

Püf Noktaları ve İpuçları El Kitabı VIP

Afshine AMIDI ve Shervine AMIDI

April 30, 2019

Ayyüce Kızrak ve Yavuz Kömeçoğlu tarafından çevrilmiştir

Veri işleme

❑ **Veri artırma** – Derin öğrenme modelleri genellikle uygun şekilde eğilmek için çok fazla veriye ihtiyaç duyar. Veri artırma tekniklerini kullanarak mevcut verilerden daha fazla veri üretmek genellikle yararlıdır. Temel işlemler aşağıdaki tabloda özetlenmiştir. Daha doğrusu, aşağıdaki girdi görüntüsüne bakıldığında, uygulayabileceğimiz teknikler şunlardır:

Orijinal	Çevirme	Rotasyon	Rastgele kırpma
			
- Herhangi bir değişiklik yapılmamış görüntü	- Görüntünün anlamının korunduğu bir eksene göre çevrilmiş görüntü	- Hafif açılı döndürme - Yanlış yatay kalibrasyonu simüle eder	- Görüntünün bir bölümüne rastgele odaklanma - Arka arkaya birkaç rasgele kesme yapılabilir

Renk değişimi	Gürültü ekleme	Bilgi kaybı	Kontrast değişimi
			
- RGB'nin nüansları biraz değiştirilmesi - Işığa maruz kalırken oluşabilecek gürültü	- Gürültü ekleme - Girdilerin kalite değişkenliğine daha fazla toleranslı olması	- Yok sayılan görüntüler - Görüntünün parçalardaki olası kayıplarını kopyalanması	- Gün içindeki ışık ve renk değişimini kontrolü

❑ **Küme normalleştirme** – Bu, $\{x_i\}$ kümesini normalleştiren, γ, β hiperparametresinin bir adımıdır. μ_B ve σ_B^2 'ye dikkat ederek, kümeyi düzeltmek istediklerimizin ortalaması ve varyansı şu şekilde yapılır:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

Genellikle tam-tüm bağlı/evrimsimli bir katmandan sonra ve doğrusal olmayan bir katmandan önce yapılır. Daha yüksek öğrenme oranlarına izin vermeyi ve başlangıç durumuna güçlü bir şekilde bağımlılığı azaltmayı amaçlar.

Bir sinir ağının eğitilmesi

❑ **Dönem (Epok/Epoch)** – Bir modelin eğitimi kapsamında, modelin ağırlıklarını güncellemek için tüm eğitim setini kullandığı bir yinelemeye ifade etmek için kullanılan bir terimdir.

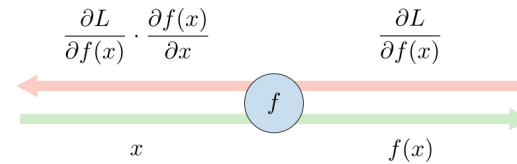
❑ **Mini-küme gradyan (bayır) iniş** – Eğitim aşamasında, ağırlıkların güncellenmesi genellikle hesaplama karmaşıklıkları nedeniyle bir kerede ayarlanan tüm eğitime veya gürültü sorunları nedeniyle bir veri noktasına dayanmaz. Bunun yerine, güncelleme adımı bir toplu işdeki veri noktalarının sayısının ayarlayabileceğimiz bir hiperparametre olduğu mini kümelerle yapılır. Veriler mini-kümeler halinde alınır.

❑ **Yitim fonksiyonu** – Belirli bir modelin nasıl bir performans gösterdiğini ölçmek için, L yitim (kayıp) fonksiyonu genellikle y gerçek çıktıların, z model çıktıları tarafından ne kadar doğru tahmin edildiğini değerlendirmek için kullanılır.

❑ **Çapraz-entropi kaybı** – Yapay sinir ağlarında ikili sınıflandırma bağlamında, çapraz entropi kaybı $L(z, y)$ yaygın olarak kullanılır ve şöyle tanımlanır:

$$L(z, y) = - \left[y \log(z) + (1 - y) \log(1 - z) \right]$$

❑ **Geriye yayılım** – Geri yayılım, asıl çıktıyı ve istenen çıktıyı dikkate alarak sinir ağındaki ağırlıkları güncellemek için kullanılan bir yöntemdir. Her bir ağırlığa göre türev, zincir kuralı kullanılarak hesaplanır.

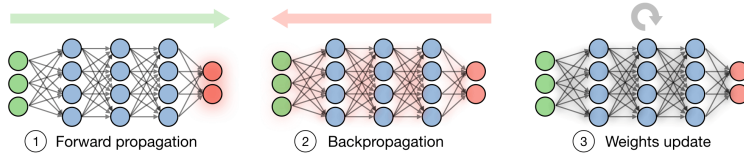


Bu yöntemi kullanarak, her ağırlık kurala göre güncellenir:

$$w \leftarrow w - \alpha \frac{\partial L(z, y)}{\partial w}$$

❑ **Ağırlıkların güncellenmesi** – Bir sinir ağında, ağırlıklar aşağıdaki gibi güncellenir:

- Adım 1: Bir küme eğitim verisi alın ve kaybı hesaplamak için ileriye doğru ilerleyin.
- Adım 2: Her ağırlığa göre kaybın derecesini elde etmek için kaybı tekrar geriye doğru yayın.
- Adım 3: Ağın ağırlıklarını güncellemek için gradyanları kullanın.



Parametre ayarlama

□ **Xavier başlangıcı (iklendirme)** – Ağırlıkları tamamen rastgele bir şekilde başlatmak yerine, Xavier başlangıcı, mimariye özgü özellikleri dikkate alan ilk ağırlıkların alınmasını sağlar.

□ **Transfer öğrenme** – Bir derin öğrenme modelini eğitmek çok fazla veri ve daha da önemlisi çok zaman gerektirir. Kullanım durumumuza yönelik eğitim yapmak ve güçlendirmek için günler/haftalar süren dev veri setleri üzerinde önceden eğitilmiş ağırlıklardan yararlanmak genellikle yararlıdır. Elimizdeki ne kadar veri olduğuna bağlı olarak, aşağıdakilerden yararlanmanın farklı yolları:

Eğitim boyutu	Görselleştirme	Açıklama
Küçük		Tüm katmanlar dondurulur, softmax'taki ağırlıkları eğitilir
Orta		Çoğu katmanlar dondurulur, son katmanlar ve softmax katmanı ağırlıkları ile eğitilir
Büyük		Önceden eğitilerek elde edilen ağırlıkları kullanarak katmanlar ve softmax için kullanılır

□ **Öğrenme oranı (adımı)** – Genellikle α veya bazen η olarak belirtilen öğrenme oranı, ağırlıkların hangi hızda güncellendiğini belirler. Sabitlenebilir veya uyarlanabilir şekilde değiştirilebilir. Mevcut en popüler yöntemin adı Adam'dır ve öğrenme hızını ayarlayan bir yöntemdir.

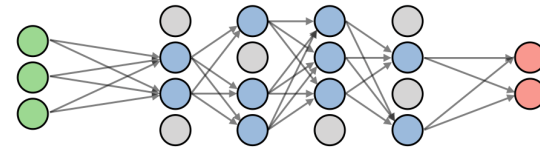
□ **Uyarlanabilir öğrenme oranları** – Bir modelin eğitilmesi sırasında öğrenme oranının değişmesine izin vermek eğitim süresini kısaltabilir ve sayısal optimum çözümü iyileştirebilir. Adam optimizasyonu yöntemi en çok kullanılan teknik olmasına rağmen, diğer yöntemler de faydalı olabilir. Bunlar aşağıdaki tabloda özetlenmiştir:

Yöntem	Açıklama	w	b
Momentum	- Osilasyonların azaltılması/ yumuşatılması - SGD (Stokastik Gradyan/Bayır İniş) iyileştirmesi - Ayarlanacak 2 parametre	$w - \alpha v_{dw}$	$b - \alpha v_{db}$
RMSprop	- Ortalama Karekök yayılımı - Osilasyonları kontrol ederek öğrenme algoritmasını hızlandırır	$w - \alpha \frac{dw}{\sqrt{s_{dw}}}$	$b \leftarrow b - \alpha \frac{db}{\sqrt{s_{db}}}$
Adam	- Uyarlanmalı Moment tahmini/kestirimi - En popüler yöntem - Ayarlanacak 4 parametre	$w - \alpha \frac{v_{dw}}{\sqrt{s_{dw}} + \epsilon}$	$b \leftarrow b - \alpha \frac{v_{db}}{\sqrt{s_{db}} + \epsilon}$

Not: diğer yöntemler içinde Adadelat, Adagrad ve SGD.

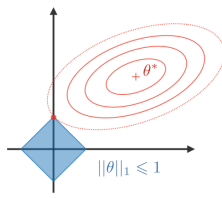
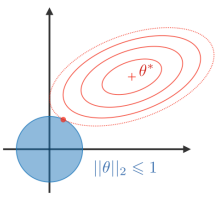
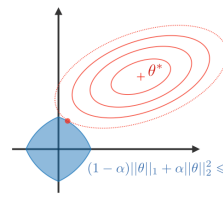
Düzenleştirme

□ **Seyreltme** – Seyreltme, sınır ağlarında, $p > 0$ olasılıklı nöronları silerek eğitim verilerinin fazla kullanılmaması için kullanılan bir tekniktir. Modeli, belirli özellik kümelerine çok fazla güvenmekten kaçınmaya zorlar.



Not: Çoğunlukla derin öğrenme kütüphaneleri, 'keep' ('tutma') parametresi $1 - p$ aracılığıyla seyreltmeyi parametrize eder.

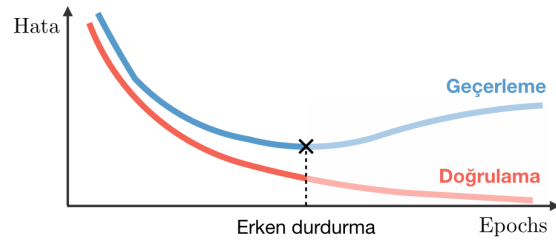
□ **Ağırlık düzenleştirme** – Ağırlıkların çok büyük olmadığından ve modelin eğitim setine uygun olmadığından emin olmak için, genellikle model ağırlıklarında düzenleştirme teknikleri uygulanır. Temel olanlar aşağıdaki tabloda özetlenmiştir:

LASSO	Ridge	Elastic Net
<ul style="list-style-type: none"> - Katsayıları 0'a düşürür - Değişken seçimi için iyi 	Katsayıları daha küçük yapar	Değişken seçimi ile küçük katsayılar arasında ödünleşim sağlar
		
$\dots + \lambda \theta _1$ $\lambda \in \mathbb{R}$	$\dots + \lambda \theta _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda \left[(1 - \alpha) \theta _1 + \alpha \theta _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$

	Sayısal gradyan	Analitik gradyan
Formül	$\frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{2h}$	$\frac{df}{dx}(x) = f'(x)$
Açıklamalar	<ul style="list-style-type: none"> - Maliyetli; Kayıp, boyut başına iki kere hesaplanmalı - Analitik uygulamanın doğruluğunu anlamak için kullanılır - Ne çok küçük (sayısal dengesizlik) ne de çok büyük (zayıf gradyan yaklaşımı) seçimi yapılmalı, bunun için ödünleşim gerekir 	<ul style="list-style-type: none"> - 'Kesin' sonuç - Doğrudan hesaplama - Son uygulamada kullanılır

* * *

□ **Erken durdurma** – Bu düzenleme tekniği, onaylama kaybı bir sabillığe ulaştığında veya artmaya başladığında eğitim sürecini durdurur.



İyi uygulamalar

□ **Küçük kümelerin ezberlenmesi** – Bir modelde hata ayıklama yaparken, modelin mi-marisinde büyük bir sorun olup olmadığını görmek için hızlı testler yapmak genellikle yararlıdır. Özellikle, modelin uygun şekilde eğitilebildiğinden emin olmak için, ezberleyecek mi diye görmek için ağ içinde bir mini küme ile eğitilir. Olmazsa, modelin normal boyutta bir eğitim setini bırakmadan, küçük bir kümeyi bile ezberleyecek kadar çok karmaşık ya da yeterince karmaşık olmadığı anlamına gelir.

□ **Gradyanların kontrolü** – Gradyan kontrolü, bir sinir ağının geriye doğru geçişinin uygulanması sırasında kullanılan bir yöntemdir. Analitik gradyanların değerini verilen noktalardaki sayısal gradyanlarla karşılaştırır ve doğruluk için bir kontrol rolü oynar.