Derin Öğrenme El Kitabı VIP

Afshine Amidi ve Shervine Amidi

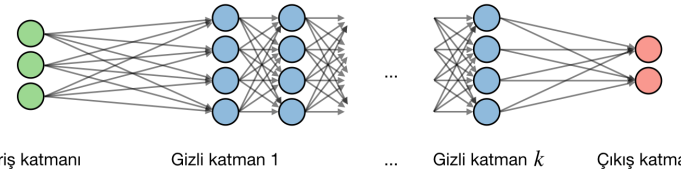
April 30, 2019

Ekrem Çetinkaya ve Omer Bukte tarafından çevrilmiştir

Sinir Ağları

Sinir ağları, katmanlarla inşa edilen bir modeller sınıfıdır. Sinir ağlarının yaygın kullanılan çeşitleri evrişimsel sinir ağları ve yinelenen sinir ağlarını içerir.

❒ **Mimari** – Sinir ağları mimarisi aşağıdaki figürde açıklanmaktadır:

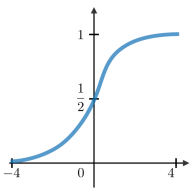
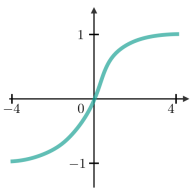
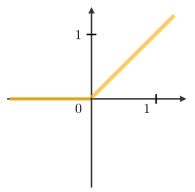
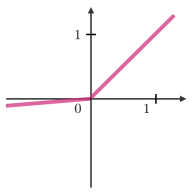


Ağın i. sırasındaki katmana i ve katmandaki j. sırasındaki gizli birime j dersek, elimizde:

$$\boxed{z_j^{[i]} = {w_j^{[i]}}^T x + b_j^{[i]}}$$

burada w, b, z değerleri sırasıyla ağırlık, eğilim ve ürünü temsil eder.

❒ **Etkinleştirme fonksiyonu** – Etkinleştirme fonksiyonları gizli birimlerin sonunda, modele lineer olmayan karmaşıklıklar katmak için kullanılır. Aşağıda en yaygın kullanılanlarını görebilirsiniz:

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0,z)$	$g(z) = \max(\epsilon z, z)$ ile $\epsilon \ll 1$
			

❒ **Çapraz-entropi kaybı** – Sinir ağları içeriğinde, çapraz-entropi kaybı $L(z,y)$ sık olarak kullanılır ve aşağıdaki gibi tanımlanır:

$$\boxed{L(z,y) = -\Big[y\log(z) + (1-y)\log(1-z)\Big]}$$

❒ **Öğrenme oranı** – Öğrenme oranı, sıklıkla α veya bazen η olarak belirtilir, ağırlıkların hangi tempoda güncellendiğini gösterir. Bu derece sabit olabilir veya uyarlamalı olarak değişebilir. Mevcut en gözde yöntem Adam olarak adlandırılan ve öğrenme oranını uyarlayan bir yöntemdir.

❒ **Geri yayılım** – Geri yayılım sinir ağındaki ağırlıkları güncellemek için kullanılan ve bunu yaparken de asıl sonuç ile istenilen sonucu hesaba katan bir yöntemdir. Ağırlık w değerine göre türev, zincir kuralı kullanılarak hesaplanılır ve aşağıdaki şekildedir:

$$\boxed{\frac{\partial L(z,y)}{\partial w} = \frac{\partial L(z,y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}}$$

Sonuç olarak, ağırlık güncellenmesi aşağıdaki gibidir:

$$\boxed{w \longleftarrow w - \eta \frac{\partial L(z,y)}{\partial w}}$$

❒ **Ağırlıkları güncelleme** – Sinir ağında ağırlıklar, aşağıdaki gibi güncellenir:

- 1. Adım: Bir eğitim verisi kümesi alınır.
- 2. Adım: Denk gelen kaybı elde etmek için, ileri yayılım gerçekleştirilir.
- 3. Adım: Gradyanları elde etmek için kayba geri yayılım uygulanır.
- 4. Adım: Ağın ağırlıklarını güncellemek için gradyanlar kullanılır.

❒ **Düşürme** – Düşürme, eğitim verisinin aşırı uymasını engellemek için sinir ağındaki birimleri düşürmek yoluyla uygulanan bir tekniktir. Pratikte, nöronlar ya p olasılığıyla düşürülür ya da $1-p$ olasılığıyla tutulur.

Evrişimsel Sinir Ağları

❒ **Evrişimsel katman gereksinimleri** – Girdi boyutuna W, evrişimsel katman nöronlarının boyutlarına F, sıfır dolgulama miktarına P dersek, belirlenmiş bir boyuta sığacak neuron sayısı N şu şekildedir:

$$\boxed{N = \frac{W - F + 2P}{S} + 1}$$

❒ **Küme normalleştirmesi** – γ, β Hiper-parametresinin, $\{x_i\}$ kümesini normalleştiren bir adımıdır. μ_B, σ_B^2 ifadelerine düzeltmek istediğimiz kümenin ortalaması ve varyansı dersek, normalleştirme işlemi şu şekilde yapılır:

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

Bu işlem genelde tamamıyla bağlantılı/evrişimsel olan bir katmandan sonra ve lineer olmayan bir katmandan önce yapılır. Bu işlem ile daha yüksek öğrenme derecesi elde etmeye imkan sağlamak ve de öndeğer atamaya olan güçlü bağımlılığı azaltmak amaçlanır.

Yinelenen Sinir Ağları

❒ **Kapı çeşitleri** – Aşağıda tipik bir yinelenen sinir ağlarında karşımıza çıkan farklı kapı örnekleri görülebilir:

Girdi kapısı	Unutma kapısı	Kapı	Çıktı kapısı
Hücreye yaz/yazma ?	Hücreyi sil/silme ?	Hücreye ne kadar yazmalı ?	Hücrenin ne kadarını açığa çıkarmalı ?

❒ **LSTM** – Uzun, kısa vadeli hafıza (LSTM) ağı, 'unutma' kapılarını ekleyerek yok olan gradyan probleminden kurtulabilen bir çeşit RNN modeldir.

Pekiştirmeli Öğrenme ve Kontrol

Pekiştirmeli öğrenmenin hedefi, bir hedefin bir ortamda nasıl değişiklik geçireceğini öğrenmesini sağlamaktır.

❒ **Markov karar süreci** – Markov karar süreci (MDP) 5 öğelidir $(\mathcal{S},\mathcal{A},\{P_{sa}\},\gamma,R)$ ve bu ifadeler şunları temsil eder:

- $\mathcal{S}$, hallerin setidir
- $\mathcal{A}$, aksiyonların setidir
- $\{P_{sa}\}$ $s \in \mathcal{S}$ ve $a \in \mathcal{A}$ için hal değişimlerinin olasılıklarıdır
- $\gamma \in [0,1[$ azaltma unsurudur
- $R : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$ veya $R : \mathcal{S} \longrightarrow \mathbb{R}$ algoritmanın en yüksek düzeye çıkartmak istediği ödül fonksiyonudur

❒ **Prensip** – π prensibi hal-aksiyon eşleşmesini yapan $\pi : \mathcal{S} \longrightarrow \mathcal{A}$ fonksiyonudur.

Dipnot: Eğer s hali verildiğinde $a = \pi(s)$ aksiyonunu uyguluyorsak, π prensibini yerine getirdik deriz.

❒ **Değer fonksiyonu** – π prensibi ve s hali verildiğinde, V^π değer fonksiyonu aşağıdaki gibi tanımlanır:

$$\boxed{V^\pi(s) = E\Big[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + ...|s_0 = s,\pi\Big]}$$

❒ **Bellman denklemi** – Ideal Bellman denklemleri, ideal prensip π^* değerinin değer fonksiyonu V^{π^*} değerini simgeler:

$$\boxed{V^{\pi^*}(s) = R(s) + \max_{a\in\mathcal{A}} \gamma \sum_{s'\in S} P_{sa}(s')V^{\pi^*}(s')}$$

Dipnot: s hali verildiğinde, ideal π^* prensibini şu şekilde tanımlarız:

$$\boxed{\pi^*(s) = \underset{a\in\mathcal{A}}{\text{argmax}} \sum_{s'\in\mathcal{S}} P_{sa}(s')V^*(s')}$$

❒ **Değer iterasyon algoritması** – Değer iterasyon algoritması iki adımdan oluşur:

- Değere ilk değer atarız:

$$\boxed{V_0(s) = 0}$$

- Daha önceki değerlere göre değere iterasyon uygularız:

$$\boxed{V_{i+1}(s) = R(s) + \max_{a\in\mathcal{A}} \left[\sum_{s'\in\mathcal{S}} \gamma P_{sa}(s')V_i(s')\right]}$$

❒ **Maksimum ihtimal tahmini** – Maksimum ihtimal hal geçişi olasılıklarını aşağıdaki şekilde tahmin eder:

$$\boxed{P_{sa}(s') = \frac{\#\text{times took action } a \text{ in state } s \text{ and got to } s'}{\#\text{times took action } a \text{ in state } s}}$$

❒ **Q-Öğrenimi** – Q-Öğrenimi modelden bağımsız bir Q tahmini yapılan bir yöntemdir ve aşağıdaki gibi yapılır:

$$\boxed{Q(s,a) \leftarrow Q(s,a) + \alpha\Big[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\Big]}$$