Q-LEARNİNG İLE YOL PLANLAMASI

Furkan ÇETİNER

180201042

fcetinerrr@gmail.com

Oğuzhan BORLAK

180201075

oguzhanborlak@hotmail.com

1-Özet

Pekiştirmeli öğrenme (reinforcement learning), öznelerin (agent) bir görevi en yüksek kazançla tamamlayabilmek için hangi eylemleri gerçekleştirmeleri gerektigi ile ilgilenen bir makine öğrenmesi tekniğidir. Bu tür öğrenme algoritmaların girdisi öznelerin görev yapacakları farklı durumlardan oluşan bir ortam, yapabilecekleri eylemler, ortamdaki durumuna göre yapabilecekleri eylemleri belirleyen prensipler, bir durumdan diğer duruma geçtiklerinde elde edecekleri kazançtır.

2-Giriş

Uygulama çalıştırıldığında karşımıza %30'u engellerden oluşan 50*50'lik bir matris ve hesapların yapılmasını sağlayacak butonların bulunduğu bir ara yüz gelmektedir. Kullanıcı matris üzerinden mouse'un sol tuşuna tıklayarak istediği kareyi başlangıç, mouse'un sağ tuşuna tıklayarak bitiş noktasını seçebilir. "R MATRİSİ" butonuna tıklandığında o karenin durumuna göre puanlar r matrisine eklendi. "Q MATRİSİ" butonuna tıklandığında oluşturulan r matrisini kullanarak q matrisi oluşturulmaktadır. "HAREKET" butonuna tıklandığında oluşturulan Q matrisi değerlerine göre en yüksek q değerleri seçilerek en kısa yol haritada çizdirilmektedir. "ADIM GRAFİĞİ" butonuna tıklandığında ise ajanın ilk 100 yolda yaptığı adım sayıları grafikte gösterilmektedir. "MALİYET GRAFİĞİ" butonuna tıklanıldığında ise ajanın en kısa yolda kazandığı q değerleri grafikte gösterilmektedir.

3-Yöntem

Projede Java programlama dili kullanılmıştır. Proje geliştirilirken ajanın öğrenebilmesi için Q-learning algoritması kullanılmıştır. Kullanıcı ara yüzü için Java Swing kütüphanesinden faydalanılmıştır.

public void DuvarUret() {

2500 karenin %30u için rastgele kareler seçilerek setceza true yap.

}

public void DuvarBas() {

Eğer isceza true ise arkaplanı siyah yap.

Değilse arkaplanı beyaz yap.

}

public void R_Matrisi(){

Öncelikle matrisin tüm değerlerini -5 yap.

Sol karelerin kontrolü için eğer hedef sol kare boşluksa 0, duvarsa -5, ödülse 100 değerini r matrisine ekle.

Sağ karelerin kontrolü için eğer hedef sağ kare boşluksa 0, duvarsa -5, ödülse 100 değerini r matrisine ekle.

Üst karelerin kontrolü için eğer hedef üst kare boşluksa 0, duvarsa -5 ödülse 100 değerini r matrisine ekle.

Alt karelerin kontrolü için eğer hedef alt kare boşluksa 0, duvarsa -5 ödülse 100 değerini r matrisine ekle.

Sol üst karelerin kontrolü için eğer hedef sol üst kare boşluksa 0, duvarsa -5 ödülse 100 değerini r matrisine ekle.

Sağ üst karelerin kontrolü için eğer hedef sağ üst kare boşluksa 0, duvarsa -5 ödülse 100 değerini r matrisine ekle.

Sol alt karelerin kontrolü için eğer hedef sol alt kare boşluksa 0, duvarsa -5 ödülse 100 değerini r matrisine ekle.

Sağ alt karelerin kontrolü için eğer hedef sağ alt kare boşluksa 0, duvarsa -5 ödülse 100 değerini r matrisine ekle }

void Q_Doldur(){

Öncelikle q matrisinin her indeksini 0.0 ile doldur. }

void Q_Matrisi() {

for(i 100e kadar)

{

While(anlık durum hedef olana kadar){

int[] aksiyon = olasiaksiyonlar(anlikdurum);

olasiaksiyonlar fonksiyonuna ajanın anlık konumunu yolla.

int index = rand.nextInt(aksiyon.length);

int sonrakidurum = aksiyon[index];

indekslerden birini sonraki duruma eşitle.

q = Q[anlikdurum][sonrakidurum];

maxQ = maxQ(sonrakidurum);

r = R[anlikdurum][sonrakidurum];

tanımlamalarını yaptıktan sonra

r + (0.9 * maxQ) formülü ile q matrisini doldur.

anlikdurum = sonrakiduruma eşitle.

boolean sondurummu fonksiyonu ile robotun konumunun hedef olup olmadığı kontrol edilir.

synchronized void hareket() {

while(ajanın konumu hedef konumuna eşit olana kadar){

geçişler dizisine olasiaksiyonlar fonksiyonundan dönen değerleri ekle.

Geçişlerin uzunluğuna kadar olası q değerlerini arrayliste ekle.

En büyük q değerini bir sonraki adım seç.

if (en büyük q değerinin bulunduğu kare hedef değilse){

o karenin arka planını sarı yap.
}

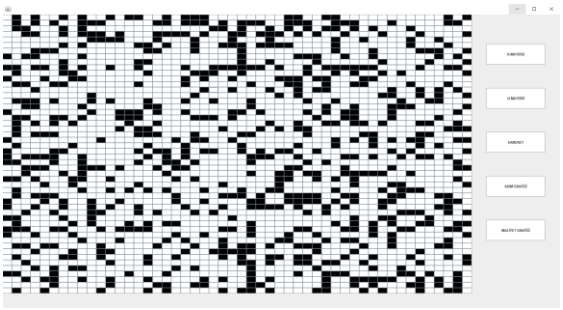
ajanın konumunu güncelle.}}

int[] olasiaksiyonlar fonksiyonu ile ajanın gidebileceği kareleri belirle.

double maxQ fonksiyonu en büyük değerler geri döndürülür.

4- Deneysel Sonuçlar

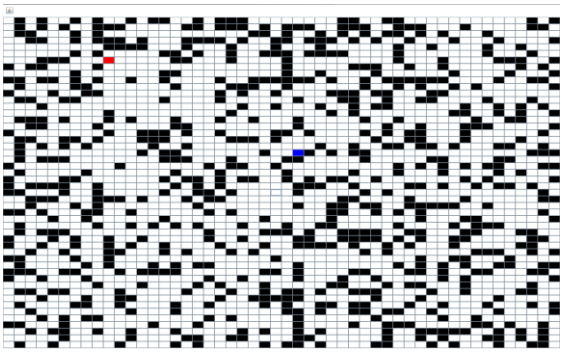
Uygulamanın giriş ekranı bu şekilde 50*50'lik bir matris ve butonlar bulunmaktadır.



Harita oluşturulduktan sonra engel.txt dosyasına her karenin durumu yazdırılmıştır.

```
(0,0,W)
(0,1,B)
(0,2,W)
(0,3,B)
(0,4,W)
(0,5,W)
(0,6,B)
(0,7,W)
(0,8,B)
(0,9,W)
(0,10,W)
(0,11,B)
(0,12,W)
(0,13,B)
(0,14,B)
(0,15,W)
(0,16,W)
(0,17,B)
(0,18,W)
(0,19,B)
(0,20,B)
(0,21,B)
(0,22,W)
(0,23,W)
(0,24,W)
(0,25,W)
(0,26,W)
(0,27,W)
(0,28,W)
(0,29,W)
(0,30,B)
(0,31,B)
(0,32,W)
```

Mousenin sol tıkı ile ajan konumubelirlenmektedir ve rengi kırmızıdır.Sağ tıkı ile hedef konumu belirlenmektedir ve rengi laciverttir.

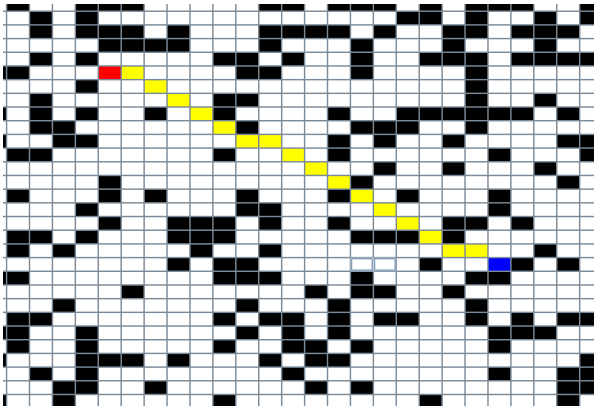


Butonlar bu şekildedir.

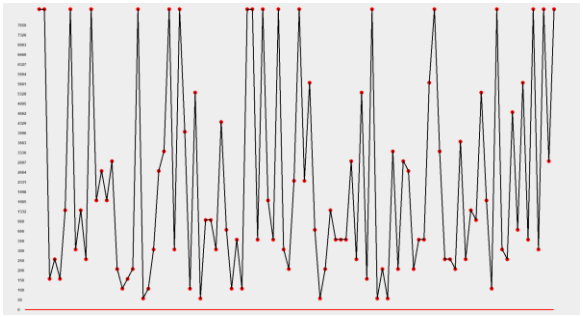
“R MATRİSİ” butonu ile seçilen başlangıç ve bitiş noktasına göre duvar puanları, boş geçiş puanları ve hedef puanı r matrisine eklenmektedir.

“Q MATRİSİ” butonu ile Q matrisi hesaplanır.

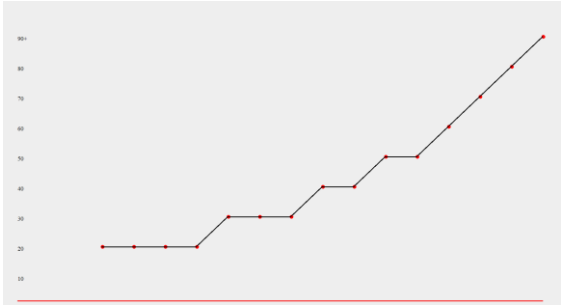
“HAREKET” butonu ile Q matrisine göre hesaplanan en kısa yol ara yüzde çizdirilir.



“ADIM GRAFİĞİ” butonu ile hedefe ulaşılan ilk yüz yolda ajanın yaptığı adım sayıları grafikte gösterilir.



“MALİYET GRAFİĞİ” butonu ile hedefe ulaşılan en kısa yolda ajanın kazandığı q değerleri grafikte gösterilir.



5-Sonuç

Projede istenildiği gibi 50 * 50'İlk matriste, matrisin rastgele %30'luk kısmına engel oluşturup, oluşturulan engellerin indeksleri ve duvar olup olmadığı engel.txt dosyasına yazdırılması sağlanmıştır.

Grafiksel ara yüzde belirlenen yollar, engeller ve duvarlar gösterilmiştir.

Kullanıcıdan bu ara yüzde ajan başlangıç noktası, hedef noktası seçebilmektedir.

Başlangıç karesinden hedef kareye giden en kısa yol grafiksel ara yüzde gösterilerek yol planı grafik üzerinde çizdirilmiştir.

İlk yüz yol için ajanın hedef noktaya ulaşıncaya kadar yaptığı adım sayıları grafiği çizdirilmiştir.

En kısa yol için ajanın hedef noktaya ulaşıncaya kadar topladığı q değerleri grafiği çizdirilmiştir.

6-Kaynakça

https://github.com/technobium/q-learning-java

https://medium.com/deep-learning-turkiye/q-learninge-giri%C5%9F-6742b3c5ed2b

https://medium.com/deep-learning-turkiye/python-i%CC%87le-q-learning-ef6413aa896e

https://www.youtube.com/watch?v=EnCoYY087Fc&t=786s&ab_channel=TheCodingLibrary