

AutoML e Sua Contribuição para o Desenvolvimento de Modelos para a Aprendizagem Profunda

Lucas Henrique Lirio Costa¹

¹Instituto de Ciências Exatas e Aplicadas - Universidade Federal de Ouro Preto (UFOP)
João Monlevade - MG - Brasil

lucas.lirio@aluno.ufop.edu.br

Abstract. *This work aims to study Automated Machine Learning (AutoML), focusing on the Neural Architecture Search (NAS) stage in order to automatically find neural architectures that solve problems with minimal or none human intervention. Recent research in the area of Neural Architecture Search was studied and a framework was chosen to implement a sample code. The proposed continuation of this study is to implement AutoML in a work with Convolutional Neural Networks for facial recognition of people wearing masks*

Resumo. *Este trabalho tem como objetivo estudar o Aprendizado de Máquina Automatizado (AutoML), focando na etapa da Busca de Arquiteturas Neurais (Neural Architecture Search) a fim de encontrar automaticamente arquiteturas neurais que resolvam problemas com mínima ou nenhuma interferência humana. Foram estudadas pesquisas recentes na área da Busca de Arquiteturas Neurais e escolhido um framework para implementar um código de exemplo. A proposta de continuidade deste estudo é implementar o AutoML em um trabalho com Redes Neurais Convolucionais para reconhecimento facial de pessoas utilizando máscara.*

1. Introdução

Nos últimos anos, o Aprendizado de Máquina, mais especificamente no campo da Aprendizagem Profunda, tem sido amplamente aplicado em várias áreas de estudo e obtendo bons resultados em resolver diversos desafios de Inteligência Artificial, como nas áreas de detecção de objetos, classificação de imagens, processamento de linguagem natural, etc. Com essa crescente evolução, têm surgido redes neurais cada vez mais profundas e complexas, porém elas dependem de serem projetadas manualmente por especialistas através de um processo de tentativa e erro, que pode ser longo e exaustivo, o que faz com que mesmo os especialistas necessitem de tempo e recursos significativos para criar modelos que obtenham um bom desempenho. Para reduzir todo esse esforço no desenvolvimento da Rede Neural, foi proposta uma ideia de automatizar todo o pipeline do Aprendizado de Máquina, permitindo que pessoas construam automaticamente aplicações sem muita exigência de conhecimento estatístico e em Aprendizado de Máquina, surgindo, assim o AutoML - Automated Machine Learning.

2. Conceitos gerais e revisão da literatura

Com a crescente demanda de produção e uso de dados nas corporações, e do uso da Aprendizagem Profunda (um ramo do Aprendizado de Máquina) para a criação de Inte-

ligências Artificiais capazes de trabalhar com esses dados, o AutoML (Automated Machine Learning) surgiu da necessidade de tentar automatizar o processo da criação e da configuração de uma rede neural que seja eficiente para o problema em questão.

O pipeline do AutoML consiste nas etapas de preparação de dados, engenharia de recursos, geração de modelo e avaliação de modelo. As duas etapas iniciais são comuns e bem conhecidas, sendo a primeira responsável por coletar, limpar e acrescentar dados à sua base. Na etapa da engenharia de recursos a base de dados é trabalhada de forma a coletar e separar a informação que será útil para o modelo da IA.

A etapa da geração de modelos pode ser dividida na criação do espaço de busca e nos métodos de otimização. No espaço de busca são definidos os princípios de design dos modelos de AM, divididos em modelos tradicionais (como K-Vizinhos Próximos e Máquina de Vetor de Suporte) e Redes Neurais Profundas. Os métodos de otimização são classificados em Otimização de Hiperparâmetros, e Otimização de Arquitetura, onde o primeiro otimiza os parâmetros relacionados à etapa do treinamento, como a taxa de aprendizado e o tamanho do lote, e o último indica os parâmetros relacionados à estrutura modelo, como por exemplo, o número de camadas para arquiteturas neurais ou o número de vizinhos para o modelo de K-Vizinhos Próximos.

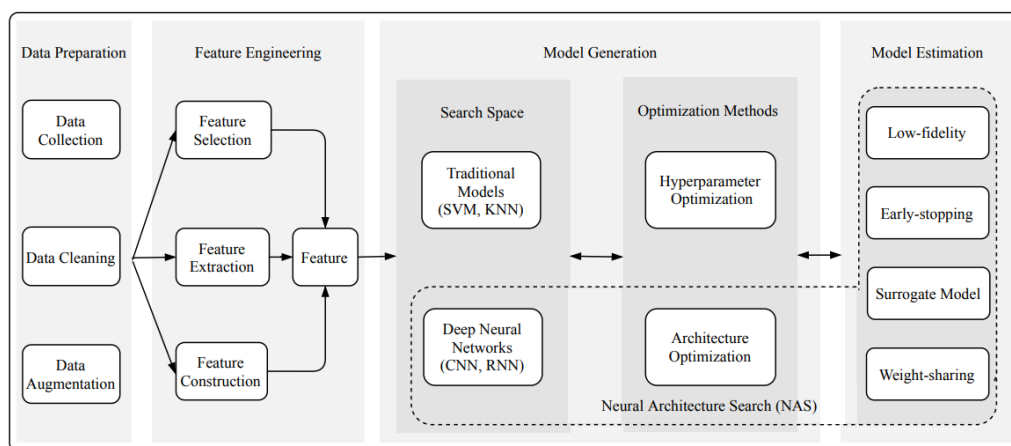


Figura 1. Uma visão geral do pipeline do AutoML abrangendo suas principais etapas

Sendo um subcampo importante do AutoML e que tem ganhado notável relevância nos últimos anos, o processo do Neural Architecture Search (NAS) – ou Busca de Arquitetura Neural – tem sido estudado e aprimorado para desenvolver algoritmos que alcancem um resultado considerável na busca automática pela melhor arquitetura de rede neural, dado um determinado problema. O NAS consiste em três componentes principais: espaço de busca, estratégia de busca e estratégia de avaliação. O espaço de busca por uma rede neural define, a princípio, quais arquiteturas podem ser representadas para o problema, pois cada cenário diferente requer espaços de busca diferentes. A estratégia de busca vai detalhar como o espaço de busca escolhido será explorado, a fim de encontrar com eficiência a melhor arquitetura do modelo. Por fim, a estratégia de estimativa de desempenho se refere ao processo de avaliar o desempenho do modelo escolhido a fim de o comparar com os outros e, finalmente, escolher o melhor modelo.

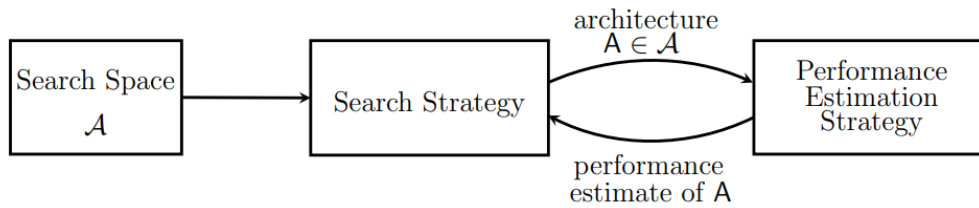


Figura 2. Abstração do processo de Busca de Arquitetura Neural

3. Metodologia

Para o desenvolvimento do trabalho, a linguagem de programação escolhida foi o Python, devido à sua facilidade de uso, grande utilização pela comunidade e suporte a diversas bibliotecas para a área da Inteligência Artificial. Foi escolhido também o ambiente de desenvolvimento Google Colaboratory, dada a facilidade de se ter um ambiente de execução em nuvem e com servidores dedicados, o que comparado à execução em uma máquina local, economiza tempo de configuração do ambiente, como instalação de bibliotecas e drivers, e contorna o problema da disponibilidade de hardware, caso a máquina tenha poucos recursos.

O passo posterior foi escolher um framework de Python que oferecesse suporte para métodos de AutoML, mais especificamente métodos de NAS, a fim de criar um código de exemplo e, no futuro, poder utilizar os recursos oferecidos para implementar a automação de rede neural em um trabalho onde a rede foi escolhida e configurada manualmente. Para isso, foram feitas diversas pesquisas a respeito de algoritmos de NAS e quais frameworks ofereciam esses algoritmos. Segundo o benchmark feito com alguns dos principais algoritmos disponibilizada no site da ferramenta NNI (Neural Network Intelligence) [Microsoft 2021], o modelo DARTS obteve uma melhor performance no quesito da acurácia, quando testado em famosas bases de dados de imagens como: CIFAR-10, CIFAR-100, Fashion-MNIST, OUI-Adience-Age, ImageNet-10-1e ImageNet-10-2. Os resultados, seguidos do cálculo de sua média e variância, além de grifado a melhor acurácia por casa base de dados, são mostrados a seguir:

Tabela 1. Desempenho dos algoritmos de NAS por base de dados

NAS	AutoKeras (%)	ENAS (macro) (%)	ENAS (micro) (%)	DARTS (%)	NAO-WS (%)
Fashion-MNIST	91,84	95,44	95,53	95,74	95,2
CIFAR-10	75,78	95,68	96,16	94,23	95,64
CIFAR-100	43,61	78,13	78,84	79,74	75,75
OUI-Adience-Age	63,2	80,34	78,55	76,83	72,96
ImageNet-10-1	61,8	77,07	79,8	80,48	77,2
ImageNet-10-2	37,2	58,13	56,47	60,53	61,2
Média	62,24	80,8	80,89	81,26	79,66
Variância	406,94	194,19	210,94	166,03	180,77

Dado o bom desempenho do DARTS e dentre os frameworks que oferecem suporte ao mesmo, decidiu-se utilizar o framework NASLib [Ruchte et al. 2020] devido ao fato de oferecer uma biblioteca projetada para ser modular e flexível, além de possuir uma documentação bem explicada, e alguns *Python notebooks* introdutórios desenvolvidos pelos próprios criadores da ferramenta, o grupo AutoML Freiburg, o qual possui um amplo

time de pesquisa e desenvolvimento na área do AutoML. Com base neste *Python notebook* introdutório ao NASLib foi feito um outro *notebook* [Lucas H L Costa 2023] mais resumido para que fossem executados os métodos da ferramenta, a fim de testar e estudar seu funcionamento, para depois ser mostrado na apresentação do presente trabalho.

4. Conclusão

Através deste estudo foi possível entender melhor o AutoML e sua importante contribuição para o campo do Aprendizado de Máquina, além de estudar a etapa da Busca de Arquitetura Neural, fundamental para a escolha do melhor modelo. Por fim, foi escolhido um framework para a implementação de um código de exemplo para ser apresentado e servir de base para ser reutilizado na continuação deste trabalho.

Referências

- Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017.
- He, X., Zhao, K., and Chu, X. (2021). Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622.
- Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., and Veeramachaneni, K. (2021). Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, 54(8):1–36.
- Lucas H L Costa (2023). Exemplo de utilização do NASLib para AutoML. https://colab.research.google.com/github/LucasHLirio/tcc_1/blob/main/Exemplo_NASLib_TCC_I.ipynb.
- Microsoft (2021). Neural Network Intelligence. <https://github.com/microsoft/nni>.
- Noé, Í. T. (2021). Redes neurais convolucionais aplicadas ao reconhecimento facial em indivíduos com máscara. *Trabalho de Conclusão de Curso – Instituto de Ciências Exatas e Aplicadas -Universidade Federal de Ouro Preto*.
- Ruchte, M., Zela, A., Siems, J., Grabocka, J., and Hutter, F. (2020). Naslib: A modular and flexible neural architecture search library. <https://github.com/automl/NASLib>.