

ARAŞTIRMA MAKALESİ / RESEARCH ARTICLE

YAPAY SINIR AĞI KULLANARAK MEME KANSERİ HASTALIĞININ TAHMİNİ

Mariya KİKNADZE¹

¹İstanbul Aydın Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul
mariyamirabella@gmail.com
ORCID ID: 0000-0003-3928-5333

Ahmet GÜRHANLI²

²İstanbul Aydın Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul
ahmetgurhanli@aydin.edu.tr
ORCID ID: 0000-0002-2568-7991

GELİŞ TARİHİ/RECEIVED DATE: 16.01.2020 KABUL TARİHİ/ACCEPTED DATE: 25.12.2020

Özet

Günümüzde meme kanseri (breast cancer) dünyadaki en önemli kötü huylu hastalıklardan biridir. ABD'de meme kanseri, kadınlarda tüm onkolojik hastalıklar arasında birinci sırada yer alır ve akciğer kanserinden sonra onkolojide ölüm nedeninin ikincisidir. Meme kanserinin erken teşhisinde ve tedavisinde son zamanlarda elde edilen büyük başarılarla rağmen, ilk aşamalarda teşhisi için yeni yaklaşımlar ve algoritmalar geliştirilmeye devam etmektedir. Meme kanseri, diğer kötü huylu hastalıklar gibi birçok sınıflandırmaya sahiptir. Histolojik, moleküler, fonksiyonel, TNM sınıflandırması bunlardan bazılarıdır. Çoğu kanser vakası hastalığın geç aşamalarında ancak teşhis edilebilir ve tedavi sıklıkla cevap vermez ve hasta kaybedilir. Bu sebepten meme kanserinin erken evrelerde teşhisi hayati önem taşır. Bu çalışmada sınıflandırma testi doğruluğunu, hassasiyet ve özgüllük değerlerini ölçerek sunmakta olan Wisconsin Meme Kanseri Teşhisi (WDBC) veri seti kullanılmaktadır. Uygulamada, veri seti eğitim aşaması için %70 ve test aşaması için %30 olarak bölünmüştür. Bu çalışma yapay sinir ağı kullanarak meme kanseri tahmininde optimizasyon algoritmalarının ve parametrelerin nasıl seçilmesi gerektiğini incelemekte ve farklı seçimlerinin nasıl sonuç verdiğini göstermektedir.

Anahtar Kelimeler: Yapay Sinir Ağları; Meme Kanseri; Meme Kanseri Tahmini

PREDICTION OF BREAST CANCER USING ARTIFICIAL NEURAL NETWORKS

Abstract

Breast cancer is one of the most important malignant diseases in the world. In the United States, breast cancer ranks first among all oncological diseases in women and is the second leading cause of cancer mortality after lung cancer. Despite recent great success in the early detection and treatment of breast cancer, new approaches and algorithms are still being developed for early diagnosis. Breast cancer has many classifications, like other malignant diseases: histological, molecular, functional, TNM classification. Most cases of cancer can be diagnosed in the later stages of the disease, and treatment is often not responding and the patient

is lost. Therefore, early detection of breast cancer is vital. This study uses the UCI Breast Cancer Wisconsin (Diagnostic) Data Set (WDBC), which is presented by measuring test classification accuracy, sensitivity, and specificity values. The data set was divided into 70% for the training phase and 30% for the testing phase. This study demonstrates the importance of optimization algorithm selection and parameters in the diagnosis of Breast Cancer using Artificial Neural Networks and investigates how they should be chosen. The accuracy results of different optimization algorithms and parameter values are reported.

Keywords: Artificial Neural Networks; Breast Cancer; Breast Cancer Diagnosis

1.GİRİŞ

Makine öğrenimi şu anda birçok bilim ve üretim alanlarında kullanılmaktadır. Tıp da bu alan için bir istisna değildir. Makine öğrenimi sayesinde hastaları sınıflandırmak, en uygun tedavi yöntemini belirlemek, bir hastalığın süresini ve sonucunu tahmin etmek, komplikasyon riskini değerlendirmek, belirli bir hastalık tipinin en karakteristik sendromlarını bulmak gibi birçok görev çözülmüştür. Meme kanseri, normal glandüler hücrelerin kansere dönüşmesinden kaynaklanan bir hastalıktır. Dünyada, meme kanseri kadınlar arasında en yaygın kanser türüdür. Kadınlar arasında yaşam süresi boyunca 13 ile 90 yaş arası, 13 kişiden biri ya da 9 kişiden biri bu hastalığa yakalanmaktadır (Aleksandroviç, Ryazanov, 2016). Diğer birçok kanserde olduğu gibi, meme kanserinin erken teşhisi hayatı kurtarabilir. Dolayısıyla meme kanserinin erken evrelerde kesin tanı koyulması hastanın yaşam kalitesini mümkün olan en iyi seviyede tutmak için çok önemlidir. Bununla birlikte, düzenli mamogramlar bile bu hastalığın zamanında teşhisini garanti etmez. ABD bilim adamları, göğüs yoğunluğunu otomatik olarak sınıflandırmak ve böylece meme kanserini tespit etmek için veri tabanlı yazılım geliştirmişler. Testler bu sistemin insan radyologları kadar doğru bir "teşhis uzmanı" olduğunu göstermiştir (Wolberg, Street, Mangasarian, 1992). Bu algoritma, göğüsün yoğunluğunun net bir tanıya izin vermediği durumlarda doktorlara yardımcı olabilir. Makine öğrenme modellerinin uygulanması hastalık tahmini ve prognozu için daha sonradan hastaların tedavisini iyileştirmeyi amaçlayan kanser çalışmalarının ayrılmaz bir parçası haline geldi. İlgili meme kanseri çalışmalarından elde edilen iki veri seti, iyi performans gösteren ve veri kaybı olmayan uygun verileri ve grafiksel veritabanları kullanarak yatay ve dikey entegrasyona dayalı bir veri entegrasyonu yaklaşımı uygulanarak birleştirilir.

Donald Hebb'in (1949) modern sinir ağları teorisini bulduğu bilinmektedir. Nörolog Hebb beynin nasıl öğrendiğini inceledi. Beynin çalışmasının en temel birimi sinir hücresi iki sinir hücresi birbiriyle nasıl ilişkilidir ve sinir ağları teorisini bu temele dayandırdı. Hebb'in bu temele dayanarak fikir başlatıldı ve yüzlerce teoriye sahip olmaktadır. Günümüzde gerçek hayatımızda kullanılan başarı oranı %99 olan birçok yapay sinir ağı (YSA) modeli vardır. Yapay sinir ağı ile makine öğrenmesi , görüntü işleme (Shi, He, 2010 ve Ramirez-Quintana, Chacon-Murguia, Chacon-Hinojo 2012), karakter tanıma, sınıflandırma, tahmin, kümeleme, ses işleme (Uncini, 2003), veri filtreleme ve en uygun şekle sokma gibi birçok uygulama yapmak mümkündür. Bu alanlarda yapay sinir ağlarının tercih edilmesinin temel nedenlerinden biri, kullanılan algoritma ne olursa olsun her tür veri, öğrenme hatalarını en aza indirmek ve bu nedenle gerçekçi tahmin yapabilmektedir. Meme kanserinde klinik veri seti temelinde doğru tahmin yapılabilmesi için Yapay Sinir Ağı modelinin doğru optimizasyon algoritması ile uygulanması ve parametre aralıklarının doğru belirlenmesi kritik önem taşır. Bu nedenle çalışmamızda aşağıdaki optimizasyon algoritmalarının nasıl sonuç verdiği araştırıldı:

1.1 Stokastik gradyan inişi (Stochastic Gradient Descent SGD)

Stokastik gradyan inişi (SGD) derin öğrenmede, nesnel işlev genellikle eğitim veri setindeki her örnek için kayıp işlevlerinin ortalama değeridir. $F_i(x)$ n veri, indeks i ve parametre vektörü x ile eğitim verisi örneğinin bir kayıp fonksiyonu olduğunu varsayıyoruz, o zaman objektif fonksiyonumuz var.

$$f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \quad (1)$$

x'deki objektif fonksiyonun gradyanı şu şekilde hesaplanır:

$$\nabla f(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x) \quad (2)$$

Degrade iniş kullanılırsa, bağımsız değişkenin her yinelemesi için hesaplama maliyeti, n ile doğrusal olarak büyüyen O(n)'dir. Bu nedenle, modelin eğitim verilerinin örneği büyük olduğunda, her bir yineleme için degrade iniş maliyeti çok yüksek olacaktır.

Stokastik gradyan inişi (SGD) her bir yinelemenin hesaplama maliyetini azaltır. Stokastik gradyan inişin her yinelemesinde, rastgele veri örnekleri için $i \in \{1, \dots, n\}$ indeksini eşit olarak seçeriz ve x'i güncellemek için $\nabla f_i(x)$ gradyanını hesaplarız:

$$x \leftarrow x - \eta \nabla f_i(x) \quad (3)$$

Burada η öğrenme oranıdır. Her bir yineleme için hesaplama maliyetinin O(n) gradyan inişinden sabit O(1)'e düştüğünü görebiliriz. Stokastik gradyan $\nabla f_i(x)$ 'nin $\nabla f(x)$ gradyanının tarafsız bir tahmini olduğu unutulmamalıdır (Ruder 2017).

$$E_i \nabla f_i(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x) = \nabla f(x) \quad (4)$$

Bu, ortalama olarak, stokastik gradyanın gradyanı iyi bir tahmini olduğu anlamına gelir.

1.2 Adagrad

s_t değişkenini, geçmiş gradyan varyansını aşağıdaki gibi biriktirmek için kullanırız.

$$\begin{aligned} g_t &= \partial_{\omega} l(y_t, f(x_t, \omega)), \\ s_t &= s_{t-1} + g_t^2, \\ \omega_t &= \omega_{t-1} - \frac{\eta}{\sqrt{s_t + \epsilon}} \cdot g_t. \end{aligned} \quad (5)$$

η öğrenme oranıdır ve ϵ , 0'a bölmememizi sağlayan bir katkı sabitidir. Son olarak, $s_0 = 0$ değerini başlatırız.

Momentumda olduğu gibi, her bir koordinat için bireysel öğrenme hızını dikkate almak için bu durumda yardımcı değişkeni izlememiz gerekir. Bu, Adagrad'ın maliyetini SGD'ye kıyasla önemli ölçüde artırmaz, çünkü ana maliyet genellikle $l(y_t, f(x_t, w))$ ve türevinin hesaplanmasından oluşur. Momentumda olduğu gibi, yardımcı bir değişkeni izlememiz gerekir, bu durumda koordinat başına bireysel bir öğrenme oranına izin vermek için Adagrad'ın $l(y_t, f(x_t, w))$, ve türevini hesaplaması olduğundan, Adagrad'ın SGD'ye göre maliyetini önemli ölçüde artırmaz.

1.3 RMSprop

RMSprop algoritması, hız planlamasını koordinat uyarlamalı öğrenme hızlarından ayırmaya izin veren basit bir düzeltme olarak kullanılmaktadır. Sorun, Adagrad'ın $g_t = s_t - 1 + g_t^2$ durum vektöründe g_t gradyanının karelerini biriktirmesidir. Sonuç olarak, algoritma yakınsadığı için, s_t normalleşme eksikliği nedeniyle, esasen doğrusal olarak, kısıtlamalar olmadan büyümeye devam eder (Ruder 2017). Bu sorunu çözenin bir yolu s_t / t kullanmak olacaktır. Makul g_t dağıtımları için, bu yakınsama yapacaktır. Ne yazık ki, prosedür değerlerin tam yörüngesini hatırladığından, limitin davranışının önemli hale gelmesi çok uzun zaman alabilir. Bir alternatif, ortalama sızdıran değerini bazı parametreler için $s_t \leftarrow \gamma s_{t-1} + (1 - \gamma) g_t^2$ $\gamma > 0$ ve diğer tüm parçaları değişmeden tutmak RMSprop verir.

$$\begin{aligned} s_t &\leftarrow \gamma s_{t-1} + (1 - \gamma) g_t^2 \\ x_t &\leftarrow x_{t-1} - \frac{\eta}{\sqrt{s_t + \epsilon}} \odot g_t \end{aligned} \quad (6)$$

$\epsilon > 0$ sabiti genellikle sıfır veya çok büyük adım boyutlarına bölünmememizi sağlamak için 10^{-6} olarak ayarlanır. Bu genişleme göz önüne alındığında, öğrenme hızını η koordinat başına uygulanan ölçeklemeden bağımsız olarak kontrol etmektedir.

1.4 Adadelta

Adadelta, AdaGrad'ın başka bir sürümüdür. Aralarındaki fark, öğrenme hızının koordinatlara uyarlanma miktarını azaltmasıdır. Ayrıca, geleneksel olarak bir öğrenme oranına sahip olmadığından değişim miktarını gelecekteki değişim için kalibrasyon olarak kullanır. Özetle, Adadelta iki durum değişkeni kullanır: s_t gradyanın ikinci momentinin ortalama sızıntısını depolamak için ve Δx_t ikinci değişiklik anının ortalama sızıntısını modelin kendisinde saklamak için.

$$\begin{aligned} s_t &= p s_{t-1} + (1 - p) g_t^2, \\ g'_t &= \sqrt{\frac{\Delta x_{t-1} + \epsilon}{s_t + \epsilon}} \odot g_t, \\ x_t &= x_{t-1} - g'_t, \\ \Delta x_t &= p \Delta x_{t-1} + (1 - p) x_t^2. \end{aligned} \quad (7)$$

Bir öncekinden farkı, değişim oranının ortalama karesi ile gradyan ortalama ikinci momenti arasındaki ilişki alınarak hesaplanan değiştirilmiş bir gradyan g'_t ile güncellemeler gerçekleştirilmizdir. g'_t kullanımı yalnızca tanımlama kolaylığı içindir. Uygulamada, bu algoritmayı g'_t için ek geçici alan kullanmak zorunda kalmadan uygulayabiliriz. Daha önce olduğu gibi, η , önemsiz olmayan sayısal sonuçlar sağlayan, yani sıfır adım büyüklüğünden veya sonsuz varyanstan kaçınan bir parametredir. Genel olarak, bunu $\eta = 10^{-5}$ olarak ayarlanır.

1.5 Adam

Adam'ın temel bileşenlerinden biri, hem momentumun hem de gradyanın ikinci momentinin bir tahminini elde etmek için üstel sızdıran ortalamalar kullanmasıdır. Yani, durum değişkenlerini kullanır

$$\begin{aligned} v_t &\leftarrow \beta_1 v_{t-1} + (1 - \beta_2) g_t^2 \\ s_t &\leftarrow \beta_2 s_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \quad (8)$$

Burada β_1 ve β_2 negatif olmayan ağırlık parametreleridir. Onlar için normal seçim: $\beta_1 = 0.9$ ve $\beta_2 = 0.999$. Yani, varyans tahmini momentum teriminden çok daha yavaş hareket eder. Eğer $v_0 = s_0 = 0$ değerini başlatırsak, başlangıçta daha düşük değerlere önemli bir önyargıya sahip oluruz. Bu, terimleri yeniden normalleştirmek için $\sum_{i=0}^t \beta^i = \frac{1-\beta^{t+1}}{1-\beta}$ kullanılarak çözülebilir. Buna göre, normalize edilmiş durum değişkenleri

$$\hat{v}_t = \frac{v_t}{1-\beta_1^t} \text{ ve } \hat{s}_t = \frac{s_t}{1-\beta_2^t} \quad (9)$$

Şimdi güncelleme denklemlerini yazabiliriz ve ilk olarak, gradyanı elde etmek için RMSProp'a çok benzer bir şekilde yeniden ölçeklendirilir

$$g'_t = \frac{\eta \hat{v}_t}{\sqrt{\hat{s}_t + \epsilon}} \quad (10)$$

RMSprop'tan farklı olarak, güncellememiz gradyanın kendisi yerine \hat{v}_t momentumunu kullanır. Dahası, yeniden ölçekleme $\frac{1}{\sqrt{\hat{s}_t + \epsilon}}$ yerine $\frac{1}{\sqrt{\hat{s}_t + \epsilon}}$ kullanarak gerçekleştiği için küçük bir kozmetik farkı vardır. Önceki pratikte pratikte biraz daha iyi çalışıyor, bu nedenle RMSProp'dan sapma genellikle sayısal kararlılık ve sadakat arasında iyi bir denge için $\epsilon = 10^{-6}$ seçeriz. Güncellemeleri hesaplamak için tüm parçalarımız var, bu biraz antiklimaktiktir ve formun basit bir güncellemesine sahibiz. Sonra, Adadelta ve RMSprop'ta gördüğümüz gibi parametreleri güncellemek için kullanılır. Adam güncelleme kuralını verir:

$$\omega_{t+1} = \omega_t - \frac{\eta \hat{v}_t}{\sqrt{\hat{s}_t + \epsilon}} \quad (11)$$

1.6 AdaMax

Adam güncelleme kuralındaki s_t faktörü gradyanı l_2 normuna göre ters orantılı olarak ölçeklendirir ve geçmiş gradyanları (v_{t-1} terimi üzerinden) ve geçerli gradyan $|g_t|$ 'dir (Ruder 2017).

$$s_t \leftarrow \beta_1 s_{t-1} + (1 - \beta_2) |g_t|^2 \quad (12)$$

Bunu Adam güncellemesine uyarlayarak AdaMax güncelleme kuralını elde ederiz.

$$\omega_{t+1} = \omega_t - \frac{\eta}{s_t} \hat{v}_t \omega_{t+1} = \omega_t - \frac{\eta}{s_t} \hat{v}_t \quad (13)$$

Burada

$$s_t = \beta_2^\infty s_{t-1} + (1 - \beta_2^\infty) |g_t|^\infty = \max(\beta_2 s_{t-1}, |g_t|) \quad (14)$$

1.7 Nadam

Nadam (Nesterov hızlandırılmalı Uyarlanabilir Moment Tahmini) Adam ve NAG'yi birleştirir. İçinde Nesterov hızlandırılmış gradyan (NAG) Adam'a dahil etmek için, momentum terimini değiştirmemiz gerekiyor (Ruder 2017).

$$\begin{aligned} g_t &= \nabla_{\omega_t} J(\omega_t) \\ m_t &= \gamma m_{t-1} + \eta g_t \\ \omega_{t+1} &= \omega_t - m_t \end{aligned} \quad (15)$$

Burada objektif fonksiyonumuz J , momentum bozulma terimi γ ve η adım boyutudur. Yukarıdaki üçüncü denklemin genişlenmesi şunu verir:

$$\omega_{t+1} = \omega_t - (\gamma m_{t-1} + \eta g_t) \quad (16)$$

Momentum vektörü ve mevcut gradyan yönünde bir adım içerdiğini bir kez daha göstermektedir. Nesterov hızlandırılmış gradyanı güncelleyerek gradyan yönünde daha kesin bir adım atmamızı sağlar. Bu nedenle, yalnızca g_t de NAG değerini bulmak için geçerli parametreleri güncellemek için ileriye yönelik momentum vektörünü doğrudan uygulayarak:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \omega_{t+1} &= \omega_t - \frac{\eta \hat{m}_t}{\sqrt{s_t + \epsilon}} \end{aligned} \quad (17)$$

\widehat{m}_{t-1} momentum vektörü akımın önyargı düzeltilmiş tahmini ile \widehat{m}_t momentum vektörü Nadam güncelleme kuralını verir.

$$\omega_{t+1} = \omega_t - \frac{\eta}{\sqrt{s_t + \epsilon}} \left(\beta_1 \widehat{m}_t + \frac{(1-\beta_1)g_t}{1-\beta_1^t} \right) \quad (18)$$

2. LİTERATÜR ARAŞTIRMASI

Bu konuyla ilgili birçok çalışma yapılmıştır. Aslında makine öğrenimi matematiksel istatistiklerin birleştiği yerde, optimizasyon yöntemleri ve klasik matematiksel disiplinler, aynı zamanda hesaplama verimliliği ve yeniden eğitim sorunları ile ilişkili kendi özellikleri vardır. Birçok endüktif eğitim yöntemi klasik istatistiksel yaklaşımlara alternatif olarak geliştirilmiştir. Yapılan çalışmalarda birçok yöntem, bilgilerin çıkarılmasıyla ve yapay veri analizi (Data Mining) ile yakından ilgilidir.

Xrulyov K.A.ve Ryazanov M.A. (2016) Azure Machine Learning ile meme kanseri tanısında teşhis için incelenen hastalar hakkındaki verilerin analizini kullanarak bir web servisi geliştirmişler.

Fogel D. B., Wasson E.C., Boughton E.M. ve Porto V.W. (1997) hasta yaşına sahip sinir ağları ile radyoaktif özellikleri kullanarak meme kanseri tespiti için veri analizi çalışmasını yapmışlardır.

Revett K., Gorunescu F., Gorunescu M., El-Darzi E. ve Ene M.,(2005) ve Gorunescu M., Gorunescu F., ve Revett K.,(2007) ham kümeler ve muhtemel sinir ağları içeren hibrid bir modele dayanan bir meme kanseri tıbbi modeli için bir karar destek sistemi geliştirmişler.

Hsiao Y.H., Huang Y.L., Liang W.M., Kuo S.J. and Chen D.R., (2009) vasküler parametreler (harmonik ve harmonik olmayan 3D Dopplerografi) kullanarak iyi veya kötü huylu göğüs tümörlerinin belirlenmesi için bir MLP sınıflandırıcı analizi çalışmasını yapmışlardır.

E.Harwich, K.Laycock., (2018) ve JASON The MITRE Corporation (2017) "Birleşik Krallık'da İngiliz bilim adamları "Ulusal Sağlık Sisteminde Yapay Zeka" ve ABD' nin önde gelen amerikalı teknoloji bilim adamı Jason "Sağlık ve Sağlık Hizmetleri İçin Yapay Zeka" adlı 2017 yılında çalıştıkları bir rapor yayınladı. Her iki çalışmada Yapay Zeka kullanarak genel nüfusa yüksek nitelikli tıbbi bakım sağlanması analiz edilmiştir. Kanser tanısı alanında Yapay Zeka kullanımı, Yapay Zekanın görevleri ve yöntemleri hakkında çalışma yapılmıştır.

Mihaylov.I , Nisheva.M , and Vassilev.D (2019) doğru teşhis için makine öğrenme modellerini kullanarak meme kanserinde sağ kalım süresinin klinik verilere dayanarak öngörülmesini sağlayan çalışma yapılmıştır. Bu çalışmada hastanın hayatta kalma süresinin, tümör evresini, tümör boyutunu ve yaşının tanısını orijinal olarak geliştirilen tümörle entegre klinik özellik olduğunu tahmin etmektedir. Çalışmada veri normalizasyonu ve sınıflandırmasının yanı sıra, uygulamalı makine öğrenimi yöntemi sağ kalım süresi tahmininin doğruluğu açısından umut verici sonuçlar vermektedir. Bu çalışmada doğrusal destek vektör regresyonu, çekirdek ridge regresyonu, K en yakın komşu regresyonu, karar ağacı regresyonu ve kement regresyonu modelleri en doğru yaşam prognozu sonuçlarını elde etmişler. Aynı yöntemleri kullanarak meme kanseri verileri üzerindeki performansı için önerilen yaklaşım olarak Python tabanlı iş akışı geliştirmişler.

3. MEME KANSERİ TAHMİNİNDE KULLANILAN VERİ SETİ

Yapay Sinir Ağını Kullanarak Meme kanseri Teşhisinde kullanılan veri seti Kaliforniya Üniversitesi (UCI) Makine Öğrenimi Deposundan alınmış meme kanseri (BC) veritabanıdır (Wolberg, Street, Mangasarian,1992). Makine öğrenme algoritmalarının deneysel analizi için kullanılacak birçok veri kümesiyle açık veri havuzu Madison'daki Wisconsin Üniversitesi Hastanesi'nden Dr. William H. Wolberg tarafından oluşturulmuş veri setidir. Bu çalışmada kullanılan veri kümesinde bulunan özelliklerden bazıları yarıçap, doku, çevre, yumuşaklık, kompaktlık, alan, içbükeylik, içbükey noktalar, her hücre çekirdeği için fraktal boyut, simetri ve veri seti için kullanılan UCI makine öğreniminde 569 örnek ve 32 fonksiyondan oluşan bir Wisconsin Diagnostic Breast Cancer (WDBC) veri setidir. WDBC veri setindeki 699 meme kanseri verisinin 458 tanesi iyi huylu (benign) ve 241 tanesi kötü huylu (malignant) kanser hücrelerinin örneklerini göstermektedir. Veri kümesinde iyi huylu kanser hücrelerinin dağılımı daha homojendir ve kötü huylu kanser hücrelerinde yapısal maligniteler bulunur. Veritabanında toplam 11 öznelik ve değer aralıkları aşağıdaki tabloda gösterilmektedir.

Tablo1. Veri Setleri ve değer aralıkları

Veri seti	Değer aralığı
Sample code number	ID Numarası
Clump Thickness	1-10
Uniformity of Cell Size	1-10
Uniformity of Cell Shape	1-10
Marginal Adhesion	1-10
Single Epithelial Cell Size	1-10
Bare Nuclei	1-10
Bland Chromatin	1-10
Normal Nucleoli	1-10
Mitoses	1-10
Class	2 / 4

Meme kanseri teşhisi uygulaması tahmininde kullanılan özneliklerin id numarasından sonraki on tanesi 1 ile 10 arasındaki özneliklerdir. Son değerimiz ise sonuç kısmı olarak görülmektedir ve eğer iyi huylu ise 2, kötü huylu ise 4 değerini almaktadır.

Kullanılacak veri setinin ilk on verisi Tablo 2' de yer almaktadır.

Tablo 2: Veri Setinde Yer Alan İlk On Veri

Sample code number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2
1017122	8	10	10	8	7	10	9	7	1	4
1018099	1	1	1	1	2	10	3	1	1	2
1018561	2	1	2	1	2	1	3	1	1	2
1033078	2	1	1	1	2	1	1	1	5	2
1035283	1	1	1	1	1	1	3	1	1	2

Kullanılacak veri setinde verilerin id numarası sonuca herhangi bir değişiklik oluşturmayacağından dolayı bu alanı çıkarıyoruz ve iyi huylu mu kötü huylu mu olup olmadığı kısmı içinde yapay sinir ağı modelimiz iki çıkışa sahip olduğundan dolayı iki alan ekleyip bu durumu belirtmiş olmamız gerekmektedir.

Son durumda tablo 2’de yer alan on tane verinin uygulama tarafından anlaşılacağı formatı tablo 3’teki gibidir.

Tablo 3: YSA Modeli İçin Yeniden Düzenlenen Veri Setinin İlk On Verisi

5	1	1	1	2	1	3	1	1	1
5	4	4	5	7	10	3	2	1	1
3	1	1	1	2	2	3	1	1	1
6	8	8	1	3	4	3	7	1	1
4	1	1	3	2	1	3	1	1	1
8	10	10	8	7	10	9	7	1	0
1	1	1	1	2	10	3	1	1	1
2	1	2	1	2	1	3	1	1	1
2	1	1	1	2	1	1	1	5	1
1	1	1	1	1	1	3	1	1	1

4. KULLANILAN YAPAY SİNİR AĞI MODELİ

Bu çalışmada veri setinde hastalığı teşhis etmek için 9 öznitelik vardır. Bu sebepten giriş katmanında 9 nöron vardır ve tümörler iyi huylu veya kötü huylu olarak ayrıldığı için çıkış katmanında 2 nöron ve ara katmanda 10 nöron bulunmaktadır. Uygulamanın başlangıcında rastgele bias ve ağırlık değerleri oluşturulmuş ve uygulama süresinde bu değerler güncellenerek son değerler bulunmuştur. Uygulamada çıkış nöronlarında hatanın geri yayılması nedeniyle değerlendirmede hata olasılığı en aza indirilmiştir.

Aktivasyon fonksiyonu olarak, ayırt edilmesi kolay olduğu için sigmoid fonksiyonu tercih edilmiştir.

$$\text{sigmoid}(s) = \frac{1}{1+e^x} \quad (19)$$

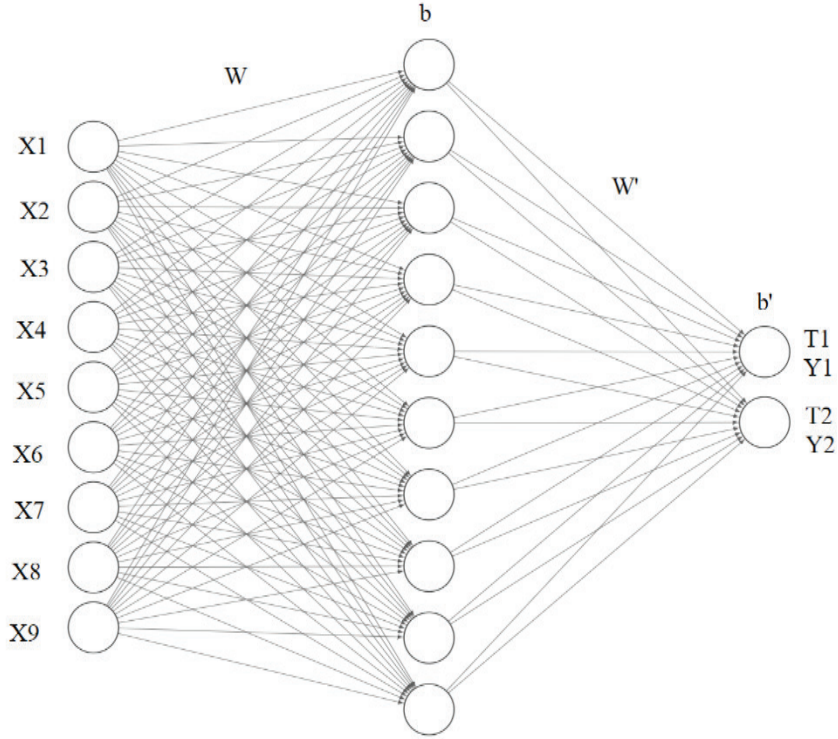
Aşağıdaki formüllere göre hesaplamalar yapılmıştır.

$$\theta_n = f(\sum_{i=1}^9 x_i \cdot \omega_{in} + b_n), n = 1, \dots, 10. \quad (20)$$

Burada b_n ve ω_{in} ara katmanın bias ve ağırlıklarını, θ_n ara katmanın çıkışı anlamına gelmektedir.

$$Y_n = f(\sum_{i=1}^{10} \theta_i \cdot \omega'_{in} + b'_n), n = 1, 2. \quad (21)$$

Burada Y_n son katmanın çıkışını temsil etmektedir. b_n ve ω_{in} ise çıkış katmanı biası ve ağırlıklarıdır. Burada öğrenme oranı 0.5 olarak alınmıştır.



Şekil 1 Çalışmadaki Yapay Sinir Ağı Modeli

Aşağıdaki formüllerle çalışmadaki güncellemeler yapılmıştır.

$$\delta_i = (T_j - Y_j) \cdot Y_j \cdot (1 - Y_j)$$

$$\Delta W'_{ij} = (t + 1) = 0.5 \cdot \delta_j \cdot \theta_{ij} = 1, 2 \text{ ve } i = 1, \dots, 10.$$

$$W'_{ij}{}^{yeni} = W'_{ij}{}^{eski} + \Delta W'_{ij}(t + 1)$$

$$\Delta W_{ij}(t + 1) = 0.5 \cdot \theta_j \cdot (1 - \theta_j) \cdot \delta_{1,2} \cdot W_{ij} \cdot X_i, \quad j = 1, \dots, 10 \text{ ve } i = 1, \dots, 9.$$

$$W_{ij}{}^{yeni} = W_{ij}{}^{eski} + \Delta W_{ij}(t + 1) \quad (22)$$

$$\Delta b'_{ij}(t + 1) = 0.5 \cdot \delta_j, i = 1, 2.$$

$$b'_{ij}{}^{yeni} = b'_{ij}{}^{eski} + \Delta b'_{ij}(t + 1)$$

$$\Delta b_{ij}(t + 1) = 0.5 \cdot \theta_j \cdot (1 - \theta_j) \cdot \delta_{1,2} \cdot W_{j,i}, i = 1, \dots, 10 \text{ ve } j = 1, \dots, 9.$$

$$b_{ij}{}^{yeni} = b_{ij}{}^{eski} + \Delta b_{ij}(t + 1)$$

$$hata = \frac{1}{2} \sum_{i=1}^2 (t_n - y_n)^2$$

Hedefdeki çıkış değerlerini formülde $t_{1,2}$, ifade etmektedir.

5. YAPAY SİNİR AĞINDA KULLANILAN OPTİMİZASYON ALGORİTMASININ SEÇİMİ VE PARAMETRELERİN AYARLANMASI

5.1 Optimizasyon Algoritması

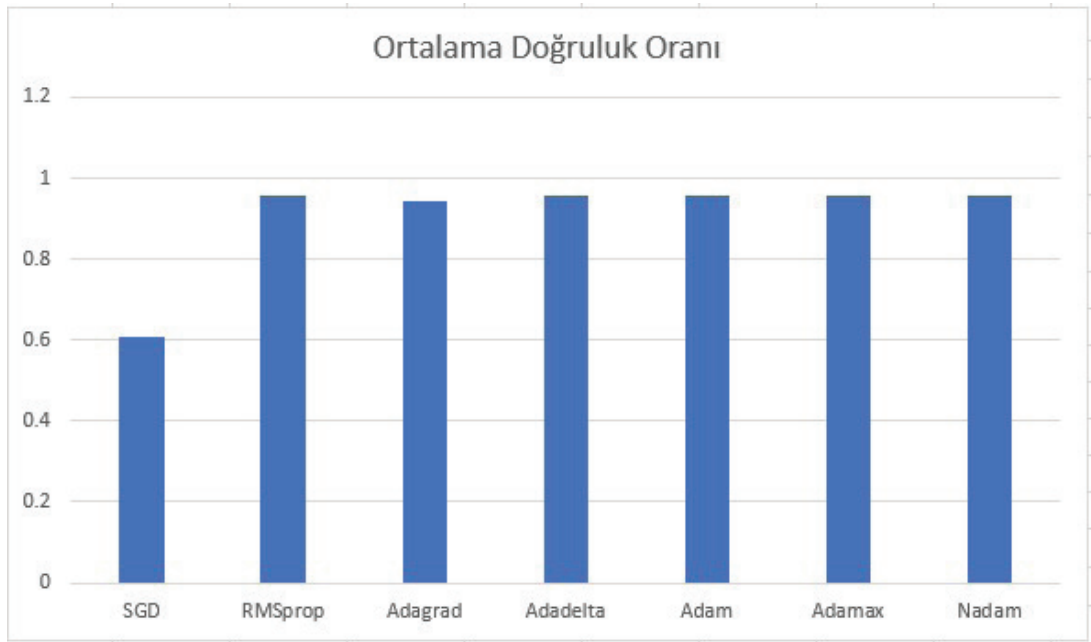
Optimizasyon yöntemi olarak 'SGD' (Mei, 2018), 'RMSprop' (Teileman ve Hinton, 2012), 'Adagrad' (Duchi, Hazan, ve Singer, 2011), 'Adadelata' (Zeiler, 2012), 'Adam' (Diederik ve Ba, 2014), 'Adamax' (Diederik ve Ba, 2014) ve 'Nadam' (Dozat, 2016) algoritmalarından en iyi doğruluk oranı vereni tespit etmek için hepsi ile testler gerçekleştirildi. Tablo 4 bu algoritmaların kullandığı formülleri listelemektedir.

Tablo 4: Optimizasyon Fonksiyonlarının Formülleri

Optimizasyon Algoritması	Formül
Stochastic Gradient Descent (SGD)	$Q(w) = \frac{1}{n} \sum_{i=1}^n Q_i(w),$
Root Mean Square Propagation (RMSprop)	$w := w - \frac{\eta}{\sqrt{v(w, t)}} \nabla Q_i(w)$
Adaptive Gradient Descent (Adagrad)	$G = \sum_{\tau=1}^t g_{\tau} g_{\tau}^T$
Adaptive Learning Rate (Adadelata)	$\Delta x_t = -\frac{\eta}{\text{RMS}[g]_t} g_t$
Adaptive Moment Estimation (Adam)	$w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w + \epsilon}}$
Adaptive Moment Estimation Maximum Adamax	$w_{t+1} = w_t - \frac{\alpha}{S_t} \cdot \hat{V}_t$
Nesterov-accelerated Adaptive Moment Estimation (Nadam)	$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{S}_t + \epsilon}} \left(\beta_1 \hat{V}_t + \frac{1 - \beta_1}{1 - \beta_1^t} \cdot \frac{\partial L}{\partial w_t} \right)$

Tablo 5: Farklı optimizasyon yöntemleri için ortalama doğruluk oranı

Optimayzer	Ortalama doğruluk oranı
SGD	0.607
RMSprop	0.957
Adagrad	0.942
Adadelta	0.957
Adam	0.957
Adamax	0.957
Nadam	0.957

**Şekil 2.** Farklı optimizasyon yöntemleri için ortalama doğruluk oranı grafiği

Tablo 5 ve Şekil 2'deki grafikte görüldüğü üzere SGD ve Adagrad dışındakiler birbirine yakın doğruluk oranı vermektedirler ve bu uygulamada tercih edilebilirler. Birbirine yakın doğruluk değerler verseler de en iyi doğruluğu ADAM verdiği için bundan sonraki incelemeler ADAM algoritması kullanılarak yapıldı.

5.2 Batch Size

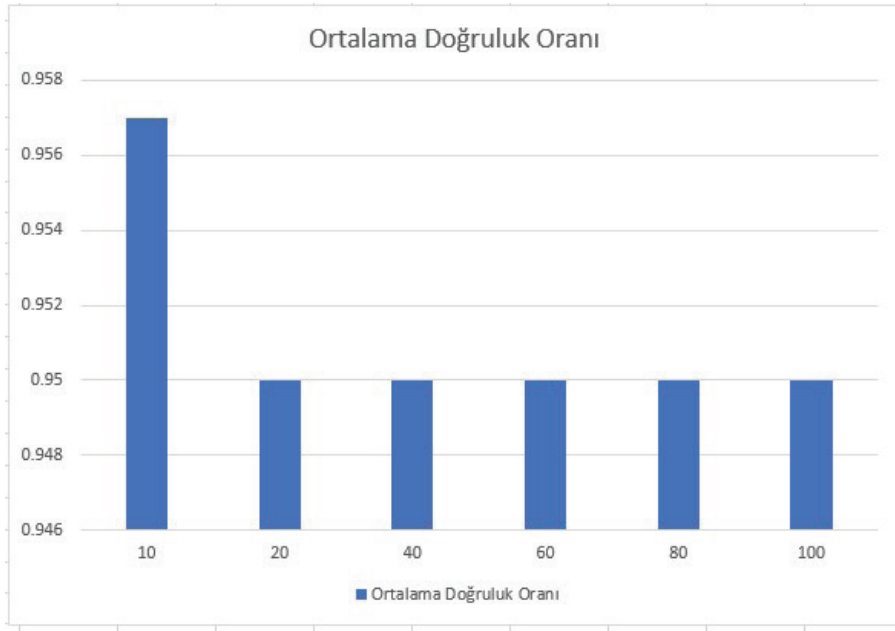
Grup boyutu (Batch Size) ağ üzerinden dağıtılacak örnek sayısını belirlemektedir. Örneğin, 1050 eğitim örneğiniz olduğunu ve batch_size değerini 100 olarak ayarlamak istediğinizi varsayalım. Algoritma ilk 100 örneği (1'den 100'e kadar) eğitim veri kümesinden alır ve ağı eğitir. Sonra ikinci 100 örneği (101'den

200'e kadar) alır ve ağı tekrar eğitir. Tüm örnekleri ağ üzerinden dağıtana kadar bu işlemi yapmaya devam edebiliriz. Son örnek kümesinde bir sorun ortaya çıkabilir. Örneğimizde, geriye kalan 100'e bölünmeyen 1050'yi kullandık. En basit çözüm, son 50 örneği alarak ağı eğitmektir.

Numunelerin gruplara bölerek kullanmanın faydaları şunlardır. Ağı daha az örnek kullanarak eğittiğiniz için, genel eğitim prosedürü daha az bellek gerektirir. Bu, tüm veri setinin makinenin belleğine sığmadığı durumlarda özellikle önemlidir. Ağlar genellikle mini paketlerle daha hızlı öğrenir. Bunun nedeni, her yayılmadan sonra ağırlığı güncellememizdir. Örneğimizde 11 paket dağıttık (10 tanesi 100 örnek ve 1 tanesi 50 örnek vardı) ve her birinin ardından ağımların parametrelerini güncelledik. Tüm örnekleri dağıtım sırasında kullansaydık, ağ parametresi için sadece 1 güncelleme yapılırdı.

Tablo 6: Batch Size için ortalama doğruluk oranı

Batch Size	Ortalama doğruluk oranı
10	0.957
20	0.950
40	0.950
60	0.950
80	0.950
100	0.950



Şekil 3. Batch Size için ortalama doğruluk oranı grafiği

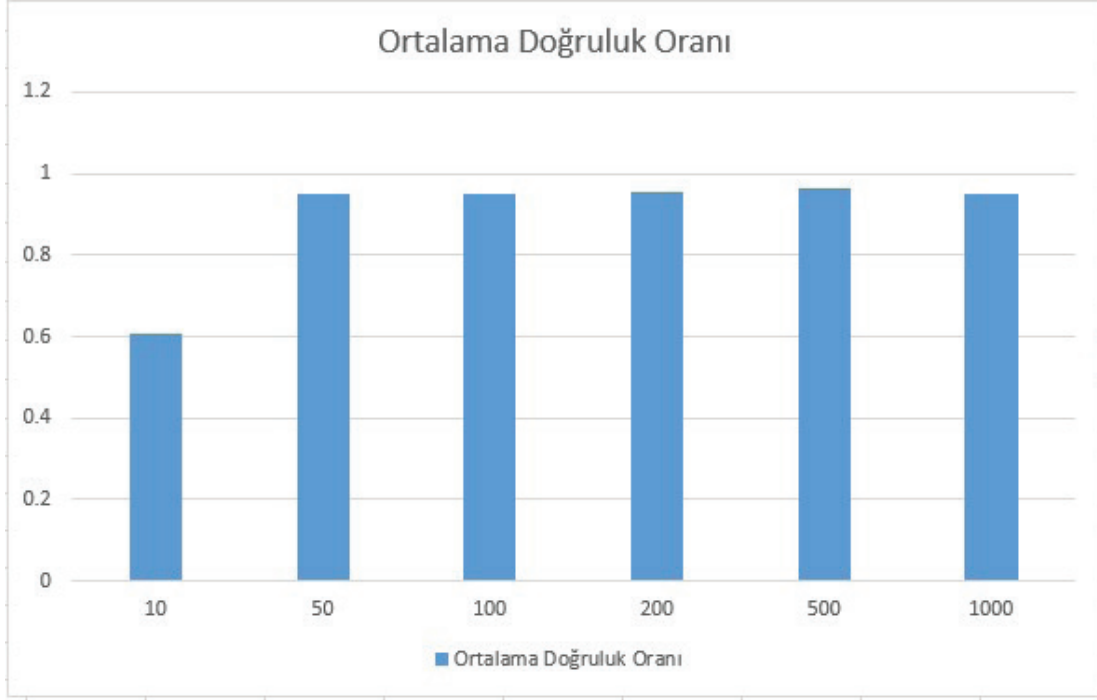
Tablo 6 ve Şekil 3'teki grafiğe göre en iyi sonuç batch size 10 olduğunda bulunuyor.

5.3 Epoch

Devir sayısı (Epoch) öğrenme algoritmasının tüm eğitim verileri kümesinde kaç kez çalışacağını belirleyen bir hiperparametredir. Bir epoch, eğitim veri setindeki her örneğin modelin dahili parametrelerini güncelleme fırsatı bulduğu anlamına gelir. Epoch bir veya daha fazla gruptan (batch) oluşur. Örneğin, yukarıda belirtildiği gibi, bir grup olduğu bir döneme grup gradyanı iniş (batch gradient descent) öğrenme algoritması denir. Her döngünün bir dizi eğitim verisinden geçtiği devir sayısı için bir for döngüsü düşünebiliriz. Bu döngü için, her bir numune grubu üzerinde yinelenen başka bir iç içe for döngüsü vardır, burada bir grup belirli sayıda "grup boyutu (batch size) " örneğine sahiptir. Devir sayısı geleneksel olarak büyüktür, genellikle yüzlerce veya binlercedir, bu da öğrenme algoritmasının model hatası en aza indirilene kadar çalışmasına izin verir. Literatürde devir sayısının örneklerini 10, 100, 500, 1000 ve daha büyük olarak örnekleri vardır. Tipik olarak, X eksenini boyunca evreleri Y eksenini üzerinde zaman ve hata veya model becerisi olarak gösteren çizgi grafikler oluşturulur ve bu grafiklere bazen öğrenme eğrileri denir. Bu grafikler, modelin yeniden eğitilip öğrenilmediğini, iyi anlaşılmadığını veya bir eğitim veri kümesi için uygun olup olmadığını teşhis etmeye yardımcı olabilir.

Tablo 7: Epoch için ortalama doğruluk oranı

Epoch	Ortalama doğruluk oranı
10	0.607
50	0.950
100	0.950
200	0.957
500	0.964
1000	0.950



Şekil 4. Epoch için ortalama doğruluk oranı grafiği

Tablo 7 ve Şekil 4'te görüldüğü gibi epoch 500 olunca doğruluk oranı en iyi oluyor.

6. SONUÇ

Yapay sinir ağlarının doğrusal bir yapısı olmadığından dolayı, bu makalede bahsettiğimiz değişkenlerin doğru belirlenmesi daha fazla önem arz etmektedir. Bu parametre değerleri için net ve sabit bir değer verilemeyeceği gibi yapay sinir ağının eğitiminde kullanılan veri setinin türüne göre de değişiklik göstermektedir. Yaptığımız çalışmalarda optimizasyon algoritmasının ve parametre değerlerinin sonucu nasıl etkilediğini gördük. Başlangıç parametrelerimizi güncellediğimizde ortalama doğruluk oranı 0.607'den 0.96'ya yükselmiştir. WDBC veri setindeki öznitelikler ile yapay sinir ağları kullanarak meme kanseri tahmini için optimizasyon algoritması tercihi ve parametre ayarlarının aşağıdaki gibi olması, testlerimizde en iyi doğruluk oranını vermiştir.

- Optimizasyon Algoritması: ADAM
- Grup Boyutu (Batch Size): 10
- Devir Sayısı (Epochs): 500

7. Kaynaklar

- Aleksandroviç, X.K., and M.A. Ryazanov.** 2016. available in <http://elibrary.asu.ru/xmlui/bitstream/handle/asu/2682/vkr.pdf?sequence=1&isAllowed=y>, last accessed November, 2020.
- Duchi, J., E. Hazan, and Y. Singer.** 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Machine Learning Research*, 12, 2121-2159.
- Fogel, D.B., E.C. Wasson, E.M. Boughton, and V.W. Porto.** 1997. A step toward computerassisted mammography using evolutionary programming and neural networks., *Cancer Letters*, 119 (1), 93-97.
- Gorunescu, M., F. Gorunescu, and K. Revett.** 2007. Investigating a Breast Cancer Dataset Using a Combined Approach: Probabilistic Neural Networks and Rough Sets, *Proceedings of the 3rd ACM International Conference on Intelligent Computing and Information Systems -ICICIS07, Cairo, Egypt*, 246-249.
- Harwich, E., and K. Laycock.** 2018. Thinking on its own: AI in the NHS, available in <http://www.reform.uk/publication/thinking-on-its-own-ai-in-the-nhs/>, last accessed November, 2020.
- Hsiao, Y.H., Y.L. Huang, W.M. Liang, S.J. Kuo, and D.R. Chen.** 2009. Characterization of benign and malignant solid breast masses: harmonic versus nonharmonic 3D power Doppler imaging, *Ultrasound Medicine & Biology* 35(3), 353-359.
- Huang, G., Y. Sun, Z. Liu, D. Sedra, and K.Q. Weinberger.** 2016. Deep networks with stochastic depth. *Proceedings of the European Conference on Computer Vision*, Springer, 646–661.
- Huo, Z., and H. Huang.** 2017. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization, *Thirty-First AAAI Conference on Artificial Intelligence*
- Ishii, M., and A. Sato.** 2017 Layer-wise weight decay for deep neural networks, *Pacific-Rim Symposium on Image and Video Technology*. Springer, 276–289.
- Jason, The Mitre Corporation,** 2017. Artificial Intelligence for Health and Health Care, available in HYPERLINK"https://www.healthit.gov/sites/default/files/jsr-17-task-%20002_aiforhealthandhealthcare12122017.pdf"https://www.healthit.gov/sites/default/files/jsr-17-task-002_aiforhealthandhealthcare12122017.pdf , last accessed November, 2020.
- Mihaylov, I., M. Nisheva, and D. Vassilev.** 2019. Application of Machine Learning Models for Survival Prognosis in Breast Cancer Studies
- Nesterov, Y.** 1983. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$, *Doklady AN USSR* 269, 543–547.
- Ramirez-Quintana, J.A., M.I. Chacon-Murguia, and J.F. Chacon-Hinojos.** 2012. Artificial Neural Image Processing Applications: A Survey. *Engineering Letters*, 20(1), 68-81.

Revett, K., F. Gorunescu, F. M. Gorunescu, E. El-Darzi, and M. Ene. 2005. A breast cancer diagnosis system: a combined approach using rough sets and probabilistic neural Networks. Computer as a tool Eurocon, Belgrade, 1124- 1127.

Sebastian, R. 2017. An overview of gradient descent optimization algorithms, Insight Centre for Data Analytics, NUI Galway Aylien Ltd., Dublin

Shi Z., and L. He. 2010. Application of Neural Networks in Medical Image Processing, Proceedings of the Second International Symposium on Networking and Network Security (ISNNS '10), China, 2-4.

Smith, S.L., P.-J. Kindermans, C. Ying, and Q.V. Le. 2018. Don't decay the learning rate, increase the batch size, in International Conference on Learning Representations (ICLR)

Staelin, D.H., and C.H. Staelin. 2011. Models for Neural Spike Computation and Cognition. CreateSpace, Seattle, Washington.

Uncini A., 2003. Audio signal processing by neural Networks, Neurocomputing, (55) 3-4, 593 – 625.

Wolberg, W. H., W.N. Street, and O.L. Mangasarian. 1992. Breast cancer Wisconsin (diagnostic) data set. UCI Machine Learning Repository