

Introdução

Em 2017, Snell et al. [1] apresentaram uma abordagem promissora para a classificação few-shot de imagens: as Redes Prototípicas. Tais redes mapeiam as imagens em um espaço métrico no qual amostras de uma mesma classe se aglomeram. Sendo assim, para cada classe, um referencial, denominado protótipo, é obtido por meio da média aritmética das amostras no espaço. Para classificar uma imagem, a distância euclidiana da amostra a todos os protótipos é tomada, supondo-se que a menor distância determine a classe. Contudo, Ji et al. [2] propuseram o uso de pesos no cálculo do protótipo, o que resultou na criação das Redes Prototípicas Ponderadas. Em suma, os pesquisadores objetivaram considerar os diferentes níveis de representatividade das imagens. Desta forma, imagens ruins teriam menor influência na determinação dos protótipos.

Objetivo

Ji et al. [2] empregaram uma rede rasa na obtenção dos pesos para o cálculo dos protótipos, decisão que trouxe resultados promissores. Entretanto, uma alternativa instigante seria modelar o problema como uma tarefa de otimização metaheurística. Tendo em vista o exposto, o bolsista tenciona aprimorar a representatividade dos protótipos empregando técnicas metaheurísticas.

Material e Métodos

Para implementar as redes supracitadas bem como executar testes, utilizou-se o Google Colab. Por outro lado, para os experimentos, passou-se a utilizar as nuvens Google Cloud e Microsoft Azure.

Em termos de hardware, empregou-se placas gráficas NVIDIA Tesla K80 no processamento das imagens. Em termos de software, fez-se uso da linguagem Python 3.7, do framework PyTorch 1.8, e das bibliotecas Opytimizer [3] e Ray. Dentre os experimentos executados até o momento, pode-se citar o treinamento e teste das Redes Prototípicas (RP) e de uma versão simplificada das redes ponderadas, as Redes Prototípicas com Pesos Aleatórios (RPPA). As bases de dados consideradas foram a Omniglot e a mini-ImageNet. Outrossim, devido ao tempo de treinamento das redes, fez-se preciso implementar o algoritmo de Otimização por Enxame de Partículas (PSO, no inglês) com paralelismo. Para tal, utilizou-se a biblioteca de computação distribuída, Ray, em conjunto com o algoritmo presente na Opytimizer. Os experimentos com o algoritmo, no entanto ainda não foram finalizados. O bolsista pretende coletar os resultados bem como testar uma segunda técnica.

Resultados e Discussão

Segue abaixo a tabela com os resultados obtidos com os modelos nas bases mencionadas.

Tabela 1. Resultados obtidos com base em 15 execuções. Negrito indica os melhores resultados, de acordo com o teste dos postos sinalizados de Wilcoxon com 95% de confiança.

| Modelo | Base | Acurácia |
|-----------|----------------------|----------------------|
| RP | Omniglot | 99.62 ± 0.01% |
| RP | mini-ImageNet | 66.98 ± 0.17% |
| RPPA | Omniglot | 99.52 ± 0.01% |
| RPPA | mini-ImageNet | 64.98 ± 0.18% |

É possível afirmar que os pesos aleatórios resultam em acurácias próximas às do modelo tradicional.

Conclusão

As Redes Prototípicas constituem um modelo simples para a classificação de imagens rotuladas em um regime no qual tais dados são escassos. Contudo, a abordagem pode ser aperfeiçoada com a adição de um conjunto adicional de parâmetros, os pesos propostos por Ji et al. [2]. Com os experimentos realizados até então, pode-se afirmar que as RPs obtiveram melhores resultados do que as RPPAs. Quanto às metaheurísticas, ainda não é possível afirmar que são alternativas melhores para o cálculo dos pesos.

Agradecimentos

Agradecemos a Deus pelo acolhimento, a FAPESP pela concessão da bolsa 20/16092-5, e aos nossos amigos e familiares por todo apoio e carinho.

Referências

¹ Jake Snell, Kevin Swersky, e Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, **2017**.
² Zhong Ji, Xingliang Chai, Yunlong Yu, Yanwei Pang, e Zhongfei Zhang. Improved prototypical networks for few-shot learning. *Pattern Recognition Letters*, 140:81–87, **2020**.
³ Douglas Rodrigues Gustavo H. de Rosa e João P. Papa. Opytimizer: A nature-inspired python optimizer, **2019**.