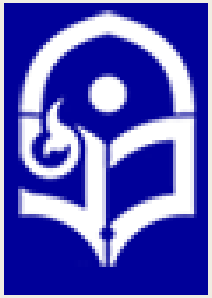
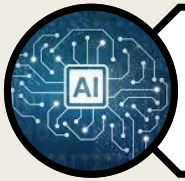


دانشگاه فرهنگیان
تبریز

هوش مصنوعی

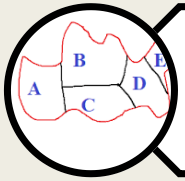
ابراهیم خلیل عباسی
ebrahim.Khalil.abbasi@gmail.com

کارگاه آموزشی
۱۳۹۹-۰۲-۲۹

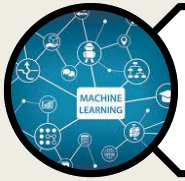


تعریف هوش مصنوعی

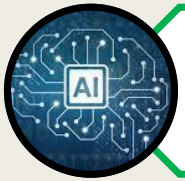
حل مساله با جستجو



مسائل ارضا محدودیت

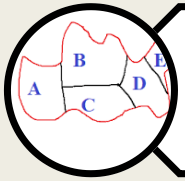


یادگیری ماشین

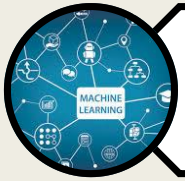


تعریف هوش مصنوعی

حل مساله با جستجو



مسائل ارضا محدودیت



یادگیری ماشین

هوش مصنوعی

سیستم هایی که عقلانی (منطقی) فکر می کنند	سیستم هایی که مانند انسان فکر می کنند
❑ مطالعه تواناییهای ذهنی از طریق به کارگیری مدل های محاسباتی (منطق گرایی). ❑ مطالعه محاسباتی که استدلال و عمل را امکان پذیر می سازد. ❑ سیستمی که با توجه آگاهی خود، کار درست انجام می دهد. ❑ قیاس منطقی معرفی شده توسط ارسطو: سقراط انسان است، تمامی انسان ها فانی هستند، پس سقراط فانی است. ❑ موانع اصلی: ❖ کارهای غیرارادی در قیاس منطقی قابل استنتاج نیستند. ❖ بیان موارد غیرقطعی در قیاس منطقی امکان پذیر نیست. مانند: اکثر انسان ها پر تلاش هستند. رضا انسان است. رضا پرتلاش است؟؟	❑ تلاش های جدید در الزام کامپیوترها به تفکر، به عبارت دیگر ماشین هایی دارای ذهن در معنای کامل کلمه ❑ خودکار نمودن فعالیت هایی که انسان ها آن ها را مرتبط با تفکر می دانند مانند تصمیم گیری، حل مساله، یادگیری، ... ❑ نیاز به روشی برای تعیین چگونگی تفکر در انسان: خودکاوی، روانکاوی، تصویربرداری مغز. ❑ هنگامی که به اندازه کافی نظریه ای (تئوری) از ذهن (mind) را در اختیار داشته باشیم آنگاه توصیف آن نظریه به عنوان برنامه کامپیوتری ممکن خواهد بود. هنوز به اندازه کافی شناخت از ذهن انسان وجود ندارد.
سیستم هایی که عقلانی (منطقی) عمل می کنند	**سیستم هایی که مانند انسان عمل می کنند**
❑ هوش محاسباتی عبارت است از مطالعه طراحی عامل های (agent) هوشمند. ❑ هوش مصنوعی مربوط به رفتار هوشمندانه در **مصنوعات** (artifact) است. ❑ سیستم هوشمند سیستمی است که دارای عملکرد عقلانی (منطقی) است. ❑ هنوز راه طولانی برای ساختن این عامل ها وجود دارد اما حاصل آن بسیار عالی خواهد بود.	❑ هنر ساخت ماشین هایی که اعمالی انجام می دهند که اگر این توابع توسط انسان انجام شوند نیازمند هوشمندی است. ❑ مطالعه چگونگی الزام کامپیوترها به انجام اعمالی که فعلا انسان ها آنها را بهتر انجام می دهند. ❑ قابلیتها: پردازش زبان طبیعی، استدلال خودکار، بینایی ماشین ❑ این نگرش به هوش مصنوعی بیش از حد پیچیده شده و ثانیا عدم شناخت کافی نسبت به خود انسان باعث شد تا دانشمندان نتوانند ابزاری بسازند که مانند انسان عمل کند.

تست تورینگ

- **هدف: یک سیستم چقدر هوشمند است و مانند انسان عمل می نماید؟**
- یک پرسشگر باید تشخیص دهد کدام یک انسان است و کدامیک کامپیوتر (سیستم هوشمند).
- پرسشگر برای تصمیم گیری و تشخیص، محدود به جواب سوالاتی است که به صورت نوشته در اختیار هر دو قرار داده شده است.
- کامپیوتر زمانی در آزمون موفق می شود که انسان پرسشگر بعد از طرح تعداد قابل قبولی پرسش، تشخیص این موضوع که آیا پاسخ ها از جانب یک انسان بوده است یا یک کامپیوتر، برای وی امکان پذیر نباشد. در این حالت گفته می شود که سیستم مورد نظر دارای عملکرد همانند انسان است.

(Alan Turing, 1950)

خردمندانه عمل کردن

روش عامل منطقی / تعقل گرا/ خردمندانه (The rational agent approach)

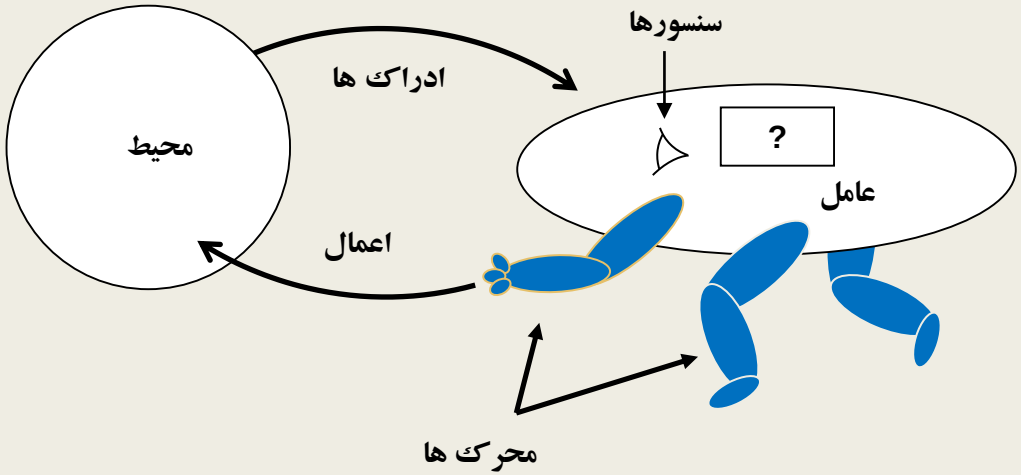
- **عامل (agent):** چیزی که عمل می نماید (act).

- تمامی برنامه های کامپیوتری عملی را انجام می دهند، اما از **عاملهای کامپیوتری هوشمند** انتظار کار بیشتری می رود:
  - به صورت خودمختار عمل نمایند (به غیر از راه اندازی اولیه و دانش اولیه به انسان نیاز ندارد)
  - محیط خود را درک نمایند (قابلیت حرکت، بینایی ماشین، حس؟؟)
  - برای یک مدت زمان طولانی عملیات خود را انجام دهند، (محدودیت مهم در انسان)
  - خود را با تغییرات وفق دهند، (قابلیت یادگیری ماشین)
  - اهداف را ایجاد و تعقیب نمایند. (به منظور کسب دانش جدید)

- از عامل خردمند انتظار می رود به نحوی عمل نماید که بهترین نتیجه را بدست آورد، و زمانی که قطعیت وجود ندارد، بهترین نتیجه مورد انتظار را بدست آورد (اگر هیچ کار خوبی بر اساس منطق وجود نداشته باشد اما باید کاری انجام شود)

- **برتری روش عامل-خردمند نسبت به سایر روشها:**
  - بسیار کلی تر و کامل تر از روش "قوانین تفکر" است زیرا استنتاج درست یکی از چندین مکانیزم ممکن برای حصول خردورزی(rationality)است.
  - توسعه علمی آن آسان تر از روش هایی است که بر مبنای رفتار و تفکر انسانی هستند. نسبت به تفکر و رفتار انسان، بسیار تابع پیشرفت های علمی است.
  - استاندارد و قالب خردورزی از نقطه نظر ریاضیات کاملا تعریف شده و به طور کلی کامل است و می تواند برای طراحی عامل ها مورد استفاده قرار دارد: تئوری روش های رسمی، تئوری کنترل، الگوریتم های اکتشافی برای مسائل NP-Complete

عامل انسانی – Agent



Intelligent Agent – عامل هوشمند

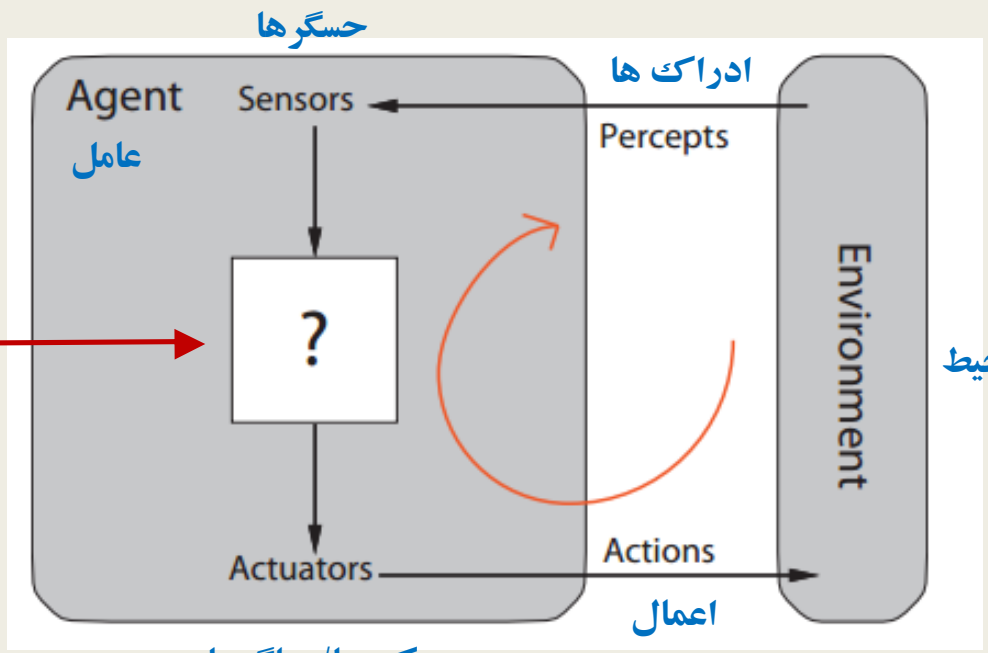
طراحی عامل هوشمند بر اساس عامل انسانی

بهترین و نزدیکترین الگوی قابل شناسایی توسط انسان به عنوان سازنده عامل هوشمند

در قبال این درک از محیط چه عملی انجام شود که خردمندانه/بهترین باشد؟

توسعه تابع عامل

انسان



نوع عامل	معیار کارایی	محیط	محرک ها	حسگرها
راننده تاکسی خودکار	ایمن، سریع، قانون، آرامش سفر، حداکثر نمودن سود	جاده ها، ترافیک، عابرین پیاده، مشتریان	فرمان، شتاب دهنده، ترمز، سیگنال، بوق، چراغ	دوربین ها، ماهواره، سرعت سنج، GPS، کیلومتر شمار، شتاب سنج، حسگرهای موتور، صفحه کلید، ...

تابع عامل

- **ادراک (Percept):** ورودی هایی که از طریق حسگرها به عامل می رسند.

- **دنباله ادراک (Percept Sequence):** تاریخچه کامل همه اطلاعاتی است که عامل تاکنون درک کرده است.
  - ❖ با مشخص نمودن گزینش عمل عامل برای هر دنباله ادراک ممکن، می توان ادعا کرد که کم و بیش همه چیز در مورد عامل گفته شده است.
  - ❖ لذا، می توان رفتار یک عامل را توسط تابع ریاضی بیان نمود.

از لحاظ ریاضی، رفتار عامل با استفاده از **تابع عامل** (agent function) توصیف می شود که هردنباله ادراک مفروض را به یک عمل نگاشت می نماید.

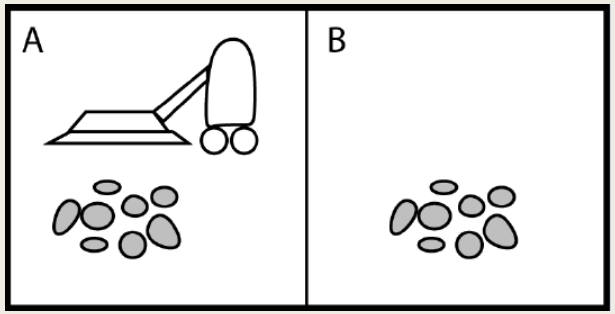
$$f : P^* \longrightarrow A \quad , f(p0, p1 \ldots pn) = a0$$

تابع عامل : دنباله ادراک ⟶ **عمل**

چالش مهم نحوه پیاده سازی تابع عامل است

تابع عامل - عامل هوشمند برای جاروبرقی

- **ادراک:** عامل جاروبرقی می تواند مکان خود و همچنین این که در آن مکان آشغال وجود دارد یا نه را درک کند. مانند: [A,Dirty]
- **اعمال:** عامل جاروبرقی میتواند به چپ حرکت کند، به راست حرکت کند، آشغال را جمع نماید و یا هیچ کاری نکند.



جاروبرقی با دو مکان برای جاروکردن

Percept sequence	Action
[A, Clean]	*Right*
[A, Dirty]	*Suck*
[B, Clean]	*Left*
[B, Dirty]	*Suck*
[A, Clean], [A, Clean]	*Right*
[A, Clean], [A, Dirty]	*Suck*
.	:
[A, Clean], [A, Clean], [A, Clean]	*Right*
[A, Clean], [A, Clean], [A, Dirty]	*Suck*
:	.

جدول ادراک-عمل برای پیاده سازی تابع عامل

```
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
```

برنامه برای پیاده سازی تابع عامل

رفتار خوب – مفهوم عقلانیت (Rationality)

- **عامل خردمند/منطقی/عاقل:** عاملی که باید تحت شرایط خاصی کار **درست/بهترین** کار را انجام دهد.
  - **از لحاظ مفهومی**
    - تمامی ورودی ها (رکوردهای) جدول به درستی پر شده اند.
    - برنامه نوشته شده به ازای تمامی ادراک ها، **درست ترین** عمل را برمی گرداند.

- **مفهوم کار درست:**
  - یک عامل در یک محیط بر اساس ادراک های خود دنباله ای از اعمال را انجام می دهد.
  - این دنباله ای از اعمال منجر می شود که محیط در وضعیت های / حالت های (state) مختلف قرار گیرد.
  - اگر دنباله وضعیت های سیستم مطلوب باشد، نتیجه می گیریم که عامل کار خود را خوب انجام داده است.
  - مثال: راننده خودکار در یک جاده لغزنده و برفی در بین دو ماشین در حال حرکت است. اگر ماشین جلویی ترمز نماید،

راننده خودکار باید سرعت خود را کم کند و یا ترمز نماید؟

از یک وضعیت مطلوب در حال حرکت برسیم به یک وضعیت مطلوبی که راننده خودکار با ماشین جلویی تصادف نکرده و برای ماشین عقبی هم مشکل ایجاد نکرده است.

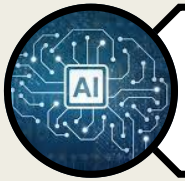
- **معیار کارایی:** برای اندازه گیری میزان موفقیت عامل، نیاز به یک معیار داریم
  - داشتن کف اتاق تمیز در عامل جاروبرقی
  - سالم به مقصد رساندن مسافر توسط عامل راننده خودکار

ساختار عامل های هوشمند

- وظیفه هوش مصنوعی طراحی یک برنامه عامل (agent program) است که تابع عامل – نگاشت از دنباله ادراک به عمل– را پیاده سازی نماید.
- فرض بر این است که این برنامه برروی نوعی وسیله محاسباتی با سنسورها و محرک های فیزیکی – که معماری نامیده می شود– اجرا خواهد شد.

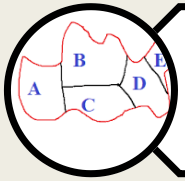
agent = architecture + program

- **برنامه های عامل دارای ساختار مشابهی هستند**: ادراک جاری را به عنوان ورودی از سنسورها می گیرد و عمل را به عنوان خروجی به محرک ها برمی گرداند.
- **برای هر دامنه کاربرد خاص نیاز است برنامه عامل متناسب نوشته شود:**
  - استفاده از درخت جستجو برای پیدا نمودن بهترین راه حل (عمل) از بین اعمال موجود
  - انتساب مقادیر مجاز به متغیرها در سیستم های پارامتریک برای پیدا نمودن تنها راه حل (عمل) موجود

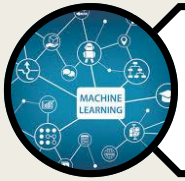


تعریف هوش مصنوعی

حل مساله با جستجو



مسائل ارضا محدودیت



یادگیری ماشین

برنامه عامل مبتنی بر جدول جستجو

Percept sequence	Action
[A, Clean]	*Right*
[A, Dirty]	*Suck*
[B, Clean]	*Left*
[l3, Dirty]	*Suck*
[A, Clean], [A, Clean]	*Right*
[A, Clean], [A, Dirty]	*Suck*
.	:
[A, Clean], [A, Clean], [A, Clean]	*Right*
[A, Clean], [A, Clean], [A, Dirty]	*Suck*
:	.

- ایجاد آن زمانبر است.
- بسیار بزرگ خواهد بود (مثلا در شطرنج ۱۰۱۵۰ سطر)
- قابلیت یادگیری ندارد.

به جای جدول یک برنامه نوشته شود.

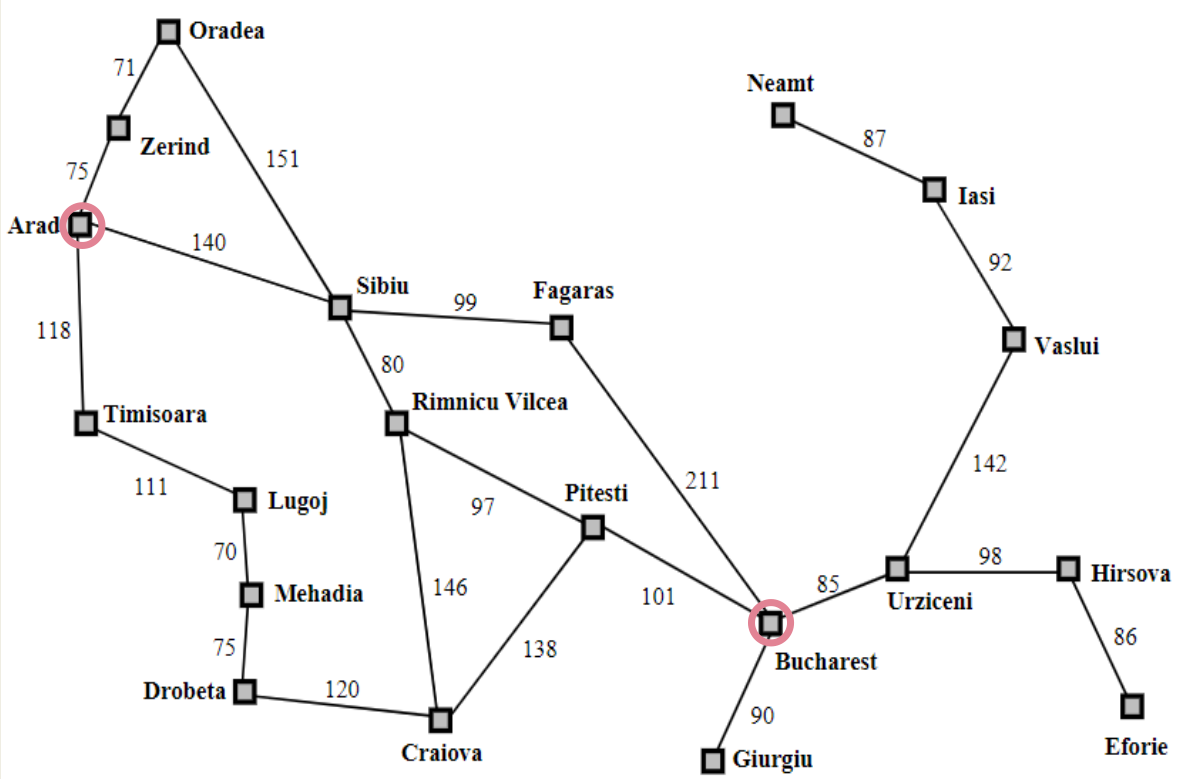
```
function REFLEX-VACCUM-AGENT( [location, status]) returns an action

    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

برای هر مساله/مشکل الگوریتم به عنوان راه حل آن پیاده سازی می شود.

حل مساله با جستجو - یک راه حل کلاسیک برای توسعه برنامه عامل

- عامل هوشمند با پیدا نمودن یک رشته از اعمال که به حالت های مطلوب منجر می شوند تصمیم می گیرند که چه کاری را انجام دهند.



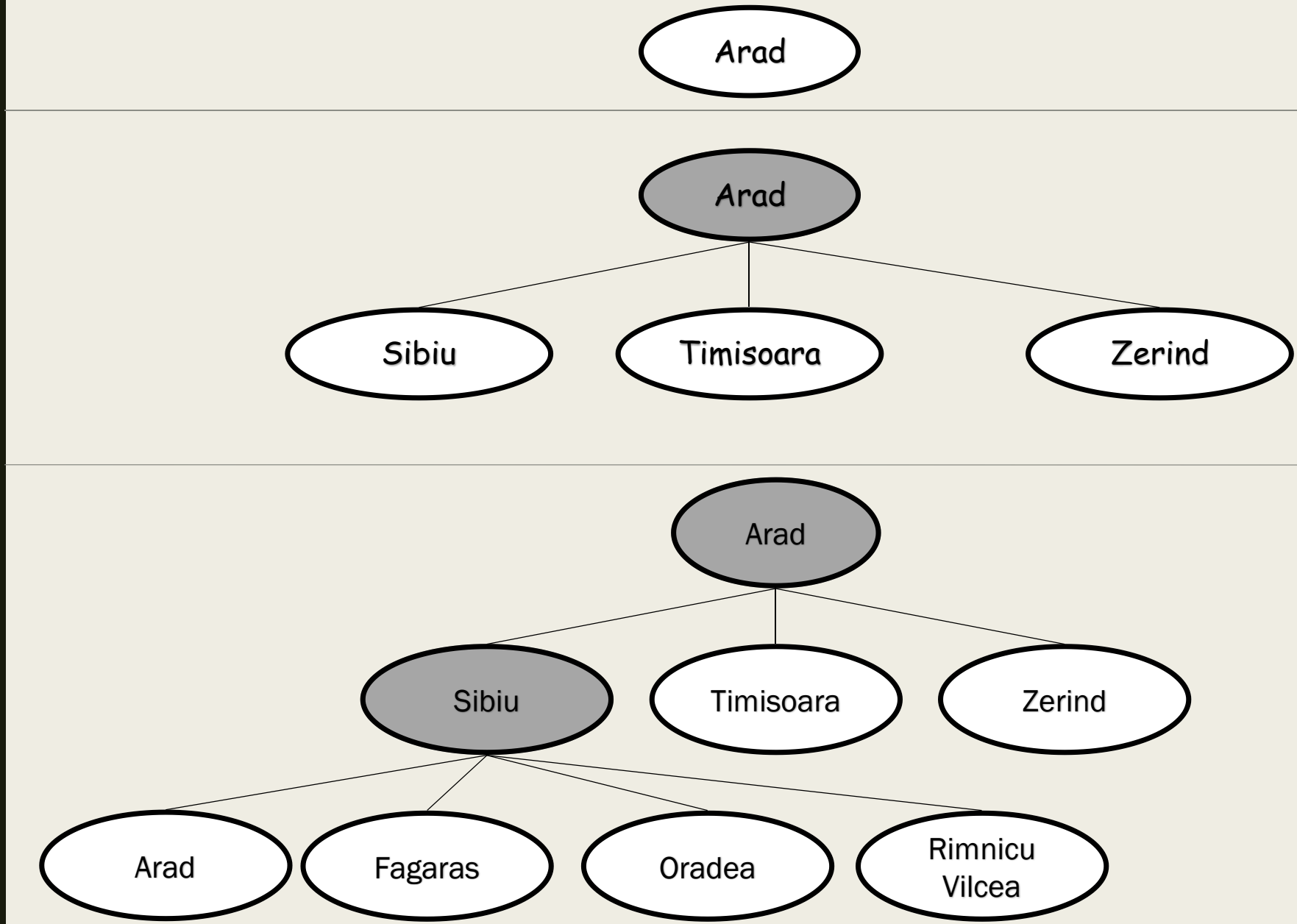
- **صورت مسأله**: رفتن از Arad به Bucharest
- **فرموله کردن هدف**: رسیدن به Bucharest
- **فرموله کردن مساله:**
  - ❖ **حالت ها**: شهرهای مختلف
  - ❖ **اعمال**: حرکت بین شهرها
- **جستجو برای یافتن بهترین پاسخ (مجموعه ای از اعمال):**

دنباله ای از شهرها مثل:Arad--> Sibiu --> Fagras --> Bucharest

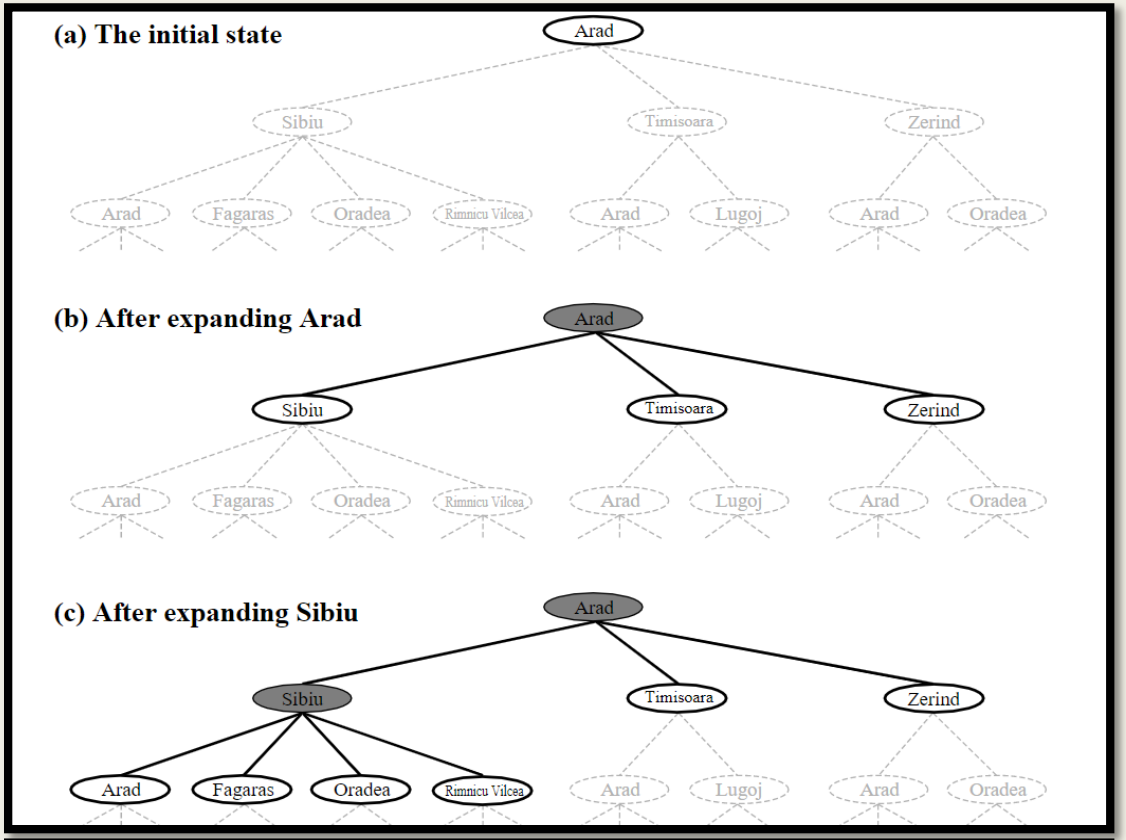
❖یک راه حل (solution) عبارت است از یک دنباله از اعمال، لذا الگوریتم های جستجو با در نظر گرفتن چندین دنباله ممکن از اعمال کار می کنند.

الگوریتم جستجو یک درخت جستجو ایجاد می نماید.

الگوریتم های جستجو - پیاده سازی با درخت جستجو



Breadth-First Search (BFS) - جستجوی اول سطح/عرض




```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  frontier ← a FIFO queue with node as the only element
  explored ← an empty set
  loop do
      if EMPTY?(frontier) then return failure
      node ← POP(frontier)   /* chooses the shallowest node in frontier */
      add node.STATE to explored
      for each action in problem.ACTIONS(node.STATE) do
          child ← CHILD-NODE(problem, node, action)
          if child.STATE is not in explored or frontier then do
              if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
              frontier ← INSERT(child, frontier)
```

- الگوریتم های مختلف روش های متفاوتی برای ایجاد درخت جستجو دارند.

Breadth-First Search (BFS) - جستجوی اول سطح/عرض

Depth	Nodes	Time	Memory
2	110	.11 milliseconds	107 kilobytes
4	11110	11 milliseconds	10.6 megabytes
6	10^6	1.1 seconds	1 gigabytes
8	10^8	2 minutes	103 gigabytes
10	10^{10}	3 hours	10 terabytes
12	10^{12}	13 days	1 petabytes
14	10^{14}	3.5 years	99 petabytes
16	10^{16}	350 years	10 exabytes

- ضریب انشعاب b برابر ۱۰ است.
- تعداد ۱ میلیون گره در ثانیه قابل تولید است.
- هر گره نیازمند ۱۰۰۰بایت فضای ذخیره سازی است.

الگوریتم و اکتشاف (Algorithm vs. Heuristics)

- **الگوریتم** به صورت مجموعه ای از قوانین و یا مراحل مرتب تعریف می شود که برای حل یک مساله خاص استفاده می شود.
- یک الگوریتم (در صورت وجود) اگر به نمونه های از مسائلی که در یک دسته قرار دارند اعمال شود برای آنها راه حل خواهد داشت.
- مثال: الگوریتم اجرای اولویت عملگرها در یک برنامه کامپیوتری

❖ 3 + 9 * (1 / 6 + 9)

❖ 9 * 6 – 7 / (1 + 9)

- **راه حل اکتشافی** روالی است که تجربه و روش سعی–خطا را برای حل مساله استفاده می نماید.
- یک راه حل اکتشافی (در صورت وجود) اگر به نمونه های از مسائلی که در یک دسته قرار دارند اعمال شود ممکن است برای برخی راه حل نداشته باشد.
- مثال: روال تشخیص کدهای با عملکرد مشابه (Clone Code Detection) با تحلیل لغوی

❖
```
for (int i=0; i < 10; i++)
    print(i);
```
```
for (int k=0; k < 10; k++ )
    print(k);
```

❖
```
for (int i=0; i < 10; i++)
    print(i);
```
```
i = 0 ;  while (i<10) {
    print (i);  i++; }
```

?

Informed (Heuristic) Search Strategies - جستجوی آگاهانه (اکتشافی)

- از یک دانش ویژه یک مساله هوش مصنوعی فراتر از تعریف مساله بهره می گیرد.

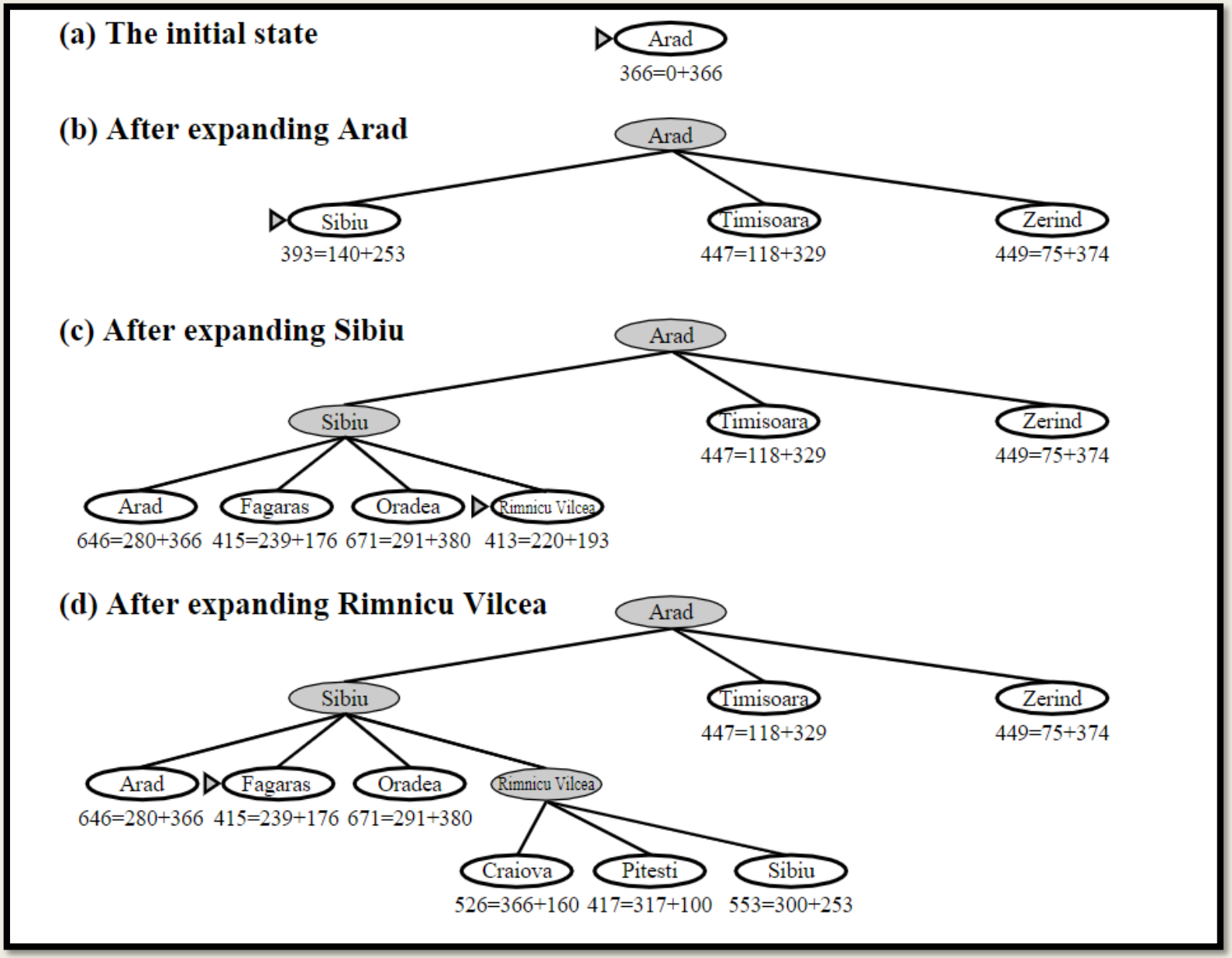
$$f(n) = g(n) + h(n)$$ تابع ارزیابی

- **تابع اکتشافی:** $h(n)$ هزینه تخمینی ارزان ترین مسیر از گره n به یک حالت نهایی
- **تابع واقعی:** $g(n)$ هزینه واقعی (فاصله) بین دو گره
- گره با کمترین مقدار $f(n)$ برای گسترش انتخاب می شود.

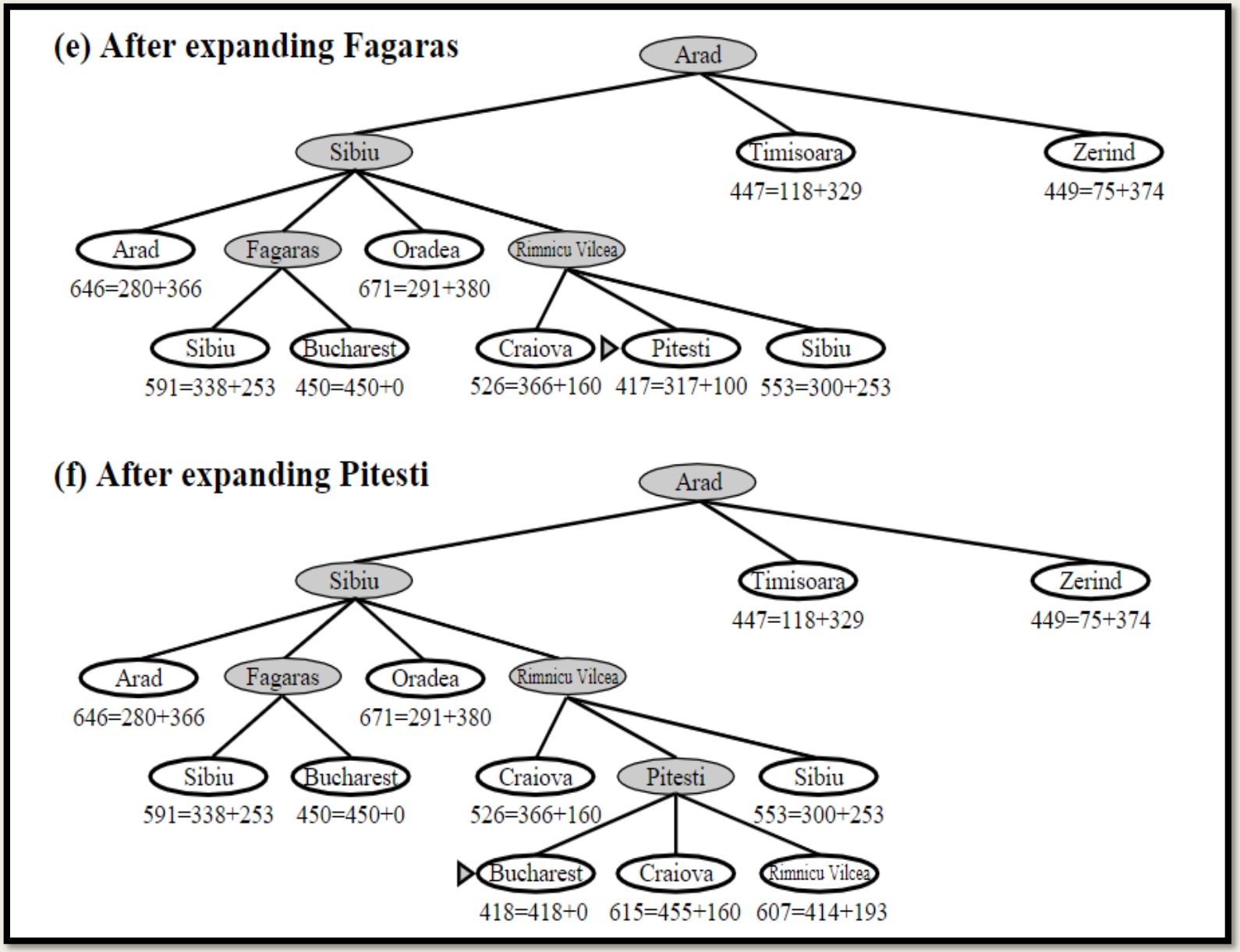
Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

h(n)

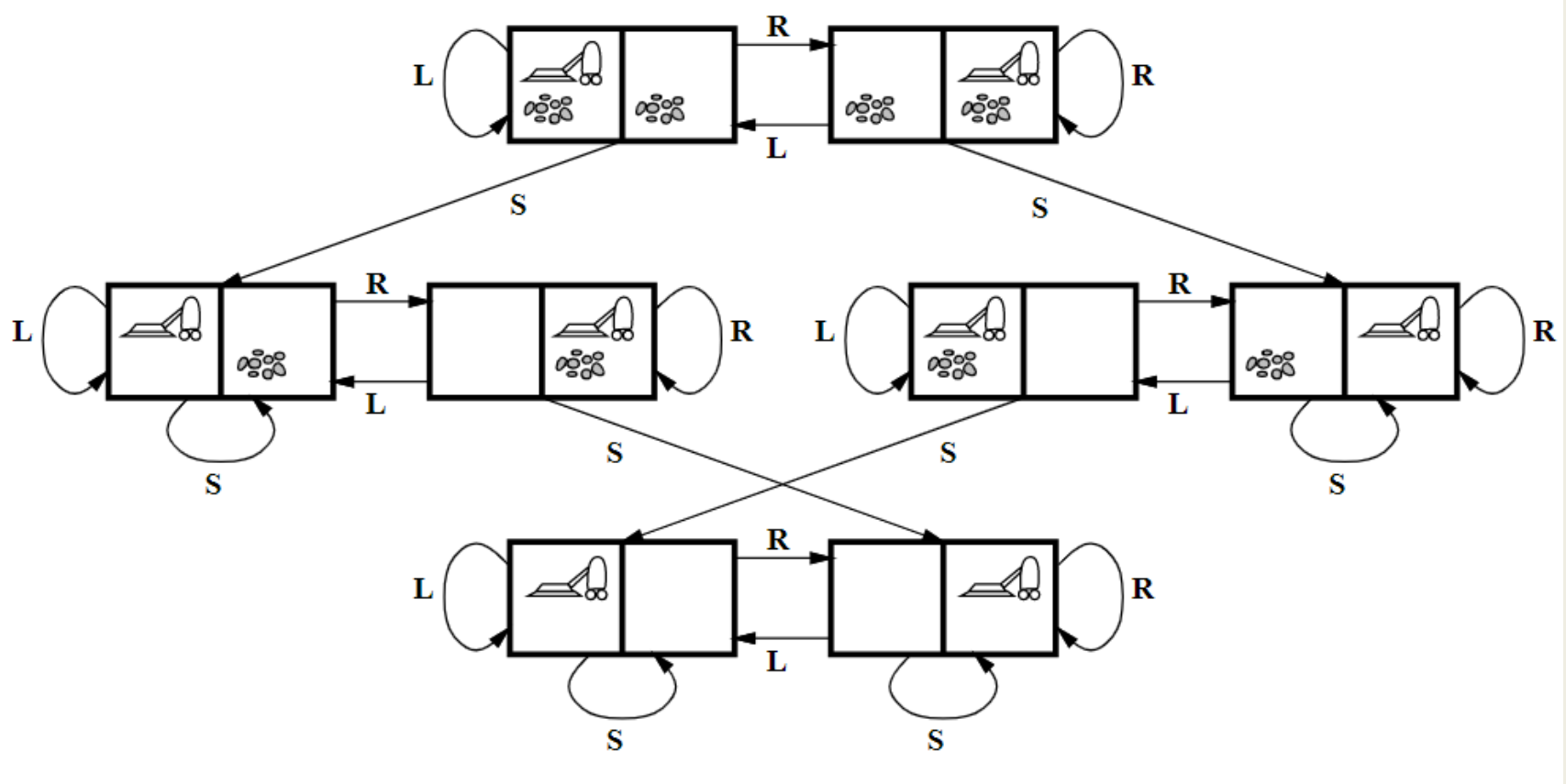
A* search: Minimizing the total estimated solution cost



A* search: Minimizing the total estimated solution cost



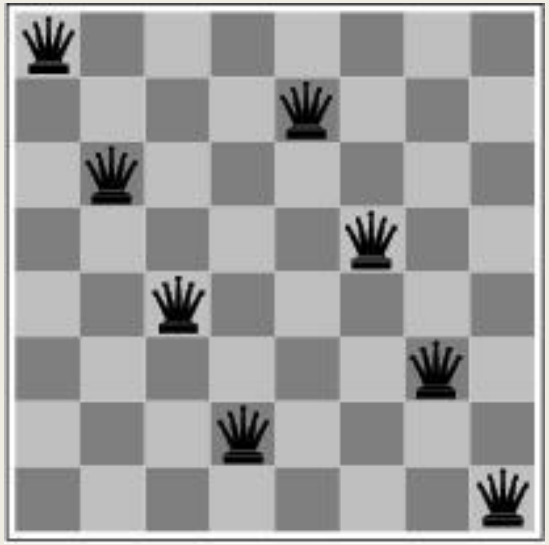
درخت جستجو برای مساله عامل هوشمند جاروبرقی



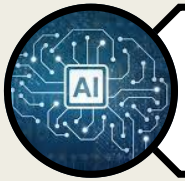
۸-وزیر (8-Queens)

هدف قرار دادن ۸ وزیر در یک صفحه است که هیچ یک از آنها دیگری را تهدید نکنند.

- دو نوع فرموله سازی وجود دارد:
  - **حالت کامل** (complete-state formulation) با ۸ وزیر برروی صفحه شروع می شود و آنها را جابجا می نماید.
  - **افزایشی** (incremental formulation): با شروع با یک حالت خالی، هر عمل یک وزیر را به حالت اضافه می نماید.

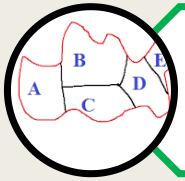


- **حالت ها:** هر ترتیبی از ۰ تا ۸ وزیر در صفحه یک حالت است.
- **حالت اولیه:** هیچ وزیری برروی صفحه نیست.
- **اعمال:** افزودن یک وزیر به یک خانه خالی
- **مدل انتقال:** صفحه ای که با افزودن یک وزیر به خانه مشخص شده بدست می آید.
- **آزمون هدف:** ۸ وزیر برروی صفحه قرار دارند و هیچ کدام همدیگر را تهدید نمی کنند.
- در این فرموله بندی، 64*63*62* ... * 57 = 1.8 * 10^14 دنباله ممکن برای بررسی وجود دارد. یک فرموله بندی بهینه عبارت است ازپرهیز از قراردادن یک وزیر در خانه ای است که تهدید می شود.
  - ***حالت ها :*** *تمامی ترتیب های از n وزیر (0<=n<=8) ، یک وزیر مستقر در هر ستون در n ستون سمت چپ که هیچ وزیری دیگری را تهدید نمی کند.*
  - ***اعمال:*** *افزودن یک وزیر به خانه در ستون خالی چپ به گونه ای که مورد تهاجم دیگر وزیران قرار نگیرد.*
  - *این فرموله بندی فضای حالت را به ۲۰۵۷ کاهش می دهد و راه حل ها براحتی پیدا می شوند. اما هنوز تعداد حالات زیاد است.*

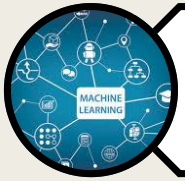


تعریف هوش مصنوعی

حل مساله با جستجو



مسائل ارضا محدودیت



یادگیری ماشین

Constraint Satisfaction Problem (CSP)

مسائل CSP

- هر مساله شامل سه عنصر X,D, C به شرح زیر است:

  - X مجموعه متناهی از متغیرها $X_1,X_2 , \dots ,X_n$
  - D دامنه های نامتناهی برای هر یک از متغیرها $D_1,D_2 , \dots ,D_n$
  - C مجموعه ای متناهی از محدودیت ها

- هر محدودیت C_i زیرمجموعه ای از متغیرها و ترکیبهای ممکنی از مقادیر برای آن زیرمجموعه ها است.

- هر حالت با **انتساب** مقادیری به چند یا تمام متغیرها تعریف میشود.

- انتسابی که هیچ محدودیتی را نقض نکند، انتساب **سازگار** نام دارد.

- انتساب **کامل** آن است که هر متغیری در آن باشد.

- **راه حل** CSP یک انتساب کامل است اگر تمام محدودیتها را برآورده کند.

مساله رنگ آمیزی نقشه

- X = {WA, NT, Q, NSW, V, SA, T}
- D = {red, green, blue}
- **محدودیت ها**: دو منطقه مجاور همرنگ نیستند.
- چون ۹ مکان هم مرز وجود دارد، لذا ۹ محدودیت وجود دارد:

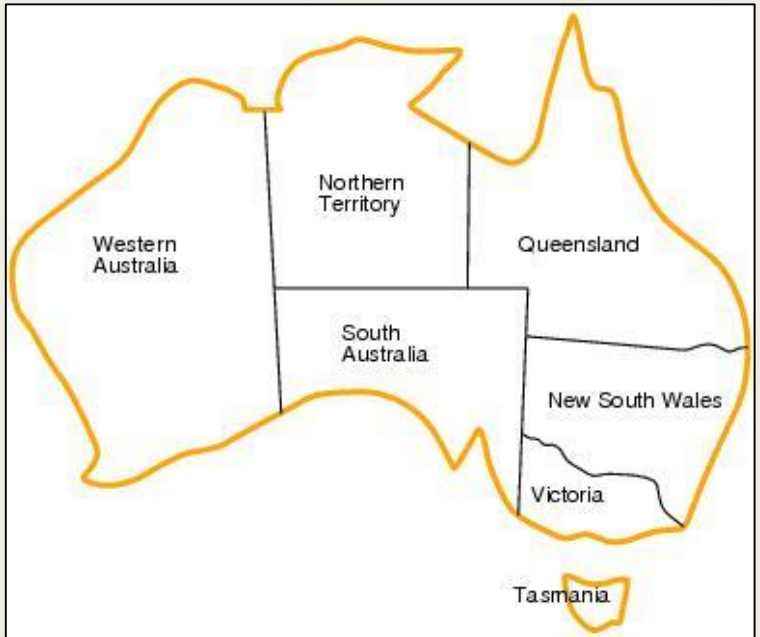
C = {SA≠WA, SA≠NT, SA ≠Q, SA ≠NSW, SA ≠V, WA ≠NT, NT ≠Q, Q ≠NSW, NSW ≠V}

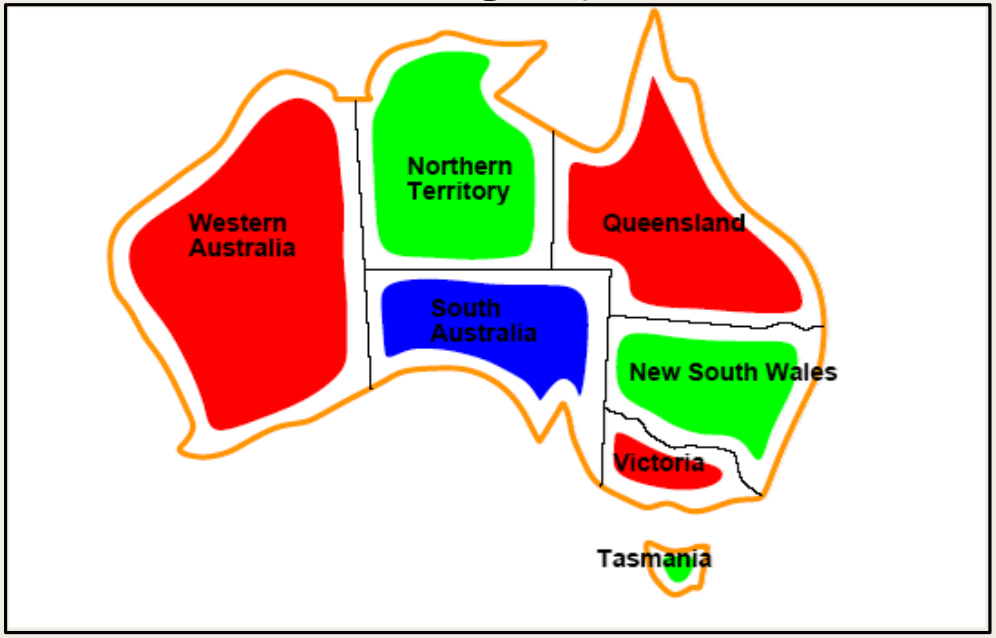
- SA≠WA را می توان به شکل زیر نوشت:

{(red, green), (red, blue), (green, red), (green, blue), (blue, red), (blue green)}

- تعداد زیادی راه حل وجود دارد. به عنوان نمونه

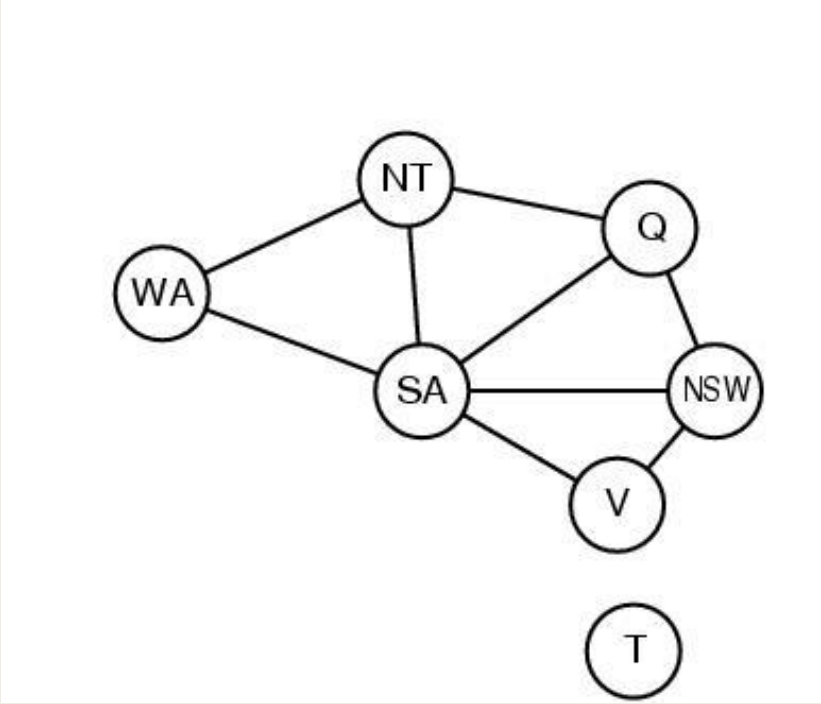
{WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green}





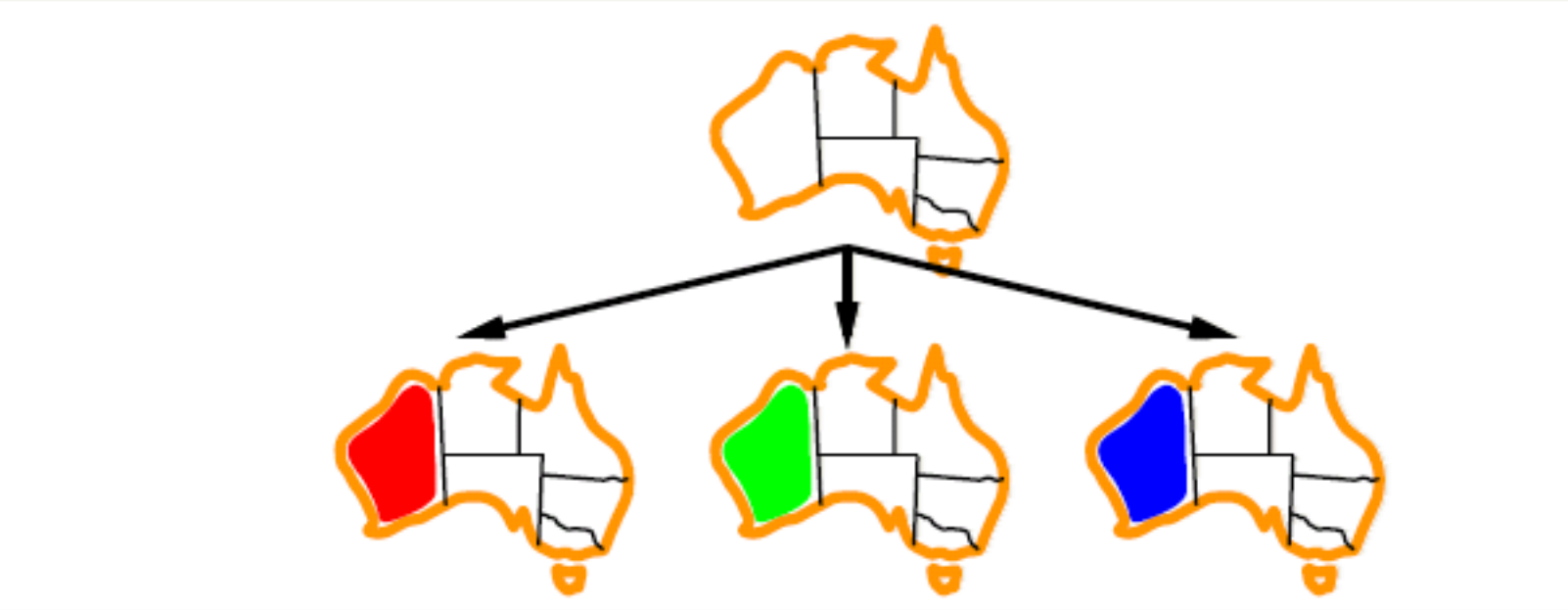
گراف محدودیت

- به منظور فهم بهتر، یک مساله CSP را می توان به صورت یک گراف محدودیت – Constraint Graph – نشان داد.

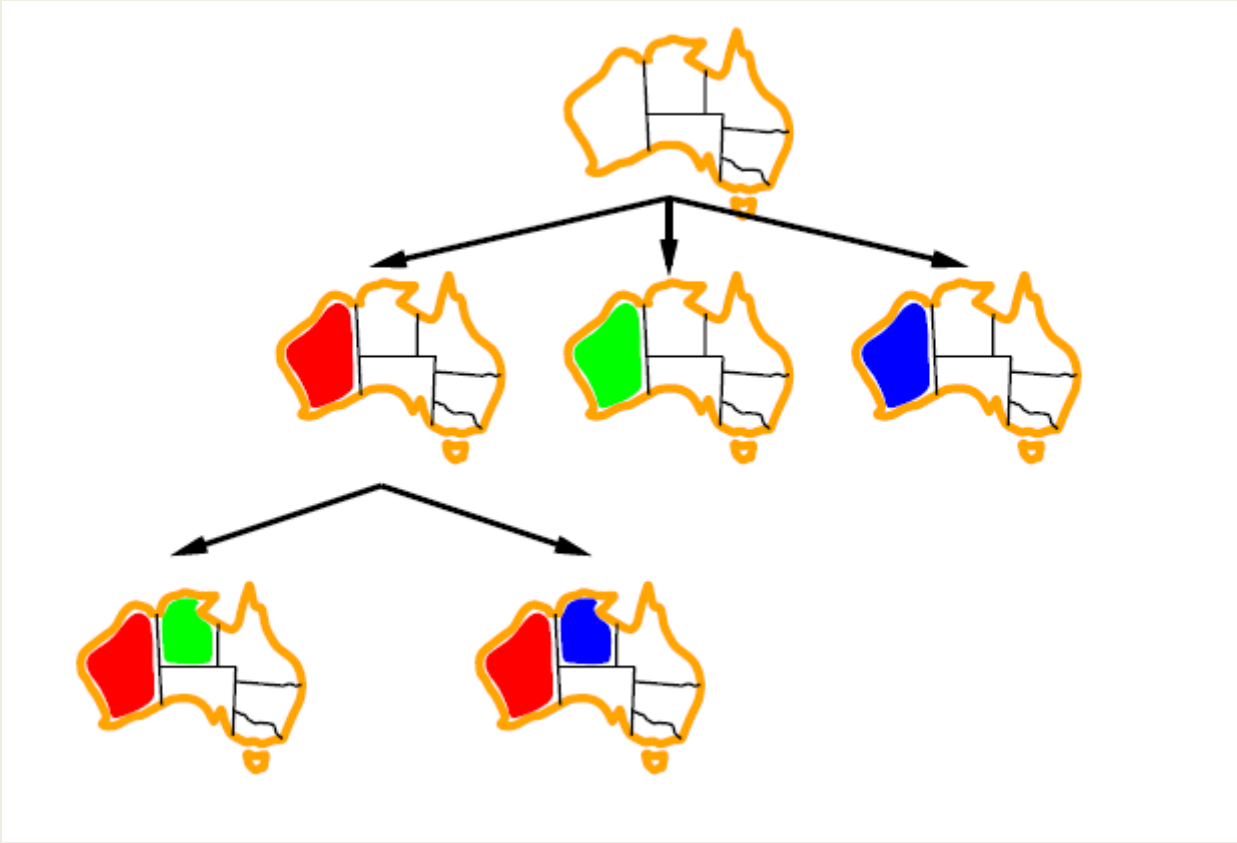


- در گراف محدودیت:
  - **گره ها**: متغیرها
  - **یالها**: محدودیتها

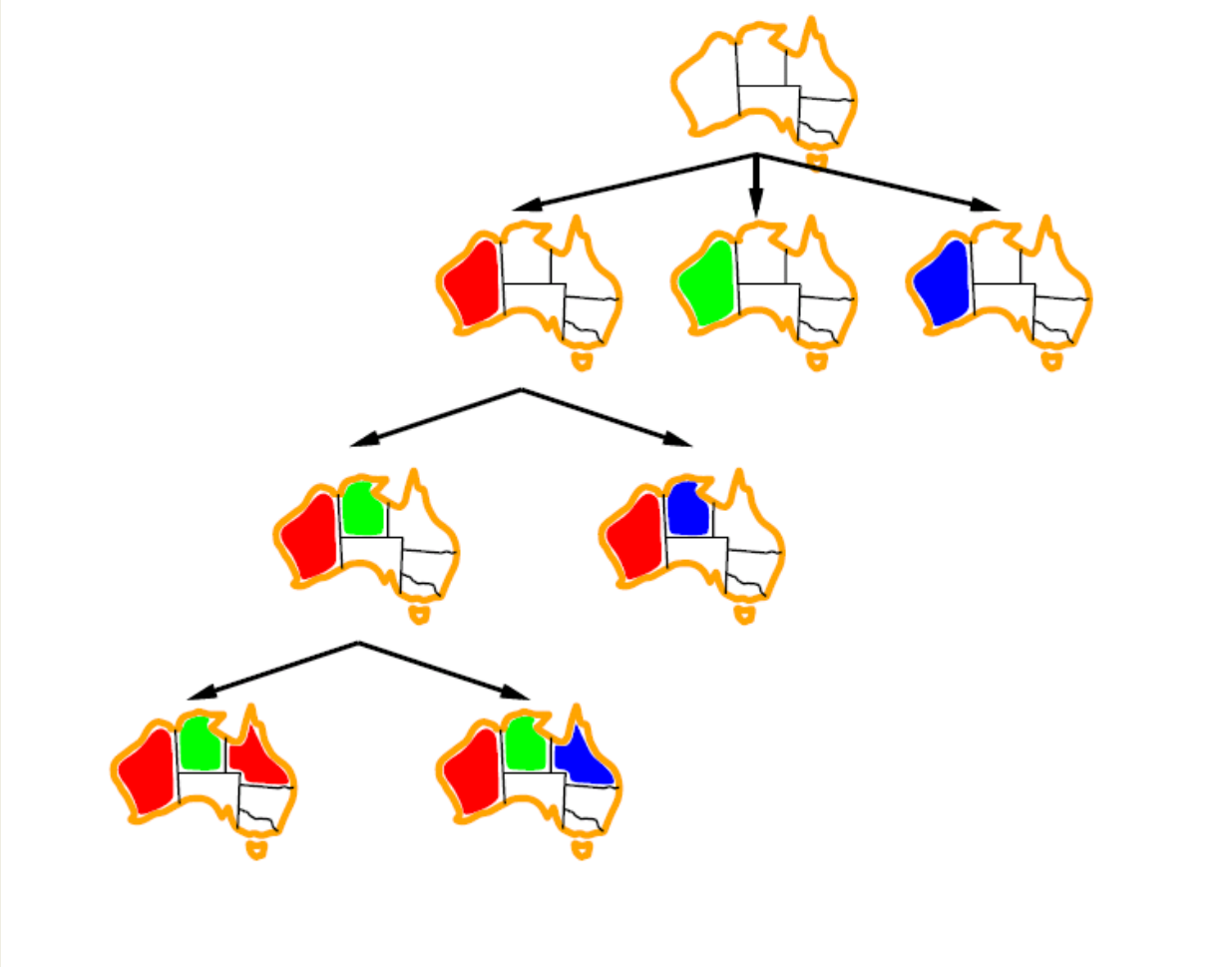
جستجوی عقبگرد برای حل مساله رنگ آمیزی (Backtracking)

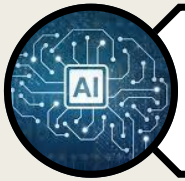


جستجوی عقبگرد برای حل مساله رنگ آمیزی (Backtracking)



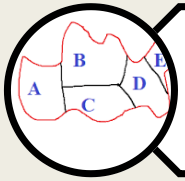
جستجوی عقبگرد برای حل مساله رنگ آمیزی (Backtracking)





تعریف هوش مصنوعی

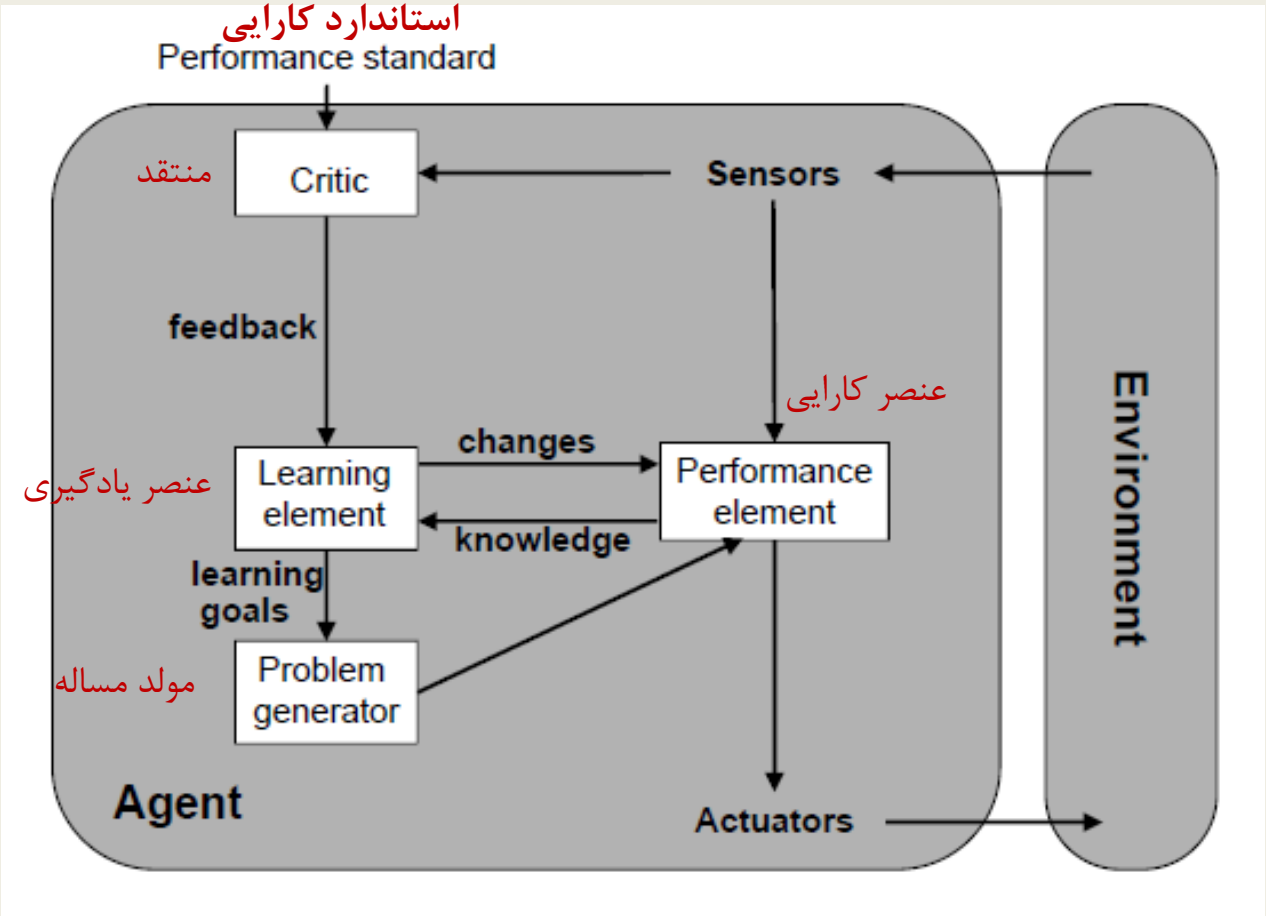
حل مساله با جستجو



مسائل ارضا محدودیت

یادگیری ماشین

عامل یادگیرنده



عامل یادگیرنده

یادگیری ماشین

- **سخت بودن نوشتن برنامه هایی برای حل مسائلی همچون تشخیص چهره**
  - مشخص نبودن نحوه عملکرد مغز انسان برای نوشتن برنامه آن
  - حتی اگر ایده خوبی برای پیاده سازی آن وجود داشته باشد، برنامه ای که برای آن باید نوشت، بسیار پیچیده خواهد بود.

- قصد داریم تا به جای نوشتن دستی یک برنامه، تعداد زیادی نمونه که در آن خروجی درست به ازای هر ورودی مشخص شده است را استفاده نماییم.

- یک الگوریتم یادگیری ماشین به کمک این نمونه ها، برنامه ای را تولید می کند که کار مورد نظر ما را انجام دهد.

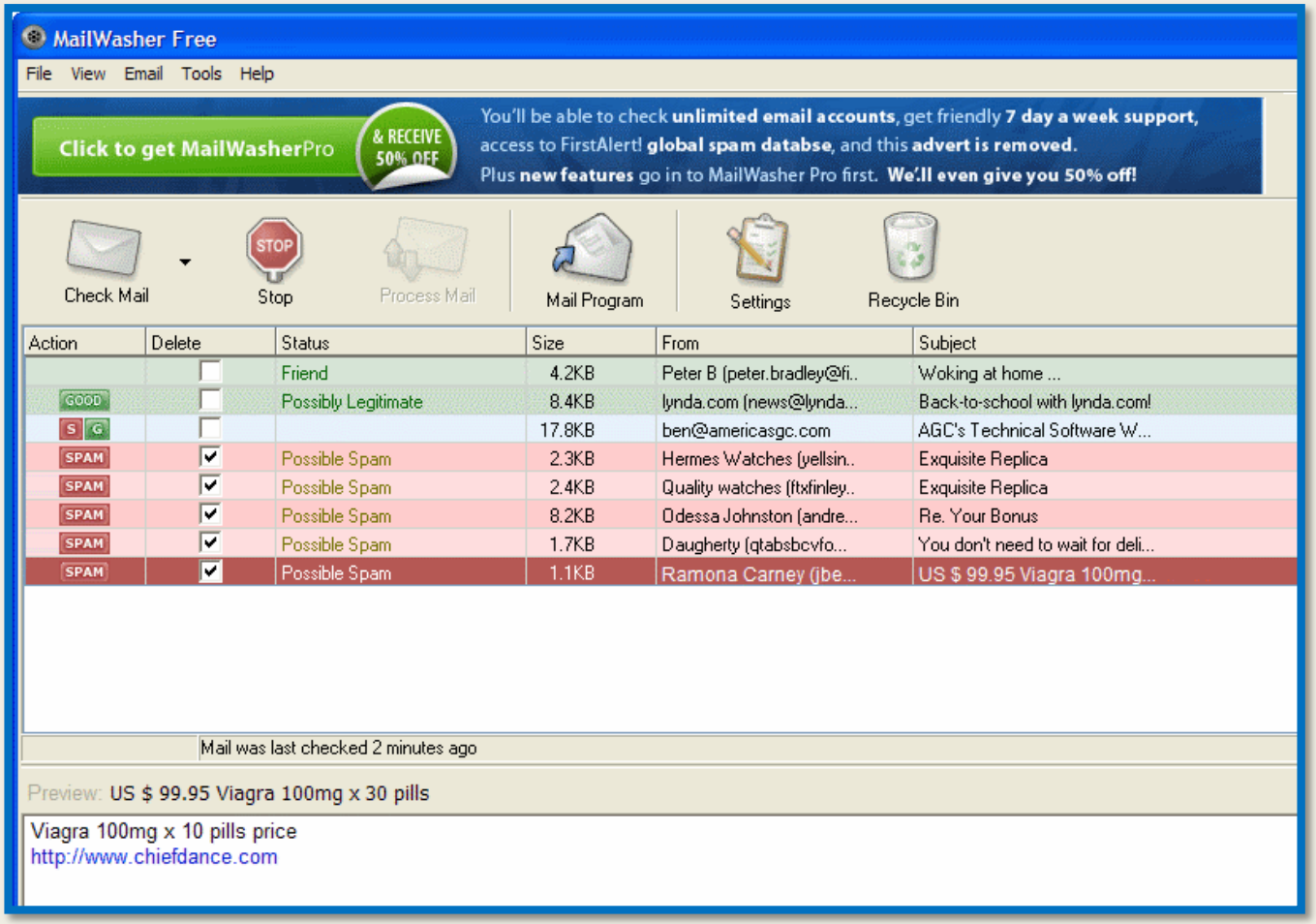
  - برنامه تولید شده توسط الگوریتم یادگیری بسیار متفاوت از برنامه های متداول دستی خواهد بود. این برنامه ممکن است از میلیون ها عدد درست شده باشد.
  - اگر این کار به درستی انجام شود، برنامه تولید شده می تواند برای نمونه های جدید به خوبی نمونه هایی که بر روی آن آموزش داده شده است، عمل نماید (تعمیم).

- در واقع در این روش ما تنها ساختار برنامه را مشخص می نماییم و کامپیوتر پارامترهای فراوان آن را برای ما مشخص می نماید

تعریف یادگیری ماشین

- **تعریف ۱:** بررسی سیستم هایی که می توانند از روی داده، آموزش ببینند.

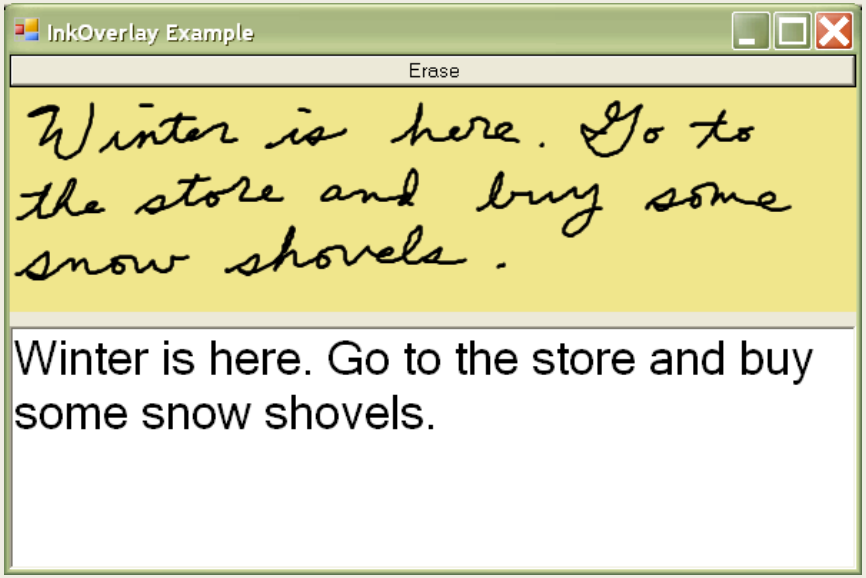
- **مثال:** یک سیستم میتواند بر روی پیام ایمیلها به نحوی آموزش ببیند که بتواند هرزنامه (اسپم) بودن و یا نبودن یک پیغام ایمیل را تشخیص دهد.

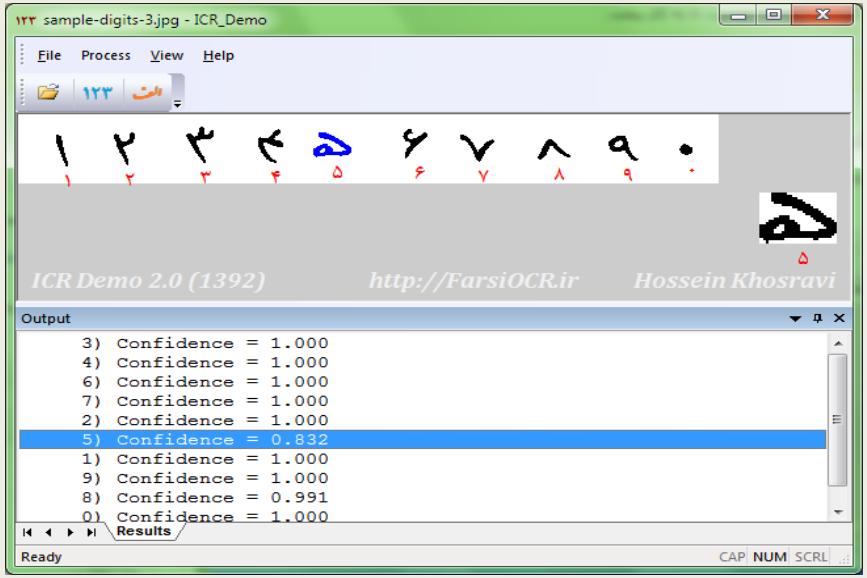


تعریف یادگیری ماشین

- **تعریف ۲:** وقتی می توان گفت که یک برنامه از روی تجربیات E برای کاربرد Tو معیار کارآیی P آموزش دیده است که کارآیی آن در کاربرد مورد نظر در T که با P مشخص شده، با استفاده از تجربیات E بالا برود.

- **مثال: مساله یادگیری برای تشخیص دست نوشته**
  - T :تشخیص و دسته بندی کلمات دستنویس
  - P : درصد کلمات درست دسته بندی شده
  - E :مجموعه داده های کلمات دستنویس



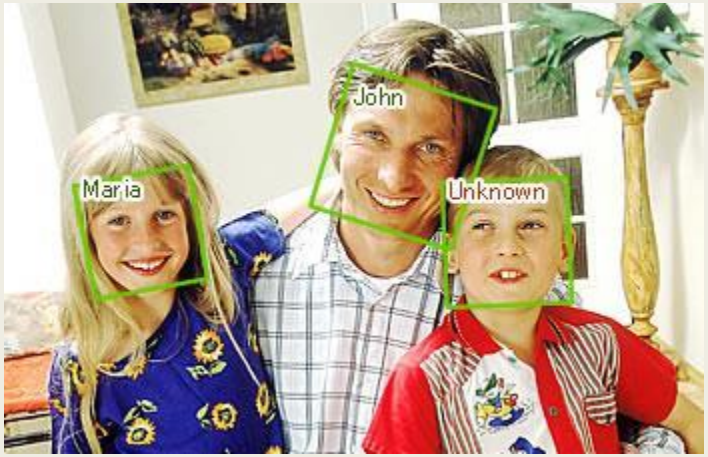


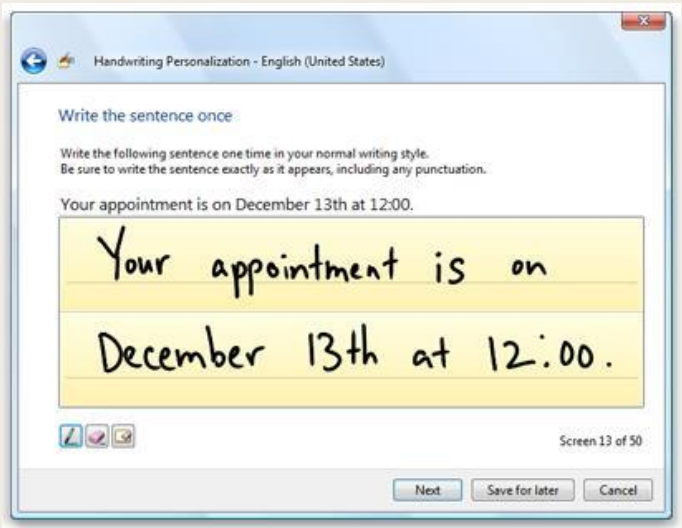
کاربردها

- **تشخیص ارقام دستنویس:** در این مساله تعدادی نمونه آموزشی از ارقام دستنویس به همراه برچسب آنها داده شده است و باید یک سیستم یادگیر طوری آموزش ببیند که در صورت دیدن یک نمونه جدید، بتواند آن را به درستی دسته بندی نماید.

کاربردها

- **تشخیص الگو:**
  - تشخیص چهره و یا حالت چهره
  - تشخیص دست نوشته و یا کلمات گفته شده
  - پردازش تصاویر پزشکی





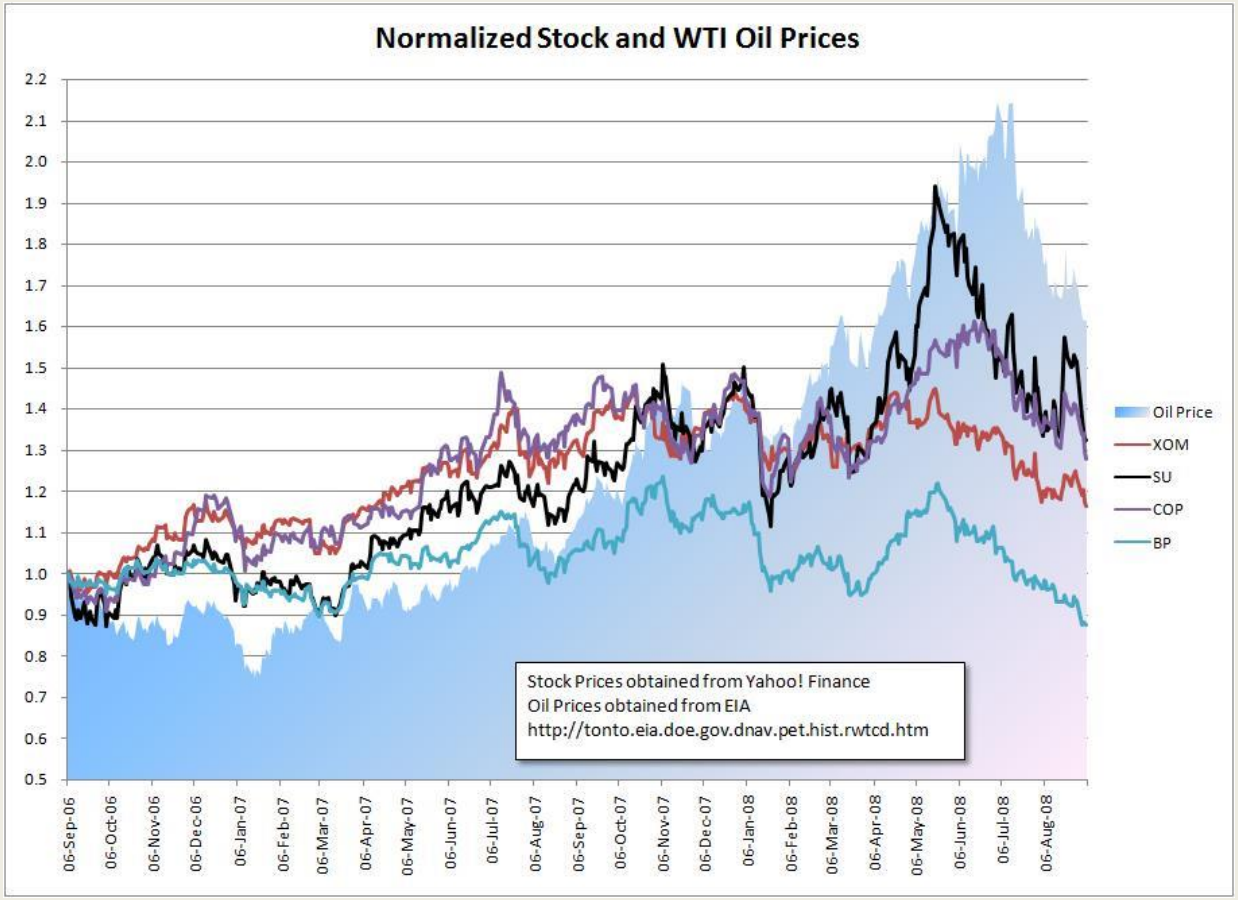
کاربردها

- **تصاویر دیجیتال و ویدئو:**
  - موقعیت یابی، رهگیری و تشخیص اشیا
  - رانندگی با ماشین

کاربردها

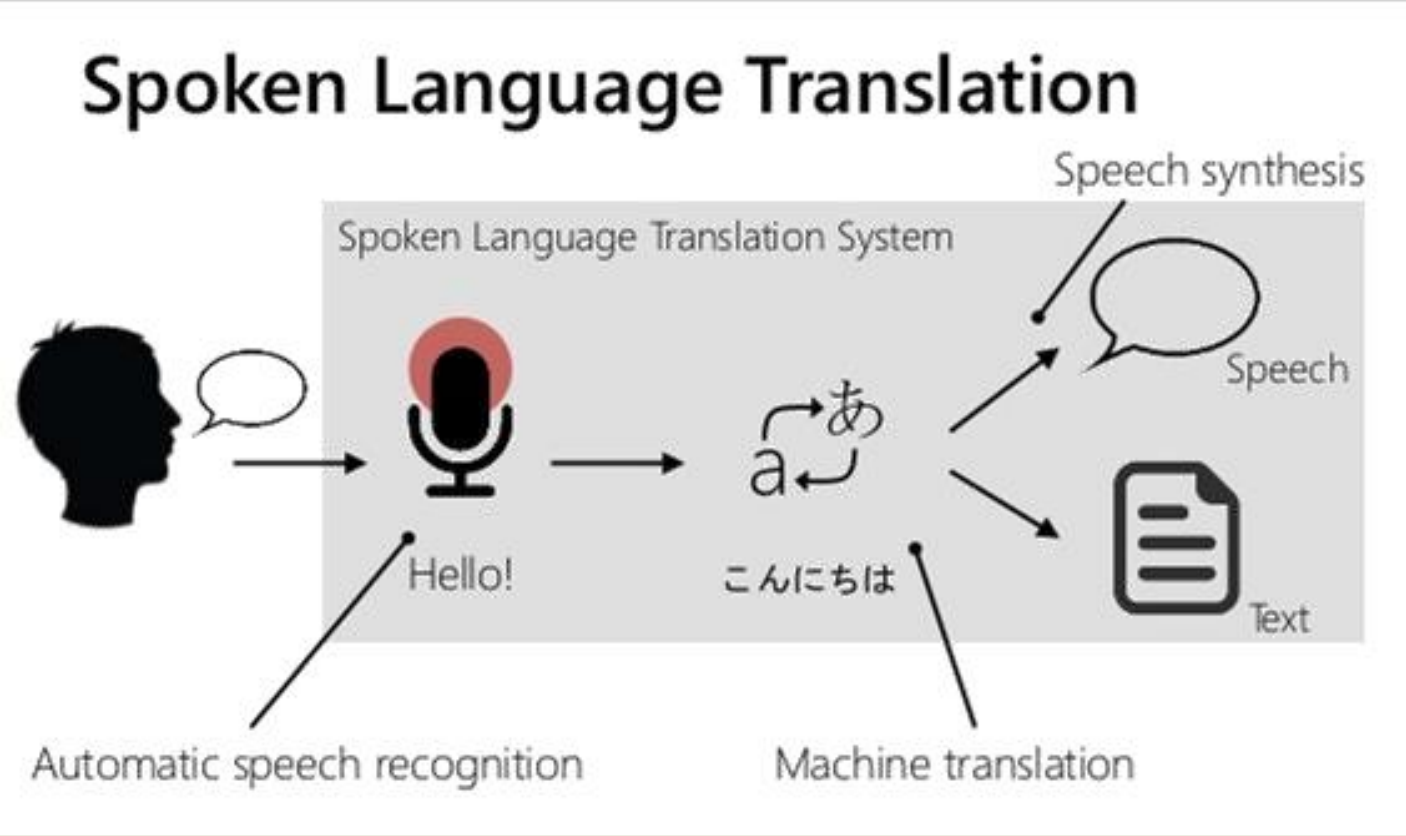
- **تشخیص ناهنجاری:**
  - توالی غیر معمول از معاملات کارت های اعتباری
  - الگوی غیر معمول در صدای موتور ماشین

- **پیشگویی:**
  - قسمت سهام و یا نرخ ارز در آینده



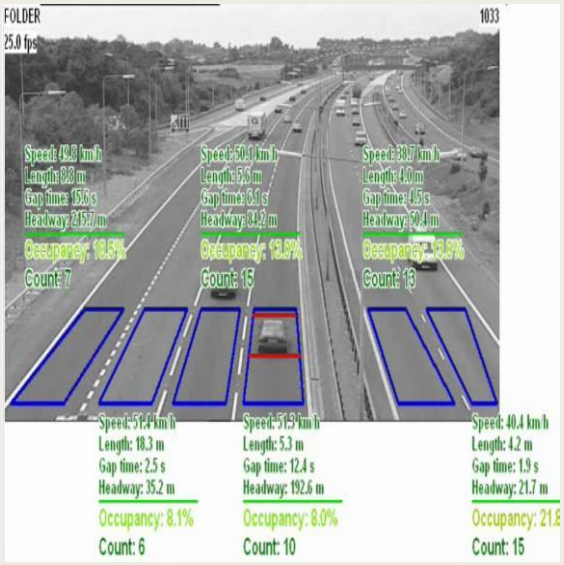
کاربردها

- تشخیص گفتار و ترجمه ماشینی:



کاربردها

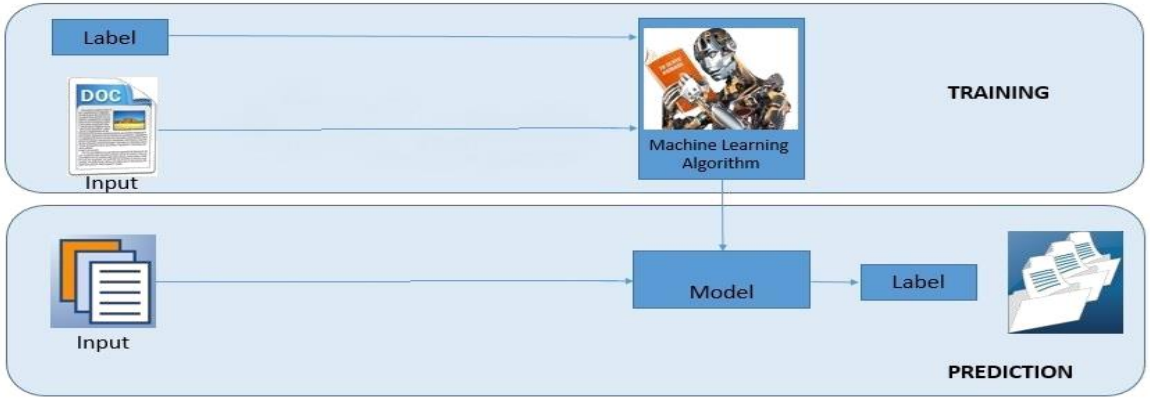
Boston Dynamics


FOLDER
1033
25.0 fps
Speed: 49.8 km/h
Length: 3.8 m
Gap time: 15.6 s
Headway: 245.7 m
Occupancy: 18.5%
Count: 7
Speed: 50.1 km/h
Length: 5.6 m
Gap time: 8.1 s
Headway: 84.2 m
Occupancy: 13.8%
Count: 15
Speed: 38.7 km/h
Length: 4.0 m
Gap time: 4.5 s
Headway: 50.4 m
Occupancy: 13.8%
Count: 13
Speed: 51.4 km/h
Length: 18.3 m
Gap time: 2.5 s
Headway: 35.2 m
Occupancy: 8.1%
Count: 6
Speed: 51.9 km/h
Length: 5.3 m
Gap time: 12.4 s
Headway: 192.6 m
Occupancy: 8.0%
Count: 10
Speed: 40.4 km/h
Length: 4.2 m
Gap time: 1.9 s
Headway: 21.7 m
Occupancy: 21.8
Count: 15

NRJ 22A
Building Maintenance

روش های یادگیری ماشین

- **یادگیری با نظارت (Supervised Learning):**
  - آن به یک سیستم، مجموعه‌ای از جفتهای ورودی-خروجی ارائه شده و سیستم تلاش میکند تا تابعی از ورودی به خروجی را فرا گیرد. بدین ترتیب این نوع یادگیری، تعدادی نمونه ورودی را به همراه خروجی مطلوب گرفته و سیستمی را میسازد تا بتواند خروجی را برای ورودیهای جدید تخمین بزند.



Data Set

Label

0
1
2
3
4
5
6
7
8

فاز آموزش

Input

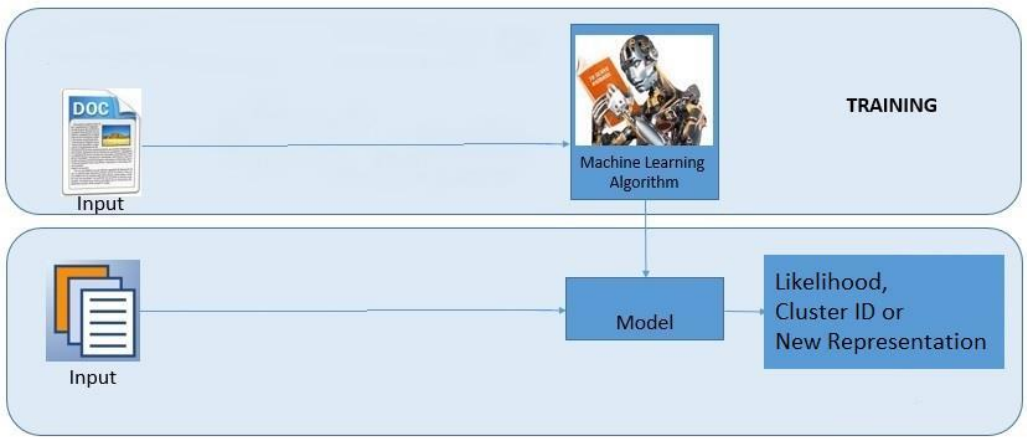
Output

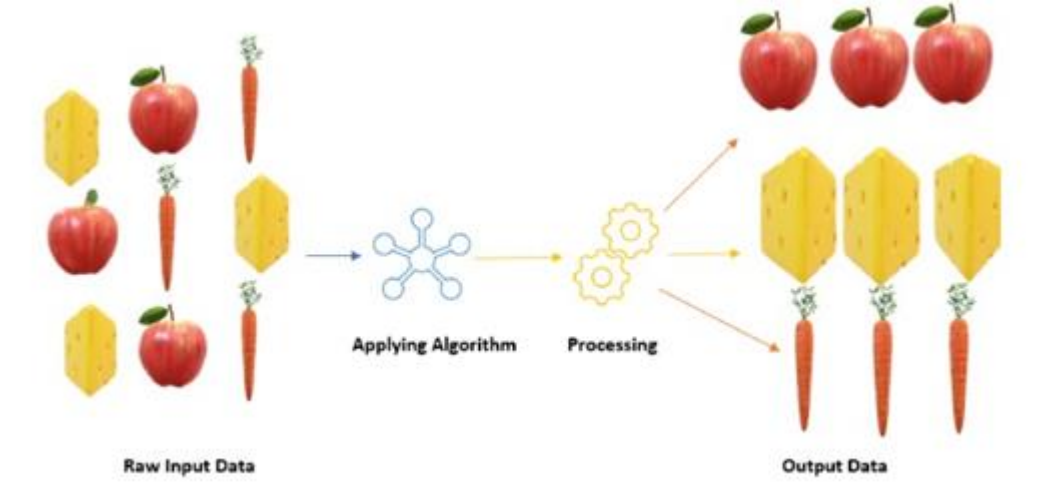
7

فاز پیش بینی

روش های یادگیری ماشین

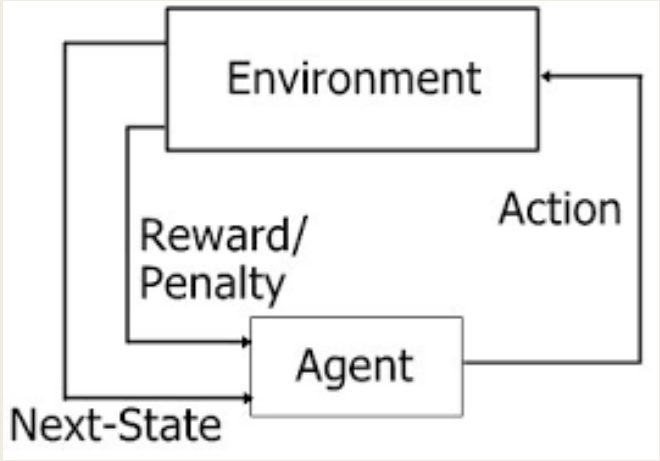
- **یادگیری بدون نظارت (Unsupervised Learning):**
  - یک مجموعه از مثالهای یادگیری وجود دارد که در آن فقط مقدار ورودیها مشخص است و اطلاعاتی در مورد خروجی صحیح در دست نیست. این نوع از یادگیری میتواند برای ساخت یک بازنمایی جدید از ورودی مورد استفاده قرار بگیرد.





روش های یادگیری ماشین

- **یادگیری تقویتی (Reinforcement Learning):**
  - در این روش یادگیری، عامل هوشمند (Agent) با توجه به وضعیتی که در محیط دارد، عملی را بر روی محیط انجام میدهد و منتظر نتیجه عملش میماند. این نتیجه میتواند در قالب یک پاداش یا تنبیه باشد. اگر نتیجه در قالب پاداش باشد، عمل انجام شده مطلوب بوده و عامل به هدفی که در آن محیط دارد نزدیک شده است. ولی اگر نتیجه در قالب تنبیه باشد، عمل انجام شده نامطلوب بوده و عامل از هدفش دور شده است. عامل باید یاد بگیرید که چه اعمالی را انجام دهد تا پاداش بیشتری را کسب کند و در نهایت به هدفش برسد.

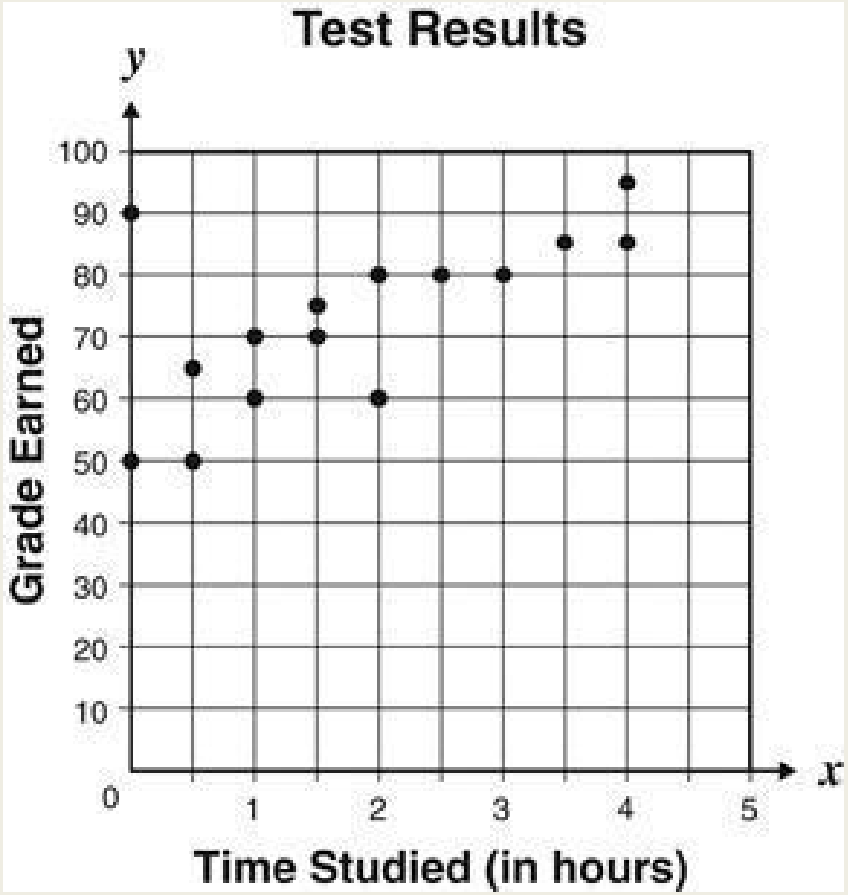


- همه ما در کودکی با الگویی مشابه یادگیری تقویتی راه رفتن را آموختیم. زمانی که پس از چندین گام برداشتن به زمین میخوردیم(**تنبیه**،) سعی میکردیم اعمال حرکتی خود را به گونه ای اصلاح کنیم تا تعادل خود را به هنگام راه رفتن حفظ کنیم (**پاداش**.) در نهایت هم به هدف خود که راه رفتن بود رسیدیم.

رگرسیون – یک الگوریتم یادگیری ماشین

- **مثال : دادگان تاثیر مطالعه در نمره**

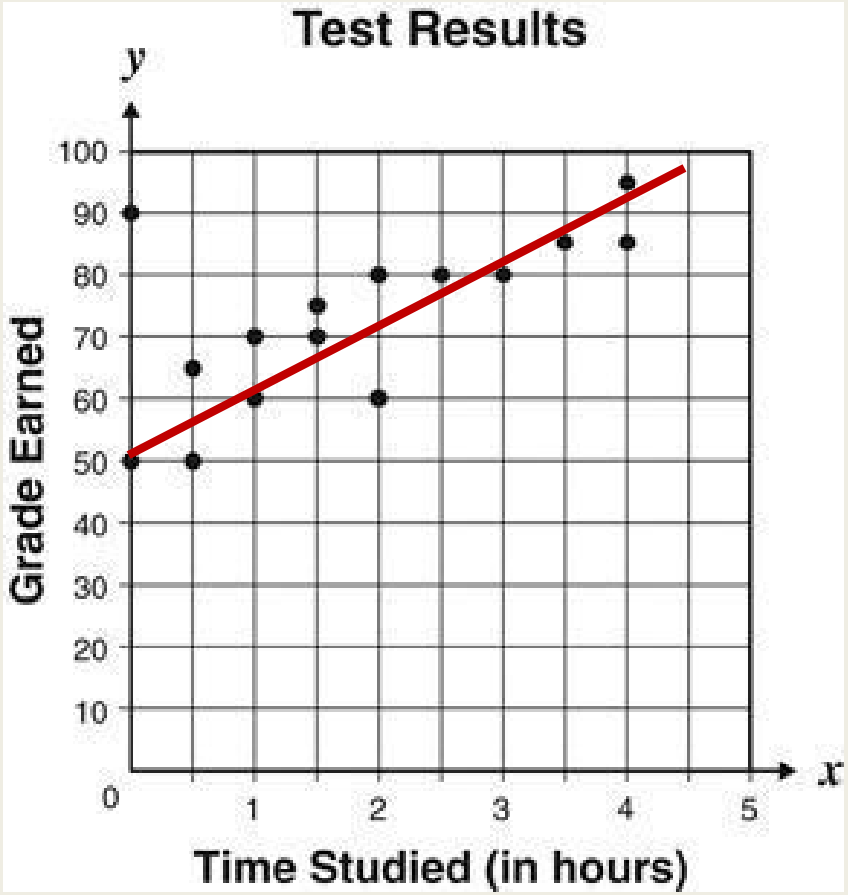
این دادگان نشاندهنده تاثیر تعداد ساعات مطالعه یک فرد بر نمره ای است که در آن درس خواهد گرفت.



رگرسیون

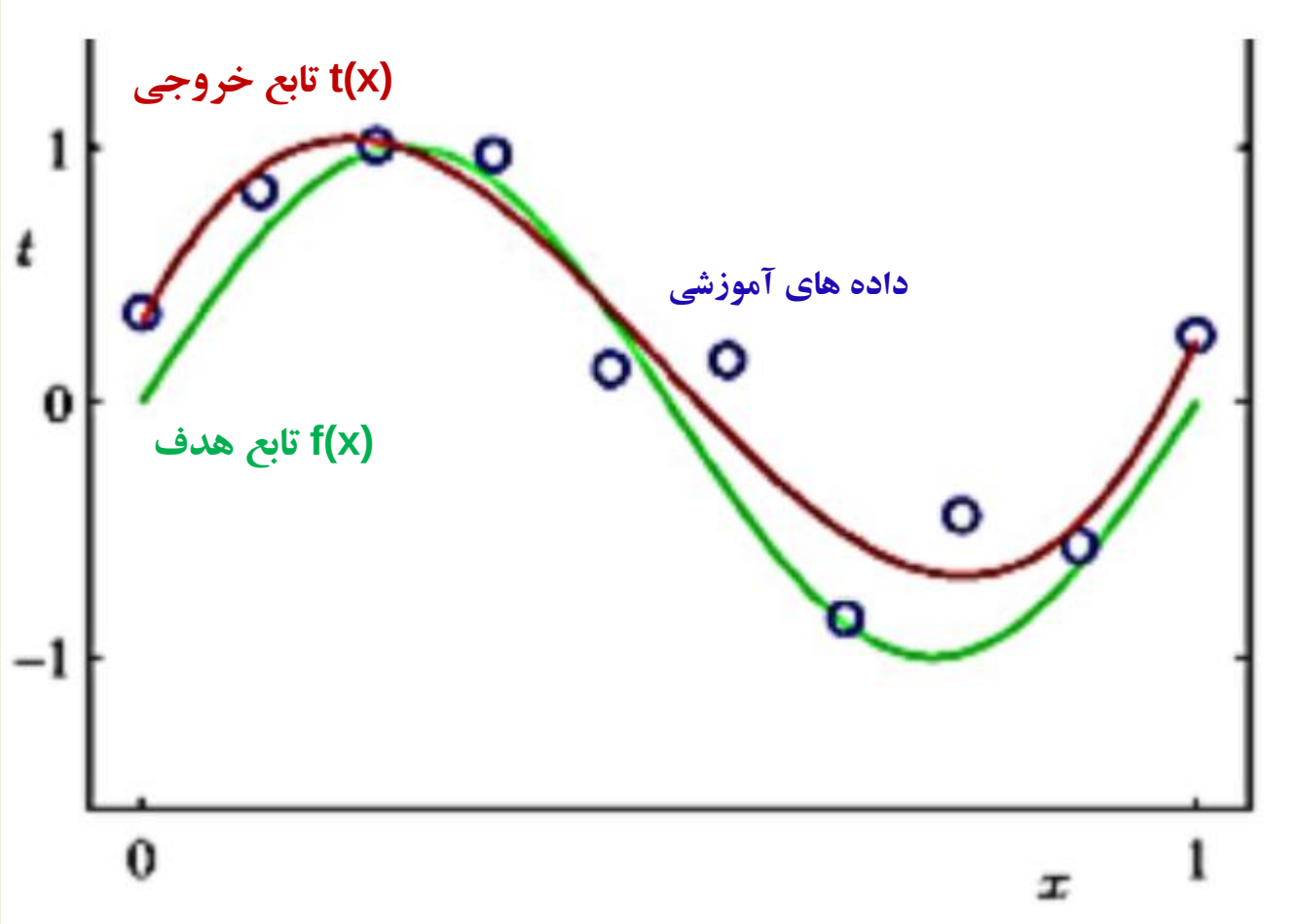
- **مثال : دادگان تاثیر مطالعه در نمره**

این دادگان نشاندهنده تاثیر تعداد ساعات مطالعه یک فرد بر نمره ای است که در آن درس خواهد گرفت.



- آیا می توان تابعی تخمین زد که تعداد ساعات مطالعه را به نمره کسب شده نگاشت نماید. در این صورت می توان در آینده با درنظرگرفتن تعداد ساعات مطالعه یک فرد نمره ای که کسب خواهد کرد را پیش بینی نمود.

رگرسیون



$$t(x) = f\ (x) + \epsilon$$

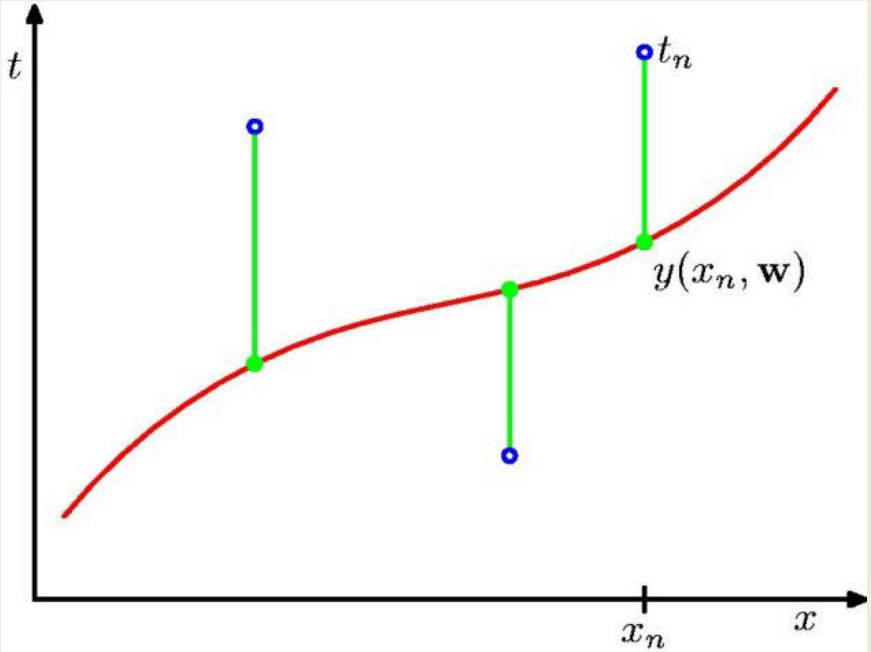
- که در این رابطه X مقدار ورودی تابع، $f(X)$ تابع هدف، ϵ نویز و ، $t(X)$ خروجی داده های موجود است.
- هدف رگراسیون تطبیق یک منحنی بر روی این نقاط آموزشی است.
- در واقع ما نیاز به یک منحنی پارامتریک داریم تا با تغییر پارمترهای آن، سعی کنیم منحنی را به دادههای آموزشی نزدیک کنیم.

رگرسیون

- **فرموله نمودن مساله**
- در مساله رگرسیون داده ها معمولا به صورت یک زوج مرتب تعریف می شود که در آن x ویژگی ورودی و t خروجی مطلوب است.

$$D = \left(\left(x^{(1)}, t^{(1)}\right), \left(x^{(2)}, t^{(2)}\right), \ldots, \left(x^{(N)}, t^{(N)}\right)\right)$$

- با استفاده از این دادههای آموزشی میتوان فرضیه (*hypothesis*)یا تخمین زننده تابعی را بدست آورد که مقدار x را به tنگاشت میکند.



$$J(W) = \sum_{n=1}^{N} [t^{(n)} - y\,(x^{(n)}, w)]^2$$

تابع هزینه

- برای اصلاح پارامترها با هدف کاهش تابع هزینه $J(\mathbf{w})$، از روش کاهش گرادیان(*gradient descent*)استفاده میکنیم.
- برای پارامتر W در خلاف جهت گرادیان حرکت کن تا به ازای مقدار W مقدار J حداقل شود.

$$W = W - \lambda \frac{\partial J}{\partial W}$$

گرادیان تابع هدف

رگرسیون در پایتون: پیش بینی قیمت خانه

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

رگرسیون در پایتون: پیش بینی قیمت خانه

1. Import libs

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

2. Check out the data

```
USAhousing = pd.read_csv('USA_Housing.csv')
```

3. Training a Linear Regression Model:

3.1. X and y arrays

```
X = USAhousing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
                'Avg. Area Number of Bedrooms', 'Area Population']]

Y = USAhousing['Price']
```

3.2. Train Test Split

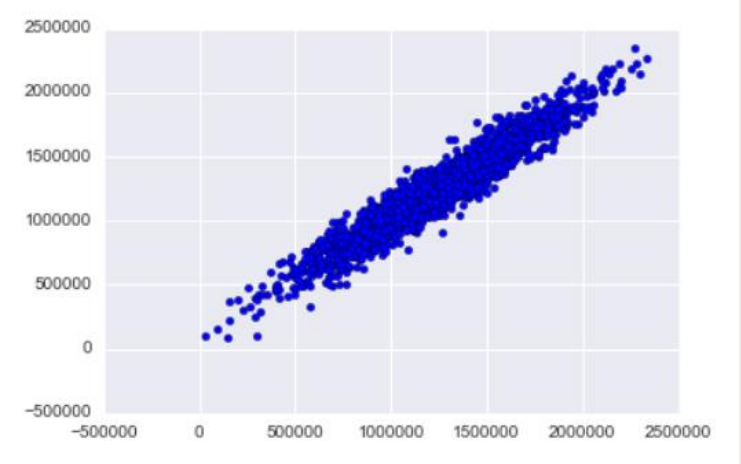
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
```

4. Creating and Training the Model

```
lm = LinearRegression()
lm.fit(X_train,y_train)
```

5. Predictions from our Model

```
predictions = lm.predict(X_test)
plt.scatter(y_test,predictions)
```



خلاصه

- تمرکز حوزه هوش مصنوعی بر ساخت و توسعه عامل های هوشمند است.
- ساخت و توسعه عامل هوشمند نیازمند معماری سخت افزاری و برنامه عامل است.
- هدف برنامه عامل توسعه تابع عامل است که انتخاب یک عمل بر اساس درک از محیط را پیاده سازی می نماید.
- برای پیاده سازی برنامه عامل الگوریتم های مختلف برای کاربردهای مختلف وجود دارد.
- عامل های یادگیرنده به عنوان نسل جدید عامل های هوشمند از یادگیری ماشین برای کسب دانش های جدید استفاده می نمایند.
- یادگیری ماشین به عنوان یک روش در هوش مصنوعی با یادگیری از طریق مدلسازی داده های نمونه می تواند رخدادهای آینده را پیش بینی نماید.

مراجع

- *Artificial Intelligence, A Modern Approach*, Stuart Russell, Peter Norvig, Third Edition, 2010.
- *Machine Learning*, Tom M. Mitchell.
- ***مفاهیم یادگیری ماشین***، محمد علی کیوان راد.