

2 Revisão Bibliográfica

Atualmente, em empresas cujo processo produtivo é automatizado, há uma imensidão de dados operacionais disponíveis, porém pouca informação consegue ser extraída a partir destes dados (Lin *et al.*, 2000). Com o intuito de auxiliar as empresas a fazerem melhor uso da sua base de dados, sistemas de apoio à decisão podem ser utilizados. Uma das principais tarefas destes sistemas é auxiliar os operadores humanos em momentos onde os mesmos devem intervir no processo produtivo. Com o objetivo de determinar quando, onde e com que intensidade deve ser feita a intervenção, pode-se utilizar técnicas de inferência que visam a relacionar as variáveis operacionais do processo com variáveis cujo valor é indicativo do desempenho do processo (como, por exemplo, a qualidade do produto final). O uso de tais técnicas depende, no entanto, de modelos que descrevam adequadamente a dependência entre as variáveis intermediárias e as variáveis de processo.

Convém mencionar que a atividade de modelagem de processos visa a representar os principais fenômenos que ocorrem em um dado processo através de equações que correlacionam as variáveis de processo mais significativas (Valdman, 1999). Constata-se, porém, que a modelagem da qualidade do produto torna-se um assunto muito mais complexo quando se deseja modelar um processo polimérico (Kiparissides, 1996). Isto se deve ao fato de que as propriedades moleculares e morfológicas de um polímero influenciam fortemente suas propriedades físicas, químicas, térmicas, reológicas e mecânicas, e, conseqüentemente, a sua aplicação final. Além disso, medidas da qualidade do polímero, tais como índice de fluidez ou densidade, não são obtidas em tempo real, estando disponíveis apenas com frequência limitada. Desta forma, modelos capazes de prever a qualidade do polímero a partir das condições operacionais tornam-se essenciais para o monitoramento adequado da qualidade do produto e para a melhoria do desempenho operacional e econômico da planta.

A modelagem utilizada em sistemas de inferência de processos químicos pode ser caracterizada como fenomenológica, empírica ou baseada no histórico do processo (Ohshima e Tanigaki, 2000). Um esquema simplificado desta classificação encontra-se na Figura 1.

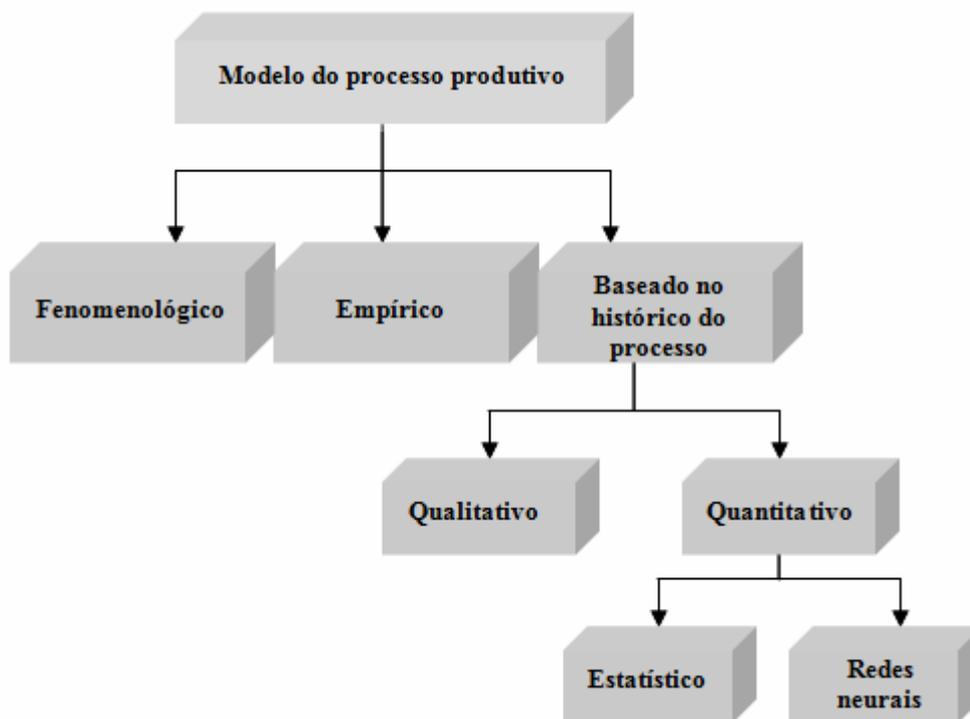


Figura 1 – Classificação dos modelos usados em sistemas de inferência

A seguir, será apresentada uma breve descrição dos tipos de modelos usados em sistemas de inferência, juntamente com os principais trabalhos desenvolvidos no âmbito de processos de polimerização, tendo como base a literatura científica pertinente.

2.1. Modelos fenomenológicos

O desenvolvimento de modelos que representem o comportamento de processos produtivos geralmente requer uma compreensão dos fenômenos físico-químicos que ocorrem durante todo o processo analisado. Modelos fenomenológicos expressam estes fenômenos através de relações matemáticas derivadas das leis fundamentais da física, química e físico-química.

No entanto, o desenvolvimento de modelos fenomenológicos detalhados para processos de polimerização não é uma tarefa trivial. As dificuldades envolvidas nesta tarefa são derivadas do grande número de reações complexas que ocorrem simultaneamente durante o processo de polimerização, bem como da necessidade de estimar um grande número de parâmetros cinéticos para o modelo (Sharmin *et al.*, 2006).

Um dos modelos fenomenológicos mais conhecidos na área de processos poliméricos é o modelo de McAuley e MacGregor (1991), cujo desenvolvimento baseou-se em simplificações do modelo cinético proposto por McAuley e colaboradores (1990). McAuley e MacGregor (1991) propuseram a seguinte relação entre a qualidade do produto (representada pelo índice de fluidez do polímero) e as variáveis de processo medidas:

$$\ln(MI) = k_0 \left(\frac{1}{T} - \frac{1}{T_0} \right) + 3,5 \ln \left(k_1 + k_2 \frac{[C_4]}{[C_2]} + k_3 \frac{[C_n]}{[C_2]} + k_4 \frac{[H_2]}{[C_2]} + k_5 \frac{[R]}{[C_2]} \right) \quad (2.1)$$

onde:

MI – índice de fluidez do polímero (*melt index*)

T – temperatura do reator em escala absoluta

T_0 – temperatura de referência em escala absoluta

$[C_4]$ – concentração de buteno no reator

$[C_2]$ – concentração de eteno no reator

$[C_n]$ – concentração de olefinas de ordem superior no reator

$[H_2]$ – concentração de hidrogênio em fase gasosa no reator

$[R]$ – concentração de cocatalisador no reator

$k_0, k_1, k_2, k_3, k_4, k_5$ – parâmetros ajustáveis do modelo

Segundo a metodologia proposta por McAuley e MacGregor (1991), o cálculo do índice de fluidez do polímero deve ser realizado em duas etapas. A primeira etapa faz uso de um modelo instantâneo de propriedades, que descreve a relação entre as variáveis de processo e o valor da propriedade do polímero em um dado momento. A segunda faz uso de um modelo cumulativo de propriedades, que descreve a relação entre as propriedades instantâneas e cumulativas do polímero dentro do reator. O desenvolvimento desta modelagem baseou-se nas seguintes pressuposições: a reação de polimerização é muito mais rápida do que o

tempo de residência do polímero no reator; todo o polímero gerado dentro de um curto espaço de tempo possui estrutura química similar; a estrutura do polímero em um determinado instante de tempo é determinada pelas condições operacionais do processo naquele momento. McAuley e MacGregor (1991) também propuseram uma técnica de atualização dos parâmetros ajustáveis do modelo através do uso do filtro recursivo de Kalman, o que permitia que os parâmetros fossem atualizados assim que os resultados de análises laboratoriais das propriedades do polímero tivessem sido disponibilizados.

2.2. Modelos empíricos

Modelos empíricos representam o comportamento do processo através de regressões lineares ou não-lineares de dados experimentais. O desenvolvimento de modelos empíricos envolve basicamente a análise quantitativa paramétrica dos efeitos experimentais que podem ser verificados nas variáveis de processo quando determinadas modificações pré-fixadas das condições operacionais são provocadas (Valdman, 1999). A modelagem empírica não requer, portanto, um conhecimento profundo do processo. Além disso, as equações obtidas através de modelos empíricos são geralmente simples, o que possibilita a sua resolução em tempo real. Convém, contudo, salientar que os modelos empíricos apresentam baixo poder de generalização, isto é, eles fornecem resultados satisfatórios apenas para as condições operacionais consideradas durante o desenvolvimento do modelo.

Na literatura pesquisada, não foram encontradas propostas de modelos empíricos relevantes à área de processos poliméricos.

2.3. Modelos baseados no histórico do processo

Visando ao desenvolvimento de sistemas inteligentes para a indústria química, é freqüente o uso de modelos baseados no histórico do processo. Isto se deve ao fato que o único requisito para o desenvolvimento destes modelos é a disponibilidade de grandes quantidades de dados históricos do processo. Os dados históricos, por sua vez, podem ser processados de maneira qualitativa ou

quantitativa (Venkatasubramanian *et al.*, 2003^b). Dois dos principais métodos qualitativos baseados no histórico do processo são os sistemas especialistas e os métodos de modelagem de tendências. Por outro lado, métodos quantitativos baseados no histórico do processo podem ser classificados como estatísticos e não-estatísticos.

No que diz respeito aos métodos qualitativos baseados no histórico do processo, cabe salientar que a primeira tentativa de capturar conhecimento e tirar conclusões a partir do mesmo foi através do desenvolvimento de sistemas especialistas. Um sistema especialista pode ser descrito como um algoritmo computacional, cujo objetivo é imitar o comportamento cognitivo de um especialista humano que deve resolver determinados tipos de problemas. O sistema é constituído por uma base de dados que contém um grande número de regras do tipo *if-then-else* e por um mecanismo de inferência que realiza buscas na base de dados e propõe conclusões a partir dos dados presentes. Dado que o sistema especialista não utiliza qualquer conhecimento sobre os fenômenos físicos que ocorrem durante o processo, o método não funciona quando o mesmo se depara com uma situação que não havia sido previamente definida na sua base de dados.

No que diz respeito aos métodos quantitativos baseados no histórico do processo, cabe destacar o uso de redes neurais artificiais como ferramenta de modelagem de processos. Dado que a maioria dos processos químicos são não-lineares (Lennox *et al.*, 1998), as relações dinâmicas entre causa e efeito nestes sistemas são melhor descritas através de expressões matemáticas não-lineares. Diferentemente dos modelos lineares tradicionalmente utilizados, redes neurais artificiais são capazes de aproximar as relações dinâmicas entre causa e efeito de maneira genérica e não-linear (Lennox *et al.*, 2001). Visto que o desempenho dos algoritmos utilizados para o monitoramento e controle de processos produtivos industriais depende da precisão do modelo do processo incluído nos mesmos, o uso de redes neurais artificiais pode melhorar o desempenho dos algoritmos aplicados a sistemas não-lineares.

No âmbito de processos de polimerização, o desenvolvimento de sistemas de inferência de propriedades poliméricas utilizando redes neurais artificiais tem se mostrado como uma área de pesquisa bastante ativa (Bhat & McAvoy, 1990; Chan & Nascimento, 1994; Qin & McAvoy, 1992). Alguns dos trabalhos mais

recentes nesta área envolvem o uso de comitês de redes neurais (Zhang *et al.*, 1997), a implementação de modelos neurais em sistemas de controle de qualidade (Ogawa *et al.*, 1999), bem como o uso combinado de redes neurais artificiais e outras técnicas (Rallo *et al.*, 2002; Abonyi *et al.*, 2003; Sharmin *et al.*, 2006). Uma descrição mais detalhada destes trabalhos será apresentada a seguir.

Zhang e colaboradores (1997) fizeram uso de um conjunto de redes neurais artificiais operando em paralelo, isto é, de um comitê de redes neurais, para inferir a qualidade do produto de um processo polimérico. Neste contexto, a qualidade do produto era calculada através da combinação ponderada dos resultados obtidos em cada uma das redes. Desta forma, a acurácia do comitê estava diretamente relacionada com o peso atribuído ao resultado de cada rede no momento da combinação ponderada. Zhang e colaboradores (1997) propuseram o uso de regressão de componentes principais para determinar os pesos de cada rede neural. A metodologia em questão foi aplicada para dados de uma planta piloto de metacrilato de metila e os melhores resultados foram obtidos para um comitê de redes neurais artificiais composto por 30 redes. Segundo Zhang e colaboradores (1997), o uso do comitê possibilitou uma melhoria da modelagem do índice de fluidez devido à escassez de dados operacionais para o treinamento de uma única rede neural artificial. De fato, os pesquisadores alertaram para o fato de que, em situações onde existem quantidades suficientes de dados operacionais para o treinamento de uma única rede neural artificial, o uso de um comitê de redes não implica em um aumento significativo da acurácia e robustez da modelagem neural.

Ogawa e colaboradores (1999) propuseram uma metodologia para estimar o índice de fluidez do polietileno de alta densidade e usar o valor estimado em um sistema de controle de qualidade. Os pesquisadores basearam-se no modelo desenvolvido por Watanabe e colaboradores (1993), que estima o índice de fluidez do polímero a partir dos valores das variáveis de processo e que foi desenvolvido através de correlações empíricas de dados operacionais coletados em regime permanente. Assim como o modelo desenvolvido por McAuley e MacGregor (1991), o modelo de Watanabe e colaboradores (1993) apresenta parâmetros que podem ser atualizados em tempo real. Devido ao fato da etapa de atualização dos parâmetros adicionar uma quantidade significativa de esforço computacional ao cômputo do índice de fluidez, Ogawa e colaboradores (1999)

propuseram algumas simplificações ao modelo, com as quais foi possível obter bons resultados de inferência sem atualizar os parâmetros do modelo. O modelo simplificado foi, então, utilizado em um sistema de controle em cascata, cujo objetivo era controlar o valor do índice de fluidez do polímero.

Rallo e colaboradores (2002) desenvolveram uma metodologia para inferir a qualidade do polietileno de baixa densidade usando modelos baseados em redes neurais. Os pesquisadores testaram diferentes tipos de redes neurais artificiais, bem como diferentes técnicas de pré-processamento dos dados. Em relação ao pré-processamento dos dados operacionais, os pesquisadores determinaram, em um primeiro momento, o menor número de variáveis de processo necessárias para modelar o índice de fluidez do polímero. Esta seleção foi feita utilizando-se uma rede neural artificial não-supervisionada do tipo SOM (*Self-Organizing Map*). Em seguida, os dados operacionais referentes às variáveis selecionadas foram divididos em grupos de treinamento e de teste. Esta divisão foi feita de duas formas: a primeira técnica de divisão dos dados considerou a ordenação temporal dos dados e separou os dados mais antigos para treinamento e os mais recentes para teste; a segunda técnica agrupou os dados segundo uma lógica *fuzzy* e, em seguida, determinou os grupos que serviriam como dados de treinamento e os que serviriam como dados de teste. Em relação às redes neurais artificiais, três modelos foram analisados: o primeiro era composto por uma rede neural artificial não-supervisionada com arquitetura do tipo ART (*Adaptive Resonance Theory*), enquanto que os outros dois modelos eram compostos por redes neurais de bases radiais.

Abonyi e colaboradores (2003) desenvolveram uma metodologia baseada em redes neurais não-supervisionadas do tipo SOM para visualizar dados operacionais e prever a qualidade do produto em processos poliméricos. Primeiramente, os pesquisadores utilizaram uma rede do tipo SOM, que fornece uma representação compacta da distribuição dos dados operacionais, para agrupar os diversos tipos de resinas produzidas com base nas condições típicas de operação. Em seguida, o método dos mínimos quadrados ortogonais foi utilizado para determinar as variáveis de processo que mais afetavam a qualidade do produto, e a projeção bidimensional destas variáveis foi utilizada para o monitoramento do processo. Com o intuito de prever a qualidade do produto (i.e. valores de índice de fluidez e densidade do polímero), os pesquisadores

identificaram vários modelos lineares locais a partir do diagrama de Voronoi dos vetores protótipos da rede SOM. A modelagem proposta foi aplicada para dados de uma planta industrial de polietileno de alta densidade.

Sharmin e colaboradores (2006) propuseram um sistema de inferência da qualidade do polietileno de baixa densidade baseado em métodos de regressão multivariável. Os pesquisadores fizeram uso da ferramenta estatística PLS (*Partial Least Squares*), desenvolvendo, em um primeiro momento, um modelo universal, cujo objetivo era relacionar os dados operacionais com o valor do índice de fluidez de todas as resinas produzidas em uma determinada planta industrial. Este modelo foi capaz de representar adequadamente o comportamento do índice de fluidez em momentos onde o tipo de resina produzido estava sendo modificado. No entanto, o modelo universal não permitiu modelar adequadamente o comportamento do índice de fluidez quando o processo de produção de uma resina já havia entrado em regime permanente. De forma a superar esta deficiência, os pesquisadores desenvolveram, em um segundo momento, modelos individuais para cada tipo de resina produzido. Estes modelos individuais, por sua vez, permitiram inferir o comportamento do índice de fluidez quando uma mesma resina estava sendo produzida em regime permanente.

Dado que o presente trabalho faz uso de redes neurais artificiais como ferramenta de modelagem da qualidade de um processo polimérico, os principais conceitos referentes a esta ferramenta serão apresentados de maneira mais detalhada na Seção 2.4.

2.4. Redes neurais artificiais

Para aplicações em problemas de engenharia, uma rede neural artificial é usualmente vista como uma caixa preta que aceita entradas, processa estas entradas de acordo com uma função de transferência não-linear, e produz saídas (Power e Bahri, 2004). No entanto, uma rede neural artificial pode ser definida como uma máquina cujo objetivo é modelar o comportamento do cérebro animal (Haykin, 1999). Desta forma, uma rede neural artificial pode ser vista como um processador, composto por várias unidades de processamento paralelas, que possui uma propensão natural a armazenar conhecimento experimental e disponibilizar o mesmo para uso. Neste contexto, redes neurais artificiais

assemelham-se ao cérebro animal em dois aspectos: a rede adquire conhecimento do ambiente em sua volta através de um processo de aprendizado; pesos sinápticos, que representam as forças de conexões entre neurônios, são usados para armazenar o conhecimento adquirido. O procedimento através do qual uma rede neural artificial adquire conhecimento é denominado de algoritmo de aprendizado, e a sua função é modificar os pesos da rede de forma que um determinado objetivo seja atingido.

O funcionamento de uma rede neural artificial está diretamente relacionado às suas unidades de processamento de informação, denominadas de neurônios. A Figura 2 apresenta uma ilustração esquemática de um neurônio artificial, cujo modelo constitui a base para o desenvolvimento de redes neurais artificiais. Nesta figura, pode-se identificar três elementos básicos do modelo de um neurônio: conexões e pesos (w), uma função de adição (Σ) e uma função de ativação (Φ). O funcionamento do modelo em questão pode ser descrito em função destes três elementos da seguinte forma: primeiramente, um sinal de entrada x_j é multiplicado pelo peso w_{kj} , onde k é o índice do neurônio e j é o índice do sinal de entrada. Em seguida, os sinais de entrada são combinados linearmente através da função de adição. O resultado da adição é, então, submetido a uma função de ativação, cujo objetivo é limitar a amplitude do valor de saída do neurônio. Cabe destacar que o modelo apresentado inclui também um valor de referência externo (b_k), cujo objetivo é aumentar ou diminuir o valor de entrada na função de ativação.

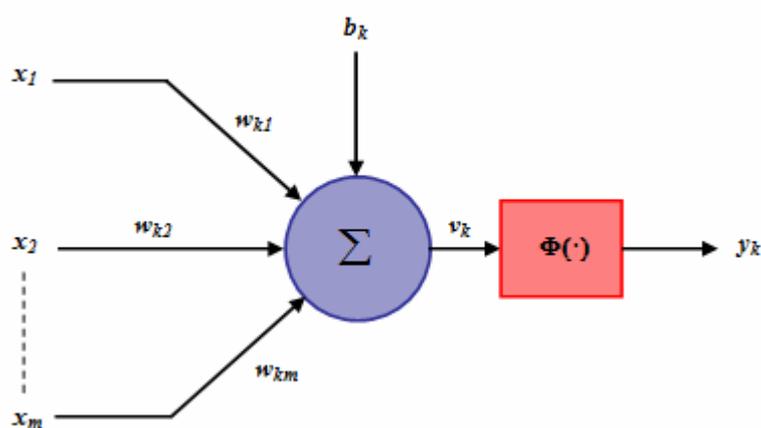


Figura 2 – Modelo esquemático de um neurônio artificial (adaptado de Haykin, 1999)

O funcionamento do neurônio apresentado na Figura 2 pode ser descrito matematicamente através das seguintes equações:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (2.2)$$

$$y_k = \Phi(v_k) \quad (2.3)$$

Cabe salientar que as equações (2.2) e (2.3) são aplicáveis quando o valor de referência (b_k) é considerado como um dos sinais de entrada da rede. O uso destas equações geralmente pressupõe que $x_0 = 1$ e $w_{k0} = b_k$.

De maneira geral, redes neurais artificiais podem ser classificadas através da sua função de ativação, da sua arquitetura e da sua estratégia de aprendizado. Cada uma destas três propriedades será descrita em maior detalhe nas seções 2.4.1, 2.4.2 e 2.4.3, respectivamente.

2.4.1. Função de ativação

A função de ativação (Φ) usada no modelo de um neurônio desempenha um papel primordial na determinação do valor de saída do modelo. Bakshi e Stephanopoulos (1993) propõem a existência de dois tipos de funções de ativação. O primeiro tipo é denominado de função de ativação global e o seu efeito é de gerar uma saída ativa para um amplo intervalo de valores de entrada, fornecendo, portanto, uma aproximação global para os dados de entrada. O segundo tipo, denominado de função de ativação local, produz uma saída ativa apenas na vizinhança imediata dos seus dados de entrada. As Tabelas 1 e 2 apresentam algumas das funções de ativação globais e locais, respectivamente.

Função linear	$\Phi(v_k) = v_k$
Função degrau limiar	$\Phi(v_k) = +\zeta$ para $v_k > 0$ $\Phi(v_k) = -\zeta$ para $v_k \leq 0$
Função sigmóide	$\Phi(v_k) = \frac{1}{1 + e^{-(v_k)}}$
Função tangente hiperbólica	$\Phi(v_k) = \tanh(v_k) = \frac{e^{(v_k)} - e^{-(v_k)}}{e^{(v_k)} + e^{-(v_k)}}$

Tabela 1 – Funções de ativação globais

<i>Função gaussiana</i>	$\Phi(v_k) = \exp\left(-\frac{v_k^2}{2\sigma^2}\right)$ <p>onde σ = desvio-padrão da gaussiana</p>
-------------------------	--

Tabela 2 – Função de ativação local

2.4.2. Arquitetura

A arquitetura de uma rede neural artificial diz respeito à direção das conexões existentes na rede. De maneira geral, redes neurais artificiais podem ser classificadas como *feedforward* ou recorrentes no que diz respeito à sua arquitetura (Pacella e Semeraro, 2007). As conexões entre neurônios feitas de maneira *feedforward* permitem que o fluxo de informações se dê em uma única direção, indo necessariamente dos neurônios de entrada para os neurônios de saída. Em redes recorrentes, dois tipos de conexões (*feedforward* e *feedback*) permitem que a informação se propague nas duas direções, isto é, tanto dos neurônios de entrada para os de saída, quanto dos neurônios de saída para os de entrada.

2.4.3. Estratégia de aprendizado

O aprendizado de uma rede neural artificial pode se dar de maneira supervisionada ou não-supervisionada (Gurney, 1997). No primeiro caso, usa-se primeiramente um conjunto de dados de treinamento, cujo objetivo é fornecer padrões de valores de entrada e de saída. Terminado o treinamento da rede, a mesma encontra-se apta a calcular os valores de saída correspondentes a outros conjuntos de dados de entrada. No caso do aprendizado não-supervisionado, os dados de entrada são separados em grupos de acordo com algum critério de semelhança. A medida de semelhança é geralmente feita através de uma métrica para cálculo de distância, de modo que sempre que um dado não seja semelhante a nenhuma das classes previamente formadas, forma-se uma nova classe e o dado em questão torna-se o padrão de referência da nova classe.

Visto que o presente trabalho faz uso de redes neurais artificiais de aprendizado supervisionado (comumente denominadas de redes neurais artificiais

supervisionadas), alguns dos conceitos mais relevantes à compreensão deste tipo de aprendizado serão discutidos em maior detalhe na seção 2.4.3.1.

2.4.3.1. Aprendizado supervisionado

O aprendizado supervisionado também é denominado de aprendizado com professor (Haykin, 1999). Com o intuito de compreender o funcionamento deste tipo de aprendizado, pode-se imaginar que existe um sistema formado por um professor e uma rede neural artificial, e que este sistema é afetado pelo ambiente externo (Figura 3). Neste contexto, diz-se que o professor detém conhecimento sobre o ambiente externo, o qual pode ser representado por um conjunto de dados de entrada e saída, enquanto que a rede neural artificial não possui qualquer conhecimento sobre o ambiente. Ao ser exposta ao ambiente (i.e. a um determinado conjunto de dados de treinamento), a rede neural artificial recorre ao conhecimento do professor para gerar uma resposta. A diferença entre a resposta desejada e a resposta calculada pela rede é, então, denominada de erro.

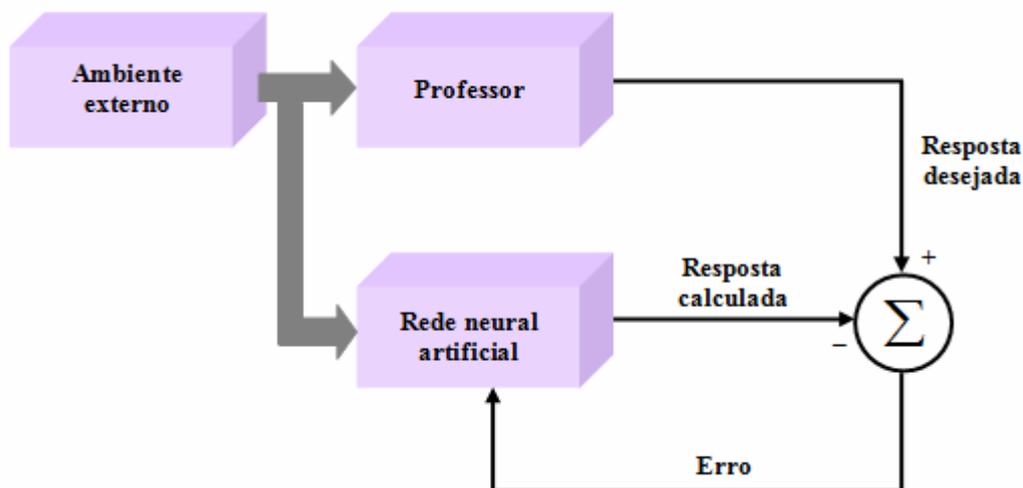


Figura 3 – Representação esquemática do aprendizado com professor (adaptado de Haykin, 1999)

O aprendizado da rede depende tanto da influência dos dados de treinamento, quanto dos valores de erro. O objetivo do treinamento é, portanto, transferir o conhecimento do professor sobre o ambiente para a rede, ou seja, minimizar o valor do erro. Quando o objetivo for atingido, ou seja, quando a rede

tiver concluído seu aprendizado, pode-se dispensar o professor e deixar apenas a rede neural artificial em contato com o ambiente externo.

O aprendizado supervisionado pode ser automatizado através de algoritmos de treinamento. O algoritmo de retropropagação do erro (*error back-propagation algorithm*) é um dos algoritmos mais utilizados para o treinamento supervisionado. De fato, estima-se que mais de 90% das aplicações de redes neurais artificiais utilizam o algoritmo de retropropagação do erro para o treinamento da rede (Fu *et al.*, 2005). Devido tanto à sua importância, quanto ao fato de ter sido utilizado no presente estudo, o algoritmo de retropropagação do erro será apresentado em maiores detalhes na seção 2.4.3.1.1.

2.4.3.1.1.

Treinamento através da retropropagação do erro

Redes neurais artificiais treinadas através da retropropagação do erro devem possuir arquitetura *feedforward* e funções de ativação continuamente diferenciáveis (De Souza Jr., 1993). Apesar de existirem várias funções de ativação não-lineares adequadas a este método de treinamento, a função sigmoideal é comumente a mais usada (Mendes, 2005). Isto se deve, em parte, ao fato de que o uso de um número suficiente de neurônios sigmoideais permite que qualquer função contínua seja aproximada com qualquer grau de acurácia por uma rede neural artificial treinada por retropropagação do erro (Hecht-Nielsen, 1990).

De modo a ilustrar o procedimento de treinamento supervisionado através da retropropagação do erro, faz-se referência à rede neural artificial cuja configuração encontra-se apresentada na Figura 4. Cabe destacar que a rede neural artificial em questão possui três camadas de neurônios e uma arquitetura *feedforward*. A primeira camada da rede é denominada de camada de entrada e, neste caso, possui $n+2$ neurônios. O último neurônio pertencente à camada de entrada é denominado de neurônio *bias* e sua função é adicionar um parâmetro (o peso de *bias*) ao modelo da rede. A segunda camada da rede é denominada de camada escondida apresenta, neste caso, $m+1$ neurônios escondidos e um neurônio de *bias*. A última camada da rede é denominada de camada de saída e, no caso da rede considerada, apresenta apenas um neurônio. Convém mencionar que não existe uma relação pré-estabelecida entre o número de neurônios na(s) camada(s) escondida(s) e o número de neurônios na camada de entrada. Na

realidade, estes dois valores devem ser ajustados levando-se em consideração o tipo de problema que será modelado.

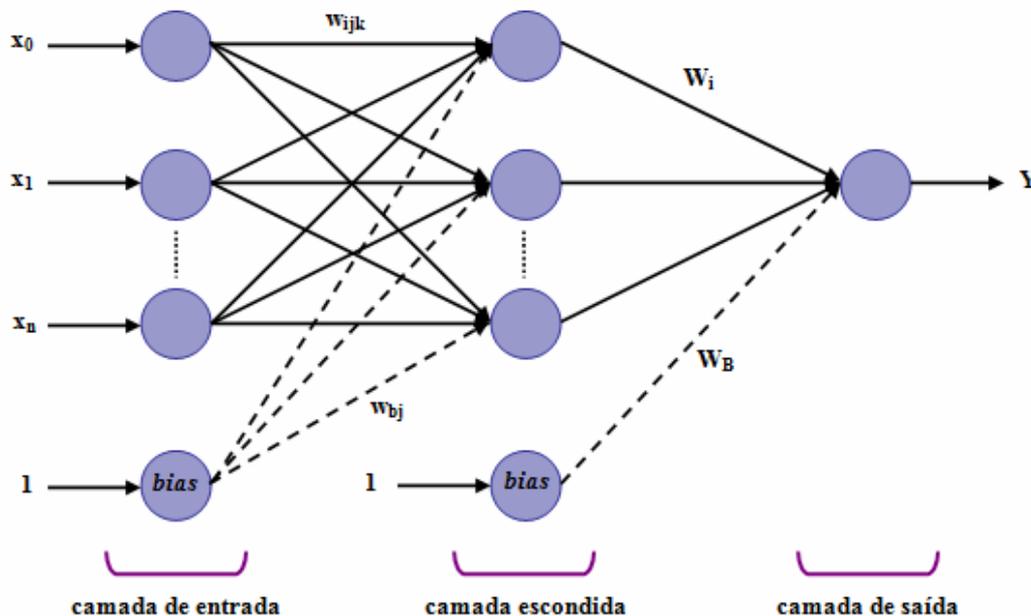


Figura 4 – Representação esquemática de uma rede neural artificial de arquitetura *feedforward* (adaptado de Haykin, 1999)

Os neurônios de entrada recebem, cada um, valores de entrada representados por x_0, x_1, \dots, x_n , onde o índice numérico representa o índice do neurônio de entrada. Os neurônios de entrada estão conectados aos neurônios da camada escondida através dos pesos w_{ij} , onde esta notação representa o peso do neurônio i da camada de entrada, que está conectado ao neurônio j da camada escondida. Os neurônios de *bias* recebem comumente um valor de entrada constante e igual a um. O neurônio de *bias* da camada de entrada está conectado aos neurônios da camada escondida através dos pesos w_{bj} , onde j representa o índice do neurônio da camada escondida ao qual o neurônio de *bias* está conectado. O neurônio de *bias* da camada escondida está conectado ao neurônio de saída através do peso W_B . Os neurônios da camada escondida estão conectados ao neurônio de saída através dos pesos W_i , onde i representa o índice do neurônio da camada escondida. Finalmente, o neurônio de saída gera um valor de saída representado por Y .

O treinamento da rede apresentada na Figura 4 através da retropropagação do erro pode ser feito através dos seguintes passos (Haykin, 1999):

1º Passo – Inicialização

Todos os pesos devem ser inicializados com valores pequenos, de forma a evitar que ocorra a paralisia da rede durante o treinamento. Haykin (1999) sugere que esta inicialização seja feita a partir de uma distribuição com média zero e cuja variância faça com que o desvio-padrão dos campos locais de indução dos neurônios fique na faixa de transição entre a parte linear e a parte saturada da função de ativação sigmoidal. Baughman e Liu (1995) recomendam inicializar os pesos com valores randômicos entre 0 e 1, enquanto que Fu e colaboradores (2005) sugerem inicializar os pesos com valores randômicos entre -0,1 e 0,1.

2º Passo – Apresentação dos exemplos de treinamento

Existem dois possíveis modos de apresentação dos exemplos de treinamento. O primeiro é denominado de modo seqüencial, enquanto que o segundo é denominado de modo batelada. Quando se realiza uma apresentação seqüencial dos exemplos de treinamento, deve-se alimentar a rede, em cada época, i.e. iteração, com um valor de cada uma das variáveis de entrada (x_0, x_1, \dots, x_n). Deve-se, então, prosseguir para os passos 3 e 4 e, ao fim do passo 4, retornar ao segundo passo. A apresentação dos exemplos de treinamento deve continuar até que todos os dados tenham sido apresentados. Neste ponto, considera-se terminada a primeira época e o procedimento é repetido para as épocas subseqüentes.

Neste procedimento, é feita uma atualização seqüencial dos pesos, ou seja, todos os pesos são atualizados após cada apresentação. Desta forma, em uma única época, os pesos serão atualizados tantas vezes quantas forem o número de apresentações dos exemplos de treinamento. Cabe ainda destacar que o número de apresentações corresponde ao número de dados de entrada que serão alimentados a qualquer um dos neurônios de entrada. Apesar deste procedimento ser de simples implementação e proporcionar soluções eficientes para problemas grandes e difíceis, o seu uso não garante a convergência do algoritmo de treinamento (Haykin, 1999).

Por outro lado, quando os exemplos de treinamento são apresentados em batelada, pode-se garantir a convergência do algoritmo a um mínimo local (Haykin, 1999). Neste caso, a atualização dos pesos é feita depois que todos os

exemplos de treinamento são apresentados, isto é, os pesos são atualizados uma única vez em cada época.

3º Passo – Cálculo forward

Nesta etapa do treinamento, deve-se realizar, a cada apresentação p , a seqüência de cálculos apresentada na Tabela 3, onde Φ é uma função de ativação e d corresponde à resposta desejada. É importante mencionar que a função de ativação utilizada na camada escondida não precisa ser necessariamente igual à função de ativação usada na camada de saída.

<i>Cálculos realizados</i>	<i>Equação utilizada</i>
1. Cálculo de v	Primeira apresentação: $v_j^0 = \sum_i (x_i^0 w_{ij}^0) + w_{bj}^0$ Demais apresentações: $v_j^p = \sum_i (v_i^{p-1} w_{ij}^p) + w_{bj}^p$
2. Cálculo de y	$y_j^p = \Phi(v_j^p)$
3. Cálculo de V	$V^p = \sum_j (y_j^p W_j^p) + W_B^p$
4. Cálculo de Y	$Y^p = \Phi(V^p)$
5. Cálculo do erro	$e = d^p - Y^p$

Tabela 3 – Seqüência de cálculos realizada a cada apresentação durante a etapa de cálculo *forward* do treinamento por retropropagação do erro

O esquema apresentado na Figura 5 visa a auxiliar a compreensão da etapa de cálculo *forward* do treinamento por retropropagação do erro.

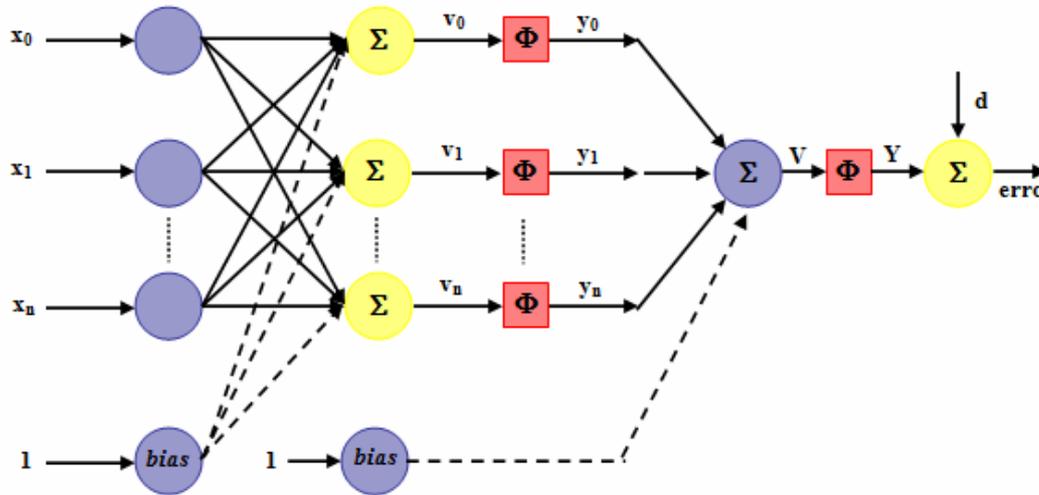


Figura 5 – Representação esquemática da etapa de cálculo *forward* do treinamento por retropropagação do erro

4º Passo – Cálculo backward

Quando os exemplos de treinamento são apresentados de modo seqüencial, deve-se realizar, a cada apresentação p , a seqüência de cálculos apresentada na Tabela 4, onde δ representa o gradiente local da rede, Φ' é a derivada da função de ativação, α é a constante de momento, η é denominado de parâmetro de aprendizado e e^p é o erro calculado na etapa do cálculo *forward*.

Por outro lado, quando se utiliza o modo batelada de apresentação de exemplos de treinamento, o cálculo *backward* só é realizado depois que todos os exemplos de treinamento foram submetidos ao cálculo *forward*. Desta forma, a seqüência de cálculos apresentada na Tabela 4 é realizada uma única vez em cada época.

<i>Cálculos realizados</i>	<i>Equação utilizada</i>
1. Cálculo dos gradientes locais da rede	Camada de saída: $\delta^p = \Phi'(V^p)e^p$ Camada escondida: $\delta_j^p = \Phi'(v_j^p)\delta^p W_j^p$
2. Atualização dos pesos	Camada de saída: $W_i^{p+1} = W_i^p + \alpha W_i^{p-1} + \eta \delta^p Y^p$ Camada escondida: $w_{ijk}^{p+1} = w_{ijk}^p + \alpha w_{ijk}^{p-1} + \eta \delta_j^p y_j^p$

Tabela 4 – Seqüência de cálculos realizada a cada apresentação durante a etapa de cálculo *backward* do treinamento por retropropagação do erro

Neste ponto, cabe uma breve discussão sobre os parâmetros α e η utilizados nas equações de atualização dos pesos. O parâmetro de aprendizado η é um número positivo que controla a taxa através da qual o valor dos pesos é alterado. Quanto mais baixo for o valor de η , mais épocas serão necessárias para minimizar o valor do erro. Por outro lado, quanto maior o valor de η , maior a tendência do treinamento se tornar instável e não conseguir, conseqüentemente, atingir o erro desejado (Baughman e Liu, 1995). A constante de momento α é um parâmetro adicional utilizado para acelerar o ajuste dos pesos. Quanto maior for o valor da constante de momento, menor será o tempo de treinamento necessário para atingir um determinado valor de erro. No entanto, existe um valor de α a partir do qual o algoritmo de treinamento começa a oscilar (Baughman e Liu, 1995). De maneira geral, recomenda-se, sempre que possível, diminuir o parâmetro de aprendizado e aumentar a constante de momento ao longo do treinamento (Haykin, 1999).

5º Passo – Iteração

O procedimento apresentado anteriormente (passos 2 a 4) deve ser repetido até que o critério de parada seja atingido. É importante mencionar que o objetivo do treinamento por retropropagação do erro é minimizar o erro da rede neural artificial. Dado que o erro médio quadrático da rede geralmente decresce com o aumento do número de épocas de treinamento, a dificuldade reside em determinar o momento em que o treinamento deve ser encerrado, isto é, o número máximo de iterações que serão realizadas durante o treinamento da rede (Haykin, 1999).

Neste sentido, convém considerar que o treinamento da rede seja terminado quando a mesma apresentar desempenho máximo. Visto que o desempenho da rede está diretamente relacionado à sua capacidade de generalização, é possível utilizar uma medida da capacidade de generalização da rede como critério de parada do treinamento. Este procedimento é denominado de parada antecipada (*early-stopping*) e necessita da existência de um conjunto de dados de validação. O objetivo deste procedimento é validar o desempenho da rede neural artificial com dados que sejam diferentes dos dados utilizados para atualizar os pesos da rede durante o seu treinamento. Neste caso, utiliza-se como critério de parada a minimização do erro calculado para os dados de validação.

No Capítulo 3, será apresentada uma descrição do processo industrial de produção de polietileno de baixa densidade. Além disso, serão detalhadas as atividades realizadas durante o pré-processamento dos dados operacionais, bem como as configurações de redes neurais artificiais consideradas neste estudo.