



Makine Öğrenmesi

8. hafta

- Takviyeli Öğrenme (Reinforcement Learning)
 - Q Öğrenme (Q Learning)
 - TD Öğrenme (TD Learning)
- Öğrenen Vektör Parçalama (LVQ)
 - LVQ2
 - LVQ-X

Yrd. Doç. Dr. Umut ORHAN

1



Takviyeli Öğrenme

Takviyeli öğrenme (Reinforcement Learning), bulunduğu ortamı algılayan ve kendi başına kararlar alabilen bir sistemin, hedefine ulaşabilmesinde doğru kararlar almayı nasıl öğrenebileceğini gösterir. Bu yöntem robotik, oyun programlama, hastalık tespiti ve fabrika otomasyonu gibi alanlarda sıklıkla kullanılır.

Yrd. Doç. Dr. Umut ORHAN

2

Takviyeli Öğrenme

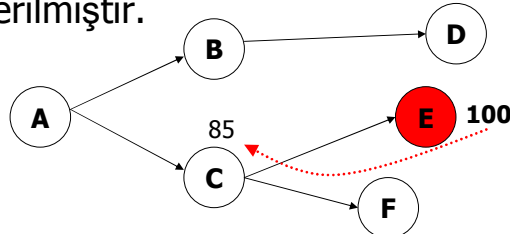
Takviyeli öğrenmede bir eğitmen bulunur fakat denetimli öğrenmedeki gibi sisteme çok detay vermez veya veremez. Bunun yerine öğrenen sistem bir karar verdiğinde bu kararın doğru olduğu durumlar için sistemi ödüllendirir ve yanlışlar için de cezalandırır. Amaç, öğrenen sistemin denediği olası durumların hedef olup olmadığının kontrolü ve denenen doğru veya yanlış tüm durumların hatırlanmasıdır.

Yrd. Doç. Dr. Umut ORHAN

3

Takviyeli Öğrenme

Karar verilen durumlar ardarda gelen diziler şeklinde hatırlanırsa sonunda başarıya ulaşan duruma bağlı olarak hatırlanan ardışıl durumlar dizisindeki her karara büyük ödülün hisseler dağıtılır. Aşağıda örnek bir durum ağacı gösterilmiştir.



Yrd. Doç. Dr. Umut ORHAN

4

Takviyeli Öğrenme

Ödül veya cezayı belirleyen genellikle bir değer (amaç veya hedef de denilir) fonksiyonu (V) vardır. Davranış politikası (Π) ile t anındaki durumda iken (s_t) yapılabilecek hareketlerden (a_t) optimumu seçilebilir. Takviyeli öğrenme, değer fonksiyonunun ürettiği en büyük ödüle sahip davranış politikasını tercih eder. Optimum davranış politikasının tercihi şöyle ifade edilir.

$$\Pi = \arg \max_{a_t} (V(s_t, a_t))$$

Yrd. Doç. Dr. Umut ORHAN

5

Q Öğrenme

Takviyeli öğrenme yaklaşımları içerisinde en çok bilineni Q-Öğrenme yöntemidir. Sıklıkla labirent ve arama problemlerine uygulanır. Bu yöntemi 1989'da ilk kez öneren Watkins değer fonksiyonu için Q harfini kullanması dolayısıyla yöntem bu adı almıştır. Kullanılacak değer fonksiyonunun (Q) matematiksel modelinin genellikle kolayca belirlenebilir olmaması yüzünden sadece anlık duruma bağlı ödüller kullanılır.

Yrd. Doç. Dr. Umut ORHAN

6

Q Öğrenme

Mevcut duruma sezgisel olarak veya rasgele bir değer (r_t) verilebilir. Fakat ödülün gerçek değeri en büyük ödüle ulaşınca netleşir. Büyük ödüle ulaşan hareketler dizisi ödülün hareket uzaklığı oranında yararlanır. Büyük ödülün yararlanma oranı (γ) genelde $[0, 1]$ arasında bir değer olarak seçilir. Değer fonksiyonunun güncellenmesi aşağıdaki denklem ile gerçekleşir.

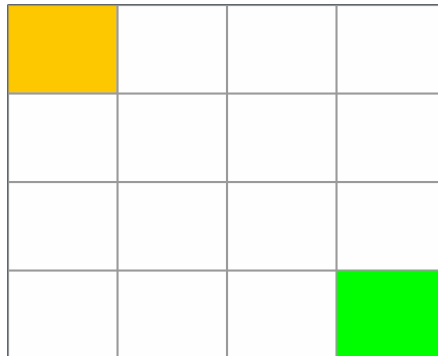
$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

Yrd. Doç. Dr. Umut ORHAN

7

Örnek

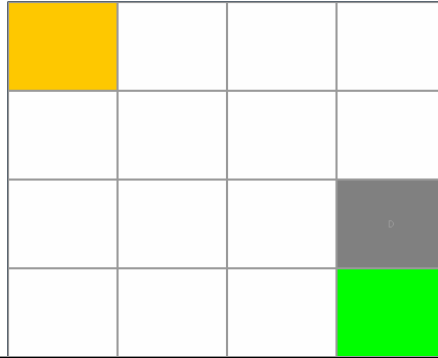
Klasik bir problem olarak bir robotun 4x4 bir matris üzerinde sol üst köşeden başlayarak sağ alt köşedeki çıkışı arama adımlarını izleyelim.



8

Örnek

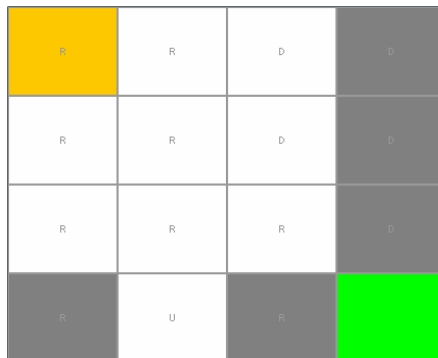
Robot rasgele tahminlerle hücreler arasında gezinirken çıkışı keşfediyor. Çıkışa hangi hücreden geçti ise onu işaretliyor.



9

Örnek

Robot rasgele gezinmelere devam ederken her seferinde bulduğu 1 hücreyi işaretliyor.



10

Örnek

Tüm iterasyonlar bittiğinde en kısa yolu aşağıdaki gibi çiziyor.

R	R	D	D
R	R	D	D
R	R	R	D
R	U	R	

11

TD Öğrenme

1988'de Sutton, Q algoritmasında delta kuralını kullanarak bir iyileşme yapıp TD (Temporal Difference) öğrenme adıyla sunmuştur. Delta kuralında eğim azaltma için kullanılan (η) parametresi, büyük ödüle giden birden fazla yolun kesişimindeki durumları Q öğrenmesine göre daha değerli hale getirmektedir.

$$Q_t = Q(s_t, a_t)$$

$$Q_t = Q_t + \eta \left(r_{t+1} + \gamma \max_{a_{t+1}} Q_{t+1} - Q_t \right)$$

Yrd. Doç. Dr. Umut ORHAN

12



Sarsa: Aktif Politikalı TD

TD öğrenme yöntemi bir sonraki en iyi hareket değerine göre davranış politikasını seçtiği için pasif politikalı yöntem olarak adlandırılır. Sarsa gibi aktif politikalı yöntemlerde ise bir sonraki davranış politikası da belirlenebilir. Dolayısıyla problemin durum ağacı üzerinde 1 seviye daha derine indiği söylenebilir.



ÖDEV

Sarsa algoritmasını örnek bir uygulama için MATLAB ile hazırlayınız.



Takviyeli Öğrenen YSA: LVQ

YSA mimarisine sahip olan LVQ (Learning Vector Quantization) modeli, takviyeli öğrenmeden esinlenerek ortaya konulan danışmanlı bir öğrenme yöntemidir.



Takviyeli Öğrenen YSA: LVQ

Bazen örneklerle ait sınıf değeri bilinemez ve bu yüzden öğrenen modele hedef değerleri vermek mümkün olmaz. Bununla beraber modelin ürettiği çıktının doğru veya yanlış olduğu belirtilebiliyorsa takviyeli öğrenme ile sınıflandırma yapılabilir.



LVQ Ağı

LVQ ağında amaç $N \times M$ boyutlu bir matrisi M boyutlu birkaç vektöre haritalamaktır.

Öğrenme işlemi, girdi matrisini temsil edecek vektörler kümesini tespit etmek için yürütülür.

Diğer bir deyişle LVQ ağının görevi öğrenme yolu ile referans vektörler denilen bu temsilcileri belirlemektir.

Yrd. Doç. Dr. Umut ORHAN

17



LVQ Ağının Katmanları

LVQ ağı 3 katmandan oluşmaktadır:

- Girdi katmanı: Dışarıdan alınan örnekler (girdi vektörleri) ağı gösterilir.
- Kohonen katmanı: Her eleman sınıflı bir referans vektörüdür. Girdi vektörüne en yakın olan referans vektörü burada belirlenir.
- Çıktı katmanı: Her sınıf için bir çıkış bulunur. Sadece belirlenen sınıfa ait çıkış aktif yapılır.

Yrd. Doç. Dr. Umut ORHAN

18



LVQ Ağında Öğrenme

- LVQ ağında girdi vektörlerinin sınıfları, referans vektörleri üzerinde uygulanan en yakın 1 komşu kuralına göre belirlenir.
- Uzaklık ölçütü olarak genellikle Öklit uzaklığı kullanılır.
- Sonuca bağlı olarak her iterasyonda bir referans vektörü güncellenir.

Yrd. Doç. Dr. Umut ORHAN

19



LVQ Ağında Öğrenme

- Girdi vektörü ile referans vektörler arasındaki mesafeler hesaplanır.
- En kısa mesafeyi sağlayan referans vektör belirlenir.
- Bu referans vektörün sınıfı çıkışta aktif edilir.
- Belirlenen referans vektörü, sonucun doğruluğuna göre belirli bir oranda girdi vektörüne yaklaştırılır veya uzaklaştırılır.

Yrd. Doç. Dr. Umut ORHAN

20

LVQ Ağında Öğrenme

Girdi vektörüne en yakın referans vektörü şu ifadeyle belirlenir.

$$i = \arg \min_j \|x_t - v_j\|$$

Çıkış doğruysa referans vektör güncellenir.

$$v_i = v_i + \lambda(x_t - v_i)$$

Çıkış yanlışsa;

$$v_i = v_i - \lambda(x_t - v_i)$$

Yrd. Doç. Dr. Umut ORHAN

21

LVQ Ağında Öğrenme

Öğrenme katsayısı λ genelde $[0 \ 1]$ aralığında seçilir.

YSA mimarisine sahip olan LVQ modeli aynı zamanda geriyayılmadaki eğitim azaltmayı da kullanır.

Dolayısıyla λ değeri her iterasyonda azaltılır.

Yrd. Doç. Dr. Umut ORHAN

22

LVQ Özet

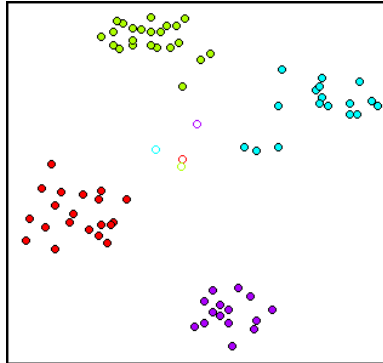
- Ağın öğrenmesi temel çok katmanlı ağlardan daha hızlı ve kolaydır. Doğrusal olmayan problemlerde de kullanılabilir.
- Öğrenme katsayısının sıfır değerine inmemesi durumunda ağ doğru ağırlık değerlerinden uzaklaşır ve öğrendiklerini unutabilir.
- Bazı problemlerde aynı referans vektörü tehlikeli şekilde sürekli olarak kazanır.
- İki sınıfın arasında veya sınırlara yakın vektörlerin sınıfı belirlenemeyebilir.

Yrd. Doç. Dr. Umut ORHAN

23

Örnek

Kümeleme testlerinde iyi bilinen Ruspini veri kümesinde LVQ eğitimi izleyelim.



Yrd. Doç. Dr. Umut ORHAN

24



LVQ2 Ağı

LVQ ağı, farklı sınıflara ait noktaların arasında veya uç sınır bölgelerde kalan örnek noktaları sınıflandırmada zorlanır. Bu konuda bir iyileştirme olarak LVQ2 önerilmiştir.

Hatalı sınıflandırılan örneklere en yakın iki referans vektör tespit edilir. Girdi vektörü bu ikisinin sınırlandığı pencere içinde ise ve vektörler farklı sınıflardansa güncellenir.

Yrd. Doç. Dr. Umut ORHAN

25



LVQ-X Ağı

- LVQ2 ağının eğitiminin iyileştirilmiş halidir. Yanlış sınıflandırma yapıldığında iki referans vektörü birden güncellenir.
- Bu işlem, öğrenme hızını artırır, öğrenme zamanını kısaltır ve ağın genelleme yeteneğini artırır.

Yrd. Doç. Dr. Umut ORHAN

26



LVQ-X Ağı

LVQ-X ağında, her iterasyonda, yarışmayı kazanan iki tane işlem elemanı bulunur. Bunlar;

- Global: Girdi vektörüne en yakın referans vektörüne sahip olanı göstermektedir.
- Yerel: Girdi ile aynı sınıftan referans vektörlerinden girdi vektörüne en yakın olanı ifade etmektedir.



LVQ-X Ağı

Girdi örneğine daha yakın olan referans vektör, örnek ile farklı sınıftandır ve bu yüzden uzaklaştırılmalıdır.

Girdi, doğru sınıflandırılabilsin diye diğer vektör de örneğe daha yakın hale getirilmelidir.

$$v_1 = v_1 + \lambda(x_i - v_1)$$
$$v_2 = v_2 - \lambda(x_i - v_2)$$