

# Proposta de uma Rede Neural Artificial para Detecção de Ruído Impulsivo

Bruna de Souza Ribeiro e Rausley Adriano Amaral de Souza

**Resumo**— Este artigo tem como objetivo apresentar uma rede neural artificial para detectar amostras de um sinal contaminadas por ruído impulsivo. Com resultado dos testes realizados, é possível concluir sobre o bom desempenho do modelo proposto, sendo próximo ou superior ao algoritmo usado em comparação.

**Palavras-Chave**— Ruído impulsivo, inteligência artificial, redes neurais, distribuição  $S\alpha S$ .

**Abstract**— This paper aims to present an artificial neural network to detect signal samples contaminated by impulsive noise. With the result of the tests presented, it is possible to identify the good performance of the proposed model, being close or superior to the algorithm used in comparison.

**Keywords**— Impulsive noise, artificial intelligence, neural network,  $S\alpha S$  distribution.

## I. INTRODUÇÃO

Em diferentes sistemas de comunicação é comum encontrar a presença de amostras indesejadas, ou como também são conhecidas, os ruídos [1]. A distribuição dessas amostras é, por vezes, modelada por uma função Gaussiana [1], mas neste trabalho será dado destaque a um outro tipo, o chamado ruído impulsivo (RI). O RI é caracterizado pela ocorrência de amostras com alta amplitude em relação à média do sinal, ocorre por algum tipo de interferência humana ou por fenômenos naturais e pode ser modelado por diferentes distribuições [1]. Uma delas é a  $\alpha$ -estável simétrica (*symmetric  $\alpha$ -stable*,  $S\alpha S$ ), a distribuição adotada neste artigo.

É notável que, em se tratando da recepção e transmissão de dados, esse tipo de interferência atua de forma negativa, e nesse sentido, há uma diversidade de algoritmos e técnicas para a sua mitigação, por meio de estatísticas como o desvio padrão [1] ou com o uso da inteligência artificial [2]. Esta última vem ganhando destaque no campo das comunicações devido à sua capacidade de solucionar problemas complexos nos chamados modelos fim-a-fim [2]. Diante disso, este trabalho apresenta o desenvolvimento e a implementação de uma rede neural artificial para a detecção de amostras com RI.

## II. RUÍDO IMPULSIVO E REDE NEURAL ARTIFICIAL

A distribuição  $S\alpha S$  é um caso particular da  $\alpha$ -estável quando o parâmetro de simetria  $\beta = 0$  [1]. Seus três outros parâmetros são: o expoente característico  $\alpha$ ,  $0 < \alpha \leq 2$ , cuja redução implica em uma maior impulsividade; o parâmetro de escala ou dispersão  $\gamma > 0$ , que controla a intensidade do ruído e o parâmetro de localização  $\lambda$ ,  $-\infty < \lambda < \infty$ , que refere-se

Bruna de Souza Ribeiro e Rausley A. A. de Souza, Instituto Nacional de Telecomunicações (Inatel), Santa Rita do Sapucaí, MG, e-mail: bruna.br@gea.inatel.br, rausley@inatel.br.

à mediana quando  $0 < \alpha \leq 1$  e à média quando  $1 < \alpha \leq 2$ . Por não possuir uma forma fechada para a função densidade de probabilidade [1], é possível descrevê-la por meio de sua função característica, dada por  $\varphi(t) = \exp(j\lambda t - \gamma|t|^\alpha[1 + j\beta \text{sign}(t)\omega(t, \alpha)])$ , em que  $\omega(t, \alpha) = \tan(\alpha\pi/2)$  para  $\alpha \neq 1$  e  $\omega(t, \alpha) = (2/\pi) \log|t|$  para  $\alpha = 1$  e  $\text{sign}(\cdot)$  é função sinal. Vale notar que para  $\alpha = 2$ , tem-se a distribuição Gaussiana.

Tanto o conjunto de treinamento quanto os dados utilizados para os testes do modelo são descritos por  $y_i = w_i + b_i v_i$ , em que  $y_i$  é a  $i$ -ésima amostra a ser classificada como impulsiva ou não impulsiva, ou ruidosa ou não ruidosa,  $i = 1, 2, \dots, N$ ,  $w_i$  a  $i$ -ésima amostra de ruído aditivo Gaussiano Branco com média zero e variância  $\sigma_w^2$ ,  $b_i$  é uma variável aleatória de Bernoulli com probabilidade de sucesso  $p_{RI}$  e  $v_i$  é  $i$ -ésima amostra i.i.d. de RI proveniente de uma distribuição  $S\alpha S$  com parâmetros  $\lambda = 0$ ,  $\gamma = 1$  e diferentes valores de  $\alpha$ . A variável  $b_i$  determina, portanto, se a amostra a ser classificada  $y_i$  está corrompida ( $b_i = 1$ ), com probabilidade  $p_{RI}$ , ou não corrompida ( $b_i = 0$ ) com RI, desse forma é a saída (*label*) desejada do modelo. Conclui-se que a rede neural é do tipo classificador binário.

Os três tipos de dados de entrada para a classificação da amostra pela rede neural são: o valor da amostra  $y_i$ , o desvio absoluto mediano (MAD) [3] e a estatística *Rank Ordered Absolute Deviation* (ROAD) [4]. A estatística MAD é uma medida estatística robusta à presença de *outliers* se comparada a outras medidas como a média e o desvio padrão, comumente usados para a detecção de ruídos [3]. Essa estatística é definida por  $MAD = \text{median}(|y_i - M|)$ , em que  $y_i$  é o valor da  $i$ -ésima amostra e  $M$  é a mediana do conjunto de amostras. A estatística ROAD mede a correlação de uma amostra com seus vizinhos mais próximos. Se a amostra possui amplitude muito diferente de seus vizinhos, o valor ROAD será grande, caso contrário, tem-se um valor pequeno. O cálculo dessa estatística pode ser encontrado em [2, Section III.B].

A rede neural proposta foi treinada a fim identificar uma amostra corrompida a partir do valor da amostra, da estatística MAD e da estatística ROAD, conforme anteriormente definido. A rede construída possui 3 neurônios na camada de entrada, duas camadas com 21 neurônios cada e apenas 1 neurônio na saída [2]. A função de ativação da primeira camada intermediária é a tangente hiperbólica, a segunda camada possui uma função linear do tipo  $f(x) = x$  e a terceira camada, a função sigmoid. As escolhas da quantidade de neurônios e das funções de ativação foram feitas baseadas em [2] e por meio do treinamento de diferentes estruturas para se encontrar a estrutura com melhor desempenho. A função de ativação da última camada foi escolhida por apresentar em

TABELA I  
ACURÁCIA(ACUR.), *Recall* E PRECISÃO(PREC.) DOS TESTES COM O ALGORITMO CITADO E O MODELO PROPOSTO.

	$p_{RI}$ Métrica	0			0.2			0.5			0.8		
		Acur.	<i>Recall</i>	Prec.	Acur.	<i>Recall</i>	Prec.	Acur.	<i>Recall</i>	Prec.	Acur.	<i>Recall</i>	Prec.
$\alpha = 2.0$	Algoritmo [1]	<b>1</b>	-	-	0.8021	0.0004	<b>1</b>	0.4971	0.0004	<b>1</b>	0.2008	0.0001	<b>1</b>
	Rede neural	0.9994	-	-	<b>0.8025</b>	<b>0.007</b>	0.6039	<b>0.5044</b>	<b>0.0176</b>	0.8622	<b>0.2263</b>	<b>0.035</b>	0.9215
$\alpha = 1.4$	Algoritmo [1]	<b>1</b>	-	-	0.8025	0.0277	<b>1</b>	0.5089	0.0194	<b>1</b>	0.2089	0.0149	<b>1</b>
	Rede neural	0.9996	-	-	<b>0.8051</b>	<b>0.0465</b>	0.8864	<b>0.531</b>	<b>0.0711</b>	0.9045	<b>0.2693</b>	<b>0.0933</b>	0.9672
$\alpha = 0.8$	Algoritmo [1]	<b>1</b>	-	-	0.8197	0.1029	<b>1</b>	0.5316	0.0627	<b>1</b>	0.2207	0.0269	<b>1</b>
	Rede neural	0.9998	-	-	<b>0.8293</b>	<b>0.1568</b>	0.9642	<b>0.5992</b>	<b>0.2108</b>	0.9425	<b>0.3664</b>	<b>0.2127</b>	0.9868
$\alpha = 0.2$	Algoritmo [1]	<b>1</b>	-	-	0.8288	0.1411	<b>1</b>	0.5208	0.0468	<b>1</b>	0.2147	0.0214	<b>1</b>
	Rede neural	0.9996	-	-	<b>0.8909</b>	<b>0.4801</b>	0.9453	<b>0.7379</b>	<b>0.5011</b>	0.9572	<b>0.5774</b>	<b>0.4871</b>	0.9727

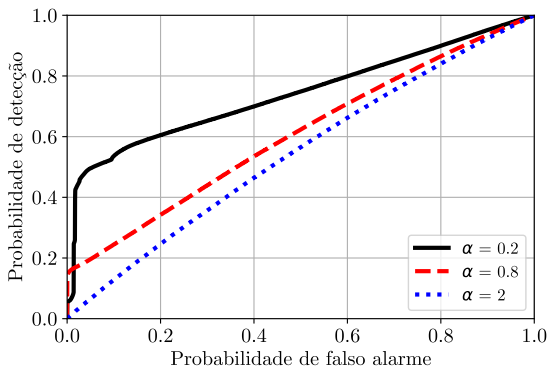


Fig. 1. Desempenho do detector de RI para diferentes valores de  $\alpha$ .

sua saída valores entre 0 e 1, permitindo classificar a amostra com a adoção de um limiar. O limiar ótimo encontrado na fase de treinamento foi 0.55. Se a saída da rede neural tiver valor menor que o limiar, o sistema decide pela ausência de RI, resultando um valor de saída 0, caso contrário a saída é classificada como 1, identificando a possível presença de RI. O treinamento ocorreu durante 100 épocas, por meio do algoritmo de otimização Adam [5], com passo de aprendizagem igual a 0.01 e *mini-batch* igual a 250. O conjunto de treinamento foi composto por 300 exemplos com 250 amostras cada. A função de erro a ser minimizada é dada por  $J = (-1/m) \sum_{i=1}^m (b_i \log(s_i) + (1 - b_i) \log(1 - s_i))$ , em que  $J$  é o erro,  $b_i = 1$  ou  $b_i = 0$  na  $i$ -ésima amostra,  $s_i$  é o valor gerado na saída da rede, ou seja na saída da última camada implementada pela função sigmoid e  $m$  é a quantidade total de amostras.

### III. RESULTADOS E CONCLUSÃO

A Figura 1 revela o desempenho do detector proposto, implementado utilizando a linguagem de programação Python, em termos da relação entre a taxa de verdadeiro positivo (probabilidade de detecção) e a taxa de falso positivo (probabilidade de falso alarme) para um valor fixo de  $p_{RI} = 0.8$  e  $\alpha = \{0.2; 0.8; 2\}$ . É nítida a melhoria do desempenho da rede neural à medida que ocorre um aumento da impulsividade do RI, i.e. um decréscimo do parâmetro  $\alpha$ .

Após o treinamento da rede neural, foi feito um comparativo de desempenho com um algoritmo presente na literatura [1] que utiliza o desvio padrão para as classificações e, mais importante, não é baseado na construção de uma rede neural.

A análise de comparação é feita com o uso de três métricas: acurácia, *recall* e precisão. A acurácia mede o percentual de amostras classificadas corretamente para ambas as classes, ou seja ausência e presença de RI [5, Page 4]. O *recall*, também conhecido como taxa de verdadeiro positivo, é a porcentagem de amostras classificadas corretamente como ruidosas em relação a todas as amostras que de fato foram corrompidas por RI [5, Eq. (3.2)]. A precisão determina a porcentagem de amostras classificadas corretamente como ruidosas em relação a todas as amostras identificadas como ruidosas [5]. A Tabela I mostra o resultado do teste para um conjunto de dados com 100 exemplos, contendo 250 amostras e com diferentes valores de probabilidade de ocorrência do RI,  $p_{RI}$ , e  $\alpha$ .

Com esses resultados é possível notar que na maioria dos casos o modelo proposto supera o algoritmo em [1]. Analisando o *recall* percebe-se que o modelo identificou um percentual maior de ruído sempre que  $p_{RI} > 0$ . Sendo que à medida em que a ocorrência de RI aumenta, o modelo destaca-se ainda mais. A acurácia foi inferior apenas quando não houve a presença de RI. A precisão foi menor para todos os casos e isso se justifica pelo erro presente no modelo. A precisão foi menor para todos os casos e isso se justifica pelo erro presente devido ao treinamento. Porém é importante destacar que a precisão foi, na maioria dos testes, superior a 90%, com apenas um caso em torno de 60% em que havia somente ruído Gaussiano.

Portanto, conclui-se que o modelo apresenta bons resultados e, por se tratar de um trabalho inicial, é importante sua continuação com a análise de outras métricas e aplicações, bem como um estudo maior da estrutura e treinamento da rede neural a fim de melhorar seu desempenho.

### REFERÊNCIAS

- [1] L. S. Costa, D. A. Guimarães, and R. A. A. de Souza, "Sensoriamento espectral cooperativo via teorema dos círculos de Gerschgorin sob ruído impulsivo," in *XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, Sept. 2018.
- [2] R. Barazideh, S. Niknam, and B. Natarajan, "Impulsive noise detection in OFDM-based systems: A deep learning perspective," in *IEEE Annual Computing and Commun. Workshop and Conf.*, pp. 0937–0942, Jan. 2019.
- [3] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764–766, 2013.
- [4] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. Image Process.*, vol. 14, pp. 1747–1754, Nov. 2005.
- [5] A. Géron, *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2017.