OTIMIZAÇÃO DO ALGORÍTMO DE BACKPROPAGATION PELO USO DA FUNÇÃO DE ATIVAÇÃO BI-HIPERBÓLICA

Geraldo Miguez
COPPE / PESC, Universidade Federal do Rio de Janeiro, Brasil
geraldomiguez@gmail.com

Nelson Maculan Filho
COPPE / PESC, Universidade Federal do Rio de Janeiro, Brasil
maculan@cos.ufrj.br

Adilson Elias Xavier
COPPE / PESC, Universidade Federal do Rio de Janeiro, Brasil
adilson@cos.ufrj.br

Resumo

O Algoritmo de Backpropagation é uma das ferramentas mais utilizadas para o treinamento de Redes Neurais Artificiais. Entretanto, em algumas aplicações práticas ele pode ser muito lento. Para permitir uma utilização mais ampla, muitas técnicas têm sido discutidas para acelerar o seu desempenho. Este trabalho apresenta uma nova estratégia baseada no uso da Função Bi-Hiperbólica que oferece maior flexibilidade e uma avaliação computacional mais rápida. A eficiência e a capacidade de discriminação da metodologia proposta são demonstradas através de um conjunto de experimentos computacionais com problemas tradicionais da literatura.

Palavras-chave
Redes Neurais; Backpropagation; Otimização; Função Bi-hiperbólica; Inteligência Artificial

Abstract

Back propagation algorithm is one of the most used tools for training artificial neural networks. However, in some practical applications it may be very slow. To allow a broader use, many techniques were discussed to speed up its performance. This paper presents a new strategy based in the use of the Bi-hyperbolic function that offers more flexibility and a faster evaluation time. The efficiency and the discrimination capacity of the proposed methodology are shown through a set of computational experiments with traditional problems of the literature.

Keywords
Neural Networks; Backpropagation; Optimization; Bi-hyperbolic Function; Artificial Intelligence

Introdução

As redes do tipo Perceptron de Múltiplas Camadas (Multilayer Perceptrons – MLP) têm sido um dos modelos de Redes Neurais Artificiais (RNA) mais amplamente utilizados na construção de sistemas para a solução de diferentes problemas, tais como classificação de padrões (reconhecimento), controle e processamento de sinais.

Para atender esta diversidade de aplicações, é necessário um eficiente algoritmo de treinamento. O algoritmo mais utilizado tem sido o Backpropagation. É um método computacionalmente eficiente para o treinamento de redes MLPs e que resolve o problema de realizar a propagação reversa do erro em RNAs com múltiplas camadas. Entretanto, ele apresenta algumas limitações na sua utilização, dificultado a sua aplicação de uma forma mais ampla. Sendo um método baseado no uso de gradientes, existe a possibilidade de convergência para um mínimo local, falhando em encontrar o mínimo global. Apresenta, também, uma lentidão muito grande no seu processamento, mesmo nos casos em que consegue atingir o seu objetivo de apresentar um erro dentro dos limites desejados. Esta demora no processamento dificulta a sua utilização em uma gama maior de aplicações práticas, em especial em aplicações de médio e grande porte (SCHIFFMANN et al, 1994), (OTAIR et al, 2005).

Um dos fatores possivelmente responsável pela lentidão deste processo de convergência é a função de ativação usada em seus neurônios, pois, sendo o processo de aprendizagem da rede essencialmente iterativo, uma função mais lenta para ser calculada torna todo o procedimento lento. Acredita-se que a razão para isto é a saturação da função de ativação usada para as camadas ocultas e de saída, pois, uma vez que a saturação de uma unidade ocorre, o gradiente descendente assume valores muito pequenos, mesmo quando o erro de saída é grande. O problema de otimizar a eficiência e a taxa de convergência do algoritmo de backpropagation tem sido objeto de interesse de muitos pesquisadores, sendo ainda uma área aberta a novos estudos.

A proposta apresentada neste trabalho prevê a utilização de uma nova função de ativação, a Função Bi-Hiperbólica, com características que atendem às necessidades do algoritmo de backpropagation e oferecem vantagens de possibilitar uma maior flexibilidade na representação dos fenômenos modelados, com o uso de dois parâmetros, um a mais do que nas funções tradicionalmente utilizadas para esta finalidade. Isto implica na possibilidade de melhor enfrentar o problema da saturação, além de permitir melhor tratamento para evitar os mínimos locais. Outra vantagem, observada empiricamente, é a de ser computacionalmente 90,5% mais rápida de ser avaliada do que a função logística. Este resultado foi obtido através de simulação programada na linguagem FORTRAN, usando o compilador COMPAQ, em um computador tipo IBM PC, com 800 Mhz de clock (XAVIER, 2005).

Outra vantagem do uso desta Função Bi-Hiperbólica reside em possibilitar, por sua maior flexibilidade, a capacidade de poder aproximar qualquer função de uma forma mais sintética, permitindo a utilização de um menor número de neurônios, melhorando ainda mais o desempenho do algoritmo backpropagation, agindo diretamente na topologia da rede (XAVIER , 2005).

Foi desenvolvido um protótipo em MATLAB que, através de uma interface gráfica, permitiu a obtenção de resultados altamente favoráveis, apresentados posteriormente neste trabalho.

Redes Neurais Artificiais

Uma Rede Neural Artificial funciona pela criação de ligações entre unidades de processamento matemático, chamados de neurônios. O conhecimento é codificado na rede pela força destas conexões entre diferentes neurônios, chamada de peso, e pela criação de camadas de neurônios que trabalham em paralelo. O sistema aprende através de um processo de determinação do número de neurônios, ou nós, e pelo ajuste dos pesos das conexões com base nos dados usados para o treinamento. O poder computacional de uma RNA é devido

basicamente à sua estrutura paralela pesadamente distribuída e à sua habilidade de aprender e, conseqüentemente, generalizar (HAYKIN, 2001).

Os neurônios são considerados as estruturas que constituem o cérebro. O neurônio biológico é basicamente o dispositivo computacional elementar do sistema nervoso, que possui algumas entradas e uma saída, conforme o esquema que pode ser visto na Figura 1. As entradas ocorrem através das conexões sinápticas, que conectam a árvore dendrital aos axônios de outras células nervosas. Os sinais que chegam dos axônios de outras células nervosas são pulsos elétricos conhecidos como impulsos nervosos ou potenciais de ação e constituem a informação que o neurônio processará de alguma forma para produzir como saída um impulso nervoso no seu axônio (FYFE, 2000).

Os Neurônios Artificiais são as unidades de processamento das RNAs. Eles são simplificações do conhecimento que se tinha do neurônio biológico, feitas por McCulloch e Pitts (KÓVACS, 1996). O modelo desenvolvido apresenta vários terminais de entrada (X), representando os dendritos, e um terminal de saída (Y), representando o axônio. As sinapses têm seu comportamento simulado pelo acoplamento de pesos (W) a cada terminal de entrada do neurônio artificial, que podem assumir valores positivos ou negativos, emulando sinapses inibitórias ou excitatórias, conforme representado na Figura 2. A ativação do neurônio artificial é obtida através da aplicação de uma função de ativação que pode ativar ou não a saída, dependendo da soma ponderada dos valores de cada entrada atingir um limiar pré-determinado. A função de ativação limita a faixa de amplitude permitida do sinal de saída a algum valor finito. Tipicamente, a amplitude normalizada da saída de um neurônio é restrita ao intervalo unitário fechado [0, 1] ou, alternativamente, [-1, 1]. O modelo neural usado inclui uma polarização externa (*bias*), que tem o efeito de aumentar ou diminuir o argumento da função de ativação (φ), que define a saída do neurônio em termos do potencial de ativação.

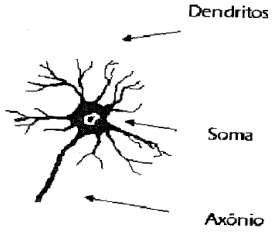


Figura 1: Neurônio do sistema nervoso central dos vertebrados

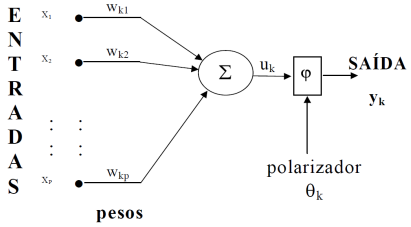


Figura 2: Neurônio Artificial

O neurônio pode ser descrito, em termos matemáticos da seguinte forma:

$$u_k = \sum_{j=1}^{p} w_{kj} x_j$$

$$v_k = u_k - \theta_k$$

$$y_k = \varphi(v_k)$$

Onde $x_1, x_2, \ldots, x_p$ são os sinais de entrada; $w_{k1}, w_{k2}, \ldots, w_{kp}$ são os pesos sinápticos do neurônio k; u_k é a saída proveniente da combinação linear dos sinais de entrada e dos pesos; θ_k é o bias; $\varphi(\bullet)$ é a função de ativação; e y_k é o sinal de saída do neurônio (HAIKIN, 2001).

Funções de Ativação

Um dos componentes mais importantes do neurônio artificial é a função de ativação ou transferência. Ela tem por objetivo limitar a amplitude válida do sinal de saída do neurônio a um valor finito. Normalmente, esta amplitude normalizada se encontra em um intervalo fechado unitário [0, 1] ou, em alguns casos, [-1, 1].

As funções de ativação mais comumente utilizadas e disponibilizadas na literatura são apresentadas abaixo. Também são descritas as suas derivadas, que têm grande importância no

método de treinamento de redes neurais artificiais conhecido como Backpropagation (XAVIER, 2005), (HAYKIN,2001).

a) Função Degrau

$$\varphi_1(v) = \begin{cases} 1 & se \quad v \geq 0 \\ 0 & se \quad v < 0 \end{cases}$$

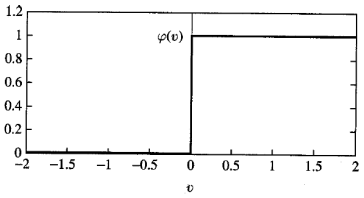
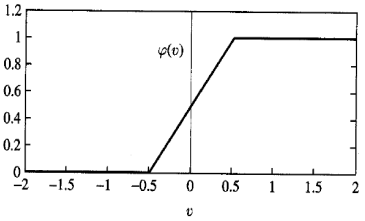


Figura 3: Função Degrau

A derivada desta função é $\varphi_1'(v) = 0$ para $\forall \quad v \neq 0$ e não é definida para $v = 0$. A descontinuidade na origem associada ao valor nulo da derivada nos demais pontos restringe muito a utilidade prática desta função. Ela está representada na Figura 3.

b) Função Patamar

$$\varphi_2(v,b) = \begin{cases} 0, & se \quad v \leq -b; \\ (v+b)/2b, & se \quad -b < v < b; \\ 1, & se \quad v > b; \end{cases}$$



Sendo $b = 1/2 \tan\alpha$, onde α é o ângulo de inclinação. Figura 4: Função Patamar

A representação desta função está na Figura 4.

A sua derivada não é definida nos pontos $v = -1/2$ e $v = 1/2$, nos demais valores assume:

$$\varphi_2'(v,b) = \begin{cases} 0, & para \quad v < -b; \\ 1/2b, & para \quad -b < v < b; \\ 0, & para \quad v > b; \end{cases}$$

A insensibilidade da derivada fora do intervalo (-*b*, *b*) limita consideravelmente o uso prático dessa função de ativação $\varphi_2(\bullet)$.

c) Função Logística

Esta é a forma de função de ativação mais utilizada na construção de redes neurais artificiais. Ela é definida como uma função estritamente crescente que exibe um balanço entre o comportamento linear e o comportamento não-linear. Ela é definida por:

$$\varphi_3(v,a) = \frac{1}{1+e^{-av}}$$

onde ***a*** é o parâmetro de declividade da função logística.

A derivada da Função Logística é definida por:

$$\varphi_3'(v,a) = a\varphi_3(v,a)(1-\varphi_3(v,a))$$

Segundo Xavier (XAVIER, 2005), a Função Logística oferece a importante flexibilidade dada por sua inclinação na origem, $\varphi'_3(0,a) = a/4$, ser variável com o parâmetro ***a***. Através da variação deste parâmetro ***a*** foram obtidas Funções Logísticas de diferentes declividades, como pode ser visto na Figura 5. Além disso, a Função Logística apresenta propriedades de simetria e de completa diferenciabilidade, ou seja, pertence à classe de funções c^{∞}.

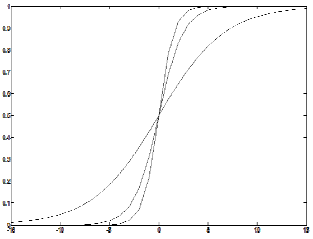


Figura 5: Função Logística – Efeito da variação do parâmetro ***a***

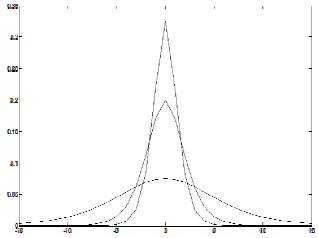


Figura 6: Derivadas da Função Logística variando o parâmetro ***a***

Variando o parâmetro ***a*** foram obtidas as derivadas da função logística de diferentes declividades, apresentadas na Figura 6.

d) Função de Elliott (ELLIOTT,1993) (XAVIER, 2005)
Esta função, apresentada na Figura 7, é definida por:

$$\varphi_4(v)=\left(\frac{v}{1+|v|}+1\right)/2$$

A sua derivada, apresentada na Figura 8, é definida por:

$$\varphi_4'(v)=\frac{1}{2(1+|v|)^2}+1$$

Ela apresenta a inclinação de sua derivada na origem invariante, $\varphi'_4(0)=1/2$, independente de qualquer transformação de escala, fato que limita fortemente a flexibilidade dessa função e seu decorrente uso prático.

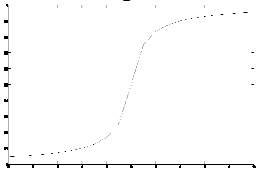


Figura 7: Função de Elliot

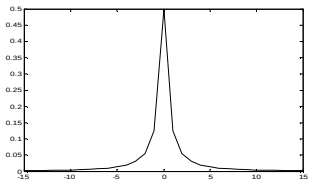


Figura 8: Derivada da Função de Elliot

e) Função Bi-Hiperbólica (XAVIER, 2005)
A função Bi-Hiperbólica Assimétrica em sua forma mais geral é definida por:

$$\phi x(v,\lambda,\tau_1,\tau_2)=\sqrt{\lambda^2(v+1/4\lambda)^2+\tau_1^{\,2}})-\sqrt{\lambda^2(v-1/4\lambda)^2+\tau_2^{\,2}})+1/2$$

Sua derivada é definida por:

$$\phi' x(v,\lambda,\tau_1,\tau_2)=\frac{\lambda^2(v+1/4\lambda)}{\sqrt{\lambda^2(v+1/4\lambda)^2+\tau_1^{\,2}}}-\frac{\lambda^2(v-1/4\lambda)}{\sqrt{\lambda^2(v-1/4\lambda)^2+\tau_2^{\,2}}}$$

A função $\phi x(\bullet,\lambda,\tau_1,\tau_2)$ apresenta a desejada propriedade de possuir diferenciabilidade infinita, ou seja, pertence à classe de funções c^{∞}, o que permitirá a sua utilização de algoritmos de otimização mais robustos, além de apresentar as seguintes propriedades triviais consentâneas às demais funções de ativação:

$$\lim_{v\to-\infty}\phi x(v,\lambda,\tau_1,\tau_2)=0$$

$$\lim_{v\to\infty}\phi x(v,\lambda,\tau_1,\tau_2)=1$$

$$\lim_{v\to-\infty}\phi' x(v,\lambda,\tau_1,\tau_2)=0$$

$$\lim_{v\to\infty}\phi' x(v,\lambda,\tau_1,\tau_2)=0$$

Se considerarmos o caso particular obtido igualando-se os valores dos parâmetros $\tau_1=\tau_2=\tau$, a função $\phi x(\bullet,\lambda,\tau_1,\tau_2)\underline{\Delta}\phi x(\bullet,\lambda,\tau)$, assume uma forma mais consentânea à outras funções de ativação, tendo imagem no intervalo [0, 1] e oferecendo a propriedade de simetria, conforme retratado pelos gráficos da Figura 9 e Figura 10.

$$\varphi x(v,\lambda,\tau)=\sqrt{\lambda^2(v+1/4\lambda)^2+\tau^2)}-\sqrt{\lambda^2(v-1/4\lambda)^2+\tau^2)}+1/2$$

$$\varphi' x(v,\lambda,\tau) = \frac{\lambda^2(v+1/4\lambda)}{\sqrt{\lambda^2(v+1/4\lambda)^2+\tau^2}} - \frac{\lambda^2(v-1/4\lambda)}{\sqrt{\lambda^2(v-1/4\lambda)^2+\tau^2}}$$

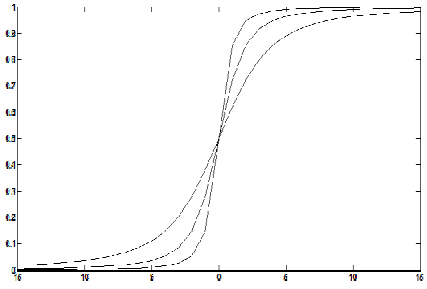


Figura 9: Curvas Bi-Hiperbólica variando λ com τ fixo

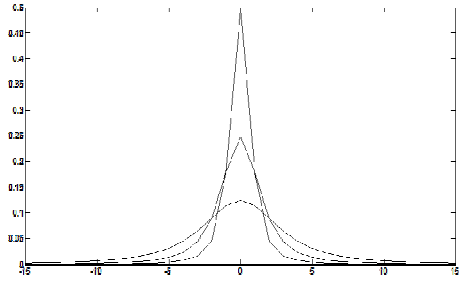


Figura 10: Derivadas das Curvas Bi-Hiperbólica variando λ com τ fixo

A função $[\varphi x(v,\lambda,\tau) - 1/2]$ é anti-simétrica, ou seja:

$$\varphi x(v,\lambda,\tau) - 1/2 = -[\varphi x(-v,\lambda,\tau) - 1/2]$$

No ponto $v = 0$, são observados os seguintes valores para a função φx e sua derivada:

$$\varphi x(0,\lambda,\tau) = 1/2$$

$$\varphi' x(0,\lambda,\tau) = \frac{\lambda}{2\sqrt{1/16+\tau^2}},$$

$$\lim_{\tau\to 0} \varphi' x(0,\lambda,\tau) = 2\lambda$$

Na figura 9, são mostrados gráficos na forma simétrica da função Bi-Hiperbólica correspondentes a quatro valores diferentes para o parâmetro λ, mantendo-se o parâmetro τ constante. Pode-se ver um efeito similar àquele produzido pela variação do parâmetro ***a*** na função logística. Dessa forma pode-se associar o parâmetro λ à inclinação da função na origem.

A função $\varphi x(\bullet,\lambda,\tau)$ apresenta, ademais, os seguintes comportamentos assintóticos:

$$\lim_{\lambda\to\infty} \varphi x(v,\lambda,\tau) = \varphi_1 x(v)$$

$$\lim_{\tau\to 0} \varphi x(v,\lambda,\tau) = \varphi_2\left(v, \frac{1}{4\lambda}\right)$$

Como bem ressalta Xavier (XAVIER, 2005), a existência de dois parâmetros, um a mais que as demais funções de ativação, enseja a essa função uma maior flexibilidade para representar mais adequadamente os fenômenos modelados com redes neurais.

Numa rede neural multicamadas, por exemplo, essa maior flexibilidade engendra, certamente à função de ativação Bi-Hiperbólica o poder de aproximar qualquer função de uma forma mais sintética, com menor número de neurônios. Através da manipulação conveniente de seus parâmetros, a função $\varphi x(\bullet,\lambda,\tau)$, oferece ademais a possibilidade de poder enfrentar mais convenientemente o desastroso fenômemo de saturação, além de poder evitar um indesejado mínimo local. Um forte indicador destas possibilidades pode ser observado no gráfico de sua derivada, na Figura 10, onde pode-se ver que ela apresenta uma taxa de variação do crescimento bem mais acentuada do que o das derivadas das demais funções.

Arquitetura da rede neural

O projeto de uma rede neural artificial começa com a seleção de uma arquitetura apropriada e com o treinamento através dos exemplos e de um algoritmo específico. Esta fase é a chamada de aprendizagem. Em seguida é feito a avaliação com os dados não usados no treinamento para determinar o seu desempenho na tarefa específica. Esta fase é a chamada de generalização. O projeto de uma rede neural artificial é baseado diretamente nos dados do

mundo real, fazendo com que a rede forneça um modelo implícito do ambiente no qual está inserida, além de realizar a função de processamento de informações.

No projeto de uma rede neural do tipo MLP o dimensionamento das camadas de entrada e de saída será sempre determinado pela natureza do próprio problema.

Entretanto, a determinação de quantas camadas ocultas e de quantos neurônios estas devem possuir, não é uma tarefa que permita uma reposta exata. Existem, para este problema, soluções aproximadas, as chamadas heurísticas, que procuram estimar estas variáveis. Estas heurísticas expõem sempre o compromisso entre a convergência e a generalização da rede. Considera-se Convergência a capacidade da rede de aprender todos os padrões de entrada usados no seu treinamento. Uma rede muito pequena em relação ao problema em análise não será capaz de aprender os dados de treinamento do problema, ou seja, a rede não possuirá parâmetros ou pesos sinápticos suficientes (HECHT-NIELSEN, 1989) (HAYKIN, 2001).

Generalização é a capacidade da rede neural responder adequadamente a padrões fora dos usados no treinamento. Uma rede muito grande, com número de neurônios muito superior ao necessário, não responderá corretamente a estes novos padrões e perderá a capacidade de generalizar, uma vez que, durante o processo de treinamento a ajuste dos pesos sinápticos da rede a levarão a memorizar especificamente estes vetores de entrada e o ruído presente nestes dados de treinamento.

A capacidade de generalização de uma rede neural é afetada pelo tamanho e eficiência dos dados de treinamento, pela arquitetura da rede e número de processadores nas camadas ocultas e pela complexidade do problema. Na prática, as heurísticas são utilizadas em conjunto com séries de tentativas e ajustes na arquitetura e definições da rede. O principal objetivo é obter uma rede que generalize, ao invés de memorizar os padrões usados no treinamento (STATHAKIS, 2009), (HORNIK, 1989) e (HECHT-NIELSEN, 1989).

Aprendizagem

A propriedade mais importante de uma Rede Neural Artificial é a sua capacidade de aprender a partir do seu ambiente e melhorar seu desempenho através do aprendizado, que se resume no problema de obter um conjunto de parâmetros livres que permita à rede atingir o desempenho desejado. O tipo de aprendizagem é determinado pela forma através da qual é efetuada a mudança nos parâmetros.

Neste processo primeiramente a rede é estimulada pelo ambiente e sofre mudanças em seus parâmetros livres como resultado deste estímulo. Devido às mudanças ocorridas em sua estrutura interna, ela passa a responder de uma nova forma ao ambiente.

Os dois paradigmas básicos de aprendizagem são o aprendizado através de um tutor (Aprendizado Supervisionado) e o aprendizado sem um tutor (Aprendizado Não-Supervisionado). Uma terceira forma chamada de Aprendizagem por Reforço utiliza um crítico.

No Aprendizado Supervisionado, uma série de padrões, representados pelos vetores de entrada, é associada com os resultados desejados como resposta, é apresentado à rede. Os parâmetros internos da rede, chamados de pesos sinápticos, são alterados sistematicamente de forma a aproximar os resultados obtidos aos das respostas desejadas. Este procedimento consiste em minimizar os erros obtidos na comparação entre os resultados desejados e os calculados para os padrões usados no treinamento**.** (HAYKIN, 2001).

Algoritmo de backpropagation

O treinamento de um Perceptron de Múltiplas Camadas (MLP) consiste em ajustar os pesos e os thresholds (bias) de suas unidades para que a classificação desejada seja obtida. Quando um padrão é inicialmente apresentado à rede, ela produz uma saída e, após medir a distância entre a resposta atual e a desejada, são realizados os ajustes apropriados nos pesos de modo a reduzir esta distância. Este procedimento é conhecido como Regra Delta.

Esse tipo de rede apresenta soluções para funções linearmente não-separáveis e necessita de um algoritmo de treinamento capaz de definir de forma automática os pesos. O algoritmo mais utilizado para o treinamento destas redes MLP é uma generalização da Regra Delta denominada de Backpropagation.

Durante o treinamento com o algoritmo Backpropagation, a rede opera em uma seqüência de dois passos. No primeiro, um padrão é apresentado à camada de entrada da rede. O sinal resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No segundo passo, a saída obtida é comparada à saída desejada para esse padrão particular. Se esta não estiver correta, o erro é calculado. Este erro é propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados à medida que o erro é retropropagado.

Na Regra Delta padrão é implementado um gradiente descendente no quadrado da soma do erro para funções de ativação lineares. Entretanto, como a superfície do erro pode não ser tão simples, as redes ficam sujeitas aos problemas de mínimos locais.

A Regra Delta Generalizada, ou Backpropagation, funciona quando são utilizadas na rede unidades com uma função de ativação semi-linear, que é uma função diferenciável e não decrescente.

A Taxa de Aprendizado é uma constante de proporcionalidade no intervalo [0,1], pois este procedimento de aprendizado requer apenas que a mudança no peso seja proporcional à meta. Entretanto, como o verdadeiro gradiente descendente requer que sejam tomados passos infinitesimais, quanto maior for essa constante, maior será a mudança nos pesos, aumentando a velocidade do aprendizado. Tal situação pode levar a uma oscilação do modelo na superfície de erro. Procura-se, então, utilizar a maior taxa de aprendizado possível que não leve a uma oscilação, resultando em um aprendizado mais rápido. O treinamento das redes MLP com backpropagation pode demandar muitos passos no conjunto de treinamento, resultando em um tempo de treinamento consideravelmente longo. Se for encontrado um mínimo local, o erro para o conjunto de treinamento pára de diminuir e estaciona em um valor maior que o aceitável (HAYKIN, 2001).

A utilização da função de ativação Bi-Hiperbólica apresenta uma vantagem grande por possuir dois parâmetros que ajudam a obter um ajuste mais preciso. Outro fator que beneficia este uso é dado pela mudança maior na inclinação de sua derivada, conforme pode ser visto na Figura 10, o que contribui para diminuir o problema da saturação, que ocorre muitas vezes no treinamento das redes neurais (XAVIER, 2005).

Estudo comparativo

A proposta apresentada neste trabalho para o problema de otimizar a eficiência e a taxa de convergência do algoritmo de Backpropagation, prevê a utilização de uma nova função de ativação, a Função Bi-Hiperbólica, com características que atendem às necessidades do algoritmo de backpropagation. Ela oferece a vantagem de possibilitar maior flexibilidade na representação dos fenômenos modelados. Conta com o uso de dois parâmetros, um a mais do que nas funções tradicionalmente utilizadas para esta finalidade. Isto implica em melhor enfrentar o problema da saturação, além de permitir tratamento para evitar os mínimos locais. Outra vantagem, observada empiricamente, é a de ser computacionalmente 90,5% mais rápida de ser avaliada do que a função logística. Além disso, a sua maior flexibilidade possibilita a capacidade de poder aproximar qualquer função de uma forma mais sintética, permitindo a utilização de um menor número de neurônios. Isto melhora ainda mais o desempenho do algoritmo backpropagation, agindo diretamente na topologia da rede (XAVIER, 2005).

Para permitir uma avaliação destas características descritas, comparando-as com a função de ativação tradicionalmente utilizada, foi desenvolvido um protótipo em MATLAB, que através de uma interface gráfica, permitiu a obtenção de resultados altamente favoráveis, apresentados posteriormente neste trabalho. Ele apresenta as funções necessárias aos

treinamentos e testes, permitindo a execução do ciclo de treinamento e a verificação dos resultados obtidos pela comparação de desempenho com o modelo usando a função logística.

Foi adotada no protótipo uma rede neuronal artificial do tipo MLP, progressiva e completamente conectada. O número de nós fonte na camada de entrada da rede é determinado pela dimensionalidade do espaço de observação, que é responsável pela geração dos sinais de entrada. O número de neurônios na camada de saída é determinado pela dimensionalidade requerida da resposta desejada. A existência de camadas ocultas se deve para permitir a extração de estatísticas de ordem superior de algum desconhecido processo aleatório subjacente, responsável pelo "comportamento" dos dados de entrada, processo sobre o qual a rede está tentando adquirir conhecimento. Este é um valor arbitrário e pode variar em função da análise do desempenho do modelo. Outro ponto importante se refere à determinação do número de neurônios em cada uma das camadas escondidas. Como não existem regras determinadas para tal especificação, foi adotada a heurística proposta por Hecht-Nielsen (HECHT-NIELSEN, 1989).

Para possibilitar a avaliação da função proposta, o protótipo desenvolvido faz o treinamento em duas redes distintas, com os mesmos parâmetros básicos e com o uso de funções de ativação diferenciadas. Uma das redes utiliza como função de ativação a Função Logística e, a outra rede, utiliza como função de ativação a Função Bi-Hiperbólica.

Para a execução dos testes foram utilizados os seguintes parâmetros em comum nos dois modelos:

a) Topologia inicial da Rede Neural:
   - Uma camada externa com 10 nós, um para cada uma dos nove atributos descritivos das características observadas e mais um para o controle do bias;
   - Uma camada escondida com 21 nós, definida com base na heurística proposta por Hecht-Nielsen (HECHT-NIELSEN, 1989).
   - Uma camada de saída com 1 nó;
b) Nível de Erro Médio Quadrático considerado: menor que 0,001;
c) Taxa de aprendizado: 0,05
d) Amostra usada no treinamento: 200 instâncias;
e) Amostra usada para avaliação do modelo: 483 instâncias;

Base de Dados para teste do modelo

Para possibilitar a obtenção de dados comparativos, foi utilizada a base de dados conhecida como "Wisconsin Breast Cancer Data", disponível no site *http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)* da *University of Wisconsin-Madison*. Ela tem sido bastante utilizada em artigos publicados na área médica e de reconhecimento de padões, facilitando as comparações com os resultados a serem obtidos (PRECHELT, 1994).

É uma base com um razoável número de amostras, atributos e padrões bem definidos, sendo formada por dados de amostras obtidas através da realização de biópsias em massas com suspeitas de malignidade, encontradas em exames de mamas humanas. Cada amostra apresenta um identificador e nove atributos descritivos das características observadas, que utilizam uma escala numérica padronizada. A cada amostra está associado o resultado da avaliação feita por especialistas, classificando-as como benignas (resultado negativo) ou malignas (resultado positivo). Foram utilizadas 683 amostras, sendo 444 classificadas como benignas (65 %) e 239 classificadas como malignas (35 %) (MANGASARIAN,1990), (WOLBERG,1990).

Resultados preliminares

Foi feita uma avaliação preliminar utilizando-se parâmetros básicos que apresentaram um resultado bastante animador. Para atingir o mesmo grau de acertos, com a mesma arquitetura, a convergência do modelo com a Função Bi-Hiperbólica necessitou de apenas

duas épocas, enquanto que o modelo equivalente utilizando a Função Logística convergiu em 62 épocas, ou seja, em número de iterações de aproximadamente 3% daquele observado no modelo que utilizou a ativação pela Função Logística. Considerando a topologia da rede, estes resultados foram obtidos quando o número de neurônios desta configuração do modelo, que utiliza a Função Bi-Hiperbólica, foi reduzido para apenas sete, contra os 21 utilizados no modelo com a ativação pela Função Logística. Esta redução tem uma importante conseqüência diminuindo substancialmente o número de iterações e em muitos casos, o poderá ser verificado a diminuição não só do tempo de processamento, como da complexidade computacional, uma vez que este número de neurônios não é mais função direta do número de atributos apresentados na camada de entrada, o que ocorre quando é utilizada, por exemplo, a heurística proposta por Hecht-Nielsen (HECHT-NIELSEN, 1989) que foi a que apresentou o melhor resultado para o modelo que usou a Função Logística.

Para a preparação inicial dos dados, foi feita uma aleatorização das instâncias, para evitar alguma tendência não conhecida devido a, por exemplo, a temporalidade da obtenção das amostras. Foi feito também uma normalização dos atributos originais para uma escala de valores entre zero e um. Nenhum destes procedimentos altera as características das amostras, visando apenas facilitar a visualização dos dados.

Foi feito um teste de sensibilidade para os parâmetros em ambos os modelos. Assim, estão apresentados abaixo os parâmetros que ofereceram o melhor resultado em termos de acerto e de número de épocas (iterações usadas no treinamento com o conjunto de amostras destacado para tal fim).

Modelo com Função de Ativação usando a Curva Logística

O parâmetro variável da curva Logística que apresentou o melhor resultado, em termos de acertos, foi ***a*** igual a 0,3 que convergiu em 62 épocas, apresentando sete diagnósticos errados, o que corresponde a um percentual menor que 1,5% de respostas erradas. Um exemplo da tela do modelo está apresentado na Figura 11.

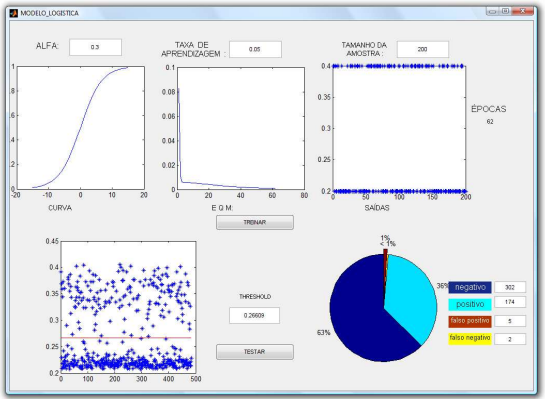


Figura 11: Tela do modelo para a Curva Logística

Um dos parâmetros mais importantes na definição de uma RNA é o número de neurônios na camada oculta, uma vez que, quanto maior for esse valor, maior será o número de pesos a serem ajustados. Para avaliar a influência do número de neurônios na camada oculta sobre o erro quadrático médio, foram feitos treinamentos independentes da rede utilizando um número variável destes neurônios.

Foram utilizadas arquiteturas contendo de 21 neurônios na camada oculta, valor obtido pelo uso da heurística proposta por Hecht-Nielsen (HECHT-NIELSEN, 1989), até o limite experimental de 5 neurônios ocultos. Para evitar a influência da inicialização dos pesos por valores aleatórios, todos os testes forma feitos com os mesmos valores iniciais para os pesos das ligações entre os neurônios das diversas camadas.

Considerando-se apenas as redes que apresentaram o melhor resultado obtido, com sete diagnósticos errados em 483 instâncias avaliadas, obtivemos os valores apresentados na

Figura 12. Podemos verificar que, em alguns casos, o mesmo resultado foi obtido por uma mesma arquitetura, mas com a utilização de parâmetros diferentes.

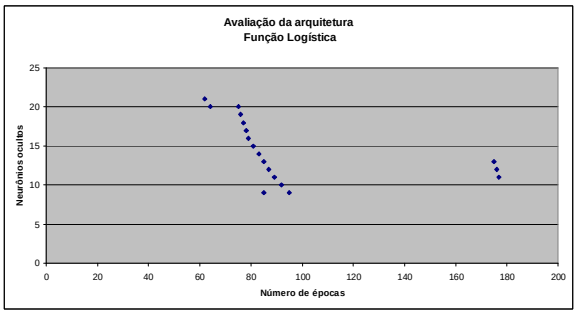


Figura 12: Avaliação das arquiteturas para a Função Logística

Modelo com ativação pela Função Bi-Hiperbólica

Para possibilitar uma avaliação comparativa do desempenho do modelo que utiliza a Função Bi-Hiperbólica, foram feitos testes variando conjuntamente o parâmetro λ, que pode ser associado com a inclinação da curva na origem, e o parâmetro τ, pode ser associado com o afastamento da curva às duas assíntotas horizontais.

Foram feitos os treinamentos e avaliações combinando entre si estes parâmetros. O melhor resultado obtido, em termos de acertos, foi a obtenção de sete diagnósticos errados. Isto foi obtido em 371 combinações no total, sendo que em 21 delas este resultado foi obtido com apenas 2 épocas, e em 25 destas com apenas 7 neurônios na camada oculta. Isto demonstra o enorme poder de convergência do modelo, bem como a sua capacidade de operar com uma rede de arquitetura com menos neurônios, o que facilita o seu uso em ambientes computacionais com menos recursos disponíveis. Um exemplo da tela do modelo está apresentado na Figura 13.

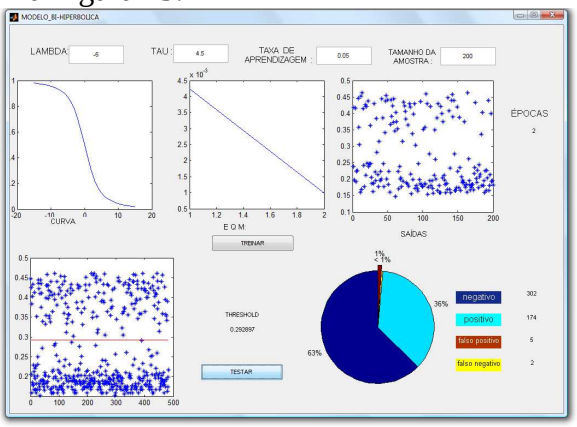


Figura 13: Tela do modelo Função Bi-Hiperbólica com 7 neurônios ocultos

O processamento deste modelo com a variação dos parâmetros citada anteriormente permitiu, também a obtenção de outros resultados muito interessantes, como por exemplo, considerando como melhor resultado o número de épocas, importante no caso de sistemas computacionais mais lentos ou com necessidade de treinamento mais rápido, foram obtidas em 68 combinações que convergiram em apenas uma época e que apresentaram apenas entre 8 e 9 diagnósticos errados. Considerando-se apenas as redes que apresentaram o melhor resultado obtido, com sete diagnósticos errados em 483 instâncias avaliadas, obtivemos os valores apresentados na Figura 14. Podemos verificar que, em alguns casos, o mesmo resultado foi obtido por uma mesma arquitetura, mas com a utilização de parametrização diferente.

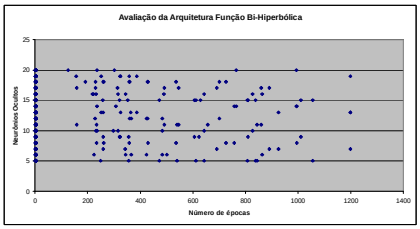


Figura 14: Avaliação das arquiteturas para a Função Bi-Hiperbólica

Conclusões

Os resultados obtidos demonstraram a grande viabilidade de utilização da função de ativação Bi-Hiperbólica, ecoando as previsões de maior capacidade de generalização, convergência mais rápida, maior velocidade de cálculo e arquitetura de rede com menor número de neurônios.

A arquitetura da rede utilizada com a função de ativação Bi-Hiperbólica possibilitou maior rapidez no treinamento, menor consumo de recursos e maior precisão na obtenção de resultados.

Outro fator importante que se pode inferir dos resultados é que a atividade de configuração da arquitetura da rede com o uso desta função, que normalmente é obtida através de processos heurísticos e de tentativas e erros, fica facilitada uma vez que uma ampla combinação de parâmetros diferentes possibilita a obtenção dos resultados desejados.

Referências

ELLIOTT, David L. A Better Activation Function for Articial Neural Networks, Institute for Systems Research, ISR Technical Report TR 93-8, 1993.

FYFE, Colin. Artificial Neural Networks, Departament of Computing and Information Systems. The University of Paisley, 2000.

HAYKIN, S. *Redes Neurais: princípios e prática*. 2. ed. Porto Alegre, Bookman, 2001.

HECHT-NIELSEN , R. Theory of the Backpropagation Neural Network; Neural Networks, 1989. IJCNN., International Joint Conference . pp 593 – 605. Washington, USA

HORNIK, K. Multilayer Feedforward Networks are Universal Approximatorz, Neural Networks, Vol. 2, pp. 359-366, 1989.

KÓVACS, Z. L., *Redes neurais artificiais: fundamentos e aplicações*. São Paulo, Edição Acadêmica, 1996.

OTAIR, M. A., SALAMEH, W. A., Speeding Up Back-Propagation Neural Networks, in Proceedings of the 2005 Informing Science and IT Education Joint Conference, Flagstaff, Arizona, USA.

SCHIFFMANN W., JOOST M., WERNER, R., Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons, University of Koblenz, Institute of Physics, Koblenz, 1994.

STATHAKIS, D. How many hidden layers and nodes? International Journal of Remote Sensing Vol. 30, No. 8, 20 April 2009, 2133–2147

XAVIER, Adilson Elias, Uma Função de Ativação para Redes Neurais Artificiais Mais Flexível e Poderosa e Mais Rápida. Learning and Nonlinear Models – Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. 1, No. 5. PP. 276-282, 2005.

PRECHELT, L.,Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules, Universit at Karlsruhe, Technical Report 21/94, 1994

MANGASARIAN, O. L., Wolberg, W. H., "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.

WOLBERG, W. H., MANGASARIAN, O. L., "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990, pp 9193-9196.