Bursa Teknik Üniversitesi – Bilgisayar Mühendisliği (Yüksek Lisans)
24435004012 - Serkan BULUT
Derin Pekiştirmeli Öğrenme Final Ödevi

DEEP Q LEARNING İLE BEŞ ARACIN BULUNDUĞU TESİSTE ÜÇ VARDİYA HALİNDE ARAÇLARA PERSONEL ATANMASININ SAĞLANMASI

Genel Tanım

Elimizde mevcut araçlara uygun olacak şekilde çeşitli ehliyetlere sahip personel bulunmaktadır. Bu personelin üç vardiya halinde birbirinden farklı beş aracın kullanımı için ataması yapılacaktır.

Oyun Tanımı

15X15 kareden oluşan bir oyun tahtası üzerinde, solda yukarıdan aşağıya 1'den 3'e doğru olacak şekilde araçlar, yukarıda 1'den 15'e kadar personel bulunacaktır. Oyun (0,0) konumundan yani oyun tahtasının sol üst köşesinden başlayacaktır. Agent oyun tahtasında gezinen boş bir kare olarak düşünülebilir.

Genel Değerler

Araç sayısı: 5
Vardiya sayısı: 3
Personel sayısı: 15 (örnek olarak 13 de değerlendirilecek.)
State sayısı: Araç sayısı X Vardiya sayısı X Personel sayısı => 5X3X15 = 225
Eylemler (Actions): 0-Sağa Git 1-İşaretle 2-ÖncekiVardiyayıYaz
Ödüller (Rewards): Oyun tahtasındaki her bir hareket için -2, uygun işaretlemeler için -1, satır sonuna gelinmiş ve işaretleme yok ise yani araca atama yapılmamışsa -3, ehliyetin uygun olmadığı işaretlemeler için -3, ataması önceden yapılmış bir araca yeniden işaretleme yapıldığında -5, önceden ataması yapılmış araca önceki vardiyadan personel ataması yapılırsa -5, boşta atanabilecek personel varken önceki vardiyadan personel atanıyorsa -5, tüm araçlara atama yapılmışsa 500.

Çevre (Environment): Oyun alanında agent sağa doğru ilerler, işaretleme veya önceki vardiyanın personelini yazma eylemlerini yapabilir. (0,0) noktasında (14,14) noktasına doğru ilerleyerek araçlara atama yapması beklenir. -1 ve 500 ödül puanı dışındaki ödül puanlarını aldığında aynı karede yani statede kalması sağlanır. Mevcut satır bir aracı temsil ettiğinden agent bu satırda işaretleme yada önceki vardiya ataması yaparsa bir alt satırın başına geçer. Son satırda bu eylemler olursa oyun biter.

```
ehliyetler = ['X','E','X','B','B','B','C','E','C','D','D','D','E','C','X']
personelid = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
araclar    = ['A','B','C','D','E','A','B','C','D','E','A','B','C','D','E']
```

	1.P	2.P	3.P	4.P	5.P	6.P	7.P	8.P	9.P	10.P	11.P	12.P	13.P	14.P	15.P
A1			X												
B1				X											
C1							X								
D1										X					
E1		X													
A2			X												
B2					X										
C2									X						
D2											X				
E2								X							
A3			X												
B3						X									
C3														X	
D3												X			
E3													X		

Oyunun sonunda yukarıdaki örneğe benzer bir tablo oluşması beklenir. Sarı ile işaretle araçlara bir önceki vardiyanın personeli atanmıştır.
Sonucun yazılı hali;
aracpersonel=[3, 4, 7, 10, 2, 3, 5, 9, 11, 8, 3, 6, 14, 12, 13] şeklinde bir dizidir.

Deep Q Learning

DQN (Deep Q-Network), Q-learning algoritmasının derin sinir ağlarıyla birleştirilmiş halidir. DQN'de, Q-değerlerini tablo yerine bir yapay sinir ağı tahmin eder. Bu ağ genellikle tam bağlantılı (fully connected - FC) katmanlardan oluşur. Bunlar; giriş katmanı (state temsil edilir), bir veya birkaç gizli katman, ve çıkış katmanı (her bir eylem için Q değeri bulunur).

Bu öğrenme ile daha büyük state sayılarına sahip oyunlaştırmalar yapılabilir. Ajanın deneyimleri bir **Replay Memory** (tekrar belleği) içinde saklanır; bu da öğrenme sırasında verinin çeşitliliğini artırarak korelasyonları azaltır.

Ağ, belirli aralıklarla bu bellekten örnekler alarak **optimize()** fonksiyonu aracılığıyla eğitilir. Kayıp fonksiyonu ile genellikle hedef Q değeri ile tahmin edilen Q değeri arasındaki fark minimize edilmeye çalışılır. Ayrıca, öğrenmenin istikrarını artırmak için hedef ağ (target network) kullanılır ve bu ağ, ana ağdan belirli aralıklarla kopyalanarak güncellenir.

Daha basit ifadeyle; ajan başta random ilerledikçe deneyimleri arasından seçtiği actionlar ile ödül-ceza elde eder. Bu tecrübesini belleğe kaydeder. Önceki ödül ve tahmin edilen sonraki ödül değerleri ile iki network bu tecrübeler ile optimize edilir. Yani öğrenim sağlanır.

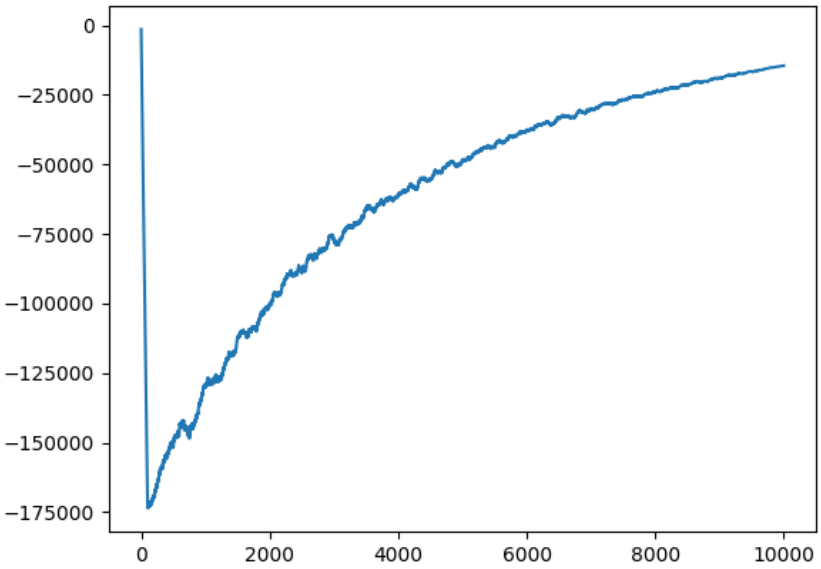
Eğitim sonucu bir sinir ağı modelinde tutulur. Test aşamalarında bu modelden actionlar çağrılır.

Bu Projede Deep Q Learning

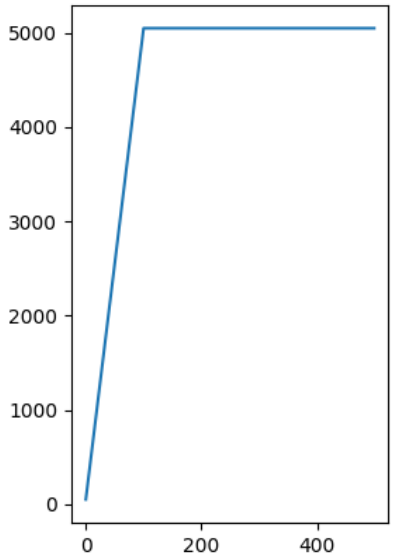
learning_rate_a = 0.001	learning rate (alpha)
discount_factor_g = 0.9	discount rate (gamma) Bu değer büyüdükçe uzun vade düşünür. Küçük ise anlık ödüllerin peşinden gider.
network_sync_rate = 10	policy and target network kaç adımda bir güncellenecek. Her güncellemede policy target'a kopyalanır.
replay_memory_size = 1000	Bellek boyutu
mini_batch_size = 32	Replay memory'den eğitim için rastgele seçilecek örnek sayısı.

Q learning ile DQL için aynı Vardiya environmenti kullanıldı. Sonuç grafikleri aşağıdadır.

Q Learning Grafiği



DQN Grafiği



Bu iki eğitim süresi arasında fark olduğu görülmektedir. Q learning eğitimi daha fazla episode ihtiyaç duymaktadır. Deep Q Learning daha kısa zamanda eğitimi tamamlamıştır.