

بسمه تعالی

هوش مصنوعی

عاملها - ۲

نیمسال دوم ۱۴۰۰-۱۳۹۹

دکتر مازیار پالهنک

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

- تعریف عامل
- نگاشت دنباله درک به عمل
- نمایش نگاشت بصورت جدول
- مقیاس کارآئی برای نشان دادن موفقیت عامل
- آنچه عقلانی است وابسته است به:
 - مقیاس کارآئی، دانش قبلی از محیط
 - اعمالی که در اختیار عامل است، دنباله درک عامل تا آن لحظه
- تفاوت عقلانیت با عقل کل بودن
- نیاز به جمع آوری دانش و یادگیری
- خودمختاری
- عوامل مهم در طراحی عامل:
 - کارآئی، محیط
 - اقدامگرها، حسگرها

ویژگیهای محیط کار

■ کاملاً مشاهده پذیر، نیمه مشاهده پذیر:

■ آیا حسگرها حالت کامل محیط در همه زمانها را در اختیار عامل قرار می دهند.

■ قطعی - غیر قطعی

■ حالت بعدی با دانستن حالت فعلی و عمل فعلی قابل دانستن است.

■ مرحله ای (واقعه ای) - ترتیبی

■ تجربه عامل به مراحل مجزا شکسته می شود که به هم وابسته نیستند. **مرحله ای**

■ شناسائی قطعات معیوب روی نوار نقاله واقعه ای می باشد.

■ شطرنج و رانندگی تاکسی ترتیبی هستند.

■ ایستا – پویا:

- اگر هنگام عمل عامل محیط بتواند تغییر کند محیط پویا است در غیر این صورت ایستا می باشد.
- اگر محیط تغییر نکند ولی مقیاس کارآئی عامل وابسته به زمان باشد محیط نیمه پویا است.

■ گسسته – پیوسته:

- تعداد محدود و واضحی حس و عمل
- تمایز گسسته/پیوسته قابل اعمال به حالت، زمان، ادراکات و اعمال
- شطرنج گسسته است چون تعداد حالات محدودی دارد.
- رانندگی تاکسی پیوسته است.

■ تک عامله – چند عامله:

- حل جورچین تک عامله
- رانندگی تاکسی چند عامله فقط همین عامل توی شهر رانندگی نمی کنه بلکه راننده های دیگه و پیاده روها و عامل های دیگه هم هستن پس میشه چند عامله

انواع محیط

تک عاملی	گسسته	ایستا	واقعه ای	قطعی	قابل مشاهده	
بله	بله	بله	خیر	بله	بله	جدول کلمات متقاطع
نه	بله	بله	خیر	بله	بله	شطرنج بدون ساعت
نه	بله	نیمه	خیر	بله	بله	شطرنج با ساعت
نه	نه	نه	خیر	نه	نه	رانندگی تاکسی

- یک تقسیم دیگر به عنوان **شناخته** و **ناشناخته** نیز وجود دارد.
- مربوط به اینکه عامل (یا طراح) قوانین حاکم بر آن دنیا را می شناسد یا خیر.
- یک دنیای ناشناخته می تواند کاملاً مشاهده پذیر باشد،
- و یک دنیای شناخته می تواند نیمه مشاهده پذیر باشد (مثل راننده تاکسی)

ساختار عاملها

- کار هوش مصنوعی طراحی برنامه عامل است.
- برنامه عامل نگاشت دنباله درک به عمل را انجام می دهد.
- این برنامه بر روی یک ماشین محاسباتی اجرا خواهد شد که به آن معماری می گوئیم. بنابر این:
- $\text{عامل} = \text{معماری} + \text{برنامه}$
- برنامه و معماری باید با هم سازگار باشند.
- اسکلت همه برنامه های عامل مشابه است: گرفتن درک – بازگرداندن عمل
- **برنامه فقط یک درک از محیط می گیرد.**

یک مثال ساده

```
function TABLE-DRIVEN-AGENT(percept) returns an action
  persistent: percepts, a sequence, initially empty
               table, a table of actions, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action  $\leftarrow$  LOOKUP(percepts, table)
  return action
```

Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

یک عاملی است که به صورت جدولی پیاده سازی شده
ورودی یک درک است و خروجی یک عمل است

- برای هر عاملهای ساده شاید مفید
- در نظر بگیرید:
- عاملی که از یک دوربین با نرخ ۳۰ قاب بر ثانیه و دقت تصویر ۴۸۰×۶۴۰ پیکسل استفاده می کند
- برای شطرنج در حدود ۱۰^{۱۵۰} ورودی لازم است.

- ایرادها به روش جدولی بطور کلی
 - اندازه جدول بسیار بزرگ است
 - طراح زمان برای ایجاد جدول ندارد
 - هیچ عاملی نمی تواند جدول را با تجربه اش یاد بگیرد.
 - پر کردن جدول نیز همواره ساده نیست.
- تلاش داریم که کد و حافظه کمتری استفاده کنیم.
- مثال: جدول جذر

انواع عاملها

- عامل انعکاسی ساده
- عامل انعکاسی ساده با حالت
- عامل هدف مبنا
- عامل سودمندی مبنا
- همه آنها می توانند قابلیت یادگیری داشته باشند.

عامل انعکاسی ساده: مثال

برای مثال دنیای جارو:

■ .

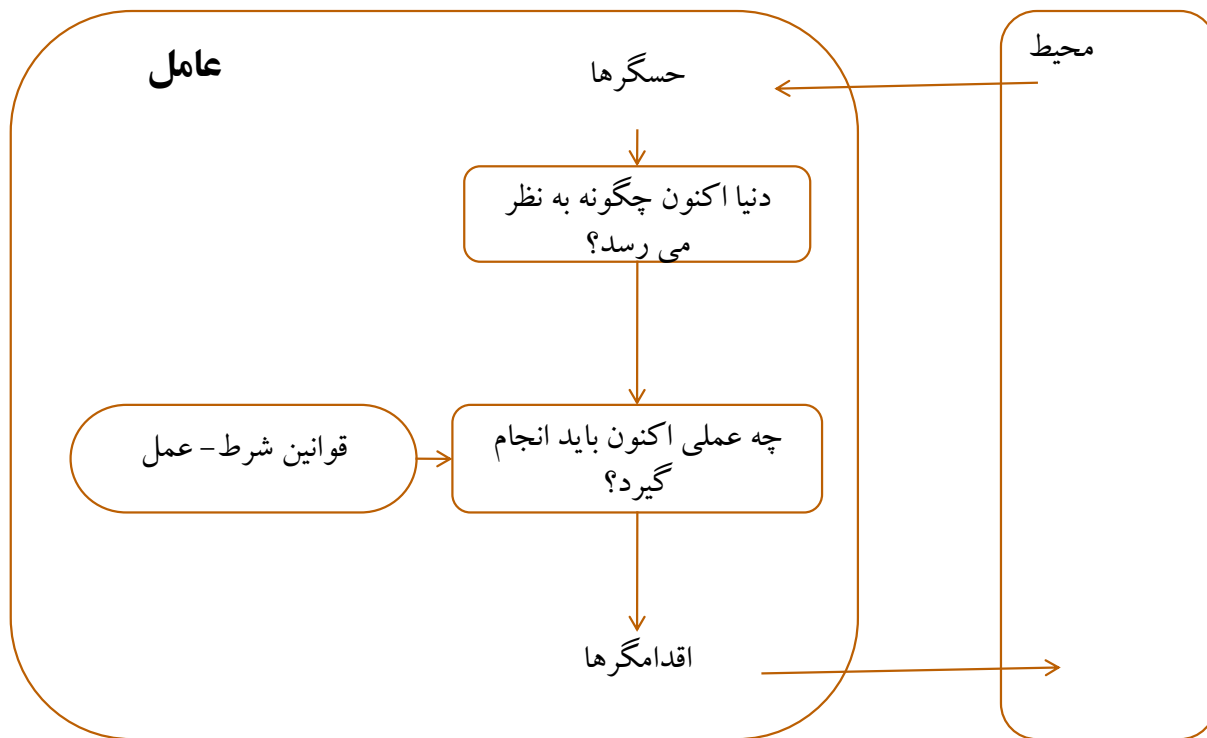
این میشه درک:

```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

این عامل یک درک رو میگیره و سریعا یک عمل انجام میده ینی سریع واکنش انجام میده و کاری هم به دنباله درک نداره

از محیط درک میاد و توسط حسگرها دریافت میشه و از روی این درک می بینیم الان دنیا به چه صورت است

Simple reflex agents عامل انعکاسی ساده



static: ینی متغییر rules وقتی که از تابع خارج می‌شیم از بین نمی‌رن و توی حافظه می‌مونن

مثال

- function Simple_Reflex-Agent(percept) returns an action
 - static: rules، مجموعه ای از قوانین شرط-عمل
 - $state \leftarrow \text{INTERPRET-INPUT}(\text{percept})$
 - $rule \leftarrow \text{RULE-MATCH}(state, rules)$
 - $action \leftarrow \text{RULE-ACTION}[rule]$
- return action

ممکن است نیاز به یک تجرید داشته باشیم

اگر خودروی مقابل تو ترمز کرد آنگاه ترمز کن

■ برای یک عامل انعکاسی ساده محیط باید کاملاً مشاهده-پذیر باشد.

■ بطور مثال در دنیای جارو اگر عامل فقط حسگر زباله داشته و حسگر مکان نداشته باشد.

■ با دریافت [dirt] مکش می کند،

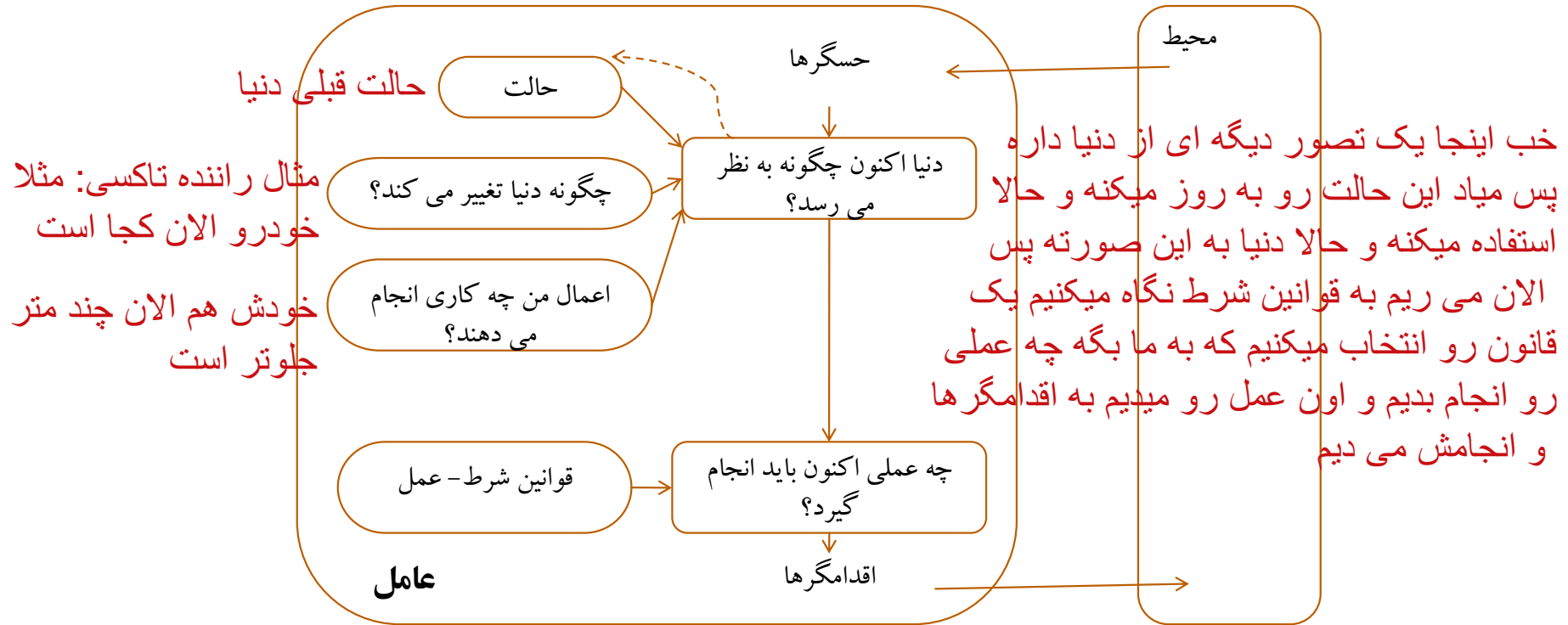
■ با دریافت [clean] چه عملی انجام می شود؟

■ اگر در A باشد و به چپ برود شکست می خورد. چون همیشه توی چپ می مونه

■ اگر در B باشد و به راست برود شکست می خورد. چون همیشه توی راست می مونه

میخوایم عامل انعکاسی باشه ولی توی یک محیط نیمه مشاهده پذیر باشه حالا باید چی کار کنیم؟

عامل مدل – مبنا Model-Based Reflex agents



■ دانش اینکه دنیا چگونه کار می کند را مدل می گویند.

```

function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
               model, a description of how the next state depends on current state and action
               rules, a set of condition–action rules
               action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action

```

Figure 2.12 A model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

عامل هدف مبنا

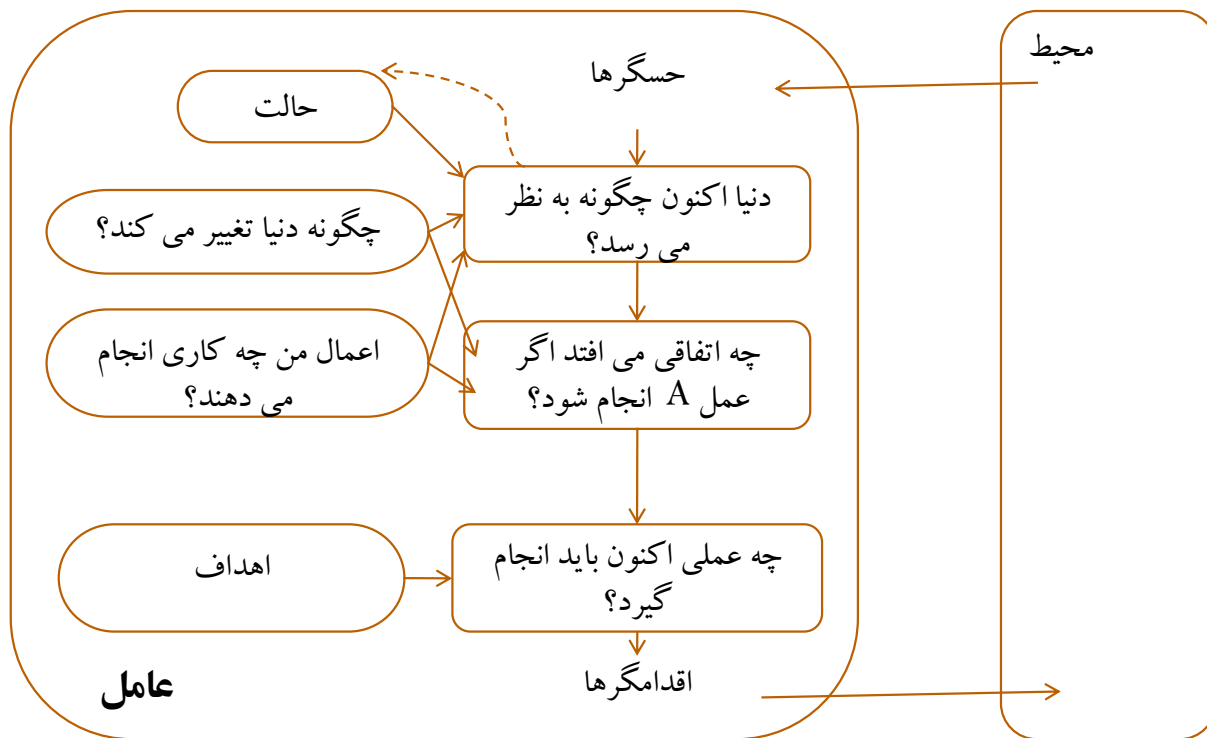
- دانستن حالت فعلی همیشه برای تصمیم گیری برای عمل فعلی کافی نیست.
- مثال: سرچهارراه می توان به راست، چپ یا مستقیم رفت.

هدف هم لازمه که مشخص بشه چه کاری باید انجام بشه
عامل انعطاف پذیرتر عاملی میشه که دارای هدف باشه

عاملی که ما رو به هدف می رسونه رو انتخاب میکنیم و اینطوری عامل انعطاف پذیرتر میشه

عامل هدف - مبنا

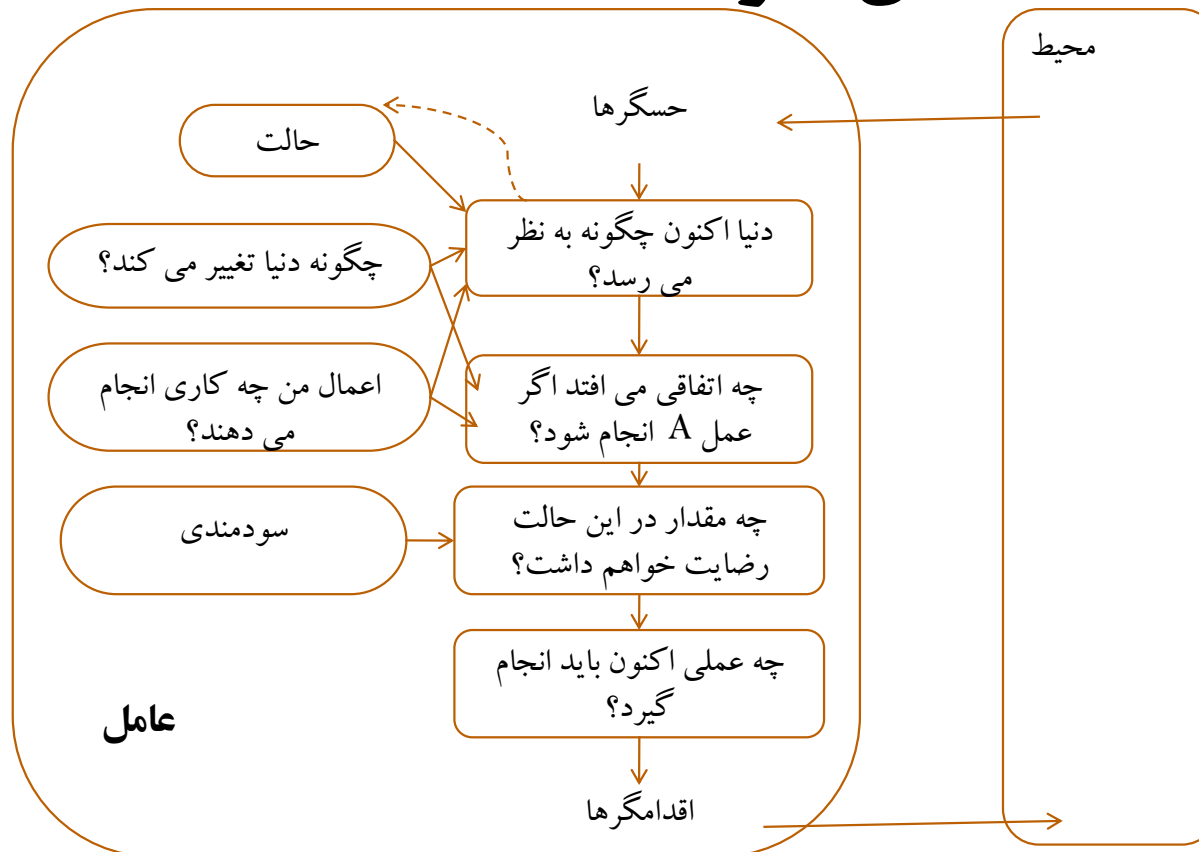
Goal-based agents



عامل هدف مبنا

- گاهی هدف با یک عمل بدست می آید
- گاهی با انجام یک سری اعمال
- استفاده از جستجو و طرحریزی برای یافتن اعمال
- رفتار عامل هدف مبنا می تواند تغییر داده شود تا به مقصد جدیدی برسد، قوانین عامل انعکاسی او را فقط به یک مقصد می رسانند.

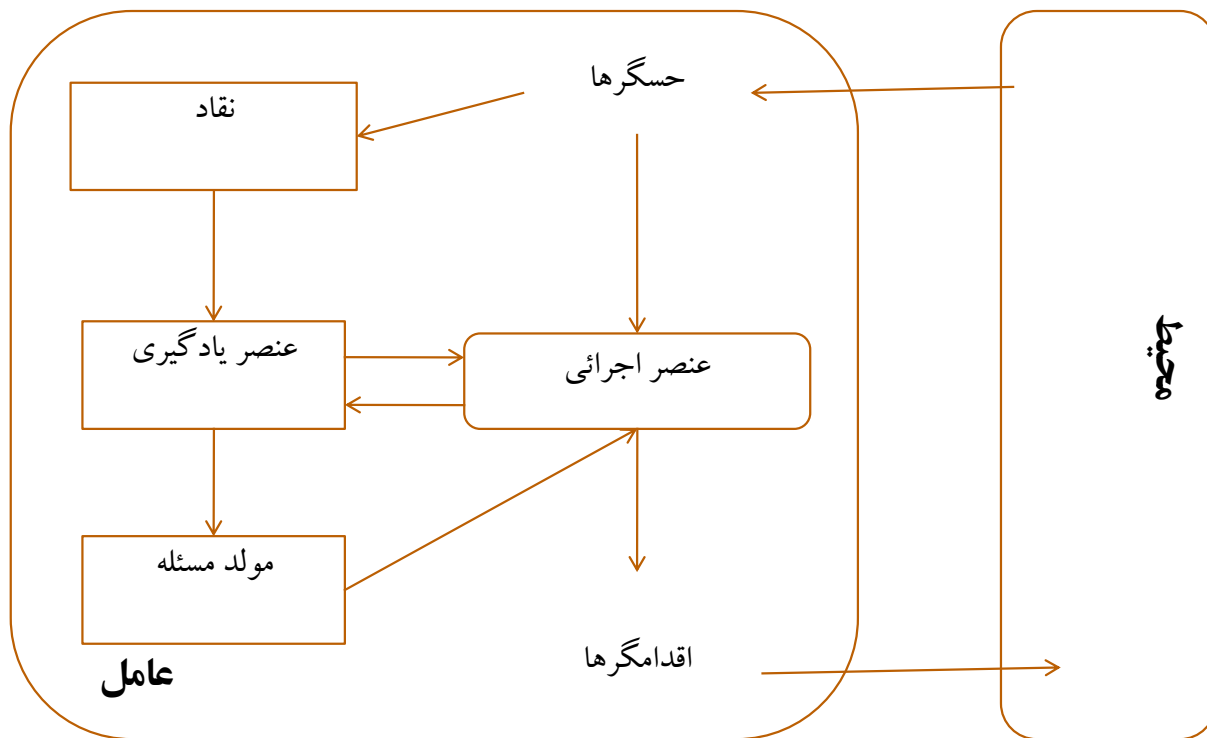
عامل سودمندی - مبنا



نقاد: بررسی میکنه میزان خوبی یا بدی اون عملی که انجام شده
عنصر یادگیری: کار این عنصر اینه که این عنصر اجرایی ما رو توش تغییر ایجاد میکنه و این رو اصلاح میکنه
مولد مسئله:

Learning agents

عامل یادگیر



خلاصه

■ ویژگیهای محیط

- مشاهده پذیر و نیم مشاهده پذیر، قطعی و غیرقطعی، مرحله ای و واقعه ای
- ایستا و پویا، گسسته و پیوسته، تک عاملی و چند عاملی
- شناخته و ناشناخته

■ ساختار عاملها

- عدم کارآئی استفاده از جدول

■ انواع عاملها

- انعکاسی ساده
- انعکاسی با مدل
- هدف مبنا
- سودمندی مبنا
- یادگیری



مازیار پالهنګ

هوش مصنوعی

■ تذکر مهم: پاورپوینت وسیله ای برای کمک به تدریس و یک
ارائه شفاهی می باشد و به هیچ وجه یک جزوه درسی نیست و
لازم است حتماً مرجع درس مطالعه شود.