

**T.C.**

**YILDIZ TEKNİK ÜNİVERSİTESİ**

**FEN BİLİMLER ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**TEZLİ YÜKSEK LİSANS PROGRAMI**

**BLM 5109: KOLLEKTİF ÖĞRENME**

**OPTİMİZASYON ALGORİTMALARINI KARŞILAŞTIRMA**

**AHMED UĞUR**

**23501027**

**METİN USLU**

**235B7014**

**Öğretim Üyesi**

**Prof. Dr. Mehmet Fatih AMASYALI**

**İSTANBUL**

**Kasım, 2024**

**Optimizasyon Algoritmalarını Karşılaştırma**

**Özet**

Bu çalışmada, derin öğrenme modellerinin eğitimi sırasında kullanılan farklı optimizasyon algoritmalarının performanslarını karşılaştırmak amaçlanmıştır. Bir adet görüntü (CIFAR-10) ve bir adet metin (AG News) veri kümesi üzerinde, sıfırdan oluşturulan derin öğrenme modelleri kullanılarak Stokastik Gradient Descent (SGD), Momentum’lu SGD ve Adam optimizasyon algoritmaları değerlendirilmiştir. Tüm algoritmalar ve veri kümeleri için aynı mini-batch boyutu ve epoch sayısı kullanılmış, her bir veri kümesi için 5 farklı başlangıç noktası belirlenerek veri kümesi başına toplamda 15 farklı optimizasyon süreci yürütülmüştür.

Deneyler sonucunda, Adam algoritmasının en hızlı yakınsama ve en yüksek doğruluk değerlerine ulaştığı gözlemlenmiştir. Momentum’lu SGD, SGD'ye göre daha hızlı yakınsamış ve daha yüksek performans sergilemiştir. SGD algoritması ise daha yavaş bir yakınsama göstermiştir. Eğitim ve doğrulama kayıplarının yanı sıra, t-SNE ile görselleştirilen yörüngeler de algoritmaların farklı davranışlarını ortaya koymuştur. Elde edilen bulgular, optimizasyon algoritmalarının model performansı üzerindeki kritik rolünü vurgulamaktadır.

**1. Giriş**

Derin öğrenme modellerinin eğitimi, büyük ve karmaşık veri kümeleri üzerinde etkin bir şekilde çalışabilmeleri için uygun optimizasyon algoritmalarının kullanılmasını gerektirir. Optimizasyon algoritmaları, modelin öğrenme sürecinde ağırlıkların güncellenmesini sağlayarak kayıp fonksiyonunu minimize etmeye çalışır. Bu süreçte seçilen optimizasyon yöntemi, modelin yakınsama hızını, genel performansını ve eğitim süresini doğrudan etkiler.

Stokastik Gradient Descent (SGD), derin öğrenme alanında en temel ve yaygın olarak kullanılan optimizasyon algoritmalarından biridir. SGD, her bir mini-batch için kayıp fonksiyonunun gradyanını hesaplayarak ağırlıkları günceller. Basitliği ve düşük hesaplama maliyeti nedeniyle tercih edilir; ancak, büyük veri kümelerinde yakınsama hızı yavaş olabilir ve lokal minimumlara takılma olasılığı vardır.

Momentum’lu SGD, standart SGD'nin eksikliklerini gidermek amacıyla geliştirilmiştir. Bu yöntem, önceki gradyan güncellemelerini hesaba katarak optimizasyon sürecine "momentum" ekler. Böylece, optimizasyon yolu üzerindeki salınımlar azaltılır ve model daha hızlı ve istikrarlı bir şekilde yakınsar. Momentum terimi, gradyanların güncellenmesinde ataleti temsil eder ve lokal minimumların aşılmasına yardımcı olur.

Adam (Adaptive Moment Estimation), adaptif öğrenme oranı belirleme mekanizmasına sahip gelişmiş bir optimizasyon algoritmasıdır. Adam, her bir parametre için ilk ve ikinci momentleri (ortalama ve varyans) takip ederek öğrenme oranını dinamik olarak ayarlar. Bu özellik, farklı parametrelerin farklı hızlarda öğrenmesini sağlar ve optimizasyon sürecini hızlandırır. Adam, özellikle derin ve karmaşık ağların eğitiminde popüler bir seçimdir.

Bu çalışmada, yukarıda bahsedilen üç optimizasyon algoritmasının performanslarını karşılaştırma ve derin öğrenme modellerinin eğitiminde hangi algoritmanın daha etkin olduğunu belirlemek amaçlanmıştır. CIFAR-10 ve AG News veri kümeleri üzerinde sıfırdan oluşturulan farklı sinir ağı modelleri kullanılarak deneyler gerçekleştirilmiştir. Tüm algoritmalar için ortak bir mini-batch boyutu ve epoch sayısı belirlenmiş, ayrıca 5 farklı başlangıç noktası seçilerek her bir algoritma bu noktalardan başlatılmıştır. Bu sayede, her bir veri kümesinde 15 olmak üzere toplamda 30 farklı optimizasyon süreci yürütülmüş ve algoritmaların performansları çeşitli metrikler ve görselleştirmeler aracılığıyla değerlendirilmiştir.

Çalışmanın temel amacı, optimizasyon algoritmalarının modelin öğrenme süreci üzerindeki etkilerini analiz etmek ve hangi algoritmanın belirli koşullar altında daha iyi performans gösterdiğini ortaya koymaktır. Elde edilen bulguların, derin öğrenme modellerinin eğitiminde optimizasyon algoritması seçimine yönelik rehberlik etmesi hedeflenmektedir.

**2. Veri Kümesi**

Bu çalışmada farklı optimizasyon algoritmalarının performanslarını değerlendirmek ve karşılaştırmak amacıyla görüntü ve metin içeren iki farklı veri kümesi kullanılmıştır. Bunlar, görüntü sınıflandırma için yaygın olarak kullanılan CIFAR-10 ve metin sınıflandırma için yaygın olarak kullanılan AG News veri kümeleridir.

**2.1. CIFAR-10 Görüntü Veri Kümesi**

CIFAR-10, renkli görüntülerden oluşan ve 10 farklı sınıfı içeren bir veri kümesidir. Veri kümesi, toplamda 60.000 adet 32x32 boyutunda renkli (RGB) görüntü içermektedir. Bu görüntüler eğitim ve test verisi olarak ikiye ayrılmıştır. Eğitim seti 50.000 adet test seti ise 10.000 adet görüntüden oluşmaktadır. Her bir sınıf için 6.000 adet görüntü bulunmakta olup sınıflar dengeli bir şekilde dağıtılmıştır. Veri kümesindeki sınıflar; uçak, otomobil, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon şeklindedir.

CIFAR-10 veri seti, derin öğrenme ve özellikle konvolüsyonel sinir ağları (CNN) alanında yaygın olarak kullanılmaktadır. Görüntülerin küçük boyutlu ve çeşitli olması, modellerin genel performansını ve genelleme yeteneklerini değerlendirmek için ideal bir platform sunmaktadır. Bu çalışmada, farklı optimizasyon algoritmalarının CNN modeli üzerindeki etkilerini incelemek için CIFAR-10 veri seti tercih edilmiştir. Tablo 1’de veri kümesinden çeşitli sınıflardaki örnekler yer almaktadır.

***Tablo 1*** *– CIFAR-10 Veri Kümesi Örnekleri*



**2.2. AG News Metin Veri Kümesi**

AG News, metin sınıflandırma ve doğal dil işleme (NLP) alanlarında kullanılan bir haber metni veri setidir. Veri seti, çeşitli haber kaynaklarından toplanmış ve dört ana kategoriye ayrılmış toplam 127.600 metin örneği içermektedir. Bu örnekler eğitim ve test verisi olarak ikiye ayrılmıştır. Eğitim seti 120.000 adet, test seti ise 7.600 adet metinden oluşmaktadır. Her bir kategori için 30.000 adet metin bulunmakta olup veri kümesindeki kategoriler; dünya haberleri, spor, iş, bilim ve teknoloji şeklindedir.

Her kategori için metinler, ilgili alandaki haber başlıkları ve kısa özetlerden oluşmaktadır. AG News veri seti, metin sınıflandırma modellerinin performansını değerlendirmek, doğal dil işleme tekniklerini test etmek ve farklı algoritmaların metin verisi üzerindeki etkisini incelemek için sıklıkla kullanılmaktadır.

Bu çalışmada, optimizasyon algoritmalarının performansını görüntü ve metin verisi üzerinde değerlendirmek amacıyla CIFAR-10 ve AG News veri kümeleri kullanılmıştır. CIFAR-10 ve AG News’in çeşitliliği ve sınıflardaki örneklerin dengesi, derin öğrenme modellerinin eğitiminde farklı optimizasyon tekniklerinin etkisini incelemek için uygun veri kümeleridir. Modelin optimizasyon sürecinde elde edilen sonuçlar, algoritmaların yakınsama hızlarını, doğruluklarını ve genel performanslarını karşılaştırmak için kullanılmıştır.

**3. Deneysel Analiz**

Bu çalışmada farklı optimizasyon algoritmalarının performanslarını karşılaştırmak amacıyla gerçekleştirilen deneysel çalışmanın detayları sunulacaktır. Çalışma kapsamında kullanılan veri kümesi, model mimarisi, veri ön işleme adımları, hiper parametre ayarları ve optimizasyon süreçleri ayrıntılı bir şekilde açıklanacaktır.

**3.1. CIFAR-10**

CIFAR-10 veri seti, renkli görüntülerden oluştuğu ve piksel değerleri 0 ile 255 arasında olduğu için, modelin daha hızlı ve etkili öğrenebilmesi amacıyla veri üzerinde bazı ön işleme adımları uygulanmıştır. Tüm görüntülerin piksel değerleri 255’e bölünerek [0, 1] aralığına normalize edilmiştir. Bu işlem, ağın daha stabil bir şekilde öğrenmesini ve gradyanların daha sağlıklı hesaplanmasını sağlar. Sınıf etiketleri kategorik veriler olarak tutulduğundan, modelin çıktı katmanında kullanılmak üzere one-hot encoding işlemi uygulanmıştır. Bu sayede, her bir etiket, sınıf sayısı boyutunda bir vektöre dönüştürülmüştür.

Her bir veri kümesinin kendi içinde tutarlılığı sağlamak adına tüm optimizasyon algoritmaları aynı model mimarisi üzerinde değerlendirilmiştir. Şekil 1’de CIFAR-10 veri kümesi için Konvolüsyonel Sinir Ağı (CNN) modeli bileşenleri yer almaktadır. Buna göre giriş katmanında 32x32 piksel boyutunda ve 3 renk kanalına sahip RGB görüntüler yer almakta olup modelin bileşenlerine ait detaylar Tablo 3’de gösterilmiştir.

***Şekil 1*** *– Görüntü Veri Kümesi için CNN Modeli*diyagram, ekran görüntüsü, metin, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Tablo 3*** *– Görüntü Veri Kümesi için CNN Model Detayları*

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Param** |
| conv2d (Conv2D) | (None, 30, 30, 32) | 896 |
| max\_pooling2d (MaxPooling2D) | (None, 15, 15, 32) | 0 |
| conv2d (Conv2D) | (None, 13, 13, 64) | 18,496 |
| max\_pooling2d (MaxPooling2D) | (None, 6, 6, 64) | 0 |
| flatten (Flatten) | (None, 2304) | 0 |
| dense (Dense) | (None, 64) | 147,520 |
| dense (Dense) | (None, 10) | 650 |

Her bir veri kümesinde model, eğitime başlamadan önce 5 farklı başlangıç noktası belirlenmiştir. Bu başlangıç noktaları, modelin ağırlıklarının rastgele olarak farklı değerlerle başlatılmasıyla elde edilmiştir. Her bir başlangıç noktası için ağırlıklar kaydedilmiş ve daha sonra her algoritmanın kaydedilen başlangıç ağırlıkları kullanması sağlanmıştır. Bu yaklaşım her algoritmanın aynı başlangıç noktasından başlayarak performanslarının karşılaştırılmasına olanak tanır.

Deneylerde karşılaştırılan optimizasyon algoritmalarında Stokastik Gradient Descent (SGD) için learning\_rate=0.01, Momentum’lu SGD için learning\_rate=0.01, Momentum=0.9 ve Adam için learning\_rate= 0.001 olarak kullanılmıştır. Bu hiper parametreler literatürde yaygın olarak kullanılan ve her bir algoritma için optimal performans sağlayan değerler göz önünde bulundurularak seçilmiştir. Tüm algoritmalar ve veri kümeleri için ortak mini-batch büyüklüğü olarak **32???** ve epoch sayısı **10???** olarak belirlenmiştir. Eğitim sırasında her bir epoch sonunda modelin eğitim ve doğrulama (test) verisi üzerindeki kayıp, doğruluk, zaman ve ağırlık bilgileri kaydedilmiştir.

Her bir optimizasyon algoritması ve başlangıç noktası için modelin ağırlıklarının zaman içindeki değişimi yani yörüngeleri kaydedilmiştir. Bu yörüngeler, her epoch sonunda ağırlıkların vektörlere dönüştürülerek kaydedilmesiyle elde edilmiştir. Daha sonra, tüm yörüngeler birleştirilerek yüksek boyutlu ağırlık uzayından iki boyutlu bir uzaya indirgenmiş ve görselleştirilmiştir. Yörüngeleri görselleştirmek amacıyla t-SNE (t-distributed Stochastic Neighbor Embedding) kullanılmıştır. T-SNE ile Elde edilen iki boyutlu veriler kullanılarak her bir algoritmanın yörüngeleri çizilmiş ve algoritmalara göre renklendirilmiştir. Bu sayede, farklı optimizasyon algoritmalarının ağırlık uzayında nasıl farklı yollar izlediği görselleştirilmiştir. Şekil 2’de görüntü veri kümesi üzerinde işletilen optimizasyon süreçleri sonrası farklı algoritmaların yörüngeleri görselleştirilmiştir.

***Şekil 2*** *– Görüntü Veri Kümesi için Yörüngeler*

diyagram, ekran görüntüsü, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3.2. AG News**

AG News veri seti içerisinde 4 farklı kategoride haber başlıkları(title), haber açıklamaları (description) bulunan ve farklı uzunluklara sahip bir veri kümesi olduğu için modelin daha hızlı ve başarılı olabilmesi amacıyla veri üzerinde bazı ön işleme adımları uygulanmıştır. İlk olarak küçük harfe çevirme, html ifadelerinin kaldırılması, ekstra boşlukların kaldırılması, kısaltmaların düzeltilmesi, özel karakterlerin kaldırılması, dolgu sözcüklerinin (stopwords) kaldırılması, kelimelerin köklerine(stemming) yada anlamlarına (lemmatization) indirilmesi gibi ön işleme adımları uygunlanmıştır. Title ve Description başlıkları da birleştirilerek yeni bir kolon “Text” oluşturulmuştur. Son olarak Text özelliği üzerinde Tokenization yapılarak kelimeler vektörler haline getirilerek sayısallaştırılmış ardından padding uygulanarak metinlerin uzunlukları aynı boyuta getirilmesi sağlanmıştır.Sınıf etiketleri kategorik veriler olarak tutulduğundan, modelin çıktı katmanında kullanılmak üzere one-hot encoding işlemi uygulanmıştır. Bu sayede, her bir etiket, sınıf sayısı boyutunda bir vektöre dönüştürülmüştür.

Her bir veri kümesinin kendi içinde tutarlılığı sağlamak adına tüm optimizasyon algoritmaları aynı model mimarisi üzerinde değerlendirilmiştir. AG News veri kümesi için ilk katmanda Embedding sahip bir Derin Öğrenme modeli bileşenleri yer almaktadır. Buna göre giriş katmanında 100 birim karakter uzunluğunda ve 128 vektör uzunluğunda olup modelin bileşenlerine ait detaylar Tablo 4’de gösterilmiştir.

***Tablo 4*** *– Metin Veri Kümesi için Derin Öğrenme Model Detayları*

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Param** |
| embedding\_4 (Embedding) | (None, 100, 128) | 1280000 |
| flatten\_4 (Flatten) | (None, 12800) | 0 |
| dense\_8 (Dense) | (None, 128) | 1638528 |
| dropout\_4 (Dropout) | (None, 128) | 0 |
| dense\_9 (Dense) | (None, 4) | 516 |

**4. Sonuç**

Çalışmada, derin öğrenme modellerinin eğitiminde kullanılan üç farklı optimizasyon algoritmasının performansları üzerine yapılan karşılaştırmada aynı model mimarisi, mini batch boyutu ve epoch sayısı kullanılarak 5 farklı başlangıç noktasından başlamak üzere toplam 30 optimizasyon süreci yürütülmüştür. Elde edilen sonuçlar optimizasyon algoritmalarının modelin öğrenme süreci ve performansı üzerindeki etkileri açıkça göstermektedir.

Stokastik Gradient Descent (SGD) basit ve temel bir algoritma olarak daha yavaş yakınsaması ve düşük performans göstermesi beklenmekteydi. Sonuç olarak diğer algoritmalara göre daha düşük doğruluk elde ederek (yaklaşık %60) daha yavaş yakınsamıştır. Bu, büyük veri kümelerinde ve karmaşık modellerde performans sınırlamalarına yol açabilir.

Momentum’lu SGD, SGD’ye göre daha hızlı yakınsama yapabilecek bir momentum ile daha iyi bir performans beklenmekteydi. Sonuç olarak SGD’ye kıyaslayacak olursak daha hızlı yakınsamış ve daha yüksek doğruluk değerine (yaklaşık %70) ulaşmıştır. Momentumun eklenmesi, optimizasyon sürecini hızlandırmıştır.

Adam algoritmasının diğer iki algoritmaya göre daha hızlı yakınsama yapacağı beklenmekteydi. En yüksek eğitim ve doğrulama sonuçlarına ulaşarak (yaklaşık %80) diğer iki algoritmaya göre daha hızlı yakınsama gerçekleştirdi. Adam algoritması ile modelin ağırlıkları daha etkin bir şekilde güncellenerek hızlı ve kararlı bir öğrenme süreci gerçekleşmiştir. Şekil 3 ve 4’te farklı algoritmaların aynı eğitim ve test verisi üzerindeki doğruluk ve kayıp grafikleri görülmektedir.

***Şekil 3*** *– Görüntü Veri Kümesi için Farklı Algoritmaların Doğruluk Grafikleri*

çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Şekil 4*** *– Görüntü Veri Kümesi için Farklı Algoritmaların Kayıp Grafikleri*

öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi, diyagram, eğim, bayır içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yörüngeleri iki boyutlu uzayda incelendiğinde SGD algoritmasında daha dağınık ve zikzak şeklinde bir yol izleyerek yavaş yakınsama yaptığı, Momentum’lu SGD’nin standart SGD’ye göre daha düzgün ve hızlı bir şekilde ilerlediği, Adam algoritmasının ise başlangıçta çok daha yüksek adımlarla ilerleyerek hızlı yakınsama yaptığı gözlemlenmiştir.

Optimizasyon algoritmaları, derin öğrenme modellerinin eğitiminde kritik bir rol oynamaktadır. Adam algoritması, geçmiş gradyan bilgisinden yararlanarak daha hızlı yakınsaması sayesinde en iyi performansı göstermiştir. SGD üzerinde yapılan iyileştirmeler ile momentumun eklenmesi, standart SGD’nin performansının önemli ölçüde artmasını sağlamıştır. Modelin daha hızlı yakınsaması ve daha yüksek doğruluk değerlerine ulaşmasında optimizasyon algoritmalarının hiper parametrelerinin doğru ayarlanması önemli rol almaktadır. Optimizasyon algoritmalarının performansını daha da iyileştirmek için hiper parametrelerin kapsamlı bir şekilde optimize edilmesi ve en uygun hale getirilmesi gerekmektedir. Bu hiper parametreler arasında; öğrenme oranı, momentum, epoch sayısı, batch boyutu vb. yer almaktadır.

Öğrenme oranının çok yüksek seçilmesi sürecin hızlı bir şekilde hedefe ulaşmasını sağlayabilir ancak aynı zamanda kayıp fonksiyonunun minimum olduğu noktanın aşılarak sapmalara neden olabilir. Çok düşük bir öğrenme oranı ise daha hassas bir yakınsama sağlayabilir ancak eğitim süreci önemli ölçüde yavaşlayabilir. Momentum katsayısının yüksek belirlenmesi hedeflenen minimumun aşılmasına, düşük belirlenmesi ise momentumun avantajlarının kaybedilmesine neden olur. Bu nedenle modelin performansına bağlı olarak bu hiper parametreler aşamalı denemeler sonucunda kademeli olarak ayarlanmalıdır. Bunun yanı sıra Grid Search, Random Search, Hyperband gibi otomatik hiper parametre optimizasyonları kullanılarak daha verimli ve hızlı şekilde performansın en yüksek olduğu ayarlamalar yapılabilir.