Gömülü Yapay Zekâ

Yapay Zekâ ve Veri Dersleri Rehberi · Kısım VI: Sistem ve Uygulama

Kaynak kısıtları | Nicemleme | Budama ve damıtma | Verimli mimariler | Cihaz ve hızlandırıcı | 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_36_Gomulu_Yapay_Zeka.html

GİRİŞ

Ders Hangi Soruyu Sorar?

> **Yapay zekâ modelleri küçük, düşük güçlü cihazlarda nasıl çalışır?**

Gömülü yapay zekâ, eğitilmiş modelleri telefon, kamera ya da mikrodenetleyici gibi belleği, işlem gücü ve enerjisi kısıtlı cihazlarda çalıştırmayı inceler. Ders modeli **nicemleme**, budama ve damıtmayla küçültmeyi, cihaza taşımayı ve gecikmeyle enerjiyi ölçmeyi kapsar; en uç biçimine TinyML denir.

Bir bakışta

YAKLAŞIM kaynağı kısıtlı çıkarım	**GİRDİ** eğitilmiş model, cihaz kısıtları	**ÇIKTI** küçük ve hızlı model
BAŞARI ÖLÇÜTÜ gecikme, enerji, bellek	**MATEMATİK YÜKÜ** orta	**VERİ İHTİYACI** eğitim verisi zaten var

Kısa Tarihçe

2015
Hinton, Vinyals ve Dean bilgi damıtmayı tanıtır: büyük modelin bilgisi küçük modele aktarılır.

2016
Han ve ark. Deep Compression ile budama ve nicemlemeyi birleştirip ağları onlarca kat küçültür.

2017
MobileNets yayımlanır, TensorFlow Lite duyurulur: telefonda derin ağ çalıştırmak olağanlaşır.

2018
Google, uç cihazlar için Edge TPU hızlandırıcısını duyurur.

2019
İlk tinyML zirvesi yapılır; Warden ve Situnayake TinyML kitabını yayımlar.

2020
MCUNet, ImageNet sınıflandırmasını bir mikrodenetleyicide %70'in üzerinde doğrulukla çalıştırır.

DERS

Temel Kavramlar

Nicemleme
32 bit kayan nokta yerine 8 bit ya da daha az tamsayı.

Budama
Önemsiz ağırlık, kanal ya da katmanları atmak.

Bilgi damıtma
Küçük öğrenci model büyük öğretmeni taklit eder.

Verimli mimariler
MobileNet, derinlemesine ayrılabilir evrişim.

Çıkarım motoru
Modeli cihazda çalıştıran küçük yazılım katmanı.

Hızlandırıcılar
NPU, DSP, Edge TPU: sinir ağına özel donanım.

Bellek bütçesi
Birkaç yüz KB SRAM ve flash: ağırlık ve ara sonuçlar.

Gecikme ve enerji
Doğruluk kadar önemli iki başarım ölçütü.

DERS

Tipik İş Akışı



Eğit	Küçült	Dönüştür	Cihaza yükle	Çalıştır ve ölç
sunucuda, tam duyarlık	nicemleme, budama	TFLite, ONNX	MCU, telefon, kamera	gecikme, enerji

Örnek: Anahtar sözcük algılayan küçük bir ağ bilgisayarda 32 bitlik kayan noktayla eğitilir, 8 bitlik tamsayıya nicemlenir ve birkaç yüz kilobayt belleği olan bir mikrodenetleyiciye yüklenir. Cihaz mikrofonu sürekli dinler, sesi buluta göndermez ve çok az güç harcar.

DERS

Derste Neler İşlenir?

1. Gömülü donanım: MCU, bellek ve enerji bütçesi
2. Sensör verisi toplama ve ön işleme
3. Küçük sinir ağları ve verimli mimariler
4. Eğitim sonrası nicemleme
5. Nicemlemeye duyarlı eğitim
6. Budama ve seyrek ağlar
7. Bilgi damıtma
8. Model dönüştürme: TensorFlow Lite ve ONNX
9. Mikrodenetleyicide çıkarım: TFLite Micro, CMSIS-NN
10. Hızlandırıcılar: NPU, DSP ve Edge TPU
11. Gecikme, bellek ve enerji ölçümü; MLPerf Tiny
12. Cihazda öğrenme ve uzaktan model güncelleme

Kazandırdığı beceriler

- TensorFlow Lite ile model dönüştürmek
- 8 bitlik nicemleme uygulamak
- Arduino'ya model yüklemek
- Edge Impulse ile uçtan uca proje
- gecikme, bellek ve enerji ölçmek

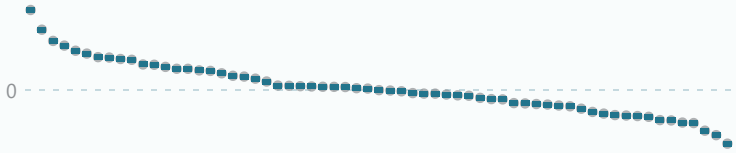
Tipik bir dönemin akışıdır; sıra ve kapsam üniversiteden üniversiteye değişir.

DENE

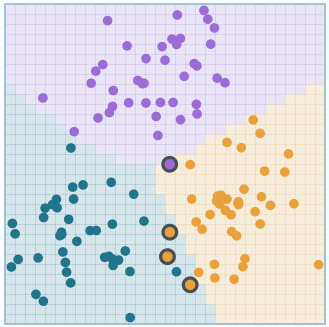
Modeli Mikrodenetleyiciye Sığdır

Bir motorun üstündeki titreşim sensörü arızayı kendisi tanımalı, ama işlemcinin belleği 512 KB. Ağırlıkları kaç bitle saklayacağınızı seçin; boyut, hız ve doğruluk birlikte değişir.

Ağırlıklar (büyükten küçüğe): gri gerçek, mavi 32 bit



Test verisi: doğruluk %95,0



Bellek: **1172 KB**

cihaz sınırı 512 KB

Gecikme: **120 ms**

Enerji: **4,8 mJ**

gecikme ve enerji tahminidir

Ağırlık başına bit: **32**

32 | 16 | 8 | 4 | 2

32 bit: model **1172 KB** → sığmaz
Doğruluk **%95,0** (32 bitte %95,0)
Tahmini gecikme 120 ms, enerji 4,8 mJ

● normal ● dengesizlik ● rulman arızası. Ağ burada eğitildi (63 ağırlık); boyut, aynı oranla 300 bin ağırlıklı gerçek model için.

Gömülü yapay zekâ modeli küçültmeyi işin parçası sayar: 8 bit **nicemleme** çoğu zaman doğruluğu korur, belleği dörtte bire indirir. 2 bitte ise karar sınırları dağılır.

KONUM

Ön Koşullar

Bu dersten önce bilinmesi yararlı olan dersler ve neden gerektikleri.

Ders	Neden gerekir
Derin Öğrenme	nicemleme, budama, küçük modeller
Mikroişlemciler	bellek, yazmaç, kesme ve çevre birimleri: cihazın kısıtlarını anlamak
C ile Programlama	gömülü yazılımın dili; işaretçiler ve sabit bellek yönetimi

KONUM

Bu Ders Sana Göre mi?

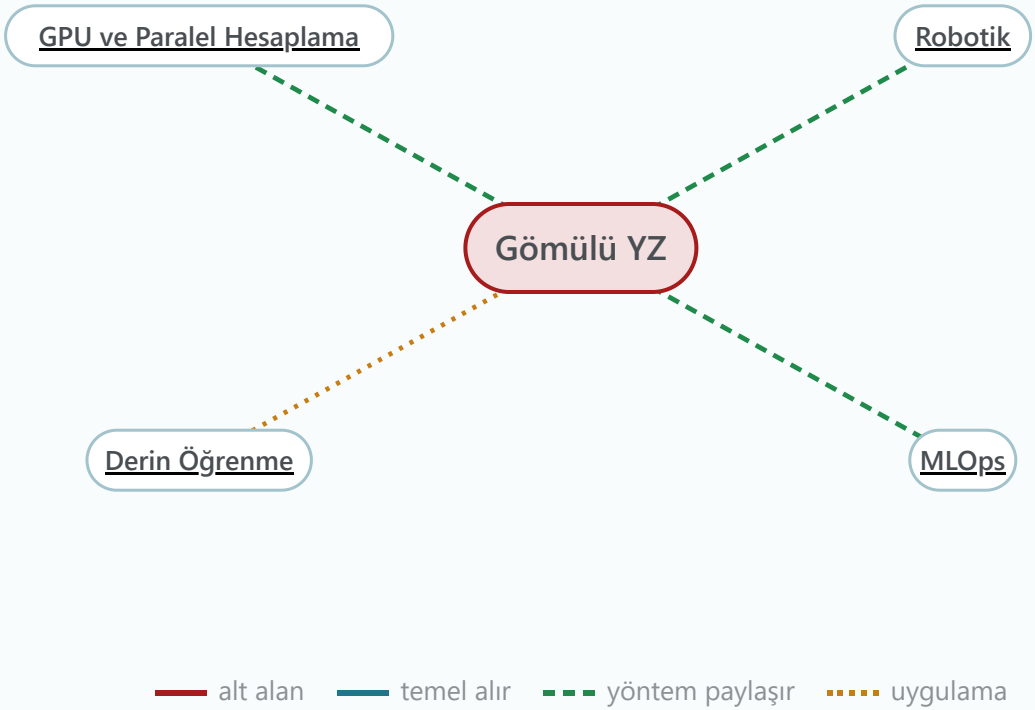
Kendine sor; "hayır" dediklerin için sağdaki sütun nereden başlayacağını gösterir.

Kendine sor	Hayırsa önce
Bir evrişimli ağı eğitip doğrulama doğruluğunu raporlayabiliyor musun?	Derin Öğrenme
C'de işaretçilerle ve dinamik bellek ayırmadan kod yazabiliyor musun?	C ile Programlama dersi
Bir mikrodenetleyiciye program yükleyip sensörden veri okuyabiliyor musun?	Mikroişlemciler dersi ve bir Arduino başlangıç seti
Bir sayının 8 bitlik tamsayıyla nasıl temsil edildiğini açıklayabiliyor musun?	Sayısal tasarımdaki ikili sayılar ve sabit nokta

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

KONUM

Dersin Komşuları



ÖRTÜŞME

Örtüştüğü Dersler

İlişki	Ders	Ortak konu
uygulama	Bu ders Derin Öğrenme yöntemlerini uygular	nicemleme, budama, küçük modeller
yöntem paylaşır	Robotik ve Robot Görüsü ile yöntem paylaşır	cihaz üzerinde gerçek zamanlı çıkarım
yöntem paylaşır	Makine Öğrenmesi Sistemleri ve MLOps ile yöntem paylaşır	modeli cihazlara dağıtma ve güncelleme
yöntem paylaşır	GPU Programlama ve Paralel Hesaplama ile yöntem paylaşır	donanım hızlandırıcılar

Ayrıştığı Noktalar (1/2)

Ölçüt	Gömülü YZ	Derin Öğrenme	GPU ve Paralel Hesaplama
Temel soru	Yapay zekâ modelleri küçük, düşük güçlü cihazlarda nasıl çalışır?	Çok katmanlı ağlar ham veriden kullanışlı temsilleri nasıl kendisi öğrenir?	Binlerce çekirdek aynı hesabı nasıl paylaşır?
Yaklaşım	kaynağı kısıtlı çıkarım	çok katmanlı temsil öğrenme	veri paralel hesaplama
Girdi	eğitilmiş model, cihaz kısıtları	ham veri: görüntü, metin, ses	paralelleştirilebilir algoritma
Çıktı	küçük ve hızlı model	öğrenilmiş temsil ve tahmin	hızlandırılmış program
Başarı ölçütü	gecikme, enerji, bellek	doğruluk, kayıp, ölçeklenme	hızlanma, bellek bant genişliği
Matematik yükü	orta	yüksek: türev, matris, olasılık	düşük-orta
Veri ihtiyacı	eğitim verisi zaten var	çok; GPU gerekir	gerekmez

- **Derin Öğrenme:** Derin öğrenme doğruluk için modeli büyütür; bu ders aynı modeli doğruluğu az kaybederek küçültür.
- **GPU ve Paralel Hesaplama:** GPU programlama büyük hızlandırıcılardan en yüksek başarımı alır; bu ders birkaç miliwatt ile çalışan cihazlarda çıkarım yapar.

AYRIŞMA

Ayrıştığı Noktalar (2/2)

Ölçüt	Gömülü YZ	MLOps	Robotik
Temel soru	Yapay zekâ modelleri küçük, düşük güçlü cihazlarda nasıl çalışır?	Bir model laboratuvardan üretime nasıl taşınır ve orada nasıl ayakta tutulur?	Bir makine fiziksel dünyayı nasıl algılar, konumunu nasıl bilir ve nasıl hareket eder?
Yaklaşım	kaynağı kısıtlı çıkarım	yazılım mühendisliği ve MÖ	algıla, planla, eyle döngüsü
Girdi	eğitilmiş model, cihaz kısıtları	eğitilmiş model, veri boru hattı	algılayıcı ölçümleri
Çıktı	küçük ve hızlı model	çalışan ve izlenen hizmet	motor komutları, yol
Başarı ölçütü	gecikme, enerji, bellek	gecikme, kesintisizlik, veri kayması	görev başarısı, güvenlik
Matematik yükü	orta	düşük	yüksek: dönüşümler, denetim
Veri ihtiyacı	eğitim verisi zaten var	akan üretim verisi	fiziksel deney ve benzetim

- **MLOps:** MLOps modeli sunucularda dağıtır ve izler; bu ders modeli bağlantısı olmayabilen cihazların içine yerleştirir.
- **Robotik:** Robotik algı, konum ve hareketi bütün olarak ele alır; bu ders robottaki modelin gerçek zamanda çalışmasını sağlar.

UYGULAMA

Uygulama Alanları

Uyandırma sözcüğü
Hoparlör ve telefonda sürekli dinleyen model.

Akıllı kamera
Görüntüyü buluta göndermeden nesne algılama.

Giyilebilir cihazlar
Saatte düşme, adım ve kalp ritmi analizi.

Kestirimci bakım
Motor titreşiminden arıza belirtisi yakalamak.

Telefonda YZ
Fotoğraf iyileştirme, çevrimdışı çeviri.

Tarım ve çevre
Pille çalışan sensörlerde ses ve zararlı tespiti.

UYGULAMA

Riskler ve Etik

Küçültmenin gizli bedeli

Nicemleme ve budamada ortalama doğruluk az düşer ama nadir sınıflar ve azınlık gruplar orantısız kayıp yaşayabilir.

Güncellenemeyen cihaz

Sahaya dağılmış binlerce cihazdaki hatalı ya da açığı olan model kolayca geri çağrılamaz.

Fark edilmeyen gözetim

Sürekli dinleyen mikrofon ve kameralar, veri cihazda kalsa bile çevredekileri habersizce izleyebilir.

Model çalınması

Cihazdaki model dosyası okunup kopyalanabilir; tersine mühendislikle düşmanca saldırılara da açık hâle gelir.

Ayrıntılı tartışma: Açıklanabilir Yapay Zekâ ve Yapay Zekâ Etiği.

UYGULAMA

Araçlar ve Örnek Veriler

Araçlar ve kütüphaneler

TensorFlow Lite | ONNX Runtime | ExecuTorch | Edge Impulse | CMSIS-NN | Arduino | NVIDIA Jetson

Tanınmış veri setleri

Speech Commands | Visual Wake Words | MNIST | CIFAR-10 | ImageNet | MLPerf Tiny

Başarı yalnız doğrulukla ölçülmez: aynı tabloya model boyutu, tepe bellek, gecikme ve enerji de yazılır.

UYGULAMA

Sık Karıştırılanlar

Yaygın düşünce	Doğrusu
Küçük cihazda model eğitilir.	Eğitim çoğunlukla sunucuda yapılır; cihazda yalnızca çıkarım çalışır.
Nicemleme doğruluğu mutlaka bozar.	8 bitlik nicemleme çoğu modelde doğruluğu çok az düşürür; nicemlemeye duyarlı eğitim kaybı daha da azaltır.
Daha az parametre, daha hızlı modeldir.	Hız bellek erişimine, işlem türüne ve donanım desteğine bağlıdır; parametre sayısı yetmez.
Gömülü YZ yalnız internet yoksa gerekir.	Gizlilik, gecikme ve enerji için bağlantı olsa bile veriyi cihazda işlemek tercih edilir.

SONRASI

Bu Dersten Sonra

Bu dersi temel alan ya da genişleten dersler

Bu ders çoğunlukla uç noktadır; öteki dersleri temel alır.

İlgili meslekler

Gömülü yazılım mühendisi

Uç YZ mühendisi

Makine öğrenmesi mühendisi

Donanım-yazılım tasarımcısı

IoT çözüm mimarı

Gömülü YZ, derin öğrenmeyi donanıma bağlar: buradan robotik, otonom araçlar ve nesnelerin interneti uygulamalarına geçilir. Donanımda derinleşmek isteyenler için hızlandırıcı tasarımı ve derleyiciler, yazılımda ise MLOps ile cihaz filolarının güncellenmesi sonraki adımlardır.

Kaynaklar

- Warden ve Situnayake, *TinyML*, O'Reilly, 2019.
- Sze, Chen, Yang ve Emer, *Efficient Processing of Deep Neural Networks*, Morgan & Claypool, 2020.
- Han, Mao ve Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", *ICLR*, 2016.
- Howard ve ark., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *arXiv:1704.04861*, 2017.
- Hinton, Vinyals ve Dean, "Distilling the Knowledge in a Neural Network", *arXiv:1503.02531*, 2015.

Son

Bölüm 36: Gömülü Yapay Zekâ

← Önceki bölüm · Alanın haritası · Sonraki bölüm →