Agent Planlama ve Akıl Yürütme Kalıpları

Bir agent'ın "karar ver" adımının somut kalıpları: ReAct, plan-and-execute, reflection, ve bir loop'un ne zaman durması gerektiği.

Orta 20 dk

Son güncelleme: 22 Ağustos 2026
Çevrimiçi oku: https://www.learnforgex.com/tr/topics/agent-planning-and-reasoning

Agent Planlama ve Akıl Yürütme Kalıpları

"Bir AI Agent Nedir?", agent loop'unu gözlemle, karar ver, eyleme geç olarak anlatmıştı -- ama "karar ver" adımını "model karar verir"de bırakmıştı. Agent tasarımları arasındaki gerçek farkın çoğu, tam olarak o adımda yaşıyor. Bu ders, bir modelin "işte hedef ve şu ana kadar olanlar"ı "işte sıradaki eylem"e nasıl çevirdiğine dair üç somut kalıbı kapsıyor: **ReAct**, **plan-and-execute**, ve **reflection** -- ve hepsinin er ya da geç yanıtlaması gereken soruyla birlikte: loop ne zaman durur?

ReAct Kalıbı: Reasoning ve Acting

ReAct ("Reason + Act"in kısaltması), her eylemle görünür bir akıl yürütme adımını iç içe geçirir: bir tool call seçmeden önce, model önce kısa bir metin üretir -- şu an ne bildiğini, hâlâ neye ihtiyacı olduğunu, ve sıradaki eylemin neden mantıklı olduğunu açıklayan -- ve ancak ondan sonra eylemin kendisini üretir. Sıradaki gözlem (tool'un gerçek sonucu) geri beslenir, ve döngü tekrarlanır: akıl yürüt, eyleme geç, gözlemle, akıl yürüt, eyleme geç, gözlemle. Avantajı doğrudanlıktır -- her karar, mümkün olan en taze bilgiyle, bir seferde bir adım olarak verilir, ve görünür akıl yürütme metni, agent'ın belirli bir eylemi *neden* seçtiğini, yalnızca hangi eylemi seçtiğini görmekten çok daha kolay hâle getirir. Bedeli, modelin bir anlamda her tek adımda yeniden planlaması, ki bu önceden daha geniş bir plana bağlanmaktan daha az verimli olabilir.

Plan-and-Execute: Önceden Planlama

Plan-and-execute, loop'u iç içe geçirmek yerine iki ayrı faza böler. Önce, hedef verildiğinde, model herhangi bir eylem almadan önce baştan çok adımlı bir plan üretir -- "X'i ara, sonra X'i kullanarak Y'yi hesapla, sonra ikisini bir yanıtta birleştir." Sonra, (genelde çok daha basit) bir execution adımı, gerektiğinde tool'ları çağırarak o planı adım adım işler. Bir adımın gerçek sonucu planın geri kalanını geçersiz kılıyorsa plan revize edilebilir, ama

varsayılan davranış onu sonuna kadar takip etmektir. Bu, ReAct'in adım-adım uyarlanabilirliğinin bir kısmını, herhangi bir şey gerçekten çalışmadan önce tüm amaçlanan sıranın daha net, daha incelenebilir bir resmiyle takas eder -- bir hedef bilinen bir alt-görev kümesine temiz bir şekilde ayrıştığında, ve execution'dan önce tüm planı görmek önemli olduğunda kullanışlıdır (örneğin bir insan inceleme adımı için -- bkz. "Agent Davranışını Kontrol Etmek"teki "Human-in-the-Loop: Riskli Eylemlerden Önce Onay").

Reflection: Kendi İşinizi Kontrol Etmek ve Revize Etmek

Reflection, bir eylem (ya da hedefe yönelik tüm bir deneme) tamamlandıktan sonra ayrı bir adım ekler: modelden, çıktısını nihai kabul etmeden önce eleştirmesi istenir -- bu gerçekten hedefi yanıtlıyor mu? sorulana göre bu sonuç doğru görünüyor mu? -- ve eleştiri bir sorun bulursa, durmak yerine revize edip tekrar denemesi istenir. Bu, doğrudan "LLM Yetenekleri ve Sınırlamaları"daki "Hallucination: Kendinden Emin, Akıcı, Yanlış"ın tek bir yanıt seviyesinde anlattığı hata moduna hedefliyor: bir modelin, yanlış olan akıcı, kendinden emin bir yanıt üretmesi. Reflection bu riski ortadan kaldırmaz -- kontrolü yapan aynı model, yanıtı üreten modelle aynı sınırlamalara sahiptir -- ama atlanmak yerine loop içinde kendi kararı olarak ele alınan özel bir eleştiri adımı, aksi hâlde kontrolsüzce geçecek bazı hata sınıflarını yakalar.

Bu Kalıplar Arasında Seçim Yapmak

Bu kalıplar birbirini dışlamaz, ve gerçek agent'ların çoğu birden fazlasının parçalarını birleştirir. Başlangıç noktası olarak: ReAct, doğru sıradaki adımın gerçekten bir önceki adımın döndürdüğü şeye bağlı olduğu ve tüm sıranın önceden bilinemediği hedeflere uyar. Plan-and-execute, bilinen bir alt-görev kümesine temiz bir şekilde ayrışan, özellikle planı çalıştırmadan önce göstermenin değer kattığı hedeflere uyar. Reflection, diğer ikisine bir alternatiften çok, ikisinden birine bir eklemedir -- adımların nasıl karar verildiğinden bağımsız olarak, bir sonuç nihai yanıt olarak kabul edilmeden önce uygulanan bir kontrol.

Termination: Ne Zaman Duracağını Bilmek

Bu kalıpların her biri er ya da geç aynı soruyu yanıtlamak zorunda: loop gerçekte ne zaman biter? Tipik olarak üç koşul onu sonlandırır: modelin kendisi hedefin karşılandığına karar verip nihai bir yanıt üretir (normal durum); model kendi başına o sonuca varmadan önce, maksimum adım sayısı gibi harici bir sınıra ulaşılır; ya da hiçbir loop'un düzeltemeyeceği, kurtarılamaz bir hata oluşur (örneğin bir tool'un sürekli başarısız olması). İlki hedeftir; ikincisi, bu olmadığında bir güvenlik ağıdır -- asla sonuca varmayan bir karar fonksiyonu, aksi hâlde sonsuza kadar döner, elde hiçbir şey olmadan zaman ve maliyet tüketir. Bu kategorinin sıradaki dersi "Agent Davranışını Kontrol Etmek", o güvenlik ağının -- bir adım sınırı korkuluğunun -- tam olarak nasıl inşa edildiğini kapsıyor.

Best Practices

- Kalıbı hedefin şekline göre eşleştirin -- bkz. "Bu Kalıplar Arasında Seçim Yapmak" -- en son kullanılan kalıba varsayılan olarak yönelmek yerine.
- Yalnızca nihai yanıtı değil, her kararın görünür bir kaydını tutun (ReAct'in akıl yürütme metni, ya da bir plan-and-execute planı) -- bu, yanlış bir nihai yanıtı sonradan debug edilebilir kılan şeydir.
- Bir reflection adımını, özellikle "kendinden emin ama yanlış"ın maliyetli olduğu çıktılar için ekleyin -- bkz. "Reflection: Kendi İşinizi Kontrol Etmek ve Revize Etmek" -- riski göz önüne almadan tekdüze uygulamak yerine.

Yaygın Hatalar

- **ReAct, plan-and-execute, ve reflection'ı birbirini dışlayan seçimler olarak ele almak.** "Bu Kalıplar Arasında Seçim Yapmak"ın açıkladığı gibi, pratikteki çoğu agent bunları birleştirir -- özellikle reflection genelde bir alternatif değil, bir eklemedir.
- **"Model bittiğine karar verir"in ötesinde hiçbir termination koşulu olmayan bir loop inşa etmek.** "Termination: Ne Zaman Duracağını Bilmek", bir adım sınırı korkuluğunun neden modelin kendi durma kararının bir yedeği olarak önemli olduğunu, onun yerine geçmediğini anlattı.
- **Bir reflection adımının yanlış bir yanıt riskini tamamen ortadan kaldırdığını varsaymak.** "Reflection: Kendi İşinizi Kontrol Etmek ve Revize Etmek"in belirttiği gibi, kontrolü yapan adıma da "Hallucination: Kendinden Emin, Akıcı, Yanlış"daki aynı altta yatan model sınırlamaları uygulanır.

Özet, Cheat Sheet ve Terimler Sözlüğü

Özet

- "Bir AI Agent Nedir?"deki agent loop'unun "karar ver" adımı tek bir sabit mekanizma değil -- ReAct, plan-and-execute, ve reflection, bunun nasıl çalışabileceğine dair üç somut kalıp.
- ReAct, her eylemle görünür bir akıl yürütme adımını iç içe geçirir, en taze gözleme dayanarak her adımda yeniden planlar.
- Plan-and-execute, planlamayı (baştan tam bir çok adımlı plan) execution'dan (o planı işlemek, yalnızca gerekirse revize etmek) ayırır.
- Reflection, bir çıktıyı nihai kabul etmeden önce bir öz-eleştiri adımı ekler, "LLM Yetenekleri ve Sınırlamaları"nın anlattığı aynı "kendinden emin ama yanlış" hata moduna hedeflenir.

- Bu kalıplar pratikte birbirini dışlayan seçimler olmaktan çok birleşir, ve hepsinin er ya da geç bir termination koşuluna ihtiyacı var -- sırada, "Agent Davranışını Kontrol Etmek"te derinlemesine kapsanıyor.

Cheat Sheet

- ReAct = akıl yürüt -> eyleme geç -> gözlemle -> tekrarla, her adımda yeniden planla.
- Plan-and-execute = baştan tam plan yap -> adımları çalıştır -> yalnızca bir sonuç planı geçersiz kılarsa revize et.
- Reflection = bir çıktıyı sonlandırmadan önce eleştir; bazı "kendinden emin ama yanlış" hataları yakalar, riski ortadan kaldırmaz.
- Termination = model hedefin karşılandığına karar verir (normal durum), YA DA bir adım sınırına ulaşılır (güvenlik ağı), YA DA kurtarılamaz bir hata oluşur.

Terimler Sözlüğü

- **ReAct:** her eylemle görünür bir akıl yürütme adımını iç içe geçiren, her gözlemden sonra sıradaki adıma yeniden karar veren bir agent kalıbı.
- **Plan-and-execute:** herhangi bir eylem almadan önce tam bir çok adımlı plan üreten, sonra onu adım adım çalıştıran bir agent kalıbı.
- **Reflection:** bir agent'ın çıktısına, nihai kabul etmeden önce uygulanan, kendinden-emin-ama-yanlış hataları yakalamayı hedefleyen bir öz-eleştiri adımı.
- **Termination koşulu:** bir agent loop'unun durmasına neden olan şey -- modelin hedefin karşılandığına karar vermesi, bir adım sınırı korkuluğu, ya da kurtarılamaz bir hata.