

GT-JeDi - Curso de Desenv. de Jogos

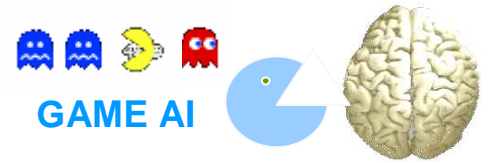
IA para Jogos



Gustavo Pessin

2006/2 - A07

Redes Neurais Artificiais



➤ Cronograma

➤ Conceitos

➤ Neurônio artificial

➤ Redes: *Perceptron* e *Multi Layer Perceptron* (MLP),

➤ Exemplos

➤ Separando cores

➤ Prevendo dinâmica molecular

➤ Calculando trajetos em robótica simulada

➤ Como fazer para...

➤ Criar um conjunto de dados de treino e teste

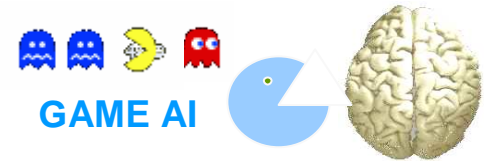
➤ Criar topologia da rede neural

➤ Treinar a rede

➤ Aplicar a rede no seu trabalho

Redes Neurais Artificiais

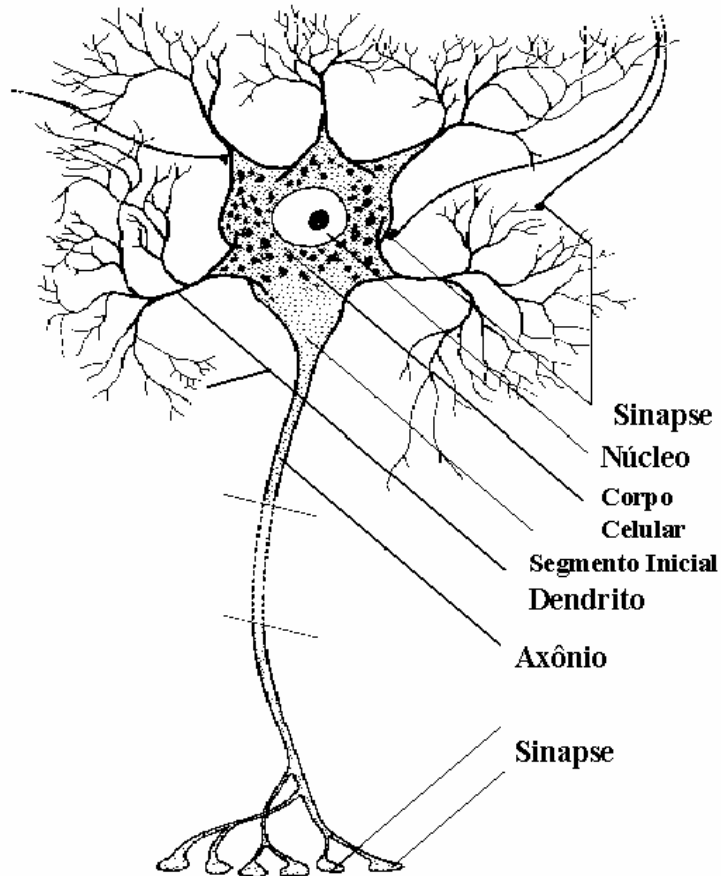
e Princípios



Modelo matemático de uma analogia ao funcionamento do cérebro.

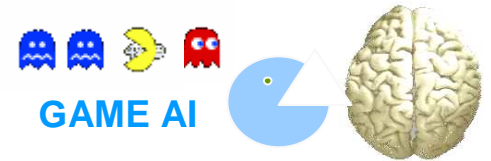
A idéia básica:

unidades de processamento simples computando certas funções matemáticas, dispostas em camadas e interligadas por um grande número de conexões apresentam grande poder computacional.



Redes Neurais Artificiais

e Princípios



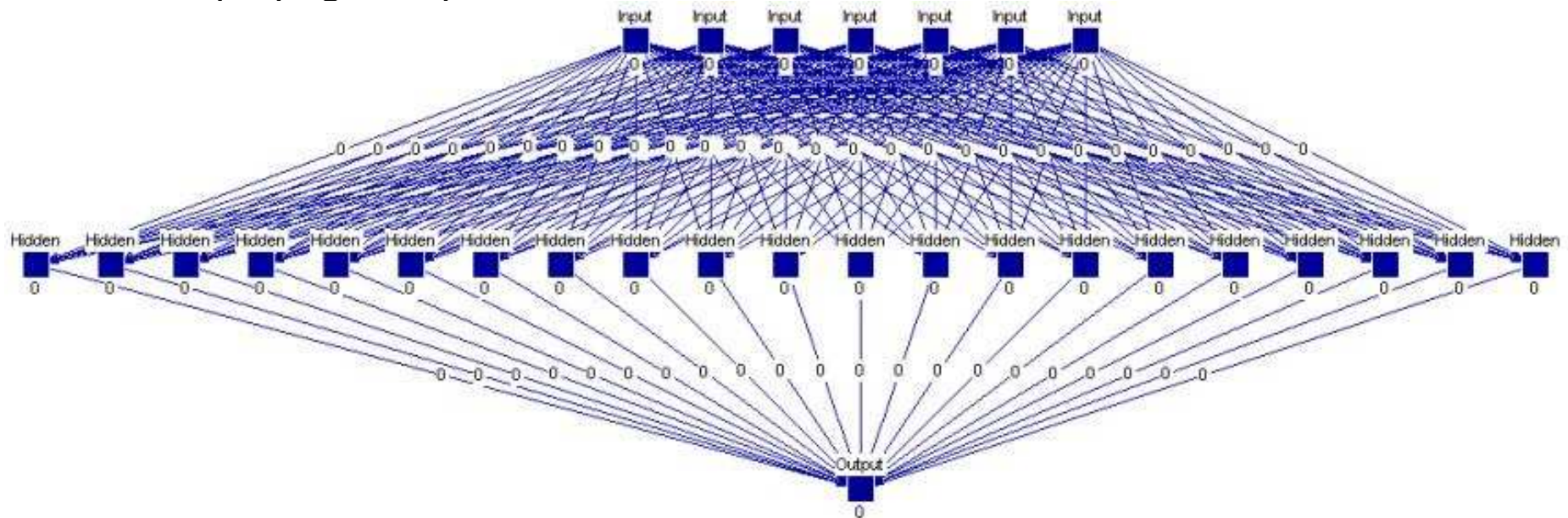
Uma rede neural é composta de vários nodos conectados por links direcionados:

Um nodo recebe uma série de entradas e é **ativado** de acordo com a composição de suas entradas

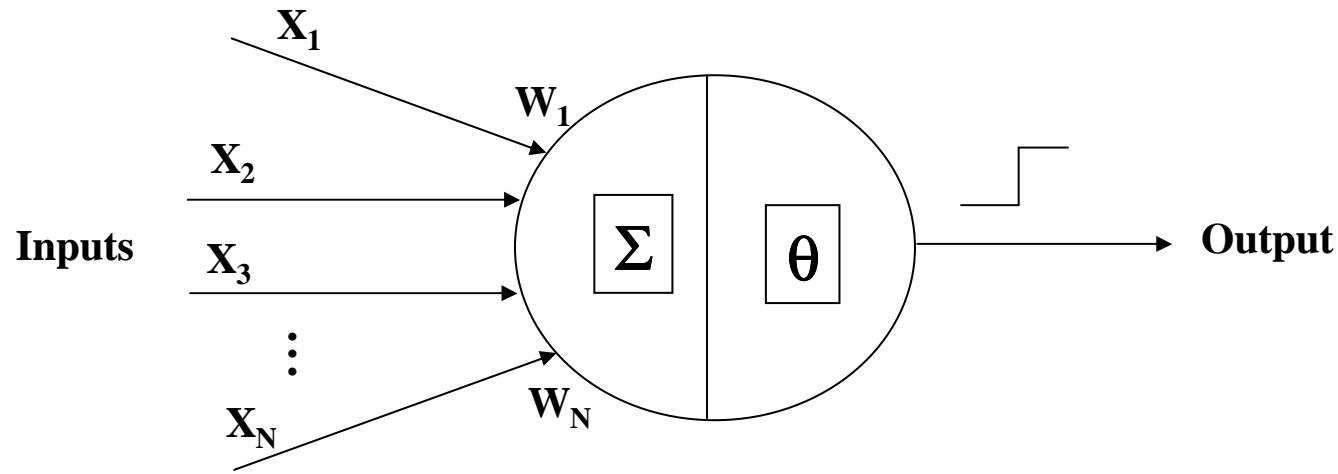
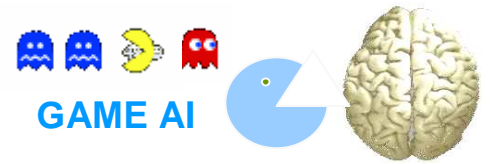
Cada entrada possui um peso, que determina a força daquela entrada na ativação

Cada neurônio tem uma função de ativação que determina a sua saída de acordo com a entrada

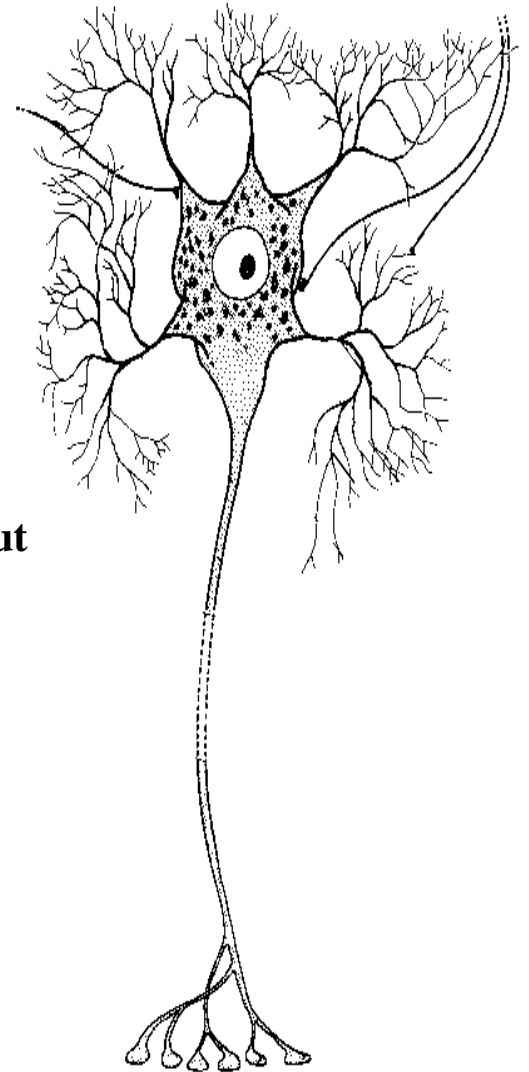
Saídas são propagadas para outros neurônios



Redes Neurais Artificiais



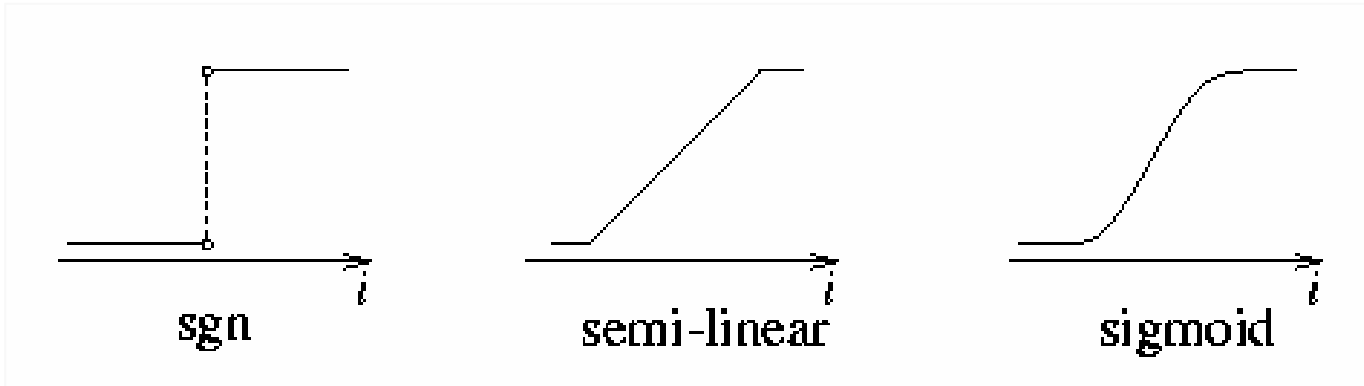
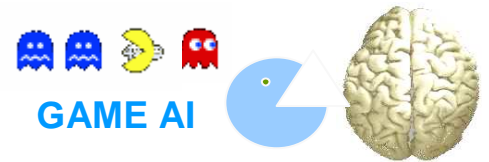
Neurônio de McCulloch e Pitts



REDES NEURAIIS:

- Perceptron - Funções de Transferência: “normaliza saída”

Objetivo - adaptar o resultado da soma ponderada em um intervalo de saída adotado



$$F(X) = \text{Sgn}(X)$$

If $X \geq 0$
Then $Y = 1$
Else $Y = 0$ (ou -1)

ou

If $X \geq \text{Limiar}$
Then $Y = 1$
Else $Y = 0$ (ou -1)

$$F(X) = \text{Linear}(X, \text{Min}, \text{Max})$$

If $X < \text{Min}$
Then $Y = 0$

If $X \geq \text{Min}$ and $X \leq \text{Max}$
Then $Y = X$

If $X > \text{Max}$
Then $Y = 1$

Obs.: $Y = X$ ou $Y = \text{Normaliza}(X)$

$$F(X) = \text{Sigmoid}(X)$$

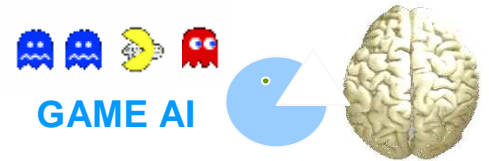
Assimétrica

$$Y = \frac{1}{1 + \text{Exp}(-x)}$$

Simétrica

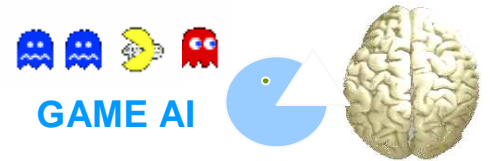
$$Y = \text{TanHip}(X)$$

Redes Neurais Artificiais



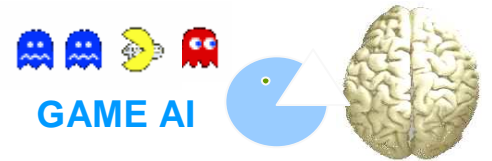
- Uma Rede Neural Artificial é um sistema distribuído paralelo, naturalmente propenso para armazenar conhecimento de exemplos e generaliza-lo.
- Em particular, a semelhança que Redes Neurais Artificiais apresentam em relação ao cérebro humano pode ser de uma forma mais fundamental associada com as seguintes colocações:
 - 1. O conhecimento é adquirido em uma rede neural através de um processo de aprendizagem.
 - 2. O conhecimento é armazenado em conexões (sinapses) existentes entre os elementos da rede, representadas através de pesos.

Redes Neurais Artificiais



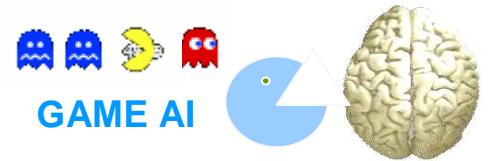
- Aplicações
 - Reconhecimento de formas
 - Tratamento de sinal
 - Visão, fala
 - Previsão e modelagem
 - Auxílio à decisão
 - Robótica

Redes Neurais Artificiais



➤ Aprendizado

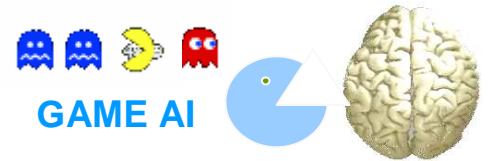
- Redes Neurais Artificiais possuem a capacidade de aprender por exemplos e fazer interpolações e extrapolações do que aprenderam.
- No aprendizado conexionista não se procura obter regras como na abordagem simbólica da Inteligência Artificial, mas determinar a intensidade de conexões entre neurônios.
- Algoritmo de aprendizado: conjunto de procedimentos bem definidos para adaptar os parâmetros de uma Rede Neural Artificial para que a mesma possa aprender.
- Não há um único algoritmo de aprendizado (diferentes maneiras de ajustar pesos)



➤ Modelo de aprendizado **Perceptron**

➤ Modelo proposto por Rosenblatt em 1958, conhecido como perceptron, era composto por uma estrutura de rede tendo como unidades básicas nodos MCP e de uma regra de aprendizado.

➤ Em 1962 , Rosenblatt demonstrou o teorema de convergência do perceptron, o qual mostra que um nodo MCP treinado com o algoritmo de aprendizado do perceptron sempre converge, caso o problema em questão seja linearmente separável.

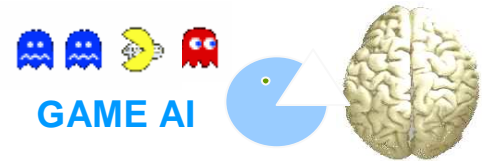


➤ Modelo de aprendizado **Perceptron**

➤ A topologia original descrita por Rosenblatt era composta por unidades de entrada, por um nível intermediário formado pelas unidades de associação e por um nível de saída formado pelas unidades de resposta.

➤ Embora esta topologia original possua três níveis, ela é conhecida como perceptron de uma única camada, já que somente o nível de saída (unidades de resposta) possui propriedades adaptativas.

Redes Neurais Artificiais

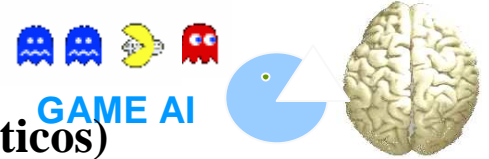


➤ Modelo de aprendizado **Perceptron**

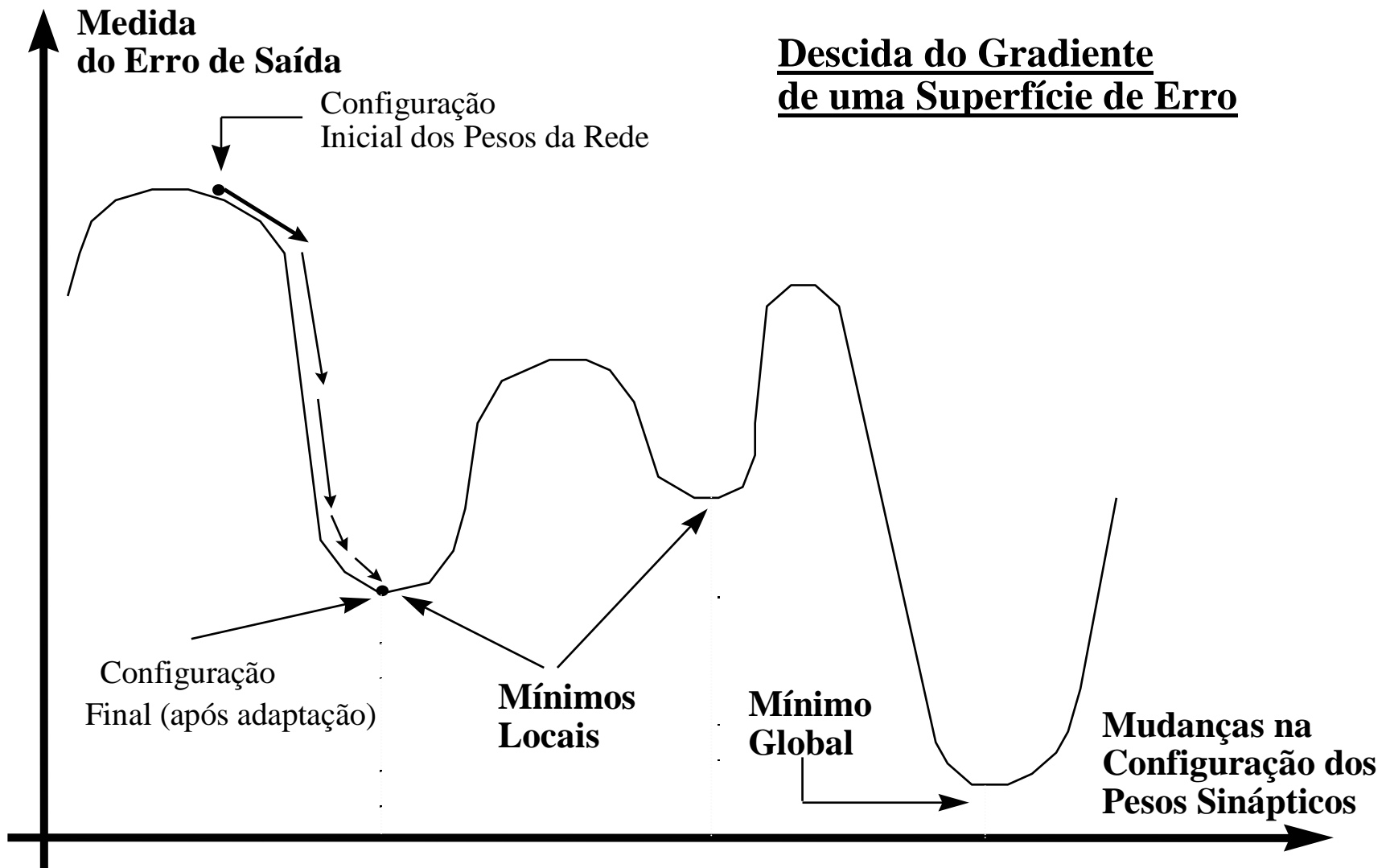
➤ Apesar de ter causado grande euforia na comunidade científica da época, o perceptron não teve vida muito longa.

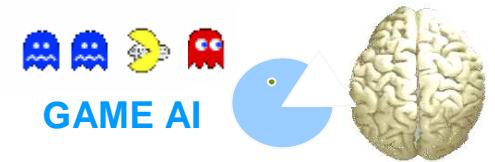
➤ Duras críticas à sua capacidade computacional causaram grande impacto sobre as pesquisas em Redes Neurais Artificiais, o que levou a um grande desinteresse pela área durante os anos 70 e início dos anos 80.

➤ Esta visão pessimista sobre a capacidade do perceptron e das Redes Neurais Artificiais de uma maneira geral mudou com a descrição, em 1986, do algoritmo *backpropagation*.



- **Perceptron - Aprendizado = Adaptação das conexões (pesos sinápticos)**





REDES NEURAIAS:

- **Perceptron - Aprendizado = Adaptação das conexões (pesos sinápticos)**
- **Método da Descida do Gradiente**

Erro quadrático: $E = \frac{1}{2} \sum_i (D_i - A_i)^2$ **Ajuste dos Pesos:** $\Delta W_{ij} = -\alpha \frac{\partial E}{\partial W_{ij}}$

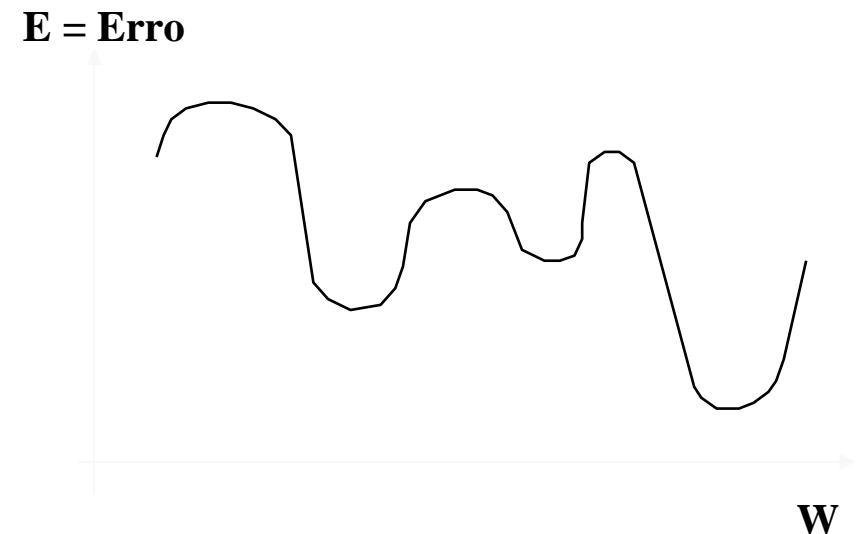
Derivação da regra de reajuste dos pesos (saída linear):

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial A_i} \frac{\partial A_i}{\partial W_{ij}} = \delta_i X_j$$

$$\frac{\partial E}{\partial A_i} = - (D_i - A_i) = \delta_i$$

$$\frac{\partial A_i}{\partial W_{ij}} = X_j$$

$$\Delta W_{ij} = -\alpha \cdot \delta_i \cdot X_j = \alpha \cdot (D_i - A_i) \cdot X_j$$



**Dada uma entrada X_i e o peso W_i ,
o neurônio fornece uma saída A_i .
Queremos treiná-lo para responder D_i**

PERCEPTRON

Rede Neural com apenas 2 entradas:

X, Y - Entradas (Valores numéricos)

W_1, W_2 - Pesos Sinápticos

$$\text{Saída} = \sum W_i \cdot X_i + \text{Biais}$$

$$= W_1 \cdot X + W_2 \cdot Y + C$$

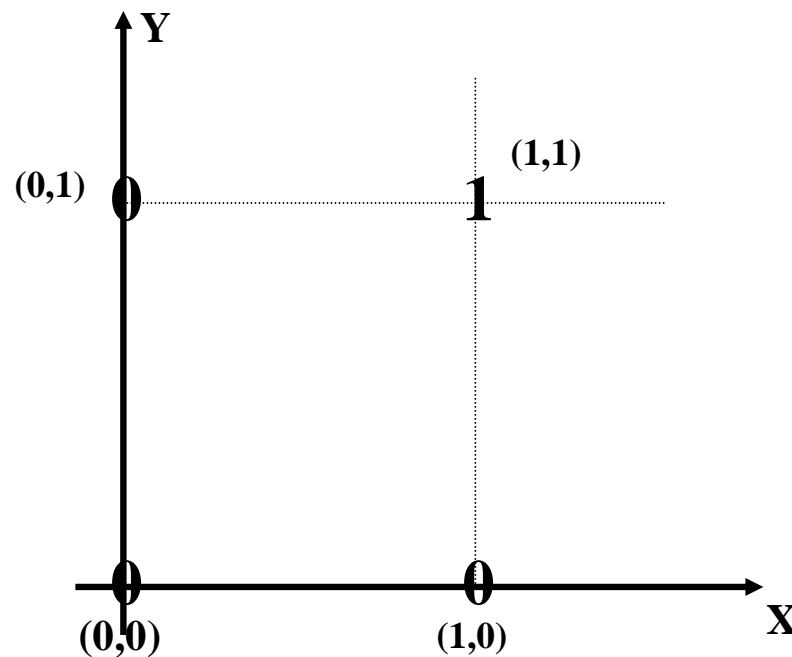


GAME AI



CLASSIFICAÇÃO

X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1



Representação
Geométrica do
Problema

PERCEPTRON

Rede Neural com apenas 2 entradas:

X, Y - Entradas (Valores numéricos)

W_1, W_2 - Pesos Sinápticos

$$\text{Saída} = \sum W_i \cdot X_i + \text{Biais}$$

$$= W_1 \cdot X + W_2 \cdot Y + C$$

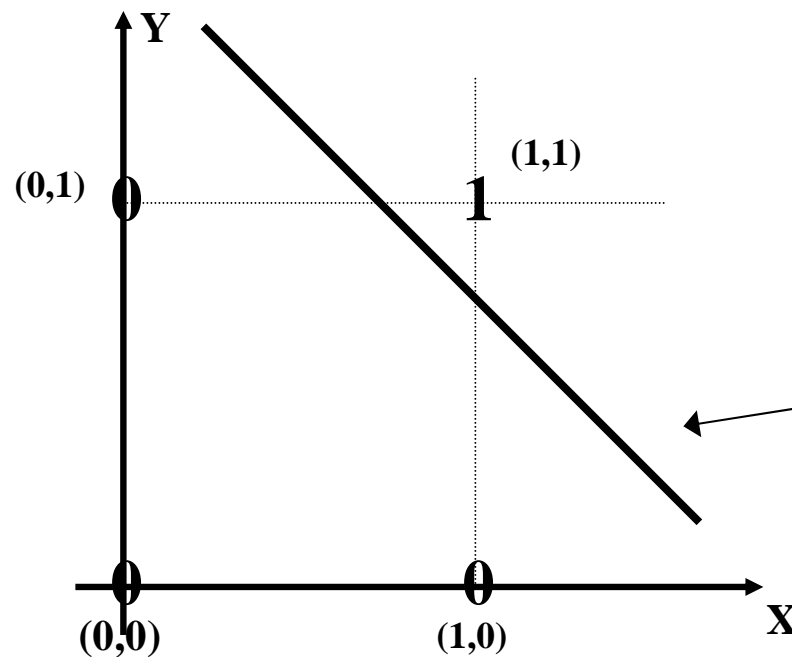


GAME AI



CLASSIFICAÇÃO

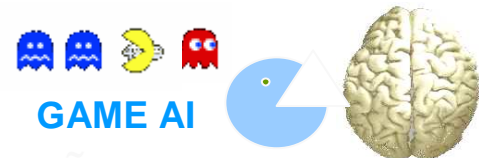
X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1



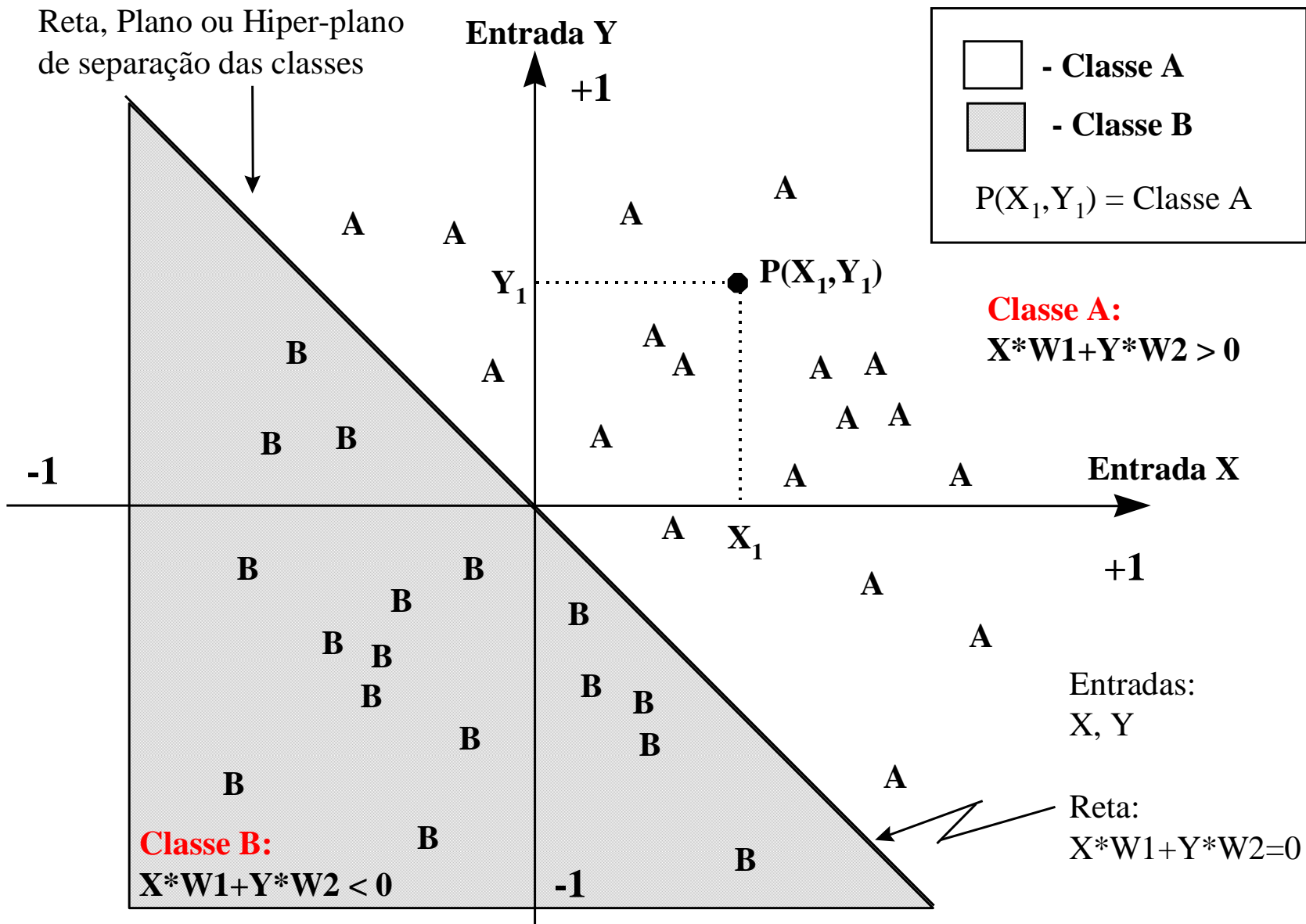
Representação
Geométrica do
Problema

Como classificar?
Separar as classes

REDES NEURAIS:



• **Perceptron** - Aprendizado = Adaptação dos (hiper)planos de separação





GAME AI



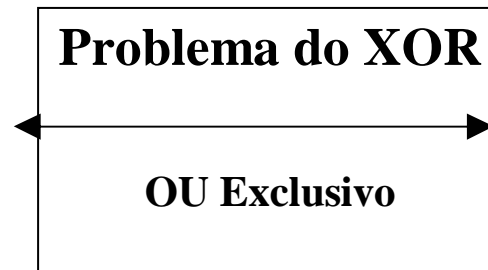
REDES NEURAIAS:

Histórico e Princípios

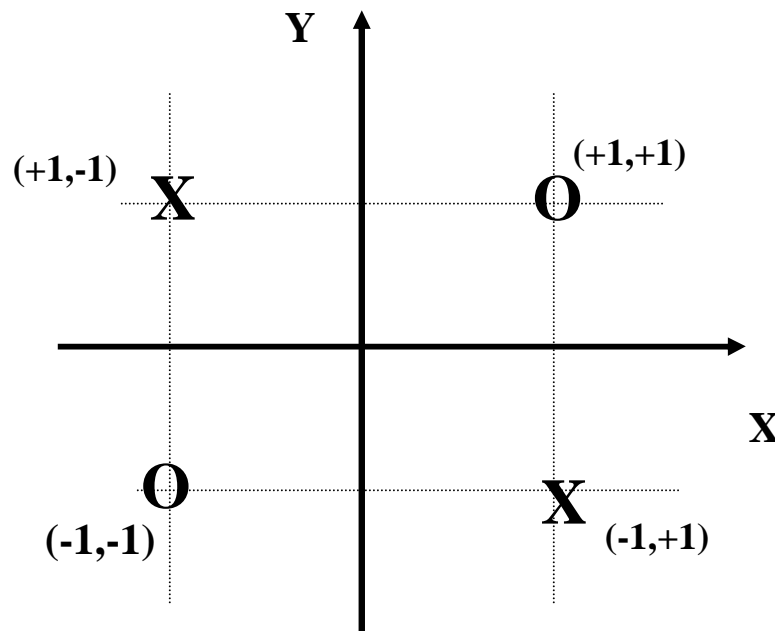
- Perceptron - **Problema do Aprendizado do XOR**

Minsky & Papert 1969 (“Perceptrons”) - Problema não linearmente separável!

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



X	Y	XOR
-1	-1	0
-1	+1	X
+1	-1	X
+1	+1	0



Sistema de Equações:

$$A.X + B.Y = S$$

$$-1.X + -1.Y = -1$$

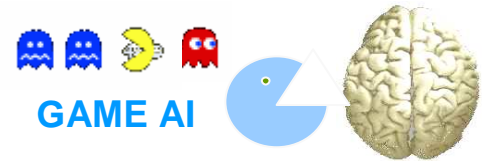
$$-1.X + +1.Y = +1$$

$$+1.X + -1.Y = +1$$

$$+1.X + +1.Y = -1$$

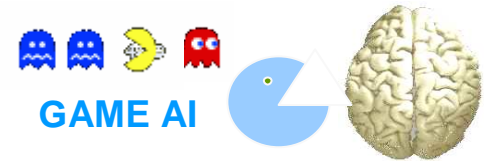
Sem solução!!!

Redes Neurais Artificiais

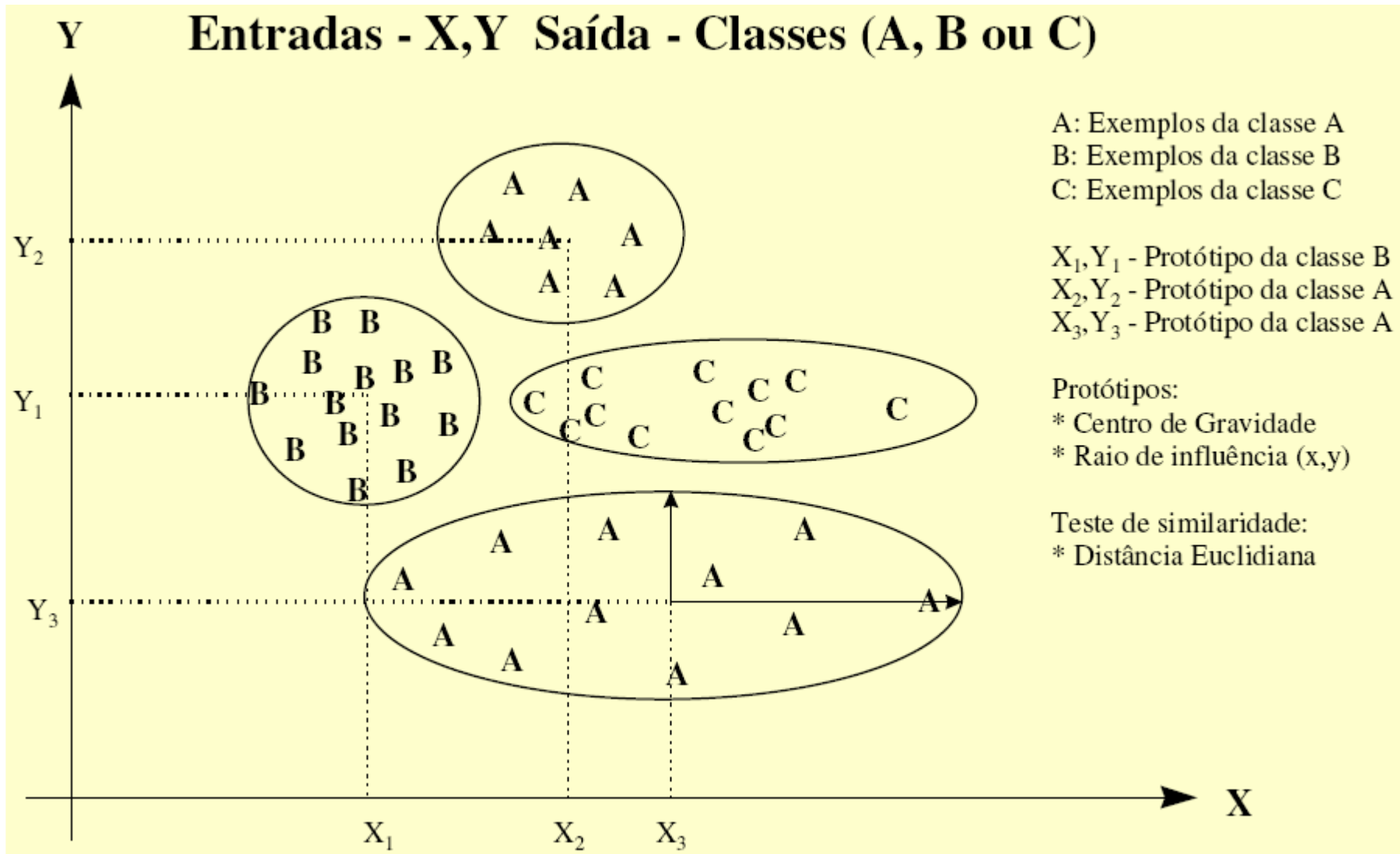


- MLP (*Multi Layer Perceptron*)
 - Redes de uma só camada resolve apenas problemas linearmente separáveis.
 - A solução para problemas não linearmente separáveis passa pelo uso de redes com uma ou mais camadas intermediárias, ou escondidas.

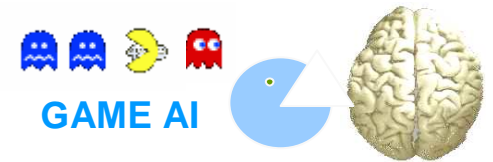
Redes Neurais Artificiais



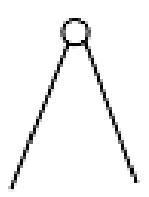
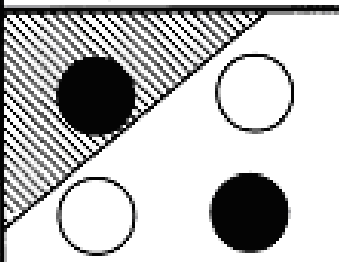
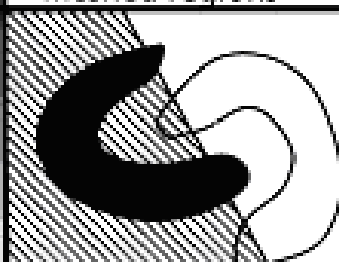
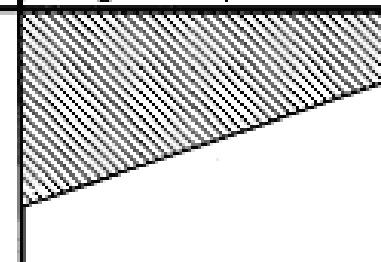
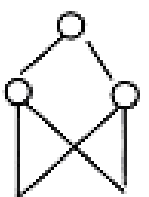
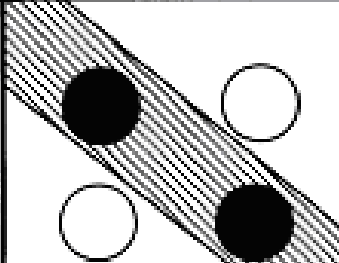
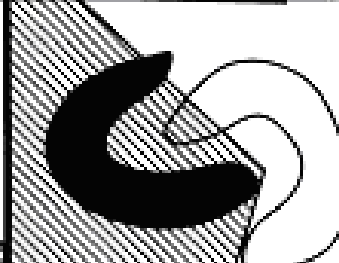
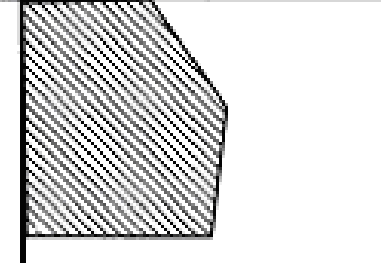
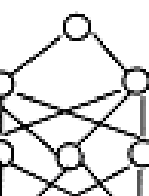
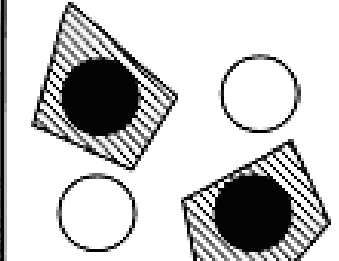
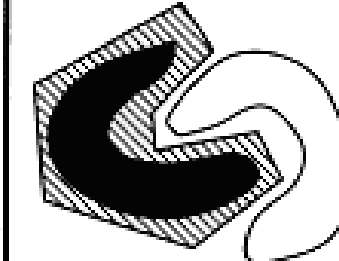
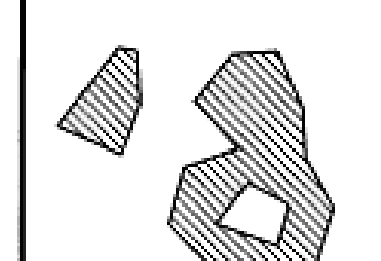
➤ MLP (Multi Layer Perceptrom) com back-propagation



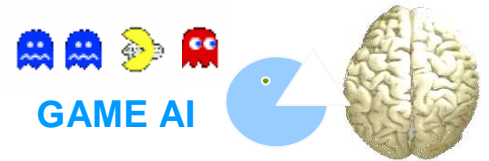
Redes Neurais Artificiais



➤ MLP (Multi Layer Perceptrom) com back-propagation

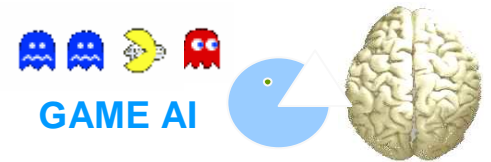
Structure	Description of decision regions	Exclusive-OR problem	Classes with meshed regions	General region shapes
 Single layer	Half plane bounded by hyperplane			
 Two layer	Arbitrary (complexity limited by number of hidden units)			
 Three layer	Arbitrary (complexity limited by number of hidden units)			

Redes Neurais Artificiais



- Back-propagation
 - Funcionamento Geral
 - Inicia-se os pesos com um valor aleatório
 - Usando o *training set*, alimenta a rede com os valores de entrada e observa a saída
 - Calcula o erro
 - Ajusta os pesos para diminuir o erro (*backpropagating* na rede), e repete o processo
 - Cada Iteração é chamada de *Epoch*
 - Pode ser executado um número fixo de vezes ou até que o erro seja pequeno o suficiente

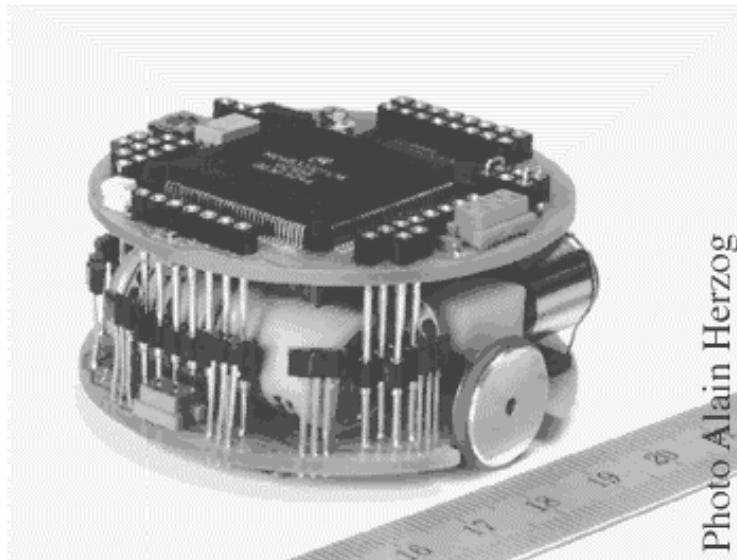
Redes Neurais Artificiais



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

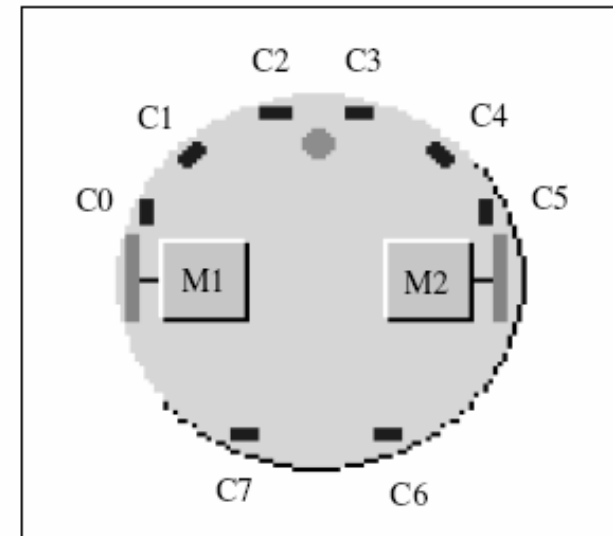
➤ ANN em Jogos...

- Agentes reativos: Robo Khepera



Controle Sensorial-Motor

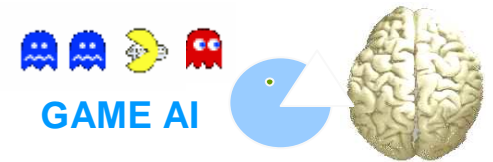
Aprendizado



IF $S1 < \text{Limite}$ and $S2 < \text{Limite}$ and
 $S3 > \text{Limite}$ and $S4 > \text{Limite}$
THEN Action(Turn_Left)

IF $S2 > \text{Limite}$ and
 $S3 > \text{Limite}$ and
 $S2 > S3$ and
 $S1 > S4$
THEN Action(Turn_Right)

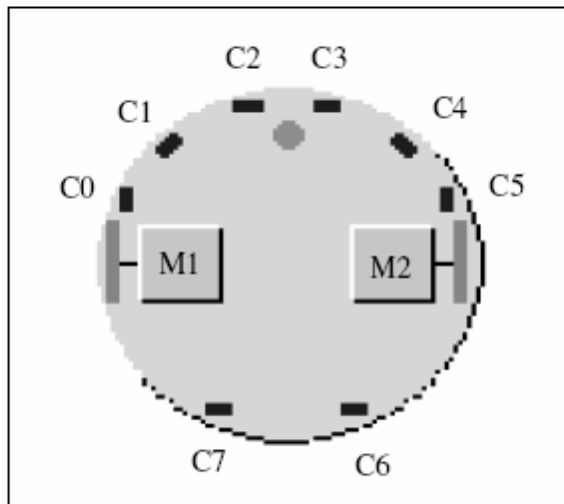
Redes Neurais Artificiais



ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos...

- Agentes reativos: Robo Khepera



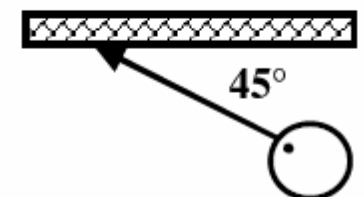
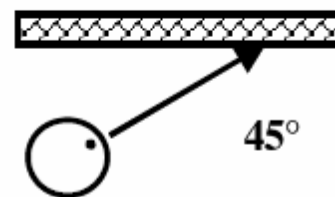
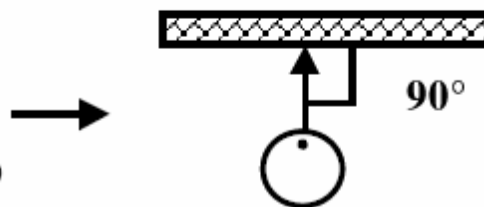
Controle Sensorial-Motor

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



Redes Neurais Artificiais

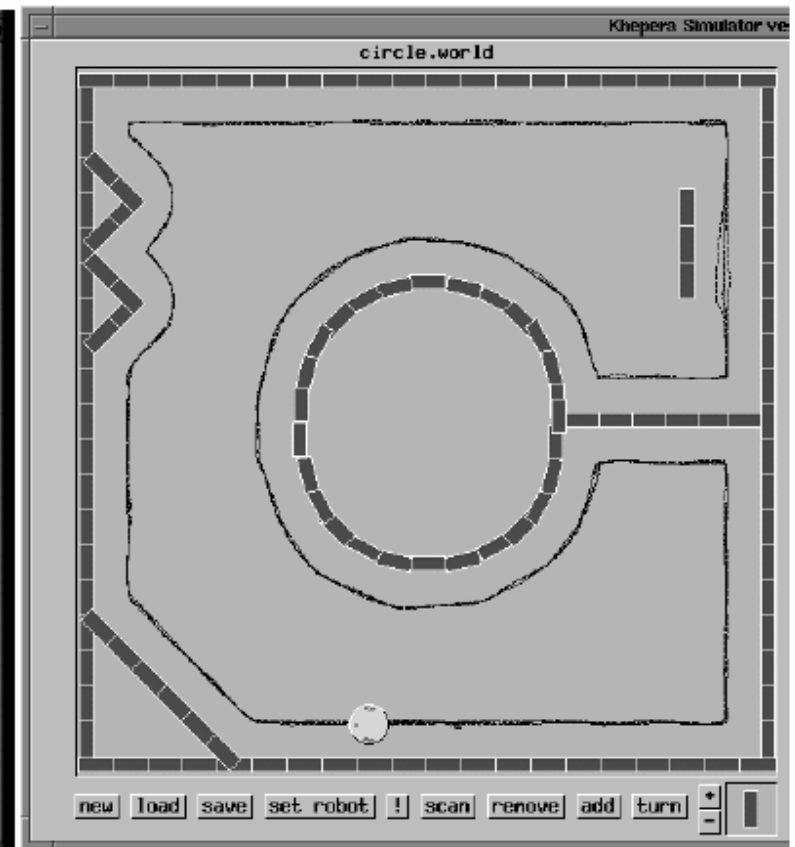
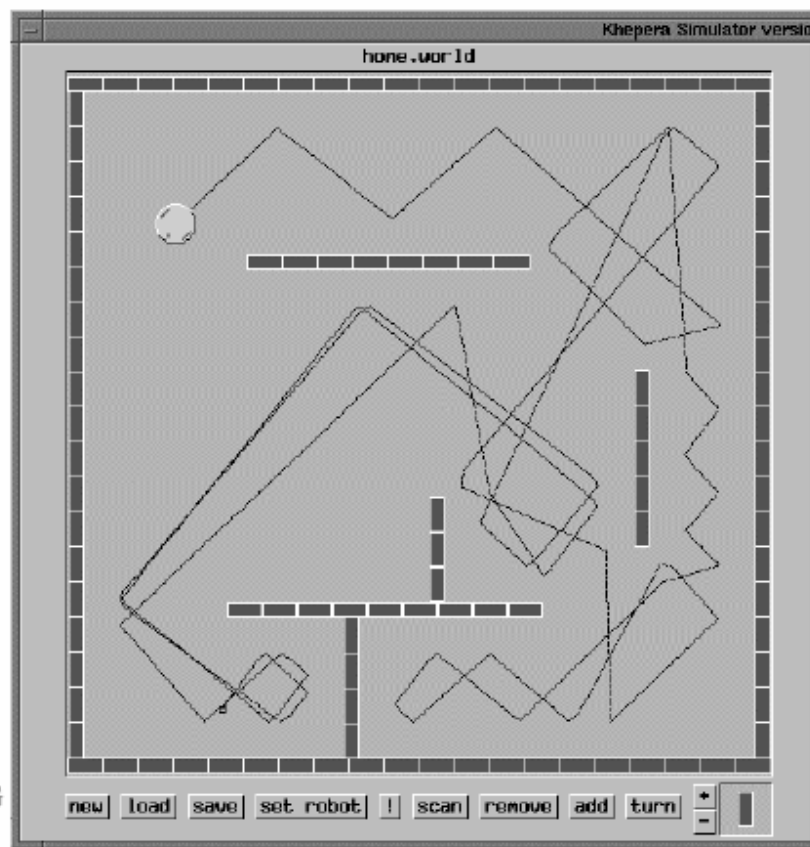


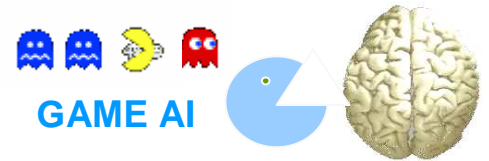
ANN – Artificial Neural Networks (Redes Neurais Artificiais)

➤ ANN em Jogos...

- Agentes reativos: Resultado do Aprendizado...

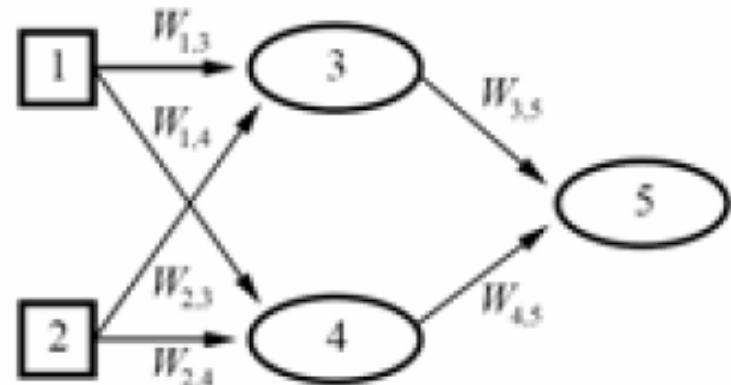
- Evitar paredes
- Seguir paredes



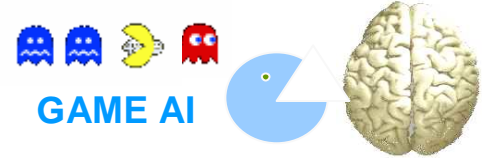


➤ Feed-forward network

- Normalmente organizadas em camadas
 - Neurônios de uma camada recebem a ativação dos neurônios da camada anterior
- Exemplo Clássico: 3 camadas
 - Camada de Entrada (Input Layer)
 - Camada Escondida (Hidden Layer)
 - Camada de Saída (Output Layer)

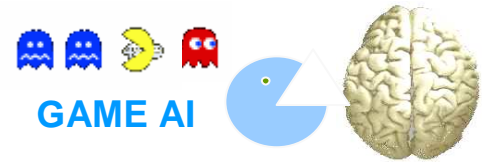


Redes Neurais Artificiais



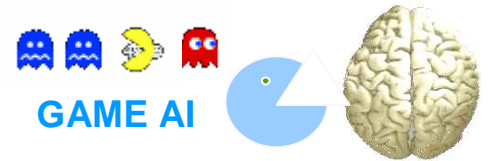
- Camada de entrada
 - Cada neurônio representa uma variável de entrada
 - Quanto mais neurônios, mais difícil o treinamento
 - Rede trabalha com números
 - Variáveis Booleanas: 0 e 1
 - Variáveis Discretas:
 - Variáveis Contínuas: normalização

Redes Neurais Artificiais



- Camada oculta
 - Quantos neurônios?
 - Tentativa e erro!
 - Sugestão:
 - 3 testes:
 - Dobro da entrada;
 - Igual a entrada;
 - Metade da entrada;

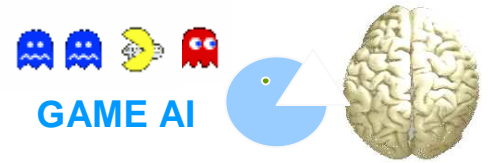
Redes Neurais Artificiais



➤ Occam's Razor

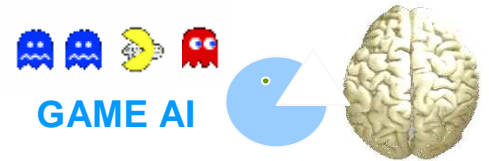
- O problema do tamanho da rede pode ser visto de maneira simplificada usando o Occam's Razor: Qualquer máquina de aprendizagem deve ser grande o suficiente para resolver o problema, porém não maior.
- Dificuldade: o que é grande o suficiente?

Redes Neurais Artificiais



- Camada de saída
 - Quantos Neurônios?
 - Depende do problema... um pode bastar...
 - Função de ativação
 - Valor discreto (classificação) [act_logistic]
 - Valor contínuo (aproximação) [act_identity]

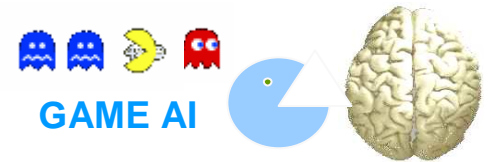
Redes Neurais Artificiais



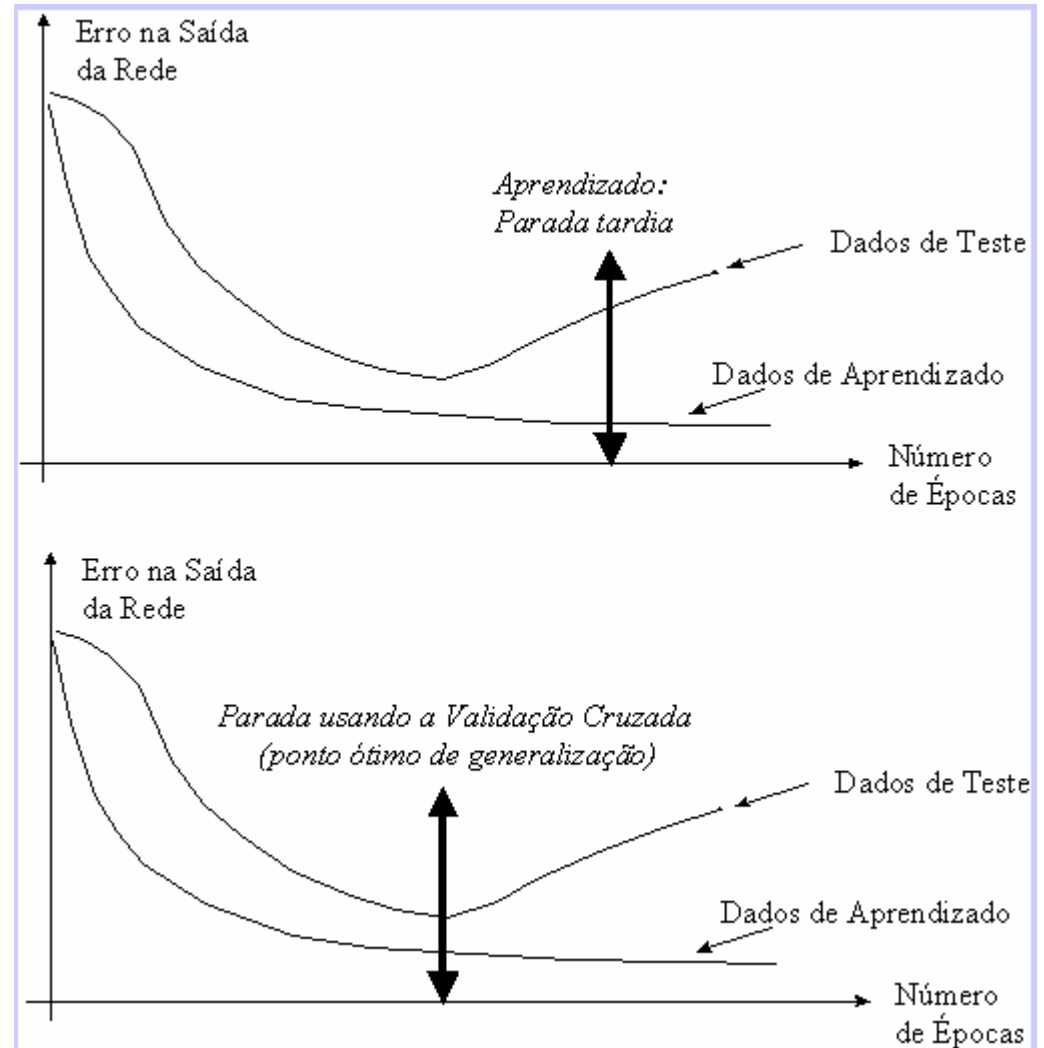
➤ Treinamento

- Determina os valores dos pesos de forma que uma entrada gere a saída desejada.
- O treinamento é feito a partir de uma série de pares de entrada e saída.
 - Deseja-se minimizar o erro.
 - Algoritmo mais comum: *Back Propagation*
- Deve-se tomar cuidado para que o training set seja representativo do conjunto de entradas e saídas possíveis
- Em caso de mudança significativa nos conjuntos de entrada e saída, deve-se treinar a rede novamente.

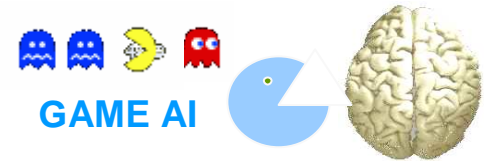
Redes Neurais Artificiais



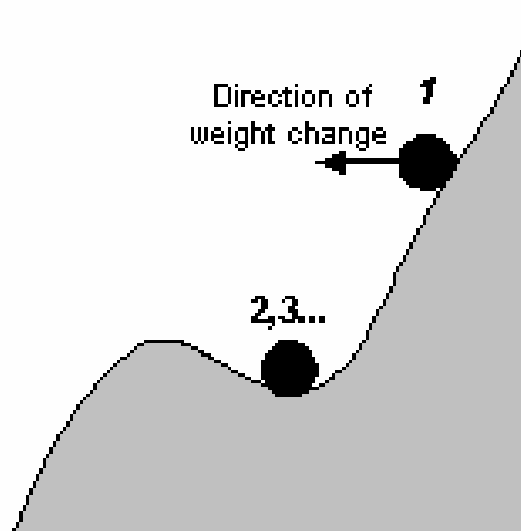
➤ Curvas de aprendizado



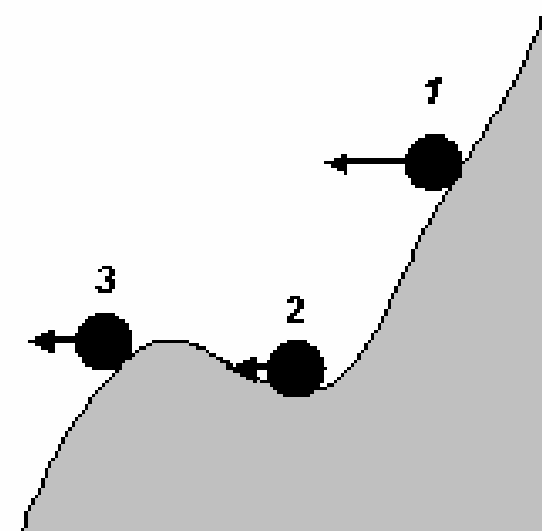
Redes Neurais Artificiais



- Momentum (parâmetro do back-propagation)
 - Aprendizagem com momento usa o incremento anterior para aumentar a velocidade e estabilizar a convergência

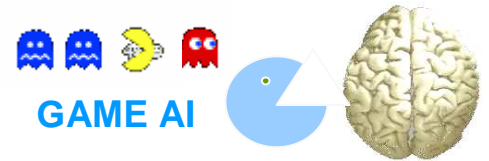


Gradient Descent



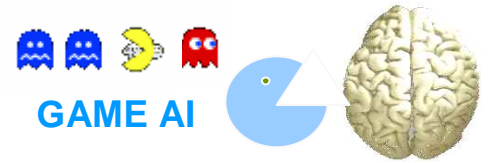
Gradient Descent with Momentum

Redes Neurais Artificiais



- Critério de parada
 - Número de iterações
 - ERRO MSE : Não há garantia de que o sistema possa atingir um MSE especificado
- Problema: overtraining
- Parada com validação cruzada
 - dois conjuntos: treinamento e validação
 - conjunto de validação: 10-25% do total dos casos
 - com certa frequência (5 a 10 iterações) verifica-se o desempenho no conjunto de validação

Redes Neurais Artificiais



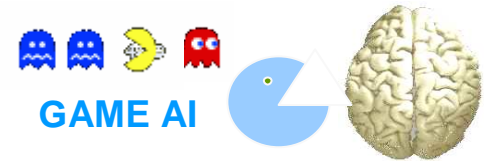
➤ **Vantagens e Limitações das Redes Neurais Artificiais:**

- Aplicações de Machine Learning e Sistemas Adaptativos;
- Aplicadas em tarefas onde temos bases de exemplos disponíveis sobre um determinado problema, realizando a aquisição automática de conhecimentos;
- Associação de padrões de entradas e saída;
- Classificação de padrões de forma supervisionada ou não;
- Aproximação de funções desconhecidas através de amostras destas funções;
- Trabalhar com dados aproximados, incompletos e inexatos;
- Paralelismo, generalização, robustez;
- “Tarefas complexas realizadas por seres humanos”;

➤ **Limitações:**

- Trabalhar com conhecimentos simbólicos de alto nível;
- Composição e construção de conhecimentos estruturados;
- Dificuldade de explicitação dos conhecimentos adquiridos;
- Dificuldade para definir a estrutura da rede, seus parâmetros e a base de dados;
- Falta de garantia de uma convergência do algoritmo para uma solução ótima;

Redes Neurais Artificiais



PAPERS / DOCUMENTAÇÃO:

- **FAQ:** [Http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/top.html](http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/top.html)
[Http://www.faqs.org/faqs/ai-faq/neural-nets/](http://www.faqs.org/faqs/ai-faq/neural-nets/)
[Ftp://ftp.sas.com/pub/neural/FAQ.html](ftp://ftp.sas.com/pub/neural/FAQ.html)
- **Osorio - Neural:** [Http://www.inf.unisinos.br/~osorio/neural.html](http://www.inf.unisinos.br/~osorio/neural.html)
- **Livro On-Line:** [Http://www.inf.unisinos.br/~osorio/neural/Neuro-book.html](http://www.inf.unisinos.br/~osorio/neural/Neuro-book.html)
- **UCI-ML:** <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- **Neuroprose**

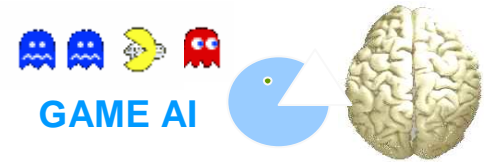
* ASSOCIAÇÃO:

- **Connectionist List**
- **Comp.ai.neural-nets**

• SOFTWARES:

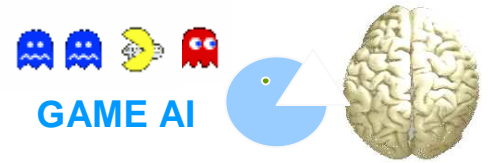
- **PDP++** <http://www.cnbc.cmu.edu/PDP++/PDP++.html>
- **SNNS** <http://www-ra.informatik.uni-tuebingen.de/SNNS/>
- **NevProp** Nevada Propagation Software
- **INSS** Contactar Osório...
- **Outros** <http://www.inf.unisinos.br/~osorio/neural/software.html>

Redes Neurais Artificiais



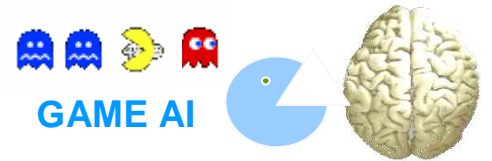
➤ Exemplo: dinâmica molecular

Redes Neurais Artificiais



➤ Exemplo: cores

Redes Neurais Artificiais



➤ SNNS

- Calculando trajetos em robótica simulada
 - Como fazer para...
 - Criar um conjunto de dados de treino e teste
 - Criar topologia da rede neural
 - Treinar a rede
 - Aplicar a rede no seu trabalho