Eurasian Academy of Sciences
Eurasian Econometrics, Statistics & Empirical Economics Journal
2025 Volume:26 S: 87- 125
Published Online September 2025 (http://econstat.eurasianacademy.org)
http://dx.doi.org/10.17740/eas.stat.2025-V25-06

A NEW SPRING FOR STATISTICAL METHODS: LARGE LANGUAGE MODELS (LLMS)

Kutluk Kağan SÜMER*

*Prof. Dr., İstanbul Üniversitesi, İktisat Fakültesi, Ekonometri Bölümü, kutluk@istanbul.edu.tr, ORCID ID: 0000-0002-5648-381X

Received Date:25.07.2025 Accepted Date:02.09.2025

ABSTRACT

Large Language Models (LLMs) are the cornerstone of modern AI systems capable of humanlike reasoning, language understanding, and text generation. Their success relies not only on deep learning architectures but also on a comprehensive statistical foundation. This article provides an extensive examination of statistical techniques underlying LLMs, including probability theory, statistical learning theory, Bayesian inference, Markov chains, the Expectation–Maximization algorithm (EM), dimensionality reduction (PCA, SVD), probabilistic graphical models, variational inference, and sampling methods such as MCMC. It further explains how these methods are integrated within the Transformer architecture and contemporary LLM training pipelines. Applications in natural language processing, healthcare, finance, and law are also explored in detail.

Keywords: Large Language Models, Statistical Learning, Bayesian Inference, Transformer, EM Algorithm, PCA, SVD
JEL Classification: C45, C38, C55, C63

İSTATİSTİKSEL YÖNTEMLERİN YENİ BAHARI: BÜYÜK DİL MODELLERİ (BDM)

ÖZET

Büyük dil modelleri (LLM'ler), günümüz yapay zekâ uygulamalarının merkezinde yer alan ve insan benzeri dil üretme, anlama, akıl yürütme kabiliyetleriyle dikkat çeken derin öğrenme sistemleridir. Bu modellerin başarısının temelinde yalnızca derin sinir ağları değil, aynı zamanda güçlü bir istatistiksel altyapı bulunmaktadır. Bu makale, LLM'lerin ardındaki olasılık teorisi, istatistiksel öğrenme kuramı, Bayesci yöntemler, Markov zincirleri, beklenti–maksimizasyon algoritması (EM), boyut indirgeme (PCA, SVD), olasılıksal grafik modeller, varyasyonel çıkarım ve örnekleme yöntemleri (MCMC) gibi tüm temel istatistiksel teknikleri kapsamlı bir biçimde ele almaktadır. Ayrıca bu tekniklerin Transformer mimarisi ve modern LLM eğitim süreçlerinde nasıl kullanıldığı ayrıntılı olarak gösterilmiştir. Makale, doğal dil işleme, sağlık, finans, hukuk gibi alanlarda LLM tabanlı istatistiksel modellerin uygulamalarını da örneklerle incelemektedir.

Anahtar Kelimeler: Büyük Dil Modelleri, İstatistiksel Öğrenme, Bayesci Çıkarım, Transformer, EM Algoritması, PCA, SVD
JEL Sınıflandırması: C45, C38, C55, C63

1 – Dilin Olasılıksal Temeli ve Bilgi Teorisi (Yeniden Yazım)

Büyük dil modelleri, dilin içsel belirsizliğini matematiksel bir yapıya oturtarak, kelimeleri ve cümleleri olasılıksal bir süreç çerçevesinde temsil eder. İnsan dili, deterministik bir yapıdan çok, tarihsel, kültürel ve bağlamsal etkilerin şekillendirdiği geniş bir olasılık uzayında yaşamaktadır. Bu nedenle modern dil modelleri, dile ilişkin temel varsayımı şu şekilde kurar: Her kelime dizisi, bir olasılık dağılımından çekilmiş rastgele bir örnektir. Dolayısıyla bir dil modeli, aslında dilin bu olasılık dağılımını tahmin etmeye çalışan bir fonksiyondur. Eğer bir kelime dizisini $W=(w_1,w_2,...,w_T)$ olarak yazarsak, model bu dizinin gerçekleşme olasılığını şu şekilde ifade eder:

$$P(W) = P(w_1, w_2, \dots, w_T)$$

Olasılık zincir kuralı, dil modellemenin temel yapı taşını oluşturur ve bu diziyi ardışık koşullu olasılıkların çarpımı şeklinde yeniden yazmamızı sağlar:

$$P(W) = \prod_{t=1}^{T} P(w_t \mid w_{1:t-1})$$

Aslında bu formül, dil modelinin bir "tahmin makinesi" olarak davranmasının matematiksel yansımasıdır. İnsan nasıl bir cümlenin devamını sezgisel olarak tahmin ediyorsa, model de geçmiş sözcükleri "bağlam" olarak kabul eder ve sıradaki kelimenin olasılığını hesaplar.

Bu yaklaşım dilin olasılıksal doğasına dayanır. Örneğin "yarın hava…" ifadesinden sonra "güzel" kelimesi, "bisiklet" kelimesine kıyasla çok daha yüksek bir koşullu olasılığa sahiptir. Çünkü bağlamın sağladığı bilgi, sonraki kelimenin dağılımını daraltarak belirsizliği azaltır. LLM'lerin yaptığı tam olarak budur: Dağılımları öğrenir, belirsizliği minimize eder ve bağlama uygun sözcükleri yüksek olasılıkla seçer.

1.1 Olasılık Dağılımlarının Dil Modellemesindeki Rolü

Bir büyük dil modelinin çıktısı aslında tek bir kelime değildir; kelimelerin bulunduğu sözlük üzerinde tanımlanan bir ihtimal dağılımıdır. Model, her bir kelime için bir sayı üretir ve bunları "logit" olarak adlandırır. Bu logit değerleri softmax fonksiyonundan geçirilerek normalize edilir:

$$P(w_t = i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Bu ifade, modelin bir sonraki kelime için oluşturduğu kategorik dağılımdır. Kategorik dağılım, kelime seçiminin ardında yatan rastgele sürecin en yalın matematiksel karşılığıdır çünkü tek bir denemede çok kategorili bir sonuç üretir. Dilin üretilişi de bir yönüyle böyle işler; bir sonraki sözcük tek bir seçenektir, fakat aday havuzu çok geniştir.

Bir adım daha yukarıda, çoknomlu dağılım dil modellerinin mini-batch eğitim süreçlerinde karşımıza çıkar. Çok sayıda cümle aynı anda işlendiğinde, o bağlamda türeyen kelime frekansları aslında çoknomlu dağılımı takip eder. Bu tür istatistiksel ilişkiler, modelin neden bazı kelimelere daha fazla yoğunlaştığını anlamamızı sağlar.

Dil modellemede sık kullanılan bir diğer dağılım Normal (Gauss) dağılımıdır. Özellikle gömülü (embedding) uzaylarının matematiksel yapısı çoğu zaman Normal (Gaussiyen) davranışlara yaklaşır. Yüksek boyutlu semantik uzayda kelime vektörlerinin kümelenme eğilimleri, anlam yakınlıklarının Gauss (Normal) dağılımı üzerinden modellenmesini kolaylaştırır. Bir kelime embedding'i (gömülü boyutu) çoğu zaman şöyle ifade edilir:

$$x \sim \mathcal{N}(\mu, \Sigma)$$

Bu ifade, kelimelerin anlamlarının tek bir noktaya indirgenemeyeceğini, aksine bir belirsizlik bölgesi olarak temsil edilmesi gerektiğini anlatır.

1.2 Bilgi Teorisi: Dilin Matematiksel Karmaşıklığı

Dilin içsel karmaşıklığını ölçmek için bilgi teorisi kullanılır. Claude Shannon'ın geliştirdiği entropi kavramı, bir rasgele değişkenin ne kadar belirsizliğe sahip olduğunu matematiksel olarak ifade eder. Eğer dildeki kelimelerin olasılık dağılımını P(x) ile gösterirsek, entropi şu şekilde hesaplanır:

$$H(X) = -\sum_{x} P(x) \log P(x)$$

Her dilin kendine özgü bir entropisi vardır. Örneğin Türkçe'de bazı eklerin sık kullanılması, İngilizce'de kelime sırasının daha sınırlı bir şekilde kurulması, Japonca'da cümlenin fiille sonlanması gibi kurallar doğal entropiyi değiştirir. Dil modelleri bu entropiyi dolaylı olarak öğrenir.

LLM'lerin eğitiminde kullanılan temel kayıp fonksiyonu çapraz entropidir:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P_\theta(w_t \mid w_{<t})$$

Bu ifade, modelin dağılımı ile gerçek dağılım arasındaki uzaklığı ölçer. Eğer model doğru tahmin yapıyorsa kayıp azalır, yanlış kelimelere yüksek olasılık veriyorsa kayıp yükselir. Eğitim süreci bu kaybı minimize etmeye çalışır ve böylece model giderek daha tutarlı bir dil dağılımı öğrenir.

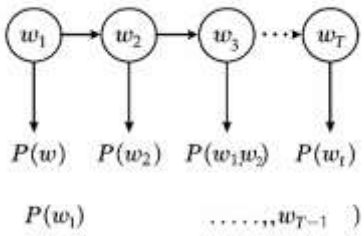
Performansı göstermek için geleneksel ölçü "perplexity"dir:

$$\text{Perplexity} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T} \log P(w_t \mid w_{<t})\right)$$

Perplexity dilin belirsizliğini temsil eden sezgisel bir ölçüdür. Düşük perplexity (şaşkınlık, karışıklık, tereddüt) daha iyi tahmin gücü anlamına gelir. İnsan dil yeteneğini ölçseydik muhtemelen benzer bir metrik kullanırdık; çünkü insan beyni de bağlam üzerinden olasılık tahminleri yapar.

1.3 Olasılığın Görsel Bir Tasviri

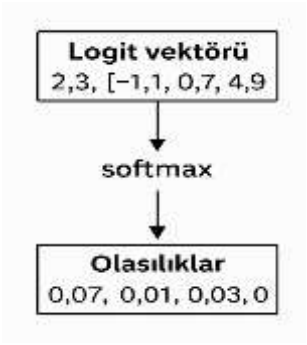
Bu kavramların tümü, aşağıdaki gibi basit bir şemada görselleştirilebilir:



Bu çizim yalnızca ardışıklığı göstermiyor; aynı zamanda geçmişten geleceğe aktarılan bilginin matematiksel temsili olduğunu da hatırlatıyor. İnsan zihni de cümleleri bu şekilde kurar; kelimeler birbirlerinin ağırlıklı olasılıklarını belirler ve anlam zaman içinde akarak oluşur.

1.4 Softmax'ın Görsel Anlatımı

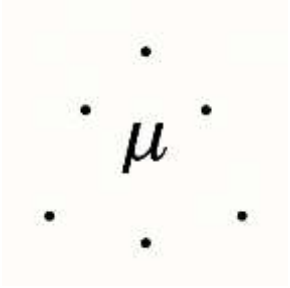
Bir modelin logitlerini olasılıklara dönüştürme süreci de şu şekilde düşünülebilir:



Bu örnek, modelin büyük olasılıkla "dördüncü kelimeyi" seçeceğini gösterir. Ancak seçim deterministik değildir; dilin büyüsü burada yatar. LLM'ler, anlam üretmek için olasılık dağılımını kullanır, rastlantısallıkla yaratıcılık arasında denge kurar.

1.5 Embedding (Gömülü) Uzayının Estetiği

Yüksek boyutlu embedding uzaylarında kelimeler adeta bir anlam manzarası oluşturur:



Bu basit çizim, kelimelerin rastgele dağılmadığını, semantik olarak ilişkili kelimelerin birbirine yakın kümelendiğini hatırlatır. LLM'ler bu geometrik işaretleri kullanarak anlam çıkarır.

2 – İstatistiksel Öğrenme Kuramı: ERM, SRM, Taraflılık-Sistematik hata varyansı (Bias–Variance) ve Düzenlileştirme

Büyük dil modellerinin eğitimi, ne kadar devasa veri ve parametre kullanırsa kullansın, özünde istatistiksel öğrenme kuramının temel ilkelerine sıkı sıkıya bağlıdır. Bu kuramın merkezinde, modelin verilerden anlamlı bir fonksiyon öğrenmesini mümkün kılan iki önemli yaklaşım bulunur: Empirik Risk Minimizasyonu (ERM) ve Yapısal Risk Minimizasyonu (SRM). LLM'lerin eğitim sürecinin, yüzeyde karmaşık sinir ağı optimizasyonu gibi görünmesine rağmen, bu kuramsal temellerden nasıl beslendiğini göstermeye çalışacağız.

Dil modellemede amaç, belirli bir modelin parametrelerini öyle bir biçimde ayarlamaktır ki, model önceden görülmemiş metinlerde dahi tutarlı tahminler yapabilsin. Bu hedef genellikle "genelleme" olarak adlandırılır ve genelleme başarısı, istatistiksel öğrenmenin omurgasını oluşturur.

2.1 Empirik Risk Minimizasyonu: Öğrenmenin En Saf Formu

Bir modelin eğitimi sırasında, elimizdeki veri kümesini $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ olarak düşünelim. Burada x_i bağlamı (örneğin cümlenin önceki kelimeleri), y_i ise modelin tahmin etmeye çalıştığı kelimedir. Dil modelinin eğitimi, bu veri noktaları üzerinde bir kayıp fonksiyonunun ortalamasını minimize etmeyi hedefler:

$$R_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i)$$

Bu ifade, modelin "gözlemlenen" veriye ne kadar iyi uyduğunu ölçer. Burada kayıp fonksiyonu L, LLM'lerde genellikle negatif log-olasılıktır:

$$L = -\log P_\theta(y_i \mid x_i)$$

Yani model, her doğru kelime için olasılığı mümkün olduğunca yukarı çekmeye çalışır; yanlış kelimelere giden olasılığı ise bastırır. Bu yaklaşım, yüzeyde teknik görünse de, sezgisel olarak şu anlama gelir: Modeli binlerce, milyonlarca örneğe bakmaya ve bu örneklerdeki örüntüleri öğrenmeye zorluyoruz. İnsanların bir dili nasıl öğrendiğini düşündüğümüzde bile, doğru ifadeleri tekrar tekrar duymanın uzun vadede doğru olasılık tahminleri yapmamızı sağladığını görürüz.

Ancak ERM'nin bir kusuru vardır: Model yalnızca gördüğü örneklerde iyi performans göstermek üzere eğitilir. Eğer model aşırı karmaşıksa ve veri yetersizse, iyi bir genelleme yapmadan yalnızca veriyi ezberleme eğilimi gösterebilir.

2.2 Yapısal Risk Minimizasyonu: Öğrenmeyi Dengede Tutmak

Vapnik'in öncülüğünü yaptığı Yapısal Risk Minimizasyonu (SRM), öğrenme sürecinin yalnızca empirik kaybı minimize etmeye dayanmasının yeterli olmadığını savunur. Asıl amaç, yalnızca eğitim verisini iyi açıklamak değil, aynı zamanda genelleme kapasitesini en üst düzeyde tutmaktır.

SRM, bunu modelin karmaşıklığını cezalandırarak yapar. Matematiksel olarak ifade edecek olursak, SRM'nin hedeflediği fonksiyon şudur:

$$R_{\text{SRM}}(f) = R_{\text{emp}}(f) + \lambda\Omega(f)$$

Burada Ω(f) modelin karmaşıklığını ölçen bir terimdir. LLM'lerde bu çoğu zaman parametrelerin normudur.

Bu düşünce aslında çok insani bir sezgiye dayanır: Ne kadar karmaşık bir açıklama yaparsak hata yapma ihtimalimiz o kadar artar. Basit açıklamalar, çoğu zaman daha sağlam temellere dayanır. Büyük dil modelleri de bu mantığı içselleştirir ve dev parametre sayılarına rağmen düzenlileştirme teknikleri sayesinde aşırı öğrenme sorununu aşar.

2.3 Bias–Variance Dengesi: Öğrenmenin Paradoksu

Makine öğrenmesinin en çarpıcı kavramlarından biri bias–variance dengesidir. Bir modelin öğrenme hatası, üç bileşenden oluşur:

Bias (yanlılık, sistematik hata) , modelin veriyi fazla basitleştirdiği durumlarda ortaya çıkar. Varyans ise modelin veriye aşırı uyum sağladığı, yani ezberlediği durumudur. LLM'lerin yüksek parametreli yapılarında varyans'ın teorik olarak çok yüksek olması beklenirdi. Ancak şaşırtıcı bir şekilde, geniş modeller daha iyi genelleme performansı gösteriyor. Bu durum Çifte iniş (double descent) fenomeniyle ilişkilendirilir; model parametreleri kritik bir eşiği aştığında, varyans artışı tersine döner ve genelleme beklenmedik şekilde iyileşir.

Bu fenomenin altındaki matematik hâlâ araştırılmakta olsa da, pratikte geniş LLM'lerin neden daha iyi çalıştığını açıklamaya yardımcı olur.

2.4 Düzenlileştirme: Öğrenmeyi Sağlamlaştırmak

Düzenlileştirme teknikleri, öğrenme sürecini istikrarlı hâle getirir. En klasik yöntem L2 düzenlileştirmesidir:

$$\lambda\|\theta\|^2$$

Bu terim, model parametrelerinin sonsuz büyüklüğe kaçmasını engeller ve fonksiyonun pürüzsüz kalmasını sağlar. Dil modellerinde bu yalnızca matematiksel bir zorunluluk değildir; aynı zamanda dilin kendisinin pürüzsüz bir doğası vardır. İnsan dili, aşırı uçlara gitmez; anlam akışı düzenlidir. L2 düzenlileştirme, modelin öğrenmesini bu doğal akışa yaklaştırır.

Dışlanma (dropout) ise tamamen farklı bir sezgiyle çalışır. Her eğitim adımında bazı nöronları rastgele "kapatarak" modelin alternatif bağlantılar üzerinden düşünmesini sağlar. Matematiksel olarak olmasa da felsefi olarak şu fikre dayanır: Eğer bir beyin sürekli aynı bağlantıları kullanırsa, başka yolları öğrenemez. Dışlanma (dropout), bu diğer yolların keşfedilmesini sağlar.

2.5 Optimizasyon: Öğrenmenin Motoru

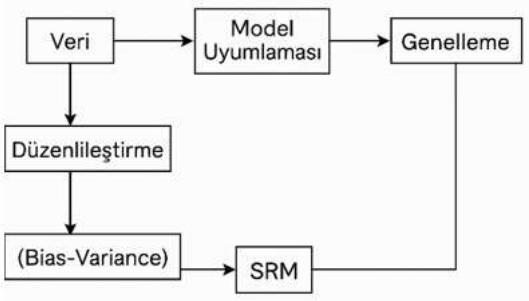
Tüm bu teorik çerçeve, nihayetinde bir optimizasyon probleminin içinden geçer. LLM'lerde kullanılan optimizasyon çoğunlukla adam tabanlı yöntemlere dayanır. En basit formuyla gradyan azaltma (gradient descent) şu şekilde yazılır:

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta\mathcal{L}$$

Fakat pratikte Adam gibi yöntemler, momentum ve adaptif öğrenme oranı gibi iyileştirmeler sayesinde daha hızlı ve stabil bir yakınsama sağlar. Dil modellerinin yüz milyarlarca parametre içermesi düşünüldüğünde, bu optimizasyon yöntemleri olmadan eğitim neredeyse imkânsız olurdu.

2.6 Öğrenme Sürecinin Görsel Bir Anlatımı

Bu kavramların birbirini nasıl beslediğini göstermek için aşağıdaki gibi soyut bir çizim yardımcı olabilir:



Bu şema, dil modellerinin yalnızca veriyi ezberleyen mekanik sistemler olmadığını, doğru dengeler kurulduğunda kavramsal bilgiye yaklaşabildiklerini gösterir.

Bayesian Çıkarım, Posterior (sonsal olasılık) Mantığı, Markov Zincirleri ve MCMC'nin LLM'lerle İlişkisi

Bayesci düşünme biçimi, belirsizliğin matematiksel olarak nasıl ele alınabileceğine dair en zarif yaklaşımlardan biridir ve özellikle büyük dil modellerinin temelindeki birçok fikri sezgisel olarak besler. Dilin doğası gereği belirsizlik içermesi, Bayes yöntemlerini dil modellemenin teorik çerçevesi ile uyumlu hâle getirir. Bir cümlenin devamında hangi kelimenin geleceğini tahmin ederken, aslında insan zihni de Bayesci şekilde çalışır: Önce genel bir beklenti öncül (prior) belirler, ardından bağlamdan gelen yeni bilgileri (likelihood~benzerlik) bu beklentiyle birleştirerek güncellenmiş bir inanç (posterior) oluşturur. Büyük dil modellerinin yaptığı şey, bu süreci devasa veri ve hesaplama gücüyle otomatikleştirmektir.

Bayes teoremi matematiksel olarak basit görünse de yorumlandığında oldukça derin bir içeriğe sahiptir:

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)}$$

Bu ifade, bilinmeyen parametrelerin olasılık dağılımının, gözlemlenen veriler ışığında nasıl güncellenmesi gerektiğini gösterir. Dil modellemede θ, modelin parametrelerini temsil ederken, Xgözlemlenen kelime dizisinin kendisidir. Ancak modern LLM'ler doğrudan Bayesian posterior hesaplamaları yapmaz; yine de eğitimin doğası, Bayesci bakış açısının izlerini taşır.

3.1 Prior, Likelihood ve Posterior'un Dile Uyarlanması

Bir cümlenin başında hiçbir bağlam yokken, dil modelinin olası kelimeler üzerine dağılımı aslında bir prior'dır. Türkçe'de bir cümlenin büyük ihtimalle özne ya da belirleyici bir unsurla başlaması beklenir. Bu bekleyiş, kültürel ve dilbilgisel bir önbilginin sonucudur. Bağlam ilerledikçe, modelin gördüğü yeni kelimeler likelihood'ı belirler ve posterior dağılım —yani bir sonraki kelime için güncellenmiş beklenti— giderek daralır.

Bu durumu soyut bir örnekle düşünelim. "Doktor ameliyathaneye…" ifadesinden sonra "girdi" kelimesi, "gömlek" kelimesinden çok daha yüksek bir posterior değere sahiptir. Çünkü likelihood, önceki kelimelerle semantik uyumu yüksek kelimeleri güçlendirmiştir.

Bu sezgiyi matematiksel olarak şöyle ifade edebiliriz:

$$P(w_t \mid w_{1:t-1}) \propto P(w_{1:t-1} \mid w_t)P(w_t)$$

Her ne kadar modern LLM'ler doğrudan Bayes formunu kullanmasa da, bu ifade dil modelinin öngörü mantığını sezgisel olarak yansıtır.

3.2 Markov Zincirleri: Dilin Zamansal Yapısını Yakalamak

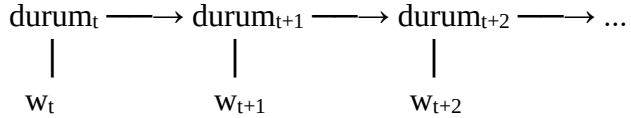
Bayes yöntemleriyle birlikte, Markov zincirleri de dilin modellenmesinde tarihsel olarak kritik bir rol oynamıştır. Markov varsayımının temel fikri, geleceğin yalnızca yakın geçmişe bağlı olduğudur. Basit bir Markov modeli şu şekilde tanımlanır:

$$P(w_t \mid w_{1:t-1}) \approx P(w_t \mid w_{t-1})$$

Bu ifade, dilin zaman içinde ilerleyişini minimal bağlamla anlamaya çalışır. Modern LLM'ler ise bağlamı sonsuz bir pencerede tutarak Markov bağımlılığını çok genişletilmiş bir hâle getirir. Ancak fikir aynı kalır: Dil, birbirini takip eden durumlar zinciri olarak modellenebilir.

Markov süreçleri, dilin yapısındaki geçiş olasılıklarını temsil eder. Geçmişte n-gram modelleri, bu yaklaşımın basit bir uygulamasıydı. LLM'ler, tüm cümleyi ve hatta tüm belgeyi bağlam olarak kabul ederek Markov zincirinin bir genellemesini öğrenir.

Aşağıdaki çizim, dilin Markov benzeri bir şekilde ilerlediğini tasvir eder:



3.3 Markov Zinciri Monte Carlo (MCMC): Zor Posteriorların Hesaplanabilir Hâle Gelmesi

Bayesian çıkarımın en büyük zorluklarından biri, posterior dağılımın genellikle analitik olarak hesaplanamamasıdır. Çünkü çoğu durumda payda olan:

$$P(X) = \int P(X \mid \theta)P(\theta)d\theta$$

hesaplanması imkânsız kadar karmaşık bir integraldir. MCMC yöntemleri, bu integrali kapalı formda çözmek yerine posterior dağılımdan örneklemeye dayanır. Böylece karmaşık dağılımlar, rastgele örneklerle yaklaşık olarak temsil edilebilir.

MCMC'nin dil modellemesiyle ilişkisi ilk bakışta açık görünmeyebilir; fakat LLM'lerin ürettiği metin örnekleme yöntemleri —özellikle Isı örneklemesi (temperature sampling), çekirdek örneklemesi (nucleus sampling) ve en üst k örneklemesi (top-k sampling) MCMC'nin felsefesine oldukça yakındır. Model, olasılık dağılımının tamamını değil, yalnızca en makul bölgelerini örnekleyerek bir sonraki kelimeyi üretir.

MCMC yöntemlerinin temel yapısı şöyledir:

$$\theta_{t+1} \sim q(\theta_{t+1} \mid \theta_t)$$

ve kabul-kuralına dayanan süreçle ilerler:

$$\alpha = \min\left(1, \frac{P(\theta_{t+1} \mid X)}{P(\theta_t \mid X)}\right)$$

Eğer yeni örnek yeterince iyi ise kabul edilir; aksi hâlde eski örnekle devam edilir. Dil modeli üretimi de benzer biçimde çalışır: En olası olmayan kelimeler elenir, yüksek olasılıklılar arasında kontrollü bir rastgelelik sağlanır.

3.4 Örnekleme Yöntemleri: Dil Üzerinde Olasılıkların Dansı

Örnekleme, modern LLM'lerin yaratıcılığının merkezindedir. Deterministik bir argmax seçimi metni sıkıcı ve mekanik hâle getirirken, olasılıksal örnekleme (sampling) cümlelere çeşitlilik, akışkanlık ve doğallık katar.

Temperature sampling, olasılık dağılımının keskinliğini ayarlar:

$$P(w_t = i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

Burada T düşürüldükçe dağılım keskinleşir ve model daha muhafazakâr davranır. T büyütüldükçe dağılım genişler, model daha yaratıcı hâle gelir.

Top-k sampling, yalnızca en olası k kelimenin seçilmesine izin verir ve şu sezgiyi taşır: İnsan da konuşurken pek çok olasılığı zihinsel olarak eleyip yalnızca makul seçenekler arasında karar verir.

Nucleus sampling (p-sampling) ise daha esnek bir yaklaşım sunar:

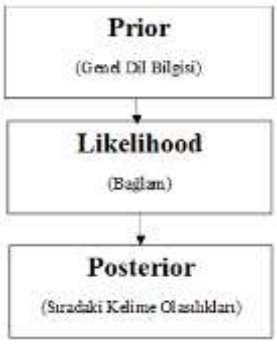
$$\sum_{i \in S_p} P(w_t = i) \geq p$$

Burada S_p, toplam olasılığı p'yi aşan en küçük kelime kümesidir. Bu yöntem, dil üretiminde çeşitliliği olasılıksal bir çekirdek üzerinden kontrol eder.

Bu sampling yöntemleri MCMC ile aynı zihinsel modele sahiptir: Olasılık uzayında rastgele bir yürüyüş yapılır, ancak bu yürüyüş yüksek olasılıklı bölgelerde yoğunlaşır.

3.5 Dilin Bayesian Yorumunun Görsel Çizimi

Aşağıdaki çizim, Bayesian çıkarımın dil üzerindeki etkisini sezgisel olarak açıklar:



Bağlam arttıkça prior'ın etkisi azalır ve likelihood daha baskın hâle gelir; tıpkı gerçek insan konuşmasında olduğu gibi.

4 – EM Algoritması, Gizli Değişkenler, HMM Geleneği ve LLM'lere Giden Yol

Modern büyük dil modellerinin parladığı çağdan önce, doğal dil işleme alanında hüküm süren kavramlardan biri "gizli değişkenli istatistiksel modeller"di. Bu modeller, gözlemlediğimiz verinin arkasında, doğrudan göremediğimiz ama davranışı açıklayan gizli nedenler olduğu varsayımına dayanıyordu. Bir cümledeki kelimeler gözlemlenebilir, ama o cümledeki sözdizimsel yapılar, anlam sınıfları, konu etiketleri çoğu zaman gizlidir. İstatistiksel modeller bu görünmeyen yapıları tahmin etmeye çalışırken, onlara rehberlik eden en önemli yöntemlerden biri, beklenti–maksimizasyon algoritması, yani EM algoritması oldu.

EM, ilk bakışta soyut bir optimizasyon prosedürü gibi görünse de, içeriğinde son derece sezgisel bir hikâye barındırır: Elinde hem görülen hem de görülemeyen bileşenlerden oluşan bir sistem vardır; doğrudan en iyi parametreleri bulamıyorsun, ama "gizli olanı" tahmin edip "görüneni" daha iyi açıklamaya çalışarak, adım adım iyileşebiliyorsun. Dil modellerinin tarihsel evriminde, özellikle gizli Markov modelleri (HMM) ve istatistiksel makine çevirisi için kullanılan IBM modelleri gibi yapılar, EM'in sahnedeki başrol oyuncularıydı. Bugünkü LLM'ler, mimari olarak farklı görünseler de, bu gizli değişkenli düşünme biçiminden doğrudan etkilenmiştir.

4.1 Gizli Değişkenlerin Dünyası

Gizli değişken kavramını, bir tiyatro sahnesi metaforuyla düşünebiliriz. Seyirci, sahnede oynanan oyunu, yani gözlemlenen veriyi görür. Oyuncuların motivasyonları, yönetmenin tercihleri, provalarda alınan kararlar ise sahne arkasındadır; görünmezler ama oyunun gidişatını belirlerler. İstatistiksel bir modelde ise bu sahne arkası, genellikle Zile gösterilen gizli değişkenlerdir. Gözlemlenen veriyi X, modelin parametrelerini θile gösterirsek, sistemin ortak olasılığı şu şekilde yazılır:

$$P(X, Z \mid \theta)$$

Oysa bizim gözümüz yalnızca X' 'i görür. Dolayısıyla ilgilendiğimiz büyüklük, marjinal olasılık olan

$$P(X \mid \theta) = \sum_{Z} P(X, Z \mid \theta)$$

ifadesidir. Ancak bu toplam, özellikle büyük ve karmaşık gizli yapıların olduğu modellerde, pratikte hesaplanamaz hâle gelir. İşte EM algoritması, bu hesaplanamaz marjinal olasılığı

doğrudan optimize etmeye çalışmak yerine, "gizli değişkenler üzerinden dolanıp" çözüm bulmayı önerir.

Dil bağlamında düşündüğümüzde Z, bir cümlenin sözdizimsel ağacı olabilir, kelimelere atanan konu etiketleri olabilir, çeviride hizalanan kelime eşleşmeleri olabilir. Gözlemlenen kelimeler sabittir, fakat onların altında yatan soyut dil yapıları gizlidir.

4.2 EM Algoritmasının Kalbi: Beklenti ve Maksimizasyon

EM algoritmasının mantığını anlamak için, önce "tam veri log-olasılığı" dediğimiz bir ifadeye bakmak gerekir:

$$\log P(X, Z \mid \theta)$$

Eğer hem X hem Z biliniyor olsaydı, parametre tahmini çok daha kolay olacaktı; çünkü tam veri üzerinden klasik maksimum olasılık tahmini (MLE) yapabilirdik. Ne var ki gerçek hayatta Zgözlemlenebilir değildir. EM, bu ikilemi bir tür "hayali tam veri" yaklaşımıyla çözer. Yani şunu söyler: Elimdeki parametre tahmini $\theta^{(old)}$ile, gizli değişkenlerin dağılımını tahmin edebilirim. Bu tahminle, tam veri log-olasılığının beklenen değerini alır, sonra bu beklenen değeri en büyükleyen yeni bir parametre seti $\theta^{(new)}$bulurum.

Bu süreç iki adımda işler. İlk adım, beklenti adımıdır (E-adımı). Burada şu fonksiyonu tanımlarız:

$$Q\left(\theta, \theta^{(\text{old})}\right) = \mathbb{E}_{Z|X,\theta^{(\text{old})}}[\log P(X, Z \mid \theta)]$$

Bu ifade, mevcut parametreler altında gizli değişkenlerin dağılımını kullanarak, tam veri log-olasılığının beklentisini hesaplar. İkinci adım, maksimize etme adımıdır (M-adımı):

$$\theta^{(\text{new})} = \arg\max_{\theta} Q\left(\theta, \theta^{(\text{old})}\right)$$

Bu iki adım, dönüşümlü olarak tekrarlanır. İçsel olarak şu hikâye anlatılır: "Mevcut parametrelerle gizli yapıyı tahmin et; sonra bu tahmin edilmiş gizli yapı üzerinden parametrelerini güncelle; sonra tekrar gizli yapıyı güncelle..." Böylece parametreler, adım adım gözlemlenen veriye daha iyi uyan bir çizgiye doğru sürüklenir.

Bu mekanizmanın arkasında Jensen eşitsizliği yatar. Marjinal log-olasılık

$$\log P(X \mid \theta) = \log \sum_{Z} P(X, Z \mid \theta)$$

doğrudan optimize edilemezken, EM bir alt sınır tanımlar ve bu alt sınırı monoton olarak yükseltir. Böylece her iterasyonda, modelin veriyi açıklama gücü artar veya en azından azalmadığı garanti edilir.

4.3 Gizli Markov Modelleri: Dil İçin EM'in İlk Tiyatrosu

EM'in dil dünyasındaki en tanınmış sahnelerinden biri gizli Markov modelleridir (HMM). HMM'de, gözlemlenen değişkenler genellikle kelimeler ya da semboller, gizli değişkenler ise "durum" dediğimiz, mesela sözcük türleri (isim, fiil, sıfat) veya fonetik birimlerin sınıfları olabilir. HMM'nin yapısı, bir zincir şeklinde tasvir edilir:

$$\begin{array}{ccccccccc} Z_1 & \longrightarrow & Z_2 & \longrightarrow & Z_3 & \longrightarrow & \ldots & \longrightarrow & Z_T \\ \downarrow & & \downarrow & & \downarrow & & & & \downarrow \\ X_1 & & X_2 & & X_{T-1} & & & & X_T \end{array}$$

Bu diyagramda üstteki Z zinciri, gizli durumların zaman içerisindeki gelişimini; alttaki X'ler ise bu gizli durumların ürettiği gözlemleri temsil eder. HMM'nin ortak olasılığı şu biçimdedir:

$$P(X, Z \mid \theta) = P(Z_1) \prod_{t=2}^{T} P(Z_t \mid Z_{t-1}) \prod_{t=1}^{T} P(X_t \mid Z_t)$$

Burada parametreler, başlangıç durum olasılıkları, durum geçiş olasılıkları ve gözlem olasılıklarıdır. Bu parametreleri, yalnızca gözlemlenen Xüzerinden tahmin etmek istediğimizde, EM devreye girer. HMM özelinde kullanılan EM türevi, Baum–Welch algoritması olarak bilinir.

Baum–Welch, E-adımında, ileri–geri (forward–backward) algoritmasını kullanarak her zamandaki gizli durum olasılıklarını ve durum geçişlerinin beklenen sayısını hesaplar. M-adımında ise bu beklenen sayıları kullanarak geçiş ve gözlem olasılıklarını yeniden tahmin eder. Bu döngü, modelin dil verisini daha iyi açıklayabildiği bir parametre setinde duruncaya kadar sürer.

4.4 İstatistiksel Makine Çevirisi, IBM Modelleri ve EM

EM'in dil dünyasındaki bir diğer büyük sahnesi, istatistiksel makine çevirisidir. Nöral makine çevirisi ve LLM tabanlı çeviri sistemlerinden önce, cümleler arasındaki çeviri eşleşmelerini modellemek için IBM modelleri kullanılıyordu. Bu modellerde, bir kaynak dil cümlesindeki kelimelerle hedef dil cümlesindeki kelimeler arasındaki hizalama yapısı gizliydi. Yani hangi İngilizce kelimenin hangi Türkçe kelimeye karşılık geldiğini doğrudan bilmiyorduk; bildiğimiz tek şey, iki dildeki cümle çiftleriydi.

İşte burada, hizalama değişkenleri gizli değişkenler olarak tanımlanır ve EM ile öğrenilir. Ortak olasılık kabaca şu biçimde yazılabilir:

$$P(f, e, a \mid \theta)$$

Burada e kaynak dil cümlesi, f hedef dil cümlesi, a ise kelime hizalama yapısıdır. Marjinal olasılıkta, gizli hizalamalar toplanır:

$$P(f \mid e, \theta) = \sum_{a} P(f, a \mid e, \theta)$$

4.5 EM'den LLM'lere: Kavramsal Köprü

Şimdi doğal bir soru ortaya çıkar: Transformer (dönüştürücü) tabanlı büyük dil modelleri doğrudan EM kullanmıyorsa, EM'in bu uzun hikâyesi LLM'lerle nasıl ilişkilidir? Cevap, kavramsal düzeyde, gizli değişken düşüncesi ve beklenti–güncelleme döngüsünde saklıdır.

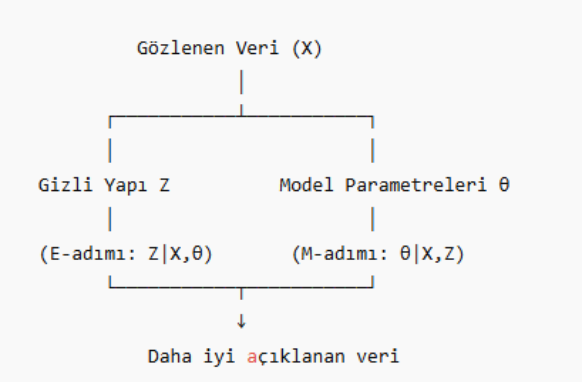
Transformer (dönüştürücü)katmanlarını, bir anlamda gizli temsil güncelleme adımları olarak görebiliriz. Her katmanda, modelin iç durumları olan gizli temsil vektörleri, girişteki ve önceki katmandaki bilgilere bakarak yeniden hesaplanır. Bu, EM'in E-adımı gibi, görünmeyen temsilin güncellenmesidir. Daha sonra geri yayılım (backpropagation), parametreleri güncelleyerek M-adımına benzer bir rol oynar: "Bu gizli temsil güncellemesi daha iyi olsun" diye parametreleri ayarlar. Elbette teknik olarak EM ve gradient descent (gradyan boyut azaltma) farklı iki mekanizmadır; ama genel düşünme biçimi, görünen veriyi açıklamak için gizli bir iç dünya tahmin etmek ve bu iç dünyayı adım adım iyileştirmektir.

Buna ek olarak, modern modellerde kullanılan mixture-of-experts (MoE) (uzman karması) gibi yapılar, tam anlamıyla gizli değişkenli karışım modellerinin yeniden doğmuş hâlidir. Hangi "uzmanın" devreye gireceği, gizli bir seçim değişkeni ile belirlenir. Eğitim sırasında, bu seçim dağılımı öğrenilir ve modelin parametreleri bu gizli karışımı daha iyi kullanacak şekilde

güncellenir. Bu tablo, EM'in karışım modelleri için yaptığı klasik güncelleme döngüsüyle şaşırtıcı derecede benzer bir ruh taşır.

4.6 Görsel Bir Özet: EM'in Dilden LLM'lere Uzanan İzleri

Bu kavramları kafada toparlamak için basit bir çizim düşünebiliriz:



Bu şema, ister HMM eğitiyor olalım, ister IBM modeli, ister MoE'li bir Transformer; temel sezginin benzer olduğunu gösterir: Gizlenen bir yapıyı tahmin eder ve bu yapıyı daha iyi açıklayacak parametreler buluruz.

5 – Boyut İndirgeme, PCA, SVD, Embedding Uzayları ve LLM'lerin Temel Geometrisi

Büyük dil modellerinin yüzeydeki işleyişi kelimeleri, cümleleri ve belgeleri işlemek üzerine kurulu olsa da, bu görünümün altında derin bir geometrik gerçeklik yatar. LLM'lerin gücü, dilin matematiksel bir uzayda temsil edilmesinden doğar. Bu temsil, kelimelerin ve anlamların yüksek boyutlu vektörler hâline dönüştürülmesiyle başlar. Bu vektörleri, dilin içsel anlam yapısının geometrik gölgeleri gibi düşünebiliriz. Modern modellerde bu uzay genellikle 768, 1024 veya 4096 boyutlu olabilir. Ancak dilin anlam yapısı bundan çok daha düşük boyutludur. Boyut indirgeme yöntemleri, bu girift uzayı daha yönetilebilir bir forma sokar ve LLM'lerin semantik yapıları daha verimli işlemesini sağlar.

Bu nedenle PCA (Principal Component Analysis~Temel Bileşenler Analizi) ve SVD (Singular Value Decomposition~Tek Değerli Ayrışma), yalnızca matematiksel araçlar değil, aynı zamanda embedding (gömülü) uzayının nasıl oluştuğunu ve LLM'lerin eğitilirken nasıl bir iç geometri kazandığını anlamamız için anahtar kavramlardır. Bu bölümde, bu geometrik yapıları adım adım açmadan önce, yüksek boyutluluğun dil için neden hem bir yük hem de bir fırsat olduğunu görerek başlayalım.

5.1 Dilin Yüksek Boyutlara Doğru Açılan Kapısı

Doğal dilin karmaşıklığı, kelimelerin ve cümlelerin çok sayıda farklı bağlamda çok farklı anlamlar kazanmasından kaynaklanır. Örneğin "sepet" kelimesi hem finansal bir enstrüman setini hem de sazdan yapılan taşıma kabını ifade edebilir. Bu tür çok anlamlılık, tek bir boyutta temsil edilemeyecek kadar zengindir. Dolayısıyla embedding'lerin yüksek boyutlu uzaylarda bulunması, bu nüansları temsil etmenin bir zorunluluğudur.

Bununla birlikte, yüksek boyutlu uzayların kendine özgü bir tuhaflığı vardır: Noktalar birbirine eşit derecede uzakta olma eğilimindedir. Bu boyutluluk laneti (curse of dimensionality) olarak bilinen olgudur. Dil uzayını makul şekilde öğrenebilmek için, bu yüksek boyutlu karmaşıklığı belirli bir altta yatan düşük boyutlu yapıya indirgemek gerekir. PCA ve SVD bu ihtiyacın doğal sonucu olarak ortaya çıkar.

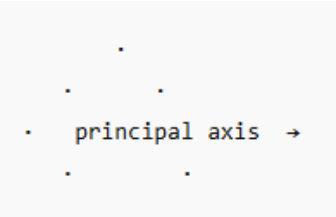
5.2 PCA: Anlamın Ana Eksenlerini Bulmak

Principal Component Analysis (PCA), verinin en fazla değiştiği yönleri bulmaya çalışan bir teknik olarak düşünülebilir. Bu yönler, dilin semantik açıdan en baskın örüntülerini içerir. PCA'nın matematiksel tanımı şu şekildedir:

$$\text{maximize } w^{\top}\Sigma w \qquad \text{durumunda} \qquad \| w \| = 1$$

Burada Σ, embedding matrisinin kovaryansıdır. Bu ifade, vektörlerin varyansı en yüksek olduğu yönü bulmaya çalışır. PCA yalnızca bir boyut indirgeme yöntemi değildir; aynı zamanda anlamın baskın yönlerini ortaya çıkaran bir mercek gibidir. Embedding uzayındaki ilk birkaç principal component genellikle cinsiyet, çoğulluk, zaman veya semantik alan gibi geniş dil ilişkilerine karşılık gelir.

PCA'nın geometrik yorumu, aşağıdaki gibi basit bir çizimde görülebilir:



Bu eksen, verinin en çok yayıldığı yönü temsil eder. Dil modellerinde bu yayılma, anlam çeşitliliğinin en baskın paternlerine karşılık gelir.

5.3 SVD: Embedding Uzayının Gizli Yapısını Ayrıştırmak

PCA'nın ardındaki matematiksel işlem aslında tekil değer ayrışımıdır (SVD). Bir metin veri kümesini ifade etmek için kullandığımız embedding matrisi X, SVD ile şu şekilde ayrışır:

$$X = U\Sigma V^{\top}$$

Bu ayrışımda Udoküman vektörlerini, Vkelime vektörlerini, Σise yapının "enerji" dağılımını temsil eden tekil değerleri ifade eder. Eğer embedding matrisi devasa bir uzayda gereksiz karmaşıklık taşıyorsa, düşük tekil değerlere karşılık gelen bileşenleri atarak daha kompakt bir uzay oluşturabiliriz.

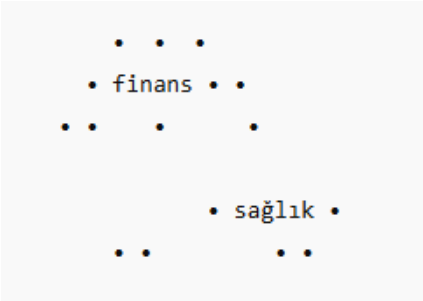
Daha da önemlisi, SVD dil modellemenin tarihsel köklerinde yer alır. Word2Vec'in negatif örnekleme (negative sampling) yöntemiyle üretilmiş embedding'lerinin aslında bir SVD'nin düşük dereceli bir yaklaşımı olduğu gösterilmiştir (Levy & Goldberg, 2014). Dolayısıyla LLM'lerde kullandığımız modern embedding uzayları, bu tarihsel temelin doğal bir devamıdır. SVD'nin geometrik yorumunu şöyle düşünebiliriz:

X (yüksek boyutlu)
↓ SVD
U —— Σ —— V^{T}
↓ düşük rütbeli yapı
X_r (indirgenmiş anlam uzayı)

5.4 Embedding Uzayının Geometrisi: Anlamın Dört Bir Yanı

Dil modelleri kelimeleri yalnızca semboller olarak değil, anlam kümeleri olarak işler. Bu kümeler embedding uzayında çoğu zaman öbeklenmiş hâlde bulunur. Semantik olarak benzer kelimelerin birbirine yakın olması, modelin benzer cümle yapılarında bu kelimeleri kolayca birbirinin yerine kullanabilmesini sağlar.

Aşağıdaki soyut çizim bu kümelenmeyi gözümüzde canlandırabilir:



Bu kümeler arasında vektörel ilişkiler vardır. Word2Vec ile popülerleşen şu klasik analoji, embedding uzayının vektörel doğasını temsil eder:

$$v(\text{kral}) - v(\text{adam}) + v(\text{kadın}) \approx v(\text{kraliçe})$$

Bu tür ilişkiler yalnızca bir istatistiksel tesadüf değildir; embedding uzayının doğrusal yapısının bir sonucudur. Bu doğrusal yapı SVD'nin sağladığı matris ayrışımının dil uzayına yansımasıdır.

5.5 Yüksek Boyutlu Uzaylarda Mesafe, Açı ve Benzerlik

LLM'ler, kelimeler arasındaki benzerliği genellikle kosinüs benzerliğiyle ölçer:

$$\cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|}$$

Kosinüs benzerliği, vektörlerin aynı yönde olup olmadığını ölçer. Yüksek boyutlu embedding uzayında, mesafeden çok açı önemlidir; çünkü noktaların mutlak mesafeleri yüksek boyutlarda anlamını kaybeder. Bu nedenle semantik yakınlık, iki vektörün yönsel benzerliğinde saklıdır.

Transformer mimarisindeki self-attention (kendi kendine bakım) mekanizmasının kalbi olan scaled dot-product attention (Ölçülmüş sonuç üretimi bakımı) tam olarak bu ilişkiyi kullanır:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V$$

Burada Q ve K arasındaki iç çarpım, kelimeler arasındaki benzerliği temsil eden bir geometrik ölçüdür. Yani Transformer'ın anlam bulma mekanizması, embedding uzayının geometrisi üzerine kuruludur.

5.6 LLM'lerde Boyutun Estetiği

LLM'ler ilk bakışta devasa boyutlarıyla korkutucudur; yüz milyarlarca parametre, binlerce boyutlu embedding alanları, karmaşık transformasyonlar… Fakat işin derininde, bu karmaşa daha basit bir yapının tekrarlı ve genişletilmiş bir versiyonudur: Dilin anlam bileşenlerini yüksek boyutlu ama yapılandırılmış bir uzaya yerleştirmek. Bu uzayın estetiği, PCA ve SVD gibi yöntemlerin ortaya çıkardığı düşük boyutluluğun altında yatan düzenle ilgilidir. Dilin içsel düzeni vardır ve boyut indirgeme yöntemleri bu düzeni sezgisel olarak açığa çıkarır.

Yüksek boyutlar, anlamı taşımak için gereklidir; ama aşırı yüksek boyutlar modelin kavrama yeteneğini bozabilir. Bu nedenle LLM'ler, embedding boyutlarını dikkatli seçer; 256 boyut çok sığ kalırken, 4096 boyutun ötesi modelin "düzleşmesine" neden olabilir. Bu hassas denge, aslında PCA'nın ve SVD'nin bize uzun yıllar önce öğrettiği bir gerçeği yansıtır: Boyut, taşıdığı varyans kadar anlamlıdır.

6 – Olasılıksal Grafik Modeller, Bağımlılık Yapıları ve Transformer'ın İçsel Graf Mantığı

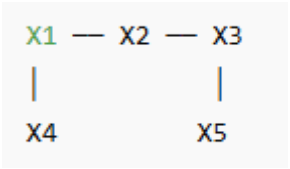
Büyük dil modelleri hakkında konuşurken genellikle "sinir ağı", "katman sayısı", "parametre sayısı" gibi kavramlara odaklanırız; ama bu modellerin başka bir yüzü daha vardır: Onlar, aslında dev birer olasılıksal grafın, öğrenilmiş ve sürekli güncellenen nüshaları gibidir. Olasılıksal grafik modeller, rastgele değişkenler arasındaki bağımlılık ve bağımsızlık ilişkilerini çizgiler ve düğümler üzerinden temsil eden yapılar olarak, LLM'lerin ne yaptığını sezgisel olarak anlamamız için güçlü bir dil sunar. Dil, kelimeler arasındaki ilişkilerin bir ağ içinde dolaştığı karmaşık bir yapıysa; grafik modeller bu ağın matematiksel haritasıdır.

Burada önce olasılıksal grafik modellerin temel kavramlarını, ardından bu kavramların dil modellemesiyle nasıl temas ettiğini, son olarak da Transformer'ın self-attention mekanizmasının, aslında dinamik ve veriyle belirlenen bir graf yapısı gibi düşünülebileceğini ele alacağız.

6.1 Grafik Modellerin Temel Fikri: Bağımlılıkların Haritası

Olasılıksal grafik modellerin çıkış noktası oldukça basittir: Çok sayıda rastgele değişkeniniz varsa ve hepsinin birbiriyle doğrudan ilişki içinde olduğunu varsayarsanız, bu değişkenlerin ortak dağılımını modellemek neredeyse imkânsız hâle gelir. Ancak çoğu gerçek sistemde, her şeyin her şeyle doğrudan ilişkili olması gerekmez; bazı değişkenler, diğerlerinden bağımsız olabilir ya da yalnızca sınırlı sayıda komşusuna bağlı olabilir. Grafik modeller, işte bu "yerel bağımlılık" fikrini çizgiler ve düğümler üzerinden ifade eder.

Basit bir gösterimle, her rastgele değişkeni bir düğümle, aralarındaki koşullu bağımlılıkları ise kenarlarla gösteririz:



Bu küçük şema bile, hangi değişkenlerin birbirine doğrudan bağlı olduğunu, hangilerinin yalnızca dolaylı yollarla etkileştiğini sezdirir. Dil uzayında bu düğümler kelimeler, sözdizimsel etiketler, anlam kategorileri ya da gizli temsil vektörleri olabilir.

Grafik modellerin gücü, ortak dağılımı parçalara ayıran faktörizasyon yapısından gelir. Örneğin yönlendirilmiş bir graf modelinde, her düğüm yalnızca ebeveynlerine bağlıdır ve ortak olasılık şöyle yazılır:

$$P(X_1, \dots, X_n) = \prod_{i=1}^{n} P\big(X_i \mid Pa(X_i)\big)$$

Burada $Pa(X_i)$, graf üzerindeki ebeveyn düğümlerini ifade eder. Bu faktörizasyon, hem hesaplamayı kolaylaştırır hem de modelin temsil gücünü daha anlaşılır kılar.

6.2 Yönlendirilmiş ve Yönlendirilmemiş Yapılar: Nedensellik ve Simetri

Grafik modeller iki ana aileye ayrılır: yönlendirilmiş (Bayesian ağlar) ve yönlendirilmemiş (Markov rastgele alanları). Yönlendirilmiş grafiklerde oklar, belirli bir koşullu bağımlılık yönünü, çoğu zaman nedensel bir yorumu ima eder. Örneğin

$$A \rightarrow B \rightarrow C$$

gibi bir yapı, C'nin B'ye, B'nin ise A'ya bağlı olduğunu, ama C'nin doğrudan A'dan etkilenmediğini (B üzerinden dolaylı etkilendiğini) ima eder.

Dil açısından bakarsak, örneğin bir fiilin zaman kipinin, cümlenin genel zaman çerçevesine bağlı olması ama bazı koşullarda özne seçiminden doğrudan etkilenmemesi gibi ince bağımlılık ilişkilerini bu tür ağlar üzerinden düşünebiliriz.

Yönlendirilmemiş grafiklerde ise kenarlar, simetrik ilişkileri ifade eder ve ortak dağılım genellikle potansiyel fonksiyonlarla yazılır:

$$P(X) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C)$$

Burada $\mathcal{C}$, grafın klik kümelerini, $\psi_C$ise o klik üzerinde tanımlı potansiyel fonksiyonları temsil eder. Dil modelleme bağlamında bu potansiyeller, belirli kelime gruplarının birlikte ortaya çıkma eğilimlerini kodlayan, daha esnek ama daha zor eğitimli yapılar olarak düşünülebilir.

6.3 Grafik Modeller ve Dil: Bağımlılıkların Cümle İçinde Akışı

Doğal dil, grafik modeller için neredeyse ideal bir oyun alanıdır. Her cümlede, kelimeler arasında karmaşık bir bağımlılık zenginliği bulunur. Bazı kelimeler, diğerlerini neredeyse belirler; bazıları ise yalnızca dolaylı anlam etkileri yaratır. Örneğin "eğer" kelimesi, ardından gelecek koşullu bir yapıyı çağırır; "çünkü" kelimesi neden-sonuç bağlantısı kurar.

Geleneksel istatistiksel dil modellemede, bu bağımlılıklar çoğu zaman zincir şeklinde (Markov zincirleri, n-gram modelleri) temsil edilmiştir. Ancak daha gelişmiş grafik modeller, bir cümledeki bağımlılıkların yalnızca sol–sağ yönünde değil, tüm cümle boyunca çapraz referanslarla örüldüğünü yakalamaya çalışmıştır. Örneğin bağımlılık çözümlemesi (dependency parsing), her kelimeyi diğerleriyle ilişkilendiren ağaç yapıları üretir:

```
      ROOT
       |
      fiil
      /  \
   özne  nesne
```

Bu tür ağaçlar, bir nevi grafik modelin özel bir hâlidir. Transformer mimarisi ortaya çıktığında, bu tür yapıları açıkça tanımlamasa da, attention (özen~ilgi~bakım) mekanizması aracılığıyla benzer ilişki ağlarını "öğrenir" hâle geldi.

6.4 Transformer'ın Self-Attention'ı: Dinamik Bir Graf Olarak Okunması

Transformer'ın kalbinde yer alan self-attention mekanizması, her token'ın, dizideki diğer tüm token'larla etkileşime girmesine izin verir. Matematiksel formül basittir ama derin bir anlam taşır:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Burada Q, Kve Vmatrisleri, sırasıyla sorgu (query), anahtar (key) ve değer (value) vektörlerini temsil eder. $QK^\top$çarpımı, her token'ın diğer token'larla ne kadar ilişkili olduğunu ölçen bir benzerlik matrisi üretir. Bu matrisin softmax ile normalize edilmesi, her satırı olasılık dağılımına dönüştürür. Böylece her kelime, diğer kelimeler üzerine tanımlı bir olasılık dağılımı üzerinden ağırlıklı ortalama alır.

Bu yapıyı grafiksel bir gözle okursak, her attention başlığı, cümle üzerindeki olası bir yönlendirilmiş graf için ağırlıklı kenarları öğrenen bir mekanizma olarak görülebilir. Örneğin bir attention başlığı, özne–fiil ilişkilerine; bir diğeri, zamir–referans ilişkilerine; bir başkası ise zamansal bağlara odaklanabilir. Bu durumda, self-attention'ın ürettiği matrisi, bir grafın komşuluk matrisi gibi düşünebiliriz:

```
Tokenlar:  t1   t2   t3   t4

Attention ağırlıkları (t2 için):
t1 — 0.1
t2 — 0.4   (kendi üzerine)
t3 — 0.3
t4 — 0.2
```

Bu tabloyu, t2 düğümünden diğer düğümlere giden yönlendirilmiş ve ağırlıklı kenarlar olarak yorumlayabiliriz. Böyle bakınca Transformer, her ileri geçişte, dizinin üzerinde yeni bir olasılıksal graf örüyor ve bu graf üzerinden bilgi yayılımı sağlıyor.

6.5 Faktörizasyon, Koşullu Bağımsızlık ve Attention'ın Rolü

Grafik modellerin en önemli avantajlarından biri, koşullu bağımsızlık ilişkilerini net biçimde ifade etmesidir. Örneğin bir Bayesian ağında, belirli bir düğüm, ebeveynleri verildiğinde bazı diğer düğümlerden bağımsız olabilir. Bu bağımsızlık ilişkileri, hem hesaplamayı hızlandırır hem de modelin yorumlanabilirliğini artırır.

Transformer'da ise, teoride her token (Gösterge~simge) her token'a bağlıdır ve tam anlamıyla "yoğun" bir graf yapısı ortaya çıkar. Ancak pratikte, attention ağırlıkları bu yoğun grafın içinden seyrek ve anlamlı alt-graplar seçer. Yüksek ağırlık alan kenarlar, güçlü bağımlılıkları; düşük ağırlık alanlar ise zayıf, çoğu zaman ihmal edilebilir ilişkileri temsil eder. Böylece model, dilin içindeki önemli bağlantıları vurgulayan, önemsizleri ise perdeleyen bir faktörizasyon mekanizmasına sahip olur.

Bu durumu, klasik faktörizasyon formülü ile sezgisel bir şekilde ilişkilendirebiliriz. Dil modelinin ortak olasılığını:

$$P(w_1, \dots, w_T) = \prod_{t=1}^{T} P(w_t \mid w_{<t})$$

şeklinde yazdığımızda, aslında her bir koşullu dağılımın, attention yoluyla belirlenen bir alt bağlama indirgenmiş olduğunu hatırlamak gerekir. Yani model, geçmişin tamamını değil, geçmişteki "ilişkili" noktaları daha çok dikkate alır. Attention, bu ilişkililiği belirleyen veri güdümlü bir graf çıkarımı olarak görülebilir.

6.6 Grafiksel Yorumun Dil Anlayışına Katkısı

Transformer ve LLM'leri grafik modeller açısından okumak, yalnızca teorik bir oyun değildir; pratikte de model içi açıklanabilirlik için oldukça önemlidir. Örneğin bir cümledeki belirli bir tahminin hangi kelimelerden ne kadar etkilendiğini, attention ağırlıkları üzerinden görselleştirebiliriz. Bu görselleştirme, aslında grafın belli bir düğümünden çıkan kenar ağırlıklarını izlemekten başka bir şey değildir.

Basit bir örnek olarak, "Doktor ameliyathaneye girdi çünkü hasta kritikti." cümlesinde, "kritikti" kelimesinin anlamını çözümlerken modelin "hasta" ve "ameliyathaneye" kelimelerine

yüksek attention vermesi, semantik ağın bu üç düğüm arasında yoğunlaştığını gösterir. Bu durum, grafik model dilinin bize sunduğu "odaklanmış alt-yapı" kavramına çok benzer: Büyük ağ içinde küçük, anlam yüklü alt grafikler.

7 – Optimizasyon, Kayıp Fonksiyonları, Gradient (düğüm~değişme~irtifa~meyil)Akışı ve LLM Eğitim Dinamikleri

Büyük dil modellerinin yüzeyde gördüğümüz etkileyici yeteneği, aslında arkada işleyen dev bir optimizasyon sürecinin ürünüdür. Bir LLM, dildeki anlam ve yapıyı "ezberlemez"; bunun yerine, devasa miktardaki veriye bakarak, hangi parametre kombinasyonunun bu veriyi en iyi açıklayacağını bulmaya çalışır. Bu arayış, yüksek boyutlu bir uzayda kayıp fonksiyonunun en alçak noktalarını keşfetme çabasıdır. Bu bölümde, dil modellerinin neden "öğrenebildiğini", gradient descent'in (boyut azaltma) neden işe yaradığını ve optimizasyon yüzeylerinin neden düşünüldüğünden daha karmaşık ama bir o kadar da düzenli olduğunu ele alacağız.

7.1 Kayıp Fonksiyonları: Dilin Matematiksel Ekonomisi

Bir LLM'in öğrendiği her şey, bir kayıp fonksiyonunun minimize edilmesine dayanır. Eğitim sırasında en sık kullanılan kayıp fonksiyonu çapraz entropi (cross-entropy)'dir. Dil modelleme bağlamında, modelin tahmin ettiği olasılık dağılımı $p_\theta(w_t \mid w_{<t})$ile gerçek kelimenin tekil dağılımı arasındaki mesafeyi ölçer. Matematiksel olarak:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log p_\theta(w_t \mid w_{<t})$$

Bu ifade, modelin doğru kelimeye yeterince yüksek olasılık verip vermediğini ölçer. Eğitim boyunca bu kayıp milyonlarca, hatta yüz milyonlarca örnek üzerinden hesaplanır. Kayıp fonksiyonunun görevi sadece modelin hata yapıp yapmadığını söylemek değildir; aynı zamanda, doğru yönde güncellenmesi için gerekli olan ince gradyan dokunuşlarını da oluşturur. Benzetme yapmak gerekirse, kayıp fonksiyonu dilin öğretmeni, gradient ise öğretmenin verdiği geri bildirim, parametreler ise öğrencinin bir sonraki derse hazırlanmak için yaptığı düzeltmelerdir.

7.2 Gradyan Azaltma: Derin Öğrenmenin Atar Damarı

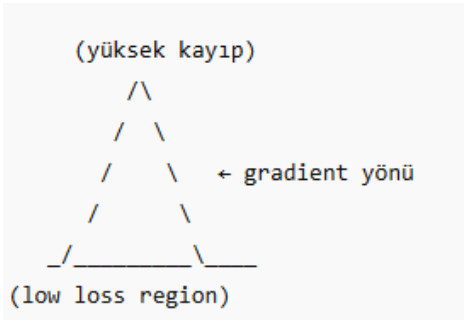
Bir sinir ağının öğrenebilmesi için parametrelerinin değiştirilmesi gerekir. Gradient descent (GD), bu değişikliğin yönünü ve miktarını belirler. Eğer kayıp fonksiyonunu bir dağlık arazi gibi düşünürsek, gradient descent bu dağın en alçak noktasını bulmaya çalışan bir gezgin gibidir. Dağın eğimi, gezginin nereye doğru adım atması gerektiğini söyler.

Bir parametre güncellemesi şu şekilde ifade edilir:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t)$$

Burada η, öğrenme hızıdır; çok büyük seçilirse model dağın bir yamacından diğer yamacına savrulur, çok küçük seçilirse model ilerleyemez. Bu hassas denge, LLM eğitiminde uzun yıllardır büyük bir araştırma konusudur.

Gradient descent'in sezgisel gücünü anlamak için, kayıp yüzeyini iki boyutlu bir çizimle gözümüzde canlandırabiliriz:



Bu yüzey, LLM parametrelerinin içinde dolaştığı milyarlarca boyutlu bir arazinin sadece küçük bir iki boyutlu kesitidir.

7.3 Stokastik (Tesadüfi) Gradient Descent (SGD): Verinin Gürültüsünü Avantaja Çevirmek

Gerçek dünyada, tüm veri üzerinden kayıp ve gradient hesaplamak çok pahalıdır. Bu nedenle SGD, yani mini-batch gradient descent (mini yığın~parti boyut azaltma), eğitim sürecinin fiili standardıdır. Her adımda veri kümesinden küçük bir örnek grubu alınır ve gradient yalnızca bu örnekler üzerinden hesaplanır.

Bu yöntem, iki önemli avantaj sağlar:

1. Gereksiz hesaplamayı azaltır ve eğitimi hızlandırır.
2. Gradient'e biraz "gürültü" ekler; bu gürültü modelin kayıp yüzeyinde yerel minimumlara takılmasını engelleyebilir.

Bu sezgiyi şöyle bir çizimle düşünebiliriz:

deterministic GD → düz ve kararlı iniş stochastic GD → zikzaklı ama daha serbest hareket

LLM'lerin devasa boyutlu kayıp yüzeyleri düşünüldüğünde, SGD'nin bu rastlantısallığı modelin etkili öğrenmesini sağlayan unsurlardan biridir.

7.4 Kayıp Yüzeyleri: Düşündüğümüzden Daha Düz, Ama Yine de Kaygan

İlk bakışta, milyarlarca parametreli bir modelin kayıp yüzeyi hayal edilemeyecek kadar karmaşık olmalıymış gibi gelir. Ancak matematiksel bulgular, yüksek boyutlu uzayların şaşırtıcı bir şekilde daha düzenli olduğunu göstermiştir. Özellikle geniş ağlarda, kötü yerel minimumların neredeyse hiç olmadığı, kayıp yüzeylerinin nispeten düz olduğu görülmüştür.

Bu durumu daha somut görmek için, literatürde sıkça verilen bir kesit çizimini düşünebiliriz:

Bu yüzey, keskin bir çukurdan çok, düz bir vadinin içinden geçen geniş bir inişe benzer. LLM'lerde "flat-minima" (düz minimumlar) olarak bilinen bu bölgeler, modelin daha iyi genelleme yapabildiği yerlerdir. Keskin minimumlar aşırı uyum riskini temsil ederken, düz olanlar modelin farklı girişlerde benzer davranmasını sağlar.

7.5 Momentum, Adam ve Optimizasyonun Modern Yüzü

Klasik gradient descent yeterli değildir. Çünkü yüksek boyutlu yüzeylerde, gradient çok dalgalı olabilir, bazı yönlerde çok hızlı düşerken bazı yönlerde neredeyse düz olabilir. Bu nedenle momentum temelli yöntemler geliştirilmiştir.

Momentum yaklaşımı, geçmiş gradient'leri akılda tutarak parametre güncellemesini daha pürüzsüz hâle getirir:

$$v_{t+1} = \beta v_t + (1-\beta)\nabla_\theta \mathcal{L}(\theta)$$
$$\theta_{t+1} = \theta_t - \eta v_{t+1}$$

Adam algoritması ise momentum ile RMSProp'un birleşmiş hâlidir ve modern LLM eğitiminde sıklıkla kullanılır:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Adam algoritmasının başarısının ardında, gradient'in yönünü ve ölçeğini ayrı ayrı takip etmesi yatar. Dil modellemenin karmaşık parametre uzayında, bu adaptif güncellemeler kritik önem taşır.

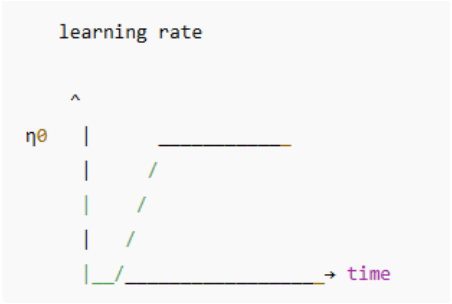
7.6 Eğitim Dinamikleri: Öğrenme Hızının Zamanla Azaltılması ve Müfredat Öğrenmesi (Curriculum Learning)

Eğitim boyunca öğrenme hızını sabit tutmak, çoğu zaman etkili değildir. Genellikle şu yaklaşım izlenir:

$$\eta_t = \eta_0 \cdot f(t)$$

Burada $f(t)$, zamanla azalan bir fonksiyondur. Kosinüs çürümesi (Cosine decay), doğrusal çürüme) (linear decay, Kızışma+Çürüme (warmup + decay) gibi stratejiler LLM eğitiminde standart hâline gelmiştir.

Warmup özellikle önemlidir. Eğitimin ilk adımlarında gradient'ler çok dengesiz olabilir. Bu nedenle öğrenme hızı yavaşça artırılır:



Müfredat öğrenmesi (Curriculum learning) ise modelin önce basit görevleri, sonra karmaşık olanları öğrenmesini sağlar; tıpkı bir öğrencinin önce temel kavramları öğrenip sonra soyut matematiğe geçmesi gibidir.

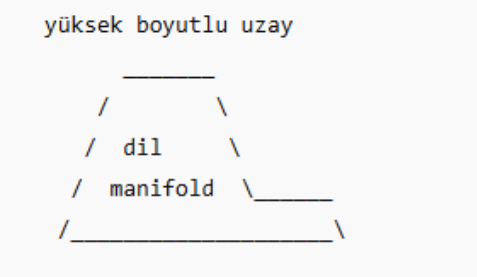
7.7 Genelleme: LLM'ler Neden "Ezberlemiyor" da Gerçekten Öğreniyor?

Derin öğrenmenin en ilginç yönlerinden biri, modelin kapasitesi çok büyük olsa bile, doğru eğitimle aşırı uyum yapmamasıdır. LLM'ler trilyonlarca kelime görür ve milyarlarca parametre içerir; ama bu parametreler veriyi birebir ezberlemek yerine, dilin genelleştirilmiş bir temsilini öğrenir.

Bu durumu açıklamak için teoride üç ana argüman kullanılır:

1. **Flat minima → iyi genelleme**
2. Model, keskin minimum yerine geniş ve düz bir çözüme yerleşir.
3. **Implicit regularization (örtük düzenleme)**
4. SGD ve Adam gibi algoritmalar, farkında olmadan modelin daha düzgün bir çözüm bulmasını sağlar.
5. **Dilin doğal yapısının düşük boyutluluğu**
6. Trilyon kelime bile, dilin içsel manifoldunun yalnızca yüzeyini temsil eder; model bu manifoldun şeklini öğrenir.

Bu sezgiyi şöyle bir manifold çizimiyle düşünebiliriz:



Model, bu manifold (çok çeşitlilik) üzerinde etkili bir projeksiyon öğrenir.

8 – Transformer Mimarisi: Katmanlar, Attention, Pozisyonal Kodlama (Positional Encoding) ve İstatistiksel Yorum

Büyük dil modellerinin kalbinde, artık neredeyse ikonik hâle gelmiş bir mimari vardır: Transformer. "Attention is all you need" başlıklı makaleyle ortaya çıkan bu yapı, dil modellemesinde istatistiksel düşünceyle lineer cebir'in buluştuğu bir kesişim noktası gibidir. Yüzeyde gördüğümüz şey, üst üste yığılmış katmanlar, self-attention blokları, residual (kalıntı) bağlantılar ve normalizasyon katmanlarından oluşan uzun bir zincir. Fakat bu zincirin her halkası, istatistiksel bir anlam taşır.

Transformer'a uzaktan baktığımızda, girişte token dizisini alan ve çıkışta her token için yeni bir temsil üreten bir fonksiyon görürüz. Bu fonksiyon aslında, bağlam koşullu bir olasılık dağılımını yaklaşık olarak öğrenmeye çalışır. Her katman, bu dağılımı daha rafine bir hâle getiren bir istatistiksel operatör gibi düşünülebilir.

8.1 Girdi Temsili: Token'lardan Sürekli Vektörlere

Bir cümlenin model tarafından işlenebilmesi için, önce kelimelerin (ya da alt birim token'ların) birer vektöre dönüştürülmesi gerekir. Bu süreçte her token, sözlükteki benzersiz kimliğini temsil eden bir indeksle başlar. Ardından bu indeks, embedding matrisine bakılarak sürekli bir vektörle eşleştirilir. Eğer embedding matrisini $E \in \mathbb{R}^{V \times d}$ile gösterirsek, burada V sözlük büyüklüğü, d ise embedding boyutudur. Bir token'ın vektörü basitçe:

$$x_t = E_{w_t}$$

şeklinde yazılabilir. Bu, istatistiksel açıdan, kategorik bir değişkeni sürekli bir uzayda kodlayan bir dönüşümdür. Bu vektörlerin tümü bir araya geldiğinde, cümlenin ilk temsilini oluşmuş olur. Ancak bu temsilin eksik bir yanı vardır: Sıra bilgisi. Yani model, "kedi köpeği kovaladı" ile

"köpek kediyi kovaladı" cümlelerini aynı çoklu-küme gibi görebilir. Bu problemi çözmek için positional encoding (pozisyon kodlama) devreye girer.

8.2 Positional Encoding (pozisyon kodlama): Sıranın Matematiksel İzini Bırakmak

Transformer mimarisinin en radikal taraflarından biri, RNN ve LSTM'lerden farklı olarak, sıralı işlemi bir zorunluluk hâline getirmemesidir. Tüm token'lar aynı anda işlenebilir; fakat bunun için modelin pozisyon bilgisini başka bir yoldan edinmesi gerekir. Positional encoding, tam da bu noktada devreye girer. Girdinin her bir pozisyonuna, konumu temsil eden bir vektör eklenir.

Vaswani ve arkadaşlarının önerdiği sinüs–kosinüs tabanlı positional encoding formülü şu şekildedir:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

Burada pospozisyonu, i ise boyut indeksini temsil eder. Bu fonksiyonlar, farklı frekanslarda salınımlar üreterek pozisyonun sürekli bir temsilini oluşturur. Bu yaklaşım, dilin sıralı yapısını Fourier benzeri bir tabana projekte etmek gibi düşünülebilir; her pozisyon, birden çok frekans bileşeninin birleşimiyle kodlanır. İstatistiksel açıdan bakıldığında, bu kodlama, sıranın kendisini rastgele bir etiket olarak değil, düzenli ve ölçülebilir bir özellik olarak modellemenin yoludur.

Sonuçta, girişteki her token, içerik bilgisini taşıyan embedding vektörü ile pozisyon bilgisini taşıyan positional encoding vektörünün toplamından oluşan bir temsil kazanır:

$$h_t^{(0)} = x_t + PE_t$$

Bu $h_t^{(0)}$vektörleri, Transformer'ın derin katmanlarına giren ilk istatistiksel öznelerdir.

8.3 Self-Attention: Koşullu Dağılımın İç Çarpım Üzerinden Değerlendirilmesi

Transformer'ın ruhu, self-attention mekanizmasıdır. Bu mekanizmayı soyut bir istatistiksel operatör olarak düşünmek oldukça faydalıdır. Her token, diğer token'lara olan ilgisini ölçer; bu ilgi, lineer dönüşümler ve iç çarpımlar üzerinden sayısallaştırılır.

Her token için üç farklı vektör hesaplanır: query (q_t), (sorgu) key (k_t) (anahtar) ve value (v_t) (değer) . Bunlar, giriş temsilinin farklı ağırlık matrisleriyle çarpılmasıyla elde edilir:

$$q_t = W_Q h_t, k_t = W_K h_t, v_t = W_V h_t$$

Buradaki W_Q, W_K, W_Vmatrisleri, öğrenilen parametrelerdir. Bütün token'lar için hesaplandığında, Q, K, Vmatrisleri oluşur. Attention skorları, query–key (sorgu-anahtar) iç çarpımlarıyla hesaplanır:

$$\alpha_{t,j} = \frac{q_t \cdot k_j}{\sqrt{d_k}}$$

Bu skorlar, bir normalizasyon adımıyla olasılık dağılımına dönüştürülür:

$$a_{t,j} = \frac{\exp(\alpha_{t,j})}{\sum_l \exp(\alpha_{t,l})}$$

Ve nihayet, her token'ın yeni temsili, diğer tüm token'ların value vektörlerinin ağırlıklı ortalaması olarak elde edilir:

$$h_t{}' = \sum_j \quad a_{t,j} v_j$$

Bu işlemi istatistiksel bir bakışla okuduğumuzda, self-attention'ın, bir token'ın koşullu dağılımını, diğer token'ların "gizli özelliklerine" göre güncelleyen bir kernel operatörü gibi davrandığını görürüz. $q_t$ve $k_j$arasındaki iç çarpım, bu iki token arasındaki benzerliği ölçer; softmax ise bu benzerlikleri birer olasılık ağırlığına dönüştürür. Sonuç, koşullu bir beklenen değer hesabıdır: "Bağlamdaki diğer token'ları, bana ne kadar benzediğinize göre tartıyorum ve yeni temsilimi bu tartılmış ortalama üzerinden güncelliyorum."

8.4 Çok Kafalı Attention: Farklı İstatistiksel Bakış Açıları

Tek bir attention başlığı, dizideki ilişkileri belirli bir projeksiyon uzayında analiz eder. Ancak dil, tek bir bakış açısıyla yakalanamayacak kadar çok katmanlıdır. Çok kafalı (multi-head) attention, bu nedenle devreye girer. Farklı başlıklar, W_Q, W_K, W_Vmatrislerinin farklı kopyalarını kullanarak, aynı girdiyi farklı lineer alt uzaylara projekte ederler. Her başlık için ayrı ayrı attention uygulanır ve sonunda bu başlıkların çıktıları birleştirilir.

Bunu şu şekilde yazabiliriz:

$$\text{head}_i = \text{Attention}(QW_Q^{(i)}, KW_K^{(i)}, VW_V^{(i)})$$
$$\text{MultiHead}(Q,K,V) = [\text{head}_1; \ldots; \text{head}_h]W_O$$

Her başlık, dilin farklı bir istatistiksel özelliğine odaklanabilir: biri uzun menzilli bağımlılıkları, biri sözdizimsel ilişkileri, biri anlam benzerliklerini, bir diğeri zaman–kip bağlamlarını yakalayabilir. Bu yapı, istatistiksel modellemede "karışım modelleri" veya "çoklu çekirdek" kullanımıyla benzer bir mantık taşır. Aynı veriye farklı çekirdekler uygulayıp, sonuçları bir üst uzayda birleştirmek gibidir.

8.5 Residual (kalıntı) Bağlantılar ve Layer (katman) Normalizasyon: Öğrenmenin Stabilizasyonu

Transformer'ın her katmanında yalnızca attention yoktur; attention'ın ardından gelen bir feed-forward ağı, ardından residual bağlantı ve normalizasyon katmanları da bulunur. Bu bileşenler, yalnızca mühendislik ayrıntıları değil, aynı zamanda istatistiksel kararlılık araçlarıdır.

Residual bağlantılar, giriş vektörünü, katmanın çıktısına ekler:

$$\tilde{h}_t = h_t^{in} + \text{SubLayer}(h_t^{in})$$

Bu yapı, gradyanların kaybolmasını engeller ve modelin derinliğini artırırken eğitim dengesini korur. İstatistiksel açıdan bakarsak, residual bağlantı, her katmanda yapılan dönüşümü bir "düzeltme terimi" gibi ele alır; her yeni katman, önceki tahmine küçük ama anlamlı bir güncelleme ekler. Sanki model, "mevcut tahminimi tamamen çöpe atmıyorum, yalnızca üzerine bir düzeltme yazıyorum" demektedir.

Layer normalization ise her token'ın temsil vektörünü, boyutları boyunca normalize eder. Bir vektör h için:

$$\text{LayerNorm}(h) = \frac{h-\mu}{\sigma} \odot \gamma + \beta$$

Burada μ ve σ, vektör bileşenlerinin ortalaması ve standart sapması; γ ve β ise öğrenilebilir ölçek ve kaydırma parametreleridir. Bu normalizasyon, dağılımı istatistiksel olarak daha kararlı hâle getirir; gradyanların aşırı dalgalanmasını engeller ve eğitim sürecini hızlandırır. Yani LayerNorm, parametre uzayının belli bölgelerinde kaybolmamızı önleyen bir düzenleyici gibi çalışır.

8.6 Feed-Forward (İleri Besleme) Katmanlar: Yerel, Ama Güçlü Dönüşümler

Her attention bloğunun ardından gelen konum-bağımsız feed-forward ağları, her token'ın temsilini kendi içinde dönüştürür. Basit bir yapı ile:

$$\mathrm{FFN}(h) = W_2\sigma(W_1 h + b_1) + b_2$$

Bu dönüşüm, istatistiksel öğrenme açısından doğrusal olmayan regresyonun bir formudur. Attention, kelimeler arasındaki ilişkileri hesaplarken, feed-forward ağları aynı kelimenin içsel özelliklerini zenginleştirir. İkisi bir araya geldiğinde, hem bağlamsal hem yerel bir istatistiksel model elde ederiz.

8.7 Transformer'ı Büyük Resimde Okumak

Tüm bu bileşenler bir araya geldiğinde, Transformer'ın her katmanını şu şekilde düşünebiliriz: Önce attention ile dizideki istatistiksel bağımlılıklar hesaplanır, sonra bu bağımlılıklar feed-forward ağlarıyla işlenir, residual ve normalizasyon ile stabil hâle getirilir ve bir sonraki katmana aktarılır. Katmanlar arttıkça, temsil giderek daha soyut, daha geniş bir bağlamı kapsayan, daha karmaşık istatistiksel özetler hâlini alır.

Bu sürecin graf düzeyinde bir tasviri, zihinde şöyle canlanabilir:

```
Girdi vektörleri  →  Attention grafı  →  Yerel dönüşümler  →  Normalize edilmiş yeni uzay
        ↓                   ↓                   ↓
    Katman 1            Katman 2            Katman L
```

Her katman, dilin istatistiksel yapısının başka bir boyutunu yakalayan bir filtre gibi çalışır.

9 – LLM Uygulama Alanları: İstatistiksel Temeller, Kavramsal Boyut ve Modelin "Alan Anlayışı"

Büyük dil modelleri, yalnızca metin üreten sistemler değildir; aynı zamanda istatistiksel bilgi işleme makineleridir. Bir hukuk metniyle tıbbi bir raporun aynı model tarafından işlenebilmesi, dilsel yüzeyin gerisinde, istatistiksel bir yapıların bütünlüğü olduğuna işaret eder. Bu yapıların merkezinde, kelimelerin ve kavramların yüksek boyutlu embedding uzayında modellenen benzerlik ilişkileri vardır. Fakat uygulama alanları birbirinden keskin şekilde farklılaştıkça, bu ilişkilerin formu da değişir. Örneğin sağlık alanında "akut", "kronik", "prognosis", "metastaz" gibi terimlerin semantik uzaklıkları tıbbi kavramlar etrafında kümelenir; hukuk metinlerinde ise "mülkiyet", "kast", "takdir yetkisi", "yargı denetimi" gibi terimler kendi kategorizasyon ağları içinde yerleşir.

LLM'lerin çok yönlülüğünün ardındaki temel istatistiksel gerçek, "ortak embedding manifoldunun bölgesel uzmanlaşmaya izin vermesi"dir. Anlam uzayı sürekli olduğundan, bir alanın kavramları başka bir alanın kavramlarından tamamen ayrı değildir; yalnızca manifoldun farklı bölgelerinde toplanırlar. Dilin çok alanlı kullanımını mümkün kılan şey de budur.

Bu noktada dil modelleri, ham sembollerle çalışan sistemler olmaktan çıkar; belirli alanların bilgi dağılımlarını öğrenmiş istatistiksel yapılar hâline gelir. Modelin gördüğü metinler, alan

uzmanlığının dağılımını içerir. Bu dağılım, kayıp fonksiyonunun içinde saklıdır. Eğitimin her adımında model, alanın "bilgi yüzeyini" biraz daha iyi tahmin etmeyi öğrenir. Bunu şu istatistiksel ifade üzerinden anlayabiliriz:

$$\theta^* = \arg\min_{\theta} \ \mathbb{E}_{X\sim\mathcal{D}}[-\log p_\theta(X)]$$

Burada $\mathcal{D}$, dağılımı "sağlık", "finans", "hukuk", "bilimsel yayınlar" vb. alanlara yayılan bir veri kümesidir. Model, tüm bu alanların ortak dağılımını yaklaşık öğrenir. Bu nedenle model, tek bir alanın uzmanı değildir; çoklu alanların ortak manifoldunu yakalamaya çalışan meta-öğrenici bir sistemdir.

Bu yetenek, özellikle karmaşık bağlam bağımlılıklarında kendisini gösterir. Örneğin finans alanında bir metinde geçen "kaldıraç" kelimesi, sağlık alanında geçen "kaldırma kuvveti" kavramından tamamen farklı bir semantik alana işaret eder. Transformer'ın çok kafalı attention mekanizması, bu çoklu bağlamları ayrıştırır; kafalardan biri finansal bağlamı öğrenirken, bir başkası fiziksel bağlamı öğrenebilir. Böylece model, bağlamı tanımayı bir istatistiksel sınıflandırma probleme dönüştürmeden, dağılım içi varyasyon olarak işler.

Embedding Uzayında Alan Ayrışması

LLM'lerin eğitiminde ortaya çıkan ilginç fenomenlerden biri, embedding uzayında alanların doğal olarak ayrışmasıdır. Sağlık terimleri bir araya kümelenir, finansal terimler başka bir bölgede yoğunlaşır, hukuki kavramlar ise farklı bir alt uzayda konumlanır. Bu ayrışma, PCA veya t-SNE gibi yöntemlerle görselleştirildiğinde daha belirgin hâle gelir.

Basit bir çizimle:

Bu kümeler birbirine uzak görünse de, manifoldun tamamı bağlantılı bir yüzey oluşturur. Bu durum, modelin bir alandan diğerine transfer edebilme kapasitesini açıklar. Bir hukuk metninde geçen tıbbi bir terim, modelin tıbbi manifold bölgesini aktive eder; ama cümlenin geri kalanı hukuk manifoldunda kalmaya devam eder.

Bu esneklik, LLM'lerin çok alanlı uygulamalarını mümkün kılar.

Alan Uzmanlığının İstatistiksel Doğası

Birkaç yıl öncesine kadar "alan modeli" fikri, bir yapay zekânın belirli bir disipline özel olarak eğitilmesi anlamına geliyordu. Bugün ise alan uzmanlığı, embedding manifoldunun belirli bölgelerinin daha keskin, daha yüksek çözünürlüklü hâle gelmesiyle oluşuyor. Bu durum, şu istatistiksel metaforla açıklanabilir:

- Genel bir LLM = geniş ama düşük çözünürlüklü bir anlam manifoldudur.
- Bir alana ince-tuning (fine-tuning) = manifoldun belli bir bölgesinin daha yüksek çözünürlükle işlenmesidir.
- Uzman model = manifoldun yalnızca bir bölgesini temsil eden dar ama ultra-yoğun bir modeldir.

Bu farkı matematiksel bir benzetmeyle göstermek istersek, genel LLM'in tahmin ettiği dağılım:

$$p_\theta(w \mid c)$$

iken, alan ince ayarından sonra tahmin edilen dağılım:

$$p_{\theta'}(w \mid c, A)$$

şeklindedir. Burada A, alan bilgisini temsil eder. Bu, koşullu bir dağılımın alt dağılıma dönüşmesi gibidir.

Bağlam-Uygulama Ayrışmasının Önemi

Bir LLM'in uygulama başarısı, yalnızca verinin içeriğine değil, bağlamın istatistiksel işlenişine bağlıdır. Aynı kavram, farklı bağlamlarda farklı koşullu dağılımlar üretir. Örneğin "Tedavi başarılı oldu." cümlesi sağlık alanında yüksek olumlu bir sonuç bildirirken, finans alanında "Tedbir başarılı oldu." ifadesi yatırım stratejisinin etkili olduğunu anlatır.

Bu iki bağlamın ayırt edilebilmesi, attention'ın bağlam seçici yapısından gelir. Self-attention'ın matematiksel formülünde:

$$a_{t,j} = \text{softmax}(\frac{q_t k_j^\top}{\sqrt{d_k}})$$

bağlam q_t'nin değerleri, alan bilgisini otomatik olarak açığa çıkarır. Böylece model, alan odaklı bir filtreleme işlemini istatistiksel olarak eğitilmiş parametrelerle gerçekleştirir.

Modelin Alanlar Arası Genelleme Yeteneği Neden Bu Kadar Güçlü?

İki temel sebep var:

1. **Dil alanlar arası paylaşılan kavramsal yapılar içerir.**

   Mantık, nedensellik, tanımsal ilişkiler, kavramların hiyerarşisi alanlar arasında ortak geometri oluşturur.

2. **Model çok büyük ve çok çeşitli veriyle eğitilmiştir.**

   Bu veri, dilin tüm yüzeysel varyasyonlarının ortak dağılımını öğrenmesine imkân tanır. Böylece model, bir alandaki ifadeleri başka bir alandaki kavramların bağlamıyla bile ilişkilendirebilir. Buna benzer bir fenomen, istatistiksel fizikteki "genelleşmiş simetriler" kavramına benzetilebilir; farklı yüzeylerdeki davranışlar aynı temel kurallarla yönetilir.

Sağlık, Finans, Hukuk ve Bilimsel Metinlerde LLM Davranışı: Alan İstatistiği ve Örnek Vaka Analizleri

Büyük dil modelleri, farklı alanlara yayılan metinleri işlerken aslında aynı çekirdek istatistiksel mekanizmaları kullanır, fakat bu mekanizmaların yüzeye yansıması alanın doğasına göre değişir. Sağlık alanında kullanılan dil, yüksek riskli kararlarla iç içedir; finans metinlerinde belirsizlik ve gelecek tahmini ön plandadır; hukukta normatif yapı, yorum ve içtihat ağı hakimdir; bilimsel metinlerde ise kanıtın gücü, deneysel tasarım ve istatistiksel testler dilin içine gömülmüştür. LLM bu alanların her birinde aynı matematiksel kalple çalışır, ancak öğrendiği dağılımlar farklılaşır.

Sağlık alanından başlayalım. Klinik metinler, tanılar, laboratuvar sonuçları, tedavi planları ve prognoz değerlendirmeleri gibi bileşenlerden oluşur. Bir LLM, bu tür metinlerle eğitildiğinde, aslında örtük olarak bir tür koşullu olasılık modeli öğrenir. Örneğin bir hastanın semptom

vektörünü x, olası tanıyı $y$ile gösterirsek, model dolaylı biçimde $p_\theta(y \mid x)$dağılımına yaklaşır. Burada klasik istatistikten alışkın olduğumuz duyarlılık (sensitivity) ve özgüllük (specificity) gibi kavramlar devreye girer. Bir tanı modelinin duyarlılığı

$$\text{Sensitivite} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

ve özgüllüğü

$$\text{Spesifisite} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

şeklinde tanımlanır. LLM doğrudan bu oranları hesaplamasa bile, ürettiği önerilerin doğruluğu aynı yapıdaki sayımlar üzerinden değerlendirilebilir. Bir klinik LLM, örneğin şüpheli bir akciğer grafisi için "yüksek olasılıkla pnömoni" dediğinde, aslında posterior bir olasılık tahmini yapar. Bu tahminin kalitesi, geleneksel istatistikteki ROC eğrileri ve AUC değerleri ile ölçülebilir. ROC eğrisinde, modelin eşik değerini değiştirirken, doğru pozitif oranıyla yanlış pozitif oranının izlediği eğri, LLM'nin tanı kararlarının ne kadar ayrışabilir olduğunu gösterir. Bu tür değerlendirmeler, sağlık alanında LLM kullanımının yalnızca "dil" değil, aynı zamanda saf istatistiksel performans meselesi olduğunu gösterir.

Finans alanında ise dil, sayılarla birlikte akmaktadır. Risk raporları, bilanço analizleri, piyasa yorumları, araştırma notları gibi metinler, hem deterministik hesaplar hem de olasılıksal senaryolar içerir. LLM'ler bu metinleri işlerken genellikle iki düzeyde çalışır: yüzeyde doğal dil üretir, derinde ise risk ve beklenti kavramlarını içeren bir istatistiksel dünya kurar. Bir finansal enstrümanın beklenen getirisi

$$\mathbb{E}[R] = \sum_i \quad p_i r_i$$

şeklinde yazılırken, varyans ve kovaryans yapıları portföy riskini belirler. Örneğin iki varlık için kovaryans

$$\text{Cov}(R_1, R_2) = \mathbb{E}[(R_1 - \mu_1)(R_2 - \mu_2)]$$

ifadesiyle verilir. LLM bu formülleri bilmek zorunda değildir; ancak finans literatürünün içinde büyümüşse, metinlerden öğrenmiş olduğu ilişkiler bu kavramlarla uyumlu bir dağılım davranışı üretir. Bir raporda geçen "volatilite artışı", "aşağı yönlü risk", "korelasyon çöküşü" gibi ifadelerin embedding uzayında oluşturduğu kümelenmeler, aslında risk kavramının semantik manifold içindeki geometrisini temsil eder. Böylece model, yalnızca "cümle kurmaktan" çok, finans dünyasının istatistiksel sezgilerini yeniden ifade etmeye başlar.

Hukuk alanı, LLM'ler için biraz farklı bir meydan okumadır. Çünkü burada yalnızca olgular yoktur; aynı zamanda normatif değerlendirmeler, kural yorumları, içtihat zincirleri ve çoğu zaman belirsizliğin dil üzerinden yönetildiği uzun metinler vardır. Bir mahkeme kararını düşündüğümüzde, olgular kümesini F, hukuk normlarını N, yorum kurallarını ise Iile gösterip, kararın sonucunu Colarak düşünebiliriz. Teorik olarak

$$C \sim f(F, N, I)$$

şeklinde soyut bir fonksiyon yazmak mümkündür. LLM, büyük miktarda karar metniyle karşılaştığında, bu fonksiyonun istatistiksel bir yaklaşıkçısı hâline gelir. Her yeni dava anlatımı, yeni bir Fvektörüdür; karar, C'nin alanındaki bir noktadır; normlar ve içtihatlar ise metinler içinde gömülü birer bilgi kaynağıdır. Model, bu ilişkileri doğrudan formülize etmese de, dil içindeki tekrarlardan öğrenir. "Davacının talebinin reddine" dair kalıplar, belirli olgusal desenlerle birlikte görülmeye başlandığında, LLM bu desenleri içsel olarak modeller.

Bu noktada hukuki metinler için olasılıksal bir bakış açısı devreye girer. LLM, belirli bir olgu seti Fverildiğinde, belirli bir karar türünü c üretme eğilimindedir. Bunu şu şekilde yazabiliriz:

$$p_\theta(C = c \mid F, \text{bağlam})$$

Bu dağılım, katı deterministik bir hukuk anlayışından uzak, ama pratik hukuki uygulamanın belirsizliğini yansıtan bir modeldir. Bu nedenle LLM'lerin hukuk alanında kullanımı, yalnızca metin özetleme ya da tematik sınıflandırma değildir; aynı zamanda, hukukun istatistiksel yüzünün bir yansıması hâline gelir. Elbette burada etik ve normatif riskler de büyüktür; bunlara bir sonraki bölümde döneceğiz.

Bilimsel metinler ve akademik literatür, LLM'lerin belki de en çok çekildiği alanlardan biridir. Çünkü bu metinler, zaten baştan sona istatistiksel kavramlarla örülüdür. Bir klinik araştırma raporunda p-değeri, güven aralığı, etki büyüklüğü; bir fizik makalesinde hata çubukları, ölçüm belirsizlikleri; bir psikoloji çalışmasında varyans analizi ve regresyon modelleri dolaşır. LLM, böyle metinlerle eğitildiğinde, adeta istatistiksel düşüncenin dil içindeki izlerini takip etmeyi öğrenir. Örneğin bir hipotez testini ele alalım. Boş hipotez H_0, alternatif hipotez H_1ve test istatistiği $T(X)$ile temsil edildiğinde, p-değeri

$$p = P(T(X) \geq T(x_{\text{obs}}) \mid H_0)$$

şeklinde tanımlanır. LLM bu formülü bilmek zorunda değildir; ancak binlerce makalede aynı yapı ile karşılaştığında, "düşük p-değeri" ifadesinin "istatistiksel olarak anlamlı" kararına bağlandığını, "güven aralığı 0'ı içermiyorsa" ifadesinin belirli sonuçlarla beraber göründüğünü öğrenir. Bu, düşüncenin istatistiksel kalıplarının dil üzerinden modele sızmasıdır.

Bilimsel alanlarda LLM kullanımı, yalnızca metin üretimi ile sınırlı kalmaz. Model, deney tasarımı önerileri, veri analizi adımlarının planlanması, olası yanlılık kaynaklarının listelenmesi gibi daha derin seviyelerde de rol oynayabilir. Bir deneyde karıştırıcı değişkenleri ($confounders$) belirlemek, aslında bir tür grafik model problemi olarak görülebilir; model, metinden yola çıkarak hangi değişkenlerin hangilerini etkileyebileceğine dair sezgisel bir graf çıkarır. Bu graf, formel olmasa bile, istatistiksel düşünceye oldukça yakındır.

Tüm bu alanlarda ortak bir motif belirir: LLM, dilin içinde gömülü istatistiği emer. Sağlıkta tanı olasılıkları, finansda risk dağılımları, hukukta olgu–sonuç ilişkileri, bilimde hipotez testleri; hepsi dilsel kalıplar üzerinden modele taşınır. Model, bu kalıpların frekansını, bağlamlarını, birlikte görünme biçimlerini öğrenir ve sonunda, sadece grameri değil, alanların istatistiksel reflekslerini de içselleştirmiş bir yapı hâline gelir.

Uygulama Alanlarında İleri Seviye Davranış Analizi, Vaka Örnekleri ve İstatistiksel Hata Dinamikleri

Sağlık, finans, hukuk ve bilimsel metinlerde LLM davranışı yalnızca kavramların anlamıyla değil, aynı zamanda bu kavramların birbirleriyle kurduğu istatistiksel ilişkilerle belirlenir. Bir

modelin belirli bir vakayı nasıl yorumladığı, kelimelerin embedding uzayındaki konumlarından, attention başlıklarının bağlamı nasıl ağırlıklandırdığına kadar geniş bir mekanizma örgüsünün ürünüdür. Bu nedenle uygulama alanlarında model performansını değerlendirirken, yüzeysel metriklerden çok daha derine inmek gerekir. Model, bazen bir kelimeyi yanlış anlamaz; o kelimenin istatistiksel konumunu yanlış okur. Bu bölümde bu tür karmaşık olayların iç işleyişini inceleyeceğiz.

Önce sağlık alanında tipik bir senaryo düşünelim. Bir klinik raporda şu cümle yer alsın: "Hastanın karın bölgesinde 48 saat içinde ilerleyen lokalize ağrı mevcut olup, bulgular apandisit ile uyumludur." İyi eğitilmiş bir LLM, burada "ilerleyen ağrı", "lokalize", "apandisit ile uyumlu" gibi medikal ipuçlarını kullanarak tanıyı doğru yönde tahmin eder. Fakat model, aynı cümle içinde geçen başka bir terimin—örneğin "gaz şikâyeti"—fazla ağırlık alması hâlinde, olasılık dağılımı yanlış bir semantik yönelime kayabilir. Bu olay, attention mekanizmasının istatistiksel dengesizliğinden kaynaklanır: belirli token'ların key vektörleri, modelin eğitim verisindeki dağılım nedeniyle gereğinden fazla "çekici" olabilir. Eğer modelin eğitiminde gaz şikâyeti sıkça benign (tehlikesiz) durumlarla eşleşmişse, posterior olasılık dağılımı şu şekilde değişebilir:

$$p(\text{akut apandisit} \mid x) \rightarrow p(\text{gaz distansiyonu} \mid x)$$

Bu, salt kelime–kelime eşleşmesi değil, embedding uzayındaki semantik yakınlıkların öncelik kazanmasından kaynaklanır. Klinik raporların düzensiz yapısı, kısaltmalar, terminolojideki değişkenlik ve bağlamsal atlamalar bu istatistiksel dalgalanmayı büyütebilir. Bu nedenle sağlık modellerinde çok sık kullanılan bir yaklaşım, özel eğitilmiş domain-adaptive (alan uyarlamalı) fine-tuning (ileri besleme) ve yapısal regularization yöntemleridir; modelin karar uzayının belirli bölgelerinin aşırı aktif olmasını engeller. İstatistiksel olarak bu durum, posterior'u yeniden şekillendiren bir prior eklemek gibidir.

Finans alanında bir örnek ele alalım. Bir analist raporunda "Likidite riski kısa vadede arttı, ancak bilanço yapısı güçlü olduğu için uzun vadeli temerrüt riski düşüktür." ifadesi geçsin. Bu cümlede kısa vadeli risk yükselirken uzun vadeli risk düşmektedir; yani metin iki farklı zaman ufkunda iki farklı olasılık dağılımını ima eder. LLM'ler çoğu zaman bu tür çok katmanlı risk ifadelerini ağırlıklandırmakta zorlanır. Çünkü embedding uzayında "risk artışı" ve "risk azalması" terimleri genellikle güçlü, karşıt semantik vektörler üretir ve model bu karşıtlığı tek bir eksen üzerinde işlemeye eğilimlidir. Oysa burada iki eksenli bir yapı vardır: zaman ufku ve risk yönü. Bu durum matematiksel olarak şöyle temsil edilebilir:

$$p(\text{risk} \mid \text{kısa}) \neq p(\text{risk} \mid \text{uzun})$$

Fakat LLM, çoğu durumda bu iki koşullu dağılımı tek bir gölge dağılımda birleştirir:

$$p_\theta(\text{risk} \mid \text{bağlam})$$

Bu gölge dağılım, bazen kısa vadeli sinyalleri uzun vadeli bağlamlara taşır. Bu istatistiksel karışma, finans modellerinde sık görülen yanlış çıkarımların kaynağıdır. Örneğin model, kısa vadeli volatilite artışını tüm ufuklara genelleyebilir. Bu, risk modellemede "volatilite klasterleşmesi" fenomenine benzer bir biçimde, embedding klasterleşmesinin hatalı bir yansımasıdır. Bu nedenle finans LLM'lerinde bağlam ayrıştırıcı attention maskeleme teknikleri kullanılmaya başlanmıştır; kısa vadeli ifadelerin uzun vadeli bağlam ağırlıklarını bozmasını engeller.

Hukuk alanında hata dinamikleri çok daha inceliklidir. Bir dava özetinde "Sanığın kastı bulunmadığından ceza tayin edilmemiştir." cümlesi geçtiğini düşünelim. Model, burada "kastın bulunmaması" ile "ceza tayin edilmemesi" arasındaki hukuki nedenselliği doğru yakalayabilir; fakat aynı bağlamda şöyle bir ifade yer alıyorsa—"Bu nedenle maddi tazminat talebi reddedilmiştir."—model bazen kast–ceza ilişkisini maddi tazminata da genelleyebilir. Oysa ceza hukuku kast ile ilgilidir; maddi tazminat ise özel hukuk sorumluluğudur ve çoğu zaman kusur kavramıyla ilişkilidir. LLM bu ayrımı her zaman içselleştirmez. Bu hata, istatistiksel olarak "yanlış nedensel bağımlılık öğrenme" problemiyle örtüşür. Bir olgu kümesi F, iki farklı sonuç kümesi C_1(ceza hukuku) ve C_2(özel hukuk) ile ilişkili olabilir:

$$p(C_1 \mid F) \neq p(C_2 \mid F)$$

Fakat model, eğitiminde bu iki ilişkiyi ayrı ayrı görmediğinde, tek bir birleşik dağılım öğrenir:

$$p_\theta(C \mid F)$$

Bu durum, hukuki bağlamların ayrışmasını engeller ve LLM'nin karar tavsiyelerinde "alanlar arası aşırı genelleme" denilen bir fenomen yaratır. Bu fenomen hem etik hem hukuki açıdan kritik bir risk oluşturur; çünkü yanlış alana ait normatif sonuçlar tahmin edilirse, model yanılgısı gerçek hayatta ciddi karar hatalarına yol açabilir.

Bilimsel alanlarda ise modelin hataları genellikle metodolojik kavramların embedding uzayındaki yakınlık ilişkilerinden doğar. Örneğin "korelasyon" ve "nedensellik" kavramlarını ele alalım. Bu iki kavram semantik olarak yakın görünür, ancak istatistiksel olarak ayrıdır. Bir çalışmada "iki değişken arasında güçlü bir korelasyon bulundu" ifadesi, LLM tarafından bazen "A, B'ye neden oluyor olabilir" şeklinde yorumlanabilir; çünkü metinlerde korelasyon bulgusunun çoğu zaman nedensel spekülasyonlarla bir arada geçtiği örneklerle karşılaşmıştır. Bu durumda model, şu yanlış çıkarımı yapmaya eğilimli olabilir:

$$\mathrm{Corr}(X, Y) \Rightarrow \mathrm{Caus}(X \rightarrow Y)$$

Oysa istatistikte bilindiği gibi, korelasyon şu şekilde tanımlanır:

$$\rho_{XY} = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

ve hiçbir şekilde nedenselliği garanti etmez. Model bu farkı sezgisel olarak ayırt edemediğinde, bilimsel yazı üretiminde hatalı çıkarımlar yapabilir. Bu nedenle bilimsel LLM uygulamalarında, nedensellik ve korelasyon kavramları embedding uzayında özel olarak ayrıştırılır; bu ayrıştırma genellikle kontrastif öğrenme teknikleri ile yapılır. İstatistiksel olarak bu işlem, iki kavramın vektör dağılımlarını farklı modal manifoltlara itmek gibidir.

Bu hataların tümü, LLM'lerin "dil alanlarının istatistiksel haritalarını" nasıl öğrendiğiyle ilgilidir. Sağlık, finans, hukuk ve bilimsel disiplinler, doğal dilin içine gömülü farklı istatistiksel ilişkiler taşır. Bir LLM bu ilişkileri doğru öğrenirse alan uzmanlığı gösterir; yanlış öğrenirse alan-özel hatalar üretir. Bu nedenle model değerlendirmesi yalnızca doğruluk, F1 veya BLEU gibi yüzeysel metriklerle değil, alan istatistiğinin iç yapısını hedefleyen daha karmaşık yöntemlerle yapılmalıdır.

Uygulama Alanlarında İleri Davranış Analizi

Uygulama alanlarını konuşurken şimdiye kadar hep metin merkezli örnekler üzerinden yürüdük; fakat modern LLM ekosisteminde metin artık çoğu zaman tek veri formu değil. Sağlıkta görüntüleme verileri, sensör çıktıları ve yapılandırılmış elektronik sağlık kayıtları; finansta zaman serileri, fiyat akışları ve işlem defterleri; hukukta şema hâline getirilmiş

mevzuat ağları ve içtihat grafikleri; bilimsel dünyada ise deneysel tablolar, grafikleri ve şekillerle birlikte anılıyor. LLM'ler bu farklı modalitelerle ilişki kurmaya başladığında, istatistiksel davranışları da yeni bir boyuta taşınıyor.

Örneğin çok modlu bir klinik sistemde metin, görüntü ve laboratuvar verisi aynı anda işleniyor olsun. Metin tarafında bir LLM, görüntü tarafında bir konvolüsyonel ağ veya transformer tabanlı görsel model, sayısal tarafta ise klasik istatistiksel özetler (ortalama, varyans, z-puanları) devreye giriyor. Bu durumda model aslında ortak bir latent uzayda birleşmiş üç farklı dağılımı öğrenmeye çalışıyor. Metin için öğrenilen temsil vektörlerini $h^{(text)}$, görüntü için $h^{(img)}$, sayısal bulgular için $h^{(lab)}$ile gösterirsek, çok modlu bir füzyon katmanı bu temsilleri birleştirip tek bir birleşik temsil *z*üretiyor:

$$z = f(h^{(text)}, h^{(img)}, h^{(lab)})$$

Bu birleşik temsil, tanı ya da risk tahmini gibi bir çıktıya bağlandığında, artık tek bir modalitenin hataları değil, tüm modalitelerin ortak istatistiksel dengesizlikleri rol oynamaya başlıyor. Örneğin görüntü verisinde eğitimin büyük kısmı belirli bir demografiye aitse, metin tarafında bu demografi az temsil edilmiş bile olsa, birleşik model belirli gruplar üzerinde sistematik yanlılık üretebilir. Bu, çok modlu domain shift'in tipik bir örneğidir.

Domain shift, yani eğitim dağılımı ile kullanım anındaki dağılımın farklılaşması, LLM'lerin tüm alanlardaki en kritik istatistiksel problemlerinden biridir. Teorik çerçevede bunu basitçe şöyle yazabiliriz: Eğitim sırasında gözlemlenen dağılım $P_{\text{train}}(X, Y)$, kullanım sırasında karşılaşılan dağılımdan $P_{\text{test}}(X, Y)$farklıdır. En sık rastlanan biçimi, kovaryat kayması denen durumdur:

$$P_{\text{train}}(X) \neq P_{\text{test}}(X), P(Y \mid X) \text{ yaklaşık sabit}$$

Bu, özellikle finans ve sağlık alanında çok tipiktir. Pandemi başladığında önceki yılların hasta profilleriyle yeni vaka profilleri arasındaki fark, bir gecede tüm klinik modelleri domain shift ile yüz yüze bırakmıştır. Benzer şekilde, finansal kriz günlerinde normal dönemlerde hiç görülmeyen fiyat hareketleri, eğitilmiş modellerin tahmin uzayının dışına düşer. LLM'ler de bu tür kaymalarda, eğitimde hiç görmedikleri bağlam kombinasyonlarını üretmek zorunda kalır ve bazen istatistiksel sezgilerini tamamen kaybedebilir.

Domain shift'in daha karmaşık bir türü, etiket dağılımının kaydığı label shift durumudur:

$$P_{\text{train}}(Y) \neq P_{\text{test}}(Y)$$

Hukuk alanında örneğin belirli bir dönemde açılan dava türlerinin dağılımı değiştiğinde, geçmiş içtihat üzerinden eğitilmiş model, yeni dönemin ağırlıklarını yansıtamaz. Bir anda idari davaların oranı artmışsa, eski karar metinlerine dayanan bir LLM, idari yargıdaki yeni eğilimleri doğru yorumlayamayabilir. Bu, modelin "tarihsel prior"larının güncel dağılımla uyumsuz hâle gelmesi anlamına gelir.

Domain adaptasyonu bu noktada devreye girer. Modern LLM pratiklerinde genellikle iki tür adaptasyon kullanılır: alan uyumlu ön-eğitim (domain-adaptive pretraining) ve ince ayarlı görev eğitimi (fine-tuning). Alan uyumlu ön-eğitimde model, genel dil veri kümesinden sonra sadece belirli alanın metinleri üzerinde ek bir dil modelleme aşamasına tabi tutulur. Bu,

parametrelerin o alanın tipik istatistiksel kalıplarına doğru kaydırılması anlamına gelir. Matematiksel olarak, modelin önce genel dağılımı

$$p_\theta(x)$$

öğrendiğini, ardından alan dağılımına göre bir güncelleme yaptığını düşünebiliriz:

$$p_{\theta'}(x) \propto p_\theta(x) \cdot w_A(x)$$

Burada $w_A(x)$, alan A'ya ait örnekleri ağırlıklandıran bir terim gibi düşünülebilir; teknik olarak doğrudan böyle uygulanmasa da, sezgisel olarak alan eğitimi posterioru alan yönünde günceller. Fine-tuning ise belirli bir göreve koşullu dağılımı öğrenir. Örneğin sağlık alanında "metinden ICD kodu tahmini" gibi bir görevde model, artık

$$p_{\theta''}(y \mid x, A)$$

şeklinde daha dar bir dağılımı optimize eder.

Bu adaptasyon süreçlerinde LoRA gibi düşük dereceli adaptasyon tekniklerinin giderek öne çıkması, istatistiksel açıdan şunu anlatır: Modelin tüm ağırlıklarını değiştirmek yerine, yalnızca düşük rütbeli bir alt uzayı güncellemek çoğu zaman yeterlidir. Büyük dil modelinin çekirdeği sabit kalırken, alan-özel küçük matrisler eklenir. Sembolik olarak şöyle yazılabilir:

$$W_{\text{yeni}} = W_{\text{eski}} + AB^\top$$

Burada Ave Bküçük boyutlu matrislerdir. Bu yapı, lineer cebir açısından düşük rütbeli bir düzeltme, istatistiksel açıdan ise genel prior üzerine alan-özel bir posterior düzeltmesi gibidir. Böylece model, hem genel dili hem de alanın özgün dilini taşıyabilir.

Graf yapılarının entegrasyonu da uygulama alanlarında giderek önem kazanıyor. Özellikle hukuk ve bilimsel metinlerde, kavramlar arasında açıkça tanımlanmış ilişkiler vardır: mevzuat maddeleri başka maddelere atıf yapar, makaleler birbirini kaynakça üzerinden ilişkilendirir. Bu ilişkileri grafikler hâlinde temsil etmek mümkündür. LLM, metni işlerken bu grafiklerden gelen yapısal sinyalleri de kullanırsa, artık yalnızca sıradaki kelimeyi tahmin eden bir model değil, aynı zamanda bilgi ağında gezinmesini bilen bir yürüyücü hâline gelir. Rastgele yürüyüş (random walk) kavramı burada ön plana çıkar:

$$P(v_{t+1} = j \mid v_t = i) = \frac{w_{ij}}{\sum_k \quad w_{ik}}$$

Bu geçiş olasılıkları temsili, bir makaleden diğerine, bir içtihattan diğerine atlayan bir gezginin davranışını tanımlar. Bir LLM bu yürüyüşü metin üretimi ile birleştirdiğinde, bazen metnin içinde gizli bir graf üzerinde gezinir; atıflara, referanslara ve kavramsal bağlantılara göre içerik üretir. Bu, özellikle bilimsel özetleme ve hukukta karar analizi gibi görevlerde modelin istatistiksel davranışını belirgin biçimde zenginleştirir.

Tüm bu mekanizmalar –çok modlu veri, domain shift (alan sıçraması), domain adaptasyonu, düşük rütbeli düzeltmeler, grafik entegrasyonu– uygulama alanlarındaki LLM davranışının yüzeyde gördüğümüz "akıllı cevaplardan" çok daha karmaşık bir istatistiksel altyapıya dayandığını gösteriyor. Bu altyapı doğru tasarlandığında model güven verir; yanlış ya da eksik tasarlandığında ise sistematik hatalar ve derin yanlılıklar üretir.

Böylece uygulama alanları bölümünün sonuna yaklaşırken, doğal olarak şu soruya geliyoruz: Tüm bu istatistiksel mekanizmalar, risk ve belirsizliği nasıl yönetiyor? Yanlılık, adalet, güvenilirlik ve "hallucination" dediğimiz olgu, modellerin altında yatan istatistiksel süreçlerde nasıl ortaya çıkıyor? Şimdi bu sorulara geçme zamanı.

İstatistiksel Riskler, Yanlılık (Bias), Belirsizlik ve Güvenilirlik

Büyük dil modellerinin hayranlık uyandıran tarafı olduğu kadar tedirgin eden tarafı da, son derece kendinden emin cümlelerle yanlış, eksik veya taraflı sonuçlar üretebilmeleridir. Bu olgu, çoğu zaman halüsinasyon (hallucination) diye adlandırılıyor; ancak gerçekte istatistiksel bir fenomenle karşı karşıyayız: Model, aslında bildiği dağılımın dışındaki noktalara da aynı özgüvenle olasılık atamaya eğilimli. Bu, belirsizliğin doğru yönetilememesinden, kalibrasyon hatalarından ve verinin kendisinde var olan yanlılıklardan besleniyor.

Belirsizliği anlamak için önce iki temel kategoriye ayırmak faydalı: aleatorik (şansa bağlı) ve epistemik (hakikatle ilgili~bilişsel) belirsizlik. Aleatorik belirsizlik, doğanın içsel rastgeleliğiyle ilgilidir; örneğin bir zarın atılmasında kaç geleceğini bilemeyiz, çünkü süreç rasgele tanımlıdır. Epistemik belirsizlik ise modelin bilgisizliğinden kaynaklanır; yeterince veri görmediği için emin değildir. LLM'ler için bu ayrım, şu şekilde yorumlanabilir: Dilin kendi doğasından gelen çok anlamlılık ve bağlam belirsizliği aleatorikken, modelin hiç görmediği bir alanın jargonunu yanlış anlaması epistemiktir.

Saf istatistikte belirsizlik çoğu zaman dağılım üzerinden ifade edilir. Bir tahmin modeli için güven aralığı çizilir, posterior dağılım hesaplanır ya da tahmin birlikte bir varyans değeri raporlanır. LLM'ler ise genellikle yalnızca en olası kelimeyi veya cümleyi sunar. Oysa model bir yanıt verdiğinde, içsel olarak bir olasılık dağılımına sahiptir. Bu dağılımın kalibrasyonu önemlidir. İyi kalibre edilmiş bir modelde, yüzde 80 güvenle yaptığı tahminler gerçekten de vakaların yaklaşık yüzde 80'inde doğru çıkar. Bunu ölçmek için kullanılan metriklerden biri beklenen kalibrasyon hatasıdır (expected calibration error, ECE):

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|\,B_m\,|}{n} \,|\, \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \,|$$

Burada $B_m$güven aralığına göre bölünmüş örnek gruplarını, $\mathrm{acc}(B_m)$o gruptaki gerçek doğruluğu, $\mathrm{conf}(B_m)$ise modelin rapor ettiği ortalama güveni ifade eder. LLM'ler pratikte sıklıkla aşırı kendine güvenli davranırlar; yani confdeğeri acc'den büyüktür. Bu aşırı özgüven, modelin "yanlış hâlde bile doğruymuş gibi konuşması" fenomenini yaratır.

Yanlılık (bias) ise başka bir katmandan gelir. Eğitim verisi, toplumdaki güç ilişkilerinin, tarihsel eşitsizliklerin, dildeki stereotiplerin bir yansımasıdır. Model, bu veriler üzerinden dil dağılımını öğrenirken, ayrımcı kalıpları da istatistiksel bir "gerçek" gibi içselleştirebilir. Örneğin belirli mesleklerin belirli cinsiyetlerle daha sık birlikte anılması, embedding uzayında meslek vektörlerinin cinsiyet vektörlerine kaymasına neden olabilir. Bunu sezgisel bir vektör hesabıyla temsil edebiliriz:

$$v(\text{doktor}) - v(\text{erkek}) \approx v(\text{mühendis}) - v(\text{erkek})$$

gibi ilişkiler güçlenirken,

$$v(\text{doktor}) - v(\text{kadın})$$

vektörü daha zayıf kalabilir. Bu tür kaymalar, modelin üretiminde sistematik önyargılara yol açar; kadın doktor yerine erkek doktoru örneklemesi daha olası hâle gelir. İstatistiksel bakışla, bu bir tür örnekleme yanlılığıdır; eğitim dağılımında belirli kombinasyonların sık, bazılarının seyrek olması, modelin tahmin uzayını eğri bir biçimde şekillendirir.

Yanlılığı analiz etmek için çeşitli adalet metrikleri kullanılır. Örneğin, iki grup için pozitif karar olasılıklarını karşılaştırdığımızda, eşitlik beklentisi

$$P(\hat{Y} = 1 \mid A = 0) \approx P(\hat{Y} = 1 \mid A = 1)$$

şeklinde ifade edilebilir. Burada A duyarlı bir özellik (örneğin cinsiyet, etnik köken), $\hat{Y}$ model kararını temsil eder. LLM bazlı sistemlerde bu tür oranlar doğrudan çıkmayabilir; fakat belirli senaryoların modelle nasıl ifade edildiğine bakılarak dolaylı olarak ölçülebilir. Örneğin "başarı hikâyesi" isteklerinde modelin hangi demografik figürleri daha sık üretmesi, bu nispi olasılıkların pratiğe yansımasıdır.

Halüsinasyon (hallucination) dediğimiz olgu, istatistiksel açıdan şöyle okunabilir: Model, gerçek dünya bilgi dağılımı $P_{\text{gerćek}}(x)$yerine yalnızca kendi öğrendiği dil dağılımı $P_\theta(x)$üzerinden örnekleme yapar. Dil dağılımı, gerçeğin bir gölgesidir; bilgi kaynakları sınırlıysa veya eğitim verisi eksikse, $P_\theta$ile $P_{\text{gerćek}}$arasındaki fark büyür. Model, aslında hiç gözlenmemiş ama dilsel olarak makul görünen örnekleri yüksek olasılıkla seçebilir. Bu durum, özellikle kaynak gösterme ya da teknik ayrıntı sunma gibi görevlerde kendini belli eder. Model, "bilinmeyen"i açıkça ifade etmek yerine, öğrenilmiş kalıplardan türetilmiş bir "uydurulmuş ama tutarlı" metin üretir.

Belirsizliğin güvenilir şekilde yönetilmesi için iki tür strateji öne çıkar. İlki, modelin cevap üretirken kendi eminlik düzeyini sinyallemesi, yani içsel olasılık dağılımının belirsizliğini dışarı vurmasıdır. Bu, sıcaklık (temperature) ve örnekleme stratejileriyle birlikte, belirli görevlerde daha temkinli davranmasını sağlayabilir. İkinci strateji, modelin yanıtlarının dış bilgi kaynaklarıyla çapraz doğrulanmasıdır. Özellikle hukuk, sağlık ve bilimsel alanlarda, LLM yanıtlarının veri tabanları, kılavuzlar veya makalelerle otomatik karşılaştırılması, hatalı örneklerin riskini azaltır. Bu ikinci strateji, doğru tasarlanırsa, modelin P_θ'sini daha sık $P_{\text{gerćek}}$'e projekte eden bir düzeltme mekanizması gibi çalışır.

Güvenilirlik meselesi, istatistiksel hataların ötesinde bir toplumsal boyut taşır. Bir modelin istatistiksel olarak iyi performans göstermesi, tüm gruplar için adil sonuçlar ürettiği anlamına gelmez. Bu noktada "riskin dağılımı" kavramı önem kazanır. Aynı toplam hata oranına sahip iki modelden biri, hatalarını rastgele dağıtıyorsa, diğeri belirli bir grupta yoğunlaştırıyorsa, ikincisi etik açıdan çok daha sorunludur. İstatistiksel formalizmde benzer toplam riskler eşdeğer görülebilir; fakat toplumda doğruluk kadar adalet de bir performans kriteridir.

Sonuçta istatistiksel risk, yanlılık, belirsizlik ve güvenilirlik birbirine sıkıca bağlıdır. LLM'lerin altındaki istatistiksel teknikler ne kadar zarif olursa olsun, verinin doğasındaki bozukluklar, modelin tasarımında yapılan seçimler ve kullanım bağlamındaki güç ilişkileri, çıktının etik niteliğini belirler. Bu nedenle LLM'ler için geliştirilen her yeni istatistiksel yöntem, aynı zamanda yeni bir sorumluluk alanını da beraberinde getirir.

10 – Genel Tartışma: İstatistiksel Tekniklerin Bütünleşmesi ve LLM Paradigması

Bu noktaya kadar parçalar hâlinde incelediğimiz her şey, aslında tek bir bütünün farklı yüzleri. Olasılık teorisi, bilgi teorisi, istatistiksel öğrenme kuramı, Bayesian çıkarım, EM algoritması, Markov zincirleri, PCA ve SVD gibi boyut indirgeme yöntemleri, olasılıksal grafik modeller, optimizasyon teknikleri, Transformer mimarisi ve uygulama alanlarında ortaya çıkan risk ve yanlılık dinamikleri; bunların hiçbiri LLM'lerden bağımsız, ayrı dünyalar değil. Tam tersine, büyük dil modelleri bu tekniklerin hepsini içeren bir tür "istatistiksel orkestrasyon" olarak görülebilir. Bir anlamda, modern LLM paradigması, istatistiksel düşüncenin son kırk yılda geçirdiği evrimin somutlaşmış bir sentezidir (Bishop, 2006; Murphy, 2023).

Dil modellemenin en temel düzeyinde, hâlâ $P(w_t \mid w_{<t})$şeklindeki koşullu olasılık düşüncesi yatıyor. Bu düşünce, hem klasik n-gram modellerini hem de günümüzün dev transformer mimarilerini birleştiren basit ama güçlü bir iplik gibi. Bilgi teorisinin entropi ve çapraz entropi kavramları, kayıp fonksiyonunun matematiksel gövdesini oluşturuyor (Cover & Thomas, 2006). İstatistiksel öğrenme kuramı, empirik risk minimizasyonu ve düzenlileştirme sayesinde, bu kaybı minimize ederken modelin genelleme yapabilir olmasını sağlıyor. Bias–variance tartışması, LLM'lerin neden aşırı parametreli olmalarına rağmen işe yaradığını anlamamızda hâlâ temel bir kavramsal çerçeve sunuyor (Hastie et al., 2009).

Bayesian bakış açısı, posterior mantığıyla belirsizliğin nasıl güncelleneceğini anlatırken, Markov zincirleri ve MCMC yöntemleri yüksek boyutlu dağılımlarla baş etmenin algoritmik yollarını gösteriyor (Gelman et al., 2013). Her ne kadar modern LLM'ler kelimenin klasik anlamıyla "tamamen Bayesci" olmasa da, örnekleme stratejilerinin ve temperature temelli olasılık manipülasyonlarının arkasında bu fikirlerin gölgesi net şekilde görülebilir. Bir dil modeli, aslında her adımda posterior benzeri bir inanç güncellemesi gerçekleştiriyor; yalnızca bunu açık formüllerle değil, sinir ağı parametrelerinin içkin temsil gücüyle yapıyor.

EM algoritması, gizli değişkenli modellerde parametre tahmini için geliştirilmiş bir yöntem olarak tarih sahnesine çıktı (Dempster et al., 1977; Rabiner, 1989). HMM'ler, IBM çeviri modelleri ve çeşitli karışım modelleri üzerinden dilin istatistiksel yapısını öğrenmenin yollarını açtı. Bugün transformer tabanlı LLM'lerde açık bir EM döngüsü görmüyoruz; ama her katmanda iç temsilin güncellenmesi ve bu temsil üzerinden parametrelerin geri yayılımla düzeltilmesi, kavramsal olarak "tahmin et ve güncelle" döngüsünün modern bir yorumu gibi. Özellikle mixture-of-experts (karmaşa uzmanı) yapıların yükselişi, gizli değişkenli karışım modellerini doğrudan tekrar gündeme taşıyor (Shazeer et al., 2017).

Boyut indirgeme teknikleri, özellikle PCA ve SVD, dilin yüksek boyutlu temsilini daha kavranabilir bir geometrik yapıya oturtmanın temel araçları oldu (Jolliffe, 2002). Word2Vec ve GloVe gibi erken embedding yöntemlerinin, log-olasılık matrisleri üzerinden dolaylı SVD benzeri faktörizasyonlar gerçekleştirdiği gösterildi (Mikolov et al., 2013; Pennington et al., 2014). Transformer tabanlı LLM'lerde embedding boyutları büyüdükçe, bu gömülü geometri daha da önemli hâle geldi. Embedding uzayı, dilin istatistiksel "haritası"dır; PCA bu haritada baskın anlam eksenlerini, SVD ise global yapıyı ortaya çıkarır. Bu uzayın lineerliği, analogilerin vektörel olarak ifade edilebilmesini mümkün kılar; $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$benzeri ilişkiler tesadüf değil, bu geometrik yapının ürünüdür (Mikolov et al., 2013).

Olasılıksal grafik modeller, bağımlılık yapılarının açıkça ifade edildiği bir dil sunar (Koller & Friedman, 2009). Transformer'ın self-attention mekanizması, her bir katmanda dinamik olarak öğrenilen, veriyle belirlenen bir graf gibi yorumlanabilir. Attention matrisleri, token'lar arasındaki koşullu bağımlılıkların ağırlıklı kenarlarını temsil eder; softmax ile normalize edilen bu ağırlıklar, her bir kelimenin bağlamdaki diğer kelimelere "ne kadar kulak verdiğini" gösterir (Vaswani et al., 2017). Grafik modellerdeki faktörizasyon fikri, burada attention ile yeniden canlanır; model, dilin tam ortak dağılımını değil, faktörize bir koşullu yapı üzerinden öğrenir.

Optimizasyon cephesinde SGD, Adam, momentum ve benzeri yöntemler, LLM'lerin devasa parametre uzayında bir tür "istatistiksel devinim"dir (Kingma & Ba, 2015). Loss yüzeylerinin yüksek boyutlardaki nispeten "düz" yapısı, iyi genelleme yapan geniş ağların varlığını açıklamaya yardımcı olur (Keskar et al., 2017). Bu açıdan bakıldığında, eğitim süreci yalnızca bir mühendislik detayı değil; aynı zamanda modelin hangi tür minimumlara yerleşeceğini belirleyen temel bir istatistiksel süreçtir. Düz minimumlar genellikle daha az hassas, daha kararlı ve yeni örneklere karşı daha bağışlayıcı çözümlere karşılık gelir.

Transformer mimarisi, tüm bu tekniklerin birleştiği bir tür istatistiksel işleme hattıdır. Positional encoding, dilin sırasını sürekli bir uzaya kodlar; self-attention, koşullu benzerlikleri iç çarpımlar üzerinden ağırlıklandırır; multi-head attention (çok başlı dikkat), aynı veriye farklı istatistiksel perspektifler uygular; feed-forward (ileri beslemeli) ağlar, yerel doğrusal olmayan regresyonlar olarak temsilleri zenginleştirir; residual (kalıntı) bağlantılar ve layer normalization, istatistiksel kararlılığı artırır (Vaswani et al., 2017; Ba et al., 2016). Bu bileşenler birlikte çalıştığında, model yalnızca "metin ezberleyen" bir yapı olmaktan çıkar; dilin altında yatan olasılık dağılımlarına erişen bir hesaplama mekanizmasına dönüşür.

Uygulama alanlarına döndüğümüzde, sağlık, finans, hukuk ve bilimsel araştırma gibi disiplinlerde LLM'lerin davranışını anlamak için yine istatistiğe sığınıyoruz. Klinik kararlarda duyarlılık ve özgüllük, finansal riskte kovaryans ve volatilite, hukukta içtihat ağının graf yapısı, bilimsel metinlerde hipotez testleri ve güven aralıkları; tüm bunlar LLM davranışının alan-özel değerlendirme araçları hâline gelmektedir. Model, bir bakıma bu alanların istatistiksel reflekslerini dil üzerinden öğreniyor. Bu öğrenme doğru olduğunda model ikna edici ve faydalı; yanlış olduğunda ise son derece tehlikeli bir hâl almaktadır.

Risk, yanlılık ve belirsizlik tartışması ise tüm bu teknik güzelliğin karanlık tarafını açığa çıkarmaktadır. Eğitim verisindeki örnekleme yanlılıkları, tarihten miras duran eşitsizlikler, kalibrasyonu bozulmuş olasılık tahminleri ve dağılım kaymaları, LLM çıktılarında sistematik hatalara neden olabilmektedir. (Barocas et al., 2019; Mitchell et al., 2019). Hallucination olgusu, modelin kendi öğrendiği dil dağılımını gerçek dünyanın üzerine aşırı güvenle projekte etmesi olarak okunabilir; epistemik belirsizliğin açıkça ifade edilmemesi, hatanın riskli bir özgüvenle paketlenmesine yol açmaktadır.

Tüm bu tabloya yukarıdan baktığımızda, LLM'lerin aslında birer "istatistiksel koalisyon" olduğu ortaya çıkmaktadır. Farklı istatistiksel teknikler, teorik ve pratik katmanlarda üst üste binmektedir; her katman kendi rolünü oynamakta, nihai davranış bu katmanların etkileşiminden doğmaktadır. LLM'ler ne sihir ne de yalnızca mühendislik başarısı; onlar onlarca yıllık istatistiksel düşüncenin, bilgisayar gücüyle çarpılmış bir devamıdır.

Sonuç: İstatistiksel Temelden Dilsel Yaratıcılığa

Büyük dil modellerinin altında yatan istatistiksel tekniklere baktığımızda, modern yapay zekânın aslında ne kadar klasik sorularla uğraştığını fark etmakteyiz. "Bir olasılık dağılımını veri üzerinden nasıl tahmin ederim?", "Gördüğüm örneklerden görmediklerime doğru nasıl genellerim?", "Belirsizliği nasıl ifade eder, nasıl yönetirim?", "Gizli yapıların varlığını nasıl keşfederim?" gibi sorular, istatistik biliminin en eski sorularıdır. LLM'ler yalnızca bu sorulara verilen yanıtların ölçüsünü büyütmektedir; veri miktarını, parametre sayısını, hesaplama gücünü ve dolayısıyla modelin ifade kapasitesini olağanüstü seviyelere taşımaktadır.

Bu çalışmada olasılık ve bilgi teorisinden başlayarak, istatistiksel öğrenme kuramı, Bayesian yaklaşım, EM, Markov zincirleri, boyut indirgeme, olasılıksal grafik modeller, optimizasyon, Transformer mimarisi ve uygulama alanlarının tamamı boyunca LLM'lerin istatistiksel iskeletini ortaya sermeye çalıştık. Çıkan resim, LLM'lerin yalnızca "derin sinir ağları" değil, aynı zamanda çok katmanlı istatistiksel sistemler olduğu yönünde oldu. Her bir teknik, modelin bir parçasını taşımaktadır; olasılık dağılımları, latent temsiller, çekirdek fonksiyonları, faktörizasyonlar ve gradyan akışları hep birlikte çalışmaktadır.

Diğer yandan, bu istatistiksel güç tek başına bir güven garantisi değildir. Yanlılık, belirsizlik, kalibrasyon bozuklukları ve dağılım kaymaları, modelleri hem daha kırılgan hem de toplumsal düzeyde daha riskli hâle getirebilmektedir. Özellikle sağlık, hukuk ve finans gibi yüksek etkili alanlarda, LLM tabanlı sistemlerin istatistiksel özelliklerini anlamak, etik ve hukuki sorumlulukların bir parçası hâline gelmektedir. Model performansını değerlendirirken yalnızca doğruluk oranına değil, hatanın kimin üzerinde yoğunlaştığına, hangi grupların daha büyük risk altında olduğuna, belirsizliğin ne kadar şeffaf ifade edildiğine bakmak zorundayız.

Geleceğe dönük olarak, LLM araştırmalarının önemli bir kısmı muhtemelen daha açık Bayesian mekanizmalar, daha iyi belirsizlik tahmini, daha sağlam domain adaptasyonu, daha şeffaf grafiksel temsil ve daha iyi kalibrasyon teknikleri etrafında şekillenecek. Variational inference, MCMC türevleri, derin olasılıksal programlama gibi alanlar ile büyük dil modelleri arasındaki köprüler güçlendikçe, istatistiksel temelin de daha görünür hâle geleceği söylenebilir (Blei et al., 2017; Kingma & Welling, 2014; Murphy, 2023).

Yani LLM'ler, insan dilinin olasılıksal bir aynasıdır. Bu aynada gördüğümüz yalnızca gramer ve kelime dizilişi değil; aynı zamanda dünya hakkında taşıdığımız istatistiksel sezgiler, ön yargılar, bilgi birikimi ve boşluklardır. Bu yüzden LLM'leri anlamak, biraz da kendimizi, dille kurduğumuz ilişkiyi ve belirsizliği nasıl yönettiğimizi anlamaktır. İstatistiksel teknikler bu anlamanın dili; LLM'ler ise bu dilin bugünkü en çarpıcı uygulamalarıdır.

REFERENCES

- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint* arXiv:1607.06450.
- Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and machine learning*. Fairness in Machine Learning.
- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877.

- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Wiley.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1), 1–38.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171–4186.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis* (3rd ed.). CRC Press.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer.
- Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length and Helmholtz free energy. In *Advances in neural information processing systems* (Vol. 6).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Springer.
- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed., draft). Draft manuscript.
- Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.
- Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*.
- Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint* arXiv:1301.3781.
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229.
- Murphy, K. P. (2023). *Probabilistic machine learning: Advanced topics*. MIT Press.
- Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., & Srebro, N. (2017). The role of over-parameterization in generalization of neural networks. *International Conference on Learning Representations (ICLR) workshop*.
- Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *Proceedings of EMNLP*, 1532–1543.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.

- Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *International Conference on Learning Representations (ICLR)*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.