

Aplicação de rede neural convolucional em dados de eletroencefalografia

Seidi Yonamine Yamauti



INSTITUTO

SANTOS DUMONT

ENSINO E PESQUISA

Objetivos

- Testar modelo estado da arte em classificação de dados de eletroencefalografia (EEG)
- Documentar, produzir e disponibilizar um código útil em python



EEGNet: A Compact Convolutional Neural Network for EEG-based Brain-Computer Interfaces

Vernon J. Lawhern^{1,*}, Amelia J. Solon^{1,2}, Nicholas R. Waytowich^{1,3}, Stephen M. Gordon^{1,2},
Chou P. Hung^{1,4}, and Brent J. Lance¹

¹Human Research and Engineering Directorate, U.S. Army Research Laboratory, Aberdeen
Proving Ground, MD

²DCS Corporation, Alexandria, VA

³Department of Biomedical Engineering, Columbia University, New York, NY

⁴Department of Neuroscience, Georgetown University, Washington, DC

*Corresponding Author

May 17, 2018

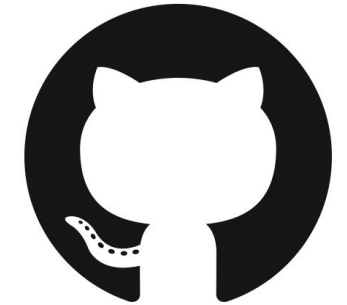
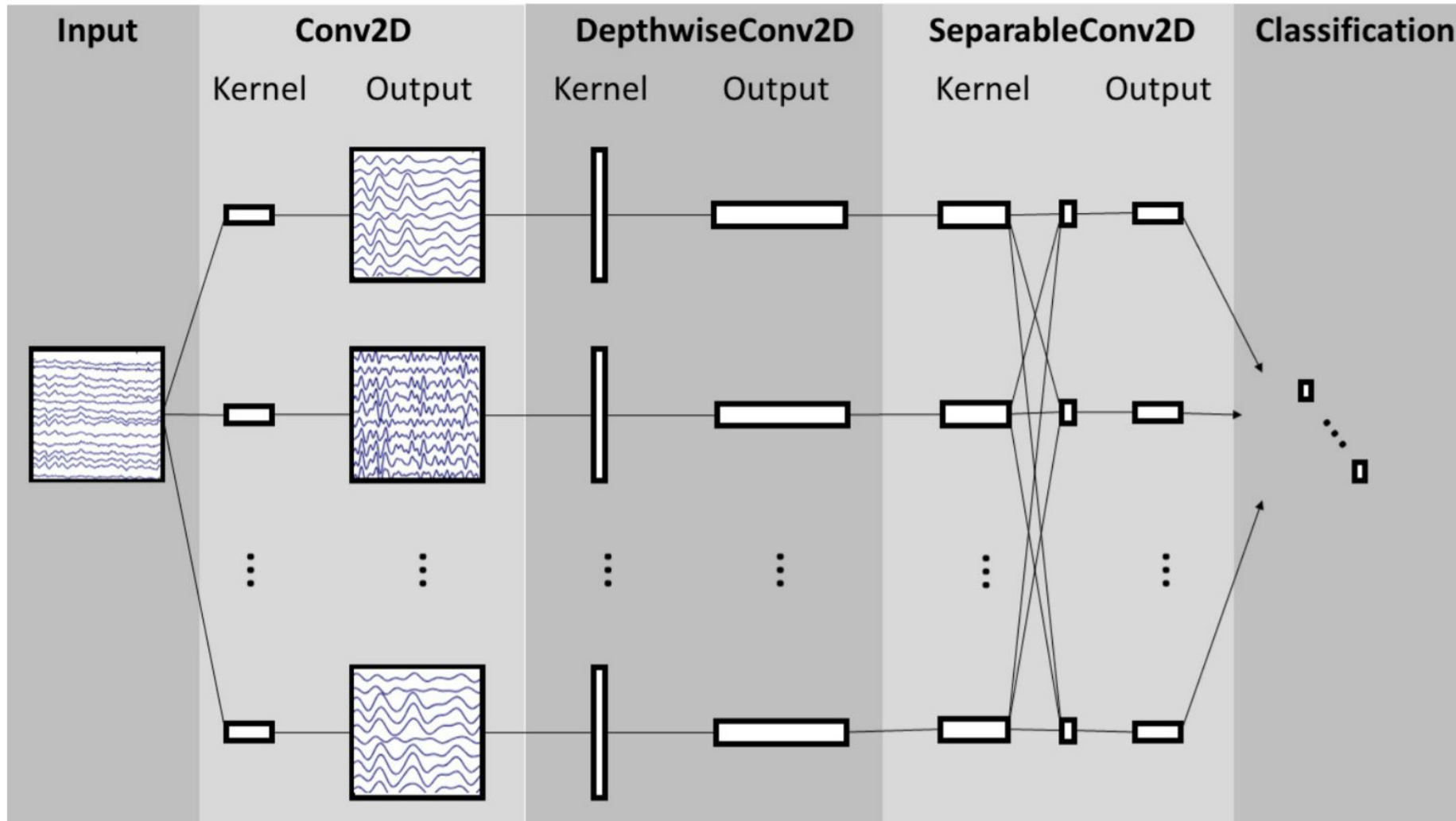
Abstract

[cs.LG] 16 May 2018

O artigo

- Desenvolvimento do modelo EEGNet [1]
 - Arquitetura de rede neural convolucional (CNN), compacta
 - O objetivo é generalizar para diversos paradigmas de interface cérebro-máquina (BCI)
 - Testado com resultados significativos (P300, ERN, MRCP e SMR)
 - Método de explicação de características selecionadas

O artigo - descrição do modelo



O artigo - descrição do modelo

Shallow Learning (Aprendizado de Máquina Clássico)

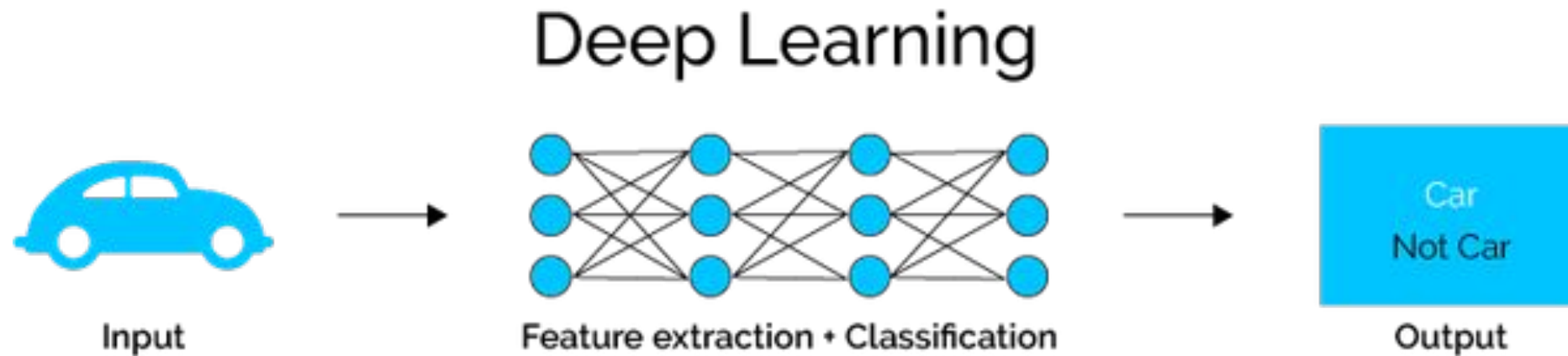
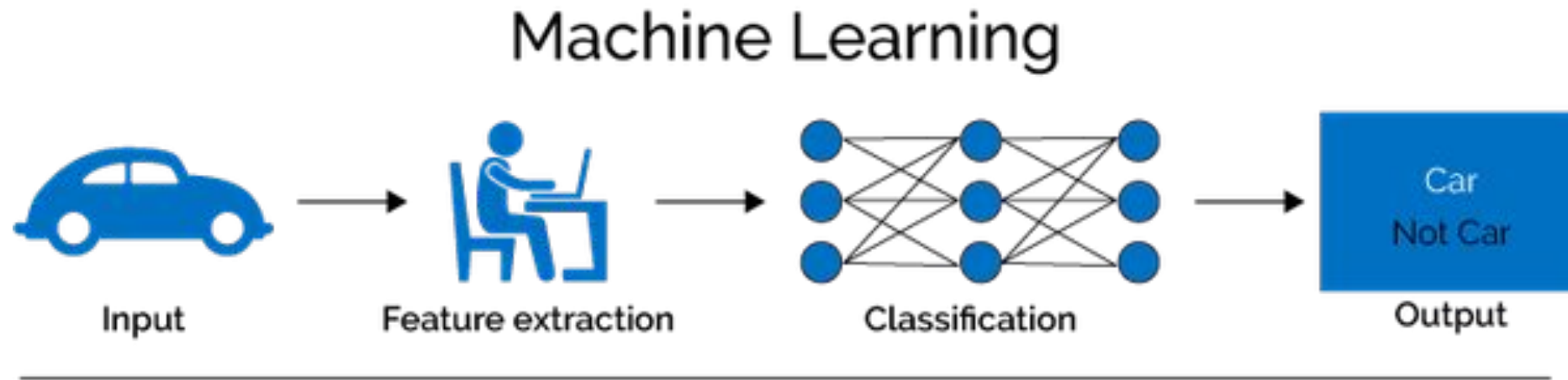
- Um modelo para extração de características
- Um modelo para classificação

Deep Learning (Aprendizado Profundo)

- Um modelo para extração de características e classificação

Ver referência [3] para mais
detalhes

O artigo - descrição do modelo



Ver referência [3] para mais
detalhes

O artigo - descrição do modelo

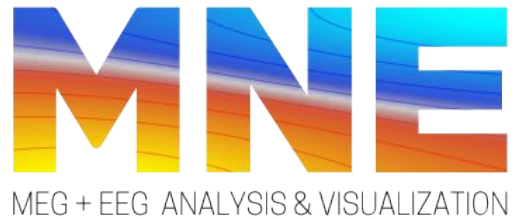
Extração de características

Block	Layer	# filters	size	# params	Output	Activation	Options
1	Input				(C, T)		
	Reshape				(1, C, T)		
	Conv2D	F_1	(1, 64)	$64 * F_1$	(F_1, C, T)	Linear	mode = same
	BatchNorm			$2 * F_1$	(F_1, C, T)		
	DepthwiseConv2D	$D * F_1$	(C, 1)	$C * D * F_1$	$(D * F_1, 1, T)$	Linear	mode = valid, depth = D, max norm = 1
	BatchNorm			$2 * D * F_1$	$(D * F_1, 1, T)$		
	Activation				$(D * F_1, 1, T)$	ELU	
	AveragePool2D		(1, 4)		$(D * F_1, 1, T // 4)$		
	Dropout*				$(D * F_1, 1, T // 4)$		$p = 0.25$ or $p = 0.5$
	SeparableConv2D	F_2	(1, 16)	$16 * D * F_1 + F_2 * (D * F_1)$	$(F_2, 1, T // 4)$	Linear	mode = same
2	BatchNorm			$2 * F_2$	$(F_2, 1, T // 4)$		
	Activation				$(F_2, 1, T // 4)$	ELU	
	AveragePool2D		(1, 8)		$(F_2, 1, T // 32)$		
	Dropout*				$(F_2, 1, T // 32)$		$p = 0.25$ or $p = 0.5$
	Flatten				$(F_2 * (T // 32))$		

Classificação

Classifier	Dense	$N * (F_2 * T // 32)$			N	Softmax	max norm = 0.25
------------	-------	-----------------------	--	--	---	---------	-----------------

Pacotes python

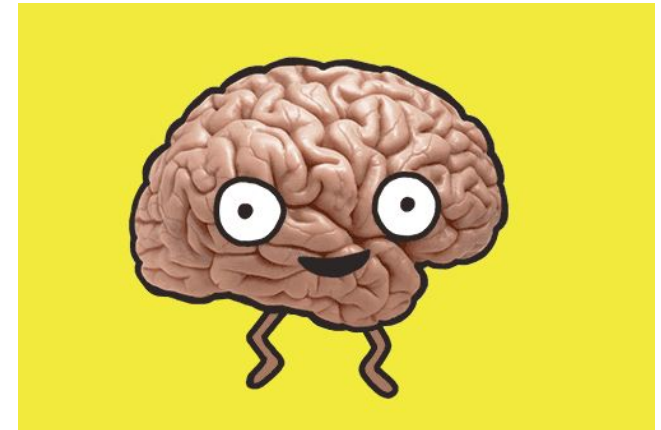


pyRiemann

matplotlib

Dados escolhidos

- **EEGBCI motor imagery**
 - Dados obtidos do banco de dados PhysioNet
 - Já em formato EDF (MNE)
 - Composto por sinais EEG de imagética motora, contendo 64 eletrodos/canais, 109 sujeitos, 14 sessões
- Dados cabíveis a paraplegia
 - melhorar classificação de imagética motora
 - exploração/entendimento de características



Resultados esperados (até 02/10)

- Pipeline de processamento dos dados escolhidos usando a EEGNet
- Comparação com resultados de outros estudos
- Código, apresentação e abstract disponibilizados

Referências

1. Lawhern, Vernon J., et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces." *Journal of neural engineering* 15.5 (2018): 056013.
2. <https://github.com/vlawhern/arl-eegmodels>
3. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436-444.

Email: seidi.yamauti@edu.isd.org.br

Github: <https://github.com/seidikun>

 institutosantosdumont.org.br

 contato@isd.org.br

   isdnarede



MINISTÉRIO DA
EDUCAÇÃO

