

3 FERRAMENTAS UTILIZADAS: REDES NEURAIS E LÓGICA FUZZY

3.1 REDES NEURAIS

As redes neurais representam uma tecnologia que têm raízes em muitas disciplinas: neurociência, matemática, estatística, física, ciência da computação e engenharia. Uma rede neural “artificial” é um dispositivo capaz de processar informação de forma distribuída e de incorporar conhecimento através de exemplos. A motivação da construção de uma rede neural nasceu da idéia de modelar a rede de neurônios humanos visando compreender o funcionamento do cérebro. Portanto, uma rede neural artificial se assemelha ao cérebro humano em dois aspectos: o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem e os pesos sinápticos, que são as forças de conexão entre neurônios, são utilizados para armazenar o conhecimento adquirido [11].

As principais vantagens de uma rede neural são enumeradas a seguir.

1. Pode ser utilizada em situações onde é difícil criar modelos adequados da realidade.
2. Tem a habilidade em inferir relações não lineares complexas.

3. Tem a capacidade de resolver problemas sem necessidade de definição de listas de regras ou de modelos precisos.
4. O conhecimento em redes neurais é adquirido através de um processo de aprendizado ou de treinamento.
5. A informação é transformada e armazenada em “densidades de conexão” conhecidas como pesos sinápticos (ou simplesmente “pesos”);
6. O conhecimento é produzido utilizando estes pesos para combinar não linearidades muito simples, fazendo com que esta não linearidade fique distribuída na rede;
7. Este conhecimento é incorporado sem que tenhamos que explicitar modelos a priori.

No entanto, tem-se um porém na utilização deste método, o número de dados de entrada a serem utilizados no modelo deve ser bastante significativo, já que se deve ter uma parte de dados para treinamento e outra para generalização do modelo.

A modelagem em Redes Neurais, que depende do projetista, deve seguir os passos enumerados na seqüência.

1. Escolha dos exemplos (observações) a serem usados para treinamento, de modo que o ambiente seja bem representado.
Os exemplos são pares do tipo: (entrada, saída desejada).

2. Escolha dos exemplos que serão utilizados para validação (avaliação de capacidade de generalização).
3. Escolha de uma arquitetura apropriada.
4. Escolha do algoritmo para treinamento da rede.

Os exemplos, pares de entrada e saída, a serem escolhidos devem ser aqueles que melhor representam o ambiente que se quer medir, ou seja, aqueles que melhor façam uma generalização do problema.

A escolha da arquitetura e do algoritmo dependem do problema que se está procurando solução. Cada algoritmo requer uma determinada arquitetura que cabe ao projetista escolhê-la.

São vários os algoritmos que são utilizados numa rede neural, os mais comuns são o de retro-propagação do erro (*error backpropagation*), o de bases radiais e o mapa auto-organizável de Kohonen. Reter-nos-emos a explanação deste último apenas, já que este foi o utilizado na presente dissertação de mestrado.

3.2 MAPA AUTO-ORGANIZÁVEL DE KOHONEN

3.2.1 Introdução

Os mapas auto-organizáveis de Kohonen são agrupados dentro das redes neurais, que a partir de um processo de treinamento agrupa os dados. Este agrupamento faz que a projeção dos dados sobre o mapa distribua suas características de uma forma gradual [19]. O mapa de Kohonen é usado para diferentes aplicações, tais como as descritas a seguir.

1. Clustering: se podem agrupar dados do conjunto de entrada, atendendo a diferentes critérios.
2. Visualização: este agrupamento, como se realiza de uma forma ordenada, permite visualizá-lo e descobrir características novas ou relações que não se haviam previsto de antemão. Também permite visualizar a evolução temporal de um conjunto de dados.
3. Classificação: uma vez treinado o mapa, e estabelecida algum tipo de etiqueta aos *clusters*, pode-se utilizar a rede para classificar dados desconhecidos.
4. Interpolação de uma função: designados valores numéricos a cada um dos nós da rede de Kohonen, se podem designar esses valores numéricos aos vetores de entrada.
5. Quantização vetorial: aplicação de uma entrada contínua a uma saída discretizada, ou seja, obter a partir um vetor qualquer o vetor mais próximo de um conjunto previamente estabelecido.

A rede de Kohonen é também definida como um mapa auto-organizável, pois a aprendizagem da rede é feita de modo não supervisionado. A característica principal de uma aprendizagem não supervisionada é que as informações são extraídas sem que haja um par de entrada e saída. Portanto, por tratar-se de um método não-supervisionado a rede de Kohonen tem a grande vantagem de não necessitar do estabelecimento de uma saída alvo. O tipo mais comum de algoritmo de aprendizagem não supervisionada são os algoritmos de análises de grupos ou *clustering*.

Os métodos não supervisionados são costumeiramente usados em análises de dados explanatórios, ou seja, em uma análise dos dados quando não se sabe de antemão quais são os grupos naturais que se formam e se quer visualizar a quantidade e a relação que há entre esses grupos naturais. Pode-se dizer que uma das principais aplicações é a visualização de dados multidimensionais, pois um algoritmo não supervisionado atua como projeção de um espaço multidimensional a outro de dimensões visualizáveis.

Um mapa auto-organizável consiste em modificar repetidamente os pesos sinápticos de uma rede neural em resposta a padrões de ativação e de acordo com regras pré-estabelecidas, até que se desenvolva uma configuração final. O procedimento irá buscar arrumações naturais no banco de dados. Este tipo de técnica é o que mais se aproxima do que se convencionou chamar, talvez de uma forma exageradamente genérica, de “*data mining*”.

3.2.2 A Rede de Kohonen

Desenvolvida por Tuevo Kohonen na década de 80 [15], a rede neural *Self Organizing Map* (SOM) é sustentada biologicamente por mapas topológicos presentes no córtex cerebral. Esse modelo possui a capacidade de se auto-organizar, preservando topologicamente a estrutura dos padrões apresentados.

A arquitetura de um mapa auto-organizável de Kohonen é composta de uma malha, em geral bidimensional, onde todas as unidades estão conectadas a todas as componentes de entrada. Cada neurônio está, portanto, conectado às componentes de entrada através de um vetor de pesos cuja dimensão é a mesma da do vetor de entrada. A figura 4 mostra a estrutura de uma rede de Kohonen.

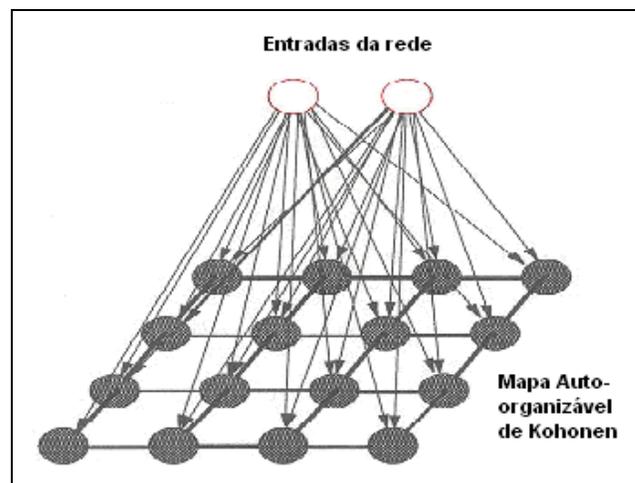


Figura 4: Representação de uma rede de Kohonen

Na rede SOM, neurônios próximos (na disposição do mapa) devem responder por funções similares, tal como no cérebro de animais mais evoluídos [7]. Em tais cérebros, determinadas áreas respondem por estímulos de mesma natureza, fala, audição, visão, etc. Neste tipo de rede neural é introduzido, portanto, o conceito de vizinhança local, isto é, o comportamento de uma determinada unidade é diretamente afetado pelo comportamento das unidades vizinhas. Com isso, o mapa auto-organizável de Kohonen tem a capacidade de alocar em um mesmo neurônio, ou em neurônios vizinhos, entradas que estão de alguma forma próximas no espaço original.

3.2.3 Processo de Aprendizagem

No processo de aprendizagem calcula-se para cada neurônio a proximidade de seu vetor de pesos ao vetor de entrada. Os neurônios competem então pelo privilégio de aprender. É importante frisar que o tamanho do vetor de entrada e do vetor de pesos de cada um dos neurônios da rede de Kohonen devem ser o mesmo. Além disso, esses vetores de pesos devem ter valores iniciais que podem ser dados aleatoriamente dentro de um determinado valor.

A medida de proximidade do vetor de entrada ao vetor de peso de cada um dos neurônios da rede pode ser feita utilizando, por exemplo, a distância euclidiana. Mas, isto não impede alguma outra métrica seja utilizada. Portanto, feita esta medida de distância entre vetores, aquele que

possuir a menor distância é o neurônio vencedor e é o único que gera um sinal de saída. Assim este neurônio e seus vizinhos são os únicos que podem aprender com esta apresentação de padrão e podem modificar seus pesos.

Como os pesos do neurônio vencedor estão próximos ao vetor de entrada, os pesos dos vizinhos não têm o porquê de cumprir isto. Portanto, as mudanças de pesos dos vizinhos costumam ser maiores que as do próprio ganhador.

3.2.4 Algoritmo de Kohonen

Vamos supor que as entradas são vetores de tamanho “p” que chamaremos de “x”. Os números são atributos do padrão

$$x = [x_1, x_2, \dots, x_p]$$

O vetor de pesos correspondente ao neurônio j é representado por:

$$w_j = [w_{j1}, w_{j2}, \dots, w_{jp}]^T$$

Suponhamos também que estabelecemos uma malha com “q” neurônios.

Vamos treinar a rede iterativamente. No passo “k” deste procedimento teremos o vetor de pesos $w_j(k)$ e a entrada $x(k)$.

Note que cada um dos vetores de entrada será apresentado diversas vezes e, portanto, $x(k)$ não representa o k-ésimo padrão e sim a k-ésima vez que um vetor de entrada é apresentado à rede.

O algoritmo, então, deve ser feito seguindo os passos que são descritos a seguir.

1. Escolha de modo aleatório os “q” vetores de pesos iniciais $w_j(0)$,
 $j = 1 \dots q$.
2. Apresente a entrada $x(k)$, onde $k = 0$ na primeira iteração.
3. Calcule a distância de $x(k)$ a cada um dos “q” vetores de pesos $w_j(k)$, $j = 1 \dots q$, e nomeie como “neurônio vencedor” aquele para o qual $\| w_j(k) - x(k) \|$ é minimizado. Onde $\| \cdot \|$ refere-se, como dito anteriormente, à norma a ser utilizada (euclidiana, por exemplo).

Suponhamos que o neurônio “i” ganhou esta competição.

4. Escolha uma vizinhança $V_i(n)$ para o neurônio vencedor “i”.
5. Atualize os pesos de todos os neurônios que pertencem a vizinhança $V_i(n)$ do seguinte modo:

$$w_j(n+1) = \begin{cases} w_j(n) + \eta(n)[x(n) - w_j(n)] & j \in V_i(n) \\ w_j(n) & \text{caso contrario} \end{cases} \quad (1)$$

onde η é um parâmetro de aprendizado.

6. Retorne ao passo 2 até a convergência.

Modificamos os vetores de pesos de todos os neurônios da vizinhança de atualização visando diminuir a distância entre a entrada apresentada e cada um dos vetores de pesos. É desta forma que estes

neurônios se especializam em responder a um determinado tipo de estímulo.

Os parâmetros $\eta(n)$ e $V(n)$, em geral, variam ao longo do treinamento. No início são maiores, permitindo uma maior flutuação em uma região mais ampla da rede, e ao final diminuem, permitindo um ajuste fino.

Após o treinamento, a cada neurônio da malha está associado um vetor de pesos que pode ser visto como representante típico dos elementos deste grupo. Vale ressaltar que o vetor de pesos resultante do treinamento não é central com relação aos elementos que respondem ao neurônio em questão.

Treinada a rede o passo final é apresentar os vários padrões de entrada e esperar que estes se ajustem “automaticamente”, onde os vetores de entrada de características semelhantes devem ficar em mesmos *clusters* ou em *clusters* vizinhos.

3.3 LÓGICA FUZZY

3.3.1 Introdução

A lógica fuzzy ou nebulosa se preocupa com os princípios formais do raciocínio aproximado, ou seja, fornece um ferramental matemático para o tratamento de informações de caráter impreciso ou vago.

Um sistema de lógica fuzzy pode operar dados numéricos e conhecimentos lingüísticos simultaneamente. Um sistema nebuloso é um sistema não-linear de mapeamento de um vetor de entrada em uma saída escalar, isto é, um mapa de números dentro de um mapa de números [18]. Uma das suas principais vantagens é que toda a formulação é feita à partir de regras lingüísticas, sem necessitar de modelo matemático formal. Com isso é capaz de incorporar tanto o conhecimento objetivo quanto o conhecimento subjetivo.

Segundo Zadeh [38], “quanto mais próximo de um problema real alguém olha, mais fuzzy este problema se torna”. Conforme a complexidade de um sistema, nossa habilidade de fazer declarações precisas e significativas sobre o seu comportamento diminui, até alcançar um limite além do qual precisão e relevância tornam-se características mutuamente exclusivas. A lógica nebulosa fornece um método para reduzir e explicar a complexidade de um sistema.

A riqueza da lógica fuzzy é que há um número enorme de possibilidades que levam a um grande número de diferentes mapas. No entanto, esta riqueza requer um cuidadoso conhecimento da teoria lógica nebulosa e dos elementos que a compõem. Quer-se neste tópico dar uma idéia geral da teoria e dos elementos que compõem esta lógica.

Nos tópicos que se seguem serão falados sobre os componentes de um sistema nebuloso, tais como: conjuntos fuzzy, operações, proposições fuzzy, inferência, fuzzificação e defuzzificação. É bem verdade que nem todos os componentes deste sistema serão utilizados neste trabalho, mas

mesmo assim será dada uma explanação, mesmo que superficial de todos estes.

3.3.2 Conjuntos Fuzzy

Nos conjuntos ditos ordinários (ou “Crisp”) a noção de pertinência é bem definida, os elementos pertencem ou não pertencem a um dado conjunto A (em um universo X). Ou seja:

$$f_A(x) = \begin{cases} 1 & \text{se e somente se } x \in A \\ 0 & \text{se e somente se } x \notin A \end{cases} \quad (2)$$

Onde essa função “f” é denominada função de característica.

Existem, no entanto, conjuntos cujo limite entre pertinência e não-pertinência é vago, podem-se citar, por exemplo, o conjunto de pessoas altas, o conjunto de temperaturas baixas, os números muito maiores do que 1, etc. São nesses conjuntos que a lógica fuzzy é utilizada, pois a função característica é generalizada, podendo assumir um número infinito de valores no intervalo entre 0 e 1. Um conjunto nebuloso A em um universo X é definido, por uma função de pertinência:

$$\mu_A(x) : X \rightarrow [0,1]$$

A seguir tem-se a função característica e a função de pertinência para o conjunto **números muito maiores que 1**, com isso pode-se ver a formulação crisp e a formulação fuzzy de um mesmo problema.

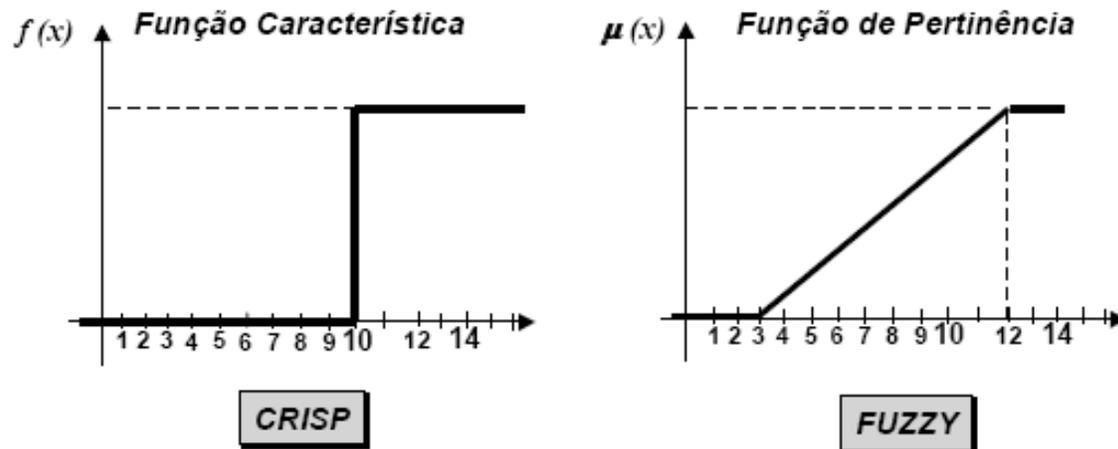


Figura 5: Comparação entre um conjunto crisp e um conjunto fuzzy

A representação matemática de um conjunto fuzzy discreto pode ser escrita como:

$$\sum_{i=1}^n \mu_A(x_i) / x_i \quad (3)$$

Já a representação matemática de um conjunto fuzzy contínuo é dada por:

$$\int_x \mu_A(x) / x \quad (4)$$

Um conjunto nebuloso pode assumir várias formas regulares ou irregulares, as mais comuns são: linear, trapezoidal, triangular, formato S, formato Z, formato Pi, gaussiana e singleton.

Definidos os conjuntos fuzzy, podemos definir agora a variável lingüística, que é a variável cujos valores são nomes de conjuntos fuzzy. A

figura a seguir ilustra o explanado anteriormente com um exemplo da temperatura de um processo qualquer.

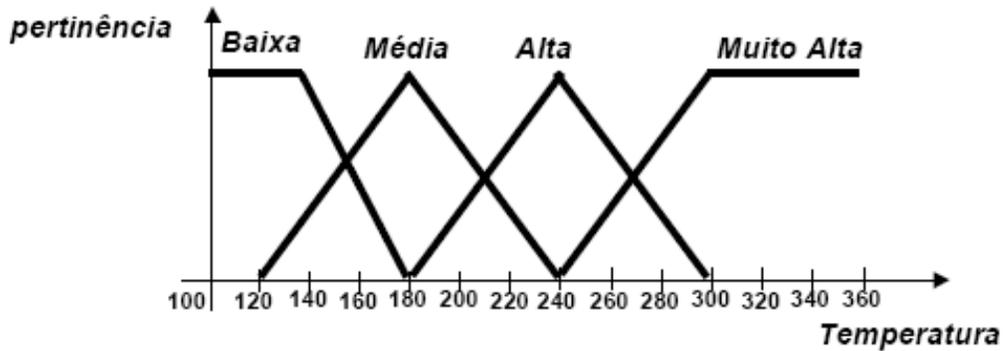


Figura 6: Diferentes variáveis lingüísticas para a variável temperatura de um processo

Como se pode ver na figura mostrada anteriormente, as variáveis lingüísticas para a variável temperatura são: baixa, média, alta e muito alta. Vê-se também que para cada uma destas variáveis têm-se conjuntos de formas diferentes.

3.3.3 Conectivos Lógicos

A lógica considerada para tratamento das variáveis nebulosas ou funções de pertinência é a lógica de processo, na qual os operadores utilizados são o **máximo** e o **mínimo**.

Para a **união** de dois conjuntos, a lógica de processo utiliza como operador o **máximo**, cujo resultado será a variável de entrada com o maior valor. Ou seja,

$$\mathbf{C = A \vee B = \max (x_1, x_2, \dots, x_n)} \quad (5)$$

Assim, se $A = 0.5$ e $B = 0.7$, então $C = A \vee B = \max (0.5, 0.7) = 0.7$.

Já para **interseção**, o operador utilizado pela lógica de processo é o **mínimo**, no qual o resultado será o valor mínimo das variáveis de entrada. Ou seja,

$$\mathbf{C = A \wedge B = \min (x_1, x_2, \dots, x_n)} \quad (6)$$

Neste caso, sendo $C = A \wedge B$, então $C = \min (0.5, 0.7) = 0.5$.

Deve-se citar também um operador que é bastante utilizado na lógica fuzzy que é o operador **complemento**, que equivale à negação. O resultado do complemento de x será $1 - x$.

É importante frisar que para lógica fuzzy os valores x_1, x_2, \dots, x_n correspondem aos graus de pertinência dos conjuntos nebulosos selecionados.

3.3.4 Proposições ou Regras Fuzzy

São frases que utilizam além dos conectivos lógicos “e”, “ou” e “não”, onde o “e” denota interseção, o “ou” denota união o “não” denota negação, utilizam também os operadores de implicação que são frases do tipo:

“se ... então ...”.

A operação do tipo “se ... então ...” é também denominada de declaração condicional fuzzy e descreve a dependência do valor de uma

variável lingüística em relação ao valor de outra. Como exemplo, podemos descrever a seguinte proposição:

se (A e/ou B) então (c).

3.3.5 Inferência Fuzzy

Na inferência fuzzy, os princípios da lógica nebulosa são utilizados para combinar as regras ou proposições (do tipo “se ... então ...” de uma base de regras fuzzy) de um mapa de entradas fuzzy em um mapa de saídas também fuzzy [18]. É na inferência fuzzy que a ativação e combinação das regras são determinadas.

A inferência de conjuntos fuzzy está alocada entre duas fases que compõem um sistema nebuloso que são a fuzzificação, que faz o mapeamento dos dados precisos de entrada para os conjuntos nebulosos, e a defuzzificação, que interpreta o conjunto fuzzy de saída.

3.3.6 Defuzzificação

Como dito anteriormente, a defuzzificação tem por função interpretar o conjunto fuzzy da saída [14]. O resultado de uma ação irá depender desta fase, já que são vários os métodos de defuzzificação. Alguns deles e que são os mais comuns serão listados a seguir.

1. **Máximo:** examina-se o conjunto nebuloso de saída e se escolhe como valor preciso o valor no universo da variável de saída para o qual o grau de pertinência é o máximo.
2. **Média dos máximos:** a saída precisa é obtida tomando-se a média entre os dois elementos extremos no universo que correspondem aos maiores valores da função de pertinência do conjunto fuzzy de saída.
3. **Centróide:** a saída precisa é o valor no universo que corresponde ao centro de gravidade do conjunto nebuloso.