

CAPÍTULO 2

2.1 – Introdução

Assim como muitas das tecnologias desenvolvidas, as redes neurais artificiais (RNA's) surgiram durante a Segunda Grande Guerra Mundial na década de 40, porém somente nos últimos 20 anos encontraram uma sólida aplicação, principalmente nas áreas da indústria, educação e pesquisa. Esse fato deve-se em grande parte aos trabalhos publicados por Hopfield em 1982 e Rumelhart e MacLelland em 1986[3], por ocasião da proposição do algoritmo de retropropagação de erro para treinamento de redes *perceptrons* de múltiplas camadas. Doravante esse fato ressurgiu o interesse na área das redes neurais artificiais.

Uma rede neural é composta por um grande número de elementos processadores, também denominados unidades de processamento, amplamente conectados entre si. Cada uma das conexões interliga somente dois elementos processadores, geralmente em um único sentido, e possui um valor que determina o grau de conectividade entre estes, denominado de peso da conexão. Esses elementos possibilitam o aprendizado das informações que são fornecidas por seus canais de entrada.

Desse modo, todo o processamento é realizado de forma distribuída entre os elementos processadores da rede, onde cada qual o realiza isolada e paralelamente,

enviando seu resultado para outras unidades através das conexões entre eles. Por isso, as redes neurais também são conhecidas como sistemas de processamento distribuído e paralelo (PDP).

A forma pela qual os elementos processadores estão interligados é denominada topologia ou padrão de interconexão.

A capacidade das Redes Neurais em resolver um determinado problema encontra-se embutida na topologia (padrão de interconexão) da rede. Ou seja, o modo pelo qual os elementos processadores estão interconectados. Desta forma uma rede neural pode ter várias propriedades, entre elas o mapeamento não-linear entre a entrada e a saída da rede.

As RNA's possuem as seguintes características comuns às redes biológicas:

- Processamento paralelo maciço: armazena conhecimento através do treinamento;
- Interligações entre neurônios feitas por conexões sinápticas: utilizadas para armazenamento de informações;
- Capacidade de generalização: permite que a RNA forneça, em tempo real, saídas razoáveis para entradas que não participaram da fase de treinamento.

As atribuições dadas as RNA's podem ser divididas em arquitetura e propriedades funcionais. A arquitetura de uma RNA define o número de camadas, de neurônios por camada e sua interconectividade. As propriedades funcionais definem o modo de treinamento, a associatividade, a classificação e o processamento de novas informações.

2.2 – Conceitos Básicos

2.2.1 – Neurônios Artificiais

Uma rede neural artificial é formada por um conjunto de processadores artificiais interligados, denominados de neurônios, eles são interligados através de conexões (pesos) de intensidades variáveis as quais são responsáveis pelo armazenamento do conhecimento da RNA, ou seja, armazenam todo o conhecimento experimental adquirido na fase de treinamento ou aprendizado e posteriormente tornam esse conhecimento disponível.

O modelo do elemento processador normalmente possui N entradas e 1 única saída, e seu processamento consiste em transferir para sua saída um valor calculado a partir de outros valores presentes nas camadas anteriores, através de uma função denominada função de transferência ou função de ativação.

Uma definição mais completa do neurônio inclui ainda a especificação de um deslocamento (“bias”) que é aplicado ao resultado da função somatória, antes desse resultado ser submetido à função de transferência. Esse deslocamento permite a obtenção de resultados mais genéricos.

Utilizando uma notação matricial tem-se:

$$P = [P(1) \ P(2) \ P(3) \dots P(R)]^T \quad (2.1)$$

$$W = [W(1,1) \ W(1,2) \ W(1,3) \dots W(1,R)] \quad (2.2)$$

sendo P a matriz que contém os R sinais de entrada, os quais são multiplicados, “ponderados”, pelos pesos das interconexões, contidos na matriz W . A Figura 2.1, mostra a estrutura de um neurônio artificial.

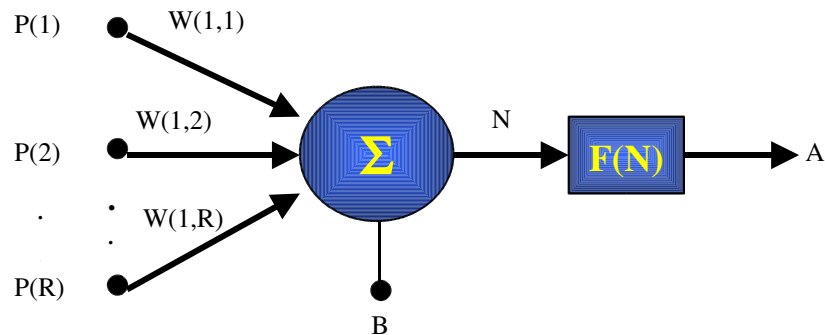


Fig. 2.1 – Neurônio Artificial

Os sinais ponderados ($W \cdot P$) são somados à polarização (B) (“bias”) para produzir um estado de ativação N que é processado pela função F , chamada de função de ativação, essa função converte a soma de ativação (N) das entradas em ativação de saída (A) e é um fator básico de diferenciação entre os neurônios. Essa função decide, baseada no potencial interno do neurônio, qual o valor a ser enviado aos demais neurônios da camada seguinte.

Tem-se vários tipos de funções de ativação e sua escolha depende do tipo de problema a ser solucionado, do tipo de rede, regra de treinamento, etc.

As funções de ativação mais usadas são:

- Função Degrau: é uma função não linear, assumindo valores 0 ou 1. A Equação 2.3 expressa seu comportamento e seu gráfico é mostrado na Figura 2.2.

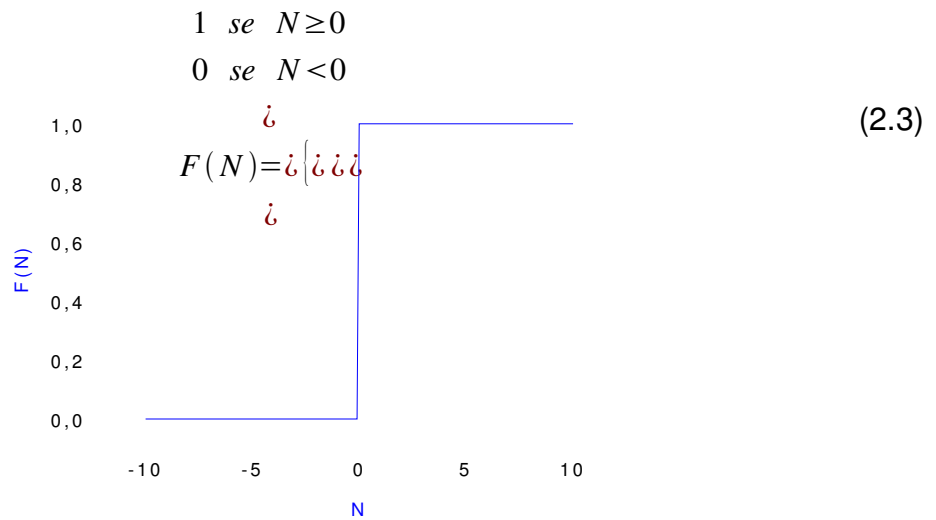


Figura 2.2 – Função Degrau

- Função Linear: essa função simplesmente retorna o valor que lhe foi passado como entrada. A Equação 2.4 expressa seu comportamento e seu gráfico é mostrado na Figura 2.3.

$$F(N) = N$$

(2.4)

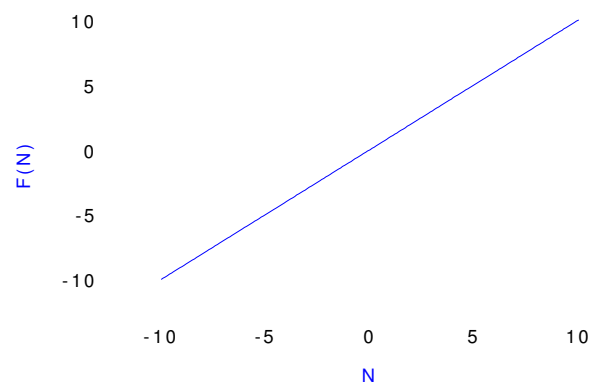


Figura 2.3 – Função Linear

- Função Bipolar: é uma função não linear, assumindo valores -1 ou 1 . A Equação 2.5 expressa seu comportamento e seu gráfico é mostrado na Figura 2.4.

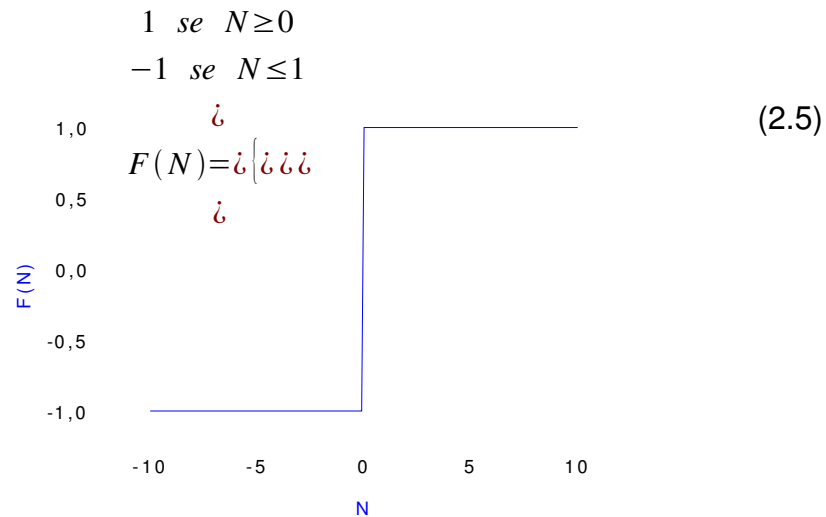


Figura. 2.4 – Função Bipolar

- Função Linear Saturada: essa função pode ser vista como um amplificador não linear. A Equação 2.6 expressa seu comportamento e seu gráfico é mostrado na Figura 2.5.

$$\begin{aligned}
 &1 \text{ se } N \geq 1 \\
 &N \text{ se } -1 < N < 1 \\
 &-1 \text{ se } N \leq -1
 \end{aligned}$$

(2.6)

$$F(N) = \begin{cases} 1 & N \geq 1 \\ N & -1 < N < 1 \\ -1 & N \leq -1 \end{cases}$$

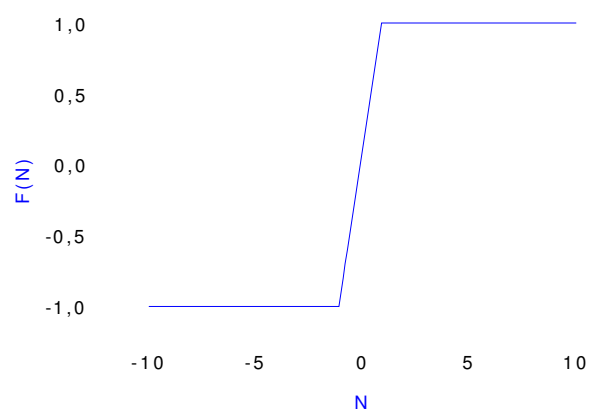


Figura 2.5 – Função Linear Saturada

- Função Sigmóide: é uma função definida não linear, variando de 0 a 1. Diferenciável em todo o seu domínio, a Equação 2.7 expressa seu comportamento, e seu gráfico é mostrado na Figura 2.6.

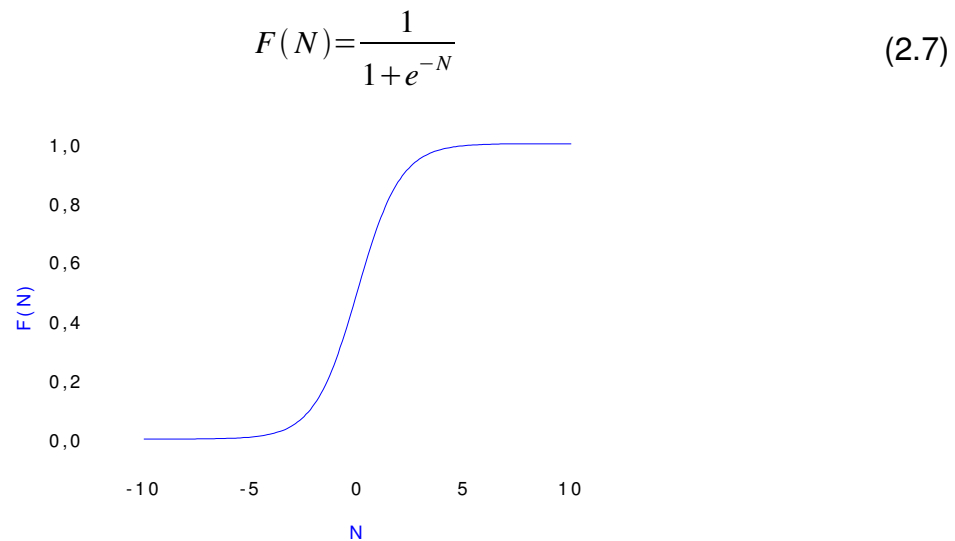


Figura 2.6 – Função Sigmóide

- Função Tangente Hiperbólica: é uma função não linear, como a função sigmoide é diferenciável em todo o seu domínio, mas varia de -1 a 1 . A Equação 2.8 expressa seu comportamento, e seu gráfico é mostrado na Figura 2.7.

$$F(N) = \frac{e^N - e^{-N}}{e^N + e^{-N}} \quad (2.8)$$

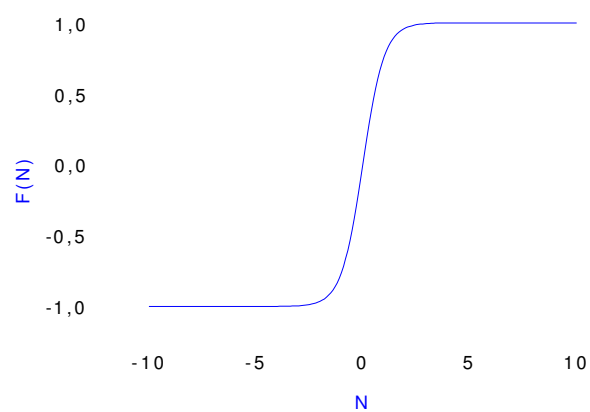


Figura 2.7 – Função Tangente Hiperbólica

Atualmente existem vários estudos voltados para a utilização de funções de ativação não convencionais [4].

2.2.2 - Uma Camada de Neurônios Artificiais

Dois ou mais neurônios podem ser combinados em uma camada, e uma rede neural artificial pode-se ter uma ou mais camadas. Primeiramente considera-se uma rede neural artificial com apenas uma camada, a fim de obter-se um melhor entendimento de seu processamento. Posteriormente, estende-se esse conceito para uma rede com mais de uma camada de neurônios.

Uma camada de neurônios de uma RNA, com R entradas e S neurônios artificiais é mostrada na Figura 2.8, utilizando a notação matricial em 2.1 e 2.2.

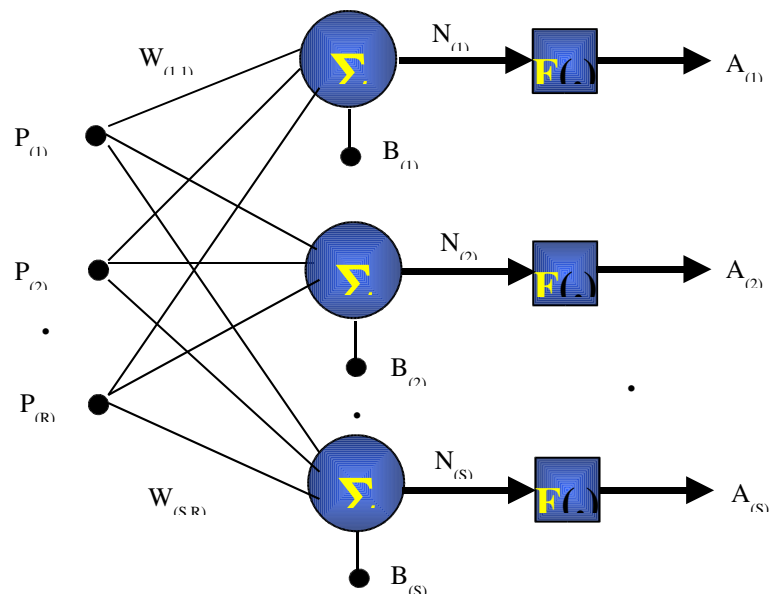


Fig. 2.8 – Uma camada de neurônios artificiais

Na Figura 2.8 considera-se que a camada de neurônios possui a mesma função de ativação (F) para todos os neurônios da camada.

Nessa rede cada um dos R elementos do vetor de entrada P são conectados à entrada de cada um neurônio através da matriz de pesos W. Cada um dos S neurônios efetua um somatório de suas entradas, a qual é somada a sua polarização (B), formando uma matriz N com o mesmo número de neurônios da camada, nesse caso igual a S. As S saídas da camada são formadas pelas saídas das funções de ativação dos neurônios da camada, formando uma matriz saída A, que para esse caso é a saída da rede.

A matriz P pode conter várias amostras de entradas para a rede neural. Nesse caso a matriz de entrada pode conter Q elementos, onde cada um desses elementos representa uma informação de entrada para a rede neural.

$$P = \begin{pmatrix} P(1,1) & P(1,2) & \cdots & P(1,Q) \\ P(2,1) & P(2,2) & \cdots & P(2,Q) \\ \vdots & \vdots & & \vdots \\ P(R,1) & P(R,2) & \cdots & P(R,Q) \end{pmatrix} \quad (2.9)$$

$$A = \begin{pmatrix} A(1,1) & A(1,2) & \cdots & A(1,Q) \\ A(2,1) & A(2,2) & \cdots & A(2,Q) \\ & \vdots & & \vdots \\ A(R,1) & A(R,2) & \cdots & A(R,Q) \end{pmatrix} \quad (2.10)$$

Cada coluna da matriz P (Eq. 2.9) representa uma informação de entrada para a rede neural. De acordo com essa matriz a rede neural teria R entradas e Q informações de entrada, e para cada vetor de entrada $P(R,Q)$ resultará em um correspondente vetor de saída $A(R,Q)$.

Na Figura 2.9 tem-se uma rede neural artificial com 3 camadas. Cada camada possui um determinado número de neurônios e sua respectiva função de ativação.

A primeira camada é denominada de camada de entrada e é responsável pelo processamento e ponderação da informação de entrada. A segunda camada é denominada de camada intermediária, também conhecida como camada escondida. Essa camada processa apenas sinais internos da rede, ficando isolada do meio externo. É responsável pela característica não-linear da rede e é grande responsável pela capacidade da rede em realizar o mapeamento não-linear entre a entrada e a saída de rede. A última camada é denominada de camada de saída. Essa camada recebe e pondera os sinais de saída da camada intermediária e fornece os sinais de saída da rede.

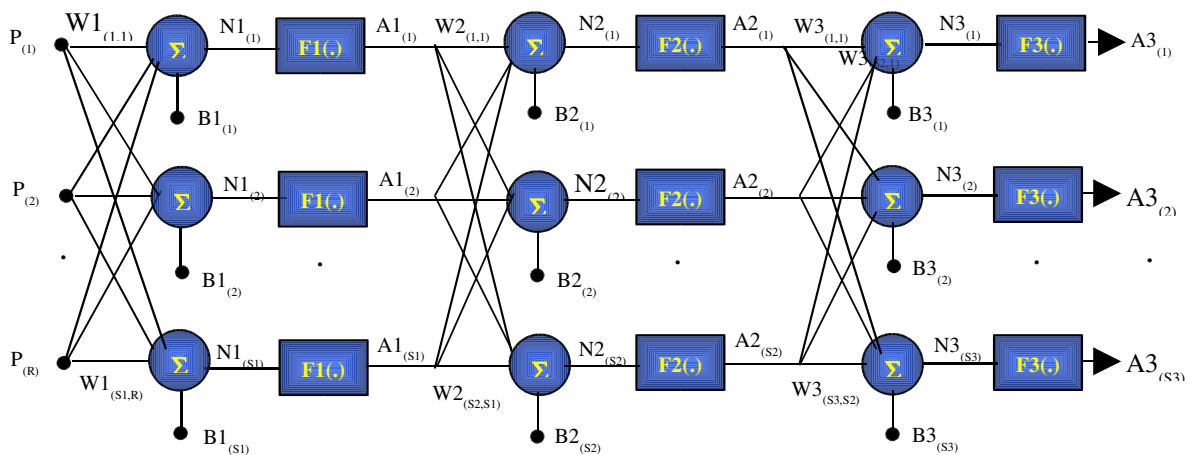


Fig. 2.9 – Três camadas de neurônios artificiais

O número de neurônios de entrada e de saída deve ser estimado de acordo com as características do problema. Já quanto ao número de neurônios da camada intermediária sabe-se que: se esse número for muito grande, a rede neural pode se especializar nos dados de treinamento e perder a capacidade de generalização [5], que é uma de suas principais características; se for pequeno a rede neural pode não ter memória suficiente para armazenar conhecimento.

As principais características das redes neurais artificiais são:

- mapeamento não-linear: uma rede neural tem a capacidade de realizar o mapeamento não-linear da relação entre a sua entrada e saída.
- processamento paralelo e distribuído: a estrutura paralelamente distribuída dos elementos processadores de uma rede neural permite que o processamento e o armazenamento da informação se distribua por toda a rede.

- Generalização de conhecimentos: a rede neural, usando como base os conhecimentos adquiridos dos exemplos apresentados na fase de treinamento, pode presumir a resposta correta para as novas entradas.

2.3 - Classificação

As redes neurais podem ser classificadas:

- Quanto ao fluxo de informações

1. *Feedforward*: um neurônio de uma determinada camada envia estímulos somente para os neurônios da camada posterior e/ou da mesma camada. Um exemplo deste tipo de rede é mostrado na Figura 2.9.
2. *Feedback*: um neurônio de uma camada envia estímulos tanto para os neurônios da camada posterior como para os neurônios da camada anterior, bem como para os neurônios da mesma camada. A Figura 2.10 mostra um exemplo deste tipo de RNA.

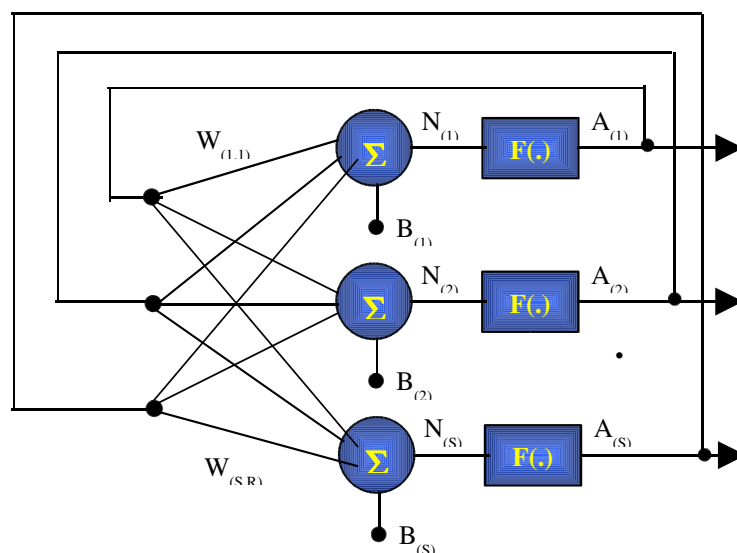


Figura 2.10 – RNA com fluxo feedback.

- Quanto ao tipo de treinamento

1. Treinamento supervisionado: consiste em apresentar à rede um padrão a ser reconhecido, juntamente com a resposta que a rede deve fornecer ao reconhecer novamente esse mesmo padrão.
2. Treinamento não supervisionado: consiste em apresentar apenas os padrões para a rede e essa se encarrega de agrupar aqueles que possuem propriedades similares.

Denomina-se ciclo ou fase de treinamento uma apresentação de todos os N pares (entrada e saída) do conjunto de treinamento no processo de aprendizado. A correção dos pesos num ciclo pode ser executada de dois modos:

1) Modo on-line (padrão): A correção dos pesos acontece a cada apresentação à rede de um exemplo do conjunto de treinamento. Cada correção de peso baseia-se somente no erro do exemplo apresentado naquela iteração. Assim, em cada ciclo ocorrem N correções.

2) Modo Batch: Apenas uma correção é feita por ciclo. Todos os exemplos do conjunto de treinamento são apresentados à rede. Seu erro médio é calculado e a partir desse erro fazem-se as correções dos pesos.

2.4 - Processo de Aprendizado

Uma rede neural aprende sobre seu ambiente através de processos iterativos de ajuste de seus pesos sinápticos (interconexões), adaptando-se aos estímulos da

entrada, para produzir a saída desejada, o treinamento. Nessa fase o conjunto de treinamento é utilizado para realizar o processo de treinamento da rede neural.

O tipo de treinamento é caracterizado pela forma como ocorrem as mudanças nos parâmetros da rede.

Depois de cada processo iterativo a rede torna-se mais instruída sobre seu ambiente. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

O algoritmo de aprendizado define as regras para a solução de um determinado problema de aprendizado. Existem vários tipos de algoritmos de aprendizagem, e diferem principalmente pelo modo como suas interconexões são modificadas. Existem algoritmos de aprendizagem específicos para determinados modelos de redes neurais.

Na fase de aprendizagem, também conhecida como treinamento, seguindo o algoritmo de treinamento escolhido, serão ajustados os pesos das conexões. É importante considerar alguns aspectos nessa fase, tais como: inicialização dos pesos da rede, modo de treinamento e o tempo de treinamento.

Uma boa escolha dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Normalmente, os valores iniciais dos pesos da rede são números aleatórios uniformemente distribuídos, em um intervalo definido.

Para treinar uma Rede Neural podem ser utilizados três mecanismos distintos de aprendizado: o aprendizado supervisionado, quando são fornecidos integralmente os resultados desejados; o aprendizado por reforço, quando apenas um parâmetro externo de comparação (ou medida relativa da adEquação) é utilizado para definir se estão agindo corretamente ou erroneamente; e o aprendizado não-supervisionado, quando a própria rede é capaz de ajustar o seu funcionamento.

Quanto ao tempo de treinamento, vários fatores podem influenciar a sua duração, porém sempre será necessário utilizar algum critério de parada. O critério de parada do algoritmo de treinamento *backpropagation* por exemplo não é bem definido, e geralmente é utilizado um número máximo de ciclos. Mas, devem ser considerados a taxa de erro médio por ciclo e a capacidade de generalização da rede. Pode ocorrer que em um determinado instante do treinamento a generalização comece a degenerar, causando o problema de "over-training", ou seja, a rede se especializa no conjunto de dados do treinamento e perde a capacidade de generalização.

O treinamento deve ser interrompido quando a rede apresentar uma boa capacidade de generalização e quando a taxa de erro for suficientemente pequena, ou seja, menor que um erro admissível. Assim, deve-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

2.5 - Coleta de Dados e Separação em Conjuntos

Os dois primeiros passos do processo de desenvolvimento de uma rede neural artificial é a coleta de dados relativos ao problema e a sua separação em um conjunto de treinamento e um conjunto de testes. Essa tarefa requer uma análise cuidadosa sobre o problema a fim de minimizar ambigüidades nos dados de treinamento. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema.

Normalmente os dados coletados são separados em duas categorias: dados de treinamento, que serão utilizados para o treinamento da rede; e dados de teste, que serão utilizados para verificar seu desempenho sob condições reais de utilização.

Além dessa divisão, pode-se usar também uma subdivisão do conjunto de treinamento, criando um conjunto de validação, que é utilizado para verificar a eficiência da rede quanto à sua capacidade de generalização durante o treinamento, e podendo ser empregado como critério de parada do treinamento.

Depois de determinados esses conjuntos eles são, geralmente, colocados em ordem aleatória para prevenção de tendências associadas à ordem de apresentação dos dados. Além disso, pode ser necessário pré-processar esses dados através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

2.6 – Fase de Teste

Durante essa fase o conjunto de teste é utilizado para avaliar o desempenho da rede com dados que não foram previamente utilizados. O desempenho da rede medido nessa fase é uma boa indicação de seu desempenho real.

Devem ser considerados ainda outros testes, como o da análise do comportamento da rede utilizando entradas especiais e o da análise dos pesos atuais da rede, pois se existirem valores muito pequenos, as conexões associadas podem ser consideradas insignificantes e assim serem eliminadas (“prunning”). De modo inverso, valores substantivamente maiores que os outros poderiam indicar que houve especialização da rede nos dados de treinamento.

2.7 - Arquiteturas

2.7.1 - Perceptron

A *Perceptron*, desenvolvida por Frank Rosenblatt em 1957, foi a primeira arquitetura de rede neural artificial. Consiste de uma rede linear de neurônios organizados em uma única camada, onde o vetor de entrada alimenta simultaneamente todos os neurônios.

Nesse modelo cada neurônio é treinado para identificar um padrão específico, como é o caso da função degrau, onde essa função de transferência pode fazer com que apenas um neurônio seja ativado, aceitando um baixo grau de distorção no padrão de teste. Essa rede simples gerou muito interesse quando foi desenvolvida devido a sua habilidade de classificar padrões linearmente separáveis [6]. Nessa implementação os pesos das conexões de entrada são multiplicados pelos valores das entradas e então somados e submetidos a uma função de transferência do tipo degrau para gerar a saída do neurônio.

Uma camada com S neurônios do tipo *Perceptron* pode ser utilizada para realizar a classificação de S padrões linearmente separáveis.

O problema das redes de um nível baseadas em *Perceptron* é que (como já foi dito acima), elas dividem o espaço de soluções em duas regiões distintas, mas em uma função lógica XOR não se consegue abranger as duas respostas em uma só região.

2.7.2 - Adaline

Esse modelo, criado por Widrow e Hoff corresponde a uma modificação do *Perceptron* onde se utiliza o algoritmo de aprendizado LMS (Least Mean Square), também conhecido como Regra Delta.

O algoritmo LMS procura minimizar o erro médio quadrático entre a saída desejada e a saída apresentada por uma rede semelhante a rede *Perceptron*. Os elementos de processamento, chamados ADaptive LINear Element, dão o nome ao modelo.

As características do *Adaline* são muito semelhantes às características do *Perceptron*. O *Adaline*, assim como o *Perceptron*, cria uma reta limite entre duas regiões de decisão. A diferença está no algoritmo de aprendizado e nas estruturas que devem ser utilizadas para implementá-lo. A arquitetura é de uma rede de uma única camada disposta linearmente, com cada neurônio possuindo uma retroalimentação do erro na sua própria saída.

Outra diferença entre os dois é que o *ADALINE* não usa uma função de transferência em seus neurônios quando o erro é calculado, fazendo apenas a soma ponderada. A função de transferência só é empregada para obter a saída final do neurônio.

2.7.3 - Kohonen

Esse tipo de arquitetura é muito utilizada em aplicações de reconhecimento de fonemas e robótica. Utiliza a regra de aprendizado do tipo competitivo, que é caracterizada pelas conexões laterais dos neurônios com seus vizinhos, estabelecendo assim uma “competição entre os neurônios” que levará a rede a um estado estável.

Esse tipo de rede possui aprendizado não supervisionado, portanto só apresenta padrões de entrada.

2.7.4 - Hopfield

Este modelo de rede neural foi introduzido por Hopfield em 1982. É uma arquitetura de rede feedback (Figura 2.10). Esse tipo de rede caracteriza-se por não possuir entradas e sim estados iniciais de seus pesos, de tal forma que a rede busque através de seu treinamento um estado de equilíbrio, o qual depende dos estados iniciais.

2.7.5 – Redes de Funções Base Radiais (RBF)

Este é um tipo de rede neural artificial *feedforward* (Figura 2.9). É caracterizada por sua simplicidade de treinamento, eficiência computacional e uso de funções gaussianas nas unidades processadoras da camada intermediária.

Nessa arquitetura a camada tem como objetivo transmitir os sinais para a segunda camada. Essa por sua vez, realiza uma transformação não linear do espaço de entrada para um novo espaço. A última camada realiza uma combinação linear das saídas das unidades intermediárias através de seus neurônios lineares.

As funções gaussianas geram uma resposta diferente de zero quando o padrão de entrada está dentro de uma região localizada dentro do espaço de entrada. A função que é mais utilizada com a rede RBF é a função de Gauss, dada por

$$F(N) = \exp \left[-\frac{1}{2} (N - \mu)^2 \right] \quad (2.11)$$

sendo μ o desvio padrão da função gaussiana e σ caracteriza o espalhamento da função em torno de sua média

de uma função gaussiana $\sigma = 0,5$ e

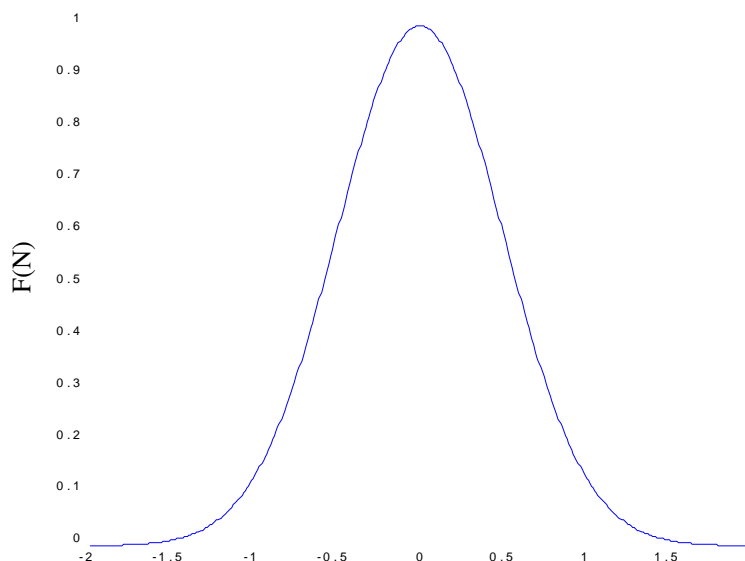


Figura 2.11 – Função gaussiana

Esse tipo de rede combina o aprendizado supervisionado com o aprendizado não-supervisionado. O procedimento de treinamento é baseado na minimização do erro quadrático da saída da rede.

As redes RBF's podem ser usadas em problemas de aproximações de funções, predição e classificação.

2.7.6 – *Adaptive Resonance Theory*

Esse tipo de rede utiliza o aprendizado competitivo; foi proposto por Gail Carpenter em 1978. É constituído de uma família de RNA's, dentre as quais destacam-se: ART1, ART2 e ART3. Uma importante característica dessas redes é a capacidade de aprender novas informações sem que as informações aprendidas no passado sejam prejudicadas. Possuem grande aplicabilidade em reconhecimento de padrões.

As três variâncias das redes ART são similares diferenciando em alguns pontos, ou seja, a rede ART1 é caracterizada por usar entradas binárias, a rede ART2 admite entradas analógicas e a rede ART3 possui um mecanismo transmissor químico nas sinapses, portanto emula a complexidade do cérebro.

2.8 - *Backpropagation*

A técnica *Backpropagation* conseguiu dar uma nova vida às redes neurais artificiais, pois permite o aprendizado das redes neurais baseadas no *Perceptron* com qualquer número de camadas, servindo inclusive para o caso do XOR. É difícil dizer quem é o real criador da solução para redes multi-nível (o *Backpropagation*), pois vários trabalhos foram feitos em lugares diferentes na mesma época. No entanto foi Rumelhart (1986) quem mais difundiu este método, cuja grande importância é o fato de ser uma regra delta generalizada para múltiplos níveis.

Uma rede que utiliza o modelo *Back-Propagation* é uma rede com neurônios em três ou mais níveis (camadas): um nível de entrada, um ou mais níveis intermediários ou ocultos (camadas escondidas) e um nível de saída (camada de saída). A função transferência é uma função do tipo função sigmóide (Eq. 2.7).

2.8.1 - Algoritmo de Aprendizado *Backpropagation*

Este aprendizado baseia-se na propagação retrógrada do erro para os níveis superiores da rede, de acordo com o grau de participação que cada neurônio teve no erro do nível anterior.

Primeiramente, a rede usa apenas o vetor de entrada para produzir sua própria saída (ativação - *forward*), depois então é possível comparar a saída atual com a saída desejada e assim determinar o erro. Se não há diferença entre a saída desejada e a saída obtida, nenhuma alteração precisa ser feita nos pesos ou nos limitantes dos neurônios. Caso contrário, os pesos têm de ser modificados para reduzir o erro (retro-propagação do erro - *backward*), gerando uma função erro.

A função erro a ser minimizada é

$$E = \frac{1}{2} \sum_{i=1}^S (T_i - A_i)^2 \quad (2.12)$$

sendo T_i a saída desejada e A_i o valor estimado na saída da rede neural para o i -ésimo neurônio da camada de saída e S o número de neurônios da camada de saída.

O cálculo da variação que o peso do neurônio deve sofrer é dado pela seguinte expressão:

$$W_{ij}(k) = W_{ij}(k+1) - W_{ij}(k) = - \left(\frac{\partial E}{\partial W_{ij}} \right) + \quad (2.13)$$

sendo η a taxa de aprendizagem e α o fator de momento, os quais serão abordados mais adiante, k indica a iteração no processo de treinamento; W_{ij} é o peso entre o j -ésimo neurônio de uma camada e o i -ésimo neurônio da camada seguinte, e $\left(\frac{\partial E}{\partial W_{ij}} \right)$ é a variação do erro em relação à interconexão W_{ij} .

O novo peso da interconexão W_{ij} é dado pela seguinte expressão:

$$W_{ij}(k) = W_{ij}(K-1) + \eta \left(\frac{\partial E}{\partial W_{ij}} \right) + \alpha (W_{ij}(k) - W_{ij}(K-1)) \quad (2.14)$$

sendo $W_{ij}(k)$ o novo valor do peso; $W_{ij}(k-1)$ é o valor do peso atual e $\left(\frac{\partial E}{\partial W_{ij}} \right)$ é o valor da variação que o peso atual deve sofrer.

2.8.2 – Deficiências do *Backpropagation*

Mesmo com o aparente sucesso do algoritmo, existem alguns aspectos que fazem com que esse não possa ser utilizado em qualquer tipo de aplicação. O maior problema é o longo processo de treinamento. O treinamento pode ter sido realizado com sucesso pode resultar em uma baixa taxa de generalização, pois não há garantia de que o mínimo global de erro tenha sido encontrado. Em outras palavras, algumas vezes, dependendo dos pesos iniciais, da curva de aprendizado e dos dados de entrada, a rede neural artificial pode até não aprender.

- Paralisação da Rede

Durante o treinamento da rede os pesos podem ser ajustados para valores muito altos. As entradas ou as saídas da camada intermediária podem alcançar valores muito altos (positivos ou negativos), e em decorrência da função sigmóide a rede vai ter um valor de ativação muito próximo de zero ou de um, conseqüentemente, o processo de treinamento pode virtualmente ficar paralisado. Isto ocorre porque um neurônio que aprendeu alguma coisa (saída com valor elevado) tende a se acomodar e não mudar seus pesos.

Essa “acomodação” da rede é proposital. Sua função é evitar que os neurônios fiquem mudando os seus pesos a todo momento. A rede nunca se estabiliza, não haveria convergência, pois ao apresentar um novo exemplo a rede sobreporia a aprendizagem do exemplo anterior.

- Mínimos Locais

A superfície de erro gerada por uma rede neural é complexa e cheia de picos e depressões. A rede pode se concentrar em uma depressão mínima local (erro mínimo local), deixando de se concentrar em uma depressão próxima muito mais profunda (erro mínimo global), onde o erro final seria mais reduzido. Existem alguns métodos para tentar fugir desse problema, como por exemplo a técnica do *momentum* que será comentada a seguir.

O *momentum* (β) é uma técnica para que a rede escape dos mínimos locais pela inércia da alteração dos pesos na direção do mínimo global, baseada na etapa anterior de correção dos pesos, permitindo usar velocidades de aprendizado maiores, sem no entanto correr o risco do algoritmo ignorar o mínimo global.

Apesar do uso dessa técnica, esse problema ainda é uma das principais deficiências encontradas em treinamento com o uso do *backpropagation*. No entanto, estudos vem sendo realizados nessa área com o intuito de amenizar esse problema, como por exemplo o uso de técnicas numéricas para otimizar o treinamento[7].

2.9 - Taxa de aprendizagem (η)

Na fase de treinamento, primeiramente, a saída inicial da rede é usada para calcular o erro inicial entre essa e a saída desejada. Então são calculados novos pesos e polarizações usando a taxa de aprendizado existente, depois uma nova saída e erro são produzidos. Se o novo erro excede o erro anterior por mais do que uma taxa pré-definida, os novos pesos, polarizações, saída e erro são descartados e a taxa de

aprendizado é diminuída. Do contrário, os novos pesos são mantidos e se o novo erro é menor do que o erro anterior, a taxa de aprendizado é aumentada.

Esse procedimento aumenta a taxa de aprendizado, mas somente na medida que a rede possa aprender sem aumentar grandemente o erro, de forma que, uma taxa de aprendizado otimizada possa ser obtida para a formação local.

O aumento da taxa de aprendizado faz com que aumente-se o passo dos neurônios (alteração dos pesos mais bruscamente) em direção ao mínimo da curva de erro, no entanto, se esse passo for muito grande, pode-se pular sobre esse valor mínimo e a rede cair em um mínimo que não é o ideal (erro mínimo local).