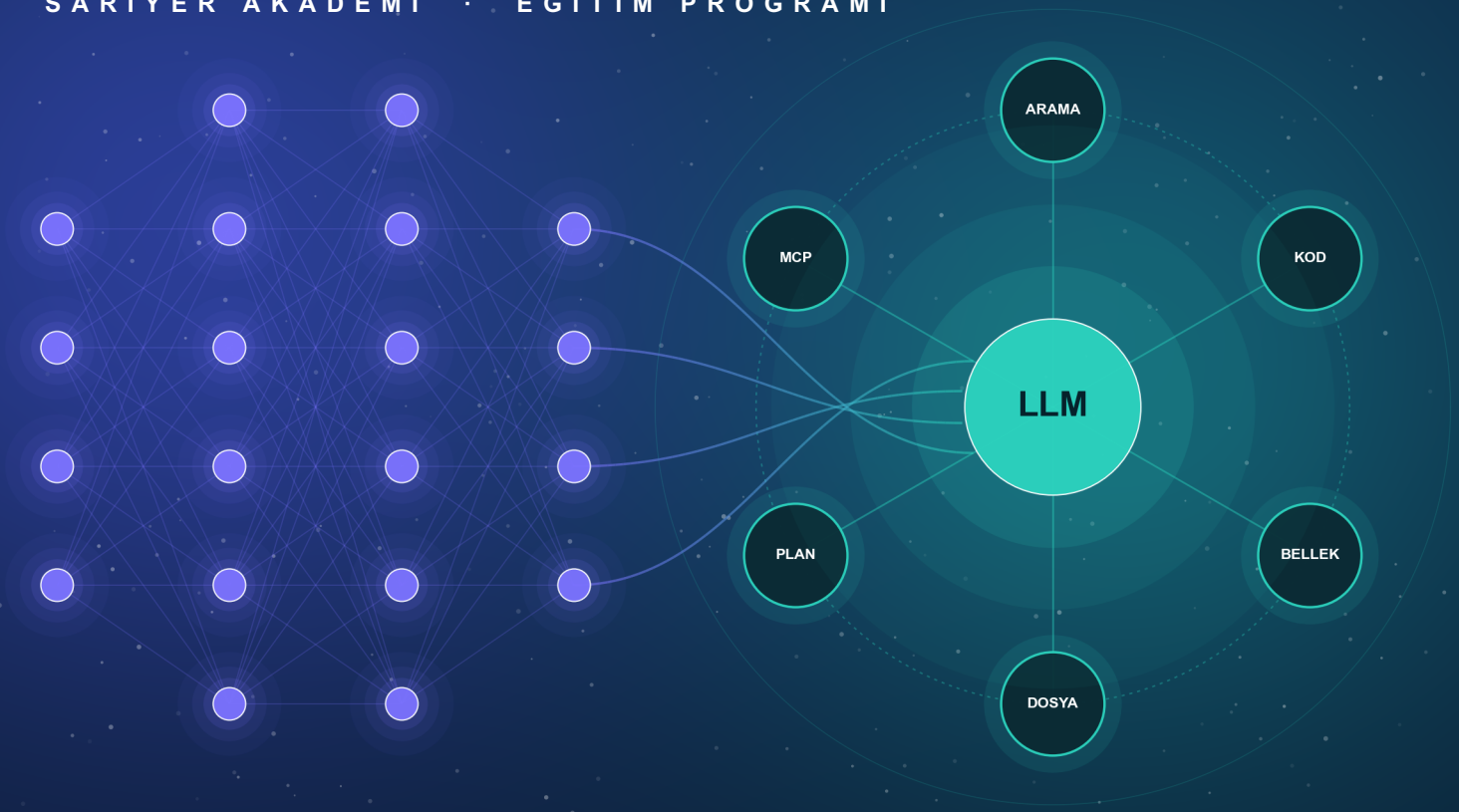
SARIYER AKADEMİ · EĞİTİM PROGRAMI



DERİN ÖĞRENME → AJANTİK MİMARİLER

2026 – 2027 EĞİTİM DÖNEMİ

Üretken Yapay Zeka

Derin Öğrenmeden Ajantik Mimarilere

Sinir ağlarından Transformer'a, LLM'lerden çoklu ajan sistemlerine uçtan uca, uygulamalı bir program.

15 oturum · 60 saat	Cumartesi · dönüşümlü	5 Ara 2026 – 17 Nis 2027
PROGRAM	09:00–13:00 / 14:00–18:00	TAKVİM

Eğitmen **Dr. Uzay Çetin**

01 · GENEL BAKIŞ

Eğitime Genel Bakış

Bu program, üretken yapay zekayı mümkün kılan iki katmanı öğretir: önce modelin kendisini (sinir ağlarından Transformer, BERT ve GPT'ye uzanan derin öğrenme hattı), ardından modelin etrafına kurulan sistemi (araç kullanan, hatırlayan, plan yapan ve birlikte çalışan ajanlar). Her oturumda bir konu anlatımı somut bir uygulamayla birleşir; program sonunda her katılımcı kendi ajan projesini sunar.

15	**60**	**2**	**1**
oturum, her biri 4 saat	saat uygulamalı ders	bölüm: derin öğrenme ve ajantik mimariler	bitirme projesi ve final sunumu

Program sonunda neler yapabileceksiniz

01
PyTorch ile bir sinir ağını sıfırdan eğitir, eğitim döngüsünün her adımını açıklarsınız.

02
CNN, RNN/LSTM ve Autoencoder ile görüntü, metin ve zaman serisi üzerinde modeller kurarsınız.

03
Transformer, BERT ve GPT mimarilerinin nasıl çalıştığını, LoRA ile bir modelin nasıl uyarlandığını anlarsınız.

04
LLM API'leri, araç kullanımı ve MCP ile bir modeli dış dünyaya bağlarsınız.

05
Sıfırdan bir ReAct ajanı yazar; RAG, bellek, planlama, yansıtma ve kod yürütmeyle genişletirsiniz.

06
Çoklu ajan sistemleri tasarlar ve ajanlarınızı ölçülebilir biçimde değerlendirirsiniz.

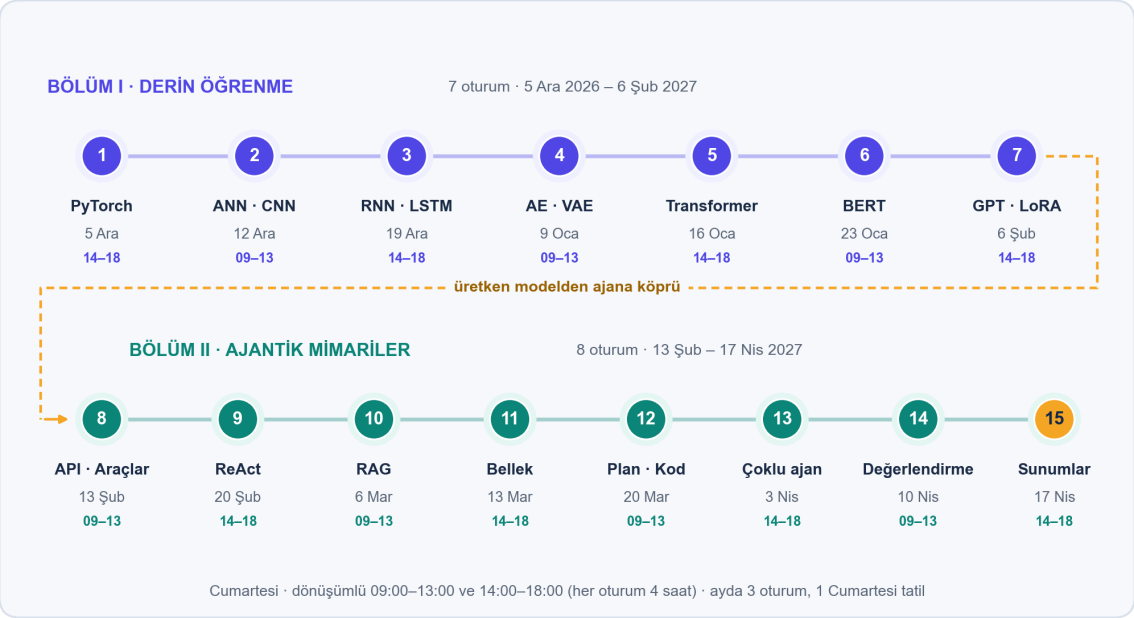
Format ve lojistik

TAKVİM	5 Aralık 2026 – 17 Nisan 2027, Cumartesi günleri. Ayda 3 oturum, ayda 1 Cumartesi tatil.
SAATLER	Dönüşümlü: 09:00–13:00 ve 14:00–18:00; her oturum 4 saat sürer. İlk oturum 5 Aralık'ta saat 14:00'te başlar.
YÖNTEM	Konu anlatımı, canlı kodlama ve proje. Her oturumda çalışan bir çıktı.
KİMLER İÇİN	Yapay zekayı uygulamalı öğrenmek isteyen gençler ve yetişkinler. Önerilen ön bilgi: temel Python ve lise matematiği; kendi dizüstü bilgisayarı (Ek C).
EĞİTMEN	Dr. Uzay Çetin

02 · YOL HARİTASI

Yol Haritası

Bölüm I modelin kendisini, Bölüm II modelin etrafında kurulan ajan sistemini öğretir.



Program bir bakışta *Başlıklara tıklayarak ilgili oturuma gidebilirsiniz.*

NO	TARİH	SAAT	OTURUM KONUSU	KAYNAK
BÖLÜM I · DERİN ÖĞRENME · 28 SAAT				
01	5 Aralık	14–18	**PyTorch'a Giriş: Tensörlerden Gradyan İnişe**	Önceki prog. · 8 Mart
02	12 Aralık	09–13	**Derin Öğrenme I: Yapay Sinir Ağları ve CNN**	Önceki prog. · 15 Mart
03	19 Aralık	14–18	**Derin Öğrenme II: RNN, LSTM ve Attention'a Giriş**	Önceki prog. · 5 Nisan
04	9 Ocak	09–13	**Autoencoder ve Üretken Modellere Giriş**	Önceki prog. · 5 Nisan
05	16 Ocak	14–18	**Uçtan Uca Transformer**	Önceki prog. · 12 Nisan
06	23 Ocak	09–13	**BERT: Ön Eğitimli Modeller ve Transfer Öğrenme**	Önceki prog. · 10 Mayıs
07	6 Şubat	14–18	**GPT ve İnce Ayar: Üretken Dil Modelleri**	Önceki prog. · 24 Mayıs
BÖLÜM II · AJANTİK MİMARİLER · 32 SAAT				
08	13 Şubat	09–13	**LLM API'leri ve Araç Kullanımı**	ch02 · ch03
09	20 Şubat	14–18	**Sıfırdan ReAct Ajanı**	ch04
10	6 Mart	09–13	**RAG, Dosya Araçları ve Callback'ler**	ch05
11	13 Mart	14–18	**Bellek Sistemleri**	ch06
12	20 Mart	09–13	**Planlama, Yansıtma ve Kod Yürütme**	ch07 · ch08
13	3 Nisan	14–18	**Çoklu Ajan Sistemleri**	ch09
14	10 Nisan	09–13	**Değerlendirme, İzleme ve Bitirme Atölyesi**	ch10
15	17 Nisan	14–18	**Bitirme Projesi Sunumları ve Kapanış**	Önceki prog. · 19 Haziran

03 · TAKVİM

Takvim

Eğitim 5 Aralık 2026 Cumartesi günü başlar. Her ay 3 Cumartesi ders yapılır, 1 Cumartesi tatil bırakılır. Her oturum 4 saat sürer ve dönüşümlü olarak 09:00–13:00 ile 14:00–18:00 saatlerinde yapılır; toplam 15 oturum Nisan 2027'de tamamlanır.

AY	1. CUMARTESİ	2. CUMARTESİ	3. CUMARTESİ	4. CUMARTESİ	5. CUMARTESİ
Aralık 2026	**5** Oturum 1 · PyTorch'a giriş · 14:00–18:00	**12** Oturum 2 · ANN ve CNN · 09:00–13:00	**19** Oturum 3 · RNN · LSTM · 14:00–18:00	**26** Tatil · ders yok	
Ocak 2027	**2** Tatil · ders yok	**9** Oturum 4 · Autoencoder · VAE · 09:00–13:00	**16** Oturum 5 · Transformer · 14:00–18:00	**23** Oturum 6 · BERT · 09:00–13:00	**30** Yedek gün · telafi için
Şubat 2027	**6** Oturum 7 · GPT · ince ayar · 14:00–18:00	**13** Oturum 8 · API ve araçlar · 09:00–13:00	**20** Oturum 9 · ReAct ajanı · 14:00–18:00	**27** Tatil · ders yok	
Mart 2027	**6** Oturum 10 · RAG ve dosya · 09:00–13:00	**13** Oturum 11 · Bellek · 14:00–18:00	**20** Oturum 12 · Plan · kod · 09:00–13:00	**27** Tatil · ders yok	
Nisan 2027	**3** Oturum 13 · Çoklu ajan · 14:00–18:00	**10** Oturum 14 · Değerlendirme · 09:00–13:00	**17** Oturum 15 · Sunumlar · 14:00–18:00	**24** Tatil · ders yok	

Bölüm I · Derin öğrenme · Bölüm II · Ajantik mimariler · Yedek gün · Tatil

TAKVİM NOTLARI

- Aralık, Şubat, Mart ve Nisan aylarında ayın son Cumartesisi tatildir. 24 Nisan, 23 Nisan Cuma ile birleşen uzun hafta sonuna denk gelir.
- Ocak ayında 5 Cumartesi vardır: 2 Ocak yılbaşı haftası nedeniyle tatil, 30 Ocak ise yedek/telafi günüdür. Bölüm I ile II arasındaki bu boşluk, gerekirse geri kalan konuları toparlamak için kullanılabilir.
- Ramazan Bayramı Diyanet takvimine göre 9–11 Mart 2027'dir (Salı–Perşembe) ve Cumartesi günlerine denk gelmez. 13 Mart oturumu bayramın hemen ertesine düştüğü için katılım düşük olursa 27 Mart tatil günü telafi için kullanılabilir.

04 · SAAT DÜZENİ

Saat Düzeni ve Oturum Akışı

Her oturum 4 saat sürer ve iki farklı saat diliminde, dönüşümlü olarak yapılır: tek numaralı oturumlar öğleden sonra, çift numaralı oturumlar sabah. İlk oturum 5 Aralık'ta saat 14:00'te başlar.

ÖĞLEDEN SONRA OTURUMLARI

14:00–18:00

8 oturum · 32 saat

1 · 5 Ara 3 · 19 Ara 5 · 16 Oca 7 · 6 Şub
9 · 20 Şub 11 · 13 Mar 13 · 3 Nis 15 · 17 Nis

SABAH OTURUMLARI

09:00–13:00

7 oturum · 28 saat

2 · 12 Ara 4 · 9 Oca 6 · 23 Oca 8 · 13 Şub
10 · 6 Mar 12 · 20 Mar 14 · 10 Nis

Öğleden sonra oturumunun akışı *14:00–18:00*

14:00	14:15	15:30	15:45	17:30
Isınma	**Kavram bloğu**	**Mola**	**Uygulama bloğu**	**Demo ve ödev**
Geçen oturumdan kısa tekrar ve sorular	Konu anlatımı ve canlı gösterim	Kahve ve sohbet	Canlı kodlama ve proje çalışması	Gösterim, tartışma, ödev
15 dk	**75 dk**	**15 dk**	**105 dk**	**30 dk**

Sabah oturumunun akışı *09:00–13:00*

09:00	09:15	10:30	10:45	12:30
Isınma	**Kavram bloğu**	**Mola**	**Uygulama bloğu**	**Demo ve ödev**
Geçen oturumdan kısa tekrar ve sorular	Konu anlatımı ve canlı gösterim	Kahve ve sohbet	Canlı kodlama ve proje çalışması	Gösterim, tartışma, ödev
15 dk	**75 dk**	**15 dk**	**105 dk**	**30 dk**

Bölüm I	Bölüm II	Toplam
28 saat	**32 saat**	**60 saat**

DÖNÜŞÜMLÜ DÜZEN

İki dilimde de oturum akışı aynıdır; yalnızca saatler değişir. Her oturum 75 dakikalık kavram ve 105 dakikalık uygulama bloğu içerir.

05 · KAYNAKLAR

Dersin İki Ana Kaynağı

Birinci yarı birinci kaynaktan, ikinci yarı ikinci kaynaktan seçilen kısımlarla kurulmuştur.

KAYNAK 1 · BÖLÜM I'İN OMURGASI

Yapay Zeka Special Force Eğitimi

Önceki program · Mart–Haziran 2025 · **uzay00.github.io/kahve/specialforce.html**

KISACA

Düzenleyen ve ekip Ka|Ve Eğitim (Karmaşık Sistemler ve Veri Bilimi Topluluğu); ortaklar Sarıyer Belediyesi, Sarıyer Akademi, Summarify ARGE. Organizatör Dr. Uzay Çetin, eğitmen Yunus Emre Gündoğmuş.

Format 12 hafta, Cumartesi 10:30–13:35, yüz yüze; 20 kişilik kontenjan, ücretsiz. Tüm derslere katılan ve proje sunan katılımcılara sertifika.

Yaklaşım Proje odaklı ve disiplinler arası: her oturumda konu anlatımı somut bir projeyle birleşir. Amaç, birlikte çalışılacak yol arkadaşları kazanmaktır.

Kapsam PyTorch, ANN/CNN, RNN/LSTM/Attention, Transformer, BERT, GPT finetuning (Falcon); FastAPI, Docker/Kubernetes, Kafka ve Jenkins ile üretime alma.

BU PROGRAMA KATKISI

- Bölüm I'in 7 oturumu doğrudan bu programdan seçildi (Ek A).
- Korunan projeler: dolar kuru tahmini, anomali tespiti, duygu analizi, BERT sınıflandırma, GPT ince ayarı.
- Çıkarılan: dağıtım/altyapı konuları ve etkinlik günleri. Güncellenen: Falcon yerine açık modeller ve LoRA; Autoencoder/VAE eklendi.

KAYNAK 2 · BÖLÜM II'NİN OMURGASI

Build an AI Agent (From Scratch)

Jungjun Hur ve Younghee Song · Manning Publications · **github.com/shangrilar/ai-agent-from-scratch**

KISACA

Kitap 336 sayfa, 10 bölüm ve bir ek; ISBN 9781633434615; Manning sayfasında yayın tarihi Temmuz 2026. Hedef okur: biraz makine öğrenmesi bilen Python geliştiricileri.

Yaklaşım Soyut ajan kütüphanelerine bağımlı kalmadan, Python ile sıfırdan bir ajan; GAIA kıyaslamasındaki çok adımlı görevleri çözen bir araştırma ajanı adım adım inşa edilir.

Konular Durumsuz API'ler ve Pydantic, araç çağırma ve MCP, context engineering, RAG, bellek, planlama ve yansıtma, CodeAct/sandbox, A2A ile çoklu ajan, LLM-as-judge değerlendirme ve izleme.

Kod deposu MIT lisanslı; scratch_agents paketi ve CH02–CH10 notebook'ları; Python 3.13+ ve uv. Testler harici servisleri simüle eder, örnekler gerçek (ücretli) API istekleri yapar.

BU PROGRAMA KATKISI

- Bölüm II'nin tamamı: CH02–CH10, oturum 8–14 (Ek B). Her oturumda ilgili notebook ve modül.
- Kavramlar çalışan kodla öğretilir; kitap kavramsal arka plan olarak okunabilir.
- Kısaltılan: CH02'nin 100 isteklik örneği, çok modelli GAIA kıyası, sandbox araç taşıma ayrıntıları.

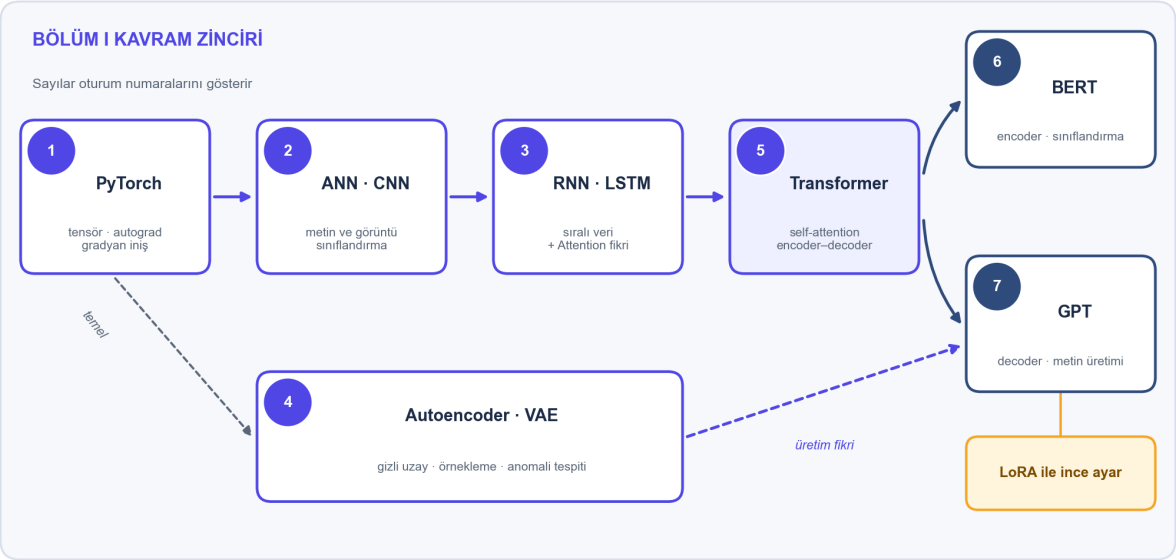
BÖLÜM I

Derin Öğrenme

Üretken yapay zekanın temeli: sinir ağlarından GPT'ye

7 oturum · 28 saat · 5 Aralık 2026 – 6 Şubat 2027

Bir model veriden nasıl öğrenir ve öğrenme nasıl 'üretmeye' dönüşür?



Şekil: Bölüm I boyunca izleyeceğimiz kavram zinciri.

NE YAPACAĞIZ?

PyTorch ile sıfırdan eğitim döngüsü; ANN, CNN ve RNN/LSTM; Autoencoder ve VAE ile üretken modellemeye giriş; Transformer, BERT ve GPT mimarileri; LoRA ile ince ayar.

NASIL ÇALIŞACAĞIZ?

Her oturumda önce kavram, sonra uygulama bloğu. Her oturum 4 saat sürer ve dönüşümlü olarak 14:00–18:00 ile 09:00–13:00 saatlerinde yapılır. Not defterleri Google Colab'da çalışır, kurulum gerekmez.

BÖLÜM SONUNDA

Bir modelin mimarisini, eğitimini ve değerlendirmesini kendi başınıza yürütebilir; Bölüm II'de kullanacağımız büyük dil modellerinin içini bilir hale gelirsiniz.

ÖNCEKİ PROGRAMLA İLİŞKİ

Bu bölüm, önceki programın (Yapay Zeka Special Force Eğitimi) derin öğrenme oturumlarından seçilen içeriklerden oluşur. İçerik üretken yapay zekaya odaklanacak şekilde yeniden sıralanmış, Autoencoder/VAE ile üretken modellemeye giriş eklenmiştir. Ayrıntılar Ek A'dadır.

BÖLÜM I · DERİN ÖĞRENME 1 / 15

OTURUM 01 · 5 ARALIK 2026 · CUMARTESİ 14:00–18:00 · 4 SAAT

PyTorch'a Giriş: Tensörlerden Gradyan İnişe

Üretken yapay zekanın yazılım ve matematik zemini

KAVRAM BLOĞU · 75 DK

- Tanışma, program ve takvim tanıtımı; Colab ortamının kontrolü
- Üretken yapay zeka manzarası: sınıflandırma ile üretim arasındaki fark ve programın yol haritası
- Tensörler, CPU/GPU ve otomatik türev (autograd) mantığı
- Kayıp fonksiyonu, gradyan iniş algoritması ve öğrenme oranının rolü
- nn.Module, optimizer ve bir eğitim döngüsünün anatomisi
- Eğitim / doğrulama / test ayrımı ve aşırı öğrenme (overfitting)

UYGULAMA BLOĞU · 105 DK

Gradyan inişi elle yazalım

- Doğrusal regresyonu önce NumPy ile, sonra PyTorch autograd ile eğitmek
- Aynı problemi nn.Module ve optimizer ile yeniden yazıp sonuçları karşılaştırmak
- Eğitilmiş modeli Streamlit ile etkileşimli bir web arayüzüne dönüştürmek

OTURUM SONUNDA KATILIMCI

- Bir eğitim döngüsünü satır satır açıklayabilir
- Öğrenme oranı ve iterasyon sayısının kayıp eğrisine etkisini yorumlayabilir
- Basit bir modeli tarayıcıda çalışan bir demoya dönüştürebilir

ARAÇLAR

PyTorch **NumPy** **Google Colab** **Streamlit**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Üç farklı öğrenme oranıyla kayıp eğrilerini çizin ve gözleminizi birkaç cümleyle yazın.

KAYNAK Önceki program · 8 Mart: PyTorch'a giriş, gradyan iniş, Streamlit. Bulut (AWS/GCP/Azure) dağıtımı bu sürümde kapsam dışıdır.

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Gradyan inişi elle yazalım**	Demo ve ödev

DÜŞÜNME SORUSU

Öğrenme oranını 10 kat büyütürsek kayıp eğrisi nasıl değişir? Bunun nedeni gradyanın yapısında nerede yatıyor?

BÖLÜM I · DERİN ÖĞRENME 2 / 15

OTURUM 02 · 12 ARALIK 2026 · CUMARTESİ 09:00–13:00 · 4 SAAT

Derin Öğrenme I: Yapay Sinir Ağları ve CNN

Metin ve görüntü için ilk sınıflandırıcılarımız

KAVRAM BLOĞU · 75 DK

- Perceptron'dan çok katmanlı ağlara (ANN): aktivasyon fonksiyonları ve geri yayılım
- Metni sayıya çevirmek: tokenizasyon ve embedding fikri (5. oturumun ön hazırlığı)
- Evrişim, filtre ve havuzlama: CNN mimarisi görüntüde neden çalışır?
- Düzenlileştirme: dropout, batch normalization, veri artırma
- Sonuçları okumak: karışıklık matrisi, precision, recall, F1

UYGULAMA BLOĞU · 105 DK

Duygu analizi, ironi tespiti ve görüntü sınıflandırma

- Kısa metinlerde duygu analizi ve ironi tespiti için embedding + ANN sınıflandırıcı
- Küçük bir görüntü veri kümesinde CNN eğitmek; veri artırmanın etkisini ölçmek
- Hata analizi: modelin nerede ve neden yanıldığını örnekler üzerinde incelemek

OTURUM SONUNDA KATILIMCI

- ANN ve CNN ile iki farklı veri türünde sınıflandırıcı kurabilir
- Doğruluk yanında F1 ve karışıklık matrisini yorumlayabilir
- Aşırı öğrenmeyi tanır, uygun düzenlileştirme yöntemini seçebilir

ARAÇLAR

PyTorch · torchvision · scikit-learn · Colab

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

CNN'de filtre ve katman sayısını değiştirin; parametre sayısı – doğruluk tablosunu doldurun.

KAYNAK Önceki program · 15 Mart: Derin Öğrenme I (ANN ve CNN); ironi tespiti, duygu analizi, görüntü sınıflandırma.

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**Duygu analizi, ironi tespiti ve görüntü sınıflandırma**	Demo ve ödev

DÜŞÜNME SORUSU

Görüntüdeki nesne 20 piksel kaysa ANN mi yoksa CNN mi daha az etkilenir? Neden?

BÖLÜM I · DERİN ÖĞRENME 3 / 15

OTURUM 03 · 19 ARALIK 2026 · CUMARTESİ 14:00–18:00 · 4 SAAT

Derin Öğrenme II: RNN, LSTM ve Attention'a Giriş

Sıralı veriyle çalışmak ve Transformer'a giden yol

KAVRAM BLOĞU · 75 DK

- Sıralı veri ve bellek ihtiyacı: RNN'in çalışma biçimi
- Kaybolan ve patlayan gradyan problemi; LSTM ve GRU kapıları
- Zaman serilerinde pencereleme, normalizasyon ve veri sızıntısından kaçınma
- Seq2seq ve bilgi darboğazı: Attention fikrinin doğuşu
- Sorgu (query), anahtar (key), değer (value) sezgisi

UYGULAMA BLOĞU · 105 DK

Dolar kuru tahmini

- Geçmiş döviz kuru verisiyle LSTM tahmin modeli kurmak
- Naif bir temel modelle (baseline) kıyaslayıp gerçekçi bir değerlendirme yapmak
- LSTM çıktısına basit bir attention katmanı ekleyip etkisini gözlemlemek

OTURUM SONUNDA KATILIMCI

- RNN ve LSTM'in belleği nasıl tuttuğunu anlatabilir
- Zaman serisi için doğru eğitim/test bölmesi yapabilir
- Attention'ın hangi problemi çözdüğünü açıklayabilir

ARAÇLAR

PyTorch **pandas** **matplotlib**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Pencere uzunluğunu (7, 30, 90 gün) değiştirip hata metriklerini karşılaştırın.

KAYNAK Önceki program · 5 Nisan: Derin Öğrenme II (RNN, LSTM, Attention); dolar kuru tahmini. Uygulama eğitim amaçlıdır, yatırım tavsiyesi değildir.

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Dolar kuru tahmini**	Demo ve ödev

DÜŞÜNME SORUSU

Modeliniz naif tahminden (dünün değeri) daha iyi değilse ne öğrenmiş olur? Finansal tahmin bu yüzden neden zordur?

BÖLÜM I · DERİN ÖĞRENME 4 / 15

OTURUM 04 · 9 OCAK 2027 · CUMARTESİ 09:00–13:00 · 4 SAAT

Autoencoder ve Üretken Modellere Giriş

Veriyi sıkıştırmayı öğrenen ağlardan yeni veri üretmeye

KAVRAM BLOĞU · 75 DK

- Autoencoder: encoder–decoder yapısı, darboğaz ve gizli uzay (latent space)
- Yeniden oluşturma hatasıyla anomali tespiti
- Ayırt edici ve üretken modeller: sınıflandırmak ile üretmek arasındaki fark
- Variational Autoencoder (VAE): olasılıksal gizli uzay, KL terimi, yeniden parametrelendirme
- Üretken model ailesi haritası: VAE, GAN, difüzyon, otoregresif modeller (kavramsal)

UYGULAMA BLOĞU · 105 DK

Anomali tespiti ve sentetik veri üretimi

- Autoencoder ile anomali tespiti; eşik seçimi ve ROC ile değerlendirme
- VAE ile örnek üretmek ve gizli uzayda gezinerek geçişleri izlemek
- Sentetik anomali üretimi: üretilen örnekler dedektörü zorluyor mu?

OTURUM SONUNDA KATILIMCI

- Gizli uzay kavramını örneklerle açıklayabilir
- Yeniden oluşturma hatasıyla anomali dedektörü kurabilir
- Üretmenin 'bir dağılımdan örnekleme' olduğunu anlatabilir

ARAÇLAR

PyTorch scikit-learn matplotlib

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Gizli uzay boyutunu 2, 8 ve 32 yaparak yeniden oluşturma kalitesini ve üretim çeşitliliğini karşılaştırın.

KAYNAK Önceki program · 5 Nisan: Auto Encoder ile anomali tespiti; proje fikri: sentetik anomali üretimi.

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**Anomali tespiti ve sentetik veri üretimi**	Demo ve ödev

DÜŞÜNME SORUSU

Gizli uzay çok küçük ya da çok büyük olursa ne bozulur? 'İyi bir üretken model' neye göre ölçülür?

BÖLÜM I · DERİN ÖĞRENME 5 / 15

OTURUM 05 · 16 OCAK 2027 · CUMARTESİ 14:00–18:00 · 4 SAAT

Uçtan Uca Transformer

Büyük dil modellerinin mimari çekirdeği

KAVRAM BLOĞU · 75 DK

- RNN'in sınırları: ardışık hesap ve uzun mesafe bağımlılıkları
- Self-attention: Q, K, V hesabı, ölçekleme, softmax ve maskeleme
- Multi-head attention, pozisyonel kodlama, artık bağlantılar ve LayerNorm
- Encoder–decoder yapısı ve tek bir Transformer bloğunun anatomisi
- Tokenizasyon: kelime, karakter ve alt-kelime (BPE) yaklaşımları

UYGULAMA BLOĞU · 105 DK

Self-attention'ı sıfırdan yazmak

- Self-attention katmanını PyTorch'ta sıfırdan kodlamak, ağırlıkları görselleştirmek
- Minik bir Transformer ile doğal dil işleme görevi çözmek
- Türkçe metni alt-kelime tokenizer'ıyla parçalamak; token sayısının maliyete etkisini görmek

OTURUM SONUNDA KATILIMCI

- Self-attention hesabını matris düzeyinde anlatabilir
- Bir Transformer bloğunun bileşenlerini ve görevlerini sayabilir
- Tokenizasyonun model ve maliyet üzerindeki etkisini yorumlayabilir

ARAÇLAR

PyTorch **Hugging Face tokenizers** **matplotlib**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Attention ağırlıklarını iki farklı cümle için görselleştirip hangi kelimelere baktığını yorumlayın.

KAYNAK Önceki program · 12 Nisan: Uçtan uca Transformer teknolojisi; Transformer'ların doğal dil işlemede kullanımı.

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Self-attention'ı sıfırdan yazmak**	Demo ve ödev

DÜŞÜNME SORUSU

Attention matrisinin boyutu dizi uzunluğunun karesiyle büyür. Bu, 100 bin tokenlık bir bağlam penceresi için ne anlama gelir?

BÖLÜM I · DERİN ÖĞRENME 6 / 15

OTURUM 06 · 23 OCAK 2027 · CUMARTESİ 09:00–13:00 · 4 SAAT

BERT: Ön Eğitimli Modeller ve Transfer Öğrenme

Hazır bir modeli kendi göreve uyarlamak ve servise dönüştürmek

KAVRAM BLOĞU · 75 DK

- Ön eğitim (pre-training) ve ince ayar (fine-tuning) paradigması
- BERT: çift yönlü encoder, maskeli dil modelleme ve sınıflandırma başlığı
- Hugging Face ekosistemi: Hub, tokenizer, model ve Trainer
- Değerlendirme: F1, sınıf dengesizliği ve hata analizi
- Çıkarım (inference) servisi: FastAPI ile modeli API olarak sunmak

UYGULAMA BLOĞU · 105 DK

BERT ile sınıflandırma ve API'ye dönüştürme

- Ön eğitimli bir BERT modelini metin sınıflandırma için ince ayarlamak
- Sonucu 2. oturumdaki ANN sınıflandırıcıyla kıyaslamak
- Modeli FastAPI ile REST servisine çevirip Streamlit arayüzünden çağırmak

OTURUM SONUNDA KATILIMCI

- Hugging Face ile bir modeli ince ayarlayıp değerlendirebilir
- Transfer öğrenmenin sıfırdan eğitime göre kazancını ölçebilir
- Eğitilmiş bir modeli HTTP servisi olarak yayınlayabilir

ARAÇLAR

Hugging Face Transformers PyTorch FastAPI Streamlit

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Servise toplu istek (batch) uç noktası ekleyin ve gecikmeyi ölçün.

KAYNAK Önceki program · 10 Mayıs: Uçtan uca BERT (BERT ile sınıflandırma) ve 22 Mart: FastAPI ile servis haline getirme.

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**BERT ile sınıflandırma ve API'ye dönüştürme**	Demo ve ödev

DÜŞÜNME SORUSU

Sınıflandırma için BERT (encoder), metin üretimi için GPT (decoder) neden daha doğal bir seçimdir?

BÖLÜM I · DERİN ÖĞRENME 7 / 15

OTURUM 07 · 6 ŞUBAT 2027 · CUMARTESİ 14:00–18:00 · 4 SAAT

GPT ve İnce Ayar: Üretken Dil Modelleri

Bölüm I'in kapanışı ve ajanlara köprü

KAVRAM BLOĞU · 75 DK

- Decoder-only mimari, nedensel maskeleme ve sonraki-token tahmini
- Metin üretimi: greedy, sıcaklık (temperature), top-k ve top-p örnekleme
- Ön eğitimden sohbet modeline: denetimli ince ayar (SFT) ve tercih optimizasyonuna genel bakış
- Parametre-verimli ince ayar: LoRA ve nicemleme (QLoRA)
- İnce ayar mı, istem mühendisliği ve RAG mı? Bir karar çerçevesi

UYGULAMA BLOĞU · 105 DK

Özel chatbot için ince ayar

- Açık ağırlıklı küçük bir dil modelini LoRA ile sohbet verisinde ince ayarlamak
- İnce ayar öncesi ve sonrası çıktıları aynı istemlerle karşılaştırmak
- Bitirme projesi duyurusu: konu havuzu ve takım kurulumu

OTURUM SONUNDA KATILIMCI

- Bir dil modelinin metni nasıl üretip örneklediğini açıklayabilir
- LoRA ile sınırlı kaynakla ince ayar yapabilir
- İnce ayar, RAG ve istem seçimini bir ihtiyaç için gerekçelendirebilir

ARAÇLAR

Hugging Face Transformers · PEFT (LoRA) · Colab GPU

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Bitirme projesi için ilgi alanınızı ve takım tercihinizi 3–4 cümleyle yazın.

KAYNAK Önceki program · 24 Mayıs: GPT teknolojisi ve GPT finetuning (Falcon). Bu sürümde güncel açık ağırlıklı modeller ve LoRA kullanılır.

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Özel chatbot için ince ayar**	Demo ve ödev

DÜŞÜNME SORUSU

Kurum içi dokümanlara dayalı bir soru-cevap sistemi için ince ayar mı, RAG mi? Hangi koşulda tersi doğru olur?

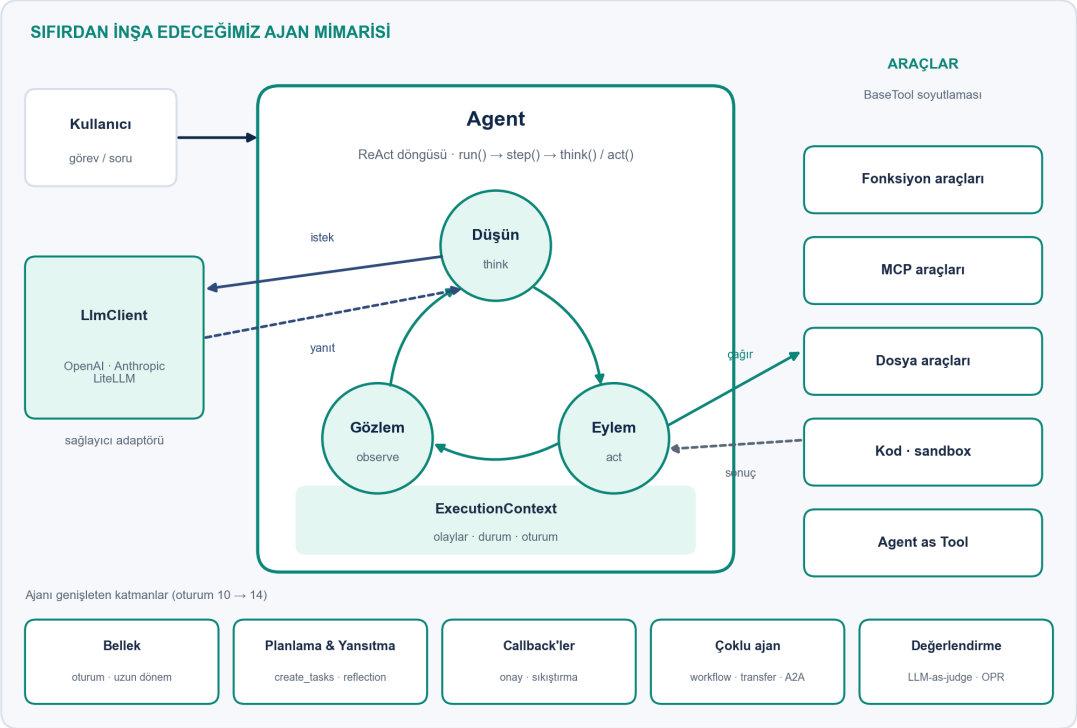
BÖLÜM II

Ajantik Mimariler

Araç kullanan, hatırlayan, planlayan ve birlikte çalışan yapay zeka sistemleri

8 oturum · 32 saat · 13 Şubat – 17 Nisan 2027

Bir dil modelini 'ajan' yapan nedir, ajanı nasıl ölçülebilir kılarız?



Şekil: Bölüm II boyunca adım adım inşa edeceğimiz ajan mimarisi.

NE YAPACAĞIZ?

API ve araç kullanımından başlayıp sıfırdan bir ReAct ajanı yazacak; ona RAG, bellek, planlama, kod yürütme ve çoklu ajan iş birliği ekleyeceğiz.

NASIL ÇALIŞACAĞIZ?

Her oturumda önce kavram, sonra uygulama bloğu. Her oturum 4 saat sürer ve dönüşümlü olarak 14:00–18:00 ile 09:00–13:00 saatlerinde yapılır. Örnekler gerçek API istekleri yapar.

BÖLÜM SONUNDA

Kendi ajanınızı tasarlar, genişletir ve ölçülebilir kriterlerle değerlendirirsiniz; bitirme projeniz bu çerçevede şekillenir.

Kaynak Bu bölüm “Build an AI Agent (From Scratch)” kod deposuna dayanır; bilgi 05 · Kaynaklar'da, depo haritası Ek B'dedir.

BÖLÜM II · AJANTİK MİMARİLER 8 / 15

OTURUM 08 · 13 ŞUBAT 2027 · CUMARTESİ 09:00–13:00 · 4 SAAT

LLM API'leri ve Araç Kullanımı

Modelden ajana ilk adım: API, yapılandırılmış çıktı, function calling ve MCP

KAVRAM BLOĞU · 75 DK

- LLM API'lerinin anatomisi: OpenAI ve Anthropic arayüzleri, LiteLLM ile tek arayüz
- API'nin durumsuz (stateless) olması ve konuşma geçmişini yönetmek
- Pydantic ile yapılandırılmış çıktı (structured output)
- Araç kullanımı: araç tanımı (JSON şeması), modelin araç çağırması, bizim çalıştırmamız
- İyi araç tasarımı ilkeleri; web arama aracını uçtan uca kurmak
- MCP (Model Context Protocol): araçları standartlaştırmak, istemci ve sunucu

UYGULAMA BLOĞU · 105 DK

İlk araç kullanan model

- Aynı isteği OpenAI ve Anthropic API'lerinde, sonra LiteLLM ile çalıştırmak
- Pydantic şemasıyla metinden yapılandırılmış veri çıkarmak
- Python fonksiyonunu araç tanımına çevirip model–araç döngüsünü elle yürütmek
- Kısa bir MCP sunucusu yazıp bir istemciden çağırmak
- Eşzamansız (async) ve hız sınırlı istekler: kısa demo

OTURUM SONUNDA KATILIMCI

- Bir LLM API'sini durumsuz yapısını bilerek doğru kullanabilir
- Bir Python fonksiyonunu modelin çağırabileceği araca dönüştürebilir
- MCP'nin hangi probleme çözüm olduğunu açıklayabilir

ARAÇLAR

OpenAI API **Anthropic API** **LiteLLM** **Pydantic** **MCP** **Tavily**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Kendi seçeceğiniz bir API'yi (hava durumu, döviz vb.) araç olarak ekleyin.

KAYNAK DEPO `notebooks/ch02 · ch03 — scratch_agents/tools/helpers.py · calculator.py · search.py · eval/gaia.py`

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**İlk araç kullanan model**	Demo ve ödev

DÜŞÜNME SORUSU

Model bir aracı 'çağırdığında' gerçekte ne olur? Aracı çalıştıran kim, sonucu modele geri veren kim?

BÖLÜM II · AJANTİK MİMARİLER 9 / 15

OTURUM 09 · 20 ŞUBAT 2027 · CUMARTESİ 14:00–18:00 · 4 SAAT

Sıfırdan ReAct Ajanı

Düşün – eylem yap – gözlemle döngüsünü kodlamak

KAVRAM BLOĞU · 75 DK

- ReAct: düşün–eylem–gözlem döngüsü; metin ayrıştırmadan yerel araç çağırmaya
- Ajan mimarisi ve bilgi akışı: merkezî depo olarak ExecutionContext
- Araç soyutlaması: BaseTool, FunctionTool ve MCP araçlarının entegrasyonu
- İletişim katmanı: LlmRequest, LlmResponse ve sağlayıcı adaptörü LlmClient
- Agent sınıfı: run(), step(), think(), act(); maksimum adım sayısı
- Araçları çıktı biçimlendirici olarak kullanıp yapılandırılmış çıktı almak

UYGULAMA BLOĞU · 105 DK

Ajanın çekirdeğini adım adım yazmak

- types → context → tools → llm → agent sırasıyla çekirdeği inşa etmek
- Ajanı web arama ve hesap makinesi araçlarıyla çalıştırmak
- GAIA kıyas testinden örnek sorular çözdürmek; çalıştırma izlerini (execution trace) okuyarak hata ayıklamak

OTURUM SONUNDA KATILIMCI

- Bir ajan döngüsünü sıfırdan kodlayabilir
- Ajanın her adımını izleyip hata ayıklayabilir
- Yalın LLM ile ajan arasındaki farkı ölçülebilir biçimde gösterebilir

ARAÇLAR

Python **asyncio** **LiteLLM** **MCP** **GAIA**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Ajanınıza kendi yazdığınız bir araç ekleyin ve 5 soruluk mini bir değerlendirme seti hazırlayın.

KAYNAK DEPO `notebooks/ch04 — scratch_agents/types.py · context.py · llm.py · agent.py · tools/base.py`

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Ajanın çekirdeğini adım adım yazmak**	Demo ve ödev

DÜŞÜNME SORUSU

Ajan sonsuz döngüye girerse ne olur? max_steps bunu çözer mi, yoksa yalnızca gizler mi?

BÖLÜM II · AJANTİK MİMARİLER 10 / 15

OTURUM 10 · 6 MART 2027 · CUMARTESİ 09:00–13:00 · 4 SAAT

RAG, Dosya Araçları ve Callback'ler

Ajanı kendi verimize bağlamak ve kontrol altında tutmak

KAVRAM BLOĞU · 75 DK

- Dahili veri problemi: modelin bilmediği veriye erişmek
- Arama yöntemleri: anahtar kelime, vektör (anlamsal) ve yapı tabanlı arama
- Embedding, parçalama (chunking) stratejileri ve vektör arama
- Dosya sistemi araçları: unzip_file, list_files, read_file, read_media_file
- Callback'lerle ajanı genişletmek; insan onayı (human-in-the-loop)
- Arama sonuçlarını sıkıştırarak bağlam maliyetini düşürmek

UYGULAMA BLOĞU · 105 DK

Kendi dokümanlarımızla konuşan ajan

- Doküman koleksiyonunda embedding ve kosinüs benzerliği ile vektör arama yapmak
- Dosya araçlarıyla arşiv, Excel ve PDF içeren soruları yanıtlatmak
- Riskli bir araç için onay callback'i yazmak

OTURUM SONUNDA KATILIMCI

- Parçalama ve embedding seçimlerinin arama kalitesine etkisini yorumlayabilir
- Ajana dosya sistemi araçları verip güvenli biçimde sınırlayabilir
- İnsan onayı gerektiren bir araç akışı kurabilir

ARAÇLAR

OpenAI embeddings **NumPy** **Tavily** **Python**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Kendi dokümanlarınızdan (ör. bir yönerge ya da SSS) 20 soruluk bir arama testi hazırlayın. Bitirme projesi önerisi (1 sayfa) bu oturumda teslim alınır.

KAYNAK DEPO `notebooks/ch05 — scratch_agents/rag.py · callbacks.py · tools/file_tools.py`

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**Kendi dokümanlarımızla konuşan ajan**	Demo ve ödev

DÜŞÜNME SORUSU

Parça boyutu çok küçük ya da çok büyük olursa arama kalitesi neden düşer?

BÖLÜM II · AJANTİK MİMARİLER 11 / 15

OTURUM 11 · 13 MART 2027 · CUMARTESİ 14:00–18:00 · 4 SAAT

Bellek Sistemleri

Bağlamı yönetmek, oturumları sürdürmek, deneyimi biriktirmek

KAVRAM BLOĞU · 75 DK

- Ajan belleğinin anatomisi: bağlam penceresi, oturum ve uzun dönem bellek
- Depolama ile sunumu ayırmak; token sayımı
- Bağlam stratejileri: kayan pencere, sıkıştırma (compaction), özetleme, hiyerarşik yönetim
- Oturum ve durum yönetimi: Session, SessionManager, çok turlu konuşma
- Duraklat–devam et (pause/resume) ve araç onayı veri yapıları
- Uzun dönem bellek: bilgi çıkarımı, ChromaDB vektör deposu ve geri çağırma

UYGULAMA BLOĞU · 105 DK

Hatırlayan ajan

- Uzun bir görevde bağlam şişmesini üç stratejiyle karşılaştırmak (token maliyeti)
- Oturum destekli çok turlu ajan; onay bekleyip kaldığı yerden devam eden iş akışı
- Oturumlar arası bilgi hatırlayan ajan: ChromaDB ile uzun dönem bellek

OTURUM SONUNDA KATILIMCI

- Bağlam penceresi dolduğunda uygun stratejiyi seçebilir
- Oturum ve durum yönetimiyle çok turlu ajan kurabilir
- Uzun dönem belleğe ne yazılıp ne yazılmayacağına karar verebilir

ARAÇLAR

tiktoken **ChromaDB** **Pydantic** **Python**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Ajanınıza kullanıcı tercihlerini hatırlayan bir uzun dönem bellek ekleyin.

KAYNAK DEPO `notebooks/ch06 — scratch_agents/memory/session.py · long_term.py · context_optimizer.py · tools/memory_tool.py`

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Hatırlayan ajan**	Demo ve ödev

DÜŞÜNME SORUSU

Her şeyi hatırlayan bir ajan iyi bir ajan mıdır? Neyi unutmasını isteriz; gizlilik açısından ne gerekir?

BÖLÜM II · AJANTİK MİMARİLER 12 / 15

OTURUM 12 · 20 MART 2027 · CUMARTESİ 09:00–13:00 · 4 SAAT

Planlama, Yansıtma ve Kod Yürütme

Ajana düşünme zamanı vermek ve ona bir bilgisayar açmak

KAVRAM BLOĞU · 75 DK

- ReAct'in sınırları; ajana düşünme zamanı vermek
- Planlama: create_tasks aracı ve görev veri yapısı
- Yansıtma (reflection): çıktıyı denetlemek ve hatadan toparlanmak
- Neden kod? Önceden tanımlı araçların sınırı ve kod tabanlı eylemlerin gücü
- Sandbox (E2B) ve çalışma alanı (workspace): dosya sistemi ve komut satırı
- Agent Skills: ilerlemeli bilgi açma ve SKILL.md yapısı

UYGULAMA BLOĞU · 105 DK

Planlayan, hatadan dönen, kod yazan ajan

- Planlama aracıyla çok adımlı bir araştırma görevini yürütmek
- Hep başarısız olan bir araçla yansıtmanın hata toparlamadaki değerini görmek
- Ajana sandbox bağlayıp bir Excel dosyasını kod yazarak analiz ettirmek
- Bir PDF birleştirme 'skill'i yazıp ajana keşfettirmek

OTURUM SONUNDA KATILIMCI

- Planlama ve yansıtmanın hangi ReAct hata türlerine çare olduğunu açıklayabilir
- Kod çalıştıran ajanı güvenli bir sandbox içinde kurabilir
- Bir skill tanımlayıp ajana kazandırabilir

ARAÇLAR

E2B sandbox **Python** **Pydantic** **Tavily**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Ajanınıza yansıtma ekleyin ve başarısız bir araç senaryosunda toparlanmasını kaydedin.

KAYNAK DEPO `notebooks/ch07 · ch08 — scratch_agents/planning.py · tools/code_execution.py · skills.py`

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**Planlayan, hatadan dönen, kod yazan ajan**	Demo ve ödev

DÜŞÜNME SORUSU

Ajana kod çalıştırma yetkisi vermek hangi riskleri doğurur? Sandbox hangilerini çözer, hangilerini çözmez?

BÖLÜM II · AJANTİK MİMARİLER 13 / 15

OTURUM 13 · 3 NISAN 2027 · CUMARTESİ 14:00–18:00 · 4 SAAT

Çoklu Ajan Sistemleri

Uzmanlaşma, yalıtım ve ağ üzerinde iş birliği

KAVRAM BLOĞU · 75 DK

- Neden çoklu ajan? Uzmanlaşma, bağlam yalıtımı ve paralellik
- Üç iş birliği deseni: Workflow, Agent as Tool, Transfer
- Workflow: Sequential, Parallel ve Loop; iş akışlarını birleştirmek
- Agent as Tool: AgentTool, bağlam yalıtımı; araştırmacı + kodcu örneği
- Transfer: ajan ağacı ve aktarım hedefleri; transfer de bir araçtır
- A2A: ağ üzerinden ajanlar arası iletişim, Agent Card, sunucu ve istemci

UYGULAMA BLOĞU · 105 DK

Ajan takımı kurmak

- Araştırma → yazım → denetim hattı (sequential) ve paralel araştırma (parallel)
- Araştırmacı ve kodcu ajanlarını agent-as-tool olarak birleştirmek
- Uzman ajanlara yönlendiren bir karşılama ajanı kurmak (transfer)
- Bir ajanı A2A sunucusu olarak yayınlayıp uzaktan çağırmak

OTURUM SONUNDA KATILIMCI

- Üç iş birliği desenini görevin özelliğine göre seçebilir
- Ajanlar arasında bağlamı yalıtabilir
- Uzak bir ajanı A2A üzerinden çağırabilir

ARAÇLAR

asyncio **E2B** **Tavily** **A2A**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Bitirme projeniz için ajan mimarisi taslağını çizin (hangi desen, hangi araçlar).

KAYNAK DEPO `notebooks/ch09 — scratch_agents/workflows/ · transfer.py · tools/agent_tool.py · remote.py · a2a_server.py`

14:00	14:15	15:30	15:45	17:30
Isınma	Kavram bloğu	Mola	**Ajan takımı kurmak**	Demo ve ödev

DÜŞÜNME SORUSU

Tek bir güçlü ajan yerine çok sayıda ajan kullanmak ne zaman gereksiz karmaşıklık, ne zaman gerçek kazançtır?

BÖLÜM II · AJANTİK MİMARİLER 14 / 15

OTURUM 14 · 10 NISAN 2027 · CUMARTESİ 09:00–13:00 · 4 SAAT

Değerlendirme, İzleme ve Bitirme Atölyesi

Bir ajanın 'çalıştığını' ölçmek

KAVRAM BLOĞU · 75 DK

- Neden 'çalıştı' demek yetmez? Ajan değerlendirme çerçevesi
- Üç boyut: cevap alaka düzeyi, atıf güvenilirliği, gereksinim uyumu
- LLM-as-judge: değerlendirme istemleri ve 1–5 ölçek; yargıcın sınırları
- Genel geçme oranı (OPR): tüm kriterleri geçen / toplam değerlendirme
- GAIA ile nihai değerlendirme; izleme ve iyileştirme döngüsü

UYGULAMA BLOĞU · 105 DK

Ajanımızı ölçmek, projeleri olgunlaştırmak

- Kendi ajanınız için küçük bir değerlendirme seti ve LLM-as-judge hattı kurmak
- Ajan sürümlerini kıyaslamak: ReAct → +RAG → +planlama/yansıtma
- Bitirme projesi atölyesi: takım çalışması ve eğitmenle kod gözden geçirme

OTURUM SONUNDA KATILIMCI

- Bir ajan için ölçülebilir kalite metrikleri tanımlayabilir
- Değişikliklerin ajan kalitesine etkisini sayılarla gösterebilir
- Projesini sunuma hazır hale getirebilir

ARAÇLAR

LLM-as-judge **GAIA** **Python**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Sunum için 10 dakikalık demo akışı ve 1 sayfalık teknik özet hazırlayın.

KAYNAK DEPO `notebooks/ch10 — scratch_agents/eval/prompts.py`

09:00	09:15	10:30	10:45	12:30
Isınma	Kavram bloğu	Mola	**Ajanımızı ölçmek, projeleri olgunlaştırmak**	Demo ve ödev

DÜŞÜNME SORUSU

LLM-as-judge'a ne kadar güvenebiliriz? Yargıcın kendi hatalarını nasıl yakalarız?

BÖLÜM II · AJANTİK MİMARİLER 15 / 15

OTURUM 15 · 17 NISAN 2027 · CUMARTESİ 14:00–18:00 · 4 SAAT

Bitirme Projesi Sunumları ve Kapanış

Gösterelim, tartışalım, birlikte devam edelim

SUNUM GÜNÜ AKIŞI

- Takım sunumları: 10 dakika demo + 5 dakika soru-cevap
- Eğitmen ve katılımcı geri bildirimi (ölçütler 'Bitirme Projesi' bölümünde)
- Mimari kararların ve değerlendirme sonuçlarının sunulması
- Öğrenilen dersler: neyi farklı yapardık?

KAPANIŞ

Kapanış ve sonraki adımlar

- Sertifika töreni ve kapanış konuşması
- Sonraki adımlar: açık kaynak katkı, topluluk ve ileri okuma listesi
- Katılımcı geri bildirim anketi

OTURUM SONUNDA KATILIMCI

- Çalışan bir ajan sistemini kendi gerekçeleriyle sunabilir
- Aldığı geri bildirimle bir sonraki adımı planlayabilir

ARAÇLAR

Bitirme projeniz **Canlı demo**

ÖDEV / SONRAKİ OTURUMA HAZIRLIK

Projenizi açık bir depoya (repo) taşıyıp README ekleyin.

KAYNAK Önceki program · 19 Haziran: Öğrenci proje sunumları.

14:00	14:15	15:30	15:45	17:30
Hazırlık	Sunumlar I	Mola	**Sunumlar II**	Kapanış ve sertifika

DÜŞÜNME SORUSU

Bu ajanı gerçek kullanıcılara açsaydık ilk hafta ne ters gidebilirdi?

06 · PROGRAM BOYUNCA

Bitirme Projesi

Her katılımcı 2–3 kişilik bir takımla kendi ajan projesini geliştirir. Proje, Bölüm II'de kurulan çerçeve üzerinde yürür ve 17 Nisan'da canlı demoyla sunulur.

Kilometre taşları

6 Şubat	Oturum 7	DUYURU	Konu havuzu paylaşılır; 2–3 kişilik takımlar kurulur.
6 Mart	Oturum 10	ÖNERI	1 sayfalık proje önerisi: problem, kullanıcı, desen, veri.
20 Mart	Oturum 12	KONTROL	Çalışan ReAct çekirdeği ve en az iki araç.
10 Nisan	Oturum 14	ÖLÇÜM	Kendi değerlendirme setinizle ölçüm ve geri bildirim.
17 Nisan	Oturum 15	SUNUM	10 dk demo + 5 dk soru; kod deposu ve özet teslimi.

Proje fikirleri

Başlangıç havuzu; kendi fikrinizle gelmeniz teşvik edilir.

01 Yerel hizmet asistanı
Belediye hizmetleri ve SSS üzerinde RAG ile çalışan; riskli işlemlerde insan onayı isteyen asistan.

02 Atıflı araştırma ekibi
Bir konuyu paralel araştıran, planlayıp yazan ve kaynakları denetleyen çoklu ajan sistemi.

03 Veri analizi ajanı
Yüklenen Excel/CSV dosyasını sandbox içinde kod yazarak analiz eden ve raporlayan ajan.

04 Hatırlayan çalışma arkadaşı
Oturumlar arasında tercihleri ve görevleri hatırlayan uzun dönem bellekli asistan.

05 İnce ayar mı, RAG mı?
Aynı problemi LoRA ile ince ayarlı model ve RAG'li ajanla çözüp ölçümle karşılaştıran köprü projesi.

06 MCP ile açık veri
Bir açık veri kaynağını MCP sunucusu olarak açan ve A2A ile uzman ajan sunan proje.

Değerlendirme ölçütleri

öneri

Çalışırlık ve teknik derinlik	%30
Ajan tasarımı (araç, bellek, plan, desen)	%25
Değerlendirme ve ölçüm	%20
Sunum ve iletişim	%15
Dokümantasyon ve takım katkısı	%10

Sertifika koşulu (öneri): oturumlara düzenli katılım ve bitirme projesinin sunulması. Kaçırılan oturumlar için 30 Ocak yedek günü kullanılabilir.

EK A

Önceki Programdan Taşınanlar

Önceki program (Yapay Zeka Special Force Eğitimi, 12 hafta, Mart–Haziran 2025) derin öğrenmeden üretime alma altyapısına uzanan geniş bir yelpaze sunuyordu. Bu program yalnızca üretken yapay zekaya hizmet eden kısımları alır ve ajantik mimarilere yer açar.

ÖNCEKİ PROGRAM	KONU	BU PROGRAMDA	NOT
8 Mart	PyTorch'a giriş, gradyan iniş, Streamlit, bulutta yayın	**Oturum 1**	Bulut dağıtımı hariç
15 Mart	Derin Öğrenme I: ANN ve CNN	**Oturum 2**	Duygu analizi ve görüntü sınıflandırma korundu
22 Mart	FastAPI ile servis haline getirme	**Oturum 6**	BERT modelini servisleştirme olarak
5 Nisan	Derin Öğrenme II: RNN, LSTM, Attention; Auto Encoder	**Oturum 3 ve 4**	Dolar kuru tahmini ve anomali tespiti korundu; VAE eklendi
12 Nisan	Uçtan uca Transformer teknolojisi	**Oturum 5**	Sıfırdan self-attention eklendi
10 Mayıs	Uçtan uca BERT	**Oturum 6**	Sınıflandırma örneği korundu
24 Mayıs	GPT teknolojisi ve GPT finetuning (Falcon)	**Oturum 7**	Güncel açık modeller ve LoRA ile
19 Nisan, 3 Mayıs, 5 Haziran	Kuyruk yapıları (Kafka, RabbitMQ, Redis), Docker ve Kubernetes, Jenkins	**Kapsam dışı**	Üretken yapay zeka odağı için çıkarıldı; ileri okuma veya ek atölye olarak sunulabilir
17 Mayıs, 12 Haziran	Akademide ve İş Dünyasında Yapay Zeka günleri (etkinlik)	**Kapsam dışı**	Bu programda etkinlik günü yok
19 Haziran	Öğrenci proje sunumları	**Oturum 15**	Bitirme projesi sunumlarına dönüştü

EK B

Depo Haritası ve Açık Noktalar

Kaynak depo haritası

Depo: **github.com/shangrilar/ai-agent-from-scratch** · MIT Lisansı, © 2025 Jungjun Hur

BÖLÜM	KONU	OTURUM	BAŞLICA MODÜLLER (scratch_agents/)
CH02	LLM API temelleri	8	eval/gaia.py
CH03	Araç kullanımı, MCP	8	tools/helpers.py · calculator.py · search.py
CH04	ReAct ajanı	9	types.py · context.py · llm.py · agent.py · tools/base.py
CH05	RAG ve dosya araçları	10	rag.py · callbacks.py · tools/file_tools.py
CH06	Bellek sistemleri	11	memory/session.py · long_term.py · context_optimizer.py
CH07	Planlama ve yansıtma	12	planning.py
CH08	Kod yürütme, skills	12	tools/code_execution.py · skills.py
CH09	Çoklu ajan	13	workflows/ · transfer.py · tools/agent_tool.py · remote.py · a2a_server.py
CH10	Değerlendirme	14	eval/prompts.py

"Önemli kısımlar" seçimi: CH02'nin 100 isteklik eşzamanlılık örneği ve çok modelli GAIA kıyası (maliyet nedeniyle), CH08'in araçları sandbox'a taşıma alt bölümü (yalnızca özet) ve CH09'un uygulama ayrıntıları kısaltılır.

Akademi ile netleştirilecek hususlar

- Mekân, ağ erişimi ve ekipman; mekânın hem 09:00–13:00 hem 14:00–18:00 diliminde uygun olduğunun teyidi
- Kontenjan ve katılımcı seçim süreci
- API anahtarları ve maliyet: katılımcılar kendi anahtarını mı getirecek, ortak anahtar veya kredi mi sağlanacak? (Bölüm II örnekleri gerçek ve ücretli API istekleri yapar.)
- Oturum 7'deki ince ayar için GPU erişimi (Colab ücretsiz katmanı yeterli olacak şekilde küçük model planlanmıştır)
- Sertifika koşulları ve program süresi: bu belge 15 oturum varsayar (Aralık–Nisan)

EK C

Hazırlık ve Kurulum

Bölüm I için (5 Aralık'tan önce)

☐ **Google hesabı ve Colab erişimi**
Not defterleri Colab'da çalışır; yerel kurulum gerekmez.

☐ **İsteğe bağlı yerel kurulum**
Python 3.11+, Jupyter, PyTorch. Oturum 7'deki ince ayar için Colab GPU önerilir.

Bölüm II için (13 Şubat'tan önce)

☐ **Python 3.13+ ve uv**
Depo kurulumu: uv sync --locked; ardından uv run jupyter lab.

☐ **Node.js ve npm (npx)**
Oturum 8–9'daki Tavily MCP sunucusu için gerekir.

☐ **API anahtarları (.env dosyası)**
Aşağıdaki tablodaki anahtarlar; hangi oturumda gerektiği belirtilmiştir.

ANAHTAR (.env)	GEREKTİĞİ OTURUMLAR	NE İÇİN
`OPENAI_API_KEY`	Tüm Bölüm II oturumları	Model çağrıları ve embedding
`ANTHROPIC_API_KEY`	Oturum 8	Anthropic örnekleri
`TAVILY_API_KEY`	Oturum 8, 9, 10, 12	Web arama aracı
`E2B_API_KEY`	Oturum 12, 13	Kod yürütme sandbox'ı
`HF_TOKEN`	Oturum 8, 9	GAIA kıyas verisi (erişim izni gerekir)

MALİYET UYARISI

Depodaki örnekler gerçek ve ücretli API istekleri yapar. Hızlı kontrol için istek sayıları azaltılabilir. Maliyet yönetimi için Ek B'deki “Akademi ile netleştirilecek hususlar” listesine bakınız.