Redes Neurais Artificial

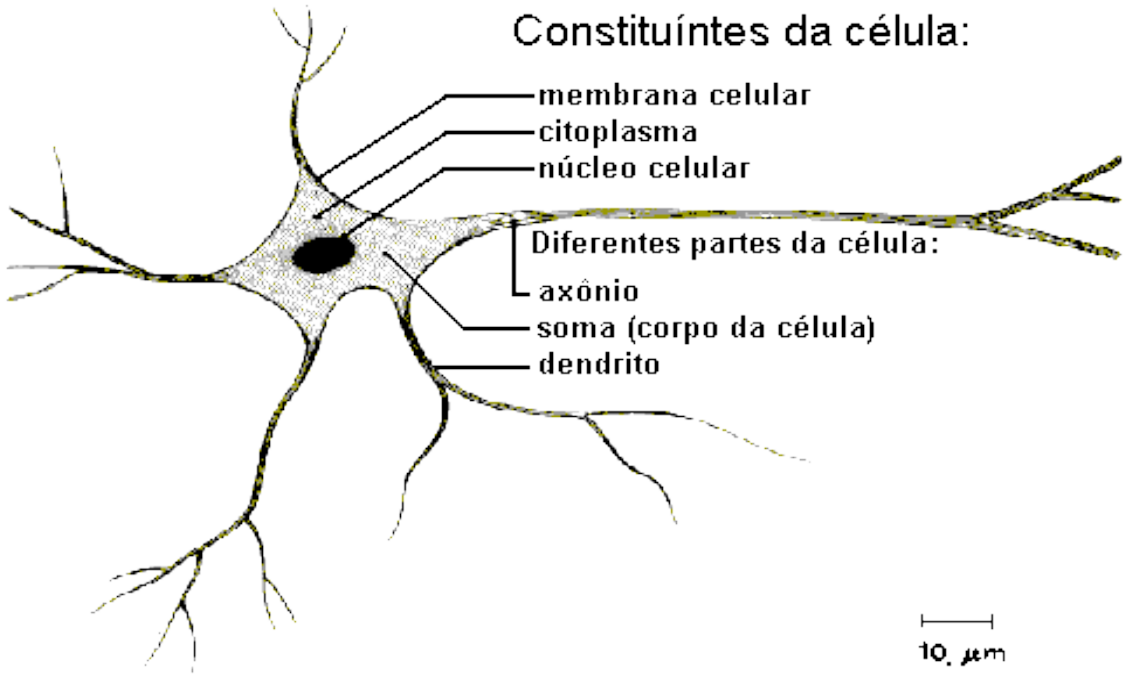
Tópicos:

- Introdução ao estudo de RNA –sua origem e inspiração biológica
- Características gerais das RN e descrição do neurônio artificial
- Aprendizado de RN e tipos de Aprendizado
- Algoritmo de Aprendizado e Topologias básicas
- Algumas Aplicações das RNA

Introdução

- Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento; já o cérebro de um mamífero pode ter muitos bilhões de neurônios.
- O sistema nervoso é formado por um conjunto extremamente complexo de células, os neurônios. Eles têm um papel essencial na determinação do funcionamento e comportamento do corpo humano e do raciocínio. Os neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo central, e pelos axônios que são longos terminais de saída.

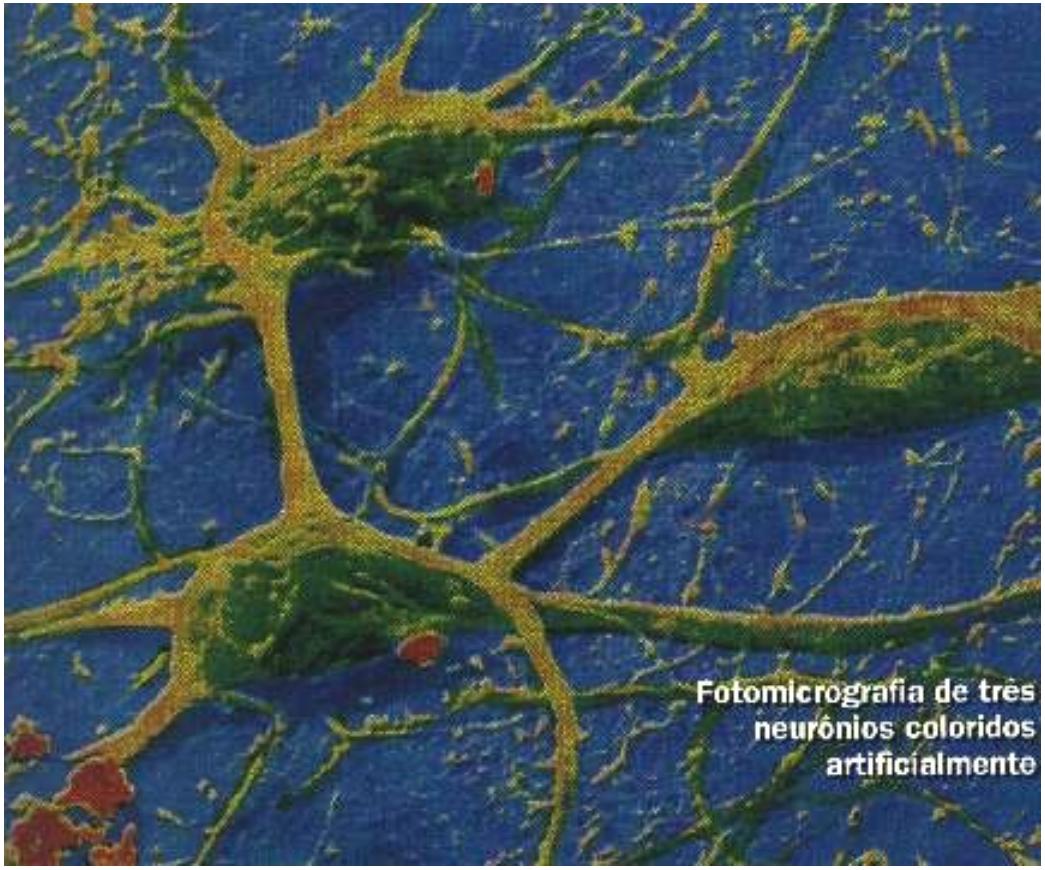
Constituíntes da célula neuronal



Inspiração biológica

- Os neurônios se comunicam através de sinapses. Sinapse é a região onde dois neurônios entram em contato e através da qual os impulsos nervosos são transmitidos entre eles. Os impulsos recebidos por um neurônio A, em um determinado momento, são processados, e atingindo um dado limiar de ação, o neurônio A dispara, produzindo uma substância neurotransmissora que flui do corpo celular para o axônio, que pode estar conectado a um dendrito de um outro neurônio B. O neurotransmissor pode diminuir ou aumentar a polaridade da membrana pós-sináptica, inibindo ou excitando a geração dos pulsos no neurônio B. Este processo depende de vários fatores, como a geometria da sinapse e o tipo de neurotransmissor.

- Em média, cada neurônio forma entre mil e dez mil sinapses. O cérebro humano possui cerca de 10 E11 neurônios, e o número de sinapses é de mais de 10 E14, possibilitando a formação de redes muito complexa.



Fotomicrografia de três neurônios coloridos artificialmente

Um Breve Histórico

- McCulloch e Pitts (1943), Hebb (1949), e Rosemblatt (1958). Estas publicações introduziram o primeiro modelo de redes neurais simulando "máquinas", o modelo básico de rede de auto-organização, e o modelo Perceptron de aprendizado supervisionado, respectivamente.
- nos anos 60 e 70, importantes trabalhos sobre modelos de redes neurais em visão, memória, controle e auto-organização como: Amari, Anderson, Cooper, Cowan, Fukushima, Grossberg, Kohonen, von der Malsburg, Werbos e Widrow.

- Alguns históricos sobre a área costumam “pular” os anos 60 e 70 e apontar um reínicio da área com a publicação dos trabalhos de Hopfield (1982) relatando a utilização de redes simétricas para otimização e de Rumelhart, Hinton e Williams que introduziram o poderoso método Backpropagation.

Características Gerais das RNs

- Uma rede neural artificial é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

Características

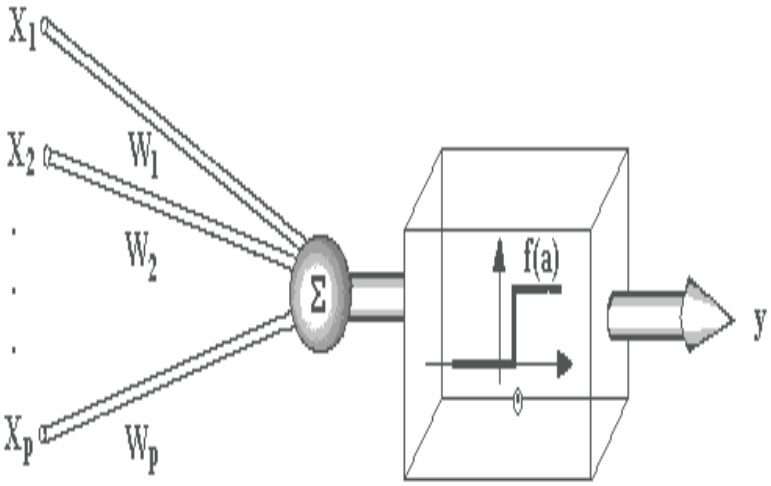
- São modelos adaptativos treináveis
- Podem representar domínios complexos (não lineares)
- São capazes de generalização diante de informação incompleta
- Robustos
- São capazes de fazer armazenamento associativo de informações
- Processam informações Espaço/temporais
- Possuem grande paralelismo, o que lhe conferem rapidez de processamento

O que é uma Rede Neural?

- A grande premissa do conexionismo para aplicações em processamento de informações e/ou inteligência artificial é o fato de que se pode analisar um problema de acordo como funcionamento do cérebro humano
- O cérebro processa informações através da ativação de uma série de neurônios biológicos. Os neurônios por sua vez, interagem numa rede biológica através da intercomunicação.

O Neurônio Artificial

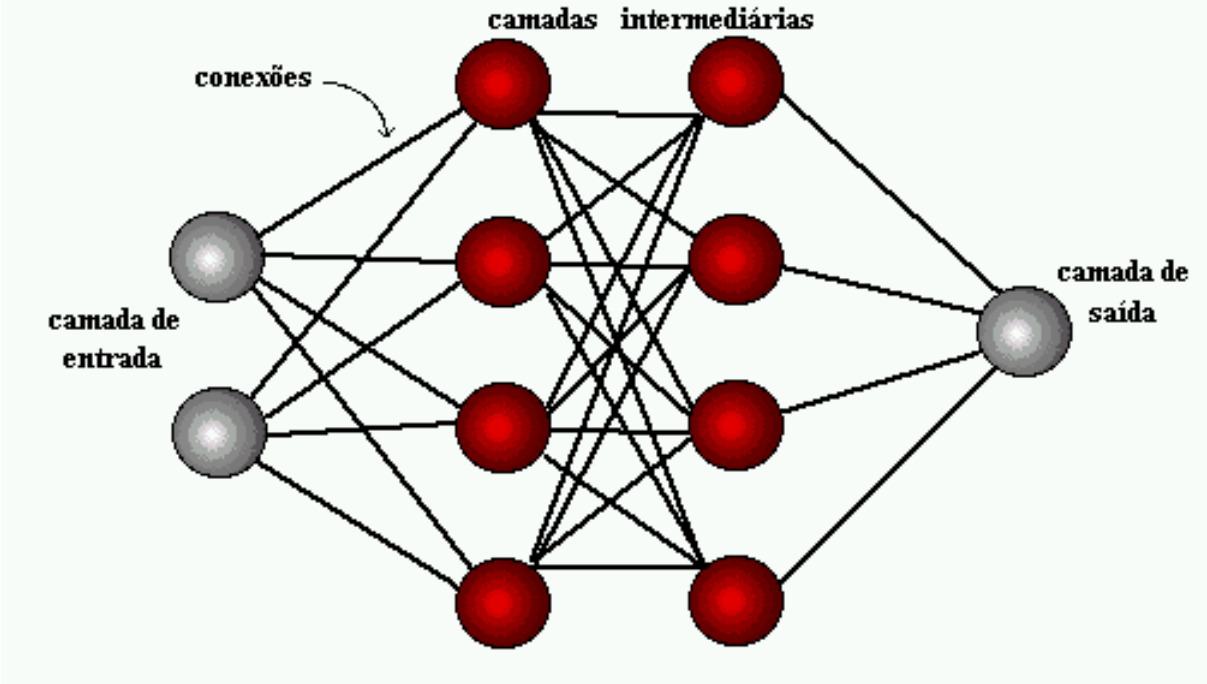
- McCullock e Pitts 1943,
- sinais são apresentados à entrada;
- cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade;
- é feita a soma ponderada dos sinais que produz um nível de atividade;
- se este nível de atividade exceder um certo limite (threshold) a unidade produz uma determinada resposta de saída.



Exemplo

- sinais de entrada X1, X2, ..., Xp (0 ou 1)
- pesos w1, w2, ..., wp, valores reais.
- limitador t;
- Neste modelo, o nível de atividade a é dado por:
  - $a = w1X1 + w2X2 + ... + wpXp$
- A saída y é dada por:
  - $y = 1$, se $a >= t$ ou
  - $y = 0$, se $a < t$.

Organização em camadas



Organização em camadas

- Usualmente as camadas são classificadas em três grupos:
  - **Camada de Entrada**: onde os padrões são apresentados à rede;
  - **Camadas Intermediárias ou Escondidas**: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
  - **Camada de Saída**: onde o resultado final é concluído e apresentado.

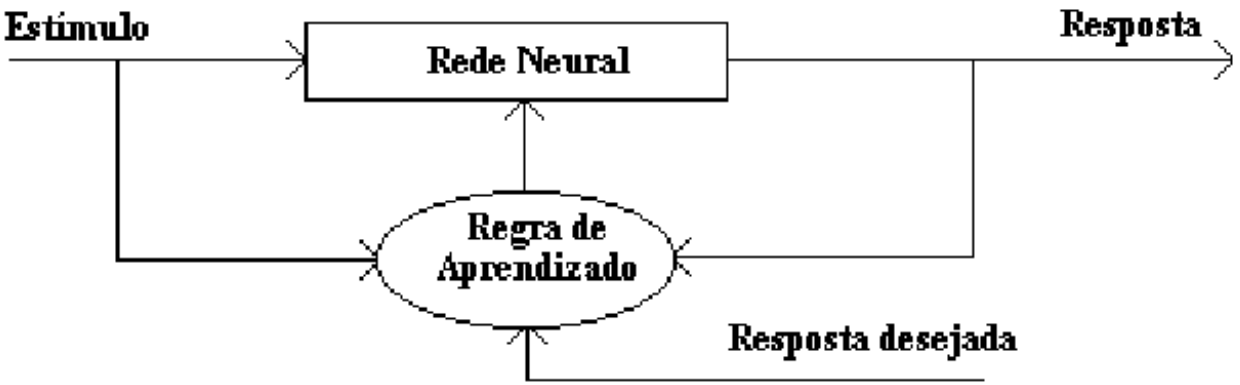
Processos de Aprendizado

- A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho.
- Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento.
- O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

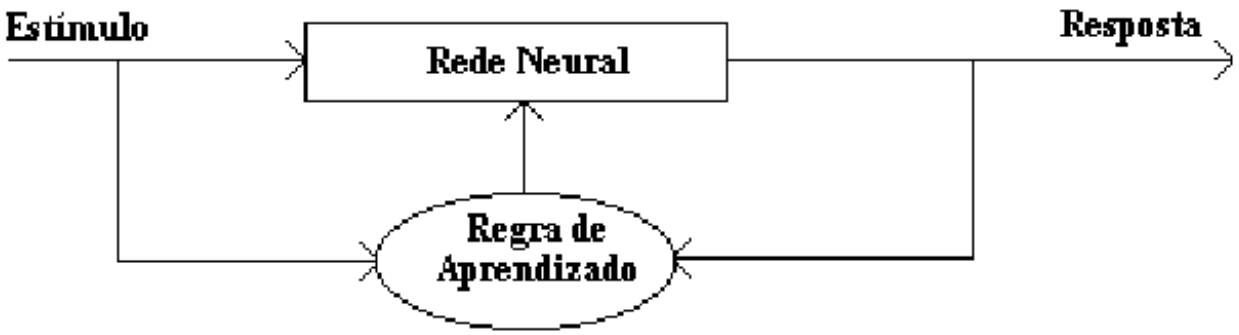
Algoritmo de Aprendizado

- algoritmo de aprendizado é um conjunto de regras bem definidas para a solução de um problema de aprendizado.
- Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais,
- estes algoritmos diferem entre si principalmente pelo modo como os pesos são modificados.

- **Aprendizado Supervisionado**, quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- **Aprendizado Não Supervisionado** (auto-organização), quando não existe uma agente externo indicando a resposta desejada para os padrões de entrada;
- **Reforço**, quando um crítico externo avalia a resposta fornecida pela rede.



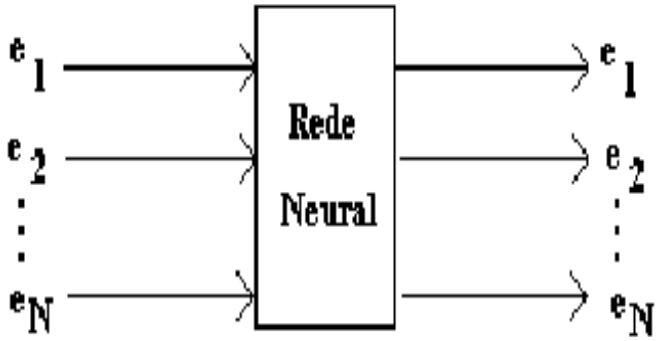
Aprendizado Supervisionado



Aprendizado Não-supervisionado

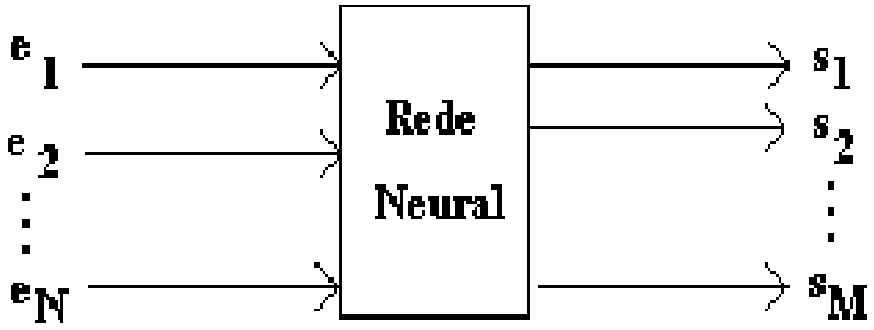
Classificação quanto a finalidade do aprendizado

Auto-associador - uma coleção de exemplos é apresentado ao sistema, a qual é suposto memorizar os exemplos. Depois, quando um destes exemplos for novamente apresentado de modo deteriorado supõe-se que o sistema restitua o original sem deterioração. Neste caso, aprende-se a funcionar como um filtro;



Rede Neural Auto-associativa

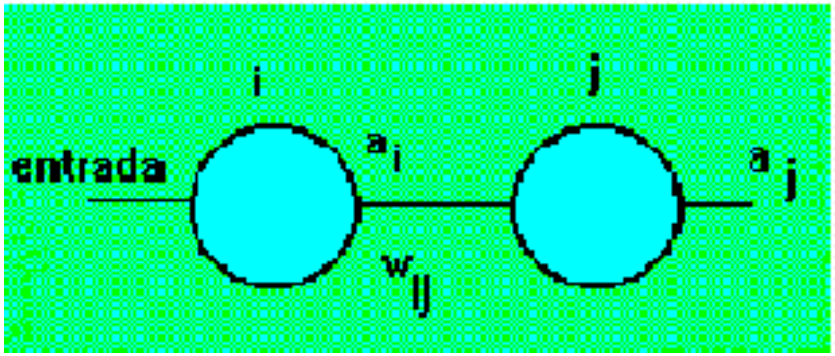
Hetero-associador - é uma variante do auto-associador que memoriza um conjunto de pares. O sistema aprende a reproduzir o segundo elemento do par mesmo que o primeiro seja apresentado contendo pequenas alterações. Este hetero-associador é também conhecido como reconhecedor de padrões, onde o primeiro elemento apresentado é o elemento a ser reconhecido e o segundo é um elemento do conjunto de padrões considerado (Barreto, 1995).



Rede Neural Hetero-associativa

Regra de Hebb

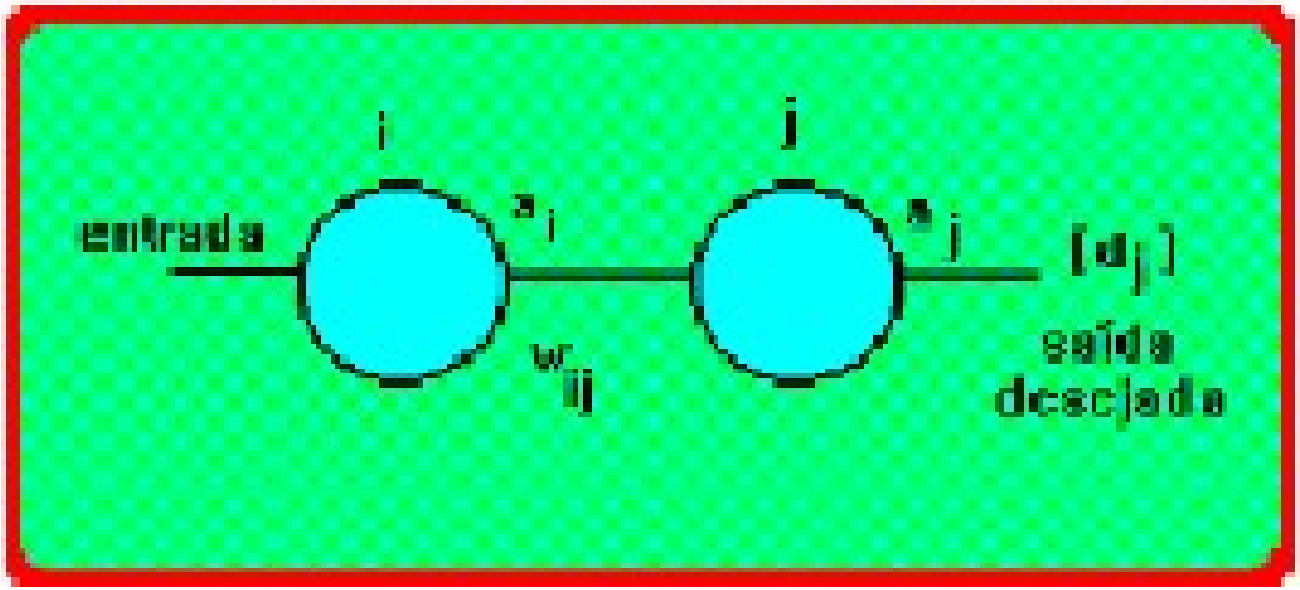
A regra de aprendizado de Hebb propõe que o peso de uma conexão sináptica deve ser ajustado se houver sincronismo entre os níveis de atividade das entradas e saídas [Hebb, 1949].



$$\Delta Wij(t) = \eta\ ai(t)aj(t)$$

Regra Delta

O treinamento supervisionado do modelo de rede Perceptron, consiste em ajustar os pesos e a polarização (bias ou threshold) de suas unidades para que a classificação desejada seja obtida. Para adaptação da polarização juntamente com os pesos pode-se considerá-lo como sendo o peso associado a uma conexão cuja entrada é sempre igual a -1 e adaptar o peso relativo a essa entrada. Quando um padrão é inicialmente apresentado à rede, ela produz uma saída. Após medir a distância entre a resposta atual e a desejada, são realizados os ajustes apropriados nos pesos das conexões de modo a reduzir esta distância. Esse procedimento é conhecido por Regra Delta. É uma variante da Regra de Hebb, introduzida por Widrow-Hoff (Widrow,1988), a diferença quanto a de Hebb é que possui uma saída desejada dj assim o peso será proporcional à saída.



$$\Delta W_{ij} = (d_j - a_j)\, a_i$$

Deste modo, tem-se o seguinte esquema de treinamento:

- Iniciar todas as conexões com pesos aleatórios;
- Repita até que o erro seja satisfatoriamente pequeno
- Para cada par de treinamento faça:
- Calcular a Resposta Obtida
- Se o erro não for satisfatoriamente pequeno então
- Atualizar os pesos: peso novo := peso anterior + taxa de aprendizado

Onde:

1. O par de treinamento corresponde ao padrão de entrada e a sua respectiva resposta desejada;

2. O erro é definido como: Resposta Desejada - Resposta Obtida

3. A taxa de aprendizado é uma constante positiva, que corresponde à velocidade do aprendizado

As respostas geradas pelas unidades são calculadas através de uma função de ativação.

Aprendizado Competitivo

No aprendizado competitivo, os neurônios são inibidos por outros neurônios de modo que a competição entre eles leva apenas um acabar excitado. Assim, enquanto uma rede neural baseada em aprendizado Hebbiano, vários neurônios de saída podem estar simultaneamente ativos, no caso do aprendizado competitivo, somente um neurônio de saída fica ativo a cada vez. Fundamentalmente existem três elementos que caracterizam o aprendizado competitivo:

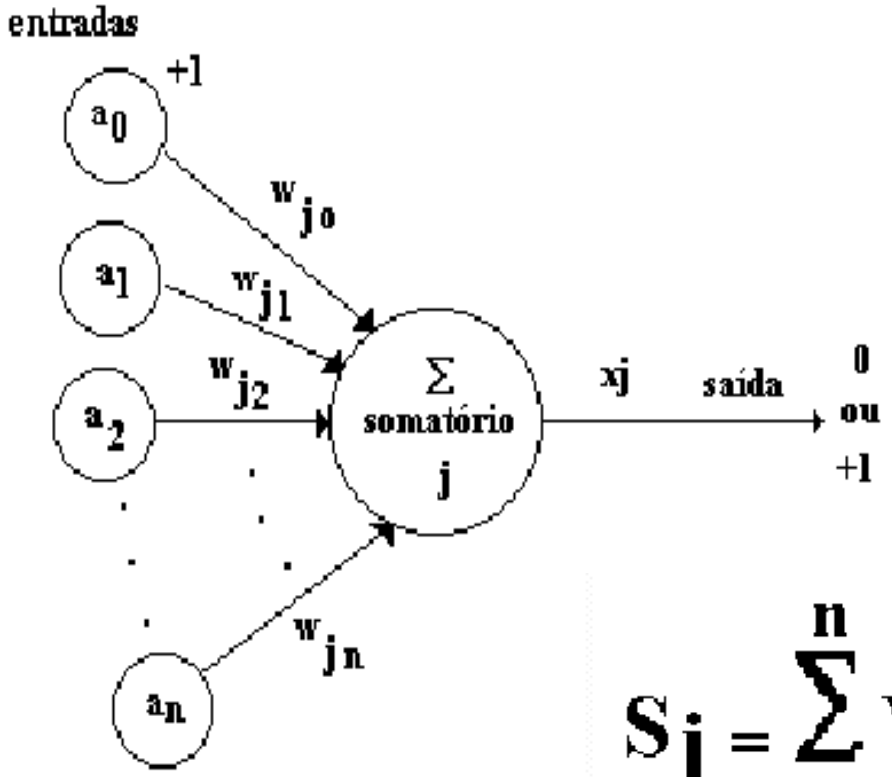
1. Existe um conjunto de neurônios idênticos, ligados por valores de conexões sinápticas de valores distribuídos de modo aleatórios.
2. Existe um valor máximo bem definido para ativação dos neurônios.
3. Existe um mecanismo que permite que os neurônios entrem em competição pelo direito de permanecerem excitados.

No aprendizado competitivo, entradas possuindo alguma semelhança tendem a excitar o mesmo neurônio na saída (Barreto, 1999).

Dois modelos de Redes Neurais Artificiais utilizam o aprendizado competitivo: a rede Counterpropagation e a rede auto-organizável de Kohonen.

Percéptron

O Perceptron foi proposto por Rosenblatt (1959) para reconhecimento de letras maiúsculas do alfabeto. É uma rede direta consistindo de unidades binárias, que aprendem a classificar padrões através de aprendizado supervisionado. Os perceptrons introduzem formalmente a uma lei de treinamento. Modelam o neurônio fazendo a soma ponderada de suas entradas e enviando o resultado 1 se a soma for maior do algum resultado inicial ajustável (caso contrário, ele envia 0). A figura mostra que $a_0, a_1, a_2, \ldots a_n$ são as unidades de entrada e $w_{j0}, w_{j1}, w_{j2}, \ldots, w_{jn}$ são os pesos das conexões e **xj** é a saída do Perceptron.



$$S_j = \sum_i^n W_{ji} a_i$$

Uma entrada especial denominada polarização ("bias" - a_0) tem um valor fixo de +1 . O perceptron testa se a soma ponderada dos pesos está acima ou abaixo de um valor de polarização usando a regra:

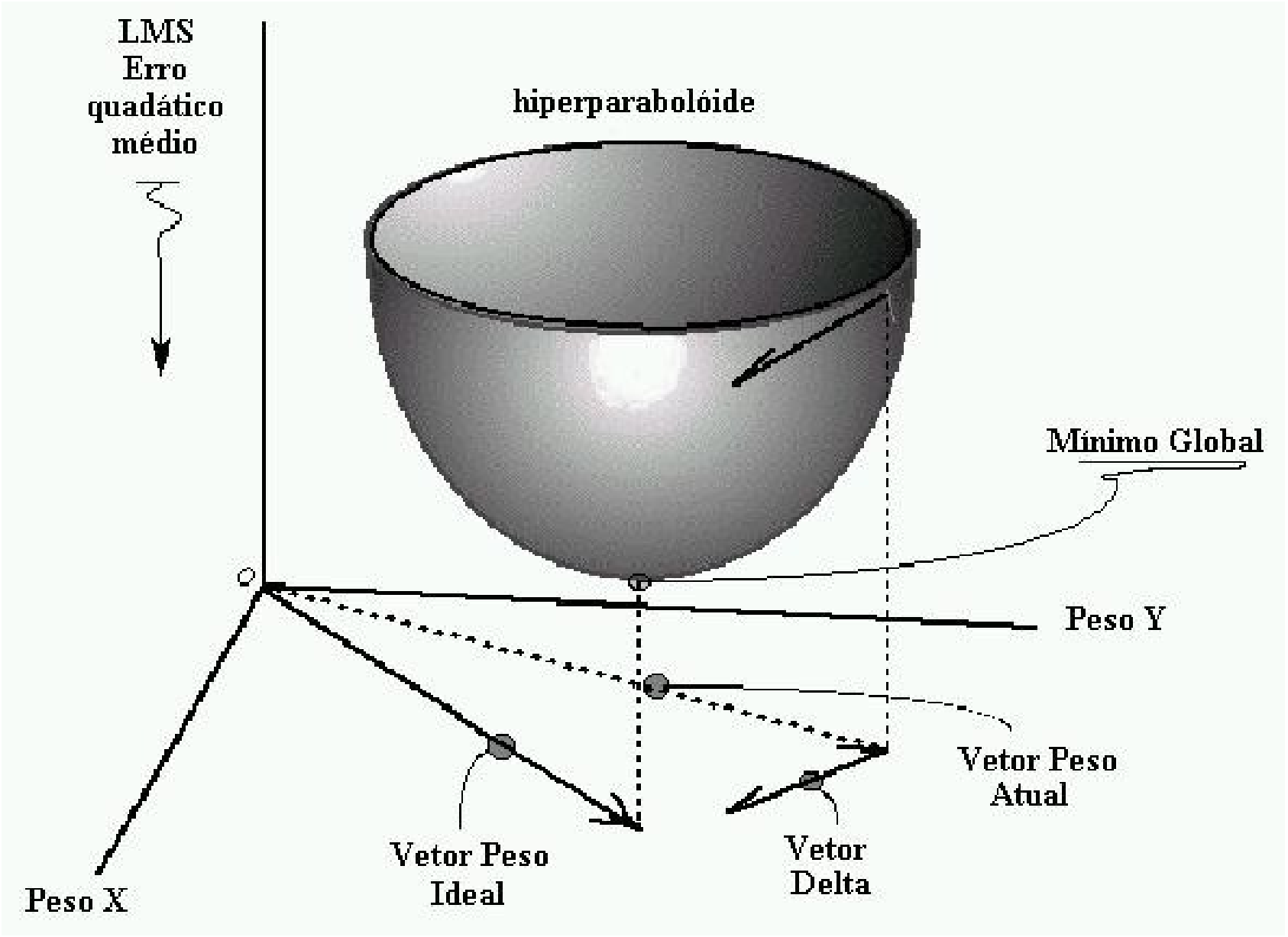
Se **Sj > 0** então **xj = 1**

Se **S =< 0** então **xj = 0**

onde **xj** = valor de saída da unidade de processamento **j**.

Treinamento Supervisionado

- O treinamento de rede Perceptron, consiste em ajustar os pesos e os thresholds (bias) de suas unidades para que a classificação desejada seja obtida.
- Quando um padrão é inicialmente apresentado à rede, ela produz uma saída.
- Após medir a distância entre a resposta atual e a desejada, são realizados os ajustes apropriados nos pesos de modo a reduzir esta distância.
- Este procedimento é conhecido como Regra Delta


LMS
Erro
quadático
médio
hiperparabolóide
Mínimo Global
Peso Y
Vetor Peso
Atual
Vetor Peso
Ideal
Vetor
Delta
Peso X

Esquema de treinamento

Iniciar todas as conexões com pesos aleatórios;

Repita enquanto o erro E > e

Para cada par de treinamento (X,d), faça:

Calcular a resposta obtida O;

Se o erro não for satisfatoriamente pequeno E > e, então:

Atualizar pesos: Wnovo := W anterior + neta E X

Onde:

O par de treinamento (X, d) corresponde ao padrão de entrada e a sua respectiva resposta desejada;

O erro E é definido como: Resposta Desejada - Resposta Obtida (d - O);

A taxa de aprendizado neta é uma constante positiva, que corresponde à velocidade do aprendizado.

Gradient Descent

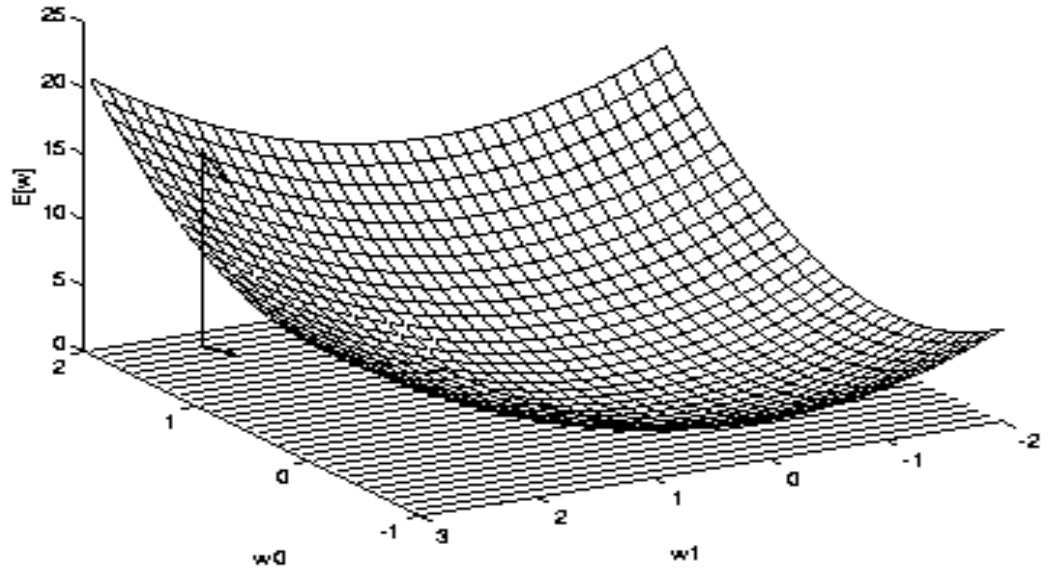
To understand, consider simpler *linear unit*, where

$$o = w_0 + w_1 x_1 + \cdots + w_n x_n$$

Let's learn w_i's that minimize the squared error

$$E[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

Where D is set of training examples



Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

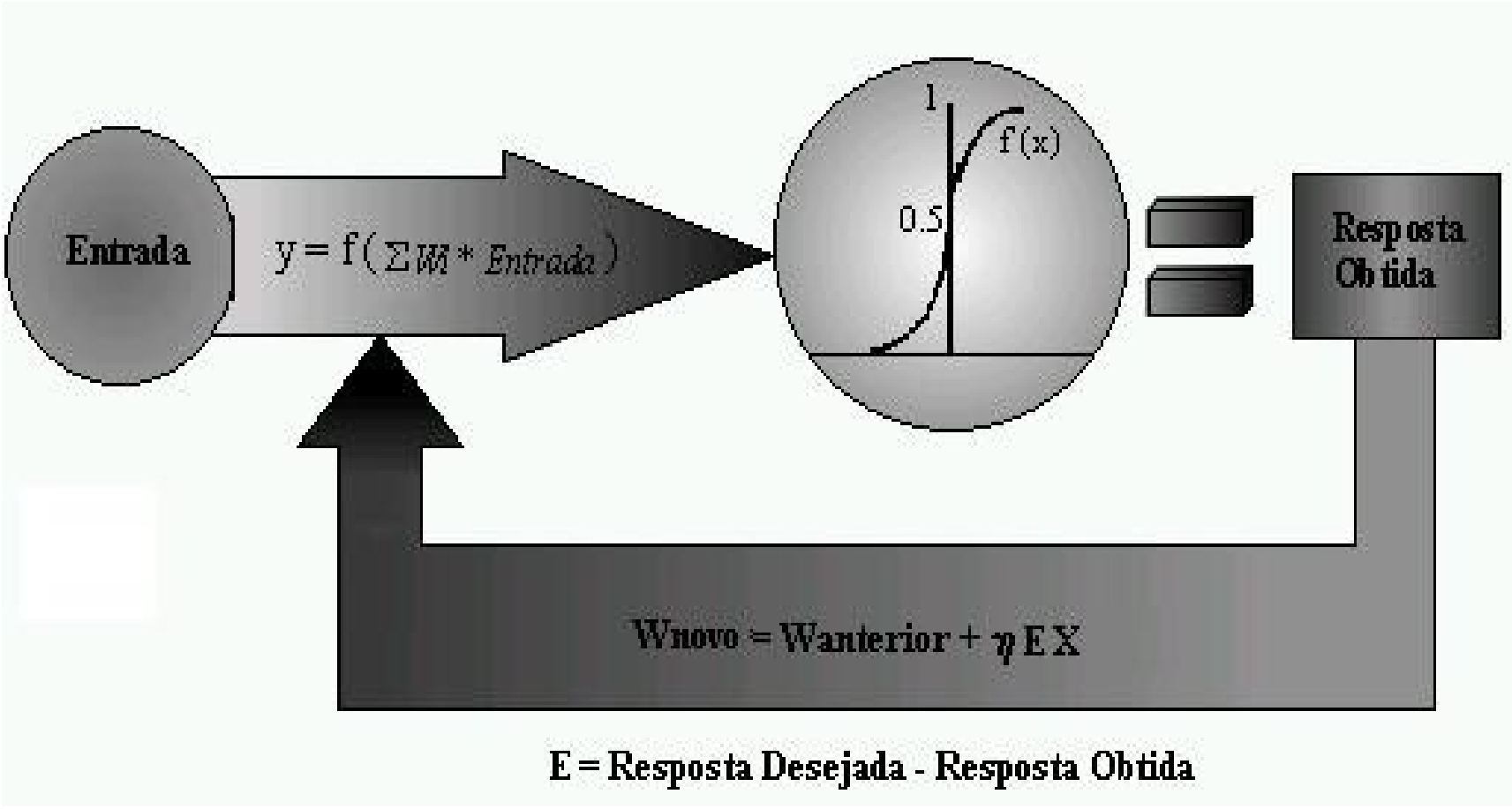
Training rule:

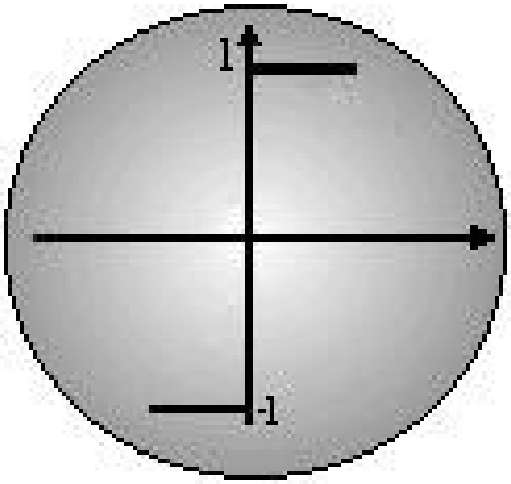
$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

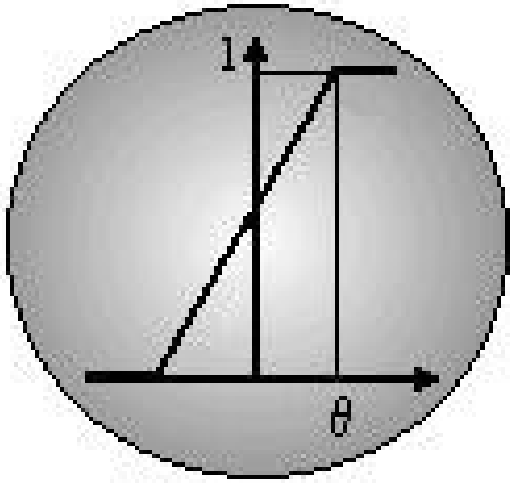
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_d (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_d \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_d 2 (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\
&= \sum_d (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - \vec{w} \cdot \vec{x_d}) \\
\frac{\partial E}{\partial w_i} &= \sum_d (t_d - o_d)(-x_{i,d})
\end{aligned}
$$

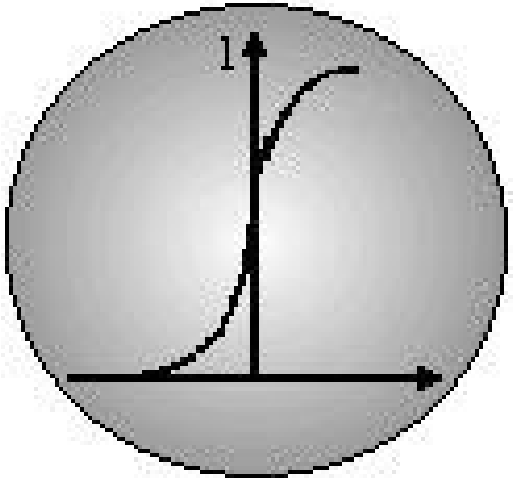

Entrada
y = f(ΣW * Entrada)
1
0.5
f(x)
Resposta Obtida
Wnovo = Wanterior + η E X
E = Resposta Desejada - Resposta Obtida



Hard Limiter



Threshold Logic

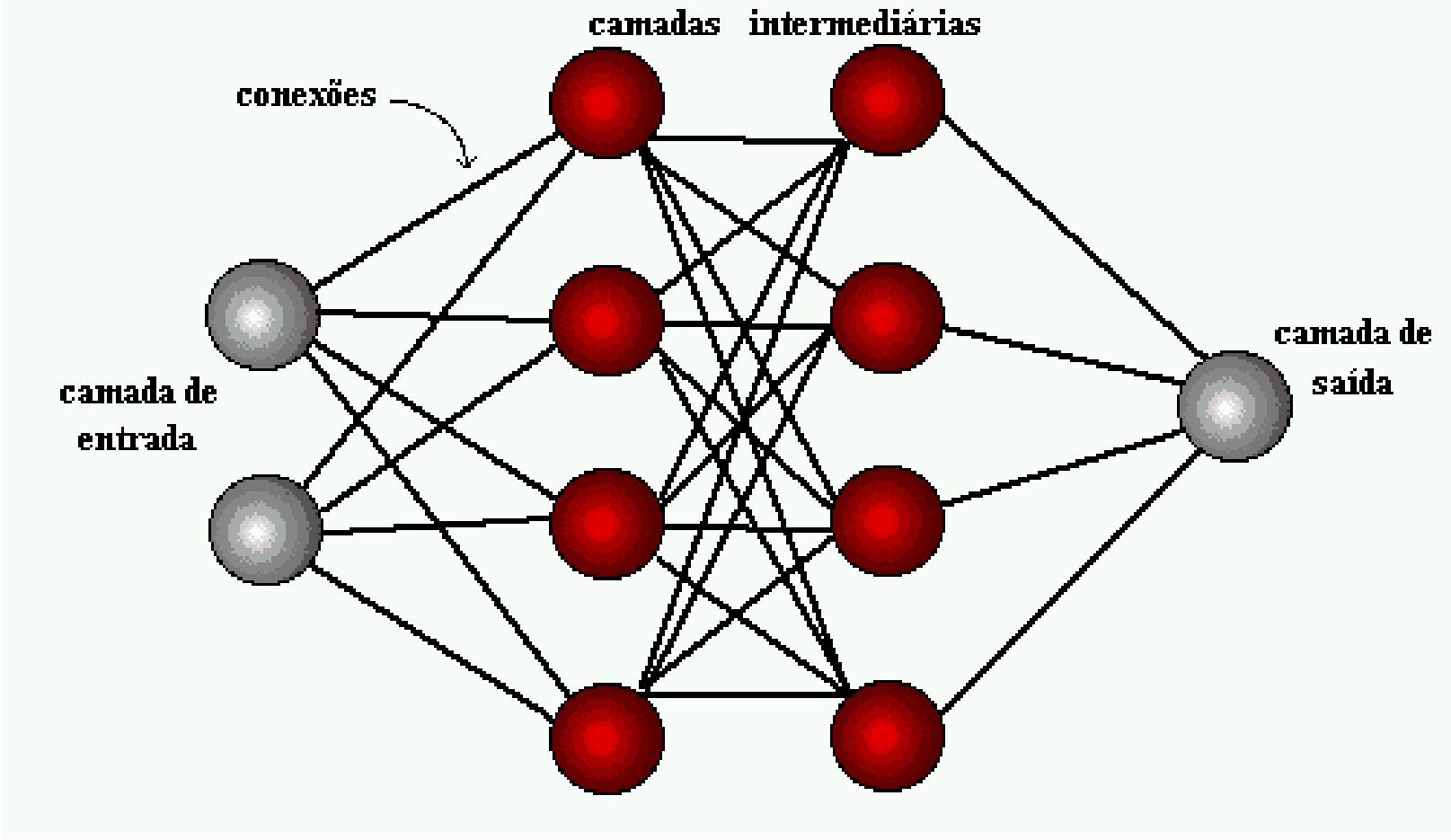


Sigmoid

Potencial de Representação

- Perceptrons representam uma superficie de um hiperplano no espaço n-dimensional
- Alguns problemas não podem ser separados por hiperplanos
- A regra de aprendizado encontra um vetor de pesos se os exemplos são linearmente separaveis
- em cc a regra converge para a “melhor aproximação”

Perceptron Multi-Camadas (MLP)



- o desenvolvimento do algoritmo de treinamento backpropagation, por Rumelhart, Hinton e Williams em 1986, precedido por propostas semelhantes ocorridas nos anos 70 e 80,
- é possível treinar eficientemente redes com camadas intermediárias, resultando no modelo de Perceptron Multi-Camadas (MLP)

- Se existirem as conexões certas entre as unidades de entrada e um conjunto suficientemente grande de unidades intermediárias, pode-se sempre encontrar a representação que irá produzir o mapeamento correto da entrada para a saída através das unidades intermediária.
- Como provou Cybenko, a partir de extensões do Teorema de Kolmogoroff, são necessárias no máximo duas camadas intermediárias, com um número suficiente de unidades por camada, para se produzir quaisquer mapeamentos.
- Também foi provado que apenas uma camada intermediária é suficiente para aproximar qualquer função contínua.

Backpropagation

- Durante o treinamento com o algoritmo backpropagation, a rede opera em uma sequência de dois passos.
  - Primeiro, um padrão é apresentado à camada de entrada da rede. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída.
  - segundo passo, a saída obtida é comparada à saída desejada para esse padrão particular. Se esta não estiver correta, o erro é calculado. O erro é propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados conforme o erro é retropropagado.

- As redes que utilizam backpropagation trabalham com uma variação da regra delta, apropriada para redes multi-camadas: a regra delta generalizada.
- A regra delta padrão essencialmente implementa um gradiente descendente no quadrado da soma do erro para funções de ativação lineares.
- Entretanto, a superfície do erro pode não ser tão simples, as redes ficam sujeitas aos problemas de de mínimos locais.

- A regra delta generalizada funciona quando são utilizadas na rede unidades com uma função de ativação semi-linear, que é uma função diferenciável e não decrescente. Note que a função threshold não se enquadra nesse requisito. Uma função de ativação amplamente utilizada, nestes casos, é a função sigmoid.
- A taxa de aprendizado é uma constante de proporcionalidade no intervalo [0,1], pois este procedimento de aprendizado requer apenas que a mudança no peso seja proporcional à neta.

- Entretanto, o verdadeiro gradiente descendente requer que sejam tomados passos infinitesimais. Assim quanto maior for essa constante, maior será a mudança nos pesos, aumentando a velocidade do aprendizado, o que pode levar à uma oscilação do modelo na superfície de erro. O ideal seria utilizar a maior taxa de aprendizado possível que não levasse à uma oscilação, resultando em um aprendizado mais rápido.
- O treinamento das redes MLP com backpropagation pode demandar muitos passos no conjunto de treinamento, resultando um tempo de treinamento considerávelmente longo. Se for encontrado um mínimo local, o erro para o conjunto de treinamento pára de diminuir e estaciona em um valor maior que o aceitável.

- Uma maneira de aumentar a taxa de aprendizado sem levar à oscilação é modificar a regra delta generalizada para incluir o termo momentum, uma constante que determina o efeito das mudanças passadas dos pesos na direção atual do movimento no espaço de pesos.
- Desta forma, o termo momentum leva em consideração o efeito de mudanças anteriores de pesos na direção do movimento atual no espaço de pesos. O termo momentum torna-se útil em espaços de erro que contenham longas gargantas, com curvas acentuadas ou vales com descidas suaves.

Algoritmo Backpropagation

- Atualize todos os pesos para números aleatórios [0,1].
- Para cada exemplo de treinamento, faça
  - Para cada unidade de saída k

    $$\delta_k = o_k (1 - o_k)(t_k - o_k)$$

  - Para cada unidade escondida h

    $$\delta_h = o_h (1 - o_h) \Sigma w_{h,k} \delta_k \text{ (k, numero de saídas)}$$

- Atualize cada peso da rede

$$w_{h,k} = w_{h,k} + \Lambda w_{h,k} \text{ onde}$$

$$\Lambda w_{h,k} = \eta \delta_k x_{h,k}$$

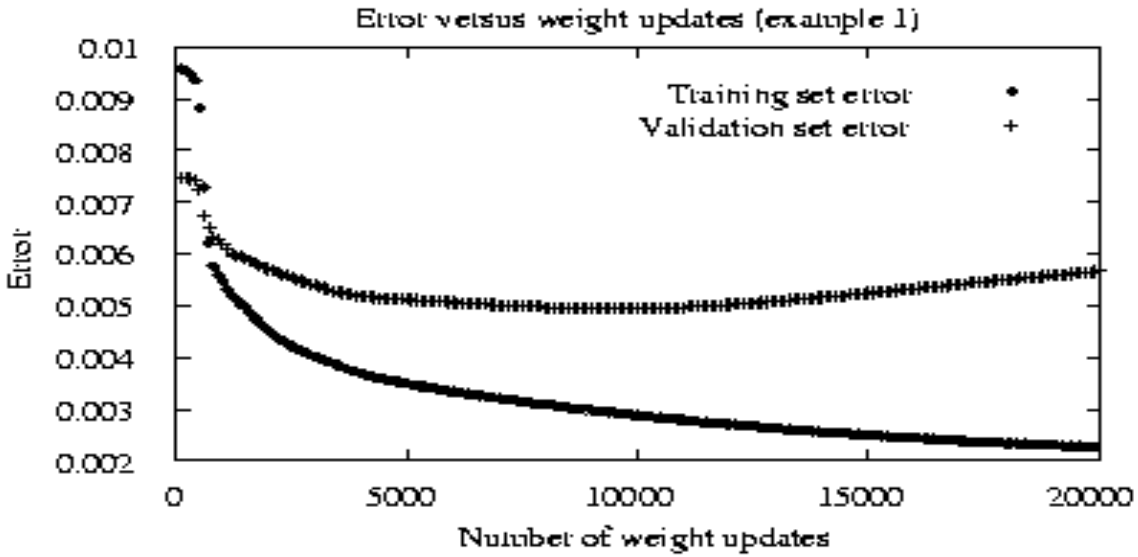
ou

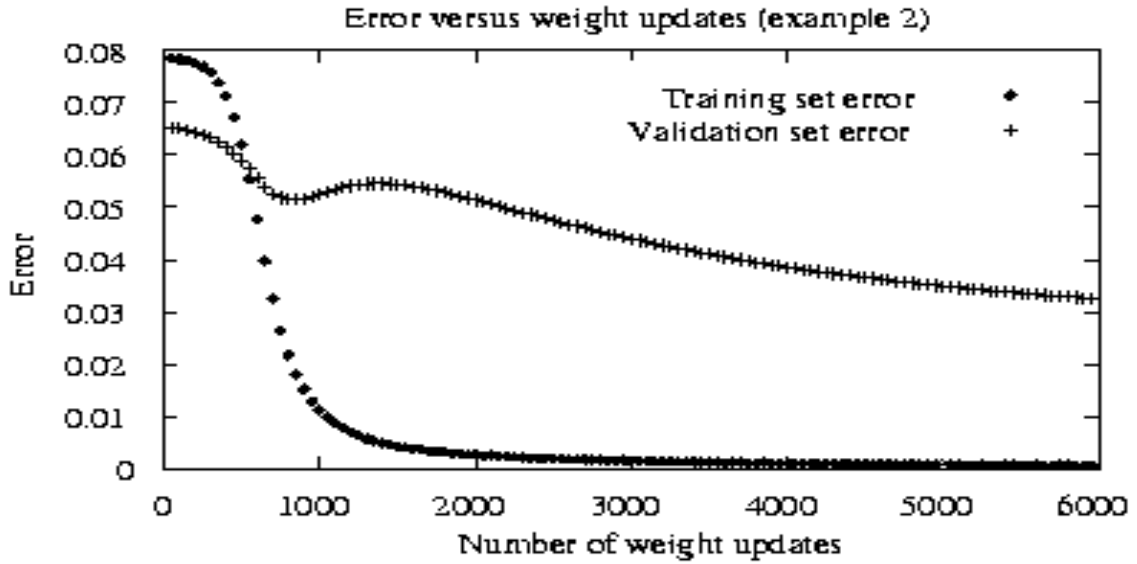
$$\Lambda w_{h,k} = \eta \delta_k x_{h,k} + \alpha \Lambda w_{h,k} (n-1)$$

Convergência do Algoritmo

- Ótimos Locais
- Adicionar momentum
- Treinar múltiplas redes com diferentes pesos iniciais

Overfitting in ANNs



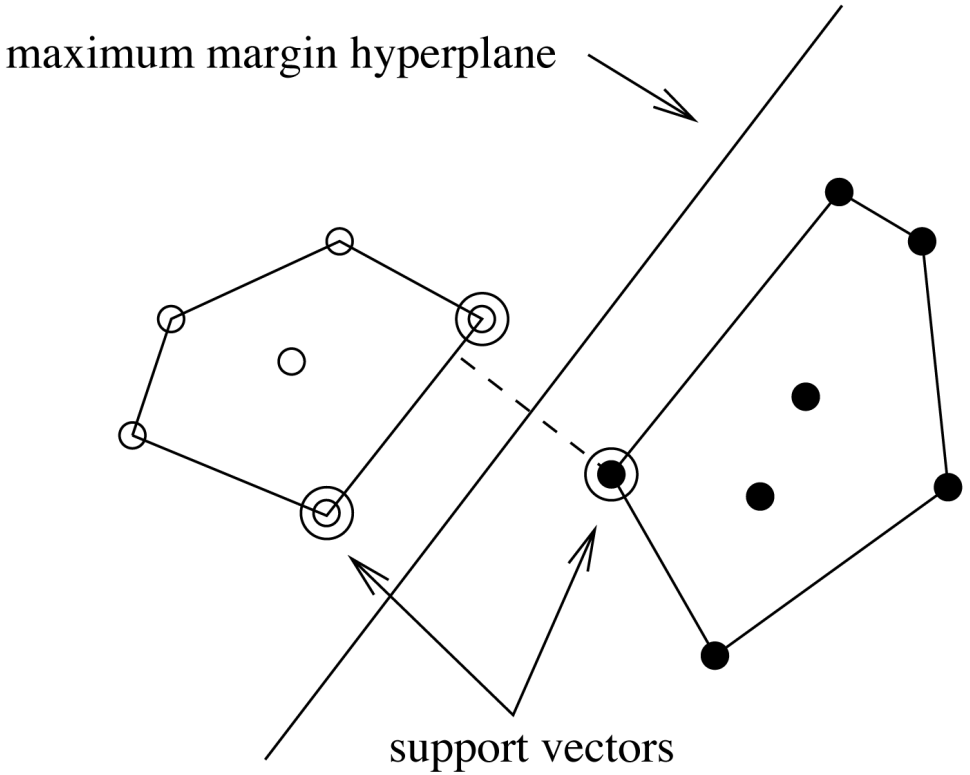


Support Vector Machines

SVM

- SVM são algoritmos para o aprendizado de classificadores.
- Menos susceptíveis a ruído
  - Um hiperplano 'maximum margem'
- Rápida nos casos não lineares
  - utiliza uma jogada matematica para evitar de criar 'pseudo-atributos'
  - espaços não-lineares são criados implicitamentes

Hiperplano



as instâncias mais próximas ao hiperplano são chamadas SV

Encontrando SV

- o hiperplano pode ser escrito

problema matemático bem definido

SVM não lineares

- mapear o problema num linear
- atributos são criados pela combinação de outros
- porem não são calculados os pseudo atributos
- produto interno
- Exemplo

$$x = b + \sum \alpha_i y_i (\vec{a}(i) \cdot \vec{a})^n$$

Funções de Kernel

- Mapear é feito com funções de kernel
- polinomial

$$x = b + \sum \alpha_i y_i \left(\vec{a}(i) \cdot \vec{a}\right)^n$$

- podemos usar outros

$$K\left(\vec{x}_i, \vec{x}_j\right) = \left(\vec{x}_i \cdot \vec{x}_j + 1\right)^d$$

$$K\left(\vec{x}_i, \vec{x}_j\right) = \exp\left(\frac{-\left(\vec{x}_i - \vec{x}_j\right)^2}{2\sigma^2}\right)$$

$$K\left(\vec{x}_i, \vec{x}_j\right) = \tanh\left(\beta\, \vec{x}_i \cdot \vec{x}_j + b\right)$$