



**Hacettepe Üniversitesi
Fen Fakültesi İstatistik Bölümü**

**Yapay Sinir Ağları ile Sınıflandırma
Classification with Artificial Neural Networks**

**Metin USLU
İleri İstatistik Projeleri I**

Ankara 2013

İçindekiler

1. Özet	3
2. Yapay Zeka	4
3. Yapay Sinir Ağları	5
3.1 Yapay Sinir Ağlarının Tarihsel Gelişimi	6
3.2 Yapay Sinir Ağlarının Uygulama Alanları	8
3.3 Yapay Sinir Ağlarının Temel Bileşenleri	9
3.3.1 Mimari Yapı	9
3.3.2 Öğrenme Algoritması	10
3.3.3 Aktivasyon Fonksiyonu	13
4. Sınıflandırma	17
5. Uygulama	18
5.1. Kanser Veriseti	18
6. Sonuç	20
7. Kaynakça	21

1. Özet

Yapay zeka tekniklerinden biri olan yapay sinir ağıları, son yıllarda öngörü ve sınıflandırma problemlerinde tercih edilen bir modelleme tekniğidir. Özellikle varsayımlardan bağımsız olması ve doğrusal olmayan sistemlerde, öngörüler açısından istatistiksel tekniklere göre kolaylık sağlayan özelliklere sahiptir.

Bu çalışmada, kadınlarda en yaygın ikinci kanser çeşidi olan göğüs kanseri verileri kullanılarak ileri beslemeli yapay sinir ağıları kullanılarak iyi huylu ve kötü huylu olarak sınıflandırılma yapılmıştır.

Çalışma da ilk olarak Yapay Zeka, ardından sırasıyla Yapay Sinir Ağları, Sınıflandırma ve Uygulama kısımları detaylı olarak anlatılmıştır. Uygulama Matlab R2012a paket programı içerisinde yer alan yapay sinir ağıları araç paketi kullanılarak yapılmıştır. Çalışmanın son bölümünde ise, uygulamadan elde edilen sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Yapay Sinir Ağları, Sınıflandırma, Meme Kanseri

2. Yapay Zeka

İnsan beyninin ve düşünme yeteneğinin taklit edilmesi isteği sanıldığı gibi aksine çok eski zamanlardan beri var olmuş bir istektir. İnsan beyni ve düşünebilme yeteneğine ilişkin ilk açıklayıcı teori geliştirme denemeleri Antik Yunan düşünürleri olan Plato (M.Ö. 427-327) ve Aristoteles'e (M.Ö. 384-322) kadar uzanmaktadır. Tarihte kendisine çalışmaları ile yer edinen Descartes (1596-1650)'de insanın düşünme yeteneğiyle ilgilenen 18. yüzyıl düşünürü olmuştur.

Yapay zekâ kavramının geçmişi modern bilgisayar bilimi kadar eskidir. Fikir babası, "Makineler düşünebilir mi ?" sorusunu ortaya atarak Makine Zekâsını tartışmaya açan Alan Mathison Turing'dir. 1943'te İkinci Dünya Savaşı sırasında Kripto Analizi gereksinimleri ile üretilen elektromekanik cihazlar sayesinde bilgisayar bilimi ve yapay zekâ kavramları doğmuştur.

İnsan beynin dünyanın en karmaşık makinesi olarak kabul edilebilir. İnsan beyni sayısal bir işlemi birkaç dakika da yapabilmesine karşın; idrak etmeye yönelik olayları çok kısa sürede yapar. Örneğin yolda giden bir şoför, yolun kayganlık derecesini, önündeki tehlikeden ne kadar uzak olduğunu, sayısal olarak değerlendiremezse dahi geçmişte kazanmış olduğu tecrübeler sayesinde aracın hızını azaltır. Çünkü o saniyelerle ölçülebilecek kadar kısa bir sürede tehlikeyi idrak etmiş ve ona karşı koyma gibi bir tepki vermiştir. Bu noktada akla gelen ilk soru şu olmaktadır: Acaba bir bilgisayar yardımı ile böyle bir zeka üretmek mümkün olabilir mi? Bilgisayarlar çok karmaşık sayısal işlemleri anında çözümleyebilmelerine karşın, idrak etme ve deneyimlerle kazanılmış bilgileri kullanabilme noktasında çok yetersizdirler. Bu olayda insanı ya da insan beynini üstün kılan temel özellik, sinirsel algılayıcılar vasıtası ile kazanılmış ve görelî olarak sınıflandırılmış bilgileri kullanabilmesidir.

Uzman Sistemler, Bulanık Mantık, Genetik Algoritma ve Yapay Sinir Ağları gibi yapay zeka alt dalları özellikle son yıllarda, geniş bir araştırma ve uygulama alanı bulmaktadır. Bu yapay zeka alt dalları kısaca özetleyecek olursak; Uzman Sistemler, uzmanların belli bir konudaki bilgi ve deneyimlerinin bilgisayara aktarılmasını amaçlayan sistemlerdir. Bulanık Mantık ise bulanık küme teorisine dayanır. Sıcak-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.

3. Yapay Sinir Ağları

YSA, insan beyninin çalışma mekanizmasını taklit ederek beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır.

Haykin (1994), beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Böylece YSA denen yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı, beynin fizyolojisinden yararlanılarak oluşturulan yapay bilgi işleme modelleri ortaya çıkmıştır. YSA'lar problemleri çözmede son yıllarda çok başarılı yaklaşımlar arasındadır (Çakar vd., 1996) YSA, karmaşık yapıda içsel ilişkilere sahip problemlere çözüm bulabilmek için geliştirilmiş bir yapay zekâ tekniğidir.

Birçok yapay sinir ağı modeli, iyi bilinen istatistiksel modellerle benzer ya da özdeş olmasına karşın, YSA literatüründe kullanılan terminoloji ile istatistik literatüründe kullanılan terminoloji arasında farklılık vardır. Literatürde YSA ile bazı istatistiksel modeller arasındaki ilişkiyi gösteren çalışmalar da bulunmaktadır (Sarle, 1994). Aşağıdaki tablo da YSA ve istatistik terminolojilerinde aynı kavramları belirten bazı terimler gösterilmektedir.

İstatistik Terminolojisi	Yapay Sinir Ağları Terminolojisi
Model	Yapay Sinir Ağı
Parametre	Ağırlık
Bağımsız değişken	Girdi
Tahmin değeri	Çıktı
Bağımlı değişken	Hedef
Artık	Hata
Güven aralığı	Hata çizgisi
Temel fonksiyon	Nöron
Bağımsız değişkenler kümesi	Girdi tabakası
Temel fonksiyonlar kümesi	Gizli tabaka
Tahmin değerleri kümesi	Çıktı tabakası
Gözlem	Örüntü
Kestirim ya da optimizasyon	Eğitim, Öğrenme ya da Adaptasyon
Örneklem adaptasyonu	Çevrimiçi öğrenme
Grup adaptasyonu	Çevrimdışı öğrenme
Kestirim ölçütü	Hata, Maliyet ya da Lyapunov fonksiyonu
Dönüşüm	Fonksiyonel bağlantı
Diskriminant analizi	Sınıflama
Regresyon	Eşleme, Yaklaşım ya da Denetimli öğrenme
Veri indirgeme	Denetimsiz öğrenme ya da şifreleme
Genelleştirme	İnterpolasyon ya da Ekstrapolasyon

3.1 Yapay Sinir Ağlarının Tarihsel Gelişimi

YSA'nın tarihsel gelişimi incelendiğinde, günümüze kadar sayısız çalışma ve uygulamadan geçtiği görülmüştür.

- 1890 - insan beyninin yapısı ve fonksiyonları ile ilgili ilk yayının yazılması,
- 1911 - insan beyninin sinir hücrelerinden oluştuğu fikrinin benimsenmesi,
- 1943 - yapay nöronlara dayanan hesaplama teorisinin ortaya atılması ve eşik değerli mantıksal devrelerin geliştirilmesi,
- 1949 - öğrenme prosedürünün bilgisayarlar tarafından gerçekleştirilecek şekilde geliştirilmesi,
- 1956 - 1962 Adaline ve Widrow-Hoff öğrenme algoritmasının geliştirilmesi,
- 1957 - 1962 Algılayıcının geliştirilmesi,
- 1965 - ilk makina öğrenmesi kitabının yayınlanması,
- 1967 - 1969 bazı gelişmiş öğrenme algoritmalarının geliştirilmesi (Grasberg vs.),
- 1969 - tek katmanlı algılayıcıların yetersizliklerinin ispatlanması,
- 1969 - 1972 doğrusal ilişkilendiricilerin geliştirilmesi,
- 1972 - korelasyon matris belleğinin geliştirilmesi,
- 1974 - geri yayılım modelinin geliştirilmesi,
- 1978 – Uyarlanabilir Rezonans Teorisi modelinin geliştirilmesi,
- 1982 yılında bir fizikçi olan Hopfield'in, YSA'nın genelleştirilebileceğini ve özellikle geleneksel bilgisayar programlama ile çözülmesi zor olan problemlere bile çözüm üretebileceğini göstermesi (Öztemel, 2003).
- 1982-1984 yıllarında Kohonen kendi adıyla anılan eğiticişiz öğrenebilen ağırları geliştirilmesi (Hooman ve Zilouchian, 2001).
- 1984 - Boltzman Makinası'nın geliştirilmesi,
- 1988 - RBF modelinin geliştirilmesi,
- 1991 – Genel Regresyon Sinir Ağı modelinin geliştirilmesi.
- 2004 yılında Virjinya Eyalet Üniversitesi'nden Zhicun Wang'ın, 3 boyutlu konfigürasyonlarda aerodinamik güçlerin kontrolünü YSA'ları kullanılarak simüle etmiştir.
- 2006 - Jorgensen bir uçağın rüzgâr tüneli verilerinden seçilen aerodinamik karakterleri öğrenerek “doğrusal geri besleme kontrolünde kullanılmasını sağlayacak yapay sinir ağı programı üzerinde çalışmasıdır.

YSA' nın genel karakteristik özellikleri:

1. YSA, bilgisayarların olayları öğrenerek, aynı durumlarla karşılaştığında karar vermesini sağlarlar.
2. Olaylarla ilgili bilgileri, örneklerden edinerek genelleme yapma yeteneğine sahip olurlar. Bu nedenle iyi genellemeler için ağı sunulan örneklerin önemi büyüktür.

3. YSA'da bilgi ağ bağlantılarındadır ve bağlantı değerine göre önemli olup olmadığı belirlenir. Yorumlanması zordur.
4. Ağ kendisine sunulan örneklerle genellemeler yaparak, daha önce görmediği örnekler içinde sonuçlar üretebilir.
5. Paralel yapıya sahiptirler. Bir çok nöron eşzamanlı olarak çalışır. YSA' nın önce örneklerle eğitimi ve arkasından testi söz konusudur. İstenen iyi bir performansa ulaşıncaya kadar ağ eğitime devam edilir. Ancak, bu durumda ağın örneklerden ezberleme özelliğinden de kaçınmak gerekir.
6. Geleneksel yöntemler eksik bilgi ile çalışmazken, YSA eksik bilgi ile çalışabilir. Olayları öğrendikten sonra eksik veri ile belirsizlik altında da karar verebilirler. Bilginin önemli olup olmadığı ağın performansına bakılarak anlaşılabilir. Buda hataya karşı toleranslı olma özelliğindendir.
7. Sinir ağları tabiatında doğrusal olmayandır. Bir gizli katman ile her türlü sürekli fonksiyona etkin bir şekilde yaklaşım gösterebilir. Ağ bu özelliğe doğrusal olmayan aktivasyon fonksiyonu ile sahiptir.

YSA' nın dezavantajları:

1. YSA kabul edilebilir sonuçlar üretir. Optimum çözümü garanti etmez. Performansı düşük çözümler elde edilmesi durumu ortaya çıkabilir.
2. Ağ için nöron sayıları, katmanların sayısı, aktivasyon fonksiyonu, öğrenme katsayısı vb. belirlenmesinde genel bir kural olmaması dezavantaj sayılabilir. Kullanıcının tecrübesine bağlıdır. Deneme yanılma neticesinde karar söz konusudur.
3. Daha önce belirtildiği gibi ağın eğitiminde örnekler çok önemlidir. Ağın performansının düşmesi çoğu zaman örneklerin doğru sunulmamasından kaynaklanmaktadır. Kullanıcı eksikliği, uygun bir ağ yapısının sağlanamaması gibi nedenler çözülebilecek bir problemde, çözümsüzlüğe yol açabilir.
4. Eğitimin ne zaman bitirileceği konusunda kesin bir sınır yoktur. Hata istenen belli bir değere ulaşıncaya, eğitimin tamamlanması yeterli görüldüğü için optimum öğrenmenin gerçekleştiği söylenmez. Ayrıca eğitimin geç sonlandırılması ağın ezberleme durumunu ortaya çıkardığı için örnekleri ezberleyen ağ, yeni örneklerle karşılaştığında uygun olmayan sonuçlar üretebilir.
5. Ağın davranışları açıklanamaz. Ağ bir kara kutu gibidir.

Bu dezavantajlara rağmen YSA ile her probleme değişik şekillerde çözümler üretilebilmekte, başarılı sonuçlar elde edilebilmektedir.

3.2 Yapay Sinir Ağlarının Uygulama Alanları

Günümüzde YSA ile ilgili çalışmalar çoğunlukla çok katmanlı ileri beslemeli ağlar ve Hopfield ağlar üzerinedir. YSA farklı alanlarda, çok çeşitli problemlerin çözümünde kullanılmaktadır.

Bunlar;

- Denetim,
- Sistem modelleme,
- Ses tanıma, el yazısı tanıma, parmak izi tanıma,
- Fizyolojik işaretleri izleme, tanıma ve yorumlama,
- Meteorolojik yorumlama,
- Otomatik araç denetimi,
- Finans, borsa, kredi kartı uygulamaları,
- Ulaştırma ve havacılık uygulamaları,
- Tıp, ilaç ve biyomedikal uygulamaları,

olarak özetlenebilir. Başlıca uygulama alanları özetle tahmin, sınıflandırma ve modelleme başlıkları altında toplanabilir.

Sınıflandırma: Müşteri / Pazar profilleri, tıbbi teşhis, risk değerlendirmeleri, şekil tanıma, mikroplar, modeller vb.

Tahmin: İleriki satışlar, üretim ihtiyacı, pazar performansı, enerji ihtiyacı, hava tahminleri vb.

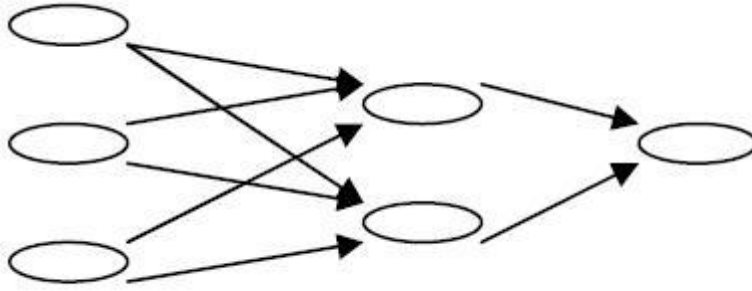
Modelleme: İşlem ve sistem kontrolü, dinamik sistemler, robot kontrolü vb.

3.3 Yapay Sinir Ağlarının Temel Bileşenleri

3.3.1 Mimari Yapı

3.3.1.1 İleri Beslemeli Çok Katmanlı YSA

İleri beslemeli YSA'larda, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan gizli katmandaki hücrelere iletir. Bilgi, saklı ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı YSA'nın, gizli katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yakınsayabileceği gösterilmiştir. En çok bilinen geriye yayılım öğrenme algoritması, bu tip YSA'ların eğitiminde etkin olarak kullanılmakta ve bazen bu ağlara geriye yayılım ağları da denmektedir. İleri beslemeli ağlara örnek olarak çok katmanlı perseptron (Multi Layer Perseptron-MLP) ve LVQ (Learning Vector Quantization) ağları verilebilir. Aşağıda ileri beslemeli ağ mimarisi aşağıda gösterilmiştir.



3.3.1.2 Geri Beslemeli Çok Katmanlı YSA

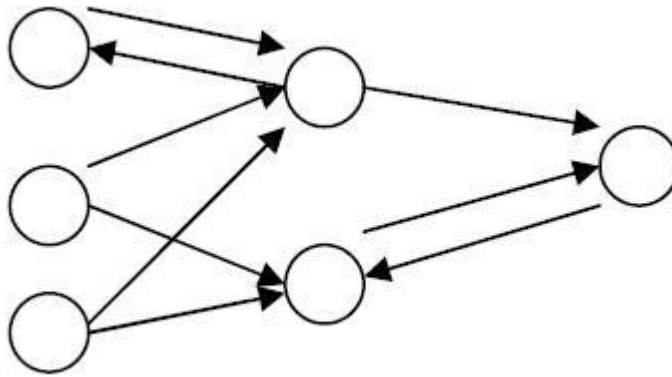
Bir geri beslemeli sinir ağı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Böylece, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Bu çeşit YSA'ların dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundur. Geri beslemeli ağlar çeşitli tipteki zaman-serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir.

Geri beslemeli YSA'larda, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli YSA, doğrusal olmayan dinamik bir davranış

gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir.

Geriye doğru hesaplamada, ağıın ürettiği çıktı değeri, ağıın beklenen çıktıları ile kıyaslanır. Bunların arasındaki fark, hata olarak kabul edilir. Amaç bu hatanın düşürülmesidir. Çıktı katmanında oluşan toplam hatayı bulmak için, bütün hataların toplanması gereklidir. Bazı hata değeri negatif olacağından, toplamın sıfır olmasını önlemek amacıyla ağırlıkların kareleri hesaplanarak sonucun karekökü alınır.

Toplam hatayı en aza indirmek için, bu hatanın kendisine neden olan nöronlara dağıtılması gerekmektedir. Bu da, nöronların ağırlıklarını değiştirmek demektir. Aşağıda geri beslemeli ağı mimarisi aşağıda gösterilmiştir.



3.3.2 Öğrenme Algoritması

YSA'ları diğer yaklaşımlardan ayıran en temel özellik onların öğrenebilmesidir. YSA'ların arzu edilen davranışı göstermesi için nöronların uygun şekilde bağlanması ve bağlantılarının uygun ağırlıklara sahip olması gerekir. Ancak bu önceden ayarlanamaz ve öğrenme yöntemi ile sağlanır. Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Birtakım metot ve kurallar, gözlem ve eğitime göre ağıdaki ağırlıklar değiştirilerek öğrenme gerçekleştirilir. YSA'lar için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

Danışmanlı Öğrenme: Bu tip öğrenmede, YSA'ya örnek olarak bir doğru çıkış verilir. İstenilen ve gerçek çıktı arasındaki farka (hataya) göre nöronlar arası bağlantıların ağırlığını en uygun çıkışı elde etmek için sonradan düzenlenebilir. Danışmanlı öğrenme algoritmasının bir "öğretmene" veya "danışmana" ihtiyacı vardır.

Danışmansız Öğrenme: Girişe verilen örnekten elde edilen çıkış bilgisine göre ağı sınıflandırma kurallarını kendi kendine geliştirmektedir. Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek yoktur. Öğrenme süresince sadece giriş bilgileri verilir. Ağı daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler (patterns) oluşturmak üzere ayarlar.

Takviyeli öğrenme: Bu öğrenme kuralı danışmanlı öğrenmeye yakın bir metoddur. Denetimsiz öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz. Hedef çıkışı

vermek için bir “öğretmen” yerine, burada YSA’ya bir çıkış verilmemekte fakat elde edilen çıkışın verilen girişe karşılık iyiliğini değerlendiren bir kriter kullanılmaktadır.

Levenberg-Marquard Öğrenme Algoritması

Geriye yayılım algoritması, yapay sinir ağlarında en çok kullanılan algoritmadır. Geriye yayılım öğrenmesi sırasında ağ, her giriş örüntüsünü, çıkış nöronlarında sonuç üretmek üzere gizli katmanlardaki nöronlardan geçirir. Daha sonra çıkış katmanındaki hataları bulabilmek için, beklenen sonuçla, elde edilen sonuç karşılaştırılır. Bundan sonra, çıkış hatalarının türevi çıkış katmanından geriye doğru gizli katmanlara geçirilir. Hata değerleri bulunduğundan sonra, nöronlar kendi hatalarını azaltmak için ağırlıklarını ayarlar. Ağırlık değiştirme denklemleri, ağdaki performans fonksiyonunu en küçük yapacak şekilde düzenlenir.

İleri beslemeli ağlarda kullanılan öğrenme algoritmaları, performans fonksiyonunu en küçük yapacak ağırlıkları ayarlayabilmek için, performans fonksiyonunun gradyenini kullanırlar. Geriye yayılım algoritması da, ağ boyunca gradyen hesaplamalarını geriye doğru yapar. En basit geriye yayılım öğrenme algoritması gradyen azalması algoritmasıdır. Bu algoritmada ağırlıklar, performans fonksiyonunun azalması yönünde ayarlanır. Fakat bu yöntem, pek çok problem için çok yavaş kalmaktadır. Bu algorithmadan daha hızlı, daha yüksek performanslı algoritmalar da vardır.

Hızlı algoritmalar genel olarak iki kategoriye ayrılabilir. İlk kategorideki algoritmalar, deneme yanılma tekniklerini kullanarak, standart gradyen azalması (steepest descent) yönteminden daha iyi sonuçlar verebilir. Deneme-yanılma işlemlerini kullanan geriye yayılım algoritmaları; momentum terimli geriye yayılım, öğrenme hızı değişen geriye yayılım ve esnek geriye yayılım algoritmalarıdır. Hızlı algoritmaların ikinci kategorisindeki algoritmalar, standart sayısal optimizasyon yöntemlerini kullanır. Bu algoritmalar; eşlenik gradyen öğrenme algoritması, Newton öğrenme algoritmaları ve Levenberg – Marquardt öğrenme algoritmasıdır.

Öğrenme algoritmaları, kendisinden önce geliştirilen algoritmalara alternatif olarak ortaya çıkmıştır ve önceki algoritmaların iyi yönlerini geliştirip, kötü yönlerini azaltmaya yönelmiştir. Levenberg – Marquardt algoritması da, Newton ve Gradyen Azalması algoritmalarının en iyi özelliklerinden oluşur ve kısıtlamalarını ortadan kaldırır.

Eşlenik gradyen öğrenme algoritmasına alternatif olarak sunulan Newton yöntemlerinde, temel adım Hessian matrisini elde etmektir. Hessian matrisi, performans fonksiyonunun ağırlıklara göre ikinci dereceden türevlerinden oluşan bir matristir. Hessian matrisi, ağırlık uzayının farklı doğrultularındaki gradyen değişimini gösterir.

$$H(n) = \frac{\partial^2 E(n)}{\partial w^2(n-1)} \quad (1)$$

Burada H Hessian matrisi, E performans fonksiyonu, w ağı sinaptik ağırlığıdır. Performans fonksiyonu, duruma göre toplam ani hata veya ortalama karesel hata olarak alınabilir. Bu çalışmada, ileri beslemeli ağların çoğunda olduğu gibi, performans fonksiyonu olarak, ortalama karesel hata kullanılmıştır.

$$E(n) = E_{ort}(n) = \frac{1}{N} \sum_{n=1}^N \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2)$$

Burada N eğitim kümesindeki toplam örüntü sayısını, e_j hata işaretini, C ağı çıkış katmanındaki bütün nöronları içeren kümeyi göstermektedir. d_j istenen değer, y_j ağı çıkışı olmak üzere hata işareti,

$$e_j(n) = d_j(n) - y_j(n) \quad (3)$$

olarak bulunabilir. Hessian matrisi hesaplandıktan sonra, tersi bulunarak ağırlıklar yenilenebilir. Ancak Hessian matrisi çok karmaşık ve ileri beslemeli bir yapay sinir ağı için hesaplanması zor bir matristir. Newton yöntemlerinin içinde, ikinci dereceden türevlerin hesaplanmadan işlem yapılan bir sınıf vardır. Bu sınıftaki yöntemler, quasi – Newton yöntemleri olarak adlandırılırlar. Quasi – Newton yöntemleri, algoritmanın her iterasyonunda, Hessian matrisinin yaklaşık bir şeklini kullanır.

Levenberg – Marquardt algoritması da quasi – Newton yöntemleri gibi, Hessian matrisinin yaklaşık değerini kullanır. Levenberg – Marquardt algoritması için Hessian matrisinin yaklaşık değeri şu şekilde bulunabilir:

$$H(n) = J^T(n) J(n) + \mu I \quad (4)$$

denklemindeki μ Marquardt parametresi, I ise birim matristir. Burada J matrisi, Jakoben matrisi olarak adlandırılır ve ağ hatalarının ağırlıklara göre birinci türevlerinden oluşur:

$$J(n) = \frac{\partial e(n)}{\partial w(n-1)} \quad (5)$$

(4) Denkleminde e, ağ hataları vektörüdür. Jakoben matrisi, hesaplamada Hessian matrisinden daha kolay olduğu için tercih edilir. Ağı gradyeni,

$$g(n) = J^T(n) e(n) \quad (6)$$

olarak hesaplanır ve ağırlıklar (7) denkleminde göre değiştirilir:

$$w(n+1) = w(n) - [H(n)]^{-1} g(n) \quad (7)$$

Marquardt parametresi, μ , skaler bir sayıdır. Eğer μ sıfırsa, bu yöntem yaklaşık Hessian matrisini kullanan Newton algoritması; eğer μ büyük bir sayı ise, küçük adımlı gradyen azalması yöntemi haline gelir. Newton yöntemleri, en küçük hata yakınlarında daha hızlı ve kesindir. Her başarılı adımdan sonra, yani performans fonksiyonunun azalmasında μ

azaltılır ve sadece deneme niteliğindeki bir adım performans fonksiyonunu yükseltecekse μ arttırılır. Bu yöntemle, algoritmanın her iterasyonunda, performans fonksiyonu daima azaltılır.

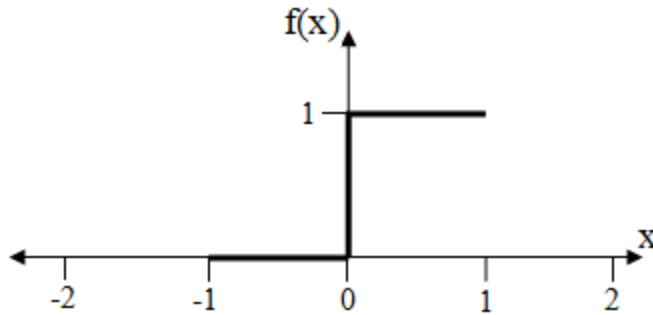
Genel olarak Levenberg – Marquardt algoritması yavaş yakınsama probleminden etkilenmez. Burada hedef, performans fonksiyonun en küçük yapacak ağırlık değerini bulmaktır.

3.3.3 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu girdi ve çıktı birimleri arasındaki eğrisel eşleşmeyi sağlar. Aktivasyon fonksiyonunun doğru seçilmesi ağırlık performansın önemli derecede etkiler. Aktivasyon fonksiyonu genelde tek kutuplu (0 1), çift kutuplu (-1 +1) ve doğrusal olarak seçilebilir. Ağırlık doğrusal olmayan yapıyı öğrenmesini sağlayan bileşenidir. En çok kullanılan aktivasyon fonksiyonları aşağıda belirtilmiştir.

3.3.3.1 Adımsal Aktivasyon Fonksiyonu

Bu aktivasyon fonksiyonu, mantıksal çıktı verir ve sınıflandırıcı ağlarda tercih edilir. Çift kutuplu ve tek kutuplu olarak ikiye ayrılır. Aşağıda tek kutuplu adımsal aktivasyon fonksiyonunun grafiksel gösterimi ve matematiksel gösterimi yer almaktadır.



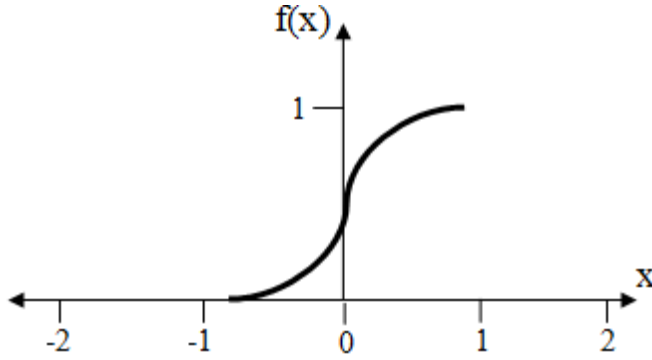
$$f(x) = \begin{cases} 0, & 0 > x \\ 1, & x \geq 0 \end{cases}$$

$f(x)$ tek kutuplu aktivasyon fonksiyonu, x girdi değeridir.

3.3.3.2 Sigmoid Aktivasyon Fonksiyonu

Sürekli ve doğrusal olmayan bir fonksiyon olması nedeniyle doğrusal olmayan problemlerin çözümünde kullanılan YSA'larda tercih edilir. Girdi değerleri hangi aralıkta olursa olsun, çıktı 0 ile 1 arasında olmaktadır. Türevlenebilir olduğu için geri yayılım algoritmaları ile kullanılabilir. Sürekli ve her bir değer birbirinden farklıdır. Asimptotik

olarak doygun noktalara yaklaşır. Sigmoid aktivasyon fonksiyonunun grafiksel gösterimi ve matematiksel gösterimi yer almaktadır.

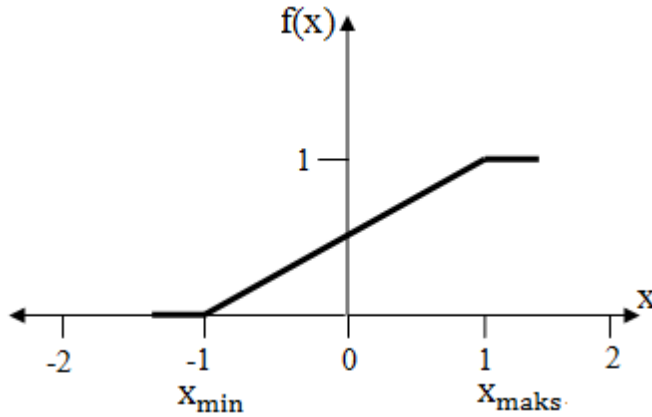


$$f(x) = \frac{1}{1 + e^{-\lambda \text{net}}}$$

$f(x)$ sigmoid aktivasyon fonksiyonu, λ eğim parametresi, net aktivasyon değeridir.

3.3.3.3 Parçalı Doğrusal Aktivasyon Fonksiyonu

Parçalı doğrusal aktivasyon fonksiyonu doğrusaldır ve net aktivasyon değerini minimum girdi değerinden küçük değere sahip girdiler için sıfıra; büyük değerler için bire eşitler. Parçalı doğrusal aktivasyon fonksiyonunun grafiksel gösterimi ve matematiksel gösterimi yer almaktadır.



$$f(x) = \begin{cases} 0, & x \leq x_{\min} \\ mx + b, & x_{\max} > x > x_{\min} \\ 1, & x \geq x_{\max} \end{cases}$$

x_{\min} veri kümesindeki minimum değere sahip değer,

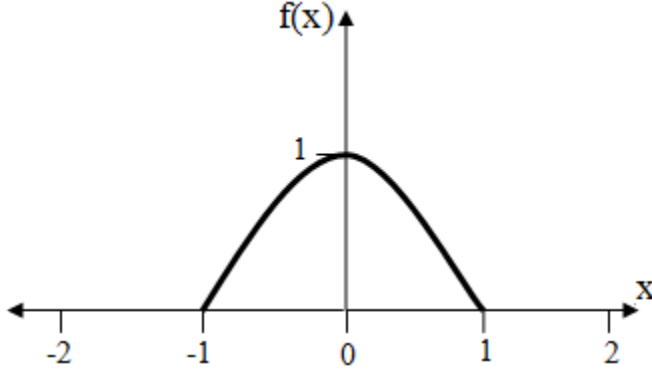
x_{\max} veri kümesindeki maximum değere sahip değer,

x girdi değeri,

b yan değeridir.

3.3.3.4 Gaussian Aktivasyon Fonksiyonu

Aşağıda görüldüğü gibi çan şeklindedir. $|net|$ büyük olduğunda $f(net)$ asimptotik olarak 0 ya da bazı sabit değerlere yakınsar. Gaussian aktivasyon fonksiyonunun grafiksel gösterimi ve matematiksel gösterimi yer almaktadır.

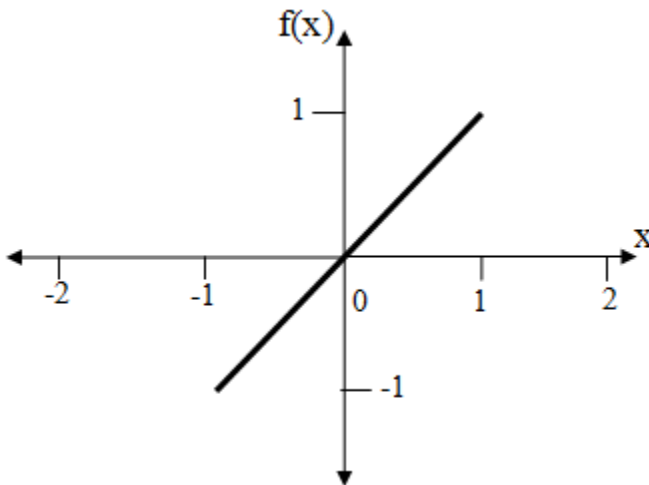


$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

z

3.3.3.5 Doğrusal Aktivasyon Fonksiyonu

Hücresinin net girdisini doğrudan hücre çıktısı olarak veren ve doğrusal bir problemi çözmek amacıyla kullanılan bir aktivasyon fonksiyonudur. Doğrusal süzgeç problemlerinde sıklıkla kullanılmaktadır. Doğrusal aktivasyon fonksiyonunun grafiksel gösterimi ve matematiksel gösterimi yer almaktadır.

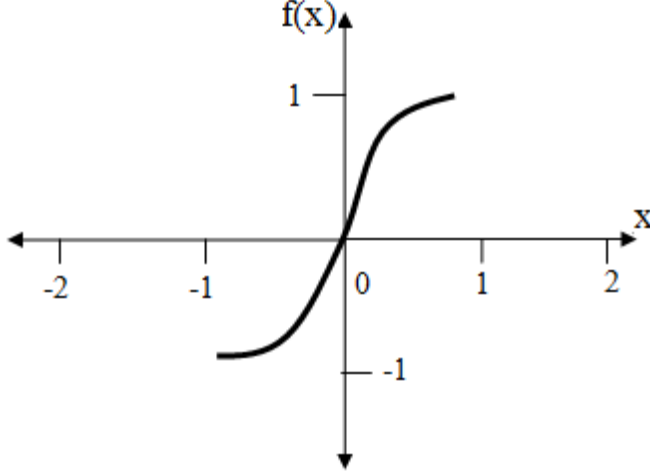


$$f(x) = x,$$

$f(x)$ doğrusal aktivasyon fonksiyonu, x girdi değeridir.

3.3.3.6 Hiberbolik Tanjant Aktivasyon Fonksiyonu

Uygulamada en çok kullanılan aktivasyon fonksiyonlarından biridir. Bu fonksiyon, aşağıda görüldüğü gibi çift kutuplu bir fonksiyondur. Aktivasyon fonksiyonunun matematiksel gösterimi aşağıda gösterilmiştir.



$$f(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}$$

$f(x)$: hiperbolik tanjant aktivasyon fonksiyonu, x girdi değeridir.

4. Sınıflandırma

Verinin içerdığı ortak özelliklere göre ayrıştırılması işlemi sınıflandırma olarak adlandırılır ve veri madenciliğinde önemli bir konudur. Karar ağaçları, Yapay Sinir Ağları, ve Genetik Algoritmalar en yaygın olarak kullanılan sınıflandırma teknikleridir.

Veri içerdığı ortak özellikleri kullanarak söz konusu verileri sınıflandırmak mümkündür. Örneğin bir firmanın müşterilerinin belirli özellikleri göz önüne alınarak sınıflandırılması sağlanabilir. Bir alışveriş merkezi merkezi yöneticileri ortalama alışveriş miktarından daha fazla alışveriş yapan müşterileri varlıklı, diğerlerini ise yoksul olarak birbirinden ayırt etmek isteyebilir. Benzer biçimde bir veri kümesi içinde ortak özellikleri ya da farklılıkları ortaya koyacak biçimde sınıflandırma yapılabilir.

Sınıflandırma bir öğrenme algoritmasına dayanır. Tüm veriler kullanılarak eğitime işi yapılmaz. Bu veri topluluğuna ait bir örnek veri üzerinde gerçekleştirilir. Öğrenmenin amacı bir sınıflandırma modelinin yaratılmasıdır. Başka bir deyişle sınıflandırma, hangi sınıfa ait olduğu bilinmeyen bir kayıt için bir sınıf belirleme sürecidir. Sınıflandırma çok çeşitli alanlarda kullanılabilir. Örneğin finansal pazardaki eğilimleri sınıflandırmak, bir banka da kredi taleplerinin değerlendirilmesinde sınıflandırma yöntemleri kullanılmaktadır.

Bu çalışma sınıflandırma tekniklerinden yapay sinir ağları, lojistik regresyon ve chaid algoritması kullanılmıştır.

5. Uygulama

Bu bölümde yapay sinir ağları kullanılarak kanser verileri üzerinde sınıflandırma yapılmıştır. Uygulama Matlab R2012a paket programı üzerinde gerekli kodlama yapılarak uygulanmıştır. Uygulama da ilk olarak veri seti, ardından sırasıyla kullanmış olduğum yapay sinir ağı mimarisi, öğrenci algoritması açıklanmıştır.

5.1. Kanser Veriseti

Bu çalışmada kullanılan göğüs kanseri verisi 9 adet özellik, 2 adet sınıf bilgisi içeren 699 gözlemden oluşan bir veri setidir. Veri setine Matlab R2012b paket programı altından ya da UCI Machine Learning Repository <http://mllearn.ics.uci.edu/MLRepository.html> sayfasından erişilebilmektedir.

Bağımsız Değişkenler

1. Kütle kalınlığı (Clump thickness)
2. Hücre büyüklüğünün üniformluğu (Uniformity of cell size)
3. Hücre şeklinin üniformluğu (Uniformity of cell shape)
4. Marjinal yapışma (Marginal Adhesion)
5. Tek epithelial hücre büyüklüğü (Single epithelial cell size)
6. Yalın çekirdek (Bare nuclei)
7. Hafif kromatin (Bland chromatin)
8. Normal Nükleoli (Normal nucleoli)
9. Mitozlar (Mitoses)

Bağımlı Değişkenler

1. İyi Huylu (Benign)
2. Kötü Huylu (Malignant)

Veri setine ait dağılımlar aşağıdaki tablo da gösterilmiştir.

Sınıf	Eğitim Kümesi		Test Kümesi		Toplam	
	Gözlem	Yüzde	Gözlem	Yüzde	Gözlem	Yüzde
İyi Huylu	394	0,66	64	0,61	458	0,66
Kötü Huylu	200	0,34	41	0,39	241	0,34
Toplam	594		105		699	

Adımları kısaca özetleyecek olursak;

- ❖ Uygulama Göğüs Kanseri verileri ele alınmıştır.
- ❖ Kanseri verileri 699 gözlemden oluşmaktadır.
- ❖ Bu veri seti için 489 gözlem eğitim algoritması ile ağ eğitilmiş, sonraki 105 gözlem ise tahmin edilerek sınıflandırma yapılmış ve gerçek sonuçlar ile karşılaştırılarak modelin performansı ölçülmüştür.
- ❖ Kanseri verileri Matlab R2012a paket programı içerisinde alınmıştır.
- ❖ Veri setinin 489 gözlemi eğitim, 105 gözlem ise test için kullanılmıştır.
- ❖ Yapay sinir ağı mimarisi girdi, gizli ve çıktı katmanları olmak üzere 3 katman kullanılmıştır.
- ❖ Girdi katmanında nöron sayısı 9, Gizli katmandaki nöron sayısı 1 ile 10 arasında değiştirilmiş olup, çıktı tabaksındaki nöron sayısı 2 olarak alınmıştır.
- ❖ Öğrenme algoritması Levenberg-Marquard geri yayılım öğrenme algoritmasıdır.
- ❖ Tüm çalışmalar Matlab R2012a paket programı üzerinde yapılmıştır.
- ❖ Performans ölçütü olarak Hata Kareler Ortalamasının Karekökü kullanılmıştır.

$$HKOK = \left(\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \right)^{1/2}$$

Burada y_i gerçek değeri, \hat{y}_i değeri ise tahmin değerini, n ise gözlem değerini göstermektedir.

Veri Seti	Yöntem	Model	HKOK
Kanser Verisi	İBYSA	9-2-2	0,295

6. Sonuç

Yapay Sinir Ağları

Gözlenen	Tahmin		
	İyi Huylu	Kötü Huylu	Yüzde
İyi Huylu	61	3	0,953
Kötü Huylu	0	41	1,00
Toplam Yüzde			0,971

Lojistik Regresyon

Gözlenen	Tahmin		
	İyi Huylu	Kötü Huylu	Yüzde
İyi Huylu	447	11	0,954
Kötü Huylu	11	230	0,976
Toplam Yüzde			0,969

Chaid Algoritması

Gözlenen	Tahmin		
	İyi Huylu	Kötü Huylu	Yüzde
İyi Huylu	63	3	0,955
Kötü Huylu	2	37	0,949
Toplam Yüzde			0,952

Yapay sinir ağlarının başarı ile kullanıldığı alanlardan birisi de sınıflandırma problemleridir. Bu çalışmada, göğüs kanserine ilişkin veri setindeki bağımsız değişkenler kullanılarak sınıflandırma yapılmıştır. Yukarıdaki sonuçlara bakıldığında denenen 3 model içerisinde sonuçların yakın olduğu görülmüştür. Yalnız YSA' nın diğerlerine göre küçük de olsa daha başarılı sonuçlar verdiği görülmüştür.

7. Kaynakça

- UCI Machine Learning Repository*. (2013). <http://mllearn.ics.uci.edu/MLRepository.html>
- Aladağ, Ç. H. (2009). Yapay Sinir Ağlarının Mimari Seçimi İçin Tabu Arama Algoritması.
- Çimen, C., Akleylek, S., & Akyıldız, E. (2009). *Şifrelerin Matematiği Kriptografi*. Ankara: ODTU Yayıncılık.
- Elmas, Ç. (2011). *Yapay Zeka Uygulamaları*. Ankara: Seçkin Yayıncılık.
- Günay, S., Eğrioğlu, E., & Aladağ, Ç. H. (2007). *Tek Değişkenli Zaman Serileri Analizine Giriş*. Ankara: Hacettepe Üniversitesi Yayınları.
- Gürsoy, U. T. (2012). *Uygulamalı Veri Madenciliği*. Ankara: Pegem Akademi.
- Özkan, Y. (2008). *Veri Madenciliği Yöntemleri*. İstanbul: Papatya Yayıncılık.
- Sağiroğlu, Ş., Beşdok, E., & Erler, M. (2003). *Mühendislikte Yapay Zeka Uygulamaları I*. Kayseri.
- Şen, Z. (2004). *Yapay Sinir Ağları İlkeleri*. İstanbul: Su Vakfı Yayınları.