Modelos Conexionistas - Redes Neurais

1 INTRODUÇÃO

Redes Neurais Artificiais ou simplesmente Redes Neurais (também conhecidas como modelos conexionistas) têm sido, ao longo dos últimos anos, uma área de pesquisa de grande interesse. Redes Neurais podem ser caracterizadas como modelos computationais com habilidades de adaptar, aprender, generalizar, agrupar ou organizar dados, nos quais a operação é baseada em processamento paralelo. A origem das pesquisas nesta área, data da época em que foi apresentado o modelo computacional de um neurônio biológico por McCulloch e Pitts, em 1943. Quando Minsky e Papert publicaram o livro Perceptrons em 1969, no qual eles mostraram as deficiências do modelo Perceptron, até então o único que possuía um algoritmo de treinamento conhecido, a maioria dos estudos em redes neurais foram redirecionados e muitos pesquisadores abandonaram a área. Uns poucos continuaram o trabalho, notadamente Teuvo Kohonen, Stephen Grossberg, James Anderson, e Kunihiko Fukushima. O interesse pela área só re-emergiu depois que alguns resultados teóricos importantes foram atingidos, dentre os quais destaca-se o algoritmo de retro propagação (Back propagation) dos erros em redes Perceptrons de múltipla camadas. Hoje em dia, a maioria das universidades tem um grupo de pesquisas em redes neurais dentro dos departamentos de Psicologia, Física, Informática, Biologia ou Engenharia.

Neste tutorial é feita uma introdução a Redes Neurais Artificiais (RNA). O ponto de vista desta abordagem é computacional. Não há preocupação com a implicação psicológica das redes e só inicialmente são citados modelos biológicos.

Neste tutorial, o capítulo 2 discute as propriedades fundamentais. O capítulo 3 descreve algumas redes clássicas e suas limitações. O Capítulo 4 continua com a descrição das tentativas para superar estas limitações e introduz a

aprendizagem do algoritmo de retro-propagação. Redes auto-organizadas são discutidas no capítulo 5. O capítulo 6 discute as redes recorrentes.

2 CONCEITOS BÁSICOS

As redes neurais possuem arquiteturas baseadas em blocos construtivos semelhantes entre si e que realizam o processamento de forma paralela. Neste capítulo nós primeiramente discutiremos estas unidades de processamento e a seguir as redes e suas diferentes topologias. Na última seção serão apresentadas algumas estratégias de aprendizado.

2.1- Os fundamentos do modelo conexionista

Uma rede neural é formada por um conjunto de unidades de processamento simples que se comunicam enviando sinais uma para a outra através de conexões ponderadas.
O componente elementar desse modelo são as unidades de processamento, também chamadas nós, neurônios ou células. Essa unidade de processamento é um modelo matemático, que possui uma inspiração no modelo biológico de um neurônio.

2.1.1 O modelo do neurônio biológico

O cérebro humano possui milhões de células nervosas (neurônios), que se interligam entre si, formando uma rede disposta em camadas. Nesta redes, são processadas todas as informações referentes aos seres humanos. A capacidade de pensar, memorizar e resolver problemas tem levado muitos cientistas a tentar modelar a sua operação. Diversos pesquisadores têm buscado criar um modelo computacional que represente a funcionalidade do cérebro, de uma maneira simples.

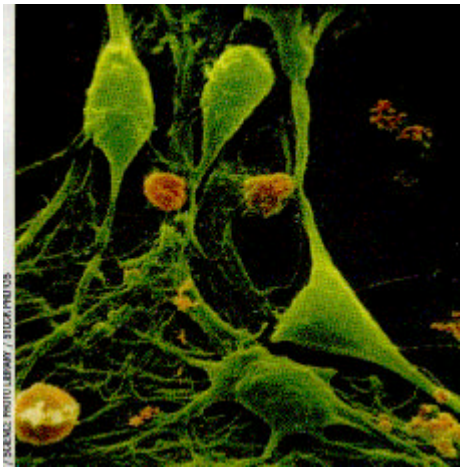


FIG. 1- MICROFOTOGRAFIA DE NEURÔNIOS HUMANOS

Um destes modelos resultou na Neurocomputação, que tem como objeto de estudo as redes neurais.

O neurônio é a célula nervosa dos organismos, encontrada no cérebro. Seu funcionamento é descrito a seguir.

Os estímulos entram nos neurônios através das sinapses, que conectam os dendritos (ramificações de entrada) de um neurônio com axônios de outros neurônios ou com o sistema nervoso.

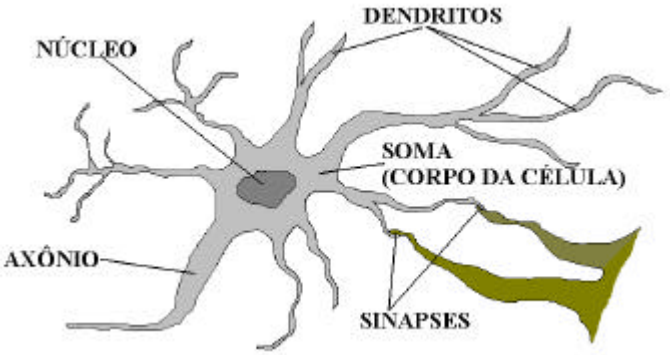


FIG. 2- NEURÔNIO BIOLÓGICO

As sinapses regulam as quantidades de informações que passam dos dendritos para o neurônio. Os sinais são então passados para o somador (corpo celular), que os adiciona e os aplica a um sensor de limiar, que determina o nível de energia mínima de entrada acima do qual o neurônio dispara uma carga elétrica.

Se a soma dos sinais é maior que o nível de limiar, o neurônio envia energia através do axônio, de onde a energia é transmitida para outras sinapses (se bem que, muitas vezes, os axônios conectam-se diretamente com outros axônios ou corpos celulares); ou também pode realimentar a sinapse original. Entretanto, se a soma for menor do que o limiar, nada acontece.

As sinapses têm um papel fundamental na memorização, pois é nelas que são armazenadas as informações. Pode-se imaginar que em cada sinapse a quantidade de neurotransmissores que podem ser liberados para uma mesma freqüência de pulsos do axônio representa a informação armazenada pela sinapse.

Em média, cada neurônio forma entre mil e dez mil sinapses. O cérebro humano possui cerca de 16 bilhões de neurônios, e o número de sinapses é de mais de 100 trilhões, possibilitando a formação de uma rede muito complexa.

2.1.2 O Modelo De Neurônio Artificial

Objetivando simular o comportamento deste neurônio biológico, McCullock e Pitts (1943), propuseram o modelo da figura 3. Neste modelo, cada sinal positivo ou negativo que entra pelo sistema, é multiplicado por um número, ou peso, que indica a sua influência na saída. Caso a soma ponderada dos sinais exceda um certo limite, é gerada uma resposta na saída.

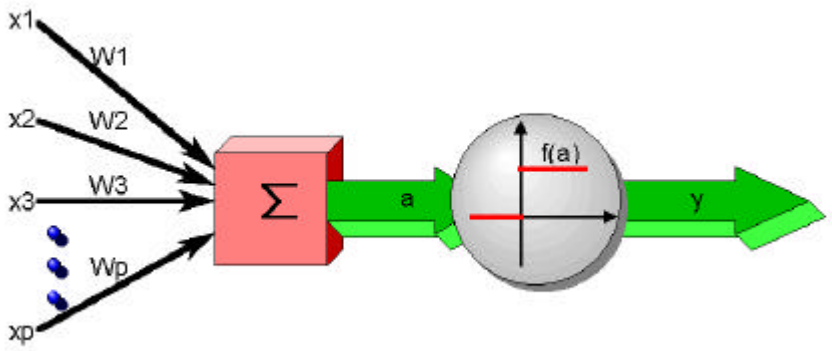


FIG. 3- NEURÔNIO ARTIFICIAL

onde:

- X_1, X_2, ..., X_p são os sinais de entrada;
- W_1, W_2, ..., W_p são os pesos aplicados aos sinais de entrada;
- Σ é a função somadora, tendo como saída "**a**" (função líquida);
- f(a) é a função de transferência (função de Ativação), que no caso do neurônio de McCullock-Pitts é uma função de limiar;
- y é a saída.

2.1.3 Funções De Ativação

Outros modelos de neurônio, implementam funções de ativação distintas. O papel da função de ativação é determinar a forma e a intensidade de alteração dos valores transmitidos de um neurônio a outro. É preciso quantificar a influência de cada entrada na ativação da unidade. Para isso definimos a função de ativação f_i que, dada uma entrada total $a_i(t)$ e a ativação atual, produz um novo valor para a ativação da unidade *i*.

Diferentes tipos de funções de ativação têm sido usados por diferentes autores. A seguir descrevemos algumas.

2.1.4 - Função de Ativação Limiar

A primeira rede com neurônios artificiais utilizava uma função de Ativação por Limiar (Rosenblatt, 1958). O modelo de neurônio usado por Rosenblatt foi o mesmo proposto por McCulloch e Pitts (1943).

Neste modelo, as unidades são ditas binárias, isto é, suas saídas assumem normalmente o valor 0 ou 1. Esta unidade responde com uma saída igual a 1 se o valor da entrada líquida for superior a um determinado valor chamado *Threshold* (que na maioria das vezes é zero), caso contrário, a saída da unidade assume o valor zero (figura 7).

$$y = f_i(a_i(t)) = \begin{cases} 1 \text{ se } a_i(t) \geq 0 \\ 0 \text{ se } a_i(t) < 0 \end{cases}$$

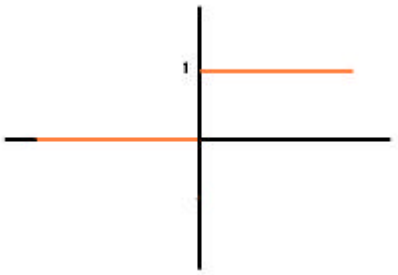


FIG. 4- FUNÇÃO DE ATIVAÇÃO LIMIAR

Algumas vezes os valores 0 e 1 são substituídos por -1 e 1. Uma rede de camada simples utilizando este tipo de unidade é denominada *Perceptron Simples* .

2.1.5 - Função de Ativação Linear

Em 1972, Kohonen definiu um novo modelo de redes neurais conhecido como Mapa Auto-Organizável de Características que, ao contrário do Perceptron, não era limitado a valores binários. Os valores das entradas, dos pesos e das saídas poderiam ser contínuos.

A saída do neurônio neste caso é representada (figura 5) por uma função linear da forma descrita abaixo:

$$y = f_i(a_i(t)) = \begin{cases} 1 \text{ se } a_i(t) \geq 1/2 \\ a_i(t) \text{ se } -1/2 < a_i(t) < 1/2 \\ -1 \text{ se } a_i(t) \leq -1/2 \end{cases}$$

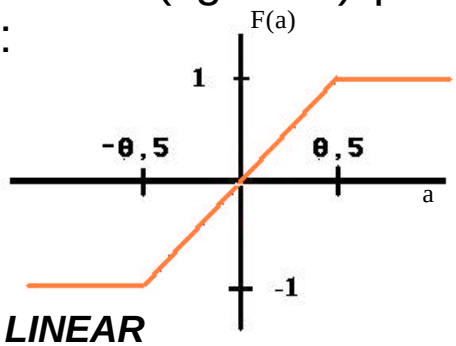


FIG. 5- FUNÇÃO DE ATIVAÇÃO LINEAR

As redes que usam este tipo de função de ativação apresentam apenas uma camada de entrada e uma de saída. Tais redes possuem uma série de limitações quanto ao que são capazes de representar (Hertz et al., 1991).

Não existe nenhuma vantagem de uma rede com mais de duas camadas ou com conexões do tipo "feedback". Em outras palavras, tudo que pode ser aprendido por uma rede de multi-camadas (mais de duas) com neurônios usando este tipo de função de ativação, pode também ser aprendido por uma de camadas simples equivalente (Hertz et al., 1991).

Na verdade, uma rede multi-camadas que utilize neurônios artificiais com funções de ativação com algum tipo de não linearidade, tem um potencial de aprendizado muito maior que uma rede com unidades lineares (Minsky e Paper, 1969). Com unidades não lineares em uma rede com pelo menos uma camada escondida é possível representar algumas associações que não são possíveis de representar com uma rede linear ou uma camada simples que utilize a mesma função de ativação. Entretanto, Minsky e Paper (1969) mostraram que não existe um algoritmo que assegure o treinamento desse tipo de rede multi-camadas.

2.1.6 - Funções de Ativação Semi-Lineares

Uma das respostas à falta de um algoritmo de treinamento, para o tipo de rede discutido anteriormente, é a utilização de redes com unidades semi-lineares (Hertz et al., 1991). As funções de ativação mais usadas por este tipo de unidades são a função logística (Figura 6) e a tangente hiperbólica.

A popularidade destas funções está ligada ao fato de suas derivadas poderem ser expressas em função das próprias funções.

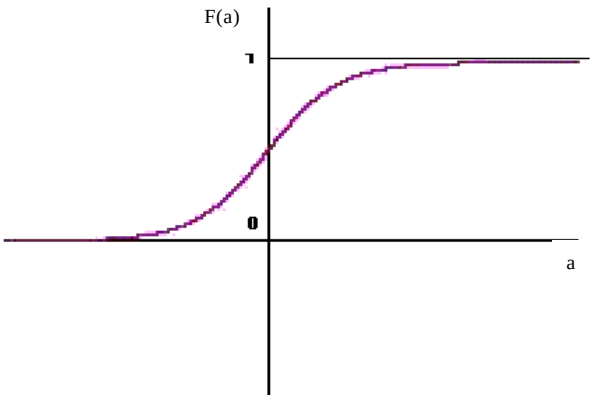


FIG. 6- FUNÇÃO DE ATIVAÇÃO LOGÍSTICA

Esta característica agiliza a implementação em computadores de modelos baseados nas mesmas. Estas funções podem ser expressas pelas equações abaixo, onde *a* representa a entrada líquida da unidade.

$$f(a) = \frac{1}{1 + e^{(-2ba)}} \qquad g(a) = \tanh(ba)$$

As equações abaixo representam as primeiras derivadas das mesmas

$$f'(a) = 2bf(1 - f) \qquad g'(a) = b(1 - g^2)$$

2.2- Os modelos de Redes Neurais Artificiais - RNAs

Uma rede de computação neural é um modelo matemático que consiste na conexão de várias unidades simples, análogas aos neurônios humanos, denominados elementos de processamento. Estes elementos são organizados em camadas, onde cada elemento tem conexões ponderadas com outros elementos, sendo que estas conexões podem tomar diferentes configurações, dependo da aplicação desejada.

A rede neural é, geralmente, constituída de uma camada de entrada, que recebe os estímulos, e de uma camada de saída, que produz a resposta. Algumas redes podem ter uma ou mais camadas internas (também chamadas ocultas ou escondidas). A capacidade computacional de uma rede neural está nas conexões entre os elementos processadores. Nos pesos que ponderam cada conexão são armazenadas as informações que a rede "aprendeu".

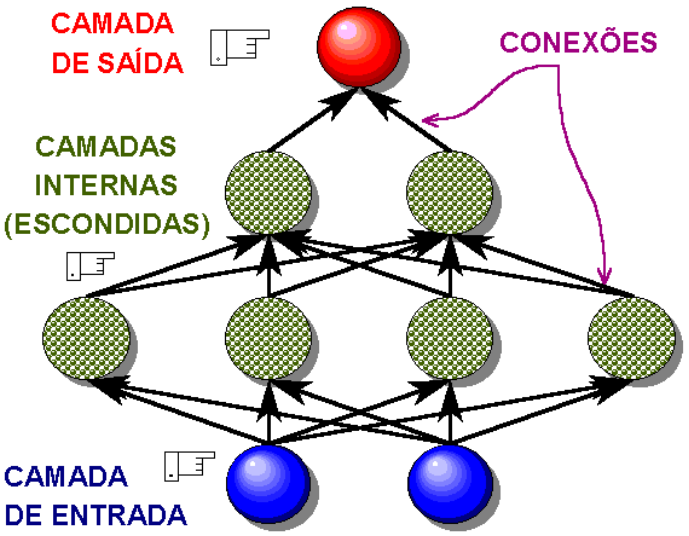


FIG. 7- REDE NEURAL ARTIFICIAL

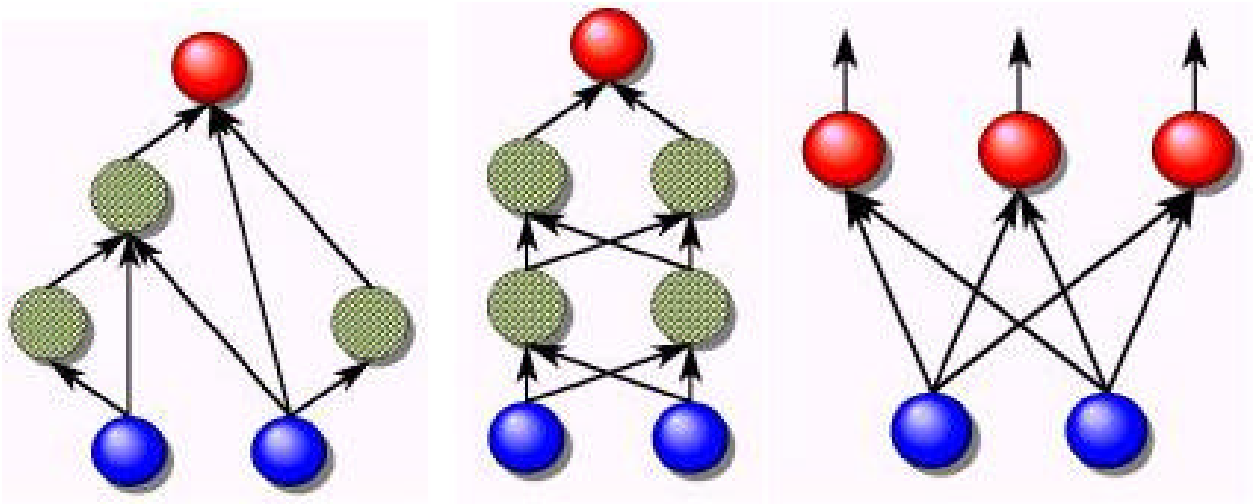
Da combinação entre elementos de processamento e conexões surgem as diferentes topologias de Redes Neurais Artificiais. Pode-se fazer uma analogia com um grafo onde os nodos são os elementos de processamento e as arestas são as conexões. A figura 7 mostra um exemplo de RNA.

2.3-Topologias

Em uma RNA os neurônios podem estar agrupados por camadas direcionadas ou não, com ligações em um sentido (para frente) ou em ambos (para frente e para trás). As redes com ligações somente para a frente, também chamadas de diretas ou *Feed-forward*, se dividem em redes monocamada e multicamadas. As redes com ligações em mais de um sentido, também chamadas de redes com ciclos, se dividem em redes recorrentes e redes competitivas.

2.3.1 Redes diretas (*feed-forward*)

São aquelas cujo grafo não apresentam ciclos. Seu aspecto geral é o da figura 8(a). Freqüentemente, é possível representar estas redes em camadas; neste caso, são chamadas redes multicamadas. No caso de serem organizadas por camadas, um neurônio de uma camada só pode se conectar a um neurônio da camada seguinte, como na figura 8(b). No caso mais simples as redes em camadas possuem uma só camada, como na figura 8(c).



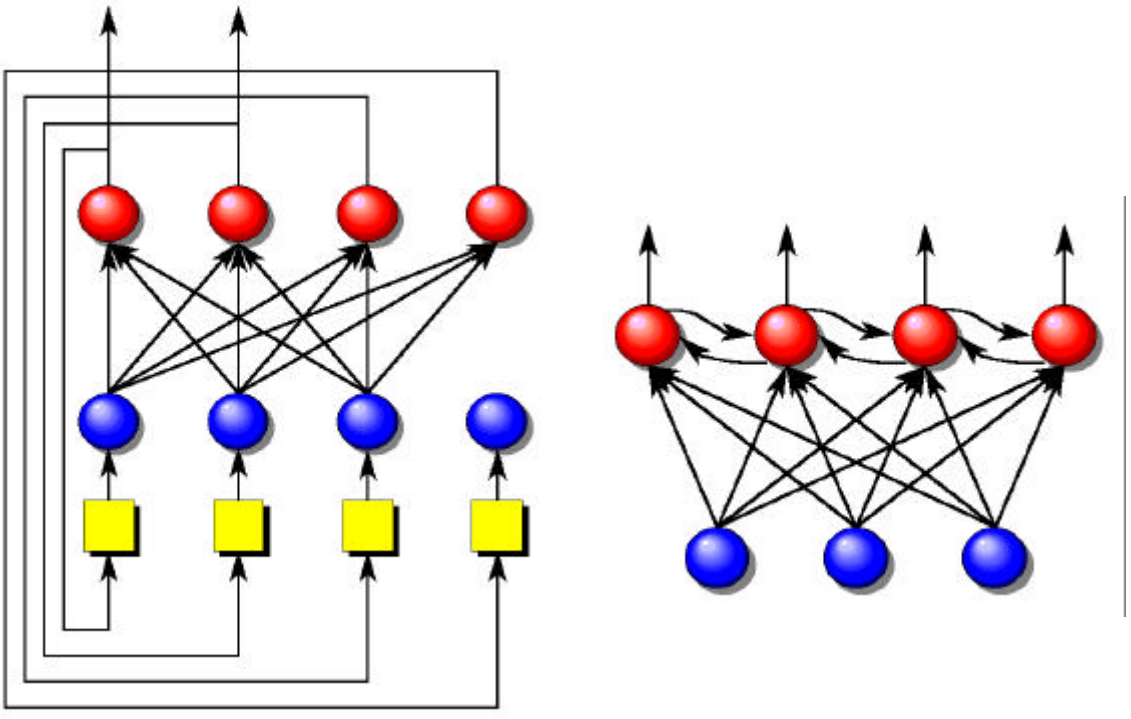
(a) (b) (c)

FIG. 8- REDES DIRETAS

No caso de possuírem mais de uma camada possuem o aspecto da figura 6b. Neste caso, os neurônios que recebem sinais de excitação pertencem à camada de entrada. Neurônios que têm sua saída como saída da rede pertencem à camada de saída. Neurônios que não pertencem a nenhuma destas camadas são internos à rede e pertencem a uma ou mais camadas internas.

2.3.2 Redes com ciclos

Redes com ciclos são aquelas cujo grafo de conectividade contém ao menos um ciclo e cujos neurônios apresentam a possibilidade de realimentarem outros neurônios. Quando, além disto, envolvem neurônios dinâmicos, são chamadas de recorrentes (figura 9a). Quando neurônios competem entre si, de forma que só as saídas de alguns são consideradas, são chamadas de redes competitivas (figura 9b).



(a) (b)

FIG. 9- REDES COM CICLOS

2.4-Treinamento e Generalização

Os sistemas de computação neural são dinâmicos e auto-adaptativos. Eles são auto-adaptáveis porque os elementos de

processamento têm capacidade de auto-ajuste. São também capazes de utilizar a experiência para modificar as respostas em certas situações. São dinâmicos porque podem estar mudando constantemente para adaptar-se a novas condições, de forma a responder satisfatoriamente a novas entradas.
Três processos compreendem a adaptabilidade dos sistemas neurais: o aprendizado; o treinamento; e, a generalização.

2.4.1 Aprendizado

Aprendizado é auto-adaptação ao nível de elemento de processamento. No processo de aprendizado, as conexões ponderadas são ajustadas para encontrar resultados específicos. Através de um processo, que chamamos treinamento e que discutiremos a seguir, o sistema se adapta, ajustando as conexões ponderadas entre os elementos de processamento.
Regras de aprendizagem são esquemas de atualização dos valores do pesos das sinapses de uma rede neural, de forma a obter da rede um padrão de processamento desejado. O processo de aprendizado pode ser feito basicamente de duas maneiras: aprendizado supervisionado ou não-supervisionado.
No aprendizado supervisionado, os exemplos das entradas e suas respectivas saídas (conjunto de padrões) são usados no treinamento da rede neural. Geralmente, este método se baseia na minimização do erro entre o valor desejado de saída das unidades da camada de saída e o valor computado pela rede para as mesmas. Baseado nessa diferença os pesos da rede são ajustados a cada apresentação de padrão, de forma a minimizar o erro na saída para o conjunto de padrões. O processo é repetido diversas vezes até que que os padrões tenham incorporado o conhecimento sobre o funcionamento dos padrões.
No aprendizado não supervisionado, apenas os exemplos de entradas são mostrados no treinamento da rede neural. As unidades respondem a características "interessantes" das entradas apresentadas no processo de treinamento. Este tipo

de treinamento está intimamente ligado com a topologia de conexão competitiva.

2.4.2 Treinamento

Treinamento é o modo pelo qual o sistema computacional neural aprende a respeito da informação que ele precisará, a fim de resolver certos problemas. O treinamento envolve a exposição do sistema a um conjunto específico de informação para encontrar um estado de auto-organização particular.

Hebb (1949) introduziu a idéia de que deu base ao desenvolvimento de redes neurais artificiais. Sua idéia básica consiste no fato de que, se um neurônio biológico recebe uma entrada de outro neurônio e ambos se encontram altamente ativados, então, a ligação entre eles é reforçada. Aplicando-se esta observação ao esquema de redes neurais artificiais poderíamos dizer que se uma unidade *i* recebe uma entrada de outra unidade *k* altamente ativada, então, a importância desta conexão deve ser aumentada, isto é, o valor do peso entre as unidades deve ser acrescido. A maneira mais simples de se representar matematicamente esta relação é representada pela equação (1), onde O_i é a saída da unidade *i* e O_k é a saída da unidade *k* e Δw_{ik} é a alteração a ser feita no peso w_{ik} que liga a unidade *k* a unidade *i*.

$$\Delta w_{ik} = O_i \; O_k \qquad (2.1)$$

Entretanto, se as duas unidades tiverem saídas negativas, isto também causará o aumento do peso entre elas e esta reação não combina com o fato fisiológico, observado por Hebb.

A equação (2.1) foi estendida e modificada e pode ser descrita pela equação (2.2), conhecida como regra de Hebb generalizada.

$$\Delta w_{ik} = h \; g(O_i, D_i) \; h(O_k, w_{ik}) \qquad (2.2)$$

A equação (2.1) pode ser vista como um caso particular da equação (2.2), onde as funções *g* e *h* simplesmente dependem do primeiro argumento e o parâmetro η é igual a um. O parâmetro η é denominado taxa de aprendizagem e D_i é a

saída desejada na unidade *i*. Algumas variações desta regra geral foram implementadas, tais como, a regra de Perceptron (Rosenblatt,1958).
Uma regra de aprendizado importante, que pode ser vista como uma variante da equação (2.2) é chamada processo de correção do erro. Neste treinamento supervisionado, o algoritmo de aprendizado consiste em alterar o valor do peso entre unidades numa forma proporcional à diferença entre as saídas desejada e computada pela rede. Esta idéia, só pode ser aplicada diretamente a uma rede de camada simples, já que no caso de existirem camadas escondidas, o valor desejado não é conhecido. Em sua forma mais simples, esta relação é matematicamente descrita pela equação (2.3).

$$\Delta w_{ik} = \eta (D_i - O_i) O_k \qquad (2.3)$$

Esta regra pode ser vista como uma variação particular da regra generalizada de Hebb (equação (2.2)). Neste caso, a função *g* é expressa como a diferença entre o valor desejado e o valor computado pela rede, enquanto a função *h* é definida como sendo igual ao valor de entrada (independente do valor do peso). A equação (2.3) pode ser generalizada, a fim de poder ser aplicada a redes de multicamadas. Esta generalização chama-se regra de retropropagação (*Back-propagation).*
Diversos outros autores desenvolveram outras regras de treinamento, como por exemplo, a Estocástica e a Competitiva.

2.4.3 Generalização

Um dos principais problemas do treinamento é a capacidade de generalização, isto é, a capacidade da rede de responder adequadamente a padrões que não fizeram parte do conjunto de treinamento. Além da habilidade de aprendizado, os sistemas computacionais neurais são também capazes de generalizar um pedaço específico ou limitado de entrada para produzir uma solução de saída. Ao fornecer-se uma entrada que nunca foi apresentada anteriormente, o sistema deve ser

capaz de generalizar esta entrada para fornecer uma resposta. Este conceito é muito importante, porque permite que o sistema forneça soluções, mesmo quando as informações de entrada são incompletas ou imprecisas. A topologia, tamanho da rede e forma de executar o treinamento da rede são bastante determinantes da capacidade de generalização.

A rede generaliza bem quando a relação entrada-saída aprendida no treinamento é representativa do problema. Uma rede super-treinada tende a aprender ruído junto com a entrada. A regularidade do mapeamento está associada a uma boa generalização. Ao adicionarmos complexidade ao modelo, por exemplo utilizando um modelo de quinta ordem ao invés de um modelo de terceira ordem, estamos aumentando a possibilidade e obter uma generalização pobre. Por ser demasiadamente complexo o modelo capta excessivos detalhes dos dados de treinamento em detrimento de propriedades mais gerais.

No caso das redes neurais os parâmetros a serem estimados são os pesos e polarizadores e portanto, quanto mais camadas e neurônios nas camadas ocultas tivermos, mais complexo será o modelo. Quanto menor a quantidade de dados mais simples deve ser o modelo.

Em geral, haverá generalização se:

$$N > \frac{w}{e} \qquad (2.4)$$

onde:

W é o número de pesos da rede.

ε é o erro admitido na fase de teste.

N é o número de exemplos de treinamento.

Por exemplo, se o erro admitido é de 10%, o número de exemplos deve ser maior que 10 x W.

Para evitar que o treinamento se especialize nos dados para os quais está sendo treinado, utiliza-se um conjunto de dados de validação, que não contribui para o treinamento. Este conjunto é usado para medir a capacidade de generalização. Os dados deste conjunto, que tipicamente possui 20% ou 30% do tamanho do conjunto de dados de treinamento, são usados

para validar, a um intervalo de épocas de treinamento, a rede obtida até àquela iteração. Isto é feito, comparando-se os erros em uma determinada iteração (época de treinamento) do conjunto de treinamento e do conjunto de validação.
Os pesos das ligações são modificados durante o treinamento de forma a minimizar o erro para os padrões apresentados. Se a quantidade de padrões apresentados é suficientemente representativa da função, a tendência durante o treinamento é que o erro, tanto do conjunto de padrões de treinamento quanto do conjunto de padrões de validação, vá diminuindo à medida que a rede vá generalizando o aprendizado, como pode ser visto na figura 10.

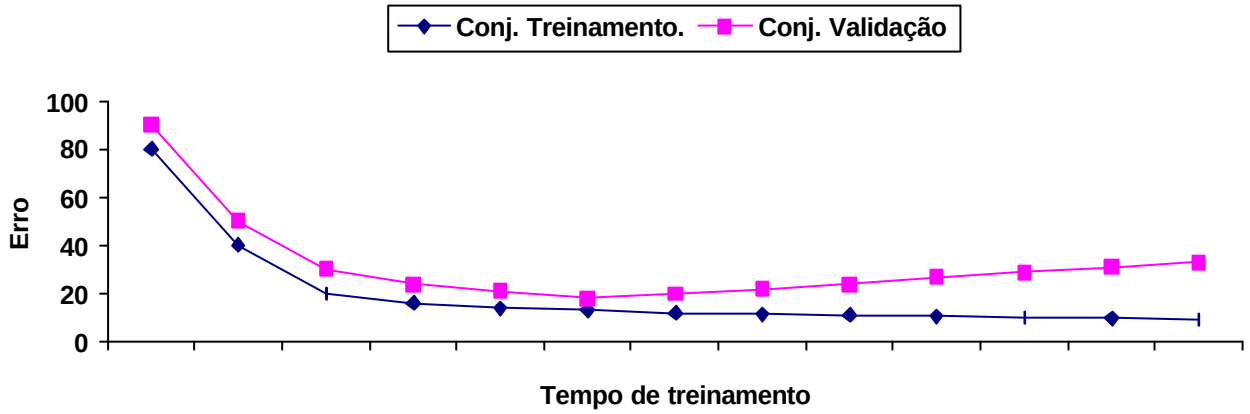


FIG. 10- COMPORTAMENTO TÍPICO DO TREINAMENTO

Entretanto, após um certo tempo o erro do conjunto de validação atinge um mínimo e começa a crescer, enquanto que o erro do conjunto de treinamento continua a cair, indicando uma tendência da rede de incorporar informações que são específicas do conjunto de treinamento, tal como, por exemplo, os erros de medidas daquele conjunto.
Uma solução é parar o treinamento quando a tendência de erro do conjunto de validação começa a divergir da tendência do conjunto de treinamento. Dessa forma, o conjunto de treinamento é usado para modificar os pesos enquanto que o conjunto de validação é usado para estimar a capacidade de generalização.
Uma outra abordagem para evitar o excesso de treinamento é limitar a capacidade da rede de absorver as correlações

espúrias entre os dados de entrada. Isto acontece basicamente quando a rede possui mais graus de liberdade (que são proporcionais ao número de ligações) do que o número de padrões de treinamento. Assim, o problema da generalização está diretamente ligado ao problema do dimensionamento da rede. Ou seja, basicamente a idéia é conseguir a menor rede que acomode as informações contidas no conjunto de dados de treinamento. O problema é justamente estimar qual é esse tamanho mínimo que permite que a rede ainda aprenda sem incorporar os dados espúrios do conjunto de treinamento.

A abordagem simples, embora bastante ineficiente, é simplesmente executar várias redes de diversos tamanhos e verificar qual é a rede mínima que consegue aprender os padrões de entrada. Mesmo que seja possível determinar a menor rede por este processo, ainda se fica muito susceptível aos parâmetros de treinamento, de forma que pode ser muito difícil determinar se a rede é muito pequena para aprender os padrões, se ela simplesmente aprende lentamente ou se devido a valores de iniciação ou parâmetros de treinamento mal escolhidos ela caiu em algum mínimo local.

A abordagem escolhida por vários autores e que trataremos a mais à frente, parte do princípio comum de iniciar o treinamento com uma rede maior do que o necessário e ir podando (removendo) partes da rede que não sejam necessárias. Como inicialmente a rede é grande, possui graus de liberdade suficientes para acomodar rapidamente as características gerais dos dados de entrada de uma forma pouco sensível às condições iniciais e aos mínimos locais. Após a acomodação inicial então a rede pode ser podada de forma a eliminar características específicas do conjunto de treinamento, favorecendo portanto os critérios desejáveis de generalidade.