FUNÇÕES DE ATIVAÇÃO HIPERBÓLICAS EM REDES NEURAIS

Ygor de Mello Canalli

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Zimbrão da Silva

Rio de Janeiro
Março de 2017

FUNÇÕES DE ATIVAÇÃO HIPERBÓLICAS EM REDES NEURAIS

Ygor de Mello Canalli

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Zimbrão da Silva, D.Sc.

Prof. Adilson Elias Xavier, D.Sc.

Prof. Leandro Guimarães Marques Alvim, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2017

Canalli, Ygor de Mello

Funções de ativação hiperbólicas em redes neurais/Ygor de Mello Canalli. – Rio de Janeiro: UFRJ/COPPE, 2017.

XIII, 92 p.: il.; 29, 7cm.

Orientador: Geraldo Zimbrão da Silva

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2017.

Referências Bibliográficas: p. 64 – 69.

1. Redes neurais. 2. Suavização hiperbólica. 3. Função de ativação. I. Silva, Geraldo Zimbrão da. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À minha esposa Sâmara, dádiva
de Deus em minha vida.

Agradecimentos

Agradeço primeiramente a Deus, pois acima de tudo e de todos, é Ele quem me concede a fabulosa oportunidade de viver e tudo o que tenho na vida: as pessoas mais preciosas, os melhores momentos e todas as oportunidades que tive para chegar até aqui. Agradeço pois Ele quem me deu forças e colocou em meu caminho todos que foram imprescindíveis para concluir este trabalho.

Agradeço à minha esposa Sâmara Antero, por sempre me incentivar a persiguir meus sonhos, e com paciência suportar toda tensão e ausência adivindas deste trabalho. Pelo consolo e afeto nos momentos de angústia, não esquecerei.

Agradeço aos meus pais, por desde cedo me ensinarem a honestidade e os mais preciosos valores morais, por plantarem em mim o gosto pelo estudo, investindo com alegria seu tempo e recursos. À minha irmã Yasmin por todo carinho dedicado durante a vida, e aos meus demais parentes, por sempre se regozijarem comigo em todas as conquistas.

Agradeço ao meu orientador acadêmico, professor Geraldo Zimbrão, por desde o período em que era estagiário na COPPETEC me dedicar atenção e demonstrar as melhores intenções para comigo, por denunciar meus erros sem deixar de confiar em meu trabalho. Aos professores Adilson Xavier e Leandro Alvim, por comporem a banca examinadora e acompanharem meu trabalho durante todo seu desenvolvimento, contribuindo com valiosos comentários e ideias.

Ao meu amigo Alexsander Andrade, por não apenas ser um dedicado colega na academia desde os primeiros períodos da faculdade, mas especialmente por ser um amigo com quem posso contar para toda a vida. Agradeço também aos amigos que ganhei durante o período da graduação na UFRRJ, pelos preciosos momentos de alegria e estudo juntos. Agradeço ao meu amigo Julio pelas produtivas conversas sobre aprendizado de máquina e redes neurais, bem como pela ajuda na revisão do presente texto. Agradeço aos meus demais colegas do PESC, com que tive a oportunidade de conviver durante estes anos. Ao meu amigo Matheus Klem, por sua tão valiosa amizade desde longa data.

Agradeço a todos os professores que despertaram em mim o interesse pelo conhecimento, me ensinaram com afinco e me fizeram chegar até aqui.

Agradeço à CAPES pelo apoio financeiro que viabilizou este trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

FUNÇÕES DE ATIVAÇÃO HIPERBÓLICAS EM REDES NEURAIS

Ygor de Mello Canalli

Março/2017

Orientador: Geraldo Zimbrão da Silva

Programa: Engenharia de Sistemas e Computação

Neste trabalho apresentamos uma versão escalada das funções de ativação hiperbólica e bi-hiperbólica para redes neurais, alterando o intervalo de atuação para atingir uma melhor capacidade de convergência. Experimentos no *data set* MNIST mostram que a versão escalada da função de ativação hiperbólica alcançou um erro até $97,12\%$ menor que a original, enquanto a versão escalada da função bi-hiperbólica atingiu até $10,24\%$ de melhora relativa quando comparada à original.

Comparando o desempenho das tradicionais funções logística, tangente hiperbólica e ReLU com a hiperbólica escalada, podemos atingir melhoras no erro de $97,44\%$, $17,63\%$, $34,41\%$, enquanto no caso da bi-hiperbólica escalada, melhoras de $97,49\%$, $20,97\%$ e $53,44\%$. Apesar da melhora expressiva, foi necessário uma busca exaustiva para escolha dos parâmetros adequados. Desta forma, utilizamos uma metodologia para ajuste automático dos parâmetros através do algoritmo de *backpropagation*, com o qual atingimos melhoras de $96,44\%$, $12,49\%$ e $12,59\%$ para a versão simétrica da bi-hiperbólica escalada, e de $96,58\%$, $11,36\%$ e $17,06\%$ para a versão assimétrica, dispensando a necessidade de uma busca exaustiva. Também mostramos que o uso da função bi-hiperbólica adaptativa é possui convergência acelerada em circunstâncias onde há limitação de tempo e poder computacional.

Uma das formas mais convencionais de suavizar a função ReLU é através da função Softplus, que sofre todavia do problema de gradiente minguante. Assim, buscando atenuar esta dificuldade, propomos uma alternativa de suavização para a função ReLU utilizando-se da técnica de penalização hiperbólica, a qual denominamos função suavização hiperbólica da ReLU, ou SH-ReLU. Nossos experimentos mostram que a SH-ReLU, unida à referida metodologia de ajuste dos parâmetros, foi capaz de superar a medida de erro da ReLU em $18,62\%$, e $36,67\%$ quando comparada à Softplus.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

HYPERBOLIC ACTIVATION FUNCTIONS ON NEURAL NETWORKS

Ygor de Mello Canalli

March/2017

Advisor: Geraldo Zimbrão da Silva

Department: Systems Engineering and Computer Science

In this work we present a scaled version of hyperbolic and bi-hyperbolic activation functions for neural networks, where the activation interval was modified in order to reach a better convergence. Our experiments on MNIST data set shows that the scaled version of the hyperbolic activation function achieved an error up to 97.12% better than the original, while the scaled version of the bi-hyperbolic function reached up to 10.24% relative improvement when compared to the original.

Comparing the performance of the traditional activation functions logistic, hyperbolic tangent and ReLU to our scaled hyperbolic, we achieved 97.44%, 17.63%, 34.41% of improvement in error, and improvements of 97.49%, 20.97% and 53.44% in the case of scaled bi-hyperbolic. Despite of the significant improvement, an exhaustive search was required to choose the appropriate parameters. Thus, we use a methodology for automatic parameter adjustment through the backpropagation algorithm during training. Using this methodology, we achieved improvements of 96.44%, 12.49% and 12.59% for the symmetric version of scaled bi-hyperbolic function, and 96.58%, 11.36% and 17.06% for the asymmetric one, without need for an exhaustive search. We also showed that the bi-hyperbolic adaptative function have a greate convergence in circumstances where there is a limitation of time and computational power.

One of more usual smoothing of ReLU activation function is the Softplus function, witch suffer from vanishing gradient problem. Thus, in order to reduce this problem, we proposed an alternate smoothing for ReLU function through hyperbolic penalty method, named Hyperbolic smoothing ReLU, or HS-ReLU. Our experiments shows that HS-ReLU, joint to refered parameter adjustment metodology, outperformed ReLU in 18.62% and Softplus in 36.67%.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

As redes neurais multi camadas (MLP - *Multi Layer Perceptron*) são conhecidas por sua capacidade de aproximar qualquer função, o que confere a elas notável utilidade (HORNIK *et al.*, 1989). Por outro lado, apesar de possuir capacidade universal de aproximação demonstrada do ponto de vista teórico, a prática de torná-las aplicáveis para solucionar problemas reais mostrou ser um grande desafio (BLUM e RIVEST, 1992, GLOROT e BENGIO, 2010, LECUN *et al.*, 2012). Existem inúmeros fatores a serem considerados para que uma rede neural seja capaz de aproximar funções do mundo real com qualidade, como por exemplo a escolha de uma arquitetura adequada. A escolha da arquitetura de uma rede neural trata-se de questões como definição da quantidade de camadas, quantidades de neurônios em cada camada, esquema de ligação entre as camadas, dentre outros. Apesar da importância da escolha da arquitetura, destacamos o problema central em redes neurais, que trata-se da escolha dos pesos das sinapses entre os neurônios, de tal forma que a rede possua baixo erro e alta capacidade de generalização. Tradicionalmente tais pesos são escolhidos através de algoritmos de minimização, como o gradiente descendente, de uma medida de erro calculada a partir da saída da rede em relação à função objetivo (HECHT-NIELSEN, 1989). Neste contexto, um fator preponderante para que o algoritmo de minimização obtenha êxito, é a escolha das função de ativação adequada (DUCH e JANKOWSKI, 1999, KARLIK e OLGAC, 2010), assunto sobre o qual tratamos no presente trabalho.

Funções de ativação sigmoidais convencionais, como a logística e a tangente hiperbólica, apesar de serem amplamente utilizadas em redes neurais, sofrem de um problema crônico conhecido como gradiente minguante (BENGIO *et al.*, 1994). O gradiente minguante é um fenômeno que ocorre devido ao fato de as derivadas destas funções tenderem a 0 quando x vai a $\pm\infty$. Nestas funções, a derivada atinge valores extremamente pequenos rapidamente, conforme x distancia-se da origem. Como a atualização do algoritmo de *backpropagation* utiliza-se do gradiente para fazer atualização dos pesos, o gradiente minguante causa uma estagnação no processo de

convergência, conhecido como saturação da rede neural (LECUN *et al.*, 2012).

Assim, idealmente, uma função de ativação cujo processo de gradiente minguante é retardado, possuirá uma maior velocidade de convergência, além de possívelmente ser capaz de atingir um erro mais baixo. Portanto, nossa proposta consiste em utilizar-se da técnica de penalização hiperbólica para solução de problemas gerais de programação não-linear sujeito a restrições de desigualdade para obter novas funções de ativação onde o processo de gradiente minguante seja retardado.

O método de penalização hiperbólica, proposto originalmente por XAVIER (1982), foi utilizado com sucesso em outros trabalhos para criar duas funções de ativação para rede neurais: a função hiperbólica e a função bi-hiperbólica (MIGUEZ, 2012, THOMAZ e MAIA, 2013, XAVIER, 2005). Neste trabalho aprimoramos estas funções de ativação hiperbólicas modificando seu intervalo original de atuação, de forma a obter melhores resultados. Apenas a título de exemplificar a vantagem de nossa função com relação ao problema de gradiente minguante, a Tabela 1.1 compara a ordem de grandeza das derivadas da funções hiperbólica escalada (com $\rho = 1$) e tangente hiperbólica.

Tabela 1.1: Comparação da ordem de grandeza das derivadas da funções hiperbólica escalada (com $\rho = 1$) e tangente hiperbólica.

x	derivada da tangente hiperbólica	derivada da hiperbólica escalada
5	10^{-4}	10^{-3}
10	10^{-9}	10^{-4}
15	10^{-13}	10^{-4}
20	10^{-17}	10^{-4}

Além disso, também propomos um novo uso para o método de penalização hiperbólica em funções de ativação para redes neurais: frequentemente a função retificadora (ou ReLU) é suavizada através da derivada da função logística, criando uma função de ativação conhecida como Softplus (DUGAS e BENGIO, 2001, GLOROT *et al.*, 2011). Da mesma forma que a logística, a função Softplus sofre do problema crônico do gradiente minguante. Assim, utilizamos a técnica de penalização hiperbólica para obter uma versão suavizada da função retificadora cujo processo de gradiente minguante seja retardado.

Apesar da notável capacidade de convergência das funções hiperbólicas, existe um problema intrínseco, que é a escolha dos parâmetros. Para atingir uma con-

vergência acelerada é necessário a escolha dos parâmetros adequados, o que pode ser muito custoso em tempo se feito através de testes exaustivos. Com isso, surge o dilema entre convergência rápida com parâmetros adequados e tempo gasto para escolha dos mesmos. Assim, a fim de contornar este dilema, apresentamos uma técnica de ajuste automático dos parâmetros destas funções de ativação.

A Tabela 1.2 resume os melhores resultados obtidos, isto é, apenas o resultado da melhor configuração (caso haja) e com a quantidade de camadas ocultas onde a melhora do erro (*cross-entropy*) em relação ao *baseline* foi mais expressiva. Os resultados para a comparação entre bi-hiperbólica modificada e bi-hiperbólica original, excepcionalmente, apresentam a melhora relativa média dos resultados, devido ao formato do experimento.

Tabela 1.2: Resumo dos melhores resultados obtidos, melhoras relativas do erro.

Função de ativação	*Baseline*	Melhora relativa
Hiperbólica modificada	Hiperbólica original	$97,12\%$
Bi-hiperbólica modificada	Bi-hiperbólica original	$10,24\%$
Hiperbólica modificada (ajuste manual)	ReLU	$34,41\%$
	Tangente Hiperbólica	$17,63\%$
	Logística	$97,44\%$
Bi-hiperbólica modificada (ajuste manual)	ReLU	$53,44\%$
	Tangente Hiperbólica	$20,97\%$
	Logística	$97,49\%$
Hiperbólica modificada (ajuste automático)	ReLU	$-15,29\%$
	Tangente Hiperbólica	$-17,68\%$
	Logística	$1,06\%$
Bi-hiperbólica simétrica modificada	ReLU	$12,59\%$
	Tangente Hiperbólica	$12,49\%$

Continua na próxima página

Função de ativação	*Baseline*	Melhora relativa
(ajuste automático)	Logística	$96,44\%$
Bi-hiperbólica assimétrica	ReLU	$17,06\%$
modificada	Tangente Hiperbólica	$11,36\%$
(ajuste automático)	Logística	$96,58\%$
Suavização hiperbólica da	ReLU	$18,62\%$
ReLU (ajuste automático)	Softplus	$36,67\%$

Assim, as contribuições deste trabalho são:

- Modificar o intervalo de atuação da função de ativação hiperbólica, de forma a obter resultados significativamente melhores;
- Modificar o intervalo de atuação da função de ativação bi-hiperbólica, obtendo melhores resultados;
- Criar uma alternativa de suavização da função retificadora com melhor capacidade de convergência, utilizando a técnica de penalização hiperbólica;
- Apresentar uma técnica eficiente de ajuste automático dos parâmetros destas funções de ativação.

A organização deste trabalho é como segue: uma breve revisão da literatura, abordando algumas das principais funções de ativação, um estudo sobre o problema do gradiente minguante, uma contextualização sobre *deeplearning*, funções de ativação adaptativas e por fim tratamos da penalização hiperbólica e das funções de ativação hiperbólicas; proposta do trabalho, descrevendo como as funções de ativação hiperbólicas foram modificadas, bem como apresentando a suavização hiperbólica da função retificadora, e por último nossa metodologia para ajuste automático dos parâmetros; avaliação experimental, contendo descrição individual de cada experimento, *data set* utilizado, configuração do experimento, métricas de comparação, detalhes de implementação e hardware e por fim os resultados obtidos, juntamente com uma discussão das principais questões observadas; por último, concluímos com considerações gerais sobre o trabalho, contribuições e diretrizes para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo faremos uma breve revisão bibliográfica, tratando primeiro de algumas das funções de ativação mais tradicionais. Logo após, uma contextualização do estudo de redes neurais na perspectiva conhecida como deeplearning e de um de seus recentes ramos de estudos, as funções de ativação adaptativas. Por fim, trataremos das funções de ativação hiperbólicas, as quais sãos objeto central de nosso estudo.

2.1 Funções de ativação tradicionais

A primeira função de ativação efetivamente utilizada em redes neurais foi a função degrau, proposta na década de 1940 por MCCULLOCH e PITTS (1943). A função degrau $\Theta(x)$ utiliza um limiar de ativação θ para produzir uma saída binária, sendo definida por

$$\Theta(x) = \begin{cases} 1 & x > \theta, \\ 0 & x \leq \theta. \end{cases} \tag{2.1}$$

Devido à descontinuidade inerente à função degrau, não é possível realizar o treinamento através de métodos de minimização de erro baseados em gradiente.

Também é possível utilizar como função de ativação a função identidade

$$I(x) = x. \tag{2.2}$$

Entretanto, utilizar a função identidade como função de ativação de um MLP faz com que este seja capaz de aprender apenas combinações lineares, eliminando a capacidade de aproximação universal, a qual é sua característica mais desejada (HORNIK *et al.*, 1989, MINSKY e PAPERT, 1969).

Posteriormente, a função degrau foi generalizada para a função logística, a qual possui formato sigmoidal, se comportando como uma versão suavizada da função

degrau. A função logística $\sigma(x)$ é dada por

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{2.3}$$

A função logística foi a primeira função sigmoidal a ser utilizada neste contexto, de forma que seu sucesso levou à utilização de diversas outras funções de ativação sigmoidais (DUCH e JANKOWSKI, 1999). As funções de ativação sigmoidais possuem inspiração na neurociência, fazendo parte da argumentação fisiológica do funcionamento das redes neurais (MALMGREN, 2000). Além da referida ligação com a neurociência, tais funções também possuem fundamentação estatística para seu funcionamento (BISHOP, 1995, JORDAN, 1995).

Uma importante função de ativação sigmoidal é a tangente hiperbólica, sendo recomendada como substituta da logística (LECUN *et al.*, 2012). A função tangente hiperbólica pode ser definida a partir do seno e coseno hiperbólicos, como vemos abaixo:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}. \tag{2.4}$$

Outra forma de entender a função tangente hiperbólica é como uma versão escalada da função logística, de forma a atuar no intervalo de ativação $[-1, 1]$, conforme as equações (2.6) a (2.13) mostram.

$$\begin{aligned}
2 \cdot \sigma(2x) - 1 &= 2\left[\frac{1}{1 + e^{-2x}}\right] - 1 && (2.5)\\
&= \frac{2}{1 + e^{-2x}} - 1 && (2.6)\\
&= \frac{2}{1 + e^{-2x}} - \frac{1 + e^{-2x}}{1 + e^{-2x}} && (2.7)\\
&= \frac{2 - (1 + e^{-2x})}{1 + e^{-2x}} && (2.8)\\
&= \frac{1 - e^{-2x}}{1 + e^{-2x}} && (2.9)\\
&= \frac{e^{x}(1 - e^{-2x})}{e^{x}(1 + e^{-2x})} && (2.10)\\
&= \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} && (2.11)\\
&= \frac{\sinh(x)}{\cosh(x)} && (2.12)\\
&= \tanh(x) && (2.13)
\end{aligned}$$

Esta simples flexão da função logística para atuar no intervalo $[-1, 1]$ faz com que a tangente hiperbólica leve a uma melhor convergência da rede neural. Isto se deve ao fato de que com as entradas normalizadas o intervalo coincide. Desta forma, os neurônios cuja função de ativação é simétrica em relação a origem tendem a produzir saídas (que são entradas de uma próxima camada) com médias próximas a zero (LECUN *et al.*, 2012). Também podemos destacar outras funções sigmoidais simétricas em relação à origem, como a função gaussiana de erro e o arco tangente.

Outra notável classe de funções de ativação utilizadas em redes neurais são as funções de base radial (*Radial Basis Functions* - RBFs) (BROOMHEAD e LOWE, 1988), as quais também também foram identificadas por vários anos por diferentes nomes (ex.: abordagem de função potencial) (DUCH e JANKOWSKI, 1999). Funções de base radial são aquelas cujo valor depende apenas da distância de um certo ponto c, chamado centro, de tal foma que, para uma função de base radial $r(x)$, temos:

$$r(x) = r(\| x - c \|). \tag{2.14}$$

Algumas RBFs comumente utilizadas são listadas abaixo.

Gaussiana: $r(x) = e^{(\epsilon x)^2}$

Multiquadrática: $r(x) = \sqrt{1 + (\epsilon x)^2}$

Quadrática inversa: $r(x) = \frac{1}{1+(\epsilon x)^2}$

Multiquadrática inversa: $r(x) = \frac{1}{\sqrt{1+(\epsilon x)^2}}$

2.1.1 O problema do gradiente minguante

Funções sigmoidais, como por exemplo a logística e tangente hiperbólica sofrem de um problema crônico conhecido como gradiente minguante (BENGIO *et al.*, 1994). Tomemos como exemplo a função logística, para a qual temos

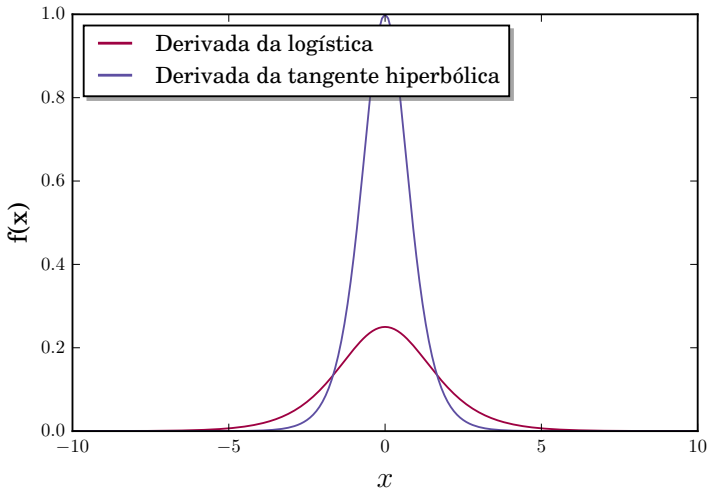
- $\lim_{x \to -\infty} \sigma(x) = 0$
- $\lim_{x \to +\infty} \sigma(x) = 1,$

isto é, a função se aproxima assintoticamente das retas $y = 0$ e $y = 1$. Tal propriedade assintótica se dá devido à função se aproximar cada vez menos das retas. Em outras palavras, a derivada da função tende a zero.

De fato, temos a derivada da função logística

$$\frac{d\sigma}{dx}(x) = \frac{d}{dx}\left(\frac{1}{1 + e^{-x}}\right) = \frac{e^x}{(e^x + 1)^2}, \tag{2.15}$$

Figura 2.1: Derivada das funções de ativação logística e tangente hiperbólica no intervalo $[-10, 10]$.



a qual possui seu limite no infinito

$$\lim_{x \to \pm\infty} \frac{e^x}{(e^x + 1)^2} = 0, \tag{2.16}$$

conforme ilutrado pela Figura 2.1. O mesmo vale para a função tangente hiperbólica. Vejamos,

$$\frac{d \tanh(x)}{dx} = \frac{d}{dx}\left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right) = \frac{4e^{2x}}{(e^{2x} + 1)^2}, \tag{2.17}$$

e da mesma forma

$$\lim_{x \to \pm\infty} \frac{4e^{2x}}{(e^{2x} + 1)^2} = 0, \tag{2.18}$$

Tal fenômeno faz com que a função tenha um valor de derivada insignificante para valores distantes de 0, o que causa estagnação no algoritmo de gradiente descendente, de forma a impedir o processo de convergência. Este efeito de gradiente minguante também é conhecido como saturação da rede neural (LECUN *et al.*, 2012), sendo um fator complicad or no treinamento de redes neurais.

2.2 Deeplearning

Tradicionalmente, os computadores são capazes de resolver muito bem problemas onde há regras claras e bem estabelecidos, problemas estes que muitas vezes se mostram complicados para a mente humana. O desenvolvimento da área conhecida como inteligência artificial fez com que o computador sobrepujasse a capacidade da mente humana neste tipo de problemas há décadas atrás GOODFELLOW *et al.* (2016). Um clássico exemplo disso foi a partida de Xadrez ocorrida em 1997 entre o campeão mundial Gary Kasparov e o computador IBM Deep Blue, onde a máquina

se mostrou capaz de superar a mente humana HSU (2002).

Por outro lado, as tarefas que nos parecem mais simples e intuitivas, como o reconhecimento da fala, de faces e objetos, são extremamente complicadas para o computador. Isso porque o computador lida bem com problemas onde as regras são matematicamente definidas, ao passo quando as regras são subjetivas e difíceis de serem elucidadas de maneira formal, torna-se extremamente complicado para o computador desempenhar tal tarefa GOODFELLOW *et al.* (2016).

Assim, a motivação para o campo de pesquisa conhecido como *deeplearning* é conceder tais capacidades cognitivas abstratas ao computador. O objetivo deixa de ser elucidar regras formais para o comportamento do computador, mas sim fazer com que o computador seja capaz de absorver conhecimento através da experiência, associando de maneira hierárquica a compreensão de conceitos mais simples, como forma de construir conceitos mais complexos. A motivação para esta abordagem é que nossa mente atua através de várias camadas de abstração. Se as camadas de abstração necessárias para compreensão destes problemas subjetivos fossem desenhadas em um diagrama, trataria-se de um diagrama profundo em termos de quantidade de camadas, daí o nome *deeplearning* (aprendizado profundo).

O *deeplearning* foi viabilizado através do desenvolvimento da pesquisa na área de redes neurais, tendo como base a ideia de se utilizar redes neurais com muitas camadas, as quais de alguma forma seriam capazes de simular a estrutura hierárquica de abstração da mente humana. Um importante artigo publicado por HINTON (2007) chamou atenção para o até então pouco explorado poder das redes neurais profundas, sendo, de certa forma, uma demarcação da área de estudos denominada deeplearning. Rapidamente redes neurais profundas se destacaram em tarefas até então difíceis de serem realizadas por máquinas, devido a sua subjetividade inerente, como por exemplo reconhecimento de imagens e de fala (BENGIO e YOSHUA, 2009, HINTON *et al.*, 2012).

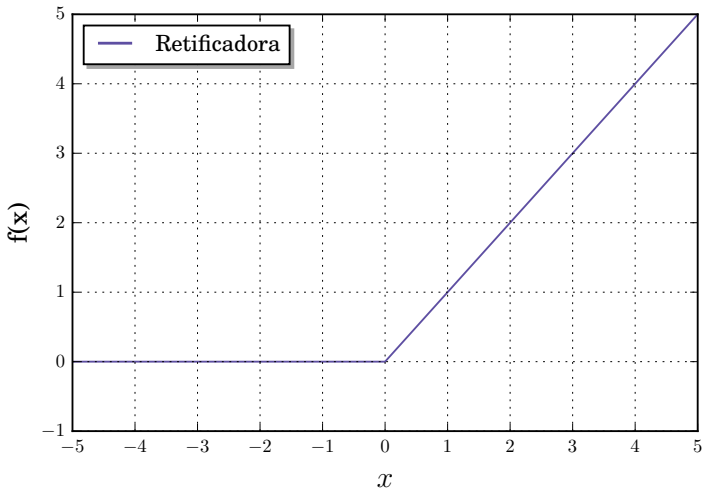
Dentre as técnicas incluídas na área chamada *deeplearning* podemos destacar: Deep MLP, Deep Boltzmann Machine, Deep Belief Networks, Autoencoders, Recursive Neural Networks, Convolutional Neural Netorks, Long Short Therm Memory, Max pooling, ReLU, Softmax, Weight decay, Sparsity e Dropout.

Dentro do contexto de redes neurais profundas, uma função de ativação de grande destaque é a retificadora, dada por

$$r(x) = \max(0, x). \tag{2.19}$$

A função refiticadora pode ser vista na Figura 2.2. Frequentemente chamamos o uso da função retificadora em um neurônio de Unidade Linear Retificada (Rectified Linear Unit - ReLU) (NAIR e HINTON, 2010).

Figura 2.2: Função de ativação retificadora



A função retificadora, em geral, possui larga vantagem em relação às tradicionais funções sigmoidais, devido à sua capacidade de não saturar. Assim, o uso da ReLU e suas variações tornou-se um importante avanço dentro de *deeplearning*.

2.3 Funções de ativação adaptativas

No contexto de redes neurais costuma-se submeter a aprendizado apenas os pesos da rede, de forma que uma arquitetura é escolhida e uma única função de ativação é pré-definida para atuar no processo de aprendizado dos pesos. Contudo, definir uma arquitetura e função de ativação capaz de realizar rapidamente o aprendizado dos parâmetros, de forma que a rede apresente grande capacidade de generalização, quase sempre é um processo dispendioso. Com isso, muitas pesquisas tem se concentrado neste assunto (AGOSTINELLI *et al.*, 2015).

Um caminho possível neste contexto de escolher a função de ativação mais adequada é fazê-lo através de um processo de aprendizado. Esforços anteriores se concentraram sobretudo em realizar escolha da função a partir de um repositório, através de custosas estratégias evolucionárias (YAO, 1999).

Pesquisas recentes tem apresentado um promissor caminho alternativo. Trata-se de utilizar o próprio algoritmo *backpropagation*, até então utilizado exclusivamene no treinamento dos pesos da rede neural, para conjuntamente aprender parâmetros de funções adaptativas. Grosseiramente, basta tomar uma função de ativação parametrizada e colocar seus parâmetros como um vetor *bias* extra, de maneira que cada neurônio, individualmente, terá sua própria configuração de função de ativação. Assim, a configuração da função de ativação será corrigida pelo *backpropagation*, de forma que seja possível atingir um bom desempenho da rede neural em um tempo

de treimento reduzido.

Tal estratégia, pode-se dizer, foi inaugurada com êxito por AGOSTINELLI *et al.* (2015), o qual obteve na ocasião estado da arte nos data sets CIFAR-10, CIFAR-100 (KRIZHEVSKY, 2009) e Higgs boson decay (BALDI *et al.*, 2015). Com o sucesso da abordagem, diversos trabalhos se sucederam empregando a técnica de aprendizado de parâmetros em novas funções de ativação adaptativas, dentre os quais destacamos HE *et al.* (2015), JIN *et al.* (2015), SCARDAPANE *et al.* (2016), TROTTIER *et al.* (2016).

Diante da adição de novos parâmetros para aprendizado, pode-se questionar a eficiência desta estratégia, por exigir mais computação em cada época para o cálculo das atualizações do *backpropagation*. Apesar do aparente prejuízo, perceba que o impacto dos novos parâmetros é mínimo, visto que redes neurais frequentemente possuem centenas de neurônios em cada camada oculta. Portanto, é necessário para cada neurônio o cálculo de todos os pesos das sinapses com os neurônios da camada seguinte, acrescendo-se apenas um peso extra por parâmetro. Uma análise quantitativa deste fenômeno pode ser vista em JIN *et al.* (2015).

2.4 Penalização hiperbólica

O método de penalização hiperbólica foi proposto originalmente por XAVIER (1982) para solução de problemas gerais de programação não-linear sujeito a restrições de desigualdade:

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0,\ i = 1, \ldots, m \\ \text{t.q.} \quad & f : \mathfrak{R}^n \to \mathfrak{R}, \\ & g_i : \mathfrak{R}^n \to \mathfrak{R},\ i = 1, \ldots, m \end{aligned} \tag{2.20}$$

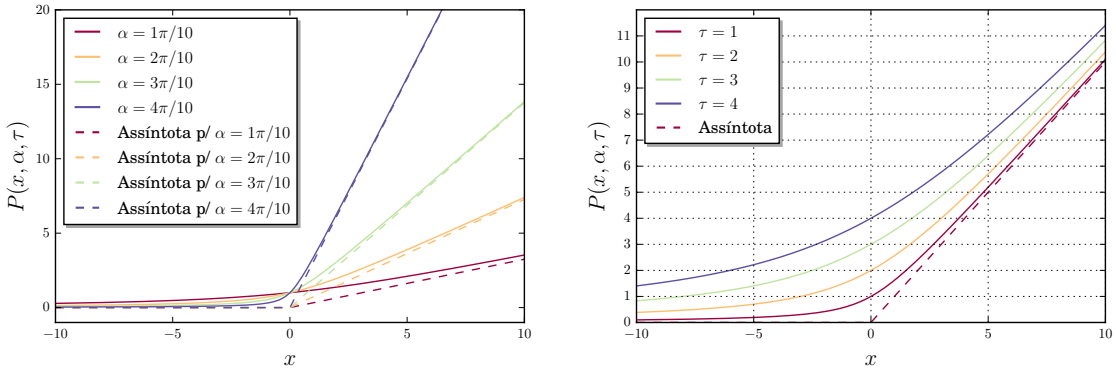
A penalização hiperbólica consiste em aplicar a função penalidade

$$P(x, \alpha, \tau) = \left(\frac{1}{2}\tan\alpha\right) x + \sqrt{\left(\frac{1}{2}\tan\alpha\right)^2 x^2 + \tau^2}, \tag{2.21}$$

onde $\alpha \in [0, \pi/2)$ e $\tau \geq 0$. Note que $P(x, \alpha, \tau)$ forma uma hipérbole cujas assíntotas possuem os ângulos 0 e $(\pi - \alpha)$ em relação ao eixo horizontal, e um intersepto τ com o eixo das ordenadas. Para uma melhor visualização, a Figura 2.3 apresenta a função penalização hiperbólica para diferentes valores de α e τ, respectivamente.

Podemos realizar a substituição $\lambda = \left(\frac{1}{2}\tan\alpha\right)^2$, de forma a colocar a função

Figura 2.3: Impacto dos parâmetros da função penalização hiperbólica



(a) variando α e fixando $\tau = 1$.

(b) variando τ e fixando $\alpha = \pi/4$.

penalização hiperbólica em um formato mais conveniente, isto é,

$$P(x, \lambda, \tau) = \lambda x + \sqrt{\lambda^2 x^2 + \tau^2}, \tag{2.22}$$

onde $\lambda \geq 0$ e $\tau \geq 0$. Com isso, a derivada da função em relação a y assume a forma

$$\frac{\partial P}{\partial x}(x, \lambda, \tau) = \lambda \left[1 + \frac{\lambda x}{\sqrt{\lambda^2 x^2 + \tau^2}}\right], \tag{2.23}$$

a qual é uma função sigmoidal com assíntotas horizontais em 0 e 2λ, isto é,

$$\begin{cases} \lim\limits_{x \to -\infty} \dfrac{\partial P}{\partial x}(x, \lambda, \tau) = 0 \\ \\ \lim\limits_{x \to +\infty} \dfrac{\partial P}{\partial x}(x, \lambda, \tau) = 2\lambda. \end{cases} \tag{2.24}$$

Além disso, temos que

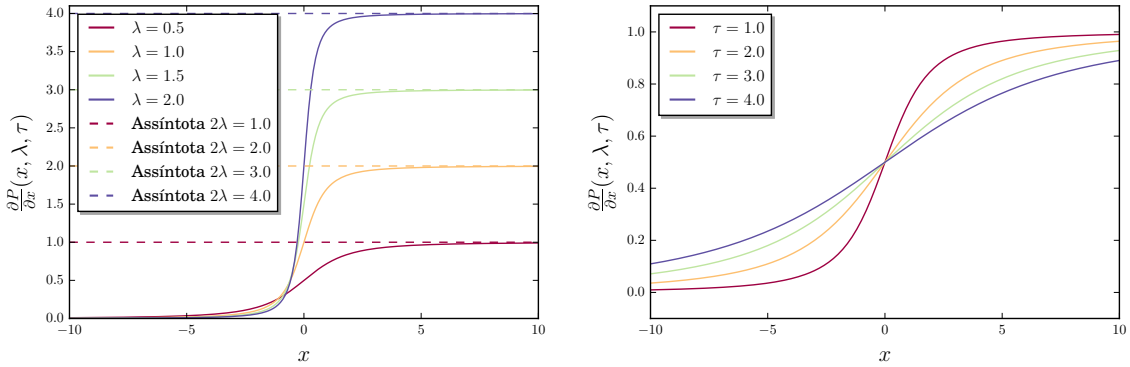
$$\frac{\partial P}{\partial x}(0, \lambda, \tau) = \lambda, \tag{2.25}$$

isto é, a derivada corta o eixo das ordenadas em $y = \lambda$. Note também que o parâmetro τ altera a inclinação da curva, de forma que esta aumenta conforme τ diminui. Desta forma, a Figura 2.4 mostra como a derivada da função penalização hiperbólica se comporta em relação aos parâmetros λ e τ.

A técnica de penalização hiperbólica tem sido aplicada com sucesso na solução de diversos problemas, dentre os quais destacamos clustering (XAVIER, 2010), Fermat-Weber, covering 2D e 3D e hub location (XAVIER e XAVIER, 2014).

Neste trabalho, a função penalização hiperbólica é utilizada como função de ativação em redes neurais artificiais, assunto sobre o qual passamos a tratar.

Figura 2.4: Impacto dos parâmetros na derivada da função penalização hiperbólica



(a) variando λ e fixando $\tau = 1$.

(b) variando τ e fixando $\lambda = 0,5$.

2.4.1 Função de ativação hiperbólica

Podemos utilizar a derivada da função penalização hiperbólica como função de ativação fixando $\lambda = 1/2$ (ou equivalentemente $\alpha = \pi/4$), e realizamos a substituição $\rho = \tau^2$. Assim, a equação 2.23 assume a forma

$$\phi(x, \rho) = \frac{1}{2}\left[1 + \frac{x}{2\sqrt{\frac{1}{4}x^2 + \rho}}\right], \tag{2.26}$$

a qual nomeamos função de ativação hiperbólica.

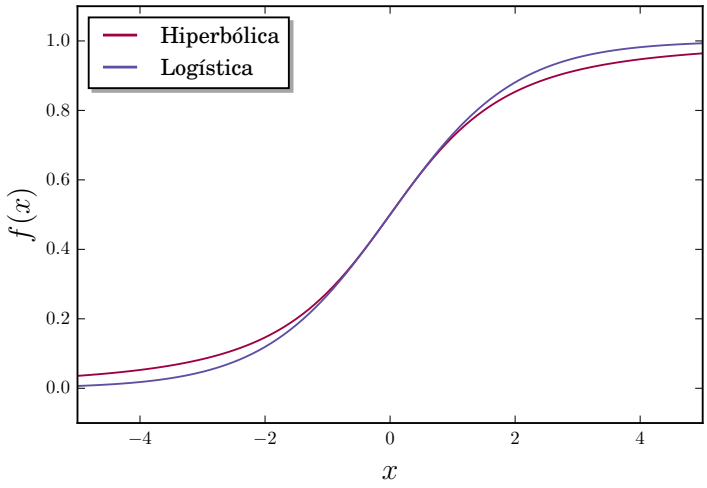
Como dito anteriormente, esta função possui forma sigmoidal, além de pertencer à classe C^∞, ou seja, é uma função infinitamente diferenciável. Adicionalmente, temos

$$\begin{cases} \lim\limits_{x \to -\infty} \phi(x, \rho) = 0 \\ \\ \lim\limits_{x \to +\infty} \phi(x, \rho) = 1, \end{cases} \tag{2.27}$$

de tal maneira que a função atua no intervalo de ativação $[0, 1]$. Desta forma, tal função se assimila à função logística, como mostra a Figura 2.5.

A função de ativação hiperbólica também pode assumir uma forma mais conveniente pela simplicidade, conforme mostram as equações (2.29) a (2.30).

Figura 2.5: Comparação entre a função de ativação hiperbólica e a função de ativação logística, com $\rho = 1$.



$$\phi(x,\rho) = \frac{1}{2}\left[1 + \frac{x}{2\sqrt{\frac{1}{4}x^2 + \rho}}\right] \tag{2.28}$$

$$= \frac{1}{2}\left[1 + \frac{x}{\sqrt{4\left(\frac{1}{4}x^2 + \rho\right)}}\right] \tag{2.29}$$

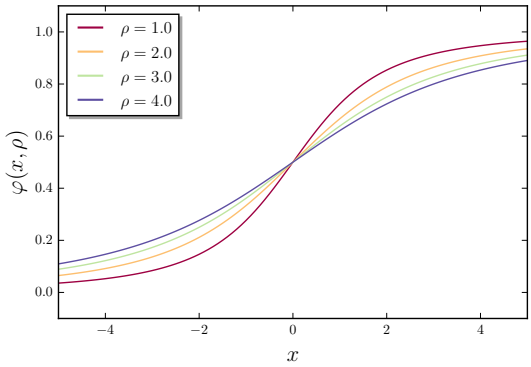
$$= \frac{1}{2}\left[1 + \frac{x}{\sqrt{x^2 + 4\rho}}\right]. \tag{2.30}$$

Além disso, podemos ver na Figura 2.6b o efeito de utilizarmos $\rho < 0$. A função sigmoidal deteriora-se de forma a criar uma descontinuidade. Temos uma curva no primeiro quadrante, com assíntota horizontal $y = 1$ e assíntota vertical $x = 2\sqrt{|\rho|}$, e uma outra situada no terceiro quadrante, esta com assíntotas horizoltal e vertical em $y = 0$ e $x = -2\sqrt{|\rho|}$, respectivamente. Portanto, o uso da função de ativação hiperbólica como proposto deve se restringir a valores de $\rho > 0$.

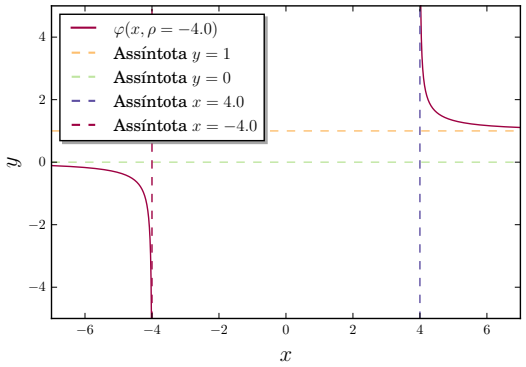
2.4.2 Função de ativação bi-hiperbólica

Xavier também propôs uma outra função sigmoidal de ativação denominada bi-hiperbólica (MIGUEZ, 2012, THOMAZ e MAIA, 2013, XAVIER, 2005), a qual é parametrizada por λ, τ_1 e τ_2, conforme vemos abaixo:

Figura 2.6: Impacto do parâmetros ρ na função de ativação hiperbólica



(a) variando ρ.



(b) tomando $\rho < 0$.

$$\psi(x) = \sqrt{\lambda^2 \left(x + \frac{1}{4\lambda}\right)^2 + \tau_1^2} - \sqrt{\lambda^2 \left(x - \frac{1}{4\lambda}\right)^2 + \tau_2^2} + \frac{1}{2}. \tag{2.31}$$

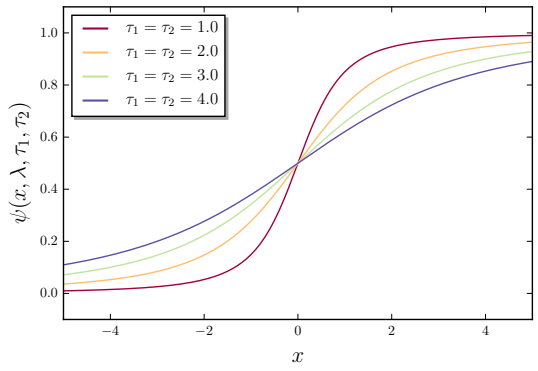
Tais parâmetros tornam-a flexível para que possa ser ajustada às condição do problema, de forma a obter uma melhor convergência (MIGUEZ, 2012, THOMAZ e MAIA, 2013, XAVIER, 2005). A função de ativação bi-hiperbólica possui comportamento sigmoidal quando $\tau_1 = \tau_2$, conforme mostra a Figura 2.7a, sendo neste caso denominada função bi-hiperbólica simétrica.

Caso $\tau_1 \neq \tau_2$, a função não possuirá forma sigmoidal, sendo neste caso denominada função bi-hiperbólica assimétrica. Tomando $\tau_1 > \tau_2$, podemos ver pela Figura 2.7b que ocorrerá uma distorção em forma de cume no lado superior direito da sigmoide. De maneira análoga, para $\tau_1 < \tau_2$ teremos uma distorção em forma de vale no lado inferior esquerdo da função, como mostra a Figura 2.7c. Tais distorções se dão pelo distanciamento de τ_1 e τ_2, de forma que quanto maior a diferença absoluta entre τ_1 e τ_2, maior a distorção.

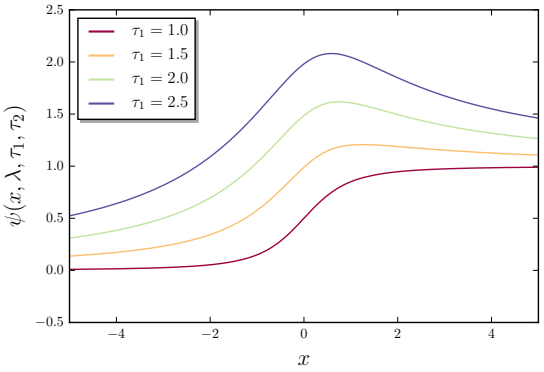
O parâmetro λ por sua vez, como podemos observar na Figura 2.7d, controla a planicidade da função, de maneira que quanto menor for λ, mais plana a função será. Utilizar valores negativos para τ_1 e τ_2 não produz efeito distinto dos valores positivos, visto que se trata de termos quadráticos. Entretanto, valores de $\lambda < 0$ fazem com que a função seja simétria em torno do eixo y, tendo portanto seu comportamento invertido, conforme a Figura 2.7e apresenta.

Da mesma maneira que a funçao de ativação hiperbólica acima apresentada, a função de ativação bi-hiperbólica possui a interessante característica de saturar mais lentamente, se comparada à funções sigmoidais similares. Com uma escolha adequada de parâmetros, tal característica faz com que a função possa convergir mais rapidamente que outras funções sigmoidais, como por exemplo a logística e tangente hiperbólica (MIGUEZ, 2012, THOMAZ e MAIA, 2013, XAVIER, 2005).

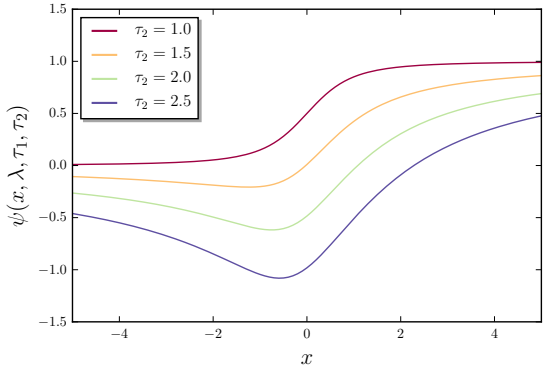
Figura 2.7: Impacto dos parâmetros λ, τ_1 e τ_2 na função de ativação bi-hiperbólica.



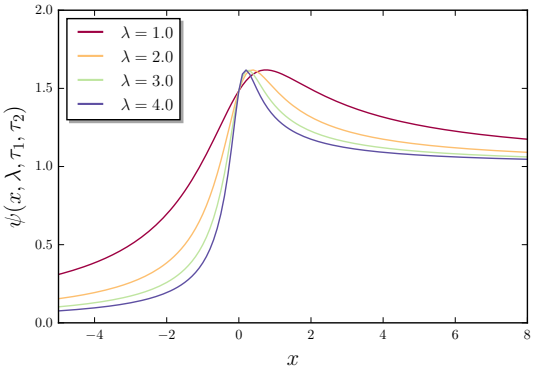
(a) $\lambda = 1$, variando τ_1 e τ_2 juntamente



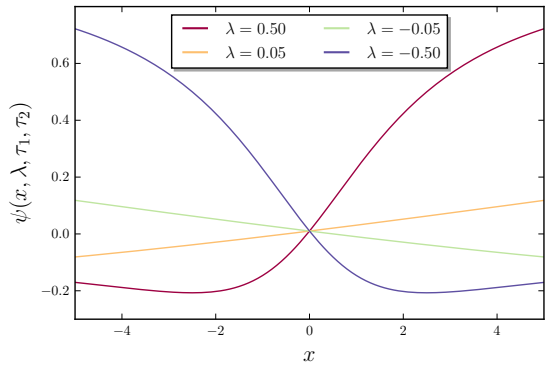
(b) $\lambda = 1$ e $\tau_2 = 1$, variando $\tau_1 > \tau_2$



(c) $\lambda = 1$ e $\tau_1 = 1$, variando $\tau_1 < \tau_2$



(d) $\tau_1 = 2$ e $\tau_2 = 1$, variando λ



(e) $\tau_1 = 2$ e $\tau_2 = 1$, efeito de $\lambda < 0$

Capítulo 3

Proposta

Neste capítulo apresentaremos nossa proposta: primeiro, uma alteração no intervalo de ativação das funções de ativação hiperbólica e bi-hiperbólica, capaz de melhorar expressivamente seus desempenhos; logo após, apresentaremos uma alternativa de suavização da função ReLU, a qual é capaz de redizir o problema do gradiente minguante presente na função Softplus; por fim, mostraremos uma solução para o problema de seleção dos parâmetros das funções anteriormente citados, criando novas funções de ativação adaptativas.

3.1 Função de ativação hiperbólica escalada

Podemos adaptar a função de ativação hiperbólica de forma que esta atue no intervalo de ativação $[-1, 1]$, assemelhando-se à função de ativação tangente hiperbólica. Tomando $\lambda = 1$ na equação 2.23 ao invés de $\lambda = 1/2$, a função passará a atuar no intervalo $[0, 2]$, bastando subtrair 1 da expressão para que passe a atuar em $[-1, 1]$. Chamaremos esta variação de função de ativação hiperbólica escalada, a qual possui a conveniente forma

$$\varphi(x, \rho) = \frac{x}{\sqrt{x^2 + \rho}}. \tag{3.1}$$

A função de ativação hiperbólica escalada possui as assíntotas

$$\begin{cases} \lim\limits_{x \to -\infty} \varphi(x, \rho) = -1 \\ \\ \lim\limits_{x \to +\infty} \varphi(x, \rho) = 1, \end{cases} \tag{3.2}$$

bem como corta o eixo das ordenadas em

$$\varphi(0, \rho) = 0, \tag{3.3}$$

A Figura 3.1 apresenta a semelhança da função de ativação hiperbólica escalada com a tangente hiperbólica.

Figura 3.1: Comparação entre a função de ativação hiperbólica escalada e a função de ativação tangente hiperbólica, com $\rho = 1$.

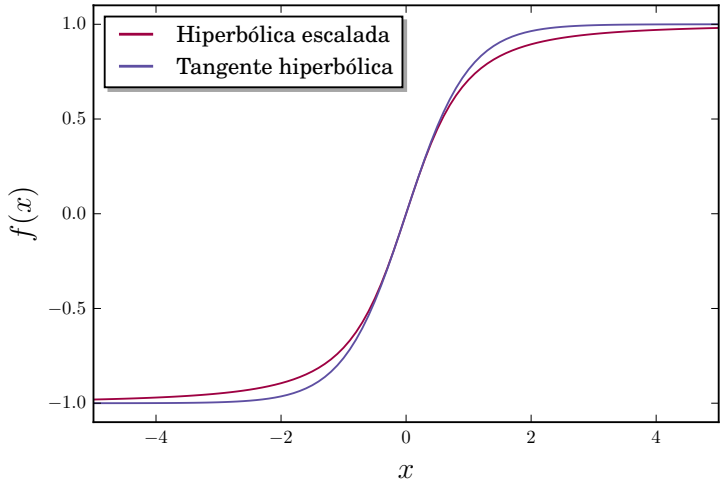
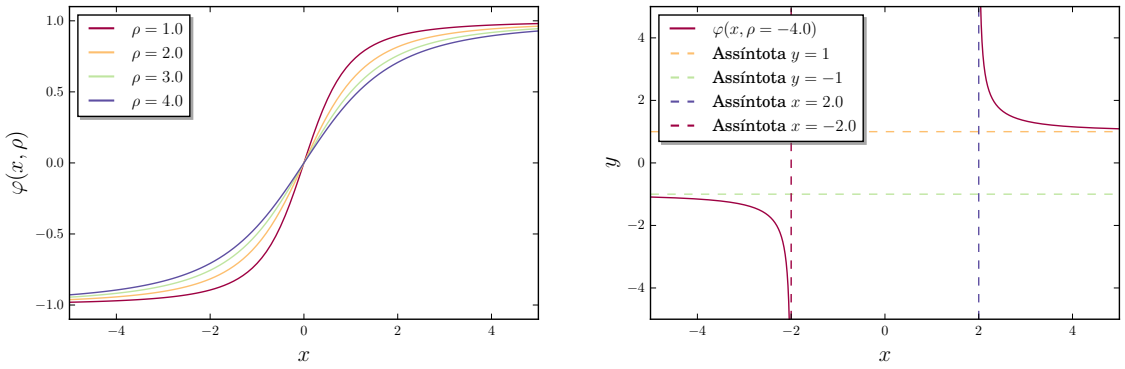


Figura 3.2: Impacto do parâmetros ρ na função de ativação hiperbólica escalada



(a) variando ρ.

(b) tomando $\rho < 0$.

3.1.1 O gradiente minguante na função hiperbólica

Assim como a função logística e tangente hiperbólica, as funções de ativação hiperbólica e hiperbólica escalada, por serem sigmoidais, também sofrem do gradiente minguante. Sem perda de generalidade, nos restringiremos à análise a derivada da função de ativação hiperbólica escalada. Temos que

$$\frac{\partial \varphi}{\partial x}(x, \rho) = \frac{\partial}{\partial x}\left(\frac{x}{\sqrt{x^2 + \rho}}\right) = \frac{\rho}{(x^2 + \rho)^{3/2}}, \tag{3.4}$$

e também

$$\lim_{x\to\pm\infty} \frac{\rho}{(x^2+\rho)^{3/2}} = 0, \tag{3.5}$$

Apesar da possibilidade de saturação da rede com a função de ativação hiperbólica escalada, a tendência é que esta seja tardia se comparada à tangente hiperbólica. Isto se deve ao fato de que, apesar da derivada se anular, este processo é muito mais demorado que com a tangente hiperbólica, conforme tipifica a relação

$$\lim_{x\to\pm\infty} \frac{\partial\varphi}{\partial x}(x,\rho) \Big/ \frac{d\tanh(x)}{dx} = \lim_{x\to\pm\infty} \frac{\partial}{\partial x}\left(\frac{x}{\sqrt{x^2+\rho}}\right) \Big/ \frac{d}{dx}\left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right) \tag{3.6}$$

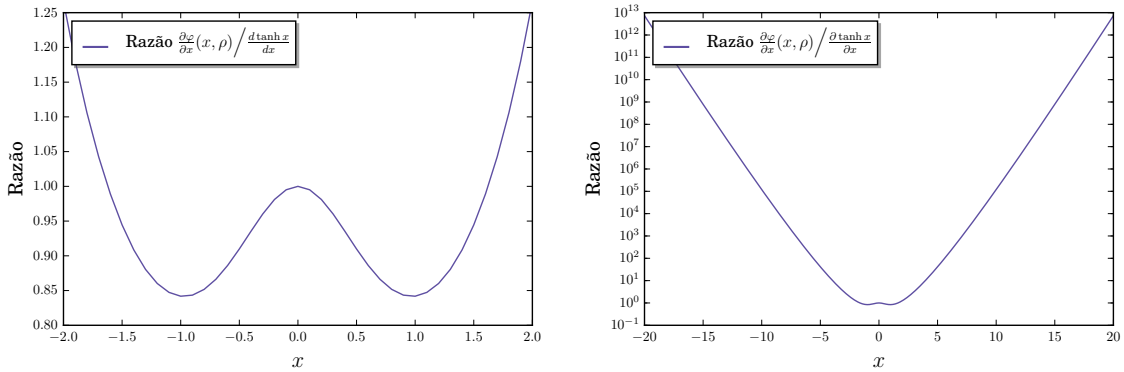
$$= \lim_{x\to\pm\infty} \frac{\rho}{(x^2+\rho)^{2/3}} \Big/ \frac{4e^{2x}}{(e^{2x}+1)^2} \tag{3.7}$$

$$= \lim_{x\to\pm\infty} \frac{e^{-2x}(e^{2x}+1)^2\rho}{4(x^2+\rho)^{2/3}} \tag{3.8}$$

$$= \infty. \tag{3.9}$$

Observe na Figura 3.3a que no intervalo $[-1, 1]$ a relação decresce, isto é, $\frac{d\tanh(x)}{dx}$ se torna cada vez maior se comparado a $\frac{\partial\varphi}{\partial x}(x,\rho)$. A partir de $x < -1$ e $x > 1$ podemos ver uma inversão deste fenômeno, de forma que rapidamente $\frac{\partial\varphi}{\partial x}(x,\rho)$ se torna maior que $\frac{d\tanh(x)}{dx}$. Tal comportamento se mantém, como vemos na Figura 3.3b, a qual mostra que para valores distantes de $x = 0$, $\frac{\partial\varphi}{\partial x}(x,\rho)$ é várias ordens de grandeza maior que $\frac{d\tanh(x)}{dx}$. Tal propriedade teórica garante uma menor saturação da função de ativação hiperbólica escalada em relação à tangente hiperbólica.

Figura 3.3: Razão $\frac{\partial\varphi}{\partial x}(x,\rho)/\frac{d\tanh(x)}{dx}$. Comparação do gradiente minguante entre as funções de ativação hiperbólica escalada e tangente hiperbólica para $\rho = 1$.



(a) intervalo $[-2, 2]$.

(b) intervalo $[-20, 20]$ em escala logarítmica.

3.2 Função de ativação bi-hiperbólica escalada

Da mesma forma que fizemos com a função de ativação hiperbólica, é possível alterarmos a formulação da função de ativação bi-hiperbólica de forma que esta passa a atuar no intervalo de ativação $[-1, 1]$, o que espera-se, trará as referidas vantagens. Para tanto, faremos uma desconstrução da mesma, de forma a compreender como realizar tal alteração mantendo-se as demais características originais.

Podemos entender a função de ativação bi-hiperbólica como sendo a subtração de duas hipérboles distintas, com mínimos equidistantes da origem no eixo das abscissas. Tais hipérboles serão chamadas h_1 e h_2, conforme apresentado abaixo.

$$h_1(x, \lambda, d, \tau_1) = \sqrt{\lambda^2(x + d)^2 + \tau_1^2}, \tag{3.10}$$

e também

$$h_2(x, \lambda, d, \tau_1) = \sqrt{\lambda^2(x - d)^2 + \tau_1^2}. \tag{3.11}$$

Desta forma, a função bi-hiperbólica pode ser definida como

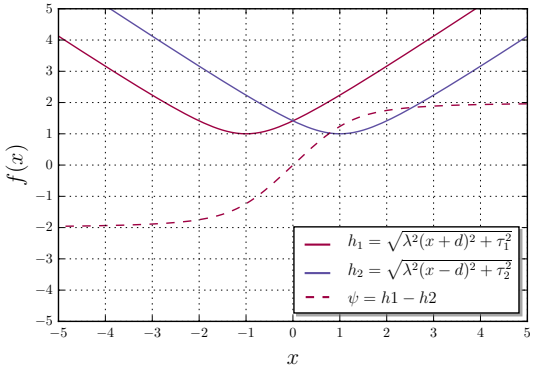
$$\psi(x, \lambda, d, \tau_1, \tau_2) = h_1(x, \lambda, d, \tau_1) - h_2(x, \lambda, d, \tau_2). \tag{3.12}$$

Com isso, são formadas duas hipérboles cuja subtração resulta na função bi-hiperbólica ψ. Alterando-se o parâmetro λ podemos controlar a angulação das hipérboles, o que resulta em um controle da planicidade de ψ. Além disso, os parâmetros τ_1 e τ_2 controlam, respectivamente, a localização do ponto mínimo no eixo das ordenadas, localizando-se nos pontos $y = \tau_1$ e $y = \tau_2$. Já o parâmetro d controla a distância que os mínimos possuem em relação à origem no eixo das abscissas, sendo portanto localizados nos pontos $x = -d$ e $x = d$. Adicionalmente, quando $\tau_1 = \tau 2$, a curva ψ possui forma sigmoidal com assíntotas horizontais em $-2\lambda d$ e $2\lambda d$. O efeito da variação de tais parâmetros pode ser melhor compreendido através da Figura 3.4.

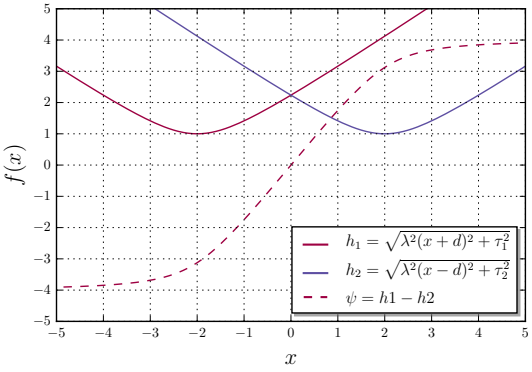
Para utilização desta função como função de ativação em redes neurais, é necessário que a função possua intervalo de ativação fixo, o que pode ser obtido utilizando-se uma fração de λ para d, de forma que a depender da fração utilizada, teremos um diferente intervalo de ativação. Com efeito, podemos verificar através da Figura 3.5 que tal estratégia é efetiva.

Portanto, para chegar à forma da função de ativação bi-hiperbólica originalmente proposta, conforme a Equação (2.31), basta tomarmos $d = 1/4\lambda$. Tal escolha de d fará com que a função atue no intervalo $[-1/2, 1/2]$. Por fim, acrescendo-se $1/2$ na fórmula, teremos uma função que atua no intervalo $[0, 1]$. Vejamos,

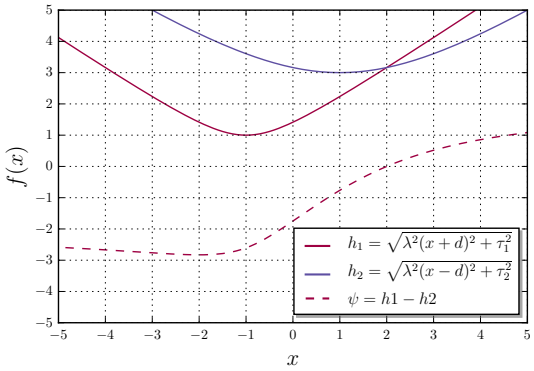
Figura 3.4: Formação da função bi-hiperbólica a partir de duas hipérboles distintas, variando λ, $d\,\tau_1$ e τ_2.



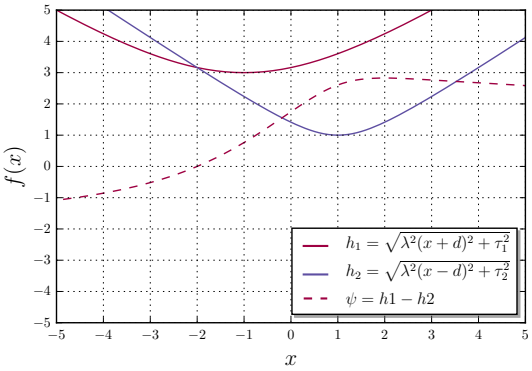
(a) $\lambda = 1$, $d = 1$, $\tau_1 = 1$ e $\tau_2 = 1$



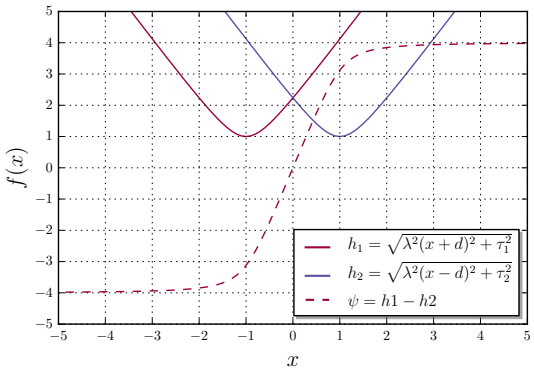
(b) $\lambda = 1$, $d = 2$, $\tau_1 = 1$ e $\tau_2 = 1$



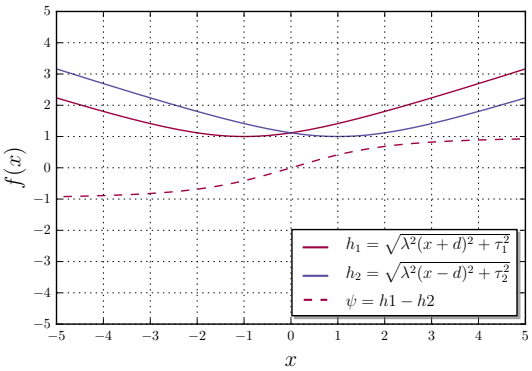
(c) $\lambda = 1$, $d = 1$, $\tau_1 = 1$ e $\tau_2 = 3$



(d) $\lambda = 1$, $d = 1$, $\tau_1 = 3$ e $\tau_2 = 1$



(e) $\lambda = 2$, $d = 1$, $\tau_1 = 1$ e $\tau_2 = 1$

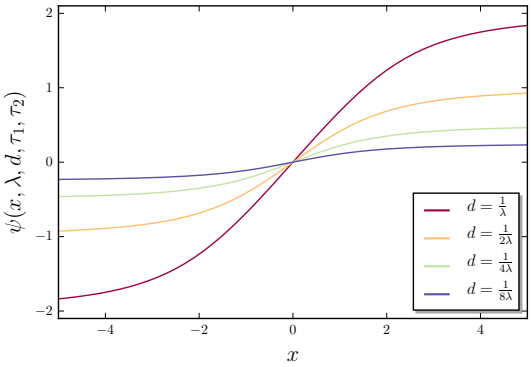


(f) $\lambda = 0,5$, $d = 1$, $\tau_1 = 1$ e $\tau_2 = 1$

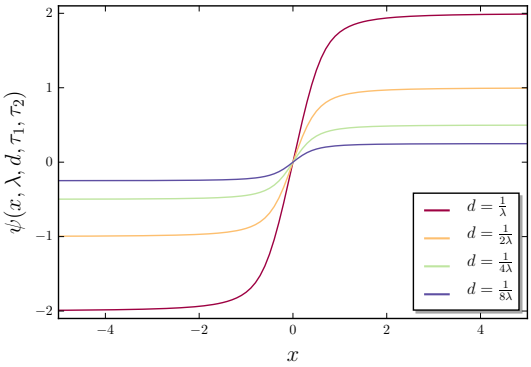
$$\psi(x, \lambda, d, \tau_1, \tau_2) = h_1\left(x, \lambda, \frac{1}{4\lambda}, \tau_1\right) - h_1\left(x, \lambda, \frac{1}{4\lambda}, \tau_2\right) + \frac{1}{2} \tag{3.13}$$

$$= \sqrt{\lambda^2\left(x + \frac{1}{4\lambda}\right)^2 + \tau_1^2} - \sqrt{\lambda^2\left(x - \frac{1}{4\lambda}\right)^2 + \tau_2^2} + \frac{1}{2}. \tag{3.14}$$

Figura 3.5: Fixando intervalo de ativação tomando d como diferentes frações de λ, com $\tau_1 = \tau_2 = 1$.



(a) $\lambda = 0,5$



(b) $\lambda = 1$

Nosso objetivo inicial era construir uma versão equivalente à função de ativação bi-hiperbólica, porém com a característica de atuar no intervalo $[-1, 1]$. Diante da análise feita, para atingirmos nosso objetivo basta tomarmos $d = 1/2\lambda$. Chamaremos esta função de bi-hiperbólica escalada, sendo esta representada por Ψ, conforme vemos abaixo:

$$\Psi(x, \lambda, \tau_1, \tau_2) = \sqrt{\lambda^2 \left(x + \frac{1}{2\lambda}\right)^2 + \tau_1^2} - \sqrt{\lambda^2 \left(x - \frac{1}{2\lambda}\right)^2 + \tau_2^2}. \tag{3.15}$$

Nos casos em que $\tau_1 = \tau_2$, chamamos a função de bi-hiperbólica escalada simétrica, podendo utilizar a notação simplificada

$$\Psi(x, \lambda, \tau) = \sqrt{\lambda^2 \left(x + \frac{1}{2\lambda}\right)^2 + \tau^2} - \sqrt{\lambda^2 \left(x - \frac{1}{2\lambda}\right)^2 + \tau^2}. \tag{3.16}$$

Caso contrário, a função também pode ser denominada função bi-hiperbólica escalada assimétrica.

3.3 Suavização Hiperbólica da ReLU

Podemos utilizar a técnica de suavização hiperbólica para obter uma versão suavizada da ReLU. Em um de seus exemplos, XAVIER e XAVIER (2014) utiliza a suavização hiperbólica sob a forma

$$\zeta(x, \tau) = \frac{x + \sqrt{x^2 + \tau^2}}{2} \tag{3.17}$$

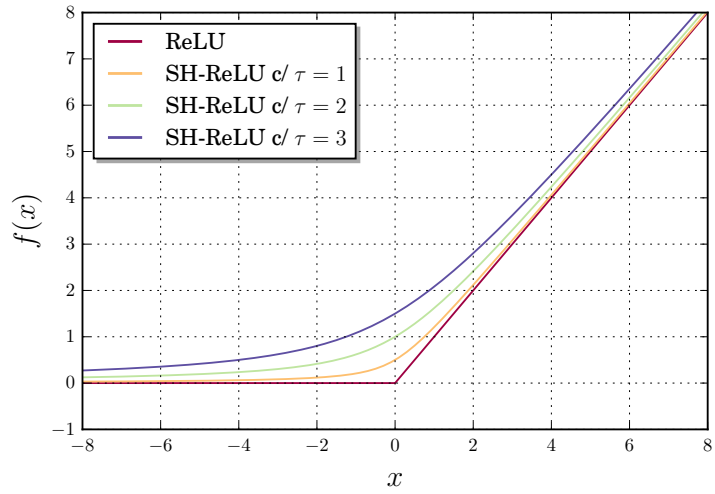
para suavizar a equação

$$r(x) = \max(0, x), \tag{3.18}$$

a qual é justamente a função retificadora. Para chegar à Equação (3.17) basta tomarmos $\lambda = 1$ na função penalidade hiperbólica $P(x, \lambda, \tau)$, conforme vista na Equação (2.22). Assim, chamaremos a função $\zeta(x, \tau)$ de suavização hiperbólica da ReLU, ou simplesmente SH-ReLU.

O parâmetro τ funciona como regulador do nível de suavização, de forma que quanto menor for τ, mais próxima a SH-ReLU estará da ReLU. Para melhor compreender o efeito da variação de τ, veja a Figura 3.6

Figura 3.6: Impacto do parâmetro τ na função SH-ReLU.



Na literatura, é comum utilizar-se a função *Softplus* (DUGAS e BENGIO, 2001, GLOROT *et al.*, 2011) como alternativa suavizada da ReLU. Esta função é a primitiva da função logística, sendo dada por

$$\varsigma(x) = \ln(1 + e^x). \tag{3.19}$$

Para uma melhor visualização das funções SH-ReLU e *Softplus* em relação à ReLU, veja a Figura 3.7a.

Perceba que a derivada da função SH-ReLU é justamente uma variação da função de ativação hiperbólica

$$\frac{\partial}{\partial x}\zeta(x, \tau) = \frac{\partial}{\partial x}\left(\frac{x + \sqrt{x^2 + \tau^2}}{2}\right) \tag{3.20}$$

$$= \frac{1}{2}\left[1 + \frac{x}{\sqrt{x^2 + \tau^2}}\right] \tag{3.21}$$

$$= \phi(x, \tau). \tag{3.22}$$

Por outro lado, a derivada da função *softplus*, por definição, é a função logística

$$\frac{d\varsigma(x)}{dx} = \frac{d}{dx}\ln(1+e^x) \tag{3.23}$$
$$= \frac{e^x}{1+e^x} \tag{3.24}$$
$$= \frac{1}{1+e^{-x}} \tag{3.25}$$
$$= \sigma(x). \tag{3.26}$$

.

Claramente, ambas funções de ativação sofrem do problema de gradiente minguante, pois

$$\lim_{x\to-\infty} \frac{1}{1+e^{-x}} = 0, \tag{3.27}$$

e também

$$\lim_{x\to-\infty} \frac{1}{2}\left[1+\frac{x}{\sqrt{x^2+\tau^2}}\right] = 0, \tag{3.28}$$

Entretanto, a SH-ReLU possui uma vantagem teórica sobre a *Softplus*, pois o efeito de gradiente minguante é retardado, como fica claro na relação

$$\lim_{x\to-\infty} \frac{\phi(x,\tau)}{\sigma(x)} = \lim_{x\to-\infty} \frac{1}{2}\left[1+\frac{x}{\sqrt{x^2+\tau^2}}\right] \Bigg/ \frac{1}{1+e^{-x}} \tag{3.29}$$
$$= \lim_{x\to-\infty} 1+\frac{x}{\sqrt{x^2+\tau^2}} \Bigg/ \frac{2}{1+e^{-x}} \tag{3.30}$$
$$= \lim_{x\to-\infty} \frac{\left(1+\dfrac{x}{\sqrt{x^2+\tau^2}}\right)(1+e^{-x})}{2} \tag{3.31}$$
$$= \lim_{x\to-\infty} \frac{1+e^{-x}+\dfrac{x}{\sqrt{x^2+\tau^2}}+\dfrac{xe^{-x}}{\sqrt{x^2+\tau^2}}}{2} \tag{3.32}$$
$$= \infty. \tag{3.33}$$

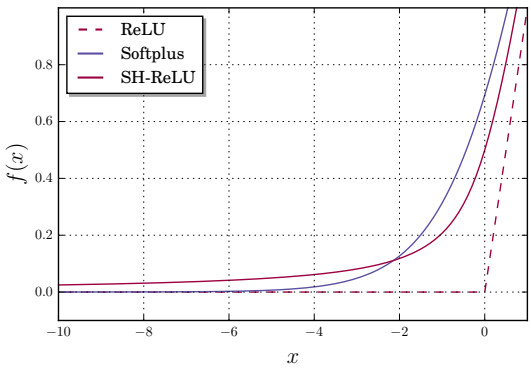
A questão do gradiente minguante das duas funções pode ser melhor compreendida através da Figura 3.7.

Com isso, a tendência é que a SH-ReLU apresente maior capacidade de convergência que a função *Softplus*.

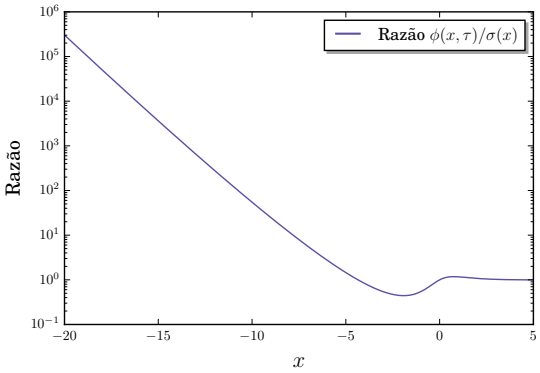
3.4 Funções de ativação hiperbólicas adaptativas

Existe um paradoxo inerente ao uso das funções de ativação paramétricas propostas neste trabalho, sobretudo no que tange ao tempo de treinamento da rede. Por um lado, funções de ativação com uma escolha de parâmetros adequadas podem ser

Figura 3.7: Comparação do gradiente minguante entre as funções SH-ReLU e *Softplus*, com $\tau = 1$.



(a) Comparação das funções ReLU, Softplus e SH-ReLU.



(b) Razão $\frac{\phi(x,\rho)}{\sigma(x)}$.

capazes de atingir a convergência em menos épocas, e até mesmo com um resultado melhor. Por outro lado, faz-se necessário uma calibração de tais parâmetros, o que invariavelmente exige avaliação de diferentes combinações.

Esta avaliação das diferentes combinações pode ser feita através de técnicas de busca exaustiva, como o *grid search*, ou até mesmo através de técnicas de computação evolucionária. Seja qual for a alternativa para avaliação das diferentes combinações de parâmetros, será necessário um tempo tão grande para isso, que tornará o argumento de convergência mais rápida ilusório do ponto de vista prático.

Também pode-se lançar um olhar sobre o *tunning* dos parâmetros exclusivamente na perspectiva de melhor qualidade nos resultados, de forma que não haja a problemática do paradoxo citado acima. Portanto, é possível realizarmos um ajuste dos parâmetros a fim de atingirmos uma capacidade superior de generalização da rede neural. Embora seja uma alternativa factível para uso prático das funções paramétricas propostas, ainda assim haverá um grande *overhead* que não pode ser ignorado. Assim, passamos a tratar de uma alternativa interessante que possa atenuar estas questões.

Cabe notar também que a análise do impacto dos parâmetros no *cross entropy*, acurácia e quantidade de épocas revelou uma faixa favorável (seções 4.3.1 e 4.3.2), a qual não necessariamente é a mesma para outros *data sets* e/ou arquitetura de rede. Tal problemática revela assim a dificuldade do uso destas funções hiperbólicas propostas do ponto de vista prático.

Entretanto, como vimos na Seção 2.3, é possível realizarmos o aprendizado de parâmetros da função de ativação através do algoritmo *backpropagation*, isto é, altera-se progressivamente os parâmetros, conforme realiza-se o treinamento da rede. Esta abordagem é interessante por permitir que um ajuste fino seja feito sem exigir testes exaustivos com as inúmeras combinações possíveis. Na literatura

citada (AGOSTINELLI *et al.*, 2015, HE *et al.*, 2015, JIN *et al.*, 2015, SCARDAPANE *et al.*, 2016, TROTTIER *et al.*, 2016), diversas funções adaptativas foram propostas a fim de serem utilizadas em conjunto com esta abordagem.

Podemos portanto nos valer desta técnica para realizar o aprendizado dos parâmetros das funções de ativação hiperbólicas propostas neste trabalho, de forma a contornar a complexa problamática da escolha dos parâmetros. Com isso, será possível gozarmos das vantagens de uma boa configuração de parâmetros, dispensando a custosa avaliação exaustiva. Finalmente, nomeamos nossas novas funções, que se utilizam da técnica adaptativa para realizar o aprendizado de seus parâmetros, os quais são discriminados juntamente.

Hiperbólica adaptativa: ρ

Bi-hiperbólica simétrica adaptativa: λ e τ

Bi-hiperbólica assimétrica adaptativa: λ, τ_1 e τ_2

Suavização hiperbólica adaptativa da ReLU: τ

Capítulo 4

Avaliação experimental

Neste capítulo faremos a avaliação experimental de todas as funções de ativação propostas no presente trabalho, sendo a organização deste capítulo como segue: primeiro descreveremos os experimentos a serem realizados com seus objetivos individuais; logo após, a metodologia utilizada para para validar a proposta, descrevendo o *dataset* utilizado, configurações do experimento e métricas; por fim, apresentaremos os resultados obtidos e faremos uma discussão das questões mais pertinentes observadas.

4.1 Objetivos dos experimentos

Primeiramente, serão feitas comparações das versões orginais das funções hiperbólica e bi-hiperbólica com suas versões escaladas. Além disso, serão comparadas as funções escaladas com algumas das funções de ativação mais convencionais: logística, tangente hiperbólica e ReLU. Após isso, avaliaremos as funções hiperbólica e bi-hiperbólica escaladas, juntamente com a suavização hiperbólica da ReLU, utlizando-se a abordagem adaptativa.

Para uma melhor organização do texto, nomeamos os experimentos de I a VII, cujos objetivos individuais são disciminados abaixo:

Experimento I Comparar o desempenho da função de ativação hiperbólica com a versão escalada apresentada neste trabalho.

Experimento II Comparar o desempenho da função de ativação bi-hiperbólica com a versão escalada apresentada neste trabalho.

Experimento III Comparar o desempenho da função de ativação hiperbólica escalada proposta neste trabalho com as funções de ativação logística, tangente hiperbólica e ReLU.

Experimento IV Comparar o desempenho da função de ativação bi-hiperbólica escalada proposta neste trabalho com as funções de ativação logística, tangente hiperbólica e ReLU.

Experimento V Comparar o desempenho das funções de ativação hiperbólicas adaptativas proposta neste trabalho com outras funções paramétricas e/ou adaptativas da literatura recente, além de funções tradicionais como a ReLU.

Experimento VI Comparar o desempenho da função de ativação bi-hiperbólicas adaptativas com outras funções paramétricas e/ou adaptativas da literatura recente, com limite reduzido de épocas e quantidade de camadas ocultas.

Experimento VII Comparar o tempo médio gasto em cada época das funções hiperbólicas em relação a funções tradicionais.

4.2 Metodologia

Passamos agora à descrição da metodologia utilizada para avaliar nossa proposta, composta por uma descrição do *dataset* utilizado; configuração do experimento, especificando algoritmo de treinamento, arquitetura da rede, inicialização dos pesos e parâmetros utilizados; descrição das métricas utilizadas.

4.2.1 *Data sets*

Em nosso trabalho utilizamos o *dataset* MNIST para realizar todas as avaliações experimententais. O MNIST consiste em uma versão resumida do *dataset* NIST para reconhecimento de texto manuscrito, contendo apenas imagens dos dígitos de 0 a 9 manuscritos. O *dataset* MNIST é composto por 60000 amostras de treinamento e 10000 de teste. Todas as imagens foram normalizadas para uma escala de cinza que varia de 0 a 255, bem como foram centralizadas e padronizadas em imagens de 28x28 pixels. Para dar robustez ao data set, as imagens do conjunto de treino e teste foram produzidas por conjuntos disjuntos de autores. Para mais detalhes sobre a contrução do *dataset*, confira LECUN e CORTES (2010). Além disso, em nossos experimentos os dados em escala de cinza foram escalados para o intervalo de 0 a 1.

4.2.2 Configuração do experimento

Como vimos anteriormente, o data set MNIST é composto por $60,000$ amostras para treinamento e $10,000$ para teste. Assim, mantemos o conjunto oficial de teste, composto por $10,000$ amostras, de forma a compatibilizar os resultados apresentados

com os da literatura. Em relação ao treinamento, das $60,000$ amostras utilizamos apenas $50,000$, utilizando as $10,000$ restantes como conjunto de validação, sendo esta divisão feita de forma aleatória. Desta forma, nosso *dataset* fica dividido em treino ($60k$), validação ($10k$) e teste ($10k$), em um experimento com limite máximo de 1000 épocas.

A fim de evitar experimentos demasiadamente longos em redes que não apresentam progresso, bem como previnir *overfitting*, utilizamos a técnica de parada prematura (do inglês *early stopping*) (CARUANA *et al.*, 2001, PRECHELT, 1998). Esta técnica consiste em definir um limite máximo de épocas sem melhora, conhecido como paciência (do ingês *patience*). Para avaliar se houve ou não melhora, compara-se o erro obtido no conjunto de validação na época atual com o obtido na época anterior. Em nossos treinamentos utilizamos uma paciência de 5 épocas sem melhora.

Primeiramente, como cada amostra do MNIST é uma imagem de 28x28 pixels, temos um total de 784 *features*. Assim, a camada inicial de nossa rede possui 784 neurônios, um para cada pixel. A quantidade de camadas ocultas varia de acordo com o experimento, entretanto, todas elas possuem em comum a característica de possuírem 800 neurônios. O valor de 800 neurônios na camada oculta para o MNIST foi utilizado com base em SIMARD *et al.* (2003). Utilizamos a função de ativação a ser avaliada tanto na camada de entrada, quanto nas camadas ocultas. Por fim, uma camada de saída com 10 neurônios, cada qual correspondente a uma das classes, ou seja, os dígitos de 0 a 9. Nesta última camada utilizamos como função de ativação a função softmax, uma alternativa diferenciável da função máximo que garante que as saídas formarão uma distribuição de probabilidade válida, isto é, que não haverá nenhuma saída com valor 0, e que a soma de todas as saídas é 1 (MIKOLOV e KOMBRINK, 2011).

A rede foi treinada utilizando o algoritmo de Gradiente Descendente Estocástico (do inglês *Stochastic Gradient Descendent*) com *mini-batchs* de tamanho 128 e taxa de aprendizado $\alpha = 0,1$. Utilizamos a regra de atualização com *momentum* de Nesterov, também conhecido como Gradiente Aceleredo de Nesterov (do inglês *Neserov's Accelerated Gradient*) (ILYA SUTSKEVER JAMES MARTENS, 2013, NESTEROV, 1983), dado pelas equações (4.1) e (4.2):

$$v_{t+1} = \mu v_t - \epsilon \nabla f(\theta_t + \mu v_t) \tag{4.1}$$

$$\theta_{t+1} = \theta_t + v_{t+1}, \tag{4.2}$$

com momentum $\mu = 0,9$. Além disso, todos os pesos da rede foram inicializados utilizando a regra normalizada de Glorot (GLOROT e BENGIO, 2010), a qual nada

mais é que um distribuição normal escalada pelo número de entradas e saídas no neurônio, mais especificamente com desvio padrão

$$\sigma = \sqrt{\frac{2}{fan_{in} + fan_{out}}}, \tag{4.3}$$

onde fan_{in} é a quantidade de entradas (i.e., quantidade de neurônios na camada anterior) e fan_{out} a quantidade de saídas (i.e., quantidade de neurônios na camada seguinte).

4.2.3 Métricas

Em todos os nossos experimentos utilizamos o *cross-entropy* como função de custo (ou erro) durante o treinamento, bem como a métrica principal para avaliar a qualidade dos resultados. Assim, ao nos referirmos aos termos erro, custo ou função de custo, estamos nos referindo, na verdade, ao *cros-entropy*. O *cross-entropy* mede a diferença entre duas distribuições de probabilidade p e q, podendo ser calculada para dados discretos através da fórmula

$$H(p, q) = -\sum_{x} p(x) \log q(x). \tag{4.4}$$

O uso *cross-entropy* no lugar do erro quadrático acelera o processo de convergência e aumenta a capacidade de generalização da rede, sobretudo quando unido a uma saída calculada através da função *softmax* (GOLIK *et al.*, 2013). Por esse motivo, muitos trabalhos de redes neurais tem utilizado o *cross-entropy* no lugar do erro quadrático, dentre os quais citamos GLOROT e BENGIO (2010), GLOROT *et al.* (2011), MAAS *et al.* (2013), MIKOLOV e KOMBRINK (2011).

Além desta, utilizamos como métrica de comparação dos modelos a quantidade de épocas, e a acurácia, a qual é dada pela fórmula

$$acc = \frac{\#\{\text{verdadeiros positivos}\} + \#\{\text{verdadeiros negativos}\}}{\#\{\text{população total}\}}. \tag{4.5}$$

Como dissemos anteriormente, o *dataset* foi dividido em conjuntos de treino, validação e teste, onde o teste é o conjunto oficial fornecido pelo MNIST. Assim, todas referência à acurácia e erro do modelo (*cross-entropy*) se tratam na verdade da aferição destas métricas no conjunto de teste, exceto quando excplicitamente indicado.

4.2.4 Implementação e hardware

Todos os experimentos foram feitos utilizando a linguagem de programação Python 3.5, utilizando-se as bibliotecas de computação científica do ecossistema SciPy (JO-

NES *et al.*, 2001–). Além disso, as redes neurais utilizadas foram construídas através da biblioteca Keras (CHOLLET, 2015). Por fim, os gráficos foram construídos através da biblioteca Matplotlib (HUNTER, 2007).

Além disso, foi utilizado um computador com processador Intel® Core™ i7-3770, com 8 GB RAM e sistema operacional Ubuntu Linux 16.04 64bits.

4.3 Resultados e discussão

Assim, apresentadas as condições sob as quais os experimentos foram realizados, passamos à descrição dos resultados obtidos e discussão dos pontos mais relevantes a serem observados.

4.3.1 Experimento I

Neste experimento faremos uma comparação do desempenhos da função de ativação hiperbólica com sua variação proposta no presente trabalho: a hiperbólica escalada. A Tabela 4.1 apresenta os resultados obtidos, contendo o *cross entropy*, quantidade de épocas do treinamento e acurácia, variando a quantidade de camadas ocultas de 1 a 6. Todos os testes aqui apresentados utilizaram o parâmetro de inclinação $\rho = 1$, visto que tal parâmetro da às curvas inclinações equivalentes às da logística e tangente hiperbólica, respectivamente.

Podemos perceber que a versão escalada alcançou um *cross entropy* expressivamente mais baixo que a hiperbólica em todos os testes. Além disso, a vantagem sobre a vesão original da função hiperbólica aumenta rapidamente conforme temos mais camadas ocultas, o que pode ser melhor visualizado através da Figura 4.1. Podemos perceber também que o menor erro se reflete diretamente na acurácia, de forma que esta obteve crescente melhora conforme a quantidade de camadas ocultas aumenta. Note também que, além de um erro menor, a versão escalada completou o treinamento em menos épocas, exceto nos testes com mais de 5 camadas, onde a convergência da função hiperbólica foi tão lenta que a parada prematura se fez rapidamente. Nestes casos, observando os valores absolutos do *cross entropy* e acurácia, pode-se considerar que a rede não foi capaz de convergir.

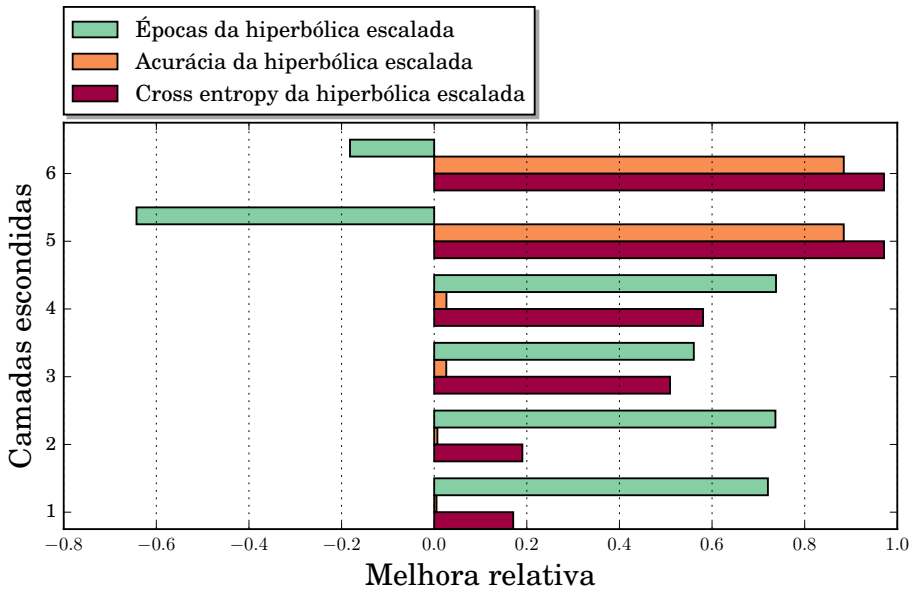
Através destes resultados, confirmamos a hipótese anteriormente levantada de que a função hiperbólica extendida teria vantagem em relação à versão tradicional, o que se deve ao fato de que ela atua no intervalo de ativação $[-1, 1]$, enquanto sua concorrente atua em $[0, 1]$.

Diante destes resultados, fica clara a larga vantagem da versão escalada para o *dataset* MNIST, tanto na qualidade dos resultados obtidos, quanto no tempo de convergência. Assim, utilizaremos somente a versão escalada da função de ativação

Tabela 4.1: Comparação de desempenho das funções hiperbólica e hiperbólica escalada de acordo com a quantidade de camadas, com $\rho = 1$.

Função	Camadas ocultas	*Cross entropy*	Épocas	Acurácia
Hiperbólica	1	0,081059	136	97,50%
Hiperbólica escalada		0,067228	38	97,99%
Hiperbólica	2	0,088034	114	97,31%
Hiperbólica escalada		0,071273	30	97,99%
Hiperbólica	3	0,149897	66	95,56%
Hiperbólica escalada		0,073555	29	98,12%
Hiperbólica	4	0,174643	84	95,47%
Hiperbólica escalada		0,073238	22	98,06%
Hiperbólica	5	2,303150	14	11,35%
Hiperbólica escalada		0,065980	23	98,17%
Hiperbólica	6	2,303819	22	11,35%
Hiperbólica escalada		0,066143	26	98,23%

Figura 4.1: Melhora relativa da função hiperbólica escalada em relação à hiperbólica, de acordo com a quantidade de camadas, com $\rho = 1$.



hiperbólica nos demais experimentos apresentados neste trabalho.

4.3.2 Experimento II

Neste experimento nosso objetivo é comparar o desempenho da função de ativação bi-hiperbólica e sua variação aqui proposta, a bi-hiperbólica escalada, ambas na versão simétrica. Diferentemente da comparação feita no Experimento I, onde comparamos as funções hiperbólicas utilizando o parâmetro fixo $\rho = 1$ devido à equivalência angular com as funções logística e tangente hiperbólica, a função bi-hiperbólica não possui uma configuração que se possa dizer trivial. Além disso, também não se pode comparar de maneira equivalente uma mesma configuração de parâmetros λ e τ em ambas funções, visto que devido à diferença na fórmula, se tratariam na verdade de angulações e níveis de distorção distintos.

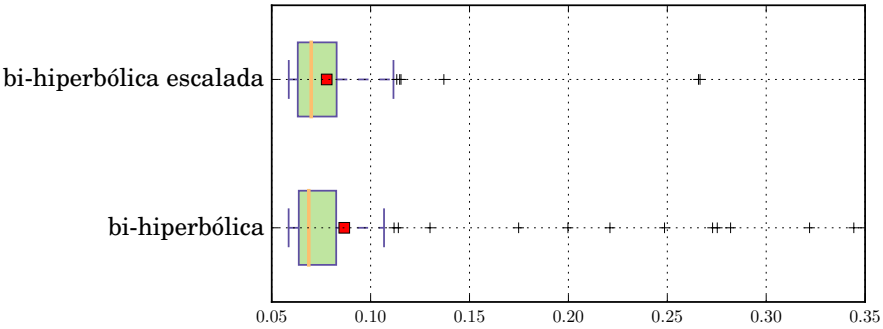
Portanto, a comparação das funções será feita através da análise descritiva dos resultados obtidos comparando o desempenho das funções com diferentes combinações de parâmetros, a saber, todas as combinações possíveis de $\lambda \in \{0,5, 1,0, 1,5, 2,0, 2,5, 3,0, 3,5, 4,0, 4,5, 5,0, 5,5\}$ e $\tau \in \{0,5, 1,0, 1,5, 2,0, 2,5, 3,0, 3,5, 4,0, 4,5, 5,0, 5,5\}$. A Figura 4.2 sumariza os resultaldos obtidos no experimento, comparando *cross entropy*, acurácia e tempo de treinamento em épocas. A tabela completa com os resultados foi suprimida visando melhorar a fluidez do texto, estando ainda disponível no Anexo A, Tabela A.1.

Primeiramente, a Figura 4.2a mostra estatísticas a respeito do *cross entropy* das diferentes configurações das funções. Note que a versão escalada da função de ativação bi-hiperbólica alcançou, em média, um custo de $0,07779$, contra $0,08667$ da função original, o que representa uma melhora relativa de $10,24\%$. Além disso, podemos notar que ocorreu uma quantidade significativamente menor de *outliers* utilizando a versão escalada e também que o *cross entropy* máximo é menor. Isso significa que a função bi-hiperbólcia escalada é mais estável em relação à versão original quando se comparam diferentes configurações de parâmetros.

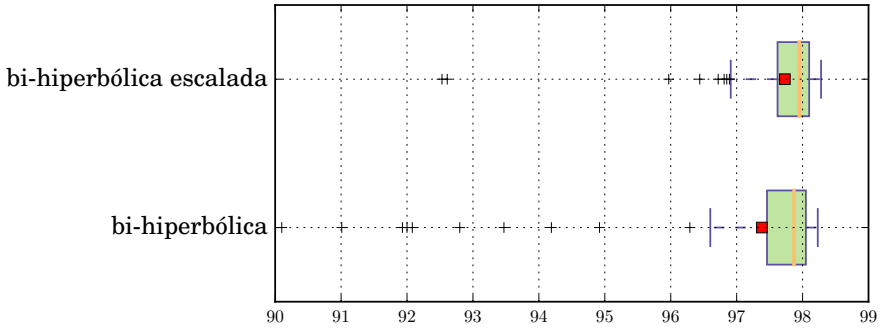
Passando à analise da acurácia, vemos na Figura 4.2b que os melhores resultados obtidos no custo se refletem na capacidade de generalização da rede. Esta conclusão se deve ao fato de que, em média, a versão escalada possui uma acurácia maior. De maneira análoga à vista anteriormente, a análise de *outliers* revela que o poder de generalização da rede utilizando a versão escalada se manteve mais estávem em relação a diferentes configurações de parâmetros.

Por fim, comparando o tempo de treinamento apresentado na Figura 4.2c, podemos perceber que a versão escalada não somente converge para resultados melhores, mas em média também leva menos épocas para atingi-los. Todavia, diferentemente do que vimos anteriormente, a análise de *outliers* mostra uma maior instabilidade no tempo de treinamento da versão escalada. Note que não apenas a função escalada apresentou um maior número de *outliers*, mas também um tempo máximo

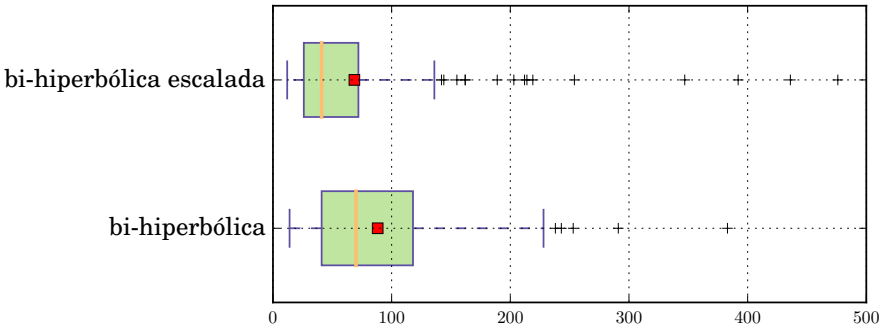
Figura 4.2: Comparação de desempenho da função bi-hiperbólica e bi-hiperbólica escalada



(a) *Cross entropy.*



(b) Acurácia.



(c) Tempo de treinamento.

maior. Isso significa que, no pior caso, a função bi-hiperbólica escalada demora mais para convergir do que a bi-hiperbólica. Este fenômeno pode ser explicado por uma permanência maior da versão escalada no processo de decréscimo de custo, o que levaria a uma mais tempo para esgotar a paciência.

Portanto, diante de *cross entropy* e acurácia em média menores e mais estáveis, e um menor tempo médio de convergência, consideramos que a função de ativação bi-hiperbólica escalada apresentou desempenho superior à versão original. Desta forma, nos próximos experimentos nos restringiremos a avaliar apenas a versão escalada.

4.3.3 Experimento III

Neste experimento faremos uma comparação entre o desempenho da função de ativação hiperbólica escalada, proposta neste trabalho, com as consagradas funções de ativação logística, tangente hiperbólica e ReLU. A Tabela 4.2 apresenta o desempenho das funções logística, tangente hiperbólica e ReLU de acordo com a quantidade de camadas, resultados que serão utilizados como *baseline* do experimento. Além disso, a tabela também mostra os resultados da melhor configuração da função hiperbólica.

Repare que os resultados deste *baseline* corroboram com LECUN *et al.* (2012), que argumenta que a tangente hiperbólica deve ser preferida à logística, pois apresenta melhor convergência. Em todos os testes, de 1 a 6 camadas, a tangente hiperbólica apresentou *cross entropy* expressivamente menor, diferença esta que se acentua conforme a quantidade de camadas aumenta. A perda de desempenho da função logística com o aprofundamento da rede é tão intenso, que a partir de 5 camadas podemos dizer que a rede nem mesmo converge, tendo *cross entropy* maior que $2,3$ e acurácia $11,35\%$. Note também que além de um erro menor, a tangente hiperbólica completou o treinamento em menos épocas, exceto no teste com 5 camadas, onde a convergência da logística foi tão lenta que a parada prematura se fez rapidamente. Também fica claro o grande poder da ReLU, visto que foi capaz de atingir rapidamente um *cross entropy* baixo. Nos testes feitos com 1 a 6 camadas ocultas, a ReLU levou por volta da metade das épocas utilizadas pela tangente hiperbólica, chegando até mesmo a um erro menor com 1 camada oculta.

Passando à análise do desempenho da função de ativação hiperbólica escalada propriamente dita, podemos ver na Tabela 4.2 que a função hiperbólica foi capaz de superar as demais no quesito *cross entropy* em todas as diferentes quantidades de camadas ocultas. Já no quesito acurácia, apenas com 1 camada oculta a ReLU alcançou melhor medida, nos demais casos a hiperbólica escalada se saiu melhor.

Também vemos na Figura 4.3 o impacto do parâmetro ρ no erro obtido e no tempo de treinamento. No que diz respeito ao erro, pode-se perceber que as configurações com ρ entre $0,2$ e $0,6$ se sairam melhor, podendo neste caso ser considerada uma boa faixa para utilização do parâmetro. O mesmo vale para a acurácia, se destacando ainda as configurações com mais camadas ocultas. Já em relação ao tempo de treinamento, as configurações com valores abaixo de $0,3$, de forma geral, concluiram o treinamento mais rapidamente. Os valores relativos à Figura 4.3 podem ser encontrados na Tabela anexa A.2.

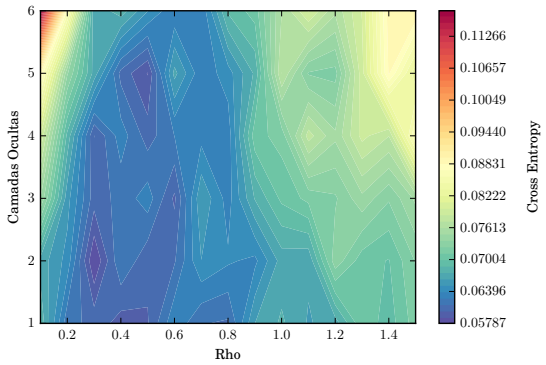
Analisaremos a melhora relativa do *cross entropy* de acordo com a quantidade de camadas ocultas. Buscando uma melhor leitura dos resultados, a Figura 4.4 apresenta apenas as 5 melhores configurações do parâmetro ρ para cada uma das

Tabela 4.2: Resultados do Experimento III.

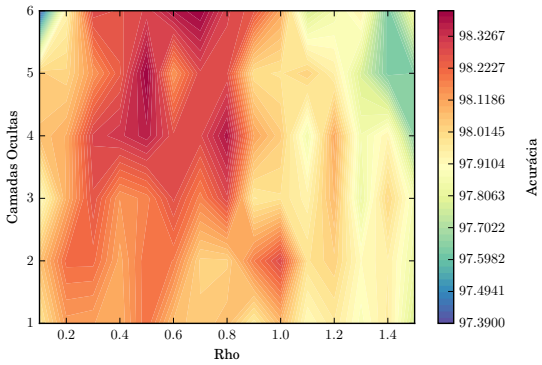
Função	Camadas ocultas	*Cross entropy*	Épocas	Acurácia
Logística	1	0,090871	129	97,34%
Tangente Hiperbólica		0,065719	42	97,98%
ReLU		0,060111	24	**98,15%**
Hiperbólica escalada		**0,059713**	**23**	98,10%
Logística	2	0,107106	90	96,81%
Tangente Hiperbólica		0,064815	30	98,17%
ReLU		0,069601	21	98,15%
Hiperbólica escalada		**0,057868**	**16**	**98,21%**
Logística	3	0,128179	74	96,28%
Tangente Hiperbólica		0,068009	31	97,98%
ReLU		0,076670	**17**	98,18%
Hiperbólica escalada		**0,060064**	22	**98,27%**
Logística	4	0,144117	97	96,35%
Tangente Hiperbólica		0,071242	19	97,98%
ReLU		0,078585	**11**	98,01%
Hiperbólica escalada		**0,060281**	13	**98,29%**
Logística	5	2,308586	10	11,35%
Tangente Hiperbólica		0,069509	19	98,14%
ReLU		0,089987	**12**	97,98%
Hiperbólica escalada		**0,059021**	15	**98,41%**
Logística	6	2,302090	29	11,35%
Tangente Hiperbólica		0,076392	23	98,06%
ReLU		0,085260	**11**	97,93%
Hiperbólica escalada		**0,062927**	20	**98,41%**

quantidades de camadas ocultas. Além disso, a comparação com a função logística foi suprimida, devido a seu desempenho defasado em relação à tangente hiperbólica e ReLU. Também podemos ver as 3 melhores combinações de parâmetro para cada diferente nível de profundidade da rede na Tabela 4.3. Tais resultados foram retirados

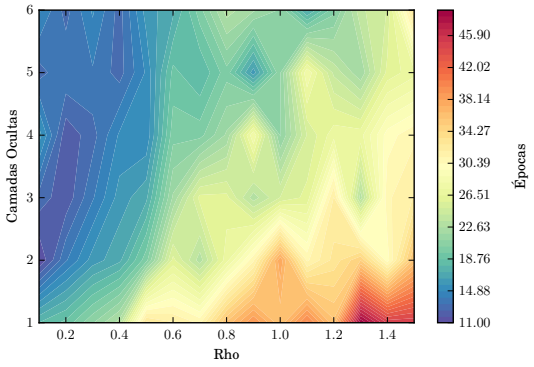
Figura 4.3: Impacto do parâmetro ρ da função de ativação hiperbólica escalada de acordo com o número de camadas ocultas.



(a) Impacto no *Cross entropy*.



(b) Impacto na acurácia.



(c) Impacto no tempo de treinamento.

da tabela completa, no Anexo A.2.

Tabela 4.3: Melhora relativa do *cross entropy* utilizando a função de ativação hiperbólica escalada de acordo com ρ, em relação à ReLU, tangente hiperbólica e logística, 3 melhores resultados, para 1 a 6 camadas ocultas. Resultados completos disponíveis em Tabela anexa A.2.

ρ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
				relu	tanh	sigmoid
1 camada oculta						
0,4	0,059713	98,10%	23	0,66%	9,14%	34,29%
0,5	0,060198	98,20%	33	-0,14%	8,40%	33,75%
0,3	0,061176	98,12%	21	-1,77%	6,91%	32,68%

Continua na próxima página

ρ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
				relu	tanh	sigmoid
2 camadas ocultas						
0,3	0,057868	98,21%	16	16,86%	10,72%	45,97%
0,5	0,060603	98,20%	20	12,93%	6,50%	43,42%
0,6	0,061377	98,18%	26	11,82%	5,31%	42,70%
3 camadas ocultas						
0,6	0,060064	98,27%	22	21,66%	11,68%	53,14%
0,3	0,060885	98,26%	14	20,59%	10,48%	52,50%
0,4	0,061910	98,14%	16	19,25%	8,97%	51,70%
4 camadas ocultas						
0,3	0,060281	98,29%	13	23,29%	15,39%	58,17%
0,5	0,060932	98,36%	15	22,46%	14,47%	57,72%
0,6	0,062752	98,27%	20	20,15%	11,92%	56,46%
5 camadas ocultas						
0,5	0,059021	98,41%	15	34,41%	15,09%	97,44%
0,4	0,061104	98,24%	13	32,10%	12,09%	97,35%
0,7	0,063066	98,28%	18	29,92%	9,27%	97,27%
6 camadas ocultas						
0,7	0,062927	98,41%	20	26,19%	17,63%	97,27%
0,6	0,062929	98,37%	17	26,19%	17,62%	97,27%
0,5	0,065240	98,29%	16	23,48%	14,60%	97,17%

Primeiramente, para 1 camada oculta a função de ativação hiperbólica foi capaz

de superar consistentemente as funções logística e tangente hiperbólica. Entretanto o mesmo não ocorre com a ReLU, pois apenas uma das combinações de parâmetro apresentou uma melhora de $0,66\%$, possuindo as demais um desempenho inferior.

Por outro lado, a partir de 2 camadas a função é capaz de alcançar um erro mais baixo que todas as demais funções, para diversas escolhas distintas de ρ. Assim, a função de ativação hiperbólica chega a uma melhora relativa próxima de 100% em relação à logística, superiores a 30% para a ReLU e próximos de 20% quanto à tangente hiperbólica. Adicionalmente, repare que para 3 e 5 camadas, todas as escolhas de parâmatros escolhidas foram superiores às funções do baseline. Em todas as configurações a acurácia acompanha o *cross entropy*, todavia a melhora é pouco expressiva.

Tais experimentos confirmam a hipótese de que a função de ativação hiperbólica possui uma grande capacidade convergência, sobrepujando consistentemente o baseline.

4.3.4 Experimento IV

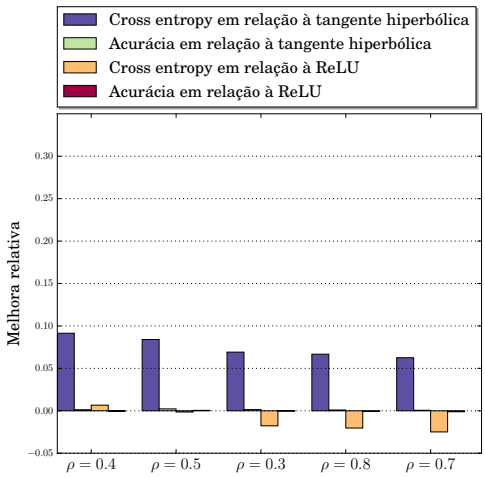
Neste experimento faremos uma comparação da nossa função de ativação bi-hiperbólica escalada com as funções logística, tangente hiperbólica e ReLU de acordo com a quantidade de camadas. Assim, o *baseline* deste experimento é equivalente ao utilizado no Experimento III (Tabela 4.2), sendo portanto suprimido para melhor fluidez do texto. Tais resultados ainda poderão ser consultados no Anexo A, Tabela A.3.

Neste experimento, medimos o desempenho da função de ativação bi-hiperbólica (na versão escalada) com diferentes combinações de parâmetros, a saber, todas as combinações possíveis de $\lambda \in \{0,5,1,0,1,5,2,0,2,5,3,0,3,5,4,0,4,5,5,0,5,5\}$ e $\tau \in \{0,5,1,0,1,5,2,0,2,5,3,0,3,5,4,0,4,5,5,0,5,5\}$. O experimento foi feito para arquiteturas com 1 a 6 camadas ocultas, e comparado com o baseline.

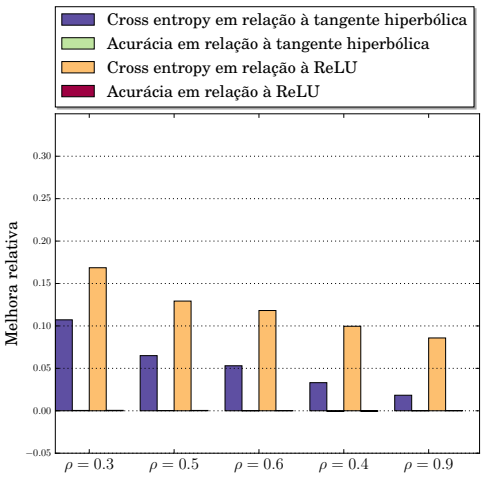
Primeiramente, faremos uma análise do impacto das diferentes combinações de λ e τ no processo de convergência, avaliando função de custo (*cross entropy*), acurácia e épocas de treinamento nas arquiteturas de 1 a 6 camadas ocultas. Tendo feita esta análise, passaremos ao objetivo final do experimento, que é comparar o desempenho da função com o baseline.

A Figura 4.5 mostra como o *cross entropy* se comporta conforme variam-se os parâmetros, de acordo com a quantidade de camadas ocutas. Podemos perceber que para 1 e 2 camadas existe a tendência de um menor custo para combinações ao redor da faixa que vai de $\tau \approx 1 \approx 2$ até $\lambda \approx 3 \approx 4$. A partir de 3 camadas esta faixa se amplia consideravelmente, de forma que a calibração dos parâmetros é menos sensível. A análise da acurácia acompanha o valor do *cross entropy*, conforme

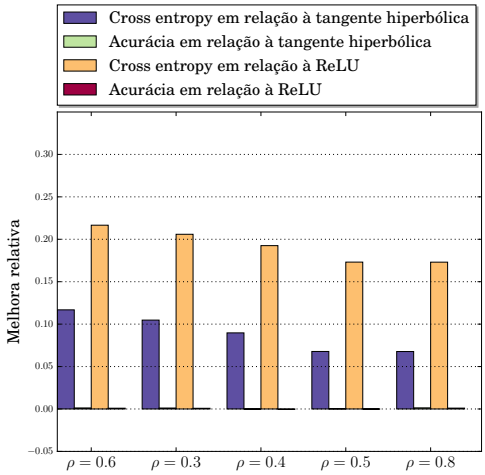
Figura 4.4: Melhora de diferentes configurações da função de ativação hiperbólica em relação à tangente hiperbólica e ReLU, de acordo com a quantidade de camadas ocultas.



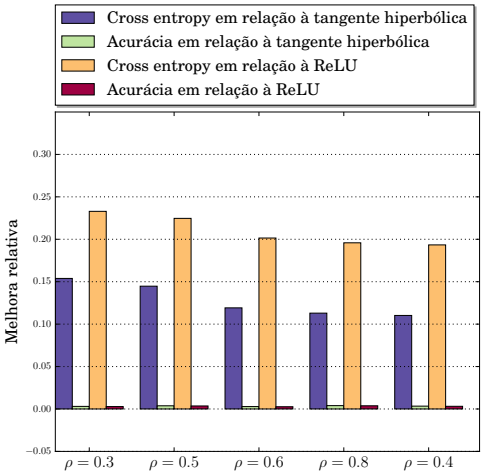
(a) 1 camada oculta



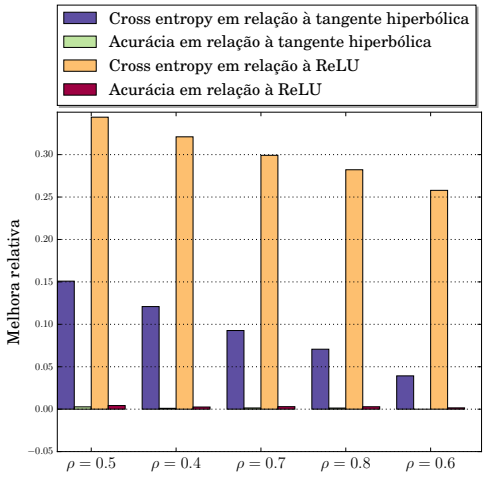
(b) 2 camadas ocultas



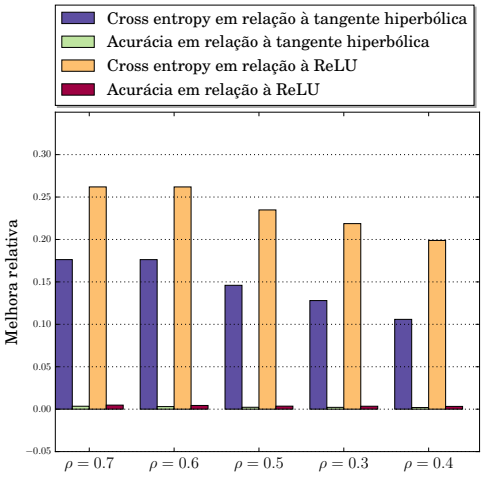
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas

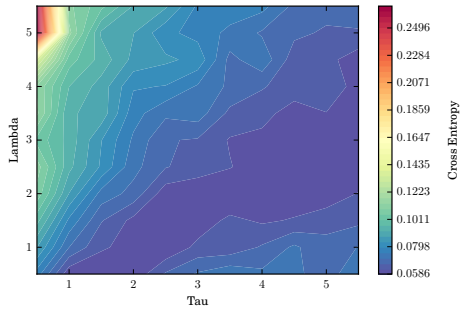


(f) 6 camadas ocultas

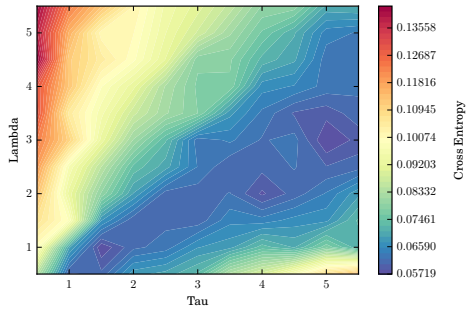
mostra a Figura 4.6

Passando à análise do tempo de convergência, podemos ver pela Figura 4.7 que

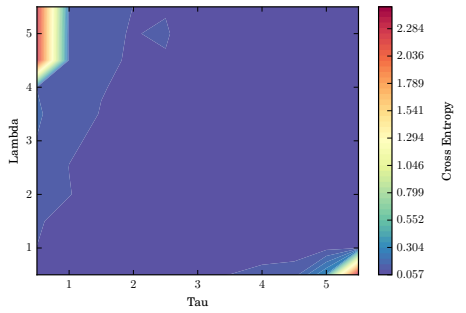
Figura 4.5: Impacto dos parâmetros λ e τ da função de ativação bi-hiperbólica no *Cross entropy*.



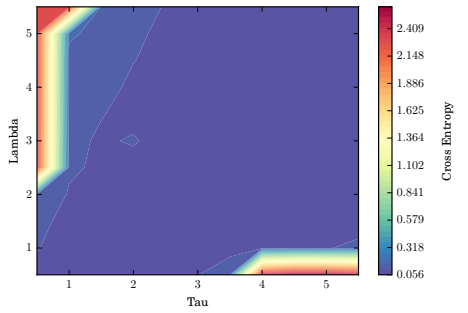
(a) 1 camada oculta



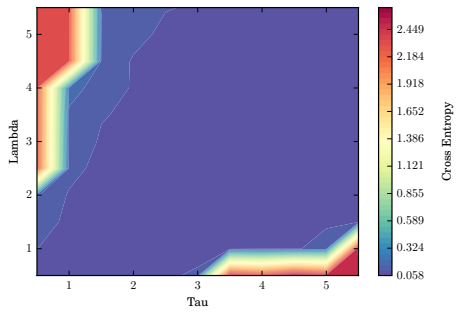
(b) 2 camadas ocultas



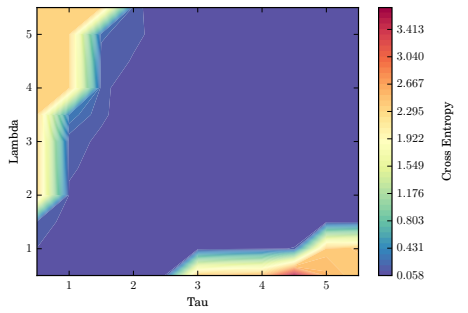
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas

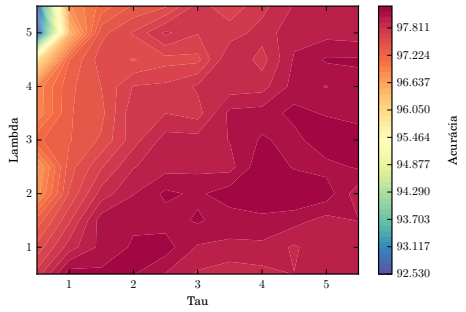


(f) 6 camadas ocultas

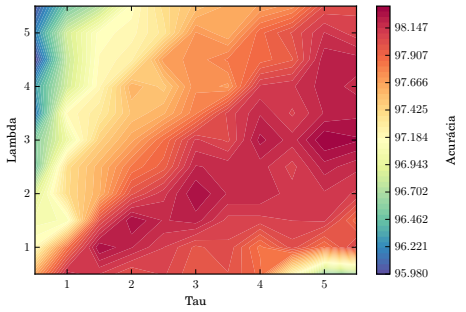
para 1 e 2 camadas ocultas a tendência é que conforme λ e τ crescem, o tempo de convergência diminui. Entretanto, vale notar que apesar de valores elevados destes parâmetros levarem a um número reduzido de épocas, não necessariamente isso significa rápida convergência. Pelo contrário, vemos que as combinações com λ e τ próximo a 5 tiveram um péssimo desempenho no quesito de erro, o que podemos considerar como não-convergência do modelo. Tal fenômeno ocorre devido ao treinamento ser feito com critério de parada de iterações sem melhora, de forma que estas combinações rapidamente deixam de progredir, esgotando a paciência e causando uma parada prematura no *backpropagation*.

Por outro lado, acima de 3 camadas vemos que o tempo de convergência passa a ser melhor justamente na faixa citada anteriormente. Desta forma, apesar de a faixa

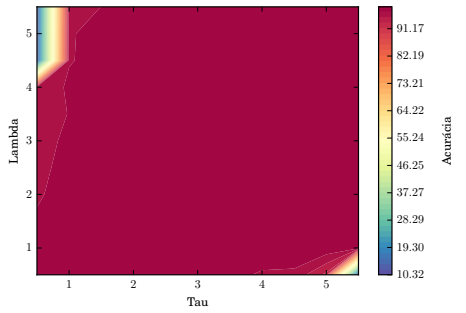
Figura 4.6: Impacto dos parâmetros λ e τ da função de ativação bi-hiperbólica na acurácia.



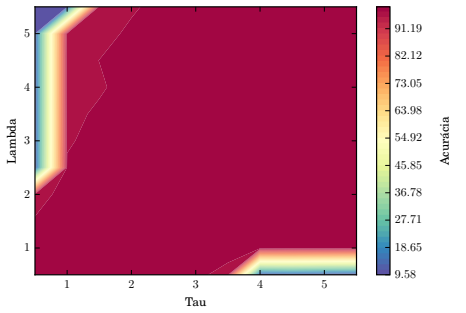
(a) 1 camada oculta



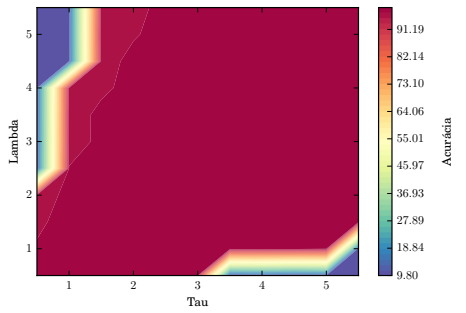
(b) 2 camadas ocultas



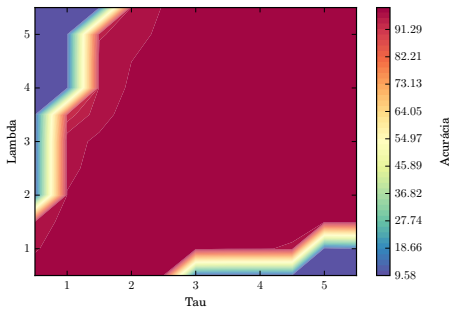
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas

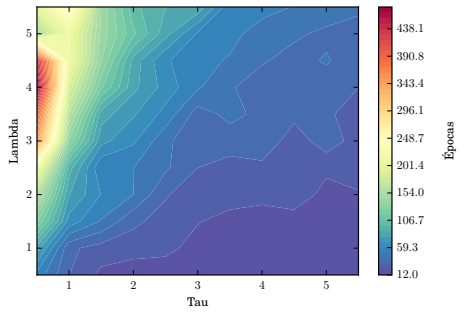


(f) 6 camadas ocultas

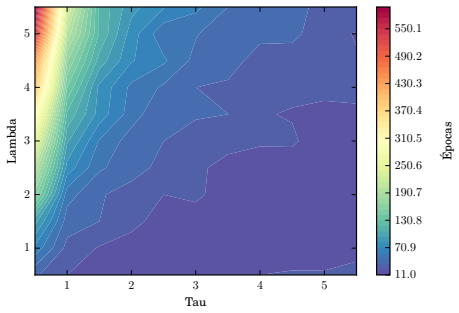
onde o erro é baixo expandir com o aumento do número de camadas ocultas, a mesma faixa é a que apresenta melhor velocidade de convergência, sendo portanto a faixa mais apropriada de combinações de parâmetros. Também vemos que combinações com τ baixo e λ alto apresentam menor quantidade de épocas, o que também é explicado pela questão da parada prematura por falta de progresso, já que estas combinações apresentaram erro elevado.

Passando à comparação da função bi-hiperbólica com o baseline, a Tabela 4.4 apresenta uma versão resumida dos resultados obtidos, contendo o *cross entropy*, acurácia e quantidade de épocas para completar o treinamento. Além destas colunas, temos também outras três colunas, que mostram a melhora relativa do *cross entropy* obtido com a combinação de parâmetros λ e τ, em relação ao baseline. Veremos aqui

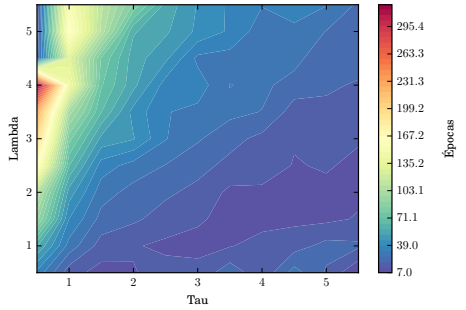
Figura 4.7: Impacto dos parâmetros λ e τ da função de ativação bi-hiperbólica no tempo de treinamento.



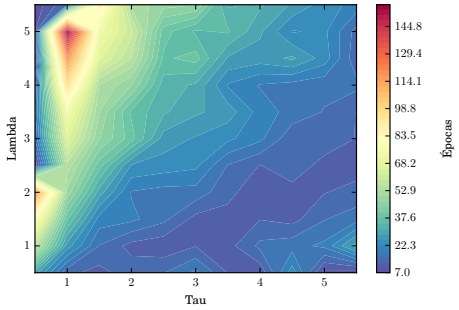
(a) 1 camada oculta



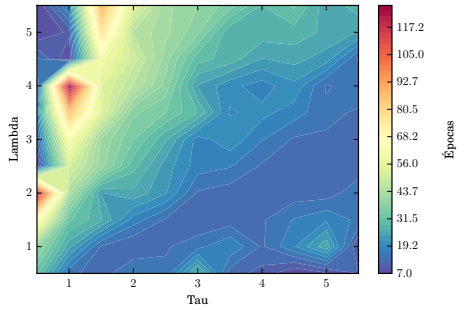
(b) 2 camadas ocultas



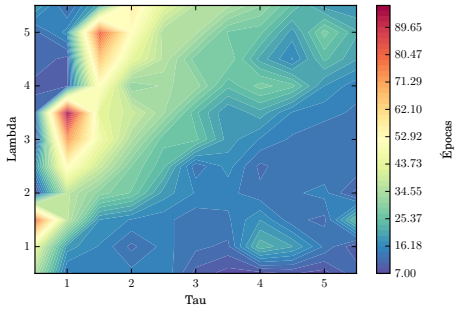
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas



(f) 6 camadas ocultas

apenas os 3 melhores resultados para cada nível de profundidade, de 1 a 6 camadas ocultas, podendo ainda os resultados completos serem vistos na Tabela anexa A.4.

Tabela 4.4: Melhora relativa do *cross entropy* utilizando a função de ativação bi-hiperbólica escalada, em relação à ReLU, tangente hiperbólica e logística tomando os 3 melhores resultados para cada nível de profundidade, de 1 a 6 camadas ocultas. Consulte a tabela completa em A.4.

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
1 camada oculta							
3,5	2,0	0,058617	98,23%	26	7,65%	14,29%	33,12%
5,5	2,5	0,058713	98,18%	24	7,50%	14,15%	33,01%
2,5	1,0	0,058921	98,22%	24	7,17%	13,85%	32,78%
2 camadas ocultas							
5,0	3,0	0,057192	98,34%	15	16,17%	12,95%	42,63%
1,5	1,0	0,057462	98,29%	22	15,77%	12,54%	42,36%
4,0	2,0	0,058585	98,19%	14	14,13%	10,83%	41,23%
3 camadas ocultas							
4,5	3,0	0,056720	98,24%	14	20,17%	14,42%	50,10%
5,5	3,5	0,058129	98,23%	15	18,18%	12,29%	48,86%
4,0	2,5	0,058137	98,36%	17	18,17%	12,28%	48,85%
4 camadas ocultas							
5,5	4,5	0,056485	98,32%	14	31,32%	19,37%	63,54%
1,0	0,5	0,057696	98,36%	14	29,85%	17,64%	62,76%
1,5	1,0	0,057998	98,23%	16	29,48%	17,21%	62,56%
5 camadas ocultas							
4,0	3,0	0,057927	98,40%	17	31,13%	20,97%	97,48%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
4,5	3,5	0,059495	98,37%	18	29,26%	18,83%	97,42%
1,5	1,0	0,059771	98,42%	14	28,93%	18,45%	97,40%
6 camadas ocultas							
2,0	1,5	0,057881	98,47%	16	53,44%	14,02%	97,49%
1,5	1,0	0,058977	98,55%	16	52,56%	12,39%	97,44%
4,0	2,5	0,059377	98,37%	12	52,23%	11,79%	97,42%

Faremos nossa análise sobretudo através dos gráficos da Figura 4.8, os quais apresentam de maneira intuitiva e resumida dos resultados contidos na Tabela anexa A.4, mostrando a melhora relativa do erro em relação à tangente hiperbólica e ReLU das 10 melhores combinações de parâmetros, para 1 a 6 camadas ocultas. Os resultados da função logística foram omitidos dos gráficos por possuirem erro muito elevado, o que prejudicaria a visualização da comparação com as demais funções do nosso baseline.

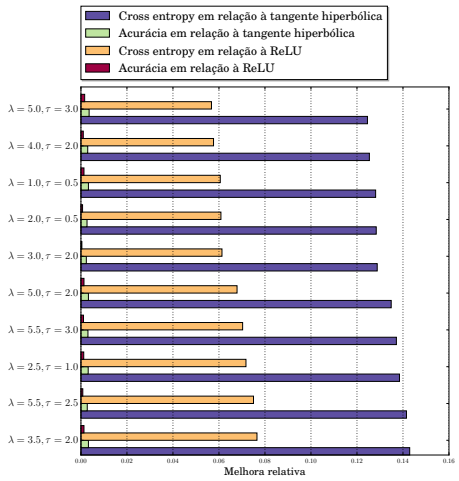
Podemos ver que a função de ativação bi-hiperbólica é capaz de superar consistentemete o baseline para todas as profundidades de rede avaliados, chegando nos melhores casos a uma melhora relativa de quase 100% em relação à logística, maiores que 50% para a ReLU e do que 20% para a tangente hiperbólica. Assim, a depender da escolha adequada de parâmetros, a função de ativação bi-hiperbólica mostra-se extremamente competitiva com suas concorrentes. A acurácia melhora juntamente ao *cross entropy*, porém de maneira tímida.

4.3.5 Experimento V

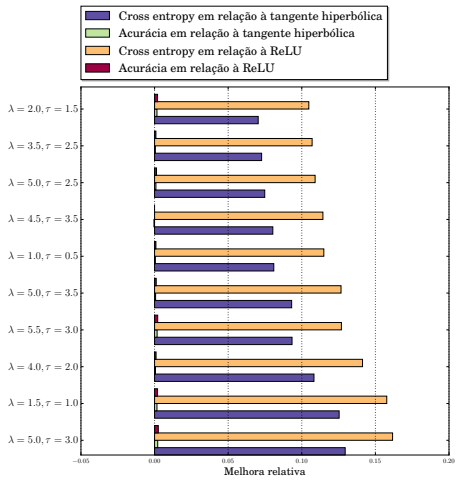
Neste experimento realizaremos uma comparação do desempenho das funções de ativação hiperbólicas adaptativas propostas neste trabalho com algumas outras propostas paramétricas e/ou adaptativas da literatura recente. Além destas, também avaliaremos o desempenho das funções logística, tangente hiperbólica, ReLU e softplus, as quais já foram descritas anteriormente no presente trabalho. Vale lembrar que os experimentos foram feitos utilizando-se as versões escaladas das funções de ativação hiperbólicas, pelas notórias vantagens obtidas através delas.

Para uma melhor leitura dos resultados, utilizaremos as seguintes abreviaturas

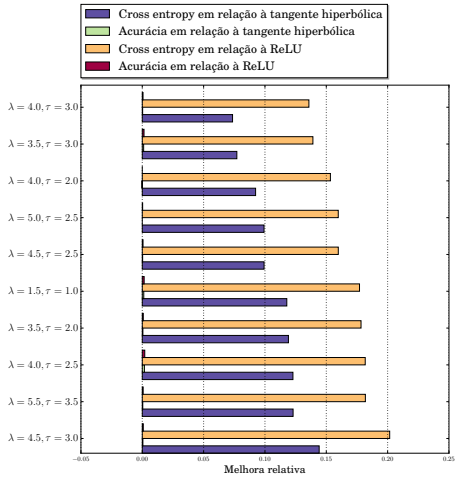
Figura 4.8: Melhora de diferentes configurações da função de ativação bi-hiperbólica em relação à tangente hiperbólica e ReLU, de acordo com a quantidade de camadas ocultas.



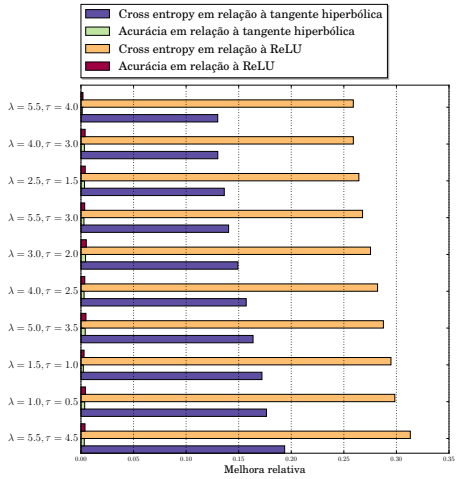
(a) 1 camada oculta



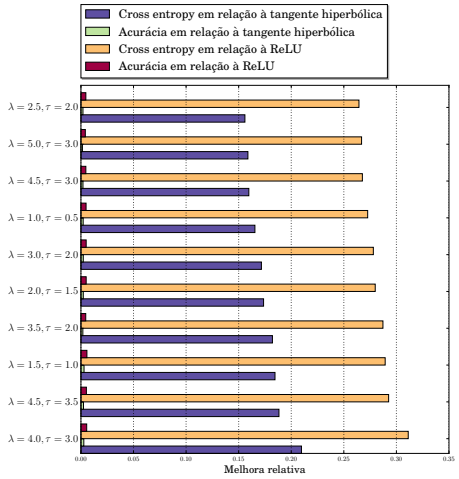
(b) 2 camadas ocultas



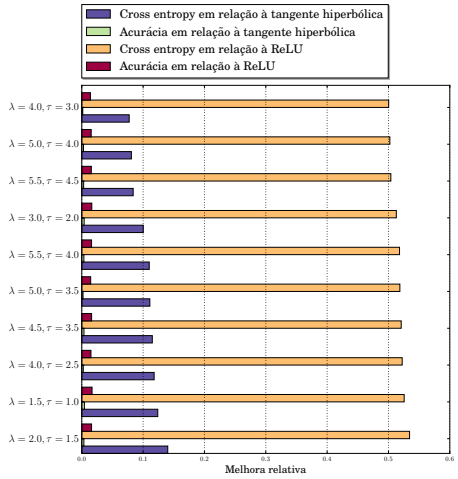
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas



(f) 6 camadas ocultas

para as funções de ativação:

HA: Hiperbólica Adaptativa

BHAA: Bi-Hiperbólica Assimétrica Adaptativa

BHSA: Bi-Hiperbólica Simétrica Adaptativa

SHA-ReLU: Suavização Hiperbólica Adaptativa da ReLU

LReLU: Leaky ReLU (MAAS *et al.*, 2013)

P-Softplus: Parametric softplus (MCFARLAND *et al.*, 2013)

PReLU: Parametric ReLU (HE *et al.*, 2015)

TReLU: Thresholded ReLU (KONDA *et al.*, 2015)

ELU: Exponential Linear Unit (CLEVERT *et al.*, 2015)

SReLU: S-sharped ReLU (JIN *et al.*, 2015)

PELU: Parametric Exponential Linear Unit (TROTTIER *et al.*, 2016)

Uma questão a ser considerada na abordagem adaptativa é a inicialização dos parâmetros, a qual é um fator preponderante para um bom desempenho da técnica. Assim, passamos a descrever como foram inicializados os parâmetros das funções avaliadas neste estudo. Primeiramente, nas funções BHAA e BHSA o parâmetro λ foi inicializado com a constante 1. Além disso, o parâmetro ρ e os parâmetros τ das funções HA, BHAA, BHSA e SHA-ReLU foram inicializados utilizando-se a regra de Glorot (GLOROT e BENGIO, 2010), a qual já foi descrita na Equação (4.3).

Agora, passamos à descrição da inicialização dos parâmetros das funções existentes na literatura: inicialmente, tomamos a Leaky ReLU com valor $\alpha = 0,3$; Parametric Softplus com $\alpha = 0,2$ e $\beta = 5,0$; Parametric ReLU com os valores $\alpha = 0,0$; Thresholded ReLU utilizando $\theta = 1,0$; ELU com $\alpha = 1,0$; Parametric ELU tomando $\alpha = \beta = 1,0$. Já na S-sharped ReLU utilizamos $t_{\text{left}} = 0$, t_{right} e a_{left} com a regra de Glorot, e por fim $a_{\text{right}} = 1,0$.

Primeiramente, faremos uma análise da qualidade do ponto de vista da qualidade dos resultados, isto é, do *cross entropy* de teste obtido ao final do treinamento. Assim, a Tabela 4.5 apresenta os resultados ao final do processo de convergência para diversas funções, para arquiteturas de 1 a 6 camadas ocultas. Para uma melhor visualização, os resultados que consideramos como convergência são apresentados na Figura 4.9, critério este que é um custo menor que 1 e em menos de 30 épocas. Também podemos visualizar a acurácia e o tempo de convergência correspondente através das figuras 4.10 e 4.11, respectivamente. Assim, estas figuras servirão de

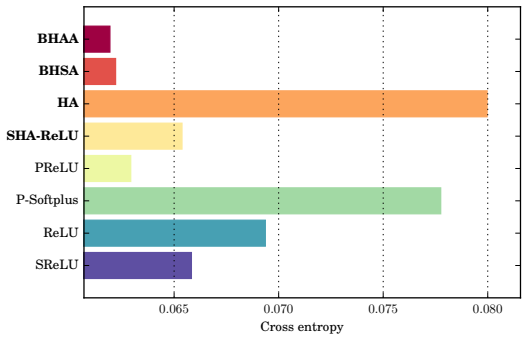
referência para nossa análise. Adicionalmente, os resultados completos deste experimento para acurácia e quantidade de épocas para convergência podem ser vistas nas tabelas anexas A.5 e A.6.

Tabela 4.5: *Cross entropy* das funções avaliadas no Experimento V.

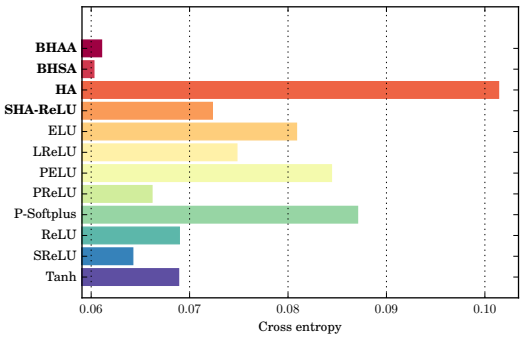
Função	camadas ocultas					
	1	2	3	4	5	6
BHAA	**0,061927**	0,061054	0,066664	0,074846	0,078889	0,089934
BHSA	0,062199	**0,060273**	**0,065787**	0,075801	0,081911	0,092505
HA	0,079967	0,101389	2,543766	2,585694	2,408141	2,412938
LReLU	0,068539	0,074803	0,075933	0,075499	0,097079	0,102241
Logística	0,080830	0,096958	0,117214	0,154980	2,306296	2,301490
P-Softplus	0,077749	0,087063	0,082305	0,086606	0,085508	0,118638
PELU	0,078080	0,084411	0,083562	0,076065	0,075589	0,085901
PReLU	0,062923	0,066177	0,074676	0,081069	0,092470	0,085398
ReLU	0,069361	0,068954	0,070965	0,080477	0,089774	0,108437
SHA-ReLU	0,065375	0,072311	0,085158	0,090635	0,086495	0,088243
SReLU	0,065825	0,064225	0,072191	0,074005	0,086183	0,090351
Softplus	0,087899	0,104508	0,105793	0,105297	0,116048	0,117245
TReLU	2,301266	2,301187	2,301076	2,301199	2,301184	2,300955
Tanh	0,067953	0,068881	0,068733	**0,069956**	**0,073910**	**0,070003**

Observando os resultados do experimento com 1 camada, as funções hiperbólicas BHAA e BHSA atingiram um custo consistentemente menor que suas concorrentes, obtento respectivamente as duas melhores marcas. Apenas a PReLU, com a 3º melhor marca, foi capaz de se aproximar dos resultados alcançados pela BHAA e BHSA. Após isso, o 4º melhor resultado foi da SHA-ReLU, nossa alternativa proposta como substituta para a Softplus. Repare que a função Softplus nem mesmo foi incluída no gráfico, por não ter alcançado nosso critério de custo menor que 1 em menos de 30 épocas. Já sua alternativa paramétrica, a P-Softplus, alcançou nosso

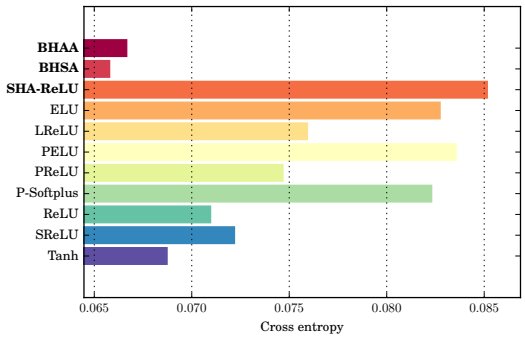
Figura 4.9: Comparação do *cross entropy* das funções de ativação hiperbólicas adaptativas com outras funções adaptativas



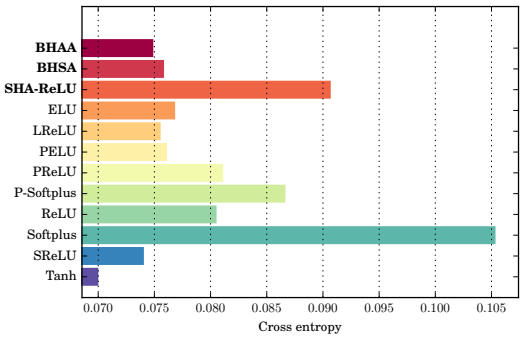
(a) 1 camada oculta



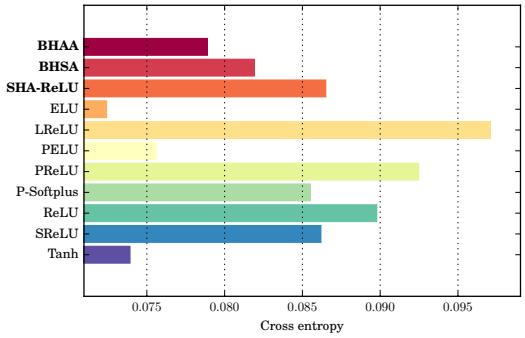
(b) 2 camadas ocultas



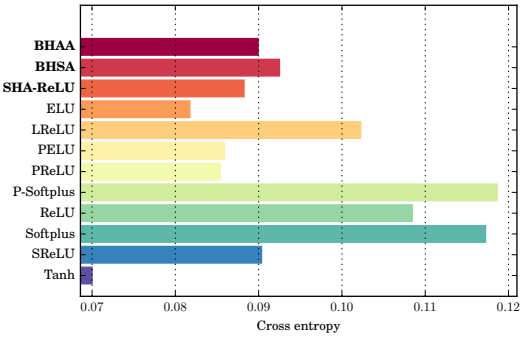
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas

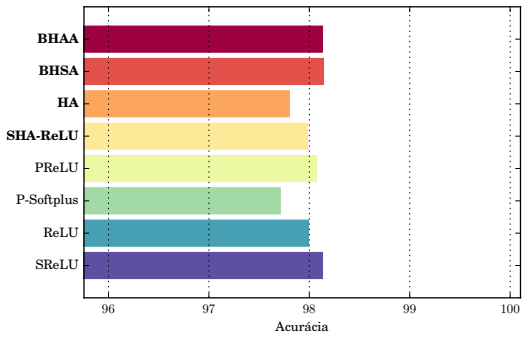


(f) 6 camadas ocultas

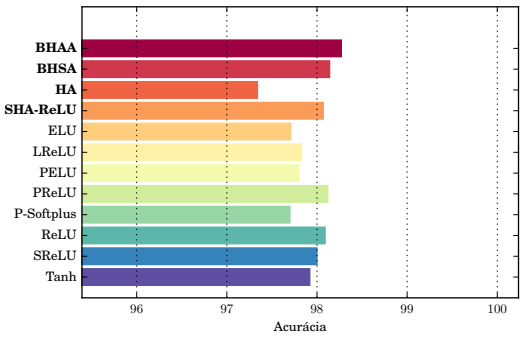
critério, porém com erro relativamente alto. Entretanto, vale notar que apesar do erro maior que da maioria, a P-Softplus alcançou a convergência rapidamente, em apenas 16 épocas. Também destacamos o desempenho de nossa função HA, a qual apesar de atingir o maior erro, foi a mais rápida a atingir, apenas 15 épocas.

No experimento com 2 camadas o cenário se altera, desta vez a BHSA supera a medida de erro da BHAA, sendo ainda as duas melhores marcas. Vale destacar que a BHSA não apenas atingiu o menor erro, como também foi a mais rápida de todas, finalizando o treinamento em somente 12 épocas. Após isso, a BHAA possui o 2º menor tempo de convergência, empatada com 14 épocas com a P-Softplus e SReLU. Desta vez, a SReLU superou a PReLU, as quais alcançaram respectivamente a 3º

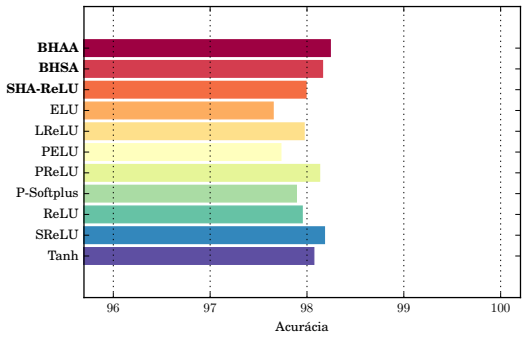
Figura 4.10: Comparação da acurácia das funções de ativação hiperbólicas adaptativas com outras funções adaptativas



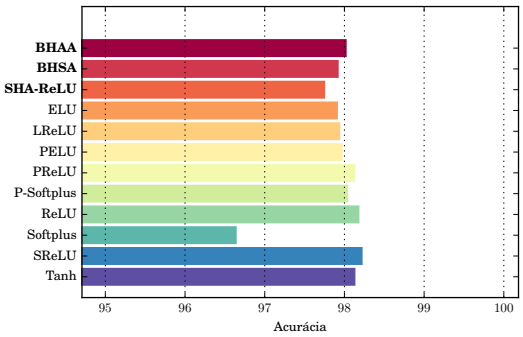
(a) 1 camada oculta



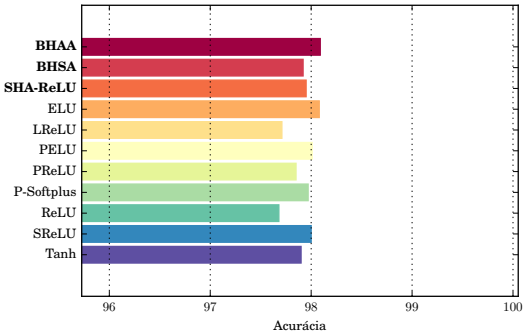
(b) 2 camadas ocultas



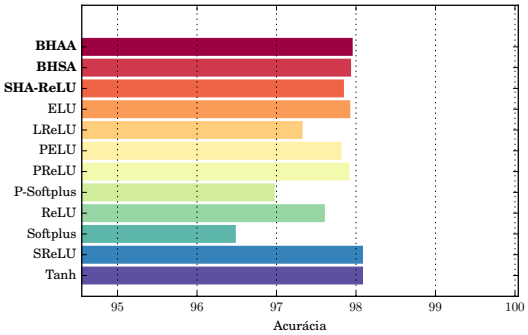
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas

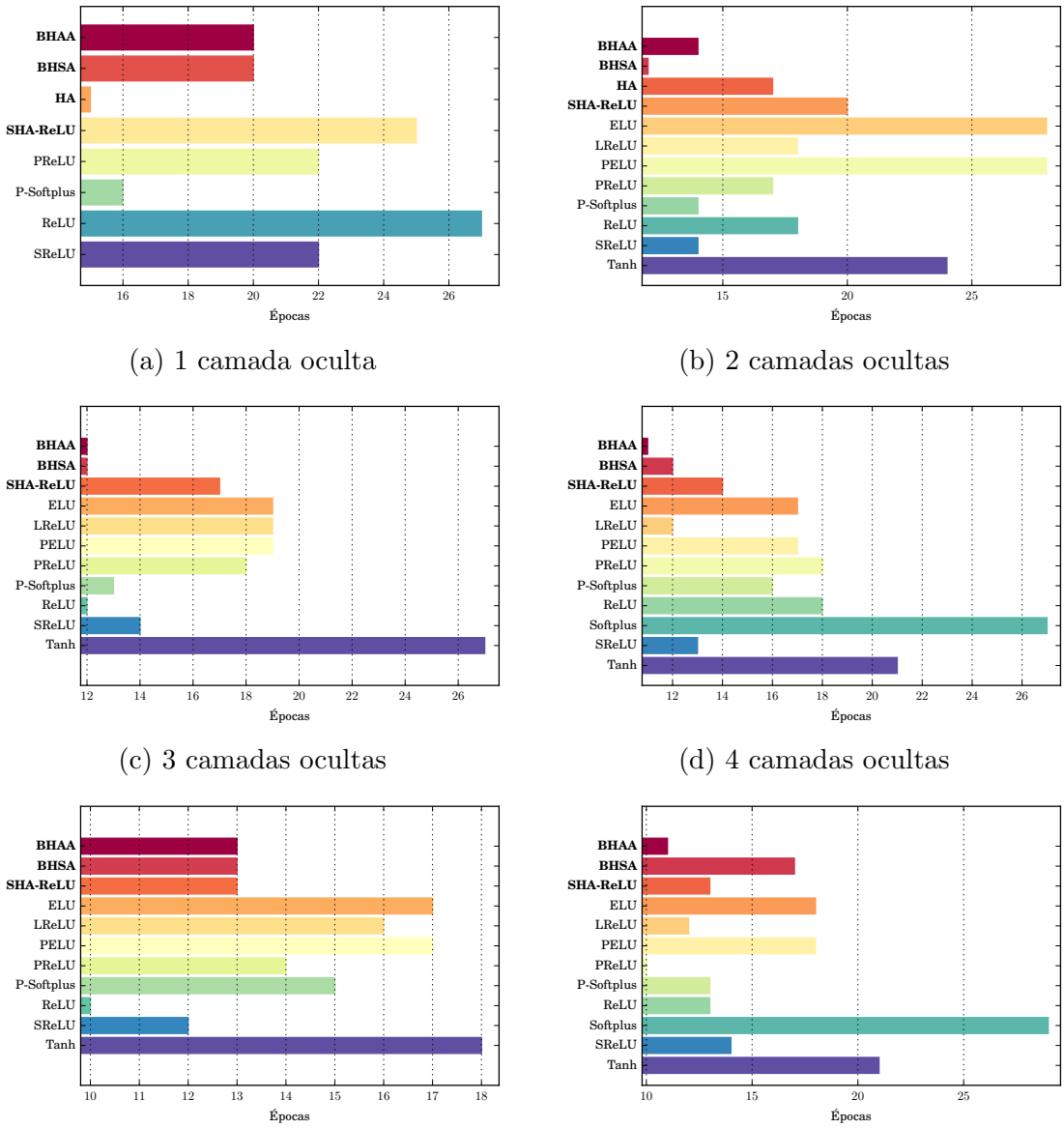


(f) 6 camadas ocultas

e 4º melhor medida de erro. Um destaque importante é que neste experimento, a tangente hiperbólica atingiu uma medida competitiva de erro, embora tenha sido uma das piores em relação ao tempo. Da mesma maneira que com 1 camada, a SHA-ReLU alcançou uma medida de erro menor que a P-Softplus, porém permanece mais lenta. Já a função HA não se mostrou muito competitiva, pois obteve a pior das medidas de erro.

Passando ao experimento com 3 camadas, as funções BHSA e BHAA se mantém respctivamente com as melhores medidas de erro, seguidas da tangente hiperbólica, ReLU e SReLU. Note também que a BHSA, BHAA e a ReLU estão empatadas com o melhor dentre todos os tempos, 12 épocas. Já a tangente hiperbólica se mantém com

Figura 4.11: Comparação do tempo de convergência (em épocas) das funções de ativação hiperbólicas adaptativas com outras funções adaptativas



(a) 1 camada oculta

(b) 2 camadas ocultas

(c) 3 camadas ocultas

(d) 4 camadas ocultas

(e) 5 camadas ocultas

(f) 6 camadas ocultas

uma convergência lenta, neste caso a pior marca. A função P-Softplus permanece com um tempo competitivo, o 2º melhor do experimento. Porém diferentemente do que vimos com duas camadas, a função P-Softplus obteve um erro menor que o da SHA-ReLU, a qual apresentou a pior marca. A função HA, por sua vez, não alcanlou nosso cirtério do convergência, sendo portanto omitida do gráfico. O mesmo ocorre nos experimentos com 4, 5 e 6 camadas.

Podemos notar uma importante diferença com 4 camadas, a função tangente hiperbólica protagoniza o cenário apresentando a menor medida de erro. Apesar do desempenho expoente no quesito erro, a tangente hiperbólica permanece com um tempo de convergência ruim. Consideramos como ruim o tempo de convergência da

tangente hiperbólica por ser o 2º pior nestas circuntâncias, perdendo apenas para a Softplus. Esta última, por sua vez, apesar de possuir o pior tempo e pior medida de erro, pela primeira vez alcançou convergência. A 2º melhor medida de erro é alcançada pela SReLU, que também obteve um desempenho muito competitivo em relação ao tempo de convergência.

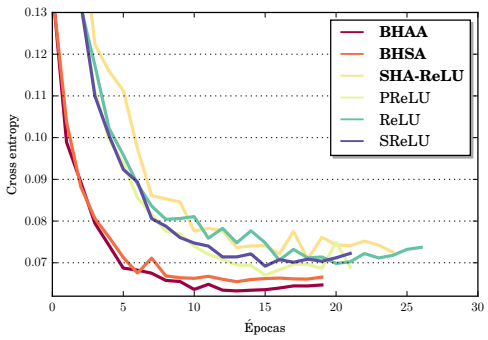
Analisando as funções de nosso interesse, percebemos que a 3º melhor marca foi atingida pela BHAA, seguida pelas funções LReLU, BHSA, PELU e ELU, todas com desempenho muito acirrado. Apesar das funções BHAA e BHSA não terem obtido as melhores marcas de erro do experimento, estas se mostraram muito competitivos, com um importante destaque para o tempo de convergência, que foram definitivamente os menores, 11 e 12 épocas respectivamente. A função SHA-ReLU, por sua vez, obteve um erro menor que sua concorrente Softplus, e por outro lado, maior que o da P-sofplus. Apesar de um erro mais elevado que o da P-Softplus, a SHA-ReLU foi capaz de finalizar o treinamento mais rápido que sua concorrente.

Com 5 camadas as melhores medidas de erro pertencem respectivamente às funções ELU e tangente hiperbólica, seguida pela PELU, e logo após a BHAA e BHSA. Apesar das melhores medidas de erro, as funções ELU e tangente hiperbólica também obtiveram os piores tempos de convergência. Em relação à SHA-ReLU, esta continua com um erro superior ao da P-Softplus, porém agora com uma diferença muito do menor que com 4 camadas, estando também com uma medida muito próxima da SReLU. Em relação ao tempo, a função ReLU se destaca desempenhando o treinamento em apenas 10 épocas, seguido da SReLU com 12 épocas, e logo após empatadas com 13 épocas nossas três funções: BHAA, BHSA e SHA-ReLU.

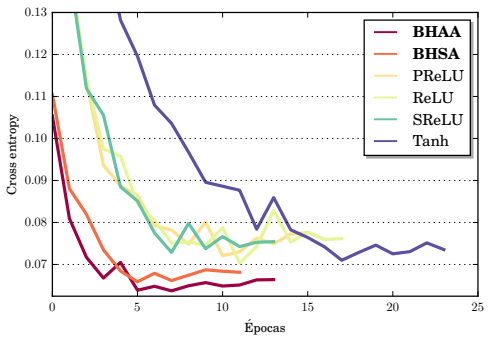
Por fim, com 6 camadas, nosso cenário se torna muito diferente do visto em redes com menor profundidade. A melhor marca de erro pertence à tangente hiperbólica, seguida pelas funções ELU, PReLU e PELU. Dentre estas, destacamos a função PReLU, que não apenas obteve a 3º melhor medida de erro, como também foi a mais rápida, concluindo o treinamento em apenas 10 épocas. Uma mudança importante a ser notada é que a função SHA-ReLU, com a 5º melhor marca, passa a alcançar uma medida de erro melhor que sua concorrente P-Softplus. Além disso, esta agora passa a apresentar também um erro inferior aos das funções BHAA e BHSA, o que não havia ocorrido nos testes com redes mais rasas. Assim, podemos perceber que conforme a arquitetura da rede aprofunda, a BHAA e BHSA pioram seu desempenho, enquanto a SHA-ReLU melhora. Por sua vez, a função BHAA apresenta a o 6º melhor erro, seguido com pequena diferença pela SReLU e BHSA. Além disso, perceba também que a BHAA se manteve com um tempo de convergência muito competitivo, perdendo apenas para a PReLU. Após estas, os melhores tempos são apresentadas pela LReLU, e empatadas as funções SHA-ReLU, P-Softplus e ReLU.

Adicionalmente, a Figura 4.10 mostra que mesmo nos casos em que nossas

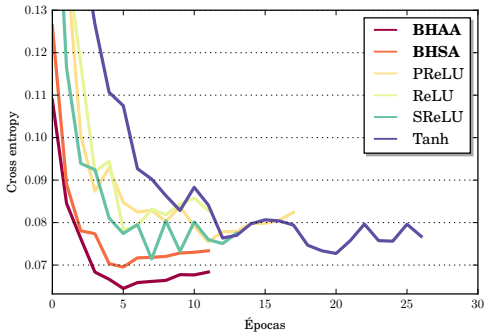
Figura 4.12: Comparação do processo de convergência das funções de ativação hiperbólicas adaptativas com outras funções adaptativas. Análise do erro de validação.



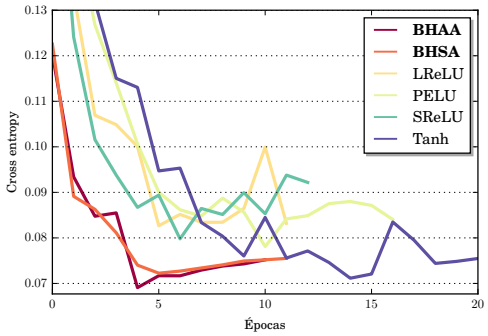
(a) 1 camada oculta



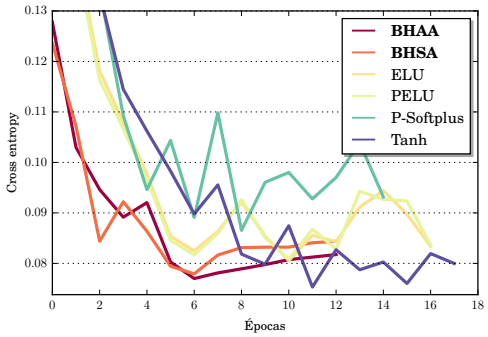
(b) 2 camadas ocultas



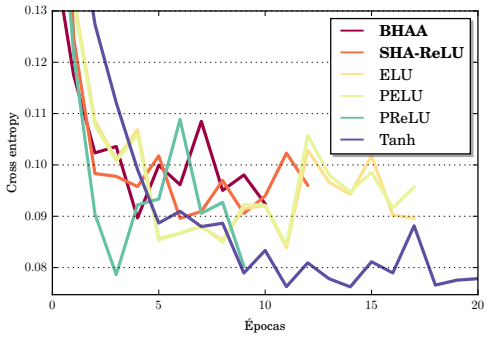
(c) 3 camadas ocultas



(d) 4 camadas ocultas



(e) 5 camadas ocultas



(f) 6 camadas ocultas

funções não apresentaram as melhores medidas de erro, a acurácia alcançou medidas muito próximas às alcançadas pelas melhores funções.

Para melhor compreender o comportamento de nossas funções adaptativas em comparação com as demais, passamos a analisar o decaimento do erro durante o treinamento. Assim, a Figura 4.12 apresenta o *cross entropy* de validação ao longo das épocas nos experimentos de 1 a 6 camadas. A fim viabilizar a compreensão das figuras, serão apresentados apenas os 6 melhores resultados de cada experimento.

Atentando para o decaimento do erro de validação das função BHAA e BHSA, podemos perceber que estas são capazes de reduzirem o erro muito mais rapidamente que suas concorrentes. Note que, apesar destas nem sempre atingirem a menor

medida de erro ao final, elas são capazes de atingir um erro significativamente baixo em condições onde se exige um treinamento expresso. Nos experimentos de 1 a 4 camadas, por exemplo, em apenas 5 épocas as funções foram capazes de reduzir o erro abaixo de $0,07$. Entretanto, os experimentos indicam que esta vantagem se extingue conforme a rede aprofunda, de forma que com 6 camadas já não é possível observá-la.

Um importante questionamento a se fazer a respeito desta abordagem adaptativa das funções hiperbólicas é se, de fato, há alteração significativa dos parâmetros conforme a rede evolui. Assim, a Figura 4.13 aprensenta a evolução dos parâmetros da função Bi-hiperbólica assimétrica adaptativa em uma rede com 4 camadas ocultas. Os gráficos apresentam a diferença entre os valors inicial e final dos parâmetros. A camada 0 trata-se da camada inicial, e as camadas 1 a 5 as camadas ocultas, em ordem de profundidade.

Podemos perceber que de fato a estratégia foi capaz de evoluir os parâmetros, entretanto, em uma escala reduzida. Apesar das variações pequenas em módulo, o ajuste fino dos parâmetros foi suficiente para atingir bons resultados, como visto anteriormente. Nota-se também uma redução progressiva da variação dos parâmetros nas camadas mais profundas.

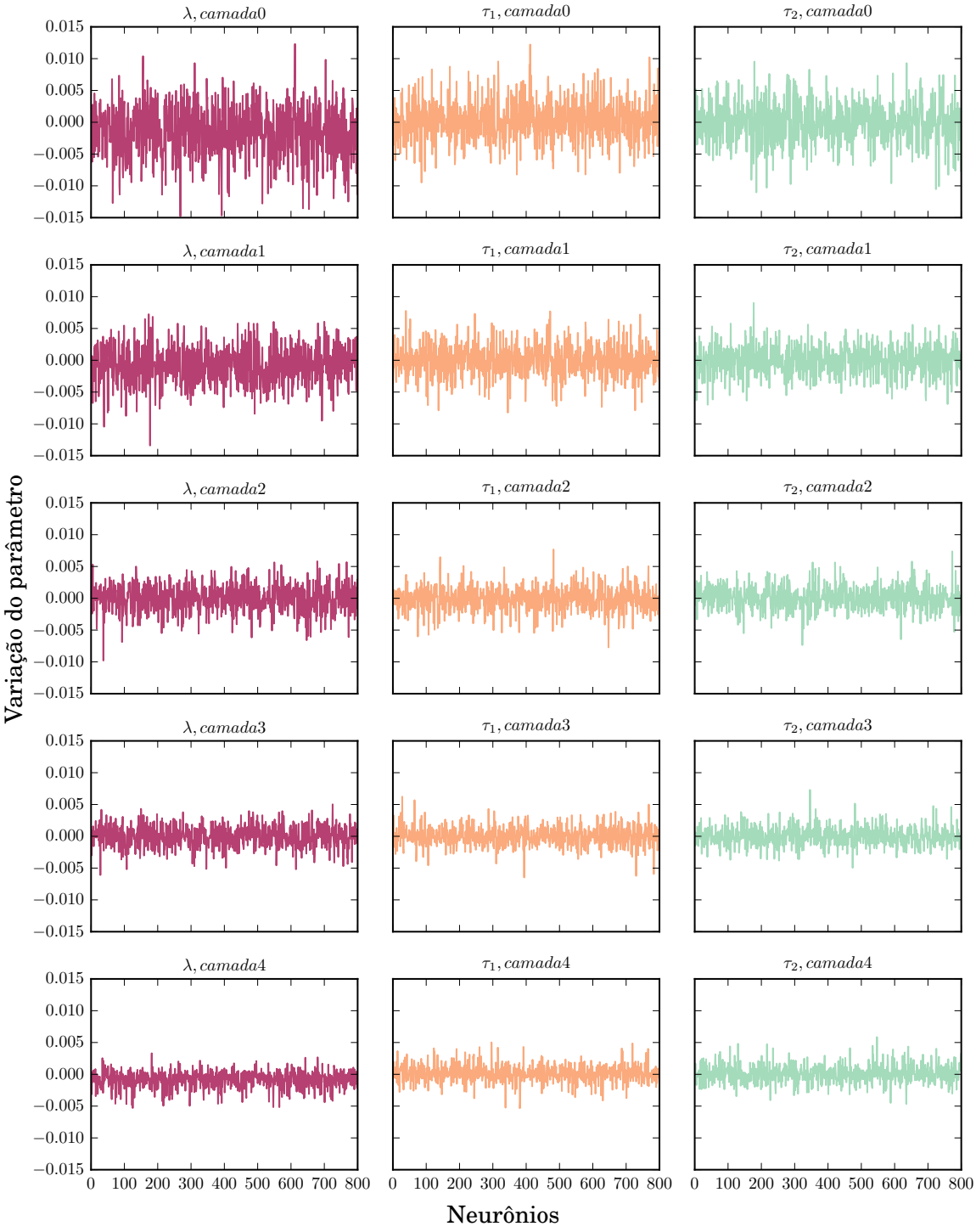
Adicionalmente, apresentamos na Figura 4.14 alguns exemplo de como ficaram as funções de ativação treinadas nesta rede. Da mesma forma que na Figura 4.13, a camada 0 trata-se da camada inicial, e as camadas 1 a 5 as camadas ocultas, em ordem de profundidade.

Diante dos resultados observados, fica evidente que as funções de ativação adaptativas proposta neste trabalho possuem excelente capacidade de convergência, sendo competitivas com muitas das funções adaptativas apresentadas na literatura recente. Podemos constatar que a SHA-ReLU, na maioria dos casos, apresentou uma convergência superior às de suas concorrentes naturais: P-Softplus, Softplus e ReLU. Por outro lado, o desempenho da função HA não foi satisfatório, não havendo alcançado bons resultados em nenhum dos experimentos. Por fim, destacamos o desempenho das funções BHAA e BHSA em redes neurais rasas, devido à sua grande capacidade de redução do erro ainda nas primeiras épocas de treinamento.

4.3.6 Experimento VI

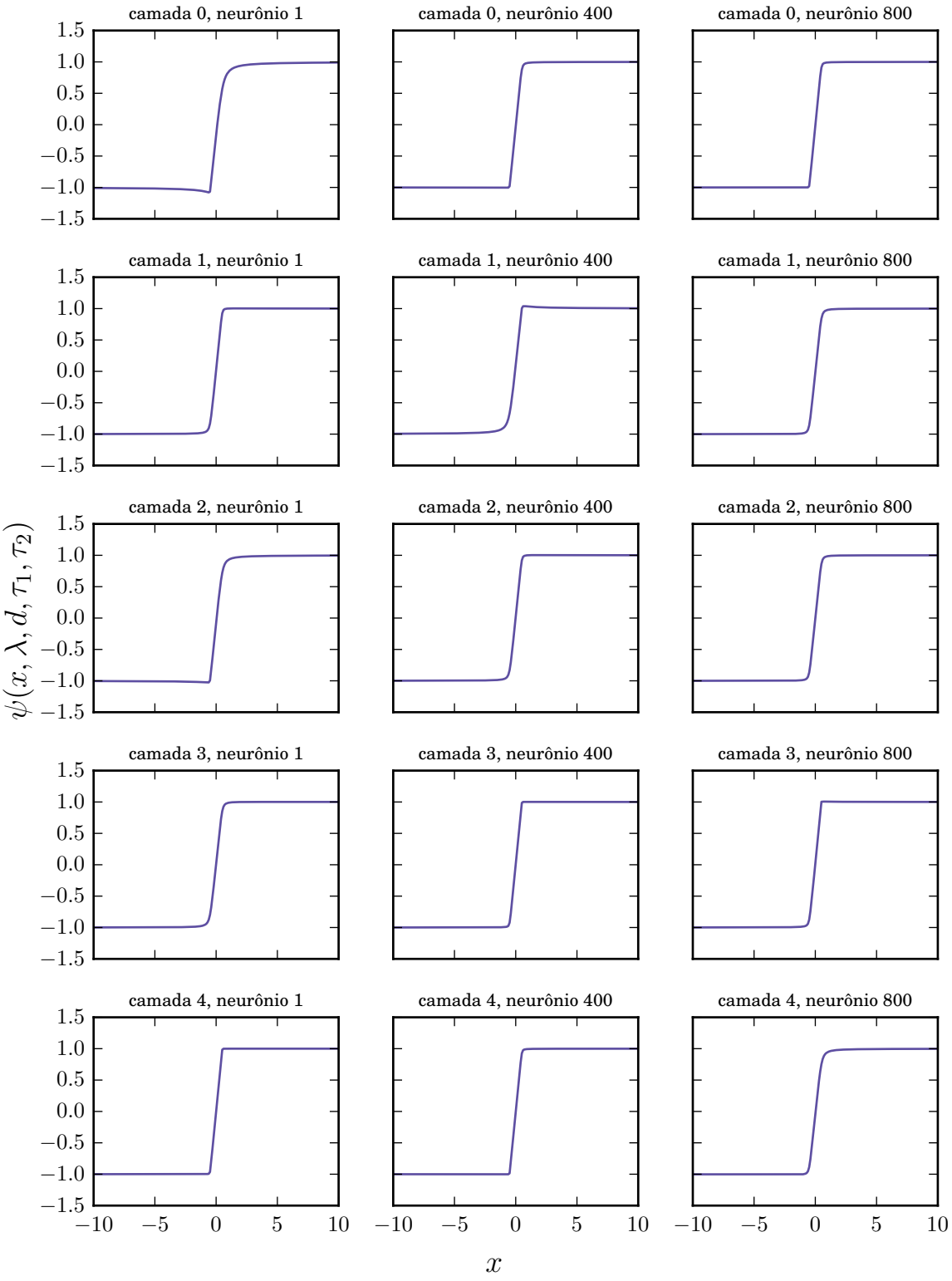
Este experimento trata-se de um complemento do Experimento V, no qual havíamos observado um rápido decréscimo no erro de validação da função de ativação bi-hiperbólica adaptativa ainda nas primeiras épocas de treinamento. Assim, em circunstâncias onde há maiores limitações de tempo para o treinamento, ou ainda em dispositivos com baixo poder computacional, o uso desta função seria favorável.

Figura 4.13: Evolução dos parâmetro da função Bi-hiperbólica assimétrica adaptativa em uma rede com 4 camadas ocultas.



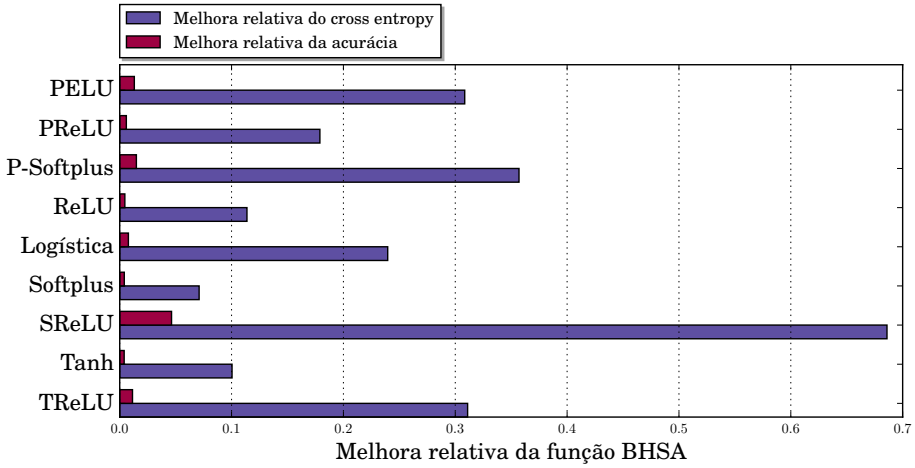
Para tanto, avaliaremos a função de ativação bi-hiperbólica adaptativa, em suas formas simétrica (BHSA) e assimétrica (BHAA), sob uma configuração diferente: será feito um treinamento com apenas 5 épocas (sem parada prematura) em uma rede neural com 4 camadas ocultas. Ao final, avaliaremos o erro e acurácia de teste obtidos, comparando com os de todas as funções vistas no Experimento V.

Figura 4.14: Evolução dos parâmetro da função Bi-hiperbólica assimétrica adaptativa em uma rede com 4 camadas ocultas.

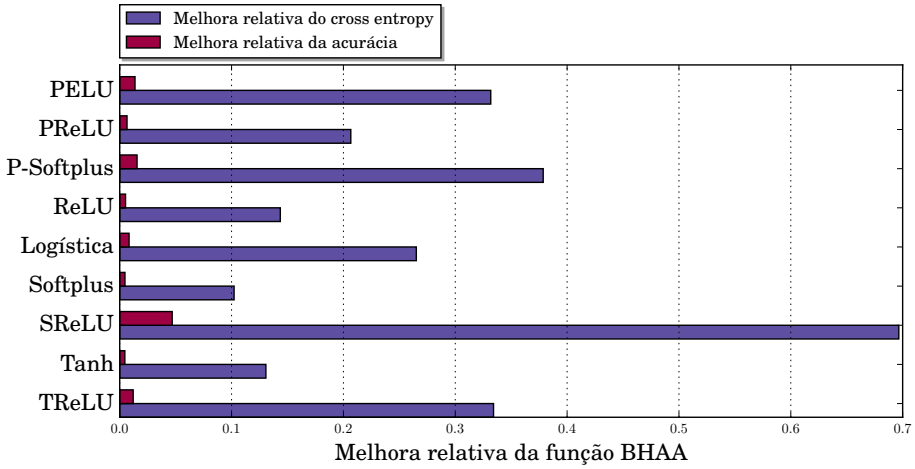


Podemos ver através da Figura 4.15 que a função bi-hiperbólica adaptativa, tanto nas versões simétrica quanto assimétrica, apresentou expressiva melhora no erro sob as criscustâncias enunciadas. Além disso, fica claro também que essa melhora no

Figura 4.15: Melhora relativa do *cross entropy* e acurácia das funções de ativação bi-hiperbólicas adaptativas em uma rede neural com 4 camadas ocultas em treinamento com apenas 5 épocas



(a) Bi-hiperbólica simétrica adaptativa



(b) Bi-hiperbólica assimétrica adaptativa

erro foi suficiente para aprimorar a previsão, visto que houve aumento na acurácia em relação a todas as demais funções de ativação.

Estes resultados corroboram com a argumentação apresentada no Experimento V, evidenciando grande vantagem no uso da função de ativação bi-hiperbólica adaptativa para treinamento de redes neurais com limitação de tempo e/ou poder computacional.

4.3.7 Experimento VII

Um questionamento a ser feito a respeito das funções de ativação propostas neste trabalho é se a formulação da função, por si só, é mais custosa de ser calculada do ponto de vista computacional que outras funções tradicionais. Assim, o presente experimento tem por objetivo avaliar o desempenho das funções de ativação propos-

tas no quesito tempo médio necessário para computar uma época. Foram avaliadas as versões escaladas não-adaptativas de nossas funções de ativação, em comparação com as funções logística, tangente hiperbólica, ReLU e Softplus. Utilizamos uma rede neural com 4 camadas ocultas, com 800 neurônios cada, conforme a arquitetura anteriormente descrita. Apresentamos na Tabela 4.6 o tempo médio gasto por época, em segundos, em um treinamento feito com 20 épocas sem parada prematura.

Tabela 4.6: Tempo médio gasto por época, em segundos.

Função	Tempo médio
Bi-hiperbólica escalada	27,271
Hiperbólica escalada	25,572
Suavização hiperbólica da ReLU	25,464
Logística	22,442
Tangente hiperbólica	22,325
ReLU	22,277
Softplus	22,928

Podemos notar que os tempos médios apresentados pelas funções tradicionais são inferiores às aqui propostas. Entretanto, cabe ressaltar que as funções tradicionais são nativas à biblioteca de redes neurais Keras, sendo implementadas de maneira otimizada. Por outro lado, nossas funções não receberam qualquer tipo de otimização específica em sua implementação.

Assim, a fim de averiguar me maneira mais precisa o tempo computacional necessário para calcular o valor das funções, realizamos um experimento em C onde todas as funções foram implementadas em sua formulação convencional, sem qualquer otimização específica, exceto a otimização convencional (nível 1) realizada pelo compilador GCC. A Tabela 4.7 apresenta o tempo total, em milisegundos, necessário para computar cada uma das funções 10^6 vezes.

Primeiramente, notamos que as funções hiperbólicas alcançaram um tempo significativamente menor que as funções logística e tangente hiperbólica. Por outro lado, a função ReLU, por sua simplicidade, obteve um tempo menor que todas as demais. Com isso, concluímos que não há desvantagem em utilizar as funções hiperbólicas propostas neste trabalho do ponto de vista do tempo gasto para computar a função propriamente dita.

Tabela 4.7: Tempo total gasto para executar 10^6 vezes cada função, em milesegundos

Função	Tempo total
Bi-hiperbólica escalada	324,916
Hiperbólica escalada	327,152
Suavização hiperbólica da ReLU	337,916
Logística	409,289
Tangente hiperbólica	379,313
ReLU	323,818
Softplus	325,741

Capítulo 5

Conclusões

Neste capítulo apresentamos uma conclusão do trabalho, discorrendo de maneira sucinta sobre o que foi abordado, como foi solucionado o problema, resultados obtidos, contribuições, limitações da proposta e direções de pesquisa.

5.1 Considerações sobre o trabalho

Neste trabalho abordamos o problema do gradiente minguante em funções de ativação de redes neurais. Vimos que em muitas das funções tradicionais, como a logística e tangente hiperbólica, o gradiente rapidamente se anula, causando uma estagnação da rede. Por esta causa, o método de penalização hiperbólica foi utilizado por MIGUEZ (2012), THOMAZ e MAIA (2013), XAVIER (2005) para criar duas novas funções de ativação paramétricas onde o efeito de gradiente minguante fosse reduzido, as quais foram denominadas função de ativação hiperbólica e função de ativação bi-hiperbólica.

Entretanto, apesar da vantagem teórica a respeito do gradiente minguante, estas funções exigem uma escolha adequada de parâmetros para apresentar uma boa convergência. Por isso, tratamos do problema da escolha de parâmetros em funções de ativação paramétricas, apresentando uma metodologia proposta recentemente na literatura, que se utiliza do algoritmo de *backpropagation* para ajustar os parâmetros da função de ativação juntamente com os pesos da rede. Desta forma, é possível usufuir das vantagens de uma boa configuração de parâmetros, sem a necessidade de realizar uma custosa busca exaustiva.

Além disso, também vimos que a suavização da função retificadora frequentemente é feita através da função logística, criando uma função de ativação conhecida como Softplus. Entretanto, assim como na logística, a Softplus sofre do problema de gradiente minguante, o que se torna um empecilho para sua capacidade de convergência. Desta forma, fim de contornar este problema, propomos utilizar a técnica

de penalização hiperbólica para criar uma nova suavização da ReLU na qual o problema do gradiente seja mais brando.

5.2 Contribuição

Diante da problemática levantada, nosso trabalho consistiu em modificar o intervalo de atuação das funções hiperbólica e bi-hiperbólcia, de forma que estas passassem apresentar uma melhor qualidade e convergência. Assim, denominamos estas novas funções de hiperbólica escalada e bi-hiperbólica escalada, e as submetemos à experimentação sobre o *dataset* MNIST, avaliando seu comportamento conforme a quantidade de camadas ocultas aumenta.

A avaliação experimental mostrou que a modificação foi capaz de reduzir o *cross-entropy* obtido pelas funções originais em até $97,12\%$ e $10,24\%$, respectivamente. Além disso, as novas funções foram submetidas a testes comparativos utilizando uma busca exaustiva de parâmetros. Os testes mostraram que a função hiperbólica escalada, no melhor caso, apresentou melhora em relação às tradicionais funções logística, tangente hiperbólica e ReLU de $97,44\%$, $17,63\%$, $34,41\%$, respectivamente. A função bi-hiperbólcia escalada, por sua vez, apresentou no melhor caso redução no erro de $97,49\%$, $20,97\%$ e $53,44\%$, no comparativo com as referidas funções.

Além disso, criamos uma alternativa hiperbólica para suavização da função retificadora, a qual possui vantagens teóricas quanto ao fenômeno de gradiente minguante quando comparada à função Softplus. Também avaliamos o uso da referida técnica de aprendizado de parâmetros através do *backpropagation* para criar uma versão adaptativas destas nossas novas funções de ativação hiperbólicas.

Estes testes mostram que a função hiperbólica escalada não obteve desempenho satisfatório utilizando-se da técnica adaptativa, visto que, no melhor caso, alcançou melhoras de $1,06\%$, $-17,68\%$ e $-15,29\%$ no erro em relação às funções logística, tangente hiperbólica e ReLU.

Por outro lado, a função de ativação bi-hiperbólica escalada mostrou resultados muito favoráveis à utilização conjunta com a técnica adaptativa. Na versão simétrica, por exemplo, esta apresentou no melhor caso redução no erro de $96,44\%$, $12,49\%$ e $12,59\%$, em relação às referidas funções logística, tangente hiperbólica e ReLU, sem que fosse necessário o ajuste de um único parâmetro manualmente. De maneira similar, a versão assimétrica apresentou melhoras de $96,58\%$, $11,36\%$ e $17,06\%$ (no melhor caso) em relação às mesmas funções.

Uma característica notável observada no processo de convergência da função bi-hiperbólica adaptativa é sua capacidade de reduzir rapidamente o erro, de maneira que ainda nas primeiras épocas atinge uma marca competitiva com os alcançados

por outras funções adaptativas (e convencionais) apenas em uma quantidade significativamente maior de épocas. Cabe ressaltar que tal característica se apresentou sobretudo em redes neurais de profundidade menor.

Atentando para nossa versão suavizada da ReLU, realizamos uma avaliação de seu desempenho quando unido à técnica adaptativa. A suavização hiperbólica adaptativa da ReLu se mostrou extremamente competitiva, alcançando (no melhor caso) uma redução no erro de $36,67\%$ e $18,62\%$, com relação às funções Softplus e ReLU, respectivamente.

Por fim, foram feitas comparações de nossas funções adaptativas com outras funções adaptativas apresentadas na literatura recente. Estes testes mostraram que não apenas nossas funções adaptativas, com exceção da hiperbólica escalada, foram superiores às tradicionais, mas também se mostraram muito competitivas com as concorrentes mais recentes, de forma a até mesmo superá-las em diversos casos. Destacamos o rápido decréscimo do erro alcançado pela função bi-hiperbólica adaptativa ainda nas primeiras épocas, o que confere a ela vantagem em circunstâncias onde há limitação de tempo ou poder computacional.

5.3 Limitações e trabalhos futuros

Uma limitação evidente de nosso trabalho é o desempenho precário da função de ativação hiperbólica escalada adaptativa, a qual na maioria das vezes nem mesmo chegou a nosso critério de convergência, e quando o fez apresentou erro muito superior aos de suas concorrentes. Uma direção de pesquisa é averiguar como aperfeiçoar esta função para que se torne mais competitiva. Uma possibilidade para este aperfeiçoamento é a forma com a qual os parâmetros são inicializados, questão sobre a qual nossa pesquisa investigou de maneira breve. Outra direção de pesquisa neste tópico é averiguar em quais outras circustâncias, possivelmente, esta função seria competitiva. Além disso, também é possível mesclar diferentes funções hiperbólicas em uma mesma rede.

Apesar do expoente desempenho da função bi-hiperbólica escalada adaptativa nas primeiras épocas em redes neurais rasas, seu desempenho não se perpetua em redes mais profundas. Desta forma, outra importante direção é investigar como conceder a esta função uma boa convergência em redes profundas. Da mesma forma que levantado a respeito da função hiperbólica, uma possível direção é aprimorar a inicialização dos parâmetros.

Também se faz necessária a experimentação da proposta em outros *data sets*, dentre os quais destacamos o ImageNet (DENG *et al.*, 2009), e os *data sets* CIFAR-10 e CIFAR-100 (KRIZHEVSKY, 2009), ou mesmo diferentes domínios, como processamento de texto. Além disso, devem ser feitos testes em MLPs com maior

profundidade, bem como outros tipos de redes neurais, como por exemplo Auto-encoders e Redes Neurais Convolucionais. Outra questão a ser avaliada é o uso de regularização e esparsidade, bem como observar o efeito de utilizar o algoritmo ADADELTA (ZEILER, 2012) para calibragem automática da taxa de aprendizado.

Outra questão a ser investigada é a variação mínima dos parâmetros observada durante o processo de treinamento. Cabe investigar como tornar a evolução dos parâmetros mais rápida, bem como averiguar o impacto disso na velocidade de convergência e capacidade de generalização do modelo.

Referências Bibliográficas

AGOSTINELLI, F., HOFFMAN, M., SADOWSKI, P., et al., 2015, "Learning Activation Functions to Improve Deep Neural Networks", *Proceedings of the 2015 International Conference on Learning Representations (ICLR'15)*, , n. 2013, pp. 1–9. ISSN: 01628828. doi: 10.1007/3-540-49430-8. Disponível em: <http://arxiv.org/abs/1412.6830>.

BALDI, P., SADOWSKI, P., WHITESON, D., 2015, "Enhanced Higgs Boson to $\tau^+\tau^-$ Search with Deep Learning", *Phys. Rev. Lett.*, v. 114 (Mar), pp. 111801. doi: 10.1103/PhysRevLett.114.111801. Disponível em: <http://link.aps.org/doi/10.1103/PhysRevLett.114.111801>.

BENGIO, Y., YOSHUA, 2009, "Learning Deep Architectures for AI", *Foundations and Trends® in Machine Learning*, v. 2, n. 1, pp. 1–127. ISSN: 1935-8237. doi: 10.1561/2200000006. Disponível em: <http://www.nowpublishers.com/article/Details/MAL-006>.

BENGIO, Y., SIMARD, P., FRASCONI, P., 1994, "Learning Long Term Dependencies with Gradient Descent is Difficult", *IEEE Transactions on Neural Networks*, v. 5, n. 2, pp. 157–166. ISSN: 1045-9227. doi: 10.1109/72.279181.

BISHOP, C. M., 1995, *Neural Networks for Pattern Recognition*. New York, NY, USA, Oxford University Press, Inc. ISBN: 0198538642.

BLUM, A. L., RIVEST, R. L., 1992, "Training a 3-node neural network is NP-complete", *Neural Networks*, v. 5, n. 1, pp. 117–127. ISSN: 08936080. doi: 10.1016/S0893-6080(05)80010-3.

BROOMHEAD, D. S., LOWE, D., 1988, "Multivariable Functional Interpolation and Adaptive Networks", *Complex Systems*, v. 2, pp. 321– 355.

CARUANA, R., LAWRENCE, S., GILES, L., 2001, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping", *Advances in neural information processing systems*, pp. 402–408. ISSN: 10495258. doi: 10.1109/IJCNN.2000.857823.

CHOLLET, F., 2015. "keras". https://github.com/fchollet/keras.

CLEVERT, D., UNTERTHINER, T., HOCHREITER, S., 2015, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)", *CoRR*, v. abs/1511.07289. Disponível em: <http://arxiv.org/abs/1511.07289>.

DENG, J., DONG, W., SOCHER, R., et al., 2009, "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*.

DUCH, W., JANKOWSKI, N., 1999, "Survey of neural transfer functions", *Neural Computing Surveys*, v. 2, pp. 163–212. Disponível em: <ftp://ftp.icsi.berkeley.edu/pub/ai/jagota/vol2{_}6.pdf>.

DUGAS, C., BENGIO, Y., 2001, "Incorporating Second-Order Functional Knowledge for Better Option Pricing", .

GLOROT, X., BENGIO, Y., 2010, "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, v. 9, pp. 249–256. ISSN: 15324435. doi: 10.1.1.207.2059. Disponível em: <http://machinelearning.wustl.edu/mlpapers/paper{_}files/AISTATS2010{_}GlorotB10.pdf>.

GLOROT, X., BORDES, A., BENGIO, Y., 2011, "Deep sparse rectifier neural networks", *AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, v. 15, pp. 315–323. ISSN: 15324435. doi: 10.1.1.208.6449.

GOLIK, P., DOETSCH, P., NEY, H., 2013, "Cross-entropy vs. Squared error training: A theoretical and experimental comparison", *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, v. 2, n. 2, pp. 1756–1760. ISSN: 19909772. doi: 10.1145/1102351.1102422.

GOODFELLOW, I., BENGIO, Y., COURVILLE, A., 2016, *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

HE, K., ZHANG, X., REN, S., et al., 2015, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", *CoRR*, v. abs/1502.0. ISSN: 15505499. doi: 10.1109/ICCV.2015.123. Disponível em: <http://arxiv.org/abs/1502.01852>.

HECHT-NIELSEN, R., 1989, "Theory of the Backpropagation Neural Network", *Proceedings Of The International Joint Conference On Neural Networks*, v. 1, pp. 593–605. ISSN: 08936080. doi: 10.1109/IJCNN.1989.118638. Disponível em: <http://ieeexplore.ieee.org/xpl/freeabs{_}all.jsp?{&}arnumber=118638>.

HINTON, G., DENG, L., YU, D., et al., 2012, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", *IEEE Signal Processing Magazine*, v. 29, n. 6 (11), pp. 82–97. ISSN: 1053-5888. doi: 10.1109/MSP.2012.2205597. Disponível em: <http://ieeexplore.ieee.org/document/6296526/>.

HINTON, G. E., 2007, "Learning multiple layers of representation", *Trends in Cognitive Sciences*, v. 11, n. 10 (10), pp. 428–434. ISSN: 13646613. doi: 10.1016/j.tics.2007.09.004. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1364661307002173>.

HORNIK, K., STINCHCOMBE, M., WHITE, H., 1989, "Multilayer feedforward networks are universal approximators", *Neural Networks*, v. 2, n. 5, pp. 359–366. ISSN: 08936080. doi: 10.1016/0893-6080(89)90020-8.

HSU, F.-H., 2002, *Behind Deep Blue: Building the Computer That Defeated the World Chess Champion*. Princeton, NJ, USA, Princeton University Press. ISBN: 0691090653.

HUNTER, J. D., 2007, "Matplotlib: A 2D graphics environment", *Computing in Science and Engineering*, v. 9, n. 3, pp. 99–104. ISSN: 15219615. doi: 10.1109/MCSE.2007.55.

ILYA SUTSKEVER JAMES MARTENS, G. E. D. G. E. H., 2013, "On the importance of initialization and momentum in deep learning." *International Conference on Machine Learning*, pp. 1139–1147.

JIN, X., XU, C., FENG, J., et al., 2015, "Deep Learning with S-shaped Rectified Linear Activation Units", .

JONES, E., OLIPHANT, T., PETERSON, P., et al., 2001–. "SciPy: Open source scientific tools for Python". Disponível em: <http://www.scipy.org/>. [Online; accessed 2017-02-14].

JORDAN, M., 1995, "Why the logistic function? A tutorial discussion on probabilities and neural networks", *Computational Cognitive Science Technical Report 9503*, pp. 1–13. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.379.718{&}rep=rep1{&}type=pdf>.

KARLIK, B., OLGAC, A., 2010, "Performance analysis of various activation functions in generalized MLP architectures of neural networks", *International Journal of Artificial Intelligence and Expert Systems (IJAE)*, v. 1, n. 4, pp. 111–122. ISSN: 2180124X. Disponível em: <http://www.cscjournals.org/csc/manuscript/Journals/IJAE/volume1/Issue4/IJAE-26.pdf>.

KONDA, K., MEMISEVIC, R., KRUEGER, D., 2015, "Zero-bias autoencoders and the benefits of co-adapting features", *ICLR*, , n. 2011, pp. 1–11.

KRIZHEVSKY, A., 2009, *Learning multiple layers of features from tiny images*. Relatório técnico.

LECUN, Y., CORTES, C., 2010, "MNIST handwritten digit database", Disponível em: <http://yann.lecun.com/exdb/mnist/>.

LECUN, Y. A., BOTTOU, L., ORR, G. B., et al., 2012, "Efficient backprop", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 7700 LECTU, pp. 9–48. ISSN: 03029743. doi: 10.1007/978-3-642-35289-8-3.

MAAS, A. L., HANNUN, A. Y., NG, A. Y., 2013, "Rectifier Nonlinearities Improve Neural Network Acoustic Models", *Proceedings of the 30 th International Conference on Machine Learning*, v. 28, pp. 6. Disponível em: <https://web.stanford.edu/{~}awni/papers/relu{_}hybrid{_}icml2013{_}final.pdf>.

MALMGREN, H., 2000, "Artificial Neural Networks in Medicine and Biology", v. 1, n. 21, pp. 1–21. doi: 10.1007/978-1-4471-0513-8. Disponível em: <http://www.springerlink.com/index/10.1007/978-1-4471-0513-8>.

MCCULLOCH, W. S., PITTS, W., 1943, "A logical calculus of the ideas immanent in nervous activity", *The Bulletin of Mathematical Biophysics*, v. 5, n. 4, pp. 115–133. ISSN: 00074985. doi: 10.1007/BF02478259.

MCFARLAND, J. M., CUI, Y., BUTTS, D. A., 2013, "Inferring Nonlinear Neuronal Computation Based on Physiologically Plausible Inputs", *PLoS Computational Biology*, v. 9, n. 7. ISSN: 1553734X. doi: 10.1371/journal.pcbi.1003143.

MIGUEZ, G. A., 2012, *Otimização do Algoritmo de Backpropagation Pelo Uso da Função de Ativação Bi-Hiperbólica*. Tese de Doutorado, Universidade Federal do Rio de Janeiro.

MIKOLOV, T., KOMBRINK, S., 2011, "Extensions of recurrent neural network language model", *Icassp*, pp. 5528–5531. ISSN: 1520-6149. doi: 10.1109/ICASSP.2011.5947611. Disponível em: <http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5947611>.

MINSKY, M., PAPERT, S., 1969, "1969), Perceptrons", *Cambridge, MA: MIT Press*, v. 18, pp. 19.

NAIR, V., HINTON, G. E., 2010, "Rectified Linear Units Improve Restricted Boltzmann Machines", *Proceedings of the 27th International Conference on Machine Learning*, , n. 3, pp. 807–814. doi: 10.1.1.165.6419.

NESTEROV, Y., 1983, "A method of solving a convex programming problem with convergence rate O(1/sqr(k))", *Soviet Mathematics Doklady*, v. 27, pp. 372–376. Disponível em: <http://www.core.ucl.ac.be/{~}nesterov/Research/Papers/DAN83.pdf>.

PRECHELT, L., 1998, "Automatic early stopping using cross validation: qualifying the criteria", *Neural Networks*, v. 11, pp. 8.

SCARDAPANE, S., SCARPINITI, M., COMMINIELLO, D., et al., 2016, "Learning activation functions from data using cubic spline interpolation", *Nips*, pp. 1–12. Disponível em: <http://arxiv.org/abs/1605.05509>.

SIMARD, P. Y., STEINKRAUS, D., PLATT, J. C., 2003, "Best practices for convolutional neural networks applied to visual document analysis", *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pp. 958–963. doi: 10.1109/ICDAR.2003.1227801.

THOMAZ, L. A., MAIA, L. S., 2013, "Métodos de implementação de redes neurais feedforward para simulação da percepção auditiva humana", v. 226.

TROTTIER, L., CHAIB-DRAA, B., ENGINEERING, S., 2016, "Parametric Exponential Linear Unit for Deep Convolutional Neural Networks", pp. 1–16.

XAVIER, A. E., 1982, *Penalização Hiperbólica: Um Novo Método para Resolução de Problemas de Otimização*. Tese de Mestrado, Universidade Federal do Rio de Janeiro.

XAVIER, A. E., 2005, "Uma Função Ativação para Redes Neurais Artificiais Mais Flexível e Poderosa e Mais Rápida", *Brazilian Neural Networks Society Journal*, v. I, pp. 276–282.

XAVIER, A. E., 2010, "The hyperbolic smoothing clustering method", *Pattern Recognition*, v. 43, n. 3 (3), pp. 731–737. ISSN: 00313203. doi: 10.1016/j.patcog.2009.06.018. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0031320309002611>.

XAVIER, A. E., XAVIER, V. L., 2014, "Flying elephants: a general method for solving non-differentiable problems", *Journal of Heuristics*. ISSN: 15729397. doi: 10.1007/s10732-014-9268-8.

YAO, X., 1999, "Evolving artificial neural networks", *Proceedings of the IEEE*, v. 87, n. 9, pp. 1423–1447. ISSN: 00189219. doi: 10.1109/5.784219.

ZEILER, M. D., 2012, "ADADELTA: An Adaptive Learning Rate Method", *CoRR*, v. abs/1212.5701. Disponível em: <http://arxiv.org/abs/1212.5701>.

Apêndice A

Tabelas complementares

A.1 Tabelas do Experimento II

Tabela A.1: Comparação de desempenho da função bi-hiperbólica e bi-hiperbólica escalada. A coluna melhora relativa trata-se do *cross entropy* da vesão escalada em relação à original.

λ	τ	*cross entropy*		acurácia		épocas		melhora relativa
		original	escalada	original	escalada	original	escalada	
0,5	0,5	0,082594	0,115352	97,45%	96,44%	136	212	-39,66%
0,5	1,0	0,062182	0,087063	98,12%	97,45%	35	84	-40,01%
0,5	1,5	0,063320	0,067607	98,08%	97,88%	17	33	-6,77%
0,5	2,0	0,080290	0,079084	97,59%	97,74%	159	85	1,50%
0,5	2,5	0,063672	0,064208	98,13%	98,02%	47	14	-0,84%
0,5	3,0	0,058907	0,098310	98,08%	97,27%	28	128	-66,89%
0,5	3,5	0,104921	0,103976	96,79%	97,20%	210	347	0,90%
0,5	4,0	0,061997	0,058617	98,04%	98,23%	42	26	5,45%
0,5	4,5	0,069835	0,108142	97,90%	96,81%	85	162	-54,85%
0,5	5,0	0,064296	0,059630	97,98%	98,23%	46	31	7,26%
0,5	5,5	0,063164	0,067385	98,14%	98,00%	46	42	-6,68%
1,0	0,5	0,070995	0,073565	97,80%	97,76%	94	36	-3,62%
1,0	1,0	0,085871	0,084942	97,49%	97,53%	125	60	1,08%
1,0	1,5	0,082105	0,068582	97,46%	97,99%	87	43	16,47%
1,0	2,0	0,061283	0,090768	98,06%	97,46%	35	118	-48,11%
1,0	2,5	0,071696	0,064735	97,81%	98,06%	105	35	9,71%
1,0	3,0	0,248597	0,067481	92,80%	98,11%	103	49	72,86%
1,0	3,5	0,062591	0,069863	98,13%	98,04%	33	50	-11,62%
1,0	4,0	0,062462	0,093128	98,15%	97,33%	24	162	-49,10%
1,0	4,5	0,083923	0,062120	97,43%	98,18%	167	39	25,98%
1,0	5,0	0,064959	0,063240	97,98%	98,09%	20	18	2,65%
1,0	5,5	0,060305	0,078298	98,14%	97,70%	30	58	-29,84%
1,5	0,5	0,068173	0,066079	97,91%	98,05%	17	13	3,07%
1,5	1,0	0,072848	0,069964	97,78%	97,93%	15	48	3,96%
1,5	1,5	0,221009	0,077746	93,47%	97,74%	147	56	64,82%
1,5	2,0	0,062474	0,077325	98,16%	97,78%	42	70	-23,77%
1,5	2,5	0,065818	0,059167	98,10%	98,23%	66	19	10,11%
1,5	3,0	0,063068	0,066136	98,10%	98,00%	49	15	-4,86%

Continua na próxima página

λ	τ	*cross entropy*		acurácia		épocas		melhora relativa
		original	escalada	original	escalada	original	escalada	
1,5	3,5	0,063399	0,059873	98,01%	98,26%	42	37	5,56%
1,5	4,0	0,060953	0,062319	98,06%	98,12%	33	25	-2,24%
1,5	4,5	0,062907	0,075066	98,07%	97,83%	54	51	-19,33%
1,5	5,0	0,068736	0,070789	97,84%	97,86%	67	41	-2,99%
1,5	5,5	0,072312	0,066686	97,88%	97,98%	116	32	7,78%
2,0	0,5	0,106782	0,074371	96,72%	97,76%	253	42	30,35%
2,0	1,0	0,099152	0,102484	97,00%	97,15%	187	144	-3,36%
2,0	1,5	0,064144	0,061578	98,10%	98,14%	58	33	4,00%
2,0	2,0	0,058544	0,061635	98,14%	98,20%	41	34	-5,28%
2,0	2,5	0,059701	0,110607	98,07%	96,85%	23	254	-85,27%
2,0	3,0	0,071577	0,063162	97,81%	97,97%	75	21	11,76%
2,0	3,5	0,272896	0,061481	92,08%	98,18%	125	26	77,47%
2,0	4,0	0,066973	0,075435	97,93%	97,95%	70	17	-12,63%
2,0	4,5	0,060753	0,095199	98,09%	97,52%	33	136	-56,70%
2,0	5,0	0,085273	0,265768	97,34%	92,61%	120	203	-211,67%
2,0	5,5	0,065209	0,067646	98,13%	98,04%	73	16	-3,74%
2,5	0,5	0,070482	0,064076	97,87%	98,03%	90	18	9,09%
2,5	1,0	0,068315	0,059583	97,96%	98,14%	81	24	12,78%
2,5	1,5	0,072351	0,062474	97,74%	98,05%	63	36	13,65%
2,5	2,0	0,067955	0,060391	97,92%	98,12%	77	23	11,13%
2,5	2,5	0,095583	0,113216	97,24%	96,91%	191	476	-18,45%
2,5	3,0	0,066295	0,060178	98,00%	98,18%	42	31	9,23%
2,5	3,5	0,199762	0,062192	94,19%	98,15%	291	19	68,87%
2,5	4,0	0,068181	0,084249	97,97%	97,48%	84	72	-23,57%
2,5	4,5	0,069412	0,088468	97,87%	97,50%	92	83	-27,45%
2,5	5,0	0,064236	0,071553	98,01%	97,96%	20	13	-11,39%
2,5	5,5	0,065304	0,072528	98,04%	97,93%	19	14	-11,06%
3,0	0,5	0,321933	0,072277	91,01%	97,89%	139	13	77,55%
3,0	1,0	0,075703	0,088199	97,69%	97,47%	143	79	-16,51%
3,0	1,5	0,101377	0,092266	96,94%	97,36%	132	100	8,99%
3,0	2,0	0,064806	0,066817	98,00%	97,98%	61	35	-3,10%
3,0	2,5	0,083144	0,063781	97,46%	98,10%	130	38	23,29%
3,0	3,0	0,066989	0,080306	98,04%	97,70%	72	81	-19,88%
3,0	3,5	0,062174	0,071387	98,09%	97,87%	26	33	-14,82%
3,0	4,0	0,070502	0,137031	97,82%	95,97%	80	436	-94,36%
3,0	4,5	0,174835	0,059013	94,92%	98,21%	228	26	66,25%
3,0	5,0	0,062229	0,062175	97,97%	98,13%	41	28	0,09%
3,0	5,5	0,081008	0,077066	97,51%	97,77%	66	57	4,87%
3,5	0,5	0,063102	0,074848	97,99%	97,76%	42	61	-18,61%
3,5	1,0	0,060081	0,064330	98,12%	98,09%	33	37	-7,07%
3,5	1,5	0,111850	0,063135	96,60%	98,00%	136	27	43,55%
3,5	2,0	0,060903	0,059982	98,21%	98,15%	27	22	1,51%
3,5	2,5	0,076739	0,064356	97,66%	98,18%	127	42	16,14%
3,5	3,0	0,094389	0,061699	97,02%	98,12%	201	31	34,63%
3,5	3,5	0,062019	0,095205	97,96%	97,33%	24	134	-53,51%
3,5	4,0	0,065671	0,059816	97,91%	98,20%	40	23	8,92%
3,5	4,5	0,090777	0,070282	97,23%	97,94%	161	49	22,58%
3,5	5,0	0,071635	0,091775	97,79%	97,42%	99	142	-28,11%
3,5	5,5	0,062908	0,093797	98,16%	97,40%	68	92	-49,10%

Continua na próxima página

λ	τ	*cross entropy*		acurácia		épocas		melhora relativa
		original	escalada	original	escalada	original	escalada	
4,0	0,5	0,114010	0,078335	96,61%	97,62%	212	53	31,29%
4,0	1,0	0,081300	0,266550	97,41%	92,53%	118	155	-227,86%
4,0	1,5	0,067854	0,068789	97,86%	97,98%	65	14	-1,38%
4,0	2,0	0,059684	0,067608	98,09%	98,01%	32	14	-13,28%
4,0	2,5	0,282000	0,060150	91,93%	98,26%	175	28	78,67%
4,0	3,0	0,074166	0,096844	97,72%	97,31%	55	189	-30,58%
4,0	3,5	0,059344	0,089018	98,12%	97,44%	33	88	-50,00%
4,0	4,0	0,070639	0,111548	97,84%	96,72%	17	219	-57,91%
4,0	4,5	0,064918	0,067983	97,99%	98,02%	44	14	-4,72%
4,0	5,0	0,063741	0,072139	98,05%	97,91%	22	15	-13,18%
4,0	5,5	0,064281	0,058713	97,90%	98,18%	52	24	8,66%
4,5	0,5	0,070404	0,061170	97,77%	98,11%	46	27	13,12%
4,5	1,0	0,083206	0,114695	97,40%	96,89%	72	392	-37,84%
4,5	1,5	0,059752	0,078857	98,23%	97,67%	42	65	-31,97%
4,5	2,0	0,068552	0,063377	97,98%	98,08%	51	27	7,55%
4,5	2,5	0,082754	0,074433	97,46%	97,75%	109	56	10,06%
4,5	3,0	0,063410	0,088480	98,08%	97,59%	39	109	-39,54%
4,5	3,5	0,080725	0,083377	97,54%	97,68%	93	51	-3,29%
4,5	4,0	0,068207	0,064763	97,93%	98,04%	16	33	5,05%
4,5	4,5	0,070183	0,068622	97,86%	98,00%	70	50	2,22%
4,5	5,0	0,098101	0,066515	97,00%	98,00%	158	37	32,20%
4,5	5,5	0,076700	0,073957	97,63%	97,86%	111	59	3,58%
5,0	0,5	0,088237	0,067433	97,34%	97,98%	113	43	23,58%
5,0	1,0	0,060454	0,062778	98,10%	98,15%	28	32	-3,84%
5,0	1,5	0,077070	0,068065	97,65%	98,01%	112	43	11,68%
5,0	2,0	0,065817	0,072748	98,06%	97,92%	70	50	-10,53%
5,0	2,5	0,130026	0,082818	96,29%	97,54%	205	56	36,31%
5,0	3,0	0,074170	0,071889	97,78%	97,91%	112	15	3,08%
5,0	3,5	0,072560	0,060004	97,74%	98,19%	81	19	17,30%
5,0	4,0	0,103223	0,060150	96,96%	98,28%	383	26	41,73%
5,0	4,5	0,066217	0,091052	97,97%	97,42%	65	83	-37,51%
5,0	5,0	0,068133	0,066043	97,86%	98,08%	84	43	3,07%
5,0	5,5	0,344306	0,060349	90,10%	98,19%	138	32	82,47%
5,5	0,5	0,067671	0,059612	97,87%	98,17%	62	15	11,91%
5,5	1,0	0,066345	0,080918	97,95%	97,84%	77	82	-21,97%
5,5	1,5	0,275261	0,065106	92,00%	98,01%	238	31	76,35%
5,5	2,0	0,071804	0,070450	97,92%	97,86%	14	42	1,89%
5,5	2,5	0,069830	0,072302	97,86%	97,81%	100	56	-3,54%
5,5	3,0	0,067871	0,060111	97,80%	98,18%	53	22	11,43%
5,5	3,5	0,082967	0,058921	97,53%	98,22%	159	24	28,98%
5,5	4,0	0,095893	0,103085	97,08%	97,22%	243	214	-7,50%
5,5	4,5	0,059570	0,074556	98,16%	97,85%	29	12	-25,16%
5,5	5,0	0,068527	0,063953	97,81%	98,10%	78	18	6,67%
5,5	5,5	0,072614	0,079628	97,76%	97,72%	97	65	-9,66%

A.2 Tabelas do Experimento III

Tabela A.2: Melhora relativa do *cross entropy* utilizando a função de ativação hiperbólica escalada de acordo com ρ, em relação à ReLU, tangente hiperbólica e logística, variando a quantidade de camadas ocultas. Tabela completa ordenada do melhor para o pior resultado.

ρ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
				relu	tanh	sigmoid
1 camada oculta						
0,4	0,059713	98,10%	23	0,66%	9,14%	34,29%
0,5	0,060198	98,20%	33	-0,14%	8,40%	33,75%
0,3	0,061176	98,12%	21	-1,77%	6,91%	32,68%
0,8	0,061335	98,07%	36	-2,03%	6,67%	32,50%
0,7	0,061609	98,04%	31	-2,49%	6,25%	32,20%
0,2	0,061920	98,13%	19	-3,01%	5,78%	31,86%
0,6	0,063732	98,07%	32	-6,02%	3,02%	29,87%
1,1	0,066266	97,91%	40	-10,24%	-0,83%	27,08%
0,9	0,066382	97,96%	39	-10,43%	-1,01%	26,95%
1,2	0,066997	97,97%	36	-11,45%	-1,94%	26,27%
1,0	0,067818	98,08%	36	-12,82%	-3,19%	25,37%
0,1	0,068480	97,98%	18	-13,92%	-4,20%	24,64%
1,4	0,069401	97,96%	45	-15,45%	-5,60%	23,63%
1,3	0,070965	97,88%	49	-18,05%	-7,98%	21,91%
1,5	0,071906	97,77%	45	-19,62%	-9,41%	20,87%
2 camadas ocultas						
0,3	0,057868	98,21%	16	16,86%	10,72%	45,97%
0,5	0,060603	98,20%	20	12,93%	6,50%	43,42%
0,6	0,061377	98,18%	26	11,82%	5,31%	42,70%
0,4	0,062668	98,11%	17	9,96%	3,31%	41,49%
0,9	0,063628	98,18%	31	8,58%	1,83%	40,59%
0,8	0,064144	98,03%	27	7,84%	1,04%	40,11%
0,2	0,064704	98,21%	14	7,04%	0,17%	39,59%
0,7	0,065160	98,03%	23	6,38%	-0,53%	39,16%
1,1	0,066633	97,99%	31	4,27%	-2,80%	37,79%
1,0	0,066918	98,28%	38	3,86%	-3,24%	37,52%
0,1	0,067102	98,06%	11	3,59%	-3,53%	37,35%
1,4	0,069965	97,95%	30	-0,52%	-7,94%	34,68%
1,3	0,070586	97,91%	35	-1,41%	-8,90%	34,10%
1,5	0,072581	97,83%	36	-4,28%	-11,98%	32,24%
1,2	0,073257	98,03%	33	-5,25%	-13,02%	31,60%
3 camadas ocultas						
0,6	0,060064	98,27%	22	21,66%	11,68%	53,14%
0,3	0,060885	98,26%	14	20,59%	10,48%	52,50%
0,4	0,061910	98,14%	16	19,25%	8,97%	51,70%
0,5	0,063399	98,17%	17	17,31%	6,78%	50,54%
0,8	0,063404	98,28%	26	17,30%	6,77%	50,54%
0,2	0,065661	98,10%	12	14,36%	3,45%	48,77%
0,7	0,066215	98,17%	26	13,64%	2,64%	48,34%
0,9	0,067700	97,97%	23	11,70%	0,45%	47,18%
1,0	0,070347	97,99%	25	8,25%	-3,44%	45,12%
1,1	0,071644	97,94%	26	6,56%	-5,34%	44,11%

Continua na próxima página

ρ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
				relu	tanh	sigmoid
1,2	0,072294	98,08%	32	5,71%	-6,30%	43,60%
1,4	0,072621	98,02%	31	5,28%	-6,78%	43,34%
1,3	0,074574	97,82%	23	2,73%	-9,65%	41,82%
0,1	0,074869	97,90%	13	2,35%	-10,09%	41,59%
1,5	0,075205	97,86%	32	1,91%	-10,58%	41,33%
4 camadas ocultas						
0,3	0,060281	98,29%	13	23,29%	15,39%	58,17%
0,5	0,060932	98,36%	15	22,46%	14,47%	57,72%
0,6	0,062752	98,27%	20	20,15%	11,92%	56,46%
0,8	0,063195	98,38%	22	19,58%	11,30%	56,15%
0,4	0,063389	98,32%	15	19,34%	11,02%	56,02%
0,7	0,063517	98,28%	20	19,17%	10,84%	55,93%
0,2	0,068463	98,10%	12	12,88%	3,90%	52,49%
0,9	0,070797	98,12%	28	9,91%	0,62%	50,88%
1,0	0,071740	98,04%	21	8,71%	-0,70%	50,22%
1,2	0,075190	98,11%	27	4,32%	-5,54%	47,83%
1,4	0,077389	97,93%	30	1,52%	-8,63%	46,30%
1,1	0,077895	97,85%	25	0,88%	-9,34%	45,95%
1,3	0,078983	97,85%	27	-0,51%	-10,87%	45,20%
0,1	0,079705	98,06%	16	-1,43%	-11,88%	44,69%
1,5	0,087271	97,65%	31	-11,05%	-22,50%	39,44%
5 camadas ocultas						
0,5	0,059021	98,41%	15	34,41%	15,09%	97,44%
0,4	0,061104	98,24%	13	32,10%	12,09%	97,35%
0,7	0,063066	98,28%	18	29,92%	9,27%	97,27%
0,8	0,064597	98,27%	20	28,22%	7,07%	97,20%
0,6	0,066781	98,14%	19	25,79%	3,92%	97,11%
0,3	0,067101	98,14%	14	25,43%	3,46%	97,09%
0,9	0,067820	98,00%	16	24,63%	2,43%	97,06%
1,2	0,072013	97,95%	24	19,97%	-3,60%	96,88%
1,1	0,072348	98,03%	28	19,60%	-4,08%	96,87%
0,2	0,073875	98,02%	14	17,91%	-6,28%	96,80%
1,0	0,077375	97,99%	21	14,02%	-11,32%	96,65%
1,3	0,079316	97,78%	22	11,86%	-14,11%	96,56%
1,5	0,082544	97,64%	27	8,27%	-18,75%	96,42%
1,4	0,088475	97,63%	26	1,68%	-27,29%	96,17%
0,1	0,088582	98,04%	13	1,56%	-27,44%	96,16%
6 camadas ocultas						
0,7	0,062927	98,41%	20	26,19%	17,63%	97,27%
0,6	0,062929	98,37%	17	26,19%	17,62%	97,27%
0,5	0,065240	98,29%	16	23,48%	14,60%	97,17%
0,3	0,066615	98,28%	15	21,87%	12,80%	97,11%
0,4	0,068307	98,26%	13	19,88%	10,58%	97,03%
0,8	0,069036	98,29%	23	19,03%	9,63%	97,00%
0,9	0,070603	98,25%	22	17,19%	7,58%	96,93%
1,0	0,076088	98,11%	21	10,76%	0,40%	96,69%

Continua na próxima página

ρ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
				relu	tanh	sigmoid
1,2	0,076435	97,85%	20	10,35%	-0,06%	96,68%
1,3	0,079873	97,94%	24	6,32%	-4,56%	96,53%
1,1	0,079915	97,77%	17	6,27%	-4,61%	96,53%
0,2	0,086699	97,94%	13	-1,69%	-13,49%	96,23%
1,4	0,088990	97,62%	25	-4,37%	-16,49%	96,13%
1,5	0,089472	97,85%	33	-4,94%	-17,12%	96,11%
0,1	0,117533	97,39%	15	-37,85%	-53,86%	94,89%

A.3 Tabelas do Experimento IV

Tabela A.3: Baseline do Experimento IV.

Função	Camadas ocultas	*Cross entropy*	Épocas	Acurácia
Logística	1	0,087650	128	97,37%
Tangente hiperbólica		0,068392	37	97,91%
ReLU		0,063475	30	98,10%
Logística	2	0,099686	97	96,98%
Tangente hiperbólica		0,065699	25	98,13%
ReLU		0,068222	18	98,09%
Logística	3	0,113659	76	96,84%
Tangente hiperbólica		0,066276	29	98,19%
ReLU		0,071049	13	98,16%
Logística	4	0,154929	87	95,96%
Tangente hiperbólica		0,070056	24	98,01%
ReLU		0,082245	13	97,94%
Logística	5	2,302588	26	11,35%
Tangente hiperbólica		0,073298	25	98,13%
ReLU		0,084105	11	97,86%
Logística	6	2,302795	24	11,35%
Tangente hiperbólica		0,067317	18	98,13%
ReLU		0,124309	10	96,90%

Tabela A.4: Melhora relativa do *cross entropy* utilizando a função de ativação bi-hiperbólica escalada com 1 a 6 camadas ocultas, em relação à ReLU, tangente hiperbólica e logística. Tabela completa ordenada do melhor para o pior.

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
1 camada oculta							
3,5	2,0	0,058617	98,23%	26	7,65%	14,29%	33,12%
5,5	2,5	0,058713	98,18%	24	7,50%	14,15%	33,01%
2,5	1,0	0,058921	98,22%	24	7,17%	13,85%	32,78%
5,5	3,0	0,059013	98,21%	26	7,03%	13,71%	32,67%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
5,0	2,0	0,059167	98,23%	19	6,79%	13,49%	32,50%
3,0	2,0	0,059583	98,14%	24	6,13%	12,88%	32,02%
2,0	0,5	0,059612	98,17%	15	6,09%	12,84%	31,99%
1,0	0,5	0,059630	98,23%	31	6,06%	12,81%	31,97%
4,0	2,0	0,059816	98,20%	23	5,76%	12,54%	31,76%
5,0	3,0	0,059873	98,26%	37	5,67%	12,46%	31,69%
4,5	2,5	0,059982	98,15%	22	5,50%	12,30%	31,57%
1,5	0,5	0,060004	98,19%	19	5,47%	12,26%	31,54%
3,0	1,5	0,060111	98,18%	22	5,30%	12,11%	31,42%
4,0	2,5	0,060150	98,26%	28	5,24%	12,05%	31,37%
4,5	2,0	0,060150	98,28%	26	5,24%	12,05%	31,37%
5,5	4,5	0,060178	98,18%	31	5,19%	12,01%	31,34%
2,5	2,0	0,060349	98,19%	32	4,92%	11,76%	31,15%
5,0	2,5	0,060391	98,12%	23	4,86%	11,70%	31,10%
2,5	1,5	0,061170	98,11%	27	3,63%	10,56%	30,21%
2,0	1,0	0,061481	98,18%	26	3,14%	10,10%	29,86%
2,0	1,5	0,061578	98,14%	33	2,99%	9,96%	29,75%
4,5	3,5	0,061635	98,20%	34	2,90%	9,88%	29,68%
5,5	4,0	0,061699	98,12%	31	2,80%	9,79%	29,61%
4,0	3,0	0,062120	98,18%	39	2,13%	9,17%	29,13%
5,5	3,5	0,062175	98,13%	28	2,05%	9,09%	29,06%
4,0	1,5	0,062192	98,15%	19	2,02%	9,07%	29,05%
4,5	3,0	0,062319	98,12%	25	1,82%	8,88%	28,90%
3,5	3,0	0,062474	98,05%	36	1,58%	8,65%	28,72%
5,0	3,5	0,062778	98,15%	32	1,10%	8,21%	28,38%
3,5	2,5	0,063135	98,00%	27	0,54%	7,69%	27,97%
5,5	2,0	0,063162	97,97%	21	0,49%	7,65%	27,94%
4,5	1,5	0,063240	98,09%	18	0,37%	7,53%	27,85%
1,5	1,0	0,063377	98,08%	27	0,15%	7,33%	27,69%
4,0	3,5	0,063781	98,10%	38	-0,48%	6,74%	27,23%
3,5	1,5	0,063953	98,10%	18	-0,75%	6,49%	27,04%
3,0	1,0	0,064076	98,03%	18	-0,95%	6,31%	26,90%
3,5	1,0	0,064208	98,02%	14	-1,15%	6,12%	26,75%
4,5	4,0	0,064330	98,09%	37	-1,35%	5,94%	26,61%
5,0	4,5	0,064356	98,18%	42	-1,39%	5,90%	26,58%
4,5	4,5	0,064735	98,06%	35	-1,99%	5,35%	26,14%
5,0	4,0	0,064763	98,04%	33	-2,03%	5,31%	26,11%
3,0	2,5	0,065106	98,01%	31	-2,57%	4,80%	25,72%
3,5	3,5	0,066043	98,08%	43	-4,05%	3,43%	24,65%
5,5	1,5	0,066079	98,05%	13	-4,10%	3,38%	24,61%
5,0	1,5	0,066136	98,00%	15	-4,19%	3,30%	24,55%
5,0	5,0	0,066515	98,00%	37	-4,79%	2,74%	24,11%
3,0	3,0	0,066686	97,98%	32	-5,06%	2,49%	23,92%
5,5	5,0	0,066817	97,98%	35	-5,27%	2,30%	23,77%
2,5	2,5	0,067385	98,00%	42	-6,16%	1,47%	23,12%
2,5	3,0	0,067433	97,98%	43	-6,24%	1,40%	23,07%
1,5	1,5	0,067481	98,11%	49	-6,31%	1,33%	23,01%
4,0	4,0	0,067607	97,88%	33	-6,51%	1,15%	22,87%
4,0	1,0	0,067608	98,01%	14	-6,51%	1,15%	22,87%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
2,5	0,5	0,067646	98,04%	16	-6,57%	1,09%	22,82%
5,0	1,0	0,067983	98,02%	14	-7,10%	0,60%	22,44%
4,5	5,0	0,068065	98,01%	43	-7,23%	0,48%	22,34%
5,5	5,5	0,068582	97,99%	43	-8,05%	-0,28%	21,75%
5,0	5,5	0,068622	98,00%	50	-8,11%	-0,34%	21,71%
5,0	0,5	0,068789	97,98%	14	-8,37%	-0,58%	21,52%
2,0	2,0	0,069863	98,04%	50	-10,06%	-2,15%	20,29%
3,5	4,5	0,069964	97,93%	48	-10,22%	-2,30%	20,18%
4,5	5,5	0,070282	97,94%	49	-10,72%	-2,76%	19,82%
3,5	4,0	0,070450	97,86%	42	-10,99%	-3,01%	19,62%
4,0	5,0	0,070789	97,86%	41	-11,52%	-3,50%	19,24%
1,0	1,0	0,071387	97,87%	33	-12,46%	-4,38%	18,55%
5,5	1,0	0,071553	97,96%	13	-12,73%	-4,62%	18,37%
3,0	0,5	0,071889	97,91%	15	-13,26%	-5,11%	17,98%
4,5	1,0	0,072139	97,91%	15	-13,65%	-5,48%	17,70%
4,0	0,5	0,072277	97,89%	13	-13,87%	-5,68%	17,54%
2,5	3,5	0,072302	97,81%	56	-13,91%	-5,72%	17,51%
4,5	0,5	0,072528	97,93%	14	-14,26%	-6,05%	17,25%
2,0	2,5	0,072748	97,92%	50	-14,61%	-6,37%	17,00%
3,0	3,5	0,073565	97,76%	36	-15,90%	-7,56%	16,07%
1,5	2,0	0,073957	97,86%	59	-16,51%	-8,14%	15,62%
4,0	4,5	0,074371	97,76%	42	-17,17%	-8,74%	15,15%
3,5	5,0	0,074433	97,75%	56	-17,26%	-8,83%	15,08%
3,5	0,5	0,074556	97,85%	12	-17,46%	-9,01%	14,94%
2,0	3,0	0,074848	97,76%	61	-17,92%	-9,44%	14,61%
3,0	4,0	0,075066	97,83%	51	-18,26%	-9,76%	14,36%
5,5	0,5	0,075435	97,95%	17	-18,84%	-10,30%	13,94%
4,0	5,5	0,077066	97,77%	57	-21,41%	-12,68%	12,08%
2,0	3,5	0,077325	97,78%	70	-21,82%	-13,06%	11,78%
0,5	0,5	0,077746	97,74%	56	-22,48%	-13,68%	11,30%
3,0	5,0	0,078298	97,70%	58	-23,35%	-14,48%	10,67%
3,5	5,5	0,078335	97,62%	53	-23,41%	-14,54%	10,63%
1,0	1,5	0,078857	97,67%	65	-24,23%	-15,30%	10,03%
3,0	5,5	0,079084	97,74%	85	-24,59%	-15,63%	9,77%
2,5	4,0	0,079628	97,72%	65	-25,45%	-16,43%	9,15%
2,0	4,0	0,080306	97,70%	81	-26,52%	-17,42%	8,38%
2,5	5,0	0,080918	97,84%	82	-27,48%	-18,32%	7,68%
3,0	4,5	0,082818	97,54%	56	-30,47%	-21,09%	5,51%
1,5	2,5	0,083377	97,68%	51	-31,35%	-21,91%	4,88%
1,5	3,0	0,084249	97,48%	72	-32,73%	-23,19%	3,88%
2,5	4,5	0,084942	97,53%	60	-33,82%	-24,20%	3,09%
2,0	4,5	0,087063	97,45%	84	-37,16%	-27,30%	0,67%
1,0	2,0	0,088199	97,47%	79	-38,95%	-28,96%	-0,63%
0,5	1,0	0,088468	97,50%	83	-39,37%	-29,35%	-0,93%
2,0	5,0	0,088480	97,59%	109	-39,39%	-29,37%	-0,95%
2,5	5,5	0,089018	97,44%	88	-40,24%	-30,16%	-1,56%
1,5	4,0	0,090768	97,46%	118	-43,00%	-32,72%	-3,56%
1,5	3,5	0,091052	97,42%	83	-43,45%	-33,13%	-3,88%
1,5	5,0	0,091775	97,42%	142	-44,58%	-34,19%	-4,71%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
2,0	5,5	0,092266	97,36%	100	-45,36%	-34,91%	-5,27%
1,0	3,5	0,093128	97,33%	162	-46,72%	-36,17%	-6,25%
1,0	2,5	0,093797	97,40%	92	-47,77%	-37,15%	-7,01%
1,5	4,5	0,095199	97,52%	136	-49,98%	-39,20%	-8,61%
1,0	3,0	0,095205	97,33%	134	-49,99%	-39,20%	-8,62%
1,0	4,0	0,096844	97,31%	189	-52,57%	-41,60%	-10,49%
0,5	1,5	0,098310	97,27%	128	-54,88%	-43,74%	-12,16%
1,5	5,5	0,102484	97,15%	144	-61,46%	-49,85%	-16,92%
1,0	4,5	0,103085	97,22%	214	-62,40%	-50,73%	-17,61%
0,5	3,0	0,103976	97,20%	347	-63,81%	-52,03%	-18,63%
0,5	2,0	0,108142	96,81%	162	-70,37%	-58,12%	-23,38%
1,0	5,5	0,110607	96,85%	254	-74,25%	-61,73%	-26,19%
0,5	2,5	0,111548	96,72%	219	-75,74%	-63,10%	-27,27%
0,5	4,0	0,113216	96,91%	476	-78,36%	-65,54%	-29,17%
0,5	3,5	0,114695	96,89%	392	-80,69%	-67,70%	-30,86%
1,0	5,0	0,115352	96,44%	212	-81,73%	-68,66%	-31,61%
0,5	4,5	0,137031	95,97%	436	-115,88%	-100,36%	-56,34%
0,5	5,5	0,265768	92,61%	203	-318,70%	-288,60%	-203,22%
0,5	5,0	0,266550	92,53%	155	-319,93%	-289,74%	-204,11%
			2 camadas ocultas				
5,0	3,0	0,057192	98,34%	15	16,17%	12,95%	42,63%
1,5	1,0	0,057462	98,29%	22	15,77%	12,54%	42,36%
4,0	2,0	0,058585	98,19%	14	14,13%	10,83%	41,23%
5,5	3,0	0,059564	98,30%	16	12,69%	9,34%	40,25%
5,0	3,5	0,059581	98,20%	17	12,67%	9,31%	40,23%
1,0	0,5	0,060378	98,18%	23	11,50%	8,10%	39,43%
4,5	3,5	0,060421	98,09%	22	11,43%	8,03%	39,39%
5,0	2,5	0,060782	98,22%	15	10,91%	7,48%	39,03%
3,5	2,5	0,060921	98,18%	18	10,70%	7,27%	38,89%
2,0	1,5	0,061074	98,29%	25	10,48%	7,04%	38,73%
1,5	0,5	0,061192	98,10%	16	10,30%	6,86%	38,62%
3,0	1,5	0,061287	98,22%	18	10,17%	6,72%	38,52%
4,0	2,5	0,061292	98,17%	19	10,16%	6,71%	38,51%
4,5	2,0	0,061434	98,12%	12	9,95%	6,49%	38,37%
3,0	2,0	0,061460	98,29%	25	9,91%	6,45%	38,35%
2,5	2,0	0,061544	98,13%	23	9,79%	6,32%	38,26%
5,5	2,5	0,061677	98,16%	15	9,59%	6,12%	38,13%
5,5	3,5	0,061725	98,23%	21	9,52%	6,05%	38,08%
2,5	1,5	0,061895	98,19%	20	9,27%	5,79%	37,91%
2,0	1,0	0,061933	98,20%	20	9,22%	5,73%	37,87%
4,0	3,0	0,062220	98,26%	24	8,80%	5,30%	37,58%
3,5	2,0	0,062520	98,19%	18	8,36%	4,84%	37,28%
4,5	2,5	0,062546	98,08%	16	8,32%	4,80%	37,26%
5,5	4,0	0,062824	98,18%	26	7,91%	4,38%	36,98%
3,0	3,0	0,063061	98,11%	31	7,57%	4,02%	36,74%
4,5	3,0	0,063139	98,17%	25	7,45%	3,90%	36,66%
5,0	4,5	0,063158	98,23%	30	7,42%	3,87%	36,64%
2,5	1,0	0,063253	98,10%	14	7,28%	3,72%	36,55%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
5,5	4,5	0,063269	98,22%	24	7,26%	3,70%	36,53%
5,0	4,0	0,063462	98,22%	29	6,98%	3,40%	36,34%
4,0	3,5	0,063714	98,19%	24	6,61%	3,02%	36,09%
5,5	5,0	0,063963	98,07%	22	6,24%	2,64%	35,84%
3,0	2,5	0,064057	98,20%	26	6,11%	2,50%	35,74%
5,0	2,0	0,064350	98,14%	13	5,68%	2,05%	35,45%
3,5	3,0	0,064506	98,04%	28	5,45%	1,82%	35,29%
4,0	1,5	0,064934	98,08%	12	4,82%	1,16%	34,86%
5,0	5,0	0,064953	98,15%	28	4,79%	1,14%	34,84%
3,5	1,5	0,065367	98,08%	15	4,18%	0,51%	34,43%
4,5	4,0	0,065745	98,14%	26	3,63%	-0,07%	34,05%
4,5	1,5	0,066605	98,10%	11	2,37%	-1,38%	33,19%
4,0	4,0	0,066632	98,11%	30	2,33%	-1,42%	33,16%
5,0	1,5	0,067199	98,11%	12	1,50%	-2,28%	32,59%
3,0	1,0	0,067534	97,99%	12	1,01%	-2,79%	32,25%
1,5	1,5	0,067751	98,00%	35	0,69%	-3,12%	32,04%
3,5	3,5	0,068183	98,07%	35	0,06%	-3,78%	31,60%
2,0	0,5	0,068673	98,00%	12	-0,66%	-4,53%	31,11%
3,5	1,0	0,068783	98,04%	12	-0,82%	-4,69%	31,00%
5,5	2,0	0,068891	98,09%	14	-0,98%	-4,86%	30,89%
2,0	2,0	0,069266	97,99%	31	-1,53%	-5,43%	30,52%
4,5	4,5	0,069735	97,95%	29	-2,22%	-6,14%	30,05%
1,0	1,0	0,069744	97,85%	31	-2,23%	-6,16%	30,04%
2,5	0,5	0,070090	98,06%	12	-2,74%	-6,68%	29,69%
2,5	2,5	0,070200	97,93%	32	-2,90%	-6,85%	29,58%
5,0	5,5	0,070670	98,01%	34	-3,59%	-7,57%	29,11%
4,5	1,0	0,070949	98,03%	13	-4,00%	-7,99%	28,83%
5,5	5,5	0,071357	98,04%	34	-4,60%	-8,61%	28,42%
4,5	5,0	0,071363	98,01%	38	-4,60%	-8,62%	28,41%
5,5	1,0	0,071686	98,07%	17	-5,08%	-9,11%	28,09%
4,0	4,5	0,072274	97,88%	30	-5,94%	-10,01%	27,50%
2,5	3,0	0,073116	97,86%	35	-7,17%	-11,29%	26,65%
3,0	0,5	0,073254	97,96%	15	-7,38%	-11,50%	26,52%
5,5	1,5	0,073254	97,88%	12	-7,38%	-11,50%	26,52%
4,0	1,0	0,073746	97,84%	13	-8,10%	-12,25%	26,02%
2,0	2,5	0,074088	97,82%	39	-8,60%	-12,77%	25,68%
4,0	5,0	0,076035	97,89%	38	-11,45%	-15,73%	23,73%
5,0	1,0	0,077510	97,85%	13	-13,61%	-17,98%	22,25%
3,0	4,0	0,077634	97,75%	35	-13,80%	-18,17%	22,12%
3,5	4,0	0,077695	97,70%	33	-13,89%	-18,26%	22,06%
3,0	3,5	0,078088	97,80%	36	-14,46%	-18,86%	21,67%
3,5	4,5	0,078440	97,82%	40	-14,98%	-19,39%	21,31%
3,0	4,5	0,078575	97,73%	43	-15,18%	-19,60%	21,18%
4,5	5,5	0,079218	97,74%	36	-16,12%	-20,58%	20,53%
1,5	2,0	0,079581	97,61%	36	-16,65%	-21,13%	20,17%
4,0	5,5	0,079659	97,67%	38	-16,76%	-21,25%	20,09%
2,0	3,0	0,081377	97,65%	41	-19,28%	-23,86%	18,37%
2,5	3,5	0,081875	97,61%	36	-20,01%	-24,62%	17,87%
3,5	5,0	0,081917	97,62%	38	-20,07%	-24,69%	17,82%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
2,5	4,0	0,083083	97,50%	45	-21,78%	-26,46%	16,66%
3,5	0,5	0,083990	98,00%	18	-23,11%	-27,84%	15,75%
0,5	0,5	0,084628	97,65%	37	-24,05%	-28,81%	15,11%
1,5	2,5	0,085041	97,60%	49	-24,65%	-29,44%	14,69%
3,5	5,5	0,086155	97,63%	47	-26,29%	-31,14%	13,57%
2,0	3,5	0,087397	97,53%	53	-28,11%	-33,03%	12,33%
3,0	5,0	0,088251	97,72%	48	-29,36%	-34,33%	11,47%
4,0	0,5	0,089401	97,89%	23	-31,04%	-36,08%	10,32%
3,0	5,5	0,091187	97,54%	62	-33,66%	-38,80%	8,53%
2,5	4,5	0,091919	97,69%	61	-34,74%	-39,91%	7,79%
2,0	4,0	0,092604	97,61%	56	-35,74%	-40,95%	7,10%
1,0	2,0	0,092953	97,41%	52	-36,25%	-41,48%	6,75%
1,5	3,0	0,093157	97,46%	58	-36,55%	-41,79%	6,55%
2,5	5,0	0,093860	97,35%	50	-37,58%	-42,86%	5,84%
2,5	5,5	0,095048	97,40%	72	-39,32%	-44,67%	4,65%
1,5	3,5	0,096176	97,32%	78	-40,98%	-46,39%	3,52%
1,0	2,5	0,096980	97,40%	78	-42,15%	-47,61%	2,71%
4,5	0,5	0,097816	97,22%	25	-43,38%	-48,89%	1,88%
1,0	1,5	0,098798	97,10%	41	-44,82%	-50,38%	0,89%
1,5	4,0	0,099140	97,22%	80	-45,32%	-50,90%	0,55%
2,0	4,5	0,099498	97,27%	72	-45,84%	-51,45%	0,19%
0,5	1,0	0,099775	97,19%	67	-46,25%	-51,87%	-0,09%
2,0	5,0	0,100840	97,21%	77	-47,81%	-53,49%	-1,16%
1,5	5,0	0,101551	97,16%	118	-48,85%	-54,57%	-1,87%
2,0	5,5	0,102482	97,16%	87	-50,22%	-55,99%	-2,80%
1,5	4,5	0,105039	97,19%	102	-53,97%	-59,88%	-5,37%
1,0	3,5	0,105191	97,19%	117	-54,19%	-60,11%	-5,52%
0,5	1,5	0,107106	97,06%	99	-57,00%	-63,03%	-7,44%
5,0	0,5	0,108566	96,62%	25	-59,14%	-65,25%	-8,91%
1,0	3,0	0,110353	97,00%	97	-61,76%	-67,97%	-10,70%
1,0	4,5	0,111708	96,81%	168	-63,74%	-70,03%	-12,06%
1,0	4,0	0,112484	96,85%	143	-64,88%	-71,21%	-12,84%
1,5	5,5	0,113094	96,76%	117	-65,77%	-72,14%	-13,45%
0,5	2,0	0,113282	96,90%	162	-66,05%	-72,43%	-13,64%
5,5	0,5	0,114195	96,68%	29	-67,39%	-73,82%	-14,55%
1,0	5,0	0,115969	96,86%	200	-69,99%	-76,52%	-16,33%
0,5	2,5	0,119193	96,52%	201	-74,71%	-81,42%	-19,57%
1,0	5,5	0,123076	96,52%	232	-80,41%	-87,33%	-23,46%
0,5	3,0	0,123339	96,58%	252	-80,79%	-87,73%	-23,73%
0,5	4,0	0,128836	96,33%	388	-88,85%	-96,10%	-29,24%
0,5	3,5	0,129555	96,16%	309	-89,90%	-97,19%	-29,96%
0,5	5,0	0,137749	96,21%	543	-101,91%	-109,67%	-38,18%
0,5	5,5	0,141898	96,03%	598	-107,99%	-115,98%	-42,34%
0,5	4,5	0,142552	95,98%	437	-108,95%	-116,98%	-43,00%
3 camadas ocultas							
4,5	3,0	0,056720	98,24%	14	20,17%	14,42%	50,10%
5,5	3,5	0,058129	98,23%	15	18,18%	12,29%	48,86%
4,0	2,5	0,058137	98,36%	17	18,17%	12,28%	48,85%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
3,5	2,0	0,058380	98,24%	12	17,83%	11,91%	48,64%
1,5	1,0	0,058468	98,31%	17	17,71%	11,78%	48,56%
4,5	2,5	0,059697	98,22%	13	15,98%	9,93%	47,48%
5,0	2,5	0,059699	98,19%	14	15,97%	9,92%	47,48%
4,0	2,0	0,060154	98,16%	12	15,33%	9,24%	47,08%
3,5	3,0	0,061167	98,30%	22	13,91%	7,71%	46,18%
4,0	3,0	0,061399	98,22%	18	13,58%	7,36%	45,98%
2,5	1,5	0,061716	98,26%	17	13,14%	6,88%	45,70%
5,0	3,5	0,061792	98,27%	18	13,03%	6,77%	45,63%
2,5	2,0	0,061858	98,29%	21	12,94%	6,67%	45,58%
3,0	2,0	0,062040	98,32%	18	12,68%	6,39%	45,42%
5,5	4,0	0,062172	98,21%	19	12,49%	6,19%	45,30%
4,5	3,5	0,062590	98,18%	17	11,91%	5,56%	44,93%
3,0	2,5	0,062646	98,17%	22	11,83%	5,48%	44,88%
1,0	0,5	0,062661	98,10%	16	11,81%	5,45%	44,87%
3,5	2,5	0,062795	98,07%	18	11,62%	5,25%	44,75%
5,5	5,0	0,063193	98,13%	23	11,06%	4,65%	44,40%
2,0	1,5	0,063346	98,17%	21	10,84%	4,42%	44,27%
5,0	3,0	0,063476	98,14%	16	10,66%	4,22%	44,15%
5,0	4,0	0,063490	98,19%	21	10,64%	4,20%	44,14%
1,5	0,5	0,063640	98,12%	11	10,43%	3,98%	44,01%
5,5	4,5	0,063656	98,18%	23	10,41%	3,95%	43,99%
5,5	3,0	0,063737	98,12%	14	10,29%	3,83%	43,92%
2,0	1,0	0,063761	98,18%	14	10,26%	3,79%	43,90%
5,0	4,5	0,065022	98,06%	20	8,48%	1,89%	42,79%
4,5	2,0	0,065201	98,05%	12	8,23%	1,62%	42,63%
4,0	3,5	0,065250	98,17%	26	8,16%	1,55%	42,59%
3,0	1,5	0,065710	98,14%	14	7,51%	0,85%	42,19%
5,5	2,5	0,066034	98,34%	12	7,06%	0,37%	41,90%
2,5	2,5	0,066134	98,30%	26	6,92%	0,21%	41,81%
4,5	4,5	0,066691	98,19%	29	6,13%	-0,63%	41,32%
4,5	4,0	0,066862	98,09%	23	5,89%	-0,88%	41,17%
4,0	1,5	0,067018	98,08%	11	5,67%	-1,12%	41,04%
3,5	1,5	0,067415	98,03%	12	5,11%	-1,72%	40,69%
3,0	3,0	0,067602	98,10%	27	4,85%	-2,00%	40,52%
4,0	4,0	0,067729	98,07%	28	4,67%	-2,19%	40,41%
5,0	5,0	0,067913	98,15%	26	4,41%	-2,47%	40,25%
3,5	3,5	0,068143	97,96%	28	4,09%	-2,82%	40,05%
5,0	2,0	0,068331	98,06%	12	3,83%	-3,10%	39,88%
3,0	1,0	0,068678	98,10%	12	3,34%	-3,62%	39,58%
5,5	5,5	0,068848	98,07%	25	3,10%	-3,88%	39,43%
5,5	2,0	0,068953	98,17%	12	2,95%	-4,04%	39,33%
2,0	2,0	0,069435	97,98%	24	2,27%	-4,77%	38,91%
2,0	0,5	0,069452	98,23%	13	2,25%	-4,79%	38,89%
4,5	1,5	0,069962	97,99%	12	1,53%	-5,56%	38,45%
2,5	1,0	0,070534	97,96%	12	0,72%	-6,42%	37,94%
1,5	1,5	0,072141	97,91%	24	-1,54%	-8,85%	36,53%
5,0	5,5	0,072197	98,07%	33	-1,62%	-8,93%	36,48%
5,5	1,5	0,072836	97,97%	14	-2,52%	-9,90%	35,92%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
1,0	1,0	0,073071	97,96%	30	-2,85%	-10,25%	35,71%
4,0	4,5	0,074223	97,94%	29	-4,47%	-11,99%	34,70%
3,0	3,5	0,074966	97,81%	31	-5,51%	-13,11%	34,04%
4,5	5,0	0,075384	97,88%	31	-6,10%	-13,74%	33,68%
5,0	1,5	0,075949	98,07%	12	-6,90%	-14,60%	33,18%
3,5	1,0	0,075968	98,07%	13	-6,92%	-14,62%	33,16%
3,5	4,0	0,076398	97,64%	26	-7,53%	-15,27%	32,78%
4,5	5,5	0,076614	97,98%	36	-7,83%	-15,60%	32,59%
4,0	1,0	0,077529	98,14%	16	-9,12%	-16,98%	31,79%
2,5	3,0	0,077622	97,84%	33	-9,25%	-17,12%	31,71%
4,5	1,0	0,078547	98,08%	17	-10,55%	-18,51%	30,89%
2,0	2,5	0,080405	97,75%	30	-13,17%	-21,32%	29,26%
4,0	5,5	0,080544	97,83%	33	-13,36%	-21,53%	29,14%
2,5	0,5	0,080665	98,01%	16	-13,53%	-21,71%	29,03%
0,5	0,5	0,084739	97,63%	37	-19,27%	-27,86%	25,44%
1,5	2,0	0,086586	97,48%	29	-21,87%	-30,64%	23,82%
3,5	4,5	0,087654	97,58%	30	-23,37%	-32,26%	22,88%
1,0	1,5	0,088441	97,56%	39	-24,48%	-33,44%	22,19%
3,5	5,5	0,088832	97,63%	38	-25,03%	-34,03%	21,84%
2,5	3,5	0,089041	97,57%	33	-25,32%	-34,35%	21,66%
2,0	3,0	0,089336	97,65%	46	-25,74%	-34,79%	21,40%
3,0	4,0	0,090231	97,56%	36	-27,00%	-36,14%	20,61%
5,0	1,0	0,090287	97,83%	22	-27,08%	-36,23%	20,56%
3,5	5,0	0,092735	97,50%	38	-30,52%	-39,92%	18,41%
3,0	5,0	0,093556	97,58%	44	-31,68%	-41,16%	17,69%
1,5	2,5	0,093734	97,39%	36	-31,93%	-41,43%	17,53%
3,0	4,5	0,094273	97,39%	31	-32,69%	-42,24%	17,06%
3,0	5,5	0,095755	97,35%	45	-34,77%	-44,48%	15,75%
4,0	5,0	0,096213	97,23%	29	-35,42%	-45,17%	15,35%
2,0	4,0	0,098853	97,29%	54	-39,13%	-49,15%	13,03%
5,5	1,0	0,099100	97,60%	21	-39,48%	-49,53%	12,81%
1,5	3,0	0,101862	97,27%	50	-43,37%	-53,69%	10,38%
3,0	0,5	0,102201	97,46%	15	-43,85%	-54,21%	10,08%
2,0	4,5	0,103331	97,20%	55	-45,44%	-55,91%	9,09%
2,0	3,5	0,103811	97,09%	43	-46,11%	-56,63%	8,66%
1,5	3,5	0,104429	97,37%	65	-46,98%	-57,57%	8,12%
2,5	4,5	0,104590	97,15%	47	-47,21%	-57,81%	7,98%
2,5	5,5	0,104695	97,17%	56	-47,36%	-57,97%	7,89%
2,5	4,0	0,105251	96,99%	36	-48,14%	-58,81%	7,40%
2,0	5,0	0,105527	96,93%	59	-48,53%	-59,22%	7,15%
0,5	1,0	0,105653	97,14%	59	-48,70%	-59,41%	7,04%
1,0	2,5	0,105732	97,11%	65	-48,82%	-59,53%	6,97%
2,0	5,5	0,105936	97,28%	83	-49,10%	-59,84%	6,79%
3,5	0,5	0,106698	97,32%	24	-50,18%	-60,99%	6,12%
1,0	2,0	0,107974	97,18%	49	-51,97%	-62,92%	5,00%
1,5	4,0	0,108026	96,88%	72	-52,04%	-62,99%	4,96%
2,5	5,0	0,108255	97,06%	53	-52,37%	-63,34%	4,75%
1,5	5,0	0,108453	97,04%	108	-52,65%	-63,64%	4,58%
1,0	3,0	0,109394	96,96%	80	-53,97%	-65,06%	3,75%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
1,5	4,5	0,111185	96,82%	77	-56,49%	-67,76%	2,18%
0,5	1,5	0,111785	96,76%	93	-57,34%	-68,67%	1,65%
1,5	5,5	0,119884	96,58%	111	-68,73%	-80,89%	-5,48%
1,0	3,5	0,123323	96,64%	100	-73,57%	-86,07%	-8,50%
4,0	0,5	0,123331	96,24%	15	-73,59%	-86,09%	-8,51%
1,0	4,0	0,124931	96,72%	144	-75,84%	-88,50%	-9,92%
1,0	4,5	0,125115	96,51%	131	-76,10%	-88,78%	-10,08%
0,5	2,0	0,128668	96,38%	110	-81,10%	-94,14%	-13,21%
1,0	5,0	0,131652	96,43%	165	-85,30%	-98,64%	-15,83%
4,5	0,5	0,133831	96,10%	29	-88,36%	-101,93%	-17,75%
0,5	2,5	0,143152	96,12%	165	-101,48%	-115,99%	-25,95%
0,5	3,0	0,153967	95,82%	198	-116,71%	-132,31%	-35,46%
0,5	4,0	0,154893	95,80%	321	-118,01%	-133,71%	-36,28%
1,0	5,5	0,155735	95,62%	156	-119,19%	-134,98%	-37,02%
0,5	3,5	0,163002	95,37%	227	-129,42%	-145,94%	-43,41%
5,0	0,5	0,310111	92,49%	19	-336,47%	-367,91%	-172,84%
0,5	4,5	2,301021	11,35%	16	-3138,64%	-3371,88%	-1924,50%
0,5	5,0	2,301063	11,35%	12	-3138,70%	-3371,94%	-1924,53%
0,5	5,5	2,301073	11,35%	11	-3138,71%	-3371,96%	-1924,54%
5,5	0,5	2,481577	10,32%	7	-3392,77%	-3644,31%	-2083,35%
4 camadas ocultas							
5,5	4,5	0,056485	98,32%	14	31,32%	19,37%	63,54%
1,0	0,5	0,057696	98,36%	14	29,85%	17,64%	62,76%
1,5	1,0	0,057998	98,23%	16	29,48%	17,21%	62,56%
5,0	3,5	0,058591	98,41%	16	28,76%	16,37%	62,18%
4,0	2,5	0,059046	98,30%	13	28,21%	15,72%	61,89%
3,0	2,0	0,059597	98,44%	18	27,54%	14,93%	61,53%
5,5	3,0	0,060219	98,29%	13	26,78%	14,04%	61,13%
2,5	1,5	0,060508	98,34%	17	26,43%	13,63%	60,94%
4,0	3,0	0,060940	98,33%	20	25,90%	13,01%	60,67%
5,5	4,0	0,060943	98,13%	17	25,90%	13,01%	60,66%
2,0	1,5	0,061148	98,26%	20	25,65%	12,72%	60,53%
3,5	2,5	0,061370	98,44%	16	25,38%	12,40%	60,39%
4,5	3,5	0,061967	98,25%	18	24,66%	11,55%	60,00%
2,0	1,0	0,062224	98,26%	12	24,34%	11,18%	59,84%
5,0	4,0	0,063193	98,24%	17	23,16%	9,80%	59,21%
5,5	3,5	0,063564	98,30%	13	22,71%	9,27%	58,97%
4,5	2,5	0,063641	98,40%	15	22,62%	9,16%	58,92%
2,5	2,0	0,063877	98,18%	18	22,33%	8,82%	58,77%
4,5	3,0	0,064482	98,24%	14	21,60%	7,96%	58,38%
3,0	2,5	0,064510	98,29%	23	21,56%	7,92%	58,36%
5,0	4,5	0,064557	98,22%	24	21,51%	7,85%	58,33%
3,5	2,0	0,064597	98,27%	14	21,46%	7,79%	58,31%
3,0	1,5	0,064787	98,26%	12	21,23%	7,52%	58,18%
2,5	2,5	0,064867	98,09%	24	21,13%	7,41%	58,13%
5,5	2,5	0,064982	98,19%	12	20,99%	7,24%	58,06%
5,0	3,0	0,065950	98,30%	15	19,81%	5,86%	57,43%
4,5	2,0	0,066166	98,11%	12	19,55%	5,55%	57,29%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
4,0	2,0	0,066323	98,19%	11	19,36%	5,33%	57,19%
3,5	3,0	0,066541	98,19%	23	19,09%	5,02%	57,05%
1,5	1,5	0,066752	98,00%	21	18,84%	4,72%	56,91%
5,0	2,5	0,066867	98,21%	12	18,70%	4,55%	56,84%
4,5	4,0	0,066882	98,18%	18	18,68%	4,53%	56,83%
5,0	2,0	0,067076	98,20%	13	18,44%	4,25%	56,71%
5,0	5,0	0,067283	98,23%	23	18,19%	3,96%	56,57%
5,5	5,0	0,067971	97,91%	15	17,36%	2,98%	56,13%
4,0	3,5	0,068807	98,07%	21	16,34%	1,78%	55,59%
2,5	1,0	0,069263	98,05%	11	15,78%	1,13%	55,29%
3,5	1,5	0,069335	98,16%	12	15,70%	1,03%	55,25%
2,0	2,0	0,069901	97,92%	19	15,01%	0,22%	54,88%
3,5	3,5	0,070605	98,26%	27	14,15%	-0,78%	54,43%
3,0	3,0	0,070915	98,16%	25	13,78%	-1,23%	54,23%
4,0	4,0	0,071154	97,94%	19	13,49%	-1,57%	54,07%
5,5	5,5	0,071900	97,90%	21	12,58%	-2,63%	53,59%
1,5	0,5	0,072262	98,05%	12	12,14%	-3,15%	53,36%
5,0	5,5	0,073076	97,98%	24	11,15%	-4,31%	52,83%
4,0	1,5	0,073140	98,07%	13	11,07%	-4,40%	52,79%
5,5	2,0	0,073157	98,14%	14	11,05%	-4,43%	52,78%
3,5	4,0	0,073422	97,92%	22	10,73%	-4,80%	52,61%
4,5	4,5	0,075240	98,04%	29	8,52%	-7,40%	51,44%
4,0	4,5	0,076047	97,94%	27	7,54%	-8,55%	50,91%
3,0	1,0	0,076256	98,12%	13	7,28%	-8,85%	50,78%
1,0	1,0	0,076370	97,85%	27	7,14%	-9,01%	50,71%
2,5	3,0	0,077191	97,81%	27	6,15%	-10,18%	50,18%
4,5	1,5	0,077321	98,13%	12	5,99%	-10,37%	50,09%
4,5	5,0	0,079517	97,63%	22	3,32%	-13,50%	48,68%
5,5	1,5	0,080772	98,06%	18	1,79%	-15,30%	47,87%
3,5	4,5	0,082461	97,72%	28	-0,26%	-17,71%	46,77%
2,0	0,5	0,083121	97,93%	15	-1,07%	-18,65%	46,35%
4,0	5,5	0,083201	97,76%	32	-1,16%	-18,76%	46,30%
2,0	2,5	0,083425	97,66%	25	-1,43%	-19,08%	46,15%
3,5	1,0	0,083668	97,91%	12	-1,73%	-19,43%	46,00%
3,0	4,0	0,083817	97,70%	31	-1,91%	-19,64%	45,90%
4,0	5,0	0,084460	97,82%	29	-2,69%	-20,56%	45,48%
3,0	3,5	0,085425	97,77%	29	-3,87%	-21,94%	44,86%
1,5	2,0	0,086414	97,63%	30	-5,07%	-23,35%	44,22%
2,5	3,5	0,088042	97,65%	32	-7,05%	-25,67%	43,17%
4,0	1,0	0,088337	97,95%	17	-7,41%	-26,09%	42,98%
5,0	1,5	0,088454	97,97%	16	-7,55%	-26,26%	42,91%
4,5	5,5	0,089450	97,70%	28	-8,76%	-27,68%	42,26%
3,5	5,0	0,089453	97,65%	35	-8,76%	-27,69%	42,26%
2,0	3,5	0,089532	97,47%	37	-8,86%	-27,80%	42,21%
0,5	0,5	0,091798	97,38%	30	-11,62%	-31,04%	40,75%
3,0	4,5	0,093229	97,58%	39	-13,36%	-33,08%	39,82%
2,5	4,0	0,094104	97,31%	33	-14,42%	-34,33%	39,26%
1,0	1,5	0,095450	97,42%	38	-16,06%	-36,25%	38,39%
2,5	4,5	0,097017	97,10%	37	-17,96%	-38,48%	37,38%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
2,5	5,0	0,098728	96,96%	39	-20,04%	-40,93%	36,28%
1,5	3,0	0,099487	97,39%	47	-20,96%	-42,01%	35,79%
2,0	4,0	0,100251	97,18%	47	-21,89%	-43,10%	35,29%
1,5	2,5	0,101686	97,12%	39	-23,64%	-45,15%	34,37%
3,0	5,5	0,101912	97,32%	47	-23,91%	-45,47%	34,22%
3,5	5,5	0,103006	97,08%	31	-25,24%	-47,03%	33,51%
2,5	0,5	0,103500	97,78%	16	-25,84%	-47,74%	33,20%
1,0	2,0	0,104460	97,23%	49	-27,01%	-49,11%	32,58%
5,0	1,0	0,104808	97,41%	20	-27,43%	-49,61%	32,35%
3,0	5,0	0,105856	97,05%	34	-28,71%	-51,10%	31,67%
2,5	5,5	0,106410	97,11%	48	-29,38%	-51,89%	31,32%
3,0	0,5	0,106450	97,29%	20	-29,43%	-51,95%	31,29%
4,5	1,0	0,107320	97,40%	18	-30,49%	-53,19%	30,73%
2,0	4,5	0,109841	97,23%	56	-33,55%	-56,79%	29,10%
1,5	3,5	0,111721	96,86%	50	-35,84%	-59,47%	27,89%
0,5	1,0	0,112178	96,92%	57	-36,39%	-60,13%	27,59%
1,0	2,5	0,115157	96,93%	55	-40,02%	-64,38%	25,67%
2,0	3,0	0,115468	97,20%	39	-40,40%	-64,82%	25,47%
2,0	5,0	0,117126	96,89%	60	-42,41%	-67,19%	24,40%
0,5	1,5	0,120762	96,78%	79	-46,83%	-72,38%	22,05%
1,5	4,0	0,123779	96,45%	53	-50,50%	-76,69%	20,11%
5,5	1,0	0,123967	97,10%	30	-50,73%	-76,95%	19,98%
2,0	5,5	0,124503	96,44%	52	-51,38%	-77,72%	19,64%
1,0	3,0	0,127262	96,37%	66	-54,74%	-81,66%	17,86%
1,5	4,5	0,127802	96,64%	66	-55,39%	-82,43%	17,51%
1,0	3,5	0,135348	96,22%	74	-64,57%	-93,20%	12,64%
1,5	5,5	0,137566	96,29%	87	-67,26%	-96,37%	11,21%
1,0	4,0	0,139473	96,03%	95	-69,58%	-99,09%	9,98%
1,5	5,0	0,140194	96,14%	73	-70,46%	-100,12%	9,51%
1,0	4,5	0,142792	96,05%	117	-73,62%	-103,83%	7,83%
0,5	2,0	0,155738	95,92%	117	-89,36%	-122,31%	-0,52%
3,5	0,5	0,159736	95,57%	13	-94,22%	-128,01%	-3,10%
1,0	5,0	0,170880	95,59%	157	-107,77%	-143,92%	-10,30%
1,0	5,5	2,300752	11,35%	16	-2697,44%	-3184,16%	-1385,04%
0,5	2,5	2,300869	11,35%	7	-2697,58%	-3184,33%	-1385,11%
0,5	3,5	2,300949	11,35%	18	-2697,68%	-3184,44%	-1385,16%
0,5	5,0	2,301057	11,35%	9	-2697,81%	-3184,60%	-1385,23%
0,5	4,5	2,301074	11,35%	7	-2697,83%	-3184,62%	-1385,24%
0,5	5,5	2,301160	11,35%	8	-2697,93%	-3184,74%	-1385,30%
0,5	4,0	2,301162	11,35%	15	-2697,94%	-3184,75%	-1385,30%
0,5	3,0	2,301219	11,35%	16	-2698,00%	-3184,83%	-1385,34%
4,0	0,5	2,457626	10,32%	11	-2888,18%	-3408,09%	-1486,29%
5,5	0,5	2,509450	9,58%	8	-2951,19%	-3482,06%	-1519,74%
5,0	0,5	2,611486	10,09%	9	-3075,25%	-3627,71%	-1585,60%
4,5	0,5	2,618242	11,35%	24	-3083,47%	-3637,36%	-1589,96%
			5 camadas ocultas				
4,0	3,0	0,057927	98,40%	17	31,13%	20,97%	97,48%
4,5	3,5	0,059495	98,37%	18	29,26%	18,83%	97,42%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
1,5	1,0	0,059771	98,42%	14	28,93%	18,45%	97,40%
3,5	2,0	0,059950	98,31%	13	28,72%	18,21%	97,40%
2,0	1,5	0,060567	98,35%	18	27,99%	17,37%	97,37%
3,0	2,0	0,060718	98,35%	14	27,81%	17,16%	97,36%
1,0	0,5	0,061171	98,34%	15	27,27%	16,54%	97,34%
4,5	3,0	0,061595	98,32%	14	26,76%	15,97%	97,32%
5,0	3,0	0,061659	98,27%	14	26,69%	15,88%	97,32%
2,5	2,0	0,061868	98,32%	22	26,44%	15,59%	97,31%
3,5	3,0	0,062020	98,42%	21	26,26%	15,39%	97,31%
5,5	4,5	0,062188	98,28%	14	26,06%	15,16%	97,30%
2,5	1,5	0,062454	98,21%	14	25,74%	14,79%	97,29%
5,0	3,5	0,062514	98,38%	15	25,67%	14,71%	97,29%
5,5	3,5	0,062986	98,28%	14	25,11%	14,07%	97,26%
4,5	2,5	0,063195	98,27%	13	24,86%	13,78%	97,26%
5,5	4,0	0,063243	98,28%	16	24,80%	13,72%	97,25%
5,5	3,0	0,063421	98,30%	13	24,59%	13,48%	97,25%
5,0	4,0	0,064095	98,23%	14	23,79%	12,56%	97,22%
4,0	2,5	0,064350	98,27%	14	23,49%	12,21%	97,21%
3,5	2,5	0,065338	98,26%	17	22,31%	10,86%	97,16%
3,0	2,5	0,066083	98,24%	18	21,43%	9,84%	97,13%
2,0	1,0	0,066153	98,10%	12	21,34%	9,75%	97,13%
4,0	2,0	0,066302	98,28%	12	21,17%	9,54%	97,12%
3,0	1,5	0,067390	98,26%	12	19,87%	8,06%	97,07%
4,0	3,5	0,067478	98,21%	20	19,77%	7,94%	97,07%
4,5	4,0	0,067946	98,31%	21	19,21%	7,30%	97,05%
5,0	4,5	0,068300	98,16%	23	18,79%	6,82%	97,03%
4,5	4,5	0,068462	98,25%	25	18,60%	6,60%	97,03%
4,5	2,0	0,069863	98,16%	13	16,93%	4,69%	96,97%
5,0	2,5	0,070042	98,09%	13	16,72%	4,44%	96,96%
5,5	5,0	0,070145	98,18%	24	16,60%	4,30%	96,95%
5,0	5,0	0,071403	98,15%	26	15,10%	2,59%	96,90%
5,5	5,5	0,071908	98,25%	28	14,50%	1,90%	96,88%
5,5	2,5	0,072371	98,21%	13	13,95%	1,26%	96,86%
3,0	3,0	0,072419	97,88%	18	13,89%	1,20%	96,85%
2,5	2,5	0,072629	98,12%	23	13,64%	0,91%	96,85%
2,0	2,0	0,072833	98,21%	26	13,40%	0,63%	96,84%
3,5	3,5	0,073132	98,09%	19	13,05%	0,23%	96,82%
2,5	1,0	0,073756	98,13%	14	12,30%	-0,62%	96,80%
4,0	4,0	0,075263	97,94%	18	10,51%	-2,68%	96,73%
4,5	5,5	0,075745	98,04%	32	9,94%	-3,34%	96,71%
4,5	5,0	0,076520	98,07%	28	9,02%	-4,40%	96,68%
1,5	1,5	0,076716	98,14%	27	8,79%	-4,66%	96,67%
1,5	0,5	0,077161	98,02%	13	8,26%	-5,27%	96,65%
3,5	1,5	0,077544	98,12%	13	7,80%	-5,79%	96,63%
4,0	1,5	0,078190	98,18%	14	7,03%	-6,67%	96,60%
1,0	1,0	0,079221	98,04%	25	5,81%	-8,08%	96,56%
5,0	5,5	0,079794	97,95%	25	5,13%	-8,86%	96,53%
5,0	2,0	0,081238	98,11%	13	3,41%	-10,83%	96,47%
4,0	4,5	0,081313	97,74%	24	3,32%	-10,93%	96,47%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
2,5	3,0	0,081750	97,77%	27	2,80%	-11,53%	96,45%
3,0	1,0	0,082150	98,03%	16	2,32%	-12,08%	96,43%
4,0	5,0	0,083194	97,69%	28	1,08%	-13,50%	96,39%
5,5	2,0	0,083574	98,03%	16	0,63%	-14,02%	96,37%
4,0	5,5	0,086007	97,55%	29	-2,26%	-17,34%	96,26%
3,5	4,5	0,087723	97,71%	28	-4,30%	-19,68%	96,19%
3,0	4,0	0,088208	97,50%	24	-4,88%	-20,34%	96,17%
2,0	0,5	0,089901	98,06%	17	-6,89%	-22,65%	96,10%
2,5	3,5	0,090553	97,53%	34	-7,67%	-23,54%	96,07%
3,5	5,0	0,090581	97,43%	27	-7,70%	-23,58%	96,07%
4,5	1,5	0,091603	97,97%	17	-8,92%	-24,97%	96,02%
3,5	1,0	0,092307	98,02%	19	-9,75%	-25,93%	95,99%
1,5	2,0	0,092377	97,32%	24	-9,84%	-26,03%	95,99%
3,0	5,0	0,092903	97,52%	37	-10,46%	-26,75%	95,97%
2,0	2,5	0,092951	97,60%	30	-10,52%	-26,81%	95,96%
3,0	3,5	0,093691	97,65%	29	-11,40%	-27,82%	95,93%
1,5	2,5	0,093997	97,59%	40	-11,76%	-28,24%	95,92%
0,5	0,5	0,095845	97,31%	32	-13,96%	-30,76%	95,84%
3,0	4,5	0,096165	97,28%	30	-14,34%	-31,20%	95,82%
1,0	1,5	0,096672	97,41%	32	-14,94%	-31,89%	95,80%
2,5	5,0	0,097546	97,41%	42	-15,98%	-33,08%	95,76%
2,5	0,5	0,098676	97,56%	15	-17,32%	-34,62%	95,71%
2,5	4,0	0,099413	97,39%	38	-18,20%	-35,63%	95,68%
2,5	4,5	0,100146	97,34%	40	-19,07%	-36,63%	95,65%
3,5	5,5	0,100690	97,29%	35	-19,72%	-37,37%	95,63%
5,0	1,5	0,101576	97,90%	19	-20,77%	-38,58%	95,59%
4,5	1,0	0,102180	97,58%	19	-21,49%	-39,40%	95,56%
3,5	4,0	0,102989	97,05%	20	-22,45%	-40,51%	95,53%
2,0	3,5	0,103323	97,18%	37	-22,85%	-40,96%	95,51%
4,0	1,0	0,104739	97,46%	14	-24,53%	-42,89%	95,45%
2,0	4,5	0,105415	97,23%	50	-25,34%	-43,82%	95,42%
3,0	5,5	0,105655	97,16%	39	-25,62%	-44,14%	95,41%
5,5	1,5	0,105742	97,59%	16	-25,73%	-44,26%	95,41%
1,0	2,0	0,105864	97,11%	44	-25,87%	-44,43%	95,40%
1,5	3,0	0,105916	96,96%	41	-25,93%	-44,50%	95,40%
2,0	4,0	0,106685	97,12%	46	-26,85%	-45,55%	95,37%
2,0	3,0	0,107672	97,24%	34	-28,02%	-46,90%	95,32%
0,5	1,0	0,111791	96,83%	40	-32,92%	-52,52%	95,14%
1,5	3,5	0,113988	97,02%	54	-35,53%	-55,51%	95,05%
2,5	5,5	0,114010	96,88%	42	-35,56%	-55,54%	95,05%
3,0	0,5	0,124425	96,96%	28	-47,94%	-69,75%	94,60%
1,0	2,5	0,129548	96,68%	52	-54,03%	-76,74%	94,37%
2,0	5,0	0,129926	96,40%	46	-54,48%	-77,26%	94,36%
1,5	4,0	0,137853	96,30%	54	-63,91%	-88,07%	94,01%
5,0	1,0	0,140360	96,91%	26	-66,89%	-91,49%	93,90%
0,5	1,5	0,141523	96,22%	66	-68,27%	-93,08%	93,85%
1,0	3,0	0,142705	95,90%	63	-69,67%	-94,69%	93,80%
2,0	5,5	0,143109	96,35%	54	-70,16%	-95,24%	93,78%
1,0	3,5	0,154101	95,77%	86	-83,22%	-110,24%	93,31%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
1,5	4,5	0,155143	95,68%	60	-84,46%	-111,66%	93,26%
1,5	5,0	0,163543	96,11%	77	-94,45%	-123,12%	92,90%
1,5	5,5	0,164830	95,65%	89	-95,98%	-124,88%	92,84%
0,5	2,0	0,169086	95,63%	121	-101,04%	-130,68%	92,66%
1,0	4,0	0,187717	95,30%	127	-123,19%	-156,10%	91,85%
1,0	4,5	2,300898	11,35%	7	-2635,74%	-3039,10%	0,07%
1,0	5,0	2,300997	11,35%	10	-2635,86%	-3039,24%	0,07%
0,5	4,0	2,301046	11,35%	8	-2635,92%	-3039,30%	0,07%
0,5	5,5	2,301102	11,35%	7	-2635,99%	-3039,38%	0,06%
0,5	3,5	2,301120	11,35%	19	-2636,01%	-3039,40%	0,06%
1,0	5,5	2,301121	11,35%	20	-2636,01%	-3039,40%	0,06%
0,5	4,5	2,301124	11,35%	15	-2636,01%	-3039,41%	0,06%
0,5	3,0	2,301164	11,35%	13	-2636,06%	-3039,46%	0,06%
0,5	5,0	2,301171	11,35%	7	-2636,07%	-3039,47%	0,06%
0,5	2,5	2,301213	11,35%	7	-2636,12%	-3039,53%	0,06%
4,0	0,5	2,371465	11,35%	11	-2719,65%	-3135,37%	-2,99%
5,5	0,5	2,474616	10,32%	12	-2842,29%	-3276,10%	-7,47%
5,0	0,5	2,495537	11,35%	9	-2867,17%	-3304,65%	-8,38%
3,5	0,5	2,515041	10,10%	13	-2890,36%	-3331,25%	-9,23%
5,5	1,0	2,562491	10,28%	9	-2946,78%	-3395,99%	-11,29%
4,5	0,5	2,661784	9,80%	7	-3064,83%	-3531,46%	-15,60%
			6 camadas ocultas				
2,0	1,5	0,057881	98,47%	16	53,44%	14,02%	97,49%
1,5	1,0	0,058977	98,55%	16	52,56%	12,39%	97,44%
4,0	2,5	0,059377	98,37%	12	52,23%	11,79%	97,42%
4,5	3,5	0,059577	98,48%	16	52,07%	11,50%	97,41%
5,0	3,5	0,059846	98,33%	15	51,86%	11,10%	97,40%
5,5	4,0	0,059914	98,47%	15	51,80%	11,00%	97,40%
3,0	2,0	0,060569	98,51%	16	51,28%	10,02%	97,37%
5,5	4,5	0,061676	98,43%	20	50,38%	8,38%	97,32%
5,0	4,0	0,061875	98,41%	18	50,22%	8,08%	97,31%
4,0	3,0	0,062113	98,29%	16	50,03%	7,73%	97,30%
1,0	0,5	0,062495	98,29%	14	49,73%	7,16%	97,29%
4,5	3,0	0,062578	98,38%	14	49,66%	7,04%	97,28%
5,5	5,0	0,063220	98,27%	22	49,14%	6,09%	97,25%
3,5	2,5	0,063784	98,41%	17	48,69%	5,25%	97,23%
3,5	3,0	0,064658	98,29%	19	47,99%	3,95%	97,19%
5,5	3,5	0,064685	98,29%	14	47,96%	3,91%	97,19%
3,5	2,0	0,065108	98,31%	15	47,62%	3,28%	97,17%
2,5	2,0	0,065520	98,30%	19	47,29%	2,67%	97,15%
4,0	3,5	0,065729	98,27%	21	47,12%	2,36%	97,15%
2,5	1,5	0,066219	98,37%	15	46,73%	1,63%	97,12%
5,0	3,0	0,066273	98,26%	13	46,69%	1,55%	97,12%
5,0	4,5	0,066824	98,25%	24	46,24%	0,73%	97,10%
4,5	4,0	0,067859	98,33%	26	45,41%	-0,81%	97,05%
3,5	1,5	0,068068	98,32%	14	45,24%	-1,12%	97,04%
4,5	2,5	0,068465	98,25%	14	44,92%	-1,71%	97,03%
4,0	2,0	0,069604	98,26%	13	44,01%	-3,40%	96,98%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
3,0	1,5	0,070039	98,28%	13	43,66%	-4,04%	96,96%
5,0	5,0	0,070101	98,21%	28	43,61%	-4,14%	96,96%
2,0	1,0	0,070685	98,26%	12	43,14%	-5,00%	96,93%
4,5	4,5	0,070725	98,00%	16	43,11%	-5,06%	96,93%
4,0	4,0	0,071684	98,23%	28	42,33%	-6,49%	96,89%
5,5	3,0	0,071909	98,16%	14	42,15%	-6,82%	96,88%
3,5	3,5	0,072315	98,02%	18	41,83%	-7,42%	96,86%
3,0	3,0	0,072866	98,23%	26	41,38%	-8,24%	96,84%
2,5	2,5	0,072937	98,21%	24	41,33%	-8,35%	96,83%
4,5	2,0	0,073648	98,18%	14	40,75%	-9,40%	96,80%
5,0	2,5	0,074043	98,18%	13	40,44%	-9,99%	96,78%
2,0	2,0	0,076210	98,10%	27	38,69%	-13,21%	96,69%
5,5	5,5	0,076414	97,98%	18	38,53%	-13,51%	96,68%
5,5	2,5	0,077159	98,14%	14	37,93%	-14,62%	96,65%
5,0	5,5	0,078298	97,75%	19	37,01%	-16,31%	96,60%
3,0	2,5	0,079604	97,96%	13	35,96%	-18,25%	96,54%
2,5	1,0	0,080803	98,09%	15	35,00%	-20,03%	96,49%
4,5	5,0	0,081055	97,50%	19	34,80%	-20,41%	96,48%
4,0	5,5	0,081495	97,85%	31	34,44%	-21,06%	96,46%
1,5	0,5	0,081569	98,16%	16	34,38%	-21,17%	96,46%
1,5	1,5	0,081891	97,70%	18	34,12%	-21,65%	96,44%
4,0	4,5	0,082360	97,72%	21	33,75%	-22,35%	96,42%
1,0	1,0	0,082849	97,67%	19	33,35%	-23,07%	96,40%
3,5	4,0	0,083297	97,77%	24	32,99%	-23,74%	96,38%
2,0	2,5	0,083880	97,67%	29	32,52%	-24,60%	96,36%
4,0	1,5	0,084204	98,07%	18	32,26%	-25,09%	96,34%
2,5	3,0	0,084986	97,69%	26	31,63%	-26,25%	96,31%
3,0	3,5	0,085109	97,67%	24	31,53%	-26,43%	96,30%
4,5	1,5	0,085497	98,04%	13	31,22%	-27,01%	96,29%
5,0	2,0	0,086192	98,07%	15	30,66%	-28,04%	96,26%
4,0	5,0	0,088805	97,72%	26	28,56%	-31,92%	96,14%
3,0	1,0	0,090088	98,00%	13	27,53%	-33,83%	96,09%
1,5	2,0	0,090933	97,52%	30	26,85%	-35,08%	96,05%
4,5	5,5	0,092139	97,66%	25	25,88%	-36,87%	96,00%
5,5	2,0	0,092708	97,83%	10	25,42%	-37,72%	95,97%
0,5	0,5	0,093660	97,68%	35	24,66%	-39,13%	95,93%
3,0	4,0	0,094016	97,45%	32	24,37%	-39,66%	95,92%
3,5	5,0	0,095181	97,36%	27	23,43%	-41,39%	95,87%
2,0	3,0	0,097073	97,48%	35	21,91%	-44,20%	95,78%
3,5	5,5	0,098324	97,37%	35	20,90%	-46,06%	95,73%
2,5	3,5	0,100720	97,26%	33	18,98%	-49,62%	95,63%
3,0	4,5	0,100948	97,01%	28	18,79%	-49,96%	95,62%
2,0	3,5	0,101057	97,35%	39	18,71%	-50,12%	95,61%
1,0	1,5	0,102931	97,38%	37	17,20%	-52,90%	95,53%
3,0	5,0	0,104019	97,28%	35	16,32%	-54,52%	95,48%
3,5	4,5	0,106324	97,31%	29	14,47%	-57,95%	95,38%
1,5	2,5	0,106891	97,26%	39	14,01%	-58,79%	95,36%
2,5	5,0	0,107281	97,06%	35	13,70%	-59,37%	95,34%
2,5	4,0	0,108729	97,09%	33	12,53%	-61,52%	95,28%

Continua na próxima página

λ	τ	*cross entropy*	acurácia	épocas	melhora relativa do *cross entropy*		
					ReLU	Tanh	Logística
2,0	0,5	0,108883	97,40%	15	12,41%	-61,75%	95,27%
2,5	4,5	0,109615	97,00%	36	11,82%	-62,83%	95,24%
3,5	1,0	0,110116	97,14%	12	11,42%	-63,58%	95,22%
2,0	4,0	0,112064	96,90%	31	9,85%	-66,47%	95,13%
5,0	1,5	0,112696	97,22%	12	9,34%	-67,41%	95,11%
5,5	1,5	0,116097	97,53%	24	6,61%	-72,46%	94,96%
3,0	5,5	0,116484	96,74%	36	6,29%	-73,04%	94,94%
1,5	3,0	0,120889	97,00%	47	2,75%	-79,58%	94,75%
4,0	1,0	0,121574	97,04%	23	2,20%	-80,60%	94,72%
1,0	2,0	0,121747	96,81%	37	2,06%	-80,86%	94,71%
2,5	5,5	0,124153	96,76%	43	0,13%	-84,43%	94,61%
2,0	4,5	0,126923	96,73%	44	-2,10%	-88,55%	94,49%
0,5	1,0	0,130901	96,57%	44	-5,30%	-94,45%	94,32%
2,5	0,5	0,133983	96,72%	16	-7,78%	-99,03%	94,18%
4,5	1,0	0,135334	96,31%	20	-8,87%	-101,04%	94,12%
2,0	5,5	0,135864	96,43%	57	-9,30%	-101,83%	94,10%
1,5	4,0	0,140963	96,06%	52	-13,40%	-109,40%	93,88%
1,5	3,5	0,141680	96,23%	43	-13,97%	-110,47%	93,85%
1,0	2,5	0,142039	96,37%	53	-14,26%	-111,00%	93,83%
2,0	5,0	0,145709	96,22%	52	-17,22%	-116,45%	93,67%
1,0	3,0	0,154538	96,26%	72	-24,32%	-129,57%	93,29%
1,5	4,5	0,169607	96,05%	65	-36,44%	-151,95%	92,63%
0,5	1,5	0,176325	95,66%	79	-41,84%	-161,93%	92,34%
1,5	5,0	0,223770	94,57%	81	-80,01%	-232,41%	90,28%
1,0	3,5	0,342074	91,99%	97	-175,18%	-408,15%	85,15%
1,0	4,0	2,300940	11,35%	10	-1750,98%	-3318,07%	0,08%
0,5	2,5	2,300963	11,35%	18	-1751,00%	-3318,10%	0,08%
0,5	3,5	2,300995	11,35%	12	-1751,03%	-3318,15%	0,08%
1,5	5,5	2,301008	11,35%	21	-1751,04%	-3318,17%	0,08%
0,5	4,0	2,301020	11,35%	10	-1751,05%	-3318,19%	0,08%
0,5	5,0	2,301030	11,35%	12	-1751,06%	-3318,20%	0,08%
0,5	5,5	2,301094	11,35%	20	-1751,11%	-3318,30%	0,07%
0,5	4,5	2,301114	11,35%	12	-1751,12%	-3318,33%	0,07%
1,0	5,0	2,301125	11,35%	17	-1751,13%	-3318,34%	0,07%
0,5	3,0	2,301158	11,35%	7	-1751,16%	-3318,39%	0,07%
1,0	4,5	2,301165	11,35%	10	-1751,17%	-3318,40%	0,07%
1,0	5,5	2,301172	11,35%	12	-1751,17%	-3318,41%	0,07%
0,5	2,0	2,301204	11,35%	8	-1751,20%	-3318,46%	0,07%
5,5	0,5	2,374140	10,32%	12	-1809,87%	-3426,81%	-3,10%
5,0	1,0	2,403129	9,58%	14	-1833,19%	-3469,87%	-4,36%
3,0	0,5	2,418674	10,10%	10	-1845,70%	-3492,96%	-5,03%
5,5	1,0	2,455961	11,35%	9	-1875,69%	-3548,35%	-6,65%
3,5	0,5	2,481416	9,58%	8	-1896,17%	-3586,17%	-7,76%
4,0	0,5	2,526212	9,82%	8	-1932,20%	-3652,71%	-9,70%
5,0	0,5	2,538549	9,74%	8	-1942,13%	-3671,04%	-10,24%
4,5	0,5	3,711205	10,32%	9	-2885,47%	-5413,03%	-61,16%

A.4 Tabelas do Experimento V

Tabela A.5: Acurácia das funções avaliadas no Experimento V.

Função	camadas ocultas					
	1	2	3	4	5	6
BHAA	98,13%	98,27%	98,24%	98,02%	98,09%	97,95%
BHSA	98,14%	98,14%	98,16%	97,92%	97,92%	97,93%
HA	97,80%	97,34%	10,09%	10,09%	9,80%	10,28%
LReLU	98,06%	97,83%	97,97%	97,94%	97,71%	97,32%
Logística	97,68%	97,20%	96,66%	96,22%	11,35%	11,35%
P-Softplus	97,71%	97,70%	97,89%	98,04%	97,97%	96,97%
PELU	97,77%	97,80%	97,73%	97,97%	98,01%	97,81%
PReLU	98,07%	98,12%	98,13%	98,13%	97,85%	97,91%
ReLU	97,99%	98,09%	97,95%	98,18%	97,68%	97,60%
SHA-ReLU	97,98%	98,07%	97,99%	97,75%	97,95%	97,84%
SReLU	98,13%	98,00%	98,18%	98,22%	98,00%	98,08%
Softplus	97,62%	97,08%	97,49%	96,64%	96,48%	96,48%
TReLU	11,35%	11,35%	11,35%	11,35%	11,35%	11,35%
Tanh	98,06%	97,92%	98,07%	98,13%	97,90%	98,08%

Tabela A.6: Quantidade de épocas das funções avaliadas no Experimento V.

Função	camadas ocultas					
	1	2	3	4	5	6
BHAA	20	14	12	11	13	11
BHSA	20	12	12	12	13	17
HA	15	17	8	9	14	15
LReLU	33	18	19	12	16	12
Logística	165	111	80	102	9	36
P-Softplus	16	14	13	16	15	13
PELU	31	28	19	17	17	18

Continua na próxima página

Função	camadas ocultas					
	1	2	3	4	5	6
PReLU	22	17	18	18	14	10
ReLU	27	18	12	18	10	13
SHA-ReLU	25	20	17	14	13	13
SReLU	22	14	14	13	12	14
Softplus	57	44	51	27	31	29
TReLU	9	7	9	12	9	12
Tanh	42	24	27	21	18	21