Artificial Intelligence with DNN

Jean-Sylvain Boige
Aricie
jsboige@aricie.fr

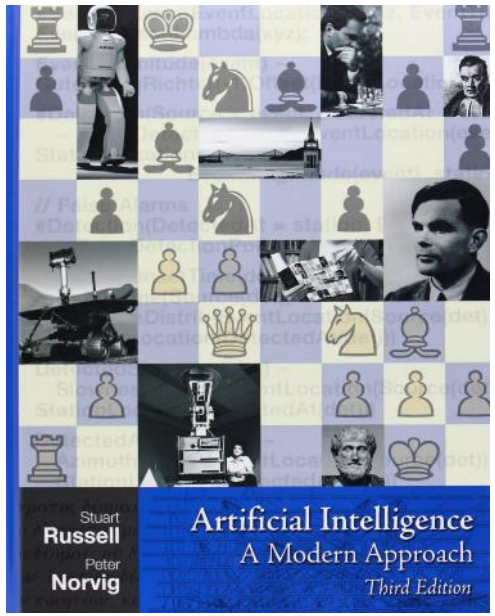
Please support our valuable sponsors

Summary

- Introduction to AI
  - What is AI?
  - Agent systems
  - DNN environment
- A Tour of AI in DNN
  - Problem Solving
    - Search, optimization
  - Reasoning
    - Logic, Knowledge bases
  - Dealing with uncertainty
    - Probabilistic networks
  - Machine learning
    - For all of the above !

Motivation

- Course at ORT engineer school
  - http://www.reddit.com/r/ia101
  - Artificial Intelligence
    A Modern Approach



- Aricie- Portal Keeper
  - Agents in DNN
  - https://ariciepkp.codeplex.com
  - Future: My Intelligence Agency
  - You(rs)?

What is AI?

Turing: Computing Machinery and Intelligence

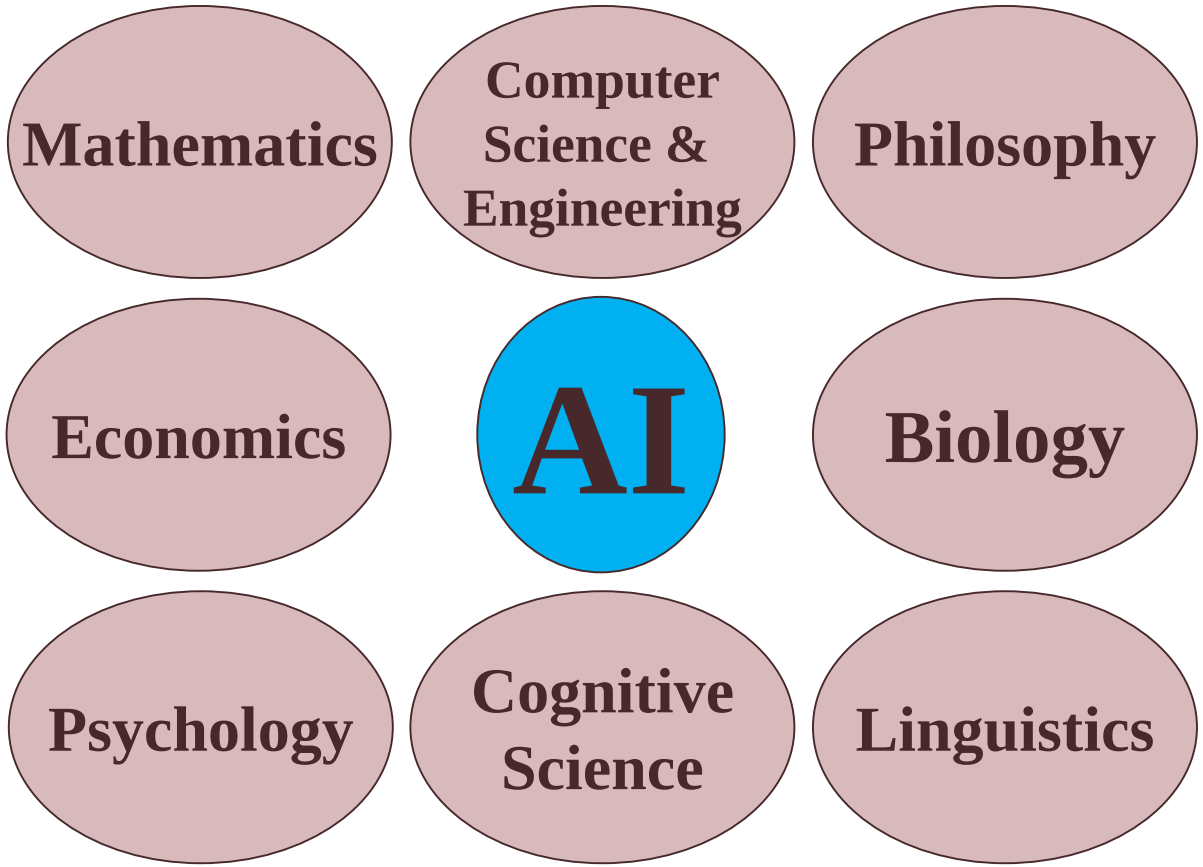
Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

Most useful approach: "acting rationally"

→ Build agents

Foundations of AI



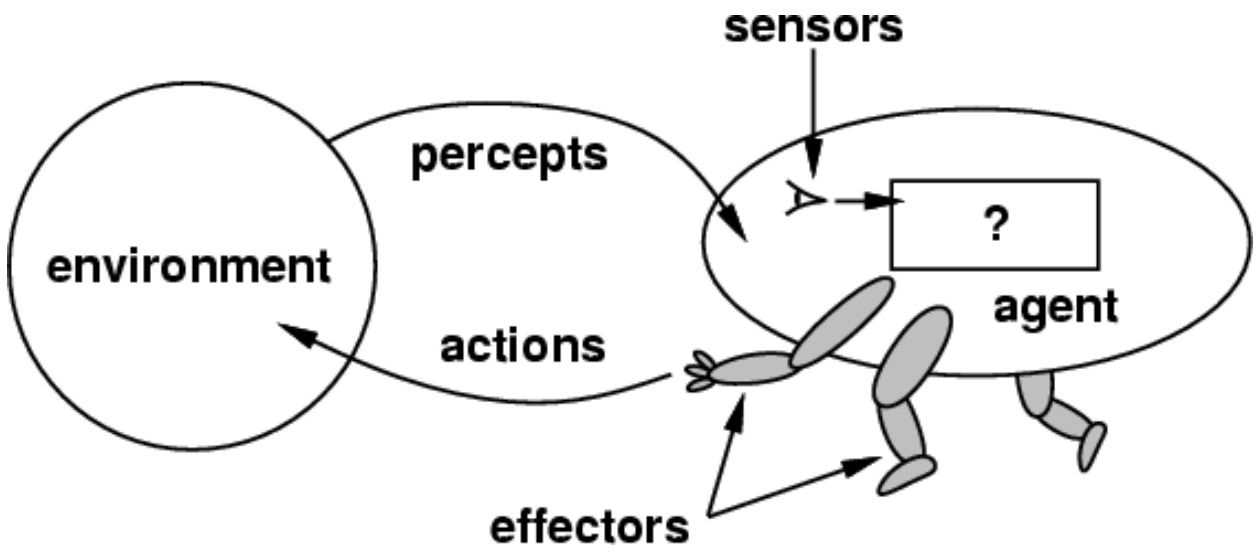
- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" adopted
- 1950s Early AI programs, Logic Theorist, Geometry Engine
- 1965 logical reasoning
- 1970s AI discovers computational complexity
  Neural network research almost disappears
- 1970s Early development of knowledge-based systems
- 1980s AI an industry, Robotics, Planning, Control theory
- 1986 Neural networks return to popularity
- 1990s AI becomes a science
- 1995 The emergence of intelligent agents, GAs, Artificial Life
- 2000s Bayesian learning, Knowledge Engineering
- 2010s Deep learning – Smart contracts

AI in your everyday life

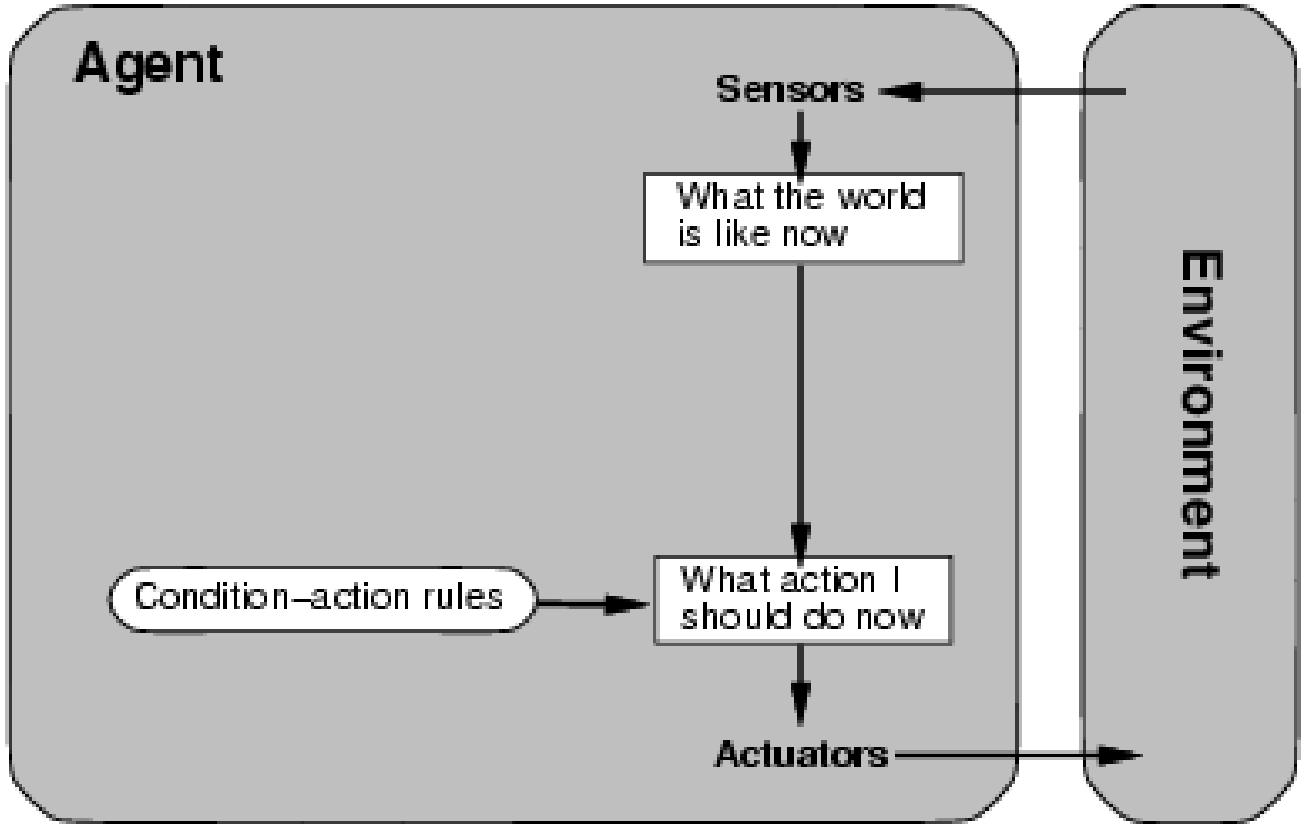
- Post Office
  - address recognition
  - sorting of mail
- Banks
  - check readers
  - signature verification
  - loan application
- Customer Service
  - voice recognition
  - Planning
- The Web
  - Identifying your age, gender, location, from your Web surfing
  - fraud detection
- Digital Cameras
  - face detection and focusing
- Computer Games
  - Intelligent characters/agents

- Agent function : maps from percepts history into actions

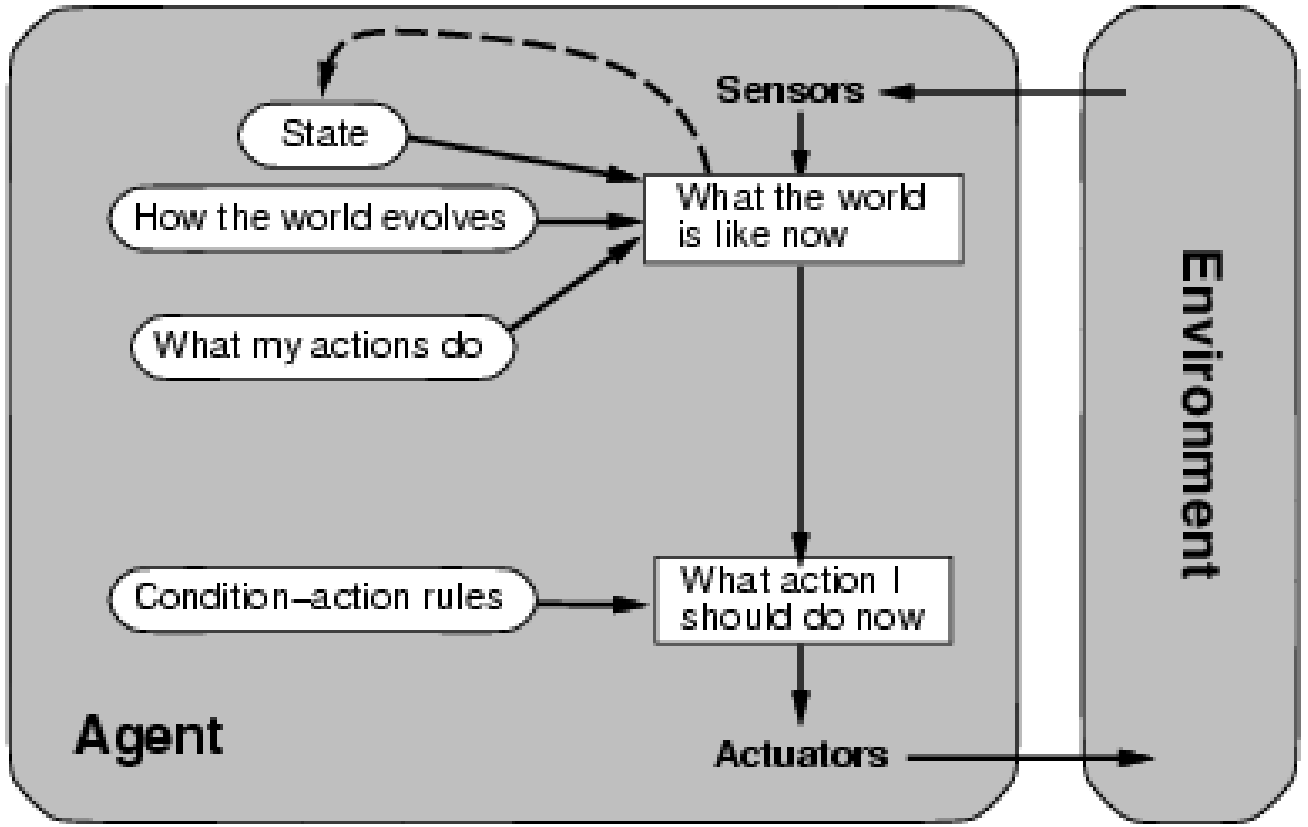
$$[f: \mathcal{P}^{\star} \rightarrow \mathcal{A}]$$



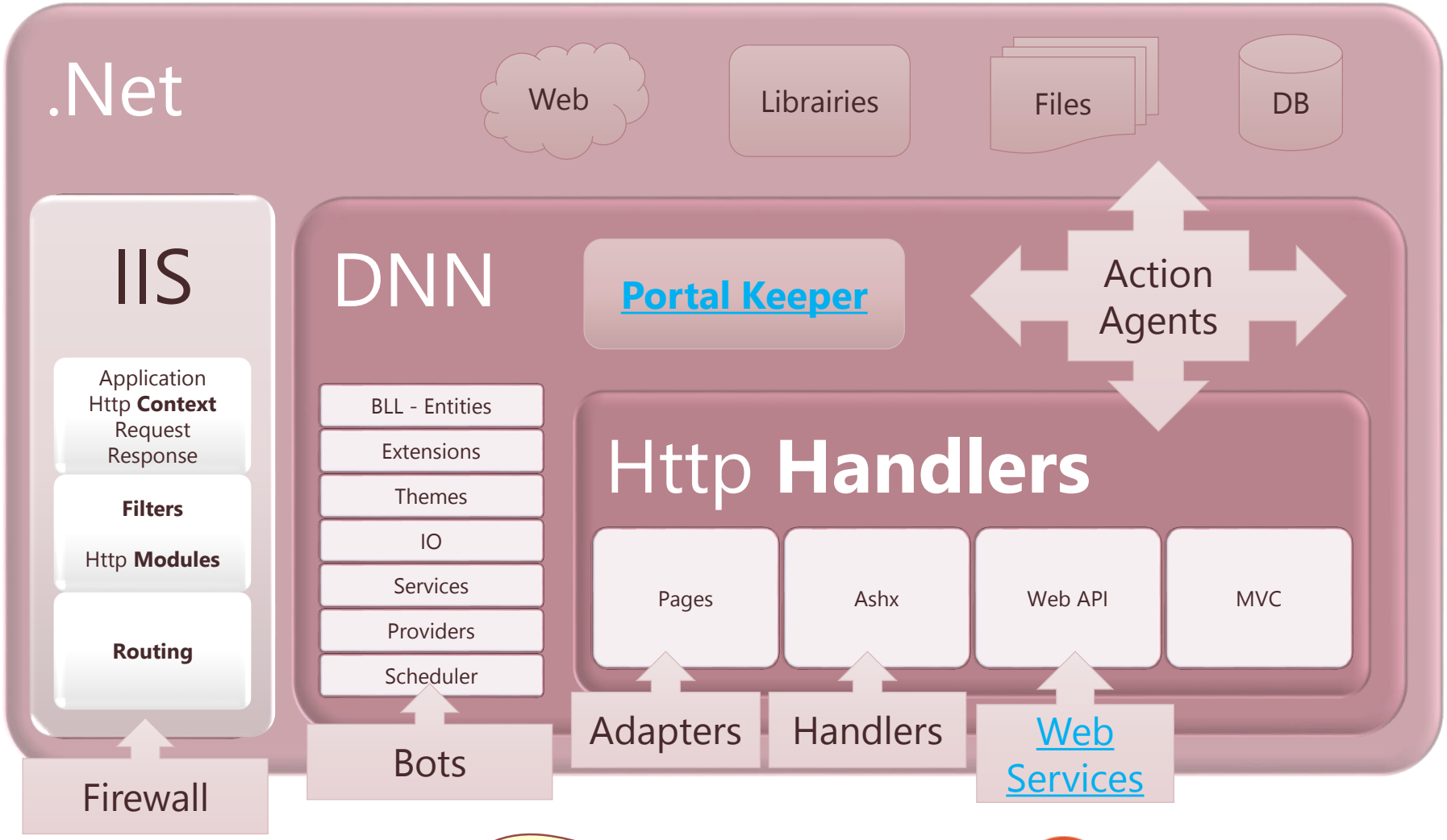
Simple reflex agents



Model-based reflex agents

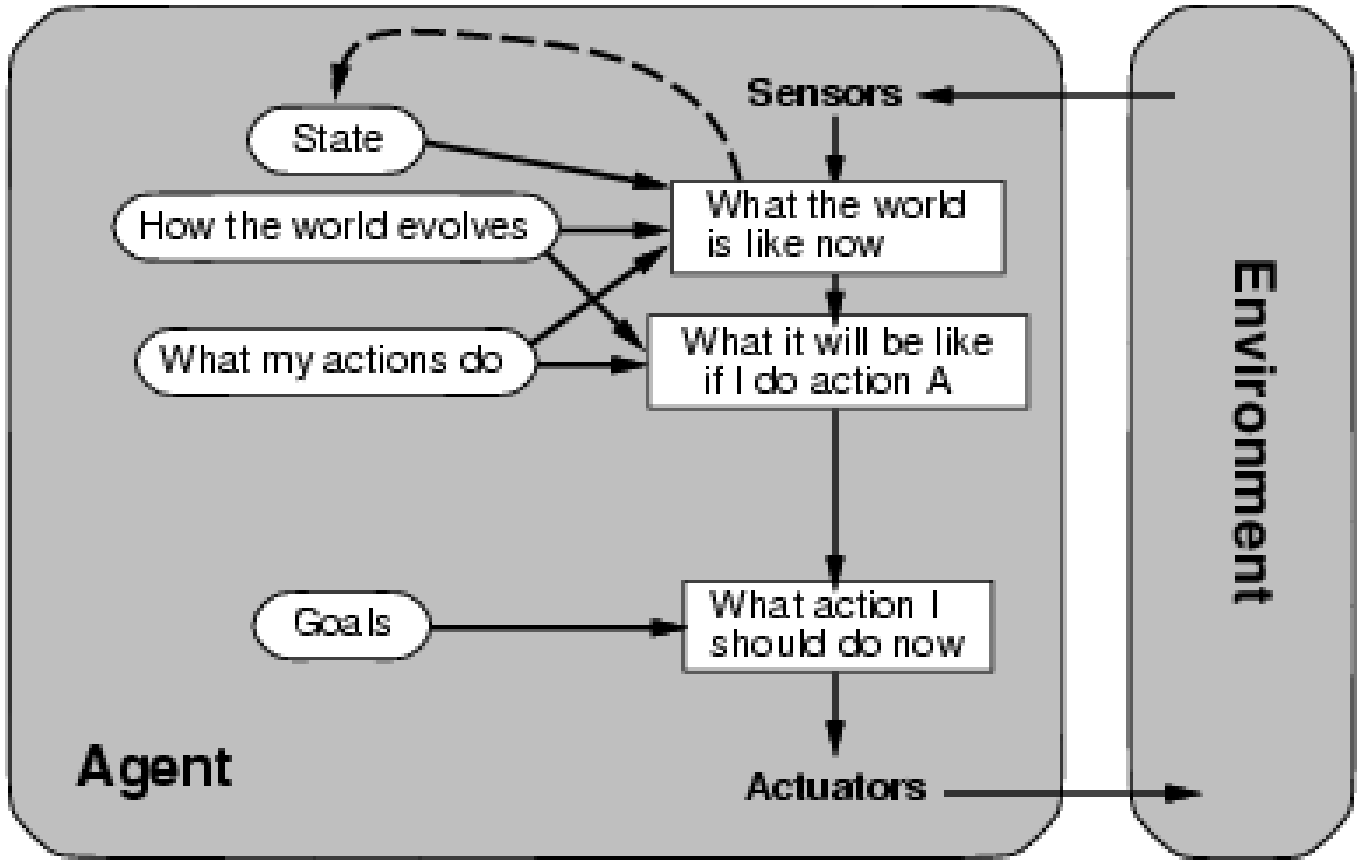


DNN and Portal Keeper



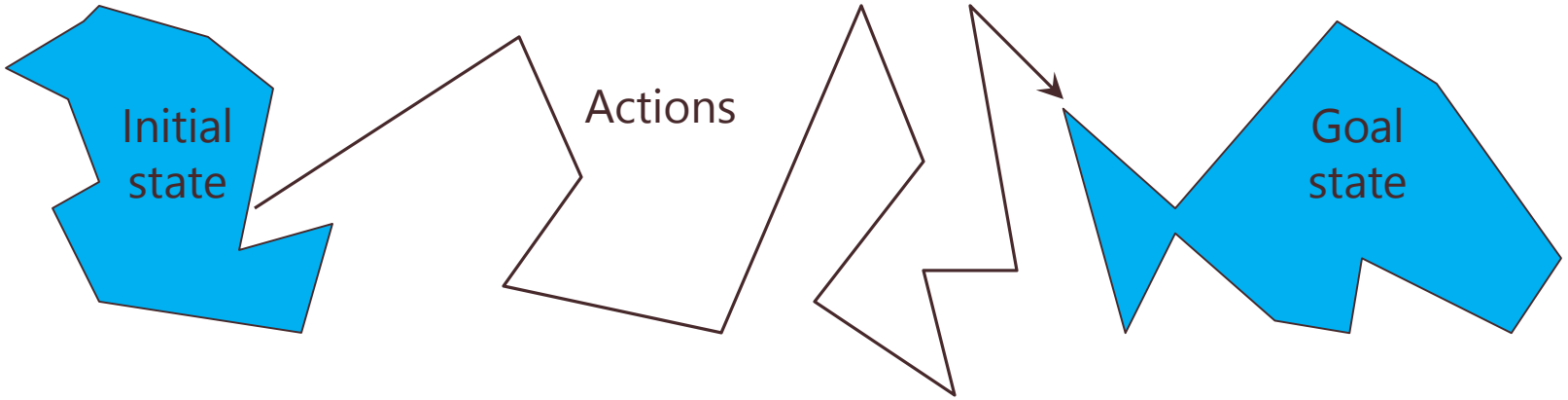
Demo 1 AI Services + POSTMAN

Goal-based agents

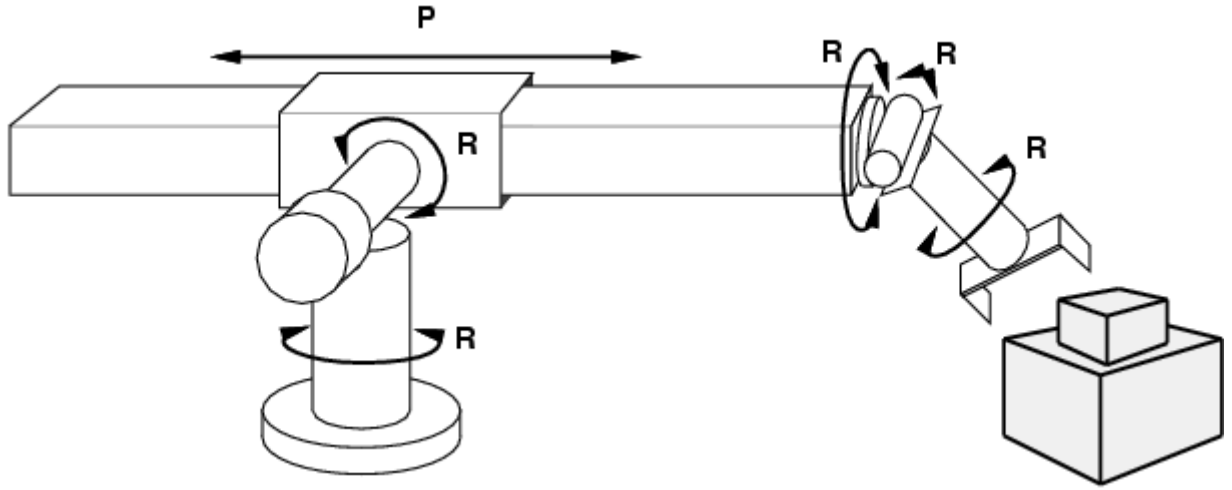


Building goal-based agents

- What is the goal to be achieved?
- What are the actions?
- What is the states representation?

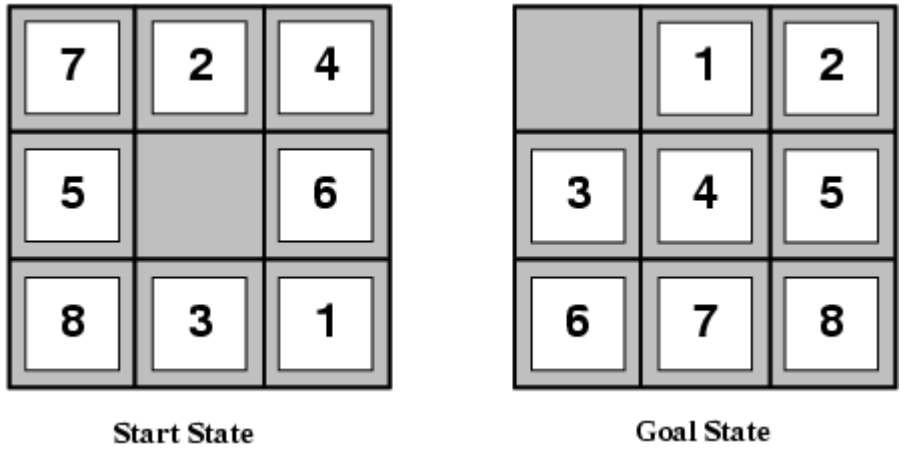


Example: robotic assembly



- states?: coordinates of joint angles, parts of the object to be assembled
- actions?: continuous motions of robot joints
- goal test?: complete assembly
- path cost?: time to execute

Example: The 8-puzzle

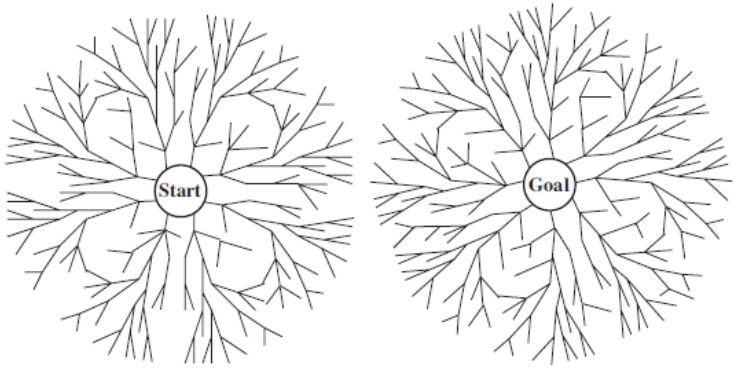


- states? locations of tiles
- actions? move blank left, right, up, down
- goal test? = goal state (given)
- path cost? 1 per move

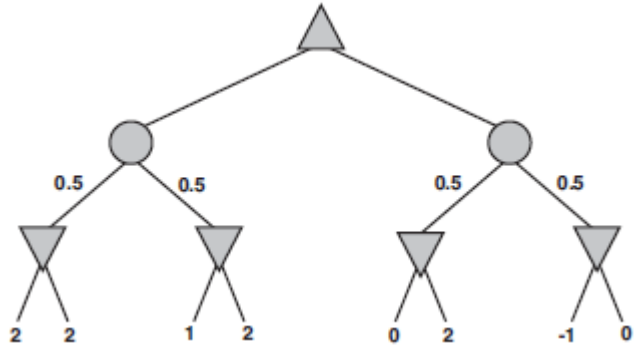
[Note: optimal solution is NP-hard]

Search

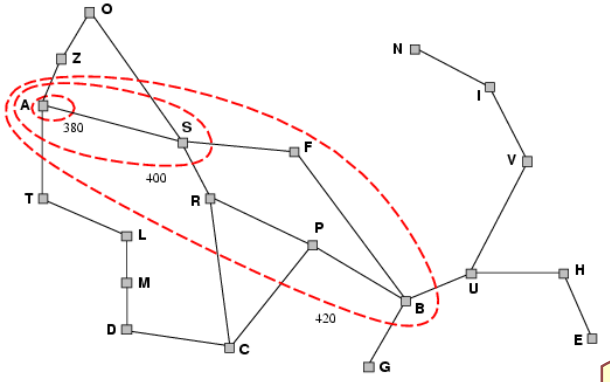
- Uninformed Search



- Adversarial Search – Games

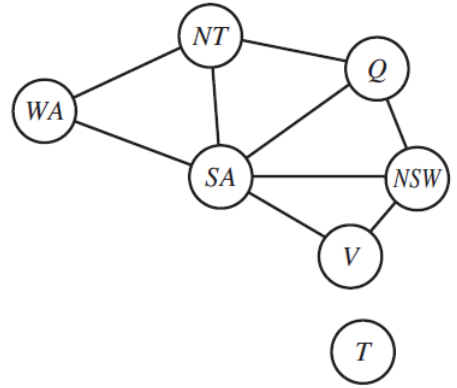


- Informed Search – Heuristics



Demo 2

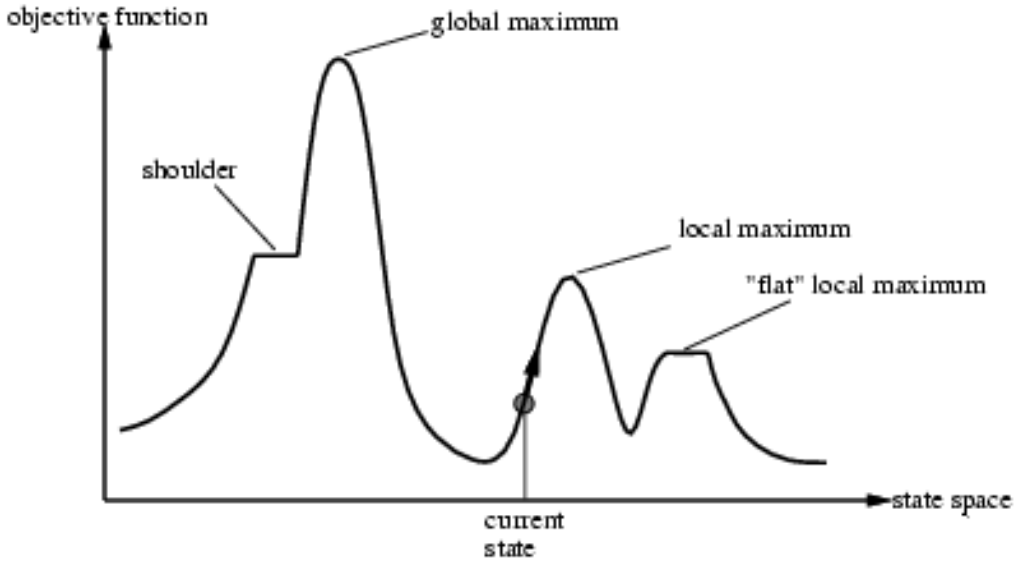
- Constraint Satisfaction problems



[PathFinding.js](), [Search]()

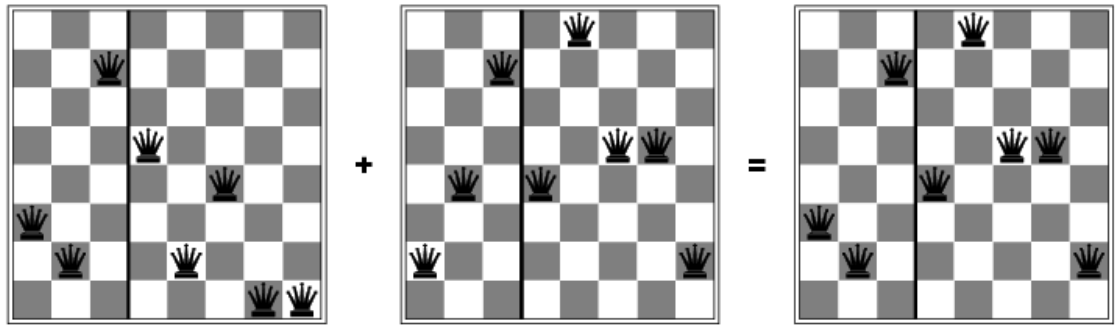
Local Search

- Hill Climbing, Gradient descent



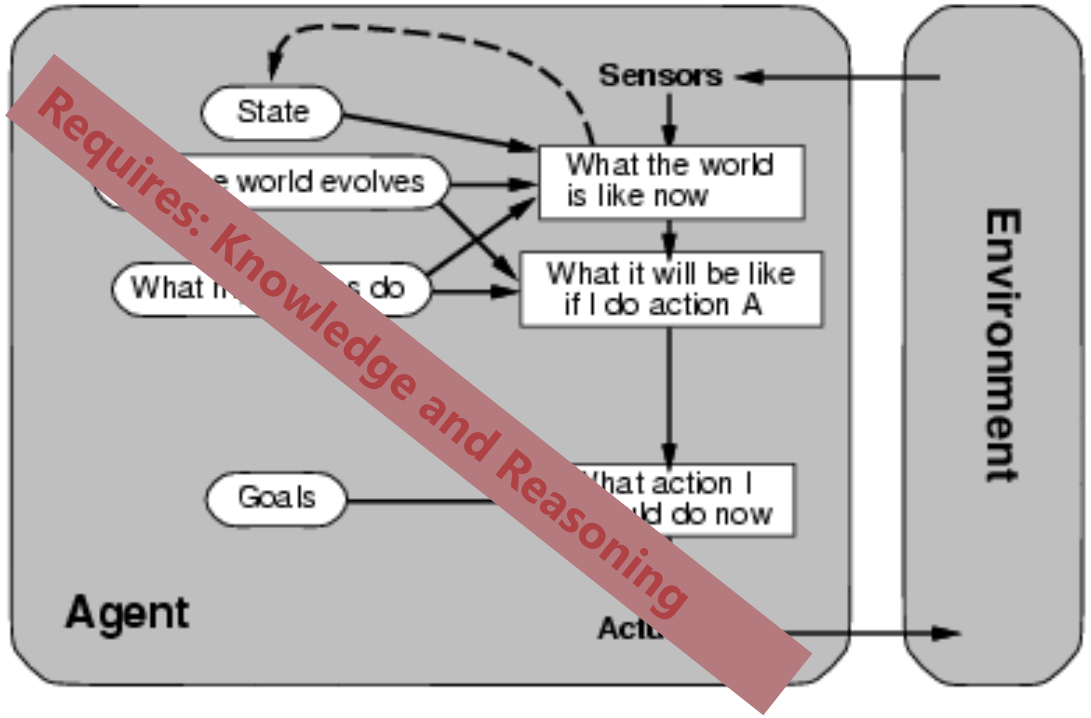
- Problem: depending on initial state, can get stuck in local maxima
  - Simulated annealing,
  - Stochastic Beam Search

- Genetic algorithms

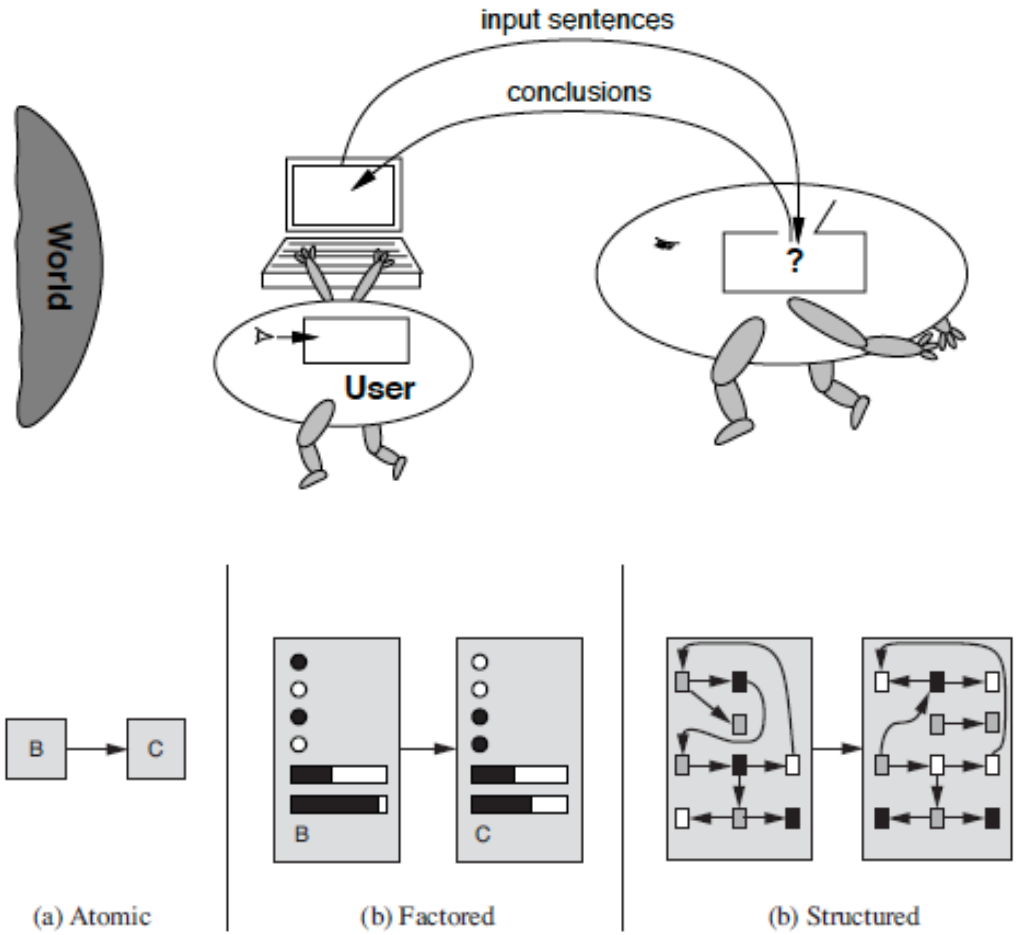


- Genetic Sharp

A Model-Based Agent



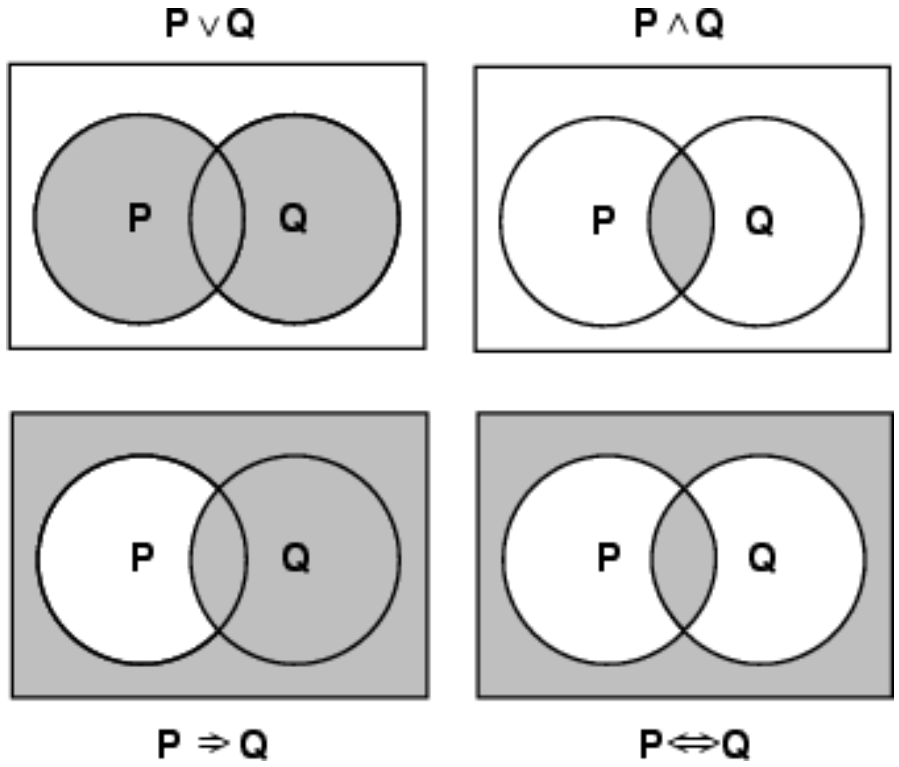
Tenuous Link to Real World



A grammar of sentences in propositional logic

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} &\rightarrow (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
&\mid \neg\ \mathit{Sentence} \\
&\mid \mathit{Sentence} \wedge \mathit{Sentence} \\
&\mid \mathit{Sentence} \vee \mathit{Sentence} \\
&\mid \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
&\mid \mathit{Sentence} \Leftrightarrow \mathit{Sentence}
\end{aligned}
$$

OPERATOR PRECEDENCE : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

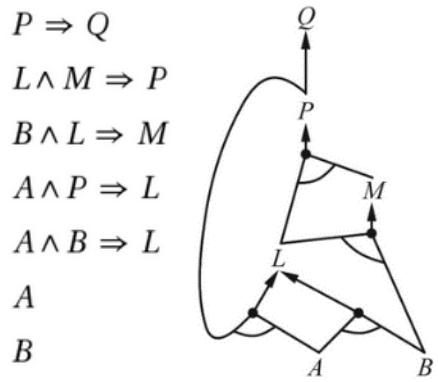


Parsing: **Grammatica**

Proving things

- A **proof** is a sequence of sentences derived by rules of inference.
- Last sentence = **theorem** (goal or query) to prove.
- Examples

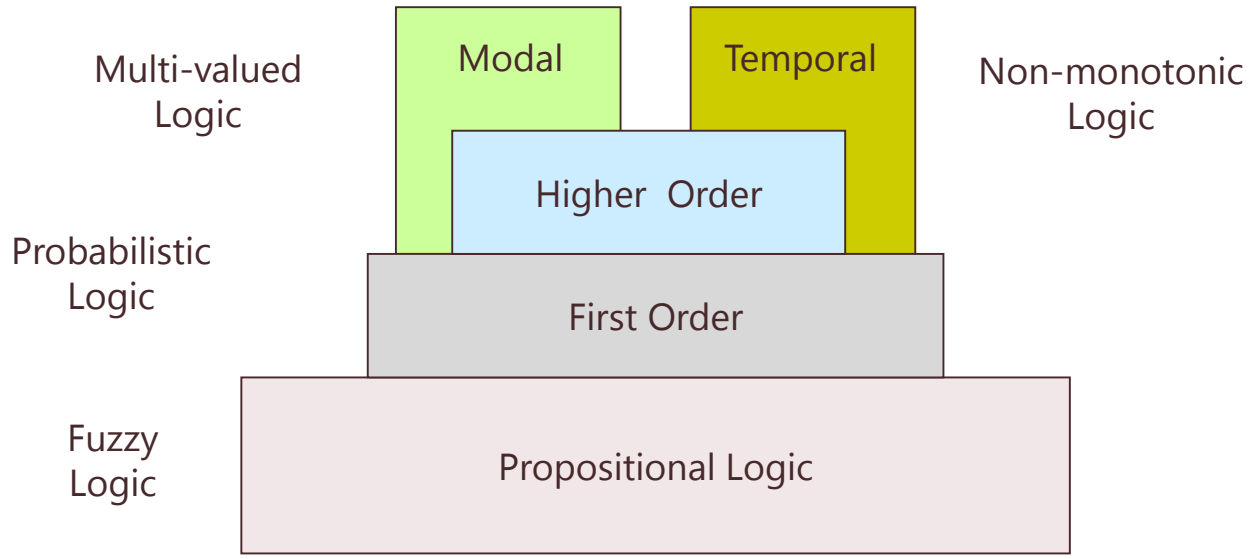
1 Humid	Premise	"It is humid"
2 Humid→Hot	Premise	"If it is humid, it is hot"
3 Hot	Modus Ponens(1,2)	"It is hot"



- But lacks expressivity

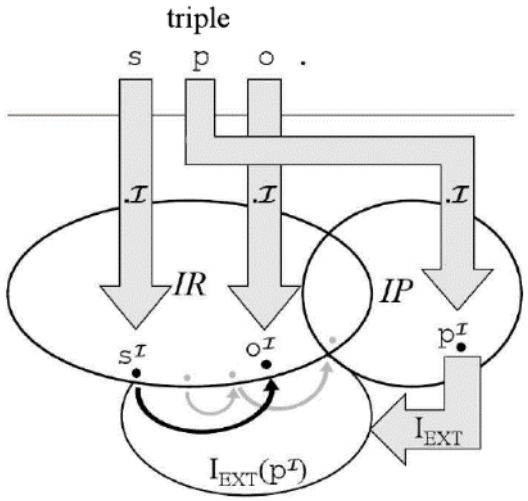
First Order Logic and KR languages

$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} &\rightarrow \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} &\rightarrow (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
&\mid \neg\ \textit{Sentence} \\
&\mid \textit{Sentence} \wedge \textit{Sentence} \\
&\mid \textit{Sentence} \vee \textit{Sentence} \\
&\mid \textit{Sentence} \Rightarrow \textit{Sentence} \\
&\mid \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
&\mid \textit{Quantifier}\ \textit{Variable}, \ldots\ \textit{Sentence} \\
\textit{Term} &\rightarrow \textit{Function}(\textit{Term}, \ldots) \\
&\mid \textit{Constant} \\
&\mid \textit{Variable} \\
\textit{Quantifier} &\rightarrow \forall \mid \exists \\
\textit{Constant} &\rightarrow A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} &\rightarrow a \mid x \mid s \mid \cdots \\
\textit{Predicate} &\rightarrow \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} &\rightarrow \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\text{OPERATOR PRECEDENCE} &: \neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{aligned}
$$

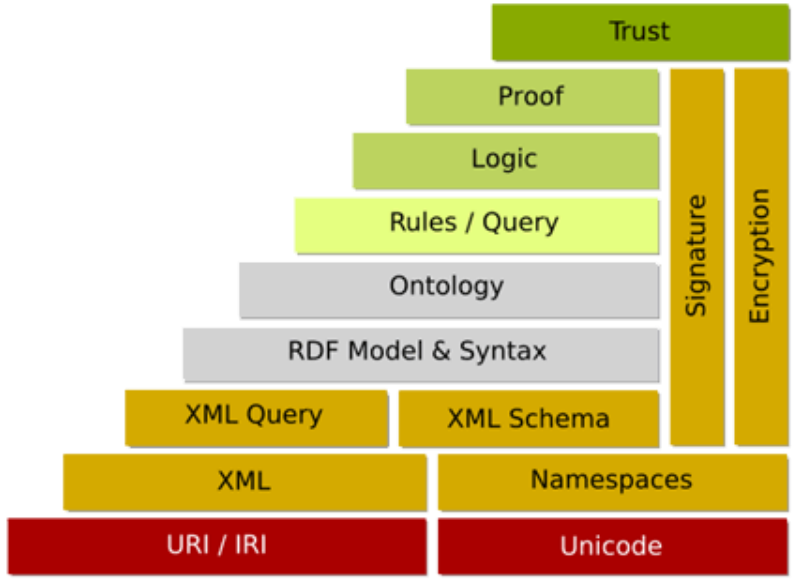


Demo 3 [Logical inference]

- Resource Description Framework
  - KR community: AAAI, W3C, **Berners-Lee**
  - RDF - triples (facts), class / subclass



  - RDFS - OWL - defined classes, constraints
  - SPARQL – Querying, Triple Stores,
  - Linked-Data - SOA

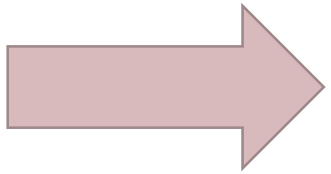


dotNetRDF
An Open Source C#.Net Library for RDF

BrightstarDB
A native **RDF** database for the **.NET platform.**

But real world is uncertain

- Uncertain **inputs**
  - Missing, noisy data
- Uncertain **knowledge**
  - Multiple, Incomplete conditions/ causes / effects
  - Probabilistic/stochastic effects
- Uncertain **outputs**
  - Abduction and induction, default reasoning
  - Incomplete inference



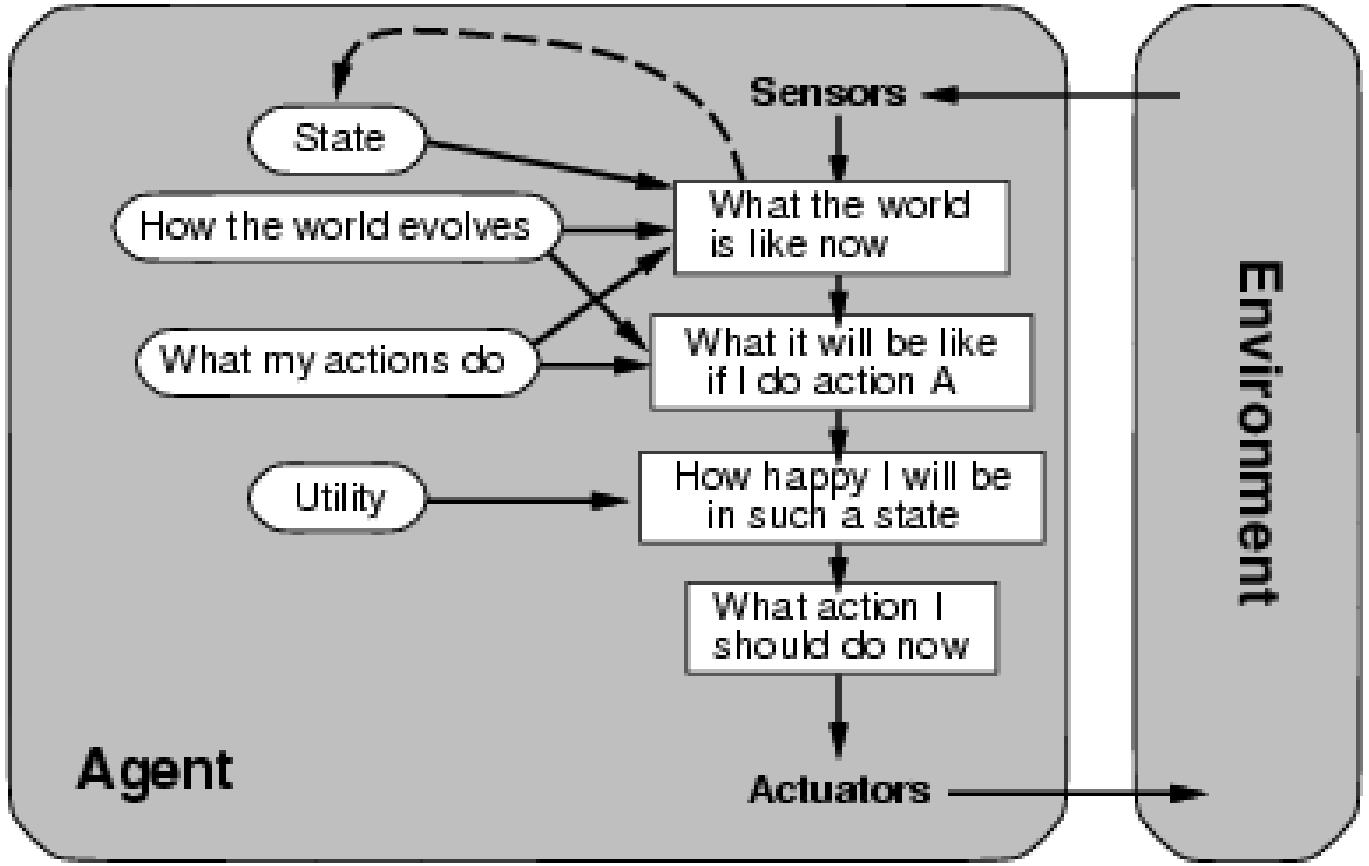
Probabilistic reasoning

→ Probabilistic results

Decision making with uncertainty

- **Rational** behavior becomes:
  - For each possible action, identify the possible outcomes
  - Compute the **probability** of each outcome
  - Compute the **utility** of each outcome
  - Compute the **expected utility** over possible outcomes for each action
  - Select the action with **Maximum Expected Utility**

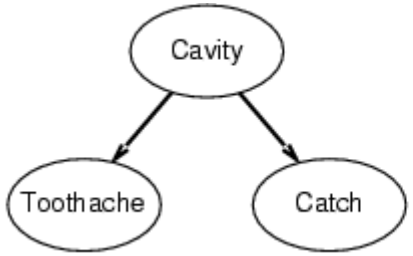
Utility-based agents

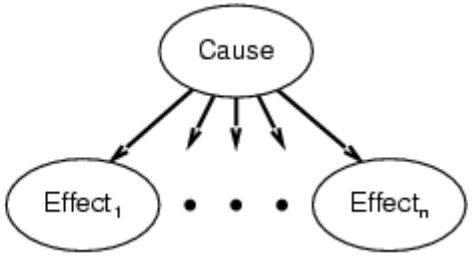


- Naïve Bayes model

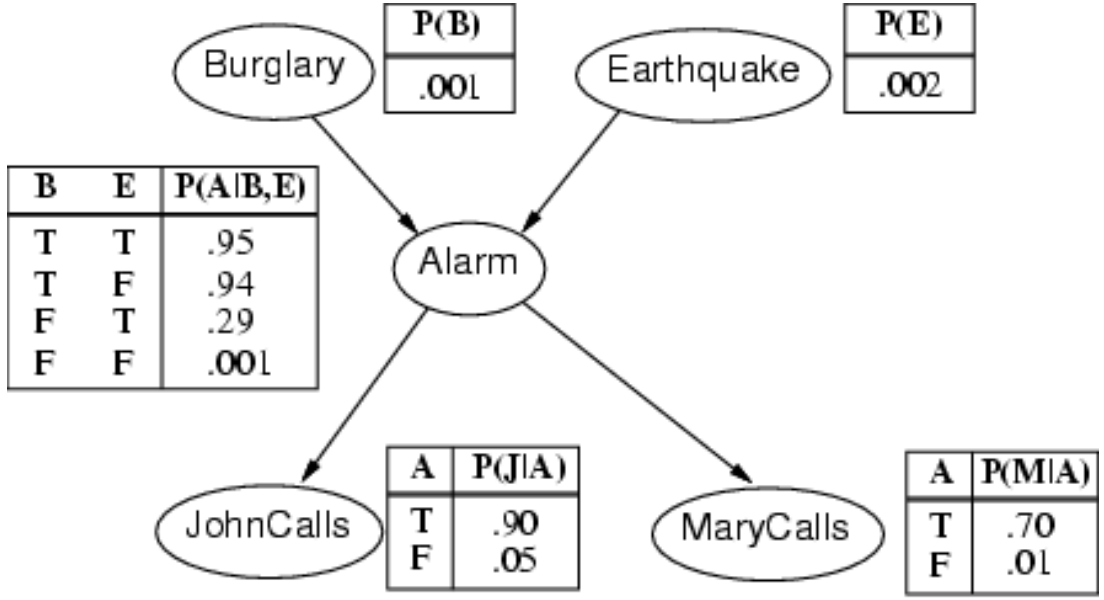
$$P(Cause, Effect_1, \dots, Effect_n)$$

$$= P(Cause) * \prod_i P(Effect_i | Cause)$$

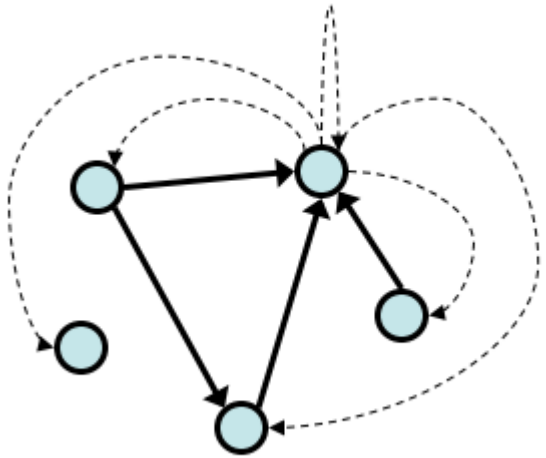


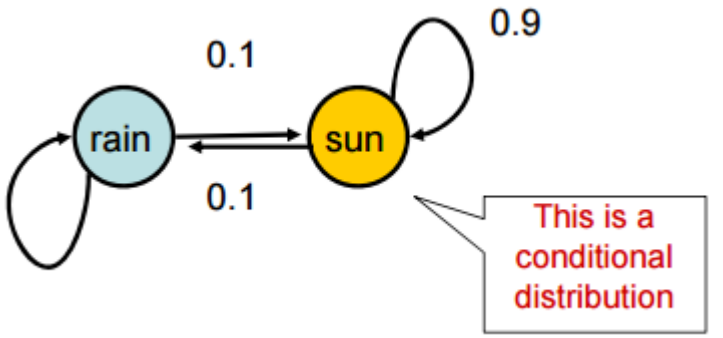


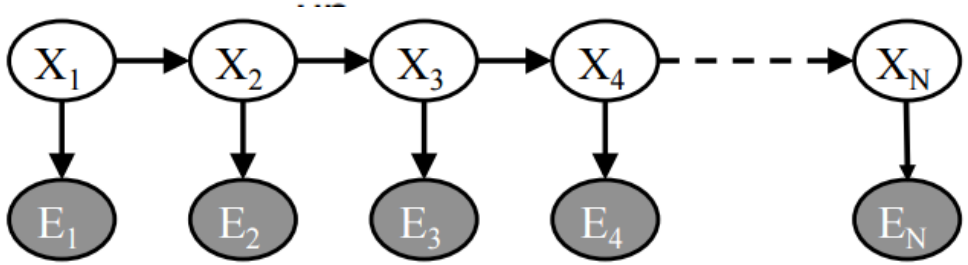
- Bayesian networks



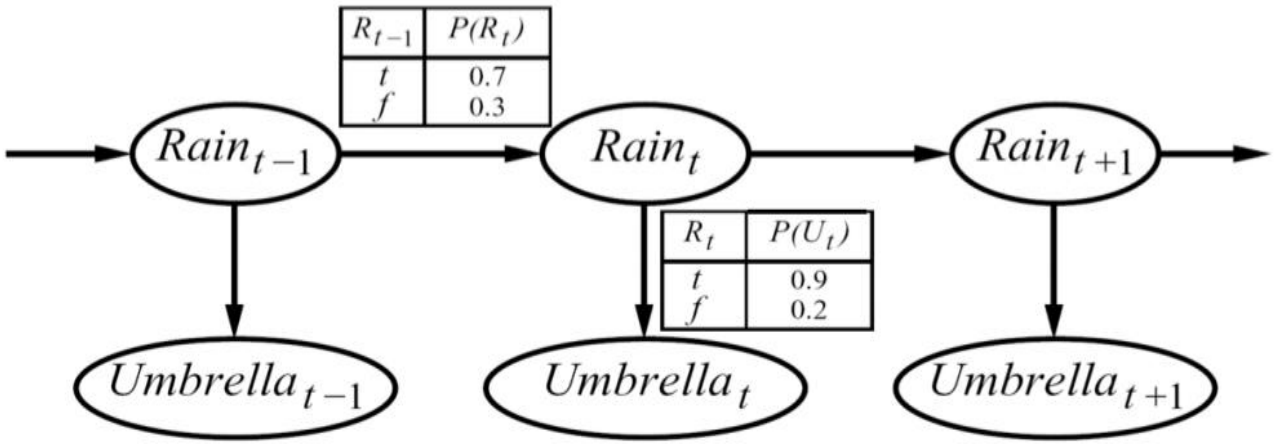
- Google 1.0 : PageRank over a web graph
  - Transitions:
    - With prob. 1-c, follow a random outlink (solid lines)
  - Stationary distribution
    - Will spend more time on highly reachable pages
- Markov processes
  - Hidden Markov Networks
    - Hidden state, observed evidence







Example: A simple weather Hidden Markov Model



- An Hidden Markov Model is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t | X_{t-1})$
  - Emissions: $P(E | X)$

Demo 4

Probabilistic inference

Real Hidden Markov Models Examples

- Natural Language processing
  - Text classification
  - Information retrieval
  - Information extraction

LUCENE.net SEARCH ENGINE LIBRARY

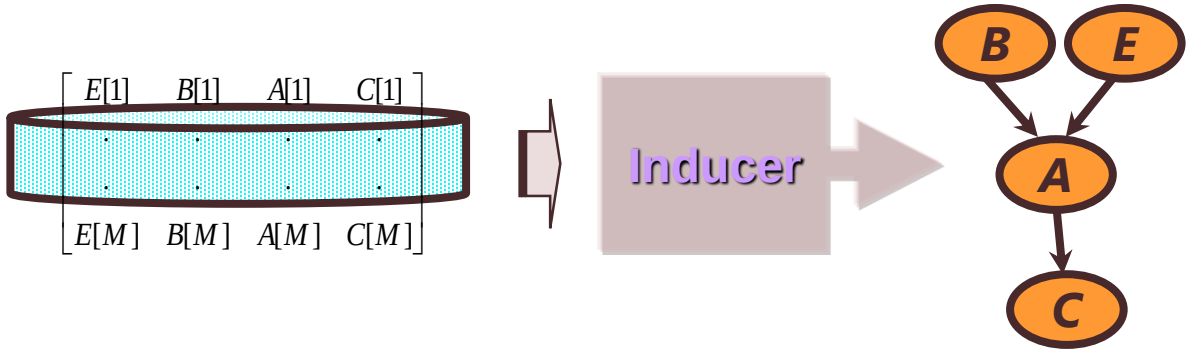
openNLP™

- Speech recognition HMMs:
  - Observations are acoustic signals
  - States are positions in words

- Machine translation HMMs:
  - Observations are words
  - States are translation options

- Radar tracking:
  - Observations are range readings
  - States are positions on a map

- →Trained probabilistic networks
  - Given training set *D*
  - Find H that best matches *D*



- Powerful toolkit: Infer.Net
  - Click-through
  - Sentiment analysis

Microsoft Cognitive Services

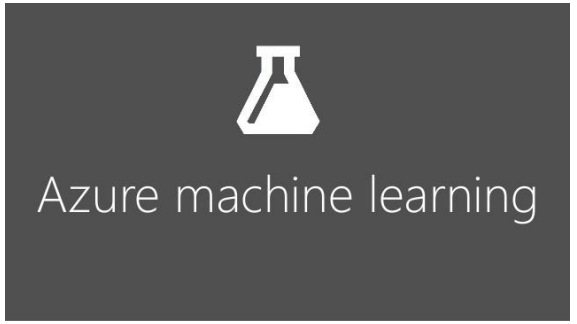
infer.net

Different Learning tasks

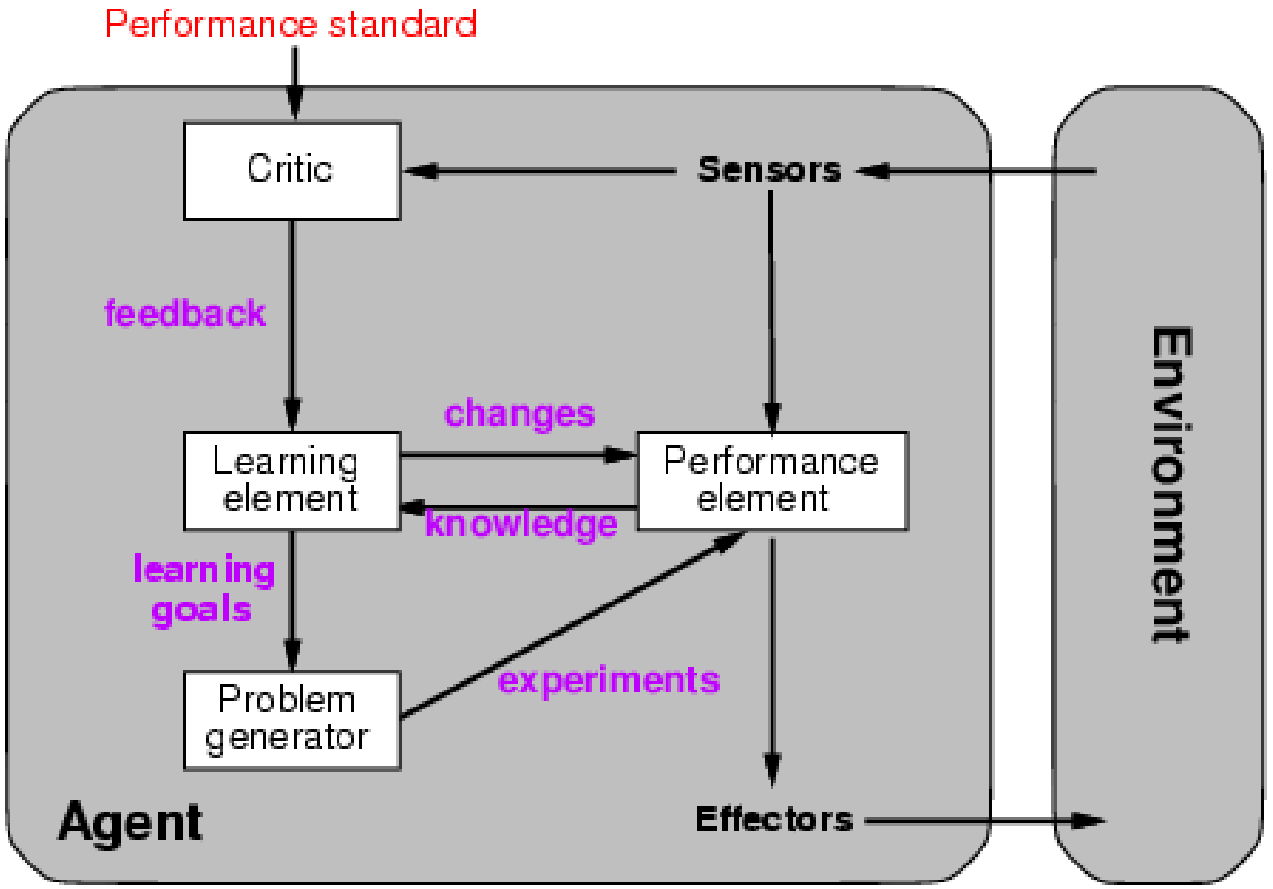
- Find ideal customers
  - Amex
- Find best person for job
  - BellAtlantic
- Predict purchasing patterns
  - Victoria Secret
- Help win games
  - NBA
- Catalogue natural objects
  - Quasars
- Bioinformatics
  - Identifying genes
  - Predicting protein function
- Recognizing Handwriting
  - Bell Labs – Lenet, US Postal
- Recognizing Spoken Words
  - Ticketmaster
- Translation
  - Google

Major paradigms of machine learning

- **Rote learning** –*Association*-based storage and retrieval.
- **Induction** – Use specific *examples* to reach *general* conclusions
- **Clustering** – *Unsupervised* identification of natural groups in data
- **Analogy** – *Correspondence* between different representations
- **Discovery** – Unsupervised, specific *goal not given*
- **Reinforcement** – *Feedback* at the end of a sequence of steps

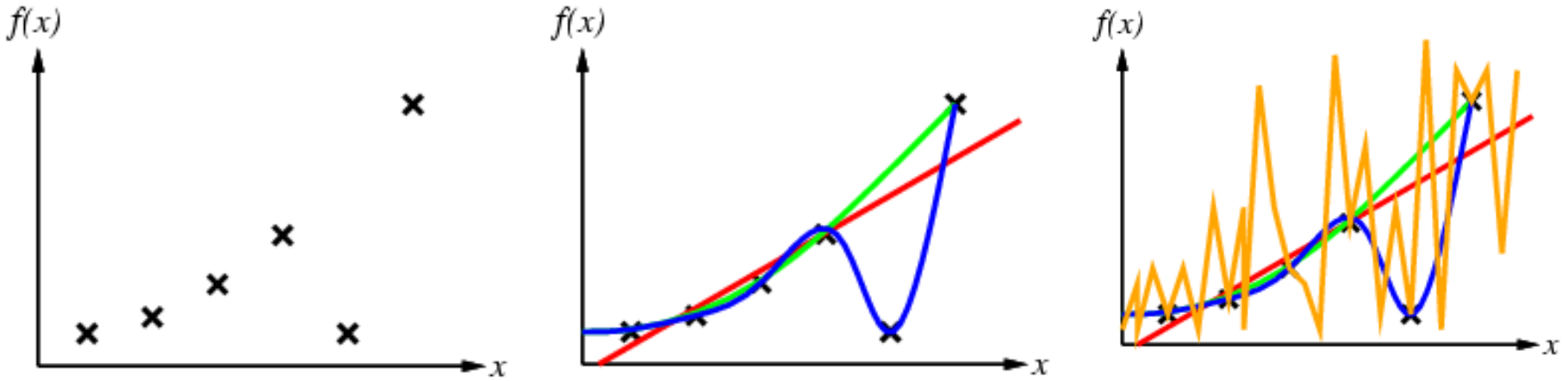


Learning agents



Inductive learning method

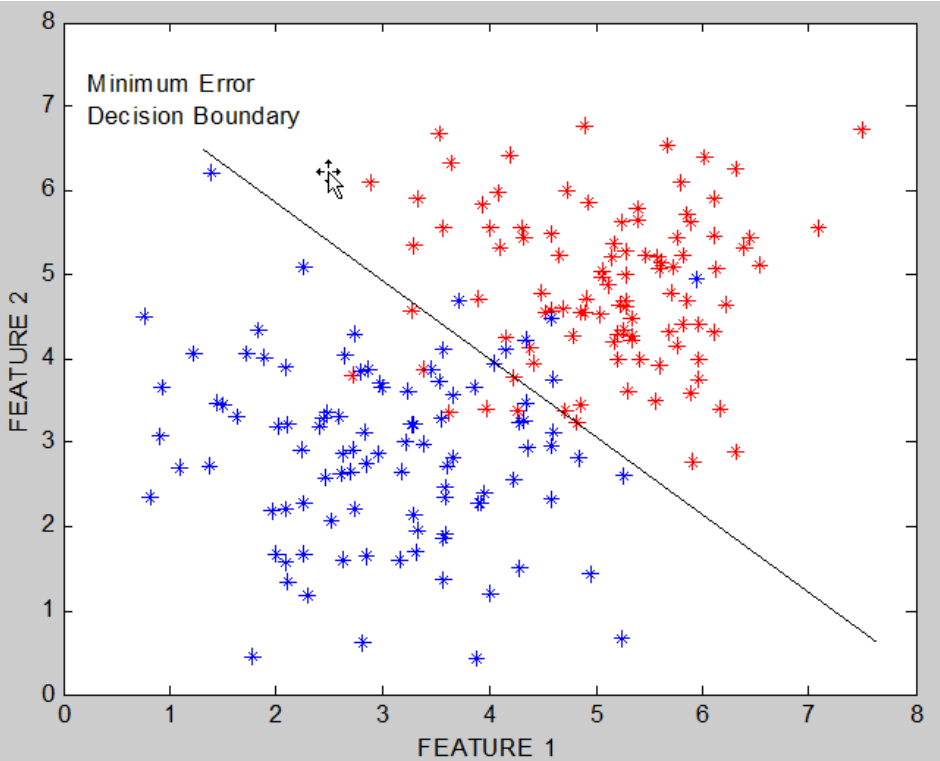
- Construct consistent hypothesis to agree on training set
- Example: Regression: curve fitting:



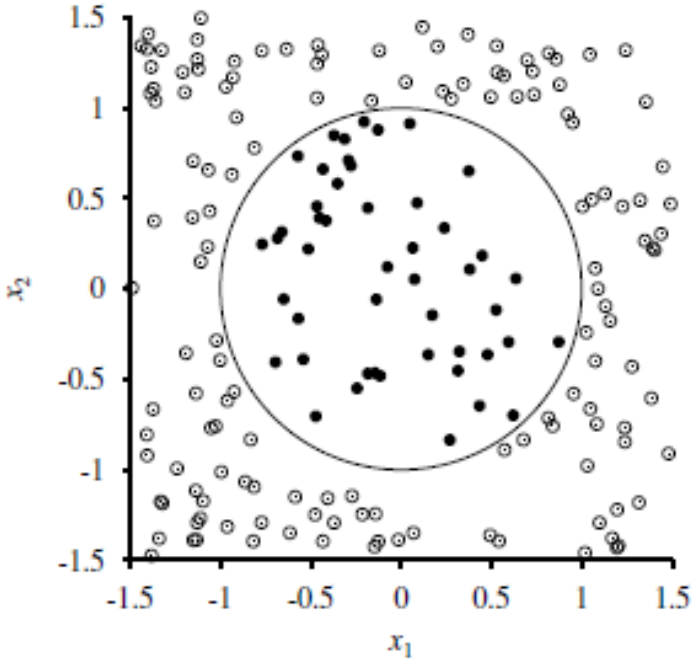
- Ockham's razor: prefer the simplest hypothesis consistent with data

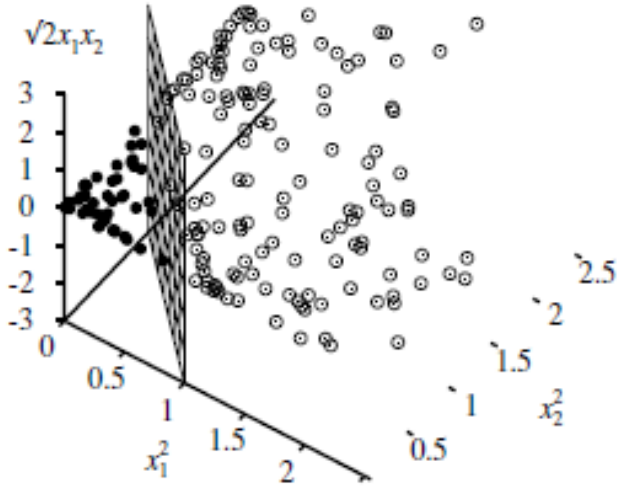
Classification

- Linear classifier



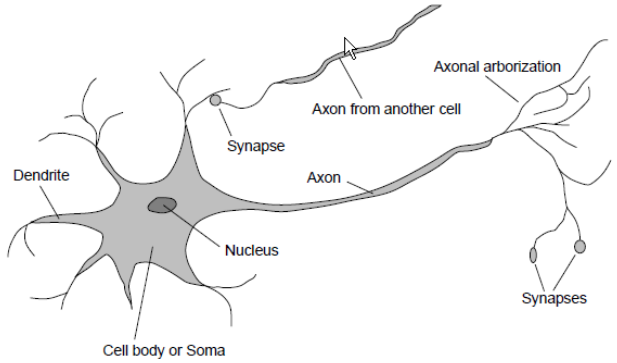
- Using higher dimensions



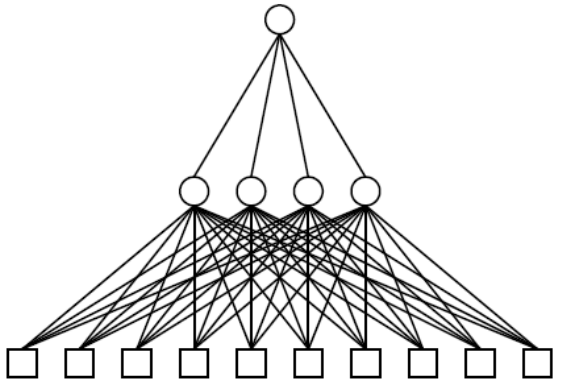


Artificial Neural Networks

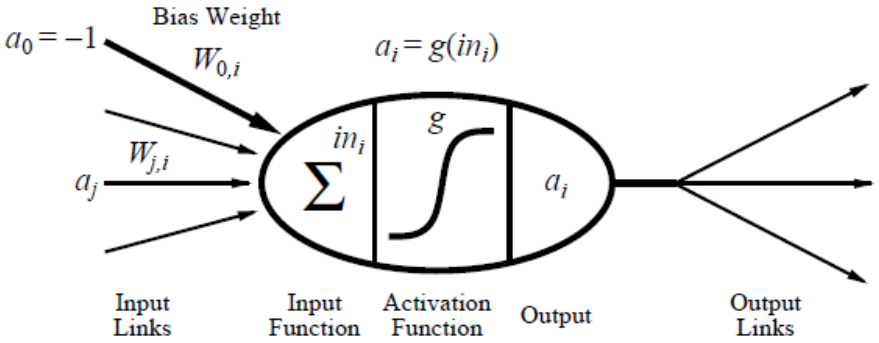
- Biological inspiration



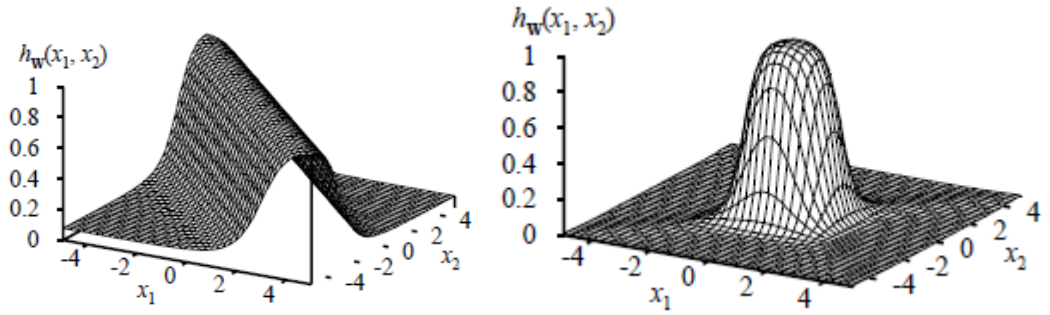
- Multiple layers



- Artificial Unit

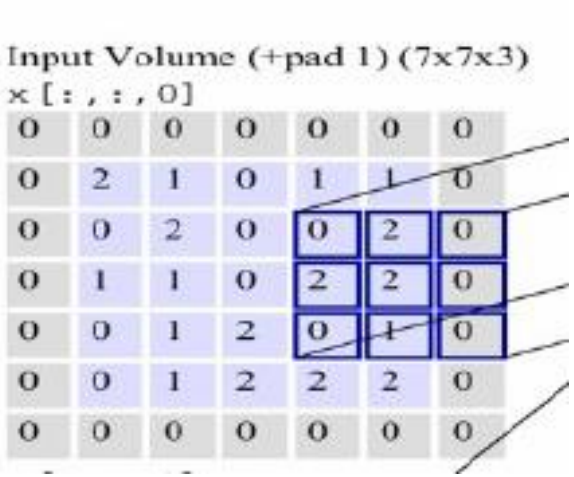


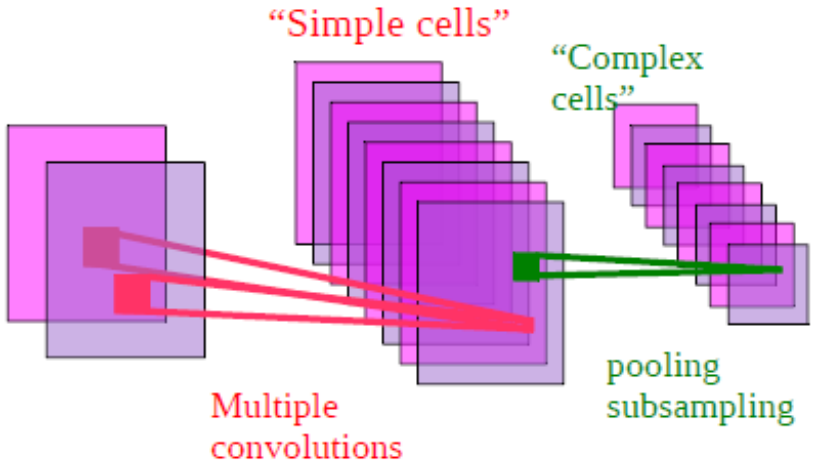
- Expressiveness



Deep learning

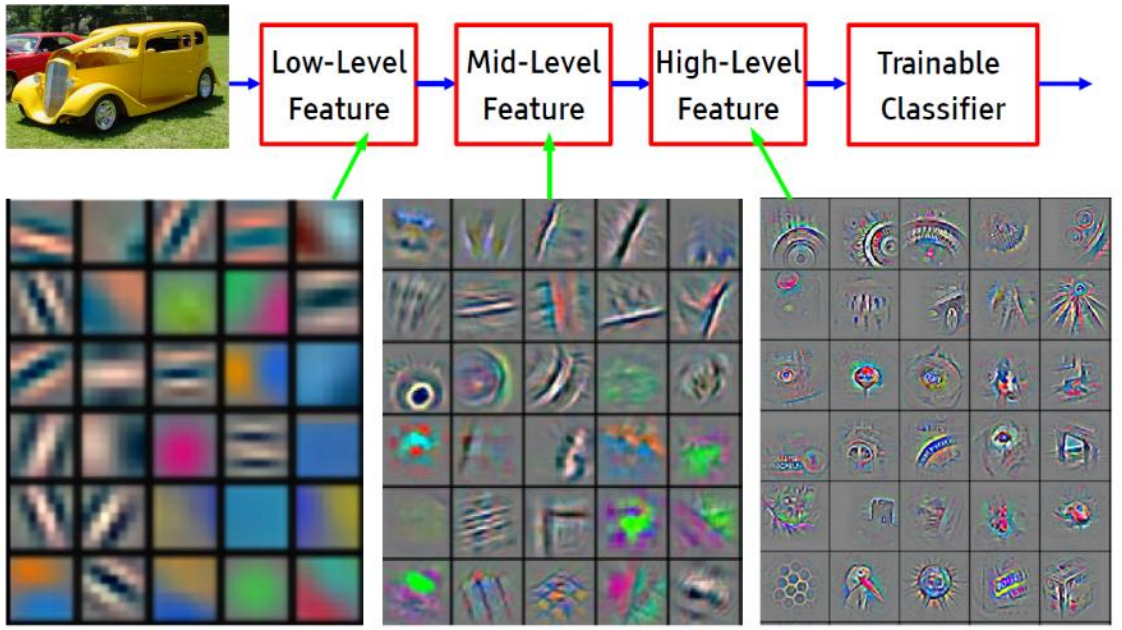
- Convoluted Networks
  - Tensor Kernels
  - Subsampling





- Deep networks
  - Traditional MultiLayer classifier

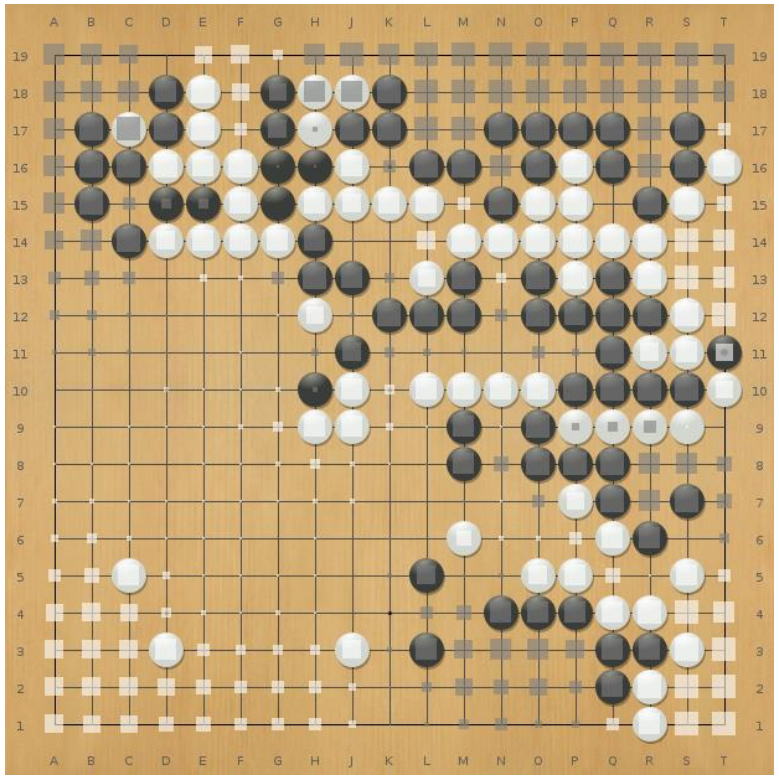
  - Deep Natural Hierarchies



Example: Go

- Game of GO
  - Simple but complex
  - Computer Go
  - 2016/03 - Alpha Go vs Lee Sedol

- Computing value Maps



- Deep learning Toolkits

TensorFlow

Microsoft CNTK

Demo 5

Train and run a convoluted network

Questions?

Please remember to evaluate the session online