

---

---

# Redes Neurais Artificiais

---

---

André Ricardo Gonçalves

`andreric [at] dca.fee.unicamp.br`

`www.dca.fee.unicamp.br/~andreric`

# Sumário

<b>1</b>	<b>Redes Neurais Artificiais</b>	p. 3
1.1	Inspiração Biológica . . . . .	p. 3
1.2	Modelo Matemático do Neurônio Biológico . . . . .	p. 4
1.3	Redes Neurais Artificiais . . . . .	p. 6
1.3.1	Topologias das Redes Neurais . . . . .	p. 7
1.3.2	Processo de Aprendizagem . . . . .	p. 9
1.3.3	Redes Multilayer Perceptron . . . . .	p. 10
1.3.4	Dificuldades de implementação da rede MLP . . . . .	p. 12
1.4	Outros Modelos . . . . .	p. 14
1.4.1	Redes de Função de Base Radial . . . . .	p. 14
1.4.2	Mapas Auto-Organizáveis . . . . .	p. 14
1.4.3	Rede de Hopfield . . . . .	p. 17
1.5	Aplicações . . . . .	p. 18
1.6	Conclusão . . . . .	p. 19
	<b>Referências</b>	p. 21

# 1 Redes Neurais Artificiais

O cérebro humano é uma máquina altamente poderosa e complexa capaz de processar uma grande quantidade de informações em tempo mínimo. As unidades principais do cérebro são os neurônios e é por meio deles que as informações são transmitidas e processadas.

As tarefas realizadas pelo cérebro intrigam os pesquisadores, como por exemplo, o cérebro é apto a reconhecer um rosto familiar dentre uma multidão, em tempo da ordem de milésimos de segundo. As respostas sobre alguns enigmas do funcionamento do cérebro se perpetuam até os dias de hoje.

O que é conhecido sobre o funcionamento do cérebro é que o mesmo desenvolve suas regras através da “experiência” adquirida em situações vividas anteriormente. Segundo (HAYKIN, 1999) o desenvolvimento do cérebro humano dá-se principalmente nos dois primeiros anos de vida, mas se arrasta por toda a vida.

Inspirando-se neste modelo, diversos pesquisadores tentaram simular o funcionamento do cérebro, principalmente o processo de aprendizagem por experiência, a fim de criar sistemas inteligentes capazes de realizar tarefas como classificação, reconhecimento de padrões, processamento de imagens entre outras.

Como resultados destas pesquisas surgiram o modelo do neurônio artificial e posteriormente um sistema com vários neurônios interconectados, a chamada Rede Neural.

## 1.1 Inspiração Biológica

Como dito anteriormente, o neurônio é a unidade básica do cérebro humano. Uma célula especializada na transmissão de informações, pois nelas estão introduzidas propriedades de excitabilidade e condução de mensagens nervosas.

De acordo com (ARBIB, 2002) o neurônio é constituído por três partes principais: a *soma* ou corpo celular, do qual emanam algumas ramificações denominadas de *dendritos*, e por fim outra ramificação descendente da soma, porém mais extensa chamada de *axônio*. Nas extremidades dos axônios estão os nervos terminais, que pelos quais é realizada a transmissão das informações para outros neurônios, esta transmissão é conhecida como *sinapse*.

Arbib (2002) observa que a *soma* e os dendritos formam a superfície de entrada do neurônio e o axônio a superfície de saída do fluxo de informação. A figura 1 identifica as partes principais do neurônio e as setas mostram o fluxo da informação

A informação transmitida pelos neurônios na realidade são impulsos elétricos,

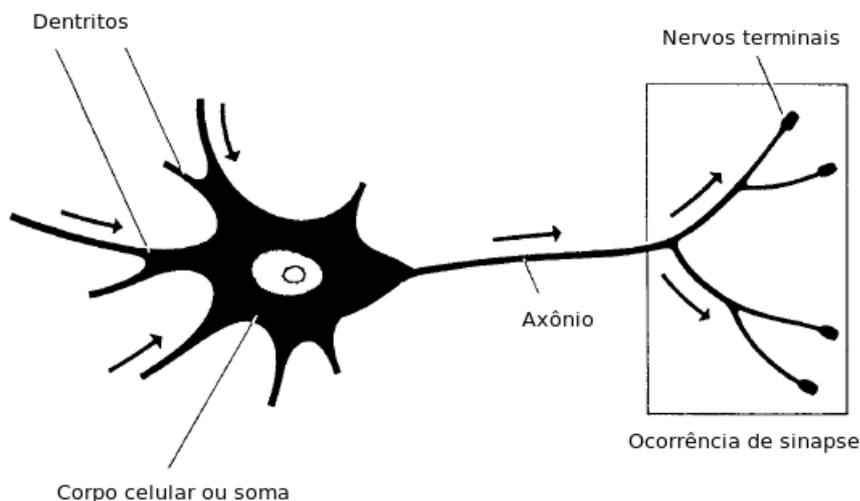


Figura 1: Modelo abstrato do neurônio biológico. [Adaptado de (ARBIB, 2002)]

que segundo (REIS, 2008) o impulso elétrico é a mensagem que os neurônios transmitem uns aos outros, ou seja, é a propagação de um estímulo ao longo dos neurônios que pode ser qualquer sinal captado pelos receptores nervosos. O estímulo resulta na capacidade dos neurônios se excitarem através da ação de um estímulo.

## 1.2 Modelo Matemático do Neurônio Biológico

A partir da estrutura e funcionamento do neurônio biológico, pesquisadores tentaram simular este sistema em computador.

O modelo mais bem aceito foi proposto por McCulloch e Pitts (1943), conhecido como *Perceptron*, o qual implementa de maneira simplificada os componentes e o funcionamento de um neurônio biológico.

Neste modelo, os impulsos elétricos provenientes de outros neurônios são representados pelos chamados *sinais de entrada*,  $(x_j)$ , dentre os vários estímulos recebidos, alguns excitarão mais e outros menos o neurônio receptor, essa medida de quão excitatório é o estímulo é representada no modelo de McCulloch e Pitts através dos *pesos sinápticos*, quanto maior o valor do peso, mais excitatório é o estímulo. Os pesos sinápticos são representados por  $w_{kj}$ , onde  $k$  representa o índice do neurônio em questão e  $j$  se refere ao terminal de entrada da sinapse à qual o peso sináptico se refere.

A *soma* é representada por uma composição de dois módulos, o primeiro é uma *junção aditiva*, somatório dos estímulos (sinais de entrada) multiplicado pelo seu fator excitatório (pesos sinápticos), e posteriormente uma *função de ativação*, que definirá com base nas entradas e pesos sinápticos, qual será a saída do neurônio. O axônio é aqui representado pela saída  $(y_k)$  obtida pela aplicação da função de ativação.

Assim como no modelo biológico, o estímulo pode ser excitatório ou inibitório, representado pelo peso sináptico positivo ou negativo respectivamente. A figura 2

apresenta o modelo de neurônio artificial proposto por McCulloch e Pitts (1943).

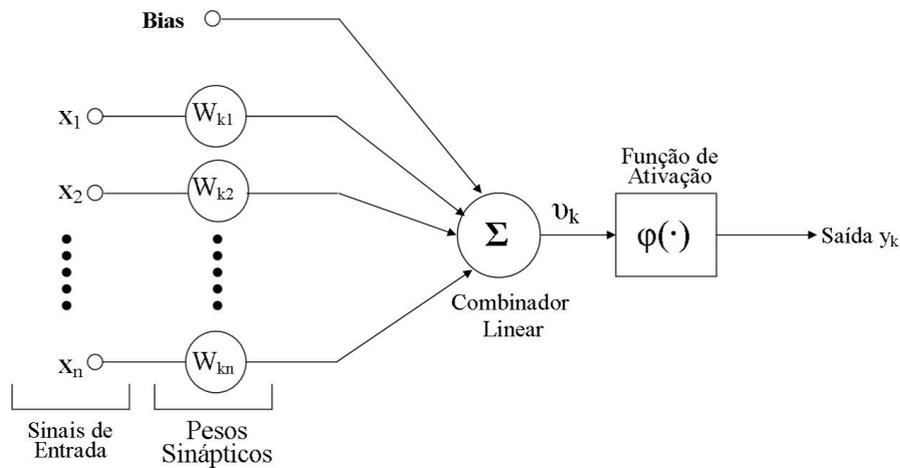


Figura 2: Modelo de neurônio artificial proposto por McCulloch e Pitts

O modelo proposto possui uma natureza binária, tanto os sinais de entrada quanto a saída são valores binários. McCulloch acreditava que o funcionamento do sistema nervoso central possuía um caráter binário, ou seja, um neurônio influencia ou não outro neurônio, mas posteriormente mostrou-se que não era dessa forma. Esta característica do modelo de McCulloch e Pitts foi referenciada como propriedade do “tudo ou nada” (HAYKIN, 1999).

Em termos matemáticos o neurônio artificial  $i$  pode ser representado como:

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (1.1)$$

onde  $m$  é o número de sinais de entrada incidentes no neurônio  $k$  e posteriormente a aplicação da função de ativação

$$y_k = \varphi(u_k) \quad (1.2)$$

Basicamente a função de ativação recebe o valor provido pela junção aditiva, denominado por (HAYKIN, 1999) como “campo local induzido”, atribuindo 1, se o campo local induzido daquele neurônio for não-negativo e 0 caso contrário. A função de ativação adotada no modelo é uma função degrau definida como segue:

$$\varphi(u) = \begin{cases} 1 & , \text{ se } u \geq 0 \\ 0 & , \text{ se } u < 0 \end{cases} \quad (1.3)$$

Outras funções de ativação foram apresentadas na literatura, dentre elas a função *sigmóide*, a qual segundo (HAYKIN, 1999) é de longe a forma mais utilizada na construção de redes neurais artificiais, tem um comportamento estritamente crescente que exhibe um balanceamento entre comportamento linear e não-linear.

Um exemplo de função sigmóide é a *função logística* definida por

$$\varphi(u) = \frac{1}{1 + \exp(-au)} \quad (1.4)$$

onde  $a$  é o parâmetro de inclinação da função sigmóide. Outro fator relevante desta função, como observado por (HAYKIN, 1999) é que esta função é diferenciável, que é uma característica importante da teoria das redes neurais.

McCulloch e Pitts mostraram que o *Perceptron* era capaz de computar funções lógicas e aritméticas, como funções lógicas do tipo “or” e “and” (BASHEER; HAJMEER, 2000).

Rosenblatt (1958) propôs o Perceptron como o primeiro modelo para aprendizagem supervisionada, chamada de Rede Perceptron ou Perceptron de Camada Única. Esta é a forma mais simples de rede neural, usada para classificação de padrões linearmente separáveis (HAYKIN, 1999).

## 1.3 Redes Neurais Artificiais

Uma rede neural típica é constituída por um conjunto de neurônios interligados, influenciando uns aos outros formando um sistema maior capaz de armazenar conhecimento adquirido por meio de exemplos apresentados e assim podendo realizar inferências sobre novos exemplos (novas situações) desconhecidos.

As redes neurais são comumente apresentadas como um grafo orientado, onde os vértices são os neurônios e as arestas as sinapses, a direção das arestas informa o tipo de alimentação, ou seja, como os neurônios são alimentados (recebem sinais de entrada), Alimentação de redes neurais é discutida na seção 1.3.1. Um exemplo de uma rede neural como um grafo orientado é mostrado na figura 3.

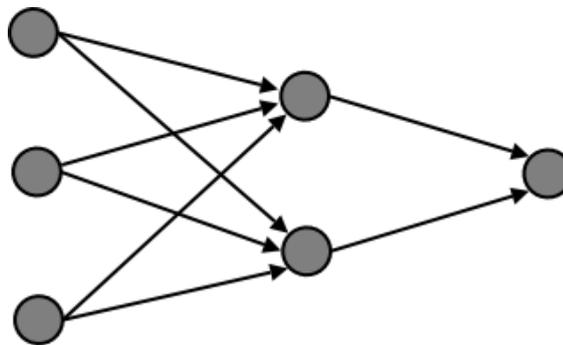


Figura 3: Rede Neural representada como um grafo orientado

De acordo com (ACHARYA et al., 2003) as redes neurais derivam seu poder devido a sua estrutura massiva e paralela e a habilidade de aprender por experiência. Essa experiência é transmitida por meio de exemplos obtidos do mundo real, definidos como um conjunto de características que levam a uma determinada situação. Se a situação gerada pela combinação de características for informada a rede, a aprendizagem é dita ser supervisionada e não-supervisionada caso contrário.

O conhecimento obtido pela rede através dos exemplos é armazenado na forma de pesos das conexões, os quais serão ajustados a fim de tomar decisões corretas a partir de novas entradas, ou seja, novas situações do mundo real não conhecidas pela rede.

O processo de ajuste dos pesos sinápticos é realizado pelo algoritmo de aprendizagem, responsável em armazenar na rede o conhecimento do mundo real obtido através de exemplos. Na literatura são relatados vários algoritmos de aprendizagem, dentre eles o *backpropagation* que é o algoritmo mais utilizado (BASHEER; HAJMEER, 2000).

Acharya et al. (2003) identifica três decisões importantes no processo de construção de uma rede neural: (i) a topologia da rede, a topologia de redes neurais serão discutidas na seção 1.3.1; (ii) algoritmo de aprendizagem e (iii) função de ativação.

### 1.3.1 Topologias das Redes Neurais

A Topologia de uma rede neural diz respeito à disposição dos neurônios na rede, como são estruturados. A topologia da rede está diretamente ligada ao tipo de algoritmo de aprendizagem utilizado.

Em geral é possível identificar três classes de topologias de rede:

#### 1. Redes alimentadas adiante (*Feed-forward networks*)

Alguns tipos de redes são estruturados em forma de *camadas*, os neurônios são dispostos em conjuntos distintos e ordenados seqüencialmente, conjuntos esses denominados de *camadas*. Nas redes alimentadas adiante, o fluxo de informação é sempre da camada de entrada para a camada de saída. Haykin (1999) distingue as redes alimentadas adiante em Redes de Camada Única e Redes de Múltiplas Camadas, apenas diferenciando do número de camadas, mas o conceito de alimentação adiante ou direta é o mesmo. A figura 4 apresenta um modelo de rede alimentada adiante.

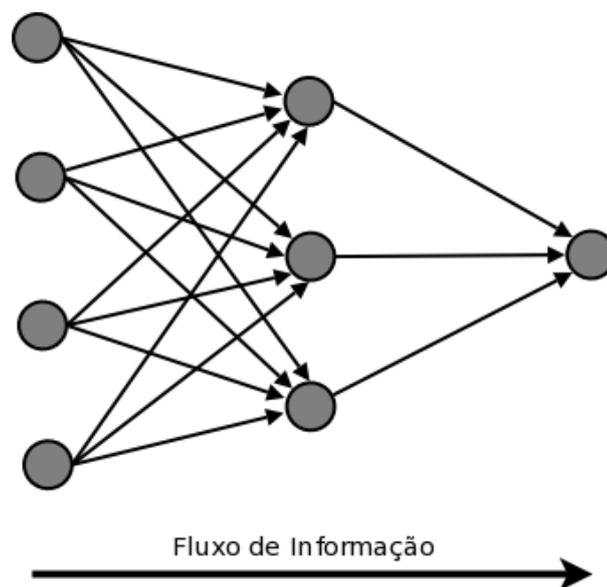


Figura 4: Rede neural alimentada adiante

Algumas características importantes de uma rede alimentada adiante são apresentadas abaixo:

- (a) Os neurônios são arranjados em camadas, a camada inicial recebe os sinais de entrada e a camada final obtém as saídas. As camadas intermediárias são chamadas de *camadas ocultas*.
- (b) Cada neurônio de uma camada é conectado com todos os neurônios da camada seguinte.
- (c) Não há conexões entre neurônios de uma mesma camada.

A informação (alimento) provém da camada de entrada ou neurônios (nós) fonte e posteriormente é transmitida para as camadas seguintes até a camada de saída. Este tipo de rede é também conhecido como *rede acíclica*, pois na representação de grafos não possui ciclos.

## 2. Redes Recorrentes (*Feed-backward networks*)

Nas redes recorrentes há a ocorrência de *realimentação*, na qual a saída de um neurônio é aplicada como entrada no próprio neurônio e/ou em outros neurônios de camadas anteriores, ou seja, há a ocorrência de um ciclo no grafo. A figura 5 mostra uma rede neural, na qual alguns neurônios são realimentados.

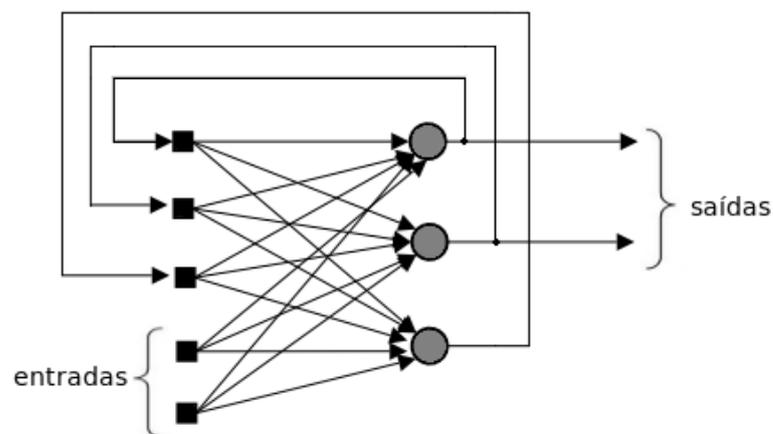


Figura 5: Exemplo de uma rede neural com realimentação

## 3. Redes Competitivas

Nesta classe de redes neurais os neurônios estão divididos em duas camadas, a camada de entrada ou nós fontes e a camada de saída, conhecida como "*grade*". Os neurônios da grade são forçados a competir entre si, com base no nível de similaridade entre o padrão de entrada e a grade de neurônios e somente o neurônio vencedor será disparado (ativado) a cada iteração (BASHEER; HAJMEER, 2000). Redes desta classe utilizam algoritmo de aprendizagem competitivo. A rede mais conhecida desta classe é a rede de Kohonen, também conhecida como Mapa Auto-Organizável.

## 1.3.2 Processo de Aprendizagem

A principal característica de uma rede neural é a sua capacidade de aprender a partir de um ambiente e de melhorar seu desempenho por meio da aprendizagem. Esta aprendizagem, como observa (BASHEER; HAJMEER, 2000), é um processo de atualização da representação interna do sistema em resposta a um estímulo externo, podendo desempenhar uma tarefa específica.

A atualização da representação é realizada sob a forma de modificação da arquitetura, ajuste dos pesos das conexões entre os neurônios e ativando regras de neurônios individuais.

As regras de aprendizagem definem como a rede deve ajustar os pesos sinápticos. Haykin (1999) identifica quatro tipos de regras de aprendizagem:

### 1. Aprendizagem por Correção de Erro

Utilizado em treinamento supervisionado, esta técnica ajusta os pesos sinápticos por meio do erro, que é obtido através da diferença entre o valor de saída da rede e o valor esperado em um ciclo de treinamento. Com isso gradualmente vai diminuindo o erro geral da rede.

### 2. Aprendizagem Hebbiana

Baseado no postulado de aprendizagem de Hebb (HEBB, 1949), que afirma: “se dois neurônios em ambos os lados de uma sinapse são ativados sincronamente e simultaneamente, então a força daquela sinapse é seletivamente aumentada”. Este processo de treinamento é feito localmente, ajustando o peso das conexões baseado nas atividades dos neurônios.

### 3. Aprendizagem de Boltzmann

Um método de aprendizagem estocástico derivado das idéias enraizadas na mecânica estatística (HAYKIN, 1999). Uma rede que implementa o método de aprendizagem de Boltzmann é dita ser uma **MB!** (**MB!**). Neste modelo os neurônios são estocásticos, podendo residir em dois estados possíveis, ligado (+1) e desligado (-1), e ainda são divididos em dois grupos funcionais, *presos* e *livres*, sendo responsáveis pela interação com o ambiente e pela explicação das restrições subjacentes dos padrões de entrada do ambiente, respectivamente. Um ponto importante na MB é que seus neurônios possuem conexões bidirecionais (HINTON; ACKLEY; SEJNOWSKI, 1985). De acordo com (HAYKIN, 1999) este modelo pode ser visto como um procedimento de aprendizagem não-supervisionado para modelar uma distribuição de probabilidade, especificada pelos padrões presos aos neurônios visíveis com probabilidades apropriadas.

### 4. Aprendizagem Competitiva

Neste modelo de aprendizagem os neurônios são forçados a competir entre si e somente um será ativo, em uma dada iteração, o vencedor, ou seja, o que tiver maior similaridade com o padrão de entrada. Todos os pesos dos neurônios próximos ao neurônio vencedor terão seus valores ajustados.

As regras que definem como serão atualizados os pesos sinápticos constituem o algoritmo de aprendizagem. O algoritmo mais utilizado na literatura é o algoritmo *backpropagation*, baseado na propagação do erro.

### 1.3.3 Redes Multilayer Perceptron

Redes **MLP!** (**MLP!**) são redes alimentadas adiante que possuem uma ou mais camadas de neurônios entre as camadas de entrada e saída, chamada de *camada oculta* (LIPPMANN, 1987). Esta camada adiciona um poder maior em relação às redes *Perceptron* de camada única, que classifica apenas padrões linearmente separáveis, sendo os neurônios ocultos responsáveis por capturar a não-linearidade dos dados.

Neste modelo todos os neurônios são ligados aos neurônios da camada subsequente, não havendo ligação com os neurônios laterais (da mesma camada) e também não ocorre realimentação.

A aprendizagem de uma rede neural MLP é um processo iterativo, conhecido como aprendizagem por experiência, no qual padrões de treinamento (exemplos) são apresentados a rede e com base nos erros obtidos, são realizados ajustes nos pesos sinápticos, com o intuito de diminuir o erro nas próximas iterações.

O principal algoritmo de treinamento é o algoritmo de retropropagação de erro (*error backpropagation*), baseado na regra de aprendizagem por correção de erro, que consiste basicamente de dois passos, um para frente e outro para trás.

O passo para frente é chamado de propagação, os valores provindos dos neurônios de entrada (nós fontes) são aplicados aos neurônios ocultos e posteriormente suas saídas são aplicadas como entradas aos neurônios da camada final, obtendo a resposta da rede. Durante este passo os pesos sinápticos da rede são todos fixos (HAYKIN, 1999).

Já o passo para trás é incumbido de ajustar os pesos sinápticos, por meio do cálculo do erro realizado na camada de saída, os pesos sinápticos entre as camadas antecessoras são ajustados de acordo com uma regra de correção de erro. A figura 6 mostra uma rede MLP.

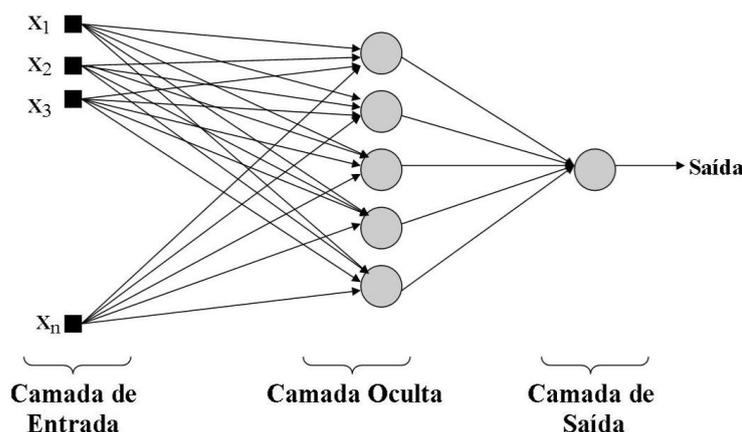


Figura 6: Arquitetura da rede MLP

O algoritmo *backpropagation* utiliza a técnica de busca por gradiente para minimizar a função custo. O algoritmo 1 mostra as etapas do *backpropagation*, baseado em (LIPPMANN, 1987).

As etapas do *backpropagation* são melhores discutidas e detalhadas abaixo.

**Algoritmo 1:** Algoritmo de treinamento *backpropagation*


---

```

1 begin
2   Atribuição de valores iniciais aos pesos sinápticos;
3   repeat
4     Apresentação à rede dos padrões de entrada e as saídas desejadas;
5     Cálculo dos valores de saída dos neurônios ocultos;
6     Cálculo dos valores de saída dos neurônios de saída (resposta real da rede);
7     Cálculo do erro (diferença entre resposta da rede e valor esperado);
8     Ajuste dos pesos sinápticos;
9   until Condição de parada não satisfeita ;
10 end

```

---

**Linha 2** O processo de atribuição de valores iniciais dos pesos sinápticos é comumente realizado de maneira aleatória, atribuindo pequenos valores, geralmente no intervalo  $\{0,1\}$ ;

**Linha 4** Sendo o treinamento supervisionado é necessária a apresentação dos padrões de entrada juntamente com a saída conhecida previamente;

**Linhas 5 e 6** O cálculo dos valores de saída são realizados pela aplicação do campo local induzido ( $v$ ) (HAYKIN, 1999) à uma função de ativação. De uma maneira matemática, o campo local induzido para o neurônio  $j$  na camada  $l$  na iteração  $n$  é obtido por

$$v_j^{(l)}(n) = \sum_{i=1}^{r+1} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \quad (1.5)$$

onde  $y_i^{(l-1)}(n)$  é o sinal de saída do neurônio  $i$  na camada anterior  $l - 1$ , na iteração  $n$ ,  $w_{ji}^{(l)}$  é o peso sináptico do neurônio  $j$  da camada  $l$ , que é alimentado pelo neurônio  $i$  da camada  $l - 1$  e  $r$  o número de neurônios na camada anterior ( $l - 1$ ). Observe que ao invés de  $r$  é utilizado  $r + 1$ , isto se deve ao fato da inclusão de um *bias*, um neurônio adicional com valor  $+1$ . O *bias* equivale a dizer:  $y_{r+1}^{(l-1)}(n) = +1$ .

Denotaremos por  $\varphi_j(v_j)$ , sendo  $\varphi(\cdot)$  a função de ativação e  $v_j$  o campo local induzido do neurônio  $j$ , o valor de saída de um neurônio  $j$  da camada  $l$ . Assim o valor de saída do neurônio

$$y_j^{(l)} = \varphi(v_j(n)) \quad (1.6)$$

Valores de saída estarão dentro do intervalo definido pela função de ativação. As funções de ativação geralmente utilizadas podem ser encontradas em (HAYKIN, 1999)

**Linha 7** O sinal de erro da saída da rede, na iteração  $n$  é calculada por  $e_j(n) = d_j(n) - y_j(n)$ , onde  $d_j$  é a  $j$ -ésima resposta desejada e  $y_j$  é a  $j$ -ésima resposta da rede;

**Linha 8** O ajuste dos pesos é o núcleo deste algoritmo, sendo realizado da camada oculta para a camada de entrada. Em qualquer camada  $l$  os valores dos pesos  $w_{ji}^{(l)}$  na iteração  $n$  serão ajustados da iteração anterior ( $n - 1$ ) de acordo com:

$$w_{ji}^{(l)}(n) = w_{ji}^{(l)}(n-1) + \Delta w_{ji}^{(l)}(n) \quad (1.7)$$

onde  $\Delta w_{ji}^{(l)}(n)$  é correção aplicada ao peso da conexão. A correção é determinada pela *regra delta modificada*, definida como segue:

$$\Delta w_{ji}^{(l)}(n+1) = \eta \delta_j^{(l)} y_i^{(l-1)} + \mu \Delta w_{ji}^{(l)}(n) \quad (1.8)$$

onde  $\eta$  é a taxa de aprendizagem,  $\delta_j^{(l)}$  é o gradiente local,  $\mu$  é a constante de momento e  $y_i^{(l-1)}$  é o sinal de saída do neurônio  $i$  na camada anterior  $l-1$ .

A taxa de aprendizagem define o tamanho do passo de atualização e a constante de momento é utilizado para que o método possa fugir do mínimo local na superfície de erro, objetivando o mínimo global.

O gradiente local é calculado de maneira distinta entre neurônios de saída e ocultos. Sendo  $L$  a camada de saída, o gradiente do neurônio  $j$ , na iteração  $n$  da camada de saída é calculado por:

$$\delta_j^{(L)}(n) = e_j^{(L)}(n) \varphi'(v_j^{(L)}) \quad (1.9)$$

onde  $\varphi'(\cdot)$  é derivada da função de ativação. Para os neurônios ocultos a correção é obtida por

$$\delta_j^{(l)}(n) = \varphi'(v_j^{(l)}) \left( \sum_{i=1}^{r+1} \delta_i^{(l+1)} w_{ij}^{(l+1)} \right) \quad (1.10)$$

sendo  $l$  uma camada oculta qualquer.

**Linha 9** Basheer e Hajmeer (2000) identificam alguns critérios de parada, dentre eles (i) erro de treinamento ( $e \leq \epsilon$ ), (ii) gradiente do erro menor que um  $\sigma$  ou (iii) utilizando técnica de *validação cruzada*.

O critério de parada (iii) é geralmente utilizado através da análise gráfica do comportamento do erro, com base na técnica de validação cruzada (KOHAVI, 1995) e (AMARI et al., 1997).

### 1.3.4 Dificuldades de implementação da rede MLP

Para que uma rede *Multilayer Perceptron* possa obter bons resultados na aplicação em problemas reais, uma boa configuração da mesma deve ser feita. Esta configuração é realizada de maneira distinta dependendo de várias características do problema, como o tipo dos dados de entrada (inteiro, *booleanos*, híbrido), número de padrões disponíveis para treinamento e teste, a dimensionalidade dos dados de entrada, entre outros. Para isso, valores ideais dos vários parâmetros da rede MLP devem ser utilizados, porém estes valores não são de fácil obtenção.

De acordo com (BASHEER; HAJMEER, 2000) alguns parâmetros são determinados por tentativa e erro, ou seja, são atribuídos vários valores distintos aos parâmetros e

analisando os resultados obtidos, a melhor configuração é escolhida. Dentre esses parâmetros, a taxa de aprendizagem, a constante de momento, o número de camadas ocultas e o número de neurônios nas camadas ocultas, possuem um maior grau de dificuldade para o ajuste dos valores.

Algumas soluções para contornar este problema foram sugeridas, como a aplicação de outras técnicas de aprendizagem de máquina, as quais são conhecidas pela alta aplicabilidade em problemas de otimização, como Algoritmos Genéticos (HAN; MAY, 1996), Otimização por Enxame de Partículas (ZHANG; SHAO; LI, 2000) e Otimização por Colônias de Formigas (GONÇALVES; CAMARGO-BRUNETO, 2008b).

Outra dificuldade é a determinação do número ideal de ciclos de treinamento da rede, que de acordo com (BASHEER; HAJMEER, 2000) é determinado por tentativa e erro. Se um número muito grande de ciclos de treinamento for aplicado, a rede entra em um processo de "memorização" dos padrões a ela apresentados, chamado de super-treinamento (*overtraining*), perdendo assim a capacidade de generalização. E se um número muito pequeno for aplicado, a rede torna-se incapaz de representar os dados. Este fenômeno é mostrado pela figura 7.

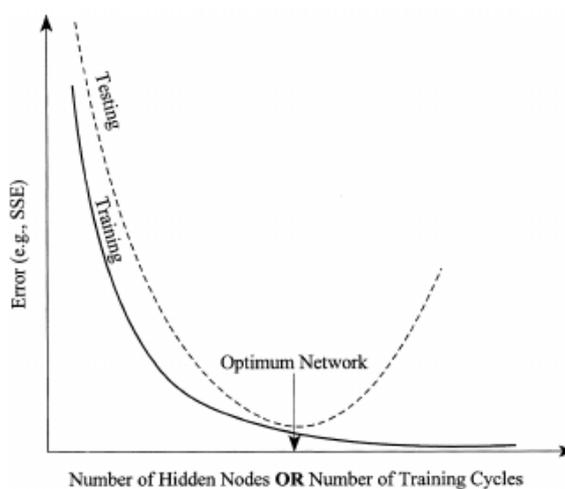


Figura 7: Número de ciclos de treinamento ótimo. [Fonte: (BASHEER; HAJMEER, 2000)]

O super-treinamento é identificado quando o erro de teste, obtido pela validação cruzada, começa a aumentar depois de ter diminuído. Para evitar o super-treinamento a aprendizagem deve ser interrompida quando os erros não variam mais, indicando que a rede está treinada, ou quando o erro de teste aumentar, depois de ter diminuído. A técnica de visualização gráfica dos erros obtidos pela utilização da técnica de validação cruzada do tipo *Holdout* é comumente utilizada (KOHAVI, 1995) e (AMARI et al., 1997). Porém nesta abordagem há a necessidade de executar um número grande de ciclos de treinamento antes da geração do gráfico. Uma maneira de automatizar este procedimento de identificação gráfica utilizando regressão linear foi proposta por (GONÇALVES; CAMARGO-BRUNETO, 2008a).

## 1.4 Outros Modelos

Diferentes modelos de redes neurais foram propostos na literatura, com diferentes algoritmos de treinamento, de topologias, de paradigmas de aprendizagem e com aplicações diversas. As redes popularmente utilizadas são: Redes de Função de Base Radial, a Rede de Kohonen ou Mapas Auto-Organizáveis e as Redes de Hopfield, que são apresentadas a seguir.

### 1.4.1 Redes de Função de Base Radial

Diferentemente das Redes Perceptron Multicamadas, as redes de Função de Base Radial, do inglês **RBF!** (**RBF!**), trabalham o projeto de uma rede neural como um problema de ajuste de curvas (aproximação) em um espaço de alta dimensionalidade (HAYKIN, 1999).

Como observado por (HAYKIN, 1999), a aprendizagem de uma rede **RBF!** é um processo de busca em uma superfície multidimensional, que forneça melhores ajustes para os dados de treinamento, sendo os “melhores ajustes” uma medida no sentido estocástico e a generalização equivale ao uso da superfície multidimensional para interpolação dos dados de teste.

As redes **RBF!** diferem das redes **MLP!** por três características principais, como observa (VON ZUBEN; ATTUX, 2008):

1. Sempre apresenta uma única camada intermediária (oculta);
2. Neurônios da saída são sempre lineares;
3. Os neurônios da camada intermediária têm funções de base radial como função de ativação, ao invés de funções sigmóides ou outras.

Uma rede **RBF!** em sua forma mais simples é uma rede constituída por três camadas, com papéis distintos. A camada inicial ou nós fonte são encarregados de prover informações do ambiente (dados de entrada), a segunda e única camada oculta da rede é responsável pela transformação não-linear do espaço de entrada para o espaço oculto, sendo o último um espaço de alta dimensionalidade (na maioria das aplicações). E por fim a camada de saída, que é linear, fornece uma resposta ao estímulo gerado pela aplicação de um padrão (dados de entrada), pela camada de entrada.

As redes **RBF!** podem ser aplicadas principalmente em classificação de padrões e aproximação de função (GUPTA; JIN; HOMMA, 2003).

### 1.4.2 Mapas Auto-Organizáveis

Os Mapas Auto-Organizáveis, do inglês **SOM!** (**SOM!**), constituem outra classe de redes neurais, as redes com aprendizagem não-supervisionada. Esta rede é também conhecida como um tipo de Rede de Kohonen, devido ao fato deste modelo ter sido proposto por Kohonen (1982).

A base biológica do **SOM!** está na forma que os neurônios se organizam, muitas vezes refletem algumas características físicas sentidas pelos estímulos externos (LIPPMANN, 1987). Ou seja, o cérebro humano é organizado em várias áreas, de modo que entradas sensoriais diferentes são representadas por mapas computacionais ordenados topologicamente (HAYKIN, 1999).

Utilizando este conhecimento biológico, a rede de Kohonen utiliza-se de duas camadas, a camada de entrada e a grade pós-sináptica ou mapa de características, sendo a última um arranjo bi-dimensional de neurônios. A figura 8 mostra um mapa auto-organizável.

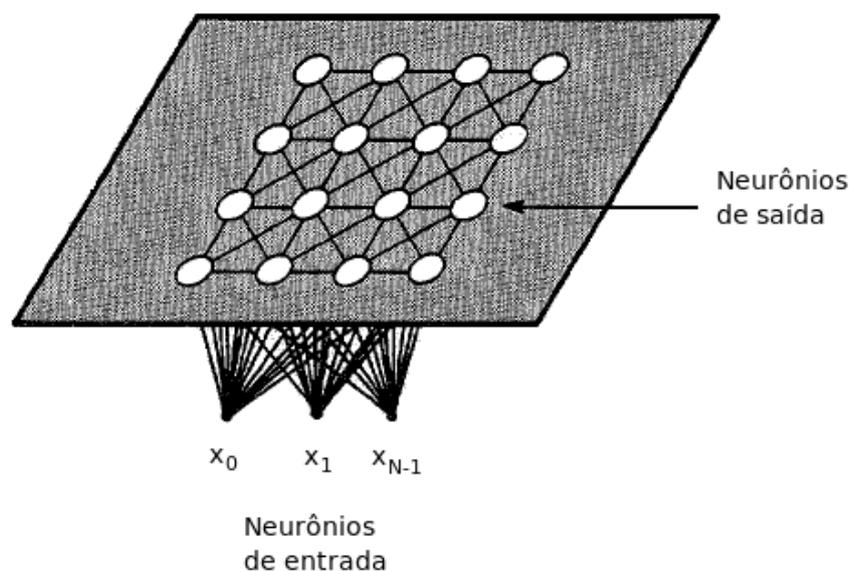


Figura 8: Disposição dos neurônios em uma rede de Kohonen. [Adaptado de (LIPPMANN, 1987)]

Como pode ser observado pela figura 8, na grade pós-sináptica os neurônios são interligados aos neurônios mais próximos e os neurônios de entrada são ligados com todos os neurônios da grade.

O mapa auto-organizável, como descrito por (LIPPMANN, 1987) é treinado utilizando uma aprendizagem híbrida composta por aprendizagem Hebbiana e competitiva, onde os neurônios da grade pós-sináptica competem entre si, com base em uma medida de similaridade com o sinal de entrada, o neurônio mais similar é dito ser o vencedor. O neurônio vencedor por sua vez excita os neurônios próximos a ele. A distância euclidiana é geralmente utilizada como medida de similaridade.

O algoritmo responsável pela formação do mapa começa pela atribuição de valores iniciais dos pesos sinápticos da grade, que segundo (HAYKIN, 1999) deve ser feito atribuindo valores pequenos gerados aleatoriamente.

Após esta etapa, Haykin (1999) identifica três processos essenciais envolvidos na formação do mapa auto-organizável:

**Competição** Para cada padrão de entrada, os neurônios da grade competem entre si, calculando a distância euclidiana entre seus pesos sinápticos e os valores do padrão de entrada,

sendo o neurônio com a menor distância o vencedor (Aprendizagem Competitiva);

**Cooperação** O neurônio vencedor determina uma área, vizinhança topológica, na qual os neurônios cooperarão entre si (Aprendizagem Hebbiana);

**Adaptação Sináptica** Determinada a vizinhança, os neurônios cooperam de modo a atualizarem seus pesos sinápticos, sendo que os mais próximos do neurônio vencedor sofrem modificações mais significativas do que os neurônios mais distantes.

A função Gaussiana é utilizada para definir como será feita a atualização dos pesos dos neurônios na vizinhança, neste caso o neurônio vencedor está localizado no centro da vizinhança. A Eq. (1.11) mostra o cálculo da excitação de um neurônio  $j$ , sendo  $d$  a distância entre o neurônio  $j$  e o neurônio vencedor  $i$ . A distância euclidiana, definida por  $D(x, y) = \sqrt{x^2 + y^2}$ , é comumente utilizada.

$$h_{ji} = \exp\left(-\frac{d_{ji}^2}{2\sigma^2}\right) \quad (1.11)$$

onde  $\sigma$  é a “largura efetiva” da vizinhança topológica. Ela mede o grau com o qual neurônios excitados na vizinhança participam do processo de aprendizagem (HAYKIN, 1999).

O autor supracitado ainda afirma que em um sentido qualitativo a vizinhança topológica gaussiana é mais biologicamente apropriada do que uma vizinhança retangular.

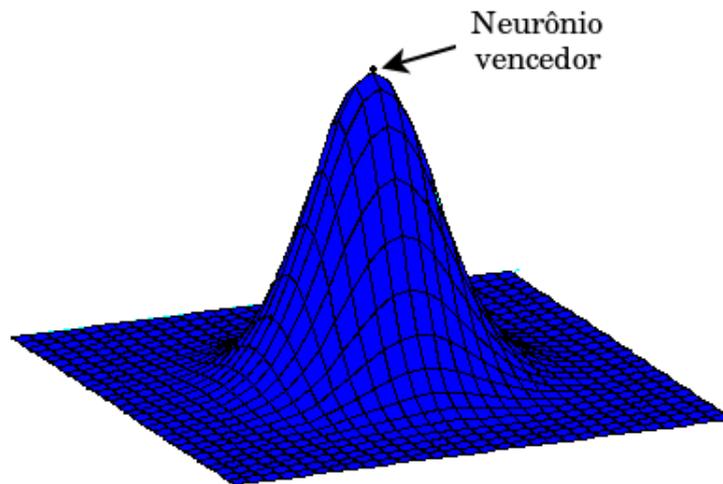


Figura 9: Vizinhança gaussiana

Com o decorrer da execução do algoritmo, os padrões de entrada similares serão mapeados em neurônios topologicamente próximos, formando áreas ou *clusters*.

O **SOM!** é aplicado à clusterização de dados, ou seja, agrupamento de dados intrinsecamente semelhantes, utilizado quando nada é conhecido sobre os dados. Os mapas de Kohonen podem ser aplicados em compressão de dados, uma vez que dados de alta dimensionalidade são mapeados em um espaço de baixa dimensão, preservando seu conteúdo (BASHEER; HAJMEER, 2000).

### 1.4.3 Rede de Hopfield

A rede de Hopfield pode ser vista como uma rede totalmente conectada agindo como uma memória associativa, capaz de armazenar padrões (HOPFIELD, 1984). De acordo com (KOVÁCS, 1996) uma memória associativa serve para armazenar um conjunto de vetores, de tal forma que se for endereçada com um vetor arbitrário  $y$ , retornará como saída aquele vetor mais próximo em algum sentido pré-definido. Geralmente é utilizada a distância de *Hamming*, como forma de mensurar a proximidade entre os vetores. A distância de Hamming entre dois vetores (*strings*) de mesmo tamanho é definido pelo número de posições que os símbolos correspondentes são diferentes (LIPPMANN, 1987).

Observando a definição é possível concluir que uma memória associativa pode ser interpretada como um classificador de padrões, onde as classes são representadas pelos vetores armazenados.

O modelo proposto por (HOPFIELD, 1982) é a implementação de uma memória associativa por uma rede recorrente, porém não há auto-relimentação. Como identificado na figura 10.

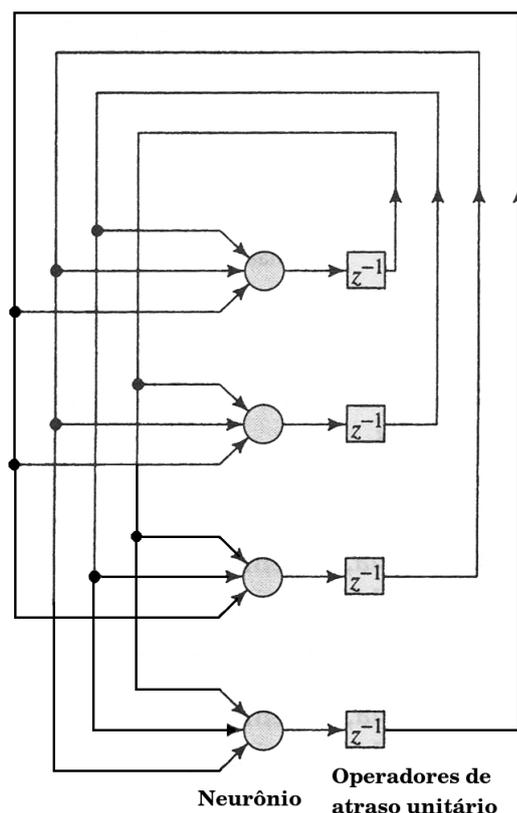


Figura 10: Memória Associativa recorrente de Hopfield. [Fonte: (HAYKIN, 1999)]

Neste modelo os vetores de entrada são normalmente binários. A rede é iniciada por um vetor  $x(0)$  e gera  $y(0)$  como saída, sendo  $y(0)$  reutilizado como entrada da rede no ciclo posterior. Esse processo é realizado até que a rede se estabilize, ou seja,  $y(k+1) \cong y(k)$ , este processo é referido como *dinâmica da rede* (KOVÁCS, 1996). O estado no qual a rede se estabilizou é dito ser um mínimo local do problema.

Com o intuito de análise o processo de evolução da rede, uma função de energia ou de *Lyapunov* é utilizada. De acordo com (KOVÁCS, 1996) a função de energia da rede submetida a sua própria dinâmica, só pode decrescer ou permanecer constante, ou seja, no decorrer da execução a rede tenta ir para estados melhores ou no mínimo iguais ao inicial.

Como observado por (LIPPMANN, 1987), a rede de Hopfield é mais apropriada quando uma exata representação binária é possível e menos apropriada quando os valores de entrada são contínuos.

Os principais focos de aplicação são: classificação e reconhecimento de padrões, quando usada como uma memória associativa; e também em otimização (HOPFIELD; TANK, 1986), quando aplicada considerando a função de energia.

## 1.5 Aplicações

Com o grande número distinto de redes neurais presentes na literatura e considerando a diversidade de tipos de aprendizagem e topologias, as redes neurais de um modo geral são aplicadas em grande parte das áreas do conhecimento. Basheer e Hajmeer (2000) destacaram sete categorias, as quais as RNA's vem sendo aplicadas com sucesso.

### 1. Classificação de Padrões

Dentre as categorias aqui apresentadas, a classificação de padrões talvez seja a categoria com maior aplicação das RNA's. Basicamente o problema consiste em classificar padrões de entrada entre classes previamente conhecidas, tomando como base de conhecimento algumas características do problema aplicado. As redes **MLP!** e **RBF!** são comumente aplicadas a esta classe de problemas.

### 2. Clusterização

A clusterização pode ser vista como um problema de classificação de padrões, quando é desconhecido previamente quantas e quais são as classes a ser classificadas. A este tipo de problema aplicam-se redes que utilizem aprendizagem não-supervisionada, podendo destacar a Rede de Kohonen.

### 3. Aproximação de Função

Quando é desejada a aproximação de funções complexas e com alta não-linearidade, algoritmos exatos nem sempre obtém boas aproximações, com isso as redes neurais surgem como uma alternativa para esse problema. As redes **MLP!** são conhecidas como aproximadores universais, podendo aproximar qualquer função em qualquer nível de precisão. Destaca-se também as redes **RBF!** em que seu processo de aprendizagem nada mais é do que a interpolação dos dados de entrada em uma superfície multidimensional.

### 4. Previsão

Com base em um histórico de cenários e as devidas ações tomadas é possível

prever qual será a ação a ser executada em um novo cenário. As redes **MLP!** e **RBF!** são aplicadas a esta categoria de problemas.

## 5. Otimização

Problemas de otimização consistem em encontrar o valor máximo ou mínimo a partir de uma função objetivo. Mesmo sendo um campo bem estabelecido na matemática, as redes neurais principalmente as redes de Hopfield foram aplicadas com sucesso em problemas de otimização complexos e não-lineares.

## 6. Associação

A associação consiste no desenvolvimento de uma rede associadora de padrões, a qual foi treinada utilizando dados com ruídos (*noise*) e posteriormente é aplicada a classificação de padrões sem ruídos. Também pode ser aplicado na reconstrução de dados corrompidos ou ausentes. Redes de Hopfield são geralmente utilizadas nesta aplicação (LIPPMANN, 1987).

## 7. Controle

Aplicação de redes neurais para auxiliar um sistema de controle adaptativo. O objetivo é gerar uma entrada de controle, tal que o sistema siga a trajetória determinada pelo modelo de referência (JAIN; MAO; MOHIUDDIN, 1996).

# 1.6 Conclusão

As redes neurais são modelos pertencentes ao paradigma de aprendizagem conexionista indutivo, na qual um conjunto de unidades simples de processamento (neurônios), são fortemente conectados, formando uma complexa e altamente robusta ferramenta de aquisição do conhecimento.

O conhecimento é armazenado na forma de pesos das conexões entre os neurônios, por pesos entende-se como nível de influência de um neurônio no neurônio seguinte, nível este que pode ser negativo (inibitório) ou positivo (excitatório).

A grande aplicabilidade das redes neurais é dada pelo fato da diversidade de modelos presentes na literatura, com diferentes modos, paradigmas e formas de aprendizagem. Destaca-se a rede *Multilayer Perceptron*, denominada por (BASHEER; HAJMEER, 2000) como o “*workhorse*” das redes neurais, sendo este um dos modelos mais largamente utilizado (BASHEER; HAJMEER, 2000).

Algumas de suas características principais são: (i) a habilidade de reconhecer e aprender as relações fundamentais entre as entradas e a saída, sem levar em consideração a dimensionalidade dos dados e a não-linearidade do sistema; (ii) a alta tolerância para dados ruidosos (*noise data*).

Redes neurais possuem suas limitações, destacando: (i) sucesso depende da qualidade e quantidade de dados disponível para treinamento; (ii) ausência de regras claras

e procedimentos efetivos para determinação da arquitetura ótima; (iii) a incapacidade de explicar de forma compreensível como são obtidas as repostas, dado um conjunto novo de entrada, característica esta que originou o termo *black boxes*. Termo este utilizado pelo não conhecimento do funcionamento interno de uma rede neural.

A fim de suprimir estas deficiências, várias abordagens foram propostas na literatura. Em conjunto com suas características positivas, pode-se concluir que as redes neurais constituem um dos mais importantes e eficazes métodos de aprendizagem de máquina presentes na literatura até então.

# Referências

- ACHARYA, U. R. et al. Classification of heart rate data using artificial neural network and fuzzy equivalence relation. *Pattern Recognition*, v. 36, p. 61–68, 2003.
- AMARI, S. et al. Asymptotic statistical theory of overtraining and cross-validation. In: *IEEE Transactions on Neural Networks*. New York: [s.n.], 1997. v. 5, n. 8, p. 985–996.
- ARBIB, M. A. *The Handbook Of Brain Theory and Neural Networks*. 2. ed. London, England: The MIT Press, 2002.
- BASHEER, I. A.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, v. 43, p. 3–31, 2000.
- GONÇALVES, A. R.; CAMARGO-BRUNETO, M. A. de O. Automatização do processo de determinação do número de ciclos de treinamento de uma rede neural artificial. In: *Aceito para: II Congresso Nacional de Extensão Universitária e XI Encontro de Atividades Científicas da Unopar*. Londrina, Paraná: [s.n.], 2008.
- GONÇALVES, A. R.; CAMARGO-BRUNETO, M. A. de O. Um novo modelo híbrido baseado em otimização por colônia de formigas e redes neurais para identificação de indivíduos com dpoc. In: *Aceito para: XI Congresso Brasileiro de Informática Aplicada a Saúde*. Campos do Jordão, São Paulo: [s.n.], 2008.
- GUPTA, M. M.; JIN, L.; HOMMA, N. *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. [S.l.]: Wiley-IEEE Press, 2003.
- HAN, S.-S.; MAY, G. S. Optimization of neural network structure and learning parameters using genetic algorithms. In: *ICTAI '96: Proceedings of the 8th International Conference on Tools with Artificial Intelligence (ICTAI '96)*. Washington, DC, USA: IEEE Computer Society, 1996. p. 200.
- HAYKIN, S. *Redes Neurais, Princípios e prática*. 2. ed. [S.l.]: Bookman, 1999.
- HEBB, D. O. *The Organization of Behavior: a neuropsychological theory*. New York: Willey, 1949.
- HINTON, G. E.; ACKLEY, D. H.; SEJNOWSKI, T. J. A learning algorithm for boltzmann machines. *Cognitive Science*, v. 9, p. 147–169, 1985.
- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Biophysics*, v. 79, p. 2554–2558, Abril 1982.
- HOPFIELD, J. J. Neurons with graded response have collective computational properties like those of two state neurons. *Biophysics*, v. 81, p. 3088–3092, Maio 1984.

- HOPFIELD, J. J.; TANK, D. W. Computing with a neural circuits: a model. *Science*, v. 233, n. 4764, p. 625–633, Agosto 1986.
- JAIN, A. K.; MAO, J.; MOHIUDDIN, K. Artificial neural networks: A tutorial. *IEEE Computer*, v. 29, p. 31–44, 1996.
- KOHAVI, R. A study a cross validation a bootstrap for accuracy estimation and a model selection. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. [S.l.: s.n.], 1995.
- KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, v. 43, p. 59–69, 1982.
- KOVÁCS, Z. L. *Redes Neurais Artificiais: fundamentos e aplicações*. São Paulo: Acadêmica, 1996.
- LIPPMANN, R. An introduction to computing with neural nets. *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, v. 4, n. 2, p. 4–22, 1987.
- MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943.
- REIS, M. C. *A unidade básica do Sistema Nervoso: Neurónio*. 2008. Website.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958.
- VON ZUBEN, F.; ATTUX, R. R. *Redes Neurais com Função de Base Radial*. 2008. Disponível em: <[ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia353\\_1s07/topico9\\_07.pdf](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia353_1s07/topico9_07.pdf)>.
- ZHANG, C.; SHAO, H.; LI, Y. Particle swarm optimisation for evolving artificial neural network. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. [s.n.], 2000. v. 4, p. 2487–2490 vol.4. Disponível em: <<http://dx.doi.org/10.1109/ICSMC.2000.884366>>.