

Matris Manipülasyonu Nedir?

Bir matrisi, **bilgisayar programlarında** ya da **matematiksel hesaplamalarda** analiz edebilmek için üzerinde **birçok işlem** yapılır. Bu işlemler sayesinde sistem çözümleri yapılabilir, veriler dönüştürülebilir veya yapay zekâ modelleri eğitilebilir. Bu işlemlerin tamamına "**matris manipülasyonu**" denir.

Matris Manipülasyonu İşlemleri

Temel İşlemler:

1. Toplama ve çıkarma

- İki matrisin aynı boyutta olması koşuluyla her eleman aynı konumdaki elemanla toplanır veya çıkarılır.

2. Skaler ile çarpma

- Bir matrisin her bir elemanı sabit bir sayı ile çarpılır.

3. Matris çarpımı (dot product)

- İki matrisin uygun şekilde çarpılması.

Boyutlar önemli: $A(m \times n)$ ve $B(n \times p)$ ise $A \cdot B = C(m \times p)$

4. Transpoz alma (transpose)

- Satırları sütun, sütunları satır yapma: A^T

5. Tersini alma (inverse)

- Bir kare matrisin çarpıldığında birim matris verecek halini bulmak: A^{-1}

6. Determinant hesaplama

- Bir matrisin kareliğini (tersi var mı yok mu gibi) anlamamıza yarar.

Gelişmiş İşlemler:

7. Matris dilimleme (slicing)

- Büyük matrisin içinden küçük parçalar seçmek (özellikle veri işleme için önemli).

8. Matrisin boyutunu değiştirme (reshape)

- Matrisin yapısını, eleman sayısını koruyarak başka boyuta dönüştürmek.

9. Sıfır veya birim matris oluşturma

- 00'lardan oluşan veya çaprazı 1 olan matris üretmek (algoritmalarda kullanılır).

10. Özdeğer ve özvektör hesaplama

- Matrisin karakteristik özelliklerini çıkarmak (özellikle yapay zekâ ve fiziksel sistemlerde).

11. Singular Value Decomposition (SVD)

- Matrisleri üç parçaya ayırarak boyut indirgeme ve veri sıkıştırma da kullanılır.

12. Rank (rütbe) bulma

- Matrisin satırlarının/ sütunlarının bağımsızlık derecesi.

13. Norm hesaplama

- Bir matrisin büyüklüğünü ölçmek (özellikle optimizasyon ve yapay zekada kullanılır).

Özetle:

Matris manipülasyonu, matrisleri **dönüştürmek, analiz etmek ve anlamlı hale getirmek için** yapılan tüm işlemleri kapsar. Bu işlemler; toplama, çarpma gibi temel işlemlerden, özdeğer hesaplama, boyut indirgeme gibi ileri düzey tekniklere kadar uzanır.

Matris Manipülasyonunun Makine Öğrenmesi ile İlişkisi

1. Matris Nedir ve Neden Önemlidir?

Makine öğrenmesinde veriler genellikle matrisler olarak temsil edilir. Yani, her gözlem (örnek) bir satıra ve her özellik (feature) bir sütuna karşılık gelir. Bu şekilde, tüm veri seti matematiksel olarak bir matris halini alır.

Matrisler, verilerin dili gibidir. Bu yapı sayesinde bilgisayarlar veri üzerinde aritmetik işlemler yapabilir, örüntüleri analiz edebilir ve öğrenme algoritmalarını çalıştırabilir.

2. Makine Öğrenmesinde Matrislerin Kullanım Alanları

a. Veri Temsili ve Modelleme

- Her veri kümesi $n \times m$ boyutunda bir matris olarak temsil edilir.

(n : gözlem sayısı, m : özellik sayısı)

- Örneğin, bir doğrusal regresyon modelinde hedef değişken (y) ile giriş matrisinin (X) çarpımı kullanılarak tahmin yapılır:

$$\hat{y} = X \cdot w$$

b. Dönüşümler ve Projeksiyonlar

- PCA (Principal Component Analysis) gibi yöntemlerde veriler, daha düşük boyutlara projekte edilir.
- Bu işlemler matris çarpımıyla gerçekleştirilir ve verinin temel yapısını koruyarak boyut indirgemeye olanak tanır.

c. Ağırlıkların ve Katsayıların Hesaplanması

- Makine öğrenmesi algoritmaları, genellikle parametreleri (ağırlıkları) matris işlemleriyle günceller.
- Gradient descent gibi optimizasyon algoritmaları, ağırlık vektörünü güncellerken matris türevlerinden faydalanır.

3. Determinant ve Makine Öğrenmesi Bağlantısı

Makine öğrenmesinde determinant:

- Bir matrisin tersinin alınabilir olup olmadığını gösterir. Bu bilgi, özellikle:
- Regresyon çözümleri
- Boyut indirgeme teknikleri
- Korelasyon matrislerinin invertlenmesi gibi işlemlerde önemlidir.

Örneğin, $(X^T X)^{-1}$ işlemi determinant sıfırta yapılamaz (matris terslenemez). Bu, modelin eğitilmesini engeller.

4. Pratik Uygulamalar

- Yüz tanıma sistemleri, yüksek boyutlu görüntü verisini özvektörler kullanarak temsil eder (örneğin Eigenfaces).
 - Doğrusal Regresyon, matris işlemleriyle model parametrelerini tahmin eder.
 - Sinir ağları, ağırlık matrisleri üzerinden verileri katman katman geçirerek öğrenme yapar.
-

Özdeğer (Eigenvalue) ve Özvektör (Eigenvector) Nedir?

Bir matrisin belirli bir vektörü, bu matrisle çarpıldığında yalnızca yönü değişmeden aynı doğrultuda kalıyorsa, bu vektöre **özvektör**, bu çarpımın sonucunda elde edilen skaler katsayıya ise **özdeğer** denir.

Özvektör (Eigenvector)

Bir kare matris **A** ile çalışılırken, öyle bir vektör **v** vardır ki, bu vektörü matrisle çarptığımızda yine kendisinin bir katı çıkıyorsa, **v** bir **özvektördür**.

Matematiksel olarak:

$$Av = \lambda v$$

Burada:

- A** : kare matris (n x n boyutunda)
- v** : sıfırdan farklı bir vektör (**özvektör**)
- λ** (lambda): bir skaler sayı (**özdeğer**)

Özdeğer (Eigenvalue)

Yukarıdaki eşitlikte **v** özvektör olduğunda, **λ** o vektöre karşılık gelen **özdeğer**dir. Yani, **A** matrisinin **v** yönündeki etkisi yalnızca uzunluğunu (veya yönünü) değiştiren skaler çarpandır.

Makine Öğrenmesinde Özdeğer ve Özvektörlerin Rolü

Makine Öğrenmesinde Özdeğer ve Özvektör Ne İşe Yarar?

1. PCA (Principal Component Analysis) - Boyut İndirgeme

Makine öğrenmesinde çoğu veri çok boyutludur. Örneğin, bir yüz görüntüsü 100x100 pikselde 10,000 özellik olur. Ancak bu özelliklerin çoğu gereksiz/gürültülü olabilir. PCA ile:

- Veriyi merkeze alırız.
- Kovaryans matrisi oluştururuz.
- Bu matrisin özdeğer ve özvektörlerini hesaplarız.
- En büyük özdeğere sahip özvektörler en çok bilgi içeren yönleri gösterir.

2. Veri Görselleştirme

Yüksek boyutlu veriler PCA ile indirgenerek insan gözünün anlayabileceği şekilde görselleştirilebilir.

3. Gürültü Azaltma

Küçük özdeğerli bileşenler az varyansa sahiptir. Bu bileşenleri atmak gürültüyü azaltır ve modelin overfitting yapmasını önler.

4. Veri Sıkıştırma

Büyük veri setlerinde en büyük birkaç özdeğer seçilerek veri minimum kayıpla sıkıştırılabilir.

5. Doğrusal Dönüşümleri Anlamak

Makine öğrenmesinde matris işlemleri çok yaygındır. Özdeğerler, bu işlemlerin veriye ne yaptığına dair bilgi verir.

6. Sinir Ağları

Sinir ağlarında eğitim sırasında bazı yönler aşırı büyür bazıları küçülür. Bu analiz, özdeğerlerle yapılabilir. Özellikle Hessian matrisi gibi yapılar öğrenme oranını etkileyebilir.

7. Terslenebilirlik ve Determinant

Bir matrisin özdeğerlerinden biri sıfırsa matris terslenemez. Bu durum, birçok ML algoritmasında sorun oluşturur.

Özet

Makine öğrenmesinde özdeğer ve özvektörlerin temel uygulama alanları şunlardır:

- PCA (Boyut İndirgeme): En bilgilendirici yönlerin seçimi.
 - Görselleştirme: Veriyi anlaşılır hale getirme.
 - Gürültü Azaltma: Önemsiz bileşenlerin atılması.
 - Sinir Ağı Eğitimi: Optimizasyon ve öğrenme analizi.
 - Matris Terslenebilirliği: Model hesaplamalarının doğruluğu.
-

Kaynakça

1. Gabel, K. *Understanding Eigenvalues and Eigenvectors in Computer Vision*. LinkedIn. <https://www.linkedin.com/pulse/understanding-eigenvalues-eigenvectors-computer-vision-kanishka-gabel>
 2. Brownlee, J. *Introduction to Eigendecomposition, Eigenvalues and Eigenvectors*. Machine Learning Mastery. <https://machinelearningmastery.com/introduction-to-eigendecomposition-eigenvalues-and-eigenvectors/>
 3. *Eigenvalues and Eigenvectors and Their Use in Machine Learning and AI*. Medium – Almonks. <https://medium.com/aimonks/eigenvalues-and-eigenvectors-and-their-use-in-machine-learning-and-ai-c7a5431ae388>
 4. *Applications of Eigenvalues and Eigenvectors*. GeeksforGeeks. <https://www.geeksforgeeks.org/applications-of-eigenvalues-and-eigenvectors/>
 5. *Determinant (mathematics)*. Britannica. <https://www.britannica.com/science/determinant-mathematics>
 6. *The Convergence of Linear Algebra and Machine Learning*. Developer Nation. <https://www.developernation.net/blog/the-convergence-of-linear-algebra-and-machine-learning/>
 7. Bhandari, D. *How Do Matrices Work in Machine Learning? Unlocking the Power of Linear Algebra*. LinkedIn. <https://www.linkedin.com/pulse/how-do-matrices-work-machine-learning-unlocking-power-deepak-bhandari-yvjqc>
 8. *Matrix Manipulation in Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/matrix-manipulation-python/>
 9. Sonderegger, D. *Matrix Manipulation*. Bookdown. <https://bookdown.org/dereksonderegger/571/1-matrix-manipulation.html#types-of-matrices>
 10. Brownlee, J. *Introduction to Matrices for Machine Learning*. Machine Learning Mastery. <https://machinelearningmastery.com/introduction-matrices-machine-learning/>
-