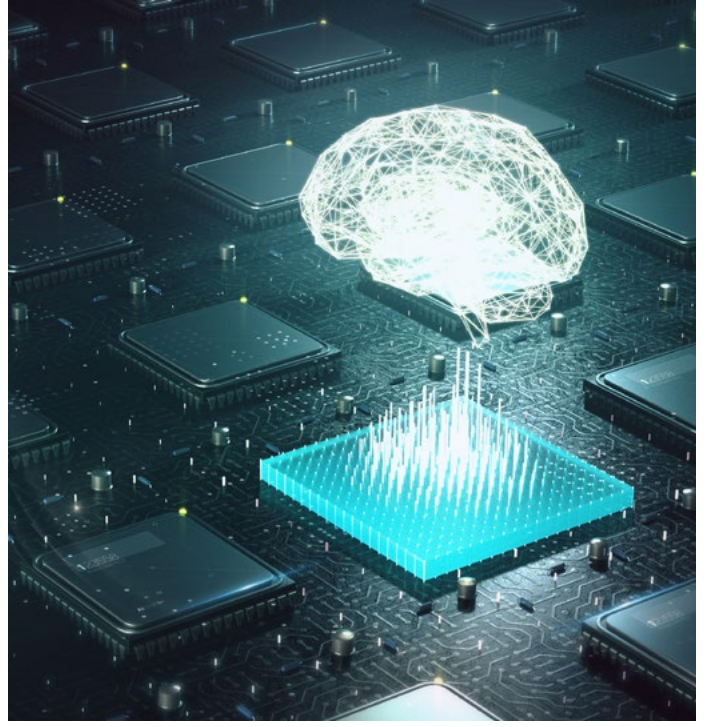


Derin Farklar: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme



Literatürde ve medyada sıkça işlenen ileri teknoloji ile ilgili farklı kavramların birbiri yerine kullanılması sıkça karşılaşılan bir olgudur. Ancak bir araştırma projesi ya da bir ürün geliştirme projesi için hazırlanacak algoritmanın; yapay zekâ, makine öğrenmesi ya da derin öğrenme algoritması olarak önceden net bir şekilde adlandırılması, ilgili projenin geleceğini olumlu yönde etkileyecektir. Nitekim yapay zekâ, makine öğrenmesi ve derin öğrenme sırasıyla dıştan içe doğru birbirini kapsayan ve aynı amaca yönelmiş teknolojilerdir. Ancak bunların çalışma şekilleri, ihtiyaç duydukları öğrenme veri miktarları ve son olarak çalışmaları için gereken kompütasyonel kapasite farklıdır. Şimdi bu kavramları açıklayalım:

Yapay Zekâ

Yapay zekâ, ister makine öğrenmesi kullansın ister kullanmasın herhangi bir tahmin veya karar işlemini gerçekleştiren teknolojilerin genel adıdır. Genel kanaatin aksine yapay zekâ makine öğrenmesi veya derin öğrenme algoritmaları olmaksızın da çalışan bir algoritma olabilir. Makine öğrenmesi algoritmaları ortaya çıkana kadar yapay zekâ çalışmaları “hard-coded” olarak nitelendirilen yani tüm mantıksal ve matematiksel işlemlerin yazılımcı tarafından bizzat kodlandığı bir yapıya dayanmaktaydı. Örneğin ilk satranç oyuncusu yapay zekâ algoritmaları tamamen böyledi. Yapay zekânın bu türü sembolik yapay zekâ olarak adlandırılmaktadır.

Makine Öğrenmesi Algoritmaları

Makine öğrenmesi algoritmaları, bilgisayarların açıkça programlanmadan bilişsel işlemler yapmasına olanak veren algoritmalar olarak nitelendirilir. Bu tanıma giren algoritmalar tarihsel olarak daha önce ortaya çıkan hard-coded yapay zekâ algoritmalarının yerini almıştır. Makine öğrenmesini hard-coded olarak kodlanmış sembolik yapay zekâ algoritmalarından ayıran özellik algoritmanın tamamen veriden öğrenmesidir. Bu farkı izah edebilmek için kabaca aşağıdaki örneği verebiliriz. Ekonomi ve sağlık haberlerini ayırtmak üzere bir algoritmamızın olduğunu varsayalım. Bu algoritma sembolik yapay zekâ ya da hard-coded olarak şöyle kodlanabilir:

```
haber_metni = "Merkez bankasından önemli açıklama"
```

```
#fonksiyonumuz
```

```
def haber_ekonomi_haberi_mi(haber_metni):
```

```
    #eğer haber içinde faiz, enflasyon gibi kavramlar varsa bu ekonomi haberidir
```

```
    if "faiz" in haber_metni:
```

```
        return "bu bir ekonomi haberidir"
```

```
    elif "obezite" in haber_metni:
```

```
return "bu bir sağlık haberidir"  
#....
```

Doğal olarak yukarıdaki şekilde kodlanmış bir yapay zekâ algoritması bir ekonomi haberinde olabilecek her spesifik kelimeyi içermek zorundadır. Ayrıca bu algoritma özellikle hem ekonomiyi hem sağlığı ilgilendiren bir haber başlığını ayırtmakta güçlük çekecektir. Bu algoritmanın kelime ikililerini özel olarak tanıyabilmesi için algoritmada açıkça kodlanması gerekmektedir. Örneğin “merkez bankası” kavramı şu şekilde ifade edilebilir:

```
...  
if "merkez" in haber_metni and "banka" in haber_metni  
...
```

Yukarıdaki örnekte görülen yapay zekâ algoritması tıpkı tanımında olduğu gibi “açıkça” kodlanmıştır. Bu nedenle “makine öğrenmesi algoritması” olarak kabul edilemez. Şimdi de aynı algoritmanın makine öğrenmesi algoritmasında nasıl bir şekilde ele alınacağını ifade edelim:

```
haber_ornekleri[0]="Merkez bankasından önemli açıklama."  
haber_ornekleri[1]="Obezite öldürüyor"  
haber_ornekleri[2]="BIST30 düşüşte"  
haber_ornekleri[3]="Yeni mide küçültme cerrahi yöntemi"  
...  
haber_etiketleri[0]="ekonomi"  
haber_etiketleri[1]="sağlık"  
haber_etiketleri[2]="ekonomi"  
haber_etiketleri[3]="sağlık"  
...  
#fonksiyon  
MAKINE_OGRENMESI_MODELI = makine_ogrenmesi_algoritmasi(haberler,etiketler)  
#bu modelle tahmin  
MAKINE_OGRENMESI_MODELI.predict("Bankalar arası faiz oranları hakkında...")  
#sonuç  
"ekonomi"
```

Yukarıda MAKINE_OGRENMESI_MODELI olarak tanımladığımız model NaiveBayes, Support Vector Machines, DecisionTree, RandomForest, xGBoost gibi herhangi bir makine öğrenmesi algoritması olabilir. Burada dikkat edilecek olursa algoritma haber_ornekleri ve haber_etiketleri üzerinden öğrenme yaparak *Faiz oranları neyi söylüyor* ifadesini ekonomi haberi olarak yorumlamaktadır. Peki bunu nasıl gerçekleştirmektedir? Arkasındaki mantık basittir. Örneğin Naive Bayes algoritması haber_ornekleri ve haber_etiketleri’ne bakarak *faiz* ifadesinin en çok *ekonomi* etiketli haberlerde olduğunu istatistiksel olarak fark etmektedir. Burada bu işlemi sembolik yapay zekâ birebir “if ’faiz’ in haber_metni” şekilde kodlama yolu ile yapmakta iken makine öğrenmesi algoritması verilerden öğrenmektedir.

Doğal olarak akla makine öğrenmesi algoritmalarının verilerden öğrenmesinin riskli tarafları gelebilir. Bu durum vakidir. “haber_ornekleri” eğitim verisinde haber_etiketleri kısmında *ekonomi* yerine *magazin* kodlanması durumunda algoritma gördüğü ekonomi haberlerini magazin haberi olarak kabul edebilir. Bu durumun somut örneği Twitter’deki metinlerden tweet atmayı öğrenen Microsoft’a ait Tay isimli bir botun sürekli olarak ırkçı

içerikli tweet atması ile sonuçlanmış ve sonunda Microsoft Tay'ı pasitfize etmiştir¹. Bu durumun nedeni ise algoritmanın Twitter'da sıkça karşılaştığı ırkçı tweet'lerden kaynaklanmıştır.

Derin Öğrenme

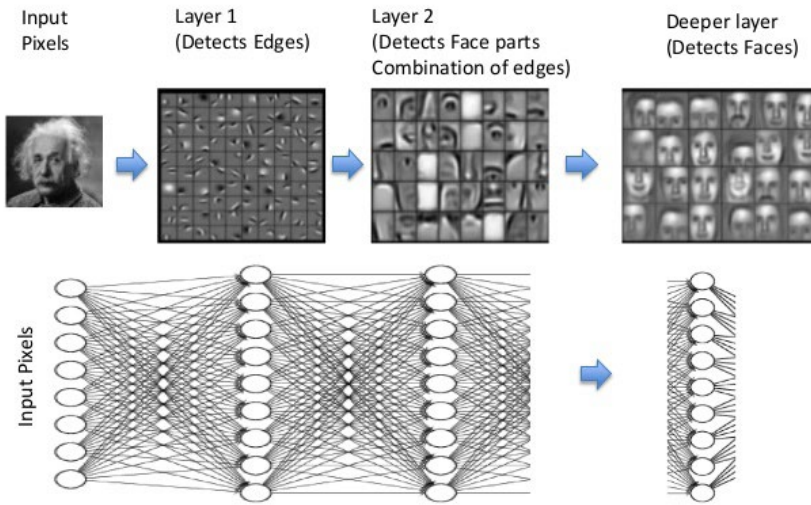
Her derin öğrenme algoritması bir makine öğrenmesi algoritmasıdır çünkü verilerden öğrenme gerçekleştirmektedir. Ancak her makine öğrenmesi algoritması derin öğrenme algoritması değildir; nitekim derin öğrenme, makine öğrenmesinin spesifik bir türüdür.

Derin öğrenme algoritması da veriye dayalı öğrenme gerçekleştirmekle birlikte, öğrenme süreci standart makine öğrenmesi algoritmalarında olduğu gibi tek bir matematiksel modele değil sinirsel ağ (neural network) olarak ifade edilen ağ diyagramlarına dayalı hesaplamalarla çalışmaktadır.

Derin öğrenme algoritması, makine öğrenmesi algoritmasının bir alt dalı olup öğrenmeye esas teşkil edilen verilerin makine öğrenmesi algoritması içinde birebir değil de **katman** olarak ifade edilen ve verinin özel bir tür temsili ile işlev görür². Bu katmanlar bizim bir bütün olarak algıladığımız sözgelimi bir fotoğrafın en küçük bilgi içerebilen parçasından tam olarak fotoğrafa dönüşene değin her aşamasını içeren temsili varlıklardır.

Aşağıdaki şekilde³ ünlü bilim adamlarının farklı fotoğraflarını tanımaya çalışan bir algoritmanın temsili çalışma şekli bulunmaktadır. Algoritma yukarıda haber örneğinde "haber_ornekleri[1]" şeklinde ifade edilen öğrenme verisinin herhangi bir satırında Einstein'a ait fotoğrafı piksellere ayırmaktadır. Bu pikseller alt kısımdaki "input pixels" kısmında sayılara dönüşmektedir (Siyah beyaz bir fotoğraf için her bir piksel 1 ila 256 değeri arasındaki bir sayısal değerdir. Siyah ile beyaz arasındaki tüm tonlar burada ifade edilir). *Layer1* kısmındaki matematiksel fonksiyonlar fotoğraf içindeki kenarları algılamakta, *Layer2* içinde ise artık fark edilebilen göz, burun ve resmin diğer unsurları ortaya çıkmaktadır. En son katmadan ise sözgelimi Einstein'a ait 1.000 fotoğrafın ortak teması olan yüzler ortaya çıkmaktadır. Ortaya çıkan tüm görseller herhangi bir katmandaki piksellerin tekrar resme dönüştürülmesiyle ortaya çıkar.

Feature Learning/Representation Learning (Ex. Face Detection)

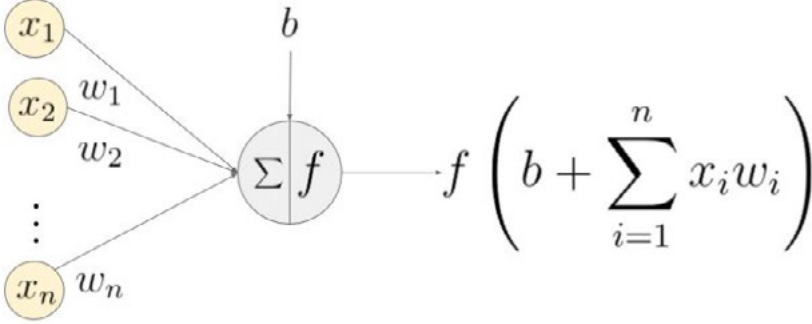


1 <http://www.teknokulis.com/haberler/internet/2016/03/24/microsoftun-twitter-kullanan-yapay-zekasi-kisa-surede-irkci-oldu>

2 Chollet, F. (2017). Deep Learning with Python. Manning Publications Co.

3 Rukshan Batuwita. (13:31:24 UTC). Deep Learning: Towards General Artificial Intelligence. Data & Analytics. <https://www.slideshare.net/RukshanBatuwita/deep-learning-towards-general-artificial-intelligence> adresinden erişildi.

İlk katmanlardaki düğümler resmin piksellerini ifade etmekte iken düğümler arasındaki çizgiler ise ağırlık değerlerini (w) ifade etmektedir. Her bir düğümden bulunan sayısal değerler çizgilerle gösterilen ağırlık değerleri ile çarpılarak müteakip düğümden sayısal değeri ortaya çıkarılır. Daha sonra bu sayısal değer özel tanımlanmış fonksiyonlara bağımsız değişken olarak girerek yeni bir sayısal değer ortaya çıkarır. Ortaya çıkan bu değer ise akış içinde sonraki düğümler için bir girdi teşkil eder ve aynı şekilde hesaplanarak en son düğümlere kadar gider.



x ile ifade edilen değerler her bir pikselin sayısal değeri, w ile ifade edilenler ise ilgili x değerinin çarpanı olan w değerini müteakip düğüm ise ağırlıklı çarpanların toplamını oluşturmaktadır. Yukarıdaki işlem Einstein'ın her resmi ve arşivdeki diğer tüm fotoğraflar için gerçekleştirilmektedir.

Yukarıdaki örnekte Einstein fotoğrafının süreç sonunda Einstein olarak tanıtılması durumu görülmektedir. Ancak etiketi belli olmayan bir fotoğrafın bu sürece girdiğinde kime ait olduğunun ortaya çıkarılabilmesi için her bir w değerleri ile düğümlerdeki fonksiyonların değerinin bilinmesi gerekmektedir. Bunun için ise ağ diyagramındaki süreç bir nevi tersine mühendislikle çözülmeye çalışır. Bunun için "Backpropagation (geri besleme)" adı verilen algoritma kullanılmaktadır. Bu algoritma öncelikle ağırlık değerleri (w)'yi tesadüfi olarak atar, bu değerlerin sonuçları ile gerçek sonuçlar arasındaki sayısal farka göre tıpkı Gradient Descent algoritmasından olduğu gibi kendini yeniler. Eldeki verilerle etiketler arasındaki ilişkiyi en iyi ifade eden konfigürasyon elde edildiğinde ise durur. Elde edilen konfigürasyonda artık w değerleri bilindiğinden algoritma belirli bir doğruluk gücünde tahmin yapar.

Derin Öğrenme algoritmalarının standart makine öğrenmesi algoritmalarına göre en büyük üstünlüğü *feature engineering* adı verilen süreci gerektirmemesidir. Bu durum bir örnekle şöyle ifade edilebilir. Diyelim ki bir şirkete ait 100 parametre ile iflas tahmini yapan bir algoritma olsun. Standart makine öğrenmesi algoritmasında hangi parametrelerin işe yarar olduğu ve modele konması gerektiği sorununu geliştirici çözmek zorundadır. Bunun için teoriden ya da parametre seçim algoritmalarından faydalanabilir. Bu süreç *feature engineering* olarak adlandırılmaktadır. Derin öğrenmede ise bu süreç yoktur. Derin öğrenme modeli, verinin yapısına göre hangi parametrelere ne ağırlık verileceğini kendisi keşfetmektedir. Derin öğrenmenin bu özelliği bir makine öğrenmesi algoritmasını daha esnek hale getirmekte ve ürüne dönüşümünde daha fazla kolaylık sağlamaktadır. Bu nedenledir ki, Kaggle gibi platformlarda derin öğrenme algoritmaları genellikle diğer algoritmalara göre açık ara öndedir. Aynı şekilde bugün ürüne dönüşmüş olan insansız otomobiller gibi teknolojilerin ardında ise derin öğrenme vardır.

Derin öğrenme algoritmalarının makine öğrenmesi algoritmalarına göre dezavantajı ise ihtiyaç duyulan donanım ve veri kapasitesidir. Derin öğrenme algoritmaları devasa miktarda kompüsyonel işlem yapmakta ve bazen GPU gibi donanımları gerektirmektedir. Birçok standart makine öğrenmesi algoritmasının masaüstü bilgisayarınızdan dahi çalıştırılması mümkün iken derin öğrenme algoritmaları için bu durum genellikle çok zor ya da imkânsızdır. Bu nedenle derin öğrenme algoritmaları için özel donanımların teminine ya da Amazon EC2 gibi bulut platformlar üzerinden kiralanmasına ihtiyaç vardır. 