

Pruning Method in The Architecture of Extreme Learning Machines Based on Partial Least Squares Regression

P. Vitor

Abstract—The extreme learning machines-ELM are useful models for performing classification and regression of data, as well as being an alternative methodology to techniques that use back-propagation to determine values of parameters used in hidden layers of the learning model. A problem that ELM may face when performing data mining techniques is to become a very generalist model with no accurate or time-consuming processing due to a high number of neurons in the hidden layers of its architecture, if a model very restricted the characteristics of the sample that was used for the training, losing its generalization capacity. As the number of neurons in the hidden layer increases, information unnecessary to the model can be included in the operations performed by the model, impairing the accuracy end of the ELM when sorting or performing regression data. To solve this problem, a pruning method based on partial least squares regression was proposed to act together with the neurons of the hidden layer of an ELM. By obtaining a ranking vector with regression values and between variables in the hidden layer of ELM, the selection of the neurons that have the best predictive capacity in the responses between the data is used. Data classification tests were performed using bases commonly used in work related to machine learning and compared to other classifier models. Lastly, it is verified that an ELM with fewer neurons in the hidden layer, where the selected neurons are the ones that contribute better to the classification of the model, improve the final accuracy of the model in comparison to an architecture with a much larger number of neurons and is statistically similar to other pruning methods of the neurons in the inner layer of the ELM, allowing gain of model performance without damaging the accuracy in the final results, without losing its ability to perform classification tasks correctly.

Index Terms—Extreme Learning Machine, Pattern Classification, Pruning Methods, Partial least squares regression.

I. INTRODUÇÃO

Para realizar técnicas de classificação e regressão de dados foi proposto por [1] um modelo de treinamento de redes neurais artificiais que possui em sua arquitetura uma camada oculta, onde sua principal característica é a geração aleatória de pesos sinápticos, apresentando melhoria no desempenho na atualização de valores internos do modelo devido a não necessidade de se atualizar toda a estrutura da rede neural. Essas estruturas são chamadas de máquinas de aprendizado extremo (*Extreme Learning Machine*- ELM). Apesar de ser uma alternativa mais eficiente aos treinos baseados em *back-propagation* [2], a ELM pode apresentar problemas de classificação de dados quando a quantidade de neurônios na

camada oculta é excessiva. Quando a número de neurônios na camada oculta é maior ou igual à quantidade de dados totais utilizadas pelo modelo, temos a situação que é chamada de *over fitting*, que seria uma super utilização dos dados para treinar a rede forçando o treinamento para que ele aconteça de forma desnecessária. Quando a quantidade de dados para treinar uma rede neural com camada oculta apresenta-se elevada, problemas podem surgir devido ao sistema tornar-se um ótimo classificador para o problema proposto, perdendo sua capacidade de generalização, isto é, torna-se um classificador excelente para o grupo de dados utilizado e não tão eficiente com novos dados coletados para o mesmo cenário inicial. Essa situação ocorre devido ao excesso de informação desnecessária que os neurônios na camada oculta podem transmitir para outras camadas da arquitetura da ELM. A partir do momento que se podem excluir informações irrelevantes ao modelo, através de uma poda dos neurônios internos desnecessários, podemos alterar a arquitetura interna da ELM, tornando-a mais enxuta e eficiente para atuar em técnicas de mineração de dados. O grande problema que os métodos de poda possuem é identificar o quanto uma informação é relevante ou não para o modelo, ou seja, a capacidade de que um dado tem em predizer uma situação real. Para encontrar uma forma de quantificar essa capacidade de predição de uma variável sobre a outra, existem métodos que são baseados em regressão por mínimos quadrados parciais [3], que são capazes de auxiliar na resolução de problemas onde se possui um grupo de variáveis e pretende-se encontrar em outro grupo a capacidade de predição da mesma. Um método proposto em [3] chamado de SIMPLS trabalha com esse conceito, buscando encontrar a regressão entre dois grupos de variáveis, possibilitando através de critérios de seleção das características, escolher o melhor subgrupo de elementos que são representativos para o problema analisado. Baseado nessa capacidade do método SIMPLS de criar um subgrupo de variáveis que são mais representativas em relação a um grupo de variáveis que fazem parte da sua estrutura, esse trabalho propõe a utilização da técnica de regressão por mínimos quadrados parciais para resolver o problema de se encontrar neurônios que sejam mais representativos para as camadas finais de uma ELM, através da aplicação do método SIMPLS em estruturas fundamentais da camada intermediária da ELM, definindo uma quantidade de neurônios internos mais coerente com a representatividade do modelo. A aplicação dessa técnica possibilitará a poda de informações que não contribuem para a efetividade do modelo, possibilitando um processamento de menor quantidade de informações para os

valores obtidos na camada interna da ELM e que contribuem para sua capacidade de atuação em técnicas de classificação. O artigo está organizado da seguinte forma. A Seção 2 apresenta os conceitos teóricos relativos a redes neurais e máquina de aprendizado extremo, além de evidenciar os conceitos de regressão por mínimos quadrados parciais e trabalhos de poda e regressão na ELM já realizados na literatura. Na Seção 3 é descrita a metodologia proposta de poda na ELM através do método proposto por [3]. Na Seção 4 resultados de experimentos numéricos são apresentados, além de avaliações estatísticas e a metodologia utilizada para realização de comparação do modelo com outros métodos de poda. Finalmente, na Seção 5 são apresentadas as conclusões.

II. REVISÃO BIBLIOGRÁFICA

A. Máquina de Aprendizado Extremo- Extreme Learning Machine

A máquina de aprendizado extremo (*Extreme Learning Machine* - ELM) é uma alternativa proposta por Huang et al [1] para resolver problemas, que sempre existiram em algoritmos de redes neurais artificiais, baseados em ajuste iterativo de parâmetros no treinamento (*back-propagation* [2]), ao realizar classificação de padrões e/ou regressão em dados. Algoritmos baseados em métodos de gradiente descendente [4] tem comportamentos diferentes quando a taxa de treinamento sofre mudanças, influenciando diretamente no resultado quando a variação envolvida é muito grande, provocando conversão muito lenta ou instabilidade e divergência da rede [1]. Para resolver os problemas das redes neurais com ajuste iterativo de parâmetros no treinamento, foi proposto um método que evita o excesso de adaptações nos atributos de uma rede neural não recorrente (*feed-forward*). A máquina de aprendizado extremo é um algoritmo de redes neurais que possui uma camada escondida e uma camada de saída na arquitetura de sua rede. Na camada escondida os pesos de entrada e o bias podem ser determinados de forma aleatória, ficando indefinidos somente os pesos que ligam a camada oculta à camada de saída. Os pesos da camada de saída são obtidos analiticamente, evitando grande número de iterações para encontrar as variáveis necessárias. A quantidade de atributos a serem calculados diminui sensivelmente, tornando a solução mais robusta para resolver problemas de regressão ou classificação de padrões [1]. Na figura 1 apresenta-se um exemplo de arquitetura de uma rede neural de múltiplas camadas, evidenciando a camada escondida e a camada de saída.

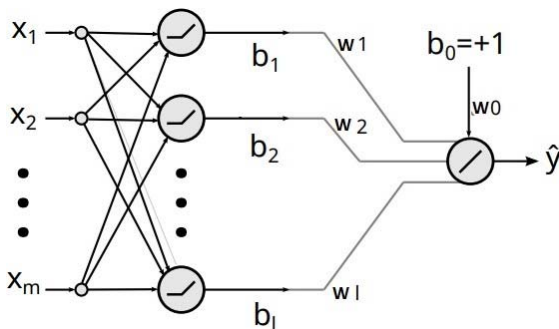


Fig. 1. Exemplo de arquitetura de uma rede neural com uma camada oculta e uma camada de saída.

Em [1] explica-se o funcionamento da obtenção das respostas de uma rede neural com camada oculta e geração aleatória de seus pesos. Considere uma rede neural de duas camadas com pesos sinápticos fixos e aleatórios na camada oculta representado por $\mathbf{W} \in \mathbb{R}^{p \times q}$, em que p é o número de unidades de entrada e q o número de neurônios ocultos. A saída da camada oculta é dada por:

$$\mathbf{H}(k+1) = f(\mathbf{W}^T(k)\mathbf{u}(k)) \quad (1)$$

Onde $\mathbf{u}(k) \in \mathbb{R}^p$ é o atual vetor de entrada e f uma função de ativação. Para o caso da ELM, f é a função de ativação aplicada ao produto escalar do vetor de entrada com os pesos escondidos. Essas funções podem ser do tipo sigmoideal, seno, base triangular, etc. A j -ésima saída da rede é dada por:

$$y_j(k+1) = \sum_{i=1}^q w_{ji}(k)H_i(k) \quad (2)$$

ELM propõe uma inicialização aleatória dos parâmetros da camada oculta, w_{ji} . Durante o treino, todas as entradas são sequência do treinamento $(\mathbf{u}(k), d(k), k=1 \dots n)$ para n como o número de entradas, são apresentadas à rede, e os estados correspondentes $(H(k), d(k))$ são coletados em suas respectivas matrizes \mathbf{H} e \mathbf{D} , onde $d(k)$ representa a saída desejada. O cálculo da matriz de pesos da camadas de saída, \mathbf{W} , é realizada por meio de uma regressão linear:

$$\mathbf{W} = \mathbf{H}^+ \mathbf{D} \quad (3)$$

em que \mathbf{H}^+ é a pseudo-inversa de Moore-Penrose da matriz das saídas da camada oculta \mathbf{H} [1].

B. Regressão por Mínimos Quadrados Parciais

A regressão por mínimos quadrados parciais, também conhecidas por *Partial Least Squares* (PLS), é um método que obtém informação através da correlação entre as variáveis que estão sendo analisadas. Esses métodos são baseados na obtenção de uma regressão linear obtida através de manipulações matriciais e estatísticas capaz de prever valores em um processo analisado [5]. É uma técnica que pode auxiliar modelos de treinamento de redes neurais artificiais, pois ela possibilita calcular uma relação entre as variáveis de entrada e a saída obtida dentro da camada interna de um modelo. No artigo utilizaremos um método de regressão por mínimos quadrados parciais para efetuarmos o cálculo da relação entre a matriz que representa as funções de ativação da camada interna de uma ELM e a resposta esperada pelo modelo, verificando a relação de relevância para as saídas da camada externa. Cada componente principal do modelo sofre uma pequena modificação para buscar a máxima covariância entre as funções de ativação da camada interna e a resposta esperada pelo modelo e passa a receber a terminologia de variável latente (VL). Não requer um conhecimento exato de todos os componentes presentes nas amostras podendo realizar a previsão de dados mesmo na presença de interferentes, desde que estes também estejam presentes por ocasião da construção do modelo [3].

A PLS utiliza a informação das funções de ativação da camada oculta na obtenção das respostas. O primeiro fator, VL, descreve a direção de máxima variância que também se correlaciona com a função de ativação. Estas variáveis latentes são na realidade combinações lineares das componentes principais calculadas pelo método PCR (*Principal Component Regression*) [6]. Há vários algoritmos para calcular a decomposição usada em PLS. Os dois mais populares são NIPALS (*Nonlinear Iterative Partial Least Squares*) [7] e SVD (*Singular Value Decomposition* ou Decomposição de Valores Singulares [6]). O método PLS assume um modelo inverso. A matriz de funções de ativação é decomposta em três matrizes, como no método PCR. Neste método, as matrizes \mathbf{X} e \mathbf{Y} são decompostas em matrizes menores de acordo com as equações 4 e 5 [8]:

$$\mathbf{X} = \mathbf{TP}^t + \mathbf{E} \quad (4)$$

\mathbf{Y} real com o resultado previsto pelo modelo. A primeira

$$\mathbf{Y} = \mathbf{UQ}^t + \mathbf{F} \quad (5)$$

Onde:

\mathbf{X} e \mathbf{Y} são as matrizes que serão decompostas;

\mathbf{T} e \mathbf{U} são matrizes de escores;

\mathbf{P} é a matriz de pesos de \mathbf{X} ; \mathbf{E} é o erro (resíduo) de \mathbf{X} ;

\mathbf{Q} é a matriz de pesos de \mathbf{Y} ; \mathbf{F} é o erro (resíduo) de \mathbf{Y} .

Após a decomposição de \mathbf{X} e \mathbf{Y} , uma relação linear é estabelecida entre os escores \mathbf{T} e \mathbf{U} , conforme a Equação [8]:

$$\mathbf{U}_n = \mathbf{b}_n + \mathbf{T}_n \quad (6)$$

Onde \mathbf{b} é o vetor dos coeficientes de regressão, e h é o número de variáveis latentes usadas no modelo PLS [8]. O termo escores está relacionado com a classificação das amostras, e os *loadings* relacionam-se com as variáveis. Uma maneira simples de interpretar o PLS é que ele forma novas variáveis \mathbf{X} (estimativa de variáveis latentes), através da combinação linear das antigas variáveis e usa esta matriz para o cálculo das estimativas de \mathbf{Y} .

Porém para encontrar um valor ideal de quanto de representatividade de uma variável sobre a outra não é uma tarefa fácil. Para otimização de modelos PLS, pode combinar-se validação cruzada com testes de validação externa. A validação cruzada é usada para determinação do modelo e do respectivo número de variáveis latentes, e o teste de validação externa para testar o desempenho final do modelo. A validação cruzada é uma técnica baseada nos dados de treinamento sendo usada para avaliar a capacidade de previsão dos modelos PLS para o conjunto de amostras. Esta validação cruzada é efetuada testando-se vários submodelos, ou seja, nesta técnica eliminam-se sucessivamente amostras do próprio conjunto de treinamento [8].

Em [8] é apresentada de forma gradual como a validação cruzada pode ser utilizada para estimar os melhores parâmetros para o modelo. Em um momento inicial, a primeira amostra dos dados de treinamento é eliminada. Seguidamente, o treinamento é efetuado usando as amostras que restam. Isso ocorre antes de ser testada na primeira amostra por comparação do

amostra é então colocada no conjunto de treinamento e a segunda amostra é eliminada, repetindo-se o procedimento. Este procedimento continua até cada uma das amostras terem sido objeto de teste, sendo retido o modelo que revelou melhor desempenho (erro mínimo) Em alguns casos para redes neurais pode-se avaliar a acurácia de acerto do modelo em relação as respostas esperadas. A eficiência do modelo por mínimos quadrados parciais pode ser avaliada pelo cálculo dos valores de Erro Quadrático Médio Normalizado (NMSE) ou pela acurácia do modelo (comparação das respostas obtidas em modelos de classificação comparadas com as respostas esperadas). Esses valores expressam a exatidão do modelo, ou seja, a proximidade entre o valor calculado pelo modelo e o valor verdadeiro ou obtido por um método de referência.

C. Modelos que Utilizam Treinamento Baseado em ELM Regularizado ou podado

A máquina de aprendizado extremo apresenta contratempos a medida que a quantidade de neurônios na camada escondida aumenta de forma desproporcional para a resolução dos problema alvo. Os tarefas de regressão/classificação dos modelos inteligentes podem se tornar extremamente dispendiosos, no que se refere a desempenho computacional, com topologias de rede que apresentam diversos neurônios na camada oculta. Quanto maior o número desses neurônios, mais custosos serão os cálculos para se obter os pesos que ligam a camada oculta até a camada de saída, além de uma maior probabilidade de acontecer erros na resposta final do modelo devido a um treinamento excessivo de atributos. Buscando resolver esse problema, foram propostos algoritmos que regularizam os elementos da camada oculta de uma ELM, através de métodos regularizadores que utilizam, para penalizar as funções de saída, regressões *lasso*, *ridge* e *elastic net* [9], coeficientes de correlação de *Matthews* [10] critérios estatísticos *Chi-Squared* e em ganho de informação [11], LARS e regularização de *Tikhonov* [12] e técnicas de *Dropout* [13]. Já em [14], aplicou-se métodos de regularização lineares e quadráticos para ELM com dados incompletos. Também foi formulada uma nova versão do ELM original através da utilização do método de regressão LARS [15] e [16] via métodos de gerenciamento da pontualidade. Métodos de regressão também foram aplicadas em modelos de redes neurais nebulosas que utilizam a ELM para atualizar parâmetros na camada oculta. Souza et al [17]–[22] utilizaram o método proposto por [23] para realizar a regularização da rede. Esse procedimento é definido como *Bolasso* (*bootstrap-enhanced least absolute shrinkage operator*) [23] onde algoritmo LARS é executado em várias replicações de *bootstrap* do conjunto de dados de treinamento. Para cada repetição, um subconjunto distinto dos regressores é selecionado [23]. Os regressores a serem incluídos no modelo final são definidos de acordo com a frequência com que cada um deles é selecionado através de diferentes ensaios [17] e um fator de consenso para verificar a quantidade de ensaios ideais para a seleção de neurônios relevantes.

Verifica-se que nas técnicas apresentadas, diversos recursos estatísticos podem ser utilizados para definir a relevância de

um neurônio para a camada de saída de uma rede neural. Em suma os modelos de redes neurais nebulosa atuam como classificadores [17]–[20] diferindo-se somente pelo neurônio lógico *fuzzy* utilizado, com previsão de séries temporais [24] e problemas de regressão linear [25]. Já Guimarães et al (2018) trabalham com modelos de rede neural *fuzzy* regularizados para problemas de imunoterapia [26], câncer de mama [21], Souza & Guimarães trabalham com modelos inteligentes na detecção de pulsares [27] e autismo em crianças [22]. Por fim Batista et. al. (2018) utilizam uma abordagem do modelo de redes neurais *fuzzy* para definir ataques cibernéticos por *SQL Injection* [28].

A ideia de poda de neurônios pode ser explicada pela visualização de uma rede neural com todas as suas ligações entre neurônios da camada oculta com a camada de saída. A aplicação de métodos de poda consistem em eliminar neurônios considerados desnecessários ou menos importantes, realizando assim uma otimização das ligações de camadas internas com a camada de saída. No trabalho de Souza [29] a rede neural fuzzy tem seus neurônios na camada oculta podados através da abordagem dos *f-scores* [30]. Portanto existe viabilidade da utilização de técnicas e abordagens estatísticas para eliminar neurônios da camada oculta das redes neurais artificiais que realizam a atualização de seus parâmetros através da ELM.

III. DESENVOLVIMENTO DO TRABALHO

Esse artigo propõe avaliar o desempenho de um modelo de treinamento de redes neurais baseado nos conceitos de máquina de aprendizado extremo aplicando ao mesmo técnicas de poda de neurônios menos relevantes, calculado através de técnicas de regressão por mínimos quadrados parciais. Esse método será utilizados para eliminar informação desnecessária entre a camada oculta de uma máquina de aprendizado extremo e a sua camada de saída quando o mesmo realiza classificação de padrões.

A. ELM Podada Via Regressão por Mínimos Quadrados Parciais

Os métodos baseados em regressão por mínimos quadrados parciais são muito usados em áreas, como por exemplo, a quimiometria, para modelar relações lineares entre medidas multivariadas, facilitando a compressão de vetores que contém variáveis preditoras em um conjunto de variáveis latentes ou de pontuação [3]. O algoritmo de [3] trabalha com a técnicas de mínimos quadrados parciais para resolver problemas que buscam maximizar a covariância entre duas matrizes. No contexto de ELM, as duas matrizes que se deseja encontrar a relação encontram-se na camada oculta da ELM. Para obter a covariância entre dois conjuntos de vetores foi utilizado o algoritmo *SIMPLS* [3]. O método que representa o algoritmo foi implementado no *Matlab*® e é representado pela função *plsregress*. A documentação do método encontra-se em ¹. O algoritmo *SIMPLS*, proposto por De Jong [3], trabalha os fatores PLS diretamente como uma combinação linear

das variáveis originais (centradas na média) em \mathbf{X} . Quando aplicado a uma simples variável dependente \mathbf{Y} , os resultados obtidos pelo método *SIMPLS* tornam-se essencialmente os mesmos daqueles obtidos pelo algoritmo NIPALS. Em suma, caso exista um desejo de se encontrar a melhor combinação linear entre o vetor de funções de ativação \mathbf{H} e as respostas esperadas \mathbf{D} , o método *SIMPLS* atua de forma a permitir que essa relação seja encontrada e mensurada. Ele pode ser resumido da seguinte forma para o problema proposto na ELM [3]:

1- Compute \mathbf{s} como $\mathbf{s} = \mathbf{H}^T \mathbf{D}$;

2- Compute as quantidades \mathbf{r} (*weights* PLS para \mathbf{H}), \mathbf{t} (*scores* PLS para \mathbf{H}), \mathbf{u} (*loadings* PLS para \mathbf{D}) e \mathbf{v} (*loadings* PLS para \mathbf{H}) como seguem:

$$\mathbf{r}_a = \mathbf{s} \quad (7)$$

$$\mathbf{t}_a = \mathbf{D} \mathbf{r}_a \quad (8)$$

$$\mathbf{t}_a = \frac{\mathbf{t}_a}{|\mathbf{t}_a|} \quad (9)$$

$$\mathbf{r}_a = \frac{\mathbf{r}_a}{|\mathbf{r}_a|} \quad (10)$$

$$\mathbf{v}_a = \mathbf{H}^T \mathbf{t}_a \quad (11)$$

$$\mathbf{u}_a = \mathbf{D}^T \mathbf{t}_a \quad (12)$$

3- Armazene \mathbf{r} , \mathbf{t} , \mathbf{u} e \mathbf{v} em \mathbf{R} , \mathbf{T} , \mathbf{U} e \mathbf{V} , respectivamente;

4- Projete \mathbf{s} no subespaço ortogonal a \mathbf{V}_a :

$$\mathbf{s} = \mathbf{s} - \mathbf{V}(\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T \mathbf{s} \quad (13)$$

5- Vá ao passo (2) para calcular o próximo vetor latente até alcançar \mathbf{A} vetores latentes;

6- Calcule o vetor de regressão como $\mathbf{b} = \mathbf{R} \mathbf{u}$

As funções de ativação da camada oculta do neurônio, representado pelo cálculo da equação 1 são alvo dessa pesquisa. Para tal, deseja-se calcular a regressão por mínimos quadrados parciais entre o vetor que representa as funções de ativação da camada oculta da ELM, \mathbf{H} (equação 1) e o vetor de respostas destinado ao treinamento, chamado aqui de \mathbf{Y}_t , possibilitando encontrar um vetor que fornece os valores de regressão entre as variáveis em questão. O resultado do método *plsregress* que será avaliado para a definição da poda dos neurônios a serem descartados serão os percentual de covariância explicada pelo modelo (θ) onde os valores próximos de 100% de covariância significam elementos muito inter-dependentes, ou seja, os maiores (menores) valores do primeira variável correspondem aos maiores (menores) valores da segunda variável analisada. De posse desses valores é possível avaliar quais relações possuem maior relevância para o modelo. Encontrar um valor ótimo é bem complexo para os valores obtidos, portanto foi abordado nesse trabalho a validação cruzada [31] para estimar o melhor valor de covariância para o modelo. Em suma, se o valor de θ for 70%, o algoritmo ele selecionará todos os neurônios que tenham valor de covariância acima

¹Partial least-squares regression. [online]. Available: <http://www.mathworks.com/help/stats/plsregress.html>.

de 70%, criando assim um subgrupo de neurônios na camada escondida. Com os neurônios selecionados após essa poda com o melhor valor de θ , são calculados novos valores de funções de ativação da primeira camada somente com os neurônios que atendem ao critério encontrado pelo método de validação cruzada. Após o novo valor das funções de ativação (equação 1) serem calculados é possível encontrar os pesos que ligam à camada interna a camada de saída do ELM (equação 3).

IV. TESTES REALIZADOS

Nesse capítulo será apresentada uma análise dos dados coletados nos testes de classificações de bases binárias. Para demonstrar como foram realizados os experimentos foram discriminados os fatores de teste, a forma de avaliação, os algoritmos e bases utilizadas além dos resultados encontrados. Por fim uma análise estatística auxiliará na validação e comparação do modelo com demais métodos na literatura. Para verificar o funcionamento dos métodos foram feitos testes referentes à resolução da classificação de uma ELM com a quantidade de neurônios na camada interna igual à quantidade de dados de cada uma das bases analisadas. A seguir algumas informações sobre as bases de dados utilizadas e testes realizados são esclarecidas. A acurácia do modelo ao classificar as bases binárias é calculada através da comparação do valor obtido pelo modelo, confrontando-o com o valor esperado. A cada acerto do neurônio (valor esperado é igual ao valor calculado) é incrementado uma unidade. No final é dividida a quantidade de acertos com o total possível de respostas corretas para a situação avaliada. O valor percentual é calculado para a acurácia de cada uma das classes.

A. Bases de Dados Utilizadas nos Testes

As bases de dados utilizadas foram obtidas do repositório em [32] e em algumas pesquisas de classificação de padrões realizadas por [33] e [34]. Ressalva-se a classe Íris que possui três tipos de classificação e foi readequada para ter somente duas saídas. A Tabela 1 detalha as bases incluindo a quantidade de amostras utilizadas para treinamento (70%) e teste (30%). Para evitar tendências nos testes executados todas as amostras disponíveis foram permutadas. Foram coletadas 30 medições de acurácia de cada uma das bases avaliadas em cada modelo analisado. As variáveis envolvidas no processo foram normalizadas com média zero e variância um. Todas as saídas do modelo foram normalizadas para o intervalo $[-1, 1]$ e as funções de ativação do tipo tangente hiperbólica. Na tabela abaixo as características das classes são apresentadas como foram distribuídas para a utilização nos testes do trabalho, onde podemos destacar a quantidade de dados destinadas a treinamento e teste, além de para cada base, evidenciar a quantidade de neurônios na camada oculta da ELM:

B. Testes

Para encontrar o melhor valor de covariância pelo método plsregress foram utilizadas técnicas de validação cruzada com o valor $\theta = 50\%$, 60% , 70% , 80% . Nesse caso, foram feitos testes utilizando esses valores de θ na fase de treinamento

TABELA I
DADOS DAS CLASSES UTILIZADAS NO TESTE

Base de Dados	Sig.	Caract	Treinamento	teste	(q)
Four Class	FCL	2	603	259	862
Haberman	HAB	3	214	92	306
Iris	IRI	4	105	45	150
Transfusion	TRA	4	523	225	748
Mammographic	MAM	5	581	249	830
Liver Disorder	LIV	6	242	103	345
Diabetes	DIA	8	538	230	768
Heart	HEA	13	189	81	270
German Credit	GER	14	390	168	558
Climate	CLI	18	377	163	540
Australian Credit	AUS	24	700	300	1000
Parkison	PAR	24	137	58	195
Ionosphere	ION	32	245	106	351

do modelo. O que obteve melhor resultado de acurácia no treinamento é utilizado para encontrar os pesos da camada de saída utilizando a pseudo-inversa, determinando a quantidade de neurônios que fará parte da arquitetura da rede neural. A partir da escolha dos melhores valores de θ , são recalculadas as funções de ativação (equação 1) e os pesos da camada de saída (equação 3). O modelo de aprendizagem de redes neurais baseado em ELM com poda na camada oculta proposto nesse artigo foi avaliado através de experimentos de classificação de padrões binários.

O algoritmo proposto nesse trabalho foi comparado com outros classificadores que realizam poda de neurônios para verificar sua capacidade de acurácia. Nos quadros 1 e 2, o valor principal é a acurácia média de 30 experimentos realizados e o valor entre parênteses é o respectivo desvio padrão. Já na Tabela 3 o valor principal é a média dos neurônios mais relevantes ao contexto avaliado juntamente com o valor do desvio padrão em parênteses nos 30 ensaios. Nos testes realizados buscou-se verificar a capacidade do modelo de aprendizagem proposto nesse artigo de aperfeiçoar a estrutura da rede através do método de poda que altera a definição da estrutura inicial (números de neurônios da camada oculta (q) igual a quantidade de amostras de cada base). Para efetuar essas comparações utilizou-se o algoritmo de treinamento proposto, a ELM em sua versão original, além de um algoritmo estado da arte de poda para ELM, o algoritmo OP-ELM, proposto por [15]. O quadro 2 apresenta os resultados de acurácia dos três métodos avaliados para as bases de teste e na tabela 3 os respectivos resultados de neurônios selecionados.

TABELA II
ACURÁCIA DOS MODELOS- TESTES DE CLASSIFICAÇÃO DE PADRÕES

Base de Dados	ELM-Podado	OP-ELM	ELM
FCL	95.04 (0.79)	93.70 (2.25)	90.65 (4.65)
HAB	75.14 (0.75)	78.24 (4.51)	74.18 (8.87)
IRI	99.14 (0.79)	99.02 (0.15)	98.41 (0.46)
TRA	76.87 (1.46)	79.42 (1.14)	77.48 (0.54)
MAM	83.99 (0.49)	84.66 (2.65)	86.75 (1.87)
LIV	83.45 (6.12)	82.07 (3.74)	79.42 (1.79)
DIA	84.98 (3.17)	85.44 (2.79)	84.68 (4.17)
HEA	88.46 (0.52)	87.15 (0.17)	91.49 (0.83)
GER	96.08 (0.68)	95.83 (1.42)	94.62 (1.76)
CLI	93.04 (0.64)	91.06 (0.51)	92.97 (0.22)
AUS	84.07 (0.59)	85.74 (0.87)	86.63 (3.81)
PAR	89.13 (1.45)	91.16 (0.82)	92.12 (0.47)
ION	93.16 (2.16)	95.12 (2.07)	96.87 (0.16)

TABELA III

NEURÔNIOS SELECIONADOS- TESTE DE CLASSIFICAÇÃO DE PADRÕES BINÁRIOS

Base de Dados	q	ELM-Podado	OP-ELM.	ELM
FCL	862	240.62 (17.12)	278.87 (54.11)	862(0)
HAB	306	75.42 (16.31)	79.12 (15.46)	306(0)
IRI	150	42.14 (12.66)	34.12 (8.63)	150(0)
TRA	748	258.54 (6.87)	216.74 (67.36)	748(0)
MAM	830	537.62 (6.14)	412.82(76.31)	830(0)
LIV	345	82.04 (26.87)	97.87 (43.86)	345(0)
DIA	768	287.44 (54.32)	180.98 (43.31)	768(0)
HEA	270	93.98 (16.09)	114.43 (6.87)	270(0)
GER	558	308.15 (43.25)	112.34 (27.40)	558(0)
CLI	540	296.41 (15.76)	250.98 (6.31)	540(0)
AUS	1000	225.58 (2.76)	246.72 (8.98)	1000(0)
PAR	195	50.98 (16.31)	48.17 (12.39)	195(0)
ION	351	112.23 (10.44)	169.98 (8.91)	351(0)

A figura 2 apresenta os resultados gráficos da comparação entre os neurônios finais em cada um dos testes. Percebe-se que a abordagem proposta nesse artigo melhorou muito a poda dos neurônios.

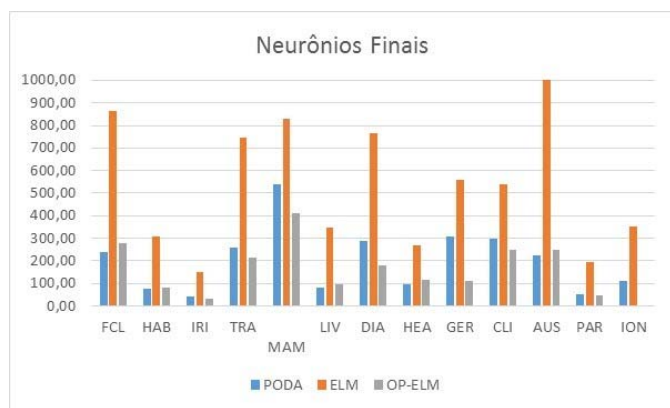


Fig. 2. Neurônios retidos nos 30 testes executados.

C. Resultados

Ao analisarmos os quadros 2 e 3, pode se verificar que o algoritmo proposto apresentou resultados relevantes na poda de neurônios desnecessários, com resultados de acurácia em destaque para quatro das bases avaliadas, principalmente nos testes em que a quantidade de neurônios estavam entre 150 e 500 neurônios na camada oculta. Porém quando avaliamos a proposta de poda, verifica-se o sucesso do algoritmo em realizar a eliminação de neurônios desnecessários de maneira eficiente em seis das bases de teste. Ao comparar o modelo desse artigo ao estado da arte em efetuar eliminação de neurônios em uma ELM, considera-se que a abordagem obteve bons resultados. De uma forma geral, os resultados obtidos de acurácia estão próximos entre si, o que inviabiliza uma análise superficial para verificar a eficiência do modelo proposto. Para tal foi aplicado uma avaliação do desempenho do modelo proposto executando uma análise estatística sobre os dados coletados na classificação de padrões.

Para realizar as avaliações estatísticas utiliza-se no teste realizado a análise de variância (ANOVA) [35] sobre os resultados de cada um dos grupos (algoritmo x fator de bloco).

De uma forma geral constata-se que o teste possui 39 (3 algoritmos e 13 bases) grupos. Através desse teste busca-se concluir se o desempenho dos algoritmos propostos nesse artigo apresenta desempenho médio igual ao estado da arte ao realizar as classificações de padrões. Na figura 3 está presente uma avaliação gráfica da relação acurácia por algoritmo e na figura 4 a relação entre a acurácia e o fator de blocagem dos problemas de classificação.

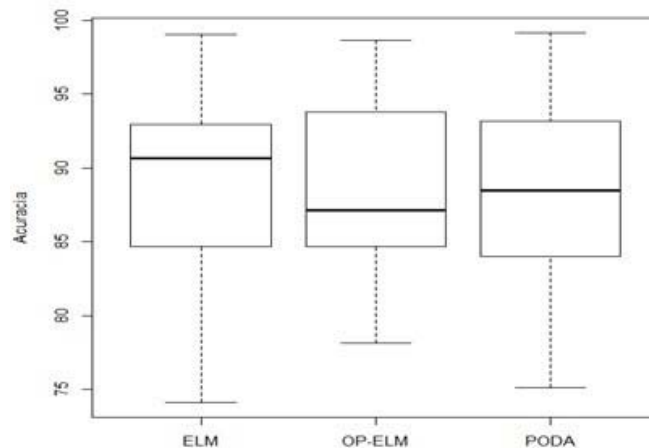


Fig. 3. Plot da relação Acurácia versus algoritmos.

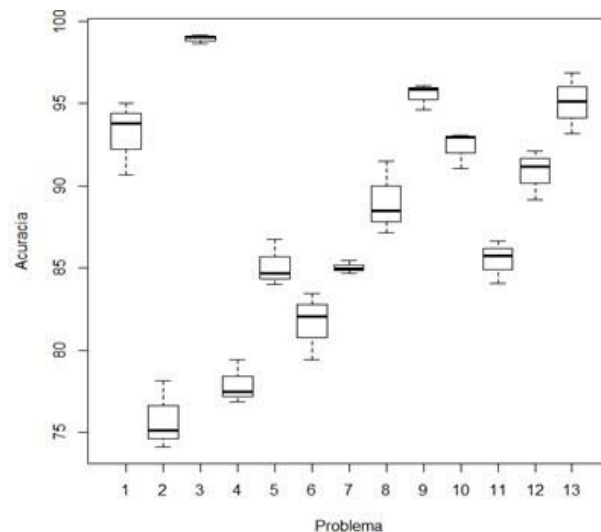


Fig. 4. Plot da relação Acurácia versus problemas em blocagem.

Após a execução do teste ANOVA conclui-se a um nível de significância de $\alpha=0.05$ que devemos aceitar a hipótese nula de igualdade dos desempenhos dos algoritmos a realizarem a classificação com essas bases de dados ($p\text{-value}=0.076$). Após confirmar todas as premissas de validação do resultado (normalidade, homoscedasticidade e independência), realizamos um teste *post-hoc* de múltiplas comparações chamado teste de Tukey [36] que realiza comparações de duas a duas entre todos os algoritmos envolvidos. As múltiplas comparações são apresentadas no quadro 4 podemos verificar que os modelos

testados para os problemas em bloco apresentam desempenhos iguais para realizar classificação de padrões.

Portanto o modelo proposto pode ser visto como uma metodologia viável para realizar classificação.

V. CONCLUSÃO

Após os testes realizados e as avaliações estatísticas finalizadas, podemos concluir que o métodos de poda apresentados nesse trabalho mantém a acurácia média de acertos na classificação dos dados quando comparados a uma estrutura de ELM com os neurônios completos na camada oculta e também com o estado da arte em efetuar poda de neurônios desnecessários na camada oculta de uma ELM. Com uma menor quantidade de neurônios, a probabilidade de acontecer o over training é menor, pois as informações desnecessárias são eliminadas, além de se reduzir os cálculos dos dados entre a camada oculta e a camada de saída. Os experimentos realizados e os resultados obtidos sugerem que o modelo é capaz de atuar como um classificador, apresentando resultados consistentes na classificação de padrões. Nos experimentos realizados podemos comprovar que os resultados do modelo proposto são semelhantes a outros classificadores, inclusive o que se baseiam em métodos de poda.

Para trabalhos futuros, métodos de otimização podem ser utilizados para avaliar os melhores valores de θ ou até mesmo outra técnica com menor custo computacional do que a validação cruzada. Experimentos com outras bases de maior quantidade de amostras também podem ser realizados para verificar a instabilidade do modelo.

AGRADECIMENTOS

Os agradecimentos são destinados ao CEFET-MG, ao Centro Universitário UNA e ao Centro Universitário UNI-BH.

REFERÊNCIAS

- [1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
- [2] J. Leonard and M. Kramer, "Improvement of the backpropagation algorithm for training neural networks," *Computers & Chemical Engineering*, vol. 14, no. 3, pp. 337–341, 1990.
- [3] S. De Jong, "Simpls: an alternative approach to partial least squares regression," *Chemometrics and intelligent laboratory systems*, vol. 18, no. 3, pp. 251–263, 1993.
- [4] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.
- [5] H. Martens, T. Naes, and T. Naes, *Multivariate calibration*. John Wiley & Sons, 1992.
- [6] M. Ferreira, A. M. Antunes, M. S. Melgo, and P. L. Volpe, "Chemometrics i: multivariate calibration, a tutorial," *Química Nova*, vol. 22, no. 5, pp. 724–731, 1999.
- [7] H. Wold, "Path models with latent variables: The nipals approach," in *Quantitative sociology*. Elsevier, 1975, pp. 307–357.
- [8] M. J. C. d. Santos *et al.*, "Estimativa da massa específica em etanol combustível por modelos de redes neurais artificiais e de regressão por mínimos quadrados parciais," 2013.
- [9] J. M. MartiNez-MartiNez, P. Escandell-Montero, E. Soria-Olivas, J. D. MartiN-Guerrero, R. Magdalena-Benedito, and J. Gómez-Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp. 3716–3721, 2011.
- [10] P. V. de Campos Souza, A. J. Guimaraes, V. S. Araújo, T. S. Rezende, and V. J. S. Araújo, "Method of pruning the hidden layer of the extreme learning machine based on correlation coefficient," in *5th IEEE Latin American Conference on Computational Intelligence LA-CCI*, vol. 5, no. 1, 2018.
- [11] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1-3, pp. 359–366, 2008.
- [12] Y. Míche, M. Van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "Trop-elm: a double-regularized elm using lars and tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.
- [13] J. Zhai, L. Zang, and Z. Zhou, "Ensemble dropout extreme learning machine via fuzzy integral for data classification," *Neurocomputing*, vol. 275, pp. 1043–1052, 2018.
- [14] Q. Yu, Y. Míche, E. Eirola, M. Van Heeswijk, E. SéVerin, and A. Lendasse, "Regularized extreme learning machine for regression with missing data," *Neurocomputing*, vol. 102, pp. 45–51, 2013.
- [15] Y. Míche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "Op-elm: optimally pruned extreme learning machine," *IEEE transactions on neural networks*, vol. 21, no. 1, pp. 158–162, 2010.
- [16] X. Luo, X. Yang, C. Jiang, and X. Ban, "Timeliness online regularized extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 3, pp. 465–476, 2018.
- [17] P. Souza and A. Lemos, "Redes neurais nebulosas para problemas de classificação," *Simpósio Brasileiro de Automação Inteligente, Natal-RN. XII SBAI-Simpósio Brasileiro de Automação Inteligente*, 2015.
- [18] P. V. C. Souza, "Regularized fuzzy neural networks for pattern classification problems," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 2985–2991, 2018.
- [19] P. V. de Campos Souza and P. F. A. de Oliveira, "Regularized fuzzy neural networks based on nullneurons for problems of classification of patterns," in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE, 2018, pp. 25–30.
- [20] P. V. de Campos Souza, G. R. L. Silva, and L. C. B. Torres, "Uninorm based regularized fuzzy neural networks," in *2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 2018, pp. 1–8.
- [21] A. J. Guimaraes, V. J. Araújo, L. de Oliveira Batista, P. V. C. Souza, V. Araújo, and T. S. Rezende, "Using fuzzy neural networks to improve prediction of expert systems for detection of breast cancer," in *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, vol. 1, no. XV. XV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), 2018, pp. 799–810.
- [22] P. V. de Campos Souza and A. J. Guimaraes, "Using fuzzy neural networks for improving the prediction of children with autism through mobile devices," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 01 086–01 089.
- [23] F. R. Bach, "Bolasso: model consistent lasso estimation through the bootstrap," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 33–40.
- [24] P. V. de Campos Souza and L. C. B. Torres, "Regularized fuzzy neural network based on or neuron for time series forecasting," in *North American Fuzzy Information Processing Society Annual Conference*. Springer, 2018, pp. 13–23.
- [25] P. V. de Campos Souza, A. J. Guimaraes, V. S. Araújo, T. S. Rezende, and V. J. S. Araújo, "Fuzzy neural networks based on fuzzy logic neurons regularized by resampling techniques and regularization theory for regression problems," *Inteligência Artificial*, vol. 21, no. 62, pp. 114–133, 2018.
- [26] A. J. Guimarães, V. J. Silva Araujo, P. V. de Campos Souza, V. S. Araujo, and T. S. Rezende, "Using fuzzy neural networks to the prediction of improvement in expert systems for treatment of immunotherapy," in *Advances in Artificial Intelligence - IBERAMIA 2018*, G. R. Simari, E. Fermé, F. Gutiérrez Segura, and J. A. Rodríguez Melquiades, Eds. Cham: Springer International Publishing, 2018, pp. 229–240.
- [27] P. V. de Campos Souza and A. J. Guimarães, "Detecção de pulsares utilizando redes neurais nebulosas baseadas em uninormas," in *Quinto Congresso Brasileiro de Sistemas Fuzzy*, vol. 1, no. 5. Sociedade Brasileira de Matemática Aplicada e Computacional, 2018, pp. 41–52.
- [28] L. O. Batista, G. A. de Silva, V. S. Araújo, V. J. S. Araújo, T. S. Rezende, A. Junio, and P. V. d. C. S. Guimarães, "Utilização de redes neurais nebulosas para criação de um sistema especialista em invasões cibernéticas," in *The Tenth International Conference on Forensic Computer Science And Cyber Law- Icofocs 2018*, Vol. 1, No. 10. Brasília Chapter Of The High Technology Crime Investigation Association (HTCIA), 2018, pp. 12–22.

- [29] P. Vitor de Campos Souza, "Pruning fuzzy neural networks based on unineuron for problems of classification of patterns," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–9, 2018.
- [30] J. Gao, Z. Wang, Y. Yang, W. Zhang, C. Tao, J. Guan, and N. Rao, "A novel approach for lie detection based on f-score and extreme learning machine," *PloS one*, vol. 8, no. 6, p. e64704, 2013.
- [31] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.
- [32] K. Bache and M. Lichman, "Uci machine learning repository," 2013.
- [33] T. K. Ho and E. M. Kleinberg, "Building projectable classifiers of arbitrary complexity," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 2. IEEE, 1996, pp. 880–885.
- [34] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.
- [35] D. C. Montgomery, *Design and analysis of experiments*. John wiley & sons, 2017.
- [36] J. Hsu, *Multiple comparisons: theory and methods*. Chapman and Hall/CRC, 1996.



Paulo Vitor de Campos Souza Formado em Sistemas de Informação pelo Centro Universitário UNA (2009), Especialista em Engenharia de Software pela UFMG (2010), Mestre em Engenharia Elétrica pela UFMG. Atua desde 2012 com desenvolvimento de softwares em java em órgãos públicos federais. Atualmente é analista de tecnologia da informação pelo CEFET-MG e professor Universitário no Centro Universitário UNA de Betim e no Centro Universitário UNI-BH em Belo Horizonte. Autor de dois livros, artigos em revistas e journals nacionais e internacionais, artigos publicados em anais de congressos nacionais e internacionais, atua como revisor de periódicos na área de inteligência artificial.