Makine Unutması

Dr. Mahir E. Ocak [*TÜBİTAK Bilim ve Teknik Dergisi*

Yapay zekâ modellerine bir şeyler öğretmek kolay bir iş değil. Ancak daha da zor olanı eğitilmiş bir yapay zekâ modeline bildiklerini unutturmak, özellikle de söz konusu yapay sinir ağlarına dayalı modeller olduğunda. Bir yapay zekâ modeline sahip olduğu hatalı, tehlikeli, zararlı ya da güncelliğini kaybetmiş bilgileri nasıl unutturulabileceği, günümüzde giderek büyüyen bir araştırma alanı.

metamorworks / iStock

Feodora Chiosea / iStock

Makine unutmasının temel amacı, eğitilmiş bir yapay zekâ modelini ele alıp yeni bir yapay zekâ modeli geliştirmektir. Öyle ki yeni modelin davranışı, unutulması istenen bilgiler hariç tutularak aynı bilgilerle eğitilmiş bir yapay zekâ modeline denk olmalıdır ya da en azından "öyleymiş gibi" davranmalıdır. Bu tanımlama pek çok soruyu akıllara getirir. Örneğin unutulması istenen bilgiler her zaman iyi ifade edilebilir mi? Arzu edilen amaca ulaşıp unutulması istenen bilgilerle hiç karşılaşmamış gibi davranan bir model geliştirmek her zaman mümkün müdür? Eğer mümkün değilse unutmanın başarısı nasıl ölçülebilir? Unutmanın başarılı olduğunu ispatlamak ya da denetlemek mümkün müdür? İnsanların yaptığı gibi makinelerin de unutmuş gibi davranması yeterli midir?

Makine Unutmasının Kullanım Alanları

Makine unutması üzerine yapılan ilk çalışmaları motive eden Avrupa Birliği'nin mahremiyet düzenlemeleri (GDPR) arasında yer alan "unutulma hakkı"ydı. Bu düzenleme, bir kullanıcıya hizmet aldığı bir şirketten kendi özel bilgilerinin kaydının silinmesini isteme hakkı verir. Örneğin e-posta hizmeti veren şirketler, bir kullanıcı hesabını kapattığında genel olarak kullanıcının bilgilerini veri tabanlarından silmeyi tercih etmez. Unutulma hakkı, bir e-posta kullanıcısına hesabını sildikten sonra hizmet aldığı şirketten kendi bilgilerinin kayıtlardan silinmesini isteme hakkı verir.

Unutulma hakkı, 2014 yılında ilk kez yürürlüğe konulduğunda makine öğrenmesi göz önünde bulundurularak hazırlanmamıştı. O dönemler yapay sinir ağlarına dayalı modellerin, gün gelip de verileri ve hesapları harmanlayan, çalışma biçimlerinin anlaşılması zor kara kutulara dönüşeceğini tahmin etmek zordu.

Makine unutmasının, unutulma hakkının dışında da kalan pek çok uygulama alanı var: Eğitimlerde kullanılmış telif haklarına tabii bilgilerin ya da eserlerin istek üzerine eğitim setinden çıkarılmasının gerekmesi, eğimlerde kullanılmış bilgilerin hatalı ya da güncelliğini yitirmiş olması, uygulamanın zararlı ya da tehlikeli sonuçlara yol açabilecek bilgiler edinmiş olması...

Yapay Sinir Ağlarına Bilgi Unutturmanın Zorluğu

Söz konusu bilgilerin, insanların okuyup anlayabileceği biçimde kodlandığı bilgisayar programları ya da tek tek belgeler hâlinde tutulduğu veri tabanları olduğunda bir sisteme bir şeyi unutturmak pek zor değildir. Programlardaki ilgili kod satırlarını ya da veri tabanlarındaki ilgili belgeleri tespit edip silersiniz. Örneğin Google, geçmişte unutulma hakkı

kapsamında gelen pek çok isteğe cevap vererek veri tabanlarındaki kullanıcı bilgilerini silmişti.

Söz konusu yapay sinir ağlarıyla modellenen programlar olduğundaysa bir yapay zekâ modeline bir şeyi unutturmak pek de kolay değildir. Bu durumun nedeni, bu modellerin tıpkı yapısından esinlendikleri insan beyni gibi bilgiyi tekil birimlerde değil dağınık bir biçimde kodlamalarıdır.

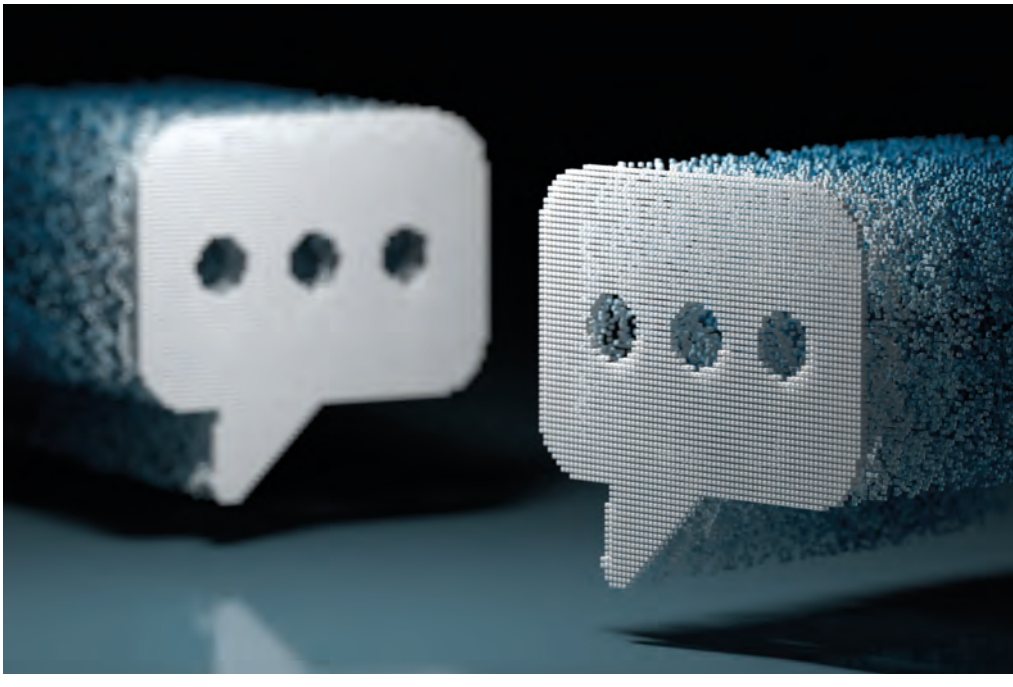
Yapay sinir ağlarıyla modellenen uygulamalarda insan beynindeki sinir ağına benzer bir yapı oluşturulur. Bu modellerde yapay nöronlar bulunur ve bu yapay nöronlar da doğal nöronlar gibi birbiriyle sinapslarla bağlantılıdır. Yapay sinir ağının bir konuda bilgi sahibi olması uzun eğitim süreçleri sonunda mümkün olur. İnsanlar yeni şeyler öğrenirken sinapslar arasındaki bağlantıların güçlenip zayıflamasına benzer biçimde yapay zekâ modelleri eğitilirken de yapay nöronlar arasındaki sinapslar güçlenir ya da zayıflar. Bilginin insan beyninde tekil nöronlarda saklanmaması gibi yapay sinir ağlarında da bilgi tekil nöronlarda kodlanmaz. Veri tabanlarından tekil belgelerin ya da sembolik programlardan satırların silinmesine benzer biçimde yapay sinir ağındaki bir nörona müdahale edilerek belirli bir bilginin sistemden silinmesi sağlanamaz. Tıpkı bir nörona müdahale edilerek bir insanın bir bilgiyi unutmasının sağlanamayacağı gibi.

Yapay sinir ağlarına bilgi unutturmanın zorluğu özellikle son 10-15 yılda bu tür yapay zekâ programlarının giderek ön plana çıkmasından sonra makine unutmasının giderek yoğunlaşan araştırmalara konu olmasına yol açtı.

Kesin Unutma

Eğer elinizde belirli bilgileri unutturmak istediğiniz bir yapay zekâ modeli varsa yapılabilecek en basit şeylerden biri tamamen yeni bir yapay zekâ modeli eğitmek olabilir. Eğer bir A algoritmasını ve bir D veri tabanını kullanarak bir $M=A(D)$ modeli elde etmişseniz ve bu modelin unutmasını istediğiniz bilgi kümesi D_r ise istenmeyen bilgileri veri tabanından silip geriye kalan $D\text{-}D_r$ bilgileriyle aynı A algoritmasını kullanarak bir $M'=A(D\text{-}D_r)$ modeli oluşturursunuz. Ancak yapay zekâ modellerini eğitmek uzun zaman alan masraflı bir iştir.

Kesin unutma için öne sürülmüş yöntemler, tüm eğitimi en başından tekrar etmeden orijinal modeli yeniden eğitilmiş bir modele dönüştürmeyi amaçlar. Bu durum orijinal yapay zekâ modelinin eğitim prosedürlerinin, bir gün elde edilecek uygulamaya bazı şeyleri unutturmak da gerekebileceğinin göz önünde bulundurularak tasarlanmasını gerektirir. Böylece unutma sürecinin tüm eğitimi en başından tekrar etmeden daha hızlı, daha verimli ve daha ucuz bir biçimde gerçekleştirilmesi mümkün olur.



Just_Super / iStock

Kesin unutma için geliştirilmiş ilk algoritmalardan biri kısaca SISA (*Sharded, Isolated, Sliced, Aggregated*) olarak adlandırılır. SISA uygulamalarında yapay zekâ modelinin eğitimi bir bütün hâlinde yapılmaz. Veri kümesi ufak parçalara (*sharded*) ayrılır. Her bir parça farklı bir alt modelin eğitiminde kullanılır ve eğitimler sırasında farklı modeller birbirinden ayrı (*isolated*) tutulur. Ayrıca her bir modelin eğitimi sırasında veriler, dilimler (*sliced*) hâlinde eğitim setine eklenir. Her bir dilim eğitime eklenmeden önce modellerle ilgili tüm parametreler kaydedilir. Geliştirilen uygulama, kullanıcıların taleplerine cevap vermek için önce her bir alt modelden bir cevap üretmesini ister daha sonra bu cevapları bir araya toplayıp (*aggregated*) kullanıcıya sunacağı nihai cevabı üretir.

SISA ile geliştirilmiş bir modelin bir bilgiyi unutmasını sağlamak görece daha kolaydır. Eğer veri tabanı başlangıçta *N* ayrı kümeye bölünmüşse bu durum yeniden belirlenmesi gereken parametrelerin sayısını *1/N* oranında azaltır. Unutma için uygulamayı meydana getiren *N* tane alt modelin tamamının yeniden eğitilmesi gerekmez, sadece ilgili bilgilerle eğitimden geçirilmiş tek bir modelin yeniden eğitilmesi yeterlidir. Üstelik modeller eğitilirken veri setindeki bilgiler dilimler hâlinde verildiği için ilgili modelin eğitiminin en başından tekrar edilmesi de gerekmez. Modeli meydana getiren parametrelerin unutulması istenen bilgilerle karşılaşmadan önceki değerlerini alıp modelin yeniden eğitimini bu noktadan başlatmak yeterlidir. SISA'nın bir dezavantajı ise modeli çok sayıda alt modele bölmenin üretilen cevapların niteliğini düşürmesi, çok sayıda alt modelin performansının tek bir büyük modelin performansının gerisinde kalmasıdır.

SISA'daki performans düşüşünün önüne geçmenin bir yolu, uygulamayı tek bir modeli eğitimden geçirerek üretmek ancak eğitimler sırasında ara ara modelle ilgili parametrelerin kaydını da tutmaktır. Bu yöntemde de SISA'da olduğu gibi tüm eğitime en başından başlamak gerekmez. Sadece modelin unutulması istenen bilgilerle karşılaştıktan sonraki eğitimlerini yenilemek yeterlidir. Ancak bu yöntemin gerçek hayatta uygulanması SISA'ya kıyasla çok daha zordur. Çok sayıda parametrenin güncellenecek olması yeniden eğitimin süresini uzatır. Ayrıca çok sayıda parametrenin, eğitimin farklı aşamalarına denk gelen çok sayıda noktadaki değerlerini kayıt altına alınma gerekliliği, çok büyük bir depolama kapasitesi gerektirir.

Kesin unutma algoritmalarının bir özelliği algoritmaların kendilerinin "unutmanın başarıya ulaştığının" ispatı olmasıdır. Unutulması istenen bilgilerin sadece tek bir modelin eğitiminde kullanılması ve bu modelin yeniden eğitimden geçirilmesi ortaya çıkan son ürünün arzu edildiği gibi istenmeyen bilgilerle hiç karşılaşmamış bir yapay zekâ modeli olacağını garanti eder. Bu yöntemler unutma problemini bir verimlilik/kesinlik problemine dönüştürür.

imaginima / iStock

Diferansiyel Mahremiyet

Diferansiyel mahremiyet (DP) bir makine unutması yöntemi değildir. Eğitimli bir modele bir şeyler unutturmak için kullanılmaz. Ancak DP prosedürleri kullanılarak eğitilmiş bir yapay zekâ modeline belirli konulardaki bilgilerin unutturulmasına gerek de kalmaz.

DP, bir öğrenme algoritmasının ortaya çıkardığı sonucun belirli bir verinin eğitim setinde kullanılmasına ya da kullanılmamasına bağlı olarak belirgin bir biçimde değişmemesini amaçlar.

Bir D eğitim seti kullanarak bir M modeli oluşturduğunuzu daha sonra da D veri setinden tek bir V kişisi ile ilgili bilgilerin çıkarılmasıyla elde edilmiş D' veri setiyle aynı algoritmaları kullanarak bir M' modeli oluşturduğunuzu düşünün. Eğer M ve M' modelleri hemen hemen aynı tahminleri yapıyor, iki modelin tahminleri arasında belirgin bir fark gözlemlenmiyorsa M modeline V kişisi ile ilgili bilgileri unutturmaya çalışmanın bir anlamı yoktur. Model zaten hiçbir zaman V kişisiyle ilgili bilgilerle karşılaşmamış gibi çalışmaktadır.

Bir uygulamanın DP'ye sahip olduğuna ve belirli bir insanın kişisel bilgilerini unutmaya gerek olmadığına karar verilebilmesi için modelin eğitiminde belirli verilerin kullanılıp kullanılmadığını anlamaya çalışan birisinin bir sonuca ulaşamaması gerekir. Bu durum şu soruyu akla getirir: Eğer yapay sinir ağları kara kutu gibi çalışan, nasıl işlediğinin anlaşılması zor sistemlerse bir bilginin, bir yapay sinir ağının eğitim setinde kullanılıp kullanılmadığı hakkında nasıl fikir edinilebilir?

Her ne kadar kara kutu gibi çalışsalar da yapay zekâ modellerinin eğitimleri sırasında hangi bilgilerin kullanıldığı hakkında bir fikir edinmek mümkündür. Bu durumun nedeni yapay zekâ eğitimleri ile ilgili yaygın bir sorundur. Bir yapay zekâ modeli, eğitimler sırasında verilen bilgiler hakkında soru sorulduğunda yüksek doğrulukla tahminler yapacak biçimde eğitilir. Ancak bu durum modelin daha önce hiç karşılaşmadığı durumlar karşısında da aynı doğrulukla tahmin yapacağı anlamına gelmez. Örneğin nesneleri tanıyıp sınıflandırmak için eğitilmiş bir model olduğunu düşünelim. Bu modele eğitim setinde yer alan bir fotoğraf gösterirseniz çok yüksek bir olasılıkla doğru cevabı verecektir. Ancak bu durum modelin eğitim setinde yer almayan bir fotoğraftaki bir nesneyi de yüksek doğrulukla

sınıflandırabileceği anlamına gelmez. İdeal olanı, modelin her koşulda büyük bir doğrulukla tahmin yapmasıdır. Ancak gerçek hayatta bunu yakalamak zordur.

Bir kişinin özel bilgilerinin bir modelin eğitiminde kullanılıp kullanılmadığını belirlemenin bir yolu da o modele, bahse konu bilgilerle ilgili sorular sormaktır. Yapay zekânın verdiği cevapların doğruluk oranı genel ortalamanın çok üzerinde çıkıyorsa bu durum, modelin eğitim sırasında bu bilgilerle karşılaştığına işaret eder. Bu yöntem "üyelik çıkarım saldırısı" olarak adlandırılır.

DP'nin amacı, üyelik çıkarım saldırıları karşısında modelin eğitim materyalleri hakkında dışarıya bilgi sızdırmasını engellemektir. Bu amaçla eğitim sırasında modele bilgiler verilirken içlerine bir miktar gürültü karıştırılır. Başka bir deyişle veriler bir miktar çarpıtılır. Böylece sonuçları fazla etkilemeden kişiye özel bilgilerdeki ayrıntıların etkileri silinmeye çalışılır. Elde edilen modelin tekil verilerdeki ufak detaylara bağımlılığı azaldığı için eğitim setinden bir materyali çıkarmak, programın yapacağı tahminler üzerinde belirgin bir değişime yol açmaz. Dolayısıyla DP algoritmaları, kişiye özel bilgilerin unutturulmasına olan ihtiyacı ortadan kaldırır.

DP algoritmalarının ne kadar başarılı olacağı verilere ne kadar gürültü eklendiğine bağlı olarak değişir. Daha fazla gürültü, daha fazla mahremiyet anlamına gelir. Ancak önemli bir noktayı da not edelim: Eğitim setindeki verilere yüksek miktarda gürültü eklemek mahremiyeti artırsa da eğitimin kalitesini düşürür, yapay zekâ modelinin ürettiği cevapların doğruluğunu azaltır. Dolayısıyla DP algoritmalarını kullanırken hassas bir dengeyi korumak da gerekir.

Kesin unutma ile karşılaştırıldığında DP önleyici bir yöntemdir. Kesin unutma yöntemleri gibi bir yapay zekâ modelini verimli bir biçimde yeniden eğitilmiş bir modele dönüştürmeyi amaçlamaz. Eğitimin en başından alınan önlemlerle tekil verilerin etkilerini sınırlamayı amaçlar. Böylece makine unutmasına olan ihtiyacı ortadan kaldırır.

Nikada / iStock

Yaklaşık Unutma

Yaklaşık unutma kapsamına giren yöntemlerin amacı, sıfırdan eğitime başvurmadan çeşitli bilgileri tamamen unutmuş bir modele mümkün olduğu kadar yakın davranışlar sergileyen bir model geliştirmektir. Bu kapsama giren yöntemler genel olarak eğitimli bir modeli alır ve bu modeli "bir şeyleri unutması için biraz daha eğitir".

Yaklaşık unutma yöntemlerinden biri "gradyan tırmanışı''dır. Bu yöntem esasen yapay zekâ modellerinin eğitiminde kullanılan "gradyan alçalması yöntemi''nin tersidir. Gradyan alçalmasında önce bir yapay zekâ modelinin parametrelerindeki ufak değişimlerin, modelin tahminlerini nasıl değiştireceği hesaplanır daha sonra ise parametreler, hata oranını azaltacak biçimde bir miktar değiştirilir. Gradyan tırmanışında da benzer bir biçimde model, parametrelerin güncellenmesine dayanır. Ancak güncellemeler hata oranını azaltacak değil artıracak bir biçimde yapılır. Bir başka yöntemde ise unutma eğitimi sırasında unutulması istenen bilgiler hatalı bir biçimde etiketlenmiş olarak modele sunulur. Böylece modelin kafası karıştırılarak bilgiler unutturulmaya çalışılır.

Yaklaşık unutma yöntemlerinin en önemli uygulama alanlarından biri, unutturulmaya çalışılan bilgilerin net olarak ifade edilemediği durumlardır. Örneğin bir yapay zekâ modeline çeşitli "kavramlar" ya da bir konuda sahip olduğu bilgi dağarcığı (mesela kuantum fiziği) nasıl unutturulabilir.

Bir kavram ya da bir konu eğitim setindeki tek bir örnek değildir. Model, konuları çok sayıda örnekten yavaş yavaş öğrenir. Öğrendiği bilgiler, yapay sinir ağında çok sayıdaki katmanda yer alan çok sayıda nöronda kodlanmıştır. Konuyla ilgili kavramlar birbiriyle bağlantılıdır. Model, bu kavramlar arasındaki ilişkileri öğrenerek konuyu kavrar. Üstelik aynı kavramlar çok sayıda farklı konuyla ilgili olabilir. Dolayısıyla bir yapay zekâ modeline kuantum fiziğini unutturmak gibi bir problemi formüle etmek bile hiç kolay değildir: Eğitim setindeki hangi bilgiler kuantum fiziğini "tanımlar?" Kuantum fiziğiyle yakından ilişkili diğer konular hakkında ne yapmak gerekir? Eğitim setindeki kuantum fiziğine verilen dolaylı referanslar hakkında ne yapmak gerekir? Kavramların ya da konuların sınırlarının net olmaması bu ve benzeri sorulara cevap bulmayı zorlaştırır ve kavram unutma probleminin net bir biçimde formüle edilmesini imkânsızlaştırır.

Söz konusu kavramlar olduğunda başvurulacak yöntemlerden biri de modelin cevap vermesini engellemektir. Pekiştirmeli öğrenme gibi yöntemlerle modelin çeşitli konularda cevap vermekten kaçınması sağlanabilir. Bu yöntemde model esasen unutmaz. Hatta ürettiği metinlerde unutturulmak istenen kavramı hiçbir zaman doğrudan kullanmasa bile sahip olduğu bilgilerden üstü kapalı bir biçimde yararlanmaya devam edebilir.

Bir başka yöntem ise kavramların yapay sinir ağında nasıl kodlandığını tespit edip model parametrelerinde değişiklikler yaparak kavramı unutturmaya çalışmak olabilir. Ancak farklı kavramların yapay sinir ağlarındaki temsilleri arasında örtüşmeler olması, bir kavramı unutturmaya çalışmanın başka kavramlar ile ilgili bilgilere de zarar verilmesine yol açar. Bu durum modelin performansını düşürür. Yapay zekâya kavramları unutturmaya çalışmak, veri tabanlarından belgelerin silinmesine değil, beyin dokusuna hasar verilmesine benzer.

Unutmayı Değerlendirmek

Makine unutması yöntemlerinin ve algoritmalarının ne ölçüde başarılı olduğunu değerlendirmek pek kolay değildir. Genel olarak göz önünde bulundurulması

dem10/iStock

gereken üç ayrı kriter vardır: Birincisi, unutma yönteminin ne kadar verimli olduğu, başka bir deyişle kullanılan algoritmanın sıfırdan eğitim ile kıyaslandığında ne ölçüde daha hızlı olduğudur. İkincisi, kullanılan yöntemin modelin performansını nasıl etkilediği yani söz konusu eğitim setinde geriye kalan bilgiler ya da konular olduğunda modelin ürettiği cevapların niteliğinde bir düşüş olup olmadığıdır. Üçüncüsü, unutmanın ne ölçüde başarılı olduğu, modelin unutulması istenen bilgileri ya da kavramları gerçekten unutup unutmadığıdır.

Unutma algoritmasının verimliliğini ya da modelin performansındaki değişimleri değerlendirmek görece daha kolaydır. Bu değerlendirmeler için gereken ölçümler zaten eğitim sürecinde rutin olarak yapılır. Makine unutması ile ilgili değerlendirilmesi en zor kriter, unutmanın başarısını ölçmektir.

Kavramları unutturmak gibi sınırları belirsiz bir hedefle kıyaslandığında eğitim verilerindeki tekil örnekleri unutturmak ilk bakışta değerlendirilmesi daha kolay bir hedef gibi görülebilir. Ancak unutulması istenen örneklerin net bir biçimde tanımlandığı durumlarda bile unutmanın başarısını değerlendirmek zor olabilir. Bu durumun bir nedeni, eğitimler sırasında öğrenilen "genellemelerdir". Modele belirli bir örneği unutturmak için yapılan işlemler başarılı olsa bile bu, modelin o örnekle ilgili kötü tahminler yapacağı anlamına gelmez. Çünkü model unutturulmak istenenleri, eğitim setindeki başka bilgilerden de öğrenebilir. Model, eğitim sırasında çok sayıda başka örnekten öğrendiği genellemelerden yararlanarak hâlâ nitelikli çıktılar üretebilir. Dolayısıyla bir yapay zekâ modelinin unutturulmaya çalışılan örnekler hakkında hâlâ yüksek doğrulukta çıktılar üretmesine bakarak unutmanın başarısız olduğu kanısına varmak her zaman doğru değildir. Örneğin bitkileri sınıflandırmak için geliştirilmiş bir modele, telif haklarına tabi bir fotoğrafı unutturduktan sonra bile model hâlâ yüksek bir doğruluk oranıyla o fotoğraftaki ağaç türünü tanıyabilir. Çünkü eğitim setindeki çok sayıda başka örnekten edindiği bilgilerle o ağaç türü hakkında çeşitli genellemeler yapmıştır. Eğitim setindeki tek bir örnek unutturulsa bile model, başka örneklerden öğrendiği genellemelerden yararlanarak yüksek doğrulukla tahminler yapabilir.

Yapay sinir ağlarının kara kutu gibi çalışmasının yol açtığı bir sorun da unutulacak örneklerin net bir biçimde tanımlı olduğu durumlarda bile "unutulacak bilgilerle hiç karşılaşmamış" bir modelin nasıl olabileceği hakkında bir tanımlama yapmanın imkânsız oluşudur. Örneğin bir sohbet robotuna Wikipedia bilgilerinin unutturulmaya çalışıldığını düşünelim. Eğitimi sırasında hiç Wikipedia makalelerinin

kullanılmadığı bir model bulmak pek mümkün değildir. Dolayısıyla hiç Wikipedia makaleleriyle karşılaşmamış bir sohbet robotunun belirli konularda nasıl metinler üreteceği hakkında net bir fikir edinmek de mümkün değildir. Bu yüzden unutturma işlemleri tamamlandıktan sonra bir karşılaştırma yaparak unutturmanın başarısı net bir şekilde ölçülemez.

Unutmaya Alternatif Yöntemler

Bir şeyleri öğrenmiş bir insana öğrendiklerini unutturmak pek kolay değildir. Ancak sanki unutmuş gibi davranabilir. Benzer biçimde ChatGPT gibi günümüzün en gelişmiş sohbet robotları da sanki unutmuş numarası yapabiliyor. Elde edilmek istenen, belirli bilgileri unutmuş bir model gibi davranması için robota arzu edilen amaca uygun talimatlar verilmesi yeterli. Belki de makinelerden de insanların yaptığından fazlasını beklememek gerekiyor. Ancak bu yöntemin uygulama alanı, sadece kullanıcılardan talimat alabilen gelişmiş dil modelleriyle sınırlı.

Unutmaya alternatif bir başka yöntem ise "hiç öğrenmemek". Bu yöntemde eğitimler sırasında modele verilen bilgiler sınırlı tutulur. Bir gün modelin unutması gerekebilecek bilgiler ise bir veri tabanında saklanır. Model, kullanıcıların isteklerine cevap verirken gerektikçe veri tabanındaki bilgilerden de yararlanır. Eğer bir gün veri tabanındaki çeşitli bilgilerin, modelin erişimi dışında kalması gerekirse basitçe veri tabanındaki ilgili belgeleri silmek yeterli olur. Temel modelin unutturma amacıyla çeşitli işlemlerden geçirilmesi de gerekmez. Araştırmalar, bu modellerin performans açısından tüm bilgilerin eğitim setine dâhil edildiği modellerle rekabet edebileceğini gösteriyor. Ancak bu yöntemin de çeşitli dezavantajları olabilir. Örneğin veri tabanında tutulması gereken bilgi miktarı yüksekse temel modelin bilgi dağarcığı ve dolayısıyla işlevselliği azalacaktır. Ayrıca bilgileri bir kara kutuda kodlamak yerine bir veri tabanında tutmak, sistemi saldırılara daha açık hâle getirir.

Özet

Yapay zekâ modelleri geliştirmek uzun zaman alan masraflı bir iş. Dolayısıyla eğitimli bir model hatalı, güncelliğini yitirmiş ya da tehlikeli sonuçlara yol açabilecek bilgilere sahipse her şeye sıfırdan başlayıp yeni bir model geliştirmeye çalışmak pratik ve uygulanabilir değil. Ancak yapay zekâ modellerine özellikle de yapay sinir ağlarına dayalı modellere bir şeyler unutturmak da çok zor. Günümüzde makinelere bir şeyler unutturmak için kullanılan yöntemlerin bazıları kesin unutmayı amaçlıyor. Bu amaca verimli bir biçimde ulaşmak içinse, eğitim prosedürlerinin makine unutmasına uygun bir biçimde önceden planlanması gerekiyor. Bazı yöntemler ise orijinal modeli alıp unutulması istenen bilgileri unutmuş bir modele mümkün olduğu kadar yakın davranışlar sergileyen bir hâle dönüştürmeyi amaçlıyor. Bu yaklaşık unutma yöntemleri özellikle kavramlar ya da konular gibi bilgi uzayının sınırlarının tanımlanamadığı durumlarda yararlı oluyor. ■

Kaynaklar

- Liu, Ken Ziyu, "Machine Unlearning in 2024", https://ai.stanford.edu/~kzliu/blog/unlearning, 2024.
- Xu, H., ve ark., "Machine Unlearning: A Survey", *ACM Computing Surveys*, Cilt 56, Makale No: 9, 2023, https://dl.acm.org/doi/10.1145/3603620.
- Bourtoule, L., ve ark., "Machine Unlearning", 42nd IEEE Symposium of Security and Privacy, https://arxiv.org/abs/1912.03817, 2021.
- Goel, S., ve ark., "Corrective Machine Unlearning", *Transactions on Machine Learning Research*, https://arxiv.org/abs/2402.14015, 2024.
- Hine, E., ve ark., "Supporting Trustworthy AI Through Machine Unlearning", *Science and Engineering Ethics*, Cilt 30, Makale No:43, 2024, https://link.springer.com/article/10.1007/s11948-024-00500-5.