

AutoML e sua contribuição para o desenvolvimento de modelos para a Aprendizagem Profunda

Aluno: Lucas Henrique Lório Costa
Orientador: Talles Henrique de Medeiros

Departamento de Computação e Sistemas - DECSI
Instituto de Ciências Exatas e Aplicadas - ICEA
Universidade Federal de Ouro Preto - UFOP

21/02/2024



Instituto de Ciências
Exatas e Aplicadas



UFOP

Introdução

Revisão Bibliográfica

Desenvolvimento

Resultados

Desafios

Conclusão

Referências

Introdução

Revisão Bibliográfica

Desenvolvimento

Resultados

Desafios

Conclusão

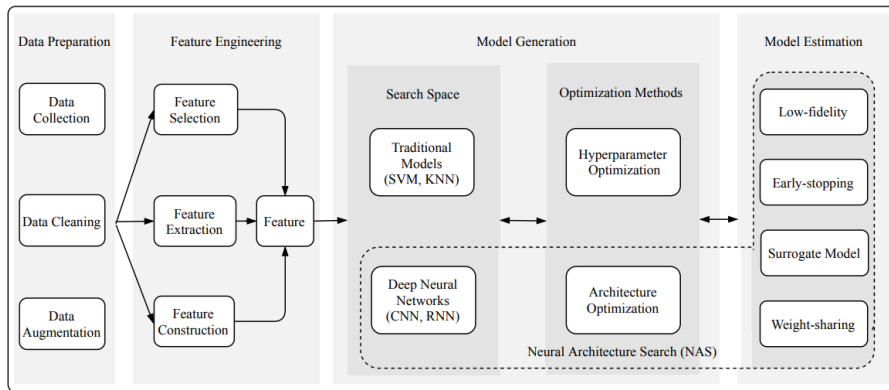


UFOP

- ▶ IA nos dias atuais
- ▶ O AutoML
- ▶ Objetivos do trabalho:
 - ▶ Implementar uma abordagem do AutoML - otimização de hiperparâmetros
 - ▶ Encontrar automaticamente os melhores hiperparâmetros para um modelo
 - ▶ Encontrar automaticamente os melhores pesos para uma função de custo multiobjetivo
 - ▶ Alcançar modelos que tenham um bom desempenho



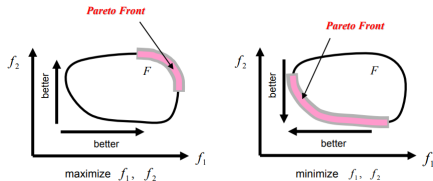
► AutoML e Otimização de Hiperparâmetros



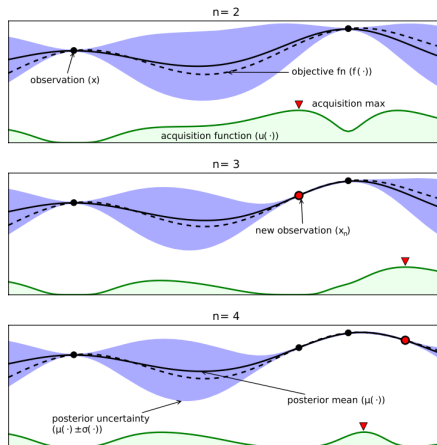
Fonte: (HE; ZHAO; CHU, 2021)[1]



- Otimização Bayesiana
- Abordagem multiobjetivo
 - Conjunto Pareto Ótimo



Fonte:(NGATCHOU; ZAREI; EL-SHARKAWI, 2005)[2]



Fonte:(SHAHRIARI et al., 2015)[3]



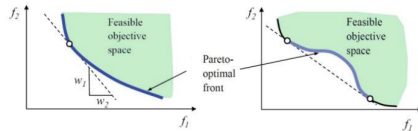
► Função de custo multiobjetivo

- Erro - Entropia Cruzada
- Complexidade - Norma L_2

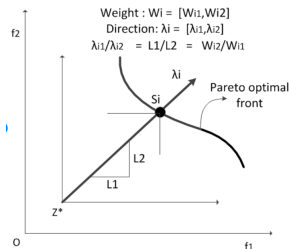
► Escalarização Chebyshev

$$g = \max_i [w_i |f_i - z_i^*|]$$

$$\text{ChebLoss} = \max (w_1 \cdot EC \mid w_2 \cdot NL2)$$



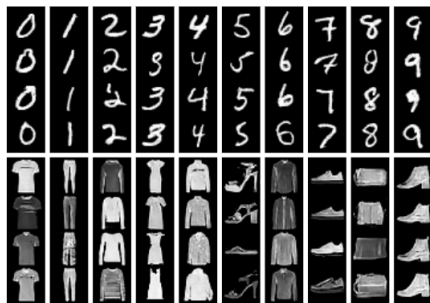
Fonte: (NAGY; MANSOUR; ABDELMOHSEN, 2020)[4]



Fonte: (WANG; ZHANG; GUO, 2013)[5]



- ▶ Linguagem: Python
- ▶ Biblioteca: KerasTuner
- ▶ IDE: Google Colaboratory (Colab)
- ▶ Experimentos com hiperparâmetros:
 - ▶ Bases de dados:
MNIST e Fashion MNIST
 - ▶ Funções de perda:
Crossentropy e CustomChebyshev
 - ▶ Modelos:
Modelo MLP e Modelo CNN
- ▶ Testes com pesos fixos da função
- ▶ Visualização do conjunto Pareto ótimo



Fonte:(BOOTH, 2020)[6]



Hiperparâmetros otimizados:

- Unidades na camada densa (oculta) $[32 - 512; 32]$
- Taxa de aprendizado $[1 \cdot 10^{-1}; 1 \cdot 10^{-2}; 1 \cdot 10^{-3}]$
- Peso 1 (w_1) da função objetiva $[0 \sim 1]$
- Peso 2 (w_2) da função objetiva $[0 \sim 1]$

flatten_input	input:	[(None, 28, 28)]
InputLayer	output:	[(None, 28, 28)]



flatten	input:	(None, 28, 28)
Flatten	output:	(None, 784)



dense	input:	(None, 784)
Dense	output:	(None, 32)

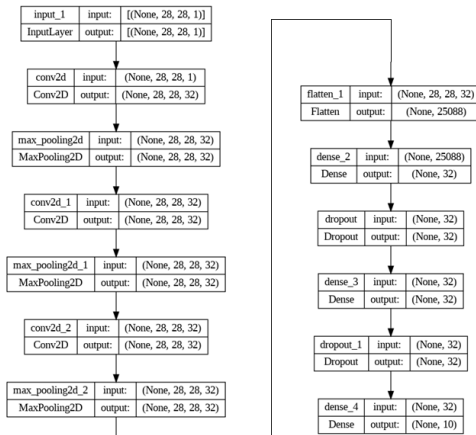


dense_1	input:	(None, 32)
Dense	output:	(None, 10)

Fonte: Produzido pelo autor.



- ▶ Taxa de aprendizado [$1 \cdot 10^{-1}$; $1 \cdot 10^{-2}$; $1 \cdot 10^{-3}$]
- ▶ Peso 1 (w1) da função objetiva [$0 \sim 1$]
- ▶ Peso 2 (w2) da função objetiva [$0 \sim 1$]
- ▶ Camadas convolucionais:
 - ▶ Unidades [32 – 512; 32]
 - ▶ Tamanho do kernel [3 – 5; 1]
 - ▶ Tamanho do stride [1 – 5; 1]
- ▶ Camadas de pooling:
 - ▶ Tamanho do kernel [2 – 4; 1]
 - ▶ Tamanho do stride [1 – 5; 1]
- ▶ Função de ativação [relu; tanh; sigmoid]
- ▶ Unidades nas 2 camadas densas [32 – 512; 32]
- ▶ Taxa de dropout [0 – 0.5; 0.1]
- ▶ Função otimizadora do modelo [adam; SGD; rmsprop]



Fonte: Produzido pelo autor.





► Experimentos com redes MLP e CNN

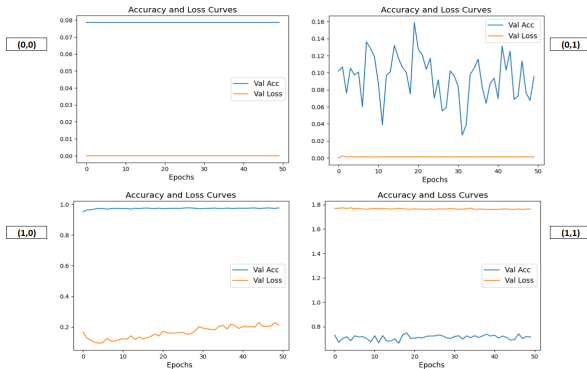
Modelo	1	2	3	4
Dataset	MNIST	MNIST	FashionMNIST	FashionMNIST
Loss function	CustomCheb	CrossEnt	CustomCheb	CrossEnt
loss_wgt1	0.9	-	0.5	-
loss_wgt2	0.0	-	0.0	-
Dense units	512	256	384	448
Learning rate	0.001	0.001	0.001	0.001
Test loss	0.1450	0.1471	0.2839	0.5861
Test accuracy	98.13 %	97.87%	88.51 %	88.78%
HP searching time (min)	38.68	16.43	64.17	20.53
HModel training time (min)	9.38	5.06	7.95	4.30

Fonte:Produzido pelo autor.

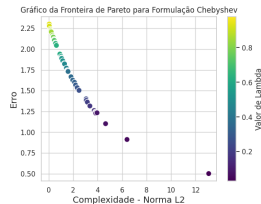
Modelo	1	2	3	4
Dataset	MNIST	MNIST	FashionMNIST	FashionMNIST
Loss function	CrossEnt	CustomCheb	CrossEnt	CustomCheb
conv1_units	352	128	128	512
conv2_units	352	96	384	192
conv3_units	192	128	192	128
kern_size1	5	5	5	4
kern_size2	5	3	4	4
kern_size3	4	5	3	4
pool_size1	2	2	4	3
pool_size2	4	2	2	2
pool_size3	3	2	3	3
stride_conv1	3	1	1	4
stride_conv2	1	2	5	3
stride_conv3	1	5	1	5
stride_pool1	1	2	3	3
stride_pool2	2	2	2	4
stride_pool3	4	5	5	4
activation	tanh	tanh	relu	relu
dense_units1	64	128	416	480
dense_units2	288	128	512	384
dropout_rate1	0.1	0.4	0.2	0.0
dropout_rate2	0.0	0.5	0.0	0.0
optimizer	SGD	adam	adam	rmsprop
Learning rate	0.1	0.001	0.001	0.001
loss_wgt1	-	0.7	-	0.8
loss_wgt2	-	0.0	-	0.0
Test loss	0.2710	0.0970	0.7022	1.510
Test accuracy	99.34 %	95.79 %	90.17%	82.90 %
HP searching time (min)	46.68	101.85	48.58	80.52
HModel training time (min)	38.38	13.30	12.38	14.15

Fonte:Produzido pelo autor.

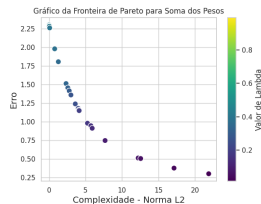
► Teste com pesos fixos e visualização da fronteira de Pareto



Fonte:Produzido pelo autor.



Fonte:Produzido pelo autor.



Fonte:Produzido pelo autor.

- ▶ Tentativa de implementar o NAS
- ▶ Testes com base de dados CIFAR-10
- ▶ Limites de recursos computacionais



- ▶ Implementação da otimização de hiperparâmetros
- ▶ Determinação dos pesos para uma abordagem multiobjetivo

Link para o repositório:

https://github.com/LucasHLirio/TCC_II



1. He, X., Zhao, K. & Chu, X. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* **212**, 106622 (2021).
2. Ngatchou, P., Zarei, A. & El-Sharkawi, A. Pareto multi objective optimization. em *Proceedings of the 13th international conference on, intelligent systems application to power systems* (2005), 84–91.
3. Shahriari, B., Swersky, K. *et al.* Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE* **104**, 148–175 (2015).
4. Nagy, M., Mansour, Y. & Abdelmohsen, S. Multi-objective optimization methods as a decision making strategy. *Int. J. Eng. Res. Technol.(IJERT)* **9**, 516–522 (2020).
5. Wang, R., Zhang, T. & Guo, B. An enhanced MOEA/D using uniform directions and a pre-organization procedure. em (2013), 2390–2397. doi:10.1109/CEC.2013.6557855.
6. Booth, S. *Bayes-Probe: Distribution-Guided Sampling for Prediction Level Sets*. https://www.researchgate.net/figure/Inferred-high-classification-samples-for-CLEVR-top-MNIST-middle-and-Fashion-MNIST_fig3_339471054. 2020.

