3 REDES NEURAIS ARTIFICIAIS

Este capitulo apresenta uma descrição sucinta da teoria básica de Redes Neurais Artificiais e sobre a criação do Comitê de Redes Neurais. Se o leitor estiver familiarizado com os fundamentos de RNAs pode saltar as seções 3.1 a 3.6, passando direito à seção 3.7 que introduz as características sobre o Comitê de RNAs cujo objetivo é obter, através da criação de Redes especializadas um desempenho superior ao de uma rede única.

3.1.O que é uma Rede Neural Artificial

A grande capacidade de processamento de informações do cérebro humano tem motivado pesquisas no sentido de encontrar modelos que reproduzam suas características computacionais, que são totalmente diferentes do computador digital convencional, possibilitando, desta forma, que se realizem certas tarefas de uma maneira semelhante ao cérebro humano. O cérebro é um sistema de processamento de informação (computador) altamente complexo, não-linear e paralelo. Ele é constituído, basicamente, de unidades estruturais elementares, chamadas de neurônios ou unidades de processamento, que podem apresentar diversas entradas e uma saída estando macissamente conectados uns com os outros na composição de uma rede neural, cuja definição pode ser vista no texto abaixo [HAYKIN, 1999]:

Uma *rede neural* é um sistema paralelo, distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. Ela 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;
- Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

As Redes Neurais Artificiais RNAs foram desenvolvidas tomando-se como base o cérebro humano. Elas fazem uma representação distribuída da

informação, na forma de conexões entre um grande número de elementos simples (neurônios artificiais). Todos esses elementos realizam operacionalmente a mesma função, conforme será visto adiante, que é executar a soma ponderada de suas entradas e executar uma transformação (linear ou não-linear) sobre este valor. Assim, as Redes Neurais Artificiais são modelos matemáticos dos neurônios biológicos e suas interconexões em redes.

A grande vantagem no uso de redes neurais artificiais para a solução de problemas complexos provém de algumas propriedades e capacidades úteis, descritas a seguir:

- *Aprendizagem*: É a habilidade da RNA de aprender automaticamente, o mapeamento desejado, entre as entradas e a saída, através de um processo iterativo de ajustes aplicados aos seus parâmetros livres (ex.: pesos sinápticos);
- *Generalização*: Corresponde à capacidade da RNA de apresentar uma saída adequada para uma entrada não presente no processo de aprendizagem;
- *Não-Linearidade*: Uma RNA é não-linear se esta for constituída de neurônios artificiais também não-lineares. Esta é uma característica importante, pois a maioria dos sistemas físicos responsáveis pela geração do mapeamento entre as sinais de entrada e a saída desejada são não-lineares;
- *Adaptabilidade*: É a capacidade que as RNAs possuem de adaptar seus pesos sinápticos perante a modificações no meio ambiente, ou seja, uma RNA treinada para operar em um ambiente específico pode facilmente ser re-treinada para absorver pequenas alterações no ambiente;
- *Tolerância a Falhas*: o conhecimento é distribuído pela RNA; desta forma, uma parte das conexões pode estar inoperante, sem mudanças significativas no desempenho de toda a RNA;
- *Resposta a Evidências*: Em sua utilização como classificadora de padrões, uma RNA pode fornecer, em sua saída, não somente a informação relativa a qual conjunto a entrada pertence, mas também uma informação sobre a confiança no resultado. Desta forma, essas informações podem ser utilizadas para rejeitar padrões ambíguos.

Estas características dotam as redes neurais artificiais da capacidade de resolver problemas complexos que não podem ser resolvidos de forma tradicional.

3.2.Modelo do neurônio

A unidade básica de processamento de uma rede neural artificial é o neurônio. Sua modelagem é inspirada no neurônio biológico, cuja figura representativa pode ser vista na Figura 6. Nesta podem ser vistas as partes constituintes do neurônio que são descritas a seguir:

- Os dendritos são os elementos receptores, as entradas do neurônio;
- O axônio é a linha de transmissão que transporta o sinal de saída do neurônio;
- As sinapses são as regiões onde a saída de um neurônio e a entrada de outro entram em contato. O tipo mais comum de sinapse no cérebro é a sinapse química, onde um processo pré-sináptico libera uma substância química transmissora que se difunde na junção entre os neurônios e então atua em um processo pós-sináptico. Logo, a sinapse converte um sinal elétrico pré-sináptico em um sinal químico e então de volta em um sinal elétrico pós-sináptico. A sinapse pode impor ao neurônio receptivo excitação ou inibição;
- O corpo celular é responsável pelo ‘processamento’ dos sinais de entrada do neurônio. Quando os valores das entradas atingem um determinado limiar, o neurônio ‘dispara’ liberando um impulso elétrico que flui do corpo celular para o axônio, que pode estar conectado à entrada de outro neurônio [HAYKIN, 1999].

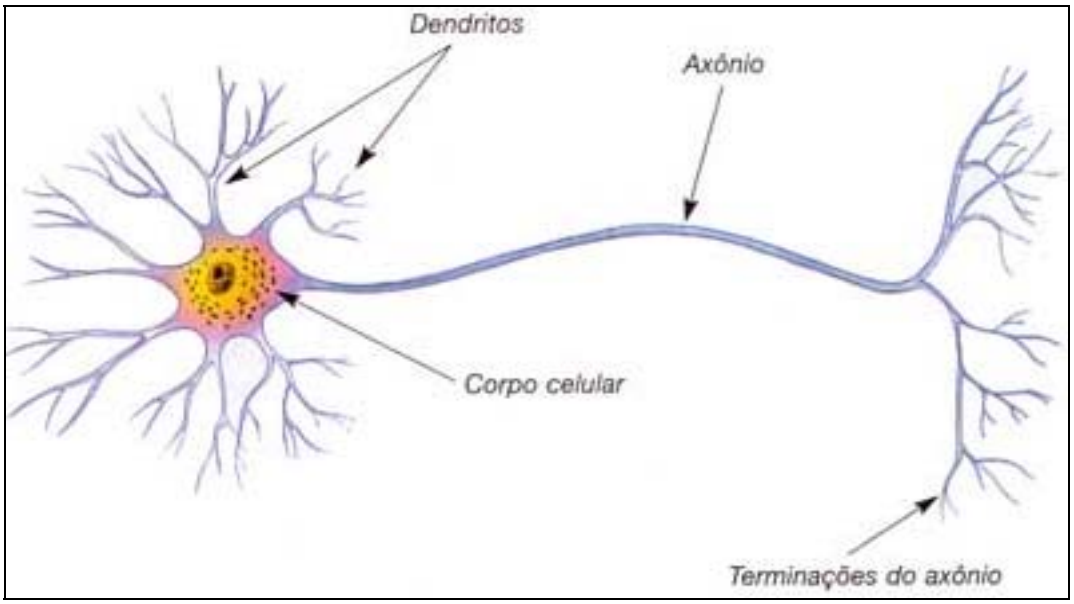

Figura 6 - Neurônio biológico.

A partir desse modelo simplificado do neurônio biológico, foram desenvolvidos modelos para o neurônio artificial, buscando reproduzir as características do neurônio biológico. Um dos trabalhos pioneiros foi o de Warren S. McCullogh e Walter Pitts, intitulado *A Logical Calculus of the Ideas Immanent in Nervous Activity*, que, em 1943, propuseram um modelo matemático para o neurônio. O neurônio tinha um número finito de entradas e uma saída.

O modelo geral do neurônio pode ser visto na Figura 7. Este modelo não mais apresenta unicamente a função de ativação limiar utilizada no neurônio de McCulloch & Pitts, mas sim uma função de ativação $f(net_j)$. Sua operação pode ser resumida da seguinte forma:

1. Os sinais de entrada apresentados às entradas s_i;
2. Cada sinal de entrada é multiplicado por um número w_{ji}, ou peso, que indica a sua influência na saída do neurônio (efeito de excitação ou inibição da sinapse);
3. É feita a soma ponderada dos sinais, produzindo um nível de atividade (corpo celular);
4. Se este nível de atividade exceder um certo limiar, a unidade 'ativa' sua saída s_j;
5. O bias θ_j tem o efeito de aumentar ou diminuir a entrada liquida da função de ativação.

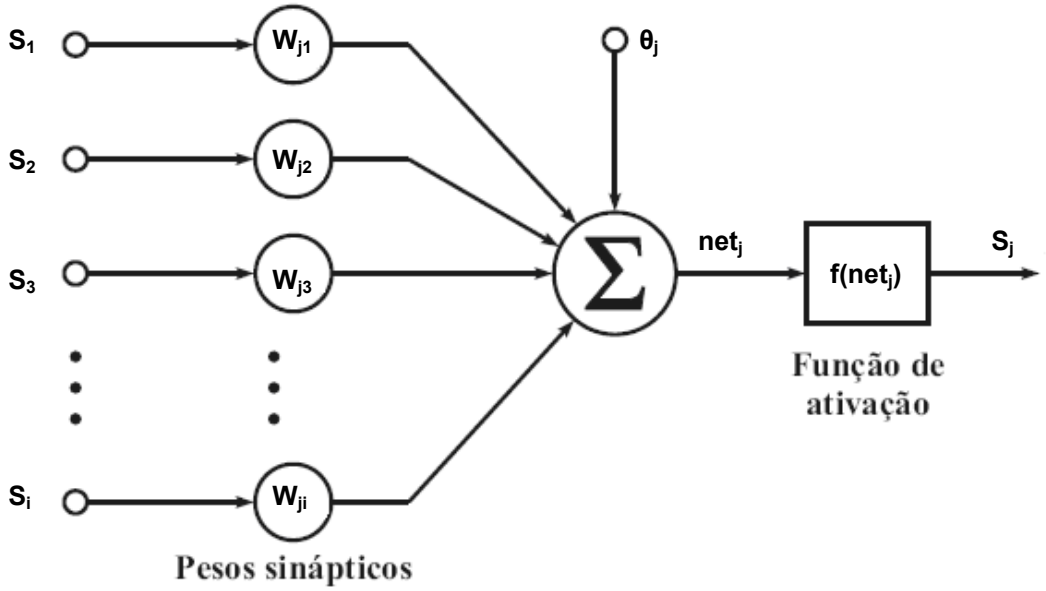

Figura 7 - Modelo matemático do neurônio.

O funcionamento deste neurônio pode ser descrito matematicamente pelas seguintes equações:

$$net_j = \sum S_i \cdot w_{ji} + \theta_j \qquad \text{eq. (2)}$$

$$S_j = f\left(net_j\right) \qquad \text{eq. (3)}$$

Onde:

i é o índice das entradas do neurônio;

θ_j é o bias aplicado ao neurônio *j*;

net_j chamado de campo local induzido, é a saída do combinador linear somado ao bias do neurônio *j*;

S_i são os sinais de entrada do neurônio;

w_{ji} são os pesos sinápticos do neurônio *j*;

$f(net_j)$ é a função de ativação do neurônio *j*, para limitar a amplitude da saída do neurônio. A função de ativação limita a faixa de amplitude permitida do sinal de saída a algum valor finito;

S_j é o sinal de saída do neurônio.

3.3.Tipos de Função de Ativação

A função de ativação f(net$_j$) é a que processa o sinal *net$_j$* para produzir a saída final do neurônio, *S$_j$*. Segundo HAYKIN (1999), existem quatro tipos básicos de função de ativação utilizados em RNAs, conforme são descritas a seguir:

(a) Função de limiar, utilizada no neurônio de McCulloch & Pitts, limita a saída do neurônio a apenas dois valores (binário: 0 ou 1, ou bipolar: -1 ou 1). Normalmente é utilizada para criar neurônios que tomem decisões binárias, como nos classificadores, possui a seguinte definição:

$$f(net) = \begin{cases} 1, & se \quad net \geq 0 \\ 0, & se \quad net < 0 \end{cases} \qquad \text{eq. (4)}$$

(b) Função linear por partes, possui a seguinte definição:

$$f(net) = \begin{cases} 1, & se \quad net \geq 1/2 \\ net, & se \quad -1/2 < net < +1/2 \\ 0, & se \quad net < -1/2 \end{cases} \qquad \text{eq. (5)}$$

(c) Função sigmóide, é a função geralmente adotada em redes neurais em virtude de ser contínua, monotônica, não linear e facilmente diferenciável em qualquer ponto. Possui a seguinte definição:

$$f(net) = \frac{1}{1 + e^{(-a*net)}} \qquad \text{eq. (6)}$$

onde *a* é o parâmetro de inclinação da função.

(d) Função tangente hiperbólica, as funções de ativação definidas nas Eqs. 4, 5 e 6 se estendem de 0 a +1. Algumas vezes é desejável que a função de ativação se estenda de -1 a +1, assumindo neste caso uma forma anti-simétrica em relação à origem. Possui a seguinte definição:

Para a forma correspondente, utiliza-se a função tangente hiperbólica, definida por:

$$f(net) = \tanh(net) \qquad \text{eq. (7)}$$

A Figura 8 apresenta os gráficos das funções de ativação, demonstrando o comportamento das mesmas.

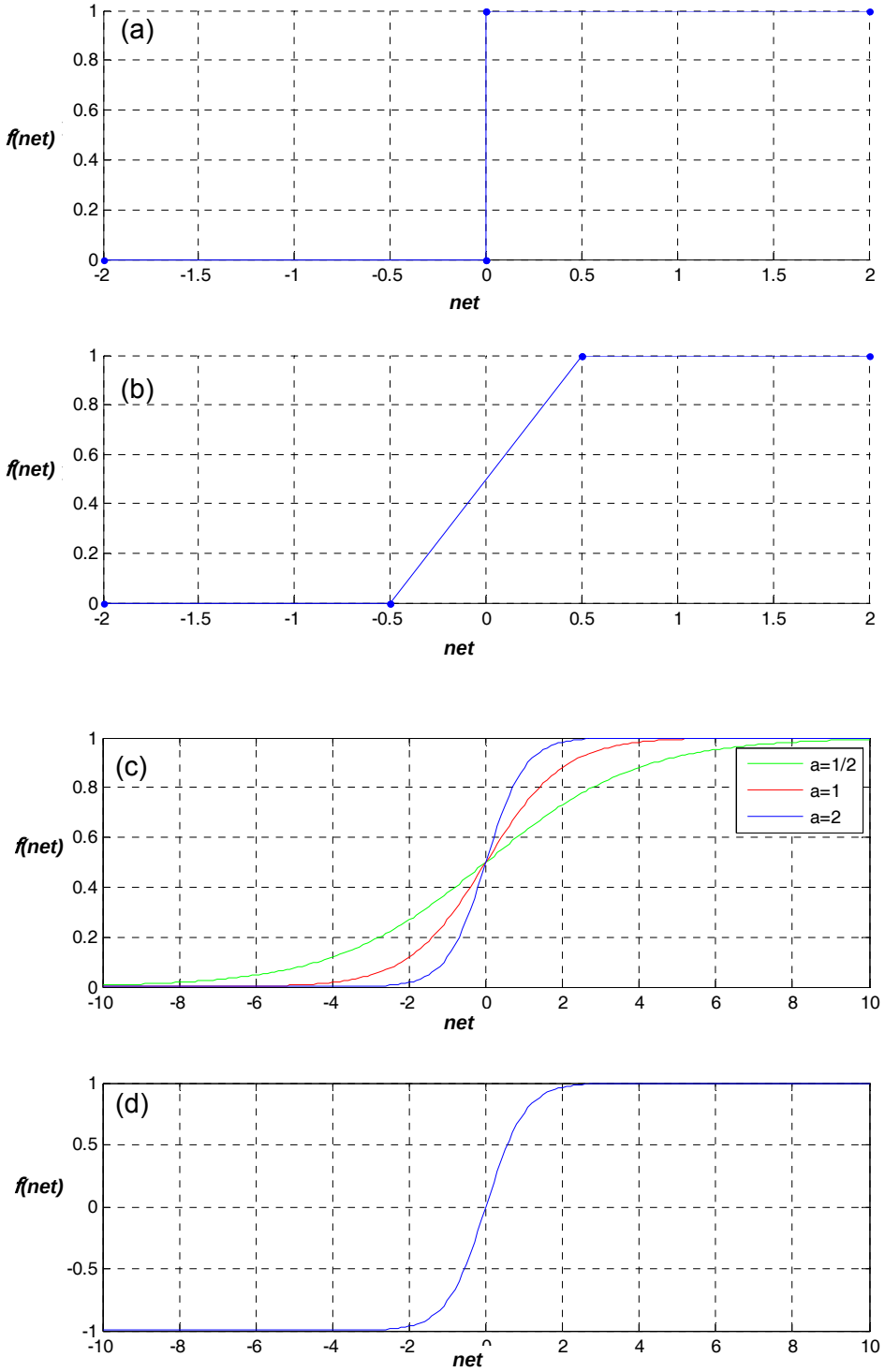

Figura 8 - Funções de ativação mais utilizadas.

3.4.Arquitetura de Redes Neurais

3.4.1.Redes feedforward de Camada Única

Nesta forma mais simples de rede neural os neurônios são organizados em uma única camada. A saída de cada um dos neurônios constitui uma saída da rede. Este tipo de arquitetura pode ser vista na Figura 9. A rede é sempre alimentada adiante, pois a camada de nós fonte fornece os sinais de entrada para a camada de saída e não vice-versa, ou seja, não há laços de realimentação. O termo camada única se refere ao fato de existir apenas uma camada de nós computacionais (neste caso, a camada de saída).

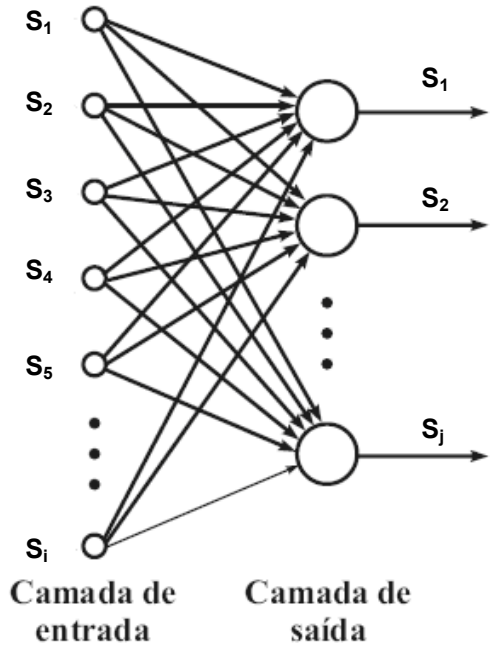

Figura 9 - Rede alimentada adiante de camada única.

3.4.2.Redes feedforward de Múltiplas Camadas

Nesta segunda classe de redes neurais existe a presença de uma ou mais camadas de nós computacionais entre as camadas de entrada e saída, as quais são chamadas de camadas ocultas ou intermediárias e constituídas, por sua vez, de neurônios ocultos. A Figura 10 apresenta esta arquitetura.

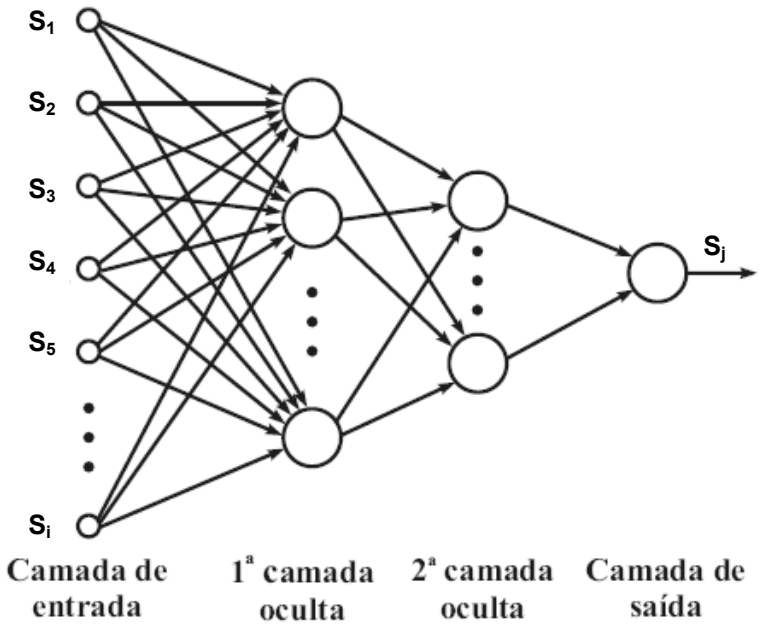

Figura 10 - Rede alimentada adiante de múltiplas camadas.

3.4.3.Redes Recorrentes

As redes neurais recorrentes diferem das redes alimentadas adiante por possuírem pelo menos um laço de realimentação. A Figura 11 apresenta uma rede deste tipo com um neurônio oculto (em cor cinza) e dois neurônios de saída. O processo de treinamento deste tipo de rede neural não será abordado nesta dissertação.

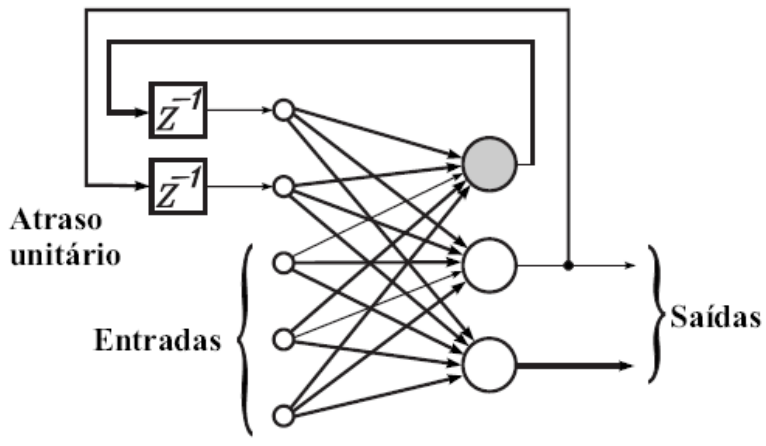

Figura 11 - Rede neural recorrente.

3.5.Paradigmas de Aprendizagem

A propriedade mais importante das redes neurais é a habilidade de aprender acerca de seu ambiente e, com isso, melhorar o seu desempenho. Isto pode ser feito através de um processo iterativo de ajustes aplicados aos pesos sinápticos da rede, chamado de treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma determinada classe de problemas.

Todos os modelos de redes neurais possuem uma regra de treinamento, onde os pesos de suas conexões sinápticas são ajustados de acordo com os padrões apresentados, ou seja, a rede aprende através de exemplos provenientes de casos reais conhecidos. Deste modo, a rede neural extrai regras básicas a partir dos exemplos, diferentemente da programação computacional tradicional (*C, Pascal, Fortran*, etc.), onde é necessário que as regras sejam previamente conhecidas.

A seguir serão apresentadas duas metodologias de aprendizagem, freqüentemente chamadas de paradigmas de aprendizagem.

3.5.1.Aprendizado Supervisionado (com professor)

O aprendizado supervisionado, também chamado aprendizado com um professor, está representado através de um diagrama de blocos na Figura 12.

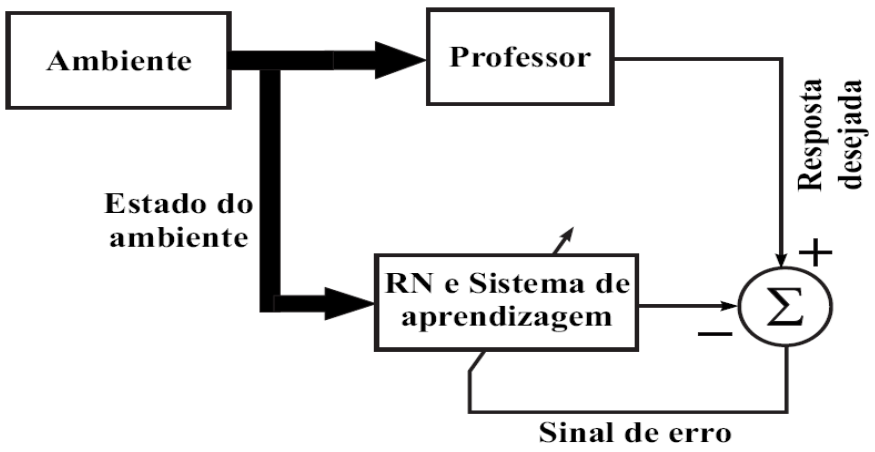

Figura 12 - Aprendizado supervisionado.

No paradigma de aprendizagem com um professor, o estado do ambiente é representado por um vetor que é aplicado à entrada da RNA e ao professor. O professor possui conhecimento sobre o ambiente, o qual é representado por um

conjunto de exemplos de entrada-saída, fornecendo assim, a resposta desejada. O sinal de saída da RNA é então comparado com a resposta desejada, gerando um sinal de erro que é utilizado para ajustar os pesos sinápticos da rede de modo que esta, após o ajuste, apresente em sua saída uma resposta tão próxima quanto possível daquela desejada. Este processo é repetido para cada exemplo de treinamento até que a RNA emule o professor. Desta forma, o conhecimento do professor foi, de certa maneira, transferido para os pesos sinápticos da rede neural, e este não se faz mais necessário, pois a RNA já é capaz de lidar com o ambiente por si mesmo.

3.5.2.Aprendizado Não-supervisionado

Como o próprio nome indica, nesta metodologia de aprendizagem não há um professor responsável pelo fornecimento da resposta desejada, ou seja, não há exemplos da função a ser aprendida pela rede. Um exemplo de rede neural que faz uso desta forma de aprendizagem é o chamado mapa auto-organizável [KOHONEN, 1997].

Para o treinamento da rede não supervisionada são utilizados apenas os valores de entrada, conforme pode ser visto no diagrama de blocos da Figura 13. Neste tipo de aprendizado, a rede utiliza os neurônios como classificadores, e as entradas como elementos a serem classificados, utilizando-se para isso um processo de competição e cooperação entre os neurônios da rede.

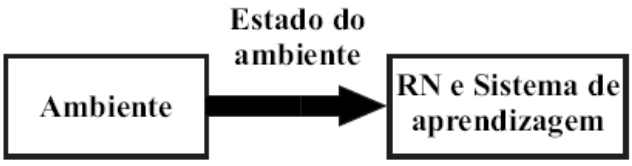

Figura 13 - Aprendizado não supervisionado.

Em sua forma mais simples, o aprendizado não-supervisionado pode ser descrito da seguinte forma: o sinal é aplicado à entrada da RNA e somente o neurônio vencedor (aquele que possui vetor de pesos mais próximo ao padrão de entrada) se tornará ativo, e os demais permanecerão inativos. Os pesos sinápticos deste neurônio são então ajustados de acordo com uma regra de aprendizagem, tendendo assim, ao fim do processo, para um valor próximo do sinal de entrada. Desta forma, sinais de entrada com características semelhantes serão identificados pela ativação do mesmo neurônio. Cada neurônio, ou conjunto de neurônios da rede, será responsável por uma única

classe de padrões de entrada, que deve ser mapeada após a finalização do treinamento.

3.6.Redes Neurais Artificiais Multilayer Perceptron

Dentre os muitos tipos de redes neurais existentes, optou-se pela utilização do modelo *Multilayer Perceptron* (MLP), pela sua facilidade de implementação e por ser um aproximador universal [HORNIK et al., 1989].

As redes MLP apresentam um grande poder computacional devido à inserção de camadas intermediárias, diferentes do modelo Perceptron original de ROSEMBLATT (1960) que possuía apenas um nível de neurônios diretamente conectados à camada de saída.

A solução de problemas não-linearmente separáveis passa pelo uso de redes com uma ou mais camadas intermediárias. A rede passa a conter então pelo menos três camadas: a de entrada, a camada intermediária ou escondida, e uma camada de saída.

Segundo (CIBENKO, 1989) uma rede com uma camada intermediária pode implementar qualquer função contínua, e, com duas camadas intermediárias é possível aproximar qualquer função matemática.

3.6.1.Algoritmo de Aprendizagem Back Propagation

Responsável pelo renascimento da RNA, este algoritmo permitiu que as redes neurais de múltiplas camadas apresentassem capacidade de aprendizado. O *Back Propagation*, baseado no modelo de aprendizado supervisionado, retropropaga o erro da camada de saída até a camada de entrada, permitindo a atualização dos pesos sinápticos entre as camadas intermediárias (ocultas).

Uma arquitetura de rede *Multilayer Perceptron* – MLP tem as seguintes características:

- O fluxo do sinal nesta estrutura é unidirecional, da camada de entrada para a camada se saída;
- O modelo de cada neurônio da rede inclui uma função de ativação não linear e diferenciável em qualquer ponto;
- A rede contém uma ou mais camadas de neurônios ocultos, que não são parte da entrada ou da saída da rede.

O treinamento da rede MLP é o processo que vai fazer com que um conjunto de pesos, determinados inicialmente de maneira aleatória, seja modificado por meio de um algoritmo, de modo que ao final do processo o conjunto de pesos obtido seja útil à solução de um determinado problema. O treinamento começa com a introdução de um conjunto de dados na entrada da rede, para os quais se espera que a saída atribua um valor que já é previamente conhecido. Como este valor de saída é conhecido, então o treinamento é do tipo supervisionado.

Este aprendizado supervisionado é baseado no método do gradiente descendente (*gradient descent*), buscando minimizar o erro global da camada de saída. Deste modo, a atualização do peso (Δw_{ij}) é proporcional ao negativo da derivada parcial do erro com relação ao próprio peso:

$$\Delta\omega_{ij} = -\eta \frac{\partial E}{d\omega_{ij}} \qquad \text{eq. (8)}$$

onde η é a taxa de aprendizado, e E é a função erro definida como:

$$E = \frac{1}{2}\sum_{i=1}^{No}\left(t_j - s_j\right)^2 \qquad \text{eq. (9)}$$

onde N_o é o número de processadores da camada de saída e t_j é o valor esperado na saída do processador j.

Derivando a equação 8, chega-se à seguinte fórmula:

$$\Delta\omega_{ij} = \eta \cdot s_i \cdot e_j$$

$$e_j = \begin{cases} (t_j - s_j) f'(net_j) & se \quad j \in camada \quad de \quad saída \\ f'(net)\sum_{k=1}^{N} \omega_{jk} e_k & se \quad j \in camada \quad escondida \end{cases} \qquad \text{eq. (10)}$$

onde η é a taxa de aprendizado; s_i é a entrada associada ao peso w_{ij}; e_j é o erro do *j*-ésimo processador; t_j é o valor desejado de saída do processador *j*; s_j é o seu estado de ativação; net_j é o seu potencial interno; *f'* é a derivada da função de ativação; e *N* é o número de processadores na camada seguinte à camada do processador *j*.

Como pode-se verificar da equação acima, o algoritmo de aprendizado do *Back Propagation* tem duas fases, para cada padrão apresentado: *Feed-Forward* e *Feed-Backward*. Na primeira etapa as entradas se propagam pela rede, da camada de entrada até a camada de saída, gerando a saída da rede em resposta ao padrão apresentado. Na segunda etapa, os erros se propagam na direção contrária ao fluxo de dados, indo da camada de saída até a primeira

camada escondida, atualizando os pesos sinápticos. Este procedimento de aprendizado é repetido diversas vezes, até que, para todos os processadores da camada de saída e para todos os padrões de treinamento, o erro seja menor do que o especificado.

Apesar do grande sucesso do *Back Propagation* nas mais diferentes aplicações, existem alguns problemas básicos: a definição do tamanho da rede, o longo processo de treinamento, e fenômenos como paralisia da rede (contornado diminuindo o valor de η) e mínimo local (que pode ser solucionado utilizando-se métodos estatísticos).

A definição do tamanho da rede, isto é, o número de camadas escondidas e número de processadores em cada uma dessas camadas, é um compromisso entre convergência e generalização. Convergência é a capacidade da Rede Neural de aprender todos os padrões do conjunto de treinamento. Generalização é a capacidade de responder corretamente aos padrões nunca vistos (conjunto de teste). O objetivo é utilizar a menor rede possível, de forma a se obter uma boa generalização, que seja capaz de aprender todos os padrões.

A taxa de aprendizado η é um parâmetro importante a ser definido no aprendizado. Esta não deve ser nem muito pequena, causando um treinamento muito lento, nem muito grande, gerando oscilações. Quando a taxa de aprendizado é pequena, e dependendo da inicialização dos pesos (feita de forma aleatória), a Rede Neural pode ficar presa em um mínimo local. Quando a taxa de aprendizado é grande, a Rede Neural pode nunca conseguir chegar ao mínimo global pois os valores dos pesos são grandes. A solução para este problema é utilizar uma taxa de aprendizado adaptativa. Além deste parâmetro, pode-se também utilizar um termo de momento [HAYKIN, 1999], proporcional à variação no valor do peso sináptico no passo anterior. Deste modo, a equação de atualização do peso sináptico w_{ij} é modificada da seguinte forma:

$$\Delta w_{ij}(t+1) = -\eta \cdot s_i \cdot e_j + \alpha \times \Delta w_{ij}(t) \qquad \text{eq. (11)}$$

A utilização do termo de momento tem a função de acelerar a convergência da rede, sem causar oscilações.

3.6.2.Generalização em RNAs

A generalização é a capacidade de uma RNA, devidamente treinada, de responder coerentemente a padrões desconhecidos. Segundo Teixeira (2001), a capacidade de generalização não é uma propriedade inerente às RNAs, ou seja,

ela não é facilmente obtida simplesmente submetendo a rede à fase de treinamento.

Alguns fatores devem ser levados em consideração para se obter uma RNA com elevada capacidade de generalização. Basicamente a generalização em uma RNA tem influência dos seguintes fatores:

- Tamanho e representatividade estatística do conjunto de dados de treinamento;
- Arquitetura da rede neural;
- Complexidade física do problema abordado.

Não existe uma regra para escolher o tamanho do conjunto de treinamento. Cada problema abordado requer uma quantidade de amostras capaz de representá-lo. Este parâmetro não é de simples estimativa, dado que o domínio do problema nem sempre é conhecido a priori.

A escolha da arquitetura do modelo neural adequada à complexidade do problema é um dos maiores desafios no estudo da capacidade de generalização. Modelos com arquiteturas muito grandes elevam a complexidade do modelo. Quando a complexidade do modelo é maior que a necessária para modelar o problema, a rede fica super-ajustada aos dados de treinamento, respondendo inadequadamente aos padrões de validação e teste. Este fenômeno de super-ajuste do modelo aos dados de treinamento é comumente chamado de *overfitting* e, reduz a capacidade de um modelo generalizar. Porém se a complexidade do problema supera a complexidade do modelo, este não é capaz de descrever e representar o domínio do problema, caracterizando assim o fenômeno de sub-ajuste ou *underfitting*.

3.6.3.Parada antecipada do treinamento

A parada antecipada do treinamento (Early Stopping) é uma técnica, proposta por Weigend et al. (1990) e citada em Teixeira (2001), baseada na divisão do conjunto de padrões em pelo menos dois conjuntos distintos, mas com mesma representatividade estatística, chamados normalmente de conjuntos de treinamento e validação.

O conjunto de treinamento é o único conjunto a ser usado durante o treinamento para a atualização dos parâmetros da RNA (pesos e termos de polarização).

O conjunto de validação é um conjunto através do qual, durante o treinamento, será também calculado um erro (erro de validação) cuja finalidade é monitorar o nível de ajuste (*fitting*) da RNA aos dados de treinamento.

O erro de validação deve ser monotonicamente decrescente a partir do início do treinamento, que deve ser interrompido no momento em que este erro começar a crescer, embora o erro de treinamento ainda seja decrescente. Este sintoma indica que o treinamento está levando a RNA à uma condição de sobreajuste e para evitá-lo, o treinamento é interrompido e os parâmetros da RNA na época anterior são considerados como os parâmetros finais obtidos com o treinamento.

A Figura 14 mostra o comportamento dos erros de treinamento e de validação no treinamento com parada antecipada '*early stopping*'.

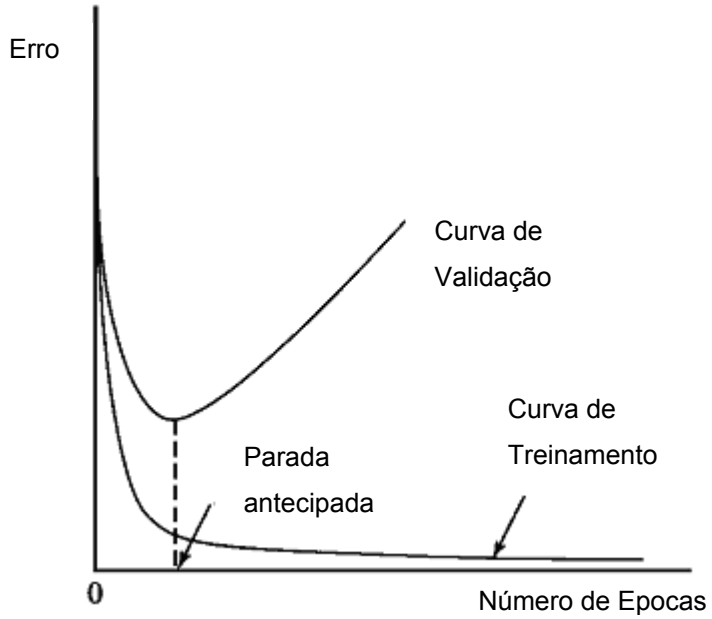

Figura 14 – Monitoramento da métrica de erro de validação ocasionando um 'early stopping'.

Dessa forma, o treinamento usando a estratégia de parada antecipada do treinamento busca elevar a capacidade de generalização do modelo, baseando-se na minimização do erro de validação, que é usado como critério de parada do algoritmo [HAYKIN, 1999].

3.7.Sistemas Multi-Net

Um sistema Multi-Net pode ser modular ou comissionado (ou em '*ensemble*') [SHARKEY, 1999]. São ditos comissionados os sistemas onde as redes integrantes são redundantes entre si (i.e., todas têm a mesma finalidade) mas, por algum motivo, são úteis para combinação de respostas.

São ditos modulares os sistemas onde cada rede exerce papel não redundante para a finalidade em questão, seguindo tipicamente o princípio 'dividir para conquistar'.

Embora esta divisão sugerida pareça adequada, deve-se ressaltar que as duas categorias propostas não devem ser vistas como mutuamente exclusivas – um sistema Multi-Net pode ser construído misturando-se comitês e sub-sistemas modulares [SHARKEY, 1999].

De maneira geral, a utilização dos sistemas Multi-Net tem dois propósitos - aumentar a eficácia de uma solução obtida individualmente ou viabilizar uma solução que antes não podia ser alcançada de maneira eficiente. A partir da próxima seção serão apresentados sistemas comissionados como a formação de comitês.

Para nossa dissertação serão utilizados os sistemas Multi-Net comissionados (ensemble). Um ensemble consiste de um conjunto de regressores ou classificadores que fornecem uma saída global baseada numa combinação das saídas individuais de cada um dos seus membros, com o objetivo de se obter um desempenho que seja superior ao desempenho individual de cada componente.

Segundo Lima (2004), em função dos bons resultados apresentados pelo método, é vasta a gama de aplicações tanto para classificação de padrões quanto para regessão.

Em Sharkey (1999) citado por Castro et al. (2005), os melhores resultados para a predição de séries temporais são obtidos pela combinação de diferentes modelos e não pela seleção do melhor modelo individual.

No caso de ensemble de RNAs, Perrone e Cooper (1993) sugerem que as RNAs componentes do ensemble devam ter diferentes arquiteturas, devam ser treinadas com algoritmos diferentes e tambem com conjuntos de dados de diferentes. Segundo Yang e Linkens (2001), a aplicação desta técnica apresenta melhoria na robustez, capacidade de predição e generalização do modelo neural.

Como citado por Lima (2004), há várias propostas para a geração da saída de um ensemble de RNAs, sendo predominante o voto majoritário para problemas de classificação e média simples, ou média ponderada, para problemas de regressão. Já em Castro et al. (2005) foi implementado um ensemble a partir de três RNAs do tipo MLP treinadas com algoritmos diferentes e a combinação do ensemble foi gerada a partir de um neurônio de saída com função de ativação linear.

3.7.1.Formação de comitês

O comitê que exemplifica um sistema comissionado é aquele que comumente combina um conjunto de componentes sintetizados para executar a mesma tarefa. Neste caso, cada componente representa isoladamente uma solução candidata para o problema de classificação ou regressão como um todo [HANSEM, 1990], podendo cada solução ser obtida por meios distintos e independentes entre si.

Uma das áreas mais ativas de pesquisa em aprendizado supervisionado é o estudo de métodos para construção de comitês de classificadores capazes de produzir ganho de desempenho, quando comparados ao desempenho de seus componentes isolados, independente das especificidades da aplicação [LIMA, 2004]. A figura abaixo exemplifica um sistema comissionado para classificação:

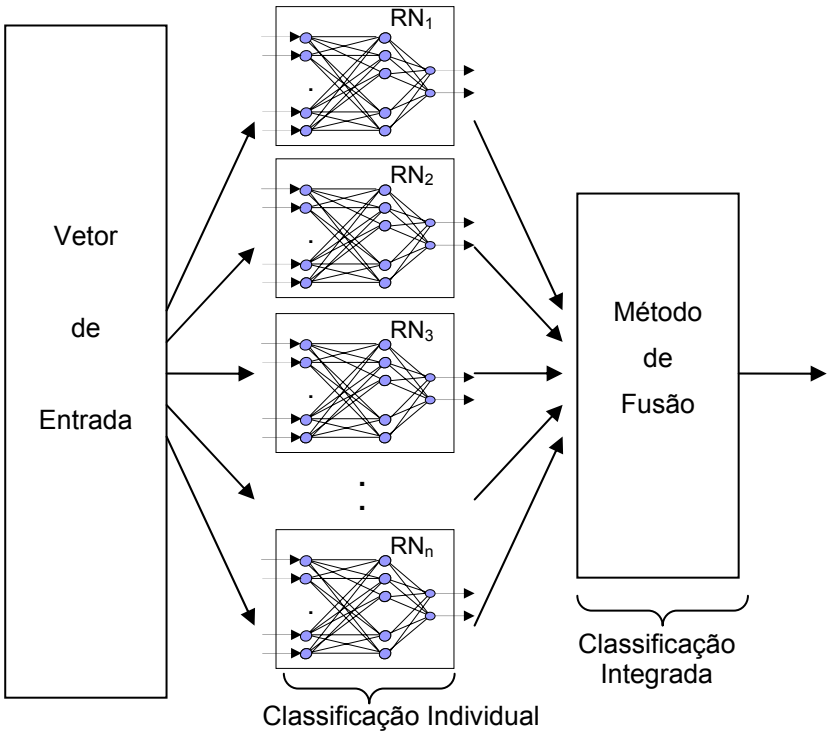

Figura 15 – Comitê de Redes Neurais.

As redes RN_1, RN_2, ..., RN_n são redundantes entre si, mas quando combinadas têm potencial de melhorar o desempenho global do sistema. Em particular, os métodos de formação dos membros (das redes) de um comitê são de fundamental interesse. Eles se reúnem em quatro grandes categorias, dependendo da característica que se varia para obtenção desses membros [SANTOS, 2001]:

- Categoria de variação do conjunto inicial de pesos;
- Categoria de variação da topologia da rede;
- Categoria de variação do algoritmo empregado;
- Categoria de variação do conjunto de treinamento.

A formação dos comitês é feita essencialmente com técnicas de modificação do conjunto de treinamento.

A classificação supervisionada depende de informações a priori sobre as classes a serem analisados. Isso se resume na extração de um conjunto de padrões, chamado conjunto de treinamento, que represente bem a natureza das classes existentes no conjunto total. A partir desse conjunto, são extraídas informações e configurados classificadores, que permitam atribuir, de maneira criteriosa, rótulos para todos os padrões. [SANTOS. 2001]

A classificação então pode ser realizada, e pode ser entendida, de maneira geral, pela partição do espaço de atributos em um número finito de regiões de tal forma que objetos de uma mesma classe recaiam, pelo menos em tese, sempre dentro de uma mesma região.

A próxima seção apresenta os métodos de formação dos membros do comitê pela variação do conjunto de treinamento.

3.7.2.Métodos de variação no conjunto de treinamento

Três métodos de formação de comitês são vistos nas próximas seções – RDP, '*Bootstrap'* e Arc-x4.

3.7.2.1.Replicação Dirigida de Padrões (RDP)

A Replicação Dirigida de Padrões (RDP) no conjunto de treinamento objetiva criar redes especializadas por classe para futura combinação.

No método RDP, para especializar uma rede na classe *k*, basta replicar no conjunto de treinamento original os *n* padrões que representam a classe em questão por um fator inteiro γ>1. Assim, admitindo que o número total de padrões no conjunto original seja *N*, o conjunto modificado terá N+(γ -1)*n padrões.

Para exemplificar o uso do RDP, considere-se como conjunto de treinamento original a seguinte tabela de dados [SANTOS, 2001]:

Tabela 8 – Exemplo de conjunto de treinamento; A, B, C, e D são atributos.

Á	B	C	D	Classe
5.1	3.5	1.4	0.2	1
4.9	3.0	1.4	0.2	1
7.0	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.3	3.3	6.0	2.5	3
5.8	2.7	5.1	1.9	3

O RDP prevê então a construção de três novos conjuntos de treinamento, para a conseqüente formação de três redes candidatas a um comitê. Supondo γ = 2 estes conjuntos ficam como nas tabelas seguintes:

Tabela 9 – Conjunto especializado na classe 1; A, B, C, e D são atributos.

Á	B	C	D	Classe
5.1	3.5	1.4	0.2	1
4.9	3.0	1.4	0.2	1
5.1	3.5	1.4	0.2	1
4.9	3.0	1.4	0.2	1
7.0	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.3	3.3	6.0	2.5	3
5.8	2.7	5.1	1.9	3

Tabela 10 – Conjunto especializado na classe 2; A, B, C, e D são atributos.

Á	B	C	D	Classe
5.1	3.5	1.4	0.2	1
4.9	3.0	1.4	0.2	1
7.0	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
7.0	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.3	3.3	6.0	2.5	3
5.8	2.7	5.1	1.9	3

Tabela 11 – Conjunto especializado na classe 3; A, B, C, e D são atributos.

Á	B	C	D	Classe
5.1	3.5	1.4	0.2	1
4.9	3.0	1.4	0.2	1
7.0	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.3	3.3	6.0	2.5	3
5.8	2.7	5.1	1.9	3
6.3	3.3	6.0	2.5	3
5.8	2.7	5.1	1.9	3

O aumento promovido no conjunto de treinamento pelo RDP tem a desvantagem de aumentar significativamente o tempo de treinamento. Além disso, a determinação do melhor valor de γ para um dado problema é, em princípio, uma questão em aberto, fora do escopo deste trabalho [SANTOS, 2001].

3.7.2.2.Bootstrap

Um comitê formado com '*Bootstrap*' é um sistema onde cada rede neural componente é treinada com uma versão em '*bootstrap*' do conjunto de dados de treinamento.

Uma versão em '*bootstrap*' [EFRON, 1993] [KOHAVI, 1995] de um conjunto de dados com n padrões é um conjunto com também n padrões, criado padrão-a-padrão, por sorteio não-viciado e com reposição, realizado no conjunto original.

Os conjuntos de treinamento formados com o '*Bootstrap*' são completamente aleatórios, mas na prática têm demonstrado resultados consideráveis [OPITZ, 1999] [BREIMAN, 1999]. Uma vantagem deste método sobre a Replicação Dirigida de Padrões RDP é o menor tempo de processamento requerido. Uma desvantagem é a falta de controle sobre a especialização produzida na rede.

A formação de comitês com o '*Bootstrap*' é referida normalmente como '*Bagging*', que é um acrônimo de '*Bootstrap Aggregating*'.

3.7.2.3.Arc-x4

O Arc-x4 é um método adaptativo de formação de conjuntos de treinamento, que leva em consideração pesos diferentes na hora de selecionar os padrões que comporão o conjunto gerado. Ele foi proposto por L. Breiman [BREIMAN, 1996], e funciona assim:

Seja *p(n)* a probabilidade com que o n-ésimo padrão do conjunto inicial de treinamento T possa ser sorteado para construção de um novo conjunto; inicia-se cada *p(n)* = 1/N, sendo N o número total de padrões em T(*n* = {1,..., N}). Então:

1. No passo Q do algoritmo, usando todos os *p(n)*'s, sorteie de T, com reposição, um novo conjunto T_P, com a mesma cardinalidade N, e treine a rede RN_P com ele;
2. Passe T pelas redes RN_1,..., RN_P criadas até então e defina *m(n)* (*n* = {1,..., N}) como o número de erros (classificações erradas) do enésimo padrão do conjunto original.

Definidos todos os *m(n)*'s, passe para o passo Q+1 com a seguinte atualização (equação 12).

$$p(n) = \frac{(1 + m(n)^4)}{\sum_{i=1}^{N} (1 + m(i)^4)} \qquad n = \{1,..., N\} \qquad \text{eq. (12)}$$

Depois de K passos, as redes RN_1,..,RN_K estarão prontas para serem combinadas.

Considere novamente o conjunto de treinamento sugerido na Tabela 8. Supondo que o primeiro padrão seja particularmente de difícil classificação, a

tendência dele, em um método adaptativo de formação de novos conjuntos, é se repetir com mais freqüência à medida que mais conjuntos são gerados.

Então, supondo que tenham sido gerados 4 novos conjuntos, uma possível formação é exemplificada na tabela abaixo [SANTOS, 2001].

Tabela 12 – Conjuntos formados com o Arc-x4 – O padrão 1 é difícil.

Conjunto	CLASSE
1	2, 6, 4, 4, 1, 5
2	5, 3, 2, 1, 1, 4
3	3, 2, 1, 1, 1, 6
4	2, 4, 1, 1, 1, 1

A formação de comitês com técnicas adaptativas é referida normalmente como '*Arcing'*, que é um acrônimo de '*Adaptative Reweighting and Combining*'.

Um outro método '*Arcing*', similar ao Arc-x4 e também bastante citado em trabalhos, é o '*Adaboost'*. Ele também funciona adaptando taxas de probabilidade para um sorteio viciado mas, ao contrário do Arc-x4, se baseia apenas no último classificador gerado para atualização dessas taxas [FREUND, 1996].

3.8.Combinação de Classificadores

Em problemas de classificação de padrões, a busca do melhor desempenho passa pelo teste de diversos esquemas classificadores [XU, 1992]. Mesmo que um dos esquemas tenha melhor resultado, os conjuntos de padrões incorretamente classificados pelos diferentes algoritmos não são necessariamente idênticos; isto sugere que diferentes classificadores, com informações potencialmente complementares, poderiam ser usados para aprimorar o melhor resultado até então obtido [KITTER et al., 1998].

Esta observação gerou o interesse por métodos de combinação de classificadores. De fato, as pesquisas têm demonstrado que um bom comitê é aquele no qual os classificadores são bem acurados individualmente, mas têm erros em partes diferentes do espaço de entrada [OPITZ, 1999].

Combinar múltiplos classificadores para resolver um problema de reconhecimento de padrões é um procedimento conveniente em casos particulares. Alguns deles são apresentados por JAIN et. al. (2000):

1. Acesso a diferentes classificadores, cada um deles desenvolvido em um contexto diferente e utilizando diferentes representações/descrições de um mesmo problema;
2. Disponibilidade de mais de um conjunto de treinamento, coletados em tempo ou ambientes diferentes. Estes conjuntos podem usar ainda diferentes atributos;
3. Diferentes classificadores treinados em um mesmo conjunto de dados que podem não apenas ter desempenhos diferentes, mas apresentar diferenças locais, de forma que cada classificador tenha uma região no espaço de atributos para a qual seu desempenho é o melhor;
4. No caso de redes neurais, é possível ter redes com diferentes inicializações. Ao invés de utilizar apenas uma e descartar as outras, a combinação poderia utilizar as vantagens de cada uma delas.

Estes casos sugerem o uso de combinadores devido à disponibilidade de classificadores ou dados. No entanto, para muitas aplicações a escolha de um classificador que possua um bom desempenho é suficiente para resolver o problema. Quando os problemas a serem resolvidos são complexos, envolvendo um grande conjunto de classes, conjuntos de atributos com dimensionalidades e características diferentes ou ainda dados ruidosos, a escolha de um único classificador pode se tornar difícil, pois provavelmente limitaria a capacidade de reconhecimento do sistema. Nestes casos, combinar classificadores pode ser interessante:

"Freqüentemente um classificador combinado dá resultados melhores do que classificadores individuais, por combinar – com o uso de alguma técnica – as decisões independentes de cada classificador e, por conseqüência, as vantagens dos classificadores individuais na solução final, resultando em considerável melhora no acerto geral" [DUIN, 2000].

3.8.1.Métodos de Combinação

Segundo XU et. al., (1992), cada classificador oferece um nível diferente de informação como saída, e que será usada pelo combinador. Os níveis são classificados em:

- *Abstrato*: o classificador *r* apenas indica um rótulo *j*, que é a classe escolhida para o padrão;

- *Ranking*: *r* faz um ranking com todas as classes, definindo uma lista na qual a classe do topo é a primeira escolha;
- *Medida* (confiança): *r* atribui uma medida a cada classe, que é o grau de confiança de x pertencer a cada classe.

O nível 3 (medida) é o que oferece informação mais relevante e o nível 1 (abstrato) menos informação sobre a decisão a ser tomada.

Nas próximas seções são apresentados alguns dos métodos de combinação mais usados na prática. Antes de descrever cada um dos métodos de combinação estudados, apresenta-se uma pequena sistematização, a título de clareza, em termos dos níveis de classificação envolvidos, como é mostrado na Tabela 13 [SANTOS, 2001]:

Tabela 13 – Sistematização dos métodos de combinação estudos.

Nível de Classificação	Métodos Estudados
Nível Abstrato	Combinação por Votações; Combinação por Probabilidades Posteriores; Combinação por Formalismo Dempster-Shafer; Combinação por Integrais Nebulosas.
Nível de Ranking	Combinação por Borda count.
Nível de Medida	Combinações lineares genéricas (incluído Média); Combinação por Integrais Nebulosas.

3.8.1.1.Votação Majoritária

Neste esquema de votação, se para um dado padrão x, mais da metade dos classificadores atribuir a ele um rótulo j, o rótulo E(x) do classificador integrado é escolhido como j também; caso contrário, há rejeição. Simbolicamente [SANTOS, 2001]:

$$E(x) = \begin{cases} j, se \quad \sum_{K} (j_k(x) = j) > \dfrac{K}{2} \quad j = \{1,2,...,M\} \\ M+1, caso \quad contrário \end{cases} \qquad \text{eq. (13)}$$

Notação:

M é o número de classes;

K é o número de classificadores;

$j_K(x)$ é o rótulo fornecido pelo classificador k para o padrão x.

3.8.1.2.Votação por Pluralidade

Na votação por pluralidade [HANSEN, 1990], para um padrão x, se dentre os classificadores envolvidos um dos rótulos é mais escolhido que qualquer outro, este rótulo é selecionado como resposta E(x) do comitê. Caso contrário, o rótulo de rejeição é usado. Simbolicamente:

$$E(x) = \begin{cases} j, se \quad \sum_K (j_K(x) = j) > \sum_K (j_K(x) = i) \quad \forall i \neq j \quad j, i = \{1,2,...,M\} \\ M+1, caso \quad contrário \end{cases} \qquad \text{eq. (14)}$$

Notação:

M é o número de classes;

K é o número de classificadores;

$j_K(x)$ é o rótulo fornecido pelo classificador k para o padrão x.

3.8.1.3.Borda Count

O '*Borda Count*' [CHO, 1995] é um método de combinação no nível de ranking.

Dado um padrão x, definem-se primeiramente medidas $\beta_j(c)$ como sendo número de classes ordenadas abaixo da classe c pelo j-ésimo classificador. Depois, o '*Borda Count*' para a classe c é calculado como:

$$B(c) = \sum_{j=1}^{K} \beta_j(c) \qquad \text{eq. (15)}$$

A decisão final do comitê, E(x), é dada pela seleção da classe com maior valor de medida, dentre os K classificadores. Em caso de empate, o rótulo de rejeição é usado:

$$E(x) = \begin{cases} j, se \quad B(j) > B(i) \quad \forall i \neq j \quad j, i = \{1,2,...,M\} \\ M+1, caso \quad contrário \end{cases} \qquad \text{eq. (16)}$$

Notação:

M é o número de classes.

Apesar da existência de diferentes métodos de formação dos membros do comitê, devido ao grande volume de dados do problema especifico em estudo (identificação do perfil do cliente irregular), os métodos utilizados para a criação dos comitês for simplesmente a divisão da base de dados em diferentes conjuntos, conforme é descrito no próximo capítulo.