

Tópicos Especiais I - Jogos

IA para Jogos



Machine Learning

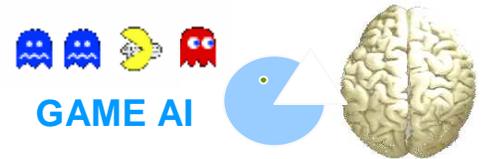
Aprendizado
Indutivo

CBR, IDT, Perfil de Usuário
Reinf. Learning, ANN

Fernando Osório

11/05/2007

IA para Jogos – Aula 03



Tópicos abordados...

- Aprendizado de máquinas em Jogos
“Machine Learning for Games”
 - Raciocínio baseado em Casos (RBC / CBR)
 - Árvores de Decisão (AD / IDT)
 - Perfil de Usuários
 - Aprendizado por Reforço (RL)
 - Redes Neurais Artificiais (RNA / ANN)
- Aplicações...

**Bibliografia
em M.L.**

Machine Learning. Tom Mitchell, 1998.
Sistemas Inteligentes. Solange Rezende, 2003.
Redes Neurais. Simon Haykin, 2001.
C4.5: Programs for machine learning. Ross Quinlan, 1993
Raciocínio Baseado em Casos. Von Wangenheim, 2003.

Machine Learning / Aprendizado de Máquinas...

➤ Conceitos básicos

- Aprendizado = Adaptação do Comportamento
 - Melhorar a Performance
 - Evitar de Repetir os Erros
 - Interação com o Meio, Experimentação, Descoberta
 - Uso da Experiência, Memória do Passado
 - Generalização – Criar Regras Gerais



Machine Learning / Aprendizado de Máquinas...

➤ Conceitos básicos

- Aprendizado = Adaptação do Comportamento
 - Melhorar a Performance
 - Evitar de Repetir os Erros
 - Interação com o Meio, Experimentação, Descoberta
 - Uso da Experiência, Memória do Passado
 - Generalização – Criar Regras Gerais

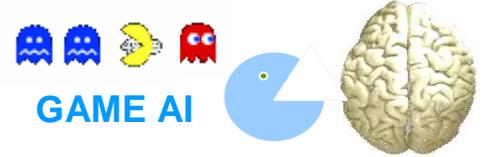


SBIA/SBRN 2003

Jogos:

- Aprendizado de Comportamentos
- Estratégias
- Ação
- Reação
- Inteligência!

*Agentes
Inteligentes*



Machine Learning / Aprendizado de Máquinas...

➤ Conceitos básicos

- Aprendizado = Adaptação do Comportamento
 - Melhorar a Performance
 - Evitar de Repetir os Erros
 - Interação com o Meio, Experimentação, Descoberta
 - Uso da Experiência, Memória do Passado
 - Generalização – Criar Regras Gerais

- Conceitos:
 - Experiência & Memória (casos passados)
 - Otimização & Adaptação (medida de performance)
 - Interação (passivo, ativo)
 - Geração de Conhecimentos Novos (regras, generalização)

Machine Learning / Aprendizado de Máquinas...

➤ Conceitos básicos

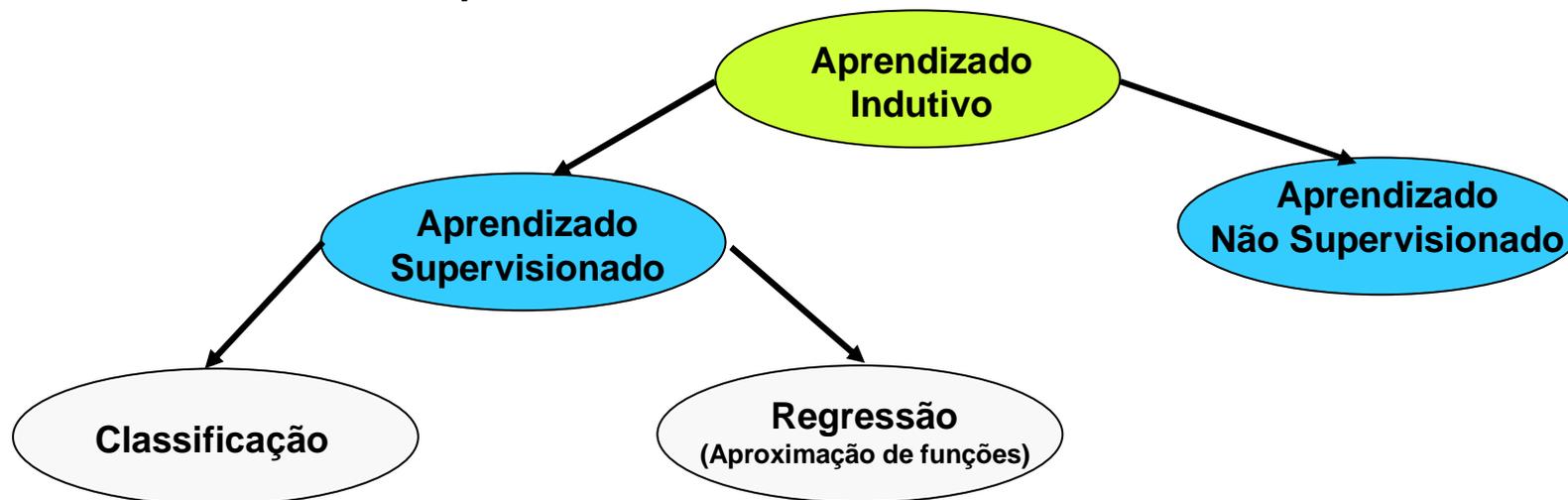
▪ Aprendizado:

- Adaptação do Comportamento
- Melhorar a Performance
- Evitar de Repetir os Erros
- Interação com o Meio, Experimentação, Descoberta
- Uso da Experiência, Memória do Passado
- Generalização – Criar Regras Gerais

▪ Conceitos:

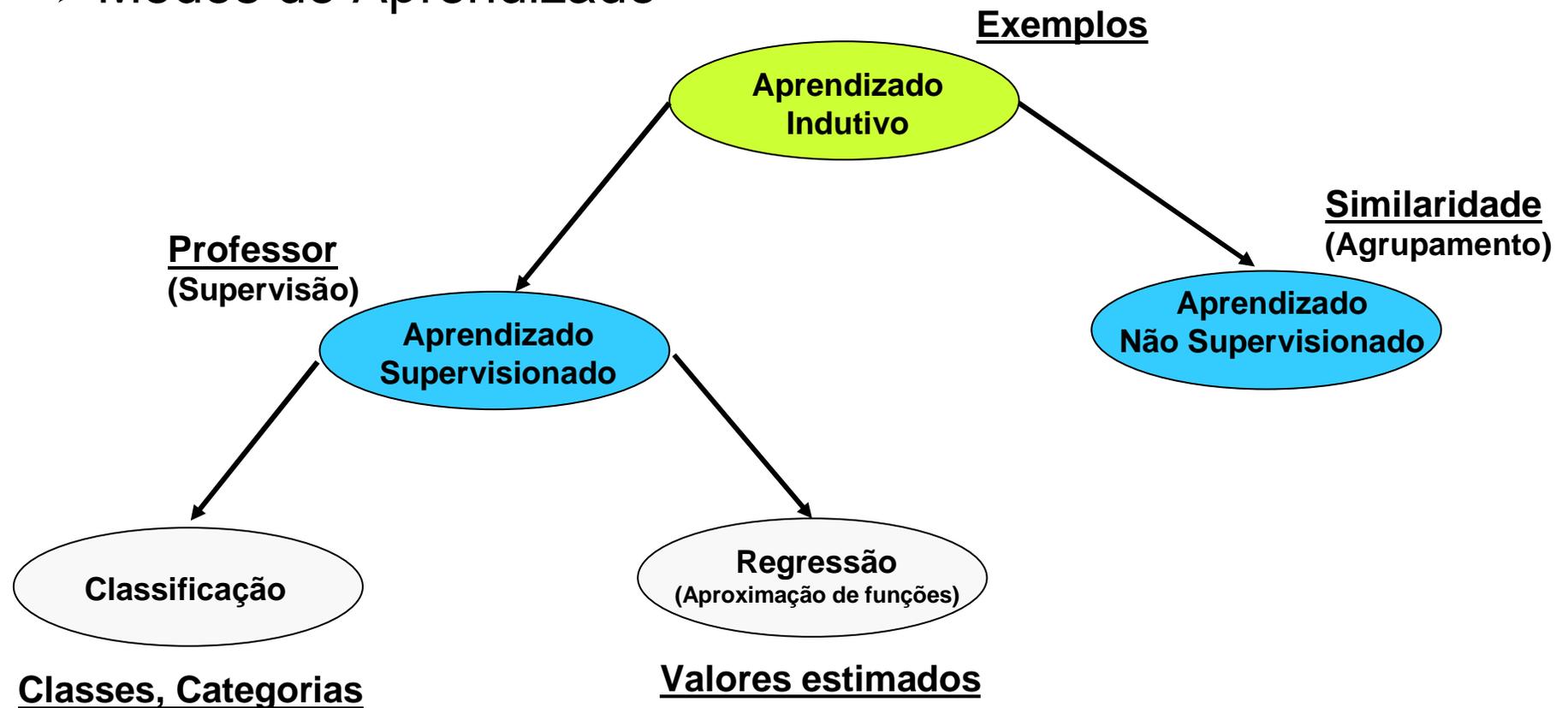
- Experiência & Memória (casos passados)
- Otimização & Adaptação (medida de performance)
- Interação (passivo, ativo)
- Geração de Conhecimentos Novos (regras, generalização)

➤ Modos de Aprendizado



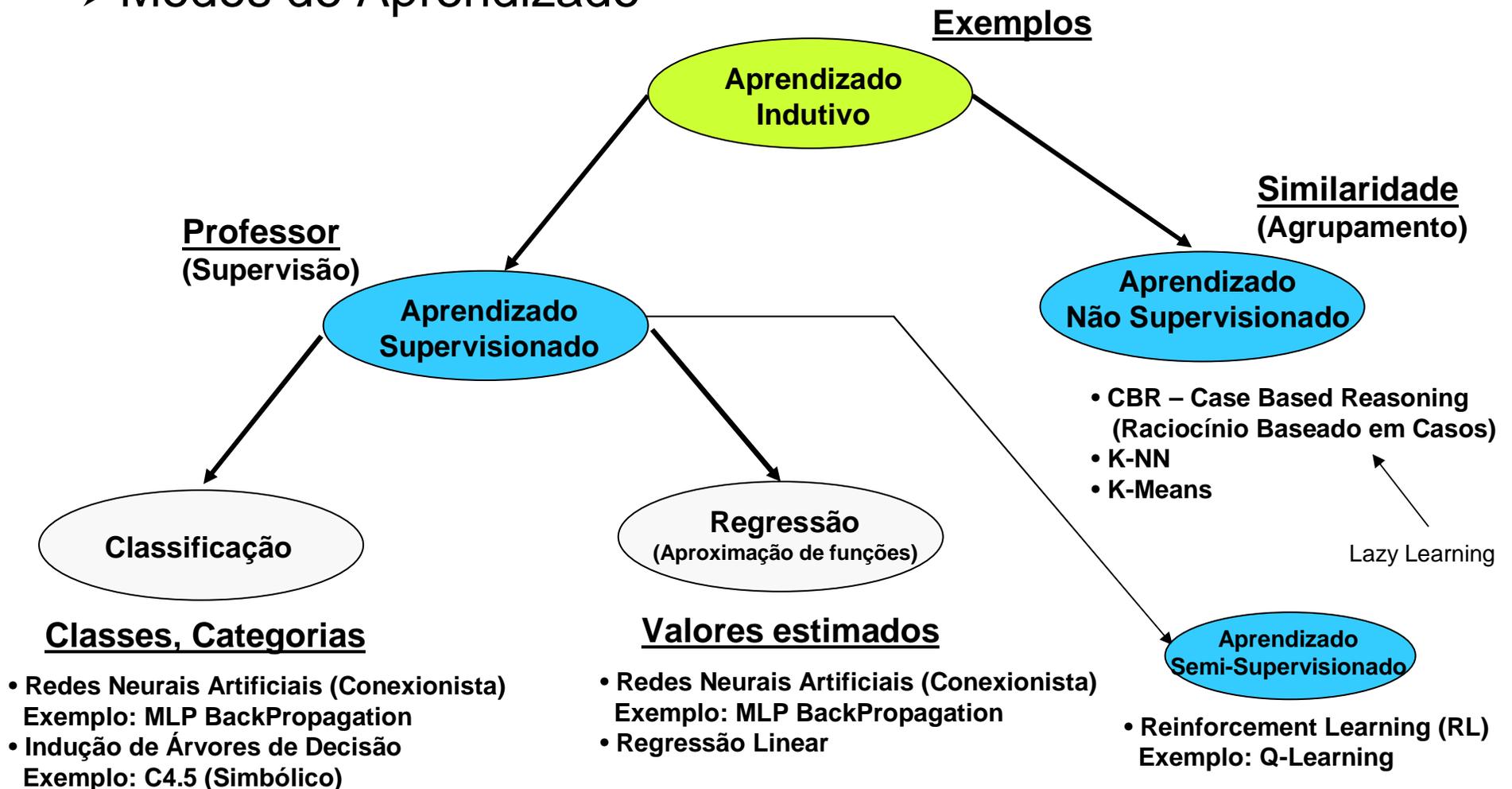
Machine Learning / Aprendizado de Máquinas...

➤ Modos de Aprendizado



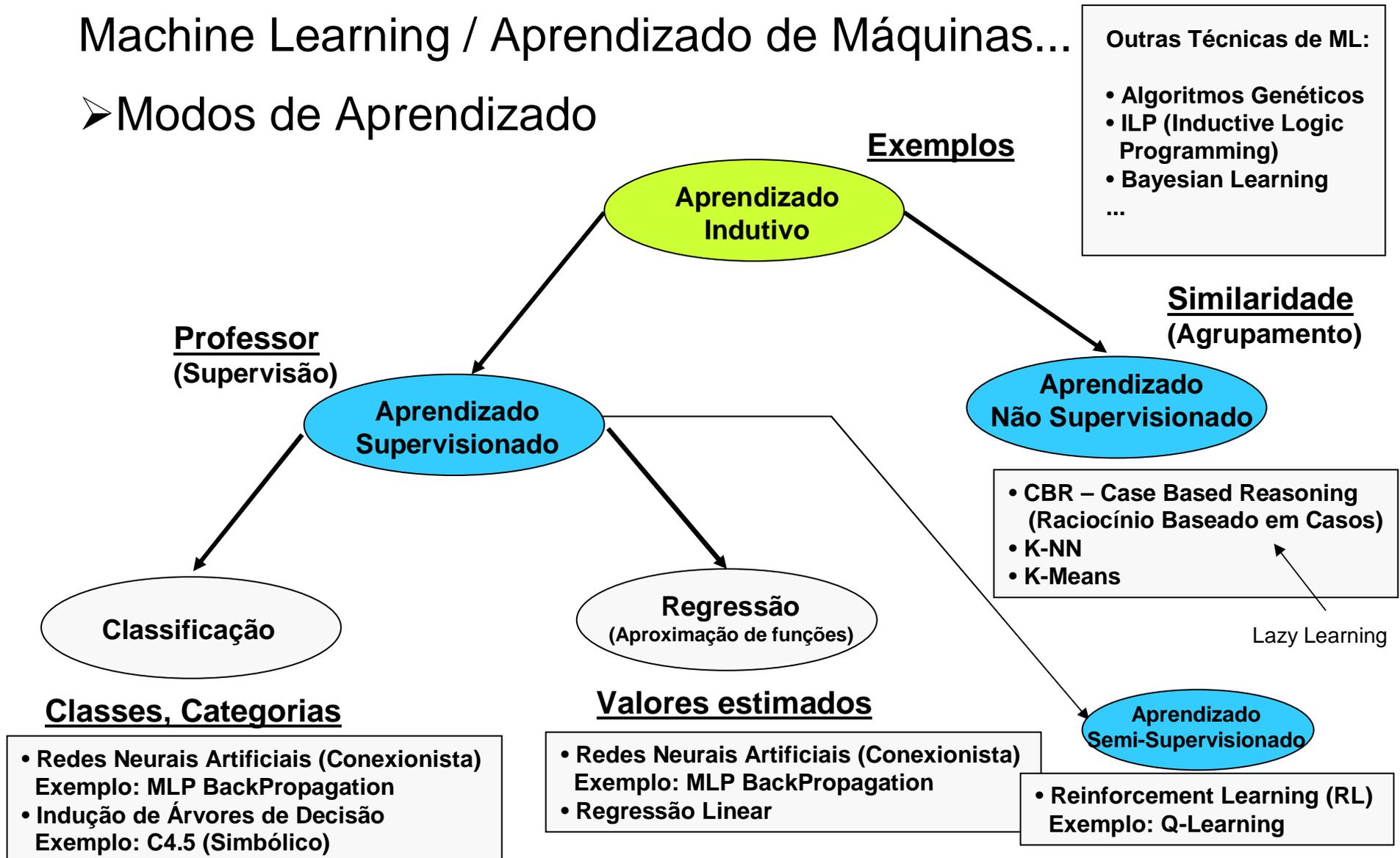
Machine Learning / Aprendizado de Máquinas...

➤ Modos de Aprendizado

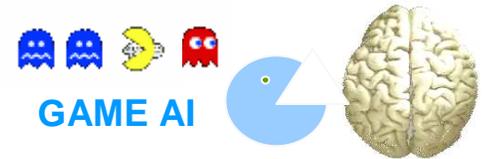


Machine Learning / Aprendizado de Máquinas...

➤ Modos de Aprendizado



ML4Games



Machine Learning... **Por onde começar?**



No aprendizado indutivo, seja ele não supervisionado ou supervisionado, voltado para classificação, regressão ou apenas agrupamento de dados, neste tipo de aprendizado é necessário um **conjunto de exemplos** que servirá para introduzir (induzir) o conhecimento a ser aprendido.

Esquema simples de Aprendizado Indutivo...



Machine Learning... **Por onde começar?**

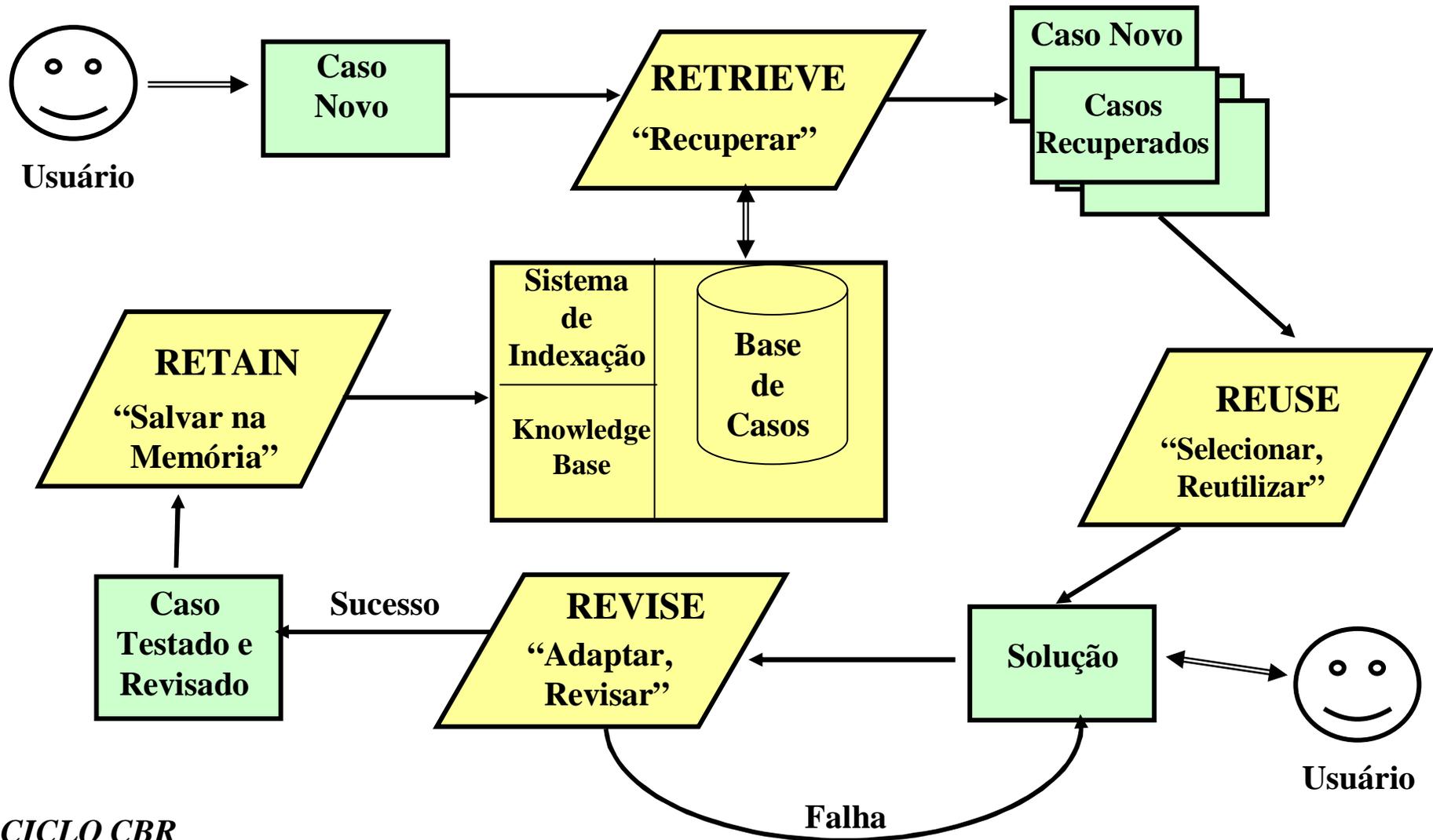


No aprendizado indutivo, seja ele não supervisionado ou supervisionado, voltado para classificação, regressão ou apenas agrupamento de dados, neste tipo de aprendizado é necessário um **conjunto de exemplos** que servirá para introduzir (induzir) o conhecimento a ser aprendido.

O exemplo mais básico de aprendizado indutivo é o presente nos **Sistemas CBR (Case-Based Reasoning)**

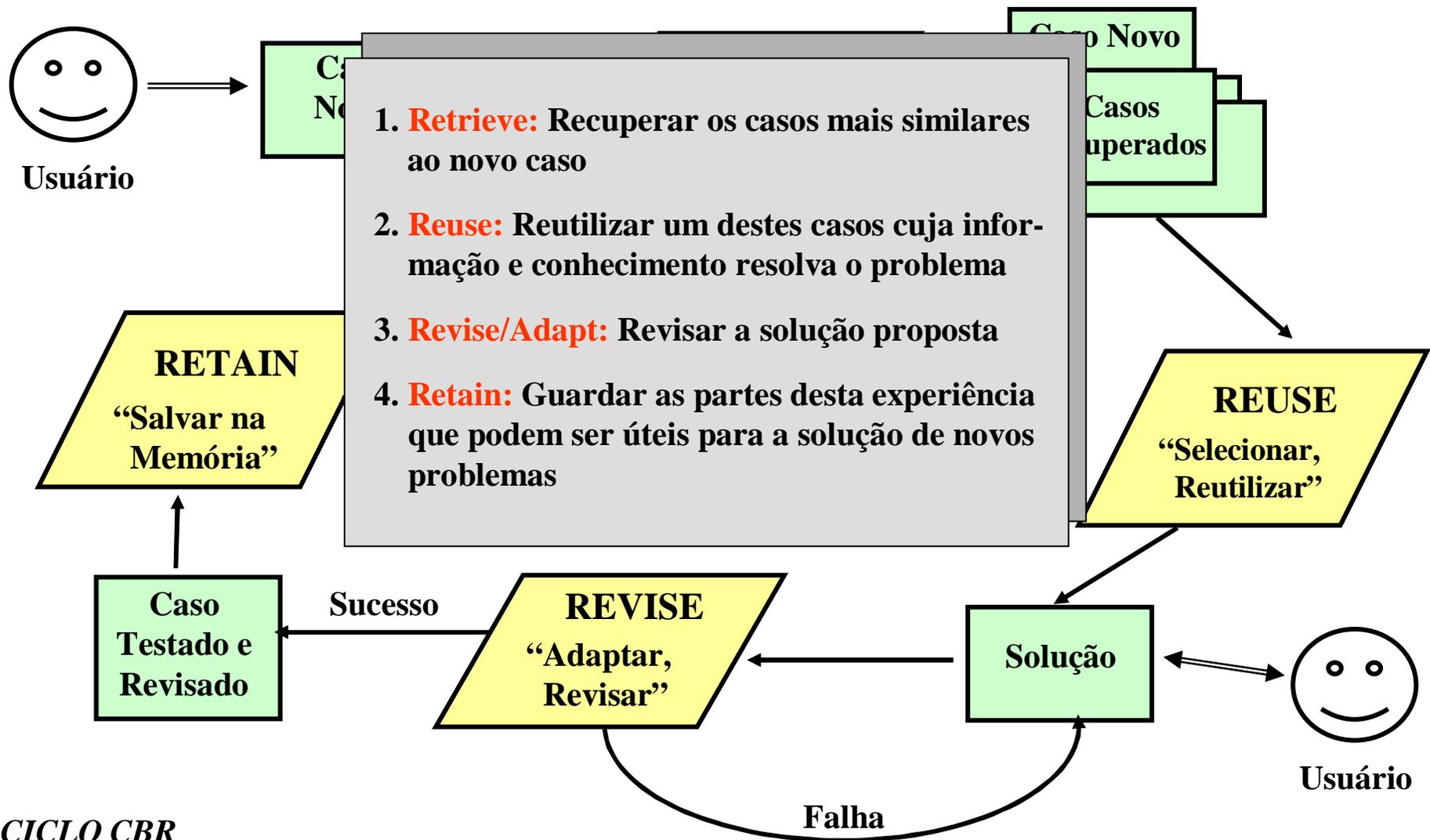
- > O que não implica que sua implementação não possa vir a ser complexa
- > e seus resultados interessantes

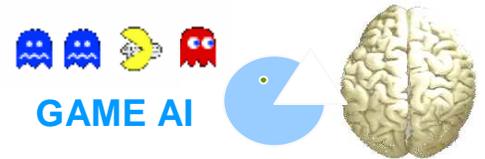
CBR – Case Based Reasoning



CICLO CBR

CBR – Case Based Reasoning





CBR – Case Based Reasoning

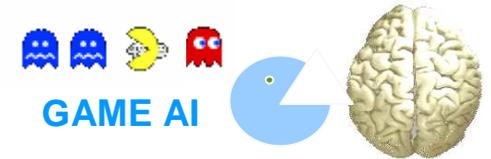
CBR em Jogos... do Xadrez ao Half Life!

- O computador Deep Blue possuía arquivadas partidas e jogadas importantes de jogos anteriores (“knowledge base of games played by grandmasters”);
- Existem citações sobre o uso de CBR em Jogos como...
 - **Space Invaders: Case-Based Plan Recognition in Computer Games.** Michael Fagan & Pádraig Cunningham. **Lecture Notes in Computer Science (LNCS): Springer-Verlag. Vol. 2689 / 2003. Jan. 2003.**
 - **Half Life** (<http://www.cs.tcd.ie/Padraig.Cunningham/nds101/>) (?)
 - **SimCity: Real-Time Case-Based Reasoning in a Complex World.** Fasciano, Mark J.; TR-96-05. CS Dept., Univ. of Chicago, Feb. 1996.
 - **BATTLE [Goo89], COACH[Col87]** apud Wangenheim 2003 (RBC).

inclusive sendo usado para acelerar simulações...

- **Using Case-Based Reasoning to Overcome High Computing Cost Interactive Simulations.** Javier Vázquez-Salceda, Miquel Sànchez-Marrè, Ulises Cortés
LNCS: Springer-Verlag Heidelberg. Vol. 2689 / 2003. Jan. 2003.

ML4Games



CBR – Case Based Reasoning

CBR em Jogos... Exemplo (para discussão)

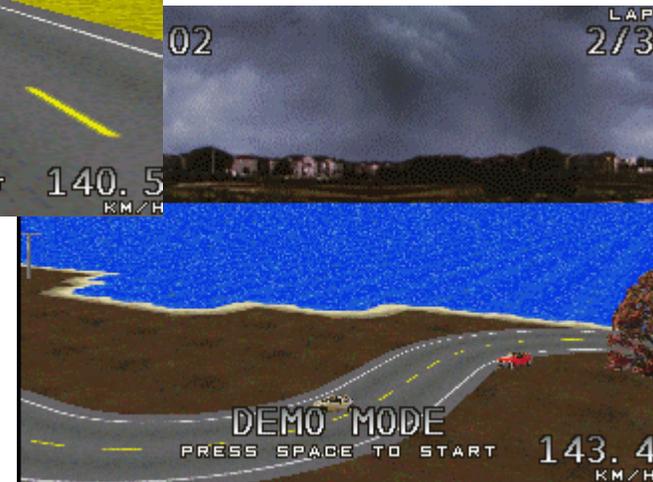


Estratégias:
Memória de
Casos...

Possibilidades



Gravando um motorista...

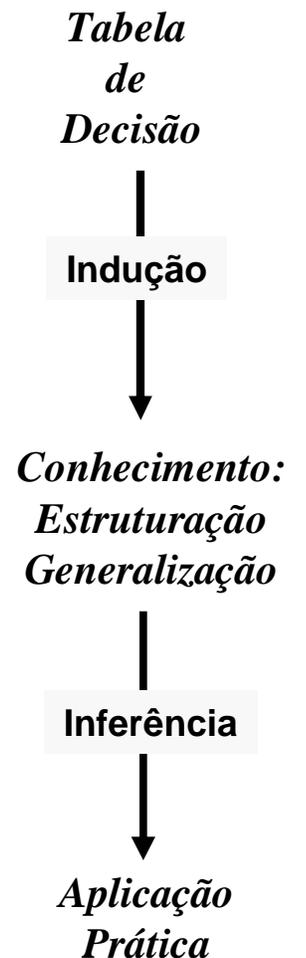


Uso das posições na pista...
Adaptação de casos...

IDT – Induction of Decision Trees (Árvores de Decisão)

➤ Aprendizado Supervisionado

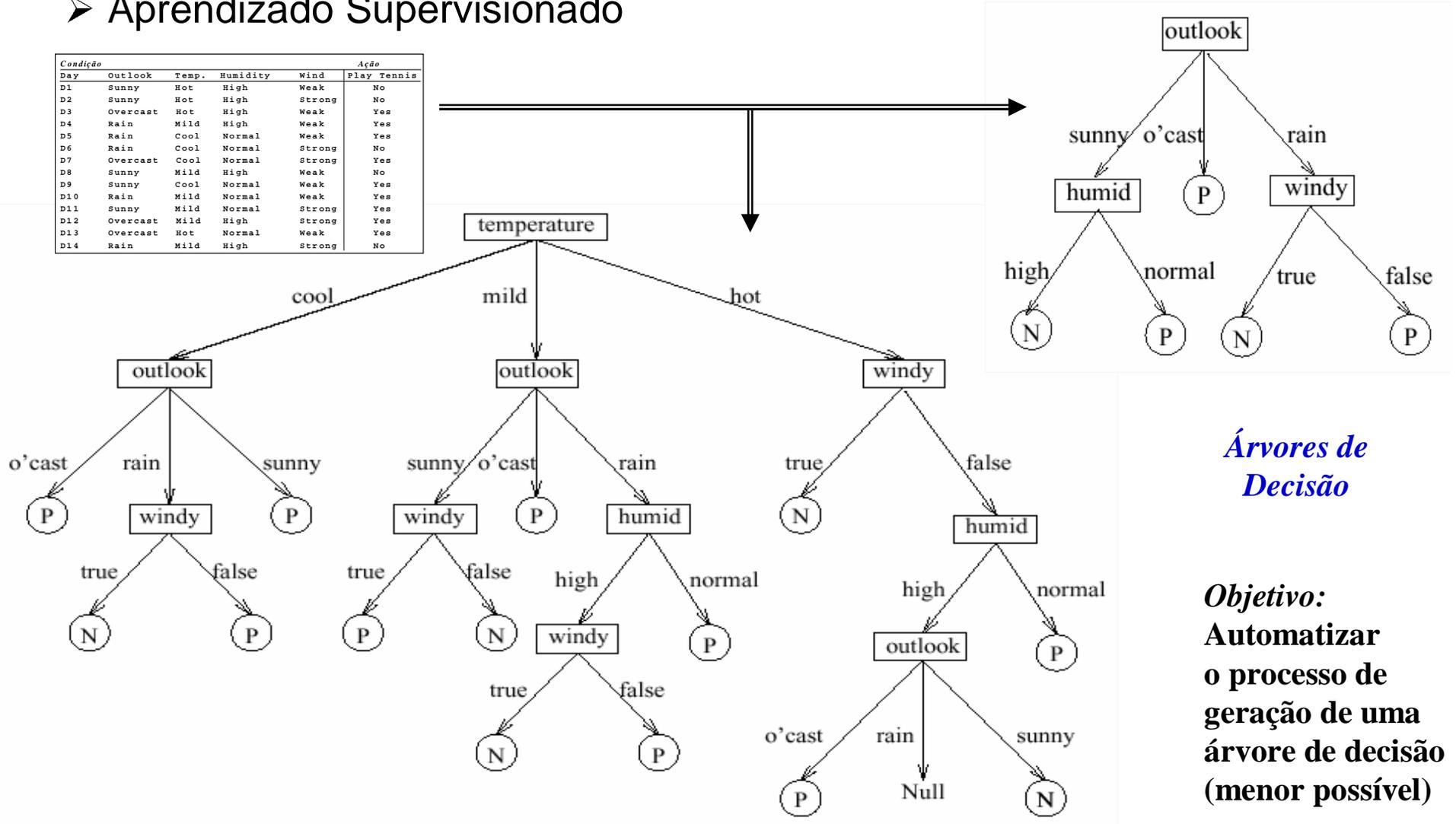
<i>Condição</i>					<i>Ação</i>
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



IDT – Induction of Decision Trees (Árvores de Decisão)

➤ Aprendizado Supervisionado

Condição	Outlook	Temp.	Humidity	Wind	Ação
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Árvores de Decisão

Objetivo:
Automatizar o processo de geração de uma árvore de decisão (menor possível)

IDT – Induction of Decision Trees (Árvores de Decisão)

➤ Aprendizado Supervisionado

Algoritmos *IDT*:

- **ID3 - Quinlan / 1986**
- **C4.5 - Quinlan / 1993**
<http://www.cse.unsw.edu.au/~quinlan/> (Atualmente See5)
- **ASSISTANT - Cestnik, Kononenko, Bratko / 1987**
- **CART (Classification and Regression Trees)**
Breitman et al / 1984
- **CN2 - Clark, Niblett / 1987**
<http://www.cs.utexas.edu/users/pclark/software.html>
- **ID5R, ITI (Incremental DT induction) - Utgoff / 1989**
<http://www.cs.umass.edu/~lrn/iti/>
- **Métodos similares... Família AQ - Michalski**
<http://www.mli.gmu.edu/msoftware.html>

Árvores de Decisão

Algoritmo Básico:

Selecionar o atributo que melhor discrimine os dados em relação as classes definidas.

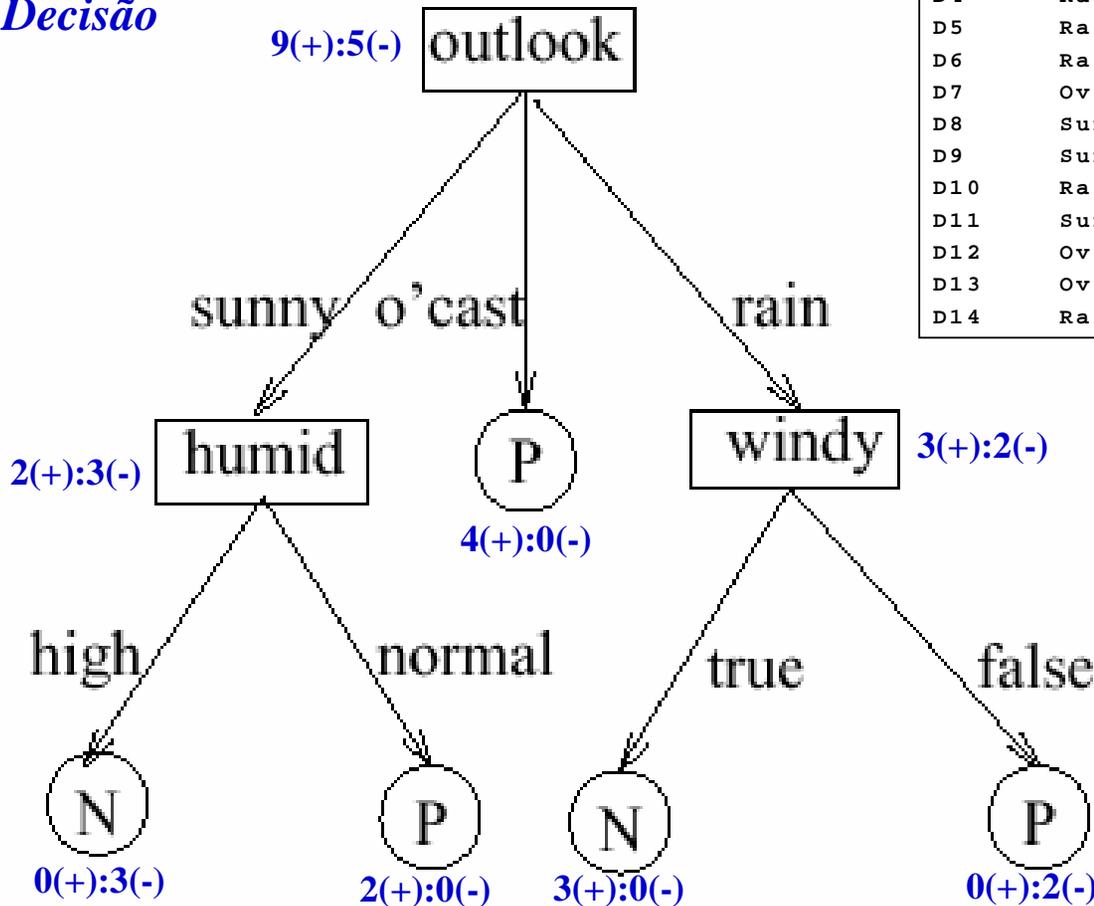
C4.5: utiliza uma medida baseada na entropia (“grau de mistura dos dados”), obtendo para cada atributo o ganho (“redução da mistura”) caso este seja selecionado como atributo discriminante.

IDT – Induction of Decision Trees (Árvores de Decisão)

➤ Aprendizado Supervisionado

Árvores de Decisão

Condição					Ação
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Nodo = Atributo testado
Aresta = Valor do atributo

Nodos: Está indicada a contagem total de YES (+) e de NO (-)

Regras obtidas: (exemplos)

IF Outlook = Overcast
Then Yes

IF Outlook = rain AND Windy = True
Then No

IDT – Induction of Decision Trees (Árvores de Decisão)

➤ Aprendizado Supervisionado

Características das Árvores de Decisão:

- **Aprendizado fácil e rápido**
- **Fácil integração em outras aplicações (obtenção de regras)**
- **Apesar de permitirem trabalhar com atributos quantitativos (discretizados de forma automática, sendo codificados como se fossem atributos nominais), as árvores de decisão são melhores na manipulação de **informações simbólicas****

SOFTWARES:

- C4.5 <http://www.cse.unsw.edu.au/~quinlan/>
- Dtree <http://fuzzy.cs.uni-magdeburg.de/~borgelt/>
- Weka <http://www.cs.waikato.ac.nz/ml/weka/>
- R <http://www.r-project.org/>
- CN2 <http://www.cs.utexas.edu/users/pclark/software.html>
- ITI <http://www.cs.umass.edu/~lrn/iti/>
- AQ Family <http://www.mli.gmu.edu/msoftware.html>
- See5 <http://www.rulequest.com/>
- CART <http://www.salford-systems.com/>
- Clementine <http://www.isl.co.uk/clementine.html>
<http://www.spss.com/clementine/>

IDT – Induction of Decision Trees (Árvores de Decisão)

➤ IDT em Jogos...

Árvores de Decisão – Usando o Conhecimento Adquirido

Decision Tree:

outlook = overcast: Yes (3.2)
outlook = sunny: ← Atributo Numérico
| humidity <= 75 : Yes (2.0)
| humidity > 75 :
| | windy = false: No (2.0)
| | windy = true:
| | | temperature <= 75 : Yes (0.4)
| | | temperature > 75 : No (1.0)
outlook = rain:
| windy = false: Yes (3.0)
| windy = true:
| | temperature <= 71 : No (2.0)
| | temperature > 71 : Yes (0.4)

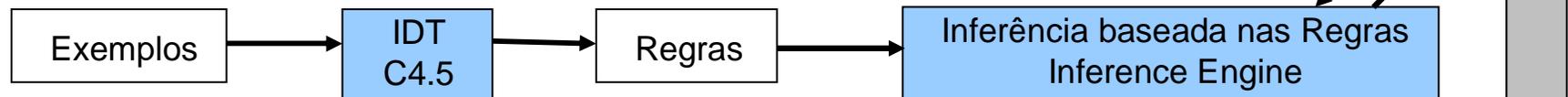
Regras IF/THEN

IF Outlook = Overcast Then Yes
IF Outlook = Rain AND Windy = False Then Yes
IF Outlook = Sunny AND Humidity <= 75 Then Yes
...

Production Rule Classifier

Rule 1:
outlook = sunny
-> class Play [100.0%]

Default class: Play



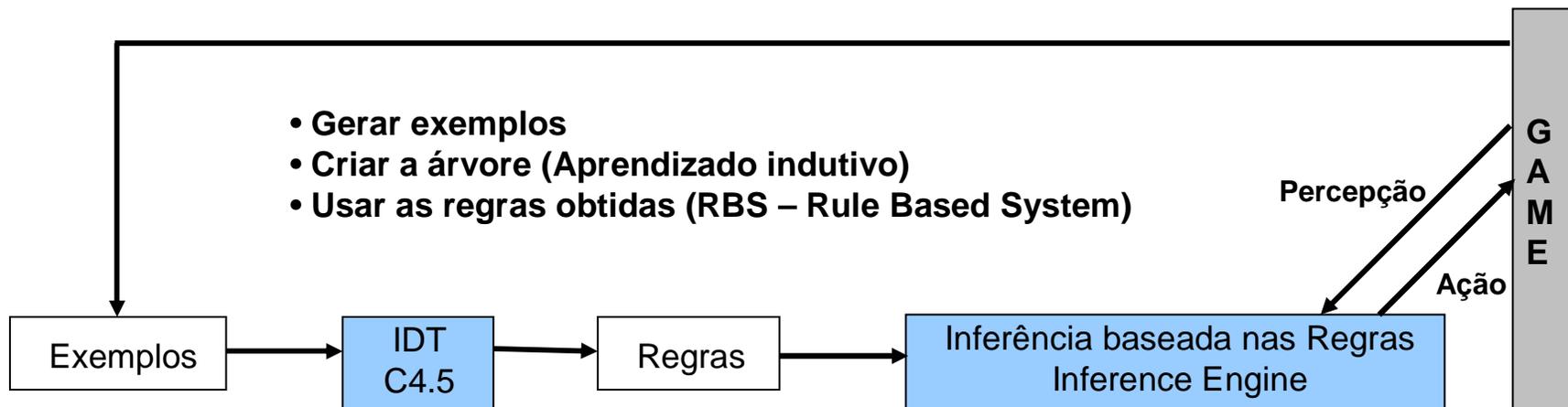
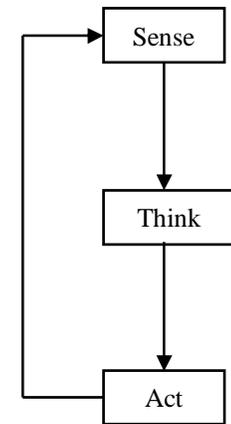
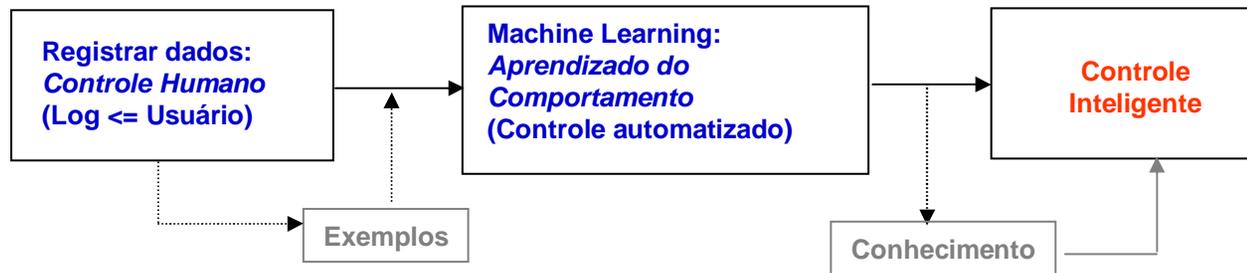
ML4Games



IDT – Induction of Decision Trees (Árvores de Decisão)

➤ IDT em Jogos...

Árvores de Decisão – Usando o Conhecimento Adquirido



IDT – Induction of Decision Trees (Árvores de Decisão)

➤ IDT em Jogos...

Aplicações possíveis - Exemplos...

- **Robocup - CMUnited RoboCup-98 Simulator**

Peter Stone and Manuela Veloso.

Using decision tree confidence factors for multiagent control.

In Proceedings of the First International Workshop on RoboCup, Nagoya, Japan, August 1997.

<http://www.cs.utexas.edu/users/pstone/papers.html>

- **Aquisição de Conhecimento para Agentes**

Yusuf Pisan

Building Characters: A Form of Knowledge Acquisition

AAAI Spring Symposium on AI and Interactive Entertainment. 2000.

<http://www.qrg.northwestern.edu/aigames.org/pastpapers.html>

- **Jogos de Tabuleiro**

- **Agentes baseados em Regras**

Ferramenta necessária: *Inference Engine / RBS (Rule Based System)*

Levantamento do Perfil de Usuários (“opponentes”)

➤ Aprendizado Não-Supervisionado

Determinando o comportamento de um agente humano...

- **Concepção de Um Ambiente Virtual Inteligente e Adaptativo**
Cássia Trojahn dos Santos, Fernando Osório
Revista Scientia – Unisinos (Artigo Submetido – V.14 N.2 – 2003)

Definição: Modelo de Usuário (Perfil) [Santos03 / SBIE]

Um modelo de usuário é uma representação explícita das características, preferências e necessidades de um usuário. Conforme Kobsa (1995), é uma coleção de informações e suposições sobre usuários individuais ou sobre grupos de usuários, necessárias para que o sistema adapte diversos aspectos de suas funcionalidades e interface. Compreende toda a informação sobre o usuário, extraída da sua interação com o sistema [Abbattista et al, 2002]. O termo perfil de usuário [Wasfi, 1999] também é utilizado com o propósito de representar o conhecimento sobre o usuário. Ao processo de aquisição e representação de um modelo dá-se o nome de modelagem de usuário.



Levantamento do Perfil de Usuários (“opponentes”)

➤ Aprendizado Não-Supervisionado

Aquisição do Modelo de Usuário [Santos03 / PIPCA]*

Os métodos para coleta dos dados podem ser agrupados em dois conjuntos:

- Explícitos
- Implícitos.

Os métodos explícitos coletam as informações diretamente do usuário, questionando-o sobre os seus interesses, preferências e necessidades. Este tipo de coleta, geralmente, é feita através do uso de formulários (identificação do perfil inicial do usuário).

Os métodos implícitos inferem informações dos usuários através do monitoramento do comportamento durante a interação com o sistema (atualização do perfil do usuário).

Exemplo:

Modelo de Usuário => Conservador (informado)

Modelo de Usuário => Agressivo (após adaptação ~ análise do comportamento)

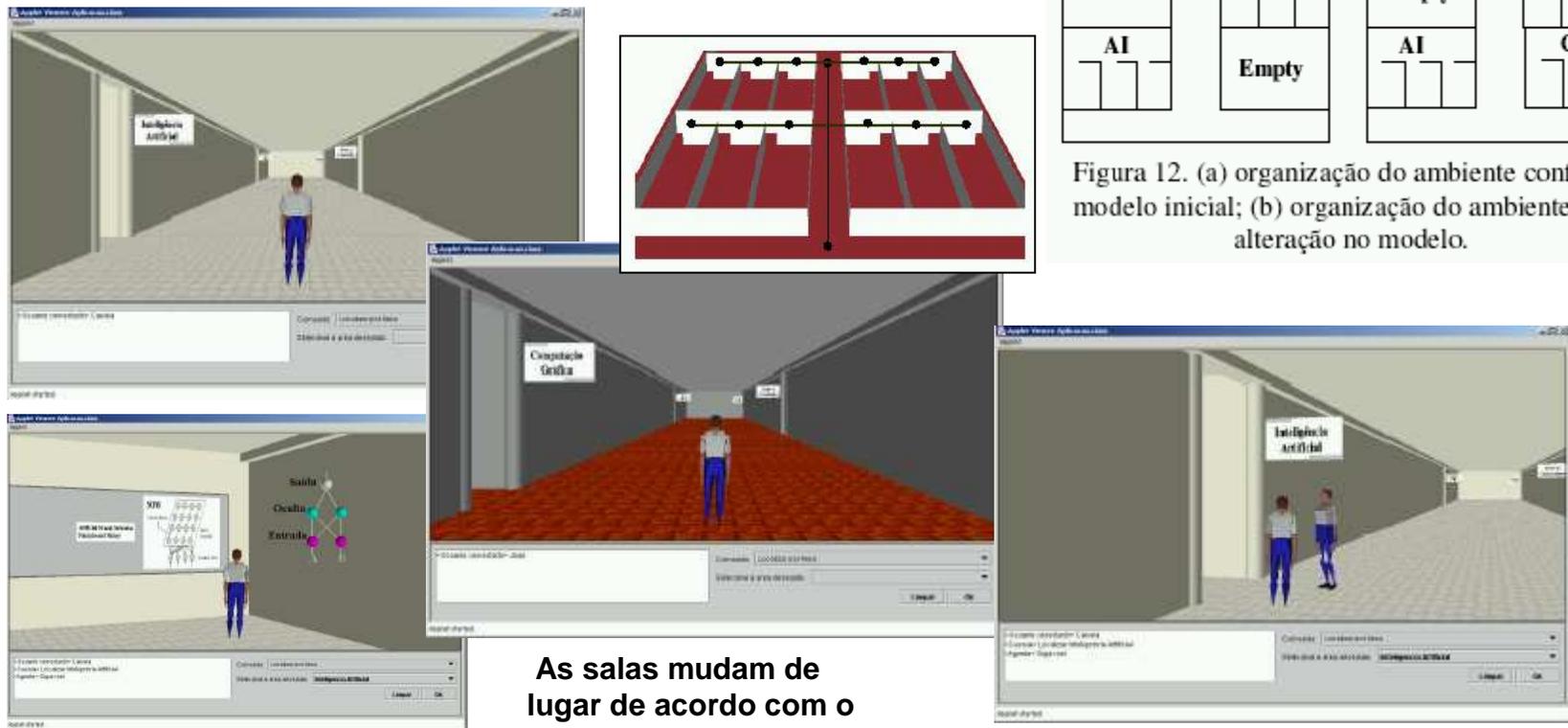
* Mais detalhes podem ser encontrados nos trabalhos relacionados a dissertação de mestrado de Cássia T. dos Santos

Levantamento do Perfil de Usuários (“oponentes”)

➤ Aprendizado Não-Supervisionado

Exemplo de Aplicação [Santos03 / PIPCA]

Ambiente Tridimensional Virtual e Adaptativo



As salas mudam de lugar de acordo com o perfil do usuário (interesses)

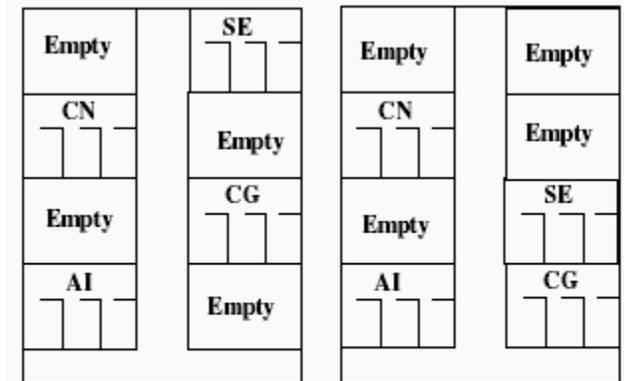


Figura 12. (a) organização do ambiente conforme modelo inicial; (b) organização do ambiente após alteração no modelo.

Levantamento do Perfil de Usuários (“opponentes”)

➤ Aprendizado Não-Supervisionado

Adaptação do Modelo de Usuário [Santos03 / PIPCA]

A atualização do modelo está baseada no uso de regras e fatores de certeza (CF):

SE Evidência(s)

ENTÃO Hipótese com CF = x

Neste trabalho, as evidências correspondem às áreas do ambiente visitadas e aos conteúdos solicitados (a partir de um mecanismo de busca) e acessados (clicados) pelo usuário. O interesse inicial do usuário por determinada área (valor inicial de $P(H)$) é obtido a partir da coleta explícita de dados e pode ser alterado durante o processo de atualização do modelo, com base em limiares de aumento e diminuição.

Os CFs associam medidas de crença (MB) e descrença (MD) em uma hipótese (H), dada uma evidência (E). Um fator de certeza 1 indica crença total em uma hipótese, enquanto -1 corresponde a descrença total. O cálculo do fator de certeza é realizado a partir das seguintes fórmulas:

$$CF = \frac{MB - MD}{1 - \min(MB, MD)}$$
$$MB \begin{cases} 1 & \text{if } P(H) = 1 \\ \frac{\max[P(H|E), P(H)] - P(H)}{1 - P(H)} & \text{otherwise} \end{cases}$$
$$MD \begin{cases} 1 & \text{if } P(H) = 0 \\ \frac{\min[P(H|E), P(H)] - P(H)}{0 - P(H)} & \text{otherwise} \end{cases}$$

Levantamento do Perfil de Usuários (“opponentes”)

➤ Aprendizado Não-Supervisionado

Adaptação do Modelo de Usuário [Santos03 / PIPCA]

As evidências são usadas para inferir a hipótese de interesse do usuário em cada área do conhecimento, a partir das regras e dos fatores de certeza correspondentes. Para a atualização do modelo, as seguintes regras foram definidas:

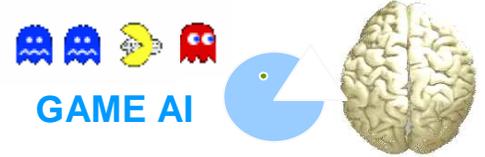
SE solicitou ENTÃO interesse em Y com CF = x

SE navegou ENTÃO interesse em Y com CF = x

SE acessou ENTÃO interesse em Y com CF = x

A cada N sessões, para cada área, são verificadas as evidências (navegação, solicitação e acesso) e feitas as inferências de interesse, com base nas regras apresentadas acima. A partir da ordenação dos fatores de certeza resultantes é estabelecido um ranking das áreas de interesse do usuário, através do qual é possível verificar as alterações no modelo inicial (obtido a partir da coleta explícita) e atualizar o modelo.

A partir dessa atualização (alterar perfil = alterar os fatores de certeza sobre o interesse em uma determinada área), é feita a re-organização do ambiente.



Levantamento do Perfil de Usuários (“opponentes”)

➤ Modelo de Usuário em Jogos... [Jogos de Ação](#)

- **Modelo de Comportamento...**

- **Tipo de Usuário: Agressivo, Impulsivo, Conservador, ...**
- **Comportamento no ambiente: Lugares que costuma visitar**
- **Análise de ações do usuário: Ataque, Defesa**

- **Baseado no comportamento...**

- **Criar estratégias de jogo**
- **Antecipar movimentos e ações**

➤ Modelo de Usuário em Jogos... [Jogos de Estratégia e Simulação](#)

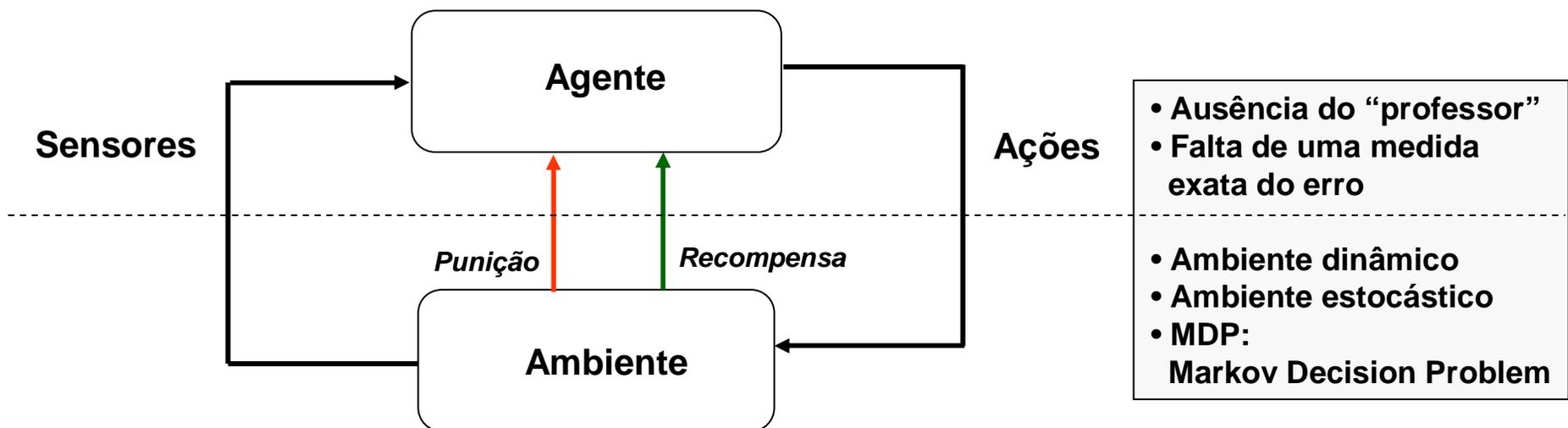
- **Modelo de Comportamento é Importante / Indispensável**

Aprendizado por Reforço (Reinforcement Learning)

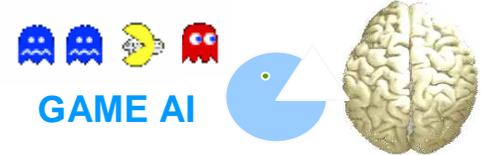
➤ Aprendizado Semi-Supervisionado

Referências:

- Artificial Intelligence: A Modern Approach. Russell & Norvig, 1995 (cap. 20).
- A Standart Interface for Reinforcement Learning Software in C++. Sutton & Samtamaria (Umass / 2003)
- Reinforcement Learning: A Survey. Kaelbling, Littman & Moore, Journal of Artificial Intelligence Research 4 (1996) 237-285



ML4Games

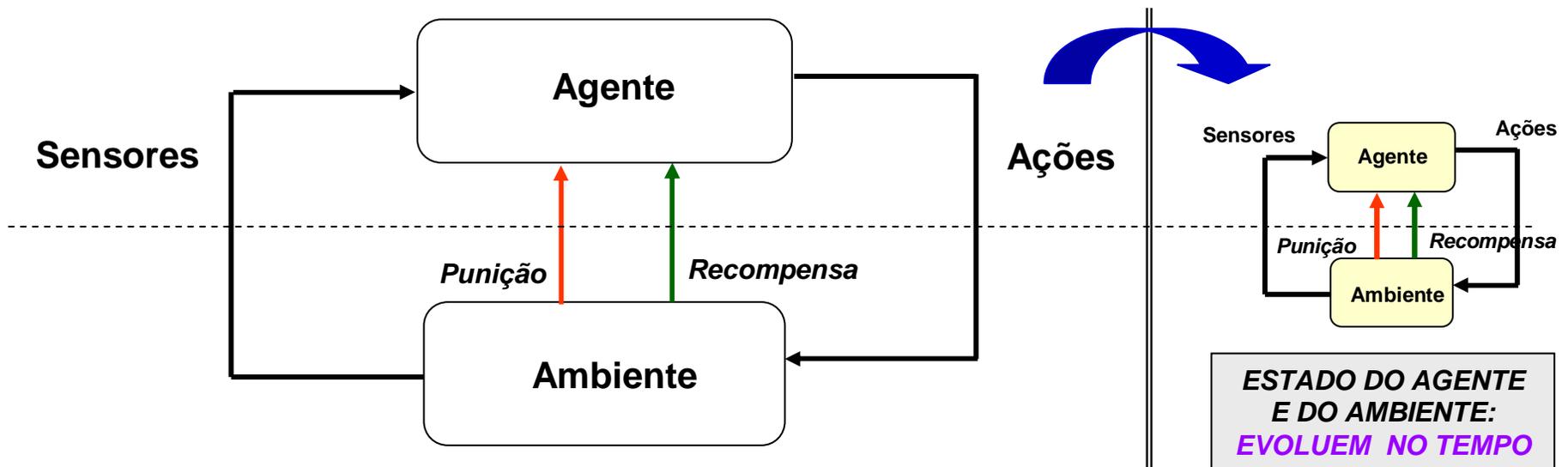


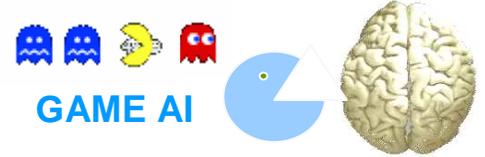
Aprendizado por Reforço (Reinforcement Learning)

➤ Aprendizado Semi-Supervisionado

Referências:

- Artificial Intelligence: A Modern Approach. Russell & Norvig, 1995 (cap. 20).
- A Standart Interface for Reinforcement Learning Software in C++. Sutton & Samtamaria (Umass / 2003)
- Reinforcement Learning: A Survey. Kaelbling, Littman & Moore, Journal of Artificial Intelligence Research 4 (1996) 237-285





Aprendizado por Reforço (Reinforcement Learning)

➤ Aprendizado Semi-Supervisionado

Conceitos Básicos:

- Aprendizado baseado no

Sucesso / Falha

Punição x Recompensa (*Punish x Reward*)

Estímulos: Pain / Pleasure - Food / Hunger

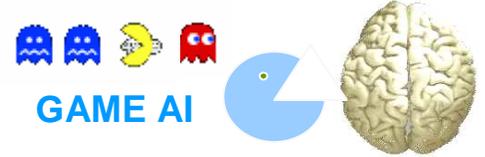
[Pavlov – 1849/1936]

- Aplicação em Jogos de Tabuleiro...

Xadrez: Aprendizado com professor => Imitar o professor!

Indica qual jogada faria...

Aprendizado sem professor => ?



Aprendizado por Reforço (Reinforcement Learning)

➤ Aprendizado Semi-Supervisionado

Conceitos Básicos:

- Aprendizado baseado no

Sucesso / Falha

Punição x Recompensa (*Punish x Reward*)

Estímulos: Pain / Pleasure - Food / Hunger

[Pavlov – 1849/1936]

- Aplicação em Jogos de Tabuleiro...

Aprendizado sem professor => Random Moves

Status: End of the Game (Terminal State)

Retorno sobre desempenho: *Feed-Back*

Vitória: Sucesso / Perda: Fracasso

- Diferencial:

Não copia o professor

Tentativa de melhorar a performance (tentativa/erro)

Aprendizado por Reforço (Reinforcement Learning)

➤ Aprendizado Semi-Supervisionado

Tipos de Aprendizagem por Reforço... segundo:

- Ambiente:
Acessível (sensores) ou Inacessível (estado interno)
- Recompensa:
Em qualquer estado
Apenas no estado final
- Agente:
Aprendiz passivo (observa passivamente o resultado das ações)
Aprendiz ativo (sugere novas ações)
- Função:
Evaluation Utility function - U(i) ou *Action-Value function*

ADP

Adaptive Dynamic Programming

$$U(i) = R(i) + \sum_j M_{ij} U(j) \quad : M_{ij} = \text{Probabilidades de Transição}$$

TD

Temporal Difference Learning

$$U(i) \leftarrow U(i) + \alpha (R(i) + U(j) - U(i))$$

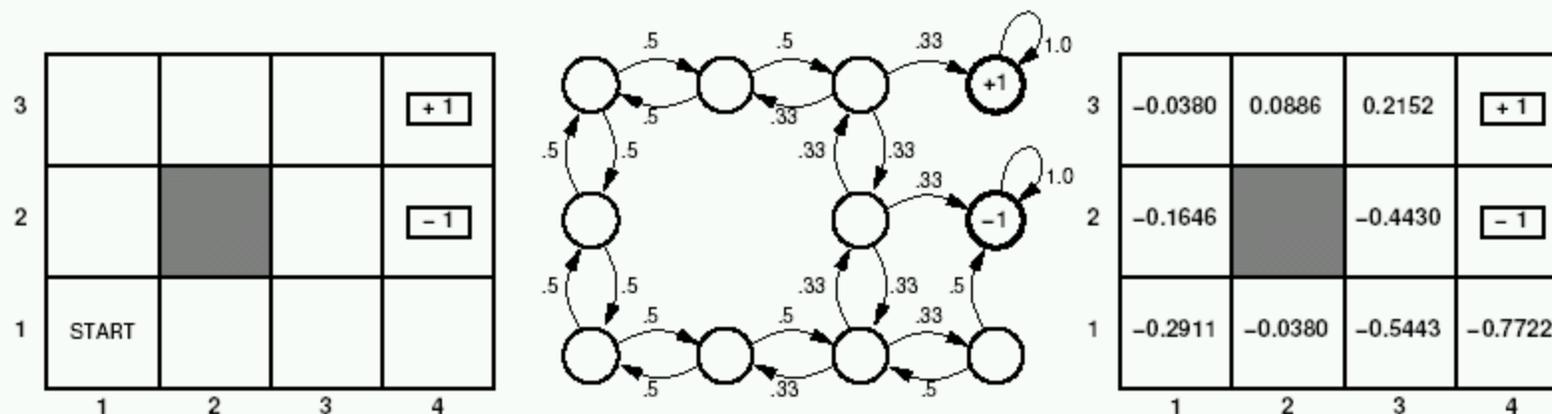
α = Learning Rate

Aprendizado por Reforço (Reinforcement Learning)

➤ Aprendizado Semi-Supervisionado

Artificial Intelligence: A Modern Approach. Russell & Norvig, 1995 (capit. 20)

Exemplo:



Seqüências de Treinamento:

- (1,1) > (1,2) > (1,3) > (1,2) > (1,3) > (1,2) > (1,1) > (2,1) > (3,1) > (4,1) > (4,2) > -1
- (1,1) > (1,2) > (1,3) > (2,3) > (3,3) > (4,3) > +1
- (1,1) > (1,2) > (1,1) > (1,2) > (1,1) > (2,1) > (3,1) > (3,2) > (4,2) > -1
- (1,1) > (1,2) > (1,1) > (1,2) > (1,3) > (2,3) > (1,3) > (2,3) > (3,3) > (4,3) > +1
- (1,1) > (2,1) > (3,1) > (2,1) > (1,1) > (1,2) > (1,3) > (2,3) > (3,3) > (4,3) > +1
- (1,1) > (2,1) > (1,1) > (1,2) > (1,3) > (2,3) > (3,3) > (3,2) > (4,2) > -1

Aprendizado por Reforço (Reinforcement Learning)

➤ Aprendizado Semi-Supervisionado

Q-Learning: Learning an Action-Value Function

“The agent learns an action-value function giving the “expected utility” of taking a given action in a given state – **Q(a,i)**”

$$U(i) = \text{Max}_a Q(a,i)$$

Q-Learning / ADP

Adaptive Dynamic Programming

$$Q(a,i) = R(i) + \sum_j M_{ij}^a \text{Max}_{a'} Q(a',j)$$

$\therefore M^{a_{ij}}$ = Probabilidade de Transição do estado “i” ao “j” para uma dada ação “a”

Q-Learning / TD

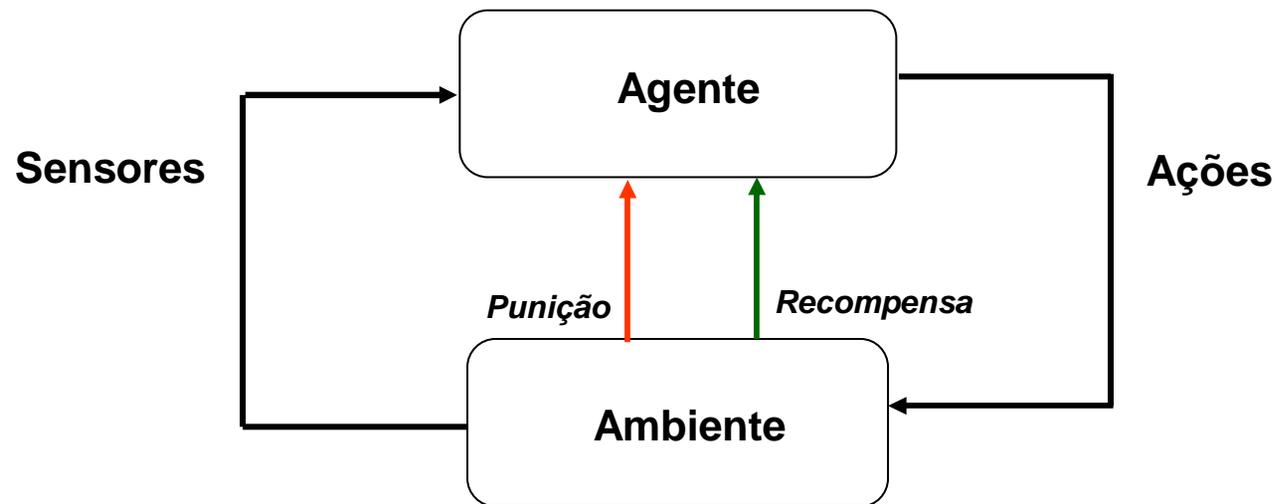
Temporal Difference Learning

$$Q(a,i) \leftarrow Q(a,i) + \alpha (R(i) + \text{Max}_{a'} Q(a',j) - Q(a,i)) \quad \therefore \alpha = \text{Learning Rate}$$

Aprendizado por Reforço (Reinforcement Learning)

➤ Reinforcement Learning em Jogos...

Aprendizado de Agentes baseado na punição e recompensa



Vantagens:

- Não precisa de um professor
- Possibilidade de realizar um aprendizado contínuo

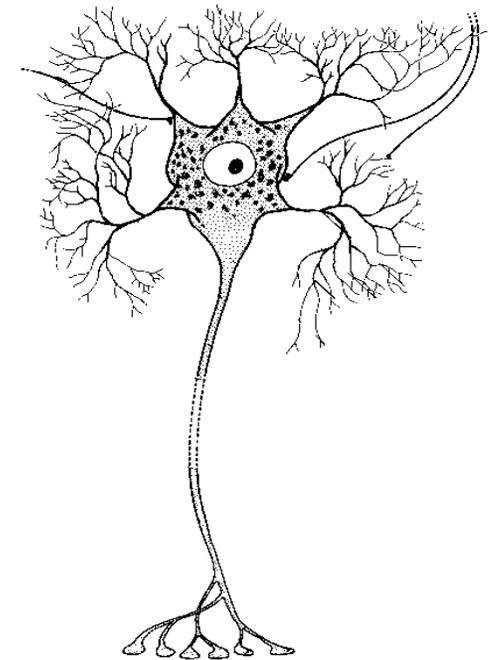
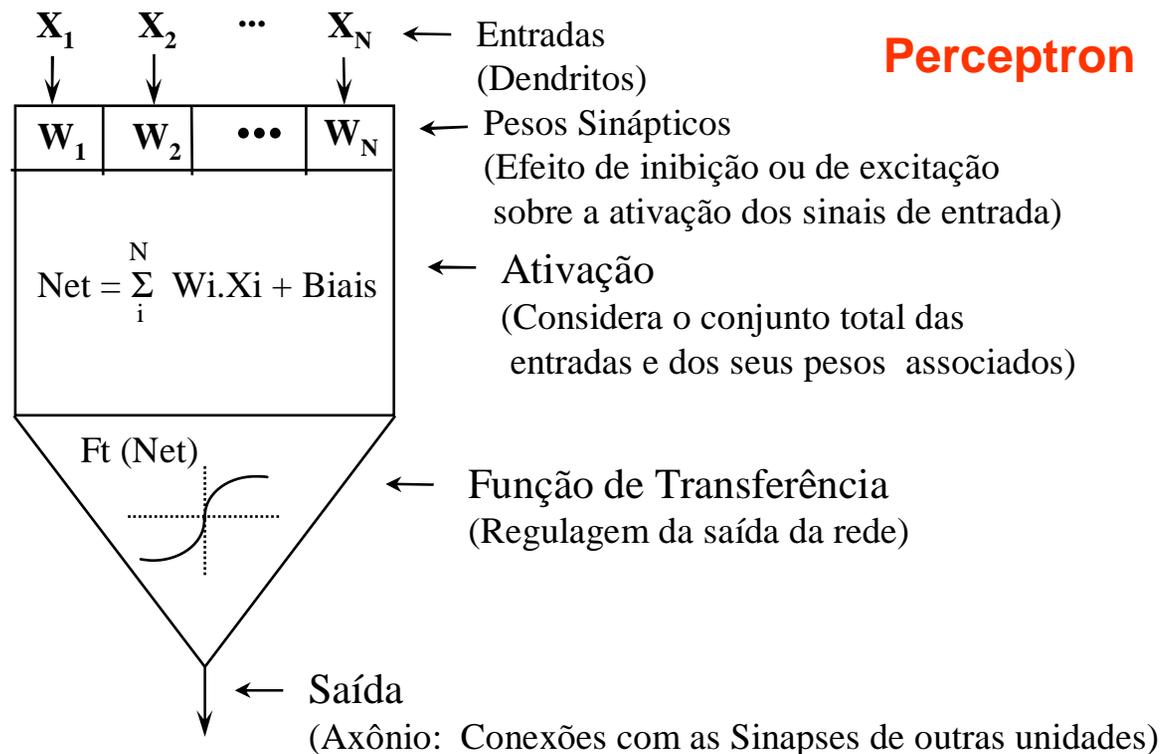
Demo: <http://reinforcementlearning.ai-depot.com/Programs.html>



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ Aprendizado Supervisionado

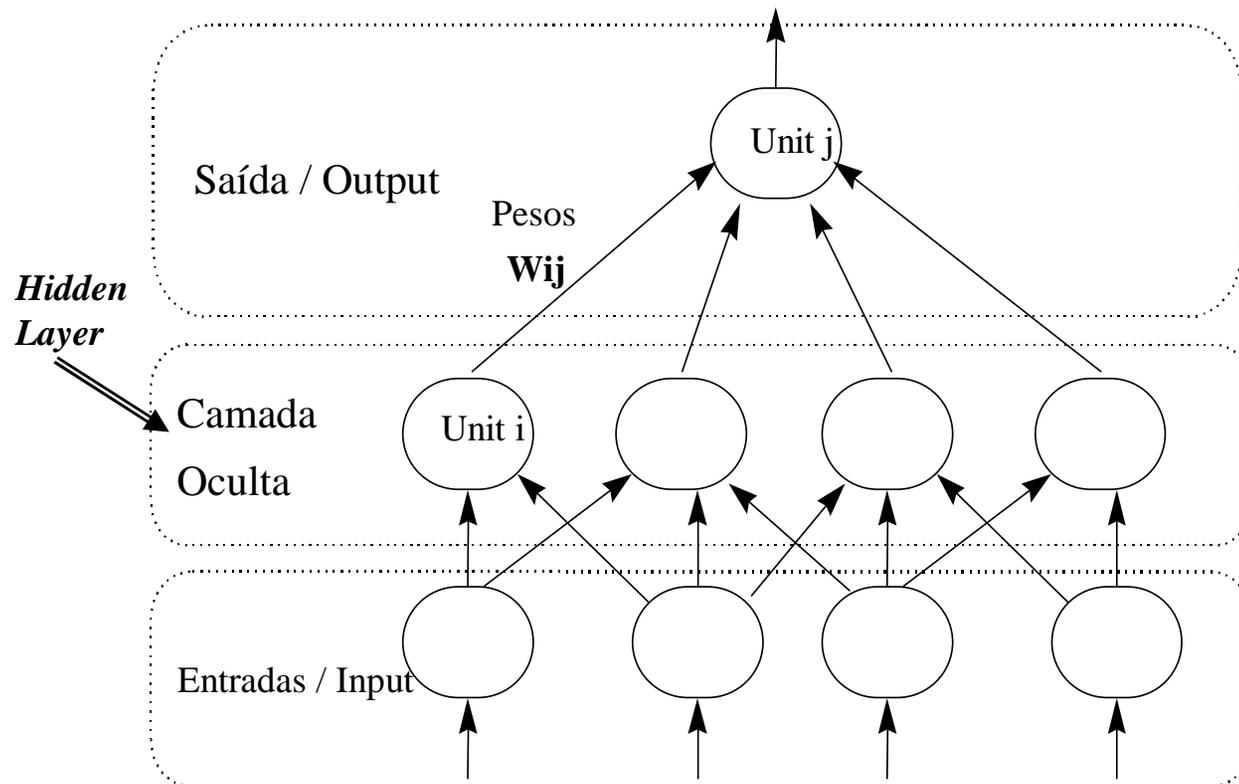
- MLP – Multi-Layer Perceptron / Algoritmo Back-Propagation



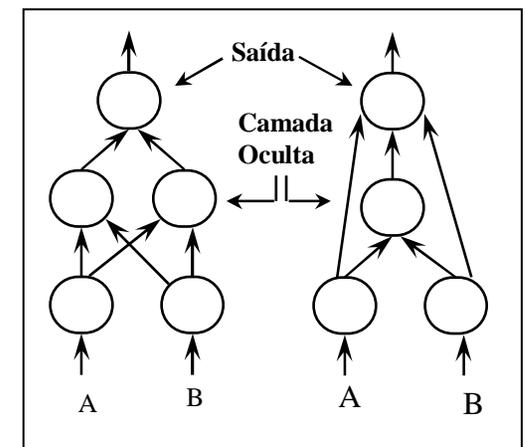
ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ Aprendizado Supervisionado

- MLP – Multi-Layer Perceptron / Algoritmo Back-Propagation



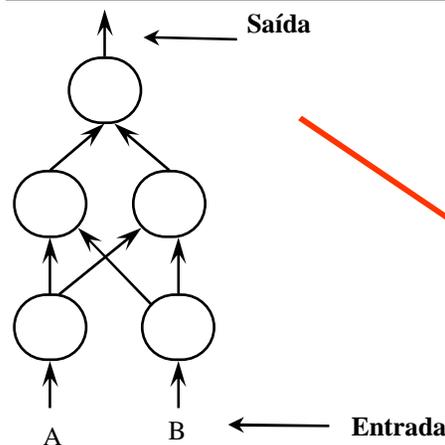
MLP
*Perceptron de
Múltiplas Camadas*



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ Aprendizado Supervisionado

- MLP – Multi-Layer Perceptron / Algoritmo Back-Propagation



Classificação / Regressão
(não linear)

Saída da Rede: A_i
Saída dos Exemplos: D_i

Erro quadrático:

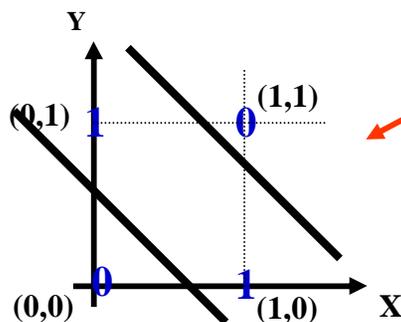
$$E = \frac{1}{2} \sum_i (D_i - A_i)^2$$

Aprendizado:
Minimizar
Erro Quadrático
na saída de rede

Exemplos

X, Y: Entradas Di (Xor)

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0



$$\text{Saída} = W_1 \cdot X + W_2 \cdot Y + C$$

Se Saída > 0 então Classe 1

Se Saída ≤ 0 então Classe 0

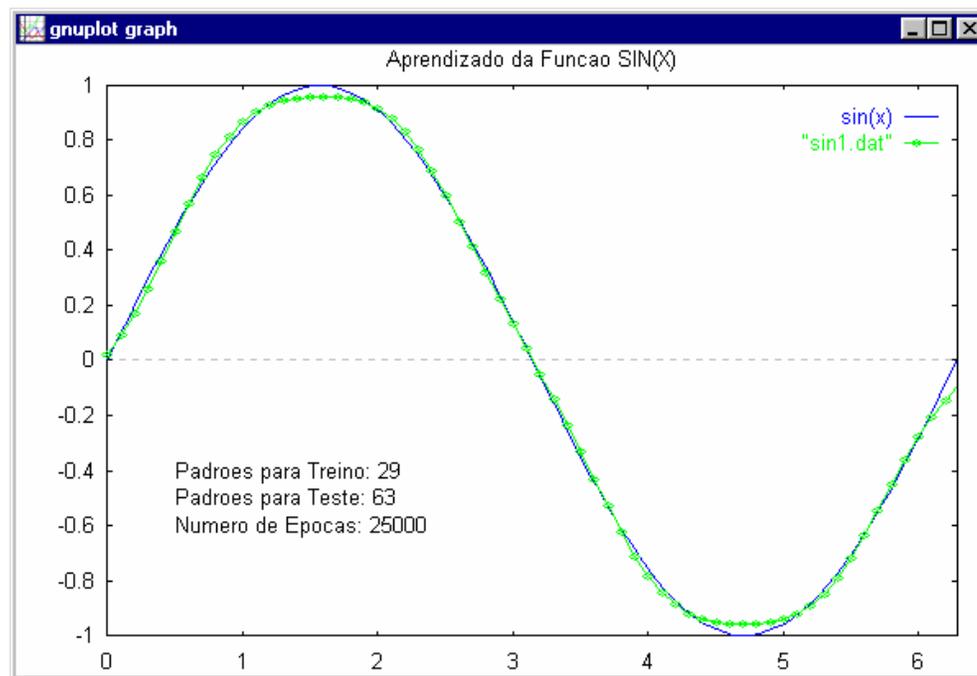
Determinar w_1 e w_2 que satisfaça as restrições...
(várias retas, várias restrições/exemplos)

Aprendizado: Ajuste dos Pesos
Processo Iterativo de Otimização

ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ Aprendizado Supervisionado

- MLP – Multi-Layer Perceptron / Algoritmo Back-Propagation



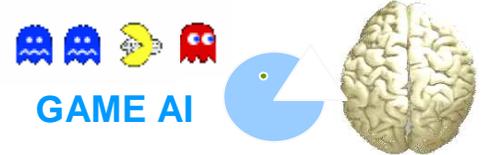
Saída de Rede

Base de Aprendizado

**Regressão:
Aproximando a
Função Seno...**

**Entrada:
X = Ângulo
Saída:
F(X) = Sin(X)**

0.0	0.000000
0.3	0.147760
0.6	0.282321
0.9	0.391663
1.05	0.433712
1.2	0.466020
1.35	0.487862
1.5	0.498747
1.65	0.498433
1.8	0.486924
1.95	0.464480
2.1	0.431605
2.4	0.337732
2.7	0.213690
3.0	0.070560
3.3	-0.078873
3.6	-0.221260
3.9	-0.343883
4.2	-0.435788
4.35	-0.467526
4.5	-0.488765
4.65	-0.499027
4.8	-0.498082
4.95	-0.485952
5.1	-0.462907
5.25	-0.429470
5.4	-0.386382
5.7	-0.275343
6.0	-0.139708



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ Aprendizado Supervisionado

- Na prática...

- 1) Criar uma base de dados com exemplos de entradas e saídas
- 2) Especificar uma topologia de rede neural
- 3) Definir os parâmetros de aprendizado
- 4) Simular... Executar o aprendizado até minimizar o erro em um “nível aceitável”, ou, rever base de dados, topologia e parâmetros

Simuladores a considerar...

SNNS <http://www-ra.informatik.uni-tuebingen.de/SNNS/> (Linux)
JNNS Java SNNS (Windows)

Ferramentas...

Interface Gráfica (JNNS / Xgui)

Linguagem de Script (batchman – Linux / Windows-Dos)

Gerador de Aplicações (snns2c – Linux / Windows-Dos)

ML4Games

ANN: SNNS



The screenshot displays the JavaNNS software interface. The main window shows a neural network diagram for an XOR problem. The network has two input nodes (in_1 and in_2), one hidden node, and one output node (result). The weights are: in_1 to hidden (4.596), in_2 to hidden (4.602), in_1 to result (4.958), in_2 to result (4.956), and hidden to result (-11.691). The current values are: in_1 (1), in_2 (1), hidden (0.857), and result (0.099). A color scale on the left indicates the magnitude of units and links.

The 'About JavaNNS' dialog box is open, displaying the following text:

Java Neural Network Simulator 1.1
based on the Stuttgart Neuronal Network Simulator 4.2 kernel

Credits:
(in order of participation)

SNNS: Andreas Zell, Günter Mamier, Michael Vogt, Niels Mache, Tilman Sommer, Ralf Hübner, Michael Schmalzl, Tobias Soye, Sven Döring, Dietmar Poselt, Kai-Uwe Herrmann, Artemis Hatzigeorgiou;
external contributions: Martin Riedmiller, Heike Speckmann, Martin Reczko, Jamie DeCoster, Jochen Biederman, Joachim Danz, Christian Wehrfritz, Randolph Werner, Michael Berthold, Hans Rudolph

Java User Interface: Igor Fischer, Fabian Hennecke, Christian Bannes

Copyright © 1990-1995 IPVR, University of Stuttgart
Copyright © 1996-2002 WSI, University of Tübingen

The 'Control Panel' shows the learning function set to 'Backprop-Momentum' with parameters: $\eta = 0.2$, $\mu = 0.5$, $c = 0.1$, and $d_{max} = 0.1$. The number of cycles is 1000 and steps is 1. Buttons for 'Init', 'Learn Current', and 'Learn All' are visible.

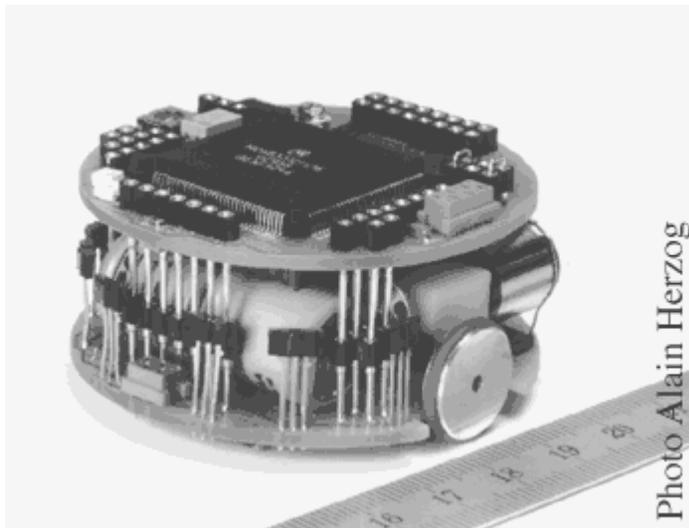
The 'Error graph' shows the error $\frac{1}{n} \sum e^2$ decreasing from approximately 1.1 to 0.1 over 2000 learning cycles.

The Windows taskbar at the bottom shows the following open applications: Iniciar, Pegasus Mail, Pesquisa Google: race..., C:\Arquivos de progra..., C:\WINNT\system32\c..., and JavaNNS. The system clock shows 15:51.

ANN – Artificial Neural Networks (Redes Neurais Artificiais)

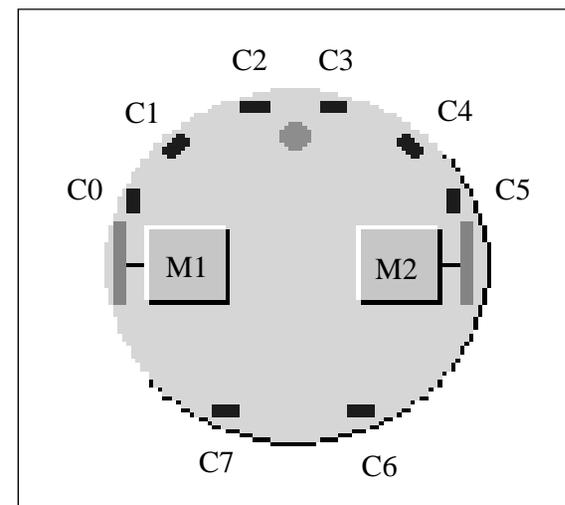
➤ ANN em Jogos...

- Agentes reativos: Robo Khepera



Controle Sensorial-Motor

Aprendizado



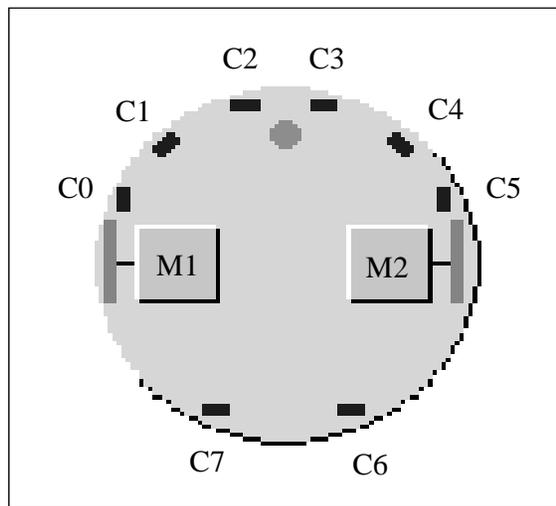
IF $S1 < Limite$ *and* $S2 < Limite$ *and*
 $S3 > Limite$ *and* $S4 > Limite$
THEN Action(Turn_Left)

IF $S2 > Limite$ *and*
 $S3 > Limite$ *and*
 $S2 > S3$ *and*
 $S1 > S4$
THEN Action(Turn_Right)

ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos...

- Agentes reativos: Robo Khepera



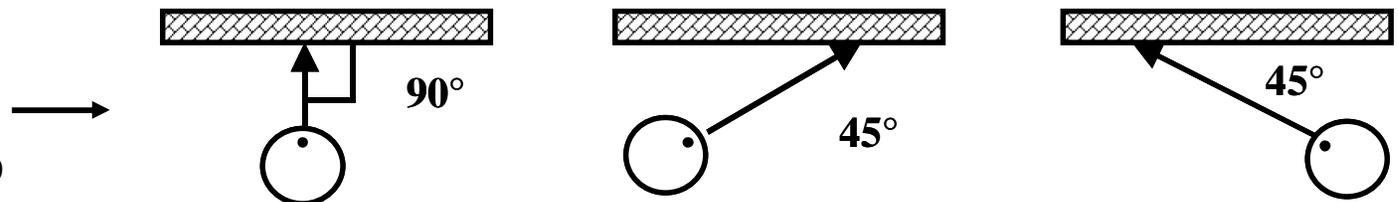
Controle Sensorial-Motor

Sensores: 8 - C0 à C7

Comandos: 3 ações (L=Turn Left, F=Forward, R=Turn Right)

C0	C1	C2	C3	C4	C5	C6	C7	L F R
4	1	4	3	6	3	0	4	0 1 0
1	3	3	6	6	6	6	6	0 1 0
6	3	234	104	772	96	3	6	1 0 0
2	6	229	104	724	107	3	4	1 0 0
563	1023	6	57	6	0	3	1	0 0 1
544	1023	3	1	4	0	5	1	0 0 1

Situações de Aprendizado

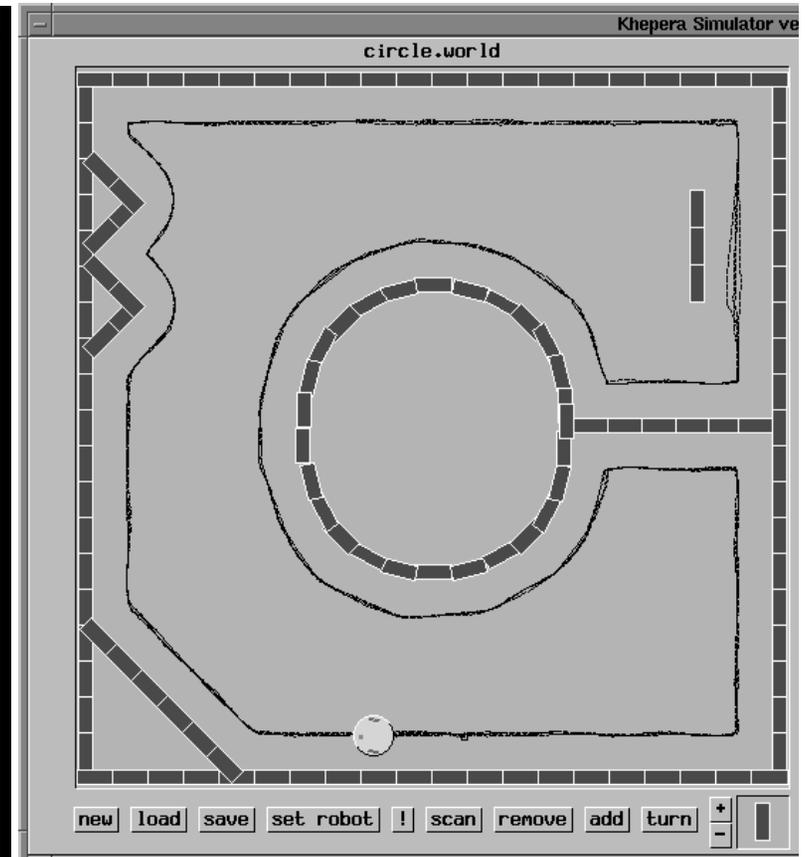
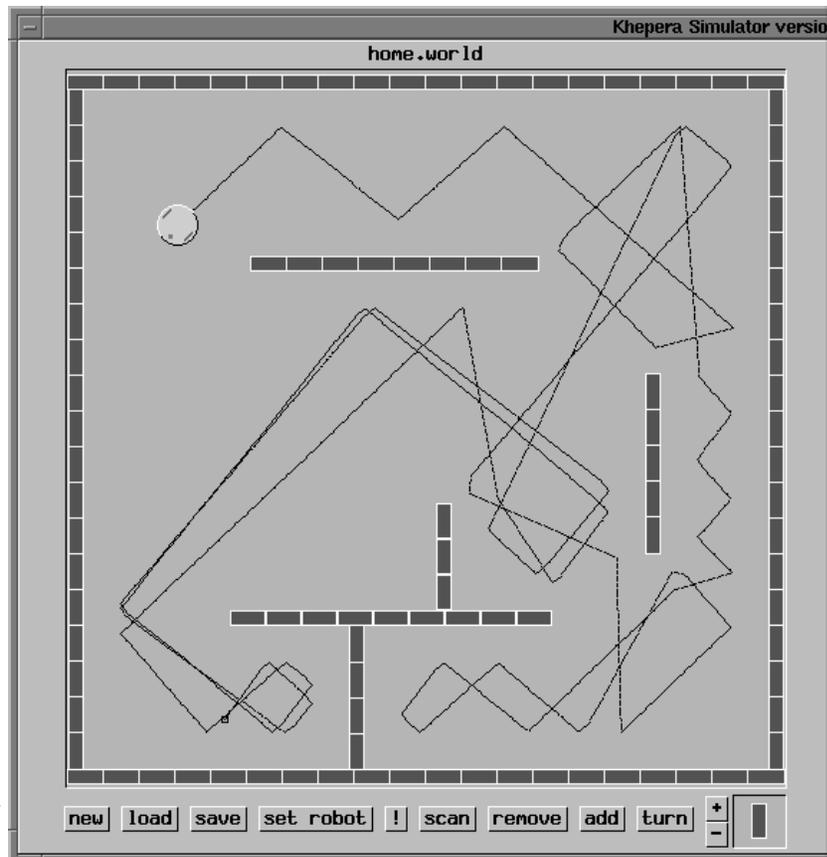


ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos...

- Agentes reativos: Resultado do Apredizado...

- Evitar paredes
- Seguir paredes



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos...

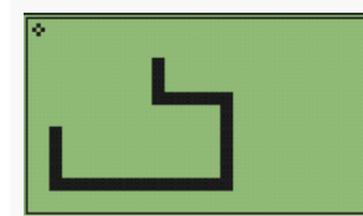
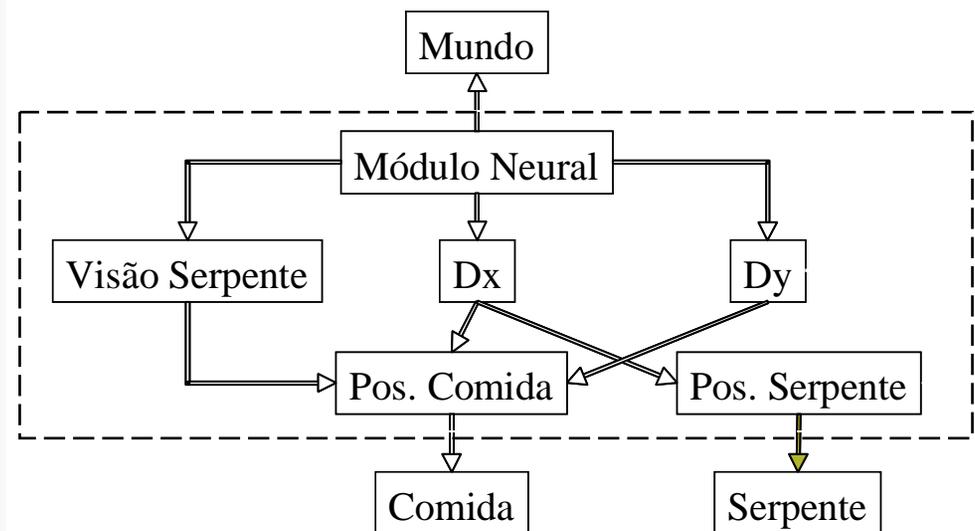
- Agentes reativos: Aprendizado Neural...



João Bittecourt
TC 2002 / Unisinos

Snake + ANNeF

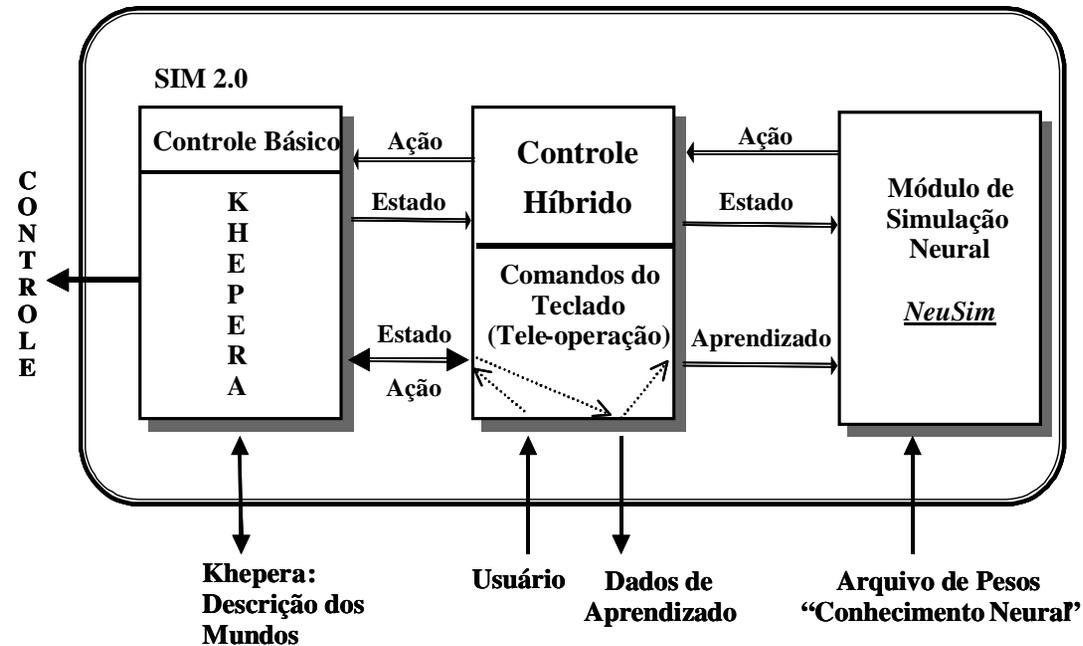
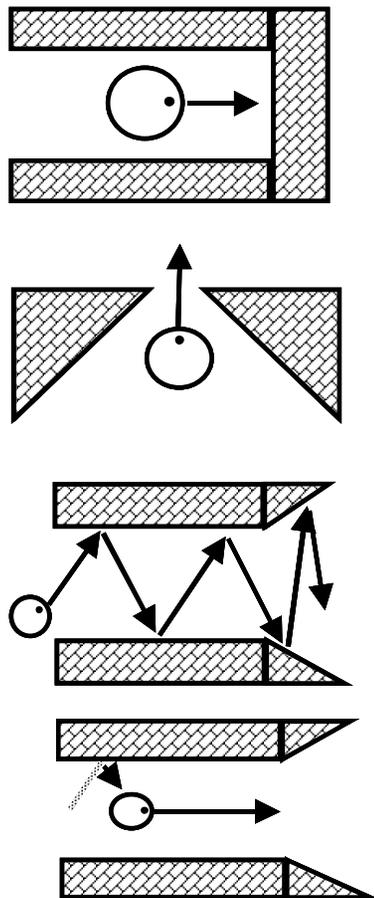
Jogo “Snake Neural”: Controle Neural da Cobra



Snake: Celular
Controle Humano

ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos... Problemas: **Comportamento reativo puro**



Tarefas:

- Evitar as paredes
- Seguir as paredes

Problemas:

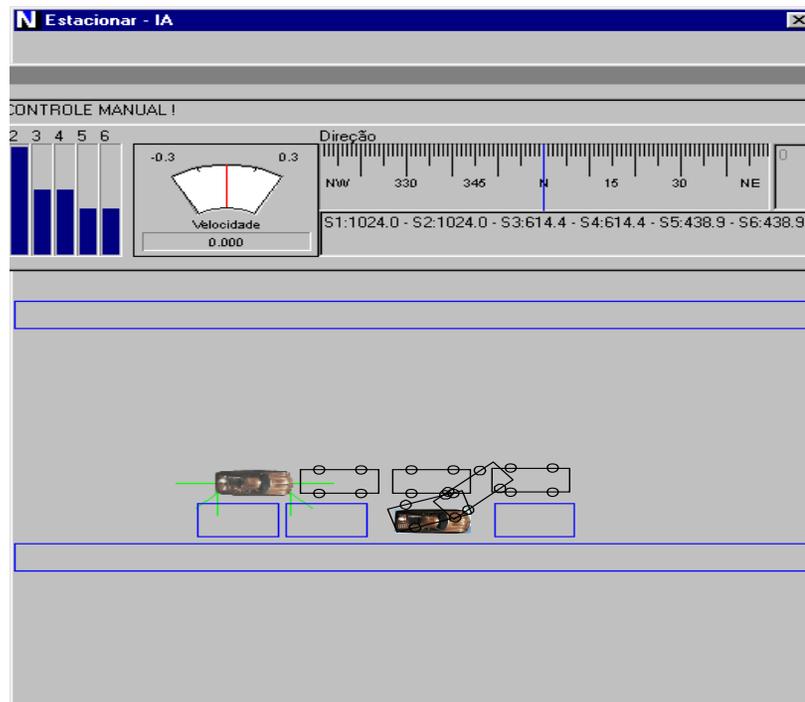
- * Tempo / Sequência
- * Controle "bem comportado"

ANN – Artificial Neural Networks (Redes Neurais Artificiais)

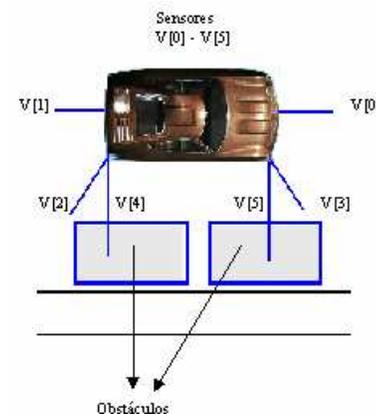
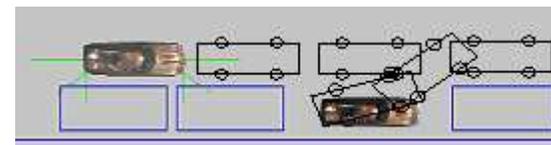
➤ ANN em Jogos... Mais exemplos

Lembrado do SEVA-A?

Pois ele foi re-implementado usando uma Rede Neural → **SEVA-N**



SEVA - Simulador de Estacionamento de Veículos Autônomos (F. Heinen / PIPCA)

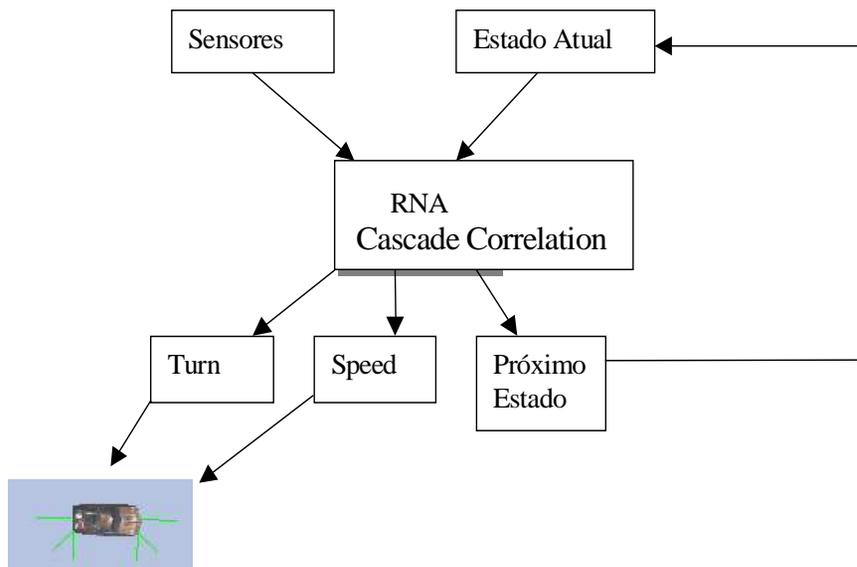
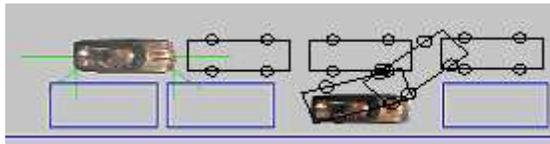


Sistema Híbrido:
Autômato Finito
+
Rede Neural
"Reativa"

ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos... Mais exemplos

SEVA - Simulador de Estacionamento de Veículos Autônomos (F. Heinen / PIPCA)



SEVA-N FSA Neural

Artigo descrevendo o SEVA-N:

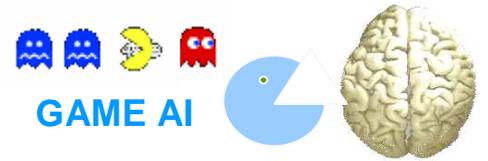
Controle da Tarefa de Estacionamento de um Veículo Autônomo através do Aprendizado de um Autômato Finito usando uma Rede Neural J-CC

F. Osório, F. Heinen, L. Fortes

SBRN 2002 – Porto de Galinhas

Perspectivas: SEVA-3D





ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos... Mais exemplos: **Modelos Comportamentais Neurais**

Modelos Comportamentais e Modelos Físicos...

Opção 1:

Physics for Game Developers. David M. Bourg. O'Reilly, 2002.

Opção 2:

Redes Neurais Artificiais usada para “simplificar e acelerar” a simulação de modelos complexos

- Behaviour Cloning
- Neural Physics Modeling

Referência clássica:

R. Grzeszczuk, D. Terzopoulos, and G. Hinton.

"Neuroanimator: fast neural network emulation and control of physics-based models."
In SIGGRAPH: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 9-20. ACM Press, 1998

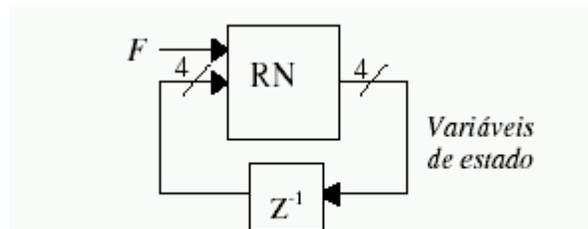
ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos... **Modelos Comportamentais Neurais**

Modelos da Dinâmica do Pêndulo Inverso

Referência:

Extração de um Autômato Finito a partir de Redes Neurais **Recorrentes** na Modelagem do Pêndulo Inverso.
Denise R. P. Simon e Adelmo L. Cechin.
IV ENIA – Campinas. 2003



Estrutura da Rede Neural Recorrente usada

4 variáveis de entrada (w , ϕ , v , x) e a variável de controle (F)

4 neurônios de saída representam as variáveis de estado no próximo passo de tempo.

Simon & Cechin 03

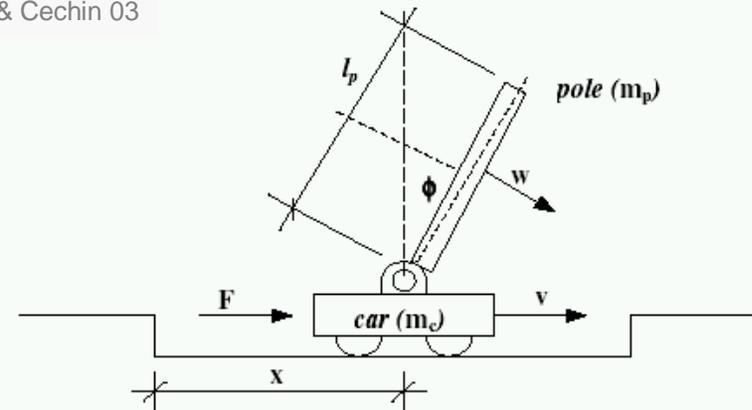


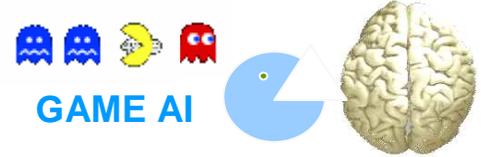
Figura 1. Pêndulo Inverso.

A Figura 1 apresenta a esquematização de um pêndulo inverso com carro. Sendo que seu comportamento dinâmico é descrito matematicamente pelas equações do processo (Barto, Sutton e Anderson 1983), apresentadas a seguir:

$$\dot{w} = \frac{k_1 \sin(\phi) - k_2 w^2 \sin(\phi) \cos(\phi) - F \cos(\phi)}{k_3 - k_2 \cos(\phi) \cos(\phi)}, \quad \dot{\phi} = w$$

$$\dot{v} = k_3 F + k_2 k_3 w^2 \sin(\phi) - k_2 k_1 \cos(\phi) \sin(\phi) k_4 k_3 - k_4 k_2 \cos(\phi) \cos(\phi), \quad \dot{x} = v,$$

onde: $k_1 = (m_c + m_p)g$, $k_2 = m_p l_p$, $k_3 = 4 / 3l_p (m_c + m_p)$, $k_4 = m_c + m_p$, $g = 9.81 m/s^2$ denota a aceleração gravitacional, $m_c = 1.0 [kg]$ e $m_p = 0.1 [kg]$ massa do carro e do pêndulo, respectivamente e $l_p = 0.5 [m]$ metade do comprimento do pêndulo (Barto, Sutton e Anderson 1983). A dinâmica do pêndulo inverso é computada com a constante de tempo $\Delta t = 0,01$.



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos... **Modelos Comportamentais Neurais**

Modelos Comportamentais...

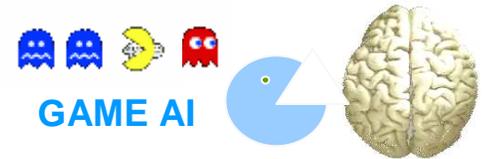
Em algumas situações não dispomos de um modelo matemático do comportamento do sistema dinâmico em questão.

Nestes casos, podemos observar o sistema e a partir dos dados de observação treinar uma Rede Neural para aproximar o comportamento do sistema real.

➤ ANN em Jogos... **Outras Aplicações Possíveis**

Reconhecimento de Padrões,
Controle de Processos e Agentes
e aplicações em jogos tradicionais (de tabuleiro).

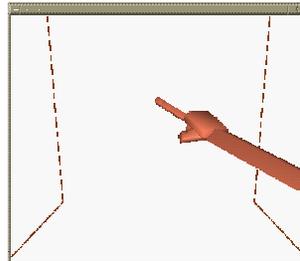
ML4Games



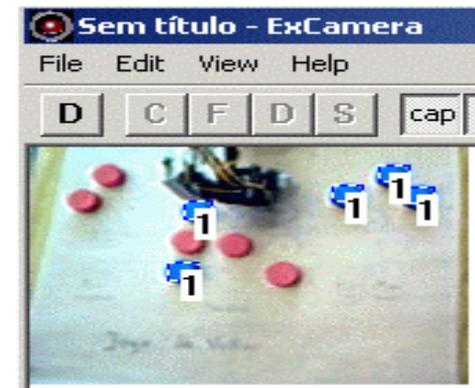
ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos... **Outras Aplicações Possíveis**

Reconhecimento de Padrões,
Controle de Processos e Agentes
e aplicações em jogos tradicionais (de tabuleiro).



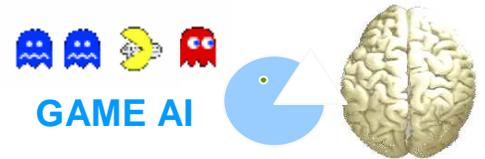
Reconhecimento de Padrões:
• Interface com dispositivos especiais – Data Glove
Reconhecimento de Gestos



Processamento de Imagens:
• Augmented Reality



IA para Jogos – Aula 03



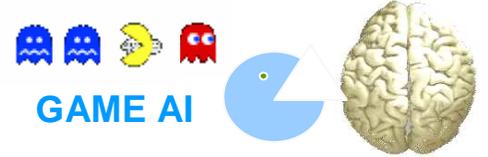
Tópicos abordados...

- Aprendizado de máquinas em Jogos
“Machine Learning for Games”
 - Raciocínio baseado em Casos (RBC / CBR)
 - Árvores de Decisão (AD / IDT)
 - Perfil de Usuários
 - Aprendizado por Reforço (RL)
 - Redes Neurais Artificiais (RNA / ANN)
- Aplicações...

**Bibliografia
em M.L.**

Machine Learning. Tom Mitchell, 1998.
Sistemas Inteligentes. Solange Rezende, 2003.
Redes Neurais. Simon Haykin, 2001.
C4.5: Programs for machine learning. Ross Quinlan, 1993
Raciocínio Baseado em Casos. Von Wangenheim, 2003.

ML4Games



Aprendizado:

➤ Aprendizado de Máquinas para Jogos...

Outras técnicas possíveis de serem usadas:

- Algoritmos Genéticos (custo computacional alto)
- Modelos probabilistas / Redes Bayesianas
- Lógica Nebulosa
- Sistemas Híbridos
- Entre outros...