

Derin Öğrenme Algoritmalarında Model Testleri: Derin Testler

Ferhat Şükrü Rende, Gültekin Bütün, Şamil Karahan

Bilişim Teknolojileri Enstitüsü, TÜBİTAK BİLGEM, Gebze, Kocaeli
{ferhat.rende, gultekin.butun, samil.karahan}@tubitak.gov.tr

Özet. Makine öğrenmesinin yeni bir alanı olan Derin Öğrenme, son dönemde geliştirilen yapay zekâ uygulamalarının başarısını oldukça yüksek düzeylere çıkarmıştır. ImageNet yarışmasında önerilen algoritmaların ve CVPR, ICLR gibi bilgisayarla görü alanına ait önemli konferanslarda yayınlanan çalışmaların birçoğu artık Derin Öğrenme temelli olmaktadır. Derin Öğrenme algoritmalarının geliştirilmesi yüksek miktarda etiketli veri ile eğitime dayanmaktadır. Algoritmalar mimari olarak karmaşık ve eğitim sırasında öğrenilmeye çalışılan parametre sayısı çoğunlukla milyon mertebesinde olmaktadır. Eğitimler iterasyonlar yapılarak gerçekleştirilmekte ve iterasyonlar sırasında fazlaca sayıda model elde edilmektedir. Elde edilen bu modellerin farklı farklı test kümeleri ile test edilmesi gerekmektedir. Bu çalışmada Derin Öğrenme algoritmasının eğitimi sonucunda ortaya çıkan modellerin birden fazla veri seti ile otomatik olarak testlerinin yapılması ve bu testler neticesinde kullanılacak modelin otomatik olarak belirlenmesine yönelik yapılan çalışmalar, elde edilen deneyimler anlatılacaktır.

Anahtar Kelimeler. Derin Öğrenme, Eğitimli Öğrenme, Test Otomasyonu

Abstract. Deep learning, which is a new area of machine learning, has brought the success of recently developed artificial intelligence applications to very high levels. Most of the algorithms that are proposed in ImageNet contest as well as studies published at computer vision conferences, such as CVPR and ICLR, are based on deep learning. Developing deep learning algorithms depends on high amounts of labeled data and training. Algorithms are architecturally complex and the number of parameters trying to be learned during training is mostly in millions range. Training is performed using iterations and during the iterations a high number of models obtained. These obtained models need to be tested with several different test sets. In this study, the automatic testing of the work models obtained from the results of deep learning algorithm training using multiple datasets, work done for automatic determination of the model to be used as results of these tests and experiences gained will be explained.

Anahtar Kelimeler. Deep Learning, Supervised Learning, Test Automation

1 Giriş

1.1 Makine Öğrenmesi

Makine öğrenmesi; çeşitli matematiksel yöntemler kullanarak mevcut verilerden bilgi edinen ve edindiği bu bilgi ile tahminler yapabilen yöntemler bütünü olarak ele alınabilir. Uygulama alanı çok geniş olmakla birlikte en yaygın olanları; optik karakter tanıma, yüz tanıma, obje tanıma, konuşulan dili anlama, tıbbi teşhis tahmini, hava tahmini, istenmeyen posta filtreleme gibi uygulamalardır.

Makine öğrenmesi algoritmaları eldeki ham veriyi (*Gözlem - Observation*) kullanmadan önce bu verileri temsil eden daha anlamlı veriler (*Öznitelik - Feature*) çıkarır. Her bir gözlem belirli bir kategoriye (*Etiket - Label*) ait olarak tanımlı olabilmektedir. Eğitim aşamasında; algoritmada kullanılan matematiksel metotlar mevcut öznitelikleri (*Eğitim verisi - Training Data*) işler ve ortaya bir model çıkarır. Artık bu model yardımı ile herhangi bir gözlem (*Test Verisi - Test Data*) hakkında tahminde bulunulabilir. Eğer eğitim aşamasında etiketler kullanıldı ise (*Gözetimli Eğitim - Supervised Training*) model, bir gözlemin kategorisini tahmin edebilmekte; eğer eğitim aşamasında etiketler kullanılmadı ise (*Gözetimsiz Eğitim - Unsupervised Training*) model, gözlemleri birbirlerine benzer olmalarına göre gruplayabilmektedir. Yüz tanıma problemi üzerinden örnek vermek gerekirse; elimizde bulunan yüz görüntüleri gözlemlerimiz, yüz görüntüsünün kime ait olduğu bilgisi etiketimiz, iki göz arası mesafenin piksel cinsinden uzunluğu, saç rengi veya ten rengi gibi bilgiler veya bir öznitelik çıkarıcı algoritma ile elde edilen bilgi de özniteliklerimiz olmaktadır.

1.2 Derin Öğrenme

Derin Öğrenme algoritmaları yapay sinir ağlarının (*YSA*) yapısal olarak daha karmaşık hali olarak düşünülebilir. *YSA* algoritmaları insandaki öğrenme işleyişinden hareketle geliştirilmiştir. Biyolojik sinir sisteminde bulunan nöronların birbirleri ile ilişki kurması gibi *YSA* sistemlerinde de nöron şeklinde tanımlanan yapılar birbirleri ile bağlantılı olacak şekilde modellenmişlerdir. Algoritmanın bu şekilde; öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahip olacağı düşünülmüştür [1]. Derin Öğrenme algoritmalarının makine öğrenmesindeki var olan algoritmalarından ayrılan yönü; çok yüksek miktarda veriye ve karmaşık yapısı ile de bu yüksek veriyi işleyebilecek çok yüksek hesaplama gücü olan donanımlara ihtiyaç duymasındadır. Son yıllarda özellikle görüntü işleme alanında etiketli veri sayısı milyon mertebelerini geçmiştir [2]. Ekran kartı temelli hesaplama gücü alanındaki büyük ilerlemelerin etkisiyle Derin Öğrenme algoritmaları çok popüler olmuştur [3]. Derin Öğrenme yöntemleri ile birçok alanda bilinen en iyi başarı düzeyleri (*State of the art*) çok daha yukarılara çıkmıştır [4]. Yüksek veri ve hesaplama gücü olanaklarına sahip büyük teknoloji firmaları (*Google, Facebook, Microsoft, Nvidia vb.*) tarafından da kullanılıp ürünlerine entegre edilmektedir. Aynı zamanda bu teknoloji firmaları kendi Derin Öğrenme yazılım kütüphanelerini geliştirici topluluklara da açarak bu alandaki hızlı ilerlemeye destek olmaktadır.

1.3 Test Otomasyonu

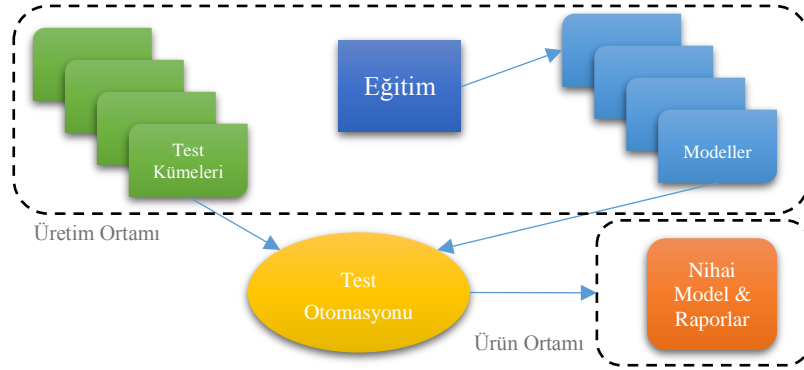
Test otomasyonunun amacı, hazır test otomasyon araçları veya ihtiyaca yönelik geliştirilen test otomasyon araçları kullanılarak testlerin otomatik koşturulması ve kontrol edilmesidir. Test otomasyonu sürekli teslimat ve sürekli test durumları için kritik öneme sahiptir [5].

Test otomasyon araçları aşağıdaki ihtiyaçları karşılamak için geliştirilip işletilmektedir:

- Genellikle tekrar ihtiva eden fonksiyonları/görevleri bir otomasyon süreci altına almak veya testlerin planlanması, tasarlanması, raporlanması ve gözlemlenmesi gibi manuel işlemlerin otomatize etmek,
- Manuel olarak işletildiğinde önemli ölçüde kaynak gereksinimi doğuran işlemleri otomasyona geçirmek,
- Testlerin güvenilirliğini artırmak,
- Manuel olarak yürütülemeyecek durumları çözebilmek [6].

2 Yapılan Çalışma

Derin Öğrenme algoritmalarının eğitiminin zaman açısından maliyetli olması dolayısı ile test otomasyon yöntemlerinin uygulanması kaçınılmazdır. Deneyim çalışmamızın ana konusu bu test otomasyonu aracının geliştirilmesidir. Geliştirdiğimiz test otomasyonu aracının tasarımı Şekil 1’de görülmektedir.



Şekil 1. Test Otomasyonu Aracı Tasarımı

Bu aracı geliştirmemizin amacı:

- Derin Öğrenme algoritmalarının eğitimi sırasında ortaya çıkan çok sayıda modelin başarı hesaplamalarını yapmak,
- Farklı test kümelerinde başarı kriterlerini hesaplamak ve koşulları sağlayan birçok model arasından -belirlenmiş bir mantığa bağlı şekilde- otomatik olarak model seçimlerini yapmak,

- Eğitimlerin uzun sürmesi dolayısı ile donanım cihazlarının kullanımını maksimize etmek, dolayısı ile algoritma eğitimi süresi boyunca ortaya çıkmış modellerin testlerini yapmak (ayrı bir zaman dilimi ayırmadan),
- Eğitim sırasında algoritmanın test kümelerinde yeterli başarıyı yakalayamayacağı- nın önceden tespitini yapıp zaman kaybetmeden tekrar eğitim aşamasına dönmek.

Geliştirdiğimiz test otomasyon aracı sayesinde, eğitimlerimiz devam ederken o ana kadar ortaya çıkan modellerin testleri otomatik olarak yapılıp raporlanırken, başarı eğrileri anlık olarak çizdirilmektedir. Tüm modellerin tüm test kümeleri ile testleri biter bitmez nihai model seçimi otomatik olarak yapılıp, gerekli olan öğeler üretim ortamından (*Development Environment*) ürün ortamına (*Production Environment*) otomatik olarak aktarılmaktadır. Böylece yoğun eğitim içeren algoritma geliştirme süreci, test otomasyonu yaklaşımı ile yoğun ve derinlemesine testler yapılarak yönetilebilmektedir.

Eğitim Aşaması: Çalışmamızın konusu iki boyutlu görüntülerin Derin Öğrenme algoritması ile sınıflandırılmasıdır. Son yıllarda büyük başarılar elde eden [7] [8] [9], görüntüler ile çalışmak üzere özelleştirilmiş Derin Öğrenme algoritmalarından [10] “Convolutional Neural Networks” (CNN) yöntemi kullanılmıştır. Eğitim verimiz yaklaşık yarım milyon görüntüden oluşmaktadır, modelimizde yaklaşık 5 milyon parametre hesaplanmakta ve dolayısı ile eğitim süremiz GPU kartlar kullanmamıza rağmen, günler-haftalar düzeyinde olmaktadır. Bir eğitim sonucunda yaklaşık 300 farklı model ortaya çıkmaktadır. Ortaya çıkan modeller test sistemine gönderilmektedir.

Test Aşaması: Her eğitim sonrası oluşan 300 farklı model, eğitim verisiyle örtüşmeyen aynı probleme ait en az 3 farklı test kümesi ile test edilmekte ve modellerin her bir test seti için belirlenen başarı eşiklerini geçip geçmediği test aracının grafik ara yüzünden ilgili algoritmanın sorumlusu tarafından kontrol edilebilmektedir. 3 farklı test seti için başarı eşiklerini geçebilen bir model elde edilemediği/edilemeyeceği anlaşıldığı takdirde ilgili algoritma için testler durdurulup; tekrar eğitim aşamasına dönülerek hiperparametre değişimi, farklı bir ön işleme uygulama, model yapısında değişiklik gibi çalışmalar yapılır. 3 farklı test seti için başarı eşiklerini geçebilen modeller mevcut ise - ki bu durumda algoritmanın düzgün bir şekilde yakınsadığı düşünülmelidir- mevcut modeller arasından en uygun olanının seçimi için model seçimi aşamasına geçilir.

Model Seçimi Aşaması: Bu aşamada elimizde 3 farklı test verisinde de başarı kriterlerini sağlayan en az bir model bulunmaktadır. Bir model olduğu durumda herhangi bir seçime gerek duymaksızın ilgili model nihai model olarak belirlenir ve ürün ortamına tüm öğeleri ile birlikte aktarılır. Eğer birden fazla model başarı kriterlerini sağlıyorsa hangi modelin seçileceğine karar verilmesi gerekir. Örnek vermek gerekirse, herhangi bir model 3. test kümesinde en yüksek başarıyı sağlarken, 2. veya 1. test kümelerinde en yüksek başarıyı veren model aynı model olmayabilmektedir. Bu durum Tablo 1’de sayısal olarak ele alınmıştır. Buna göre; eğitim neticesinde K adet modelin ortaya çıktığı düşünülecek olursa, ilk test kümesine göre karar verilirse 7. modeli, ikinci test kümesine göre karar verilecek olursa 4. modeli, üçüncü test kümesine göre karar verilirse

2. modelin seçilmesi gerekir. Bu durumda hangi modelin seçileceğine aşağıdaki yöntemler ile karar verilebilir:

1. Müşteri verisine en yakın olduğu düşünülen test kümesine göre karar verilir,
2. Her bir veri seti için tüm modellerin başarıdan verilen ödün miktarınca maliyetleri çıkarılır ve en az maliyetli model seçilir.
3. Başarıdan verilen ödün miktarına göre maliyet hesaplanırken test kümelerine belirli ağırlıklar verilerek ağırlıklandırılmış bir maliyet çıkarılır. Böylece ilk yöntem ile ikinci yöntemin avantajları birleştirilmeye çalışılır.

Maliyet hesabı yapmamızdaki amaç en genellenebilir modelin belirlenebilmesidir.

Tablo 1. Modellerin Test Kümelerindeki Başarıları

Model No	Test Kümesi 1 Başarısı	Test Kümesi 2 Başarısı	Test Kümesi 3 Başarısı	Test Kümesi N Başarısı
1	%65	%75	%96	...
2	%70	%80	%99	...
3	%69	%81	%98	...
4	%71	%92	%97	...
5	%80	%91	%95	...
6	%94	%89	%98	...
7	%95	%90	%97	...
K

Maliyet Değeri: Test kümesi i için, j . modelin yüzdesel başarısı TK_{ij} olsun, $j=1, \dots, K$ olmak üzere, TK_i^* i . test kümesinde elde edilen en yüksek başarıyı gösterirken, j . modele ait maliyet değeri Denklem 1'deki gibi tasarlanmıştır.

$$Maliyet(j) = \sum_{i=1}^N |TK_{ij} - TK_i^*|. \quad (1)$$

Testler neticesinde maliyeti en düşük model nihai model olarak seçilir. Tablo 1 için bu denklem uygulandığı takdirde 7. modelin maliyeti $|95-95| + |90-92| + |97-99| = 4$ ile en düşük maliyet olacağından 7. model nihai olarak kabul edilir ve ürüne entegre edilir. Dikkat edilecek olursa seçilecek nihai modelin herhangi bir test kümesi için en yüksek değeri vermeme durumları oluşabilmektedir. Örneğin 6. modelin test kümesi 2'deki başarısı %89 yerine %91 olsa idi; otomatik seçilecek model $|94-95| + |91-92| + |98-99| = 3$ olacağından, -herhangi bir test kümesinde en yüksek değeri vermediği halde- model 6 olacaktır.

Derin Öğrenme için oluşturduğumuz “Derin Test Sistemi”, algoritmanın eğitimi süreci ile başlamakta, üretilen her modeli tespit edip ilgili modeli anında test etmek şeklinde çalışmaktadır. Böylece eğitim süresinin bitiminde tüm testlerimizi de bitirmiş olabilmekteyiz. Dahası olumsuz bir eğitiminin sonuçları erkenden tespit edilebildiğinden, gerekli olduğunda eğitim durdurulmakta ve zaman kayıplarının önü alınmaktadır.

3 Sonuç

Elde ettiğimiz deneyimlerden hareketle yapılabilecek çıkarımlar aşağıdaki gibidir:

- Derin Öğrenme algoritmalarının geliştirilmesi için test aşamasının dikkatli ve detaylı şekilde planlanması, bir test otomasyon ortamının kurulması, algoritmanın daha kısa sürede geliştirilebilmesi için çok önemlidir.
- Her test kümesi için oluşan başarı eğrisi eğitim süresince ara ara incelenmeli, test başarıları beklenmedik şekilde ilerliyorsa algoritma eğitimi durdurulmalı ve gerekli değişiklikler yapılarak yeni model için eğitim aşamasına dönülmelidir.
- Derin Öğrenme algoritmalarının eğitiminde zor manuel iş yüklerinin çokluğu test otomasyonun yapılmasını gerektirmektedir.

Kaynakça

- [1] «Wikipedia,» [Çevrimiçi]. Available: https://tr.wikipedia.org/wiki/Yapay_sirir_a%C4%9Flar%C4%B1. [Erişildi: 16 Haziran 2016].
- [2] «image-net,» [Çevrimiçi]. Available: <http://www.image-net.org/about-stats>. [Erişildi: 16 Haziran 2016].
- [3] «Nvidia,» [Çevrimiçi]. Available: http://www.nvidia.com/content/events/geo-Int2015/LBrown_DL.pdf. [Erişildi: 16 Haziran 2016].
- [4] «devblogs.nvidia,» [Çevrimiçi]. Available: <https://devblogs.nvidia.com/parallelfoll/mocha-jl-deep-learning-julia/>. [Erişildi: 16 Haziran 2016].
- [5] «Wikipedia,» [Çevrimiçi]. Available: https://en.wikipedia.org/wiki/Test_automation. [Erişildi: 16 Haziran 2016].
- [6] «Sertifikalı Test Uzmanı Temel Seviye Ders Programı,» International Software Testing Qualifications Board, 2011.
- [7] A. Krizhevsky, I. Sutskever ve G. E. Hinton, «Imagenet classification with deep convolutional neural networks,» %1 içinde *Advances in neural information processing systems*, 2012, pp. 1097-1105.
- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke ve A. Rabinovich, «Going deeper with convolutions,» %1 içinde *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.
- [9] H. Kaiming, Z. Xiangyu, R. Shaoqing ve S. Jian, «Deep Residual Learning for Image Recognition,» *CoRR*, cilt abs/1512.03385, 2015.
- [10] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard ve L. D. Jackel, «Handwritten digit recognition with a back-propagation network,» %1 içinde *Advances in neural information processing systems*, 1990.