilgilerdir.
  + **Ağırlıklar**: Ağırlıklar bir yapay hücreye gelen bilginin önemini ve hücre üzerindeki etkisini gösterir.
  + **Toplama fonksiyonu**: Bu fonksiyon, bir hücreye gelen net girdiyi hesaplar. Bunun için değişik fonksiyonlar kullanılmaktadır. En yaygın olanı ağırlıklı toplamı bulmaktır.



* + **Aktivasyon (Etkinleştirme) Fonksiyonu**: aktivasyon fonksiyonunun amacı NET değerini belirli bir aralığa çekmektir. Özellikle derin öğrenme ile kullanıma çıkan aktivasyon fonksiyonlarıda bulunmaktadır:
    - **ReLU**: Düzleştirilmiş Doğrusal Birimler (Rectified Linear Units - ReLU). Bir ReLU birimi eğer girdi (x, net) sıfırdan küçüke 0, aksi durumda girdinin değerini alır. ReLU (x) = max(0, x)
    - **Softmax**: Girdinin belirli sınıfa ait olma olasılığını 0 ve 1 aralığında değerler üreterek belirlenmesini sağlamaktadır. Yani olasılıksal bir yorumlama gerçekleştirir.





* + **Hücrenin çıktısı**: Hücrenin çıktısı: Aktivasyon fonksiyonu tarafından belirlenen çıktı değeridir. Üretilen çıktı dış dünyaya veya başka bir hücreye gönderilir.
* **Yapay Sinir Hücresinin Çalışma Prensibi**: 
* Yapay Sinir Ağlarında Öğrenme
  + Adaptif Öğrenme ve Test Etme: Ağın eğitimi tamamlandıktan sonra öğrenip öğrenmediğini ölçmek için yapılan denemelere ise ağın test edilmesi denmektedir. Test etmek için ağın öğrenme sırasında görmediği örnekler kullanılır. Eğitimde kullanılan örnek setine eğitim seti, test için kullanılan sete ise test seti adı verilmektedir.
* Yapay Sinir Ağlarından En Çok Kullanılan Modeller:
  + Algılayıcılar
  + Çok Katmanlı Algılayıcılar - MLP (hatayı geriye yayma modelleri)
  + Vektör Kuantizasyon Modelleri (LVQ)
  + Kendi Kendini Organize Eden Model (SOM)
  + Adaptif Rezonans Teorisi Modelleri (ART)
  + Hopfield Ağları
  + Counterpropagation Ağı
  + Neocognitron Ağı
  + Boltzman Makinesi
  + Olasıklıksal Ağlar (PNN)
  + Elman Ağı
  + Radyal Temelli Ağlar (RBN)

**Yapay Sinir Ağ Modellerinde Öğrenme**

* **Tek Katmanlı Algılayıcı (TKA) - Perceptron:**

Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusal fonksiyondur. Yani ağa gösterilen örnekler iki sınıf arasında paylaştırılarak iki sınıfı birbirinden ayıran doğru bulunmaya çalışılır. Onun için eşik değer fonksiyonu kullanılmaktadır.

* + Ağın beklenen çıktısı 0 değeridir. Fakat NET girdi eşik değerinin üstündedir. Yani ağın gerçekleşen çıktısı 1 değeridir. Bu durumda ağırlık değerleri azaltılmaktadır. Ağırlıkların değişim oranı girdi değerlerinin belirli bir oranı kadardır.

Yani; Wn = Wo-λX olur.

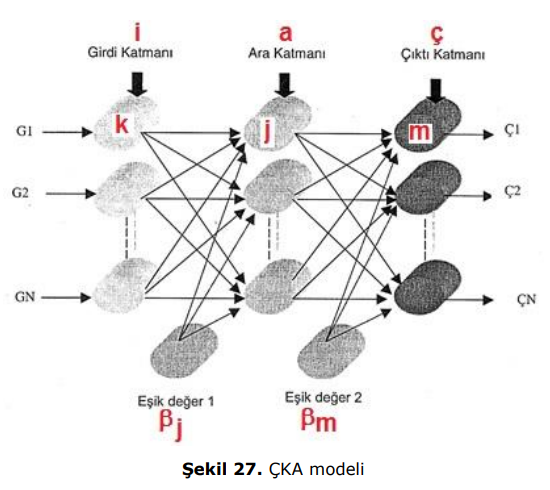
Burada λ öğrenme katsayısıdır.

* + Beklenen çıktının 1 olması ve ağın gerçek çıktısının 0 olması durumudur. Yani NET girdi eşik değerinin altındadır. Bu durumda ağırlıkların değerinin artırılması gerekmektedir.

Yani; Wn = Wo +λX

* + Örneğin; NET= w1\*x1 +w2\*x2, NET > Ф olduğundan gerçekleşen çıktı Ç = 1 ve Ç≠B2
* **Eğiticili YSA Modeli – Çok Katmanlı Algılayıcı**

Bu tür olayların öğrenilmesi için daha gelişmiş modellere ihtiyaç vardır. Bu bölümde anlatılan Çok Katmanlı Algılayıcı (ÇKA) modeli bunlardan birisidir.

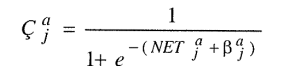


**ÇKA Ağının Öğrenme Algoritması**

ÇKA ağının öğrenme kuralı en küçük kareler yöntemine dayalı Delta Öğrenme Kuralının genelleştirilmiş halidir. O nedenle öğrenme kuralına Genelleştirilmiş Delta Kuralı da denmektedir.

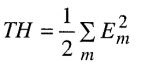
* 1. aşama - ileri doğru hesaplama: Ağın çıktısını hesaplanır.
* 2. aşama - geriye doğru hesaplama: Ağın ağırlıkları değiştirilir.

**İleri Doğru Hesaplama:**

* Giriş Katmanında; yani, girdiler olduğu gibi ara katmana gider
* Ara Katmanda; yani, ara katman elemanın çıktısı bu net girdinin aktivasyon fonksiyonundan (genellikle sigmoid fonksiyonundan) geçirilmesiyle hesaplanır.
* Çıktı Katmanında, , Yani daha önce belirtilen aktivasyon fonksiyonlardan herhangi birisini burada kullanılmalıdır.
* Ara katmanın bütün işlem elemanları ve çıktı katmanın işlem elemanlarının çıktıları aynı şekilde kendilerine gelen NET girdinin hesaplanması ve sigmoidi fonksiyonundan geçirilmesi sonucu belirlenirler.

**Geriye Doğru Hesaplama:**

* Çıktı katmanındaki m. işlem elemanı için oluşan hata (Em): Em = Bm - Çm
* Toplam Hatayı (TH) bulmak için her bir çıktı için oluşan hataların toplanması gerekir. Bazı hata değerleri negatif olacağından toplamın sıfır olmasını önlemek amacı ile ağırlıkların kareleri hesaplanarak sonucun karekökü alınır. ÇKA ağının eğitilmesindeki amaç bu hatayı en azlamaktır.

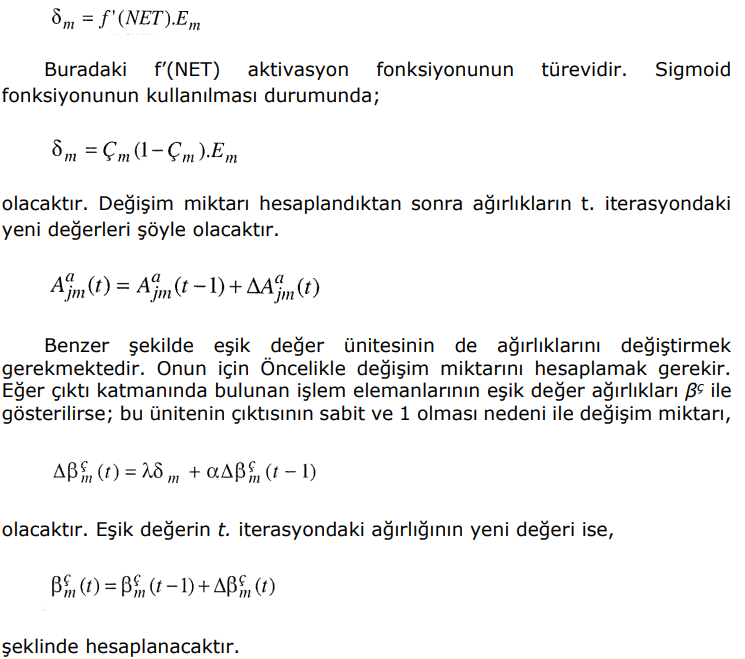


Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi:

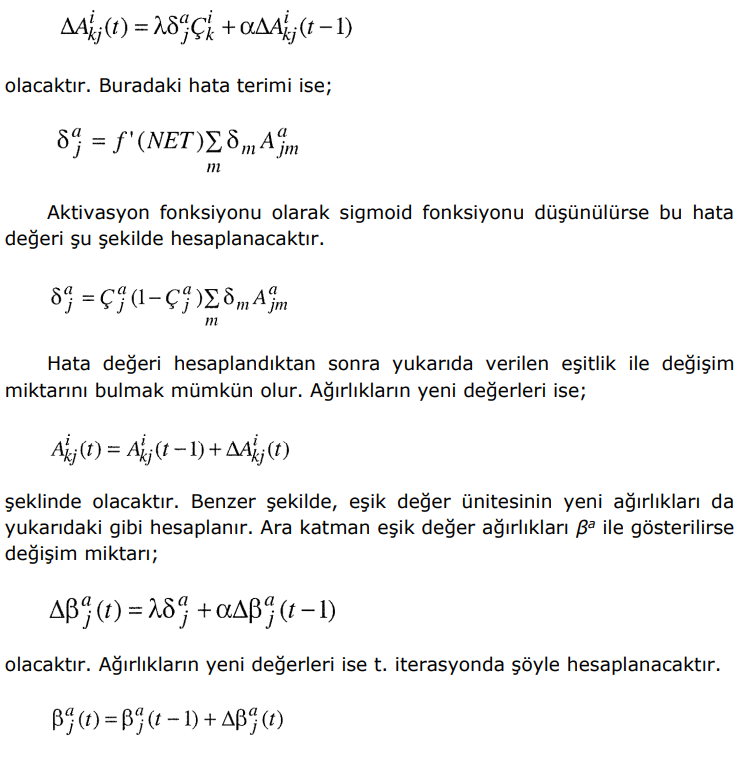
* Her bir Ağırlığın Değişim Miktarı Hesaplanır



* Burada λ öğrenme katsayısını, α momentum katsayısını göstermektedir. Öğrenme katsayısı ağırlıkların değişim miktarını, momentum katsayısı ise ÇKA ağının öğrenmesi esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlarlar. Eşitlikteki δm ise m. çıktı ünitesinin hatasını göstermektedir. Şu şekilde hesaplanır.



Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi;



**ÇKA Ağının Çalışma Prosedürü:**

* **Örneklerin toplanması**: Ağın çözmesi 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.
* **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 işlem elemanı ve kaç tane çıktı elemanı olması gerektiği bu adımda belirlenmektedir.
* **Öğrenme Parametrelerini Belirlenmesi**: Ağın öğrenme katsayısı, işlem elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
* **Ağırlıkların Başlangıç Değerlerinin Atanması**: İşlem elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin atanması yapılır. Başlangıçta genellikle rasgele değerler atanır. Daha sonra ağ uygun değerleri öğrenme sırasında kendisi belirler.
* **Öğrenme Setinden Örneklerin Seçilmesi ve Ağa Gösterilmesi**: Ağın öğrenmeye başlaması ve yukarıda anlatılan öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler (Girdi/Çıktı değerleri) belirli bir düzeneğe göre gösterilir.
* **Öğrenme Sırasında İleri Hesaplamaların Yapılması**: Yukarıda anlatıldığı şekilde sunulan girdi için ağın çıktı değerleri hesaplanır.
* **Gerçekleşen Çıktının Beklenen Çıktı ile Karşılaştırılması**: Ağın ürettiği hata değerleri bu adımda hesaplanır.
* **Ağırlıkların Değiştirilmesi**: Yukarıda anlatıldığı gibi geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.

**ÇKA Ağının Eğitilmesi**:

* Problem eğitilirken bulunan örnekler problem uzayını %100 temsil etmeyebilir.
* Oluşturulan ÇKA ağı için doğru parametreler seçilmemiş olabilir.
* Ağın ağırlıkları başlangıçta tam istenildiği şekilde belirlenmemiş olabilir.
* Ağın topolojisi yetersiz seçilmiş olabilir.

Bu durumda ağın olayı öğrenmesi için bazı değişiklikler yapılarak yeniden eğitilmesi gerekir. Bu değişiklikler arasında şunlar sayılabilir:

* Başka başlangıç değerlerinin kullanılabilir.
* Topolojide değişiklikler yapılabilir (ara katman sayısını artırmak, işlem elemanı sayısını artırmak veya azaltmak gibi),
* Parametrelerde değişiklik yapılabilir (aktivasyon fonksiyonlarının başka seçilmesi, öğrenme ve momentum katsayılarının değiştirilmesi gibi)
* Problemin gösterimi ve örneklerin formülasyonu değiştirilerek yeni örnek seti oluşturulabilir
* Öğrenme setindeki örneklerin sayısı artırılabilir veya azaltılabilir
* Öğrenme sürecinde örneklerin ağa sunulması

**ÇKA Ağının Öğrenmek Yerine Ezberlemesi:** Bazı durumlarda eğitilen ÇKA ağı eğitim setindeki bütün örneklere %100 doğru cevap üretmesine rağmen test setindeki örneklere doğru cevaplar üretememektedir.

**Bir ÇKA Ağının Oluşturulmasında Dikkat Edilmesi Gereken Bazı Önemli Noktalar**

Yapılan araştırmalar ve tecrübeler bir ÇKA ağının performansını etkileyen unsurların şunlar olduğunu göstermektedir.

* Örnekleri seçilmesi
* Girdi ve çıktıların ağa gösterimi
* Girdilerin nümerik gösterimi
* Çıktıların nümerik gösterimi
* Başlangıç değerlerinin atanması
* Öğrenme ve momentum katsayılarının belirlenmesi
* Örnekleri ağa sunulması
* Ağırlıkların değiştirilme zamanları
* Girdi ve çıktıların ölçeklendirilmesi
* Durdurma Kriterinin belirlenmesi
* Ara katmanların ve her katmandaki işlem elemanlarının sayısının belirlenmesi
* Ağların büyütülmesi veya budanması

**Eğiticisiz YSA Modeli – ART Ağları**

Hafıza (Bellek) Kavramı; ART ağlarında öğrenme doğru bilgilerin belirlenerek hafızaya alınması anlamına gelmektedir. Öğrenme sırasında kullanılan örneklerden öğrenilen bilgilere dayanarak daha sonra görülmemiş örnekler hakkında yorumlar yapılabilmektedir.

* **Kısa Dönemli Hafıza (KDH):** Bilgilerin geçici olarak tutulduğu ve zaman içerisinde yok olduğu ve yerlerine başka bilgilerin saklandığı hafızadır.
* **Uzun Dönemli Hafıza (UDH):** Bilgilerin sürekli tutulduğu ve kolay kolay unutulmadığı hafızadır. Bilginin silinmesi için çok uzun zamanın geçmesi gerekebilmektedir.
* **NOT:** ART ağlarında bilgiler hem kısa dönemli hem de uzun dönemli hafızada saklanmaktadır.

Bu modelin 3 temel özelliği vardır.

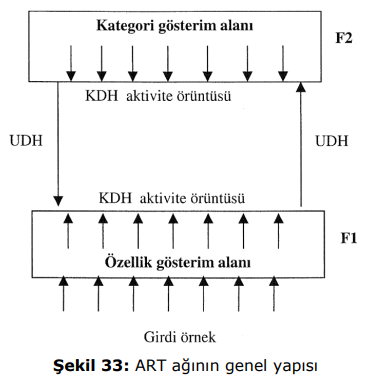
* **Normalizasyon**: Bu, özellikle biyolojik sistemlerin çevredeki büyük değişikliklere karşı adaptif olduklarının durumu göstermektedir. Örneğin insanın çok fazla gürültülü bir ortamda bir süre sonra gürültüden rahatsız olmaması sisteme adapte olunduğunu ve çevredeki olayların normalize edildiğini göstermektedir.
* **Ayrıştırabilme**: İnsanın karar verebilmesinde ve olayları yorumlayabilmesinde çevredeki olaylar arasında var olan fakat görülmesi zor farklılıkları ayrıştırmak çok önemlidir.
* **Ayrıntıların Saklandığı Kısa Dönemli Hafıza**: Belirlenen farklılıklar ve çevresel olaylar davranışlara neden olmadan önce hafızada saklanmakta ve daha sonra eyleme dönüşmektedir.

**ART Ağlarının Diğer Yapay Sinir Ağlarından Farkları**

* ART ağları gerçek zamanlı olarak oldukça hızlı ve kararlı bir şekilde öğrenme yeteneklerine sahiptirler.
* Gerçek zamanda ortam genel olarak durağan değildir. Olayların oluşumu her an beklenmedik olaylar ile değişebilmektedir. Bununda ötesinde gerçek zamanlı olaylar sürekli devam etmektedir.
* ART ağları beklenen çıktıları bir öğretmenden almak yerine kendi kendine öğrenmeye çalışır.
* ART ağları ağa sunulan farklı nitelikteki ve değişik durumlardaki örnekler karşısında kendi kendilerine kararlı (stabil) bir yapı oluşturabilirler.
* ART ağları çevredeki olayları sürekli öğrenmeye devam eder.
* ART ağlarında hem aşağıdan yukarı hem de yukarıdan aşağıya ağırlık değerleri vardır.

**ART Ağlarının Yapısı**

Adaptif Rezonans Teorisi (ART) ağları genel olarak iki katmandan oluşmaktadır. Bu katmanlar F1 ve F2 olarak isimlendirilmiştir. F1 katmanı girdinin özelliklerini gösterirken F2 katmanı kategorileri (ayrıştırılmış sınıfları) göstermektedir. Bu iki katman birbirlerine UDH ile bağlanmaktadırlar. Girdi bilgileri F1 katmanından alınır ve sınıflandırma ise F2 katmanında yapılır.



ART Ağlarının Çalışma Prensibi;

* Aşağıdan yukarı (F1 den F2'ye) bilgi işleme
* Yukarıdan aşağı (F2 den F1 'e) bilgi işleme

**ART Ağının Eğitilmesi ve Öğrenmesi:**

ART konusunda yukarıdaki açıklamaları dikkate alarak öğrenme olayı basitleştirilmiş ve aşağıda adım adım açıklanan öğrenme kuralı geliştirilmiştir:

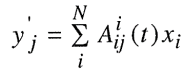
1. Adım: Ağırlıklara başlangıç değerleri atanır;
   1. F2 ve F1 arasındaki geriye doğru ağırlıkların bütün değerleri başlangıçta 1 değerini alır.



* 1. F1 ve F2 arasındaki ileri doğru ağırlıkların başlangıç değerleri, F1 katmanındaki işlem elemanı sayısı n olmak üzere ise şu şekilde atanmaktadır.



1. Adım: Benzerlik katsayısının (p) değeri belirlenir; Benzerlik katsayısı p ile gösterilmekte olup 0 ile 1 arasında bir sayıdır. Bu katsayı iki vektörün aynı sınıfın elemanı sayılabilmesi için birbirlerine ne kadar benzemesi gerektiğini belirler.
2. Adım: Girdi setinden bir örnek ağa gösterilir; Girdi setindeki örnek X (x1,x2,.. xn) vektörü olarak (her bir elemanı xi olarak tanımlanmış şekilde) ağa gösterilir.
3. Adım: F2 katmanındaki işlem elemanlarının çıktılarının hesaplanması; F2 katmanındaki her işlem elemanının çıktı değeri şu şekilde hesaplanmaktadır:



1. Kazanan elemanın seçilmesi; Kazanan eleman en büyük (maksimum) çıktıya sahip işlem elemanıdır. Bu elemanın sahip olduğu ağırlık vektörüne en uygun sınıf (kategori) gösterim vektörü denmektedir. Bunun k. işlem elemanı olduğu varsayılırsa kazanan elemanın çıktısı



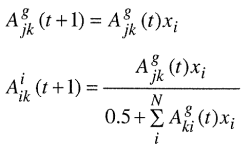
1. Adım: Uygunluk testinin yapılması; Burada kazanan elemanın ağırlık vektörünün ile girdi vektörünü temsil edip edemeyeceğine karar verilmektedir. Bunun için önce girdi vektöründe bulunan 1 sayısı (s1) belirlenir.



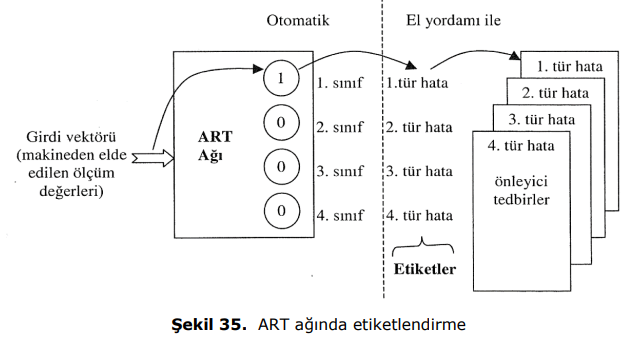
Daha sonra kategori gösterimi vektörü (kazanan elemanın ağırlık vektörü) ile girdi vektörünün uyuştuğu 1 sayısı (s2) bulunur. Bunu veren formül ise şöyledir:



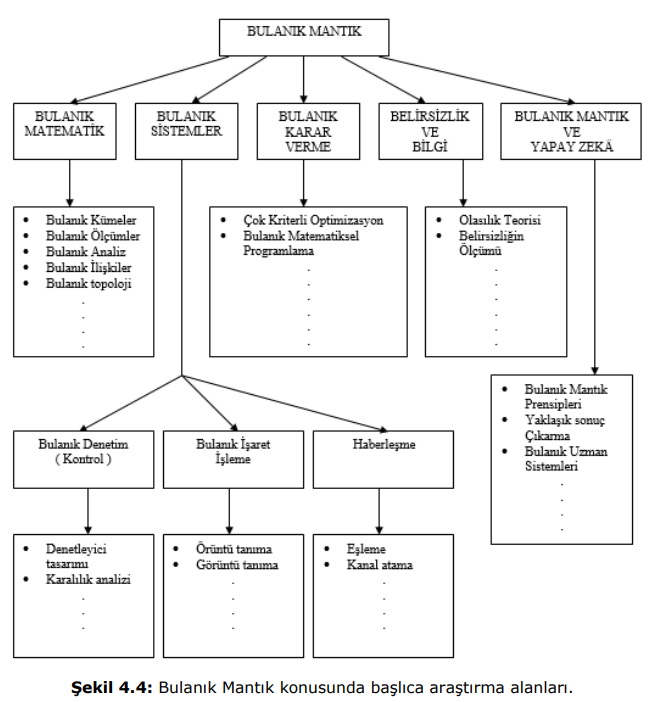
Eğer, s2/s1 ≥ p ise o zaman iki vektör birbirinin benzeri kabul edilir. Yani kategori gösterim vektörü girdi vektörünü temsil edebiliyor demektir. Bu durumda ağırlıklar şu şekilde değiştirilir.



**ART Ağlarında Etiketlendirme**; Grupların neyi temsil ettikleri öğrenme bittikten sonra tasarımcı tarafından belirlenir. Buna sınıfın etiketlenmesi denir.



**Bulanık Mantık**



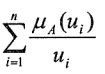
Bulanık mantık sistemlerinin avantajlarını kısaca özetlemek gerekirse;

* Problem eğer basit matematiksel durumlar ile modellenebiliyorsa geleneksel yöntemler yeterli olacaktır. Fakat karmaşıklığı olan bir problem ile karşı karşıya gelindiğinde bulanık mantık sistemin analizini daha iyi yapabilmektedir.
* Bir sisteme gelen sinyallere önişlem uygulanmasından ve geniş skalaya sahip değerleri az sayıda tanımlanan üyelik fonksiyonları ile modelleyebilmesinden ötürü bulanık mantık metodu kullanılarak problem çok daha hızlı şekilde çözümlenir.
* Tanımlanacak olan az sayıda bulanık küme ve üyelik fonksiyonları üzerinde kural sayısı da az olarak belirleneceğinden ötürü bulanık mantık ile sonuca ulaşmak daha hızlı olmaktadır.
* Bulanık mantık ile oluşturulan yazılımların boyutlarının daha az olmasından ötürü, dış bellek kullanımına gerek kalmamaktadır.
* Bulanık mantık, doğrudan uzman bilgi ve tecrübelerinden yararlanmaya olanak sağlar.

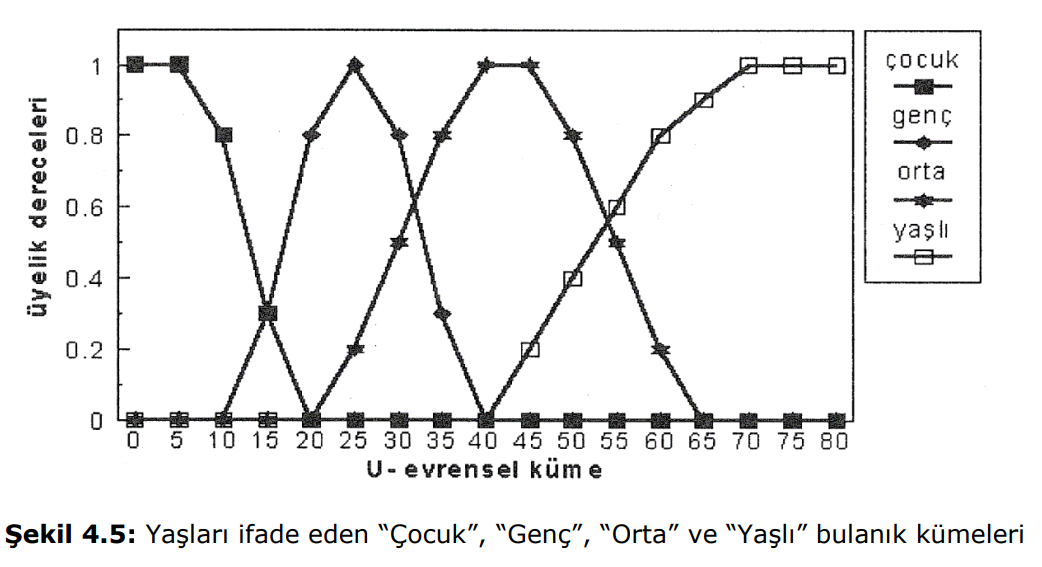
Bulanık mantık sistemlerinin dezavantajlarını kısaca özetlemek gerekirse;

* Bulanık denetimde kullanılan kurallar uzman bilgi ve tecrübesine bağlıdır.
* Üyelik fonksiyonlarının seçimi için belirli bir yöntem yoktur. En çok kullanılan ve en uygun yöntem, deneme yanılmadır.
* Bulanık mantık yöntemine sahip bir sistemin önceden nasıl cevap vereceğinin kestirimi yapılamaz. Ayrıca sistemin kararlılık analizi de yapılamaz. Fakat benzetim çalışması bu durumlar için yapılabilir.

**Zade’nin Açılım İlkesi**



Örnek: Evrensel A kümesi 0-100 yaş arası kişileri ifade etsin. Yaşları ifade eden “Çocuk”, “Genç”, “Orta” ve “Yaşlı” bulanık kümelere üyelik fonksiyonları Şekil 4.5'deki gibi tanımlanabilir



5 senelik bir ara ile aşağıdaki bulanık değerleri elde ederiz(0(Sıfır) üyelik değerleri yazılmamıştır):

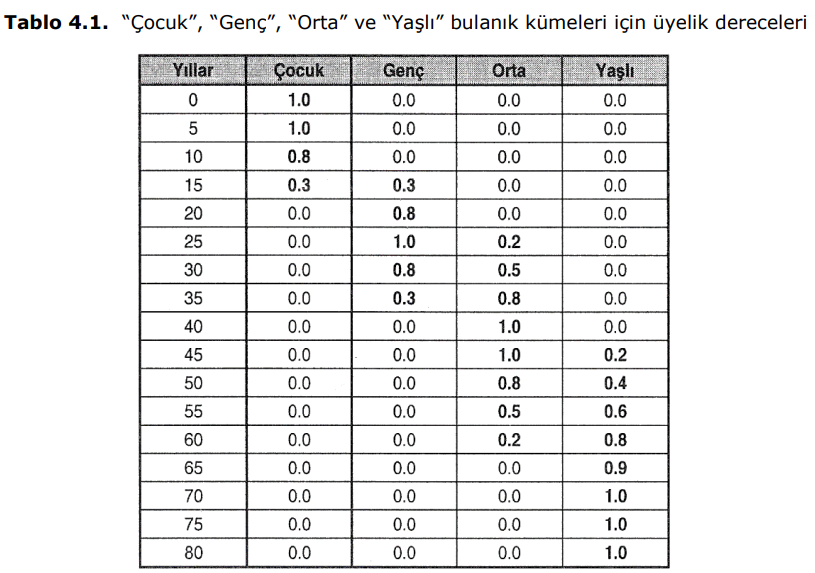
Çocuk = μçocuk = 1/0 + 1/5 + 0.8/10 + 0.3/15

Genç = μgenç = 0.3/15 + 0.8/20 + 1/25 + 0.8/30 + 0.3/35

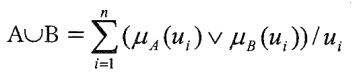
Orta = μorta = 0.2/25 + 0.5/30 + 0.8/35 + 1/40 + 1/45+0.8/50+0.5/55 + 0.2/60

Yaşlı = μyaşlı = 0.2/45 + 0.4/50 + 0.6/55 + 0.8/60 + 0.9/65 + 1/70 + 1/75 + 1/80

Bulunan üyelik derecelerini ifade eden tablo aşağıda gösterilmiştir:

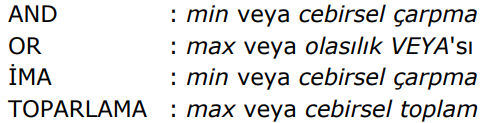


Bulanık Kümelerle ilgili işlem yaparken 3 Temel işlem bulunur;

* Bulanık Kümelerin Birleşmesi
  + 
  + 
  + Tabloya Göre Birleşim. Genç ve Orta İçin
    - (genç U orta) = μ(genç U orta) = 0.3/15 + 0.8/20 + 1/25 + 0.8/30 + 0.8/35 + 1/40 + 1/45 + 0.8/50 + 0.5/55 + 0.2/60
* Bulanık Kümelerin Kesişmesi
  + 
  + 
  + Tabloya Göre Kesişim. Orta ve Yaşlı İçin
    - (orta ∩ yaşlı) = μ(orta ∩ yaşlı) = 0.2/45 + 0.4/50 + 0.5/55 + 0.2/60
* Bulanık Kümenin Tersi
  + 
  + Tabloya Göre Ters Alma. Çocuk için;
    - ~(çocuk) = 0.2/10 + 0.7/15 + 1/20 + 1/25 + 1/30 + 1/35 + 1/40 + 1/45 + 1/50 + 1/55 + 1/60 + 1/65 + 1/70 + 1/75 + 1/80

**Çıkarım Mekanizması**

Çıkarım mekanizmasında, üç ana işlem türü mevcuttur. AND, OR, İMA ve TOPARLAMA işlemleri için en sık kullanılan operatörler aşağıdaki gibidir:

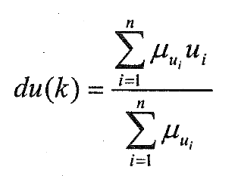


**Çıkış Birimi veya Durultucu**

Genellikle üyelik fonksiyonlarının birleşiminin, ağırlıklarının hesaplanmasına dayalıdır. Örnek olarak aşağıdaki durulama yöntemleri gösterilebilir;

* Maksimum Üyelik İlkesi
* Ağırlık Merkezi Yöntemi(En Çok Kullanılan)
* Ağırlıklı Ortalama Yöntemi
* Ortalama Maksimum Üyelik Yöntemi

Ağırlık merkezi durulama yönteminin karmaşıklığı, çıkış üyelik fonksiyonlarının biçimine bağlıdır ve kesin değerlerin elde edilmesinde aşağıdaki bağıntıdan faydalanılır.



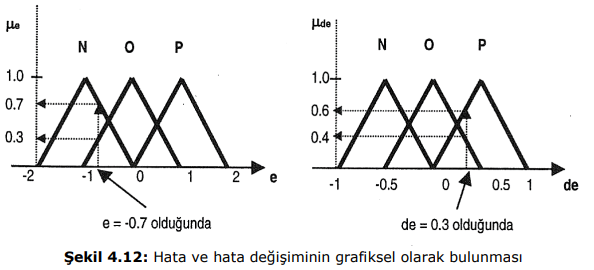
**Mamdani Tipi Bulanık Modelleme**

**Örnek**:

Bir kontrol sistemi olarak bulanık mantığın kullanıldığı ve ayarlanması gereken parametrenin hatasının (e) ve bu hatanın türevinin (de) sisteme giriş olarak uygulandığı varsayılsın. Burada hata, ayarlanan parametrenin istenenle gerçek değerleri arasındaki farkı, hatanın türevi ise bu hatanın değişimini ifade etmektedir. e(k)= -0.7 ve de(k) = 0.3 için kontrol işareti du(k) bulunuz?

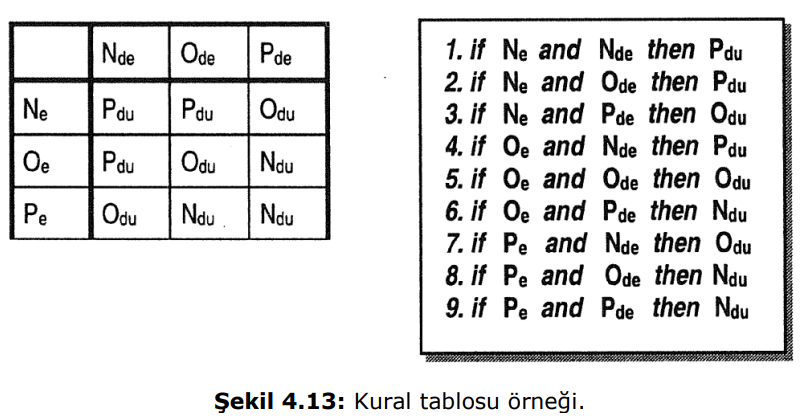
**Çözüm**

1. **Adım;**
   1. **Bulanıklaştırıcı**

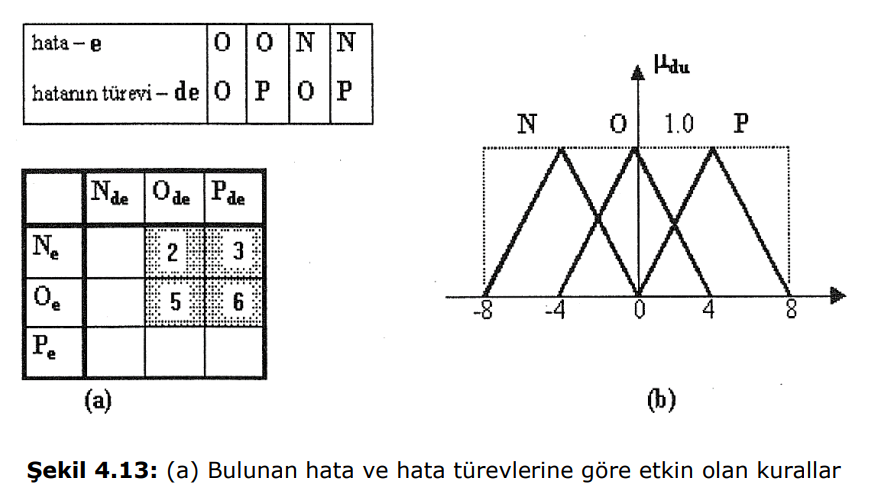


* Burada hata ve türevi için uygun olarak, e = -0.7 ve de = 0.3 değerleri ele alınmıştır.
* Bu değerler için bulanıklaştırma sonucunda bulunan üyelik değerleri:
* O üyelik fonksiyonu için μe=0.3 ve μde=0.4,
* N ve P üyelik fonksiyonları için ise μe =0.7 ve μde =0.6 olmaktadır.

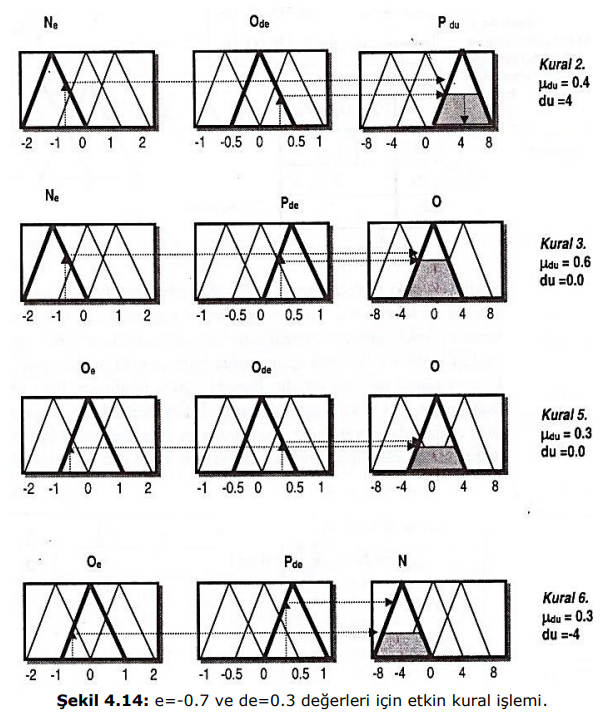
1. **Adım;**
   1. **Bulanık Çıkarım**
      1. Kural tablosunun belirlenmesini gerektirmektedir. e(k) ve de(k) için uygun olarak üçer üyelik fonksiyonu iştirak ettiğinden kurallar sayısı 9 olacaktır. Mümkün olabilecek kurallar tablosu ve kuralların sözel ifadesi IF ... THEN ... ELSE biçiminde aşağıda verilmektedir. IF (hata ◊ X) ve (hatanın türevi◊Y) THEN (du = Z)



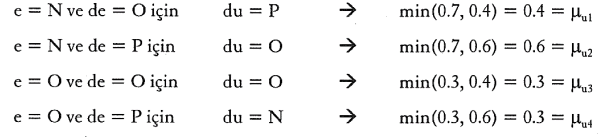
Burada du(k) değerleri, giriş değerlerine göre karar tablosundan elde edilen bulanık çıkarımlardır. Verilmiş hata ve hata türevi değerlerine göre bulunan üyelik değerlerinin dört farklı kombinasyonu olduğu görülmektedir. Bu durumda karar tablosunun dört kuralı etkin olacaktır:



Şekil 4.13'deki tablolardan da anlaşıldığı gibi, burada hata negatif ve hatanın değişimi yaklaşık sıfır ise 2. kuralın uygulanması gerektiği, yani uygulanacak kontrol işaretinin pozitif olması istenmektedir. Yine de bu tabloya göre, hata ve hatanın türevi yaklaşık sıfır ise, kontrol işareti sıfır; hata negatif ve hatanın değişimi pozitif ise kontrol işareti sıfır olmaktadır. Burada uzman tarafından hata sıfır olduğunda, hata değişiminin sıfır veya pozitif olmasına ilişkin uygun kontrol işaretinin sıfır veya negatif olması da saptanmıştır. Bu kuralların çıkış değeri, üyelik fonksiyonlarının orta değeri olarak ele alınmaktadır. Üstteki dört kural için çıkış değerleri: u1 ◊ P = 4, u2, u3 ◊ O = 0, u4 ◊ N =-4 olacaktır.



Açıklamalara dayanarak bu dört durum için elde edilen üyelik değerleri, minimum operatörüne uygulanır.

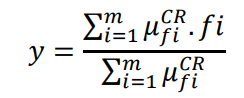


1. **Adım;**
   1. Durulaştırıcı: Ağırlık merkezi durulama yöntemine göre çıkışın hesaplanması:
      1. du(k) = (0.4\*(4) + 0.6\*(0) + 0.3\*(0) +0.3\*(-4)) / (0.4 + 0.6 + 0.3 + 0.3) = 0.25

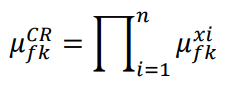
**Sugeno Tipi Bulanık Modelleme**

* Kural-1: Eğer x1 = A1 ve y1 = B1, ise f1(x,y)=p1\*x + q1\*y + r1
* Kural-2: Eğer x2 = A2 ve y2 = B2, ise f2(x,y)=p2\*x + q2\*y + r2

Sugeno bulanık sistem çıkışının (y) hesaplanması aşağıdaki denklemde verilmiştir.



Verilen denklem, fi'lerin ağırlıklandırılmış bir ortalamasıdır. Denklem 𝜇𝑓𝑖 𝐶𝑅, her çıkış değişkeni için üyelik fonksiyonlarının keskinlik (ağırlık) derecesi olup, formülasyonu aşağıda verilmiştir.



Eşitlikte: 𝜇𝑓𝑘 𝑥𝑖 , fk çıkış değişkenindeki bulanık kümelerin üyelik fonksiyon değerlerini ifade etmektedir.

**Avantajları**

* Matematiksel yönü kuvvetli olduğu için hesaplama için uygundur.
* Matematiksel denklemler ile kuralların çıkışları hesaplandığından durulaştırmaya gerek yoktur.
* Doğrusal olmayan sistemlerin çözümü için doğrusal yöntemler kullanılabilir.
* Çıkışın sürekliliği mevcuttur.
* Matematiksel analiz durumlarına uygundur.

**Dezavantajlari**

* İleri derecedeki modelleri karmaşık bir yapıdadır.
* Mamdani gibi kurallar, uzman yardımı ile alınmadığından dolayı insan sezgilerine çok benzer değildir.

**Örnek**:

Sugeno tipi bir bulanık sistemde;

Kurallar:

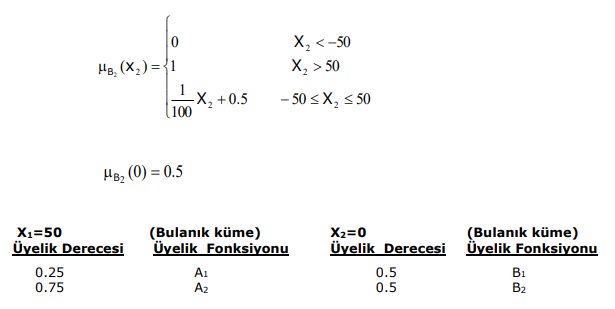
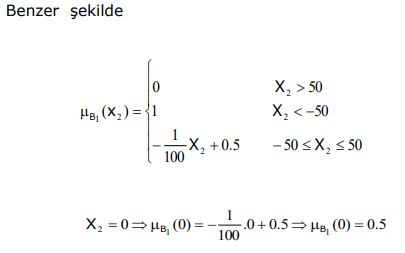
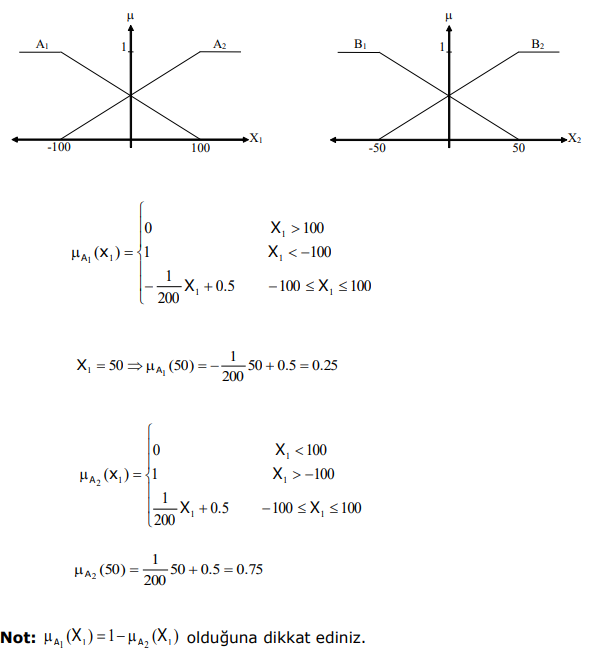
1. IF (X1 is A1 AND X2 is B1) THEN y = f1(X1,X2)
2. IF (X1 is A1 AND X2 is B2) THEN y = f2(X1,X2)
3. IF (X1 is A2 AND X2 is B1) THEN y = f3(X1,X2)
4. IF (X1 is A2 AND X2 is B2) THEN y = f4(X1,X2)

Çıkış denklemleri:

* f1(X1,X2)=3X1-2X2 +1
* f2(X1,X2)=X1+X2
* f3(X1,X2)=2X1+X2–2
* f4(X1,X2)=X1-3X2 +5

olarak verilmiştir. AND operatörü için cebirsel çarpma seçildiğine göre X1=50, X2=0 değerleri için bulanık sistemin çıkışı y’yi hesaplayınız.

**Çözüm**



1. Adım

* Bulanıklaştırıcıda yapılan hesaplar (Üyelik Dereceleri)
* Yukarıdaki Üyelik Fonskiyonlarına Göre;
  + X1 = 50 için; μA1(50) = - 50/200 + 0.5 = - 0.25 + 0.5 = 0.25
  + X1 = 50 için; μA2(50) = 50/200 + 0.5 = 0.25 + 0.5 = 0.75
  + X2 = 0 için ; μB1(0) = - 0 /100 + 0.5 = 0 + 0.5 = 0.5
  + X2 = 0 için ; μB2(0) = 0 /100 + 0.5 = 0 + 0.5 = 0.5

1. Adım

* Çıkarım Mekanizması
  + 1. kuralın kesinlik derecesi:
    - μ CR f1 = 0.25\*0.5 = 0.125
  + 2. kuralın kesinlik derecesi:
    - μ CR f2 = 0.25\*0.5 = 0.125
  + 3. kuralın kesinlik derecesi:
    - μ CR f3 = 0.75\*0.5 = 0.375
  + 3. kuralın kesinlik derecesi:
    - μ CR f4 = 0.75\*0.5 = 0.375

1. Adım

* Çıkış Denklemlerinin Hesaplanması
  + f1(X1,X2) = 3\*X1-2\*X2 + 1 için;
    - f1 (50,0) = 3\*50 - 2\*0 + 1 = 151
  + f2(X1,X2) = X1 + X2
    - f2 (50,0) = 50 + 0 = 50
  + f3(X1,X2) = 2\*X1 + X2 – 2
    - f3(50,0) = 2\*50 + 0 - 2 = 98
  + f4(X1,X2) = X1-3\*X2 + 5
    - f4(50,0) = 50 - 3\*0 + 5 = 55

1. Adım
   * Ağırlıklı Ortalama Metodu;
     1. Y = ((μ CR f1 \* f1) + (μ CR f2 \* f2) + (μ CR f3 + f3) + (μ CR f4 + f4)) / (μ CR f1 + μ CR f2 + μ CR f3 + μ CR f4 )
     2. Y = ((0.125 \* 151)+(0.125\*50)+(0.375\*98)+(0.375\*55))/(0.125+0.125+0.375+0.375)
     3. Y = 82.5

**UZMAN SİSTEMLER**

**Uzman Sistemin Bileşenleri**

* **Bilginin temin edilmesi**: Uzman sistemin uzmanlık alanı ile ilgili bilgilerin toplanması, derlenmesi ve bilgisayarın anlayacağı şekle dönüştürülmesi çalışmalarını kapsar.
* **Bilgi tabanı**: Uzman sistemin uzmanlık alanı ile ilgili toplanan bilgilerin saklandığı yerdir. Bilgiler genellikle kurallar (IF...THEN... şeklinde), bilgi çatıları (çerçeveleri), bilgi sınıfları ve prosedürlerden oluşur.
* **Bilgi mühendisi**: uzmana problem alanlarını yapılandırmak için yardım eder. Bunu uzman kişi(ler)in cevaplarını yorumlayarak, bütünleştirerek, benzetmeler yaparak bilginin formülasyonunu gerçekleştirir.
* **Çıkarım mekanizması**: Bilgi tabanında bulunan bilgileri arayan, filtreleyen, yorumlayan ve sonuçlar çıkaran yani; çözüm üreten bir mekanizmadır. Genel olarak iki türlü çıkarım vardır.
  + **İleri doğru zincirleme**: Bu durumda, ilgili problem hakkındaki gerçeklerden hareket edilerek sonuca gidilir. Çıkarım ünitesi, problemin en başından başlayarak (IF cümlesinden) sonuç kısmına (THEN...) ulaşmasıdır.
  + **Geri doğru zincirleme**: Bu durumda ise bir sonuç ele alınarak geriye doğru, “o sonucu destekleyen gerçekler var mıdır?” Sorusunun cevabı aranır. Çıkarım ünitesi; problemi çözerken kuralın en sonu olan sonuç (THEN...) cümlesi ile başlar ve şart (IF...) cümleleri tatbik edilerek çözüm bulunur.
* **Kullanıcı ara birimi**: Uzman sistemi kullanan kişiler ile uzman sistemin iletişimini sağlar. Problemlere üretilen sonuçların nasıl üretildiği ve niçin o sonuçlara varıldığını açıklar. Uzman sistemin bir uzman gibi görülmesi bu ara birimi ve açıklama yeteneğinin güçlü olmasına bağlıdır.

**Uzman Sistemlerin Kısıtları**:

* Uzman Sistemlerin dış dünya ile bağlantıları yetersizdir.
* Tabandaki bilgi yüzeyseldir.
* Aşırı derecede uzman kişilerin bilgisine muhtaçtır.
* Hemen hemen hiç öğrenme becerisi yoktur.
* Kullanım alanı şimdilik sınırlı sayılabilir.
* Akıl yürütme metotları sınırlıdır.
* Bilgi sunumu metotları sınırlıdır.

**Uzman Sistemlerin Avantajları:**

* Uzman sistemler, uzman insanlardan daha hızlı olduğu için bu hız farkı üretime artış olarak yansıyabilir.
* Uzmanların yerine bir uzman sistem devreye geçirilebilirse, uzman insanlar için yapılan harcama ortadan kalkar.
* Uzman sistemler tasarlanmasındaki amaca yönelik olarak, bir insandaki tüm özellikleri taşımasından dolayı “emin değilim”, “bilmiyorum” gibi kesinlik belirtmeyen ifadelere de yer verilmesi, değerlendirme ve kıstaslarını buna göre ayarlaması yani tam ve kesin olmayan bilgi ile çalışabilme gerçekleştirilebilmektedir.
* Uzman sistemler, ara sonuçların gösterilebilmesi açısından bu sistemlerin kullanıldığı bazı alanlarda ara sonuçları raporlar halinde görmek ve ilerisine yönelik plan ve tasarı yapmak amacı ile istenildiği anda müdahale edilebilir şekilde tasarlanabilir.
* Uzman sistemler geleceğin fabrikalarının gelişmesine katkıda bulunacak ve bu doğrultuda öngörülen insansız fabrikaların uzman sistemler ile daha da gelişmesi sağlanabilecek. Özellikle 4. sanayi devrimi (Endüstri 4.0) ile bu konuda büyük bir atılım beklenmektedir.

**Uzman Sistemlerin Dezavantajları**:

* Günümüz koşullarında her alanda yeteri kadar uzman bulunamamakta, bulunsa dahi yeteri zamanı ayıramamaktadır
* Aynı konuya uzmanların farklı bakış açıları getirmesinin bir sonucu olarak, uygulanacak alana göre değişiklik göstermesiyle birlikte uzmanların benimsediği ve doğru olarak ortaya koyduğu bilgiler tutarsızlık arz ediyor olabilir.
* Maliyet yüksekliği ve zaman fazlalığı bir sorun olarak tezahür eder, bunun nedeni ise uzman sistem tasarlamanın yüksek maliyeti ve buna ek olarak, bilgi mühendisi, uzman ve programcı ile 3-5 yıl arasında değişen sistemlerin maliyetleridir.
* Özgünlük, insana haiz bir beceri olması nedeni ile alışılagelmemiş bir durumla karşılaşıldığında, uzman insanın yeteneği ile yeni bir çözüm sunulabilirken uzman sistem bunu yapamamaktadır.

**Uzman Sistemlerin Kullanım Alanları**:

* **Yorumlama**: Sensörlerden gelen durumların tanımlanması. Burada, gözlemlerden durum yorumu çıkartılır. Gözlenen verilere sembolik anlamlar verilerek işlemler yapılır.
* **Tahmin**: Verilmiş durumlara benzer sonuçların çıkarılması. Tahmin sistemi parametrik dinamik model kullanır. Bu modelde parametre değerler, verilen duruma göre değişir. Modelden elde edilen sonuçlar tahmin için esas teşkil eder.
* **Teşhis**: Gözlem neticelerine göre sistem bozukluklarının tespiti. Gözlemlerden bozuklukları bulan sistemlerde kullanılır. Tıbbi, elektronik, mekanik ve yazılım tanıları bu sınıfa girer.
* **Tasarım**: Sınırlı şartlar altında nesne tasarımı. Tasarlama sistemi, tasarlama kanıtlamalarını sağlayan nesne konfigürasyonlarını geliştirir.
* **Planlama**: İşlemlerin tasarımı. Bu sistemler, fonksiyona sahip nesnelerin, tasarım problemleri üzerinde özelleşmişlerdir. Bunlar otomatik programlama, robotlar, projeler, yol, iletişim, deney ve askeri planlama problemleriyle ilgilidir.
* **İzleme**: Hassaslıkları planlamak için gözlemlerin karşılaştırılması.
* **Hata Ayıklama**: Hatalara sebep olan bozuklukların sunulması
* **Tamir**: Belirlenmiş yönetim planının yürütülmesi
* **Eğitim**: Öğrenci davranışlarının tespiti ve düzeltilmesi
* **Kontrol**: Sistem Davranışının yorumu, tahmini, tamiri ve izlenmesi

**Uzman Sistemler ve Uygulama Alanları**

* Biyoloji
  + CRYSALIS
    - Proteinlerin 3 boyutlu yapılarını yorumlama
  + DENDRAL
    - Moleküler Yapı Yorumlama
  + CLONER
    - Yeni biyolojik yapı tasarımı
  + MOLGEN
    - Gen kopyalama (klonlama) deneylerinin tasarımı
  + SECS
    - Kompleks organik moleküllerin tasarımı
  + SPEX
    - Moleküler biyoloji deneylerinin planlanması
* Elektronik
  + ACE
    - Telefon Ağlarındaki arızaların teşhisi
  + IN-ATE
    - Osiloskop hatalarının teşhisi
  + NDS
    - Ulusal iletişim ağının teşhisi
  + PALLADIO
    - Yeni VLSI devrelerinin tasarım ve testi
  + CADHELP
    - Bilgisayar destekli tasarım yardımı
  + SOPHIE
    - Devre arıza teşhisi yardımı
* Tıp
  + PUFF
    - Akciğer hastalıklarının teşhisi
  + VM
    - Yoğun bakım hastalıklarının incelenmesi
  + AI/COAG
    - Kan hastalıklarının teşhisi
  + CADUCEUS
    - Dahili hastalıkların teşhisi
  + MYCIN
    - Bakteriyel enfeksiyonların teşhis ve tedavisi
  + ONCOCIN
    - Kemotrapi hastalarının tedavi ve idaresi
  + ATTENDING
    - Anestezi işlem talimatı
  + GUIDON
    - Bakteriyel enfeksiyonlar için talimat
* Termodinamik
  + REACTOR
    - Reaktör kazalarının teşhis ve çözümü
  + DELTA
    - GE lokomotiflerinin teşhis ve çözümü
  + STEAMER
    - Buhar santralinin çalışma talimatı
* Madencilik
  + LITHO
    - Petrol kuyularının verilerinin yorumlanması
  + MUD
    - Sondaj problemlerinin teşhis ve çözümü
  + PROSPECTOR
    - Mineral arama çalışmaları için jeolojik verilerin yorumlanması
* Bilişim
  + BDS
    - Şalterli ağlarda bozuk kısımların teşhisi
  + YES/MVS
    - IBM MVS işletim sistemi için konrol/izleme

**Uzman Sistem Tasarımı**

* Problem, yordamsal programlama ile çözülebilir mi?
  + Cevap evet ise uzman sisteme gerek yoktur.
* Uygulama yapılacak alanın sınırları kesin olarak belli mi?
  + Tasarlanacak Uzman Sistemin bilmesi gerekenler ve yeteneklerinin sınırları tam olarak belirli olmalıdır.
* Uzman Sisteme ihtiyaç var mı?
  + Pek çok uzmanın bulunduğu bir alanda Uzman Sistem hazırlamak pek de mantıklı olmaz.
* İşbirliği yapılabilecek en az bir uzman kişi mevcut mu?
  + Tasarlanan projeye ilgi duyan en az bir uzman kişinin bulunması gereklidir
* Uzman kişi, bilgisini anlaşılabilir derecede açıklayabiliyor mu?
  + Tasarım yapan mühendis, uzman kişinin söylediklerini net bir şekilde anlayamazsa, aldığı bilgiyi bilgisayar koduna dönüştürmesi çok zor olur.

**Tasarımda Bulunması Gereken Özellikler:**

1. Yüksek Performans:
   * **Hızlı Cevap Verme**: Tasarlanan sistemin, sorulan sorulara yönelik bir sonuca makul bir sürede varabilmesi ve hatta uzman bir insandan daha çabuk karar verebilmesi gerekir.
   * **Güvenilirlik**: Hazırlanan Uzman Sistemin güvenilir olması, hata vermemesi gerekir.
   * **Anlaşılabilirlik**: Tasarlanan sistemin, bir konuda vardığı sonucun aşamalarını tek tek açıklayabilmesi gerekir. Sonuca nasıl vardığı meçhul olan bir sistemden ziyade, tıpkı bir insan uzman gibi, gerektiğinde vardığı sonucun nedenlerini açıklayabilmelidir.
   * **Esneklik**: Bir Uzman Sistemde kullanmak üzere büyük miktarda bilgi yüklemek gerekir.

**Uzman Sistemin Tasarım Adımları:**

1. Uzman Sistem tasarımı için bir araç seçilir.
2. Problemin tam olarak ne olduğu belirlenir sistemin içermesi gereken bilgi analiz edilir.
3. Sistem dizayn edilir. Yani akış diyagram ve matrisleri, sistemin kurallarının taslakları bu dizaynın içerisinde yer alır.
4. İlk adımda belirlediğimiz araç kullanılarak sistemin bir prototipi oluşturulur. Bu prototip, bilgi tabanının oluşmuş halini, test edilmesini ve birçok denemenin gerçekleştirilmesini içerir.
5. Sistem istediğimiz hale gelene kadar üzerinde genişletmeler ve testler yapılarak sistem gözden geçirilir.
6. Sistem son halini alır ve yeri geldikçe üzerinde yeni düzenlemeler yapılabilir.

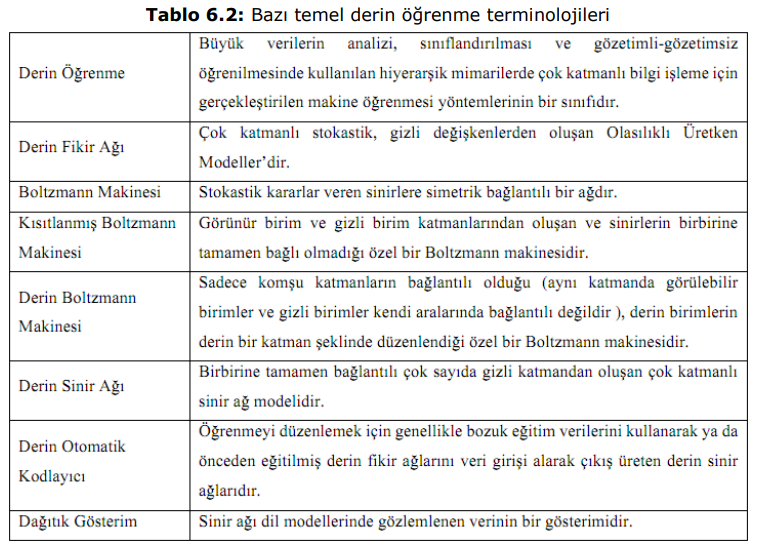
**DERİN ÖĞRENME**

Makine Öğrenmesinin Zorlukları;

* **Özellik çıkarımı**: Özellik çıkarma geleneksel makine öğrenimi sistemleri için en önemli adımlardan birisidir. Çünkü sistemin doğru çalışması, doğru özelliklerin ve özellik sayılarının seçilmesine bağlı olarak değişmektedir.
* **Gürültü**: Bir yüz tanıma işleminde, görüntülerdeki yüzler farklı boyutlarda, renklerde, açılarda, ışıklarda ya da düşük çözünürlükte olabilir. Bir metin sınıflandırma işleminde, farklı site kaynaklarından alınan metinler yanlış yazımlar, noktalama işaretleri, yaygın kullanımı olmayan kısaltmalar, eksik cümleler ve sisteme girdi olarak verilecek metinler farklı uzunluklarda cümleler içerebilmektedir.
* **Aşırı öğrenme (ezberleme):** Bir yüz tanıma ya da metin sınıflandırma işleminde, model karmaşıklığı arttırılarak her bir eğitim örneği çok iyi bir şekilde sınıflandırılabilir. Ancak modelin test verileri yani daha önceden görülmemiş veriler üzerindeki performansı düşük olabilmektedir. Bir model eğitim örneklerinde çok iyi çalışırken test verilerinde kötü performans sergiliyorsa aşırı öğrenmiştir.
* **Öğrenme algoritmasının seçimi**: Belirli bir iş için kullanılan model doğru çalışmayabilir ve alternatif algoritmalara ihtiyaç duyulabilir. Araştırmacının, problemin ihtiyacına yönelik iyi bir öğrenme algoritmasını seçilebilmesi için pek çok modeli test etmesi ve ihtiyaca uygun algoritmayı geliştirmesi ya da kullanması gerekmektedir.
* **Eksik değerler**: Eksik özellikler temel olarak veri eksikliğinden ya da tercih etmeme seçeneğinin kullanılmasından kaynaklanmaktadır.

Derin öğrenme mimarilerinin akademik ve iş dünyasında çok sıcak bir çalışma alanı olmasındaki unsurlar aşağıda sıralanmıştır:

* Metin, görüntü ve ses gibi farklı formatlarda, çok büyük ve yüksek kalitede etiketli veri setlerine erişme kolaylığı
* Paralel bir şekilde karmaşık hesaplamalar yapabilen güçlü GPU birimleri, Google Colaboratory gibi derin öğrenme modellerinin eğitilmesine olanak sağlayan ve tamamen bulutta çalışan ücretsiz GPU ortamlarının sunulması
* AlexNet, GoogLeNet, VGG, R-CNN, ResNet ve Inception modülleri gibi yeni derin öğrenme modellerinin geliştirilmesi
* TensorFlow, Keras ve PyTorch gibi kullanımı kolay ve hızlı uygulanabilir açık kaynak derin öğrenme platformlarının olması
* ReLU gibi sıfırlanan gradyan problemine çözüm geliştiren aktivasyon fonksiyonlarının keşfedilmesi
* Dropout gibi yeni düzenlileştirme teknikleri ve veri arttırma yöntemleri ile aşırı öğrenme problemiyle karşı karşıya kalmadan çok büyük ağları eğitebilme olanağı
* Model parametrelerini güncellerken daha iyi ve hızlı sonuçlar üreten RMSprop, Adagrad, Adamax, TFOptimizer ve ADAM gibi optimizatörler.



**Derin Sinir Ağı Mimari ve Modelleri**

Derin öğrenme mimarileri dört temel kategoride sınıflandırılabilir:

* Evrişimsel Sinir Ağları
  + AlexNet
  + VGG
  + GoogLeNet
  + ResNet
* Tekrarlı Sinir Ağları
  + Uzun Kısa Süreli Bellek
  + Kapılı Tekrarlayan Hücre
  + İki yönlü Tekrarlı Sinir Ağı
  + İki-yönlü Uzun Kısa Süreli Bellek
* Kısıtlanmış Boltzmann Makineleri
  + Derin İnanç Ağları
  + Derin Boltzmann Makineleri
* Otomatik Kodlayıcılar
  + Seyrek Otomatik Kodlayıcı
  + Gürültü Temizlemeli Otomatik Kodlayıcı
  + Varyasyonel Otomatik Kodlayıcı

**Evrişimsel Sinir Ağları**

Konvülasyon Sinir Ağları olarak da geçen Evrişimsel Sinir Ağları (Convolutional Neural Networks - CNN), görüntü tanıma ve doğal dil işlemine dayalı sınıflandırma modellerinin oluşturulmasında önemli rol oynamaktadır.

Sırasıyla ESA mimarisinin modeli:

1. Girdi
2. Konvolüsyon
3. ReLU
4. Havuzlama
5. Konvolüsyon
6. ReLU
7. Havuzlama
8. Tam Bağlantılı
9. Softmax
10. ÇIKTI

Her bir katmanın işlevi:

* **Girdi Katmanı**: Bu katmanda veri ham olarak ağa verilmektedir. Giriş görüntü boyutunun yüksek seçilmesi hem yüksek bellek ihtiyacını hem eğitim süresini hem de görüntü başına düşen test süresini uzatabilir. Görüntü analizinde hem ağ derinliği hem donanımsal hesaplama maliyeti hem de ağ başarısı için uygun bir giriş görüntü boyutu seçilmelidir.
* **Konvolüsyon İşlemi**: Konvolüsyon işleminin amacı matrisi belirli filtreler ile tarayarak istenilen görüntüler ve metinler için açıklayıcı özellikleri çıkarmaktadır. konvolüsyon filtresi kullanılarak birer kaydırma (stride) ile girdi matris ve filtrenin ilişkili değerleri arasında birebir çarpım gerçekleştirilir. Her bir çapım sonucu alınan tam sayılar toplanarak, özellik haritası denilen 4x4’lük çıktı matrisinin ilgili yerine yerleştirilir. Bu katman dönüşüm katmanı olarakta bilinir. Filtreler, bir önceki katmandan gelen görüntülere konvolüsyon işlemi uygulayarak çıkış verisini oluşturur. Bu katmandaki nöronların görevi belirli öznitelikleri aramaktır. Bu katmanlarda filtreleme, piksel ekleme (padding - p), piksel kaydırma (stride - s) gibi işlemler yapılmaktadır. Konvolüsyon katmanında gerçekleşen filtreleme adımının formülü şu şekilde tanımlanmaktadır;

Konvülasyon Katmanı Çıkışı = [((n+2p−f) / s) + 1] \* [ ((n+2p−f) / s) + 1] x filtre sayısı

Burada;

* + n değeri imgenin boyutunu
  + f filtre boyutunu
  + p padding değerini
  + s ise stride değerini belirtmektedir

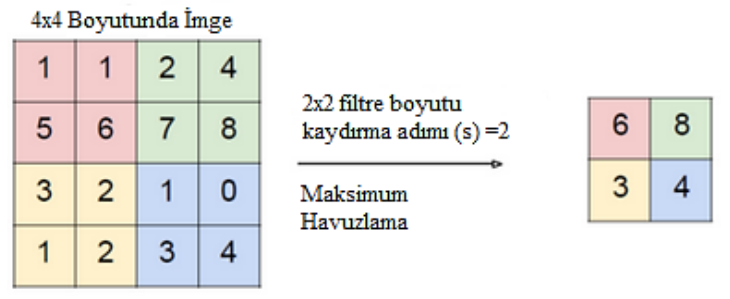
**Piksel Ekleme (Padding):** Evrişim işleminde, giriş matrisi ile çıkış matrisi arasındaki boyut farkını uyumlu hale getirmek için giriş matrisine ilave pikseller eklenmesi ile sağlanır.

p=(f-1)/2 denkleminden faydalanılarak giriş matrisine eklenecek piksel sayısı bulunur.

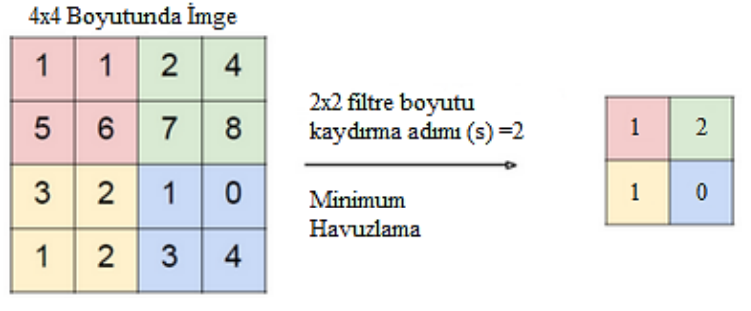
**Kaydırma Adımı (Stride):** Bu değer evrişim işlemi için giriş matrisi üzerinde filtreyi kaç piksellik adımlarla kaydıracağının bilgisini verir. Bu da doğrudan çıkış boyutunu etkileyen bir parametredir.

Konvolüsyon katmanından sonra ReLU fonksiyonu kullanılmaktadır. ReLU fonksiyonu aşağıdaki denklemde verildiği üzere negatif girdiler için 0 değerini alırken x pozitif girdiler için x değerini almaktadır. f(x) = max (0, x) Verinin ReLU fonksiyonundan geçirilerek doğrultulması, düzleştirme olarakta ifade edilmektedir.

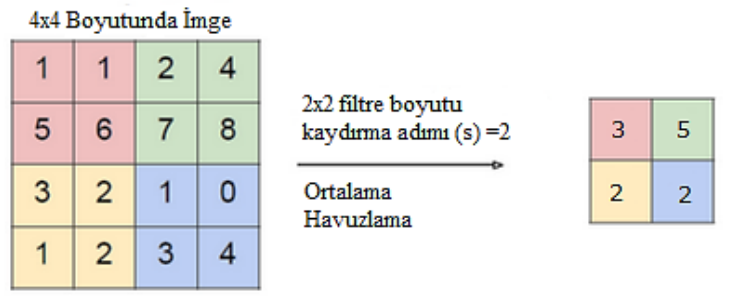
* **Havuzlama Katmanı**: Havuzlama (pooling) katmanında, kullanılacak olan maksimum havuzlama veya ortalama havuzlama teknikleri ile gelen girdi boyutu azaltılarak ağın hesapsal yükü, hafıza kullanımı ve parametre sayısı düşürülmektedir. Modelin eğitim setini aşırı öğrenmesinin (ezberlemesinin) önüne geçilmektedir.
  + **Maksiumum Havuzlama**: Buradaki amaç asıl imgenin üzerinde, 2x2 boyutunda filtre pencereleri seçip, o kısımda kalan maksimum değeri bulmaktadır.



* + **Minimum Havuzlama**: Buradaki amaç asıl imgenin üzerinde 2x2 boyutunda filtre pencereleri seçip, o kısımda kalan minimum değeri bulmaktadır.



* + **Ortalama Havuzlama**: Buradaki amaç asıl imgenin üzerinde 2x2 boyutunda filtre pencereleri seçip, o kısımdaki bulunan değerlerin ortalamasını bulmaktadır.



* **Tam Bağlantılı Katman:** Bu katman kendinden önceki katmanın tüm alanlarına bağlıdır. Tam bağlantılı katman, yapay sinir ağı yoğun (dense) katmanı olarakta isimlendirilmektedir
* **Çıktı Katmanı**: Bu katman, sınıflandırma katmanı olarakda bilinir. Birkaç tam bağlı katmanın birleşiminden oluşan son katman, bir tahmin yani çıktı üretmektedir.