

Gymnasium Kütüphanesi ve Ortamlarının Analizi

Gymnasium kütüphanesinin kurulup kullanılması ve analiz edilmesi Ubuntu 24.04 sisteminde denenip Python 3.12.7 sürümü kullanılmış olup ona göre analiz edilmiştir.

Kurulum:

Kurulum sırasında kütüphanenin Gymnasium ile tam uyumluluk sergilemesine rağmen birkaç kontrol edilip düzeltilmesi gereken hatalar ile karşılaşabiliyor.

1. Python pip install kütüphanesi kullanılarak kurulmuştur.

Terminale yada python env ile ortam oluşturulup şu komut ile indirme yapılır. Kaynak-1

```
pip install gymnasium[all]
```

```
Downloading ale_py-0.10.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64
.whl (2.2 MB)
 2.2/2.2 MB 3.9 MB/s eta 0:00:00
Downloading Cython-0.29.37-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_6
4.manylinux_2_28_x86_64.whl (2.0 MB)
 2.0/2.0 MB 4.0 MB/s eta 0:00:00
Downloading Farama_Notifications-0.0.4-py3-none-any.whl (2.5 kB)
Downloading flax-0.10.6-py3-none-any.whl (447 kB)
Downloading jax-0.6.0-py3-none-any.whl (2.3 MB)
 2.3/2.3 MB 4.0 MB/s eta 0:00:00
Downloading jaxlib-0.6.0-cp312-cp312-manylinux2014_x86_64.whl (87.8 MB)
 87.8/87.8 MB 4.0 MB/s eta 0:00:00
Downloading moviepy-2.1.2-py3-none-any.whl (126 kB)
Downloading mujoco-3.3.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.
whl (6.7 MB)
 6.7/6.7 MB 4.0 MB/s eta 0:00:00
Downloading mujoco_py-2.1.2.14-py3-none-any.whl (2.4 MB)
 2.4/2.4 MB 440.7 kB/s eta 0:00:00
Downloading opencv_python-4.11.0.86-cp37-abi3-manylinux_2_17_x86_64.manylinux201
4_x86_64.whl (63.0 MB)
 2.6/63.0 MB 3.8 MB/s eta 0:00:16
```

Kurulum işlemi

210202053 Ayhan Kubilay Kazıcı

Kurulumu yaparken box2d-py hata veriyor hatanın açıklamasında swig bulunamadı şeklinde bir uyarı yazıyor.

```
error: command 'swig' failed: No such file or directory
[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.
ERROR: Failed building wheel for box2d-py
Running setup.py clean for box2d-py
Failed to build box2d-py
ERROR: ERROR: Failed to build installable wheels for some pyproject.toml based projects (box2d-py)
```

Box2d-py hatası

internette yaptığımız araştırma sonucu swig kuruyoruz

2. Swig kurma işlemi:

Python pip install kullanılarak Ubuntu 24.04 swig kurulumu yapılıyor. Kaynak-2

```
sudo apt-get install swig3.0
ln -s /usr/bin/swig3.0 /usr/bin/swig
```

3. Swig kurduktan sonra tekrardan gymnasium kurmaya çalışıyoruz fakat bu seferde ml-dtypes tensorflow ile uyumlu değil diye başka bir uyarı veriyor.

```
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
tensorflow 2.18.0 requires ml-dtypes<0.5.0,>=0.4.0, but you have ml-dtypes 0.5.1 which is incompatible.
```

4. Tensorflow 2.18.0 sürümünden tensorflow 2.19.0 sürümüne güncelleniyor.

```
pip install --upgrade tensorflow
```

5. Deneme amaçlı jupyter notebook açılıyor ve çalıştırılıyor.

```
[1]: import gymnasium as gym

env = gym.make("CartPole-v1", render_mode="human")
```

6. Kurulduktan sonra gymnasium kütüphanesinin mujoco eklentisi çalışmadığı için yeni bir conda environment oluşturulup eski bir python sürümüne geçiliyor ve OpenGL grafik kütüphaneleri kuruluyor. Kaynak 3.

6.1 Anaconda ve git indiriliyor (bizde daha önce indirilmişti)

6.2 Mujoco kaynak 3 de verilen yönergedeki linkden indirilip .mujoco dosyasına atılıyor.

6.3 Mujoco dosyası environmente ekleniyor.

6.4 Mujoco dosyasındaki simulasyon python ile çalıştırılıp kurulum olmuş mu diye kontrol ediliyor.

6.5 Ortam için ayrı bir şekilde oluşturulması gerekiyor ve conda ayarlanıyor OpenGL kütüphaneleri kuruluyor. Mujoco environmentte kullanılmak üzere git den klonlanıyor ve test ediliyor.

7. Kodun çalıştırılması test aşaması.

```
import gymnasium as gym

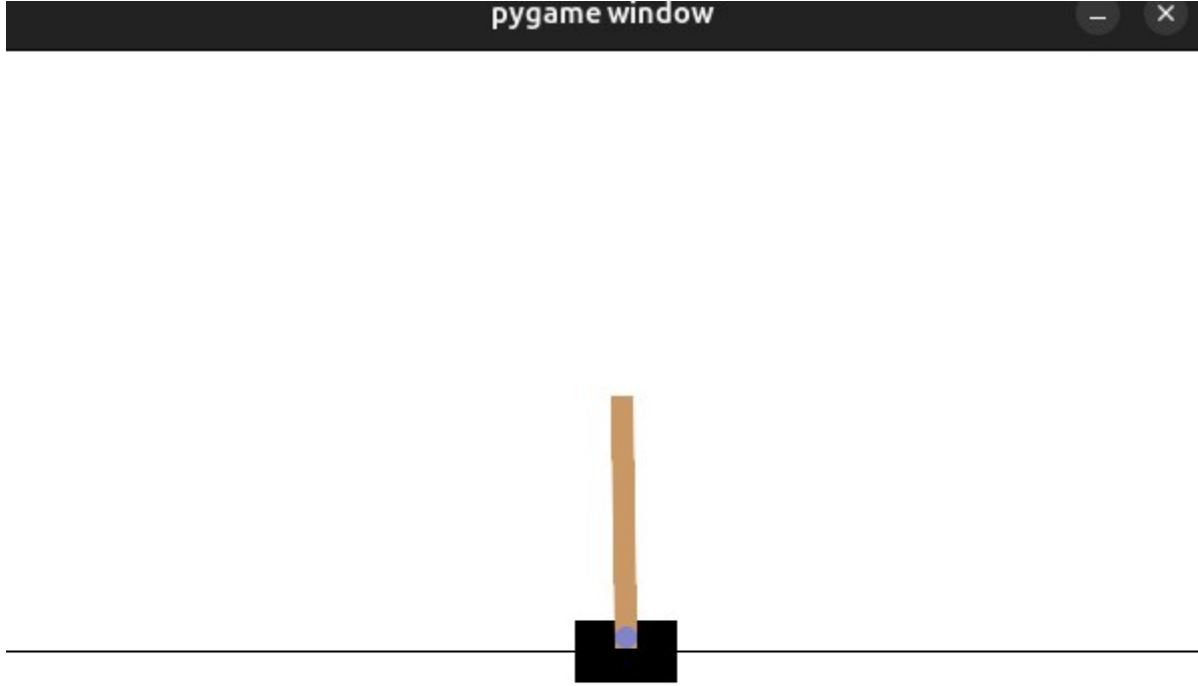
# Ortamı başlat
env = gym.make("CartPole-v1", render_mode="human")

observation, info = env.reset()
done = False

while not done:
    action = env.action_space.sample() # rastgele aksiyon
    observation, reward, terminated, truncated, info = env.step(action)
    done = terminated or truncated

env.close()
```

[gymnasiumtestCode](#)



[kod çalıştırıldığında gelen görsel](#)

Kod çalıştırıldığında karşımıza bir simulasyon geliyor.

Gymnasium nedir ne işe yarar?

Gymnasium bir python kütüphanesidir ve pekiştirmeli öğrenme algoritmaları için standartlaştırılmış ortamlar sağlar. OpenAI tarafından geliştirilmiş olan gym kütüphanesinin daha güncel ve kararlı bir modelidir.

- Gymnasium pekiştirmeli öğrenme algoritmalarını test etmek için standart görevler sunar
- Ajanların öğrenme ve karar verme yeteneklerini sınamak için kullanılır.
- Pekiştirmeli Öğrenme ortamı geliştirmek için arayüz sağlar.

Gymnasium Ortamları

Gymnasium ortamları kullanım şekillerine göre ayrılmıştır.

1. Classic Control:

Basit fiziksel kontrol problemleri için küçük boyutlu bir eylem uzayına sahiptir eğitim süresi kısadır. Stokastik bir ortam sağlar. Fakat bazı ortamlar random seed atadığından dolayı böyledir. Belirli bir kuralı vardır. Cartpole seed ayarı sabit olduğu zaman deterministik olmaktadır. [\(Kaynak 4\)](#)

Ortamlara örnek olarak:

- CartPole
- MountainCar
- Acrobot
- Pendulum

Genelde hızlı prototipleme için kullanılır eğitim süresi kısadır temel algoritmaları test etmek için idealdir. Algoritma'nın temel mantığına bakmak için kullanılabilir.

Gözlem uzayı açısından bakıldığında hepsi Box tipi sürekli uzaylardır.

- **Acrobot:** Aşağıdaki görsel Acrobot uzayının gözlem uzayıdır.

Action Space	Discrete(3)
Observation Shape	(6,)
Observation High	[1. 1. 1. 1. 12.57 28.27]
Observation Low	[-1. -1. -1. -1. -12.57 -28.27]
Import	<code>gym.make("Acrobot-v1")</code>

6 Boyutludur. Discrete'dir yani alabileceği sayılar belirlidir. Sınırları fotoğraftaki gibidir. "`env.env.book_or_nips = 'nips'`" kodu ile dinamikler değiştirilir.

- Amaç hedefe en kısa yoldan ulaşmak. Ulaşılamadığı her adım için -1 yazılır

- **CartPole:** Discrete(2), 4 boyutlu, [+4.8 inf +0.42 inf]
 - Başlangıç olarak -0.05 ile 0.05 arasında random atılıp başlanılır.
 - Bitiş kutup açısı ve araba konumuna bağlı olarak maks 500 episot olacak şekilde biter.
 - Eşik 475 puan, dik tutulduğu her süre +1 puan verilir.
- **MountainCar:** Discrete(3), shape 2, high [0.6 0.07], low [-1.2 -0.07]
 - Başlangıç: -0.6, -0.4 arasında başlangıç konumu verilir, 0 ivme ile başlar
 - Bitiş: Arabanın konumu 0.5 e eşitse yada büyükse veya episot 200 den fazla ise biter.
- **Pendulum:** Box(-2.0, 2.0, (1,)), float32, Shape 3, high [1. 1. 8.], low [-1. -1. -8.]
`gym.make('Pendulum-v1', g=9.81)` g yer çekimini temsil eder. Değiştirilebilir.
 - Başlangıç: Dairede olduğundan pi sayısının aralığında başlar. 1 veya -1 ivme arasında rastgele başlar.

Ortamların hepsi **epizodiktir**. Başlangıcı ve bitişi vardır ve bellidir. Bütün ortamlar **tam gözlemlenebilir** ortamlardır.

2. Box2D

Fizik motoru 2 boyutlu daha karmaşık ortamlar sunar. Yerçekimi, çarpışma, sürtünme gibi fizik kanunları barındırır. **Deterministik** bir ortamdır. Herhangi bir randomize bir şey oluşturmaz fizik motorunun dışına çıkmaz.

Ortamlar:

- **BipedalWalker**: Box tipindedir. Box(-1.0, 1.0, (4,), float32), 24 boyutludur,
 - high: [3.14 5. 5. 5. 3.14 5. 3.14 5. 5. 3.14 5. 3.14 5. 5. 1. 1. 1. 1. 1. 1. 1. 1.]
 - low: [-3.14 -5.-5.-5. -3.14 -5. -3.14 -5. -0. -3.14 -5.-3.14 -5. -0.-1.-1.-1.-1.-1.-1.-1.-1.-1.-1.]
 - env = gym.make("BipedalWalker-v3", hardcore=True) ortamı hardcore olarak değiştirir. - -
 - Epizodiktir. Hep aynı şekilde başlar ve bitiş koşulları belirlidir.
 - Tam gözlemlenebilir.
- **LunarLander**: Discrete eylem yapısına sahiptir. Discrete(4), 8 bboyutludur.
 - High: [1.5 1.5 5. 5. 3.14 5. 1. 1.]
 - Low: [-1.5 -1.5 -5. -5. -3.14 -5. -0. -0.]
 - Rastgele bir kuvvet ile başlar
 - İniş kazasına göre biter.
 - Tam gözlemlenebilir.
 - Epizodiktir.
 - env = gym.make(
"LunarLander-v2",
continuous: bool = False,
gravity: float = -10.0,
enable_wind: bool = False,
wind_power: float = 15.0,
turbulence_power: float = 1.5,)

Detaylı incelemek için: [Kaynak 5](#)

- **CarRacing:** Box yapısı, Box([-1. 0. 0.], 1.0, (3,), float32), boyutu (96, 96, 3)
 - high: 255
 - low: 0
 - Araba yolun merkezinde başlar
 - Tüm yol bitince bitebilir veya pistten uzaklık -100 değerine ulaşınca biter
 - Genelde Epizodiktir.
 - Kısmi gözlemlenebilir sayılır çünkü geçilen kare bir daha görülmez.
 - env = gym.make("CarRacing-v1", domain_randomize=True) → Eğer domain_randomize=True ise sürekli yeni harita geldiğinden epizodiklik biter.

```
# normal reset, this changes the colour scheme by default
env.reset()
```

```
# reset with colour scheme change
env.reset(options={"randomize": True})
```

```
# reset with no colour scheme change
env.reset(options={"randomize": False})
```

Ortamin rengi deđiřtirmesi ile ilgilidir ve domain_randomize true ise çalışır.

2 Boyutlu fizik tabanlı robotik simülasyonlarda ve genel olarak gerçek dünya problemine daha yakın testlerde kullanılır.

Ajanın karmařık ortamlarda karar vermesi gerekiyorsa, orta zorlukta benchmark istendiğinde ve yürüme iniř gibi görevler için kullanılır.

3. Toy Text

Durumu ve eylemi metin olarak ifade eder. Genellikle tablolara dayalı yöntemlerle çözülür. **Stokastik** bir ortam sağlar. Sürekli aynı řeylerin gerçekteřmesi için bir kural temel yoktur.

Örnek Ortamalar:

- **FrozenLake:** Eylem: Discrete(4), Gözlem: Discrete(16)
 - 4 yöne hareket edebilir.
 - Epizodiktir
 - Tam Gözlemlenebilir.
 - gym.make('FrozenLake-v1', desc=None, map_name="4x4", is_slippery=True)
 - is_slippery = true ise iki tarafa da eřit olasılıkla 1/3 oranında gidebilecektir deterministik olur False durumunda stokastik bir durumla karşılaşılr.
 - map_name önceden hazırlanmış haritaların isimlerini girdiğinizde farklı haritalar kullanılır.
- **Taxi:** Eylem Discrete(6), Gözlem Discrete(500)
 - Başlangıçta stokastik attığı adımlar deterministiktir.
 - Epizodiktir.
 - Tam Gözlemlenebilir.
 - gym.make("Taxi-v3")

- **CliffWalking**: Eylem: Discrete(4), Gözlem: Discrete(48)
 - Epizodiktir
 - Tam Gözlenmelebilir.
 - Ajan uçuruma basarsa başlangıca geri döner.
 - Ajan hedefe ulaştığında bölüm sona erer.
 - gym.make('CliffWalking-v0')

Temel Q-learning ve SARSA gibi algoritmaların test edilmesi için kullanılır. Durumlar açıkça görülebildiği için öğrenme süreci takip edilip anlaşılabilir.

Tablo temelli pekiştirmeli Öğrenme (value iteration, Q-table) algoritmaları öğretilirken kullanılabilir aynı zamanda düşük boyutlu ve ayrık problemlerde de kullanılabilir.

4. MuJoCo

Multi-Joint dynamics with Contact, karmaşık 3D robotik simülasyon ortamları için kullanılır. Gerçekçi fizik motoru vardır. Oluşturduğu ortam **Deterministik'tir**. Fizik motorunun vereceği tepkiler bir temele dayalıdır, değişmez.

Ortam Örnekleri:

- **HalfCheetah**: Eylem: Box(-1.0, 1.0, (6,)), float32), Gözlem boyutu: (17,)
 - high: [inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf]
 - low: [-inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf]
 - 0 a filtre ekleyerek stokastik şekilde başlatılır.
 - Bölüm uzunluğu 1000 den uzun oldu mu durdurulur.
 - Epizodiktir
 - Tam gözlemlenebilir.
 - env = gym.make('HalfCheetah-v4', ctrl_cost_weight=0.1,) Kaynak 6
- **Ant**: Eylem: Box(-1.0, 1.0, (8,)), float32), Gözlem Boyutu: (27,)
 - High: [inf inf]
 - Low: [-inf -inf]
 - Başlangıç: Ortalama 0 konumuna gürültü filtresi eklenir z eksenini biraz daha yüksekte olur.
 - Bitiş: Bölüm uzunluğu 1000 den uzun oldu mu kesilir.
 - Epizodiktir
 - Tam gözlemlenebilir.
 - env = gym.make('Ant-v4', ctrl_cost_weight=0.1, ...) → ctrl_cost_weight reward üzerinde değişiklik yapılmak için kullanılır. Çok fazla parametresi vardır. [Kaynak 7](#)

- **Hopper:** Eylem: Box(-1.0, 1.0, (3,)), float32), Gözlem Boyutu: (11,)
 - High: [inf inf inf inf inf inf inf inf inf inf inf]
 - Low: [-inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf]
 - Başlangıç: Belirli bir lokasyondan başlayıp istenildiği takdirde istenilen miktarda gürültü filtresi eklenebilmektedir.
 - Bitiş: Hazne belirtilen kapalı haznede bulunmuyorsa (Ayarlanabilir), Haznenin yüksekliği belirtilenin dışındaysa, açılı belirtilen şekilde değilse, oluşturma sırasında sağlıklı ise durdur açık ise bitirilir. Truncation, 1000 haneye ulaşıldığında.
 - Epizodiktir.
 - Tam gözlemlenebilir.
 - env = gym.make('Hopper-v4', ctrl_cost_weight=0.1, ...) → Çok fazla başlangıç opsiyonu vardır incelemek için: [Kaynak 8](#)

Ajanı sürekli aksiyon alanlarında kontrol öğrenimine maruz bırakır. Robot kinematiği, dinamik dengenin öğrenilmesi sağlanır.

Gelişmiş pekiştirmeli öğrenme algoritmalarını test ederken kullanılır.

5. Atari

Klasik atari oyunlarının emülasyonunu yapar. Ajan ekran pikselinden karar almayı öğrenir. Oluşturulan ortam **Stokastik** özelliklidir. Oyunlarda nelerin olacağı belirli değildir.

Ortam Örnekleri:

- **BreakoutNoFrameskip:** Eylem: Discrete(18), Gözlem şekli: (210, 160, 3)
 - High: 255
 - Low: 0
 - Epizodiktir
 - Kısmi Gözlemlenebilir
 - Oyun: Tuğlaları yok ederek puan kazanılır. Renklerin puanları farklıdır.
 - env = gym.make("ALE/Breakout-v5")
- **Pong:** Eylem: Discrete(18), Gözlem Boyutu: (210, 160, 3)
 - High: 255
 - Low: 0
 - Epizodiktir
 - Kısmi Gözlemlenebilir
 - Oyun: Top rakibin raketini geçerse puan kazanırsınız, sizin raketinizi geçerse puan azalır.
 - env = gym.make("ALE/Pong-v5")

- **SpaceInvadersNoFrameskip**: Eylem: Discrete(18), Gözlem Boyutu(210, 160,3)
 - High: 255
 - Low: 0
 - Epizodiktir
 - Kısmi Gözlemlenebilir
 - Uzay istilacılarını öldürdüğünüzde puan kazanırsınız. Arkadakiler daha çok puana sahiptir
 - env = gym.make("ALE/SpaceInvaders-v5")

Görsel girdilerde karar vermeyi sağlar. Ajanın piksel değerinden yola çıkarak bir şeyler öğrenmesine olanak tanır. Derin Q Ağı gibi derin öğrenme tabanlı yöntemlerin test edilmesinde kullanılır. Görsel dikkat, hedef takibi, refleks öğrenimi şeklinde öğrenimi sağlar.

210202053 Ayhan Kubilay Kazıcı

Kaynakça:

Kaynak 1 - <https://pypi.org/project/gymnasium/>

Kaynak 2 - <https://github.com/openai/gym/issues/3143>

Kaynak 3 - <https://gist.github.com/saratrajput/60b1310fe9d9df664f9983b38b50d5da>

Kaynak 4 - https://www.gymlibrary.dev/environments/classic_control/index.html

Kaynak 5 - https://www.gymlibrary.dev/environments/box2d/lunar_lander/

Kaynak 6 - https://www.gymlibrary.dev/environments/mujoco/half_cheetah/

Kaynak 7 - <https://www.gymlibrary.dev/environments/mujoco/ant/>

Kaynak 8 - <https://www.gymlibrary.dev/environments/mujoco/hopper/>