Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. 6, No. 2, pp. 142-153, 2008

REDES NEURAIS ARTIFICIAIS COM FUNÇÕES DE ATIVAÇÃO COMPLEMENTO LOG-LOG E PROBIT PARA APROXIMAR FUNÇÕES NA PRESENÇA DE OBSERVAÇÕES EXTREMAS

Gecynalda S. da S. Gomes e Teresa B. Ludermir
Centro de Informática, Universidade Federal de Pernambuco
Av. Prof. Luis Freire, s/n, Cidade Universitária, Recife/PE, 50732-970, Brasil.
{gssg, tbl}@cin.ufpe.br

Resumo - Em redes neurais artificiais as funções de ativação mais utilizadas na prática são a função sigmóide logística e a função tangente hiperbólica, dependendo das características dos dados. Entretanto, a escolha da função de ativação pode influenciar fortemente o desempenho e a complexidade da rede neural. Por isso, propomos o uso de duas funções simples, complemento log-log e probit, como funções de ativação com o objetivo de melhorar o desempenho e a complexidade da rede neural mesmo na presença de observações extremas (*outliers*).

Palavra chave - Redes Neurais, Funções de Ativação, Complemento log-log, Probit, Observações Extremas.

1. Introdução

Modelos de redes neurais artificiais (RNAs) são alternativos ao uso de modelos de regressão, principalmente, quando se trata de dados com características binárias. Para resolver problemas de classificação binária, geralmente, modelos de regressão binomial são utilizados. Esse tipo de modelo é um caso especial dos Modelos Lineares Generalizados (MLGs). A estrutura geral de um MLG é formada por uma componente aleatória, uma componente sistemática e uma função monotônica diferenciável, conhecida como função de ligação. No modelo de regressão binomial, as funções de ligação mais conhecidas são a logit (inversa da sigmóide logística), a probit e a complemento log-log, porém, a mais utilizada é a função logit, devido ao fato deste modelo fornecer o valor da razão de chances (*odds ratio*), o que não acontece quando as outras funções de ligação são utilizadas.

Em RNAs, uma das limitações é justamente a dificuldade de fornecer o valor da razão de chances de cada variável estudada. Por isso, na maioria das vezes pesquisadores optam pela regressão com o modelo logístico binomial, principalmente, na área de biomedicina. Segundo a Pubmed [25], no início de 2008, foram registrados 62.425 trabalhos que utilizaram regressão logística contra 14.105 que utilizaram redes neurais. Entretanto, o uso de RNAs fornece uma alternativa a essas análises, particularmente nas situações em que as variáveis dependentes e independentes exibem determinados relacionamentos não-lineares complexos, devido ao fato de que as RNAs são ferramentas úteis para o reconhecimento de padrões ou classificação de dados mesmo na presença de ruídos ou dados incompletos. Estes e outros fatores fazem com que o uso de modelos com redes neurais aumente a cada ano, segundo a Pubmed, o início de 2009 já registra 21.924 trabalhos com redes neurais, 55% a mais em relação ao ano anterior, contra 69.815 trabalhos com regressão logística, 12% a mais em relação ao ano anterior.

No modelo neural podem ser identificados três elementos básicos tais como um conjunto de sinapses, um somatório e uma função de ativação [15]. As funções de ativação mais utilizadas na prática são a função sigmóide logística e a função tangente hiperbólica, dependendo das características dos dados. Entretanto, alguns estudos têm mostrado a importância das funções de ativação para o aprendizado da rede neural. Hornik [16, 17] utilizou funções de ativação não-polinomiais. Singh e Chandra [31] propuseram uma classe de funções sigmóides. Skoundrianos e Tzafestas [32] propuseram uma nova função de ativação sigmoidal com bons resultados para modelagem de sistemas dinâmicos de tempo discreto. Ma e Khorasani [21] usaram a função polinomial Hermite com resultados satisfatórios. Wen e Ma [33] propuseram a função *Max-Piecewise-Linear* (MPWL) para aproximação de funções. Gomes e Ludermir [14] usaram as funções complemento log-log e probit para mostrar que quando os dados seguem uma distribuição binomial com características complemento log-log e probit o uso da função sigmóide logística na modelagem de redes neurais é inadequado. Gomes e Ludermir [13] usaram as funções complemento log-log e probit em redes neurais para aproximação e regressão.

Estes e outros artigos citados mostram que a escolha da função de ativação é considerada, por muitos especialistas, tão importante quanto a arquitetura e o algoritmo de aprendizagem da rede neural. A principal característica da função logit é que esta função assume um intervalo contínuo de valores entre 0 e 1. Quando é desejável que a função de ativação se estenda de -1 a $+1$, assumindo uma forma anti-simétrica em relação à origem utiliza-se, geralmente, a função tangente hiperbólica, caso queira manter a característica de uma função sigmóide [15]. Porém, existem situações em que a probabilidade de uma dada resposta se aproxima de 0 a uma taxa diferente da que se aproxima de 1 e essas são as principais características das funções complemento log-log e probit. Por isso, propomos o uso dessas duas funções simples como funções de ativação com o objetivo de melhorar o desempenho da rede neural e diminuir sua complexidade mesmo na presença de observações extremas (*outliers*), pois a escolha da função de ativação pode influenciar fortemente o desempenho e a complexidade da rede neural [8, 9, 10, 36].

Como não existe um padrão estabelecido para a comparação de desempenho de métodos de treinamento de redes

neurais do tipo MLP, a complexidade da rede neural foi avaliada através do número de épocas, por ser visto como um critério bem definido, apesar de insuficiente, uma vez que não reflete as diferenças na complexidade por época de treinamento de cada método [12, 30].

A importância de avaliar o comportamento dos modelos na presença de observações extremas se deve ao fato de que observações extremas podem influenciar mais o desempenho de um modelo do que de outro. Tais influências de observações extremas e da composição da amostra sobre o ajuste de modelos são sempre abordadas nos textos tradicionais que tratam de regressão linear [7, 24, 29] ou até mesmo textos que tratam de redes neurais [19].

2. Funções complemento log-log e probit

As RNAs têm sido aplicadas com sucesso em uma diversidade de problemas, como regressão, classificação, agrupamento, otimização, previsão de séries temporais, etc. As RNAs são capazes de modelar adequadamente uma variedade de problemas, devido ao seu poder computacional e à sua capacidade de representar mapeamentos não-lineares [4].

Nelder e Wedderburn [23] desenvolveram uma classe de modelos lineares generalizados baseados na família exponencial uniparamétrica, cujas médias são não-lineares num conjunto de parâmetros lineares. A função de ligação deve estar de acordo com a distribuição proposta para os dados e a escolha deve ser feita com a finalidade de facilitar a interpretação do modelo. Essa função relaciona o preditor linear, η, ao valor esperado. Para a distribuição binomial uma função de ligação deve satisfazer a condição de transformar o intervalo (0, 1) na linha dos reais. As três funções de ligação logit (η_1), complemento log-log (η_2) e probit (η_3) são, respectivamente,

$$\eta_1 = \ln(\tau/(1-\tau)), \tag{1}$$

$$\eta_2 = \ln[-\ln(1-\tau)]\mathrm{e} \tag{2}$$

$$\eta_3 = \Phi^{-1}(\tau), \tag{3}$$

em que τ é a probabilidade de ocorrência de determinado sucesso e $\Phi(\cdot)$ é a distribuição acumulada da normal padrão.

As três funções, logit, probit e complemento log-log, apresentam um comportamento quase linear no intervalo $0{,}1 \leq \tau \leq 0{,}9$. A Figura 1 apresenta o comportamento de η como função de τ para as três funções até aqui citadas. Para pequenos valores de τ próximos de 0, as ligações logit e complemento log-log encontram-se bastante próximas, decaindo mais rapidamente que a probit. Entretanto, quando τ se aproxima de 1, a ligação complemento log-log cresce mais lentamente do que as ligações probit e logit [22]. Bliss [3], utilizando um modelo binomial com ligação probit, foi quem iniciou a modelagem de proporções. Como o parâmetro de interesse, no caso da binomial, sempre é uma probabilidade, fica muito razoável que funções de distribuições acumuladas sejam utilizadas para gerarem novas ligações e consequentemente novos modelos. A função de ligação complemento log-log é recomendada por Collett [5] quando a distribuição das proporções é bastante assimétrica.

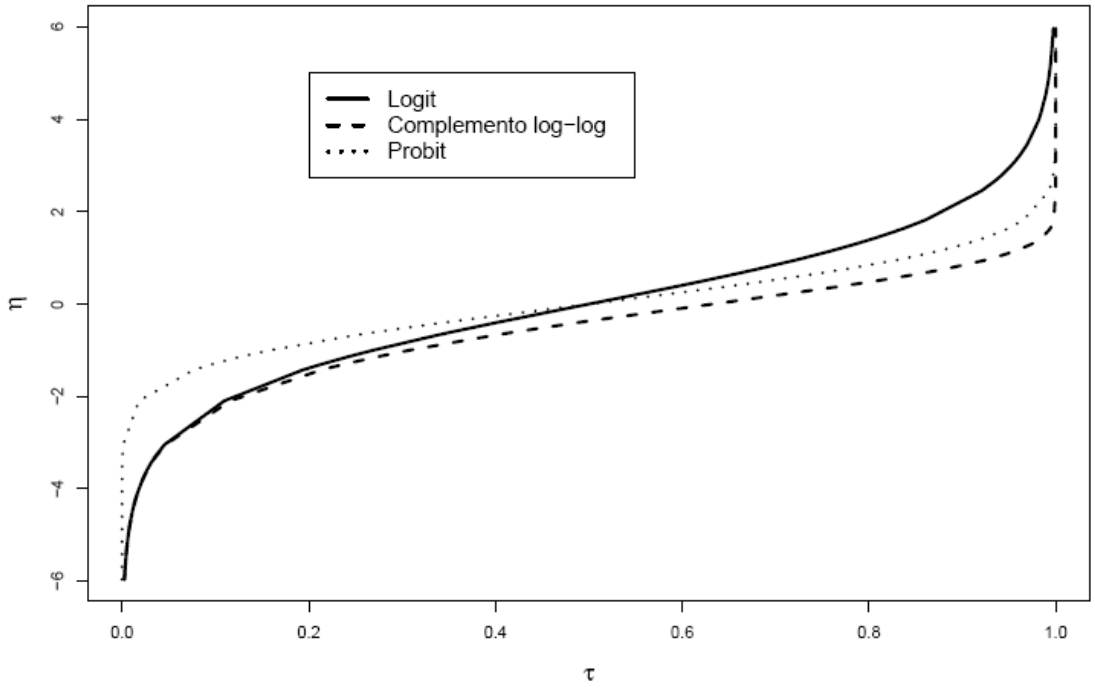

Figura 1: Gráfico de funções de ligação.

A ideia é utilizar uma rede *Multilayer Perceptron* (MLP) usando as novas funções propostas, representando as funções inversas de (2) e (3), que assumem as formas

$$f(x) = 1 - \exp(-\exp(x)), \tag{4}$$

$$f(x) = \Phi(x) = 1/\sqrt{2\pi}\int_{-\infty}^{x} e^{-x^2/2}dx, \tag{5}$$

onde as equações (4) e (5) representam as funções complemento log-log e probit, respectivamente.

O Teorema da Aproximação Universal (TAU) fornece a justificativa matemática para a aproximação de uma função contínua arbitrária em oposição à representação exata [15]. A seguir iremos mostrar que essas novas funções satisfazem o TAU cujo enunciado pode ser visto no Apêndice A.1.

As proposições a seguir estabelecem que as novas funções sejam não-constantes, limitadas e monotonicamente crescentes.

Proposição 1 *As funções complemento log-log e probit são funções monotonicamente crescentes (MC).*

Prova. Seja $x_2 > x_1$.

Então, $\exp(x_2) > \exp(x_1)$, pois $\forall\ x\ \exp(x) > 0$, logo $1-\exp(-\exp(x_2)) > 1-\exp(-\exp(x_1))$. Portanto, a função complemento log-log é MC.

Para a função probit, temos que a função $\Phi(x)$ é a distribuição acumulada da normal padrão, como foi dito anteriormente, portanto, a função $\Phi(x)$ é uma probabilidade, logo, $\Phi(x) > 0$. Como $x_2 > x_1$, logo $\Phi(x_1) > \Phi(x_2)$. Portanto, a função probit é MC.

■

Proposição 2 *As funções complemento log-log e probit são limitadas em 1 quando* $x \to +\infty$ *e em 0 quando* $x \to -\infty$.

Prova.

Para o limite superior da função complemento log-log, temos pelas propriedades de limite

$$\lim_{x\to+\infty} f(x) = \lim_{x\to+\infty} (1-\exp(-\exp(x))) = 1-0 = 1;$$

para o limite inferior, temos

$$\lim_{x\to-\infty} f(x) = \lim_{x\to-\infty} (1-\exp(-\exp(x))) = 1-1 = 0.$$

Para a função probit, a prova é direta, pois $\Phi(x)$ é uma probabilidade, portanto limitada no intervalo [0, 1].

■

Proposição 3 *As funções complemento log-log e probit são funções contínuas diferenciáveis.*

Prova.

Diferenciando as funções (4) e (5), temos

$$\frac{df(x)}{dx} = \exp(x)\cdot\exp(-\exp(x)) \qquad \text{e} \tag{6}$$

$$\frac{df(x)}{dx} = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right). \tag{7}$$

■

A partir das Proposições 1-3, temos que as funções complemento log-log e probit são não-constantes, limitadas e monotonicamente crescentes. Assim, essas funções satisfazem as propriedades requeridas no TAU, logo, podem ser usadas como função de ativação da rede neural.

É sabido que quando as funções de ativação são mudadas, deve-se trocar também as derivadas das funções de ativação usadas no cálculo dos gradientes locais presentes nas regras de aprendizagem (delta generalizada) dos pesos de saída e da camada escondida. Devido ao fato de que a regra de aprendizagem tem um impacto direto na convergência do algoritmo, as devidas mudanças foram feitas ao utilizarmos as novas funções de ativação. Este procedimento com as respectivas modificações pode ser visto no Apêndice A.2.

3. Resultados experimentais

Nesta seção, serão apresentados os resultados experimentais obtidos com a implementação das novas funções de ativação complemento log-log e probit, bem como os resultados com a função sigmóide logística para efeito de comparação com as demais. Primeiramente, foram realizados experimentos com dados sintéticos, o detalhamento destes experimentos está apresentado na Seção 3.1. Posteriormente, foram realizados experimentos com dados reais, o detalhamento destes experimentos está apresentado na Seção 3.2. Todos os experimentos foram realizados na plataforma R com o uso do pacote 'AMORE' (*A MORE flexible neural network package*) [26].

3.1. Aplicação em dados sintéticos

Executamos, neste trabalho, algumas experiências de simulação de Monte Carlo com a finalidade de avaliar a implementação das novas funções de ativação. O método Monte Carlo pode ser usado para fazer inferências sobre os parâmetros do modelo [27]. O número de réplicas de Monte Carlo foi fixado em 1000. Foram conduzidas 20 inicializações diferentes dos parâmetros ajustáveis (pesos e *bias*). Em cada réplica foram realizadas as 20 inicializações, previamente fixadas, de pesos e *bias* e a cada inicialização dos parâmetros foi obtido o erro quadrático médio (EQM) de cada modelo de rede neural, com isso obtivemos a média e o desvio-padrão de 20000 experimentos (resultado das 1000 réplicas e das 20 inicializações). Os pesos e *bias* foram inicializados aleatoriamente de uma Uniforme U(-1, 1). O número máximo de épocas foi 5000. O número de observações extremas foi fixado em $k = \{0, 5, 20\}$. A ideia é mostrar que as funções propostas são capazes de aproximar qualquer função mesmo na presença de observações extremas.

Os modelos de rede neural com funções de ativação sigmóide logística, complemento log-log e probit na camada escondida e saída linear são simbolizados por *LoLi*, *ClLi* e *PrLi*, respectivamente. Os modelos de rede neural com funções de ativação na camada de saída igual à função de ativação usada na camada escondida a saber sigmóide logística, complemento log-log e probit são simbolizados por *LoLo*, *ClCl* e *PrPr*, respectivamente.

Para avaliar as novas funções de ativação, usamos duas funções na tarefa de aproximação. As funções usadas foram

$$g_1(x) = \sin(x); \quad x \in (0,2\pi) \qquad \text{e} \tag{8}$$

$$g_2(\mathbf{x}) = 2\sin(x_1^2 - x_2) + x_1 x_2 - 1; \quad \mathbf{x} \equiv (x_1, x_2), \quad x_1, x_2 \in [0,3]. \tag{9}$$

Os padrões (vetores) de entrada que são apresentados aos nodos de entrada rede neural foram gerados aleatoriamente a partir de uma distribuição uniforme de acordo com o domínio de cada função apresentada. Foram geradas $n–k$ observações pseudo-aleatórias de cada função (8) e (9). Para as k observações extremas foram geradas k observações pseudo-aleatórias de cada função citada e adicionados valores gerados de uma distribuição Gaussiana $N(0,1)$. Com isso, os n valores foram considerados como as observações da variável resposta de cada modelo. Os experimentos foram realizados com uma rede MLP com uma camada escondida usando *backpropagation* (BP) e o *backpropagation* com o termo *momentum* (BPM). Usamos 250 exemplos, 200 (80% do total) para o treinamento e os 50 restantes para teste.

Como a função sigmóide logística é a mais usada na prática e o modelo mais simples é com a saída linear, inicialmente, foram verificadas, para cada função, nove configurações para os modelos com algoritmo de aprendizagem BP e vinte e sete para os modelos com algoritmo de aprendizagem BPM, variando o número de nodos escondidos, $q = \{4, 6, 8\}$, a taxa de aprendizagem, $\nu = \{0{,}001; 0{,}01; 0{,}1\}$ e o termo de *momentum*, $m = \{0{,}5; 0{,}7; 0{,}9\}$ usando o modelo *LoLi* que serviu de referência para a escolha da arquitetura da rede. A configuração desta arquitetura selecionada através do menor erro de validação, em cada tarefa de aproximação, está apresentada na Tabela 1.

Tabela 1: Detalhes da arquitetura das redes neurais selecionadas pelo menor erro de validação do modelo *LoLi*.

Função	Arquitetura	ν	m	No. de parâmetros ajustáveis
$g_1(x)$	1–8–1	0,1	0,9	25
$g_2(x)$	2–6–1	0,01	0,7	25

A Tabela 2 apresenta as estatísticas descritivas das observações geradas com variável resposta do modelo. Podemos observar que a variabilidade (ver desvio-padrão) dos dados aumenta de acordo com o aumento de observações extremas tanto para a função g_1 como para a função g_2. Nesta tabela está apresentado o valor da taxa com que os dados se aproximam de 1, mostrando que a probabilidade da variável resposta de cada exemplo aqui apresentado se aproxima de 0 a uma taxa diferente da que se aproxima de 1 (ver Seção 1).

O método de análise de variância (ANOVA) foi usado para para comparar o desempenho dos modelos usando o algoritmo BP com o desempenho dos modelos usando o algoritmo BPM. A ANOVA é uma técnica que pode ser usada para determinar se as médias de duas ou mais populações são iguais. O teste se baseia numa amostra extraída de cada população. Na ANOVA assume-se que as populações são normais e têm variâncias iguais, ou seja, o teste é realizado baseado na hipótese nula H_0: *as médias das populações são todas iguais*, se a hipótese nula é verdadeira, então a variância dentro de grupos é igual à variância entre grupos e ambas são uma estimativa da variância comum a todas as populações. Em outras palavras, concluiremos que as diferenças observadas entre as médias amostrais são devidas a variações aleatórias na amostra. A magnitude numérica destas variâncias é comparada formalmente através do teste de Fisher (teste F). O teste F é geralmente utilizado para comparar variâncias e decidir se são ou não significativamente diferentes, também pode ser usado para comparar dois modelos diferentes [20]. A decisão de aceitar ou rejeitar H_0 depende da escolha do nível de significância, α. Na maioria das aplicações práticas o valor escolhido é 0,05 ou 0,01 mas não há nada que justifique formalmente o uso destes valores em particular. Um enfoque alternativo consiste em calcular o menor nível de significância para o qual H_0 é rejeitada, para o valor observado da estatística de teste [18]. Esta quantidade é chamada **nível crítico**, **probabilidade de significância** ou ***p*-valor**. A ideia é que se o p-valor for maior que o valor de α pré-estabelecido ele fornece evidência de que H_0 é verdadeira, enquanto que se o p-valor for menor que o valor de α pré-estabelecido indica que existe evidência nos dados contra H_0 [18].

Tabela 2: Estatísticas descritivas dos valores gerados para a tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$.

Número de observações extremas	$g_1(x)$					$g_2(x)$				
	Média	DP	Mín	Máx	Taxa	Média	DP	Mín	Máx	Taxa
$k = 0$	– 0,098	0,697	– 1,000	0,985	56,4	0,679	0,280	0,020	1,000	43,4
$k = 5$	– 0,025	0,750	– 1,776	1,341	63,2	0,728	0,765	– 2,606	4,943	44,3
$k = 20$	0,049	0,802	– 2,356	2,233	66,2	0,902	1,246	– 1,494	7,819	46,2

DP - desvio-padrão

As Tabelas 3, 4 e 5 apresentam os resultados dos desempenhos médios dos 20000 experimentos, com seus respectivos desvios-padrão, dos modelos ajustados pelos dois algoritmos descritos anteriormente nos conjuntos de treinamento e de teste dos dados sem observações extremas ($k = 0$), com 5 observações extremas ($k = 5$), e com 20 observações extremas ($k = 20$), respectivamente. Os p-valores apresentados nessas tabelas são resultados dos testes da ANOVA. Neste caso, a hipótese nula é da forma H_0: $\mu_{BP}=\mu_{BPM}$, ou seja, a média dos EQMs para os modelos com algoritmo BP é igual à média dos EQMs para os modelos com algoritmo BPM. Para estes resultados, o nível de significância adotado como referência foi de 0,05 ($\alpha = 0{,}05$), logo, apenas para os p-valores menores do 0,05 é que o teste indicará diferenças estatisticamente significantes.

Através da Tabela 3, podemos observar que ao utilizarmos a função de ativação sigmóide logística, os desempenhos médios, na tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$, são maiores do que os resultados com as funções complemento log-log e probit, seja no conjunto de treinamento seja no conjunto de teste, tanto nos modelos com saída linear como nos modelos com saída não-linear. Para a função $g_1(x)$, observamos, ainda, que usando na camada de saída a mesma função de ativação usada na camada escondida os desempenhos médios foram melhores do que os desempenhos médios com saída linear, independentemente da função de ativação não-linear usada e do algoritmo de aprendizagem. Vale ressaltar que, apenas no modelo *PrLi* encontramos diferença estatisticamente significante, ao compararmos os desempenhos médios dos algoritmos BP e BPM, tanto no conjunto de treinamento (p-valor = 0,0053) como no conjunto de teste (p-valor = 0,0052). Para a função $g_2(x)$, percebemos que os resultados médios do modelo *LoLi* foram menores do que do modelo *LoLo*, tanto no treinamento como no

teste. Para os outros modelos isso só ocorre no conjunto de teste. No modelo *PrLi* foi encontrada uma diferença estatisticamente significante, entre os desempenhos médios dos algoritmos BP e BPM, apenas no conjunto de teste (*p*-valor = 0,0463).

Tabela 3: Média dos EQMs (desvio-padrão) de todos os modelos avaliados na tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$ para $k = 0$.

Função	Modelo	EQM Treino			EQM Teste		
		BP	BPM	*p*-valor	BP	BPM	*p*-valor
$g_1(x)$	*LoLi*	0,2257 (0,0589)	0,1539 (0,1255)	0,1397	0,2322 (0,0588)	0,1584 (0,1214)	0,1205
$g_1(x)$	*ClLi*	0,0897 (0,1145)	0,0294 (0,0808)	0,2150	0,0900 (0,1163)	0,0274 (0,0712)	0,1877
$g_1(x)$	*PrLi*	0,1386 (0,1277)	0,0013 (0,0023)	0,0053	0,1385 (0,1268)	0,0017 (0,0036)	0,0052
$g_1(x)$	*LoLo*	0,1020 (0,0187)	0,0862 (0,0353)	0,2518	0,0918 (0,0328)	0,0638 (0,0233)	0,0535
$g_1(x)$	*ClCl*	0,0348 (0,0496)	0,0545 (0,0478)	0,4032	0,0226 (0,0329)	0,0316 (0,0246)	0,5233
$g_1(x)$	*PrPr*	0,0360 (0,0426)	0,0543 (0,0488)	0,4100	0,0205 (0,0193)	0,0317 (0,0257)	0,3135
$g_2(x)$	*LoLi*	1,9967 (0,5502)	1,8321 (0,6966)	0,5857	2,3643 (0,5464)	2,1794 (0,7759)	0,5671
$g_2(x)$	*ClLi*	1,2147 (0,6706)	0,5942 (0,5804)	0,0520	1,4373 (0,6795)	0,8162 (0,6839)	0,0712
$g_2(x)$	*PrLi*	1,2368 (0,7275)	0,7279 (0,6196)	0,1297	1,5039 (0,8104)	0,9417 (0,7375)	0,1433
$g_2(x)$	*LoLo*	3,9550 (2,3916)	1,8681 (1,8588)	0,0553	4,8715 (2,7856)	2,2723 (2,2957)	0,0463
$g_2(x)$	*ClCl*	0,7928 (0,3611)	0,7865 (0,4829)	0,9753	0,9647 (0,3014)	0,9572 (0,6025)	0,9740
$g_2(x)$	*PrPr*	0,9388 (0,5325)	0,9232 (0,6487)	0,9563	1,1695 (0,6347)	1,1478 (0,8651)	0,9523

EQM - erro quadrático médio; BP - *backpropagation*; BPM - *backpropagation* com *momentum*

Na Tabela 4, em que os dados apresentam 5 observações extremas, notamos que, em todas as situações, os desempenhos médios dos modelos *LoLi* e *LoLo* são maiores do que os desempenhos médios dos outros modelos com as funções de ativação propostas. Notamos, ainda, que os desempenhos médios dos modelos com função de ativação sigmóide logística reduzem significativamente com o uso do algoritmo BPM. Na tarefa de aproximar a função $g_1(x)$, este fato ocorreu com o modelo *LoLi* no conjunto de treinamento (*p*-valor = 0,0268) e no conjunto de teste (*p*-valor = 0,0158) e com o modelo *LoLo* no conjunto de treinamento (*p*-valor = 0,0002). Na tarefa de aproximar a função $g_2(x)$, ocorreu com o modelo *LoLo* no conjunto de treinamento (*p*-valor = 0,0210) e com o modelo *LoLi* no conjunto de teste (*p*-valor = 0,0351). Com as funções propostas, a diferença entre os algoritmos ocorreu apenas em duas situações com o modelo *ClLi* no conjunto de teste (*p*-valor = 0,0379), na tarefa de aproximar a função $g_1(x)$, e com o *ClCl* no conjunto de treinamento (*p*-valor = 0,0117), na tarefa de aproximar a função $g_2(x)$.

Pela Tabela 5, em que os dados apresentam 20 observações extremas, observamos que, em todas as situações, os desempenhos médios dos modelos com as funções de ativação propostas apresentaram melhores desempenhos em relação aos modelos *LoLi* e *LoLo*. Os resultados desta tabela indicam que para dados com um elevado número de observações extremas o uso do algoritmo com o termo *momentum* não apresentou uma melhoria significativa em nenhuma das situações avaliadas (*p*-valores > 0,05).

Para comparar os desempenhos médios entre os modelos utilizamos o teste *t*-Student. O teste t-Student é um teste paramétrico utilizado na estatística para comparar duas amostras independentes com a finalidade de verificar a existência de diferença significativa entre as médias de métricas dessas amostras [18]. A Tabela 6 apresenta os *p*-valores referentes ao teste *t*-Student para comparar os desempenhos médios dos modelos. Por exemplo, ao comparar os modelos *LoLi* e *ClLi* estamos testando a hipótese nula H_0: $\mu_{LoLi} = \mu_{ClLi}$, ou seja, o desempenho médio do modelo *LoLi* é igual ao do modelo *ClLi*.

Conforme visto anteriormente, nos conjuntos de treinamento e de teste, os desempenhos médios dos modelos com funções de ativação sigmóide logística apresentaram os piores resultados ao compararmos com os outros modelos e através do teste podemos observar que, na maioria dos casos, que os desempenhos médios dos modelos *LoLi* ou *LoLo* e dos modelos com as funções propostas são diferentes e estatisticamente significantes, ou seja, rejeitamos a hipótese nula (*p*-valor < 0,05) (Tabela 6). Vale ressaltar ainda que, em algumas situações vistas anteriormente, os desempenhos médios dos modelos com função de ativação sigmóide logística apresentam uma melhora significativa quando utilizamos o algoritmo *backpropagation* com o termo *momentum*, ou seja, essa melhoria ocorre, principalmente, devido à mudança do algoritmo, fato que não ocorre com os modelos com função complemento log-log ou probit, o uso dessas funções melhoram o desempenho independentemente do uso do algoritmo.

Tabela 4: Média dos EQMs (desvio-padrão) de todos os modelos avaliados na tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$ para $k = 5$.

Função	Modelo	EQM Treino			EQM Teste		
		BP	BPM	*p*-valor	BP	BPM	*p*-valor
$g_1(x)$	*LoLi*	0,1572 (0,0239)	0,0635 (0,0300)	0,0268	0,2402 (0,0462)	0,0766 (0,0391)	0,0158
$g_1(x)$	*ClLi*	0,0634 (0,0179)	0,0434 (0,0279)	0,5564	0,0945 (0,0408)	0,0021 (0,0008)	0,0379
$g_1(x)$	*PrLi*	0,0245 (0,0218)	0,0426 (0,0280)	0,6178	0,0309 (0,0278)	0,0434 (0,0285)	0,7571
$g_1(x)$	*LoLo*	0,0703 (0,0031)	0,0456 (0,0042)	0,0002	0,1907 (0,0682)	0,0756 (0,0154)	0,1193
$g_1(x)$	*ClCl*	0,0288 (0,0100)	0,0079 (0,0028)	0,0634	0,0719 (0,0222)	0,0217 (0,0132)	0,0706

$g_1(x)$	*PrPr*	0,0190 (0,0059)	0,0220 (0,0070)	0,7531	0,0484 (0,0199)	0,0346 (0,0177)	0,6131
$g_2(x)$	*LoLi*	3,1895 (0,6791)	1,9464 (0,1699)	0,0948	3,1783 (0,5138)	1,9644 (0,1176)	0,0351
$g_2(x)$	*ClLi*	1,5319 (0,1655)	1,1607 (0,1139)	0,0833	1,5821 (0,1432)	1,3920 (0,1765)	0,4154
$g_2(x)$	*PrLi*	1,0746 (0,1351)	0,8951 (0,1273)	0,3481	1,4433 (0,1462)	1,1020 (0,1348)	0,1055
$g_2(x)$	*LoLo*	2,3911 (0,3666)	1,4215 (0,0954)	0,0210	2,5939 (0,3608)	1,7214 (0,2143)	0,0541
$g_2(x)$	*ClCl*	1,7461 (0,1143)	1,1439 (0,1779)	0,0117	1,9584 (0,1470)	1,5502 (0,2653)	0,1972
$g_2(x)$	*PrPr*	1,0648 (0,1230)	0,7780 (0,0918)	0,0802	1,3818 (0,2359)	1,2001 (0,2767)	0,6240

EQM - erro quadrático médio; BP - *backpropagation*; BPM - *backpropagation* com *momentum*

Tabela 5: Média dos EQMs (desvio-padrão) de todos os modelos avaliados na tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$ para $k = 20$.

Função	**Modelo**	**EQM Treino**			**EQM Teste**		
		BP	**BPM**	***p*-valor**	**BP**	**BPM**	***p*-valor**
$g_1(x)$	*LoLi*	1,2447 (0,1555)	1,4688 (0,1258)	0,2794	1,1896 (0,1476)	1,3798 (0,1671)	0,4062
$g_1(x)$	*ClLi*	0,8321 (0,1321)	0,8863 (0,0817)	0,7320	0,8504 (0,0771)	0,8855 (0,1055)	0,7915
$g_1(x)$	*PrLi*	0,9238 (0,0918)	0,8847 (0,0715)	0,7411	0,8448 (0,0718)	0,8255 (0,0653)	0,8445
$g_1(x)$	*LoLo*	1,2116 (0,1421)	1,1906 (0,1597)	0,9232	1,1812 (0,1467)	1,2937 (0,1696)	0,6229
$g_1(x)$	*ClCl*	0,8497 (0,0683)	0,8776 (0,0763)	0,7894	0,8426 (0,0810)	0,8423 (0,0667)	0,9973
$g_1(x)$	*PrPr*	0,8760 (0,0658)	0,8620 (0,0760)	0,8914	0,8497 (0,0666)	0,8209 (0,0630)	0,7573
$g_2(x)$	*LoLi*	2,8733 (0,1209)	2,8999 (0,1323)	0,8839	3,3200 (0,1420)	3,1984 (0,1486)	0,5626
$g_2(x)$	*ClLi*	2,3132 (0,2139)	2,0106 (0,2626)	0,3849	2,9490 (0,1262)	2,6039 (0,2597)	0,2496
$g_2(x)$	*PrLi*	2,2899 (0,2285)	1,7330 (0,1837)	0,0758	2,9121 (0,1633)	2,6429 (0,2945)	0,4357
$g_2(x)$	*LoLo*	3,4626 (0,2173)	3,2908 (0,3399)	0,6760	4,4507 (0,7827)	4,2429 (0,8703)	0,8613
$g_2(x)$	*ClCl*	2,5398 (0,2929)	1,9998 (0,2033)	0,1494	2,7978 (0,2059)	2,2834 (0,2635)	0,1436
$g_2(x)$	*PrPr*	2,2747 (0,2398)	1,7091 (0,2093)	0,0947	2,6356 (0,2244)	2,1579 (0,3554)	0,2724

EQM - erro quadrático médio; BP - *backpropagation*; BPM - *backpropagation* com *momentum*

Por isso, em alguns modelos com função de ativação sigmóide logística usando o algoritmo de aprendizagem BPM tiveram seus desempenhos médios melhorados com o uso do termo *momentum*, porém, a variabilidade dos resultados continua sendo alta, logo, a diferença dos desempenhos médios dos modelos *LoLi* ou *LoLo* com relação a alguns modelos com as novas funções não é estatisticamente significante (Tabela 6). Na maioria dos casos, entre os desempenhos médios dos modelos com as funções propostas, *ClLi–PrLi* e *ClCl–PrPr*, não observamos diferenças estatisticamente significantes.

Tabela 6: Resultados dos *p*-valores dos testes *t*-Student com nível de 95% de confiança na tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$.

Hipótese nula (H_0)	***k* = 0**				***k* = 5**				***k* = 20**			
para $g_1(x)$	**BP**		**BPM**		**BP**		**BPM**		**BP**		**BPM**	
	Trn	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**
$\mu_{LoLi} = \mu_{ClLi}$	0,003	0,002	0,012	0,006	0,003	0,015	0,316	0,037	0,030	0,029	0,001	0,011
$\mu_{LoLi} = \mu_{PrLi}$	0,041	0,031	0,001	0,001	0,000	0,001	0,309	0,251	0,047	0,026	0,000	0,003
$\mu_{ClLi} = \mu_{PrLi}$	0,798	0,795	0,157	0,148	0,094	0,108	0,492	0,916	0,711	0,479	0,494	0,317
$\mu_{LoLo} = \mu_{ClCl}$	0,001	0,000	0,065	0,006	0,001	0,058	0,000	0,008	0,017	0,030	0,048	0,012
$\mu_{LoLo} = \mu_{PrPr}$	0,000	0,000	0,065	0,007	0,000	0,031	0,005	0,050	0,024	0,028	0,040	0,009
$\mu_{ClCl} = \mu_{PrPr}$	0,521	0,436	0,495	0,503	0,207	0,221	0,958	0,715	0,607	0,526	0,443	0,409
Hipótese nula (H_0)	***k* = 0**				***k* = 5**				***k* = 20**			
para $g_2(x)$	**BP**		**BPM**		**BP**		**BPM**		**BP**		**BPM**	
	Trn	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**
$\mu_{LoLi} = \mu_{ClLi}$	0,008	0,003	0,000	0,001	0,015	0,004	0,001	0,007	0,018	0,034	0,004	0,032
$\mu_{LoLi} = \mu_{PrLi}$	0,012	0,009	0,001	0,002	0,003	0,002	0,000	0,000	0,019	0,038	0,000	0,055
$\mu_{ClLi} = \mu_{PrLi}$	0,526	0,544	0,679	0,644	0,024	0,678	0,069	0,105	0,470	0,430	0,199	0,538
$\mu_{LoLo} = \mu_{ClCl}$	0,001	0,000	0,055	0,058	0,056	0,061	0,094	0,311	0,011	0,029	0,002	0,023
$\mu_{LoLo} = \mu_{PrPr}$	0,001	0,001	0,085	0,094	0,001	0,006	0,000	0,078	0,001	0,020	0,001	0,020
$\mu_{ClCl} = \mu_{PrPr}$	0,747	0,803	0,691	0,702	0,000	0,027	0,043	0,187	0,246	0,300	0,167	0,390

BP – *backpropagation*; BPM – *backpropagation* com *momentum*; Trn – Treino; Tst - Teste
Hipótese alternativa - H_a: $\mu_1 > \mu_2$

Tabela 7: Média do número de épocas (desvio-padrão) de todos os modelos avaliados na tarefa de aproximar as funções $g_1(x)$ e $g_2(x)$.

Função	**Modelo**	***k* = 0**	***k* = 5**	***k* = 20**

		BP		BPM		BP		BPM		BP		BPM	
$g_1(x)$	*LoLi*	2316	(2172)	1934	(2302)	1400	(1393)	3468	(2313)	683	(926)	434	(125)
$g_1(x)$	*ClLi*	1853	(2366)	1793	(2407)	1079	(1508)	3113	(2329)	220	(191)	160	(101)
$g_1(x)$	*PrLi*	1802	(2401)	1903	(2332)	531	(187)	2841	(2302)	169	(124)	270	(242)
$g_1(x)$	*LoLo*	2354	(2150)	2640	(2235)	3557	(2170)	4073	(1746)	721	(949)	1007	(1524)
$g_1(x)$	*ClCl*	1864	(2359)	2020	(2274)	3202	(2204)	3730	(1721)	231	(212)	387	(470)
$g_1(x)$	*PrPr*	1753	(2436)	1886	(2352)	2930	(2187)	1753	(1774)	120	(56)	253	(295)
$g_2(x)$	*LoLi*	450	(63)	263	(180)	788	(261)	1205	(1274)	1950	(2288)	1763	(2428)
$g_2(x)$	*ClLi*	374	(131)	519	(840)	712	(986)	783	(721)	1374	(2059)	1019	(1713)
$g_2(x)$	*PrLi*	279	(168)	278	(210)	659	(612)	625	(294)	1779	(2416)	1234	(2139)
$g_2(x)$	*LoLo*	2131	(1585)	1052	(1542)	4115	(1590)	4621	(1137)	2030	(1691)	1552	(2002)
$g_2(x)$	*ClCl*	1014	(605)	753	(920)	3940	(1591)	3539	(2192)	913	(677)	1253	(1677)
$g_2(x)$	*PrPr*	726	(746)	502	(490)	3967	(2050)	3383	(2425)	1882	(2320)	1002	(1577)

BP - *backpropagation*; BPM - *backpropagation* com *momentum*

A Tabela 7 apresenta os resultados médios do número de épocas necessários para a convergência do algoritmo em cada modelo. Podemos observar que os modelos com função sigmóide logística precisam de mais iterações para convergir do que os modelos com as funções propostas, isso indica que as novas funções desempenham um papel importante na diminuição da complexidade da rede neural.

3.2. Aplicação em dados reais

Aqui, iremos realizar um experimento com dados reais para enfatizar os resultados da simulação na tarefa de aproximar funções. Os dados são de uma pesquisa realizada no Canadá, na década de 60, para calcular os graus de prestígio (variável resposta) de diferentes ocupações [2]. Além desses resultados, foram obtidas informações demográficas e sócio-econômicas, tais como percentual de empregados do sexo feminino, anos de educação, renda média e tipo de ocupação.

A base de dados desta pesquisa é composta de 102 observações, sendo que 4 profissões não obtiveram nenhum tipo de classificação quanto ao tipo de ocupação, ou seja, são profissões que não se encaixaram em nenhuma das categorias mencionadas a seguir, como *professional, managerial, technical*, *white collar* e *blue collar*. Portanto, o nosso conjunto de dados será composto de 98 exemplos, sendo 74 (aproximadamente 75% do total) para o conjunto de treinamento e 24 para o conjunto de teste. A arquitetura da rede ficou da seguinte forma 5-3-1, com 22 parâmetros ajustáveis, a uma taxa de aprendizagem $\nu = 0{,}001$e com termo *momentum* $m = 0{,}5$.

Os diferentes tipos de ocupação são distribuídos da seguinte forma, 31,6% das profissões são do tipo *professional*, 23,5% do tipo *white collar* e 44,9% do tipo *blue collar*. As pessoas que estão em profissões do tipo *professional* são pessoas que possuem maiores níveis de educação comparado com pessoas que estão em profissões do tipo *white collar* e *blue collar*.

Tabela 8: Média dos EQMs (desvio-padrão) dos modelos de regressão avaliados para os dados de graus de prestígio.

Modelo	EQM Treino					EQM Teste				
	BP		BPM		*p*-valor	BP		BPM		*p*-valor
LoLi	85,61	(80,93)	60,61	(51,43)	0,1985	95,03	(75,16)	73,07	(47,24)	0,2221
ClLi	59,03	(62,00)	43,42	(8,31)	0,2182	76,38	(74,58)	58,87	(12,72)	0,2528
PrLi	54,69	(40,52)	43,49	(8,23)	0,1819	70,43	(44,73)	58,98	(12,47)	0,2236
LoLo	181,37	(103,03)	130,09	(99,83)	0,0802	185,66	(102,12)	150,29	(113,22)	0,2519
ClCl	83,03	(83,17)	87,83	(93,76)	0,8488	103,45	(97,71)	110,58	(107,10)	0,8067
PrPr	83,31	(85,05)	69,45	(75,21)	0,5444	103,71	(9,55)	89,40	(86,13)	0,5893

EQM - erro quadrático médio; BP - *backpropagation*; BPM - *backpropagation* com *momentum*

Na Tabela 8, apresentamos os resultados da aplicação em dados reais. Podemos observar que, em todos os casos, os desempenhos médios dos modelos com as novas funções propostas são melhores do que os desempenhos médios dos modelos com função de ativação sigmóide logística, independentemente do algoritmo de aprendizagem usado (p-valor > 0,05). Notamos ainda que a variabilidade dos desempenhos nos modelos com função sigmóide logística é maior do que nos modelos com as novas funções.

Tabela 9: Resultados dos *p*-valores dos testes *t*-Student com nível de 90% de confiança para os dados do grau de prestígio.

Hipótese nula (H_0)	BP		BPM	
	Treino	Teste	Treino	Teste
$\mu_{LoLi} = \mu_{ClLi}$	0,0992	0,1915	0,0527	0,0767

$\mu_{LoLi} = \mu_{PrLi}$	0,0470	0,0830	0,0534	0,0779
$\mu_{ClLi} = \mu_{PrLi}$	0,3855	0,3667	0,5123	0,5119
$\mu_{LoLo} = \mu_{ClCl}$	0,0003	0,0027	0,0647	0,1044
$\mu_{LoLo} = \mu_{PrPr}$	0,0003	0,0030	0,0095	0,0187
$\mu_{ClCl} = \mu_{PrPr}$	0,5047	0,5037	0,2241	0,2224

BP - *backpropagation*; BPM - *backpropagation* com *momentum*
Hipótese alternativa – H_a: $\mu_1 > \mu_2$

Tabela 10: Média do número de épocas (desvio-padrão) de todos os modelos avaliados para os dados do grau de prestígio.

Modelo	BP		BPM	
LoLi	2881	(2459)	655	(753)
ClLi	1280	(1238)	232	(163)
PrLi	977	(756)	391	(804)
LoLo	4023	(4674)	1436	(965)
ClCl	1242	(979)	217	(103)
PrPr	1008	(637)	219	(102)

BP - *backpropagation*;
BPM - *backpropagation* com *momentum*

A Tabela 9 mostra que, no conjunto de treinamento, os desempenhos médios dos modelos *LoLi* ou *LoLo* e dos modelos com as funções propostas são diferentes e estatisticamente significantes, ou seja, rejeitamos a hipótese nula (p-valor < 0,10) (Tabela 8). No conjunto de teste, ao compararmos os desempenhos médios dos modelos com saída linear, não existe diferença estatisticamente significante apenas entre os desempenhos médios dos modelos *LoLi* e *ClLi* usando *backpropagation*; ao compararmos os desempenhos médios dos modelos *LoLo* e *ClCl* podemos observar que existe uma difereça *borderline*, ou seja, o p-valor é muito próximo do nível de significância de 10%. Entre os desempenhos médios dos modelos com as funções propostas, *ClLi-PrLi* e *ClCl-PrPr*, não observamos diferenças estatisticamente significantes.

A Tabela 10 apresenta os resultados médios e seus respectivos desvios-padrão do números de épocas necessários para a convergência do algoritmo. Podemos observar que nos modelos com função sigmóide logística o número médio de épocas necessárias foi maior do que nos modelos com função complemento log-log ou com função probit.

4. Conclusão

Este artigo apresenta estudos realizados através de aplicações em dados sintéticos e em dados reais para verificação dos impactos de novas funções implementadas, complemento log-log e probit, como função de ativação nos modelos de rede neural MLP com uma camada escondida. Para os dados sintéticos, essa avaliação se estende para dados com observações extremas e as principais medidas utilizadas para avaliar esse impacto foram a média e o desvio-padrão do EQM das 1000 réplicas de Monte Carlo e 20 inicializações dos pesos e *bias*, que totaliza 20000 experimentos. Para os dados reais, foram utilizados a média e o desvio-padrão do EQM das 20 inicializações. Além disso, avaliamos o nível de complexidade dos modelos através do número médio de épocas necessárias para a convergência do algoritmo. Para comparar os resultados usamos a função de ativação sigmóide logística, pois é a mais utilizada na prática. Com os resultados obtidos concluímos que as novas funções têm a capacidade de melhorar o desempenho da rede neural MLP e diminuir sua complexidade mesmo na presença de observações extremas.

Agradecimentos. Os autores agradecem ao CNPq e à FACEPE pelo auxílio financeiro e aos revisores pelas devidas sugestões.

Referências Bibliográficas

[1] Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3), 930–945, 1993.

[2] Blishen, B., Carroll, W. and Moore, C. *Census of Canada*. Statistics Canada. Technical report, 3 (6), Departments of Sociology, York University and University of Victoria, Germany, 1971.

[3] Bliss, C. I. The caculator of the dosage-mortality curve. *Ann. Appl. Biol*., 22, 134–167, 1935.

[4] Braga, A. P., Ludermir T. B. and Carvalho, A. C. P. *Redes Neurais Artificiais - Teoria e Aplicações*. ed.. LTC: Rio de Janeiro, 2007.

[5] Collet, D. *Modelling Survival Data in Medical Research*. Chapman and Hall: London, 1994.

[6] Cybenko, G. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, 2, 303–314, 1989.

[7] Draper, N. and Smith, H. *Applied regression analysis*. John Wiley: New York, 1981.

[8] Duch, W. and Jankowski, N. Survey of neural transfer functions. *Neural Computing*, 2, 163–212, 1999.

[9] Duch, W. and Jankowski, N. Taxonomy of neural transfer functions. *International Join Conference on Neural Networks*, Como, Italy and Los Alamitos, California, 2000. Computer Society and IEEE.

[10] Duch, W. and Jankowski, N. Transfer functions: hidden possibilities for better neural networks. *9th European Symposium on Artificial Neural Networks*, pages 81–94, Bruges, Belgium, 2001.

[11] Fausett, L. *Fundamentals of neural networks*. Prentice Hall: New York, 1994.

[12] Fiesler, E. and Moreira, M. Comparing Parameterless Learning Rate Adaptation Methods. *International Conference on Neural Networks* (ICNN'97), pages 1082–1087, Barcelona, Spain, 1997. IEEE.

[13] Gomes, G. S. S. and Ludermir, T. B. Aproximação de funções através de redes neurais artificiais com funções de ativação complemento log-log e probit. SBRN/WCI, Salvador, Bahia, 2008. Computer Society and IEEE.

[14] Gomes, G. S. S. and Ludermir, T. B. Complementary log-log and probit: activation functions implemented in artificial neural networks. *Eighth International Conference on Hybrid Intelligent Systems*, pages 939–942 , Barcelona, Spain, 2008. Computer Society and IEEE.

[15] Haykin, S. *Neural Networks: A Comprehensive Foundation*, 2nd ed.. Prentice Hall: New Jersey, 2001.

[16] Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251–257, 1991.

[17] Hornik, K. Some new results on neural network approximation. *Neural Networks*, 6(9), 1069–1072, 1993.

[18] Lehman, E. *Testing Statistical Hypothesis*. John Wiley & Sons: New York, 2nd edition, 1986.

[19] Limo, K. Robust Error Measure for Supervised Neural Network Learning with Outliers. *IEEE Transactions on Neural Networks*, 7 (1), 246–250, 1996.

[20] Lindman, H. R. *Analysis of variance in complex experimental designs*. W. H. Freeman & Co: San Francisco, 1974.

[21] Ma, L and Khorasani, K. Constructive Feedforward Neural Networks Using Hermite Polynomial Activation Functions. *IEEE Transactions on Neural Networks*, 16(4), 821–833, 2005.

[22] McCullagh, P. and Nelder, J. A. *Generalized Linear Models*, 2nd ed.. Chapman and Hall: London, 2nd. Edition edition, 1989.

[23] Nelder, J. A. and Wedderburn, W. M. Generalized linear models. *Journal of The Royal Statistical Society*, 3, 370–384, 1972.

[24] Neter, J., Wasserman, W. and Kutner, M. *Applied linear statistical models*. Homewood: Richard D.Irwin, 1990.

[25] PubMed. http://www.pubmed.gov. 2008.

[26] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2007.

[27] Robert, C. P. and Casella, G. *Monte Carlo Statistical Methods* (Springer Texts in Statistics), 2nd ed.. Springer-Verlag: New York, 2005.

[28] Rumelhart, D. E. and McClelland, J. L. *Parallel distributed processing: explorations in the microstructure of cognition*. MIT Press, Cambridge, MA, USA, 1986.

[29] Seber, G. *Linear regression analysis*. John Wiley: New York, 1977.

[30] Silva, L. N. C. *Análise e Síntese de Estratégias de Aprendizado para Redes Neurais Artificiais*. Master's thesis, Faculdade de Engenharia Elétrica e de Computação - UNICAMP, 1998.

[31] Singh, Y. and Chandra, P. A class +1 sigmoidal activation functions for FFANNs. *Journal of Economic Dynamics and Control*, 28(1), 183–187, 2003.

[32] Skoundrianos, E. N. and Tzafestas, S. G. Modelling and FDI of Dynamic Discrete Time Systems Using a MLP with a New Sigmoidal Activation Function. *J. Intell. Robotics Syst*., 41(1), 19–36, 2004.

[33] Wen, C. and Ma, X. A max-piecewise-linear neural network for function approximation. *Neurocomputing*, 71, 843–852, 2005.

Apêndice

A.1. Enunciado do Teorema da Aproximação Universal (TAU)

A seguir apresentamos o enunciado do TAU [15]:

Teorema 1 *Suponha que* $\varphi(\cdot)$ *seja uma função contínua não-constante, limitada e monotonicamente crescente. Suponha que* I_n *represente um hipercubo unitário* $[0,1]^n$ *de dimensão* n. *O espaço das funções contínuas em* I_n *é representado por* $C(I_n)$. *Então, dada qualquer função* $f \in C(I_n)$ *e* $\varepsilon > 0$, *existe um inteiro* M *e um conjunto de constantes reais* α_i, θ_i *e* w_{ij}, *onde* $i = 1,\ldots,k$ *e* $j = 1,\ldots,n$ *tal que podemos definir* $F(x_1,\ldots,x_m) = \sum_{i=1}^{k} \alpha_i \varphi(\sum_{j=1}^{n} w_{ij} x_j + \theta_i)$ *como uma realização aproximada da função* $f(\cdot)$, *isto é,* $|F(x_1,\ldots,x_n) - f(x_1,\ldots,x_n)| < \varepsilon$, $\forall x_1,\ldots,x_n$ *que se encontre no espaço de entrada.*

Prova. A prova do teorema pode ser vista em [1], [6], [15].

■

A.2. Desenvolvimento da regra de aprendizagem para as novas funções

O algoritmo *backpropagation* é o mais conhecido e utilizado para treinamento de redes neurais [11]. Ele é baseado na regra delta [28] que define que o valor da correção a ser aplicado aos pesos das conexões é igual à multiplicação de uma taxa de aprendizagem, o gradiente local e o sinal de entrada do nodo. Para cada registro inserido na rede durante o treinamento, a informação é alimentada através da rede para gerar um valor na unidade de saída, chamado de valor ajustado (*d*). Este valor é comparado com o valor real (*y*) e a diferença entre os dois, chamada de erro (*e*), é retropropagada pela rede para ajustar os pesos adaptativos para melhorar o ajuste para padrões similares. Neste método o vetor gradiente é calculado. Portanto, o sinal de erro na saída do nodo *j*, na iteração *l*, é definido por

$$e_j(l) = d_j(l) - y_j(l).$$

O algoritmo foi desenvolvido com o objetivo de obter os pesos que minimizem a função de erros, dada por

$$E(l) = \frac{1}{2}\sum_{j \in C} e_j^2(l),$$

onde o conjunto *C* inclui todos os nodos da camada de saída da rede. Para obter essa minimização é necessário calcular as derivadas de *E*(*l*) com relação aos pesos e *bias*. Logo, seja $\upsilon_j(l)$ dado por

$$\upsilon_j(l) = \sum_{i=0}^{m} w_j i(l) y_i(l)$$

o campo local induzido produzido na entrada da função de ativação associada ao nodo *j*, onde *m* é o número total de entradas (excluindo o *bias*) aplicadas ao nodo *j*. O peso sináptico w_{j0} é igual ao *bias* b_j aplicado ao nodo *j* na iteração *l*. Assim, temos que o sinal funcional $y_j(l)$ que aparece na saída do nodo *j* na iteração *l* é dado por

$$y_j(l) = \varphi_j(\upsilon_j(l)).$$

O algoritmo *backpropagation* aplica uma correção $\Delta w_{ji}(l)$ ao peso sináptico $w_{ji}(l)$, que é proporcional à derivada parcial $\partial E(l)/\partial w_{ji}(l)$. Todo o procedimento restante pode ser visto em [15]. Logo, a correção $\Delta w_{ji}(l)$ pode ser escrita como

$$\Delta w_{ji}(l) = \eta \delta_j(l) y_i(l),$$

onde ν é a taxa de aprendizagem e $\delta_j(l)$ o gradiente local que pode é definido como

$$\delta_j(l) = e_j(l)\varphi_{i'}(\upsilon_j(l)).$$

Sabemos que o fator $\varphi_i(\upsilon_j(l))$ envolvido no cálculo do gradiente local depende unicamente da função de ativação associada ao nodo *j*, logo, o cálculo de δ para cada nodo da MLP com cada uma das novas funções de ativação será feito usando as derivadas encontradas em (6) e (7). A seguir descreveremos as formas destas duas funções:

1. *Função complemento log-log*: para esta função sua forma geral é definida por

$$\varphi_i(\upsilon_j(l)) = 1 - \exp(-\exp(\upsilon_j(l)))$$

onde $\upsilon_j(l)$ é o campo local induzido do nodo j. De acordo com a não-linearidade, a amplitude de saída se encontra dentro do intervalo $0 \leq y_j \leq 1$. A sua derivada é dada por

$$\varphi_i'(\upsilon_j(l)) = \exp(\upsilon_j(l)) \cdot \exp(-\exp(\upsilon_j(l))).$$

2. *Função probit*: para esta função sua forma geral é definida por

$$\varphi_i(\upsilon_j(l)) = \Phi(\upsilon_j(l)) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\upsilon_j(l)} e^{-\upsilon_j^2(l)/2} d\upsilon_j(l).$$

De acordo com a não-linearidade, a amplitude de saída também se encontra dentro do intervalo $0 \leq y_j \leq 1$. A sua derivada é dada por

$$\varphi_i'(\upsilon_j(l)) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\upsilon_j^2(l)}{2}\right).$$