Yapay Sinir Ağları

Yapay Zekâ ve Veri Dersleri Rehberi · Kısım I: Çekirdek

Yapay nöron | Algılayıcı | Çok katmanlı ağ | Geri yayılım | Eğitim ve genelleme | Komşu dersler

Sercan KÜLCÜ · Giresun Üniversitesi · Bilgisayar Mühendisliği

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/ai_courses/sunumlar/Bolum_04_Yapay_Sinir_Aglari.html

GİRİŞ

Ders Hangi Soruyu Sorar?

> **Basit birimlerden kurulan ağlar ağırlıklarını ayarlayarak nasıl öğrenir?**

Yapay sinir ağları, ağırlıklı toplam alıp bir etkinleştirme fonksiyonundan geçiren basit birimlerin katmanlar hâlinde bağlanmasıyla kurulan modelleri inceler. Öğrenme, kayıp fonksiyonunu küçültecek biçimde **ağırlıkları ayarlamaktır**; bunun temel aracı geri yayılım ve gradyan inişidir.

Bir bakışta

YAKLAŞIM	GİRDİ	ÇIKTI
ağırlıklı basit birimlerden ağ	sayısal öznitelik vektörü	sınıf ya da sayı tahmini

BAŞARI ÖLÇÜTÜ	MATEMATİK YÜKÜ	VERİ İHTİYACI
kayıp ve doğruluk	orta-yüksek: türev, matris	orta

Kısa Tarihçe

1943
McCulloch ve Pitts, eşik değerli mantıksal birimlerle ilk yapay nöron modelini önerir.

1949
Hebb, birlikte etkinleşen nöronların bağının güçlendiğini öne sürer: Hebb öğrenme kuralı.

1957
Rosenblatt algılayıcıyı (perceptron) tanıtır; ağırlıklar örneklerden ayarlanır.

1969
Minsky ve Papert, tek katmanlı algılayıcının XOR gibi problemleri çözemeyeceğini gösterir.

1982
Hopfield, belleği enerji en küçüklemesiyle geri çağıran yinelemeli ağı yayımlar.

1986
Rumelhart, Hinton ve Williams geri yayılımı yaygınlaştırır; çok katmanlı ağlar eğitilebilir olur.

DERS

Temel Kavramlar

Yapay nöron
Girdilerin ağırlıklı toplamı, eşik (bias) ve çıkış.

Etkinleştirme
Sigmoid, tanh, ReLU: ağa doğrusal olmama katar.

Algılayıcı
Doğrusal ayrılabilen sınıfları ayıran tek katmanlı ağ.

Çok katmanlı ağ
Gizli katmanlarla doğrusal olmayan sınırlar öğrenilir.

İleri yayılım
Girdiden çıkışa katman katman tahmin hesaplamak.

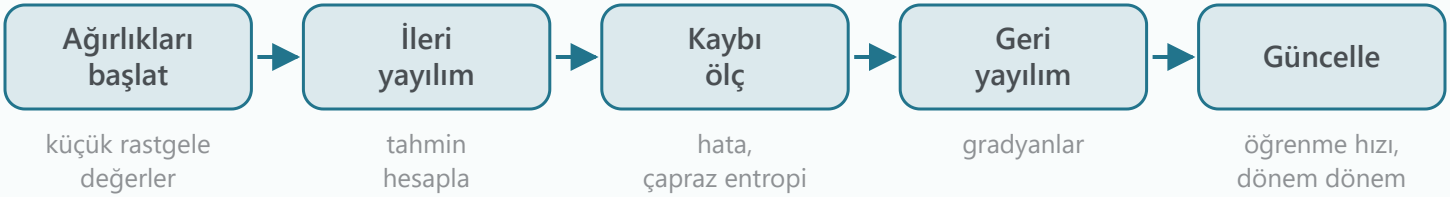
Geri yayılım
Zincir kuralıyla her ağırlığın gradyanını bulmak.

Gradyan inişi
Ağırlıkları kaybın azaldığı yöne küçük adımlarla taşır.

Düzenlileştirme
Ağırlık cezası ve erken durdurma ile aşırı uydurmayı önler.

DERS

Tipik İş Akışı



Örnek: El yazısı rakam tanıyan küçük bir ağ, 28×28 piksellik görüntüyü 784 girdi olarak alır, gizli katmandan geçirip 10 sınıfa olasılık verir. Yanlış tahminde kayıp büyür; geri yayılım her ağırlığın payını hesaplar ve gradyan inişi ağırlıkları biraz düzeltir.

DERS

Derste Neler İşlenir?

1. Biyolojik ve yapay nöron; McCulloch-Pitts modeli
2. Hebb kuralı ve algılayıcı öğrenme algoritması
3. Adaline, en küçük kareler ve delta kuralı
4. Doğrusal ayrılabilirlik ve XOR problemi
5. Çok katmanlı ağ ve etkinleştirme fonksiyonları
6. Geri yayılım: hesaplama çizgesi ve zincir kuralı
7. Kayıp fonksiyonları: kare hata, çapraz entropi
8. Gradyan inişi, öğrenme hızı ve momentum
9. Ağırlık başlatma ve gradyanın sönmesi
10. Genelleme: düzenlileştirme, erken durdurma
11. Radyal tabanlı fonksiyon ağları
12. Hopfield ağı ve öz düzenleyen haritalar (SOM)

Kazandırdığı beceriler

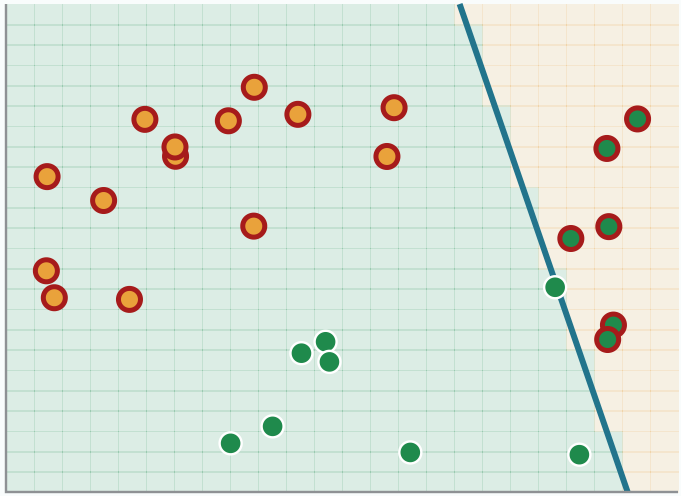
NumPy ile ağı sıfırdan yazmak · geri yayılımı elle türetmek · gradyanı sayısal olarak denetlemek · PyTorch ve Keras ile ağ eğitmek · öğrenme eğrilerini okumak

Tipik bir dönemin akışıdır; sıra ve kapsam üniversiteden üniversiteye değişir.

DENE

Mandalina mı, Limon mu?

Ayıklama bandındaki kamera her meyvenin ağırlığını ve renk tonunu ölçüyor. Tek bir yapay nöron (algılayıcı) iki türü ayırmayı öğreniyor: eğitin, ya da bir meyveye tıklayıp etiketini değiştirin.



Bir adım eğit | Bir tur | 10 tur | Sıfırla

çıktı = işaret(**+0,40**·ağırlık **+0,10**·ton **−0,37**)

Tur 0, sıradaki örnek 1/28.
Yanlış sınıflanan: **20** meyve

● mandalina ● limon. Bir limonu turuncuların ortasına taşıyın: hiçbir doğru ayıramaz, hata **sıfıra inmez**.

Kuralı kimse yazmadı: nöron her hatada ağırlıkları hatanın yönüne biraz kaydırdı. Tek nöron yalnız **doğrusal** sınır çizebilir; ayrılamayan veride katmanlı ağa ihtiyaç doğar.

KONUM

Ön Koşullar

Bu dersten önce bilinmesi yararlı olan dersler ve neden gerektikleri.

Ders	Neden gerekir
Makine Öğrenmesi	parametrik model, kayıp fonksiyonu, gradyan inişi
Doğrusal Cebir	ağırlık matrisleri, vektörleştirilmiş hesap
Optimizasyon	gradyan inişi, öğrenme hızı
Analiz (Kalkülüs)	türev, kısmi türev ve zincir kuralı: geri yayılımın matematiği
Programlama	Python ile diziler, döngüler ve fonksiyonlar

KONUM

Bu Ders Sana Göre mi?

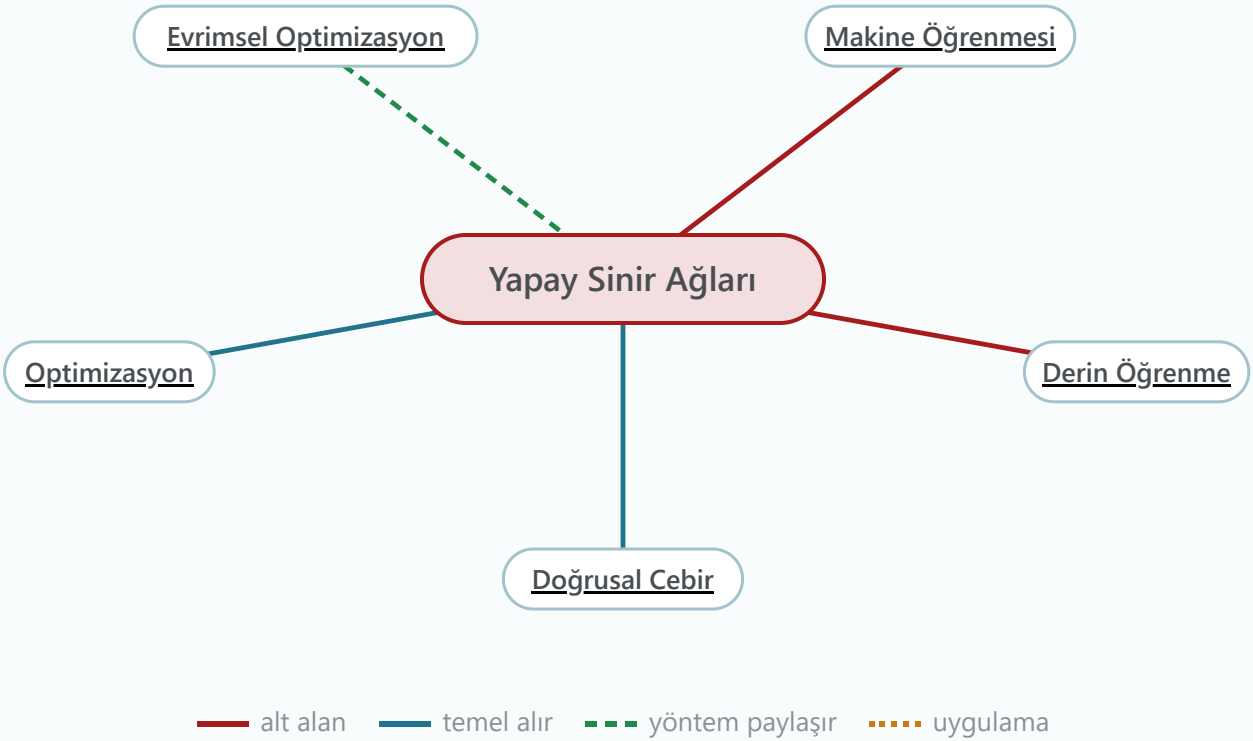
Kendine sor; "hayır" dediklerin için sağdaki sütun nereden başlayacağını gösterir.

Kendine sor	Hayırsa önce
Bileşik bir fonksiyonun türevini zincir kuralıyla alabiliyor musun?	Analiz dersindeki türev ve zincir kuralı konuları
Bir matrisi bir vektörle çarpıp sonucun boyutunu söyleyebiliyor musun?	Doğrusal Cebir
Python'da NumPy dizileriyle döngüsüz toplu işlem yapabiliyor musun?	Programlamaya giriş, ardından NumPy'a kısa bir giriş
Eğitim ve test hatası arasındaki farkı açıklayabiliyor musun?	Makine Öğrenmesi

Çoğuna "evet" diyebiliyorsan derse hazırsın; bir ya da iki "hayır" dönem içinde kapanabilir.

KONUM

Dersin Komşuları



ÖRTÜŞME

Örtüştüğü Dersler

İlişki	Ders	Ortak konu
alt alan	Bu ders, Makine Öğrenmesi dersinin alt alanıdır	parametrik model, kayıp fonksiyonu, gradyan inişi
alt alan	Derin Öğrenme, bu dersin alt alanıdır	çok katmanlı ağlar, geri yayılım, etkinleştirme fonksiyonları
temel alır	Bu ders Doğrusal Cebir dersini temel alır	ağırlık matrisleri, vektörleştirilmiş hesap
temel alır	Bu ders Optimizasyon dersini temel alır	gradyan inişi, öğrenme hızı
yöntem paylaşır	Evrimsel ve Metasezgisel Optimizasyon ile yöntem paylaşır	nöroevrim, hiperparametre arama

AYRIŞMA

Ayrıştığı Noktalar (1/2)

Ölçüt	Yapay Sinir Ağları	Makine Öğrenmesi	Derin Öğrenme
Temel soru	Basit birimlerden kurulan ağlar ağırlıklarını ayarlayarak nasıl öğrenir?	Bir program örneklerden nasıl genelleme yapar?	Çok katmanlı ağlar ham veriden kullanışlı temsilleri nasıl kendisi öğrenir?
Yaklaşım	ağırlıklı basit birimlerden ağ	istatistiksel öğrenme	çok katmanlı temsil öğrenme
Girdi	sayısal öznitelik vektörü	etiketli ya da etiketsiz örnekler	ham veri: görüntü, metin, ses
Çıktı	sınıf ya da sayı tahmini	tahmin yapan model	öğrenilmiş temsil ve tahmin
Başarı ölçütü	kayıp ve doğruluk	görülmemiş veride hata	doğruluk, kayıp, ölçeklenme
Matematik yükü	orta-yüksek: türev, matris	yüksek: doğrusal cebir, olasılık, optimizasyon	yüksek: türev, matris, olasılık
Veri ihtiyacı	orta	orta ve çok	çok; GPU gerekir

- **Makine Öğrenmesi:** MÖ karar ağacından SVM'ye birçok model ailesini kapsar; bu ders yalnız nöronlardan kurulan ağlara odaklanır.
- **Derin Öğrenme:** Derin öğrenme evrişimli, yinelemeli ve dönüştürücü mimarilere geçer; bu ders bunların dayandığı temel ağı ve eğitimini kurar.

Ayrıştığı Noktalar (2/2)

Ölçüt	Yapay Sinir Ağları	Optimizasyon	Evrimsel Optimizasyon
Temel soru	Basit birimlerden kurulan ağlar ağırlıklarını ayarlayarak nasıl öğrenir?	Bir amaç fonksiyonunu en iyi yapan değerler nasıl bulunur?	Türev yokken ya da arama uzayı çok büyükken iyi bir çözüm nasıl bulunur?
Yaklaşım	ağırlıklı basit birimlerden ağ	amaç ve kısıtların matematiksel modeli	popülasyon ve sezgisel arama
Girdi	sayısal öznitelik vektörü	amaç fonksiyonu ve kısıtlar	amaç fonksiyonu ve çözüm gösterimi
Çıktı	sınıf ya da sayı tahmini	en iyi ya da ona yakın çözüm	iyi bir çözüm (en iyisi garanti değil)
Başarı ölçütü	kayıp ve doğruluk	amaç değeri, yakınsama hızı	çözüm kalitesi, değerlendirme sayısı
Matematik yükü	orta-yüksek: türev, matris	kuramın kendisi	düşük-orta
Veri ihtiyacı	orta	gerekmez	gerekmez; değerlendirme yeter

- **Optimizasyon:** Optimizasyon gradyan yöntemlerini genel olarak inceler; bu ders onları ağırlıkları öğrenmek için kullanır.
- **Evrimsel Optimizasyon:** Nöroevrim ağırlıkları ya da mimariyi evrimle arar; bu ders ağırlıkları gradyanla, geri yayılımla öğrenir.

UYGULAMA

Uygulama Alanları

Rakam tanıma
Posta kodu ve çek üzerindeki el yazısı rakamlar.

Tablo verisi
Kredi riski, müşteri kaybı gibi sınıflandırmalar.

Fonksiyon yaklaşıklama
Ölçümlerden doğrusal olmayan ilişki öğrenmek.

Denetim
Basit robot ve süreç denetleyicileri.

Sinyal sınıflandırma
Sensör verisinden arıza ya da hareket tanıma.

İlişkisel bellek
Hopfield ağıyla bozuk örüntüyü tamamlamak.

UYGULAMA

Riskler ve Etik

Ezber ve aşırı uydurma

Küçük veride büyük ağ eğitim hatasını sıfırlar ama yeni örnekte çöker; doğrulama kümesi olmadan bu fark görülmez.

Kestirme öğrenme

Ağ asıl özniteliği değil, veri toplamanın izini öğrenebilir: fotoğraftaki cetvel, zemin rengi ya da tarayıcı damgası gibi.

Kararsız eğitim

Kötü başlatma ya da büyük öğrenme hızı kaybı patlatır; tek bir rastgele tohumla alınan sonuç tekrarlanamayabilir.

Gerekçesiz kararlar

Ağırlıklar okunabilir bir kural vermez; kredi ya da tıbbi karar gibi gerekçe istenen yerlerde bu ciddi bir sorundur.

Ayrıntılı tartışma: Açıklanabilir Yapay Zekâ ve Yapay Zekâ Etiği.

UYGULAMA

Araçlar ve Örnek Veriler

Araçlar ve kütüphaneler

NumPy | PyTorch | Keras | scikit-learn (MLPClassifier) | TensorFlow Playground | MATLAB Deep Learning Toolbox

Tanınmış veri setleri

XOR problemi | Iris | MNIST | Fashion-MNIST | İki sarmal (two spirals) | CIFAR-10

Bu derste bir ağı önce **NumPy ile elle** yazmak, geri yayılımın gerçekten ne yaptığını görmenin en iyi yoludur.

UYGULAMA

Sık Karıştırılanlar

Yaygın düşünce	Doğrusu
Yapay sinir ağları beyni taklit eder.	Esin kaynağı biyolojidir; ama yapay nöron çok kaba bir modeldir ve öğrenme kuralı beyninkinden farklıdır.
Daha çok nöron her zaman daha iyidir.	Büyük ağ eğitim verisini ezberleyebilir; boyut doğrulama kümesiyle seçilir.
Geri yayılım ayrı bir öğrenme yöntemidir.	Geri yayılım yalnız gradyanı hesaplar; ağırlıkları güncelleyen gradyan inişidir.
Eğitim her seferinde aynı sonucu verir.	Rastgele başlangıç ve karıştırma yüzünden farklı yerel en küçüklere varılabilir.

SONRASI

Bu Dersten Sonra

Bu dersi temel alan ya da genişleten dersler

Derin Öğrenme

İlgili meslekler

Makine öğrenmesi mühendisi | Derin öğrenme mühendisi | Araştırmacı | Gömülü YZ geliştiricisi

Bu ders derin öğrenmenin kapısıdır: evrişimli ağlar, yinelemeli ağlar ve dönüştürücüler burada kurulan nöron, kayıp ve geri yayılım fikirlerinin üzerine kurulur. Optimizasyon ve doğrusal cebir bilgisini de pekiştirir.

SONRASI

Kaynaklar

- Haykin, *Neural Networks and Learning Machines*, 3. baskı, Pearson, 2009.
- Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015 (çevrimiçi kitap).
- Rumelhart, Hinton ve Williams, "Learning representations by back-propagating errors", *Nature* 323, 1986.

Son

Bölüm 4: Yapay Sinir Ağları

← Önceki bölüm · Alanın haritası · Sonraki bölüm →