A Survey on Transformers in Reinforcement Learning

Wenzhe Li1, Hao Luo, Zichuan Lin, Chongjie Zhang, Zongqing Lu, Deheng Ye

https://doi.org/10.48550/arXiv.2301.03044

MAKALE ÖZETİ

YUSUF AYKUT

1. Giriş

Pekiştirmeli Öğrenm,e sıralı karar-verme için matematiksel bir formülasyon sunar. Çoğunlukla markov zincir proses'ine uygun tanımlanan pekiştirmeli öğrenme modelleri aslında problemin optimizasyonunu araştırırlar. Böylece bunu akıllıca (gibi görünen) davranışları, farklı alanlarda, otomatik bir şekilde gerçekleştirmek için kullanabiliriz.

Transformer'ların son yıllarda, CNN ve RNN yapılarından üstün performans göstererek, paradigmamızın önün açması devrim niteliğinde bir gelişmedir. Transformer'lar özellikle uzun vadeli bağımlılıkların olduğu problemlerde kullanılmaya uygunlar ve denetimli öğrenme alanındaki bir takım başarısı, RL alanında nasıl kullanılabileceği bir araştırma konusu olmuştur.

Transformer'ların RL'deki çalışması yapılandırılmış state'ler(durumlar) arasında ilişkisel çıkarımlarda bulunmak için öz-dikkat mekanizması'nın kullanılmasıdır (Zembaldi et al., 2018). Takip eden senelerdeki çalışmalar; state representasyonu öğrenimi için transformer'ları kullanmayı, offline RL'de sıralı verilen kararları ve çoklu görevleri genelleştirme üzerinedir. RL'de transformer'ların genel olarak müdahale edebileceği başlıklar representasyonun ne kadar güçlü olduğu, sıralı durumları modelleme, önceden eğitilmiş büyük ölçekli modelin (örneğin GPT) RL ile spesifik bir öğrenme gerçekleştirmesidir.

2. Problem Kapsamı

Pekiştirmeli Öğrenme, bir markov zinciri representasyon yapısında politikayı öğrenmek için ajanın eğitilmesi sürecidir. Pekiştirmeli öğrenmede tanımladığımız markov zincir yapısı daha sonra optimallik fonksiyonlarımızda kullanacağımız bazı parametreler ve durum bilgileri içerir. $M = \langle S, A, P, r, \gamma \rangle$ burada sırayla *durum(state), aksiyon, olası sonraki durum(next_state), ödül(ya da return), discount factor listelenmiştir ve klasik anlamda SARSA(state-action-return-state-action) süreciyle optimal politikaya ulaşmamızı sağlayan optimizasyon gerçekleştirilir. Bu da bir RL problemlerinin temelinde bulunan şu denklemle sağlanır:* $J(\pi) = E_{\pi}\left[\sum_t \gamma^t r(s_t, a_t)\right]$. burada yapılması gereken iki şey vardır: birincisi temsil edilen durumun(state'in) nasıl öğrenileceğini belirlemektir –bir bakıma modeli kurmaktır- ikinci kısım ise aksiyonları belirlemektir.

Offline RL. Offline RL, ajanın eğitim sırasında ortam(environment) ile etkileşime girememesidir. Ancak, ajan eğitim sırasında keyfi bir politika üzerinden veri setine erişebilir. Buradaki bir problem ajanın çevreyle etkileşim kuramaması nedeniyle veri setinde yer almayan eylemlerin sonuçlarını doğru bir şekilde tahmin edememesidir. Bu durum, ajanın bilinmeyen eylemler için yanlışlıkla yüksek değerler öngörmesine yol açabilir buna aşırı tahmin –overestimation- denir. Bunu önlemek için modelin veri setindeki dağılıma yakın kalmasını sağlayacak şekilde Offline RL yaklaşımları geliştirilmiştir.

Goal-conditioned RL. GCRL burada ajan belirli bir hedefe yönelik öğrenme gerçekleştirir. Politika hedefe yönelik olarak optimize edilmektedir -$\pi(a|s, g)$-. Bu birden fazla hedefi içerebileceği gibi, çelişkili hedeflerinde optimizasyonunu barındırabilir. Bu konuda self-imitation learning, universal value function, hindsight relabeling GCRL alanında model başarısını arttırmak adına geliştirilmiş bazı yöntemlerdir.

Model-based RL. Model-free RL yönteminde ajan ortamın bir üst-dil'i diyebileceğimiz representasyonu üzerinden doğrudan politika ve değer fonksiyonlarını öğrenir. Bunun aksine Model-based RL çevrenin bir modelini öğrenir. Bu model ile gelecekteki durumları ve ödülleri tahmin edebilir. Burada ajan environment ile etkileşime girmeden evvel öğrenme gerçekleştirir.

3. RL'de Transformers Kullanımı

Sıralı data için en efektif modellerden birisidir. Transformer'lar ile ilgili önemli nokta kendi kendine dikkat mekanizmalarıdır. Uzun vadeli bağımlılıkları yakalamaya olanak sağlar. Bunu verileri tokenleştirme ve her token'in diğer tokenlerle nasıl bir ilişkisi olduğunu öğrenmesini sağlar. Bunu yapmak için her token, üç farklı vektöre dönüştürülür: **Query, Key, Value**. Bu vektörler tokenların birbirine ne kadar dikkat etmesi gerektiğini hesaplamak ve yeni temsiller oluşturmak için matematiksel formülasyonlarda kullanılır.

Transformer'ların denetimli ve denetimsiz öğrenme problemlerinde, özellikle uzun vadeli bağımlılıklar gerektiren problemlerde ciddi başarılar gösterdiği ortadayken RL'de kullanılıp kullanılamayacağı akla gelen bir soru olmaktadır. Peki bu mükün mü?

Transformer-based RL modellerini sınıflandırmadan evvel, RL'deki sinir ağı yapılarını ele alacağız ve zorluklarını gözden geçireceğiz. Bunu yapmamızın sebebi Transformer'ların kendisinin gelişmiş birer sinir ağı olmasıdır.

Architectures for function approximators. Function approximator diye bahsedilen özellikle DRL'de genel anlamda sinir ağlarıdır. Deep Q-Network'ün ortaya çıkışı (Mnih et al., 2015) bunun ilk örneklerinden birisidir(belki ilk örneğidir?) Burada bu sinir ağlarının geliştirilmesini 3 ana başlıkta inceeyebiliriz. Birincisi, inductive bias yani algoritmanın veriden bir şeyler öğrenirken dayandığı varsayımlar'ın nasıl ağ mimarisine dahil edileceğidir? Bunu yapmayı ajanın hangi durumlarda ne yapması gerektiğine dair varsayımları olabileceği için önemsemekteyiz. İkincisi, genel sinir ağı tekniklerinin RL'de uygulanıp uygulanamayacağı ile ilgilidir. Örneğin, regularization(aşırı öğrenmeyi engellemek), skip connection, batch normalization(veriyi normaliza etmek). Üçüncüsü DRL'yi dağıtık öğrenme için ölçeklendirmek. Normalde RL'de bir eylemin beklenen ödülü (expected value) hesaplanır. Distributional learning ise ödülün sadece ortalamasını değil, tüm olasılık dağılımını öğrenir. Bu, ajana daha fazla bilgi verir ve daha iyi kararlar almasına yardım edebilir.

Supervised Learning'de oldukça başarı gösteren Transformer'ların RL'e uygulanmasundaki temel zorluk; Transformer tabanlı mimarilerin yüksek bellek gerekliliğine ihtiyaç duyması ve gecikmeler yaşanmasıdır. Bu problemleri çözmek için hesaplama ve bellek verimliliğine yönelik araştırmalar yapılmaktadır ancak önerilen çözümler henüz istenilen düzeyde değil. Bu yüzden RL ile uğraşanlar tarafından yaygın bir şekilde kullanılmamaktadır.

Bu çalışmada Transformer'ların RL'de kullanımı araştırılmaktadır ve 4 ana başlıkta sınıflandırılmıştır.

3.1 Transformers'lar ile temsil öğrenme

Temsil öğrenme, bir ajanın ham verilerden (örneğin, bir oyundaki ekran görüntüsü) anlamlı özellikler çıkararak çevreyi daha iyi anlamasını ve kararlar almasını sağlayan bir süreçtir.

Transformers, genellikle metin veya görüntü gibi dizisel verileri işlemek için tasarlanmış bir yapay sinir ağı modelidir. RL'de ise bir ajan, çevreden gelen verilerle (örneğin, oyun ekranları, zaman içindeki durumlar) etkileşime girerek öğrenir. Bu veriler çoğunlukla **dizisel** bir doğaya sahiptir. Transformers, dikkat mekanizması (attention mechanism) sayesinde dizilerdeki elemanlar arasındaki ilişkileri etkili bir şekilde modelleyebilir. Bu özellik, RL'deki karmaşık ve değişken verileri anlamak için idealdir.

Zambaldi et al. (2018) Transformers'ı RL'de ilk kez kullanarak, ajanın gözlemindeki değişken sayıdaki varlıklar (entities) arasındaki ilişkileri modellemeyi önerdi. Bu, multi-head dot-product attention (çok kafalı dikkat mekanizması) ile yapıldı. AlphaStar **(Vinyals et al. 2019)** StarCraft II gibi karmaşık bir oyunda, ajanın gözlemindeki birimler ve binalar gibi çoklu varlıkları işlemek için Transformers kullanıldı. Burada, her varlık bir entity olarak kodlandı ve şu şekilde işlenmiştir:

$$Emb = Transformer(e1,\cdots,ei,\cdots)$$

Burada e_i ajanın i - inci varlığa dair gözlemidir(observation).

Genel olarak bu konudaki yaklaşımlara baktığımızda, Transformers, RL'de temsil öğrenme için güçlü bir araçtır, özellikle dizisel verileri ve karmaşık ilişkileri modellemede etkilidir. Şu konularda Transformers'ın RL'de temsil öğrenme için kullanımının giderek yaygınlaştığını söyleyebiliriz:

- **İlişkisel Öğrenme:** Gözlemdeki varlıklar veya morfolojik yapılar gibi ilişkisel veriler olduğunda, Transformers'ın dikkat mekanizması çok faydalıdır.
- **Karmaşık Veriler:** Görsel veya dil gibi karmaşık girdileri işlemek için Transformers'ın diğer alanlardaki başarısı RL'ye aktarılmaya çalışılıyor.
- **Önceden Eğitilmiş Modeller:** RL'de önceden eğitilmiş Transformer kodlayıcılarının kullanımı artıyor, bu da RL ile görüntü ve dil alanları arasında köprü kuruyor.

Transformer'lar ile zamansal dizileri işlemek de mümkündür. Şu şekilde gösterebiliriz:

$$Emb_{0:t} = Transformer(o_0,\cdots,o_t)$$

Burada o_t , t anındaki ajanın gözlemidir ve $Emb_{0:t}$ tarihsel gözlemlerin(bulunduğumuz gözlemden öncekiler) anlık gözleme gömülmesini ifade eder.

Transformers, uzun vadeli bağımlılıkları modellemede RNN'lerden (örneğin LSTM) üstündür ve parametre sayısı arttıkça daha iyi performans gösterir. Ancak RL sinyallerinde veri verimliliği (örneğin, az veriyle iyi öğrenme yeteneği) düşük kalabilir. Bu sorunu aşmak için yardımcı görevler (Banino et al., 2021) veya önceden eğitilmiş Transformer'lar (Li et al., 2022; Fan et al., 2022) kullanılmaktadır. Kısacası, Transformers zamansal dizileri işleyerek ajanın hafızasını güçlendirir fakat veri verimliliği gibi zorluklar devam etmektedir.

3.2 Transformers'lar ile model öğrenimi

Transformer mimarisi, model tabanlı pekiştirmeli öğrenme (model-based RL) algoritmalarında dünya modelinin (world model) temel taşlarından biri haline geldi. Model tabanlı RL'de amaç, ajanın çevrenin bir modelini öğrenip bu modelle gelecekteki durumları ve ödülleri tahmin etmesi.

Transformer’lar, bu işi geleneksel RNN’lerin (Recurrent Neural Networks) yerine alarak daha etkili bir yol sunuyor. Özellikle uzun vadeli bağımlılıkları yakalamada başarılı olmaları, onları bu alanda öne çıkarmaktadır.

Mesela, **Dreamer** gibi algoritmalar (Hafner et al., 2020; 2021; 2023), geçmişe dayalı dünya modellerinin gücünü gösterdi. Kısmi gözlemlenebilir ortamlarda veya hafıza gerektiren görevlerde Transformer’lar, RNN’lere kıyasla daha veri verimli çalışıyor. **Chen et al. (2022)**, Dreamer’daki RNN yapısını Transformer tabanlı bir modelle (TSSM) değiştirip uzun vadeli hafıza isteyen işlerde daha iyi sonuçlar aldı. Aynı şekilde, **IRIS** (Micheli et al., 2022) ve **TWM** (Robine et al., 2023), Transformer’ları auto-regressive öğrenmeyle birleştirip Atari benchmark’ında dikkat çekici performans sergiledi.

Transformer’ların bir diğer avantajı, model tabanlı RL’nin baş belası olan bileşik tahmin hatasını (compounding prediction error) azaltabilmesi. **Janner et al. (2021)** ve **Chen et al. (2022)**, Transformer’ların uzun dizilerde daha az hata biriktirdiğini ortaya koydu. Bu da ajanın daha uzun vadeli planlar yapmasına olanak tanıyor.

3.3 Sıralı Karar Verme için Transformer'lar

Offline RL'de Transformer'lar: Transformer mimarileri, çevrimdışı pekiştirmeli öğrenme (offline RL) alanında önemli gelişmelere yol açmıştır. Offline RL, ajanın çevreyle doğrudan etkileşime girmeden, önceden toplanmış veri üzerinden öğrendiği bir yaklaşımdır. Bu alanda öne çıkan iki model Decision Transformer (DT) ve Trajectory Transformer (TT) olmuştur. Decision Transformer, Chen ve arkadaşları (2021) tarafından geliştirilen ve pekiştirmeli öğrenmeyi bir dizi tahmin problemi olarak yeniden formüle eden bir modeldir. DT, geçmiş durumlar, eylemler ve beklenen getiri (return-to-go) değerlerinden oluşan bir diziyi işleyerek gelecekteki en uygun eylemi tahmin eder. Benzer şekilde, Trajectory Transformer (Janner ve arkadaşları, 2021), durumlar, eylemler ve ödülleri bir dizi olarak ele alır ve Transformer mimarisini kullanarak tüm yörüngeyi modelleyerek karar verme sürecini optimize eder. Geleneksel RL algoritmalarının aksine, bu Transformer tabanlı yaklaşımlar, çevrimdışı verileri daha etkili bir şekilde kullanabilmekte ve değer fonksiyonu tahminindeki aşırı iyimserlik (overestimation) sorunlarını azaltmaktadır.

Farklı Koşullara Bağlı Karar Verme: Transformer modelleri genellikle beklenen getiri (return-to-go) değerlerine dayalı tahminler yapmak için kullanılsa da, farklı bilgileri de koşul olarak kullanabilirler. Örneğin, Generalized Decision Transformer (GDT), sadece getiri değil, farklı istatistikleri de dikkate alarak karar verme sürecini iyileştirir. ESPER (Expectile Steps-PERturbed) gibi yaklaşımlar, çevrenin stokastik (rastgele) yapısını daha iyi modelleyebilmek için geliştirilmiştir. Diffuser modeli (Janner ve arkadaşları, 2022) ise, difüzyon modelleme teknikleriyle Transformer mimarisini birleştirerek daha esnek bir karar verme süreci sunmaktadır. DoC (Decision Transformer with Distributional Constraints) modeli ise, kararları çevrenin rastgeleliğinden bağımsız hale getiren bir yapı sunarak daha güvenilir tahminler yapmayı amaçlar.

Transformer Mimarisini Geliştirmek: DT ve TT gibi temel modeller, Markov benzeri durum-eylem bağımlılıklarını öğrenme konusunda bazı zorluklar yaşayabilmektedir. Bu zorlukları aşmak için Step Transformer gibi yeni mimariler önerilmiştir. Step Transformer, adım bazlı kararları daha iyi modelleyerek Markov özelliklerini daha etkili bir şekilde yakalayabilir. SPLT (SeParated Latent

Trajectory) Transformer, model tahmini ve politika ağacını ayrı tutarak daha güvenli ve kontrol edilebilir bir karar verme süreci sağlar. DeFog (Decision Transformer under Random Frame Dropping) modeli ise, eksik veri veya gözlemlerle başa çıkmak için geliştirilmiştir. Bu model, zaman içinde kaybolan bilgileri tahmin ederek modelin sağlamlığını artırır ve gerçek dünya uygulamalarında karşılaşılabilecek veri kayıplarına karşı dayanıklılık sağlar.

Online RL ve Çok Ajanlı Uygulamalar: Transformer tabanlı modeller şimdiye kadar ağırlıklı olarak çevrimdışı RL'de kullanılsa da, çevrimiçi (online) RL için de adapte edilmesi çalışmaları yapılmaktadır. Online Decision Transformer (ODT), çevrimdışı öğrenme ile çevrimiçi ince ayarlamaları birleştirerek, hem veri verimliliğini artırır hem de keşif sürecini daha etkili hale getirir. Multi-Agent Decision Transformer (MADT) ise, birden fazla ajanın öğrenmesini koordine etmek için geliştirilmiştir. Bu model, her ajanın bağımsız olarak karar vermesini sağlarken, ortak bir politika oluşturmasına yardımcı olarak çok ajanlı sistemlerde işbirliği ve rekabet dengesi sağlar.

3.4 Genelleştirilmiş Ajanlar için Transformer'lar

Çoklu Göreve Uyum Sağlayan Transformer Modelleri. Transformer'ların çoklu görevleri yerine getiren, farklı ortam ve senaryolarda başarılı olan genel amaçlı ajanlar geliştirmek için kullanımı giderek yaygınlaşmaktadır. Multi-Game Decision Transformer (MGDT), farklı Atari oyunlarından gelen verileri işleyerek genelleştirilmiş bir politika öğrenmiştir. Benzer şekilde, Switch Trajectory Transformer (SwitchTT), farklı görevler için özel bileşenleri aktif hale getirerek daha etkili bir öğrenme süreci sağlamaktadır.

Büyük Ölçekli Modeller ve Öğrenme Yaklaşımları Gato gibi büyük ölçekli Transformer tabanlı modeller, dil, görüntü ve eylem tabanlı karar verme süreçlerini aynı çatı altında birleştirmeyi başarmıştır. Bu tür modeller, çeşitli görevler arasında genelleme yapabilmek için büyük veri kümeleriyle eğitilmiştir. Prompt-DT gibi modeller ise, az örnekle öğrenme (few-shot learning) yeteneğini artırmak için görevle ilgili örnekleri giriş olarak kullanarak esnek karar verme mekanizmaları oluşturur.

Çapraz Alanlarda Kullanım ve Gelecek Perspektifi. Transformer'ların yalnızca pekiştirmeli öğrenme değil, daha geniş karar verme problemleri için de kullanımı araştırılmaktadır. Örneğin, Uni[MASK] modeli, pekiştirmeli öğrenme, hedefe bağlı öğrenme ve dinamik tahmin gibi farklı görevleri tek bir maskeleme çerçevesinde ele almaktadır. Aynı zamanda, dil modellerinden transfer öğrenme ile Transformer tabanlı RL modellerinin performansı artırılmaktadır. Bu tür çapraz alan araştırmalarının, gelecekte daha genelleştirilebilir ve güçlü ajanlar üretmesi beklenmektedir.

3.4 Sonuç

Transformer mimarileri, pekiştirmeli öğrenme (RL) alanında temsil öğrenme, dünya modeli oluşturma, sıralı karar verme ve genelleştirme gibi çeşitli rollerde başarıyla uygulanmıştır. İlişkisel yapıları modellemedeki ve uzun vadeli bağımlılıkları yakalamadaki üstünlükleri, RL problemlerinde önemli avantajlar sağlamaktadır.

Ancak, Transformer'ların RL'deki kullanımı hala birtakım zorluklarla karşı karşıyadır. Yüksek hesaplama maliyeti, önemli bellek gereksinimleri ve veri verimliliği sorunları, bu modellerin yaygın kullanımını

sınırlamaktadır. Gelecekteki araştırma yönleri arasında, çevrimiçi ve çevrimdışı öğrenme stratejilerinin birleştirilmesi, RL ile öz-denetimli öğrenmenin entegrasyonu ve Transformer yapılarının RL'ye özgü optimizasyonu öne çıkmaktadır.

Ayrıca, genelleştirilmiş ajanların geliştirilmesi ve büyük dil modellerinden transfer öğrenme ile RL performansının artırılması da gelecek vaat eden araştırma alanlarıdır. Diffüzyon modelleri gibi yeni yaklaşımlarla RL'nin birleştirilmesi ve insan geri bildirimiyle model ince ayarı gibi yöntemler, Transformer'ların RL alanındaki potansiyelini daha da artırabilir.

Bu gelişmeler ışığında, Transformer mimarilerinin pekiştirmeli öğrenme alanında giderek daha önemli bir rol oynaması ve karmaşık karar verme problemleri için daha etkili çözümler sunması beklenmektedir.