Journal of Turkish

Operations Management

Pekiştirmeli öğrenme ile tenis oyunu simülasyonu gerçekleştirimi

Bakhtiyar Ospanov[1], M. Fatih Demirci[2*]

[1] Department of Computer Science, Nazarbayev University, Astana, Kazakhstan
e-mail: bakhtiyar.ospanov@nu.edu.kz, ORCID No: : https://orcid.org/0000-0003-4490-4709

[2] Bilgisayar Mühendisliği Bölümü, Ankara Yıldırım Beyazıt Üniversitesi, Ankara, Türkiye
e-mail: mfdemirci@aybu.edu.tr, ORCID No: https://orcid.org/0000-0003-1552-0087
*Sorumlu Yazar

Makale Bilgisi

Makale Geçmişi:
Geliş: 30.03.2023
Revize: 17.05.2023
Kabul: 27.06.2023

Anahtar Kelimeler:

Pekiştirmeli Öğrenme,
Tenis Oyunu Simülasyonu,
Makine Öğrenmesi,
Sinir Ağları

Özet

Oyun ve robotik endüstrilerinde, akıllı ve etkileşimli karakterlerin tasarımı, yapay zekadaki ilerlemelerle büyük ölçüde zenginleştirilmiştir. Yapay-zeka tabanlı bu yaklaşımlara, özellikle, geleneksel algoritmaların önceden programlanmış kural-tabanlı olması nedeniyle ihtiyaç duyulmaktadır. Makine öğrenmesi ile oyun karakterleri karmaşık oyunlarda bile özgün ve bağımsız davranışlara sahip olacak şekilde eğitilmektedir. Bu çalışma, henüz yeterince üzerinde çalışılmamış olan tenis oyununda pekiştirmeli öğrenme kullanarak zeki oyuncuların (ajanların) başarılı bir şekilde eğitilebileceğini göstermektedir. Eğitim aşamasında ajanlara temel tenis kuralları ve sonuç (kazanma/kaybetme) hakkında bilgi verilmektedir. Ayrıca, oyun içindeki performanslarına göre ajanlar ödül ya da ceza da almaktadır. Buna göre, ajanlar kendi bulundukları duruma göre en iyi davranışı bulmaya çalışır. Ajanlar, Unity'de uygulanan görsel, fiziksel ve bilişsel olarak zengin bir çevrede eğitilmektedir. Sunulan çalışmanın deneysel değerlendirmesi, genel olarak modelin etkililiğini ve başarısını göstermektedir. Gerçekleştirilen uygulama açık-kaynaktır ve uygulamaya şu adresten erişilebilir: https://bakhtiyar-ospanov.github.io/MLAT/index.html

Implementation of tennis game simulation with reinforcement learning

Article Info

Article History:
Received: 30.03.2023
Revised: 17.05.2023
Accepted: 27.06.2023

Keywords:

Reinforcement Learning,
Tennis Game Simulation,
Machine Learning,
Neural Networks

Abstract

In the gaming and robotics industries, advances in artificial intelligence have significantly improved the design of intelligent and interactive characters. These AI-based approaches are particularly essential due to the rule-based nature of traditional algorithms. Through machine learning, game characters can be trained to exhibit unique and independent behaviors, even in complex games. This study illustrates that intelligent players (agents) can be effectively trained using reinforcement learning in the game of tennis, an area that hasn't received sufficient research attention. Agents are provided with a basic understanding of tennis rules and the game's outcome (win/lose). Additionally, agents are rewarded or penalized based on their in-game performance. Consequently, agents strive to determine the best behavior depending on their current circumstances. These agents are trained in an environment that is visually, physically, and cognitively rich, implemented in Unity. The empirical evaluation of this study demonstrates the overall effectiveness and success of the model. The implemented application is open-source and can be accessed at the following address: https://bakhtiyar-ospanov.github.io/MLAT/index.html

1. Giriş ve İlgili Çalışmalar

Makine Öğrenmesi (MÖ) algoritmaları, son yıllarda farklı sektörlerde başarı ile kullanılmıştır. Gerçek dünyanın simülasyonu, MÖ algoritmalarından faydalanabilecek alanlardan biridir. Simülasyon ortamı tasarımının genel zorlukları bilinmekle birlikte, gerçekçi olarak nesneler arasında mantıksal etkileşimler ve anlamlı davranışlara dayanarak bir simülasyon elde etmenin daha da zor olduğu araştırmacılar tarafından kabul edilmektedir. Simülasyonlar, bağlamla ilgili sabit olarak belirlenmiş kurallardan ziyade deneyime dayalı olarak MÖ ile yapıldığında daha gerçekçi olmaktadır. Örnek olarak, robotik araştırmalarında Rubik küpünü çözmek için robotik bir elin eğitilmesi (Akkaya ve vd. (2019)) ve eğlence endüstrisinde Doom oyununda ortalama bir insandan daha iyi performans gösterilmesi (Lample ve vd. (2017)) bu şekilde bir simülasyon yapılması sonrasında elde edilmiştir. Akıllı, özgün ve bağlama duyarlı bir rakibe karşı oynamak, oyunları daha eğlenceli ve ilgi çekici hale getirecektir. Sunulan çalışmanın amacı, bir tenis oyununda kendi kendine oynayan zeki ajanların MÖ algoritmaları ile etkin bir şekilde eğitilmesidir.

Gerçekleştirilen uygulamada, fiziksel dünyanın etkin bir simülasyonunu sağlayan ve uygun MÖ araçları sunan Unity Gerçek Zamanlı Geliştirme Platformu kullanılmıştır. Ek olarak, ajanlar, pekiştirmeli öğrenme (PÖ) kapsamında eğitilmiş olup çalışma Python API ile programlanmıştır. Oyunda, aynı çalışma mantığını kullanan iki PÖ-Ajanı, tenis kurallarına göre filenin karşı taraflarında topa vurmak için raketleri kontrol etmektedir. Oyuncular, topa rakip oyuncunun karşılayamayacağı şekilde vurma amacını taşır. Böylece modelin gerçekçi bir ortamda eğitilmesi hedeflenmiştir. Uygun fiziksel özelliklere sahip bir tenis oyununun simüle edilmesi, geniş bir alan içindeki karmaşık fiziksel etkileşimler nedeniyle titiz kurulum ve ayarlamalar gerektirmektetir. Bunun sonucu olarak oyun endüstrisine bu şekilde bir yaklaşımın tanıtılması sağlanmaktadır. Sunulan kapsamlı simülasyon ortamı, fiziksel ve görsel olarak zengin bir içeriğe sahiptir ve masa tenisi, badminton, pickleball vb. gibi benzer spor oyunları için de kolayca uygulanabilecek bir yapıdadır. Aynı zamanda, eğitim düzeni, oyunlardaki gözlem karmaşıklığını, fiziksel parametrelerin etkisini, mantıksal etkileşimleri (strateji) ve sosyal etkileşimleri (arkadaşça/saldırgan oyun tarzı) araştırmak için de kullanılabilir. Ortaya çıkan model, bir oyun uygulamasında kullanılmasının yanı sıra, bu alanda yapılacak gelecekteki araştırmalar için temel bir performans modelidir.

MÖ, genel olarak farklı türlerdeki verilerden bilgi çıkarmak için kullanılan bir araçtır. Ayrıca, bir makinenin açıkça programlanmadan verilen verilere dayalı bir sonucu tahmin edebildiği Yapay Zekanın (YZ) bir alt alanı olarak da tanımlanır (Alzubi ve vd. (2018)). Genel olarak, MÖ yaklaşımları üç ana tipte sınıflandırılmaktadır: denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme (Ayodele (2010)). Birinci tip, çok sayıda doğru etiketlenmiş veri gerektirir ve sıklıkla sınıflandırma ve regresyon için kullanılır. Denetimsiz olan MÖ algoritmaları, kümeleme ve ilişkilendirmeyi etiket kullanmadan ve önceden kategorize etmeden yapar. Sonuncusu (PÖ) ise, bir ortamı keşfeden ve onunla etkileşime giren deneme yanılma yaklaşımını benimseyen ajanları kullanır.

PÖ'de, bir ajan, doğru bir performans durumunda verilen bir ödülü en üst düzeye çıkarmaya ve uygunsuz davranış için bir cezayı en aza indirmeye çalışır. Bu şekilde ajanlar, ödül sinyaline uyum sağlayarak, bir ortam ve kuralları hakkında bilgisini geliştirir ve buna bağlı olarak bir sonraki en iyi eylemlerini tahmin edebilir. Simülasyon dünyasının karmaşıklığı göz önüne alındığında, PÖ dinamik davranışa sahip bir ajan oluşturmak için en uygun MÖ sınıfı olarak karşımıza çıkmaktadır. Satranç, shogi ve Go oynayan PÖ ajanları (Silver vd. (2018)), video oyunları (Torrado ve vd. (2018)), 3B çok oyunculu oyunlar (Jaderberg ve vd. (2019)), dağıtım şebekesi (Yurtsever ve vd. (2022)) ve oyun araştırmaları dahil olmak üzere YZ'nın ve MÖ'nün oyun endüstrisini nasıl değiştirdiğine dair bilgilendirici örnekler vardır (Lanctot ve vd. (2019), Millington ve Funge (2009)) .

Genel olarak literatüre baktığımızda, bilgilendirici ve gerçekçi etkileşim sağlayan bir ortam elde edilmesi için PÖ'nün başarılı performans sağlayacak önemli bir potansiyele sahip olduğu görülmektedir. PÖ, çevre tasarımı, gerçek dünya kavramlarının simüle edilmesi ve araştırmacıların hipotezlerinin test edilmesi için sezgisel ve son derece özelleştirilebilir bir ortam sunar. Ayrıca, literatürde tenis oyununun simüle edilmesi için gerekli çalışmaların eksikliği de görülmektedir. Bu nedenle sunulan çalışmada henüz yeterince araştırılmamış olan tenis oyununda kendi kendine oynayarak pekiştirmeli öğrenen ajanların eğitilmesi yapılmış ve bu şekilde literatürdeki bir açığın kapatılması için önemli bir adım atılmıştır.

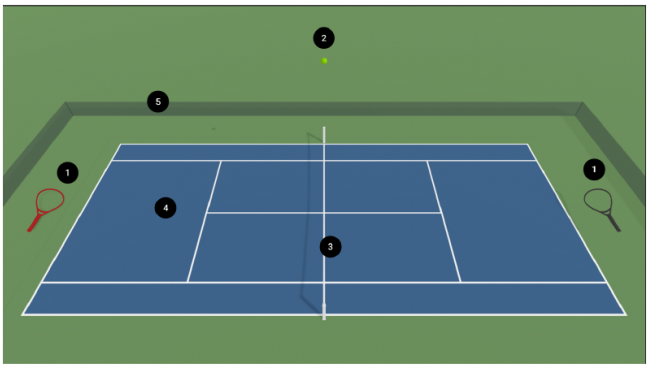
Tenis oyununa benzer kurallara sahip olan masa tenisinde literatürde PÖ-temelli çalışmalar sunulmuştur (Wang ve vd. (2022)) (Muelling ve vd. (2014)) (Tebbe ve vd. (2021)) (Gao ve vd. (2020)). Tenis ve masa tenisinde bir oyuncu, topa her seferinde aynı noktadan vurarak oyunu kazanamaz. Bu nedenle oyuncular, rakiplerini yenmek için etkili bir strateji geliştirmek zorundadır. Bu kapsamda PÖ ile ödül işlevini öğrenen çalışmaların geçmişte kullanıldığını görüyoruz. Muelling ve vd. (2014) ödül fonksiyonunun Markov karar problemine dayandığı, bir PÖ çıkarım modeli vermektedir. Başarılı sonuçlar alınmasına karşın mevcut modelde, rakibin zayıflıklarını, stratejisini, topun dönüşünü ve sensör bilgisinin kusurlu olabilme olasılığını hesaba katılmamıştır. Tebbe ve vd.

(2021) robotikte pekiştirmeli öğrenme uygulamalarının iki temel problemini (hız ve sağlamlık) ele alan masa tenisi oynayan bir robot tasarlamıştır. Bu çalışma bir önceki çalışmanın eksikliklerine benzer olarak topun hız ve dönüş parametrelerini dikkat almamaktadır.

Masa tenisinin PÖ yaklaşımı ile öğrenen modellerin geçmiş çalışmalarda daha ilgi görmesine karşın tenis oyunu için aynı şeyi söylemek söz konusu değildir. Buna karşın geçmiş çalışmalarda özel olarak tenis oyununu hedeflemese de uygulama olarak diğer oyunların yanında tenis oyununu kullanan çalışmaların bulunduğu görülmektedir, örnek (Kancharla ve Lee (2019)) (Cao ve vd. (2020)). Buna ek olarak, (Weeraman (2023)) önerilen makalede bulunan kapsamda herhangi bir tartışma ve değerlendirme bulunmayan bir uygulama linki paylaşmıştır. Bu uygulamanın sadece bir programlama hedefi ile yazılmış olmasından dolayı, sunulan çalışma ile bu çalışma karşılaştırılmamıştır.

Bu makalede sunulan modelin, teknik detayları verilerek elde edilen sonuçlar tartışılmıştır. Ek olarak, PÖ ajanlarının birbirlerine karşı oynadıkları oyunların değerlendirilmesi 45 farklı katılımcı ile sağlamıştır. Elde edilen sonuçlar, oyunlarda ajan davranışlarının, uzman bir kullanıcı davranışlarını başarılı bir şekilde simüle ettiğini göstermektedir. Makalenin geri kalan kısmında sunulan model ayrıntılı olarak açıklanacak ve kullanılan hiper parametreler belirtilecektir. Deneyler ve sonuçlar bölümünde, çeşitli konfigürasyonlara sahip ajanların eğitim ve karşılaştırmalı analizi sırasında istatistiksel gözlemler anlatılacaktır. Elde edilen model, MÖ ajanının performansını ölçmek ve kullanıcı deneyimi ile karşılaştırmak için bir test uygulamasında kullanılmıştır. Son olarak, bu makalede çalışmanın sınırlamaları ve sunulan modellerin geliştirilmesine yönelik çeşitli fikirler tartışılmıştır.

Bu çalışma, 2021 tarihinde birinci yazar tarafından ikinci yazar danışmanlığında Nazarbayev Üniversitesi Bilgisayar Bilimleri Anabilim Dalında Yapılan Yüksek Lisans tezinden üretilmiştir. Çalışmanın her aşamasında Araştırma ve Yayın Etiğine uyulmuştur.



Şekil 1. Tenis kortu görünümü. 1 - ajanlar, 2 - top, 3 - ağ bandı (file), 4 - kort alanı, 5 - sınırlar.

2. Metodoloji

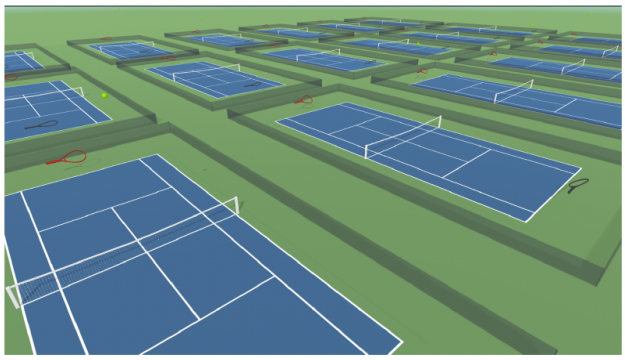
Sunulan makalenin araştırma hedefine ulaşmak ve PÖ tabanlı zeki bir tenis oyuncusu (ajanı) yetiştirmek için zengin karmaşıklığa sahip bir ortamın nasıl oluşturulduğu bu bölümde anlatılacaktır. Bunu takiben, eğitim ortamı ile birlikte mantıksal davranışlar ve etkileşimler, ve yapılan eğitimin test uygulamasındaki pratik kullanımı da bu bölümde anlatılmaktadır.

2.1. Ortam Tasarımı

Ortam tasarımı kapsamında ilk olarak uygulamada kullanılan fiziksel değerler ve öğrenme değerlerinin nasıl seçildiğinden bahsedilecektir. Daha sonra eğitim konfigürasyonu konusunda bilgi verilecektir. Son olarak oluşturulan test uygulaması açıklanacaktır.

2.1.1 Fiziksel Değerler

Bir ortamın, başarılı bir şekilde ajanları eğitmesi için hem fiziksel hem de görsel özellikleri açısından gerçek dünyaya yüksek düzeyde benzerlik göstermesi gerekir. Daha önce bahsedildiği gibi Unity, bu açıdan farklı platformlarda çalışan ideal oyun motorlarından biridir. Yapılan çalışmada ilk olarak, tenis alanının genel yapısı Unity'de oluşturularak top, raket, ağ bandı (file) ve sahne tasarlanmıştır (Şekil 1). Böylece, ajanların ve bunların bağımlı bileşenlerinin konumlandırılması ve izlenmesi elde edilmiştir. Bu tasarım ile bu alanların 18 bağımsız kopyası oluşturulmuş ve eğitim süreci bu kopyalar ile birlikte başlatılmıştır (Şekil 2). Burada birden fazla kopya kullanılmasının amacı aşağıdaki bölümde anlatılacak (Bölüm 2.1.2) öğrenme değerlerinin daha hızlı bir şekilde elde edilmek istenmesidir.

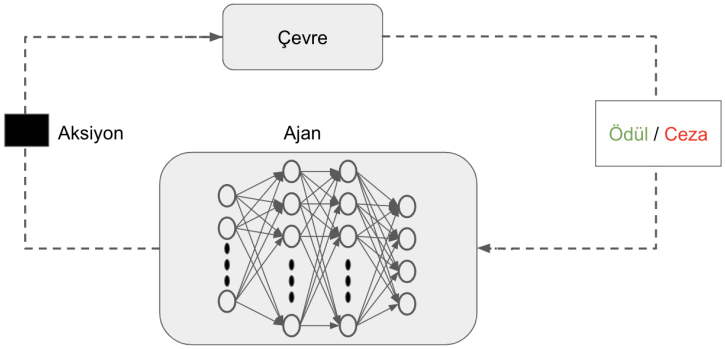


Şekil 2. Eğitim kurulumunda tenis için 18 tane ayrı örnek çalıştırılmıştır.

Hazırlanan tenis kortların her birinde, grafiksel olarak raketler ile temsil edilen iki ajan vardır (bu makalede tenis oyuncuları, ajanlar ve raketler birbirinin yerine kullanılmaktadır). Çarpıştırıcı bileşeni, top, ağ bandı, zemin ve oyun sınırları arasındaki çarpışmaları izlemek için oyuna eklenmiştir. Raket, top, ağ, zemin ve oyun sınırını belirlemede kullanılan duvarlar için şu fiziksel materyaller kullanılmıştır. Raketler için kullanılan fiziksel materyal 0.45 birim dinamik sürtünmeye, 0.5 birim statik sürtünmeye, 0.2 birim zıplama kapasitesine ve 350 gram ağırlığa sahiptir. Bu değerler ortalama bir raketin gerçek dünyadaki değerleri baz alınarak deneysel olarak belirlenmiştir. Dolayısıyla, bu parametreler ile raket ile topun çarpışması gerçekçi bir şekilde elde edilmiştir. Açısal sürükleme, raket dönerken karşılaşılan hava direncinin etkisini temsil eder (sıfır, direnç olmadığı anlamına gelir) ve 0,05'e ayarlanmıştır. Hızlı hareket eden etkileşimler nedeniyle hassas hesaplamalar için yoğun hesaplamalı sürekli çarpışma algılama algoritmaları kullanılmıştır. Bu algoritmalar ile her üç boyutta da hız kontrol ederek bir ajanın gerçekçi hareketi simüle edilmiştir. Hız ve atalet kısıtlamaları açısından ve eğitim konfigürasyonunun özgüllüğü nedeniyle x ve y eksenleri dönüş için sınırlandırılmıştır. Raket ölçüleri gerçek raketlerin ölçüleri göz önüne alınarak 1.37m uzunluk, 0.57m kafa genişliği, 0.021m kalınlığa sahip olacak şekilde belirlenmiştir.

Raketlere ek olarak, tenis oyununda topun gerçekçi olarak simüle edilmesi de oldukça önemlidir. Top için kullanılan materyal, 0.2 birim dinamik sürtünmeye, 0.1 birim statik sürtünmeye, 0.8 birim zıplama kapasitesine ve 60 gram ağırlığa sahiptir. Topun yarıçapı, ajanın arama uzayının azaltılması için gerçek boyutundan daha yüksek yapılarak 0,2m seçilmiştir. Bölüm 2.2'de bu konu üzerinde daha detaylı açıklama yapılacaktır.

Zemin, tenis oyununda oyuncuların alacakları puanları belirleyen farklı bölgelerden oluşur ve bu bölgeler elde edilen ortamda belirlenmiştir. Zemin için 0,4 birim dinamik sürtünme, 0,4 birim statik sürtünme ve 0,1 birim zıplama kapasitesi tanımlanmıştır. Ayrıca, zemin için ortalama sürtünme ve zıplama kombinasyon modları tanımlanmıştır. Kort ölçüleri 23mx8m olarak kullanılmıştır. Kortun ortasındaki ağ bandı, bir sahayı iki özdeş parçaya ayırır ve topun bölge değişimini kontrol etmek için üstte bir çarpıştırıcı içerir. Son olarak, görünmeyen duvarlar, raketlerin hareketini makul konum ötesinde kısıtlamak için tüm oyun alanı etrafında tanımlanmıştır. Genel olarak, proje ayarları, yerçekimi için 9.834 $m/s2$, oyun sırasındaki zaman ölçeği 1s, eğitim süreci için ise 10s olarak belirlenmiş olup fiziksel güncellemeler için sabit delta süresi 0.02s olarak belirlenmiştir.



Şekil 3. Sunulan pekiştirmeli öğrenme modelinin genel sunumu. Ajan, ileri beslemeli yapay sinir ağı ile çıktı aksiyon üretir. Bu aksiyon çevreye (tenis oyununa) uygulandıktan sonra oyun kurallarına göre belirli bir ödül yada ceza üretilir. Bu geri besleme tekrar ajana verilerek ajanın sonraki aksiyonu belirlenir.

2.1.2 Öğrenme Değerleri

Gözlem ve çıkarımların toplanması tek bir sınıf (Akademi sınıfı) tarafından yönetilmektedir. Hem eğitim hem de oyun sırasında tüm ajanların kararlarını kontrol eden bir ajan vardır. Akademi sınıfı ayrıca, gözlemlenen verileri aktarmak ve yapay sinir ağını eğitmek için Python API ile iletişim kurmaktan da sorumludur.

Simülasyon süreci, ajanların puan kazanmaya çalıştığı bölümleri içerir. Bir bölüm, sonlandırma koşullarından biri sağlanana kadar devam eder. Sonlandırma senaryolarında, ajanlardan biri bir puan kazanır veya oyun zaman aşımına uğrar (yani, oyunda maksimum adım sayısına ulaşılır). Her bölüm, ilgili metot çağrılarak ortam kurulumunun bazı kısımlarının yeniden başlatılması ile çalışmaya başlar. Burada, raketlerin pozisyonları, rotasyon değerleri ve hızları sıfırlanır. Ajanların iyi bir şekilde farklı koşullarda eğitilmesi için raketlerin ilk pozisyonu tenis kortunun yüzeyinde başlangıç bölgesi içinde rastgele seçilmektedir.

Simülasyon tasarımındaki bir sonraki adım, potansiyel karar üzerinde etkisi olabilecek verileri toplayarak çevreyi gözlemlemektir. Eğitim süreci, sinir ağının üzerinde çalışabilmesi için verilerin öznitelik vektör formatında gönderilmesini gerektirir. Gözlem vektörünün uzay boyutu da davranış parametrelerinde açıkça belirtilmektedir. Öznitelik seçimi, her potansiyel parametrenin öneminin analitik tahmini ile yapılmıştır.

Model, gözlemlenen bilgilere dayanarak, her ajan için bir sonraki eylemi üretir. Ajanlar sürekli olarak sahada hareket ettiğinden, belirli bir kurulum için 'Sürekli' eylem alanı türü gereklidir. Karar verme sürecine, pekiştirmeli öğrenmede amaçlandığı gibi yapılan eylemler için ödüller eşlik eder. Bu ödüller, yapay sinir ağı tarafından seçilen eylemlerin optimalliğini belirlemek için hem eğitim hem de simülasyon sırasında kullanılır.

2.2 Eğitim Konfigürasyonu

Oyunda kullanılan tüm statik parametreler yukarıda açıklanmıştır. Bu bölümde ise eğitim sırasında kullanılan dinamik değişkenler belirtilecektir. Bir gözlem vektörü 14 değer içerir: Ajanın x,y,z pozisyonu, ajanın x,y,z yönündeki hızı, ajanın rotasyonu, topun x,y,z pozisyonu, topun x,y,z yönündeki hızı ve topun anlık parametrelerine göre düşmesi öngörülen bölgenin numarası.

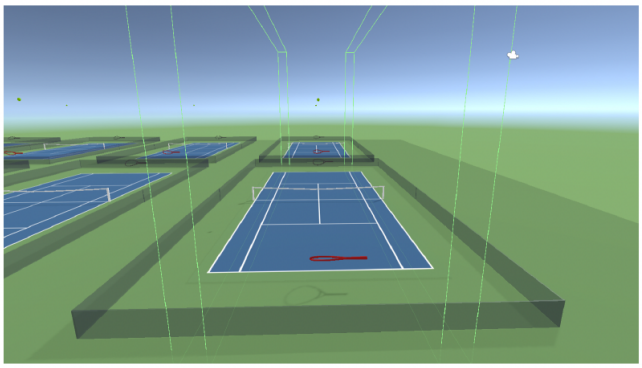
Kullanılacak yapay sinir ağı modelinde daha hızlı yakınsamasına yardımcı olduğu için tüm değerler [-1,1] aralığında normalleştirilmiştir. Karmaşık bir tekrarlayan yapay sinir ağı mimarisi sunmak yerine, mevcut gözlemleri önceki adımların verileriyle iyileştirmek için yığınlama kullanılmıştır. Bu genişletilmiş gözlem, verileri daha bağlamsal hale getirmek için küçük bir “hafızayı” taklit etmektedir.

Sağlanan girdi bilgilerini kullanarak model, 4 boyutlu bir çıktı eylem vektörü (aksiyon) üretir. Buradaki değerler [-1,1] aralığına normalize edilir ve bir raketin x,y,z yönündeki hızı ile rotasyonunu kontrol etmek için kullanılır. Ödül dağıtım sisteminde, oyun içinde bir puan kazanana 0,5 puan eklenir, kaybedenden ise 0.5 puan alınır. Bir ajanı makul davranışa daha hızlı yönlendirmek için, raket topa dokunduğunda küçük bir ödül (0,0005) verilir. Ayrıca, top fileyi (ağ bandını) aşıp rakip zemine çarptığında 0,001 puan ödül eklenir. Tenis kurallarının anlaşılmasını ve ard arda daha fazla galibiyeti teşvik etmek için, ajanlar bir oyunu kazandığında 0,5 puan ek ödül verilir, kaybettiğinde ise ajanlardan 0,5 puan ödül eksiltilir. Benzer olarak, teniste bir set kazanıldığında ajanlara 1,0 puan ödül eklenir, kaybedildiğinde ise ajanların toplam ödüllerinden 1,0 puan eksiltilir.

Bu konfigürasyon, eğitimin Unity tarafını tamamlarken, asıl eğitim ML-Agent Toolkit python paketi aracılığıyla harici olarak yürütülmektedir. Bu paket, proje için 27 hiper parametreli bir yapılandırma dosyası kullanır. Seçilen eğitim algoritmalarına dayalı olarak, hiperparametre ayarlama, ajanların bir hedefi gerçekleştirirken en uygun seçimi bulma becerisini büyük ölçüde etkiler.

Sunulan çalışmada kullanılan yapay sinir ağı ileri beslemeli olup toplam 256 gizli nöron ve normalizasyon uygulanmış iki gizli katman içermektedir. Eğitim için, ön deneylere dayalı olarak aşağıdaki hiperparametreler seçilmiştir: grup boyutu = 2048, tampon boyutu = 20480, öğrenme oranı = 0.0002 (sabit), beta = .003, epsilon = 0.15, lambda = 0.93, dönem sayısı = 4. Ödül sinyali için sadece dışsal ödül konfigürasyonu gama 0,96 ve dayanıklılık 1,0 ile tanımlanmıştır. Bu gama, bir ajanın gelecekte alabileceği ödüllerin önemini kontrol eder. Bu şekilde, ajanın içinde bulunduğu anda iyi performans göstermesi ve gelecekte de daha fazla ödül alması için daha fazla çabalaması sağlanmaktadır. Sunulan modelin genel görünümü Şekil 3'te verilmiştir.

Modelde, kendi kendine oynatma kurulumu için tanıtılan özel ayarlamalar vardır. Öğrenen ekip, 100.000 adım boyunca bir sanal eğitici ile oynar. Bu şekilde aynı rakiple uzun süre oynanarak ajanın kazanma şansı artmaktadır. Ayrıca, bu adım sayısı ile belirli bir stratejiye uyum sağlanıp bu strateji ezberlenmez. 100.000 adım sonrasında ajanlar yeni bir sanal eğitici ile eğitimlerine devam edebilir. Bir ajanın en son oynadığı rakibine karşı tekrar oynama olasılığı %50'dir. Her bir iterasyonda, ajanlar stratejilerini revize eder.



Şekil 4. Tenis kortu etrafına konulan nesnelerin hareketini sınırlayan sanal duvarlar.

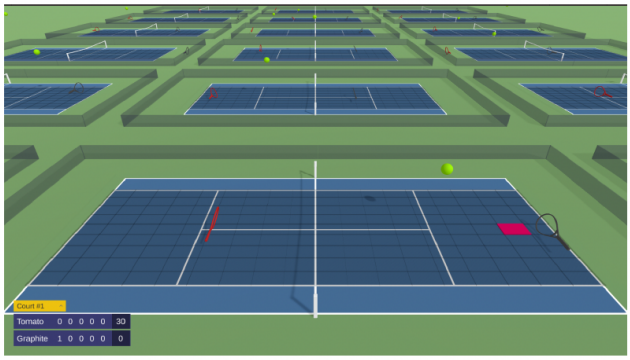
Uygun ajan davranışı, hedeflenen fiziksel parametrelerle doğrudan elde edilemez. Bir raketin ve bir topun gerçek hayattaki boyutları, tenis kortu alanına göre çok küçüktür. Bu, bir ajan için oldukça büyük bir alanda arama yapması anlamına gelir. Bu nedenle ödül sinyali çoğu zaman alınmaz. Daha açık olarak, küçük bir raketin yüzey alanına sahip bir oyuncunun, en azından büyük bir kortta herhangi bir yerde bulunabilecek bir topa dokunması beklenmektedir. Başlangıçta rastgele bir strateji verildiğinde, bunu başarmak oldukça zordur. Bunun yerine, görevin zorluğunu aşamalı olarak artırmak için kademeli öğrenme yöntemi uygulanmıştır. Bu eğitim, raketlerin %25, top ölçeğinin %150 büyütülmüş versiyonları ve z ekseni boyunca sabitlenmiş hız ile başlatılmıştır. Daha sonra her 200.000 adımda bu değerler orantısal olarak hedef değerlere ulaşana kadar düşürülmüştür. Böylece ajan, verilen ölçek parametreleriyle uygun davranış sergileyebilmiştir. 1 milyon adımdan sonra hız, z ekseninin her iki tarafındaki sınırlayıcı duvarlar olmak üzere tüm eksenlerde serbest bırakılmıştır (Şekil 4). Duvarlar arasındaki mesafe (yani ajanın hareket edebileceği bölge) kademeli öğreniminin bir parçası olarak her 200.000 adımda bir artmıştır. Bu bağlamda, rakip topa vurduğunda, ajan topun kendi tarafına nereye düşeceğini tahmin eder. Hesaplanan pozisyonun kesin konumunu belirtmek yerine, tenis kortunun her bir tarafı 9x9'luk bölgelere ayrılmıştır. Örnek bir tahmin, Şekil 5'de gösterilmektedir.

2.3. Test Uygulaması

Eğitilmiş model, Unity'de yerleşik bir tenis oyununda kullanılmıştır. Yapay Sinir ağı, hem CPU hem de GPU üzerinde çalışabilecek şekilde Unity Inference Engine tarafından işlenir. Oyun, WebGL platformu için inşa edilmiştir, ancak çok çeşitli ortamlarda da çalıştırılabilir. Gerçekleştirilen uygulamaya şu adresten erişilebilir: https://bakhtiyar-ospanov.github.io/MLAT/index.html

Oyun modunda bir ajanın z ekseni boyunca hızı, kullanıcı rahatlığı için sınırlandırılmıştır. Bir kullanıcının aynı anda üç eksen boyunca kendi konumuna göre topun konumunu uzamsal olarak hesaplaması zordur. Bu nedenle, kamera konumu, topun en az bir eksen konumunun ve topun fileye olan mesafesinin doğru bir şekilde öngörülmesi için ayarlanmıştır. Kullanıcıların, kontrol etmesi gereken yükseklik aralığı, kort ölçeğine göre alması zor olduğundan raketler z ekseni boyunca döndürebilir ve klavye girişi ile x ekseni boyunca hareket ettirebilir. Bir

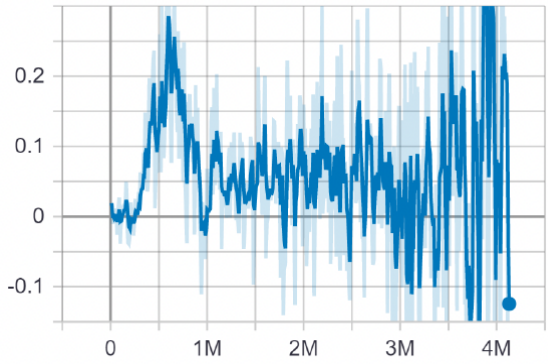
ajanın hareketini doğal yapmak için kullanıcı tarafından seçilen yöndeki hızı yavaş bir şekilde arttırılmıştır. Sınırlı eylem alanına ek olarak, oyunda top hareketinin yörünge çizgisi kullanıcıya ipucu olarak görüntülenir. Oyun, 5 setlik gerçek tenis oyununu yansıtacak şekilde tasarlanmıştır. Bir kullanıcı ayrıca iki MÖ ajanının oyununu gözlemleyebilir veya raketlerden birinin kontrolünü ele alabilir. Kullanıcılara oyuna başladıktan ve MÖ ajanına karşı birkaç oyun oynadıktan sonra, oyun hakkındaki görüşlerini almak için bir anket de verilmektedir.



Şekil 5. Topun nereye ineceğinin tahmin edildiği kırmızı ile işaretlenmiş bölge.

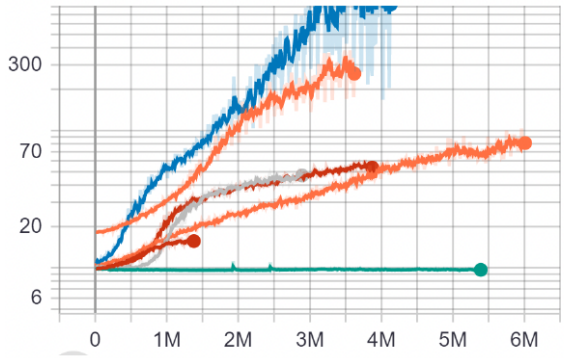
3. Deney ve Sonuçlar

Eğitim süreci ve deneyler sırasındaki istatistikler MLAgents Toolkit ve TensorBoard ile kaydedilmiştir. TensorBoard, özel metrikleri görselleştirmek ve gözlemlemek için uygun araçlar sağlamaktadır. Yapılan deneylerde, genellikle pekiştirmeli öğrenmede kullanılan ve değerinin 1'e doğru yaklaşması beklenen bir ölçüm olan kümülatif ödül kullanılmıştır. Şekil 6'da gösterildiği gibi, bir ödül sinyali başlangıçta sıfırın biraz üzerindedir ve eğitim ilerledikçe genlik artar.

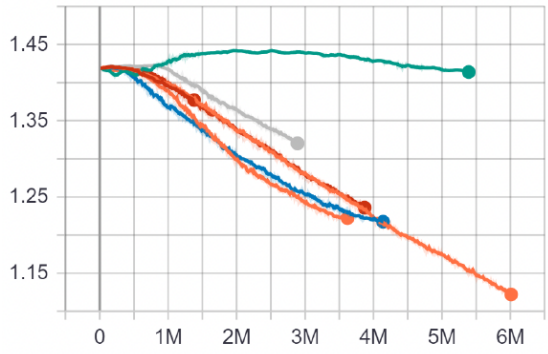


Şekil 6. Bir test sürecinde elde edilen örnek kümülatif ödül.

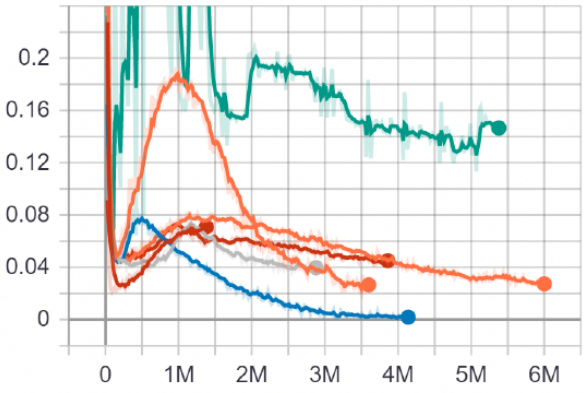
Oyunda kullanılan stratejiler geliştikçe, rakip havuzu daha güçlü stratejiler içerir ve bu da bir ajanın oyunlardaki kazanma ve kaybetme istatistiklerinin her ikisini arttırır. Bu nedenle, kümülatif bir ödül, böyle bir durum için anlamlı bir ölçüm olmamaktadır. Dolayısıyla, oyunların belirli bir bölümüne kadar (Şekil 7), ajanların topa vurma ve birbirlerine karşı oynama başarımları kullanılabilir. Bununla birlikte, daha uzun bir oyun uzunluğu, ajanların daha iyi bir performans göstermesi anlamına gelmez, aksine bu ajanların daha az kazanan atışlar yaptığı daha az agresif bir oyun sunar. Burada kullanılabilecek bir diğer önemli ölçüt, ajan kararlarının rastgeleliğini izleyen entropidir (Şekil 8). Bu kararlar zamanla azalmaktadır ki bu da oyun için uygun seçilmiş hiper parametreler anlamına gelir. Kayıp fonksiyonları ile ilgili olarak, model tarafından üretilen durumları ölçmek için değer kayıpları (Şekil 9) izlenmektedir. Grafiklerde bir ajan uzayı keşfedip öğrenirken bir artış, ardındanda alınan ödül sabitlendiğinden aşamalı bir şekilde düşüş görülür.



Şekil 7. Tenis oyunlarının farklı bölümleri içinde ajanların topa vurma ve birbirlerine karşı oynama başarımları.

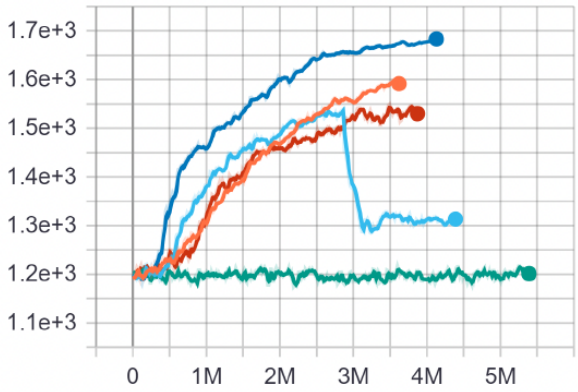


Şekil 8. Farklı konfigürasyonlar için ajanların oyunlardaki rastgeleliğini ölçen entropi değerleri. Bu değerler zamanla azalmaktadır.



Şekil 9. Farklı konfigürasyonlar için kayıp değerleri.

Eğitimin başarısını değerlendirmek için kullanılabilecek birincil ölçüt, kendi kendine oynanan oyunlara özgü olan ELO derecelendirme sistemidir (Albers ve de Vries (2001)). Bu yaklaşım, sıfır toplamlı bir oyunda (yani kazanan ve kaybeden oyuncuların puanlarının toplamının sıfır olduğu bir oyunda) iki oyuncunun birbirine göre seviyelerini göreceli olarak değerlendirir. Bu değerlendirme, mevcut puanlara ve maç sonucuna (galibiyet/mağlubiyet/beraberlik) göre hesaplanır. Güçlü oyuncu kazanırsa, zafer beklendiği için bu oyuncuya birkaç puan verilir ve bu yenilgi için zayıf oyuncudan birkaç puan düşülür. Buna karşılık, güçlü oyuncuya karşı zafer kazanması durumunda, zayıf oyuncuya yüksek puan verilir. Bir oyun berabere biterse, güçlü oyuncuya yüksek ceza, zayıf oyuncuya ise yüksek ödül verilir.



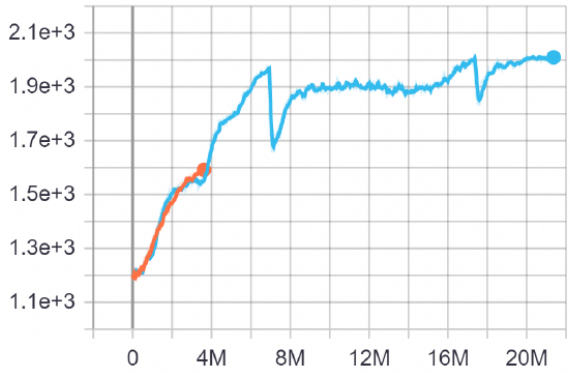
Şekil 10. Farklı konfigürasyonlar için ELO değerleri.

Başlangıçta ELO derecelendirme puanı 1200'dür. Şekil 10 çeşitli konfigürasyonlarla yapılan çalışma sonuçlarında elde edilen ELO değerlerini göstermektedir. Turuncu çizgi, ilk kurulumu temsil eder. Bu konfigürasyonda ajanın x yönündeki hızı, raketlerin ve topun büyütülmüş versiyonu ve bir oyunu kazanmak/kaybetmek için +1/-1 ile temel ödül sinyali bulunur. Bir ajan bu konfigürasyonda iyi performans gösterdiğinden dolayı, bu performans karşılaştırma için temel (baz) performans alınmıştır. Mavi çizgi, ödül sinyal performansını gösterir. Eğitim konfigürasyonu bölümünde açıklandığı gibi, topu karşılama, filenin karşı kısmına (rakip bölüme) başarılı bir şekilde top atma, tüm maçı kazanma vb. durumlar için ek ödüller verilmiştir. ELO derecelendirmesinin bu şekilde kullanılması sonrasında eğitim hızı açısından gözle görülür bir performans artışı kaydedilmiştir. Ödül sinyalinin alınması stabilize edildikten sonra, fiziksel özellikler daha önce bahsedildiği gibi gerçek dünya boyutlarıyla ilişkilendirilecek şekilde ayarlanmıştır. Bu ayarlama sonrasında, yani, raket yüzeyinin ve top boyutunun hızlı bir şekilde azalması ile daha geniş bir arama alanı oluşturduğundan ajanlar kötü performans sergilemeye başlamıştır. Bu aşamada oyun kuralları ajanlara kural tabanlı olarak verilmiştir. Bunun sonunda elde edilen başarım ve bunun temel durumla karşılaştırılması Şekil 11'de verilmiştir. ELO skoru açısından önemli bir artış sağlanmasına karşın bu süreçteki adım sayısı artırılmıştır. Sonuçta, ajanın hedeflenen senaryoda performans gösterme yeteneği, eğitim süresine göre önceliklendirilmiştir.

Bunu takiben, ajanların x yönündeki hızlanması önündeki engel kaldırılmış ve ajanlar üç boyutta da serbestçe hareket edebilecek bir konuma getirilmiştir. Bu, arama alanını önemli ölçüde genişletilmiş ve ajanların daha karmaşık etkileşimler içinde bulunmasını gerektirmiştir. Ajanlar, Şekil 10'daki düz yeşil çizgiyle gösterildiği gibi, nadiren bir topa dokunabilmiş ve genel olarak doğru bir şekilde topa vuramamıştır. Bu sorunun üstesinden gelmek için, öğelerin büyütülmüş bir versiyonu elde edilmiş ve arama alanını küçültmek için ajanların her iki tarafına z ekseni boyunca sınırlayıcı duvarlar yerleştirmiştir. Duvarlar arasındaki mesafe, eğitimin bir parçası olmuş ve zamanla arttırılmıştır.

Açık mavi çizgi, bu konfigürasyondaki performansı gösterir. Ajanlar, duvarların birbirinden çok uzak olmadığı bir yere kadar iyi performans göstermiş, ancak ardından arama alanı tekrar yönetilemez hale gelmiştir. Ajanları hedef eyleme doğru daha etkin bir şekilde yönlendirmek için, topun potansiyel iniş pozisyonu, dolaylı ipucu olarak verilmiştir (bu eğitim konfigürasyonunda açıklanmıştır). Kırmızı çizgi, bu yaklaşımı kullanmanın yararını göstermektedir. Ancak burada, oyundaki eşyaların büyütülmüş versiyonları ve tenis kortu kenarlarında bulunan sınırlayıcı duvarların kullanıldığı unutulmamalıdır.

Genel olarak, deneylerin çıktısı olarak başarıyla eğitilmiş iki model vardır. İlk model, güvenilir performans gösteren gerçekçi boyutta raketler ve ajanin x yönünde değişmeyen hızı ile eğitilmiştir. İkincisinde ise ajan gerçek dünyadaki hareket özgürlüğüne sahiptir ancak oyundaki temel nesnelerin boyutları büyütülmüştür. İkinci modelde ajanların kararsız davranış sergilediği görüldüğünden dolayı, sunulan çalışmanın test uygulamasında kullanıcıların oynaması ve ajanların performansının değerlendirmesi için ilk model kullanılmıştır.



Şekil 11. Oyun kurallarının açık bir şekilde ajanlara kural-tabanlı verilmesi sonrası elde edilen ELO değerleri.

Test uygulaması web platformunda barındırılmış ve sunulan çalışmanın değerlendirilmesi için bu kapsamda bir anket hazırlanmıştır. Ankete katılanlardan oyun, etkileşim fiziği ve kuralları hakkında bilgi sahibi olmaları için oyunu oynamaları istenmiştir. Ankete katılan toplam kişi sayısı 45'dir. Ankette katılımcılara her biri 1 dakikalık iki örnek video klip izletilmiştir. Bunların ilkinde uzman bir kullanıcının MÖ ajanına karşı oyunu, diğerinde ise iki MÖ ajanın birbirine karşı oynadığı oyun bulunmaktadır. Ankete katılanlardan iki videoyu ayırt etmesi ve bu farkın nasıl tespit ettiğini bildirmesi istenmiştir. Genel olarak, katılımcıların sadece %33.5'i iki video arasındaki farkı doğru bir şekilde bulabilmiş ve %35.8'i ise MÖ ajanını uzman bir kullanıcıyla karıştırmış ve bu soruyu yanlış yanıtlamıştır. Yanıt verenlerin %30,7'i iki oyun arasında önemli bir fark görmemiştir. Özetle, MÖ ajanını uzman bir kullanıcıyla karıştıran ve MÖ ajanı ile gerçek kullanıcı arasındaki farkı ayırt edemeyen katılımcılar, toplam katılımcıların %66.5'ini oluşturmaktadır. Bu, katılımcıların çoğunluğunun akıllı bir ajanın, uzman kullanıcı davranışını yüksek doğrulukla simüle edebileceğine inandığı anlamına gelmektedir.

Bu iddia, katılımcıların anket kapsamında bir sonraki soruya verdiği yanıtlarla da desteklenmektedir. Bu kapsamda, katılımcılardan iki oyun arasında benzerliği 1 (benzer değil) - 5 (çok benzer) ölçeğine göre değerlendirmeleri istenmiştir. Çoğunluk, MÖ ajanının tatmin edici performansını doğrulayacak bir şekilde 4 (%53,8) ve 3 (%38,5) puan vermiştir (ortalama 3,46). Ek olarak, ankette Video 1'den uzman kullanıcının ve Video 2'den MÖ ajanının performansının ayrı ayrı değerlendirmeleri istenmiştir. Uzman kullanıcı ortalamada 3,23 puan alırken ve MÖ ajanına 3,39 puan verilmiştir. Bu puanlar o kadar yüksek olmasa da, birbirine yakın çıkmıştır. Genel olarak, bu anket, MÖ ajan davranışının, yanıt verenlerin farkı anlayamayacakları veya tereddütte kalacak ölçüde uzman kullanıcının davranışını başarılı bir şekilde simüle edebileceğini göstermiştir. Bu sonuçlara göre, kullanılan model üzerinde PÖ-tabanlı eğitilen ajanlar tenis oyununu başarılı bir şekilde simüle etmektedir.

3. Sonuç

Bu çalışma, tenis oyununu pekiştirmeli öğrenme kapsamında akıllı ajanlar ile oynamak için zengin görsel ve fiziksel içeriğe sahip bir simülasyon ortamı sunmaktadır. Oyundaki gerçek dünya nesnelerinin (top, raket, ajan) fiziksel etkileşimi, grafiksel temsili ve mantıksal davranışları Unity Gerçek Zamanlı Geliştirme Platformu kullanılarak yapılmıştır. İyi kurgulanmış ortam ile eğitim sırasında doğru ve çok yönlü gözlemlerin toplanması sağlanarak elde edilen veriler/bulgular ML-Agents Toolkit aracılığı ile eğitim sırasında kullanılmıştır.

Tenis oyununu, kort alanı içindeki herhangi bir noktadan topa vurma dahil olmak üzere MÖ ajanından karmaşık etkileşimler (ajanın her üç eksen boyunca hareketi ve rotasyonu) gerektirir. Arama uzayının büyüklüğü nedeniyle model sınırlandırılmış parametreler ile eğitilmiştir. Taklit eden bir öğrenme metodolojisinin eğitim sürecine eklenmesi ile modelin potansiyel iyileştirmesi sağlanabilir. Bu şekilde ajan, uzman bir kullanıcının oyunda nasıl davrandığını taklit edebilir. Bu kapsamda, uzman kullanıcılar ve bu kullanıcıların oyunları sınırlı sayıda ise, Üretken Çelişkili Taklit Öğreniminin (Ho ve Ermon (2016)) adapte edilmesi söz konusu olabilir, aksi takdirde Davranışsal Klonlama (Torabi ve vd. (2018)) yaklaşımı ajanların eğitimi için kullanılabilir. Seyrek ödüllerin olduğu bu tür karmaşık ortamlarda, Merak ödül sinyali (Pathak ve vd. (2017)) ve Rastgele Ağ Damıtma (Burda ve vd. (2018)) gibi içsel ödüllerin tanıtılması da literatürde önerilmektedir. Sunulan çalışmanın geliştirilmesi için tüm bu yaklaşımlar ayrı ayrı ya da aynı anda birlikte de kullanılabilir.

Genel olarak, bu çalışmada ajanların pekiştirmeli öğrenme ile eğitilebileceği tenis oyunu için gerçekçi bir simülasyon platformu oluşturulabileceği gösterilmiştir. Çalışma, tatmin edici davranış sergileyen, test uygulamasında akıllı tenis ajanlarının geliştirilmesi ile sonuçlanmıştır. Yapılan deneyler, büyük oranda yapay ajanlar ile uzman kullanıcıların benzer davranış sergilediğini göstermektedir.

Araştırmacıların katkısı

Bu araştırmada; Bakhtiyar Ospanov, araştırmanın tasarlanması, uygulama kodlamasının gerçekleştirilmesi, analizlerin yapılması, verilerin toplanması, bulguların değerlendirilmesi; M. Fatih Demirci araştırma sürecinin tasarlanması, izlenmesi, kontrolü, değerlendirilmesi ve makalenin hazırlanması kısımlarına katkı sağlamıştır.

Çıkar çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Kaynaklar

Akkaya I, Andrychowicz M, Chociej M, Litwin M, McGrew B, Petron A, Paino A, Plappert M, Powell G, Ribas R. (2019). Solving rubik's cube with a robot hand. arXiv preprint arXiv:1910.07113, 10. https://arxiv.org/abs/1910.07113

Albers P, de Vries D. (2001). Elo-rating as a tool in the sequential estimation of dominance strengths. Animal Behaviour, pages 489–495. https://palbers.home.xs4all.nl/AlbersDeVries.pdf

Alzubi J, Nayyar A, Kumar A. (2018). Machine learning from theory to algorithms: an overview. In Journal of physics: conference series, volume 1142, page 012012. IOP Publishing. https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012/meta

Ayodele T. (2010). Types of machine learning algorithms. New advances in machine learning, 3:19–48. https://www.intechopen.com/chapters/10694

Burda, Y., Edwards, H., Storkey, A., Klimov, O. (2018). Exploration by random network distillation. arXiv preprint arXiv:1810.12894. https://openreview.net/forum?id=H1lJJnR5Ym

Cao Z, Wong K, Bai Q, Lin C. (2020). Hierarchical and non-hierarchical multi-agent interactions based on unity reinforcement learning. Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 2095-2097. https://www.ifaamas.org/Proceedings/aamas2020/pdfs/p2095.pdf

Gao W, Graesser L, Choromanski K, Song X, Lazic N, Sanketi P, Sindhwani V, Jaitly N. (2020). Robotic Table Tennis with Model-Free Reinforcement Learning. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, pp. 5556-5563, doi: 10.1109/IROS45743.2020.9341191. https://arxiv.org/abs/2003.14398

Ho J, Ermon S. (2016). Generative adversarial imitation learning. arXiv preprint arXiv:1606.03476. https://arxiv.org/abs/1606.03476

Jaderberg M, Czarnecki W, Dunning I, Marris L, Lever G, Castaneda A, Beattie C, Rabinowitz N, Morcos A, Ruderman A. (2019). Human-level performance in 3d multiplayer games with population-based reinforcement learning. Science, 364(6443):859–865. https://www.science.org/doi/10.1126/science.aau6249

Lample G, Chaplot D. (2017). Playing fps games with deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 31. https://dl.acm.org/doi/10.5555/3298483.3298548

Millington I, Funge J. (2009). Artificial intelligence for games. CRC Press. https://spada.uns.ac.id/pluginfile.php/629724/mod_resource/content/1/gameng_AIFG.pdf

Muelling K, Boularias A, Mohler B, Schölkopf B, Peters J. (2014). Learning strategies in table tennis using inverse reinforcement learning. Biol Cybern 108(5):603–619. https://link.springer.com/article/10.1007/s00422-014-0599-1

Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science, 362(6419):1140–1144. https://www.science.org/doi/10.1126/science.aar6404

Pathak D, Agrawal P, Efros A, Darrell T. (2017). Curiosity driven exploration by self-supervised prediction. In International Conference on Machine Learning, pages 2778–2787. PMLR. https://proceedings.mlr.press/v70/pathak17a/pathak17a.pdf

Tebbe J, Krauch L, Gao Y, Zell A. (2021). Sample-efficient Reinforcement Learning in Robotic Table Tennis. IEEE International Conference on Robotics and Automation (ICRA 2021), May 31 - June 4. https://arxiv.org/abs/2011.03275

Torabi F, Warnell G, Stone P. (2018). Behavioral cloning from observation. arXiv preprint arXiv:1805.01954. https://arxiv.org/abs/1805.01954

Torrado R, Bontrager P, Togelius J, Liu J, Perez-Liebana D. (2018). Deep reinforcement learning for general video game ai. IEEE Conference on Computational Intelligence and Games (CIG), 1–8. IEEE. https://arxiv.org/abs/1806.02448

Wang X, Liu C., Sun L. (2022). Lightweight Deep Learning Models for Resource Constrained Devices. Computational Intelligence and Neuroscience, vol. 2022, Article ID 4623561, 10 pages. https://www.hindawi.com/journals/cin/si/635175/

Weeraman A. (2023). Erişim adresi: https://github.com/aweeraman/reinforcement-learning-tennis. Son erişim: Mayıs 2023.

Yurtsever, A., Dengiz, B. , Çakır, B. & Karaoğlan, İ. (2022). Dağıtık üretim içeren dağıtım şebekesi genişleme problemi için yeni bir matematiksel model. Journal of Turkish Operations Management , 6 (1) , 1134-1152. https://dergipark.org.tr/tr/pub/jtom/issue/70951/1106004