
Ako sa roboty učia?



Pavel Petrovič

Katedra aplikovanej informatiky,

Fakulta matematiky, fyziky a informatiky UK

ppetrovic@acm.org

September 2009

Ako sa roboty učia?

Potrebujú sa roboty učiť?

Ako to robia? Ako si ukladajú poznatky?

Majú mozog alebo neurónovú sieť?

Majú radi čokoládu? Športujú?

Je to pravdepodobné?

Existuje evolúcia robotov?

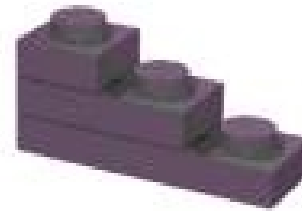
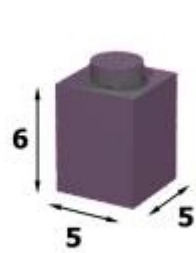
Čo na to hovoria logici?

A prečo roboty?

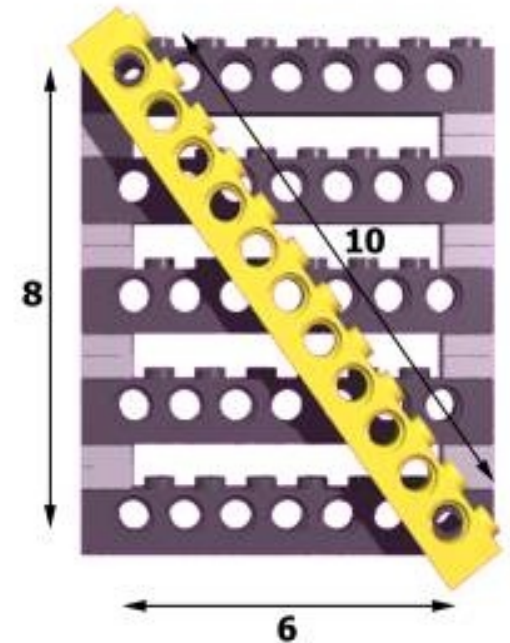
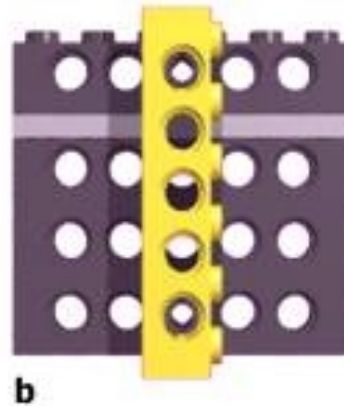
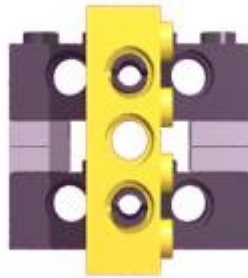
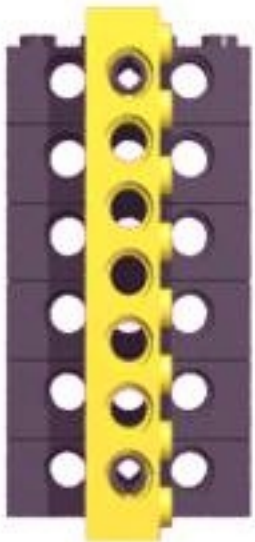
Čo znamená učiť sa?



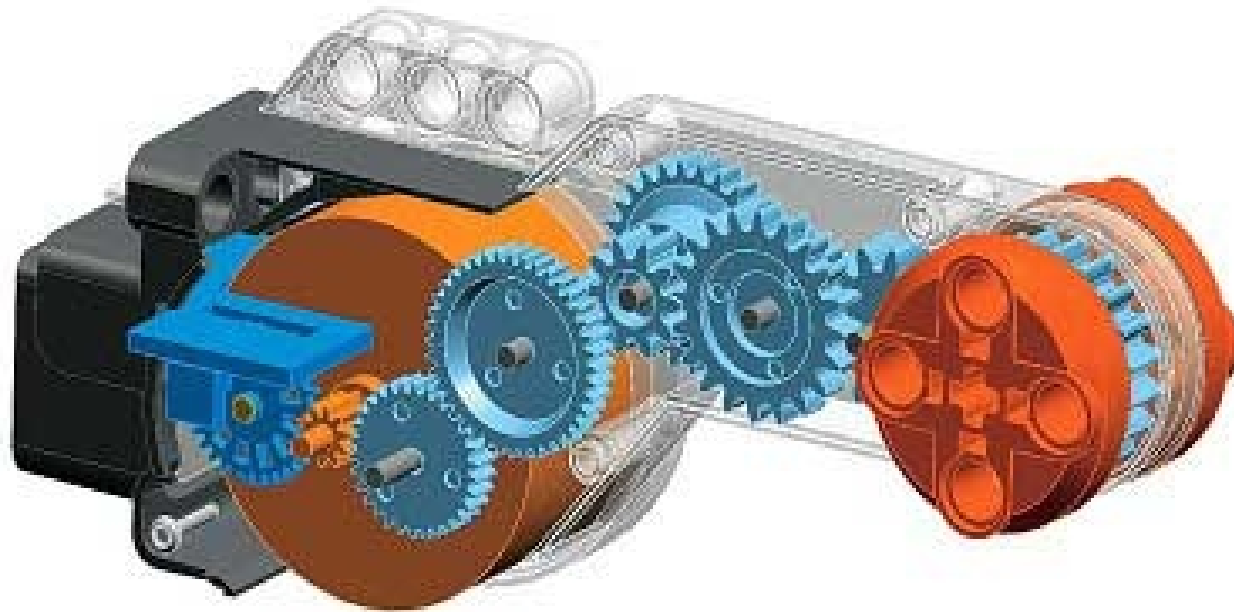
Life is learning... :-)



LEGO Geometry



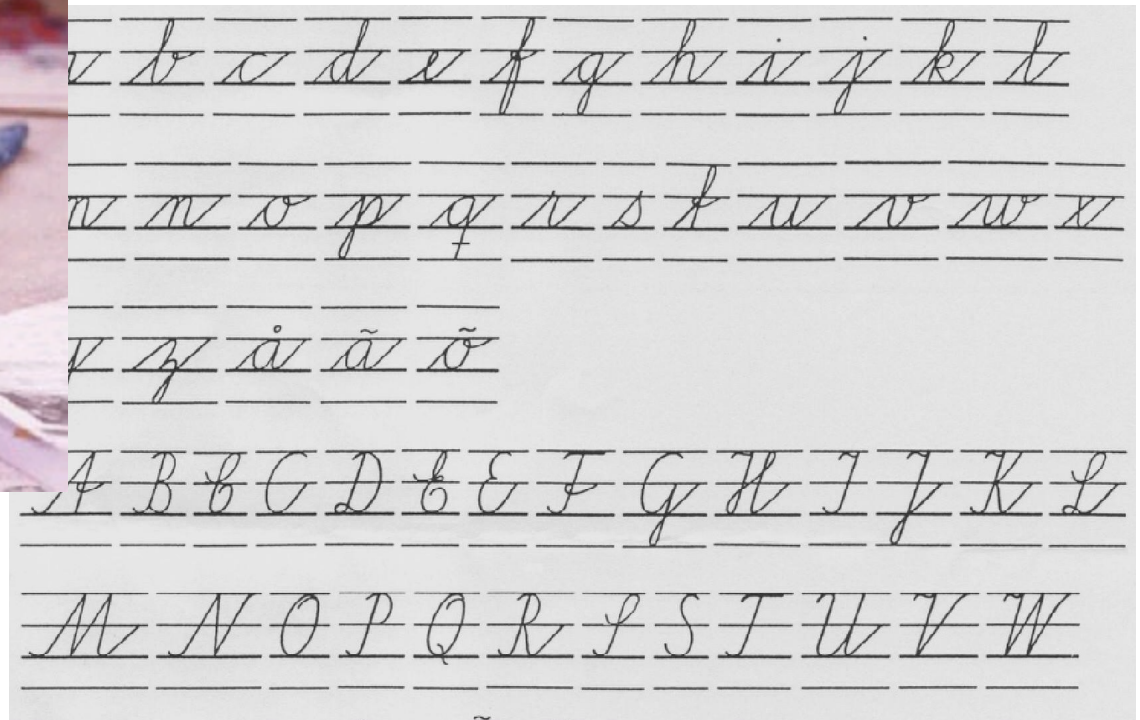
But how does the learning work inside?



Čo znamená učiť sa?



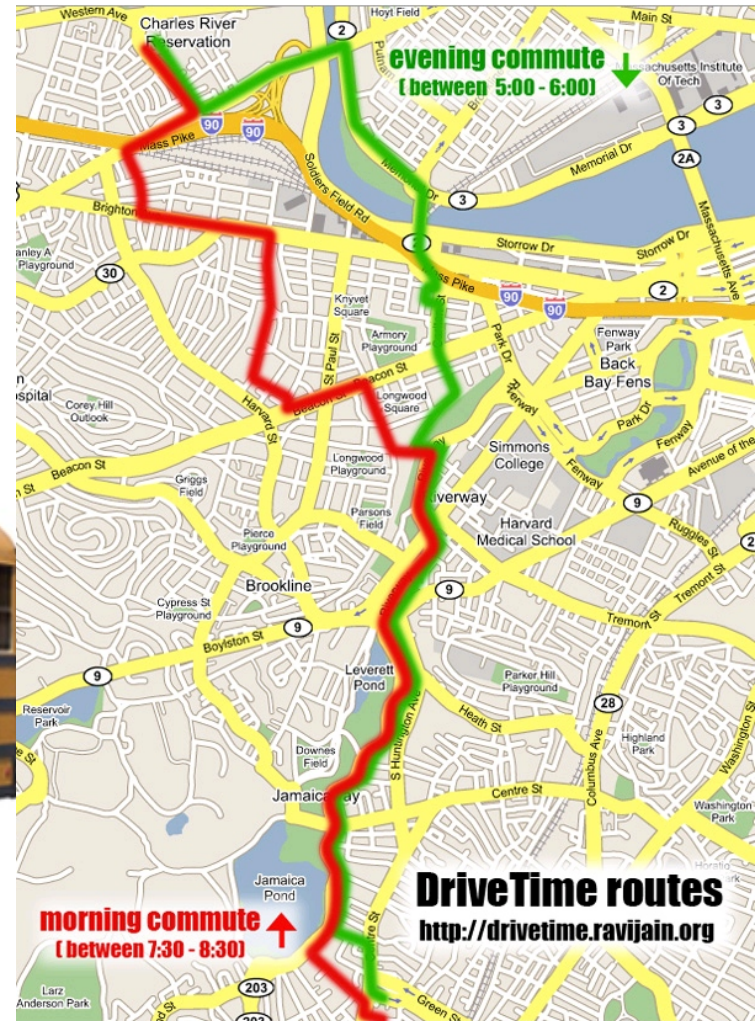
Čo znamená učiť sa?



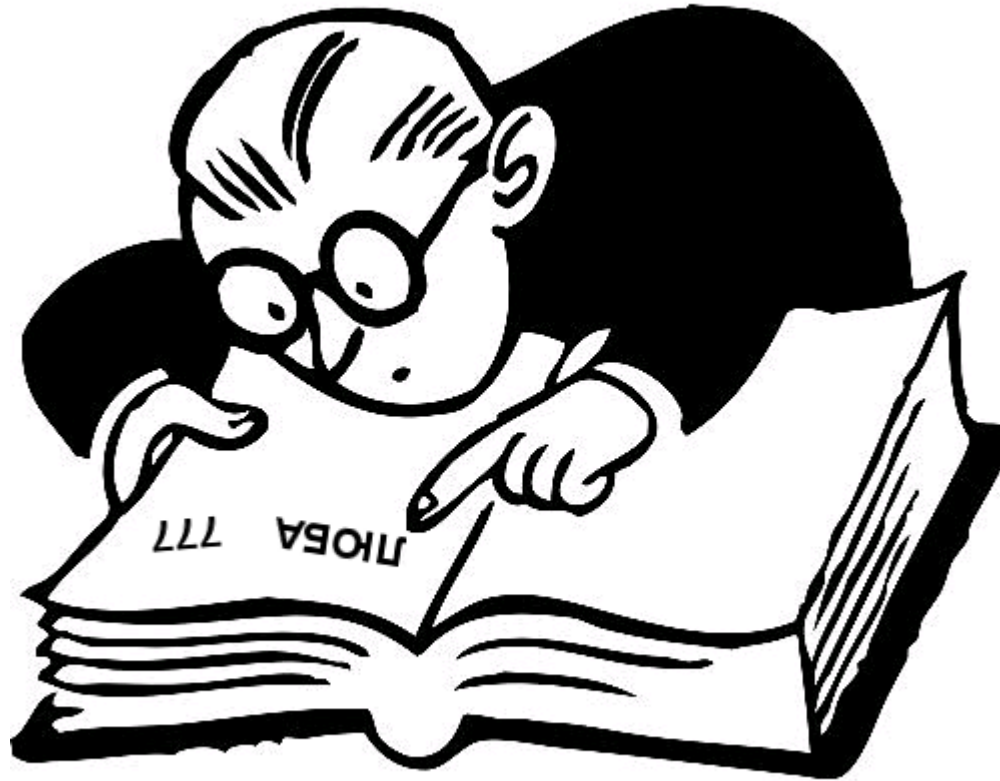
Čo znamená učiť sa?



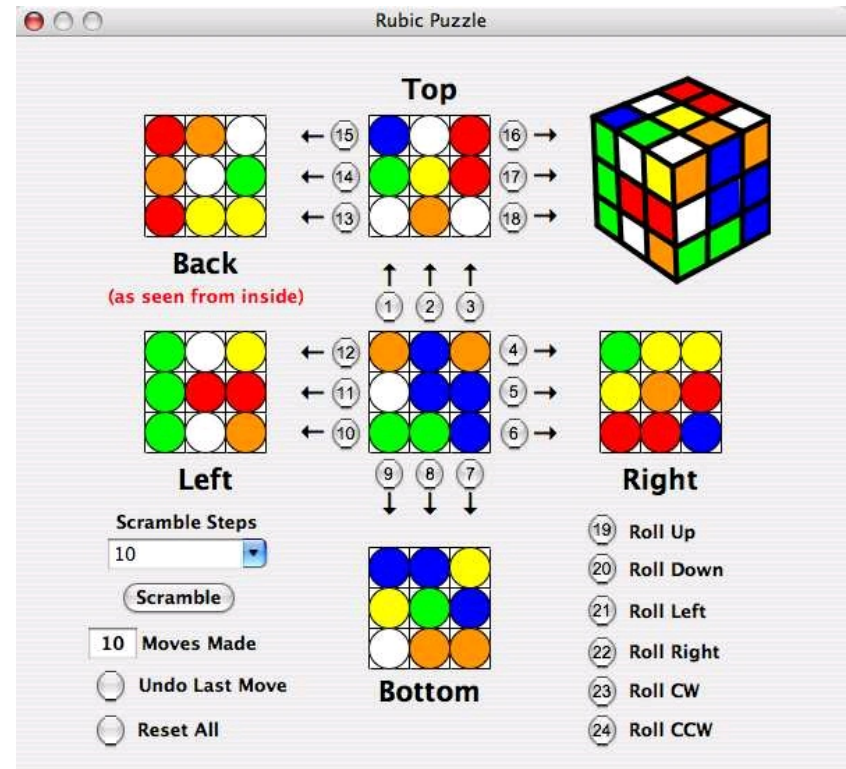
Čo znamená učiť sa?



Čo znamená učiť sa?



Čo znamená učiť sa?



Čo znamená učiť sa?

Krátky slovník slovenského jazyka: *učiť sa*

- 1. cvičením získavať zručnosť: učiť sa plávať, tancovať, hrať na husliach*
- 2. na základe výchovného pôsobenia si osvojovať, vštepovať: učiť sa disciplíny, pravdovravnosti*
- 3. získavať vedomosti: učiť sa dejepis, dobre, ľahko sa učiť; učiť sa naspamäť; čo ste sa dnes (v škole) učili?*
- 4. pripravovať sa na budúce povolanie (obyčajne remeslo): učiť sa za automechanika; hovor. chcel by sa ďalej učiť študovať*

„učiť sa na vlastnej škode“

„učiť sa na vlastných chybách“

Čo znamená učiť sa?

Encyclopedia Britannica: *learning*

the alteration of behaviour as a result of individual experience.

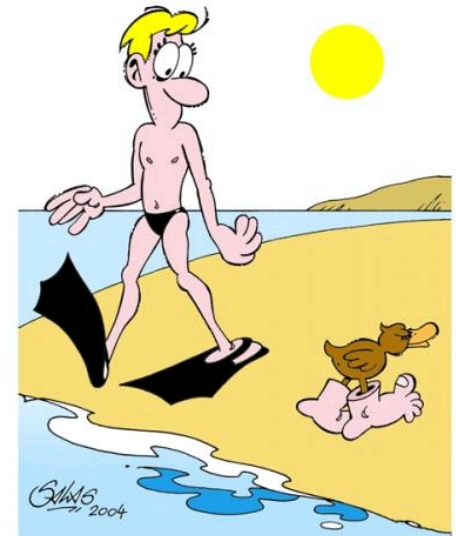
When an organism can perceive and change its behaviour, it is said to learn.

Wikipedia: *learning*

acquiring new knowledge, behaviors, skills, values, preferences or understanding, and may involve synthesizing different types of information. The ability to learn is possessed by humans, animals and some machines.

Učenie x Adaptácia?

- Adaptácia – „malé“ učenie, väčšinou sa týka prispôsobeniu sa fyzikálnym parametrom sveta
- Adaptácia druhov
- Adaptácia – zmena tvaru tela, správania, získavania potravy, životného štýlu
- Evolučná adaptácia
- Učenie sa týka jedincov
- Učenie súvisí väčšinou s kognitívnymi procesmi, zmena nastáva iba v mozgu



Učenie sa robotov

Prečo sa roboty potrebujú učiť?

Bežné roboty používané v kontrolovaných podmienkach na výrobných linkách sa väčšinou učiť nepotrebujú. Môžu sa adaptovať – prispôbiť rôznym vlastnostiam materiálov, optimalizovať svoju prevádzku, môžu byť programovateľné...

Roboty, ktoré plnia úlohy v „našom“ prostredí – v reálnom svete sa **môžu** učiť s cieľom lepšie plniť svoju úlohu.

Vlastnosti prostredia:

Neznáme = nevieme, čo tam očakávať

Dynamické = môže sa meniť

Nepredvídateľné = nedokážeme odhadnúť kedy a ako sa zmení

Učenie sa robotov

Čo sa roboty môžu naučiť?

- Mapu svojho prostredia
- Vlastnosti svojho prostredia
- Rozpoznávať predmety, tváre, ľudí
- Manipulačné úlohy
- Navigačné úlohy
- Spolupracovať s inými robotmi
- Efektívne komunikovať s ľuďmi
- Porozumieť situáciám a voliť v nich správne akcie
- Zložité úlohy

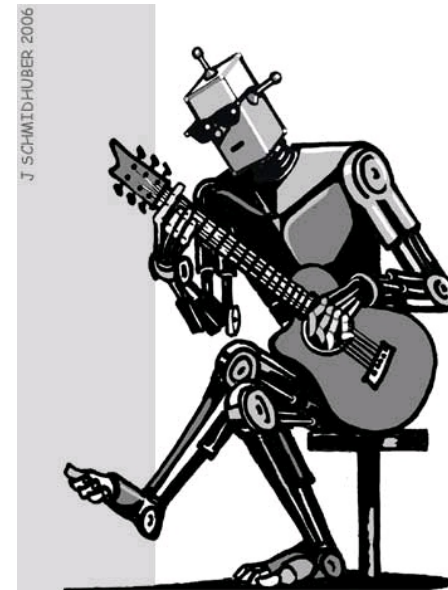
Robot Learning

Ako sa roboty môžu učiť?

Pattern Recognition & Machine Learning

(univerzálny pojem: Umelá inteligencia)

Preskúmame pár príkladov...



COGNITIVE ROBOTICS

Machine Learning – jednoduchý příklad

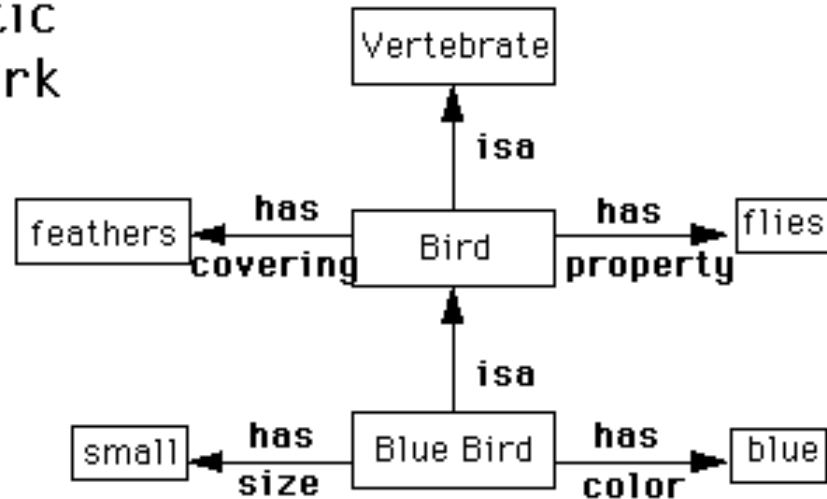
Hra zvier

Počítač	Človek	Počítač	Človek
Je to cicavec?	áno	Je to cicavec?	áno
Žije vo vode?	nie	Žije vo vode?	áno
Je to šelma?	áno	Je to veľryba?	nie
Má prúžky?	áno	Vzdávam sa. Čo je to?	delfín
Is it a tiger?	áno	Zadaj otázku, ktorá odlišuje veľrybu a delfína:	Je obrovský?
I won!		Pre delfína je odpoveď na túto otázku:	nie

ML: Reprezentácia poznatkov

Symbolická reprezentácia poznatkov

Sematic
Network



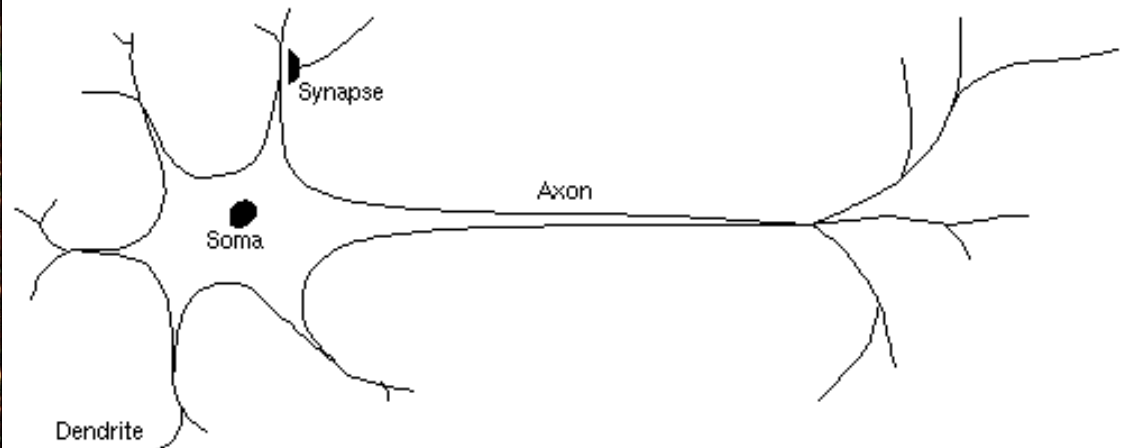
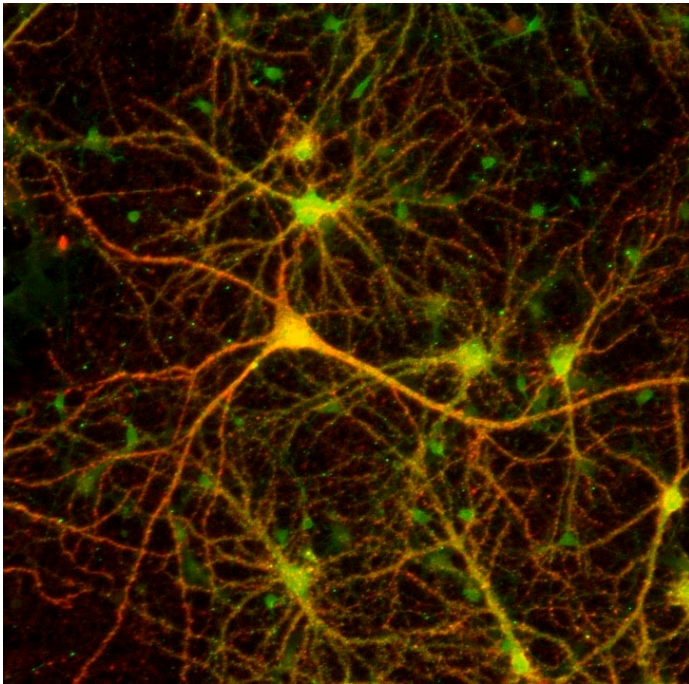
Reprezentácia poznatkov: LISP expressions
Učiaci algoritmus: predikátová logika

Symbolická reprezentácia poznatkov

GENERAL TRIANGLE {
(Generalized_by: closed planar geometric object)
(Generalization_of: acute-angled triangle, obtuse-angled triangle, right-angled triangle, equilateral triangle, isoscales-triangle)
(parameters: x-side, y-side, z-side, φ -angle, χ -angle, ψ -angle, x-altitude, y-altitude, z-altitude, x-median, y-median, z-median, r_inner_circle_radius, R_outer_circle_radius, P_perimeter=x+y+z, V_volume=(x*x-altitude)/2)
(number of sides [*<cardinality:1>* *<data type:INT>*] value: 3)
(number of angles [*<cardinality:1>* *<data type:INT>*] value: 3)
(x-side [*<cardinality:1>* *<data type:REAL>* *<if-needed: ask, measure, infer>* *<ifchanged: check consistency (x<y+z)>*] length value: UNKNOWN)
(y-side, z-side *similarly*)
(φ -angle [*<cardinality:1>* *<data type:REAL >* *<data template: .** , 0< φ <180 >* *<if-needed: ask, measure, infer>**<if-changed: check_consistency ($\varphi+\chi +\psi=180$)>*] value: UNKNOWN)
(χ -angle, ψ -angle *similarly*)
(x-altitude [*<cardinality:1>* *<data type:REAL >* *<data template: .**>* *<if-needed: ask, measure, infer>* *<if-changed: check_consistency>*] value: UNKNOWN)
(y-altitude, z-altitude *similarly*)
(x-meridian [*<cardinality:1>* *<data type:REAL >* *<data template: .**>* *<if-needed: ask, measure, infer>* *<if-changed: check_consistency>*] value: UNKNOWN)
(y-meridian, z-median *similarly*)
(P_perimeter [*<cardinality:1>* *<data type:REAL >* *<if-needed: ask, measure, infer_by: P = x+y+z>*] value: UNKNOWN)
(V_volume [*<cardinality:1>* *<data type: REAL>* *<data template: .**>* *<if-needed: ask, infer>*] value: $\sqrt{[(p/2)(p-x)(p-y)(p-z)](x*x-altitude)/2}$ }

Sub-symbolické reprezentácie

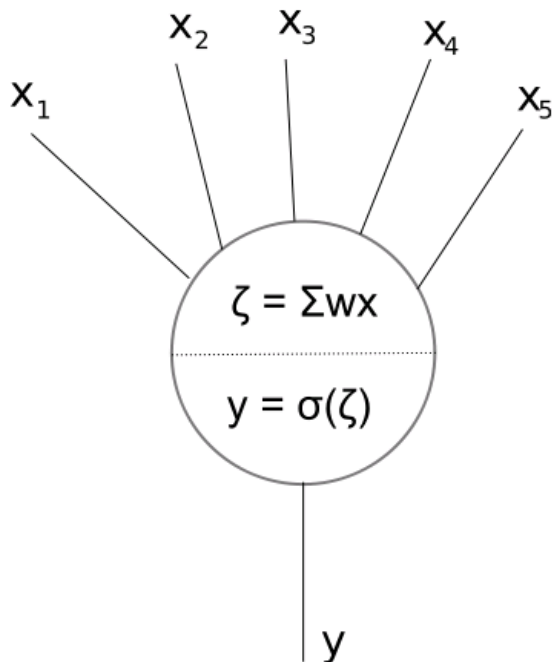
Takto to robí príroda:



*Informácia je **distribúovaná**, representovaná miliónmi číselných hodnôt, ktoré plnia mnoho rolí/významov...*

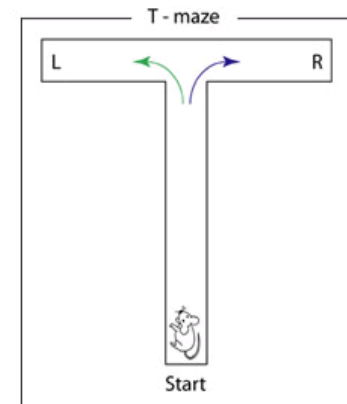
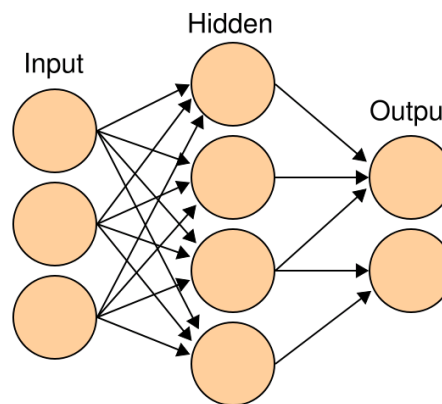
Sub-symbolická reprezentácia

Connectionists: Umelé neurónové siete môžu uchovávať poznatky, učiť sa, usudzovať, generovať akcie

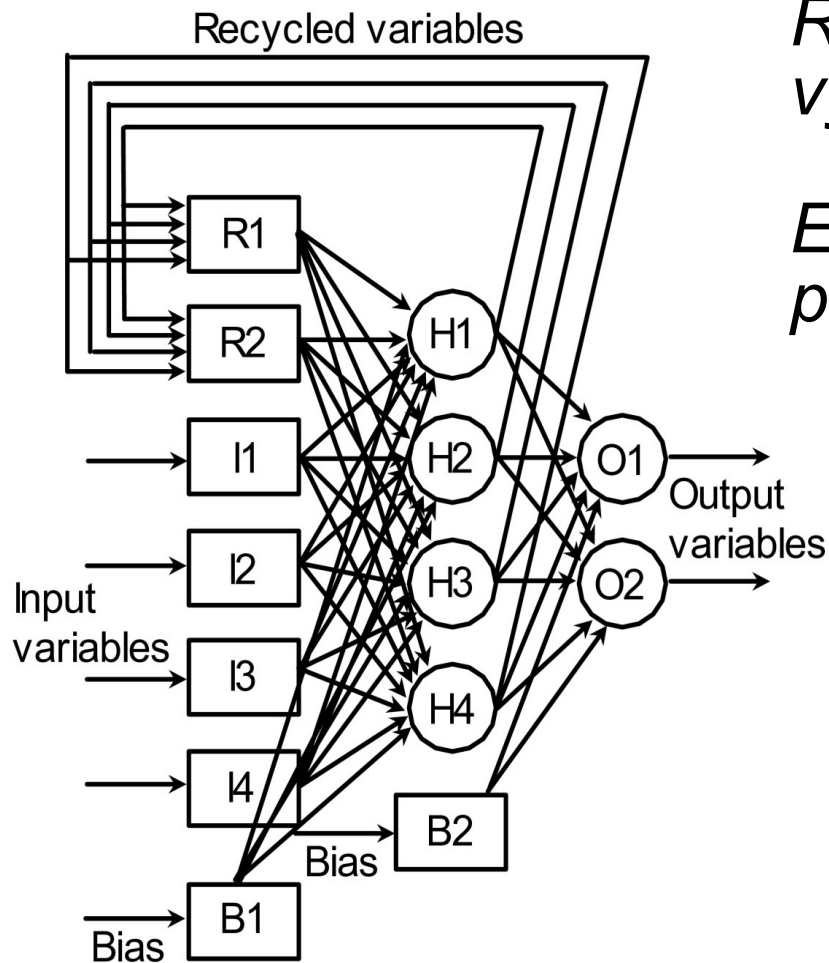


V robotike: Sensoricko-motorické systémy

Reaktívne systémy vs. Interný stav

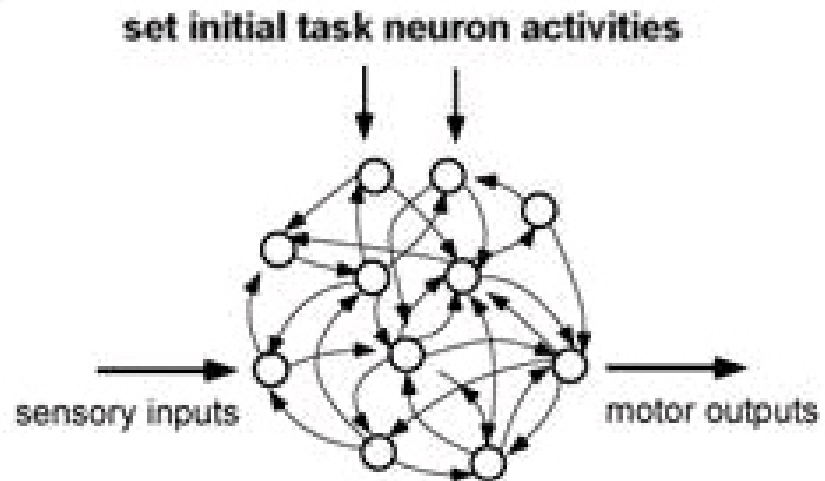


Sub-symbolická reprezentácia



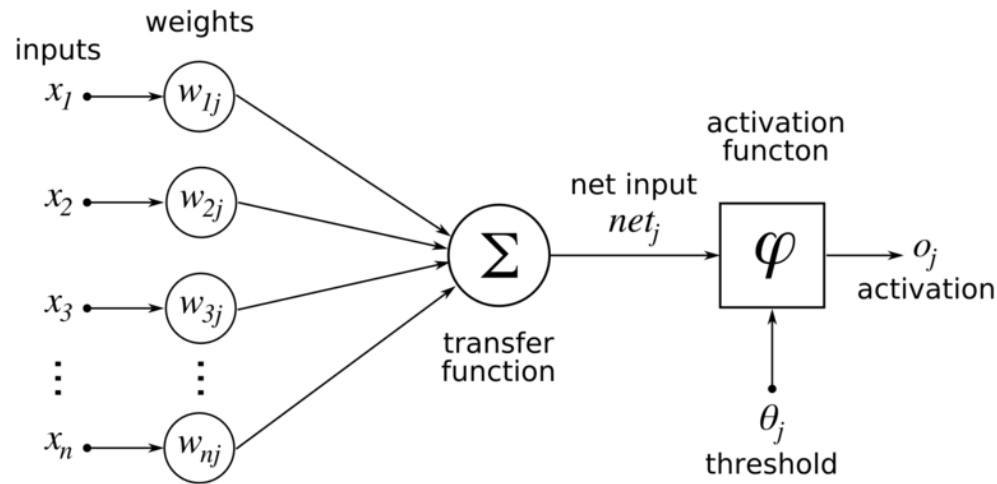
RNN môžu vypočítať ľubovoľnú vypočítateľnú funkciu

Elmanovské alebo s úplným prepojením uzlov

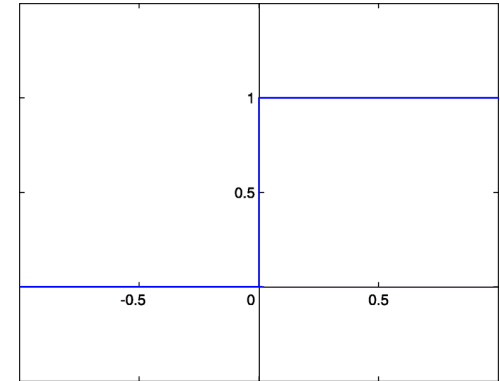


Sub-symbolická reprezentácia

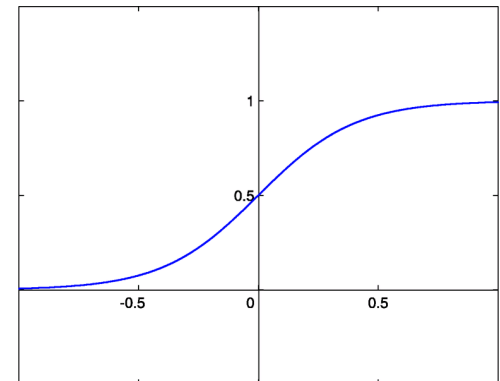
Čo dokáže uchovať jeden perceptrón?



1	0	1	1	1	1	1	1	0	1	1	0
0	0	0	0	0	1	0	1	0	0	0	1
AND	0	1	OR	0	1	NOT	0	1	XOR	0	1



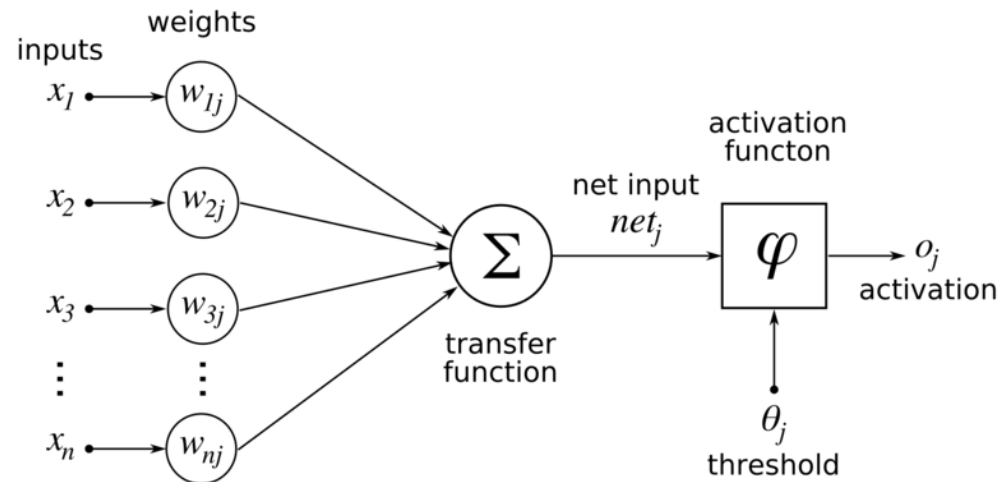
$$\sigma(t) = \frac{1}{1 + e^{-\beta t}}$$



Sub-symbolická reprezentácia

Čo dokáže uchovať jeden perceptrón?

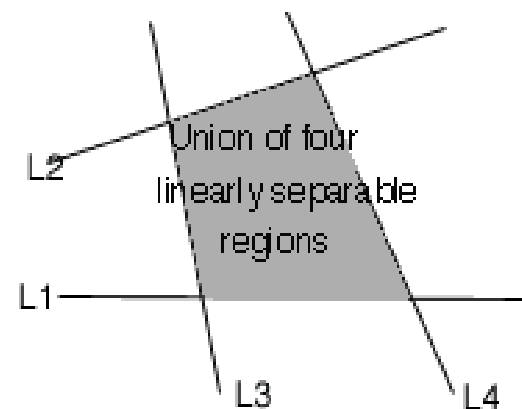
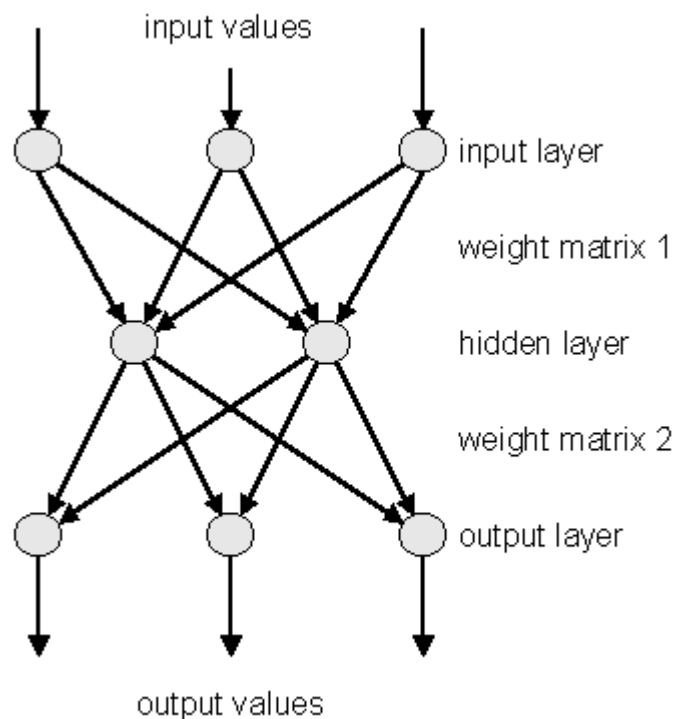
1	1	0
0	0	1
<hr/>		
XOR	0	1



1969: Minsky, Papert: Perceptrons

Sub-symbolická reprezentácia

Riešenie – viacvrstvový perceptrón



klasifikácia

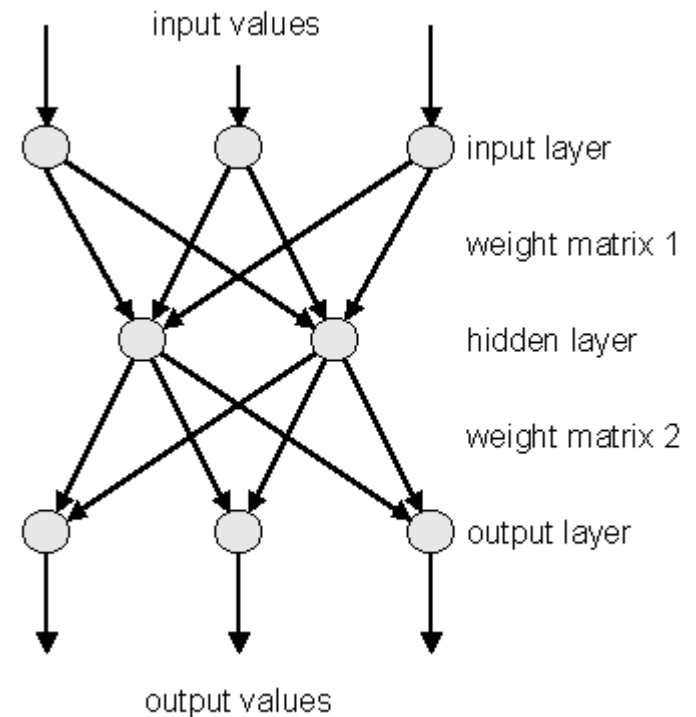
Sub-symbolická reprezentácia

Spôsob učenia?

Príklad: Backpropagation

(algoritmus spätného šírenia chýb)

Každý uzol prispieva k výslednej chybe siete určitou mierou, algoritmus upravuje váhy medzi uzlami úmerne k tejto miere.



Sub-symbolická reprezentácia

\vec{x}_j = vstupný vektor uzla j (x_{ji} = i -ty vstup j -teho uzla)

\vec{w}_j = váhový vektor uzla j (w_{ji} = váha priradená x_{ji})

$z_j = \vec{w}_j \cdot \vec{x}_j$ = váhovaná suma vstupom uzla j

o_j = výstup uzla j

t_j = správny (očakávaný) výstup uzla j

Chceme určiť $\frac{\partial E}{\partial w_{ji}}$ pre každú vstupnú váhu w_{ji} pre každý výstupný uzol j . Keďže z_j je funkcia w_{ji} nezávisle od miesta, kde sa uzol j nachádza:

$$\begin{aligned} \frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ji}} \\ &= \frac{\partial E}{\partial z_j} x_{ji} \end{aligned} \qquad \frac{\partial E}{\partial z_j} = \delta_j$$

Sub-symbolická reprezentácia

Výstupné uzly:

$$E = \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - \sigma(z_k))^2$$
$$\delta_j = \frac{\partial E}{\partial z_j} = \frac{\partial}{\partial z_j} \frac{1}{2} (t_j - o_j)^2$$
$$= -(t_j - o_j) \frac{\partial o_j}{\partial z_j}$$
$$= -(t_j - o_j) \frac{\partial}{\partial z_j} \sigma(z_j)$$
$$= -(t_j - o_j) (1 - \sigma(z_j)) \sigma(z_j)$$
$$= -(t_j - o_j) (1 - o_j) o_j$$

Pravidlo zmeny váh:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \delta_j x_{ji}$$

Sub-symbolická reprezentácia

Skryté uzly:

$$\begin{aligned}\frac{\partial E}{\partial w_{ji}} &= \sum_{k \in \text{Downstream}(j)} \frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ji}} \\ &= \sum_{k \in \text{Downstream}(j)} \frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial z_j} \cdot x_{ji}\end{aligned}$$

$$\begin{aligned}\delta_j &= \sum_{k \in \text{Downstream}(j)} \frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial z_j} \\ &= \sum_{k \in \text{Downstream}(j)} \delta_k w_{kj} o_j (1 - o_j)\end{aligned}$$

$$\delta_j = o_j (1 - o_j) \sum_{k \in \text{Downstream}(j)} \delta_k w_{kj}$$

ALVINN: Autonomous Land Vehicle In a Neural Network

- Dean Pomerleau's Ph.D. thesis (1992).
- How ALVINN Works
 - Architecture
 - Training Procedure
 - Performance
- Why ALVINN Works
 - Hidden Unit Analysis
- Integrating Multiple Networks

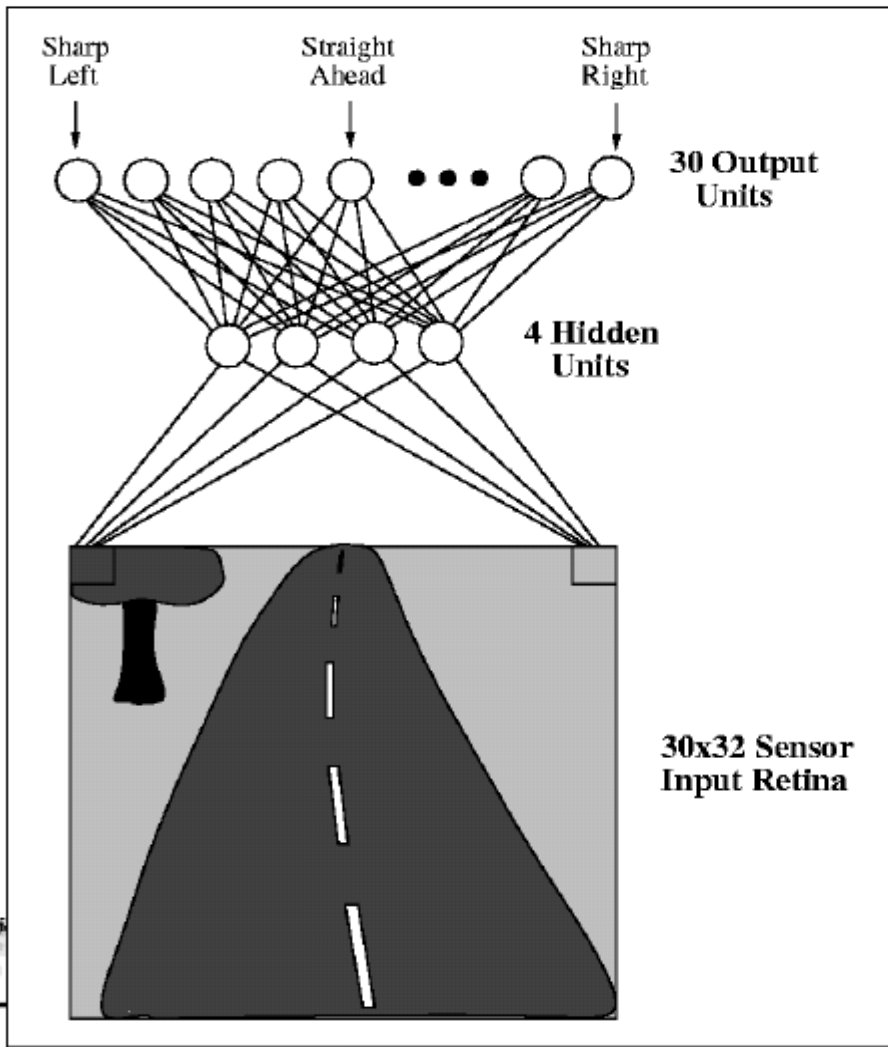


Other Applications

Interactive Systems Labs



ALVINN Network Architecture



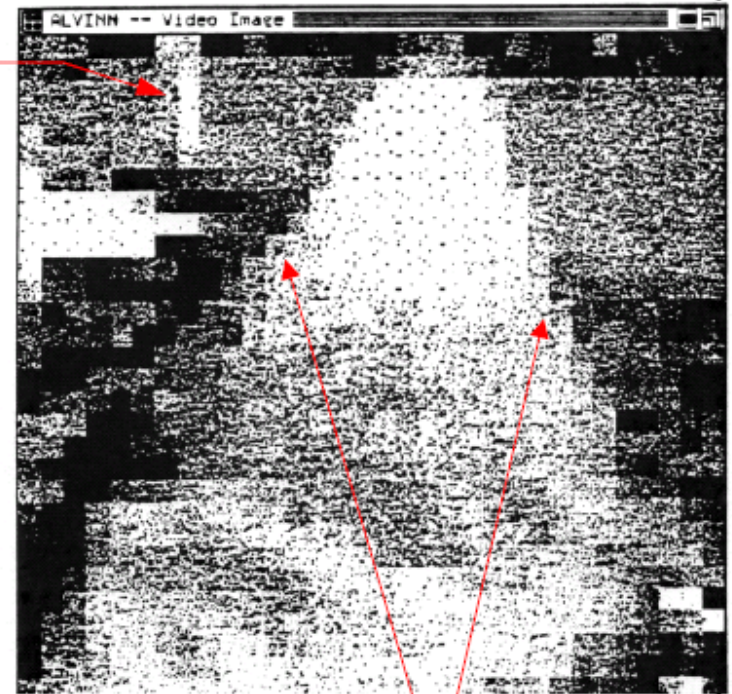
How many inputs?
 $30 \times 32 = 960$

How many weights?
 $961 \times 4 + 5 \times 30 = 3994$



Original Training Scheme

- Generate artificial road images mimicing situations the network is expected to encounter, including noise.
- Calculate correct steering direction for each image.
- Train on artificial images, then test on real roads.
- Problem: realistic training images are difficult to produce: training is expensive.



tree

road edges

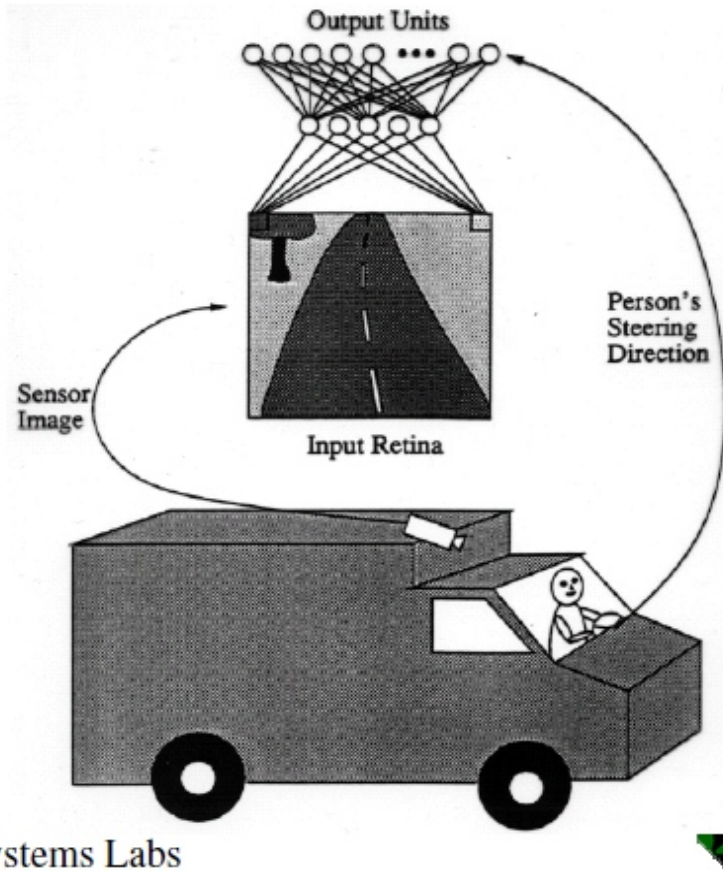


Interactive Systems Labs



Training on the Fly

- Digitize the steering wheel position.
- Train the network by having it observe live sensor data as a human drives the vehicle.
- The human “teaches” the network how to drive.
- Can this really work?
 - It's not so simple...

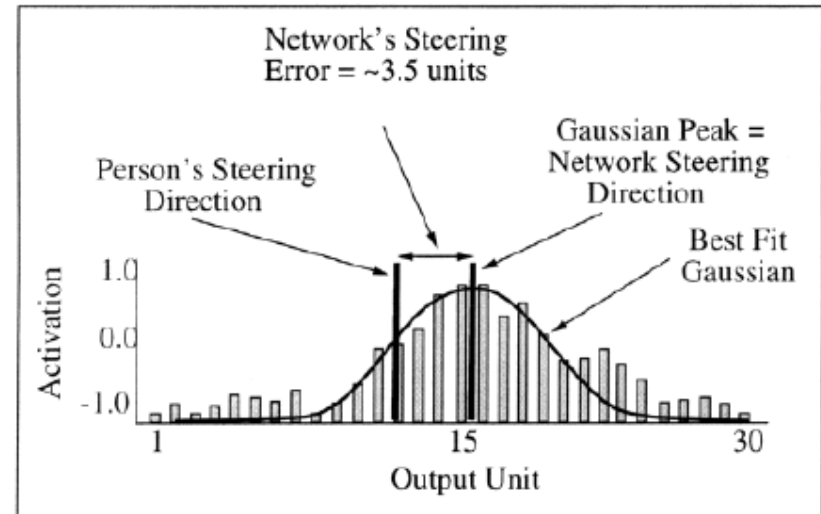


Interactive Systems Labs



Measuring Steering Error

- Train with a Gaussian bump centered over the desired steering direction.
- To test: fit a Gaussian to the network's output vector.
- Measure distance between Gaussian's peak and human steering direction.



Why use a Gaussian for the output pattern?



Interactive Systems Labs



Learning to Correct Steering Errors

- If the human drives perfectly, the network never learns to make corrections when it drifts off the desired track.
- Crude solution:
 - Turn learning off temporarily, and drive off course.
 - Turn learning back on, and let the network observe the human making the necessary corrections.
 - Repeat.

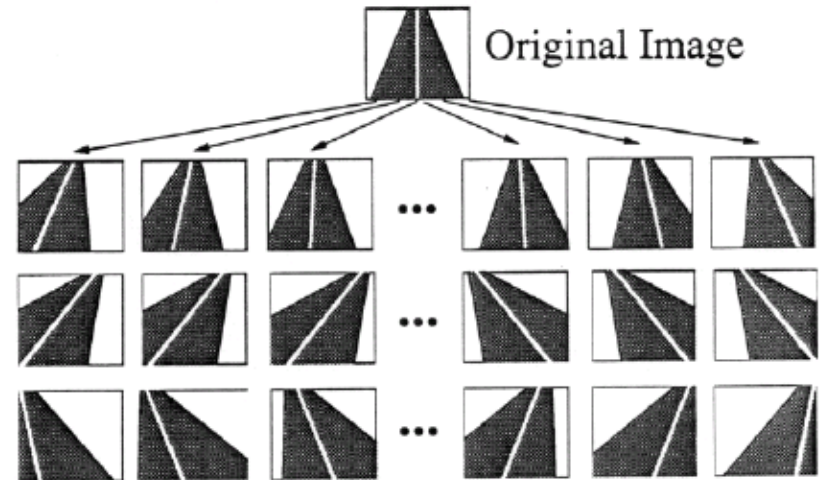


- Relies on the human driver to generate a rich set of steering errors: time consuming and unreliable.
Can be dangerous if training in traffic.



Simulating the Steering Errors

- Let humans drive as best they can.
- Increase training set variety by *artificially* shifting and rotating the video images, so that the vehicle appears at different orientations relative to the road.

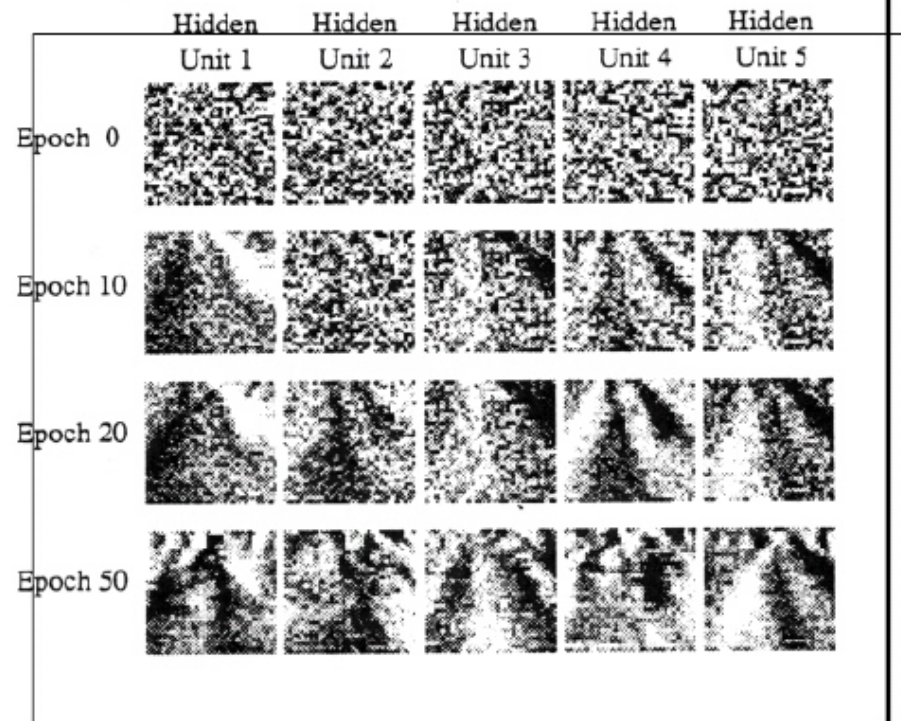


- **Generate 14 random shift/rotations for each image.**
- A simple steering model is used to predict how a human driver would react to each transformation.



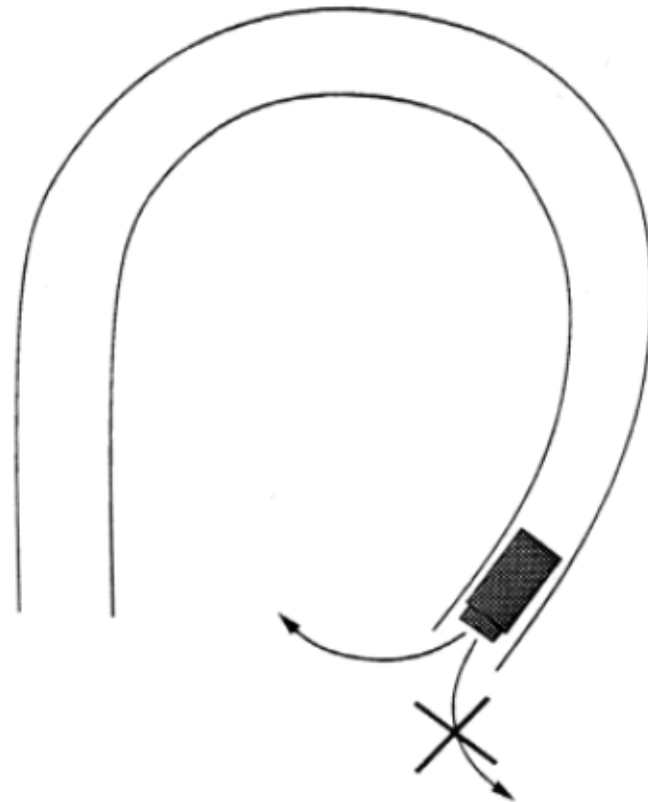
Network Weights Evolving

- Initial random weights look like “salt and pepper” noise.
- During training, the hidden units evolve into a set of complementary feature detectors.



Problem with Online Learning: Network Can “Forget”

- The network tends to overlearn recently encountered examples and forget how to drive in situations encountered earlier in training.
- After a long right turn, the network will be biased toward turning right, since recent training data focused on right turns.



Interactive Systems Labs

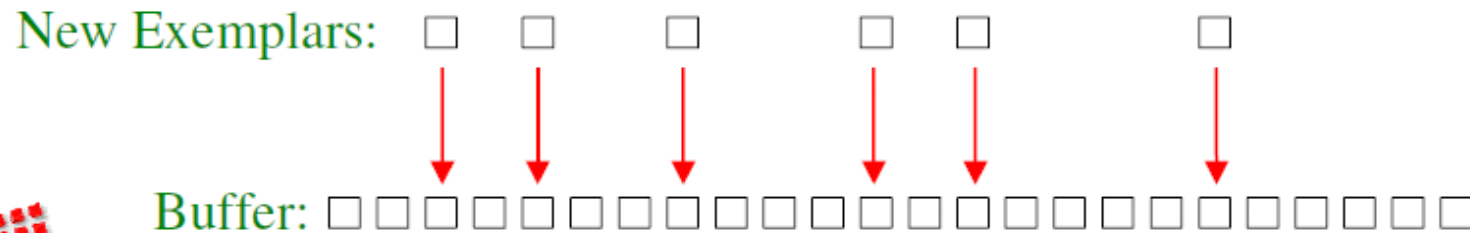


Solution: Maintain a Buffer of Balanced Training Images

This is a semi-batch learning approach. Keep a buffer of 200 training images.

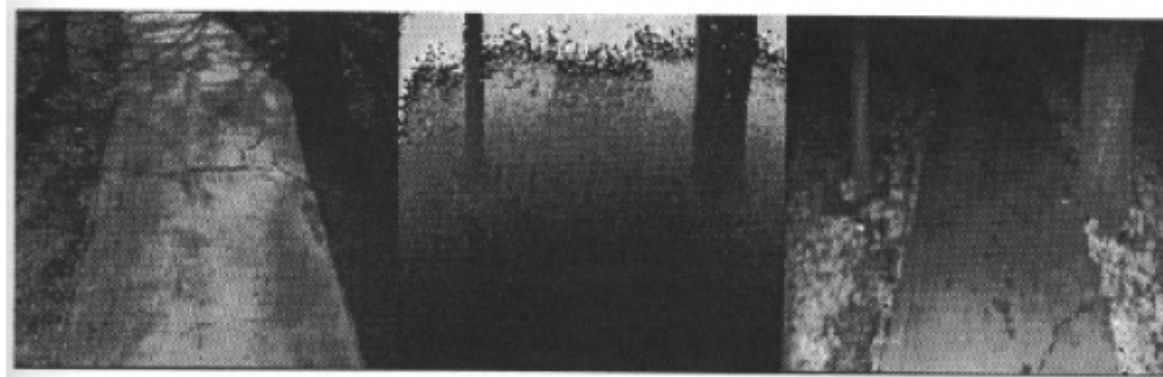
Replace 15 old exemplars with new ones derived from the current camera image. Replacement strategies:

- (1) Replace the image with the lowest error
- (2) Replace the image with the closest steering direction



Multi-Modal Inputs

- ALVINN can avoid obstacles using a laser rangefinder. It can drive at night using laser reflectance imaging.



**Regular
Video**

**Laser
Rangefinder**

**Laser
Reflectance**



Interactive Systems Labs



Comparison with the “Traditional Approach”

1) Determine which image features are important, e.g., a yellow stripe down the center of the road.

ALVINN finds the important features itself.

2) Hand-code algorithms to find the important features, e.g., edge detection to find yellow lines.

ALVINN constructs its own feature detectors.

3) Hand-code algorithm to determine steering direction based on feature positions in the image.

ALVINN learns the mapping from feature detector outputs to steering direction.



Interactive Systems Labs

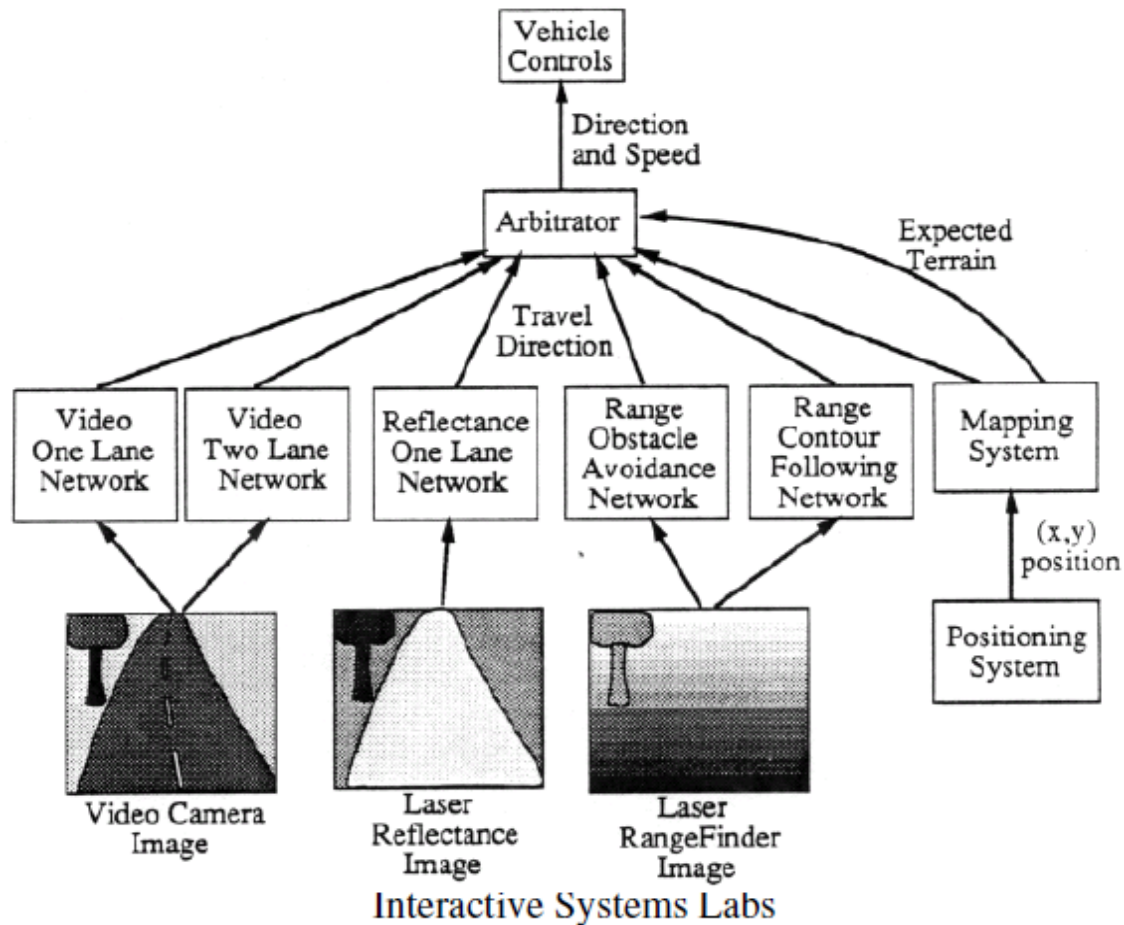


ALVINN's Shortcomings

- The single-network ALVINN architecture can only drive on one type of road (unpaved, single-lane, double-lane, lane-striped, etc.)
- Can't transition from one road type to another.
- Can't follow a route.
- Solution: rule-based multi-network integration.

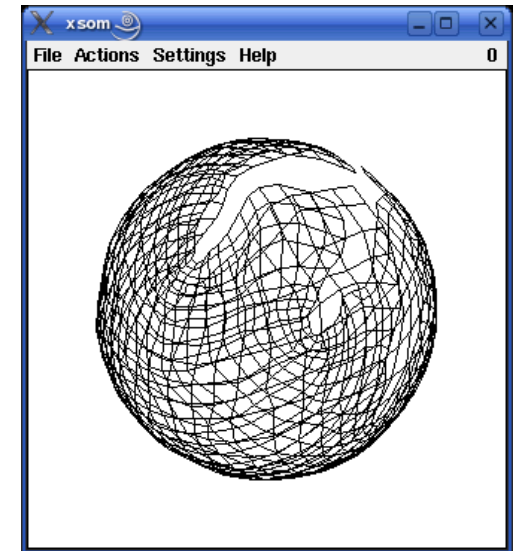
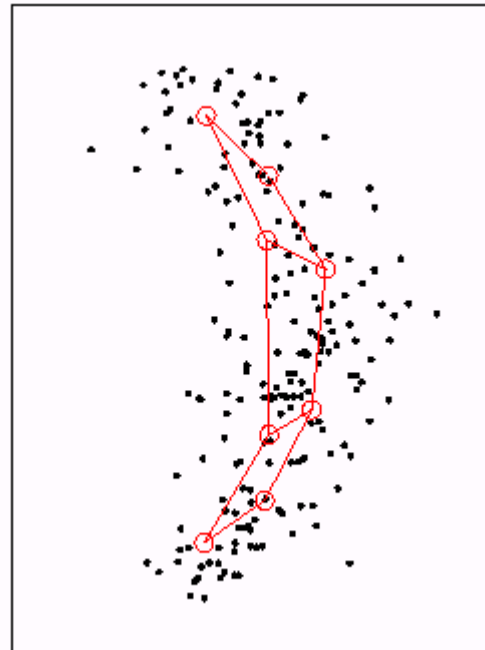
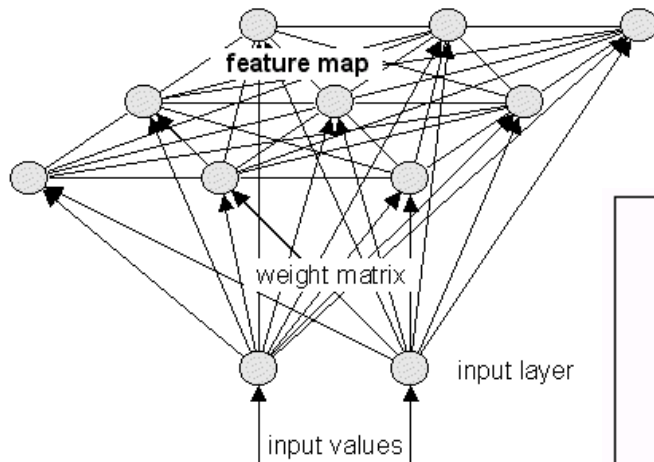


Hybrid ALVINN Architecture



Iné druhy umelých sietí

Klustrovanie, topologické mapovanie...



Homunculus

Sú umelé neurónové siete univerzálne?

Druhy učenia:

Učenie s učiteľom (supervised learning)

Učenie bez učiteľa (unsupervised learning)

Učenie odmenou a trestom (reinforcement learning)

Povaha dát – sensory

- Informácie, ktoré robot získava z reálneho sveta majú celkom iný charakter ako diskkrétne údaje uložené v počítači: senzory poskytujú zašumené dáta a algoritmy sa s nimi musia popasovať!
- Sensory nikdy neposkytujú úplnú informáciu o stave prostredia – iba merajú niektoré fyzikálne veličiny s obmedzenou presnosťou a spoľahlivosťou
- Informácia zo senzorov sa získava draho: stojí čas a iné zdroje, nie je kedykoľvek k dispozícii

Prečo pravdepodobnosť?

- V reálnom prostredí robot musí narábať s neurčitou a nepresnosťou
- svojich akcií
- meraní senzorov
- Presnosť a správnosť modelu, ktorý si robot o svojom prostredí vytvára sú kľúčové pre úspešnosť jeho misie
- Všetky dáta, ktoré sú k dispozícii je treba použiť (alebo aspoň zvážiť)
- Mnoho dát má povahu pravdepodobnosti

Aká pravdepodobnosť?

- Vlastnosť senzorov
- Presnosť senzorov
- Prešmykovanie kolies
- Obmedzená presnosť riadenia motorov
- Presnosť výroby kolies
- Zmena chovania v závislosti teploty a pod.

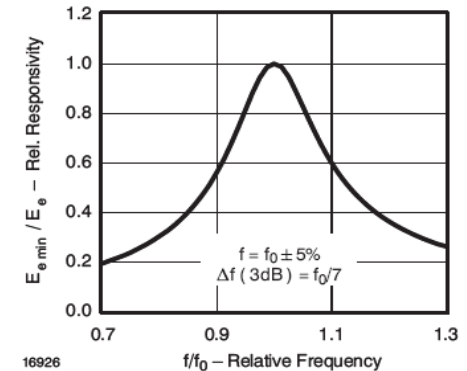
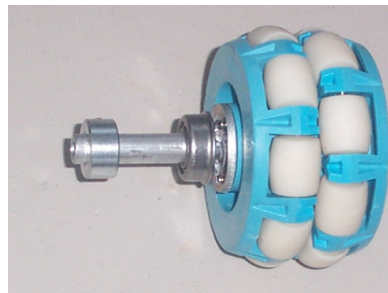


Figure 5. Frequency Dependence of Responsivity

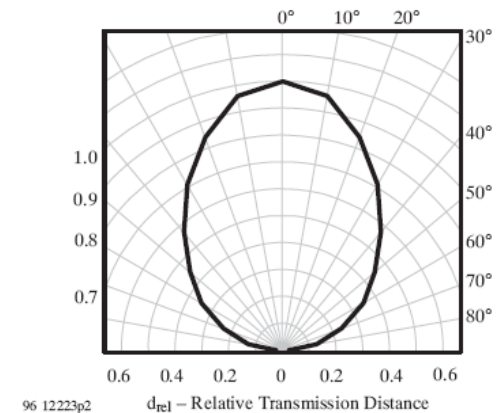


Figure 12. Directivity

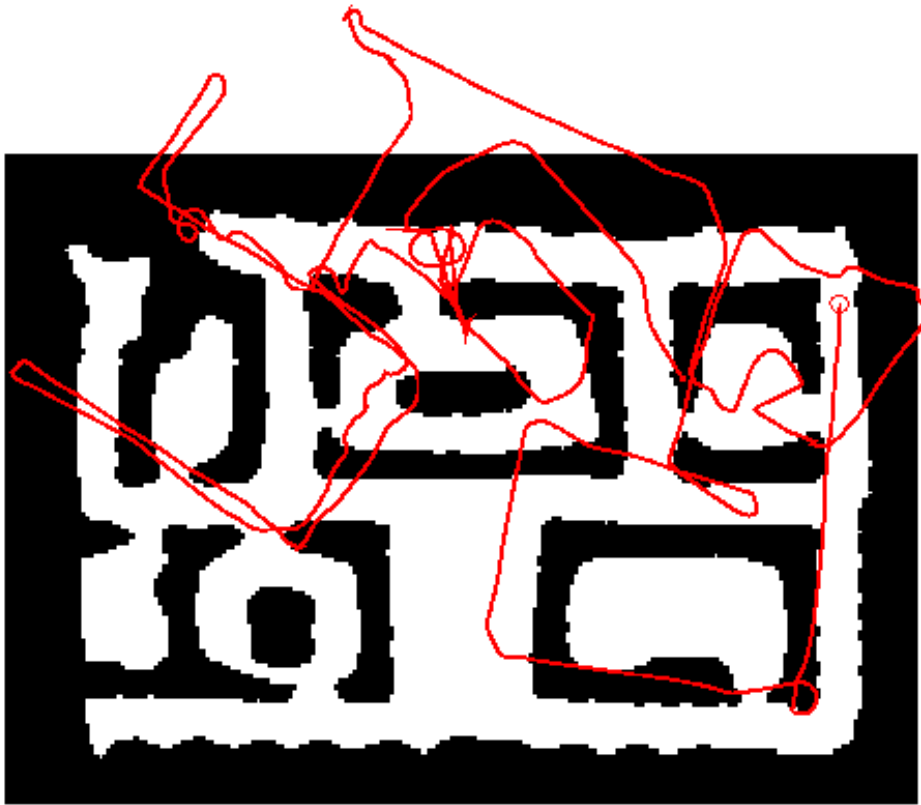
Aké pravdepodobnosti

- Tieto nepresnosti môžu byť zamerané a modelované pomocou rozdelenia náhodnej premennej (štatistika)
- Jedno čítanie zo senzora obsahuje viac informácie, ak poznáme pravdepodobnostné rozdelenie pre správanie sa tohto senzora
- Robot si nemôže dovoliť túto informáciu nevyužiť!

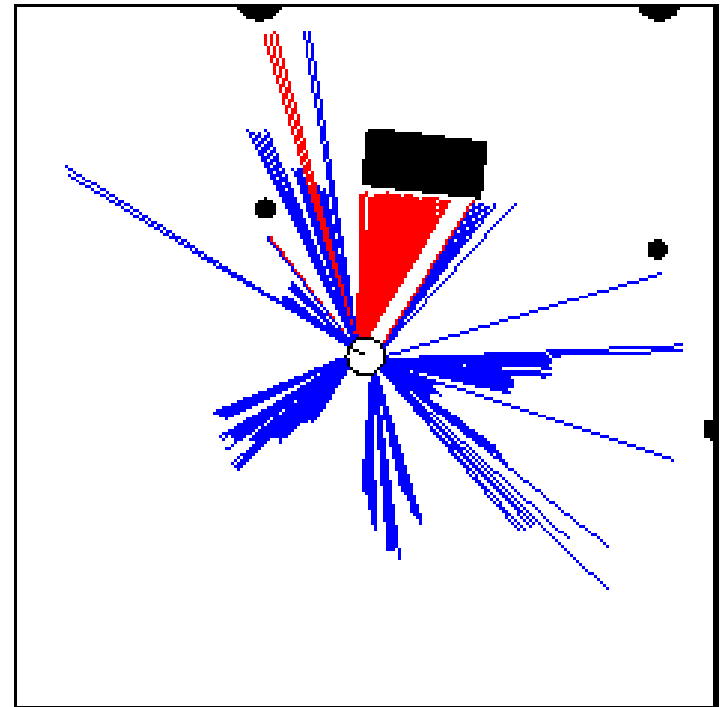
Aké pravdepodobnosti?

- Zložitejšie príklady:
- Poloha a orientácia robota (*robot pose*)
- Mapa prostredia
- Plánovanie a riadenie
- Výber akcií
- Usudzovanie...

Povaha dát



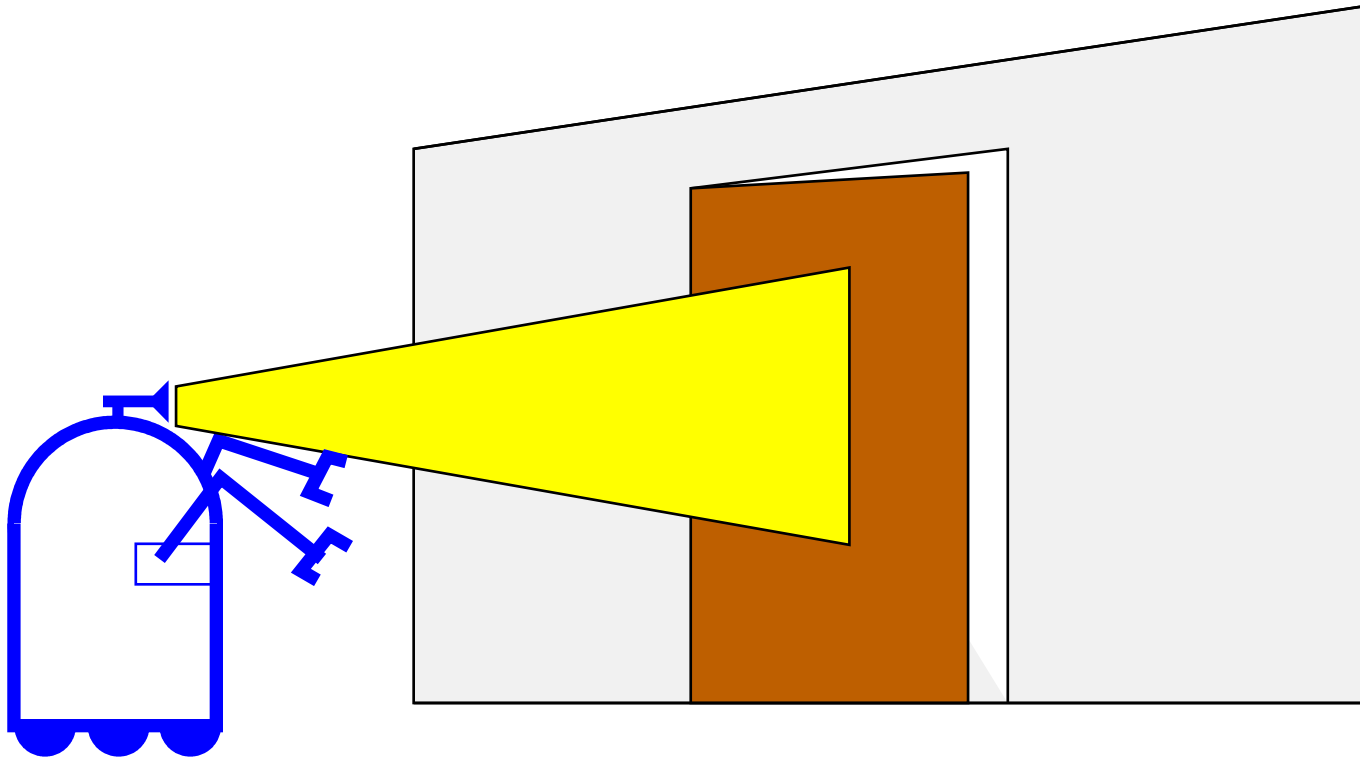
Odometry Data



Range Data

Príklad pravdepodobnostného merania

- Robot pomocou senzoru namerá z
- Aká je $P(open|z)$?



Kauzálna vs. Diagnostická znalosť

- $P(open|z)$ je **diagnostická**
- $P(z|open)$ je **kauzálna**
- Často získavame **kauzálnu** znalosť jednoduchšie
- Bayesove pravidlo u **spočítame početnosť!** znalosť:

$$P(open | z) = \frac{P(z | open)P(open)}{P(z)}$$

Príklad

- $P(z|open) = 0.6$ $P(z|\neg open) = 0.3$
- $P(open) = P(\neg open) = 0.5$

$$P(open | z) = \frac{P(z | open)P(open)}{P(z | open)p(open) + P(z | \neg open)p(\neg open)}$$

$$P(open | z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

- z zvýšilo pravdepodobnosť, že dvere sú otvorené

Kombinácia Pozorovaní

- Robot vykoná druhé meranie z_2
- Ako kombinovať túto informáciu s predchádzajúcou?
- Vo všeobecnosti, dá sa kombinovať N meraní $P(x | z_1 \dots z_n)$?

Rekurzívne Bayesovské Pravidlo

$$P(x | z_1, \dots, z_n) = \frac{P(z_n | x, z_1, \dots, z_{n-1}) P(x | z_1, \dots, z_{n-1})}{P(z_n | z_1, \dots, z_{n-1})}$$

Markovovský predpoklad: z_n nezávisí na z_1, \dots, z_{n-1} ak poznáme x .

$$\begin{aligned} P(x | z_1, \dots, z_n) &= \frac{P(z_n | x) P(x | z_1, \dots, z_{n-1})}{P(z_n | z_1, \dots, z_{n-1})} \\ &= \eta P(z_n | x) P(x | z_1, \dots, z_{n-1}) \\ &= \eta_{1\dots n} \prod_{i=1\dots n} P(z_i | x) P(x) \end{aligned}$$

Príklad: Druhé meranie

- $P(z_2|open) = 0.5$ $P(z_2|\neg open) = 0.6$
- $P(open|z_1) = 2/3$

$$P(open | z_2, z_1) = \frac{P(z_2 | open) P(open | z_1)}{P(z_2 | open) P(open | z_1) + P(z_2 | \neg open) P(\neg open | z_1)}$$
$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

- z_2 znížilo pravdepodobnosť, že dvere sú otvorené

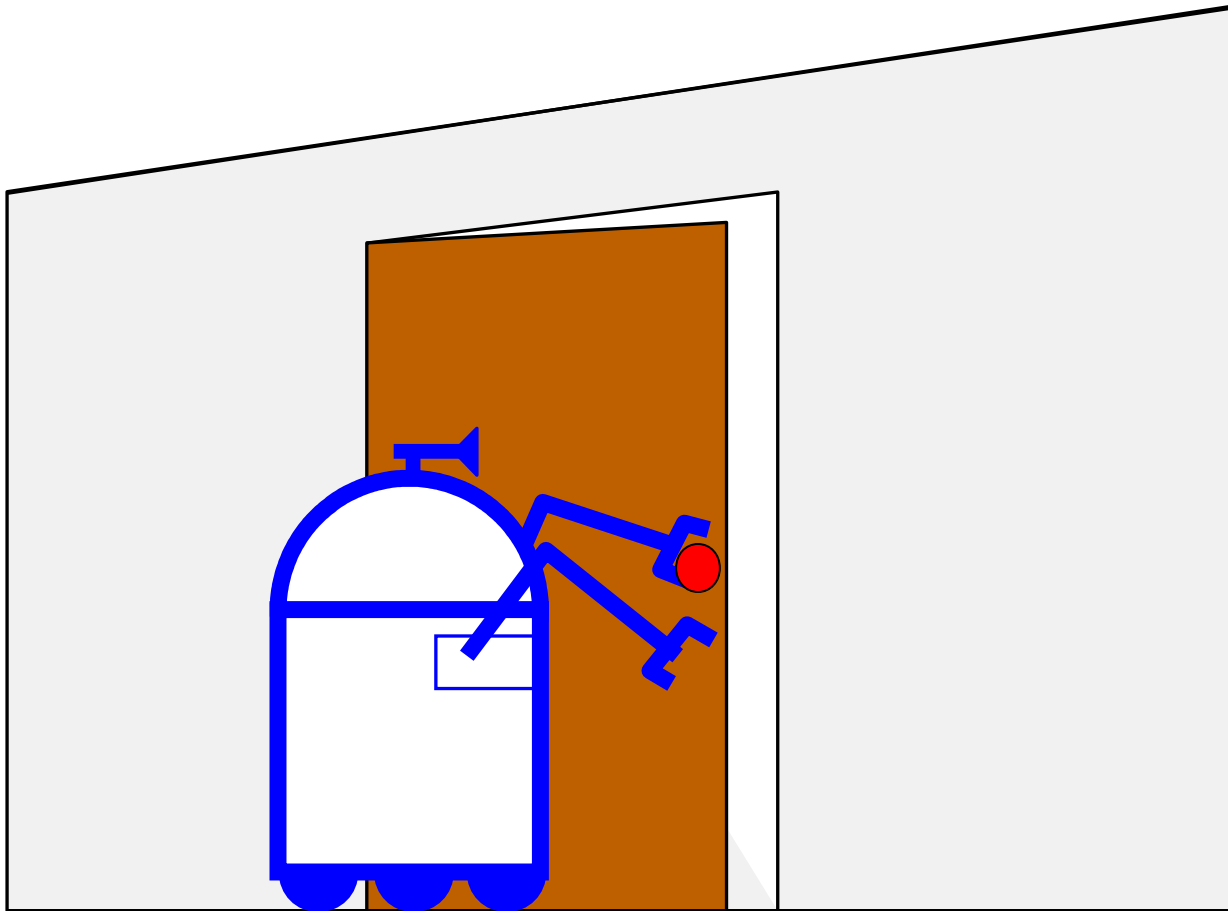
Modelovanie akcií

- Výsledok akcie u zkomponujeme do aktuálneho “belief”, pomocou podmienenej pravdepodobnosti:

$$P(x|u,x')$$

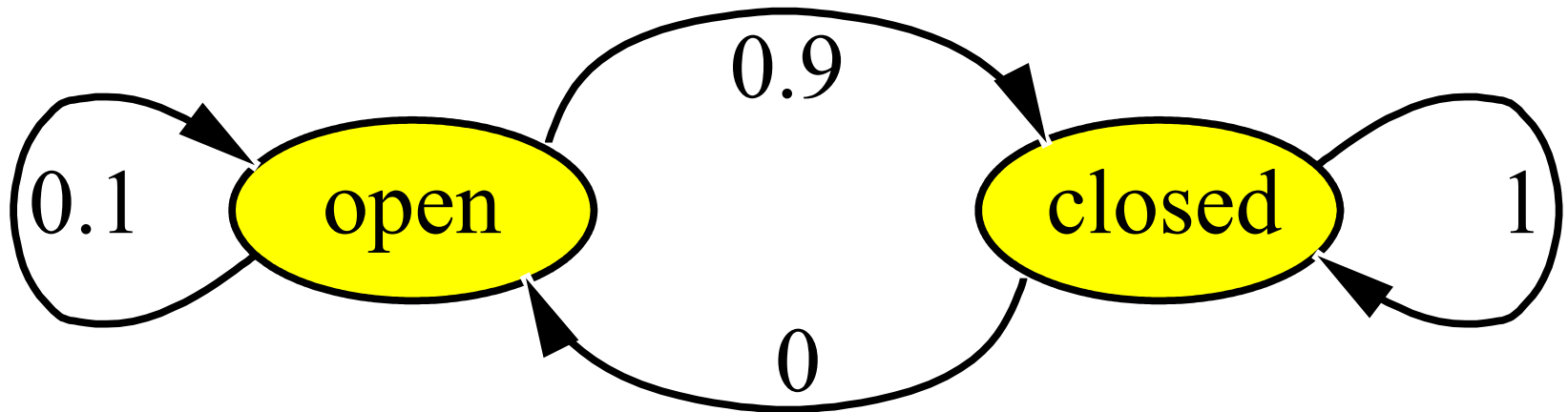
- **vykonaním akcie u sa stav zmení z x' na x**

Príklad: Zatvorenie dverí



Stavové prechody

$P(x|u, x')$ pre $u = \text{“zatvor dvere”}$:



Ak sú dvere otvorené, akcia “zatvor dvere”
uspeje s pravdepodobnosťou 90%

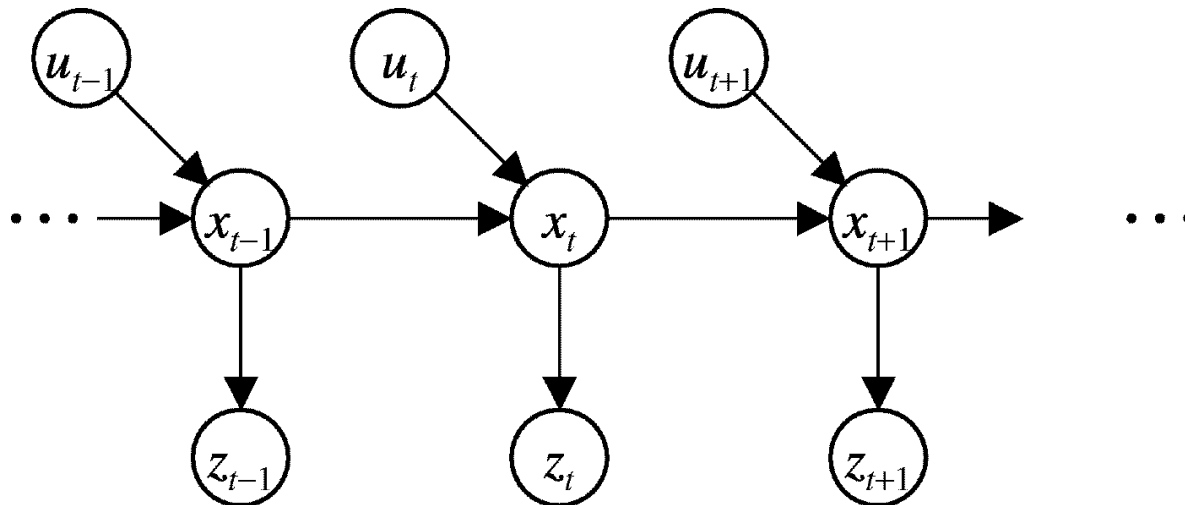
Example: The Resulting Belief

$$\begin{aligned}P(\textit{closed} | u) &= \sum P(\textit{closed} | u, x')P(x') \\ &= P(\textit{closed} | u, \textit{open})P(\textit{open}) \\ &\quad + P(\textit{closed} | u, \textit{closed})P(\textit{closed}) \\ &= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16}\end{aligned}$$

$$\begin{aligned}P(\textit{open} | u) &= \sum P(\textit{open} | u, x')P(x') \\ &= P(\textit{open} | u, \textit{open})P(\textit{open}) \\ &\quad + P(\textit{open} | u, \textit{closed})P(\textit{closed}) \\ &= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16}\end{aligned}$$

$$= 1 - P(\textit{closed} | u)$$

Markovský predpoklad



$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

$$p(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Ďalšie predpoklady

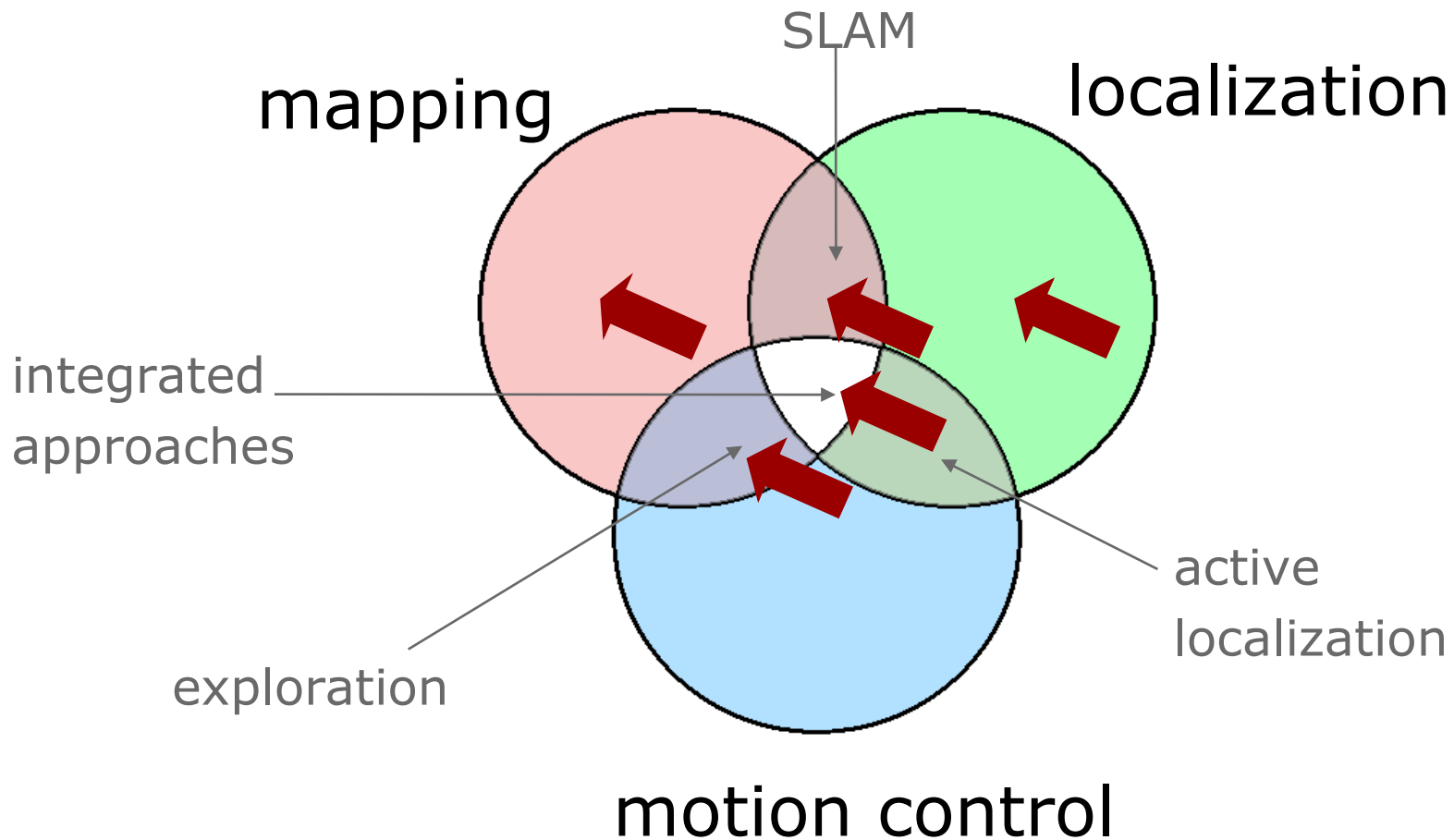
- Statický svet
- Šum nie je závislý
- Model je dokonalý, nedošlo k zjednodušeniam

Bayesovské filtre

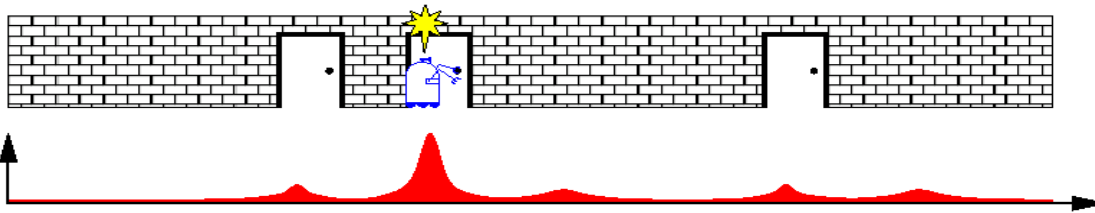
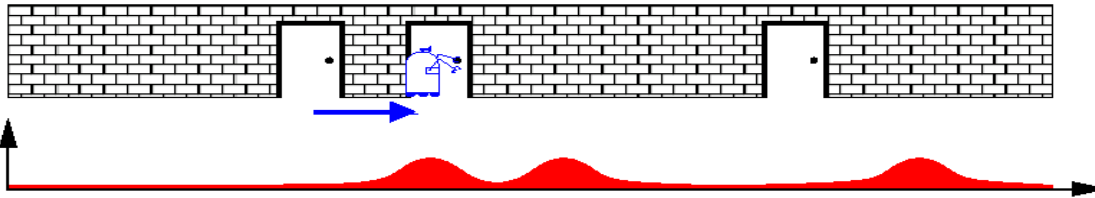
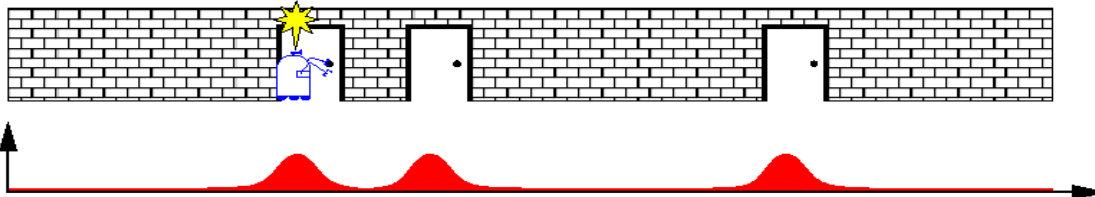
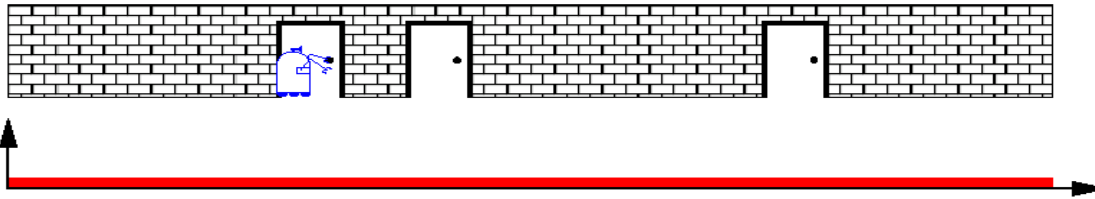
$$Bel(x_t) = \eta P(z_t | x_t) \int P(x_t | u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

- Kalmanove filtre
- Diskrétne filtre
- “Particle” filtre
- Skryté markovovské modely
- Dynamické bayesovské siete
- Partially Observable Markov Decision Processes (POMDPs)

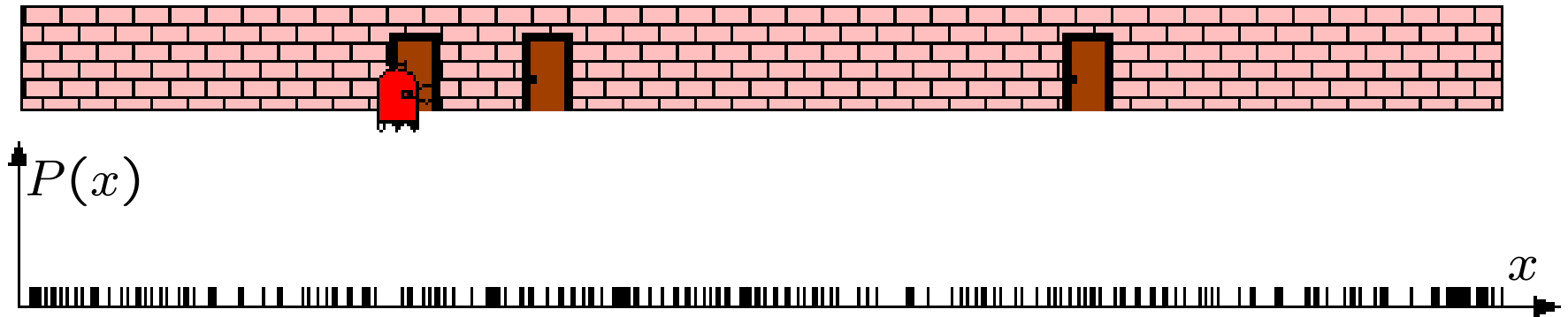
Dimensions of Mobile Robot Navigation



Pravdepodobnostná Lokalizácia

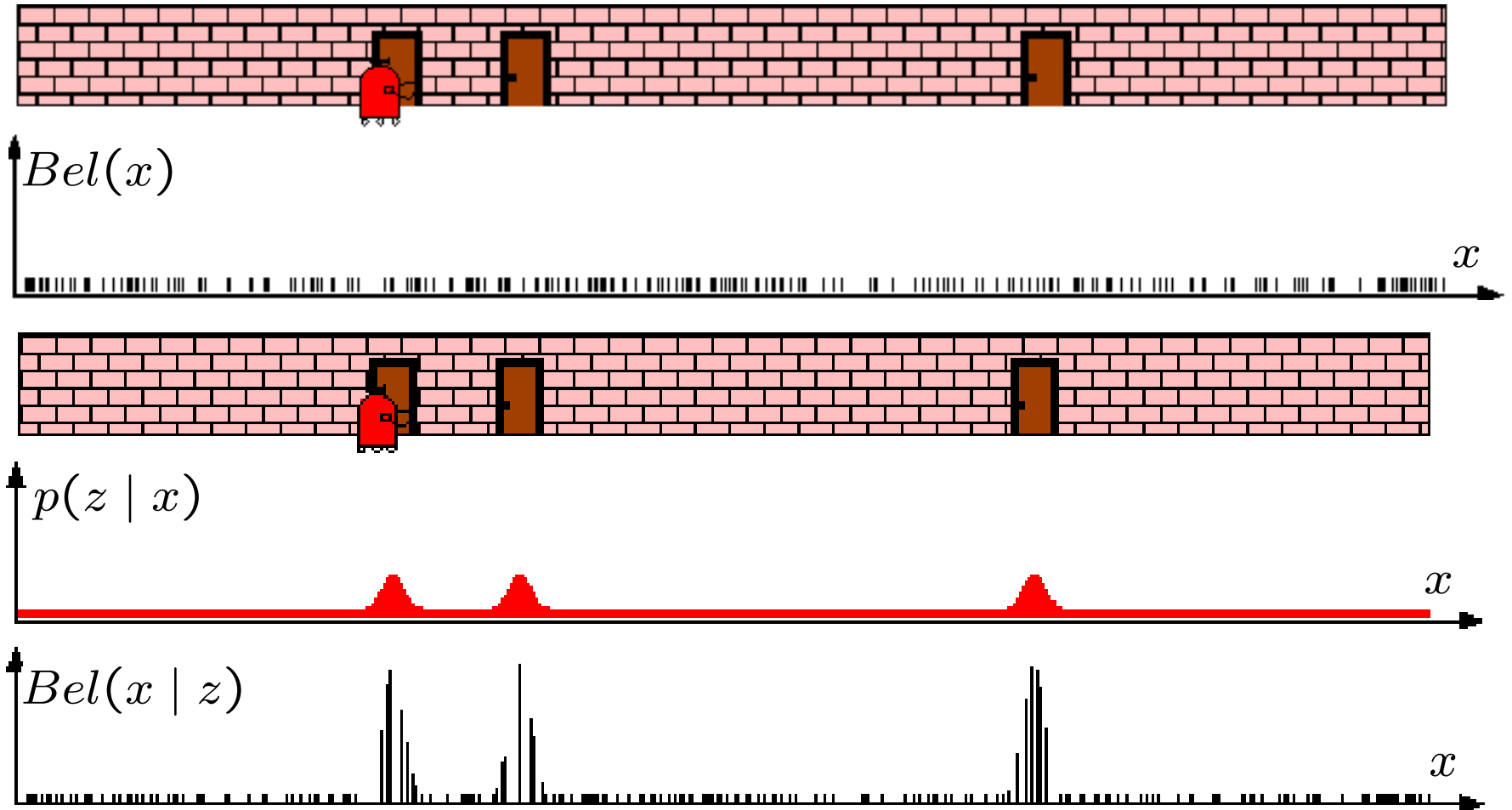


Lokalizácia pomocou “Particle” Filtrov



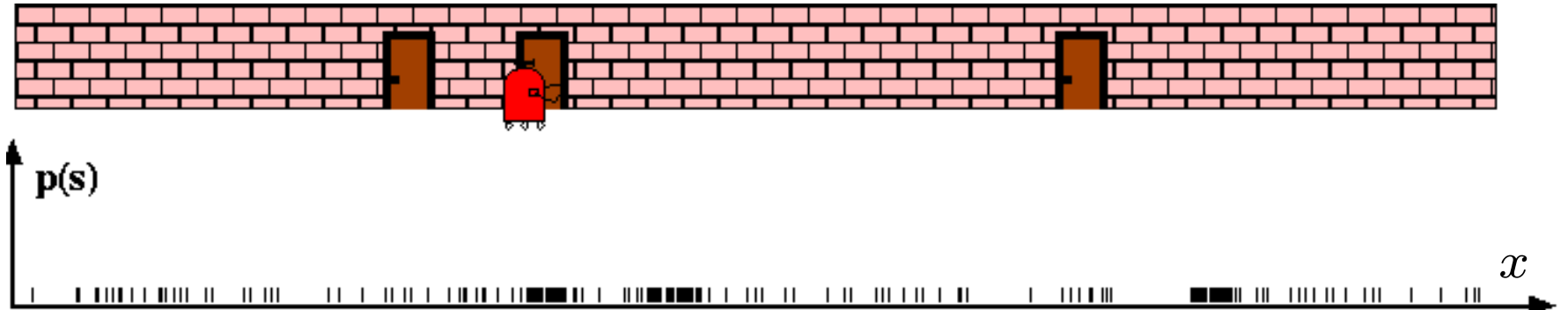
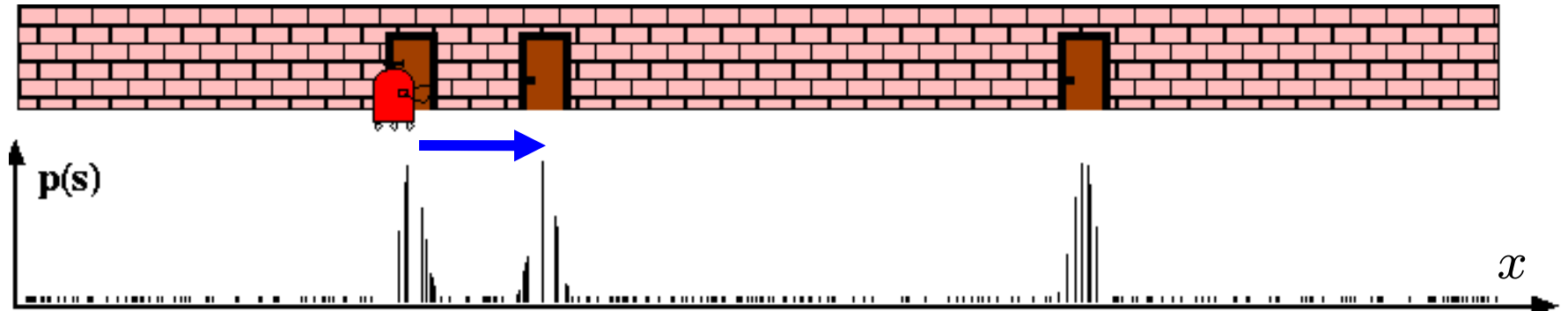
MCL: meranie

$$Bel(x | z) = \alpha p(z | x) Bel(x)$$



PF: pohyb

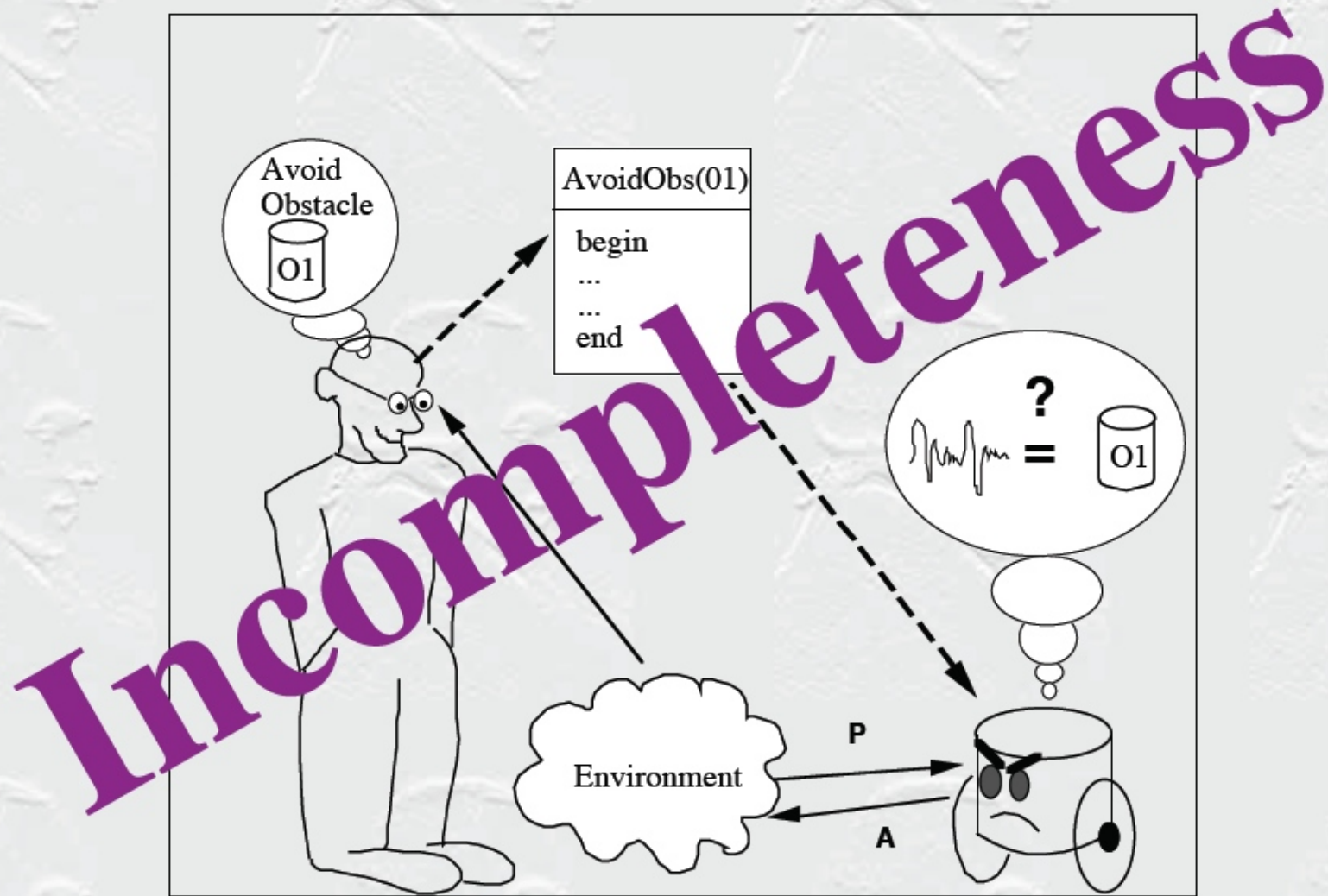
$$Bel(x | u) = \int_{x'} p(x | u, x') Bel(x')$$



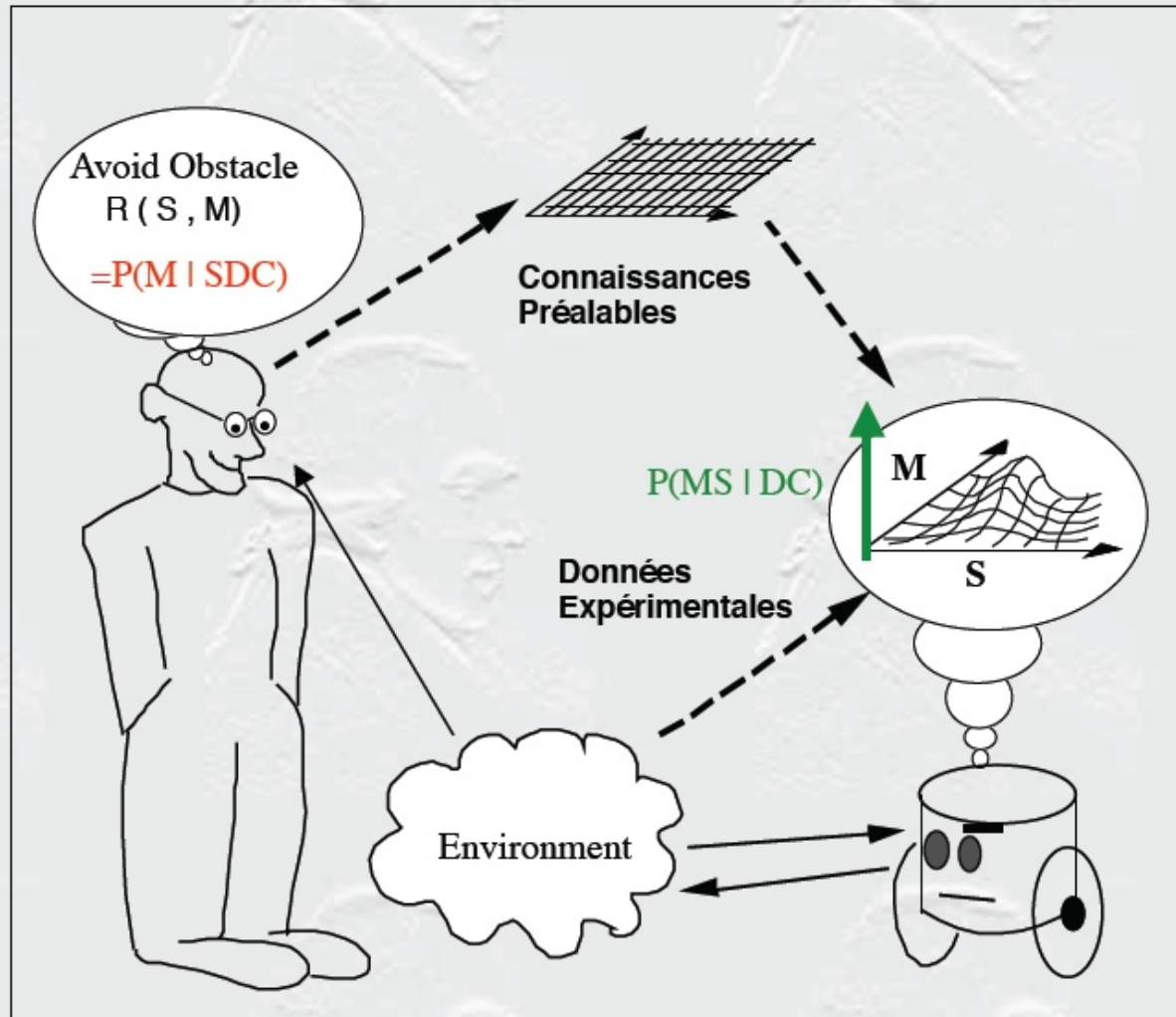
Bayesovské Programovanie Robotov

- Integrovaný prístup, ktorý všetko modeluje pomocou pravdepodobností
- Príklad: tréning robotu Khepera
- (video)

Logical Paradigm



Bayesian Paradigm



Principle

Incompleteness

Bayesian Learning

Preliminary Knowledge
+
Experimental Data
=
Probabilistic Representation

Uncertainty

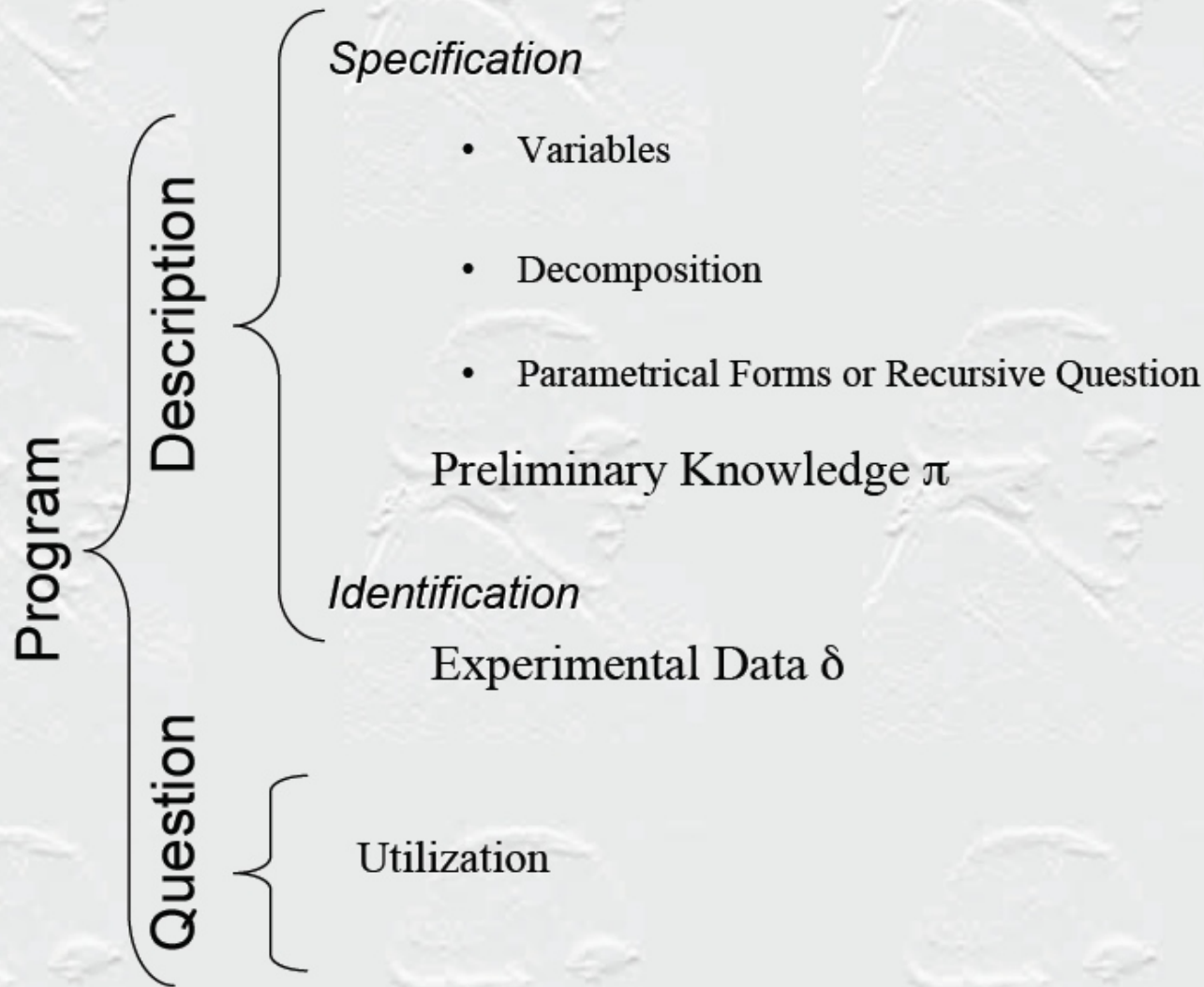
Bayesian Inference

$$P(a) + P(\neg a) = 1$$

$$P(a \wedge b) = P(a)P(b|a) = P(b)P(a|b)$$

Decision

Bayesian Program



Pushing Objects

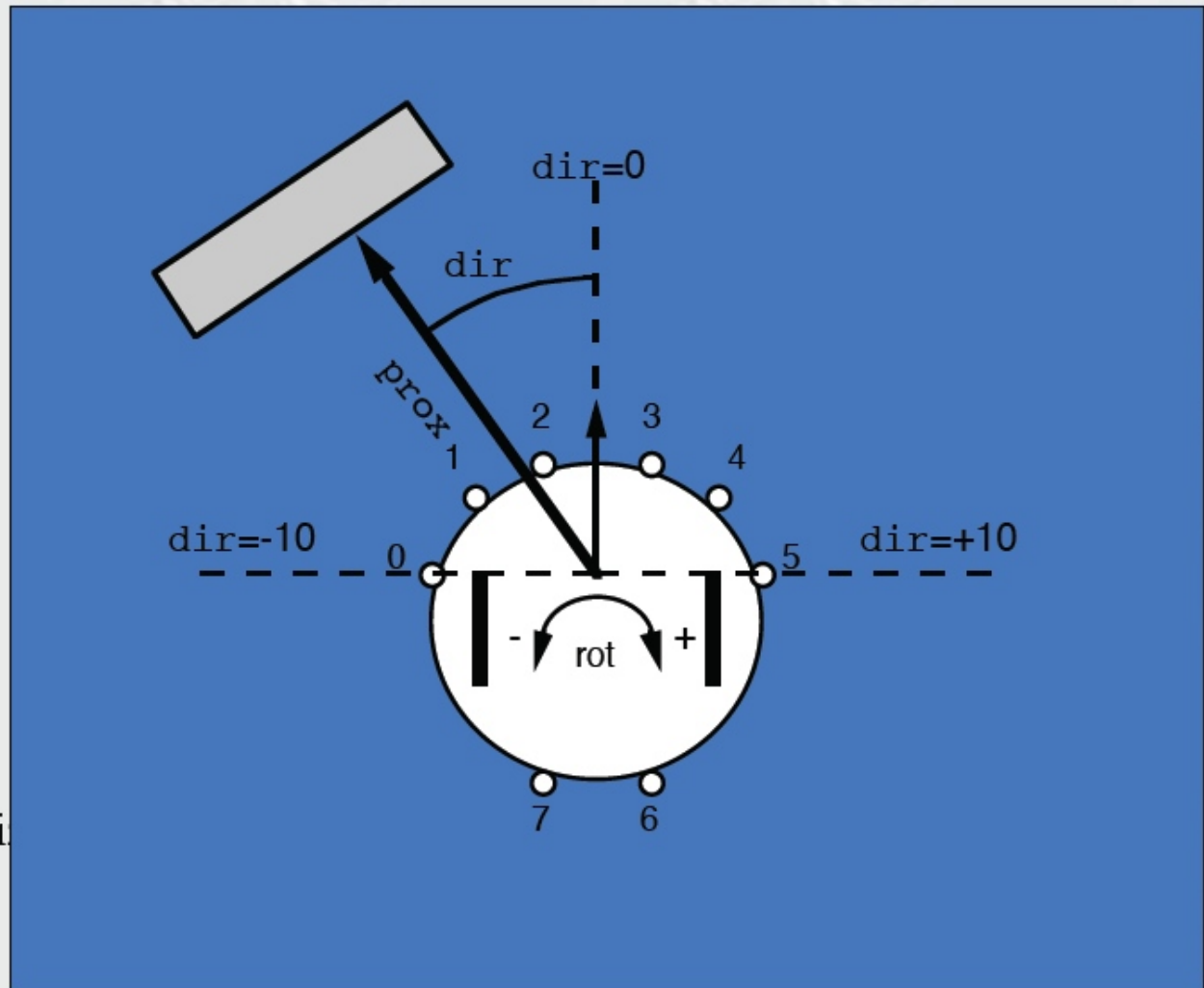
Program

Description

Question

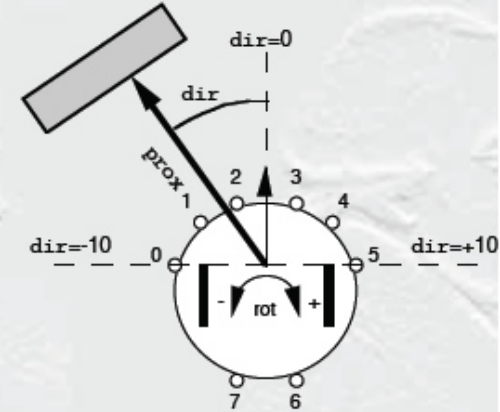
Specification

- Variables



Utili

Pushing Objects



Program

Description

Question

Specification

- Variables

$$Dir \wedge Prox \wedge Vrot$$

- Decomposition

$$P(Dir \wedge Prox \wedge Vrot \mid \delta \wedge \pi)$$

$$= P(Dir \mid \delta \wedge \pi) \times P(Prox \mid \delta \wedge \pi) \times P(Vrot \mid Dir \wedge Prox \wedge \delta \wedge \pi)$$

- Parametrical Forms

$$P(Dir \wedge Prox \mid \delta \wedge \pi) \leftarrow \text{Uniform}$$

$$P(Vrot \mid Dir \wedge Prox \wedge \delta \wedge \pi) \leftarrow \text{Gaussians}$$

- Preliminary Knowledge π

Identification

- Joystick Remote Control → Experimental Data $\delta 1$

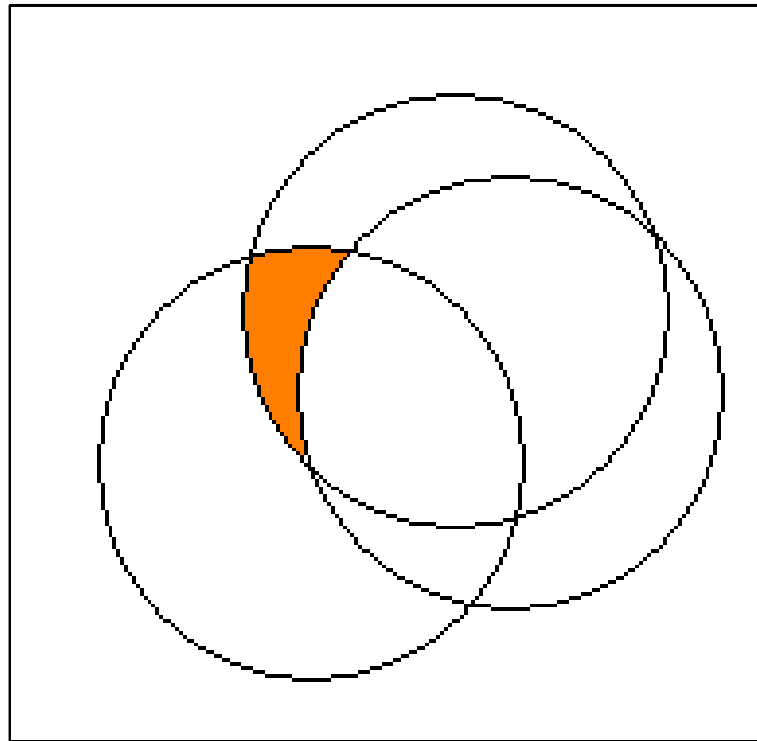
$$P(Dir \wedge Prox \wedge Vrot \mid \delta 1 \wedge \pi)$$

Utilization

$$P(Vrot \mid [Dir = d] \wedge [Prox = p] \wedge \delta 1 \wedge \pi)$$

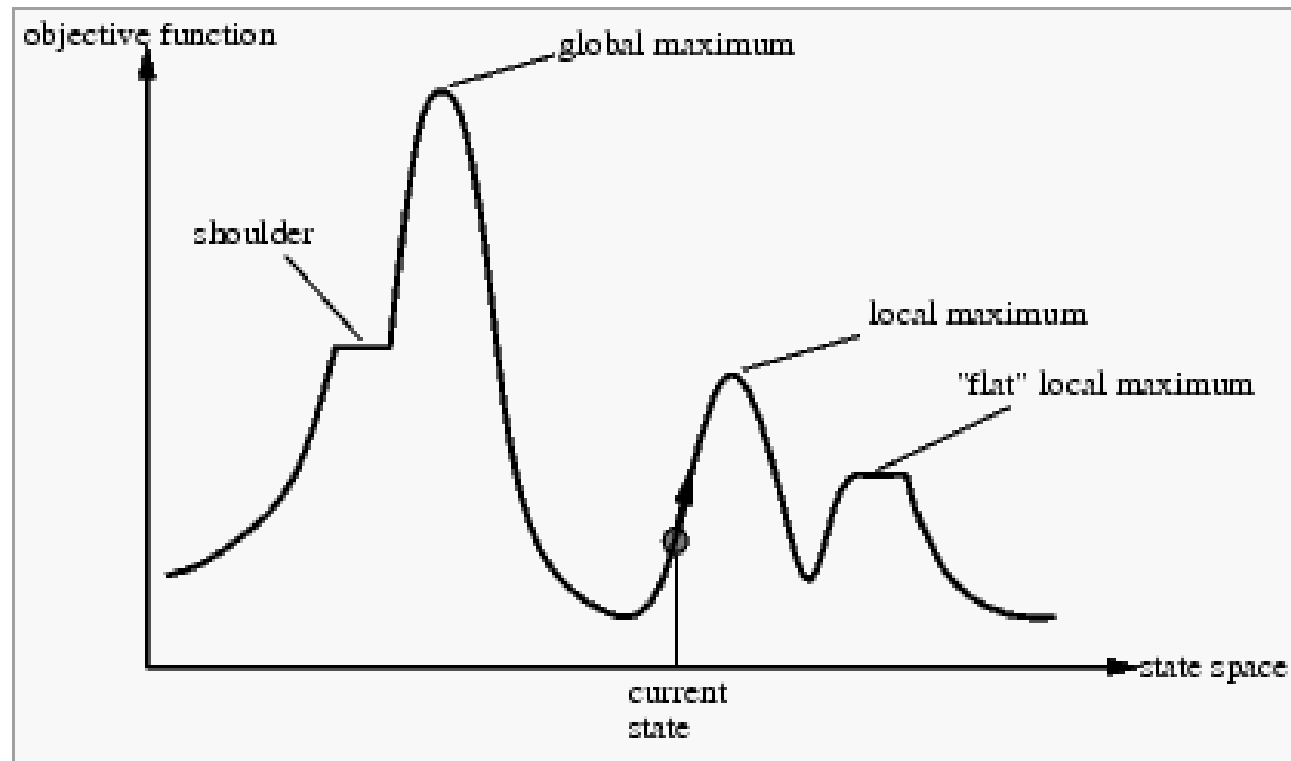
Stochastické metódy: Monte Carlo

- Zistiť obsah určitého tvaru:



Stochastické metódy: Simulované žíhanie

- Prehľadávací priestor skúmaný na základe lokálneho okolia:

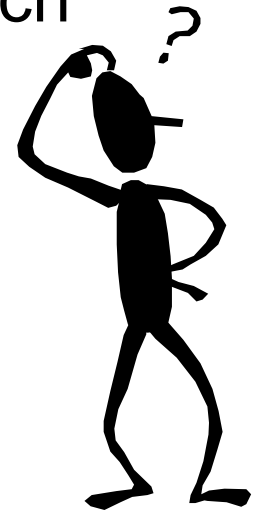


Princípy Prirodzenej Evolúcie

- Informácia o jedincoch je uložená v ich genotype, ktorý pozostáva z génov / aliel
- Úspešnejšie jedince majú väčšiu šancu prežiť a preto vyššiu pravdepodobnosť potomstva
- Populácia jedincov sa prispôsobuje meniacim sa podmienkam, keďže viac prispôsobené jedince budú prevládať v populácii
- Zmeny genotypu prichádzajú prostredníctvom mutácie a rekombinácie

Evolučné výpočty

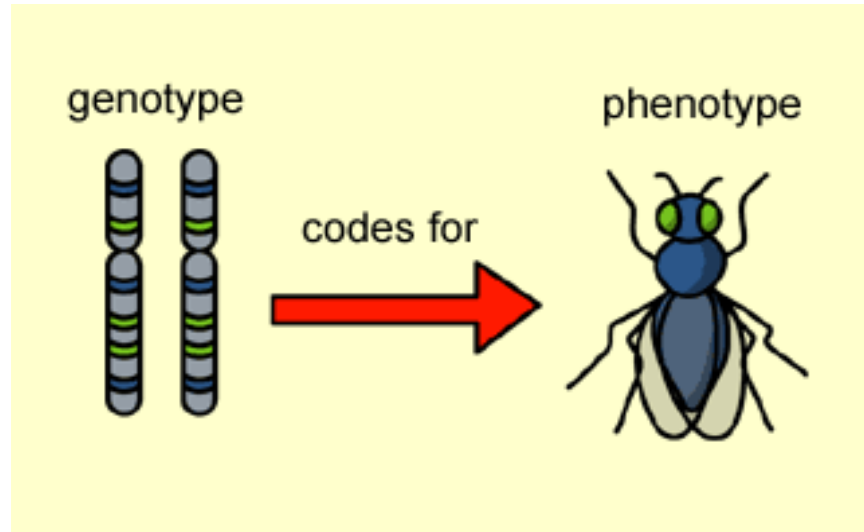
- Hľadáme riešenie nejakého problému
- Jednotné kódovanie riešení (reprezentácia)
- Fitness: účelová funkcia, ktorá číselne ohodnotí vhodnosť jedinca
- Populácia: množina náhodne vygenerovaných jedincov
- Princípy prirodzenej evolúcie:
 - výber, rekombinácia, mutácia
- Beží mnoho generácií



EA pojmy

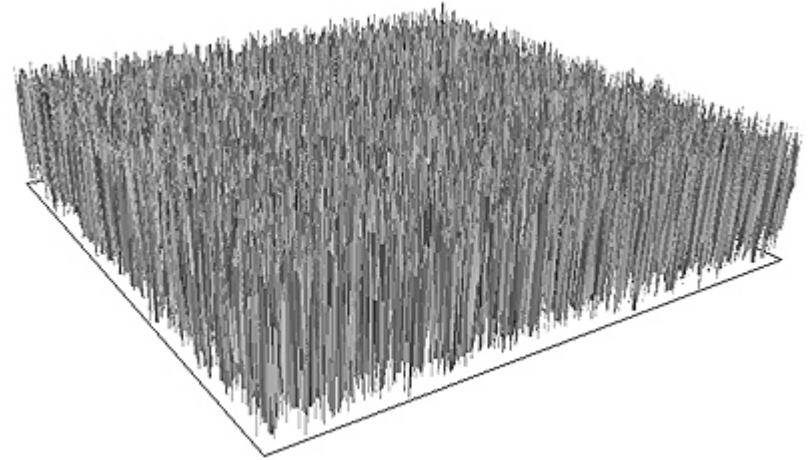
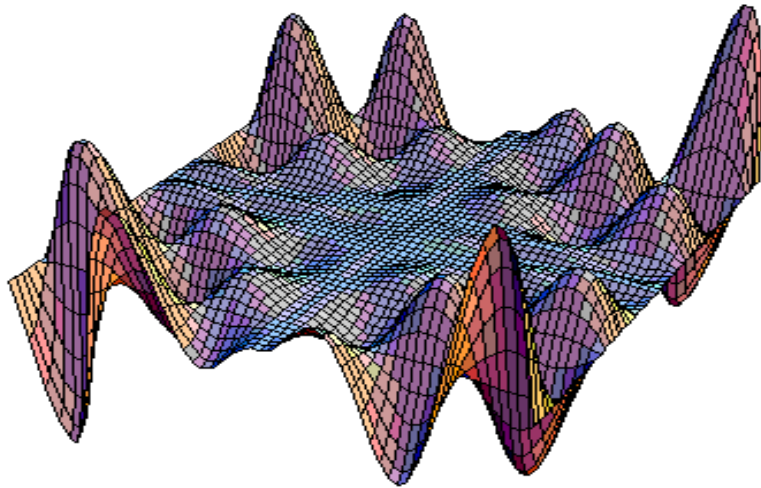
- genotype and phenotype
- fitness landscape
- diversity, genetic drift
- premature convergence
- exploration vs. exploitation
- selection methods: roulette wheel (fit.prop.), tournament, truncation, rank, elitist
- selection pressure
- direct vs. indirect representations
- fitness space

Genotype and Phenotype



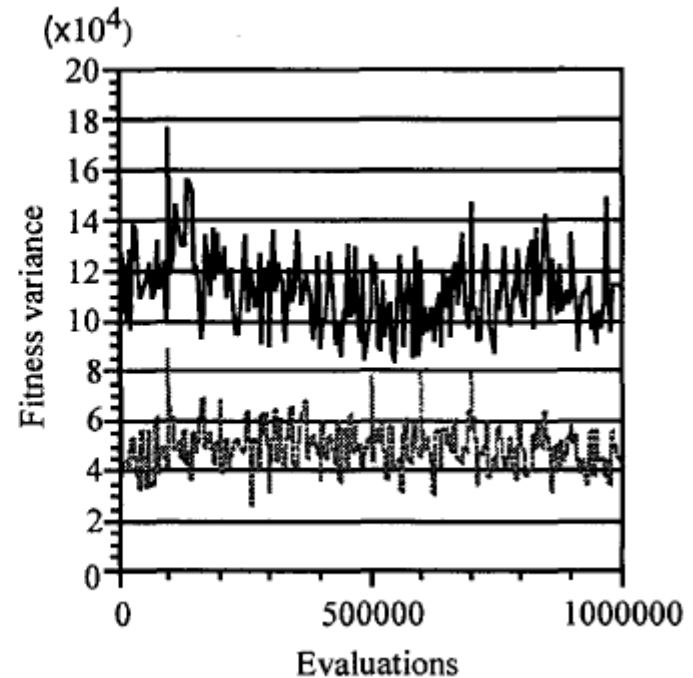
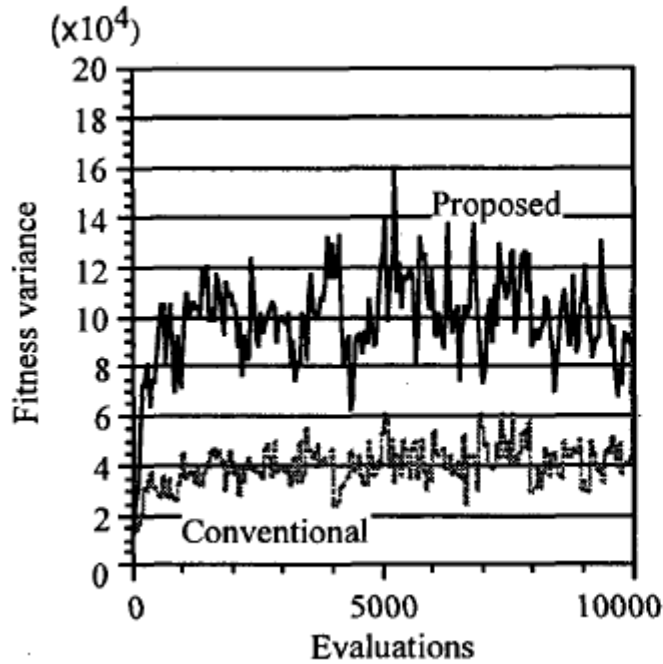
- *Genotype* – all genetic material of a particular individual (genes)
- *Phenotype* – the real features of that individual

Fitness landscape



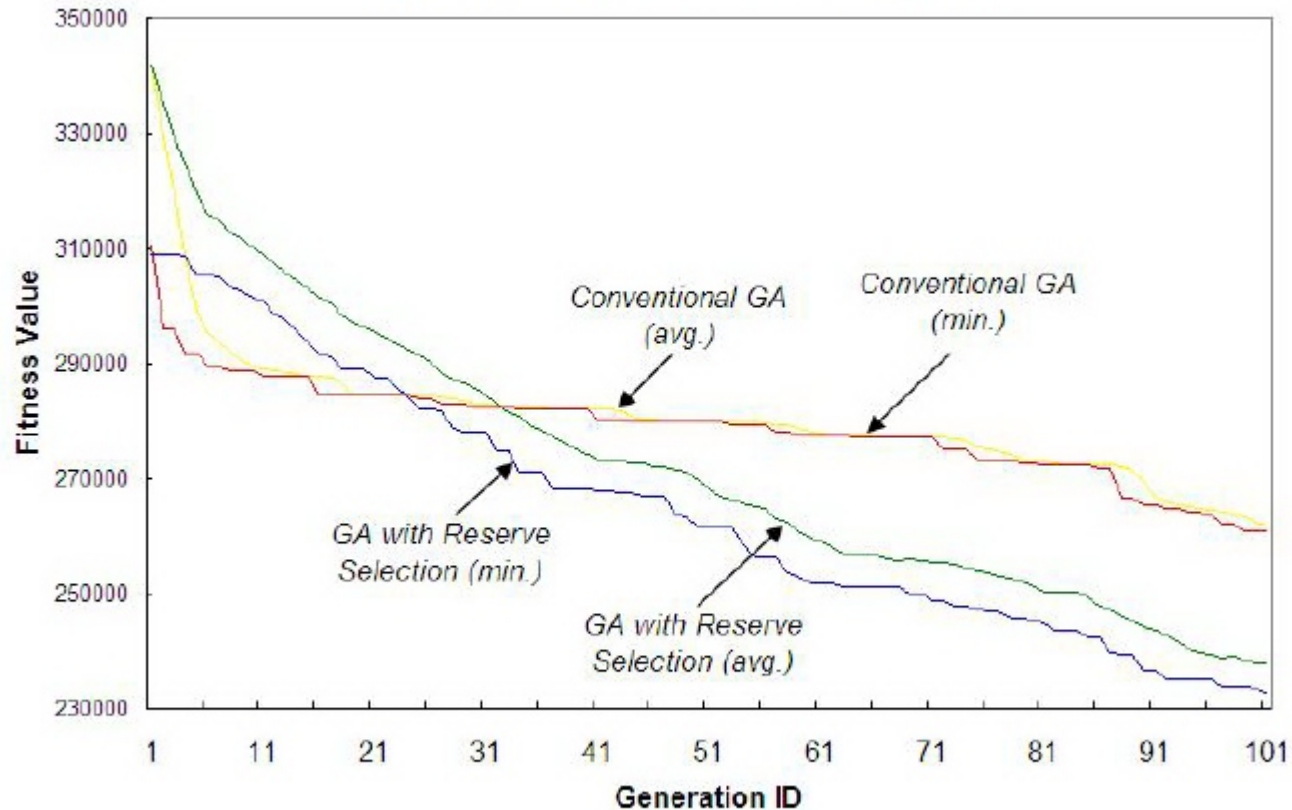
- Genotype space – difficulty of the problem – shape of fitness landscape, neighborhood function

Population diversity



- Must be kept high for the evolution to advance

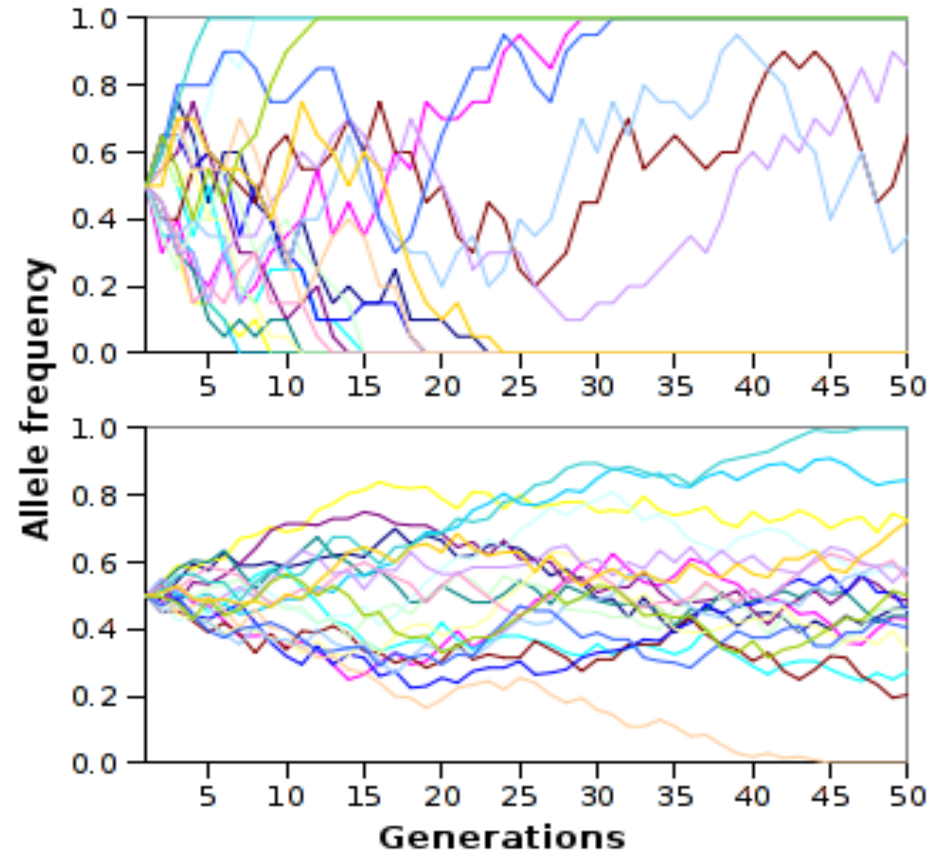
Premature convergence



- important building blocks are lost early in the evolutionary run

Genetic drift

- Loosing the population distribution due to the sampling error



Exploration vs. Exploitation

- Exploration phase: localize promising areas
- Exploitation phase: fine-tune the solution

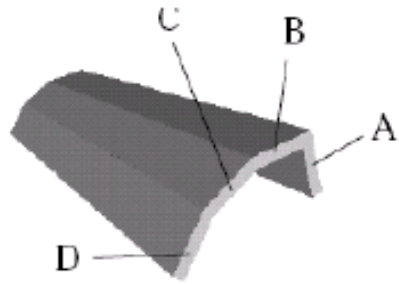
Selection methods





- roulette wheel (fitness proportionate selection),
- tournament selection
- truncation selection
- rank selection
- elitist strategies

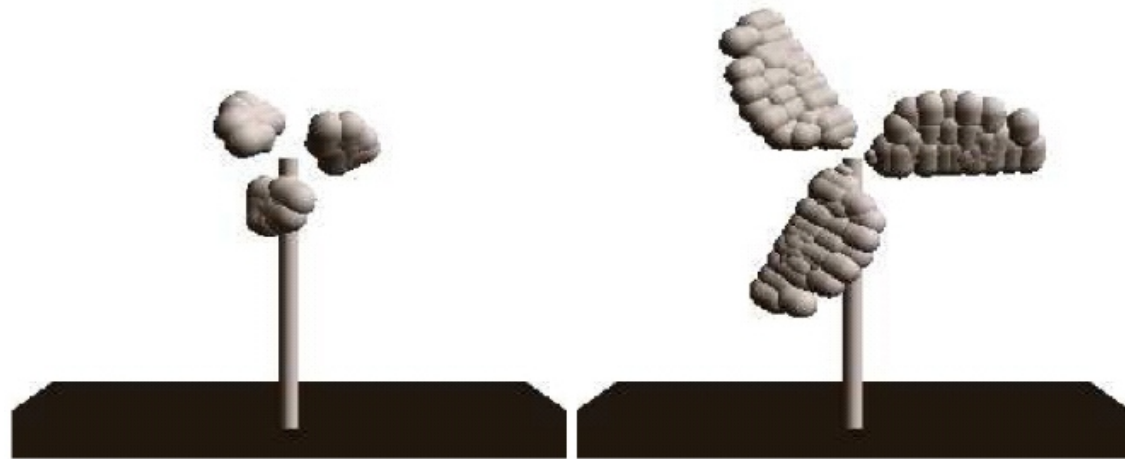
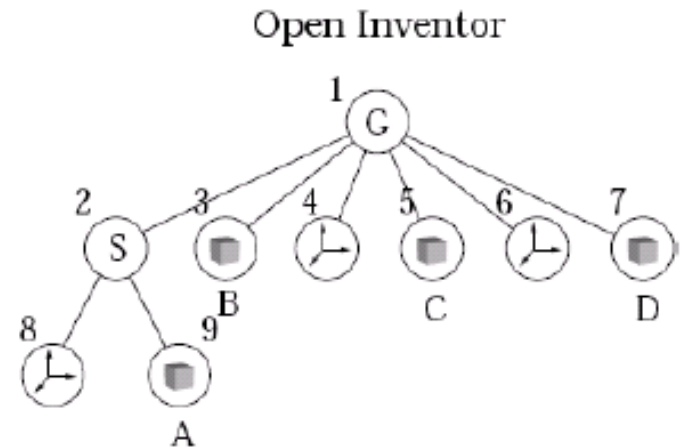
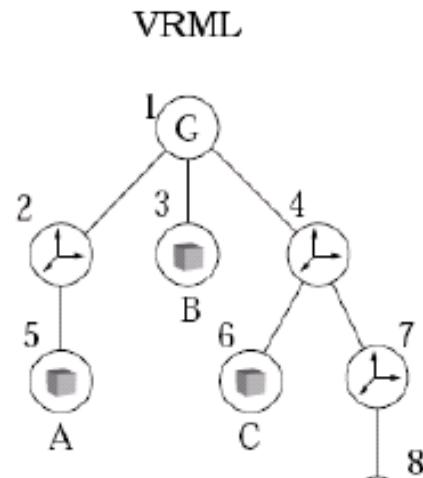
Selection pressure

- Influenced by the problem
- Relates to evolutionary operators

Direct vs. Indirect Representations



-  tree root
-  separator node
-  transformation node
-  leaf - prism

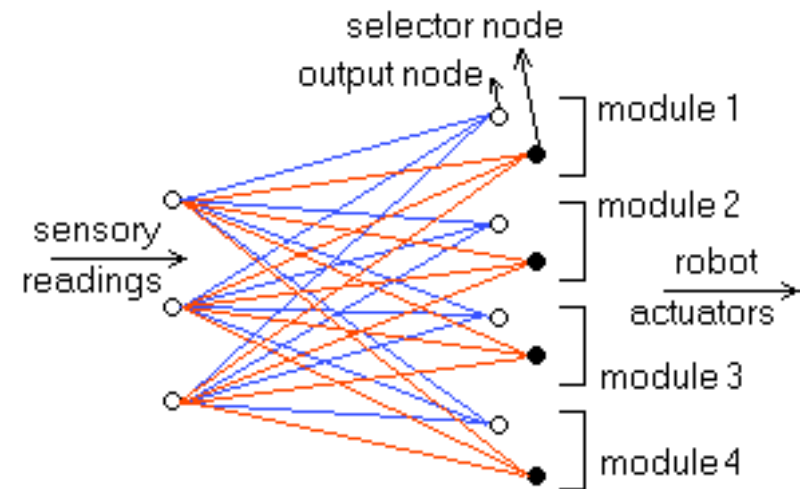
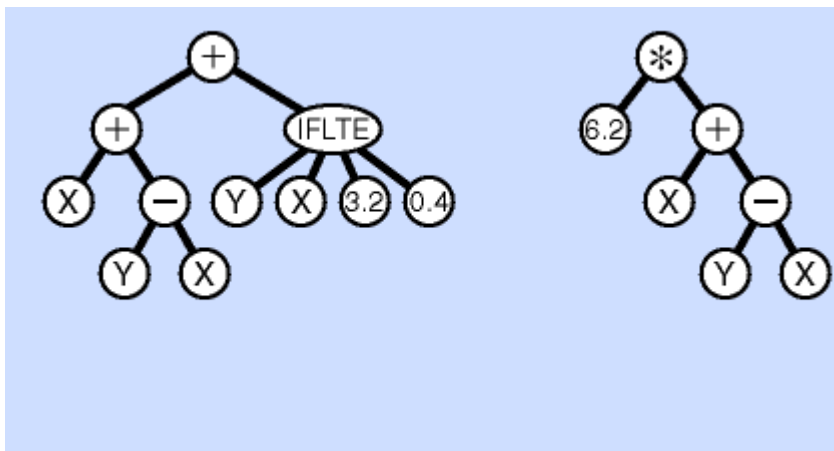


Fitness Space (Floreano)

- Functional vs. behavioral
- Explicit vs. implicit
- External vs. internal

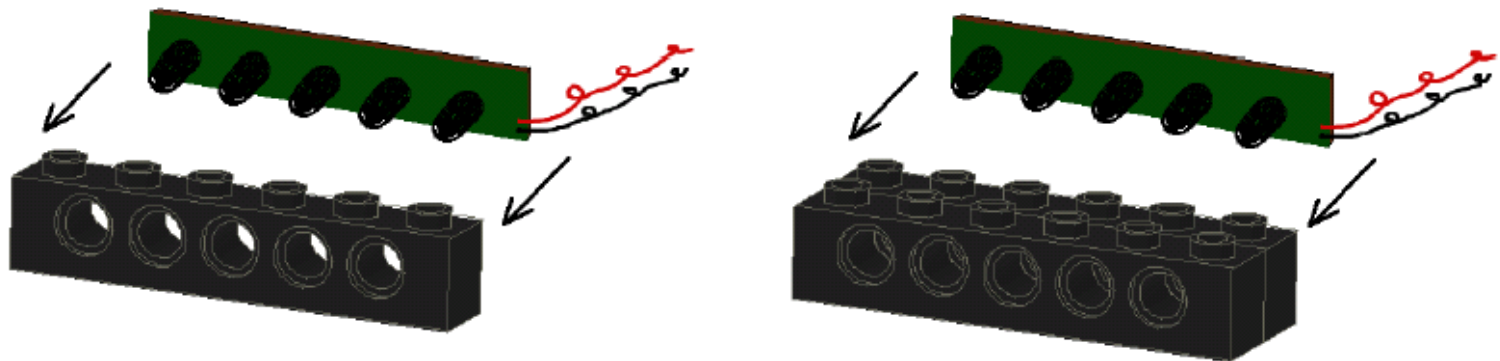
Evolutionary Robotics

- Solution: Robot's controller



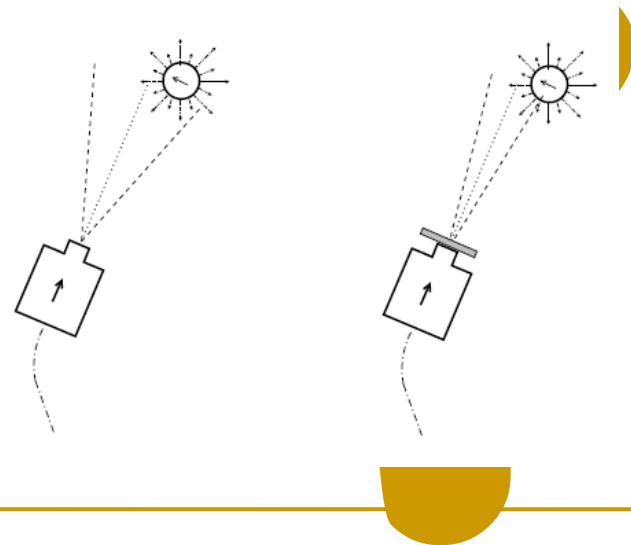
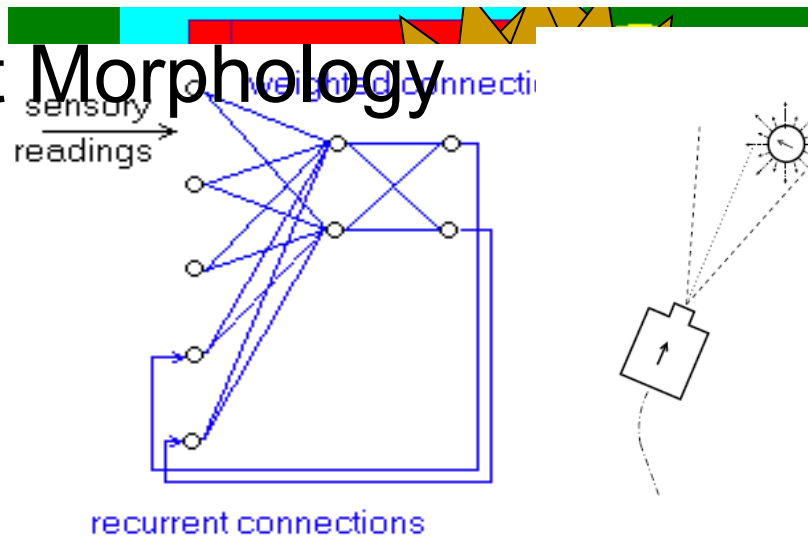
- Fitness: how well the robot performs
- Simulation or real robot

Fitness Influenced by



Robot Morphology

$\frac{\text{sensc}}{\text{Inc readin}}$



Evolvable Tasks

- Wall following
- Obstacle avoidance
- Docking and recharging
- Artificial ant following
- Box pushing
- Lawn mowing
- Legged walking
- T-maze navigation
- Foraging strategies
- Trash collection
- Vision discrimination and classification tasks
- Target tracking and navigation
- Pursuit-evasion behaviors
- Soccer playing
- Navigation tasks

Neuroevolution through augmenting topologies

- The most successful method for evolution of artificial neural networks
- Sharing fitness
- Starting with simple solutions
- Global counter
- i.e. Topological crossover – very important for preserving evolved structures

What is Learning?

