OSTİM TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
YAPAY ZEKÂ MÜHENDİSLİĞİ BÖLÜMÜ

LunarLander-v3 Ortamında DQN, Double DQN ve Dueling DQN Karşılaştırmalı Analizi

Reinforcement Learning
(YZM 403)
Final Projesi Raporu

Hazırlayan:
Aziz Deniz Akmermer
220212037

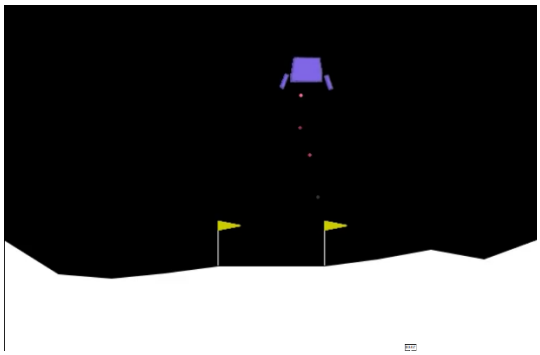
Dersi Veren:
Asst. Prof. Dr. Haydar Kılıç

2025 Güz Dönemi

1. ORTAM VE GÖREV

Ortam Adı

Gymnasium kütüphanesinde yer alan **LunarLander-v3** simülasyon ortamı kullanılmıştır. LunarLander-v3, iki boyutlu bir fizik tabanlı kontrol problemi olup bir uzay aracının belirlenen iniş pedleri arasına güvenli bir şekilde indirilmesini amaçlayan standart bir pekiştirmeli öğrenme ortamıdır.



Ajanın Amacı

Ajanın temel amacı, sınırlı yakıt ve motor gücü koşulları altında uzay aracını dengeli, kontrollü ve güvenli bir biçimde iniş pedleri arasına indiren optimal kontrol politikasını öğrenmektir. Ajan, bu hedef doğrultusunda uzun vadeli kümülatif ödülü maksimize etmeye çalışır.

Episode Bitiş Koşulları

- Uzay aracının iniş pedleri arasına **başarılı ve dengeli** bir şekilde iniş yapması
- Uzay aracının kontrolsüz şekilde yere çarpması veya devrilmesi
- Uzay aracının ortam sınırları dışına çıkması
- Ortam tarafından tanımlanan maksimum adım sayısına ulaşılması

Episode sonunda elde edilen toplam ödül, ajanın ilgili bölümdeki performansını temsil etmektedir.

2. PROBLEM TANIMI

Durum Uzayı (State Space)

Ajan, her zaman adımında 8 boyutlu sürekli bir durum vektörü gözlemler. Bu vektör; aracın konumu, hızları, açısal durumu ve iniş ayaklarının temas bilgilerini özetler. Bu çalışmada durum uzayı boyutu kod tarafında state_size = 8 olarak doğrulanmıştır.

Eylem Uzayı (Action Space)

Ajan, her adımda 4 adet ayrık eylemden birini seçer (action_size = 4). Bu eylemler; motor ateşleme kararlarını temsil eder (örn. motor kullanmama / ana motor / yan motorlar gibi). Bu çalışmada eylem uzayı boyutu kod tarafında env.action_space.n = 4 olarak doğrulanmıştır.

Ödül Tanımı (Reward Definition)

Ortamın kendi (built-in) ödül fonksiyonu kullanılmıştır; özel bir ödül yeniden-tasarımı yapılmamıştır. Ajanın hedefi, her episode boyunca aldığı ödüllerin toplamını (kümülatif ödül) maksimize etmektir. Ödül sinyali genel olarak:

- **Başarılı ve kontrollü inişi teşvik edecek** şekilde pozitif katkılar,
- **Çarpma / devrilme gibi başarısız durumları** cezalandıran negatif katkılar,
- **Gereksiz motor kullanımı / verimsiz manevraları** azaltmaya yönelik maliyetler

içerir. Böylece ajan, hem güvenli iniş davranışını hem de daha verimli kontrolü öğrenmeye yönlendirilir.

3. KULLANILAN ALGORİTMA

Algoritma Adı

- **DQN (Deep Q-Network)**
- **Double DQN (DDQN)**
- **Dueling DQN**

Temel Çalışma Mantığı (Ortak)

- Üç algoritmada da ajan, her durumda **4 ayrık eylemden** birini seçer.
- Eylem seçimi, sinir ağının ürettiği **Q-değerlerine** göre yapılır.
- Ajan, ortamla etkileşime girerek elde ettiği *(durum, eylem, ödül, sonraki durum, episode durumu)* bilgilerini **Replay Buffer** içinde saklar.
- Eğitim sırasında bu hafızadan **rastgele mini-batch** örneklenerek ağ güncellenir.
- Hedef ağ (target network) kullanılarak öğrenme süreci daha kararlı hale getirilir.

DQN (Standart DQN)

- DQN algoritmasında amaç, her eylem için doğru Q-değerlerini tahmin eden bir sinir ağı öğrenmektir.
- DQN için tam bağlantılı (fully-connected) bir ağ yapısı kullanılmıştır.
- Ağ, iki gizli katman ve bir çıkış katmanından oluşmaktadır.
- Aktivasyon fonksiyonu olarak **ReLU** kullanılmıştır.
- Hedef ağ, eğitim sürecinde **soft update** yöntemiyle kademeli olarak güncellenmiştir.

Double DQN (DDQN)

- Double DQN, DQN algoritmasındaki **Q-değerlerinin aşırı tahmin edilmesi** problemini azaltmak amacıyla kullanılmıştır.
- Bu yöntemde:
  - Bir sonraki adımda en iyi eylem **local ağ** tarafından seçilir.
  - Seçilen eylemin değeri ise **target ağ** tarafından değerlendirilir.
- Böylece eylem seçimi ve değerlendirme ayrılmış olur ve daha dengeli bir öğrenme süreci sağlanır.

Dueling DQN

- Dueling DQN algoritmasında sinir ağı, ortak katmanlardan sonra iki ayrı kola ayrılır.
- Bu kollardan biri **durumun genel değerini**, diğeri ise **eylemlerin göreli avantajını** öğrenir.
- Daha sonra bu iki bilgi birleştirilerek her eylem için Q-değeri elde edilir.
- Bu yapı, özellikle eylemler arasındaki farkın küçük olduğu durumlarda daha kararlı ve hızlı öğrenmeye katkı sağlar.

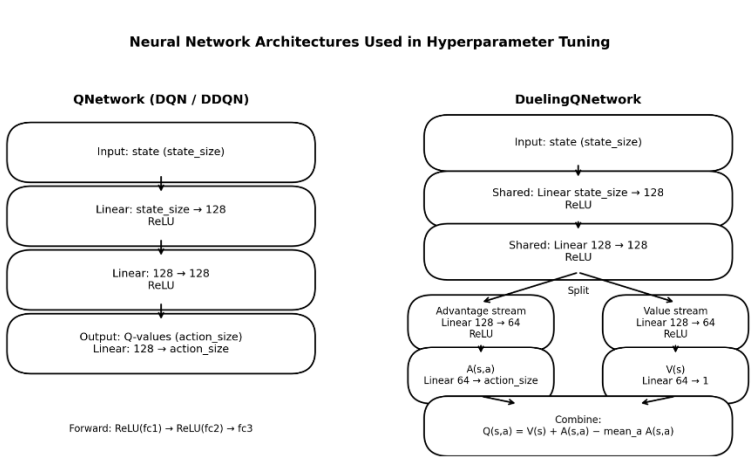
Temel Parametreler

- **Replay Buffer kapasitesi:** 100,000
- **Batch size:** 128
- **İndirim faktörü (gamma):** 0.99
- **Öğrenme oranı (LR):** 0.0001
- **Soft update katsayısı (tau):** 0.005
- **Güncelleme sıklığı:** Her 4 adımda bir öğrenme
- **Optimizer:** AdamW
- **Kayıp fonksiyonu:** Huber Loss (SmoothL1Loss)
- **Eğitim stabilitesi için:** Gradient clipping (maksimum norm = 1.0)

Keşif Yöntemi (Exploration)

- **epsilon-greedy** keşif stratejisi kullanılmıştır.
- Eğitim başlangıcında epsilon değeri yüksek tutulmuş ve ajan daha fazla keşif yapmıştır.
- Episode sayısı arttıkça epsilon değeri azaltılarak ajan, öğrendiği en iyi eylemleri seçmeye yönlendirilmiştir.

Sinir Ağı Mimarisi ve Hiperparametre Güncellemesi



Şekil 0. Hiperparametre ayarlaması sırasında kullanılan sinir ağı mimarileri: QNetwork (DQN/DDQN) ve DuelingQNetwork (Dueling DQN).

Mimariye bakınca İki farklı ağ yapısı görülmektedir. Sol tarafta yer alan **QNetwork**, DQN ve Double DQN algoritmalarında kullanılan standart yapıdır. Girişte ortamın durum vektörü alınır (state_size) ve ağın çıktısında her bir eylem için bir Q-değeri üretilir (action_size). Bu yapı iki adet tam bağlantılı gizli katmandan oluşur ve her gizli katman sonrası **ReLU** aktivasyonu uygulanır. Bu projede hiperparametre güncellemesi sonrası gizli katman genişliği **128 nöron** olacak şekilde ayarlanmıştır (state_size → 128 → 128 → action_size). Bu değişiklik, modelin durum–eylem ilişkilerini daha yüksek kapasiteyle temsil etmesini ve daha iyi bir değer tahmini öğrenmesini amaçlar.

Şeklin sağ tarafında yer alan **DuelingQNetwork** ise Dueling DQN algoritması için kullanılan dallanmalı mimaridir. Bu yapıda önce ortak iki katman ile özellik çıkarımı yapılır (state_size → 128 → 128). Daha sonra ağ iki kola ayrılır:

- **Value stream (Değer kolu):** Durumun genel olarak ne kadar iyi/kötü olduğunu öğrenir (çıktı tek bir değer üretir).
- **Advantage stream (Avantaj kolu):** Aynı durumda eylemler arasındaki göreli farkları öğrenir (çıktı eylem sayısı kadar değer üretir).

Bu iki kolun ayrılması, bazı durumlarda eylemler birbirine çok benzerken bile "durumun genel kalitesinin" daha doğru öğrenilmesini hedefler. Şekilde ayrıca her iki kolda ara katman genişliğinin **64** olarak kullanıldığı görülmektedir (128 → 64). Son aşamada bu iki bilgi birleştirilerek her eylem için Q-değeri elde edilir. Bu birleşim sırasında, avantaj değerleri eylemler arasında dengelenerek (ortalama etkisi giderilerek) daha stabil bir Q-değeri üretimi amaçlanır.

Mimari güncellemesiyle birlikte, eğitim kararlılığını ve yakınsamayı iyileştirmek için bazı hiperparametreler de değiştirilmiştir:

- **Network Width:** 64 → 128 (daha yüksek temsil kapasitesi)

- **Batch Size:** 128 → 64 (daha sık güncelleme, daha dinamik öğrenme)
- **Learning Rate:** 1e-4 → 5e-4 (daha hızlı yakınsama hedefi)
- **TAU (Soft Update):** 0.005 → 0.001 (hedef ağın daha yavaş güncellenmesiyle daha stabil öğrenme)

Bu değişikliklerin beklenen etkisi, özellikle hedef ağ güncellemesinin yavaşlatılması (tau düşürülmesi) sayesinde öğrenme sürecindeki dalgalanmaların azalması ve ortalama ödül eğrisinin daha düzenli hale gelmesidir.

4. EĞİTİM SÜRECİ

Tüm algoritmalar LunarLander-v3 ortamında aynı eğitim yapısı kullanılarak eğitilmiştir. Ajan, ortamdan elde ettiği deneyimleri Replay Buffer içinde saklamış ve bu deneyimlerden rastgele örneklenen mini-batch'ler ile öğrenme gerçekleştirmiştir.

DQN ve Double DQN algoritmaları 500 episode boyunca, Dueling DQN algoritması ise 600 episode boyunca eğitilmiştir. Ağ güncellemeleri belirli adım aralıklarında yapılmış, hedef ağlar soft update yöntemiyle güncellenmiştir.

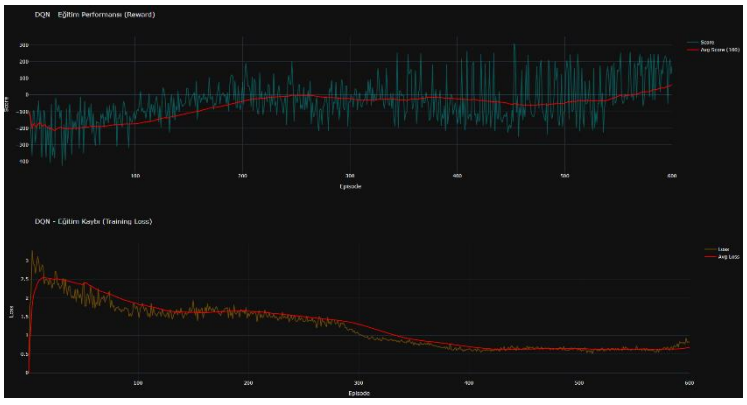
Performans takibi için 100 episode'luk hareketli ortalama ödül değeri kullanılmıştır. Ayrıca eğitim sürecinde kayıp (loss) değerleri izlenerek öğrenmenin kararlılığı değerlendirilmiştir.

5. SONUÇLAR

DQN, Double DQN ve Dueling DQN algoritmalarının eğitim sürecinde elde edilen performansları karşılaştırılmıştır. Karşılaştırma, episode başına elde edilen toplam ödül (reward) ve eğitim kaybı (loss) değerleri üzerinden yapılmıştır.

DQN Sonuçları

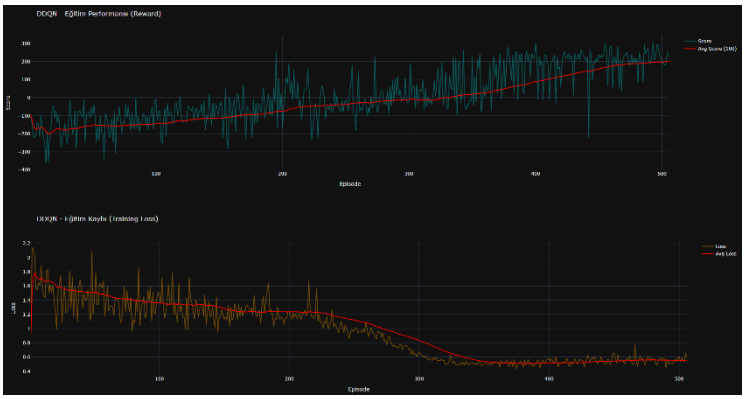
Şekil 1'de Ödül eğrisi, eğitim ilerledikçe artış eğilimi göstermiştir ancak dalgalanmalar yüksektir. Kayıp (loss) değerleri zamanla azalmış olsa da eğitim süreci boyunca belirli bir seviyede dalgalanma devam etmiştir.



Şekil 1. DQN algoritması için episode bazlı toplam ödül ve eğitim kaybı grafikleri.

Double DQN Sonuçları

Şekil 2'de Double DQN algoritmasına ait sonuçlar verilmiştir. DQN'e kıyasla ödül eğrisinin daha düzenli bir artış gösterdiği ve dalgalanmaların görece azaldığı gözlemlenmiştir. Eğitim kaybı daha kararlı bir şekilde düşmüş ve daha stabil bir öğrenme süreci elde edilmiştir.

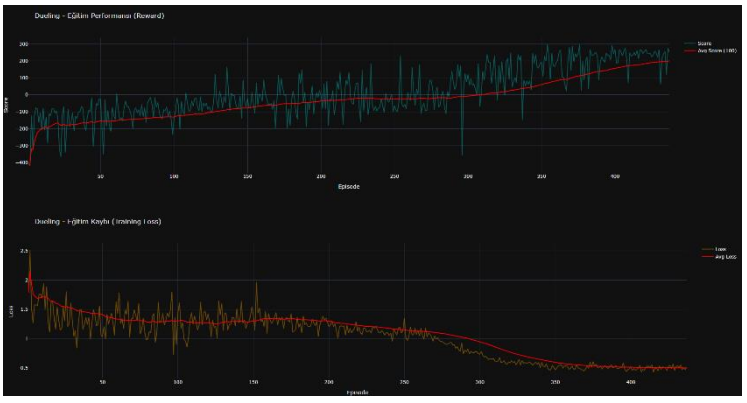


Şekil 2. Double DQN algoritması için episode bazlı toplam ödül ve eğitim kaybı grafikleri.

Double DQN algoritması eğitim süreci boyunca düzenli bir öğrenme davranışı sergilemiştir. Ortalama ödül değerleri eğitim ilerledikçe sürekli artmış, yaklaşık 300. episode sonrasında pozitif değerlere ulaşmıştır. 500. episode civarında ortalama ödül 200 seviyesini aşmış ve algoritma 506. episode'da ortamı çözmüştür. Eğitim kaybı değerleri ise zamanla azalarak daha kararlı bir seviyede seyretmiştir.

Dueling DQN Sonuçları

Şekil 3'te Dueling DQN algoritmasının eğitim sonuçları gösterilmektedir. Ödül eğrisi, eğitim süreci boyunca istikrarlı bir artış sergilemiş ve daha yüksek ortalama ödül değerlerine ulaşmıştır. Kayıp değerleri ise eğitim ilerledikçe azalmış ve kararlı bir seviyede seyretmiştir.



Şekil 3. Dueling DQN algoritması için episode bazlı toplam ödül ve eğitim kaybı grafikleri.

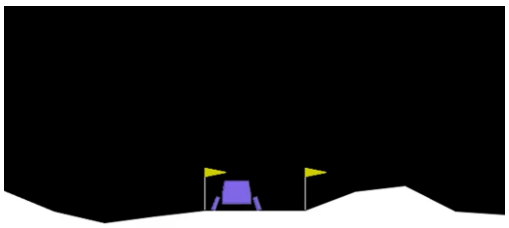
Dueling DQN algoritması eğitim süreci boyunca istikrarlı bir öğrenme davranışı göstermiştir. Ortalama ödül değerleri yaklaşık 350. episode sonrasında pozitif bölgeye geçmiş, 400. episode civarında hızlı bir artış sergilemiştir. Algoritma 516.

episode'da ortalama ödül değeri 200 eşiğini aşmış ve eğitim 600 episode'a kadar sürdürülmüştür. Eğitim sonunda ulaşılan ortalama ödül değeri **223.93** olup, öğrenme sürecinin kararlı bir şekilde tamamlandığı gözlemlenmiştir.

Dueling DQN için standart **erken durdurma yaklaşımı yerine**, sabit 600 episode'luk bir eğitim stratejisi uygulanmıştır. Eğitim sürecinde iki ayrı model kaydı alınmıştır: biri ortalama ödül 200 eşiğinin ilk kez aşıldığı anda, diğeri ise tüm eğitim tamamlandıktan sonra. Bu yaklaşım sayesinde yalnızca skor değil, aynı zamanda kayıp (**loss**) değerleri ve öğrenme eğrileri birlikte analiz edilerek daha kapsamlı bir performans değerlendirmesi yapılmıştır.

Nihai Performans Değerlendirmesi

Elde edilen sonuçlar, Double DQN ve Dueling DQN algoritmalarının standart DQN'e kıyasla daha kararlı bir öğrenme süreci sunduğunu göstermektedir. Özellikle Dueling DQN algoritması, ödül ve kayıp eğrileri açısından daha dengeli bir performans sergilemiştir.



Şekil 4. Başarılı bir inişin gösterimi.

KAYNAKÇA

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). *Human-level control through deep reinforcement learning*. Nature, 518(7540), 529–533.

[2] Van Hasselt, H., Guez, A., & Silver, D. (2016). *Deep reinforcement learning with double Q-learning*. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).

[3] Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2016). *Dueling network architectures for deep reinforcement learning*. Proceedings of the 33rd International Conference on Machine Learning (ICML).

[4] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[5] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *OpenAI Gym*. arXiv preprint arXiv:1606.01540.

[6] Farama Foundation. (2023). *Gymnasium: A standard API for reinforcement learning environments*. https://gymnasium.farama.org

[7] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). *PyTorch: An imperative style, high-performance deep learning library*. Advances in Neural Information Processing Systems (NeurIPS).