Large Language Model'ler Nasıl Çalışır?

LLM'lerin mekanik olarak ne olduğu, pretraining ve knowledge cutoff, base model ile instruction-tuned model farkı, ve in-context learning ile context kavramına ilk bakış.

Orta 20 dk

Son güncelleme: 6 Ekim 2026
Çevrimiçi oku: https://www.learnforgex.com/tr/topics/how-llms-work

---

Large Language Model'ler Nasıl Çalışır?

"Generative AI" bir önizlemeyle bitmişti: Large Language Model'ler (LLM'ler), metne uygulanmış generative AI'dır, transformer mimarisi üzerine inşa edilmiştir, ve kendinden önce gelen her şeye dayanarak sıradaki token'ı tahmin etmek üzere eğitilmiştir. Bu tek cümle doğrudur, ama çok şeyi sıkıştırır -- bu ders onu açıyor. Sonunda "pretraining"in (ön eğitim) gerçekte ne anlama geldiğini, bir sohbet asistanının altındaki ham modelden neden farklı davrandığını, ve "in-context learning"in (bağlam içi öğrenme) ne olduğunu bileceksiniz -- bu kursun geri kalanının tamamının üzerine kurulduğu bir kavram, bir sonraki dersteki "context" (bağlam) kavramının kendisiyle başlayarak.

Bir Modeli "Large Language Model" Yapan Nedir?

Bir LLM, mekanik olarak, "Deep Learning"in "Yaygın Neural Network Türleri" bölümünde anlatılan transformer tabanlı neural network'tür, devasa miktarda metinle, aldatıcı derecede basit tek bir hedefle eğitilmiştir: bir metin dizisi verildiğinde, sıradaki token'ı tahmin etmek. "Large" (büyük), aynı anda iki şeye işaret eder -- ağın boyutu (milyarlarca ağırlık, "Deep Learning"teki "Her Node Gerçekte Ne Yapar?"ı hatırlayın) ve eğitim verisinin boyutu (kamuya açık metnin anlamlı bir kısmı: kitaplar, web siteleri, kod, makaleler). "Deep Learning" ya da "Generative AI"dan altındaki mekanizmada hiçbir şey değişmedi -- değişen şey ölçektir, ve sıradaki bölümlerin göstereceği gibi, ölçek gerçekten yeni yetenekler üretmiş olduğu ortaya çıktı.

Neden Birçok Dar Model Yerine Tek Bir Genel Model?

LLM'lerden önce, çoğu NLP (doğal dil işleme) sistemi, bu kursun ilk dersindeki "Dar AI ve Genel AI"da anlatılan en katı anlamıyla narrow AI'dı: çeviri için ayrı bir model, duygu analizi için ayrı bir model, özetleme için ayrı bir model -- her biri kendi etiketli veri kümesine ve kendi eğitim sürecine ihtiyaç duyuyordu. LLM'ler var, çünkü yeterince genel metinle, basit

"sıradaki token'ı tahmin et" hedefiyle pretrain edilmiş tek bir model, dolaylı olarak dilbilgisini, olguları, akıl yürütme örüntülerini ve hatta talimatları nasıl takip edeceğini öğreniyor -- o kadar iyi ki *aynı* eğitilmiş model, her görev için ayrı ayrı eğitilmeden çeviri yapabiliyor, özetleyebiliyor, soruları yanıtlayabiliyor ve kod yazabiliyor. Bu, ölçeğin pratik getirisidir: tek bir genel amaçlı model, eskiden birçok dar, göreve özel modelin gerektirdiği şeyin yerini alıyor.

Pretraining: Dünyanın Metninden Öğrenmek

Pretraining (ön eğitim), "Machine Learning"in "Training ve Inference"inde genel olarak anlatılan (son derece pahalı, genellikle aylar süren, binlerce GPU'da çalışan) eğitim aşamasının -- burada devasa bir ölçekte uygulanmış halidir. Modele devasa miktarda metin gösterilir, ve her bir parça için, sıradaki token'ı tahmin etmesi istenir; yanlış yaptığında, ağırlıkları ("Deep Learning"teki "backpropagation" ve "gradient descent"i hatırlayın) bir dahaki sefere o tahmini biraz daha iyi yapacak şekilde ayarlanır. Trilyonlarca token boyunca tekrarlandığında, bu süreç modele dilbilgisini, dünya hakkındaki olguları (eğitim verisinin toplandığı ana kadar -- **knowledge cutoff**'u, bilgi kesim tarihi), yaygın akıl yürütme örüntülerini, ve hatta programlama dillerini -- sırf sıradaki kelimeyi tahmin etmede giderek daha iyi olmanın bir yan etkisi olarak -- öğretir.

> Bir modelin knowledge cutoff'u yaygın bir kafa karışıklığı kaynağını açıklar: bir LLM, ne kadar kendinden emin cevap verirse versin, eğitim verisi toplandıktan sonraki olaylar hakkında bilgi sahibi olamaz. Bu, yamalanacak bir hata değildir -- pretraining'in nasıl çalıştığının doğrudan bir sonucudur. Bu kursun ilerleyen "Tool'lar ve MCP" kategorisi, sistemlerin modellere yine de güncel bilgiye erişim sağlamasının nasıl mümkün olduğunu kapsıyor.

Base Model'ler ve Instruction-Tuned (Chat) Modeller

Doğrudan pretraining'den çıkan bir model -- genellikle **base model** (temel model) denir -- yalnızca bir şeyde iyidir: metni istatistiksel olarak makul bir şekilde sürdürmek. Ona "Fransa'nın başkenti" verirseniz, akıcı biçimde devam eder. Ama ona "Bu e-postayı özetle" gibi bir talimat verirseniz, bir base model gerçekten bir şeyi özetlemek yerine (bu örüntü de eğitim metninde göründüğü için) pekala *başka talimatların bir listesiyle* devam edebilir. Bu yüzden etkileşimde bulunduğunuz neredeyse her LLM -- bir sohbet asistanı, bir kodlama aracı -- pretraining'den sonra ek bir eğitim aşamasından geçmiştir, genel olarak **instruction tuning** (talimat ayarlama) denir (bazen RLHF, Reinforcement Learning from Human Feedback, denen bir teknikle birleştirilir), burada model özellikle talimatları takip etmesi, bir sohbeti sürdürmesi, ve yalnızca metni sürdürmek yerine yararlı davranması için ek olarak eğitilir. Base model'in öğrendiği devasa bilgi kaybolmaz -- instruction tuning, o bilginin *nasıl* ifade edildiğini yeniden şekillendirir.

In-Context Learning: Bir Model Ona Verdiğinizi Nasıl Kullanır?

Pretraining bir kez, çevrimdışı gerçekleşir ("Training ve Inference"i tekrar hatırlayın). Ama bir LLM açıkça davranışını *tek bir sohbet içinde* uyarlayabilir -- ona uydurma bir kelime öğretebilir, takip etmesi için bir örnek format verebilir, ya da hiçbir şeyi yeniden eğitmeden bir hatayı düzeltebilirsiniz. Buna **in-context learning** denir: modelin ağırlıkları bir sohbet sırasında hiç değişmez (hiçbir eğitim gerçekleşmez); bunun yerine, sağladığınız her şey -- talimatlarınız, örnekleriniz, ve o ana kadarki sohbet -- her tek yanıtta modele girdi olarak yeniden beslenir, ve modelin mevcut (donmuş) pretrain edilmiş yetenekleri, o girdide bulunan örüntüleri tanımasını ve takip etmesini sağlayan şeydir. Bu ayrım son derece önemlidir: model, sohbetinizi bir veritabanının yapacağı gibi "hatırlamıyor" -- her yanıt ürettiğinde mevcut girdinin tamamını baştan yeniden okuyor.

Context'e İlk Bakış

Önceki bölümdeki "sağladığınız her şey"in bir adı var: **context** (bağlam). En basit anlamda, context, modelin sıradaki token'ını üretmeden önce gerçekten gördüğü tüm metindir -- talimatlarınız, ona sağlanan her türlü arka plan bilgisi, sohbet geçmişi, ve mevcut yanıtında o ana kadar ürettiği her şey. Context, bir LLM'in davranışının inference zamanında şekillendirilebileceği *tek* kanaldır ("Training ve Inference"ten hatırlayın, inference sırasında hiçbir öğrenme gerçekleşmez) -- bir LLM'in belirli bir yanıt için "bildiği" her şey ya pretraining'den gelmiştir, ya da şu anda context'inde bir yerdedir. Bu tek fikir -- context'in yalnızca bir arayüz özelliği değil, mekanizmanın kendisi olduğu -- bu kursun geri kalanının üzerine inşa edildiği temeldir: bir sonraki ders context'in çok gerçek boyut sınırlarına bakıyor, ve ilerleyen "Tool'lar ve MCP" kategorisi, sistemlerin *doğru* bilgiyi doğru zamanda bir modelin context'ine nasıl soktuğunu tam olarak kapsıyor.

Ölçek: Parametreler, Veri ve Hesaplama

"Deep Learning"ten bir modelin **parametrelerinin**, öğrenilmiş ağırlıkları olduğunu hatırlayın -- "milyarlarca parametre," bu ayarlanabilir sayılardan milyarlarca tanesi anlamına gelir. Modern LLM'ler birkaç milyardan bir trilyonun üzerine kadar parametreye sahiptir, trilyonlarca token metinle, haftalarca ya da aylarca çalışan binlerce GPU kullanılarak eğitilir ("Deep Learning 2010'larda Neden Patladı?"yı hatırlayın -- on yıl sonra çok daha büyük bir ölçekte aynı veri-ve-hesaplama hikayesi). Araştırmacılar oldukça öngörülebilir **scaling law'lar** (ölçeklendirme yasaları) gözlemlediler: performans, model boyutu, veri boyutu ve hesaplama birlikte arttıkça düzgün biçimde iyileşme eğilimindedir -- bu, endüstrinin neden giderek daha büyük modellere yöneldiğinin bir kısmıdır, ama ölçek tek başına her sınırlamayı çözmez ("LLM Yetenekleri ve Sınırlamaları"na, bu kategorinin son dersine bakın).

Best Practices

- Bir LLM'in "bildiği" bir iddiayı değerlendirirken, bunun pretraining'den mi (ve knowledge cutoff'u kontrol edin) yoksa sağladığınız context'ten mi geldiğini sorun -- bunlar tek iki kaynaktır ("Context'e İlk Bakış"a bakın).
- Bir modelin ağırlıklarının eğitimden sonra donduğunu unutmayın -- bir sohbette söylediğiniz hiçbir şey modeli kalıcı olarak değiştirmez, yalnızca o sohbetin context'inde olanı değiştirir ("In-Context Learning: Bir Model Ona Verdiğinizi Nasıl Kullanır?"a bakın).
- Talimat takibi açısından beklenmedik bir davranışla karşılaştığınızda, düzgün instruction-tuned bir asistana karşı bir base model'in ham tamamlama davranışına daha yakın bir şeyle mi etkileşimde olduğunuzu düşünün ("Base Model'ler ve Instruction-Tuned (Chat) Modeller"e bakın).

Yaygın Hatalar

- **Bir LLM'in önceki sohbetleri bir insanın ya da bir veritabanının yaptığı gibi "hatırladığını" varsaymak.** "In-Context Learning: Bir Model Ona Verdiğinizi Nasıl Kullanır?"ın açıkladığı gibi, bir sistem bu bilgiyi yeni bir sohbette açıkça context olarak yeniden sağlamadığı sürece ayrı sohbetler arasında hiçbir şey saklanmaz -- modelin kendisi hiçbir şey tutmaz.
- **Bir base model'in talimatları iyi takip etmesini beklemek.** "Base Model'ler ve Instruction-Tuned (Chat) Modeller"in gösterdiği gibi, ham sıradaki-token tahmini ve talimat takibi gerçekten farklı eğitilmiş davranışlardır -- bir modelin ikincisini güvenilir biçimde yapması için ikinci bir eğitim aşamasına ihtiyacı vardır.
- **"Daha büyük model"i her problem için otomatik bir çözüm saymak.** Ölçek ("Ölçek: Parametreler, Veri ve Hesaplama"ya bakın) çok şeyi iyileştirir, ama bu kategorideki bir sonraki ders anlatacağı gibi, bazı sınırlamalar yetersiz boyuttan değil, LLM'lerin temelde nasıl çalıştığından kaynaklanır.

Özet, Cheat Sheet ve Terimler Sözlüğü

Özet

- Bir LLM, metindeki sıradaki token'ı tahmin etmek üzere devasa bir ölçekte eğitilmiş transformer tabanlı bir modeldir -- önceki kategoriden aynı generative AI mekanizması, yalnızca çok daha büyük ölçekte uygulanmış.
- Tek bir genel pretrain edilmiş model, eskiden her biri ayrı bir dar model gerektiren birçok görevi (çeviri, özetleme, kodlama) gerçekleştirebilir.
- **Pretraining,** bir modele, devasa miktarda metin boyunca sıradaki token'ı tahmin ettirerek dilbilgisini, olguları ve akıl yürütme örüntülerini sırf bunun sonucunda öğretir.

- **Base model'ler** metni makul biçimde sürdürür; **instruction-tuned (chat) modeller,** talimatları güvenilir biçimde takip etmek ve sohbet sürdürmek için ek bir eğitim aşamasından geçer.
- **In-context learning,** bir modelin davranışının hiçbir ağırlık değişmeden bir sohbet içinde uyarlanması anlamına gelir -- her şey **context**'ten gelir, modelin üretmeden önce gerçekten gördüğü metinden.
- LLM yeteneği ölçekle (parametreler, veri, hesaplama) iyileşme eğilimindedir, ama ölçek her sınırlamayı düzeltmez.

Cheat Sheet

- LLM = transformer + devasa ölçek + sıradaki token tahmini.
- Pretraining = bir kereye mahsus, pahalı eğitim aşaması. Knowledge cutoff = pretraining verisinin ne kadar güncel olduğu.
- Base model = ham tamamlama. Instruction-tuned model = talimatları güvenilir biçimde takip eder (instruction tuning / RLHF aracılığıyla).
- In-context learning = bir sohbet içinde davranışı uyarlamak, ağırlık değişikliği yok.
- Context = modelin sıradaki token'ından önce gördüğü her şey -- inference zamanında çıktısını şekillendirmenin tek kanalı.

Terimler Sözlüğü

- **Large Language Model (LLM):** devasa ölçekte eğitilmiş, metin tahmin eden ve üreten, transformer tabanlı bir generative AI modeli.
- **Pretraining (ön eğitim):** bir modelin devasa miktarda metin boyunca sıradaki token'ı tahmin etmeyi öğrendiği ilk, büyük ölçekli eğitim aşaması.
- **Knowledge cutoff (bilgi kesim tarihi):** pretraining verisi o tarihten önce toplandığı için, bir modelin ondan sonrası hakkında hiçbir bilgisi olmadığı zaman noktası.
- **Base model (temel model):** yalnızca pretraining'den geçmiş bir model -- metni makul biçimde sürdürür ama talimatları güvenilir biçimde takip etmez.
- **Instruction tuning / RLHF:** bir base model'i talimatları güvenilir biçimde takip eden ve sohbet sürdüren bir modele dönüştüren ek eğitim aşamaları.
- **In-context learning (bağlam içi öğrenme):** bir modelin, hiçbir ağırlık değişikliği olmadan, sağlanan context'e dayanarak davranışını tek bir sohbet içinde uyarlaması.
- **Context (bağlam):** bir modelin sıradaki token'ını üretmeden önce gerçekten gördüğü tüm metin -- talimatlar, arka plan bilgisi ve sohbet geçmişinin birleşimi.
- **Parametreler:** bir modelin öğrenilmiş, ayarlanabilir ağırlıkları -- "milyarlarca parametre," bir modelin boyutunu tarif eder.
- **Scaling law'lar (ölçeklendirme yasaları):** model boyutu, veri ve hesaplama birlikte arttıkça LLM performansının öngörülebilir biçimde iyileşme eğiliminde olduğu gözlemlenen örüntü.