

Seleção de Hiperparâmetros para o Treinamento das Arquiteturas de Rajpurkar *et al.* e Ribeiro *et al.* voltadas à Classificação de Patologias Cardíacas

Roberto Marafon Leandro, Sarah Morgana Meurer, Daniel Gomes de Pinho Zanco, Eduardo Vinícius Kuhn e Ranniery Maia

Resumo— Este artigo apresenta uma abordagem sistemática para a seleção de hiperparâmetros no treinamento de duas arquiteturas relevantes de aprendizado profundo da literatura, usadas na classificação de patologias cardíacas. Especificamente, utilizando a base de dados multirrotulo PTB-XL ECG e métricas apropriadas, foi desenvolvido um *script* em Python para realizar uma busca exaustiva em grade sobre um espaço pré-definido de valores para os hiperparâmetros. Os resultados mostram um desempenho similar para ambas as arquiteturas quando valores apropriados de hiperparâmetros são utilizados.

Palavras-Chave— Busca exaustiva, classificação multirrotulo, eletrocardiograma.

Abstract— This paper presents a systematic approach to hyperparameter selection for training two relevant deep-learning architectures from the literature used for classifying cardiac pathologies. Specifically, using the multilabel PTB-XL ECG dataset and appropriate metrics, a Python script was developed to conduct an exhaustive grid search across a predefined hyperparameter values space. Results show similar performance for both architectures when appropriate values for the hyperparameters are used.

Keywords— Exhaustive grid search, multilabel classification, electrocardiogram.

I. INTRODUÇÃO

Doenças cardiovasculares figuram dentre as principais causas de morte no mundo, representando aproximadamente 32% das fatalidades registradas em 2019 [1]. Dessas fatalidades, uma importante parte poderia ter sido evitada através do diagnóstico precoce da doença realizado por meio de um exame cardíaco não invasivo, denominado eletrocardiograma (ECG), o qual fornece uma interessante representação da atividade elétrica do coração. A partir do ECG, características cardíacas, como o ritmo e a morfologia dos batimentos, podem ser identificadas e avaliadas [2]. Todavia, a interpretação do ECG para o diagnóstico preciso das condições cardíacas é

Roberto Marafon Leandro, Sarah Morgana Meurer e Eduardo Vinícius Kuhn estão vinculados ao LAPSE—Laboratório de Processamento de Sinais e Eletrônica do Departamento de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR), Toledo, PR, Brasil (e-mails: rleandro@alunos.utfpr.edu.br, sarah_morgana18@hotmail.com e kuhn@utfpr.edu.br).

Daniel Gomes de Pinho Zanco está vinculado ao LINSE—Laboratório de Circuitos e Processamento de Sinais do Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil (e-mail: dangpzanco@gmail.com).

Ranniery Maia está vinculado ao Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte (UFRN), Natal, RN, Brasil (e-mail: rmaia@dimap.ufrn.br).

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

uma tarefa complexa, requerendo assim perícia e atenção do profissional de saúde na detecção de alterações morfológicas. Destaca-se que até 33% das interpretações de ECG por médicos não cardiologistas apresentam erros de diagnóstico significativos [3].

Visando então auxiliar no processo de diagnóstico, técnicas de aprendizado profundo vêm sendo amplamente utilizadas para o desenvolvimento de ferramentas computacionais na área de eletrofisiologia clínica [2]. Dentre elas [4], destacam-se as arquiteturas apresentadas em [5] e [6], as quais foram adaptadas em [7] para classificar 5 patologias cardíacas em um conjunto de dados multirrotulos público [8]. Apesar do bom desempenho alcançado em [7], o ajuste dos hiperparâmetros foi realizado seguindo as diretrizes dadas em [5] e [6] (quando fornecidas). Contudo, considerando a alteração do conjunto de dados e as pequenas modificações necessárias nas arquiteturas, não se pode afirmar que os hiperparâmetros utilizados conduzem ao melhor desempenho; sobretudo, frente à natureza empírica do processo de seleção de hiperparâmetros [9, Sec. 11.4], [10, Ch. 6].

Neste contexto, considerando as arquiteturas adaptadas em [7] e a importância da seleção de hiperparâmetros, o presente trabalho de pesquisa visa:

- i) desenvolver um *script* em Python para auxiliar na seleção de valores dos hiperparâmetros usados no treinamento;
- ii) conduzir experimentos usando conjunto de dados e métricas padronizadas em espaço de busca pré-definido; e
- iii) apresentar e discutir os resultados alcançados, fornecendo diretrizes consistentes para comparações de desempenho.

II. FORMULAÇÃO DO PROBLEMA

O conjunto de dados utilizado, denominado PTB-XL e disponível em [8], contém 21.837 registros de sinais de ECG de 12 derivações, com 10 segundos de duração e frequência de amostragem de 100 Hz. Esse conjunto é composto por sinais de ECG rotulados por cardiologistas, assim como por metadados contendo declarações de diagnóstico de acordo com o padrão descrito na Norma ISO 11073-91064:2009 [11]. Tais declarações de diagnóstico presentes no conjunto de dados podem ser agrupadas em 5 superclasses não mutuamente exclusivas. Vale destacar que o conjunto de dados é aqui dividido entre treinamento, validação e teste, conforme indicado em [8], sendo somente os dois primeiros conjuntos usados para seleção de hiperparâmetros [10, Ch. 6].

Visto que um indivíduo pode ser diagnosticado com mais de uma patologia cardíaca (exceto quando saudável), a natureza

do problema se enquadra em classificação multirrotulo. Nesse sentido, considera-se aqui duas arquiteturas de rede neural profunda, adaptadas em [7], a saber:

- De Rajpurkar *et al.* [5] que conta com 34 camadas, arranjadas em 1 bloco de entrada, 16 blocos residuais com 2 camadas convolucionais cada e conexões de atalho, e um bloco de saída com função de ativação *sigmoid*.
- De Ribeiro *et al.* [6] que possui 10 camadas, organizadas em 1 bloco de entrada, 4 blocos residuais (modificados) com 2 camadas convolucionais cada e conexões de atalho, e 1 bloco de saída com a função de ativação *sigmoid*.

Mais detalhes dessas arquiteturas são fornecidos em [7].

III. IMPLEMENTAÇÃO PROPOSTA

A implementação proposta¹, realizada utilizando as bibliotecas Tensorflow/Keras, é baseada na abordagem de busca exaustiva em grade² dentro de um espaço pré-definido de valores típicos para os hiperparâmetros, compreendendo: *batch size* de 16, 32, 64, 128 e 256; *optimizer* como SGD, RMSProp e Adam; e *learning rate* de 0,001, 0,01 e 0,1. Esse espaço de busca produz 45 combinações de valores de hiperparâmetros para cada arquitetura, resultando assim na execução de 90 rodadas distintas de treinamento. Utiliza-se, durante o treinamento, os *callbacks* ReduceLRonPlateau e EarlyStopping da biblioteca Keras para a adequação da taxa de aprendizagem e interrupção antecipada do treinamento (máximo de 100 épocas), respectivamente. Ao final de cada rodada de treinamento, o modelo obtido é avaliado sobre o conjunto de validação utilizando a matriz de confusão multirrotulo (MLCM), cuja metodologia foi recentemente introduzida em [13]. A partir da MLCM, diferentes métricas são então calculadas, tais como *precision*, *recall* e *F1-score* bem como os correspondentes valores médios *micro*, *macro* e *weighted*. Os resultados obtidos em cada rodada são armazenados em um arquivo .csv para posterior análise.

IV. RESULTADOS E DISCUSSÃO

A Tabela I apresenta os resultados alcançados para a métrica (*macro avg*) *F1-score* sobre o conjunto de validação, considerando tanto a arquitetura de Rajpurkar *et al.* [5] quanto a de Ribeiro *et al.* [6], tendo em vista o espaço de busca de hiperparâmetros pré-definido. (Resultados próximos de zero indicam que aquela dada combinação de valores de hiperparâmetros não promove a convergência da arquitetura durante o treinamento.) A partir desses resultados, verifica-se que a arquitetura de Rajpurkar *et al.* [5], usando *batch size* de 128, *optimizer* Adam e *learning rate* de 0,01, alcança um (*macro avg*) *F1-score* máximo de 0,71. Por sua vez, a arquitetura de Ribeiro *et al.* [6], adotando *batch size* de 64, *optimizer* RMSProp e *learning rate* de 0,01, atinge um (*macro avg*) *F1-score* máximo de 0,70. Portanto, infere-se que ambas as arquiteturas exibem desempenho similar (mesmo frente às outras métricas omitidas aqui), quando valores apropriados para os hiperparâmetros são usados; dessa forma, comparações de desempenho justas entre os modelos podem ser conduzidas.

¹Veja https://github.com/lablapse/ecg_signal_processing.

²Alternativamente, uma estratégia de busca aleatória [12] pode ser adotada.

TABELA I

RESULTADOS OBTIDOS CONSIDERANDO A MÉTRICA (*macro avg*) *F1-score* PARA AS ARQUITETURAS DE RAJPURKAR *et al.* [5] E DE RIBEIRO *et al.* [6].

		Adam			RMSProp			SGD		
		0,001	0,01	0,1	0,001	0,01	0,1	0,001	0,01	0,1
Rajpurkar <i>et al.</i> [5]	16	0,69	0,69	0,47	0,67	0,67	0	0,49	0,62	0,63
	32	0,70	0,70	0,50	0,68	0,66	0,67	0,57	0,59	0,64
	64	0,70	0,69	0,66	0,69	0,68	0,67	0,33	0,58	0,66
	128	0,67	<u>0,71</u>	0,68	0,65	0,68	0,66	0,24	0,53	0,52
	256	0,68	0,68	0,65	0,65	0,69	0,60	0,39	0,50	0,61
Ribeiro <i>et al.</i> [6]	16	0,65	0,69	0,17	0,67	0,67	0,66	0,61	0,64	0,67
	32	0,68	0,69	0,61	0,67	0,66	0,68	0,62	0,66	0,66
	64	0,68	0,69	0,53	0,63	<u>0,70</u>	0,62	0,59	0,66	0,67
	128	0,67	0,69	0,67	0,64	0,66	0,68	0,56	0,61	0,64
	256	0,64	0,66	0,68	0,65	0,68	0,69	0,45	0,63	0,66

*Valores sublinhados indicam o melhor desempenho de cada arquitetura.

V. CONCLUSÕES

Neste trabalho, foi realizada a implementação de um *script* (em linguagem Python) para auxiliar na seleção de valores para alguns hiperparâmetros pertinentes ao treinamento de duas importantes arquiteturas da literatura (discutidas em [7]), voltadas à classificação de patologias cardíacas em um conjunto de dados multirrotulo. Baseado nos resultados obtidos, observou-se que ambas as arquiteturas consideradas exibem desempenho similar quando combinações apropriadas de valores de hiperparâmetros são utilizadas, sendo tal impacto menos significativo no desempenho da arquitetura de Ribeiro *et al.* [6]. Visando dar continuidade ao presente trabalho de pesquisa, pretende-se agora expandir a busca contemplando hiperparâmetros internos das arquiteturas.

REFERÊNCIAS

- [1] World Health Organization (WHO). (2021) Cardiovascular diseases (CVDs). [Online]. Available: [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [2] G. Sannino and G. D. Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Future Generation Comput. Syst.*, vol. 86, pp. 446–455, 2018.
- [3] P. Mele, "Improving electrocardiogram interpretation in the clinical setting," *J. Electrocardiology*, vol. 41, pp. 438–439, 2008.
- [4] S. Somani *et al.*, "Deep learning and the electrocardiogram: A review of the current state-of-the-art," *Eur. Soc. Cardiology*, vol. 23, pp. 1179–1191, 2021.
- [5] P. Rajpurkar *et al.*, "Cardiologist-level arrhythmia detection with convolutional neural networks," *ArXiv*, vol. abs/1707.01836, 2017.
- [6] A. H. Ribeiro *et al.*, "Automatic diagnosis of the 12-lead ECG using a deep neural network," *Nature Commun.*, vol. 11, 2020.
- [7] S. M. Meurer, "Detecção e classificação de patologias cardíacas em electrocardiogramas utilizando redes neurais profundas," TCC do Curso de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná, Toledo, PR, 2022.
- [8] P. Wagner *et al.*, "PTB-XL, a large publicly available electrocardiography dataset," *Sci Data*, vol. 7, 2020.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
- [10] S. Raschka, Y. H. Liu, and V. Mirjalili, *Machine Learning with Pytorch and Scikit-Learn*, 1st ed. Packt, 2022.
- [11] International Organization for Standardization (ISO), "Health informatics – Standard communication protocol – Part 91064: Computer-assisted electrocardiography," Geneva, CH, International Standard ISO 11073-91064:2009, 2009.
- [12] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learning Research*, vol. 13, pp. 281–305, 2012.
- [13] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19 083–19 095, 2022.