

# Pruning Extreme Learning Machines Using the Successive Projections Algorithm

D. P. P. Mesquita, J. P. P. Gomes, L. R. Rodrigues and R. K. H. Galvão

**Abstract**— Extreme Learning Machine (ELM) is a recently proposed machine learning method with successful applications in many domains. The key strengths of ELM are its simple formulation and the reduced number of hyper-parameters. Among these hyper-parameters, the number of hidden nodes has significant impact on ELM performance since too few/many hidden nodes may lead to underfitting/overfitting. In this work, we propose a pruning strategy for ELM using the Successive Projections Algorithm (SPA) as an approach to automatically find the number of hidden nodes. The SPA was originally proposed for variable selection. In this work, it is adapted in order to prune ELMs. The proposed method is compared to the Optimally Pruned Extreme Learning Machine algorithm (OP-ELM), which is considered as a state of the art method. Real world datasets were used to assess the performance of the proposed method for regression and classification problems. The application of the proposed model resulted in much simpler models with similar performance compared to the OP-ELM. For some classification instances, the performance of the proposed method outperformed the OP-ELM method.

**Keywords**— Feedforward Networks, Extreme Learning Machine, Successive Projections Algorithm.

## I. INTRODUÇÃO

MÁQUINAS de Aprendizado Extremo (*Extreme Learning Machines*, ELM [1]) são redes neurais supervisionadas cujas principais vantagens são a simplicidade e a facilidade de implementação. Como consequência destas características, ELMs têm sido empregadas com sucesso em diversos problemas reais [2], [3] e [4].

A determinação do número de neurônios na camada oculta de uma ELM consiste em um dos poucos hiper-parâmetros ajustáveis nestes modelos. A utilização de poucos neurônios pode prejudicar a capacidade da rede em aprender de acordo com os dados de treinamento. Por outro lado, ao utilizar muitos neurônios, pode ocorrer um sobre ajuste do modelo levando ao problema de *overfitting*.

Desta forma, diversas estratégias têm sido empregadas para determinar o número ideal de neurônios na camada oculta em redes neurais. Dentre estas estratégias, algoritmos de poda são bastante populares, conforme pode ser visto nos seguintes trabalhos [5], [6] e [7]. A estratégia de poda consiste em utilizar uma rede com muitos neurônios e identificar, de acordo com algum critério, quais destes neurônios são

importantes para manter a capacidade de generalização da rede.

Apesar da existência de diversos estudos relacionados à poda de redes neurais, poucos trabalhos sobre ELMs exploram este problema. Dentre os trabalhos recentes, destaca-se o algoritmo *Optimally Pruned ELM* (OP-ELM, [5]). O OP-ELM utiliza um critério baseado em *leave-one-out* (LOO) para escolha dos neurônios e isto levou a resultados comparáveis a métodos considerados estado da arte em classificação e regressão com reduzido custo computacional.

Este trabalho propõe um novo método de poda para ELM utilizando o Algoritmo das Projeções Sucessivas (APS, [8]). O APS é um método de seleção de atributos, extensamente aplicado em problemas de quimiometria [8], [9] e [10]. O método utiliza critérios simples e possui baixo custo computacional. No presente trabalho, o APS foi adaptado para funcionar como algoritmo de poda ao invés de seleção de atributos.

O método proposto, denominado APS-ELM foi comparado com o OP-ELM em diversos problemas de classificação e regressão. O desempenho do APS-ELM em termos de erro quadrático médio e taxa de classificação, em problemas de regressão e classificação, respectivamente, mostrou-se comparável ao OP-ELM. Entretanto, foi observada uma significativa redução no número de neurônios escolhidos. O APS-ELM foi capaz de reduzir, em média, o número de neurônios em 50%.

As demais seções deste artigo estão organizadas conforme descrito a seguir. A seção II descreve o método de ELM. A seção III apresenta o Algoritmo de Projeções Sucessivas (APS). A seção IV descreve o método proposto. A seção V apresenta os resultados obtidos na aplicação do modelo proposto para diferentes bases de dados. As conclusões do trabalho são apresentadas na seção VI.

## II. MÁQUINAS DE APRENDIZADO EXTREMO

O método ELM foi originalmente proposto em [1] e utiliza como modelo uma rede neural *feedforward* de uma única camada oculta. O principal conceito empregado em ELM consiste em utilizar valores aleatórios para os pesos e constantes de polarização de todos os neurônios da camada oculta da rede. Os neurônios da camada de saída podem, então, ser ajustados utilizando o método dos mínimos quadrados. A capacidade de aproximação universal é garantida para a escolha aleatória dos pesos e constantes de polarização, desde que a função de ativação seja infinitamente diferenciável.

D. P. P. Mesquita, Universidade Federal do Ceará (UFC), Fortaleza, Ceará, Brasil, diego@diegoparente.com

J. P. P. Gomes, Universidade Federal do Ceará (UFC), Fortaleza, Ceará, Brasil, jpaulo@lia.ufc.br

L. R. Rodrigues, Instituto de Aeronáutica e Espaço (IAE), São José dos Campos, São Paulo, Brasil, leonardolrr@iae.cta.br

R. K. H. Galvão, Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, São Paulo, Brasil, kawakami@ita.br

Seja  $S = \{(x_i, y_i) : 1 \leq i \leq N\}$  o conjunto de treinamento, com  $x_i \in \mathbf{R}^l$ ,  $y_i \in \mathbf{R}^k$ . Sejam  $f: \mathbf{R} \rightarrow \mathbf{R}$ , a função de ativação e  $H$  o número de neurônios na camada oculta. É possível descrever as respostas do ELM com relação ao estímulo  $x_i$  como:

$$\sum_{i=1}^N \beta_i f(w_i x_i + b_i) \quad (1)$$

onde  $w$  denota o peso da camada de entrada,  $\beta$  denota os pesos da camada oculta e as constantes de polarização são representadas como  $b_i$ . É possível reescrever a Eq.(1) como  $A\beta$  utilizando a seguinte abreviatura:

$$A_{(i,j)} = f(w_j x_i + b_j) | 1 \leq i \leq N, 1 \leq j \leq H \quad (2)$$

Decidir os pesos de ativação da camada oculta de uma ELM, então, consiste em aplicar o método dos mínimos quadrados para encontrar uma solução aproximada para o sistema linear  $A\beta = Y$ , no qual  $Y \in \mathbf{R}^{N \times k}$  tal que a  $i$ -ésima linha de  $Y$  corresponde à  $y_i$ . A aproximação  $\hat{\beta}$  de mínimos quadrados para o valor de  $\beta$  pode ser escrita como:

$$\hat{\beta} = (A^T A)^{-1} A^T Y \quad (3)$$

### III. ALGORITMO DAS PROJEÇÕES SUCESSIVAS (APS)

O APS, conforme originalmente proposto em 2001 por Araújo et al.[8], é uma técnica de seleção de variáveis cujo objetivo é selecionar um subconjunto de variáveis com baixa colinearidade que contenha informação suficiente para a construção de um modelo de Regressão Linear Múltipla (RLM) por mínimos quadrados [8] e [9].

Para a aplicação do APS, os dados devem ser divididos em dois subconjuntos: um subconjunto de calibração e um subconjunto de validação, contendo  $N_C$  e  $N_V$  observações, respectivamente.

A aplicação do APS é composta por três fases principais: geração das cadeias de variáveis, avaliação dos subconjuntos e eliminação de variáveis. As operações realizadas em cada uma das fases são brevemente descritas a seguir.

#### A. Fase 1: Geração das Cadeias de Variáveis

A primeira fase do APS consiste em gerar várias cadeias de variáveis. A seleção das variáveis que compõem cada uma das cadeias geradas é feita de forma sequencial. Cada variável é escolhida de modo a se obter a menor colinearidade em relação ao conjunto de variáveis já escolhidas. Para isso, em cada iteração, escolhe-se a variável com a maior projeção no

subespaço ortogonal ao das variáveis que já compõem a cadeia.

A geração das cadeias de variáveis é feita usando apenas os dados de calibração. A descrição detalhada da sequência de passos a ser seguida na geração das cadeias de variáveis pode ser encontrada em [9].

A Fig. 1 ilustra, de forma gráfica, as operações realizadas na primeira iteração da primeira fase do APS para obtenção de uma cadeia de variáveis. Neste exemplo, cada observação é composta por 4 variáveis, e o conjunto de calibração possui 3 observações.

Neste exemplo, a sequência das projeções é inicializada com  $\mathbf{z}^1 = \mathbf{x}_3^1 = \mathbf{x}_3$ , onde  $\mathbf{x}_k^1 = \mathbf{x}_k$  é um vetor coluna contendo as  $N_C$  observações da  $k$ -ésima variável independente  $x_k$  presentes no conjunto de calibração e  $\mathbf{z}^m$  é o vetor de referência para o cálculo das projeções da  $m$ -ésima iteração. Para  $m \geq 1$ ,  $\mathbf{x}_k^{m+1}$  é a projeção do vetor  $\mathbf{x}_k^m$  no subespaço ortogonal ao vetor de referência  $\mathbf{z}^m$ . As projeções dos vetores são calculadas conforme a Eq.(4).

$$\mathbf{x}_k^{m+1} = \mathbf{P}^m \mathbf{x}_k^m \quad (4)$$

sendo  $\mathbf{P}^m$  a matriz de projeção no subespaço ortogonal a  $\mathbf{z}^m$ , definida conforme a Eq.(5).

$$\mathbf{P}^m = \mathbf{I} - \frac{\mathbf{z}^m (\mathbf{z}^m)^T}{(\mathbf{z}^m)^T \mathbf{z}^m} \quad (5)$$

onde  $\mathbf{I}$  é uma matriz identidade de ordem  $N_C$ .

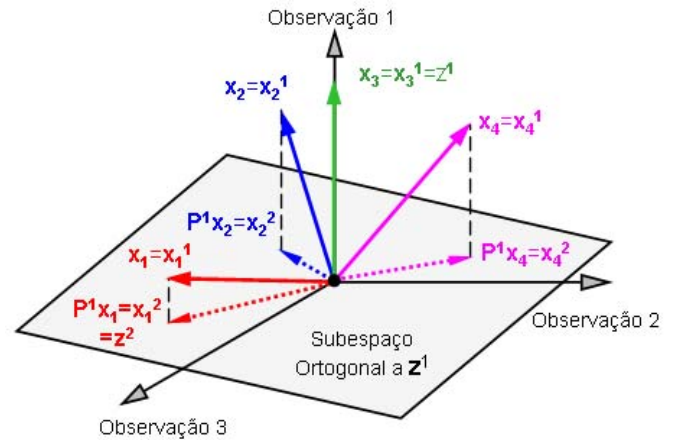


Figura 1. Exemplo gráfico das operações realizadas na primeira iteração da primeira fase do APS.

Conforme ilustrado na Fig. 1, a maior projeção obtida foi a projeção do vetor  $\mathbf{x}_1$  e, portanto, na próxima iteração,  $\mathbf{z}^2 = \mathbf{x}_1^2$  será o novo vetor de referência para o cálculo das projeções. A representação gráfica da segunda iteração deste exemplo é apresentada na Fig. 2.

Observa-se que, nesta segunda iteração, a maior projeção obtida foi a projeção correspondente ao vetor  $\mathbf{x}_2$ . A primeira fase do APS termina nessa iteração, uma vez que as próximas projeções resultariam em vetores de norma nula. Desse modo, a cadeia encontrada ao iniciar o APS com a variável  $\mathbf{x}_3$  neste exemplo é  $\{\mathbf{x}_3, \mathbf{x}_1, \mathbf{x}_2\}$ .

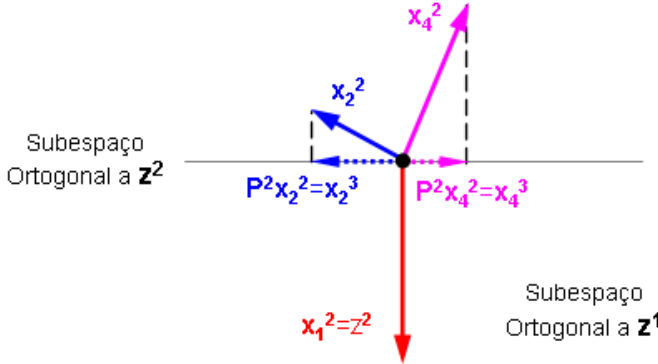


Figura 2. Exemplo gráfico das operações realizadas na segunda iteração da primeira fase do APS.

Alterando-se a variável empregada na inicialização do APS, bem como o número de iterações de projeção, podem-se obter diferentes cadeias de variáveis. A melhor cadeia é escolhida com base na capacidade de previsão do modelo RLM resultante, conforme descrito a seguir.

#### B. Fase 2: Avaliação das Cadeias de Variáveis

A segunda fase do APS consiste em avaliar as cadeias de variáveis geradas na primeira fase, de acordo com a Soma Quadrática dos Erros de Previsão (*Prediction Error Sum of Squares - PRESS*) do modelo RLM correspondente. Para isso, são usados os dados de validação. O PRESS é calculado conforme a Eq.(6).

$$PRESS = \sum_{i=1}^{N_V} (y_{v,i} - \hat{y}_{v,i})^2 \quad (6)$$

onde  $y_{v,i}$  é o valor da variável independente para a  $i$ -ésima observação do conjunto de validação e  $\hat{y}_{v,i}$  é a estimativa correspondente obtida pelo modelo RLM. O objetivo desta fase é identificar qual das cadeias geradas na fase 1 possui o menor PRESS.

#### C. Fase 3: Eliminação de Variáveis

A terceira fase do APS consiste em um procedimento de eliminação de variáveis. Nesta fase verifica-se, através de um teste estatístico, se é possível eliminar alguma variável sem comprometer significativamente o erro do modelo RLM.

Nesta fase, associa-se a cada variável selecionada na fase 2 um fator de relevância definido pelo produto entre o desvio

padrão e o módulo do coeficiente de regressão da variável em questão [9].

Posteriormente, são construídos modelos RLM que incluem progressivamente as variáveis em ordem decrescente dos fatores de relevância. Para cada nova variável incluída, calcula-se o valor de PRESS, usando o conjunto de validação.

Em seguida, determina-se o menor número de variáveis para o qual o valor de PRESS não é significativamente maior do que o mínimo valor obtido. O teste estatístico F é comumente empregado na escolha do modelo final.

### IV. MÉTODO PROPOSTO

Conforme dito anteriormente, o algoritmo das projeções sucessivas foi proposto originalmente para a tarefa de seleção de atributos. Neste trabalho foi realizada a adaptação do APS para a tarefa de poda em redes ELM.

De maneira semelhante a diversos outros métodos não lineares para classificação e regressão, a rede ELM pode ser vista como um processo em duas etapas.

A etapa inicial consiste em projetar o dado original em um espaço de dimensão maior através de uma projeção não linear. Esta etapa tem como objetivo transformar problemas de classificação não linearmente separáveis em linearmente separáveis ou problemas de regressão não linear em problemas de regressão linear.

Para a rede ELM, esta primeira etapa é realizada pela transformação não linear presente na camada oculta da rede. Desta forma, após a transformação não linear, temos um vetor de saída da camada oculta associado a cada vetor de entrada presente na rede.

A segunda etapa utiliza um método linear para classificação ou regressão. Apesar da utilização de um método linear, o modelo final apresenta comportamento não linear devido à transformação ocorrida na primeira etapa.

Tendo como base a idéia de que a segunda etapa da ELM consiste em um método de regressão/classificação linear, pode-se utilizar o APS como método de seleção de atributos. Neste caso, a seleção de atributos irá ocorrer no espaço projetado não linearmente na primeira etapa. Consequentemente, o APS irá funcionar como método para seleção dos neurônios da camada oculta.

### V. AVALIAÇÃO DE DESEMPENHO

Nesta seção, avalia-se o desempenho do método proposto para poda de neurônios da camada oculta de ELMs. Para isso, compara-se o desempenho do método ASP-ELM com o desempenho do método OP-ELM, considerado o estado da arte para poda de ELMs.

A implementação do OP-ELM usada durante os experimentos foi a fornecida em [11]. Para ambos os métodos, em todos os experimentos o número máximo de neurônios na camada interna foi fixado em 100. Durante a fase de seleção de variáveis do APS, apenas as 25 variáveis mais promissoras foram consideradas para escolha inicial. Para avaliar a qualidade das variáveis selecionadas foi adotado o método 10-fold cross-validation ao invés do método LOO.

### A. Regressão

Foram conduzidos testes de regressão utilizando-se 8 diferentes bancos de dados de aplicações reais. Todos os bancos de dados utilizados estão disponíveis na base de dados da UCI (*University of California, Irvine*) disponível em <https://archive.ics.uci.edu/ml/datasets>.

A Tabela I mostra a comparação entre os resultados obtidos com o método proposto e com o método OP-ELM. Os resultados da Tabela I refletem séries de 10 eventos similares. A Tabela II apresenta a quantidade de neurônios gerados por ambos os métodos para cada um dos 10 eventos (*folds*) empregados na avaliação de cada banco de dados.

Os resultados apresentados na Tabela I mostram que o método OP-ELM apresenta AMSE inferior ao método APS-ELM para todos os bancos de dados considerados. No entanto, a Tabela II mostra que o método proposto realizou podas significativamente maiores que as realizadas pelo método OP-ELM. Por exemplo, para o conjunto de dados de qualidade de vinhos brancos, enquanto o OP-ELM reduziu a quantidade de neurônios na camada interna de 100 para 91 (no pior dos casos), o APS-ELM manteve no máximo 30 dos neurônios iniciais da camada interna.

TABELA I. COMPARAÇÃO DOS RESULTADOS PARA PROBLEMAS DE REGRESSÃO.

BASE DE DADOS	OP-ELM		APS-ELM	
	AMSE	DESVIO	AMSE	DESVIO
White Wine	0,548	0,056	0,569	0,030
Red Wine	0,433	0,039	0,443	0,044
Parkinson	14,6E-05	6,6E-05	17,1E-05	5,9E-05
Stocks	1,039	0,136	1,20	0,29
Forest-Fire	155.995	477.360	4.874	3.432
Breast-Cancer	1.262	223	1.431	1.917
Boston Housing	18,01	4,26	24,0	5,05
Compression	55,34	7,43	67,7	11,2

### B. Classificação

Para avaliar o desempenho do método proposto em problemas de classificação, 13 bancos de dados provenientes de aplicações reais foram adotados. Os bancos de dados diferem amplamente em natureza, tamanho e número de classes. Dos 13 bancos de dados adotados, 12 estão disponíveis na base de dados do UCI. O último conjunto de dados consiste em um problema de reconhecimento de faces utilizando a base de dados conhecida como Yale Faces. Para esta base de dados foram extraídos vetores de 20 atributos utilizando Análise de Componentes Principais (*Principal Component Analysis*, PCA).

Foi adotado um esquema 1-de-S para codificação da saída dos bancos de dados adotados. Para cada uma das S classes, uma ELM é criada para classificação binária. A Tabela III mostra a quantidade média de neurônios mantida por cada um dos algoritmos para cada evento para cada banco de dados. Por exemplo, para a base de dados Yale Faces, que contém 15 diferentes classes, na décima rodada de experimentos, a média de neurônios na camada interna dos 15 classificadores utilizando o método APS-ELM é 21.9. A Tabela IV mostra a comparação entre os resultados obtidos com o método proposto e com o método OP-ELM. Assim como nos testes envolvendo problemas de regressão, os resultados refletem uma série de 10 eventos similares.

É possível observar que, assim como nos experimentos de regressão, o número de neurônios mantidos pelo método APS-ELM é bastante inferior ao número de neurônios mantido pelo OP-ELM. No entanto, ao contrário do observado nos experimentos de regressão, em 9 dos 13 casos considerados o APS-ELM supera a taxa de acerto obtida com o método OP-ELM.

TABELA II. NÚMERO DE NEURÔNIOS NA CAMADA OCULTA (APS-ELM / OP-ELM).

BASE DE DADOS	ROUND 1	ROUND 2	ROUND 3	ROUND 4	ROUND 5	ROUND 6	ROUND 7	ROUND 8	ROUND 9	ROUND 10
White Wine	19 / 81	19 / 76	16 / 81	26 / 81	12 / 86	12 / 91	17 / 91	30 / 81	14 / 71	22 / 66
Red Wine	12 / 31	12 / 21	12 / 26	12 / 31	12 / 26	12 / 41	12 / 26	12 / 36	12 / 16	12 / 36
Parkinson	36 / 91	40 / 96	43 / 91	40 / 91	31 / 96	49 / 91	52 / 91	38 / 91	41 / 86	37 / 91
Stocks	19 / 79	75 / 79	10 / 74	33 / 74	26 / 79	10 / 79	10 / 89	10 / 89	14 / 74	33 / 79
Forest Fire	63 / 1	38 / 89	48 / 1	54 / 3	45 / 14	40 / 1	35 / 4	38 / 89	58 / 1	50 / 1
Breast- Cancer	5 / 27	5 / 19	5 / 8	5 / 4	5 / 7	5 / 31	5 / 5	5 / 12	5 / 10	5 / 4
Boston Housing	33 / 38	33 / 48	33 / 83	33 / 43	33 / 83	33 / 83	33 / 83	33 / 43	33 / 13	33 / 53
Compression	58 / 98	20 / 88	34 / 98	14 / 98	48 / 98	16 / 98	27 / 98	27 / 83	29 / 93	14 / 93

TABELA III. NÚMERO DE NEURÔNIOS NA CAMADA OCULTA PARA PROBLEMAS DE CLASSIFICAÇÃO (APS-ELM / OP-ELM).

BASE DE DADOS	ROUND 1	ROUND 2	ROUND 3	ROUND 4	ROUND 5	ROUND 6	ROUND 7	ROUND 8	ROUND 9	ROUND 10
Waveform	31 / 91	24.3 / 87.7	22.3 / 91	22 / 91	26.3 / 91	26.7 / 96	22 / 96	23.7 / 96	24 / 92.7	22.3 / 86
Spam Base	58 / 82	58 / 82	58 / 82	58 / 92	58 / 87	58 / 92	58 / 56	58 / 67	58 / 87	58 / 87
Monk3	10.5 / 46	7 / 26	9 / 11	8 / 26	15.5 / 81	10.5 / 81	8.5 / 21	16 / 31	8.5 / 31	7.5 / 71
Monk2	7 / 21	7 / 16	7 / 16	7 / 21	7 / 21	7 / 11	7.5 / 21	9 / 21	8.5 / 21	7 / 26
Monk1	7 / 26	15 / 16	7.5 / 16	10.5 / 31	8.5 / 16	10.5 / 16	15.5 / 41	7 / 26	7 / 26	15 / 4
Liver	7 / 16	7 / 21	7 / 16	7 / 11	7.5 / 26	7 / 11	7 / 16	8 / 21	7 / 21	7 / 21
Hayes	12.7 / 33	15 / 18	10 / 21.3	13 / 43	17.3 / 53	19 / 28	14.7 / 11.3	15 / 23	16.3 / 36.3	13.3 / 23
Haberman	4 / 8	4 / 8	5 / 13	4 / 8	4 / 1	4 / 1	4 / 18	4 / 2	4.5 / 2	4 / 3
Diabetes	9 / 5	9 / 18	9 / 8	9 / 18	9 / 13	9.5 / 18	9 / 18	9 / 13	9 / 8	9 / 33
Column	10.5 / 11	7 / 16	8 / 11	7 / 6	8 / 26	7 / 26	7 / 21	7 / 26	7 / 26	7 / 31
Breast	31 / 26	31 / 10	31 / 26	31 / 21	31 / 29	31 / 19	31 / 25	31 / 95	31 / 35	31 / 20
Balance	5 / 24	6.7 / 89	11 / 29	16.3 / 59	9 / 52.3	7.3 / 89	8 / 34	11.7 / 55.7	9.3 / 50.7	14 / 65.7
Yale Faces	22.3 / 35.3	24.3 / 42.7	21.6 / 40.3	22.3 / 37.3	25.3 / 46.3	21.8 / 41.3	24 / 38.7	22.7 / 37.7	27 / 38.6	21.9 / 65

TABELA IV. COMPARAÇÃO DOS RESULTADOS PARA PROBLEMAS DE CLASSIFICAÇÃO.

BASE DE DADOS	OP-ELM		APS-ELM	
	TAXA MÉDIA DE ACERTO	DESVIO	TAXA MÉDIA DE ACERTO	DESVIO
Waveform	0,859	0,0047	0,833	0,0093
Spam Base	0,901	0,0051	0,919	0,0062
Monk3	0,858	0,0249	0,871	0,0254
Monk2	0,699	0,0226	0,675	0,0146
Monk1	0,701	0,0291	0,712	0,0395
Liver	0,643	0,0363	0,677	0,0334
Hayes	0,839	0,0811	0,814	0,0527
Haberman	0,736	0,0328	0,750	0,0321
Diabetes	0,744	0,0200	0,755	0,0298
Column	0,839	0,0304	0,859	0,0307
Breast	0,951	0,0094	0,966	0,0102
Balance	0,884	0,0190	0,894	0,0151
Yale Faces	0,823	0,0994	0,740	0,1119

## VI. CONCLUSÕES

O presente trabalho apresenta um novo método de poda para redes ELM utilizando o Algoritmo das Projeções Sucessivas (APS). O APS, originalmente proposto para seleção de atributos, é utilizado como método para seleção de neurônios da camada oculta da rede ELM. O método proposto foi denominado APS-ELM.

Foram realizados diversos testes utilizando bases de dados reais em problemas de classificação e regressão. O APS-ELM foi comparado com o OP-ELM, tido como estado da arte para poda em redes ELM e apresentou resultados promissores.

Em comparação ao OP-ELM, o método proposto teve, em geral, melhores resultados em tarefas de classificação e

resultados piores em tarefas de regressão. Entretanto, em todos os casos o APS-ELM proporcionou maiores níveis de poda nas redes, resultado assim em modelo menos complexos.

Futuros trabalhos poderão abordar diferentes formas de inicialização do APS bem como a sua utilização para poda de outras redes neurais *feedforward*.

## AGRADECIMENTOS

Os autores gostariam de agradecer ao CNPq (processo 456837/2014-0 e produtividade em pesquisa) e à FAPESP (processo 2011/17610-0) pelo apoio financeiro.

## REFERÊNCIAS

- [1] G.-B. Huang, Q.-Y. Zhu and C.K. Siew "Extreme Learning Machine: theory and applications", *Neurocomputing*, Vol. 70, p. 489-501, 2006.
- [2] N.-Y. Liang, P. Saratchandran, G.-B. Huang and N. Sundararajan "Classification of mental tasks from EEG signals using extreme learning machine" *International Journal of Neural Systems*, Vol. 16, p. 29-38, 2006.
- [3] Y. Bazi, N. Alajlan, F. Melgani, H. AlHichri, S. Malek and R.R. Yager, "Differential Evolution Extreme Learning Machine for the Classification of Hyperspectral Images," *IEEE Geoscience and Remote Sensing Letters*, vol.11, no.6, pp.1066,1070, June 2014.
- [4] Runxuan Zhang, Guang-Bin Huang, N. Sundararajan and P. Saratchandran, "Multicategory Classification Using An Extreme Learning Machine for Microarray Gene Expression Cancer Diagnosis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.4, no.3, pp.485,495, July-Sept. 2007.
- [5] Yoan Miche, 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, Jan. 2010.
- [6] C. L. Giles and C. W. Omlin, "Pruning recurrent neural networks for improved generalization performance," *IEEE Transactions on Neural Networks*, vol.5, no.5, pp.848,851, Sep 1994.
- [7] G. Castellano, A. M. Fanelli and M. Pelillo, "An iterative pruning algorithm for feedforward neural networks," *IEEE Transactions on Neural Networks*, vol.8, no.3, pp.519,531, May 1997.
- [8] M. C. U. Araújo, T. C. B. Saldanha, R. K. H. Galvão, T. Yoneyama, H. C. Chame and V. Visani, "The Successive Projections Algorithm for Variable Selection in Spectroscopic Multicomponent", *Chemometrics and Intelligent Laboratory Systems*, Vol. 57, No. 2, p. 65-73, 2001.

- [9] R. K. H. Galvão, M. C. U. Araújo, W. D. Fragoso, E. C. Silva, G. E. José, S. F. C. Soares and H. M. Paiva, “A Variable Elimination Method to Improve the Parsimony of MLR Models Using the Successive Projections Algorithm”, *Chemometrics and Intelligent Laboratory Systems*, Vol. 92, No. 1, p. 83–91, 2008.
- [10] S. F. C. Soares, A. A. Gomes, A. R. Galvão Filho, M. C. U. Araújo and R. K. H. Galvão, “The successive projections algorithm”, *Trends in Analytical Chemistry*, Vol. 42, p. 84-98, 2013.
- [11] Yoan Miche, Antti Sorjamaa, Amaury Lendasse “OP-ELM: Theory, Experiments and a Toolbox”, *Proceedings of the 18th International Conference on Artificial Neural Networks*, 2008.



**Diego Parente Paiva Mesquita** é graduando em computação pela Universidade Federal do Ceará, com período de graduação sanduíche na University of Alberta. O mesmo tem interesse nas áreas de aprendizado de máquina e visão computacional.



**João Paulo Pordeus Gomes** possui graduação em engenharia elétrica pela Universidade Federal do Ceará (2004), mestrado em engenharia aeronáutica pelo Instituto Tecnológico de Aeronáutica (2006) e doutorado em engenharia eletrônica e computação (2011) pela mesma universidade. Trabalhou no período entre 2006 e 2013 como engenheiro de desenvolvimento tecnológico na EMBRAER S.A e atualmente é professor adjunto do departamento de computação da Universidade Federal do Ceará. Tem interesse nos seguintes temas: aprendizado de máquina, sistemas inteligentes e monitoramento da saúde de sistemas.



**Leonardo Ramos Rodrigues** possui graduação em engenharia elétrica pela Universidade Federal do Espírito Santo – UFES (2003), mestrado em engenharia aeronáutica pelo Instituto Tecnológico de Aeronáutica – ITA (2008) e doutorado em engenharia aeronáutica (2013) pela mesma Universidade. Trabalhou no período entre 2006 e 2015 como engenheiro de desenvolvimento tecnológico na EMBRAER S.A e atualmente é tecnólogo pleno da Divisão de Eletrônica do Instituto de Aeronáutica e Espaço (IAE). Tem interesse nos seguintes temas: otimização combinatória, pesquisa operacional, aprendizado de máquina e monitoramento da saúde de sistemas.



**Roberto Kawakami Harrop Galvão** possui graduação em Engenharia Eletrônica (1995, com distinção *Summa cum Laude*) pelo Instituto Tecnológico de Aeronáutica (ITA), onde também obteve o mestrado (1997) e o doutorado (1999) na área de Sistemas de Controle. Desde 1998, é professor da Divisão de Engenharia Eletrônica do ITA. Sua área de pesquisa inclui diagnóstico e prognóstico de falhas, teoria e aplicações de wavelets, e controle preditivo baseado em modelo.