

Redes Neurais Artificiais

Airton Bordin Junior
[airtonbjunior@gmail.com]

Mestrado em Ciência da Computação – Inteligência Computacional
Profº Dr Celso Gonçalves Camilo Junior

Universidade Federal de Goiás (UFG) - Instituto de Informática - Maio/2017

Programação

- Introdução
- Heurísticas e Metaheurísticas
- Algoritmos evolucionários
- Redes Neurais Artificiais
- Referências





Introdução

- Podemos classificar os problemas computacionais em
 - Tratáveis, também chamados de polinomiais
 - Podem ser resolvidos por algoritmos determinísticos.
 - Intratáveis, ou não polinomiais
 - Sem algoritmo determinístico para resolver o problema em tempo hábil.

Tratáveis

- Polinomiais
- Algoritmos determinísticos

Intratáveis

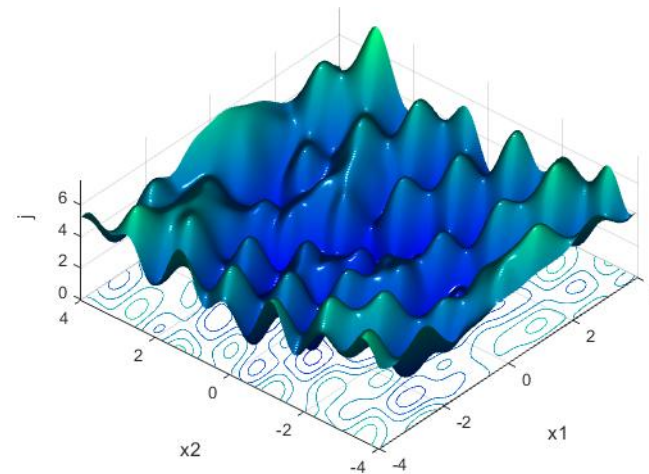
- Não polinomiais
- Algoritmos não determinísticos
- Solução determinística inviável





Heurística

- Impraticabilidade de encontrar/calcular a melhor resposta para problemas não polinomiais;
- Desafio: produzir, em tempo reduzido, soluções tão próximas quanto possíveis da solução ótima.

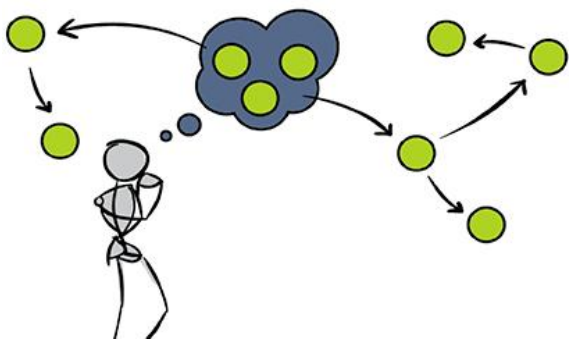




Metaheurística

Propriedades e características das metaheurísticas

[SALIBA, 2010]



Estratégias que guiam o processo de busca;

Exploração eficiente do espaço de busca - soluções ótimas ou quase ótimas;

De simples procedimentos de busca local a complexos processos de aprendizado;

Aproximados e usualmente não determinísticos;

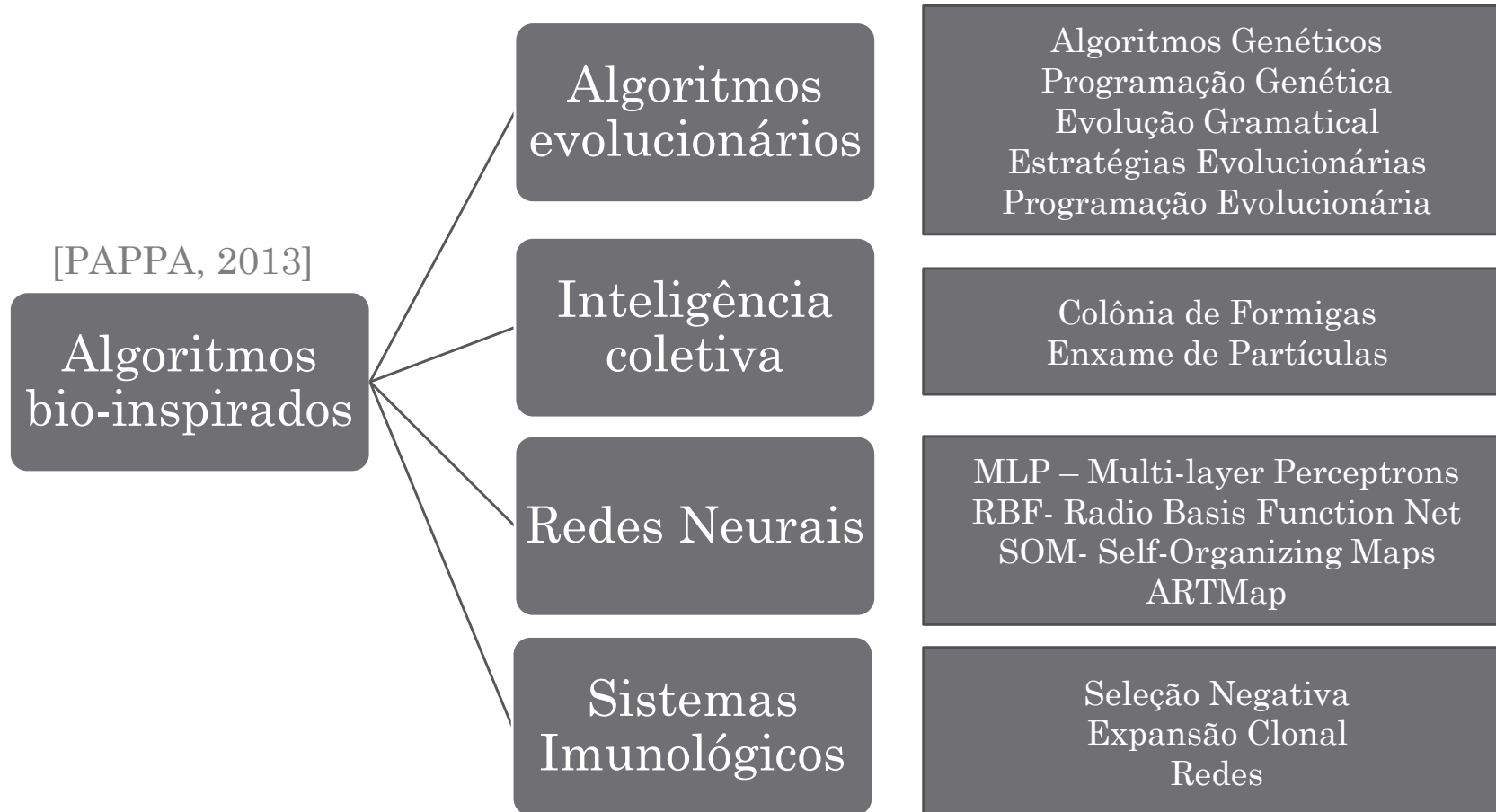
Podem incorporar mecanismos para evitar ficar presos em áreas confinadas do espaço de busca;

Não são específicas para um determinado problema;

Podem usar um conhecimento específico do problema na forma de heurísticas que são controladas por uma estratégia de nível superior.

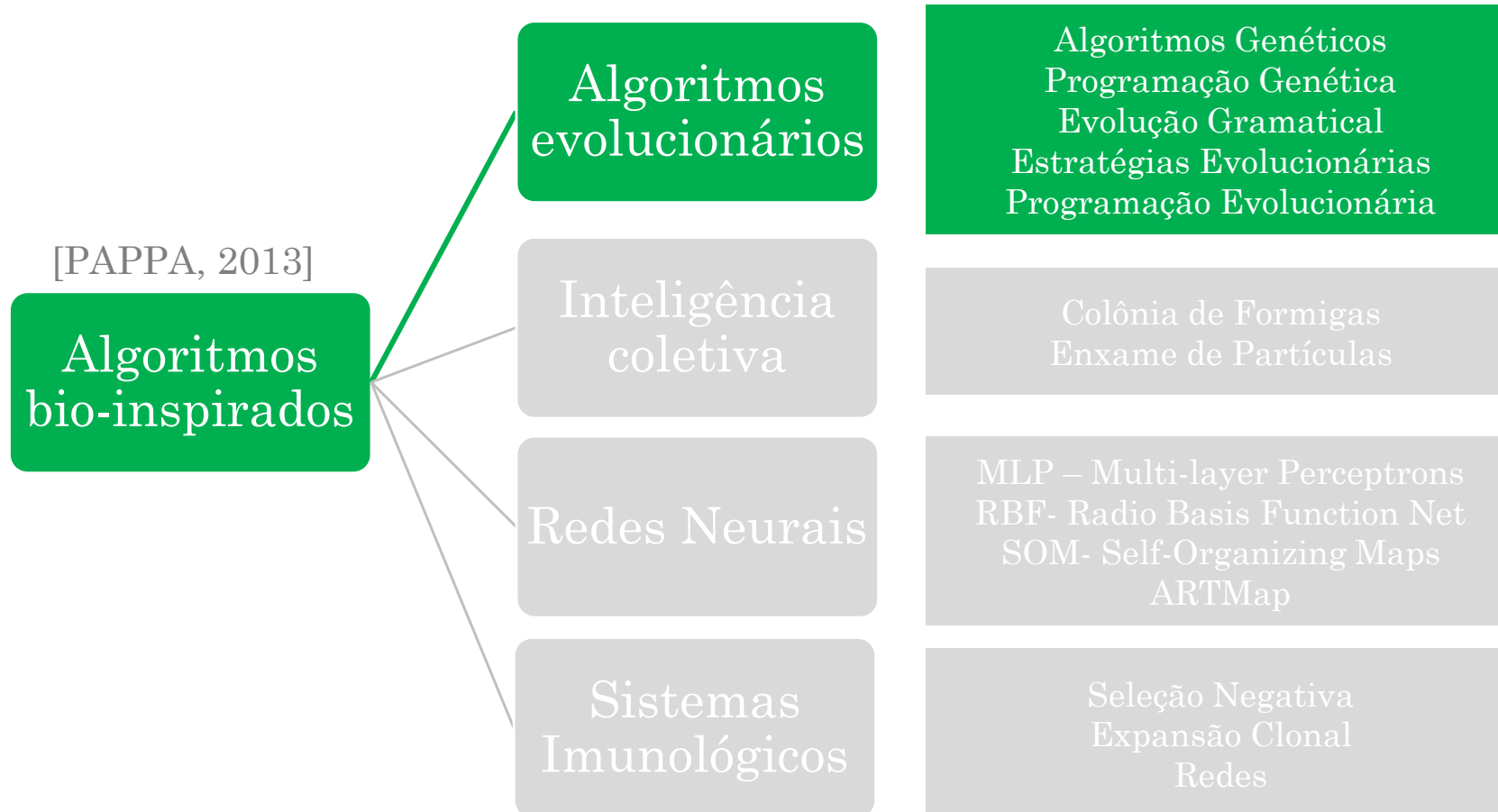


Algoritmos bio-inspirados





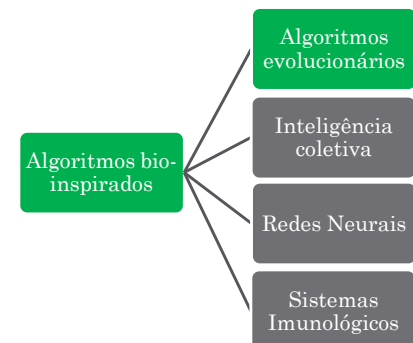
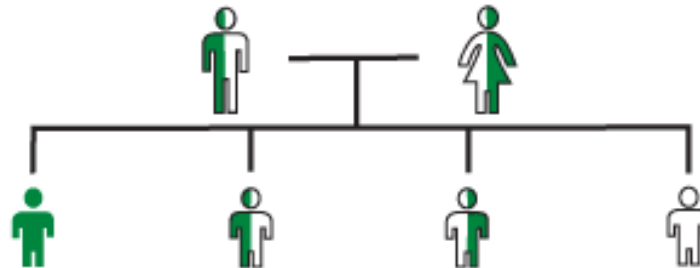
Algoritmos bio-inspirados





Algoritmos evolucionários

- Inspirados na teoria de evolução de Darwin;
- Evolução: mudança das características (genéticas) de uma população de uma geração para a próxima
 - Mutação dos genes;
 - Recombinação dos genes dos pais.

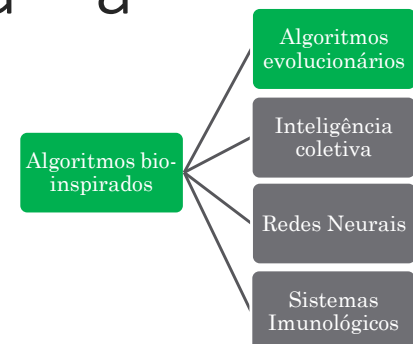




Algoritmos evolucionários

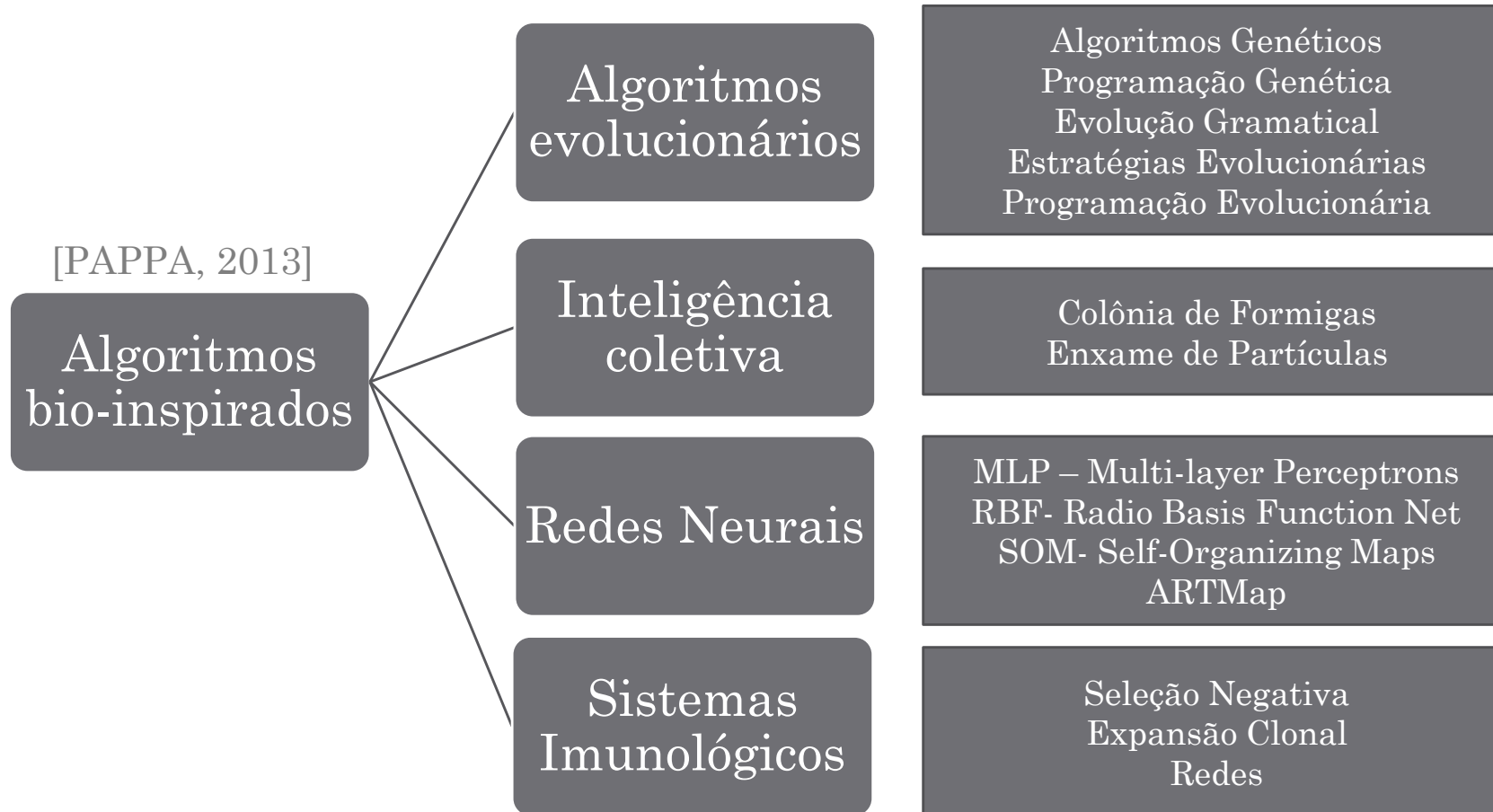
- Evolução é caracterizada basicamente por um processo constituído de 3 passos [VON ZUBEN, 2005]

1. Reprodução com herança genética;
2. Introdução de variação aleatória em uma população de indivíduos;
3. Aplicação da “seleção natural” para a produção da próxima geração.



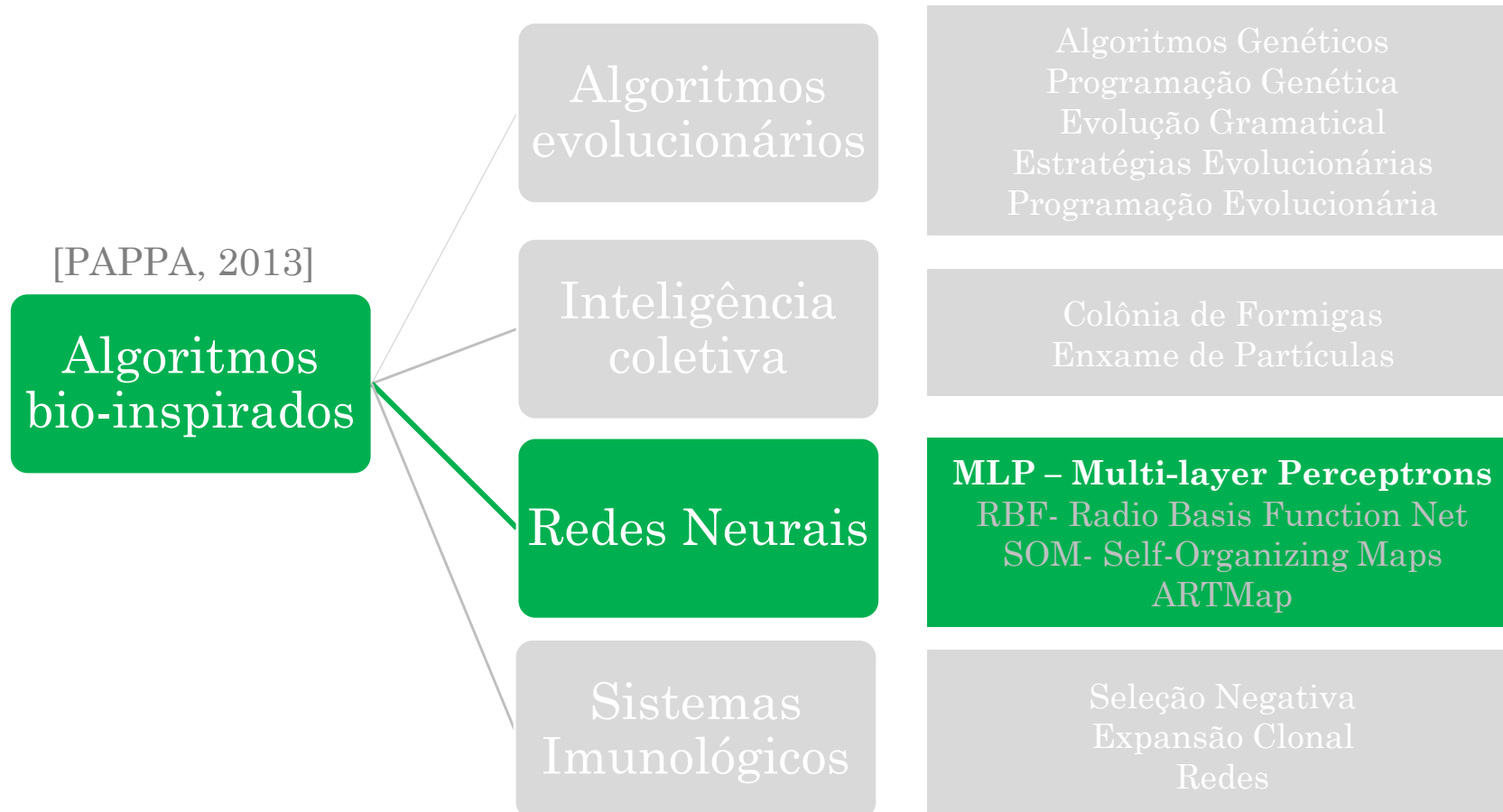


Algoritmos bio-inspirados





Algoritmos bio-inspirados



Redes Neurais Artificiais



Redes Neurais

- Sistema nervoso é formado por um conjunto extremamente complexo de células: os neurônios.
- Essencial na determinação do funcionamento e comportamento do corpo humano e do raciocínio.





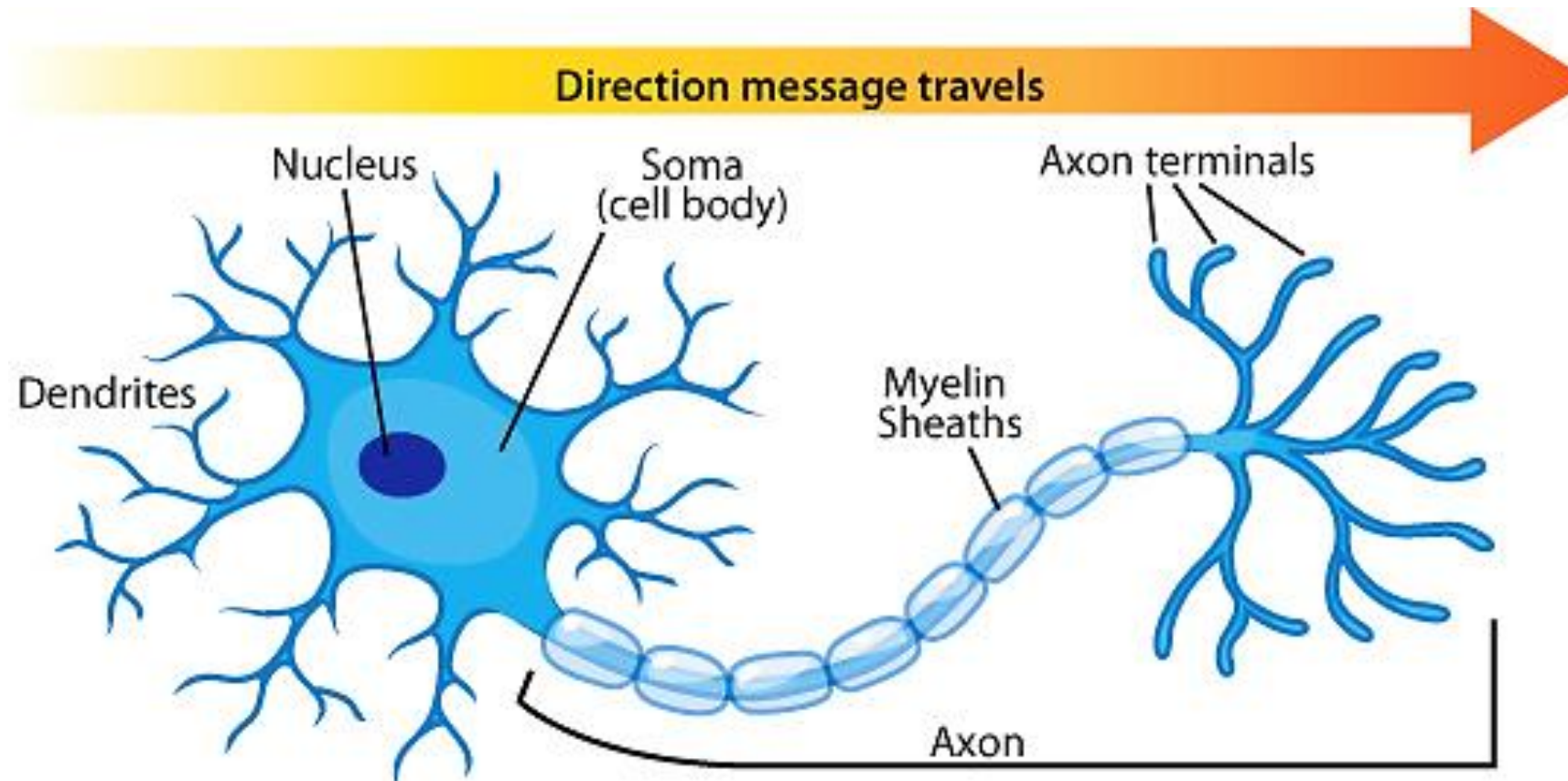
Redes Neurais

- Principais partes de um neurônio
 - **Dentritos**, que recebem os estímulos;
 - **Corpo (soma)**, responsável por coletar e combinar informações;
 - **Axônio**, que transmite os estímulos para outras células.





Neurônio





Redes Neurais Artificiais

- Modelos matemáticos que se assemelham às estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de aprendizagem e generalização;
- Inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.





Redes Neurais Artificiais

- Estrutura de processamento, composta por um número de unidades interconectadas, onde cada unidade apresenta um comportamento específico de entrada/saída, determinado pela sua função de transferência, pelas interconexões com outras unidades, dentro de um raio de vizinhança, e possivelmente pelas entradas externas. [VON ZUBBEN, 2003]





Redes Neurais Artificiais

- Circuito composto por uma grande quantidade de unidades simples de processamento inspiradas no sistema neural;
[NIGRIN, 1993]
- Sistema massivamente paralelo e distribuído, composto por unidades de processamento simples que possuem uma capacidade natural de armazenar e utilizar conhecimento
[HAYKIN, 1999]





Redes Neurais Artificiais

- Propriedades particulares de uma RNA:
 - Aprender;
 - Adaptar;
 - Generalizar;
 - Eventualmente organizar.





Histórico resumido

- **1943:** McCulloch e Pitts - construção de uma máquina baseada ou inspirada no cérebro humano;
- **1949:** Donald Hebb – “*The Organization of Behavior*” - traduziu matematicamente a sinapse dos neurônios biológicos;
- **1951:** Mavin Minsky - primeiro neurocomputador: Snark (não executou nenhuma função útil);
- **1956:** Dartmouth College – 2 paradigmas de IA: Simbólica e Conexionista;



Histórico resumido

- **1957:** Rosenblatt - primeiro neurocomputador a obter sucesso: Mark I Perceptron (reconhecimento de padrões);
- **1983:** Ira Skurnick – centro de pesquisas em neurocomputação – DARPA (Defense Advanced Research Projects Agency);
- **1986:** David Rumelhart e James McClelland – *“Parallel Distributed Processing”*



Redes Neurais Artificiais

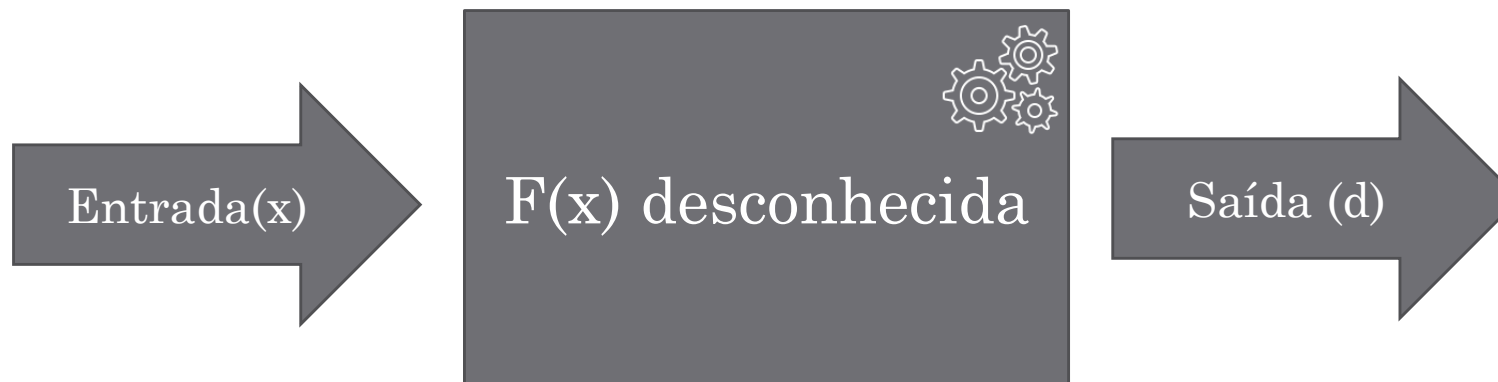
- Operação de uma unidade de processamento
 - Sinais são apresentados à entrada;
 - Cada sinal é multiplicado por um peso (influência na saída da unidade);
 - Somatório ponderado dos sinais produz um nível de atividade;
 - Se este nível de atividade exceder um certo limite (*threshold*) a unidade produz uma determinada resposta de saída.

[MCCULLOCK, PITTS, 1943]



Redes Neurais Artificiais

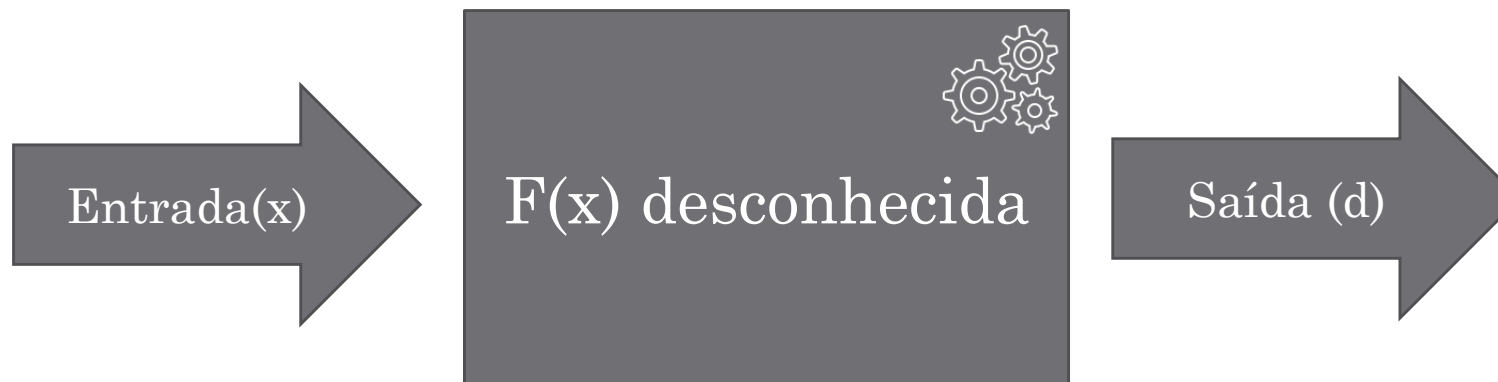
- Função matemática $F(x)$
 - Relaciona um vetor de entrada com um vetor de saída;
 - Assume-se que $F(x)$ é totalmente desconhecida
 - Pode ser tão simples como um mapeamento linear do tipo
 - $d = M.x$





Redes Neurais Artificiais

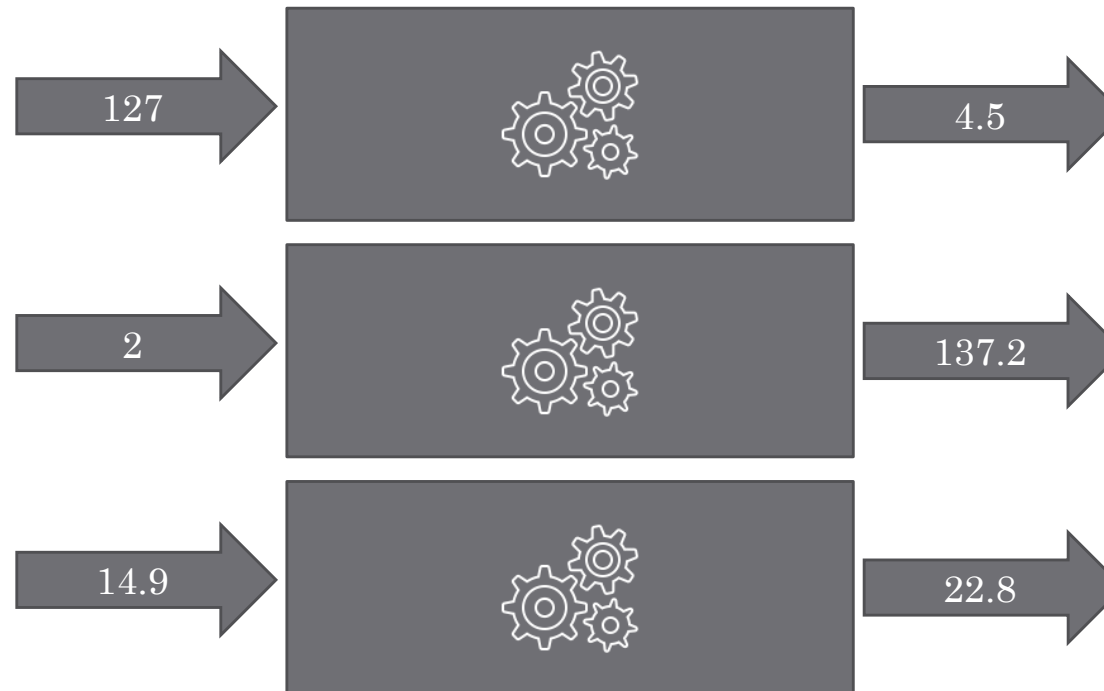
- Função matemática $F(x)$
 - A função pode ser extremamente complexa;
 - Relações não lineares, etc.
 - Objetivo: imitar o funcionamento de $F(x)$.





Redes Neurais Artificiais

- Fontes de informação sobre o problema
 - Conjunto finito de pares de entrada $\{x_i, d_i\}$
 - Exemplo:



...



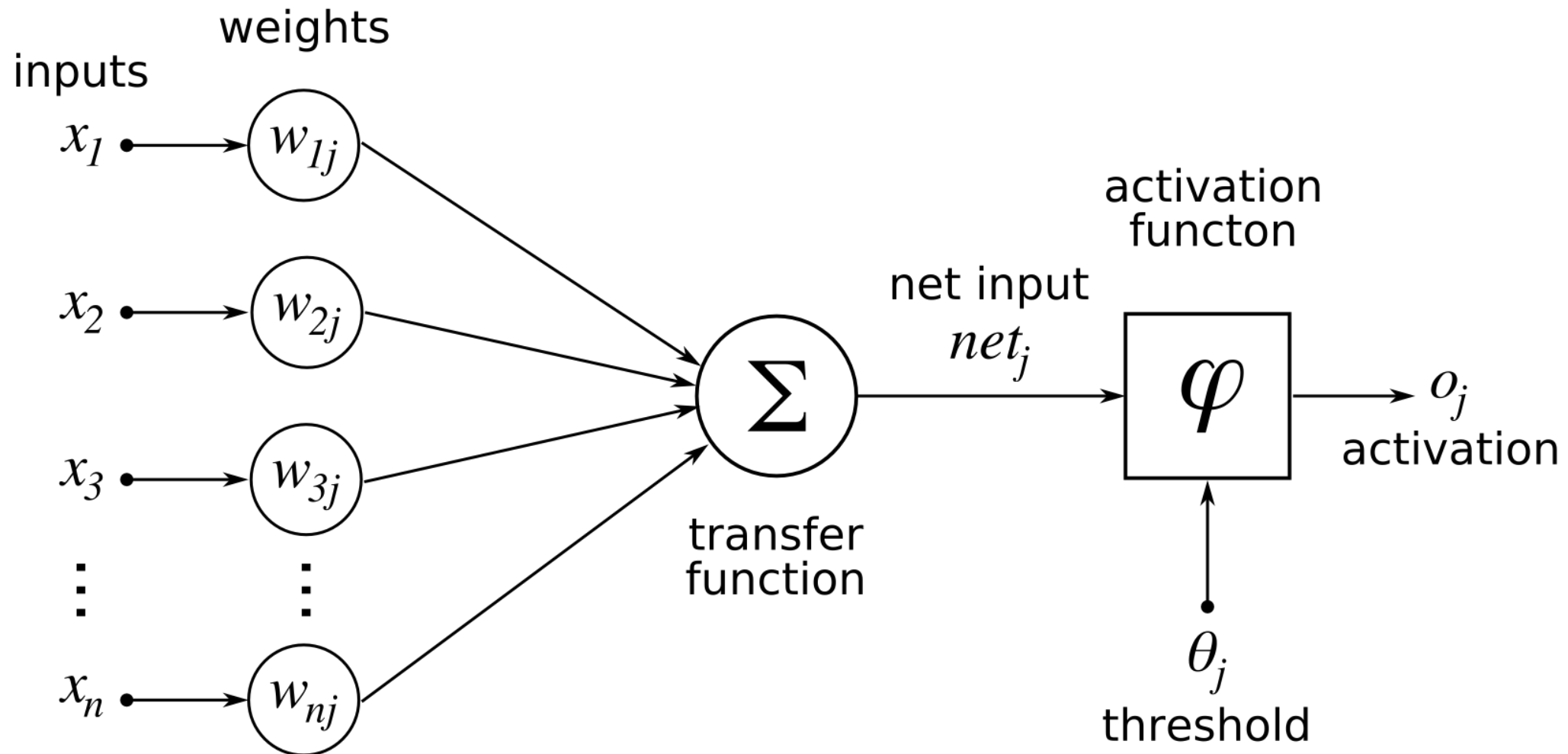
Redes Neurais Artificiais

- Fontes de informação sobre o problema
 - Após a apresentação das entradas conhecidas x e saídas d , a ideia é que a rede seja capaz de retornar uma saída d' para uma entrada x' desconhecida;
 - Saída muito próxima da real;
 - O processo de obtenção de d' a partir das entradas conhecidas é chamado de **Processo Indutivo**.



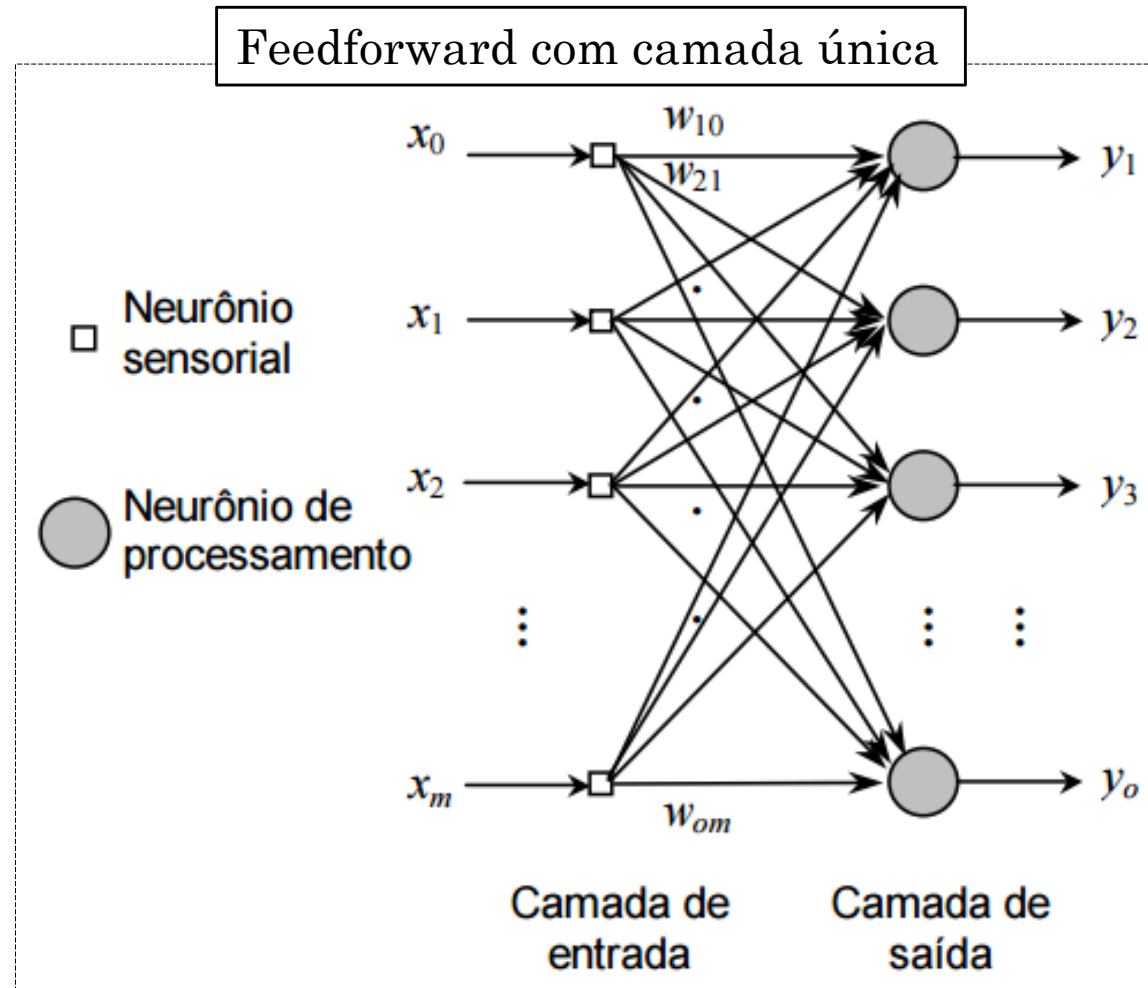


Neurônio artificial



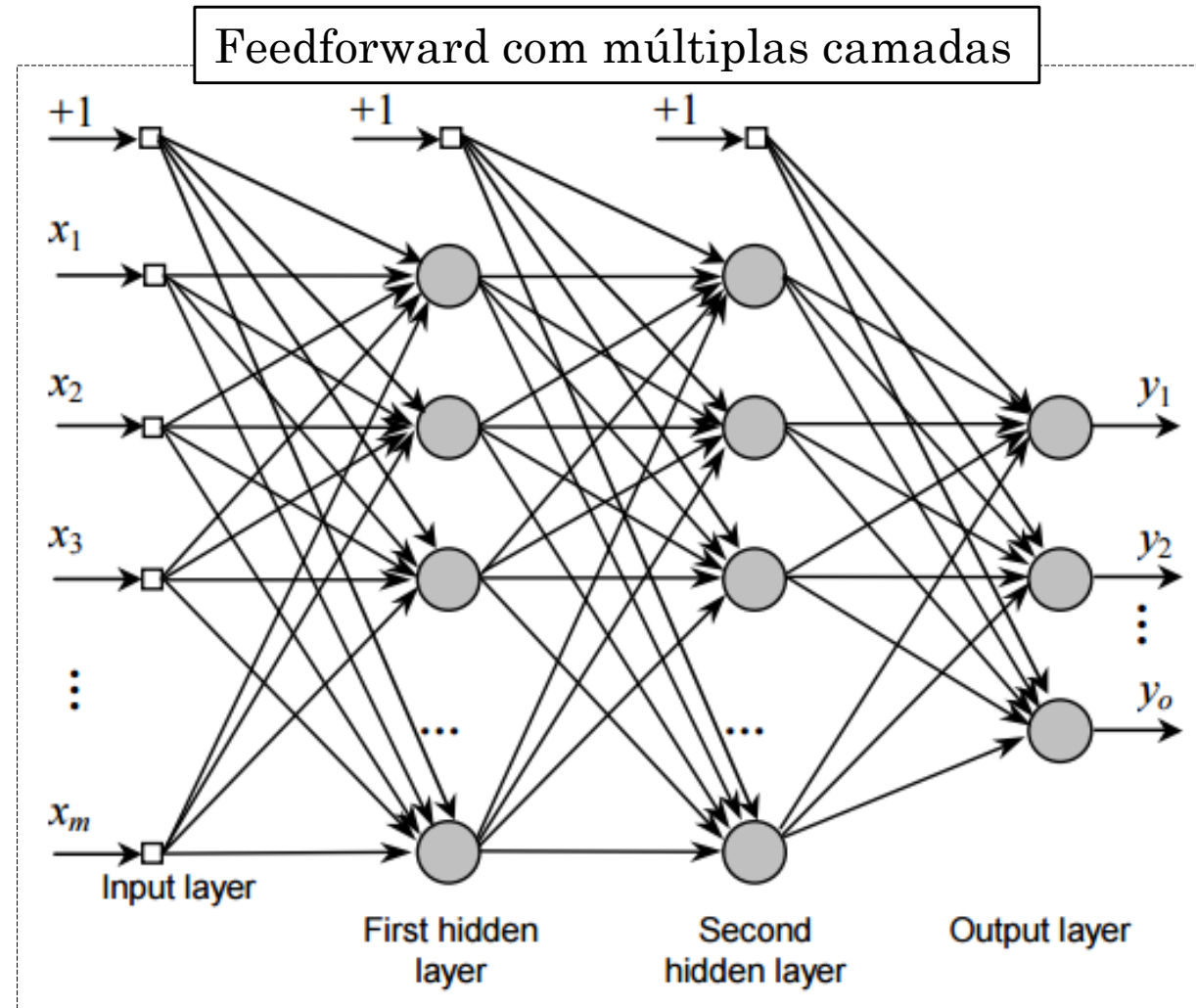


Arquiteturas de redes



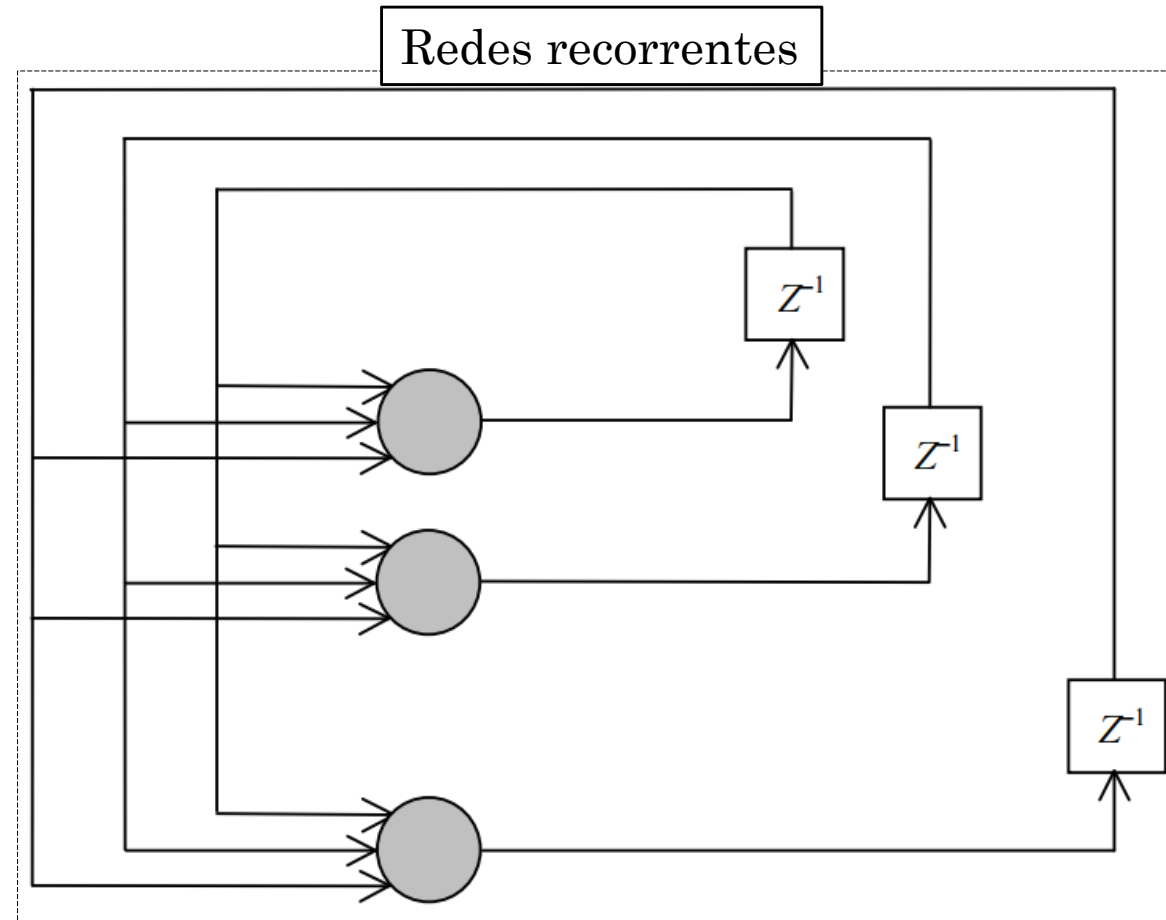


Arquiteturas de redes





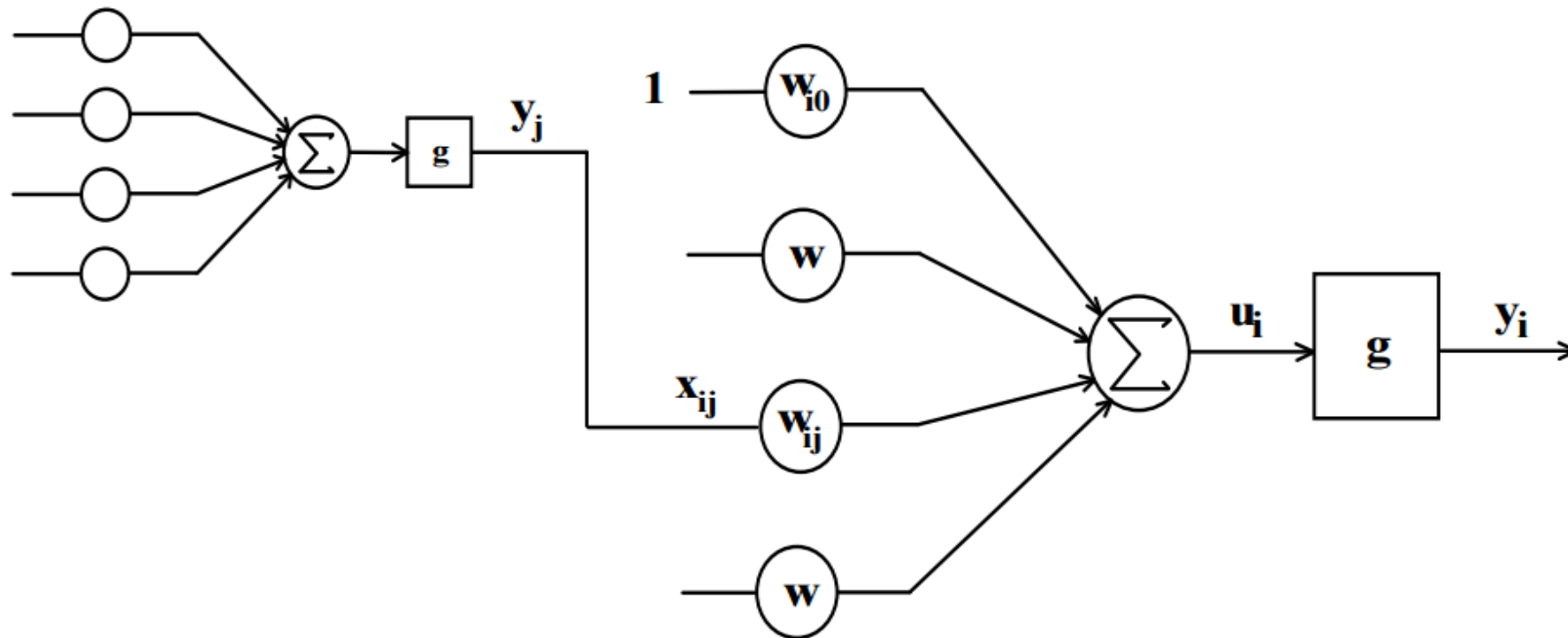
Arquiteturas de redes





Redes Neurais Artificiais

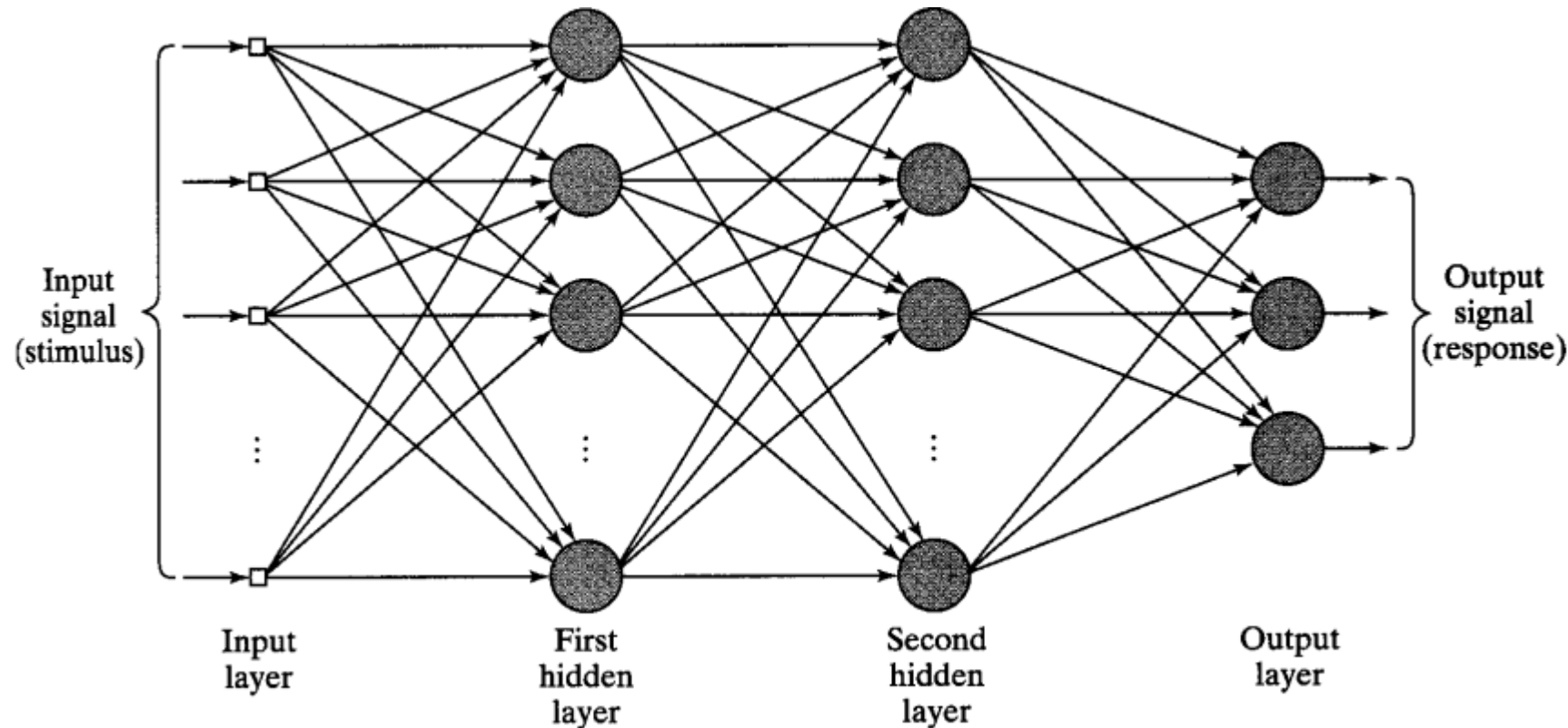
- O processo de conexão de neurônios artificiais leva à geração de sinapses e Redes Neurais Artificiais





Redes Neurais Artificiais

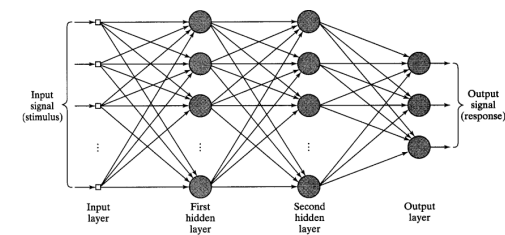
- As estruturas mais conhecidas são as em camadas
 - *Multilayer Perceptron* (MLP)





Multilayer Perceptron (MLP)

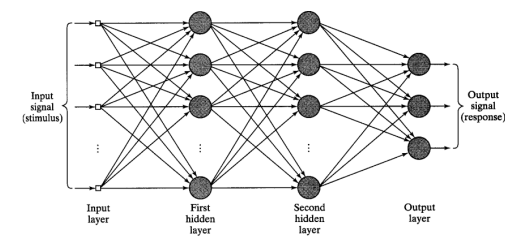
- RNAs de múltiplas camadas resolvem problemas não linearmente separáveis;
- Distinguem-se das redes de camada simples pelo número de camadas intermediárias
 - Possui uma ou mais camadas ocultas, compostas por neurônios artificiais (neurônios ocultos);
 - Adição de camadas ocultas torna a rede capaz de extrair estatísticas de ordem elevada.





Multilayer Perceptron (MLP)

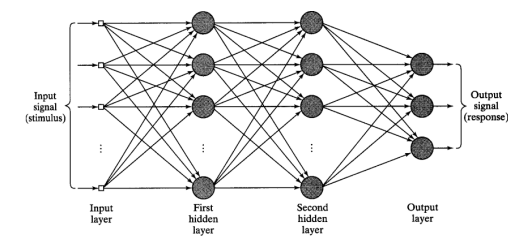
- O número de nós de entrada é determinado pela dimensão do espaço de observação;
- O número de neurônios da camada de saída é determinado pela dimensão da resposta;
- Assim, o projeto requer a consideração de:
 1. Número de camadas ocultas;
 2. Número de neurônios em cada camada oculta;
 3. Especificação dos pesos sinápticos.





Processos de aprendizado

- Se o conhecimento está armazenado nos pesos das conexões, então o processo de aprendizagem corresponde a identificar um conjunto apropriado de pesos de forma que a rede se comporte como desejado;
- Possibilidade de desenvolvimento de técnicas de aprendizagem, e a representação distribuída de conhecimento.



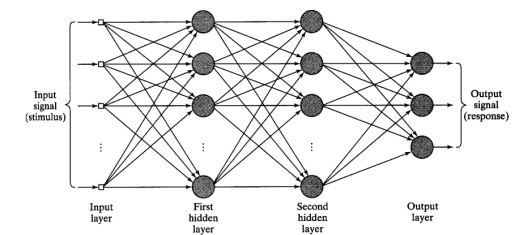


Processos de aprendizado

Estímulo

Adaptação

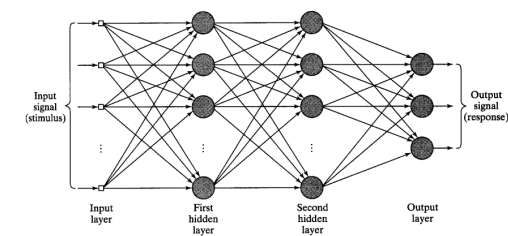
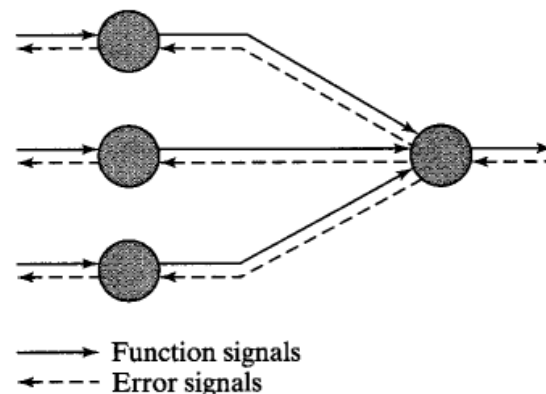
Novo
comportamento





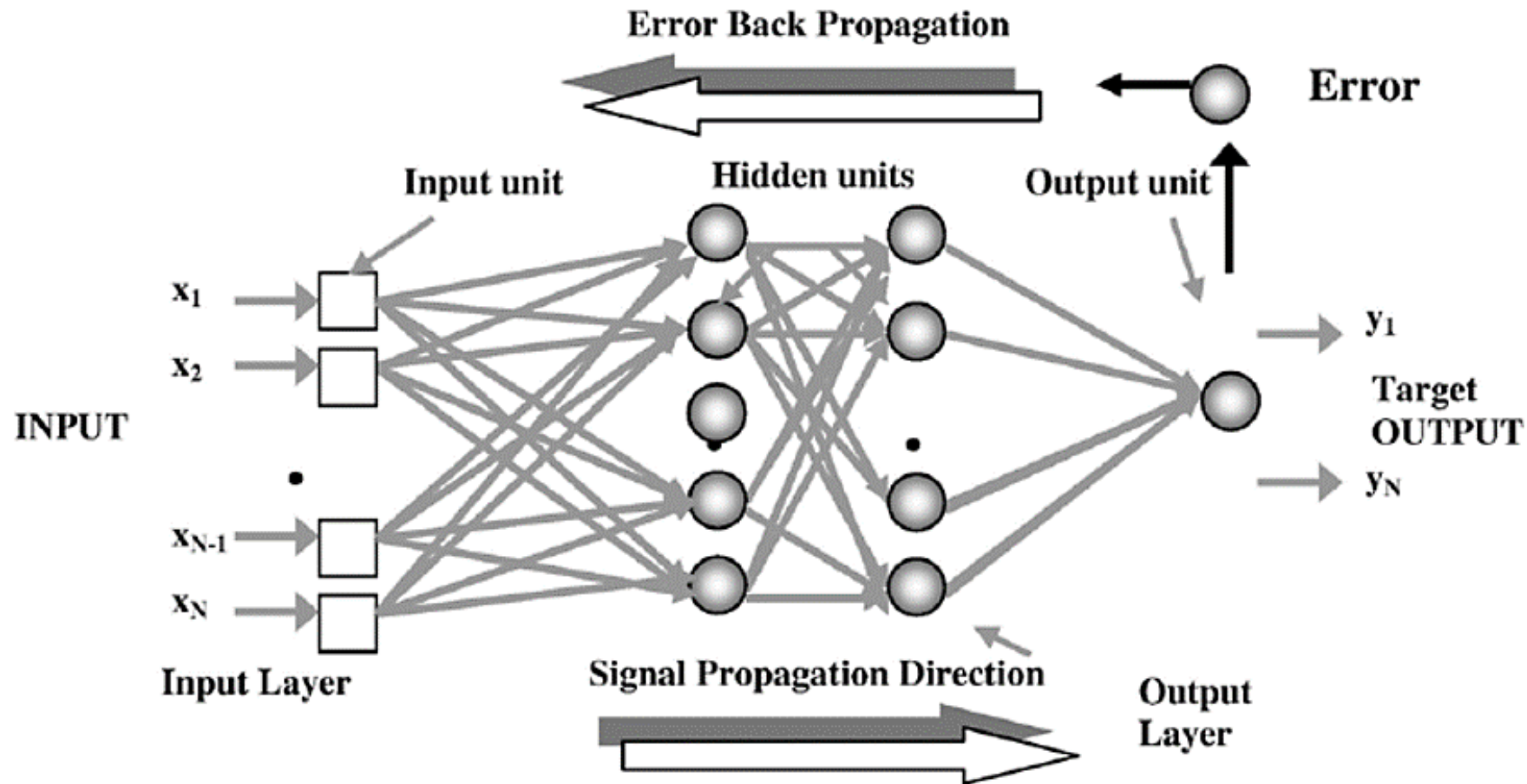
Processos de aprendizado

- O algoritmo de treinamento mais utilizado em MLP é o *Backpropagation*
 - Baseia-se na aprendizagem por correção de erros.
- Quando o valor de saída é gerado, é calculado o erro e seus valores são retro-propagados para entrada
 - Os pesos são ajustados e os valores são novamente calculados.





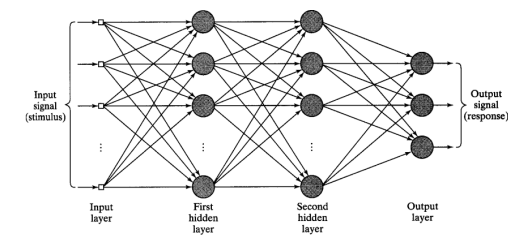
Processos de aprendizado





Processos de aprendizado

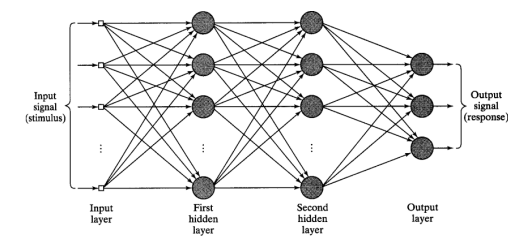
- Funcionamento geral
 1. Apresenta-se um padrão à camada de entrada
 - a. Padrão é processado camada por camada até que a camada de saída forneça a resposta processada.
 2. Resposta é comparada com a resposta desejada e se estiver errada, o erro é calculado;
 3. Valores são retropropagados da camada de saída para a camada de entrada
 - a. Pesos são ajustados.





Processos de aprendizado

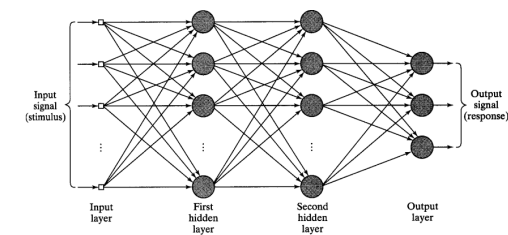
- O aprendizado é feito por meio de um processo iterativo de ajuste dos pesos sinápticos
 1. RNA é estimulada pelo ambiente de informação;
 2. Estrutura interna da rede é alterada como resultado do estímulo;
 3. Por conta das alterações na estrutura interna, a rede tem sua resposta aos estímulos do ambiente modificada.





Processos de aprendizado

- A RNA se baseia nos dados para extrair um modelo geral
 - Fase de aprendizado deve ser rigorosa e verdadeira, a fim de se evitar modelos irreais.
- Todo o conhecimento de uma rede neural está armazenado nas sinapses, ou seja, nos pesos atribuídos às conexões entre os neurônios;





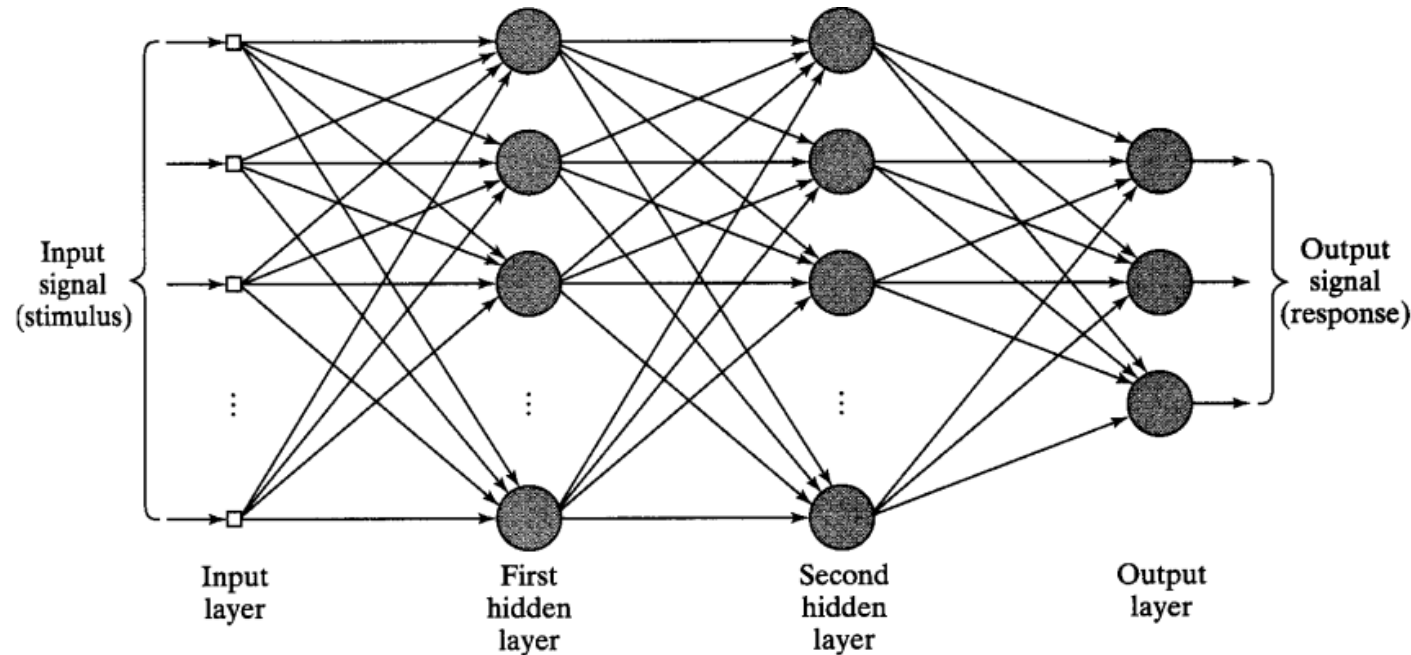
Sentido direto

- Cálculo ativações e saídas de todos os neurônios da camada oculta e de saída;

Entrada

Intermediária

Saída





Sentido direto

- Cálculo ativações e saídas de todos os neurônios da camada oculta e de saída.

$$z_i(t) = \phi_i(u_i(t)) = \phi_i \left(\sum_{j=0}^p w_{ij}(t) x_j(t) \right) = \phi_i (\mathbf{w}_i^T(t) \mathbf{x}(t))$$

Intermediária



Sentido direto

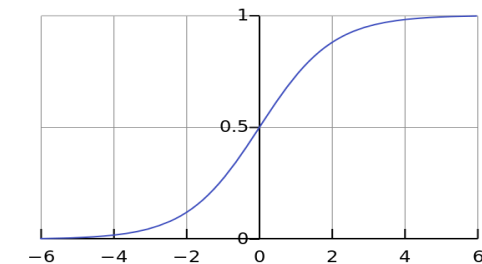
- Cálculo ativações e saídas de todos os neurônios da camada oculta e de saída;

$$z_i(t) = \phi_i(u_i(t)) = \phi_i \left(\sum_{j=0}^p w_{ij}(t)x_j(t) \right) = \phi_i (\mathbf{w}_i^T(t)\mathbf{x}(t))$$

Intermediária

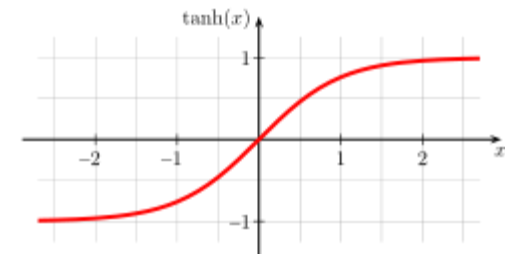
$$\phi_i(u_i(t)) = \frac{1}{1 + \exp[-u_i(t)]}$$

Função Logística



$$\phi_i(u_i(t)) = \frac{1 - \exp[-u_i(t)]}{1 + \exp[-u_i(t)]}$$

Tangente Hiperbólica





Sentido direto

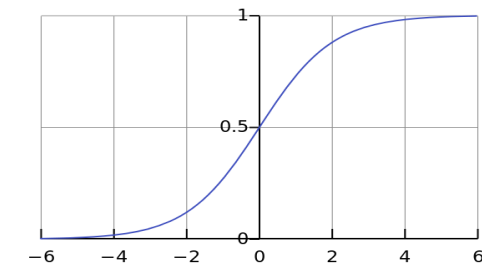
- Deve repetir para a camada de saída, considerando a saída da camada intermediária como entrada

$$y_k(t) = \phi_k(u_k(t)) = \phi_k \left(\sum_{i=0}^q m_{ki}(t) z_i(t) \right)$$

Saída

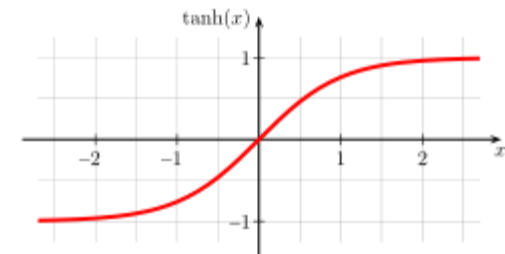
$$\phi_i(u_i(t)) = \frac{1}{1 + \exp[-u_i(t)]}$$

Função Logística



$$\phi_i(u_i(t)) = \frac{1 - \exp[-u_i(t)]}{1 + \exp[-u_i(t)]}$$

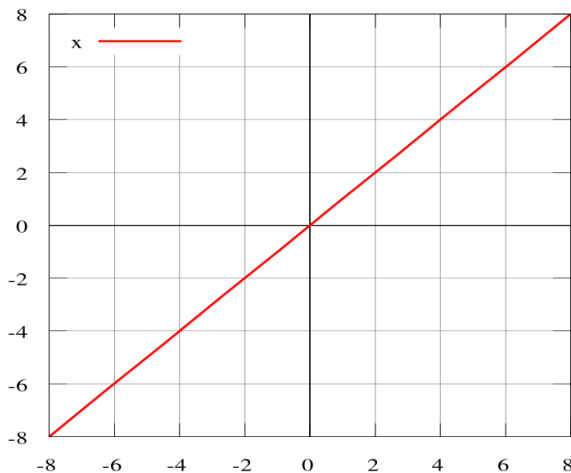
Tangente Hiperbólica





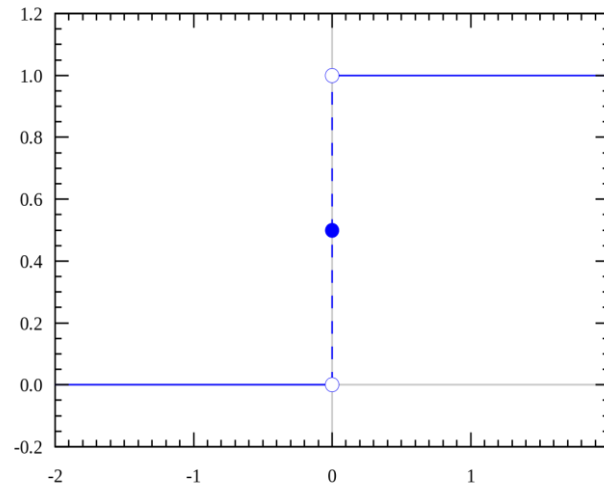
Funções de ativação

Função Identidade



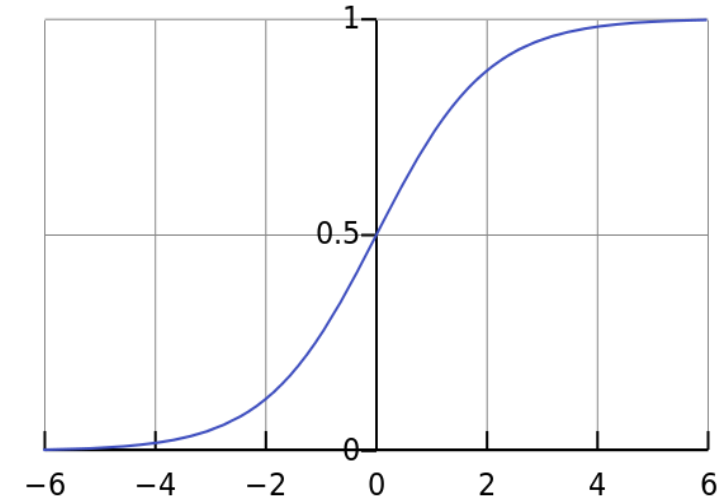
$$f(x) = x$$

Função Degrau



$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

Função Logística

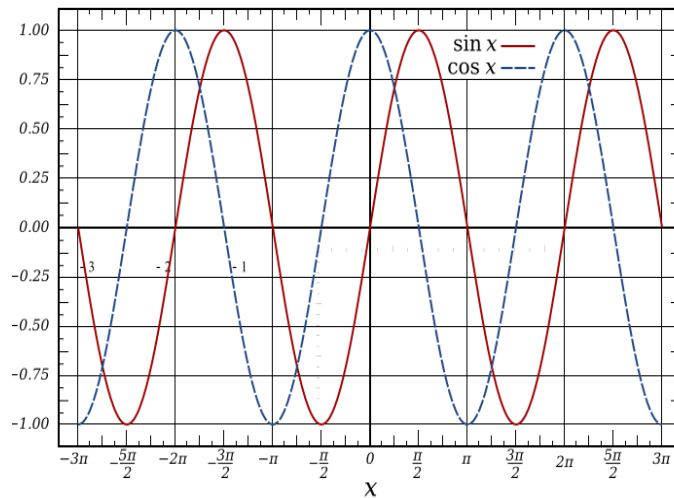


$$f(x) = \frac{1}{1 + e^{-x}}$$



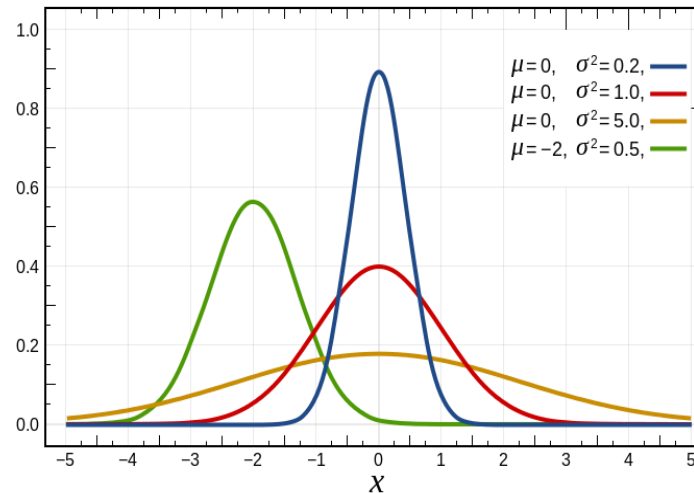
Funções de ativação

Função Senoide



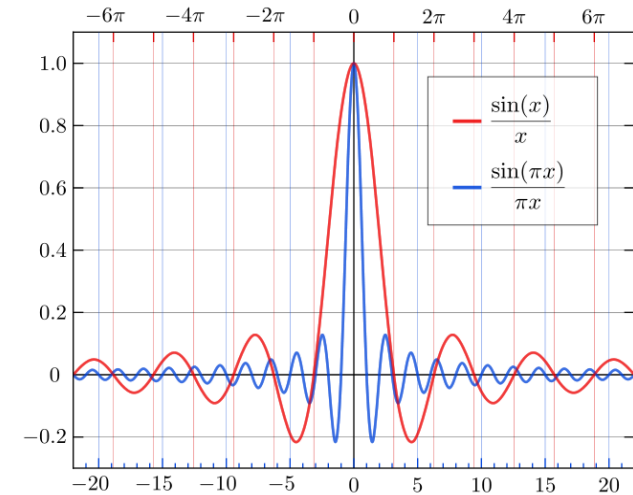
$$f(x) = \sin(x)$$

Função Gaussiana



$$f(x) = e^{-x^2}$$

Função Sinc



$$f(x) = \begin{cases} 1 & \text{for } x = 0 \\ \frac{\sin(x)}{x} & \text{for } x \neq 0 \end{cases}$$

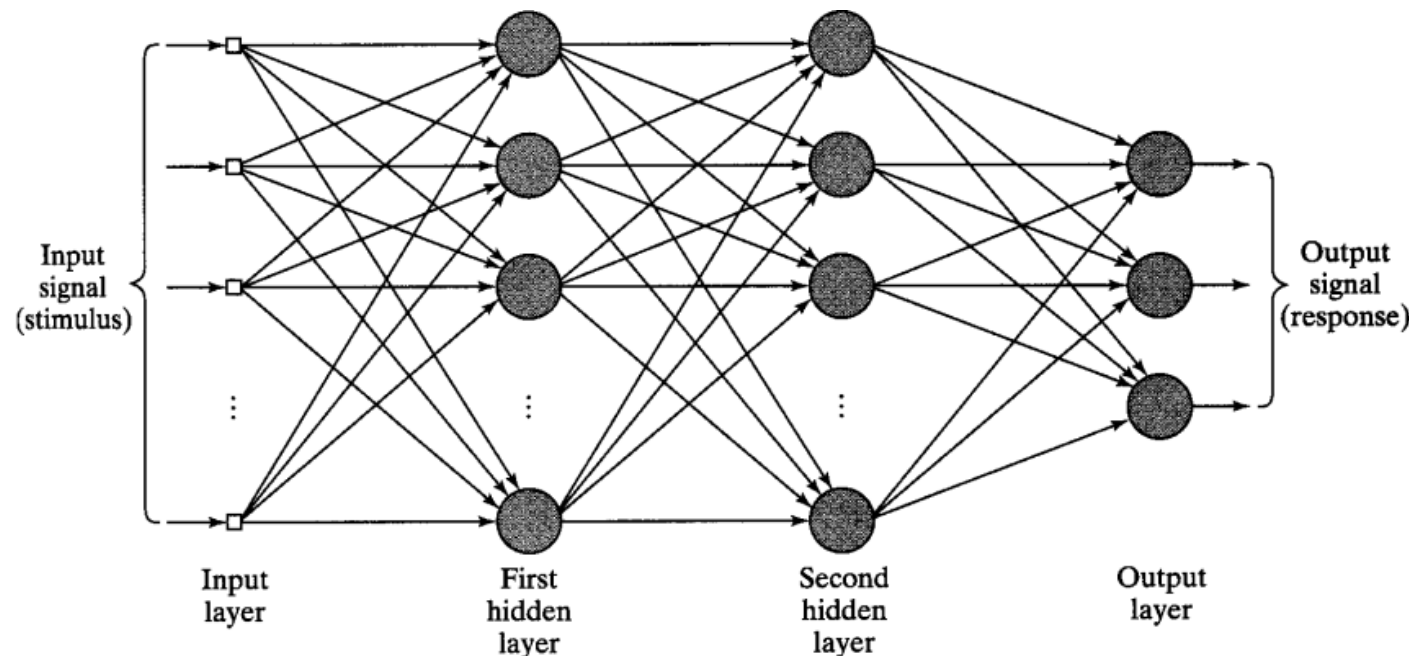


Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída.

Intermediária

Saída





Sentido inverso

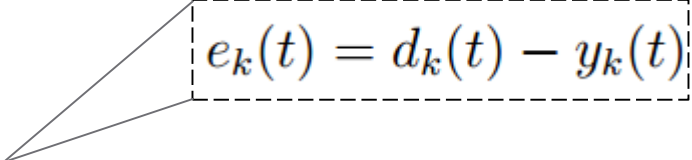
- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada saída**

$$\delta_k(t) = e_k(t)\phi'(u_k(t))$$



Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada saída**


$$e_k(t) = d_k(t) - y_k(t)$$

$$\delta_k(t) = e_k(t)\phi'(u_k(t))$$



Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada saída**

$$e_k(t) = d_k(t) - y_k(t)$$

$$\delta_k(t) = e_k(t)\phi'(u_k(t))$$

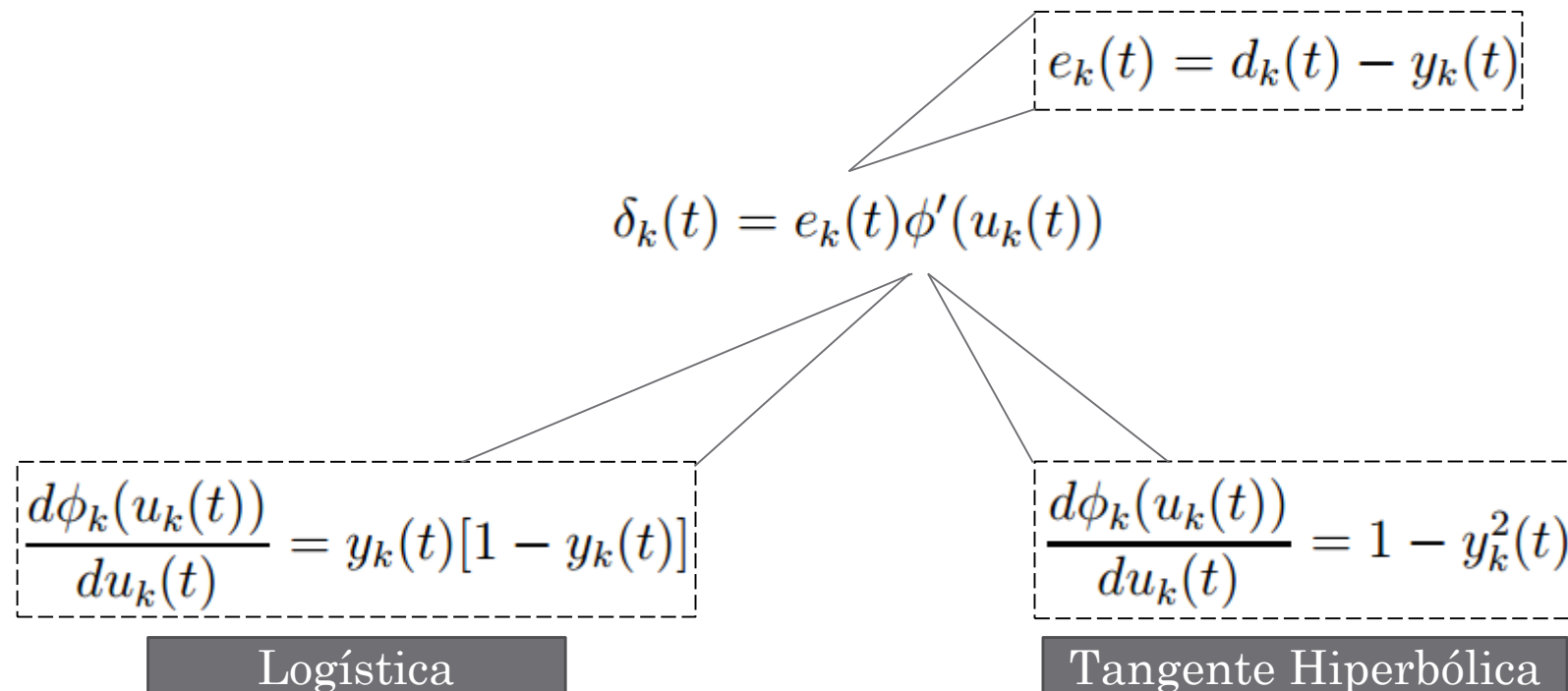
$$\frac{d\phi_k(u_k(t))}{du_k(t)} = y_k(t)[1 - y_k(t)]$$

Logística



Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada saída**





Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada intermediária**

$$\delta_i(t) = \phi'_i(u_i(t)) \sum_{k=1}^n m_{ki} \delta_k(t)$$



Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada intermediária**

$$\delta_i(t) = \phi'_i(u_i(t)) \sum_{k=1}^n m_{ki} \delta_k(t)$$

$$\frac{d\phi_i(u_i(t))}{du_i(t)} = y_i(t)[1 - y_i(t)]$$

Logística



Sentido inverso

- Cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada intermediária e saída
 - Gradientes locais **camada intermediária**

$$\delta_i(t) = \phi'_i(u_i(t)) \sum_{k=1}^n m_{ki} \delta_k(t)$$

$$\frac{d\phi_i(u_i(t))}{du_i(t)} = y_i(t)[1 - y_i(t)]$$

Logística

$$\frac{d\phi_i(u_i(t))}{du_i(t)} = \frac{1}{2}[1 - y_i^2(t)]$$

Tangente Hiperbólica



Sentido inverso

- Atualização/ajuste dos parâmetros da rede
 - Pesos sinápticos e limiares

$$\begin{aligned}w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij}(t) \\ &= w_{ij}(t) + \alpha \delta_i(t) x_j(t)\end{aligned}$$

Camada Intermediária

$$\begin{aligned}m_{ki}(t+1) &= m_{ki}(t) + \Delta m_{ki}(t) \\ &= m_{ki}(t) + \alpha \delta_k(t) z_i(t)\end{aligned}$$

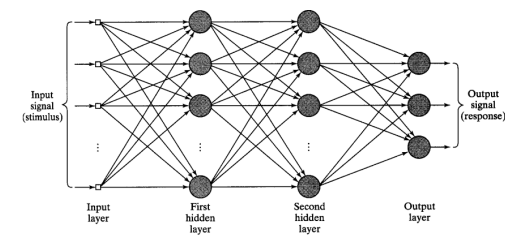
Camada Saída



Treinamento

- Pontos importantes
 - Vetor de entrada;
 - Vetor de saída;
 - Número neurônios camada oculta;
 - Funções de ativação;
 - Critério de parada e convergência
 - Geralmente erro médio quadrático por Época.
 - Avaliação da rede.

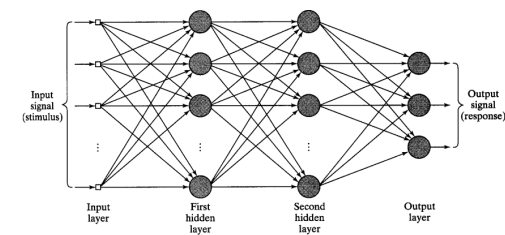
$$\varepsilon_{epoca} = \frac{1}{N} \sum_{t=1}^N \varepsilon(t) = \frac{1}{2N} \sum_{t=1}^N \sum_{k=1}^n e_k^2(t)$$





Multilayer Perceptron (MLP)

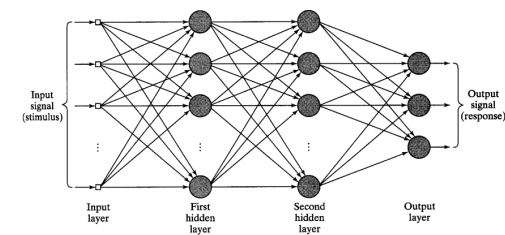
- **Hecht-nielsen** afirma que apenas com uma camada oculta já é possível calcular a função arbitrária qualquer (camada oculta deve ter $2i+1$ neurônios);
- **Cybenko** defende o uso de 2 camadas internas;
- **Kudricky** observa que, para cada 3 neurônios da 1ª camada oculta era preciso 1 da segunda camada;
- **Lippmann** afirma que a 2ª camada oculta deve ter o dobro de neurônios da camada de saída





Processos de aprendizado

- De 50 a 90% do total de dados devem ser usados para o treinamento da RNA
 - Escolhidos aleatoriamente, a fim de que a rede "aprenda" as regras e não decore exemplos.
- O restante dos dados é apresentado à RNA na fase de testes a fim de que ela possa deduzir corretamente o interrelacionamento entre os dados.





Referências

- ZUBEN, F. V. Rede MLP: Perceptron de Múltiplas Camadas
- CRUZ, A. Redes Neurais Artificiais: Multi Layer Perceptron
- ALMEIDA, R. R. IA Conexcionista: Redes Neurais Artificiais
- AFFONSO, E. T. F., SILVA, A. M., SILVA, M. P., RODRIGUES, T. M. D., MOITA, G. F. Uso Redes Neurais Multilayer Perceptron (Mlp) Em Sistema De Bloqueio De Websites Baseado Em Conteúdo
- CASTRO, F. C., CASTRO, M. C. Multilayer Perceptrons
- CASTROUNIS, A. Artificial Intelligence, Deep Learning, and Neural Networks Explained
- ANDRÉ, A. P. L. F. Redes Neurais Artificiais
- TATIBANA, C. Y., KAETSU, D. Y. Redes Neurais
- ZADROZNY, B. Métodos Estatísticos de Aprendizagem
- BACKES, A. Redes Neurais
- BARRETO, G. A. Perceptron Multicamadas e o Algoritmo de Retropropagação do Erro