Machine Learning

Machine Learning'in gerçekte ne olduğu -- training ve inference farkı, supervised /unsupervised/reinforcement öğrenme türleri, ve overfitting/underfitting'in eğitimi nasıl yanlış gösterebileceği.

Başlangıç 18 dk

Son güncelleme: 6 Ekim 2026
Çevrimiçi oku: https://www.learnforgex.com/tr/topics/machine-learning

---

Machine Learning

Önceki ders, "Yapay Zeka Nedir?", bir haritayla bitmişti: AI, Machine Learning'i içerir, Machine Learning Deep Learning'i içerir, Deep Learning de bugünün Generative AI'ının arkasındaki tekniktir. Bu ders o haritadaki ikinci daireye yakınlaşıyor. AI, "görünüşte zeka gerektiren görevleri yerine getiren yazılım" şeklindeki geniş fikirse, Machine Learning (ML) günümüz AI sistemlerinin çoğunun buna ulaşmak için gerçekte kullandığı somut *mekanizmadır*: **davranışı bir programcı tarafından elle yazılmak yerine veriden öğrenmek.** Bu dersin sonunda "eğitim" (training) kelimesinin gerçekte ne anlama geldiğini, bir sistemin öğrenebileceği üç ana yolu ve eğitilmiş bir sistemin bazen neden yanlış sonuç verdiğini açıklayan iki başarısızlık modunu -- overfitting ve underfitting -- bileceksiniz.

Machine Learning Nedir?

Machine Learning, bir sistemin bir görevdeki performansını, bir insanın açıkça programladığı kurallar yerine veriden örüntüler öğrenerek geliştirdiği AI alt alanıdır. Somut olarak: `if (email "free money" içeriyorsa) then spam = true` yazmak yerine, sisteme "spam" ya da "spam değil" olarak zaten etiketlenmiş binlerce e-posta gösterirsiniz, ve bir *öğrenme algoritması*, hangi örüntülerin (hangi kelimeler, göndericiler, biçimlendirme tuhaflıkları) iki kategoriyi ayırdığını otomatik olarak bulur. Bu sürecin çıktısı bir **model**'dir: sabit bir yapı (tahmin üreten bir fonksiyon/mimari) ile bu yapının içindeki öğrenilmiş **parametreler**in (sayılar) birleşimi. Eğitimi gerçekten yürüten şeye **öğrenme algoritması** denir -- modelin kendisi değil, modeli *üreten* süreçtir. Eğitim bittiğinde elinizde artık öğrenme algoritmasına ihtiyaç duymadan çalışan bir model kalır: hiç görmediği yepyeni bir e-postaya bakıp spam olup olmadığını tahmin edebilir.

Bu, "Yapay Zeka Nedir?" dersinin "AI, davranışı veriden veya örneklerden öğrenilen yazılımdır" diye tanıttığı fikrin ta kendisi. Machine Learning, o cümlenin pratikte anlattığı şeydir.

Neden Var?

Önceki ders, el yazımı kuralların gerçek dünyanın karmaşık problemlerinde neden çöktüğünü göstermek için "bu fotoğraf bir kedi mi?" örneğini kullanmıştı. Machine Learning, bir çözümü elle kodlamayı farklı, daha ele alınabilir bir probleme dönüştürdüğü için var: **örüntüyü kendiniz tarif etmek yerine, onun örneklerini toplayıp örüntüyü sizin için bir algoritmaya buldurursunuz.** Bu değişim çok pratik bir nedenle önemli -- devasa bir problem sınıfı için (görüntü tanıma, dili anlama, fiyat tahmini, dolandırıcılık tespiti) hiç kimse elle güvenilir, açık bir kural kümesi yazmayı başaramadı, ama bilgisayarlar defalarca yeterince örnekten bir tane *öğrenmeyi* başardı.

> Yararlı bir kontrol: mantığı makul boyutta bir `if` / `else` kural kümesi olarak kendiniz makul biçimde yazabileceğinizi düşünüyorsanız, muhtemelen machine learning'e ihtiyacınız yok -- düz kod daha basit, daha hızlı ve hata ayıklaması daha kolay olacaktır. ML'i, açık ve kararlı kurallarla kolayca çözülemeyen ve verideki örüntülerin öğrenilmesinin gerçek bir avantaj sağladığı problemler için saklayın.

Training ve Inference

Her machine learning sisteminin yaşamında iki farklı aşama vardır, ve bunları karıştırmak yeni başlayanlar için en yaygın yanlış anlama kaynaklarından biridir:

- **Training (eğitim),** bir modele birçok örnek göstererek tahminlerinin giderek iyileşmesi için iç parametrelerini ayarlama sürecidir (genellikle yavaş ve hesaplama açısından pahalıdır). Bu *bir kez* (ya da modeli güncellemek istediğinizde periyodik olarak) gerçekleşir, genellikle çevrimdışıdır, bazen dakikalar, saatler, ya da -- bu kursun ilerleyen bölümlerinde ele alınacak büyük modeller için -- özel donanımlarda haftalar sürebilir.
- **Inference (çıkarım),** *zaten eğitilmiş* bir modeli yeni bir girdi üzerinde tahmin yapmak için kullanmaktır. Bu, bir spam filtresinin gelen bir e-postayı her kontrol edişinde ya da bir telefonun yüz tanımayla her açılışında olan şeydir. Inference tipik olarak hızlıdır (milisaniyeler) ve son kullanıcıların gerçekte deneyimlediği şey budur.

Training'i "sınava çalışmak," inference'ı ise "çalıştığınız bilgiyle sınava girmek" olarak düşünün. Training aşamasını iyi geçiren ama hiçbir zaman inference için gerçekten kullanılmayan bir model pratikte işe yaramazdır -- ve inference'ta hızlı olan ama kötü veriyle eğitilmiş bir model, kendinden emin biçimde kötü tahminler üretir.

Üç Öğrenme Türü: Supervised, Unsupervised, Reinforcement

Tüm machine learning aynı şekilde çalışmaz. Sistemin hangi tür veriden öğrendiğine ve (varsa) nasıl bir "geri bildirim" aldığına göre ayrılan üç geniş kategori vardır:

- **Supervised Learning (denetimli öğrenme):** sistem, üzerinde zaten "doğru cevap" bulunan veriden öğrenir -- örneğin, zaten "kedi"/"kedi değil" diye etiketlenmiş fotoğraflar,

ya da bilinen satış fiyatına sahip evler. Sistemin görevi, girdiden doğru çıktıya olan eşleşmeyi öğrenmektir, böylece yeni, etiketlenmemiş girdiler için cevabı tahmin edebilir. Bu, açık ara en yaygın ve en olgun kategoridir, ve yukarıdaki spam filtresi örneğinin anlattığı şey de tam olarak budur.

- **Unsupervised Learning (denetimsiz öğrenme):** sisteme üzerinde **hiçbir** doğru cevap bulunmayan veri verilir, ve verideki yapıyı kendi başına bulmak zorundadır -- örneğin, müşterileri satın alma davranışına göre segmentlere ayırmak, segmentlerin ne olması gerektiğini kimse önceden söylemeden (*clustering* -- kümeleme denen bir teknik).
- **Reinforcement Learning (pekiştirmeli öğrenme):** sistem (bir *agent* olarak adlandırılır), bir ortamda eylemler gerçekleştirerek ve geri bildirim olarak bir *ödül* ya da *ceza* alarak öğrenir, zamanla hangi eylemlerin daha iyi sonuçlara yol açtığını kademeli olarak öğrenir -- tıpkı oyun oynayan bir AI'ın kendi kendine milyonlarca oyun oynayarak strateji öğrenmesi gibi. Bu kategori kavramsal olarak bu kursun ilerleyen "AI Agent'lar" kategorisine en yakın olanıdır, ama orada inşa edeceğiniz agent'lar klasik pekiştirmeli öğrenme yerine, büyük dil modelleri üzerine kurulu farklı, daha modern bir teknik kümesi kullanır.

Feature'lar ve Label'lar: Bir Modelin Gerçekte Öğrendiği Şey

Herhangi bir supervised learning sisteminin içine baktığınızda sürekli karşınıza çıkan iki terim var:

- **Feature (öznitelik),** modelin tahmin yaparken kullandığı, ölçülebilir bir girdi bilgisi parçasıdır -- bir e-posta spam filtresi için feature'lar, ünlem işareti sayısını, göndericinin bilinen bir kişi olup olmadığını ya da belirli kelimelerin varlığını içerebilir. İyi feature'lar seçmek ve hazırlamak (*feature engineering* denen bir süreç), özellikle bu dersteki klasik ML tekniklerinde bir sistem inşa etmenin en zaman alan ve beceri gerektiren parçalarından biriydi -- feature'ların büyük ölçüde elle tasarlanması gerekiyordu. Bunun her ML tekniğinde aynı ölçüde geçerli olmadığını ileride "Deep Learning" dersinde göreceksiniz.
- **Label (etiket),** bir eğitim örneğine iliştirilmiş doğru cevaptır -- "spam" ya da "spam değil," bir evin gerçek satış fiyatı, bir fotoğrafın doğru kategorisi. Label'lar, supervised learning'i "supervised" (denetimli) yapan şeydir: bir insanın (ya da otomatik bir sürecin) bunları önceden sağlamış olması gerekir.

Dataset (veri kümesi) ise, her biri kendi feature'larıyla (ve supervised learning için kendi label'ıyla) birlikte gelen büyük bir örnek koleksiyonudur, başka bir şey değil. Bu veri kümesinin kalitesi ve boyutu, çoğu zaman hangi belirli öğrenme algoritmasının kullanıldığından *daha* önemlidir bir modelin nihai performansı için -- bu kursun tekrar tekrar döneceği bir tema.

Overfitting ve Underfitting

Bir model eğitildikten sonra, eğitimin gerçekten *iyi* çalışıp çalışmadığını nasıl bilirsiniz? İki isimlendirilmiş başarısızlık modu, işlerin ters gitmesinin en yaygın iki yolunu tarif eder:

- **Overfitting,** bir modelin eğitim verisini *çok* hassas biçimde -- altındaki genel örüntü yerine gürültüsü ve tuhaflıklarıyla birlikte -- öğrenmesi durumunda ortaya çıkar. Overfit olmuş bir model, eğitildiği tam örneklerde mükemmel sonuçlar alır, ama yeni, görülmemiş veride kötü performans gösterir, çünkü altındaki kuralı öğrenmek yerine etkin biçimde "cevapları ezberlemiştir."
- **Underfitting** ise tam tersi bir sorundur: model, verideki gerçek örüntüyü bile yakalayamayacak kadar basittir (ya da yeterince eğitilmemiştir), bu yüzden hem eğitim verisinde hem yeni veride kötü performans gösterir.

Overfitting için pratik çözüm neredeyse her zaman modeli eğitim sırasında hiç görmediği veri üzerinde değerlendirmeyi (bir sonraki bölüme bakın) ve modelin ezberlemek yerine iyi genelleme yapmasını sağlayacak teknikleri -- daha fazla eğitim verisi, daha basit modeller, ya da modelin ne kadar sıkı uyduğunu bilinçli olarak sınırlamak -- kullanmayı içerir.

Bir Modelin İyi Olduğunu Nasıl Biliriz?

Standart pratik, eğitime başlamadan önce mevcut veri kümesini üç parçaya bölmektir:

- **Training set (eğitim kümesi):** modelin gerçekten öğrendiği örnekler.
- **Validation set (doğrulama kümesi):** eğitim sırasında modele gösterilmeyen, ama model/hyperparameter seçimleri yaparken (örneğin hangi mimarinin ya da hangi ayarların daha iyi sonuç verdiğine karar verirken) kullanılan, kenara ayrılmış örnekler.
- **Test set (test kümesi):** ne eğitimde ne de bu seçimlerde kullanılan, tamamen kenara ayrılmış örnekler -- modelin *final* performansını, tüm kararlar verildikten sonra, dürüstçe ölçmek için son kez kullanılır.

Bu üçe bölmenin nedeni basit: validation set'i model seçimi için kullanırsanız, modeliniz dolaylı olarak o veriye de biraz "uyum sağlamış" olur -- bu yüzden gerçekten hiç dokunulmamış, ayrı bir test set'e hâlâ ihtiyacınız vardır. Bu dersin amacı doğrulama sürecinin ayrıntılarına girmek değil, yalnızca doğru zihinsel modeli kurmak: training set öğrenir, validation set seçim yapar, test set final skoru verir. Eğitim kümesinde iyi ama test kümesinde kötü performans gösteren bir model, overfitting'in ders kitabı örneğidir.

> Bir modelin eğitim kümesi skoru, gerçek dünyadaki kalitesinin GÜVENİLİR bir ölçüsü DEĞİLDİR -- neredeyse her zaman modelin gerçek performansından daha yüksektir, tam olarak model bu örnekleri daha önce gördüğü için. Bir AI sisteminin ne kadar "doğru" olduğuna dair bir iddia okuduğunuzda, önemli soru her zaman şudur: modelin eğitim sırasında hiç görmediği *hangi veride* doğru?

Bu kursun ilerleyen AI Evaluation kategorisi, modern sistemlerin (özellikle büyük dil modellerinin) gerçekte nasıl ölçüldüğüne çok daha derinlemesine girecek -- bu ders yalnızca bunun temelini oluşturan training/test ayrımını anlamanızı gerektiriyor.

Machine Learning'den Deep Learning'e: Sırada Ne Var?

Bu derste anlatılan her şey, öğrenmeyi gerçekleştiren belirli algoritma ne olursa olsun, machine learning'e genel olarak uygulanır -- ve tarihsel olarak birçok algoritma olmuştur: karar ağaçları (decision trees), lineer regresyon, destek vektör makineleri (support vector machines) ve diğerleri, bunların çoğu son AI patlamasından onlarca yıl öncesine dayanır. Bir sonraki ders, "Deep Learning," öğrenme algoritmalarının belirli bir ailesine -- **neural network'lere (sinir ağları),** özellikle *çok katmanlı* olanlara -- yakınlaşıyor -- yeterince veri ve hesaplama gücü mevcut olduğunda eski tekniklerden çarpıcı biçimde daha iyi ölçeklendiği ortaya çıkan aile ("Yapay Zeka Nedir?" dersinin bahsettiği 2012 dönüm noktasını hatırlayın). Deep learning bu dersteki kavramların -- training, inference, supervised learning, overfitting -- yerini almadı; bunları *gerçekleştirmek için* belirli, son derece başarılı bir teknik, ve az önce öğrendiğiniz her şey ona doğrudan uygulanmaya devam ediyor.

Best Practices

- Machine learning'e başvurmadan önce, problemin düz, açık kodla çözülüp çözülemeyeceğini kontrol edin ("Neden Var?" bölümündeki ipucuna bakın) -- ML gerçek bir maliyet ekler (veri toplama, eğitim altyapısı, sürekli izleme) ve açık, kararlı kuralları olan problemler için bu maliyet haklı çıkmaz.
- Bir modeli her zaman üzerinde eğitilmediği veride değerlendirin. "Hangi test verisinde ölçüldü?" sorusuna net bir cevabı olmayan bir sayıya güvenilmemelidir.
- Veri kümenizi kodunuz kadar ciddiye alın. İyi ayarlanmış bir öğrenme algoritması, kötü, önyargılı ya da çok küçük veriyle eğitildiğinde güvenilir biçimde kötü, önyargılı ya da güvenilmez bir model üretir -- "garbage in, garbage out" (çöp girer, çöp çıkar) ilkesi, yazılımın neredeyse başka hiçbir yerinde olmadığı kadar burada geçerlidir.

Yaygın Hatalar

- **Bir modeli yalnızca eğitim performansına göre değerlendirmek.** "Overfitting ve Underfitting"te anlatıldığı gibi, bir model eğitim verisinde neredeyse mükemmel bir skor alırken gerçek, yeni girdilerde kötü performans gösterebilir. Her zaman özellikle test kümesi performansını sorun.
- **"Machine learning"in "neural network" anlamına geldiğini varsaymak.** Bir sonraki dersin netleştireceği gibi, neural network'ler (deep learning) günümüzde yaygın olarak kullanılan, ama alanla eş anlamlı OLMAYAN, birkaç ML tekniği ailesinden biridir.
- **Reinforcement learning ile supervised learning'i birbirinin yerine kullanılabilir saymak.** Bunlar çok farklı türde problemleri çözer (etiketli tahmin ile deneme-yanılma geri bildirimiyle öğrenme) -- "Üç Öğrenme Türü: Supervised, Unsupervised,

Reinforcement"e bakın -- ve bir sistemin gerçekte hangisini kullandığını karıştırmak, nasıl eğitilebileceği ya da geliştirilebileceği konusunda yanlış beklentilere yol açar.

Özet, Cheat Sheet ve Terimler Sözlüğü

Özet

- Machine Learning, sistemlerin davranışı el yazımı kurallardan değil veriden öğrendiği AI alt alanıdır -- "Yapay Zeka Nedir?"te tanıtılan genel fikrin arkasındaki somut mekanizmadır.
- Her ML sisteminin iki aşaması vardır: **training** (örneklerden öğrenmek, genellikle yavaş) ve **inference** (öğrenileni kullanmak, genellikle hızlı).
- Üç geniş öğrenme türü vardır: **supervised** (etiketli örneklerden öğrenir), **unsupervised** (etiket olmadan yapı bulur), ve **reinforcement** (ödül/ceza geri bildiriminden öğrenir).
- **Overfitting** (eğitim verisini çok hassas biçimde ezberlemek) ve **underfitting** (gerçek örüntüyü yakalayamamak), eğitimin ters gidebileceği iki ana yoldur -- kenara ayrılmış bir test kümesinde değerlendirerek yakalanır.
- Deep Learning (sıradaki ders), neural network'lere dayanan, belirli ve son derece başarılı bir ML teknikleri ailesidir -- bu dersteki kavramların yerine geçmez, onları büyük ölçekte uygulamanın bir tekniğidir.

Cheat Sheet

- Model = mimari (tahmin üreten fonksiyon) + öğrenilmiş parametreler. Öğrenme algoritması = modeli *üreten* eğitim süreci, modelin kendisi değil.
- Training = örneklerden öğrenmek. Inference = öğrenileni kullanmak.
- Feature = modelin kullandığı bir girdi. Label = doğru cevap (yalnızca supervised learning).
- Training set = model öğrenir. Validation set = model/hyperparameter seçimi yapılır. Test set = final performans dürüstçe ölçülür.
- Overfit = eğitim verisine çok yakın uyar, yeni veride başarısız olur. Underfit = çok basittir, her yerde başarısız olur.
- Her zaman modelin hiç eğitilmediği bir test kümesinde değerlendirin.

Terimler Sözlüğü

- **Machine Learning (ML):** sistemlerin davranışı açık kurallar yerine veriden öğrendiği AI alt alanı.
- **Model:** bir öğrenme sürecinin eğitilmiş çıktısı -- parametreler artı onları tahmin yapmak için kullanan bir algoritma.
- **Training / Inference:** bir ML sisteminin yaşamındaki iki aşama -- veriden öğrenmek, sonra öğrenileni yeni girdide kullanmak.
- **Supervised / Unsupervised / Reinforcement Learning:** sistemin hangi türde veriden ve geri bildirimden öğrendiğine göre ayrılan üç geniş machine learning kategorisi.

- **Feature / Label:** modelin tahmin için kullandığı bir girdi / bir eğitim örneğine iliştirilmiş doğru cevap.
- **Overfitting / Underfitting:** eğitim verisini genelleyemeyecek kadar hassas biçimde öğrenmek / gerçek örüntüyü hiç öğrenememek.
- **Training set / Validation set / Test set:** bir modelin öğrendiği veri / model ya da hyperparameter seçimi yaparken kullanılan kenara ayrılmış veri / modelin gerçekte ne kadar iyi performans gösterdiğini son kez ve dürüstçe ölçmek için kullanılan, hiç dokunulmamış veri.