

## Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması Ve Kiyaslanması

Ebubekir SEYYARER<sup>1</sup>, Faruk AYATA<sup>1</sup>, Taner UÇKAN<sup>1</sup>, Ali KARCI<sup>2</sup>

<sup>1</sup>Van Yüzüncü Yıl Üniversitesi, Bilgisayar Teknolojileri Bölümü, Van, Türkiye ([eseyyarer@yyu.edu.tr](mailto:eseyyarer@yyu.edu.tr), [farukayata@yyu.edu.tr](mailto:farukayata@yyu.edu.tr), [taneruckan@yyu.edu.tr](mailto:taneruckan@yyu.edu.tr))

<sup>2</sup>İnönü Üniversitesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye ([ali.karci@inonu.edu.tr](mailto:ali.karci@inonu.edu.tr))

Received Date : Jun.14, 2020.

Acceptance Date : Sep. 29, 2020.

Published Date : Dec. 1, 2020

**Özetçe:** Derin öğrenme (çok katmanlı yapay sinir ağı), katmanlı mimarisiyle büyük veri setlerinden öğrenme gerçekleştiren makine öğreniminin bir alt dalıdır. İşlemci hızları ve depolamadaki artış, ucuz bilgisayar donanımı ve makine öğreniminin kazanımları, derin öğrenme gelişimini yükselten faktörlerdir. Derin öğrenmede hatayı minimuma düşürmek için kullanılan gradyan iniş tabanlı optimizasyon algoritmaları da (Stochastic gradient descent (sgd), momentum, adam, adagrad, rmsprop ve adadelta), bu başarımın bir parçası olmaktadır.

Bu çalışmada uluslararası bir veri seti olan caltech 101 veri setine derin öğrenme ve optimizasyon algoritmaları uygulanmıştır. Aktivasyon fonksiyonu olarak relu, hata fonksiyonu olarak da cross entropi işlemleri tercih edilmiştir. Veri setindeki imgeler 64x64'e göre yeniden boyutlandırılmıştır. Programın her çalıştırılmasında rasgele altı kategori imge alınmaktadır ve 100 iterasyon çalıştırılmaktadır. Optimizasyon algoritmaları ile farklı sonuçlar elde edilmiş ve bu sonuçlar analiz edilmiştir. Sınıflamadaki başarı oranları sgd:%64.5, momentum:%85.56, adam:%92.31, adagrad:%71.25, rmsprop:%40.26 ve adadelta:%86.88 olarak gözlemlenmiştir.

**Anahtar Kelimeler:** Optimizasyon Algoritmaları, Derin Öğrenme, Yapay Sinir Ağları, Caltech 101

## APPLICATIONS AND COMPARISON OF OPTIMIZATION ALGORITHMS USED IN DEEP LEARNING

**Abstract:** Deep learning (multi layer artificial neural network) is a sub-branch of machine learning that enables learning from large data sets with its layered architecture. Increased processor speeds and storage, inexpensive computer hardware and machine learning gains are factors that boost deep learning development. Gradient descent based optimization algorithms(Stochastic gradient descent (sgd), momentum, adam, adagrad, rmsprop and adadelta) used to minimize error in deep learning are also part of this success.

In this study, deep learning and optimization algorithms were applied to the caltech 101 data set, which is an international data set. Relu was used as activation function and cross entropy was preferred as loss function. Images are resized to 64x64. Each time the program is run, a random six category image is taken and 100 iterations are executed. Different results were obtained with optimization algorithms and these results were analyzed. The success rates in classification were observed as sgd: 64.5%, momentum: 85.56%, man: 92.31%, adagrad: 71.25%, rmsprop: 40.26% and adadelta: 86.88%.

**Keywords:** Optimization Algorithms, Deep Learning, Artificial Neural Networks, Caltech 101

## 1. GİRİŞ

Makine öğrenimi, herhangi bir problem için var olan bilgiler ile bir çıkarım modeli ortaya koymaktır. Geliştirilen modeller, verilerden iteratif olarak öğrenmektedir. Gözetimli, Gözetimsiz, Yarı Gözetimli ve Pekiştirmeli öğrenme olarak dört türü mevcuttur. Nöral ağ, Karar Ağaçları, Kümeleme, regresyon ve derin öğrenme gibi çok çeşitli algoritmalara sahiptir.

Derin öğrenme, görüntü alma, doğal dil işleme ve nesne algılama gibi işlemlerde kullanılmaktadır. Derin öğrenme'nin amacı iyi bir genelleme elde etmektir. Global minimumu bulması şart değildir ama makul bir sürede çözüme en yakın sonucu bulması beklenmektedir (Kurt, 2019; Anadolu, 2019; Cebeci, 2020). Sonuca yaklaşmayı optimizasyon algoritmaları ile gerçekleştirmektedir.

Programın çalışmasında oluşabilecek hataları minimum seviyeye indirmek için optimizasyon yöntemleri kullanılmaktadır. Bu optimizasyon yöntemleri gradyan iniş (Gradient Descent) olarak da tanımlanmaktadır. Optimizasyon yöntemleri adım adım işletilen bir süreçtir. Bu süreçte işletilen, adım miktarına öğrenme katsayısı denilmektedir. Öğrenme katsayısı belirlenirken en uygun değer seçilmesi gerekmektedir. Öğrenme katsayısının küçük seçilmesi durumunda çözüm süreci uzamaktadır, büyük seçilmesi durumunda ise minimum noktanın gözden kaçırılmasına sebep olmaktadır. SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam algoritmaları, optimizasyon yöntemleri içinde en popüler olanlardır (Bosch ve ark., 2007; Frome ve ark., 2007; Seyyarer ve Aydın, 2017; Cebeci, 2020; Fortuner, 2019).

Vani ve ark., yaptıkları çalışmada konvolüsyonel sinir ağı (CNN) kullanmışlardır. Hint Pines veri seti üzerinde SGD, RMSProp, Adam, Adamax, Adagrad, Adadelta ve Nadam gibi optimizasyon yöntemlerini kullanarak CNN'deki hatayı minimize etmeye çalışmışlardır. Analiz sonucunda Adamax optimizasyon yöntemi ile %99,58 başarı elde etmişlerdir (Vani ve ark., 2019).

Rajakumaran ve ark., çok değişkenli lineer regresyon ve MSE hata fonksiyonunu kullanarak Denial of Service (DoS) saldırılarını tespit etmeye çalışmışlardır. Analiz sonucunda SGD optimizasyon yöntemi ile hata oranını %3,3'den %0,3'e düşürmeyi başarmışlardır (Rajakumaran ve ark., 2020).

Dogo ve ark., yaptıkları çalışmada CNN yapısına üç ayrı veri seti üzerinde uygulamışlardır. CNN'de oluşan hata oranını en aza indirmek için SGD, vSGD, SGDm, SGDm+n, RMSProp, Adam, AdaGrad, AdaDelta, Adamax ve Nadam optimizasyon yöntemlerini kullanmışlardır. Analiz sonucunda NADAM optimizasyon yöntemi ile en iyi sonucu, Adadelta optimizasyon yöntemi ile de en kötü sonucu elde etmişlerdir (Dogo ve ark., 2018).

Yazan ve ark., Momentum, AdaGrad, RMSProp, AdaDelta ve Adam optimizasyon yöntemleri kullanarak parametre güncelleme yapmışlardır. Bu optimizasyon yöntemlerini griewank fonksiyonunda test ederek Adadelta yöntemiyle en yüksek başarıya ulaşmışlardır (Yazan ve ark., 2017).

Önceki çalışmada, Uluslararası bir veri seti olan caltech 101 veri setine, tek ara katmanlı bir yapay sinir ağı uygulanmış, sonuçları analiz edilmiş ve yayına dönüştürülmüştür. Çalışmada imge ön işleme, bölütleme ve özellik seçimi gibi işlemler yapılmıştır. Öz nitelik seçimi için geometrik ve renksiz imgelere uygulanan 7 değişmez moment kullanılmıştır. Sınıflamadaki başarı oranı düşük olduğu gözlemlenmiştir.

Bu çalışmada ise aynı veri setine (caltech 101) derin öğrenme uygulanmıştır. Ayrıca optimizasyon algoritmalarının derin öğrenmedeki sonuçlarına odaklanılmaktadır. Hata oranını minimuma indirmek için gradient descent tabanlı 6 tane optimizasyon yönteminden (SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam) ve amaç fonksiyonu olarak da cross entropy hata fonksiyonundan yararlanılmıştır. Tek ara katmanlı yapay sinir ağına (önceki çalışma) optimizasyon algoritmaları kullanılmamıştır. İki çalışmanın sonuçları karşılaştırılmıştır.

Uygulamada kullanılan veri seti bölüm 2'de, optimizasyon algoritmaları bölüm 3'te, aktivasyon fonksiyonları bölüm 4'te, hata fonksiyonları bölüm 5'te, uygulama ile ilgili bilgiler bölüm 6'da ve son olarak bölüm 7'de sonuçlar kısmı ele alınmıştır.

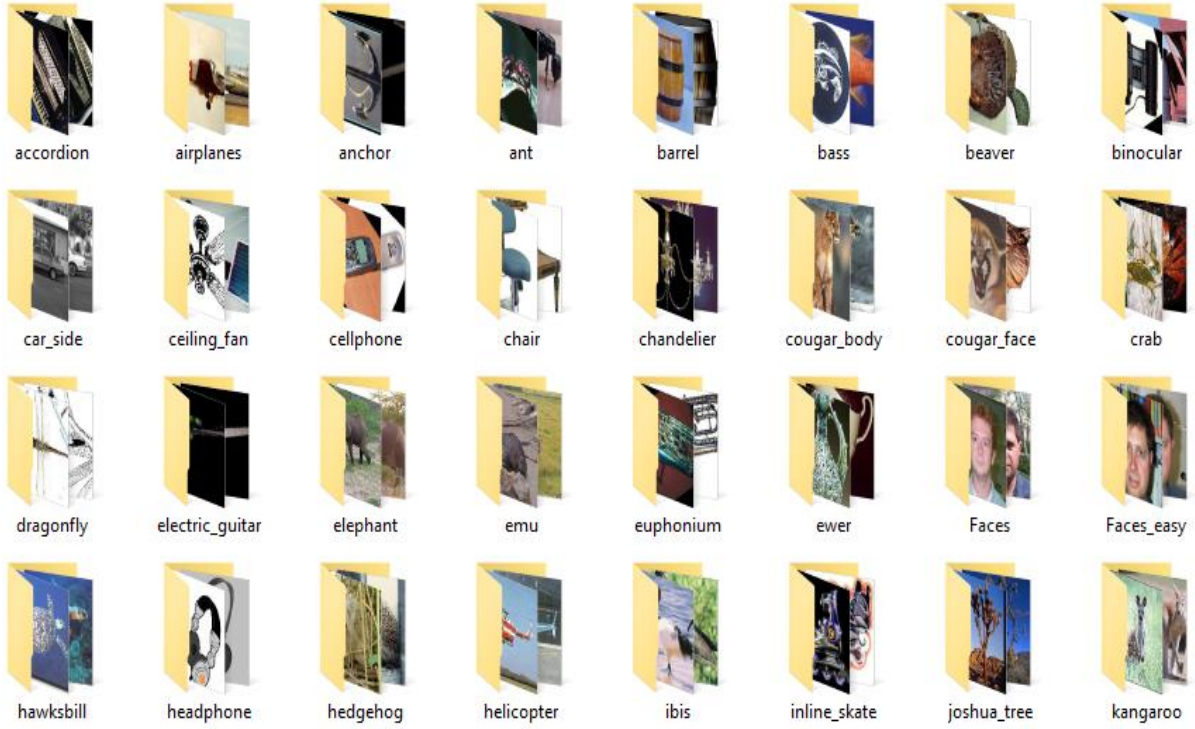
## 2. VERİ SETİ

Uluslararası bir veri seti olan Caltech 101, 101+1 ("Background Google" kategorisi sonradan eklenmiştir) kategoride imgelerden oluşmaktadır. Kategoriler, sayıları 40 ile 800 arasında değişen imgeler barındırmaktadır, fakat birçoğunda 50 imge bulunmaktadır ve her bir imge yaklaşık 300x200

ebatlarındadır. Toplamda 131 Mb'dir. 2003 yılında Fei-Fei Li, Marco Andreetto ve Marc 'Aurelio Ranzato tarafından oluşturulmuştur. Ayrıca 2006 yılında, boyutu 1.2 Gb olan Caltech 256 (256+1, "Clutter" kategorisi sonradan eklenmiştir) veri seti de literatürde yer almaktadır (Bosch ve ark., 2007; Frome ve ark., 2007; Seyyarer ve ark., 2017; Fei-Fei ve ark., 2006).

**Tablo 1.** Veri setlerinin karşılaştırılması

Veri Seti	Tarih	Kategori	Toplam İmge	En Düşük	Ortalama	En Fazla
Caltech 101	2003	101+1	9144	31	50	800
Caltech 256	2006	256+1	30608	80	100	827



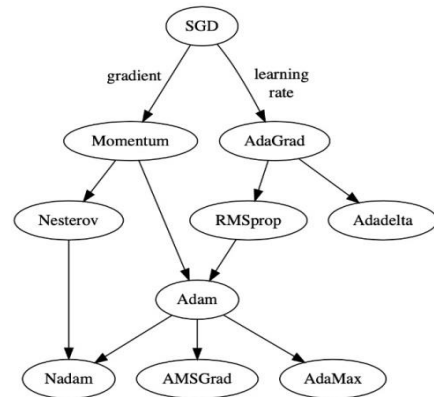
**Şekil 1.** Caltech101 imge setinden bazılarının kategorik görünümü

### 3. OPTİMİZASYON ALGORİTMALARI

Makine öğrenmesinde hata oranını en aza indirmek için sıklıkla kullanılan 6 tane optimizasyon yöntemi bulunmaktadır. Bunlar; SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam yöntemleridir. Tablo 2’de gradient descent yöntemlerinin özellikleri karşılaştırılarak sunulmuştur. Şekil 2’de gradient descent yöntemlerinin evrimsel haritası çizilmiştir.

**Tablo 2.** Gradient Descent yöntemlerinin karşılaştırılması (Alpaydın, 2018)

Yöntem	Tarih	Öğrenme Katsayısı	Gradient
SGD	1951	✓	✓
Momentum	1964		✓
Adam	2014	✓	✓
AdaGrad	2011	✓	
RMSprop	2012	✓	
Adadelta	2012	✓	



**Şekil 2.** Gradient Descent yöntemlerinin evrimsel haritası

### 3.1. SGD

Literatürdeki birçok çalışmada gradient descent ismi ile kullanılmaktadır. SGD, tüm gradient'ler yerine rastgele olmak üzere bir kısım gradient'le ağırlıkları güncellemektedir. Mevcut gradient'i ( $\partial L/\partial w_t$ ), öğrenme katsayısı ( $a$ ) ile çarparak mevcut ağırlığı ( $w_t$ ) güncellemektedir (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

$$w_{t+1} = w_t - a \frac{\partial L}{\partial w_t} \quad (1)$$

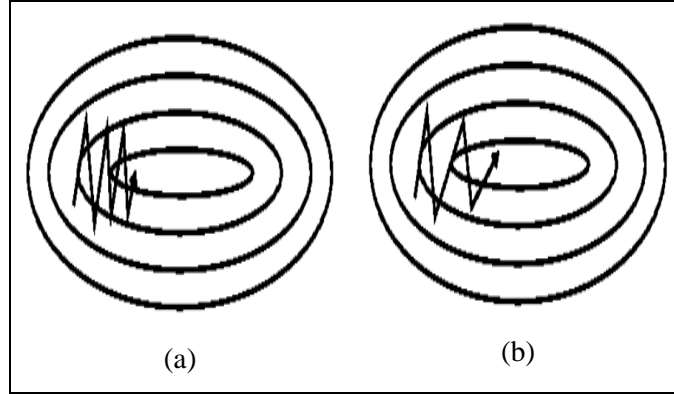
### 3.2. Momentum

SGD'de optimum nokta aranırken çok fazla salınım olmaktadır. Bu salınımları azaltmak ve dolayısıyla hedefe gitme hızını arttırmak için momentum yöntemi önerilmektedir. Bu yöntemde mevcut gradient'ler yerine momentumlu gradient kullanılmaktadır. Aslında momentum ismi biraz talihsizce olmuştur, bu yöntem, "kontrollü inme" olarak adlandırılabilir (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

$$w_{t+1} = w_t - aV_t \quad (2)$$

$$V_t = \beta V_{t-1} + (1 - \beta) \frac{\partial L}{\partial w_t} \quad (3)$$

Burada  $V_t$ 'nin başlangıç değeri 0'dır.  $\beta$ , 0 ile 1 arasında olup genelde kullanılan değeri ise 0.9'dur ve geçmiş gradientlerin işleme ne kadar katılacağını ayarlamak için kullanılmaktadır (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018). Şekil 3 a'da momentumlu SGD'nin çözüme giderken çok fazla salınım yaptığı görülmektedir. Buna karşılık şekil 3 b'de momentumlu SGD, daha az salınım yaparak çözüme ulaşmıştır.



Şekil 3. (a) momentumuz SGD, (b) momentumlu SGD (Akca, 2020)

### 3.3. Adagrad

SGD ve Momentum yöntemlerindeki sabit öğrenme katsayısı problemini ortadan kaldırmak için önerilmiştir. Her adımda farklı bir öğrenme katsayısı ile işlem yapmak için, eşitlik 2'deki öğrenme katsayısını geçmiş gradientlerin karelerinin kümülatif toplamının kareköküne bölmektedir (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (4)$$

$$S_t = S_{t-1} + \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (5)$$

Burada  $S$  başlangıçta 0,  $\epsilon$  ise işlemi sıfıra bölme hatasından kurtarmak için genelde çok küçük bir sayı ( $10^{-7}$ ) olarak alınmaktadır (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

### 3.4. RMSprop

Adagrad'da olduğu gibi sabit öğrenme katsayısı problemini çözmek için önerilmiştir. Aralarındaki fark ise, adagrad yöntemindeki gradientlerin karelerini almak yerine momentumlu gradientlerin karelerini almaktadır (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (6)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (7)$$

Burada  $S$  başlangıçta 0,  $\alpha=0.001$ ,  $\beta=0.9$  ve  $\epsilon$  ise  $10^{-6}$  olarak tercih edilmektedir (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

### 3.5. Adadelata

Adadelata ve RMSprop yaklaşımları aynı yıl içerisinde farklı kişiler tarafından önerilmiştir. Her iki yöntem de, sabit öğrenme problemini çözmeye odaklanan adagrad versiyonlarıdır. Adadelata yönteminde, adagrad ve RMSprop yöntemlerinden farklı olarak öğrenme katsayısı seçme zorunluluğu yoktur. Öğrenme katsayısı yerine, geçerli ağırlıklar ile güncellenen ağırlıklar arasındaki farkı ifade eden delta değerlerinin karelerinin momentumlu toplamları kullanılmaktadır (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1} - 1}}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (8)$$

$$D_t = \beta D_{t-1} + (1 - \beta) [\Delta w_t]^2 \quad (9)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (10)$$

$$\Delta w_t = w_t - w_{t-1} \quad (11)$$

Burada  $S$  ve  $D$  başlangıçta 0,  $\beta=0.95$  ve  $\epsilon$  ise  $10^{-6}$  olarak tercih edilmektedir (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

### 3.6. Adam

Rmsprop ve momentum yöntemlerinin avantajlı yönlerinin birleştirilmesi ile önerilen gradient descent algoritmasıdır. Momentum yöntemindeki  $V$  ile rmspropdaki  $S$ 'i kullanmaktadır (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{S}_t + \epsilon}} \cdot \hat{V}_t \quad (12)$$

$$\hat{V}_t = \frac{V_t}{1 - \beta_1^t} \quad (13)$$

$$\hat{S}_t = \frac{S_t}{1 - \beta_2^t} \quad (14)$$

$$V_t = \beta_1 V_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial w_t} \quad (15)$$

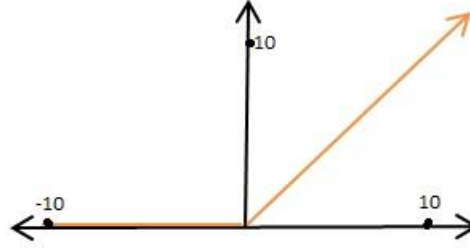
$$S_t = \beta_2 S_{t-1} + (1 - \beta_2) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (16)$$

Burada  $S$  ve  $V$  başlangıç olarak  $0$ ,  $\alpha=0.001$ ,  $\beta_1=0.9$ ,  $\beta_2=0.999$  ve  $\epsilon$  ise  $10^{-8}$  olarak tercih edilmektedir (Ruder, 2016; Akca, 2020; Yazan ve ark., 2017; Zeiler, 2012; Karim, 2018; Alpaydın, 2018).

#### 4. AKTİVASYON FONKSİYONLARI

Aktivasyon fonksiyonlarındaki amaç; ağırlık ve eşik (bias) değerlerini ayarlamaktır. Sigmoid, tanh, relu, leaky relu ve elu gibi birçok aktivasyon fonksiyonu bulunmaktadır. Günümüzün en popüler aktivasyon fonksiyonu relu fonksiyonudur. 2012 yılında ilk defa kullanılmaya başlanmıştır. Her yıl imajinal dataset (Alex.Net) yarışmalarında çok iyi sonuç vermektedir (Cebeci, 2019).

$F(x)=\max(0,x)$ ;  $x$  değeri negatifse sonuç sıfır, pozitif ise sonuç  $x$ 'tir (Cebeci, 2019).

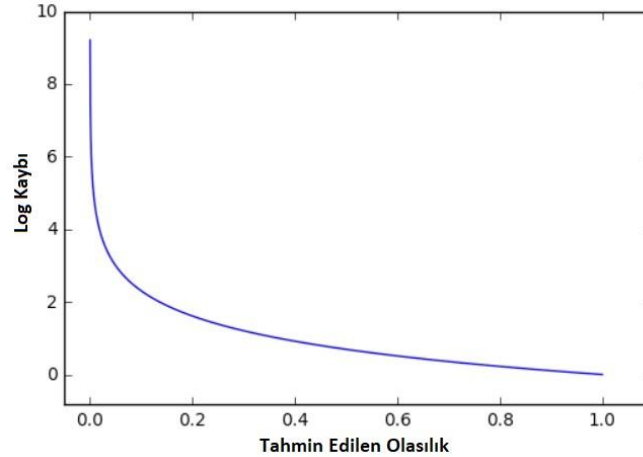


Şekil 4. Relu fonksiyonunun grafiği

#### 5. HATA (LOSS, ERROR, COST, OBJECTIVE) FONKSİYONLARI

Hata fonksiyonları; tahmin edilen değer gerçek değerden ne kadar uzak olduğunu hesaplamaktadır ve eğitim esnasında zamanla sıfıra yaklaşması beklenmektedir. Hata fonksiyonlarının Cross-Entropy, Hinge, Huber, Kullback-Leibler, MAE (L1) ve MSE (L2) gibi çeşitleri bulunmaktadır. Uygulamada çapraz entropi kullanılmıştır (Cebeci, 2019; Fortuner, 2019).

Çapraz entropi kaybı veya log kaybı, çıktısı 0 ile 1 arasında bir olasılık değeri olan bir sınıflandırma modelinin performansını ölçmektedir. Öngörülen olasılık, gerçek değerden uzaklaştıkça, çapraz entropi kaybı artmaktadır. İyi bir model, sıfır log kaybı ile sonuçlanmalıdır (Fortuner, 2019).



Şekil 5. Gerçek değer 1 iken log kaybı grafiği (Fortuner, 2019).

Çapraz entropi ve log kaybı, içeriği itibarı ile biraz farklıdır, ancak makine öğrenmede 0 ile 1 arasındaki hata oranlarını hesaplarken aynı şeyi çözmektedirler.

Çapraz entropi'de sınıf sayısı 2 iken formül;

$$(y \log(p) + (1 - y) \log(1 - p)) \quad (17)$$

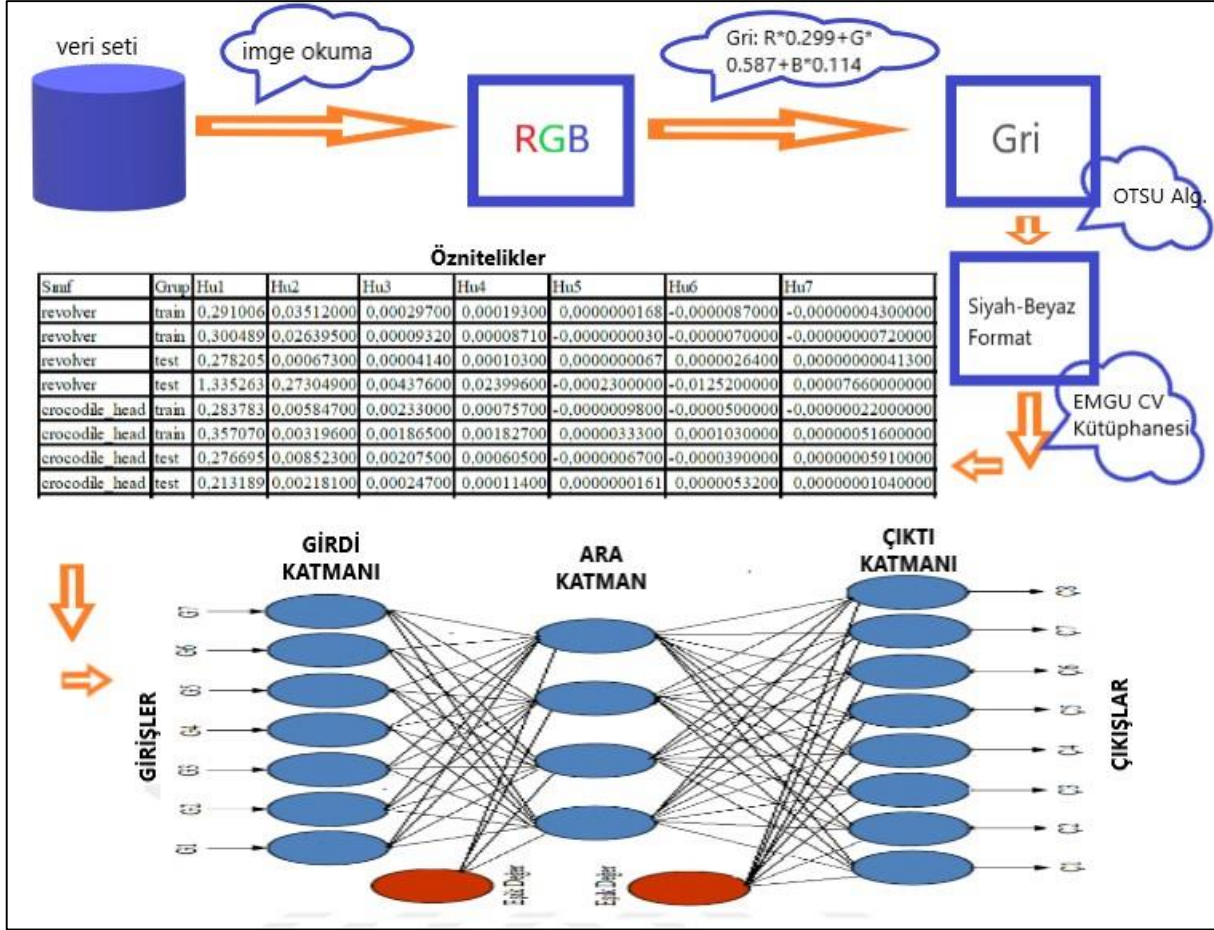
Çapraz entropi'de sınıf sayısı 2'den fazla ise formül;

$$\sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (18)$$

$M$ , sınıf sayısıdır,  $y$ ,  $c$  sınıfı etiketinin  $o$  gözlemi için doğru sınıflandırma olup olmadığını gösteren bir ikili göstergesidir (0 veya 1),  $p$ , modelin,  $o$  gözleminin  $c$  sınıfında olması ihtimalinin öngörülen olasılığıdır (Fortuner, 2019).

## 6. UYGULAMA

Önceki çalışmada yapay sinir ağı kullanılmıştır ve başarı oranı oldukça düşük olarak gözlemlenmiştir. Bunun sebebi; belirli geometrik şekli olan imgelerden öznelik çıkarmak için kullanılan 7 değişmez momentlerin kullanılmasıdır. Ayrıca yapay sinir ağının tek ara katmanlı olarak kullanılması ve optimizasyon algoritmalarının kullanılmaması da başarısızlığı getiren faktörlerden olmuştur. Önceki çalışmanın tüm aşamaları şekil 6'da gösterilmektedir.

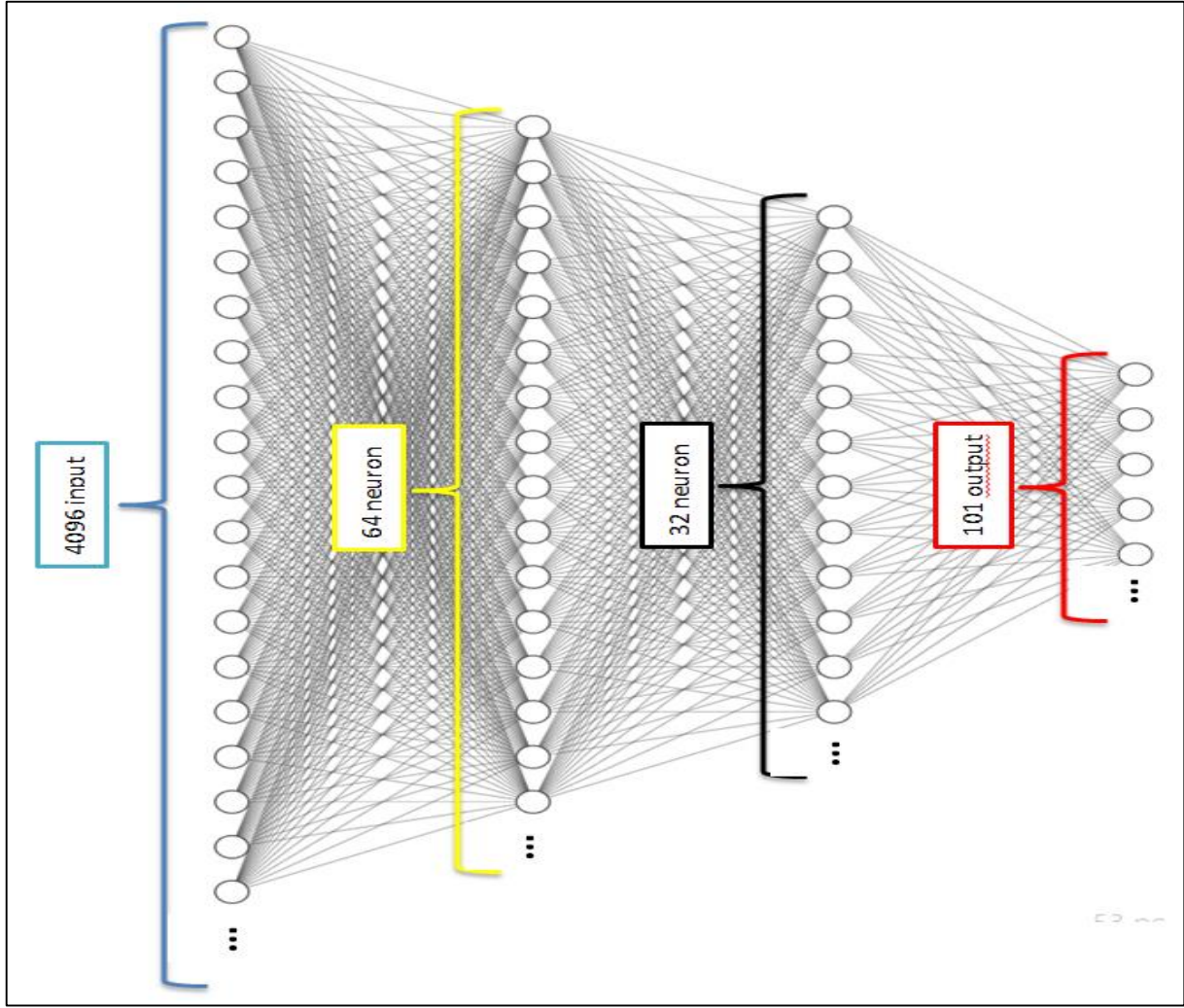


Şekil 6. Önceki çalışma aşamaları

Şimdiki çalışmada kullanılan sinir ağı, şekil 7'de gösterildiği gibi hem çok girdili hem de iki ara katmana sahiptir. Çok girdili olması, imge ile ilgili daha fazla özneliğin sınıflandırmaya etkisinin olması demektir. Öz niteliklerin fazla olması başarı oranını yükseltmektedir. Ara katmanların çok olması derin öğrenmeyi aktif hale getirmektedir. "Ara katmanlar kaç adet ve her katman kaç düğüme sahip olmalı?" sorusuna henüz net bir cevap literatürde bulunmamaktadır. Uygulamada ilk katman 64, ikinci ara katman ise 32 düğüme sahiptir.

Uygulama aşamaları aşağıdaki gibidir;

1. İmge okuma (rasgele 6 kategori)
2. Griye çevirme
3. 64x64'e göre yeniden boyutlandırmak
4. %70 eğitim, %30 test verisi olarak ayırmak (her kategori için)
5. Eğitim verisini ağı verip sinir ağını modellemek
  - a. Relu aktivasyon fonksiyonu
  - b. Cross Entropi (Loss fonksiyonu)
  - c. Optimizasyon algoritmaları (sgd, momentum, adam, adagrad, rmsprop ve adadelta)
6. Test verisini modele uygulamak
7. Sonuçları analiz etmek



Şekil 7. Şimdiki çalışmanın YSA görüntüsü

Şekil 7’de gösterilmekte olan derin öğrenme (çok katmanlı YSA) modelinin başarısı tablo 3’ de listelenmiştir.

**Tablo 3.** Yöntemlerin başarı oranları.

Yöntemler	Başarı Oranı(%)	
Tek Katmanlı YSA	25	
Çok Katmanlı YSA	SGD	64,5
	Momentum	85,56
	Adam	<b>92,31</b>
	Adagrad	71,25
	RMSProp	40,26
	Adadelata	86,88

Uygulamalar, 8 GB SSHD hafızaya, Intel i5-4210U işlemciye, 1 TB HDD sabit diske ve NVidia GeForce 840M’ye sahip bir bilgisayar üzerinde GPU tabanlı gerçekleştirilmiştir.

## 7. SONUÇLAR

Yapılan çalışmada derin öğrenme ve optimizasyon algoritmaları kullanılmıştır. Bu optimizasyon algoritmaları, uluslararası caltech 101 veri setine uygulanarak farklı sonuçlar elde edilmiştir. Literatürdeki en başarılı aktivasyon fonksiyonu olan relu ve hata fonksiyonu olarak da cross entropi fonksiyonları tercih edilmiştir. Çalışmada, imgelerin boyutları 64x64 olmak üzere değiştirilerek kullanılmıştır. İmge kategori boyutlarının yüksek olmasından dolayı, programın her çalıştırılmasında rasgele 6 kategori alınmaktadır ve 100 iterasyon çalıştırılmaktadır. Uygulama sonucunda optimizasyon algoritmalarının sınıflamadaki başarı oranları sgd:%64.5,



momentum:%85.56, adam:%92.31, adagrad:%71.25, rmsprop:%40.26 ve adadelta:%86.88 olarak gözlemlenmiştir.

Optimizasyon algoritmalarının kullanılmadığı ve tek ara katmanlı YSA'nın kullanıldığı bir önceki çalışmanın başarı oranı ise %25 olarak gözlemlenmiştir. Yapılan iki uygulamanın sonuçları tablo 3'te listelenmiştir.

Yapılan uygulamaların sonuçları dikkate alındığında, 7 değişmez moment özneliklerinin belli bir geometrik şekle sahip olan imgeler üzerinde kullanılması önerilmektedir. Tek katmanlı YSA'nın imgeler üzerinde başarılı olmadığı ve bu imgeler için derin öğrenme modellerinin kullanılması sonucuna varılmaktadır. Ayrıca büyük veri setlerinde hatayı minimuma indirmek için gradient descent tabanlı optimizasyon algoritmaları kullanılmalıdır.

Önümüzdeki çalışmalarda, optimizasyon algoritmalarının diğer makine öğrenme yöntemlerindeki başarı oranları incelenerek bu oranların artırılması hedeflenmektedir. Ayrıca hibrid bir optimizasyon algoritmasının çalışması düşünülmektedir.

## 8. KAYNAKLAR

- Akca, M.F., (2020). <https://medium.com/deep-learning-turkiye/gradient-descent-nedir-3ec6afcb9900>. Erişim Tarihi: 12.03.2020.
- Alpaydın, E., (2018). "Yapay Öğrenme". Boğaziçi Üniversitesi Yayınevi, İstanbul.
- Anadolu, B., (2019). Dijital Hikaye Anlatıcılığı Bağlamında Yapay Zekanın Sinemaya Etkisi: Sunspring ve It's No Game Filmlerinin Analizi. Erciyes İletişim Dergisi, (1), 39-56.
- Bosch, A., Zisserman, A., & Munoz, X., (2007). Image classification using random forests and ferns. In 2007 IEEE 11th international conference on computer vision (pp. 1-8). Ieee.
- Cebeci, H., (2019). "Yapay Zeka ve Derin Öğrenme A-Z™: Tensorflow". Erişim Tarihi: 17.11.2019.
- Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B. and Aigbavboa, C. O., (2018). "A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks." 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). IEEE.
- Fei-Fei, L., Fergus, R. ve Perona, P., (2006). One-Shot learning of object categories. IEEE Trans. Pattern Recognition and Machine Intelligence. In press. vol. 28, no. 4, pp. 594-611, April 2006, doi: 10.1109/TPAMI.79.
- Fortuner B., (2019). [https://github.com/bfortuner/ml-glossary/blob/master/docs/loss\\_functions.rst](https://github.com/bfortuner/ml-glossary/blob/master/docs/loss_functions.rst). Erişim Tarihi: (07.05.2019)
- Frome, A., Singer, Y., & Malik, J., (2007). Image retrieval and classification using local distance functions. In Advances in neural information processing systems (pp. 417-424).
- Karim R., (2018). <https://towardsdatascience.com/10-gradient-descent-optimisation-algorithms-86989510b5e9> . Erişim Tarihi: 05.05.2019.
- Kurt, İ., (2019). Makine öğrenmesi yöntemleri kullanılarak sesin müzikal öznelikleri ile parkinson hastalığının tespiti (Master's thesis, Trakya Üniversitesi Fen Bilimleri Enstitüsü).
- Rajakumaran, G., Venkataraman, N. & Mukkamala, R.R., (2020). "Denial of Service Attack Prediction Using Gradient Descent Algorithm." SN Computer Science 1.: 45.
- Ruder S., (2016). <https://ruder.io/optimizing-gradient-descent/>. Erişim Tarihi: 02.02.2020.
- Seyyarer, A., Aydın, T., (2017). Değişmez Momentler Kullanarak İçerik Tabanlı Görüntü Erişim Sistemi ve İmge Sınıflandırma Yöntemlerinin Karşılaştırılması. Anatolian Science - Bilgisayar Bilimleri Dergisi, 2 (1), 1-9. Retrieved from <http://dergipark.org.tr/bbd/issue/30846/333815>.
- Vani, S., ve Rao, T.V.M., (2019). "An Experimental Approach towards the Performance Assessment of Various Optimizers on Convolutional Neural Network." 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE.
- Yazan, E., ve Talu, M.F., (2017). "Comparison of the stochastic gradient descent based optimization techniques." Artificial Intelligence and Data Processing Symposium (IDAP), 2017 International. IEEE,
- Yılmaz, A., (2017). "Yapay Zeka", Kodlab Yayınevi, İstanbul.
- Zeiler, M.D., (2012). "ADADELTA: an adaptive learning rate method". arXiv preprint arXiv:1212.5701.