Manipülasyon Robotlarında Eylem Primitiflerinin Öğrenilmesi
Learning Action Primitives in Manipulation Robots

Burcu Kılıç
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, Türkiye
burcu.kilic@std.bogazici.edu.tr

Alper Ahmetoğlu
Intelligent Robot Lab
Brown University
Providence, United States
ahmetoglu.alper@gmail.com

Emre Uğur
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, Türkiye
emre.ugur@bogazici.edu.tr

***Özetçe* —Akıllı etkenler sürekli bir duyu-motor uzayında algılar ve hareket ederler. Öte yandan etkenlerin bulunduğu gerçek dünya ortamı sonlu sayıda nesnelerden oluşmakta, ve bu nesneler belli beceriler ile manipüle edilmektedir. Etkenin bulunduğu bu ortam düşünüldüğünde algı ve eylem primitifleri kullanmak doğal bir hal almaktadır. Önceki çalışmalar üst düzey eylemler (beceriler) kullanarak algı primitiflerini öğrenmekte, veya bu becerileri elle kodlanmış durumlardan öğrenmektedir. Bu çalışmada, etkileşim verisini kullanarak aynı anda eylem primitifleri ve nesne düzeyinde algı primitifleri öğrenmeyi mümkün kılan bir sinir yapısı öneriyoruz. Etkenin ortam ile etkileşimi etki odaklı eylem parametreleri aramayı sağlayan bir aktif öğrenme modülü rehberliğinde sürdürülmektedir. Deney sonuçları, modelin anlamlı eylem primitifleri bulabildiğini göstermektedir.**

Anahtar Kelimeler—aktif öğrenme, robotik, eylem primitifleri

***Abstract*—Intelligent agents sense and act in a continuous space at the sensorimotor level. However, most of our daily tasks require manipulating a finite number of objects using a fixed set of skills, suggesting the use of perceptual and action primitives. Previous studies either learn perceptual primitives from a set of high-level actions or learn these actions from hand-coded states. In this study, we propose a neural architecture that allows simultaneous learning of action primitives together with object-level perceptual primitives from interaction data. The interaction process is guided by an active learning module that enables the search for effect-focused action parameters. The experiment results show that our model can find meaningful action primitives.**

Keywords—active learning, robotics, action primitives

I. GİRİŞ

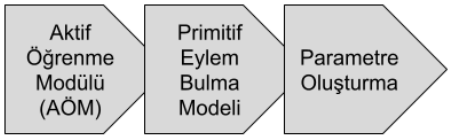
Akıllı etkenler algıçları ve eyleyicileri aracılığıyla bulundukları ortam ile etkileşim kurarlar. Bu algıçlar ve eyleyiciler ortamı sürekli bir uzayda göstermektedir. Örneğin, robot üzerine yerleştirilmiş renkli bir kamera her bir pikseldeki renk değerini 0 ila 255 arasında değişen üç bayt ile gösterirken robotun eklemlerindeki açı değeri $[-\pi, \pi]$ gibi sürekli aralıklarda gösterilmektedir. Bu gösterimlerin değerleri robotun kullandığı algıçlar ve eyleyicilere göre çeşitlilik gösterse de kullanılan gösterim sürekli uzaydadır. Öte yandan robotun içinde bulunduğu ortamlar genelde sonlu sayıda nesneler içermekte, ve bu nesneler belli başlı üst seviye beceriler ile manipüle edilebilmektedir. Örneğin, bir robota bulaşıkları bulaşık makinesine yerleştirme görevi verildiğinde robot sahip olduğu beceriler ve algıladığı nesneler üzerinden akıl yürüterek bir eylem planı oluşturabilir.

Robotun bu şekilde soyut muhakeme yapabilmesi, düşük seviye eylem değişkenleri yerine bunları soyutlayan üst seviye becerilere sahip olmasına ve ortamı düşük seviye algıç gösterimleri üzerinden algılamak yerine nesneler üzerinden algılamasına bağlıdır. Başka bir deyişle, soyut muhakeme için robotun duyu-motor sürekli gösterimleri üst seviye ayrık primitiflere çevrilmelidir. Ancak bu primitifler önceden verilmeden sonradan öğrenilebilir olmalıdır [1]. Önceki çalışmalarda robotun ortamı keşfederken öğrendiği nesne kategorileri ve eylem primitifleri ile sembolik planlama yapılabileceği gösterilmiştir [2]. Yine bu alandaki [3] çalışmasında, bir etki tahmin modelinin darboğaz katmanında nesne kategorileri öğrenilmektedir. Ancak bu çalışmada eylemlerin ayrık olarak el ile tasarlanıp verili olduğu varsayılmıştır. Gerçek ortamlarda ise robotun hareketi sürekli bir uzayda tanımlanır, ve bu nedenle ayrık eylem primitiflerinin de ayrıca bulunması gerekmektedir.

Bu çalışmada sürekli eylem parametreleri uzayında ayrık eylem primitiflerini keşfetmek için yenilikçi bir model önermekteyiz. Önerilen mimari hem nesneler hem de beceriler için gösterim öğrenirken etki tahmini yapan bir sinir ağıdır. Darboğaz katmanındaki ayrık etkileşim fonksiyonları sayesinde hem eylem primitifleri hem de nesneler kategorize edilmektedir. Ortamın keşfinin rastgele olmayan etkili bir biçimde yapılabilmesi için farklı yöntemler vardır. Durumların ziyaret sayısına göre ödül verme [4], yüksek öğrenme ilerlemesi (LP) gösteren önceden modellenmiş bölgelerden seçim yapma [5], episodik hafıza tabanlı özgünlüğe dayalı içsel ödülle keşif yapma [6] bu çalışmalardan bazılarıdır. Biz de bu çalışmamızda etki büyüklüğü bazlı aktif öğrenmeye dayalı bir keşif yöntemi önerdik. Bu çalışmadaki temel katkılarımız; (1) etki tahmin ederek nesne kategorileri ve primitif eylem bulma modeli, (2) parametre uzayını etkili şekilde arayabilen aktif öğrenme bazlı keşif modülü önermek ve (3) modelin öğrendiği eylem primitiflerinin, çok adım gerektiren hedefe ulaşma görevlerinde kullanılabileceğini göstermektir.

II. YÖNTEM

Girişte bahsedildiği üzere bir robotun soyut muhakeme yapabilmesi için eylem ve nesne sembollerini robotun ortamla etkileşiminden öğrenmek gerekmektedir. Bu bölümde, önce



Şekil 1: Aktif öğrenme modülünde elde edilen keşif verileri ile primitif eylem bulma modeli eğitilir. Daha sonra modelin eylem kodlayıcısı, primitifleri tekrar parametreleştirmede kullanılır.

robotun etkileşime girdiği ortam ve bu ortamdaki etkileşimlerin etkisini tahmin etme yöntemi anlatılacaktır. Modüllerin genel sırası Şekil 1'de gösterilmektedir.

A. Duyu-Motor Gösterimi

Bu çalışmada robotun ortam ile etkileşimini; ortamdaki nesneleri belirten durum (state - s), parametrik eylem, (a) ve eylemin etkisi (e) ile oluşturulan çokuzlu gösterim (s, a, e) ile kodladık. i nesnesinde gözlemlenen durum s_i olarak gösterilir ve Denklem (1)'de gösterildiği gibi nesnenin konumu, yönü, boyutları, tipi ve robotun kıskacına değip değmemesi ile tanımlanır.

$$s_i = (konum_i, yön_i, en_i, boy_i, derinlik_i, tip_i, değme_i) \quad (1)$$

Parametrik eylem, Denklem (2)'de gösterildiği şekilde, eylemin gerçekleştirilme sırasında (başlangıç, orta ve sonda) bulunduğu konum (x_j, y_j, z_j) ve o konumda kıskacının açık veya kapalı olmasıyla (g_j) ifade edilir. Robot, eylem parametrelerini, hedef nesne h'ye göreli olan konumlara giderek uygular.

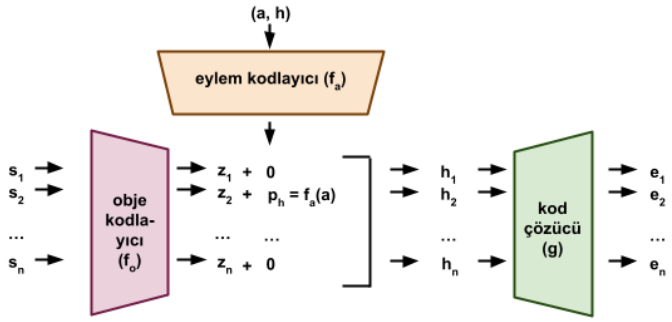
$$\begin{aligned} a = (&x_1, y_1, z_1, g_1, \\ &x_2, y_2, z_2, g_2, \\ &x_3, y_3, z_3, g_3) \end{aligned} \quad (2)$$

Eylemin nesneler üzerindeki etkisi e ile gösterilir ve e_i, nesne s_i'deki konum değişimini $(\nabla konum_i)$ tarif eder.

B. Primitif Eylem Bulma Modeli

Ayrık primitifleri keşfetmek için yöntemlerden birisi eylem parametre uzayında denetimsiz kümeleme yapmaktır. Fakat bu yöntem parametrelerin tipine ve kullanılan kümeleme algoritmasına bağlı olarak farklı sonuçlar vermektedir. Amacımız bulunan eylem primitiflerinin hedef odaklı kullanılmasıdır. Bundan dolayı hem nesne özellikleri hem de ortamda yaratılan etki ile ilintili bir şekilde eylem primitiflerini keşfetmeyi hedefledik. Bu amaçla nesne durumunu ve eylem parametrelerini girdi olarak alan ve etkiyi tahmin eden bir sinir yapısı içinde ayrıksı aktive olan darboğaz katmanında eylem primitiflerinin kategorize olması için yenilikçi bir sinir yapısı önerdik.

Bu yapı; eylem ve nesne kodlayıcı, ikili darboğaz katmanı ve bir kod çözücüden oluşmaktadır. Şekil 2'de gösterilen modelde eylem kodlayıcı $f(a)$ ve nesne kodlayıcının $f(o)$ aktivasyon katmanı olarak Doğrudan Geçiş Katmanı (Straight Through Layer) [7] kullanılır ve bu katmanda ikilileştirme işlemi, giriş değeri pozitif ise 1, değilse 0 olacak şekilde gerçekleştirilir. Çıkarılan ikili gösterimler birleştirilerek kod çözücüye verilir. Kod çözücü (g), eylem a'nın nesneler üzerindeki etkisini (e) tahmin eder.



Şekil 2: Eylem ve nesne kodlayıcılı ve ikili darboğaz katmanlı, etki tahmin ederek primitif bulma modeli. Nesne kodlayıcı (f_o), nesne s_i'leri ikili gösterim olan z_i'lere dönüştürür. Eylem kodlayıcı (f_a), verilen eylem parametresi a'yı ikili gösterim olan p_h'ye dönüştürür. p_h yalnızca hedef (h) nesne gösterimine (z_h) eklenir ve oluşturulan yeni vektörden kod çözücü (g) her nesne için e_i'yi tahmin eder.

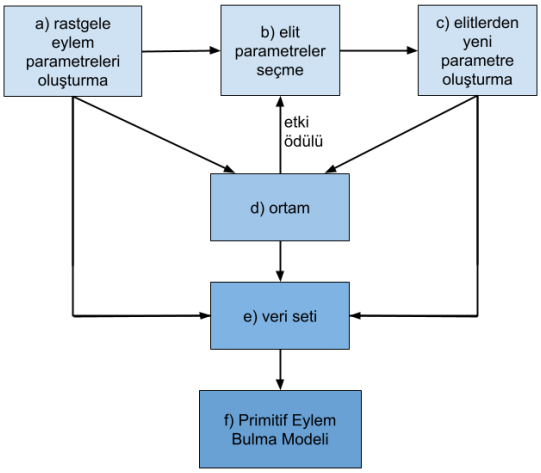
C. Aktif Öğrenme Modülü

Eylem parametreleri, robotun hareket edebileceği geniş ve sürekli bir uzayda tanımlanır. Robotun bu uzayda rastgele dolaşımı, çoğunlukla nesneye bir etkisi olmayan eylemler üretir. Modelimizin etki odaklı eylemler keşfedebilmesi için rastgele parametrelerle dolaşması yerine etki odaklı bir aktif öğrenme keşif modülüne ihtiyaç duyulmaktadır. Şekil 3'te görülen, evrim tabanlı politikadan [8] ilham alan bir eylem seçimine ve değişimli keşif ve sömürüye dayanan bir yöntem öneriyoruz. Bu modülde aşağıdaki adımlar tekrarlanmaktadır:

Öncelikle, rastgele eylem parametreleri oluşturulur (a) ve her biri rastgele oluşturulmuş ortamda (d) rastgele seçilmiş hedef nesneyle simüle edilir, oluşturduğu etki kaydedilir. Etkinin büyüklüğü, o eylemin ödülü olarak seçilir. Daha sonra elit parametreler seçilir (b). Bu aşama, oluşturulan tüm parametreler arasındaki en yüksek ödüle sahip eylemlerle gerçekleştirilir. Eylemlere etki büyüklüğünü ödül olarak verme ve elit parametre seçimi, her üç eksendeki etki için bağımsız olarak gerçekleştirilir. Buradaki amaç, robotun birbirinden ayırt edilebilecek eylemler öğrenmesini kolaylaştırmaktır. Daha sonra seçilen elit parametrelere yakın yeni parametreler oluşturulur (c) ve ortamda (d) simüle edilir. Tüm simülasyonlardaki (s, a, e) çokuzlusu toplanır ve bu verilerle (e) bir mini yığın oluşturulur. Bu mini yığın ile Şekil 2'deki primitif eylem bulma modeli (f) eğitilir.

D. Öğrenilen Primitiflerden Eylem Parametreleri Oluşturma

Robotun, öğrenilen eylem primitiflerini gerçekleştirebilmesi için eylem kodlayıcıdan çıkan ikili gösterimlerin tekrar parametrelere dönüştürülmesi gerekmektedir. Bir hedef primitife karşılık gelen eylem parametrelerini bulmak için şu optimizasyon süreci uygulanır:
Eğitimde kullanılan tüm eylem parametreleri alınır ve eylem kodlayıcıdan geçirilerek ikili gösterimleri bulunur. Bu gösterimleri hedef primitife yaklaştırmak için ikili çapraz entropi kaybı (BCE) kullanılarak eylem parametreleri Stokastik Gradyan İnişi (SGD) ile optimize edilir. Optimizasyon sonrası yığında, kodlayıcıdan hedef gösterimi çıkaran parametrelerin ortalaması, o primitifin karşılık parametresi olarak kabul edilir.



Şekil 3: Aktif Öğrenme Modülü (AÖM). a, b, c, d; robotun yaptığı keşif ve ortamla etkileşimini, e; bu keşiften oluşturulan veri setini, f de bu veri ile eğitilen primitif eylem bulma modelini gösterir.

E. Referans Modeli Olarak Rastgele Eylem Parametreleri

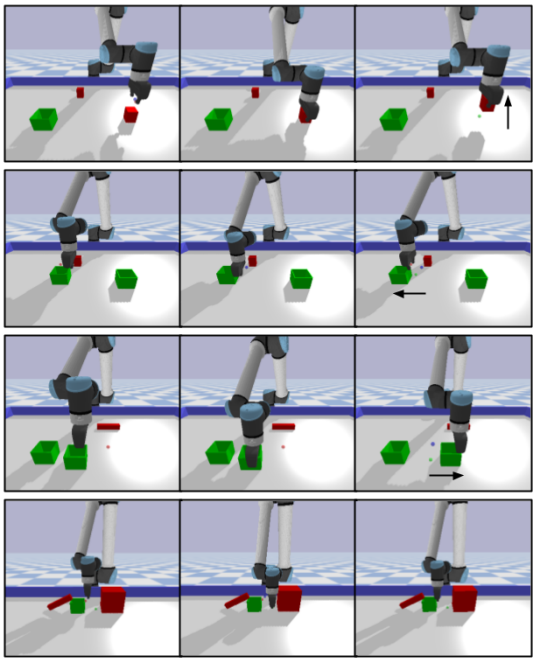
Keşif modülünün öğrenme performansını ölçebilmek için referans olarak modeli rastgele keşif verisiyle de eğittik. Bu aşama şöyle gerçekleşir:
Robot, rastgele nesnelerle oluşturulmuş deney ortamında hareket ettirilir. Rastgele bir hedef nesne seçilir, hareket için rastgele 12 eylem parametresi belirlenir ve robota bu parametreler uygulatılır. Nesnelerin başlangıç ve son konumları kaydedilip eylem sonrası nesneler üzerindeki etki hesaplanır. Eylem parametreleri ve etkiler kaydedilip daha sonra modelin eğitilmesi için bir veri seti oluşturulur.

III. Deney Ortamı

Deney ortamında, UR10 robotu rastgele konumlarda, boylarda ve tiplerde oluşturulmuş nesnelerle etkileşime girer. Ortam 6 adet nesneyle başlatılır. Nesnelerin başlangıç konumları, yönelimleri ve boyutları kaydedilip her nesne için s_i vektörü oluşturulur. Robot önce bir hedef nesne h seçer ve verilen eylem parametrelerini o nesneyi merkeze alarak uygular.

Kullanılan primitif eylem bulma modelinde, kodlayıcılar ve kod çözücü her biri 128 birimden oluşan 4 katman içerir. Eylem ve nesne kodlayıcının çıktı boyutu iki olarak ayarlanmıştır, bu nedenle modelin öğrenebileceği en çok $2^2 = 4$ farklı eylem ve nesne ikili gösterimleri bulunmaktadır.

Deneyde, keşif modülü 20 kere farklı rastgele başlangıç tohum (random seed) ile yinelenir. Her iterasyonda 20 eylem parametresi oluşturulur ve bu parametreler beşer defa simüle edilip ödüllerin ortalaması alınır. Daha sonra en iyi 5 eylem parametresi elit parametre seçilir ve 0.005 standart sapma ile yeni eylem parametreleri oluşturmada kullanılır. Bu oluşturulan yeni parametreler de 100 adımlık bir simülasyona tabi tutulur. Böylece her üç eksen için iki yüzer, toplamda 600 veriden oluşan bir mini yığın ile etki tahmin modeli 5 güncelleme dönemi, 0.00001 öğrenme çarpanı ve Adam optimizasyon algoritması ile eğitilir. Keşif sonrası öğrenilen primitifleri tekrar parametreleştirme, bahsedilen optimizasyon yöntemi ile 0.001 öğrenme çarpanı ve 5000 iterasyonda yapılır.



Şekil 4: İki bit çıktıda AÖM ile öğrenilmiş eşsiz beceriler. Üstten aşağı: 00, kaldırma eylemi; 01, sola itme eylemi; 10, sağa itme eylemi; 11, etkisiz eylem.

IV. Deney Sonuçları

A. Sistemin Keşfettiği Temel Beceriler

Bu bölümde, kodlayıcı çıktıları 1, 2 ve 3 boyutta değiştirilen primitif eylem bulma modeli AÖM ile eğitilmiş ve çıkan primitif eylemler tekrar parametreleştirilip robota uygulatılmıştır. Uygulanan beceriler bizim tarafımızdan isimlendirilmiştir. AÖM'yi karşılaştırma amaçlı referans modeliyle rastgele parametreler ile eğitilen modelin sonuçları da koyulmuştur. AÖM ile eğitilen, 2 bit çıktılı modelde öğrenilen 4 farklı beceri Şekil 4'te gösterilmiştir. Model kaldırma, sola itme, sağa itme eylemleri ve etkisiz eylem öğrenmiştir.

Modelin eylem kodlayıcısının çıktı boyutu 1 ve 3 olarak ayarlanarak da farklı deneyler yapılmıştır. 1 bit sembolde model kaldırmayı ve etkisiz eylemi öğrenmiştir. 3 bit sembolde ise modelin iterek kaldırmayı, etkisiz eylemi, ileri itmeyi ve kaldırmayı öğrendiği görülmüştür. Bir diğer deneyde de AÖM olmayan referans model; robotun, rastgele oluşturulmuş ortamlarda 100000 rastgele eylem parametresi uygulaması ile 500 epok eğitilmiştir. Modelin eylem kodlayıcısının çıktısı ikiye ayarlanmıştır. Model, yalnızca itme eylemini öğrenmiştir.

Yapılan dört farklı deneyde öğrenilen üst düzey beceriler Tablo I'de karşılaştırılmıştır. İlk üç sütun keşif modelimiz kullanılarak 1, 2 ve 3 bit çıktıda; 4. sütun da rastgele parametreler ile 2 bit modelde öğrenilen eylem primitiflerini göstermektedir. AÖM kullanıldığında 1 ve 2 bit modeller, kapasiteleri kadar eşsiz eylem öğrenmiş, 3 bit modelle yapılan deney de 'iterek kaldırma' gibi diğerlerine göre gereksinimi daha yüksek bir beceri öğrenmiştir. Rastgele parametrelerle yapılan deney ise kapasitesinin çok altında kalmış, yalnızca sağa itmeyi öğrenmiştir.

TABLO I: Farklı deneylerde öğrenilen farklı beceriler.

	1bit AÖM	2bit AÖM	3bit AÖM	2bit Referans Model
etkisiz	0 cm	0 cm	0 cm	-
kaldırma	14 cm	10 cm	14 cm	-
sağa itme	-	4 cm	-	4 cm
sola itme	-	6 cm	-	-
ileri itme	-	-	6 cm	-
iterek kaldırma	-	-	22 cm yukarı, 2 cm ileri	-

B. Etki Tahmin Hataları

Modellerin etki tahmin becerilerini ölçmek için gerçek etkilerle tahmin edilen etkinin farkına bakılır. Bunun için 2000 rastgele eylem parametresi ve ortamdan oluşan keşif verisi toplanır. Modeller sırasıyla bu veri üzerinde ileri yayılım yaparak etkileri tahmin eder. Gerçek etkilerle arasındaki mutlak farkların ortalamaları x, y ve z ekseninde ayrı ayrı alınır. Tablo II'de görüldüğü gibi en başarılı etki tahminini 2 bit sembollü AÖM kullanılan model yapmıştır. 1 ve 3 bit kullanılan modellerin hataları benzerken, rastgele eylem parametreleriyle eğitilen referans modelin performansı diğerlerine göre düşük kalmıştır. Burada 3 bitlik modelin 2 bitlik modelden daha düşük performans göstermesinin sebebinin, ortamdaki nesne ve yüksek düzey becerilerin fazla olmaması ve $2^3 = 8$ gösterimde anlamsız eylemler bulunmasının modelin genelleme becerisini engellediği olduğu kanısına vardık. Bu bulgumuz, modelde en iyi genelleştirmeyi yapabilme adına bilgiyi kısıtlamak gerektiğini gösteren Bilgi Darboğazı Yöntemi [9] ile uyuşmaktadır.

TABLO II: Farklı modellerin X, Y, Z eksenlerindeki hataları.

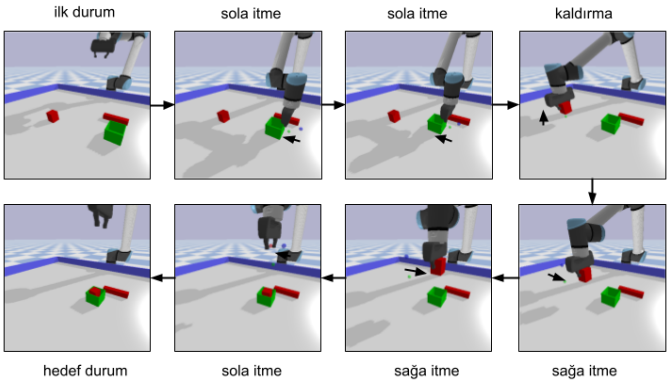
	1 bit AÖM	2 bit AÖM	3 bit AÖM	2 bit Referans Model
x	10.66 cm	10.88 cm	15.32 cm	23.82 cm
y	13.34 cm	5.74 cm	14.42 cm	74.18 cm
z	8.24 cm	14.42 cm	26.58 cm	23.2 cm

C. Çok Aşamalı Eylemler ile Hedefe Ulaşma

Öğrenilen becerilerin çok adım gerektiren hedefe ulaşmada kullanılabileceğini gösterebilmek için sonucu Şekil 5'te gösterilen çalışmayı yaptık. Eylem parametreleri için 2 bit AÖM ile öğrenilmiş model kullandık. İlk durumdan hedef duruma, yalnızca robotun öğrendiği eylem parametreleri kullanılarak gidildi. Şekil 4'te gösterildiği gibi önce yeşil kutu iki defa sola ittirildi, kırmızı küp kaldırıldı ve iki defa sağa sürüklendi, daha sonra bir defa sola sürüklenip bırakıldı. Son adımda, sola itme hareketinin son aşamasında kıskaç açıldığı için küp, içi boş yeşil kutunun içine düştü ve hedef duruma ulaşıldı. Bu deney robotun öğrendiği eylem primitiflerini kullanarak nitel olarak farklı, rastgele eylemler ile ulaşılması umulmayan daha karmaşık durumlara ulaşabileceğini göstermektedir.

V. Sonuçlar

Bu çalışmada önerdiğimiz sinir ağı bazlı yekpare bir mimari ile etki tahmin ederek farklı eylem primitifleri çıkarılabilmiş, aktif öğrenmeli keşfin rastgele eğitime göre daha anlamlı beceriler bulduğu gösterilmiş, ve bu becerilerin çok adımlı eylem dizilerinde hedefe ulaşmada kullanılabileceği denenmiştir. Önemli noktalardan biri eylem primitiflerinin mimarinin



Şekil 5: İlk durumdan hedef duruma götüren 6 adımlı eylem dizisi.

darboğaz katmanının kullanılarak öğrenilmesidir. Bu sayede öğrenilen eylem primitifleri yapay zeka planlayıcılarının kullandığı PDDL [10] formatına uygun üst-düzey eylemler ve obje sembolleri öğrenmekte, bu da sürekli uzayda gösterilen hedef bir duruma üst-düzey planlama ve muhakeme yaparak ulaşabilmenin yolunu açmaktadır.

Bu çalışmada darboğaz katmandaki sinir sayısı önceden verilmiş olup gelecekte bu sayının hataya bağlı olarak otomatik olarak ayarlanması planlanmaktadır. Modelin sembol öğrendikten sonra yeni bir ortama kendini uyarlaması çalışılabilir. Etki büyüklüğüne dayalı aktif öğrenme yerine merak tabanlı bir keşif modülü [11] kullanılabilir.

Bilgilendirme

Bilgilendirmeler Gizlenmiştir

Kaynaklar

[1] T. Taniguchi, E. Ugur, M. Hoffmann, L. Jamone, T. Nagai, B. Rosman, T. Matsuka, N. Iwahashi, E. Oztop, J. Piater *et al.*, "Symbol emergence in cognitive developmental systems: a survey," *IEEE transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 494–516, 2018.

[2] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2627–2633.

[3] A. Ahmetoglu, M. Y. Seker, J. Piater, E. Oztop, and E. Ugur, "Deepsym: Deep symbol generation and rule learning for planning from unsupervised robot interaction," *Journal of Artificial Intelligence Research*, vol. 75, pp. 709–745, 2022.

[4] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. Xi Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel, "# exploration: A study of count-based exploration for deep reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[5] "Ayrıntılar, çift-taraflı gizlilik ilkesi kapsamında gizlenmiştir."

[6] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt *et al.*, "Never give up: Learning directed exploration strategies," in *International Conference on Learning Representations*, 2020.

[7] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding straight-through estimator in training activation quantized neural nets," in *International Conference on Learning Representations (ICLR)*, 2019.

[8] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 11, pp. 241–276, 1999.

[9] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[10] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "Pddl| the planning domain definition language," 1998, [Online]. Available: https://api.semanticscholar.org/CorpusID:59656859.

[11] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.