**Hopfield Networks:**

**1.1**

Hopfield ağı, asenkron bir geri beslemeli sinir ağıdır ve özellikle hafıza ve optimizasyon problemlerini çözmek için kullanılır. İlk olarak John Hopfield tarafından 1982'de tanıtılmıştır.

Hopfield ağları, bir dizi bağlantılı düğümden (nöron) oluşan bir ağ yapısına sahiptir. Her düğüm, diğer düğümlere bağlanan ağırlıklı bağlantılarla ilişkilendirilmiştir. Her düğüm, girdi sinyallerini alır, bunları işler ve bir çıkış sinyali üretir.

Hopfield ağları, iki temel işlevi gerçekleştirir: hafıza ve asenkron dinamik. Hafıza özelliği sayesinde, ağ belirli desenleri hatırlayabilir ve daha sonra bu desenleri hatırladığı girdilerle eşleştirebilir. Asenkron dinamik özelliği, ağın asenkron bir şekilde güncellenmesine ve her düğümün zamanla dinamik olarak değişmesine olanak tanır.

Hopfield ağları, bir enerji fonksiyonu üzerinden çalışır. Enerji fonksiyonu, ağın belirli bir durumunun enerjisini temsil eder ve ağ, enerjiyi en aza indirmeye çalışır. Bu, ağın, enerji fonksiyonunun lokal minimum noktalarına doğru ilerlerken bir iterasyon süreci ile güncellendiği dinamik davranışını açıklar.

Hopfield ağları, özellikle hafıza ve optimize etme problemleri için kullanılır, örneğin desen tanıma, veri sıkıştırma ve kombinatoryal optimizasyon gibi uygulamalarda başarıyla kullanılabilirler.

**Terimler:**

Kombinatoryal optimizasyon: Birçok olası çözüm kombinasyonu arasından en iyi çözümü bulmak için matematiksel ve hesaplamalı yöntemlerin kullanıldığı bir optimizasyon alt dalıdır. Kombinatoryal optimizasyon problemleri, belirli bir problemi en iyi şekilde çözmek için bir dizi karar değişkeni veya parametre arasından seçim yapma ihtiyacını içerir.

Asenkron geri beslemeli sinir ağı: Sinir ağlarının bir türüdür ve geri besleme bağlantıları içerir. Geri beslemeli sinir ağları, çıkışların daha önceki durumlarına bağlı olarak güncellendiği ve girdi sinyallerini tekrar kullanarak iteratif bir şekilde güncellendiği ağlardır.

**1.2**

**Nedir?**

Asenkron bir geri beslemeli sinir ağıdır. Yani bir tür sinir ağıdır ve bu türün özelliği yapılan çıkışlar sonucunda oluşan değerleri tekrar tekrar kullanıp artımlı bir şekilde durumlar güncellenir.

**Ne işe yarar?**

Hopfield ağları, genel olarak hafıza, desen tanıma, veri sıkıştırma, kombinatoryal optimizasyon ve en yakın komşu eşleştirmesi gibi problemlerde kullanılabilir ve birçok farklı uygulama alanında potansiyel çözümler sunabilir.

**Neden bilmeliyiz?**

Hopfield ağları, örüntü tanıma ve depolama yoluyla insan hafızasını simüle etme kavramı ile ilişkilidir. Bununla birlikte birçok soruna çözüm sunar. Hafıza, optimizasyon problemlerine çözüm bulabilmek için bilmeliyiz.

**Gerçek hayatta kullanılıyor mu?**

Evet kullanılıyor. Şifreleme işlemlerinde, karakter-desen eşleştirmede, bozuk şekillerin yeniden oluşturulmasında, el yazı tanımlama sistemlerinde kullanılmaktadır.

**Kaynak:**

ChatGPT

<https://medium.com/@batincangurbuz/hopfield-ağ-modeli-hopfield-network-hn-ccf1548ca432>

https://tr.theastrologypage.com/hopfield-network

https://link.springer.com/book/9780387310732

**Boltzmann Machine**

**2.1**

Boltzmann Machine (Boltzmann Makinesi), yapay sinir ağları alanında bir tür olası durum ağı (probabilistic graphical model) olan enerji tabanlı bir istatistiksel modeldir. Boltzmann Makinesi, ağın gizli ve görünür (gözlemlenen) katmanlarını içeren iki katmanlı bir ağ yapısına sahiptir ve bu katmanlar arasındaki bağlantılar çift yönlüdür. Boltzmann Makinesi, her bir gizli ve görünür birimi belirli bir olasılıkla etkinleşebilir veya etkisizleştirilebilir. Bu nedenle, Boltzmann Makinesi, olası durumların geniş bir yelpazesini temsil edebilir ve genel olarak olasılık temelli bir modele dayanır.

Boltzmann Makinesi, ağdaki gizli ve görünür birimlerin durumlarını, enerji fonksiyonu olarak adlandırılan bir fonksiyonla temsil eder. Enerji fonksiyonu, ağın parametreleri tarafından belirlenir ve ağın belirli bir durumdaki enerji seviyesini hesaplar. Boltzmann Makinesi, enerji fonksiyonu kullanarak ağın durumları arasında geçişleri olasılıklarla modelleyen bir stokastik süreç olarak çalışır. Belirli bir durumdaki ağın olasılığını hesaplamak için Boltzmann dağılımı kullanılır.

Boltzmann Makinesi, Markov zinciri Monte Carlo (MCMC) yöntemleri gibi özel algoritmalara dayanarak ağın istatistiksel özelliklerini öğrenme yeteneğine sahiptir. Öğrenme süreci, ağın enerji fonksiyonunu optimize etmek için veriye dayalı bir süreçtir. Öğrenme tamamlandığında, Boltzmann Makinesi, ağın gizli birimlerini kullanarak verileri yeniden üretebilir ve bu nedenle veri üretimi, örnekleme, hatta modelleme ve tahminleme gibi bir dizi uygulamada kullanılabilir.

Boltzmann Makinesi, derin öğrenme alanının temel taşlarından biridir ve birçok farklı uygulama alanında kullanılır, örneğin görüntü işleme, doğal dil işleme, öneri sistemleri ve karmaşık veri modellenmesi gibi alanlarda başarıyla uygulanmıştır.

**Terimler:**

Markov zinciri Monte Carlo (MCMC): Karmaşık olasılık dağılımlarını keşfetmek için kullanılan bir istatistiksel simülasyon yöntemidir. MCMC, istatistiksel çıkarımlar ve model tahminlemesi gibi birçok uygulama alanında kullanılır.

Regresyon: Bir değişkenin diğer bir veya daha fazla değişkenle ilişkisini inceleyen bir istatistiksel analiz yöntemidir. Genellikle bir bağımlı değişkenin, bir veya daha fazla bağımsız değişkenle ilişkisini modellemek ve tahmin etmek için kullanılır.

**2.2**

**Nedir?**

Sınıflandırma, regresyon ve özellik öğrenimi işlemlerini yapan Boltzmann makinesi giriş veri seti üzerinde olasılıksal dağılımları öğrenebilen bir sinir ağıdır.

**Ne işe yarar?**

Zor kombinatoryal problemleri temsil edip çözebilirler. Boltzmann makineleri, makine öğrenmesi veya çıkarımında pratik sorunlar için yararlı değildir, ancak eğer bağlantı düzgün bir şekilde kısıtlıysa, öğrenme pratik problemler için faydalı olacak kadar verimli hale getirilebilir.

**Neden bilmeliyiz?**

Boltzmann makinesi hakkında bilgi sahibi olmak, makine öğrenimi alanında daha derin bir anlayış geliştirmemizi, enerji tabanlı modelleme ve stokastik hesaplamalar hakkında daha fazla bilgi edinmemizi ve çeşitli uygulama alanlarında model seçimi ve kullanımında avantaj elde etmemizi sağlayabilir.

**Gerçek hayatta kullanılıyor mu?**

Boltzmann makinesi, genellikle teorik bir model olarak kullanılmaktadır ve gerçek hayatta doğrudan uygulanması nadir görülmektedir. Örneğin, Gibbs Örneklemesi adlı bir yöntem, Boltzmann makinesinden türetilmiş bir örnekleme yöntemidir ve Markov Zinciri Monte Carlo (MCMC) adlı bir istatistiksel yöntemde sıkça kullanılır. Ancak, Boltzmann makinesinin doğrudan pratik uygulamalarda kullanılması yaygın değildir. Çoğunlukla teorik ve araştırma düzeyinde kullanılmaktadır ve gerçek hayatta kullanımı sınırlıdır.

**Kaynak:**

ChatGPT

<https://medium.com/@batincangurbuz/boltzman-makinesi-boltzmann-machines-bm-ve-kısıtlanmış-boltzmann-makineleri-restricted-boltzmann-6f2e0f839b05>

<https://tr.theastrologypage.com/boltzmann-machine>

<https://devhunteryz.wordpress.com/2018/07/25/kisitli-boltzmann-makineleri/>

**Markov Assumption ve Markov Chain**

**3.1**

Markov ön kabulü (Markov assumption), bir sürecin gelecekteki durumunun sadece şu anki duruma bağlı olduğunu ve geçmiş durumlardan bağımsız olduğunu varsayan bir istatistiksel ön kabuldür. Bu ön kabul, Markov zinciri adı verilen bir süreç sınıfını temsil eder.

Markov zinciri (Markov Chain), belirli bir durum kümesinde zaman içinde geçişlerin rastgele olarak gerçekleştiği stokastik bir süreçtir. Markov zinciri, bugünkü durumun, yalnızca bir önceki duruma bağlı olduğu bir süreçtir ve geçmiş durumlardan bağımsızdır. Bu nedenle, Markov zinciri, gelecekteki durumları tahminlemek için sadece şu anki durumu kullanır.

Markov zincirleri, birçok uygulama alanında kullanılır, örneğin zaman serileri analizi, doğal dil işleme, makineler arası iletişim ve rastgele süreçlerin modellemesi gibi alanlarda yaygın olarak kullanılır. Markov zincirleri, olasılık teorisi ve istatistik alanlarında önemli bir konudur ve birçok farklı matematiksel ve istatistiksel yöntemle incelenmiştir.

**Terimler:**

Markov zincirleri: Zaman içinde belirli bir durum kümesi üzerinde rastgele geçişlerin gerçekleştiği stokastik bir süreçtir. Örneğin, zaman serileri analizi, doğal dil işleme, makineler arası iletişim ve rastgele süreçlerin modellemesi gibi alanlarda Markov zincirleri yaygın olarak kullanılır.

**3.2**

**Nedir?**

Markov varsayımı, bir süreç veya sistemin mevcut durumunun, sadece anlık durumundan ve birkaç önceki durumundan bağımsız olduğunu ifade eden bir istatistiksel varsayımdır.

**Ne işe yarar?**

Markov varsayımı ve Markov zincirleri, birçok uygulama alanında kullanılan önemli istatistiksel araçlardır ve çeşitli problemleri modellemek ve çözmek için kullanılabilirler.

**Neden bilmeliyiz?**

Zaman serileri analizi, Doğal dil işleme, Hesaplamalı istatistik, Rastgele süreçlerin modellenmesi, Makine öğrenimi gibi yapay zekanın kullanıldığı alanlarda durumların modellenmesi için bilmemizde fayda vardır.

**Gerçek hayatta kullanılıyor mu?**

Kullanılıyor. Google’ın kullandığı internet sayfalarının PageRank'i Markov zinciri ile tanımlanmaktadır. Markov modelleri aynı zamanda kullanıcıların internet gezinti davranışlarını analiz etmek için de kullanılmıştır. Bir kullanıcının bir internet sayfasından web bağlantısı ile geçişi, birincil ya da ikincil Markov modelleri ile modellenebilir ve gelecekteki hareketleri öngörmede ve dolayısıyla internet sayfasını kullanıcı için kişiselleştirme de kullanılabilir. Metin üretiminden finansal modellemeye kadar birçok farklı alanda kullanılmıştır. Fakat en çok kullanılan alanları ise metin oluşturma ve otomatik tamamlama uygulamalarıdır.

**Kaynak:**

ChatGPT

<https://medium.com/@batincangurbuz/markov-zinciri-markov-chain-mc-33cd8a61f6fa>

<https://egealpay1.medium.com/hidden-markov-models-saklı-markov-modeli-b381380d0aca>

<https://bilgisayarkavramlari.com/2009/06/17/markof-modeli-markov-model/>

https://tr.wikipedia.org/wiki/Markov\_zinciri#:~:text=Matematikte%2C%20Markov%20Zinciri%20(Andrey%20Markov,durumlardan%20bağımsız%20olması%20anlamına%20gelir.

Aşağıdaki yöntemleri kullanarak training ve validation loss değerlerini yükseltebilirsiniz:

1. Daha düşük bir learning rate kullanın: Learning rate, ağırlık güncellemelerinin ne kadar büyük adımlarla yapıldığını belirler. Daha düşük bir learning rate kullanmak, modelin daha yavaş bir şekilde güncellenmesine ve daha stabil bir eğitim sürecine yol açabilir. Bu, loss değerlerinin daha yüksek seviyelerde kalmasını sağlayabilir.
2. Daha karmaşık bir model kullanın: Daha karmaşık bir model, daha fazla parametreye ve daha fazla kapasiteye sahip olabilir, ancak aynı zamanda daha fazla overfitting eğiliminde olabilir. Overfitting, modelin eğitim verilerine aşırı uyum sağlaması ve genelleme performansının düşmesine yol açabilir. Daha karmaşık bir model kullanarak, modelin daha fazla hatalı tahmin yapmasına ve loss değerlerinin yükselmesine neden olabilirsiniz.
3. Daha az veri kullanın: Daha az veri kullanmak, modelin genel performansını düşürebilir. Eğitim veri setinizin boyutunu azaltmak, modelin daha az doğru tahmin yapmasına ve loss değerlerinin yükselmesine neden olabilir.
4. Daha az epok kullanın: Daha az epok kullanmak, modelin daha az eğitim verisi üzerinde güncellenmesine ve daha az dengeye ulaşmasına neden olabilir. Bu, loss değerlerinin daha yüksek seviyelerde kalmasına yol açabilir.