Proposta de Avaliação Prática de Frameworks para a Distribuição de Redes de Aprendizado Profundo

Ana Luisa Veroneze Solórzano, Lucas Mello Schnorr

Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

alvsolorzano@inf.ufrgs.br, schnorr@inf.ufrgs.br

Resumo. *Frameworks para distribuir o treinamento de aprendizado profundo usam estruturas aplicáveis a sistemas de computação de alto desempenho, e propõe fácil utilização pelo usuário. Um dos principais desafios dos frameworks é escalar o treinamento para múltiplos processadores, como CPUs e GPUs, sem perder desempenho e precisão com a sobrecarga de comunicações. Este trabalho propõe uma avaliação prática de* frameworks *recentes tais como Horovod, LBANN e Tarantella para distribuição de Redes de Aprendizado Profundo.*

1. Introdução

Redes Neurais Artificiais usam modelos matemáticos para lidar com problemas reais com processamento similar a capacidade humana de realizar tarefas [Laine 2003]. Assim como o cérebro humano, Redes Neurais Artificiais Profundas também precisam aprender a partir de dados desconhecidos. Essa rede é composta por camadas compostas por neurônios artificiais que se comunicam entre si para obter a melhor previsão ao final.

O treinamento de uma rede neural pode levar dias, dependendo do modelo e dos dados utilizados. Ambientes de computação de alto desempenho são essenciais para processar aplicações como essas, que lidam com grandes quantidades de dados e executam computações exaustivas. *Frameworks* para distribuir o treinamento de redes de aprendizado profundo facilitam a utilização de múltiplos dispositivos como CPUs, GPUs e FPGAs [Ben-Nun and Hoefler 2019]. Porém, ao implementar abordagens paralelas, essas ferramentas devem preservar aspectos da rede, como garantir que o modelo esteja acessível em todos processos, que os processos tenham os mesmos parâmetros de inicialização, e que todos processos tenham os resultados de outros durante cada época de treinamento. Com isso, o maior desafio destas ferramentas está em escalar o treinamento para mais recursos computacionais e minimizar os custos de computações extra.

Horovod [Sergeev and Balso 2018], LBANN [Van Essen et al. 2015] e Tarantella [CC-HPC 2020] são alguns dos *frameworks* mais recentes para sistemas com CPUs e GPUs. Eles se propõem a facilitar a distribuição do treinamento utilizando poucas linhas adicionais de código. Acredita-se que com uma avaliação prática dessas ferramentas seja possível comparar o desempenho entre elas, identificando os algoritmos utilizados para distribuição e gargalos nas estratégias implementadas. Este estudo também pode auxiliar na decisão de qual *framework* utilizar, além de que otimizações no treinamento de Redes Neurais Artificiais de Aprendizado Profundo podem acelerar diversas pesquisas que lidam com grandes modelos e grandes conjuntos de dados.

2. Metodologia

Os experimentos serão conduzidos na Grid'5000 [Cappello et al. 2005] e no Parque Computacional de Alto Desempenho[1], utilizando GPUs com Tensor e CUDA cores. Será utilizada uma rede neural convolucional com o dataset FashionMNIST e CIFAR-10 para classificação de imagens. A primeira etapa consiste na definição de um projeto experimental que considere fatores/níveis relevantes para configuração dos treinamentos [Jain 1991]. A segunda etapa consiste em selecionar os melhores casos da primeira, aprofundando a análise com instrumentação do código dos *frameworks* com Score-P [Knüpfer et al. 2012]. Os rastros coletados permitirão identificar as fases das execuções levando a propostas de melhorias na implementação dos *frameworks*. A análise com visualização de dados será feita na linguagem R com pacotes da coleção tidyverse[2]. Os *frameworks* com suporte CPU/GPU serão o Horovod, o LBANN e o Tarantella.

3. Considerações Finais

Ao final deste trabalho, espera-se entender, por exemplo, como os *frameworks* distribuem computações entre diferentes processos, quais estratégias são implementadas para paralelização e quais algoritmos são utilizados para sincronização entre processos. Espera-se propor melhorias na implementação de pelo menos um *framework*, tal como otimizações na distribuição de carga entre recursos, avaliando e comparando o desempenho entre as ferramentas. Serão criadas visualizações para análise dos rastros, melhor percepção sobre as fases de processamento e identificação de gargalos.

Referências

Ben-Nun, T. and Hoefler, T. (2019). Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *ACM Comput. Surv.*, 52(4).

Cappello, F. et al. (2005). Grid'5000: a large scale and highly reconfigurable grid experimental testbed. In *The 6th IEEE/ACM International Workshop on Grid Computing, 2005.*, page 8, Washington, US. IEEE.

CC-HPC, C. (2020). Tarantella: Distributed deep learning framework. `http://www.tarantella.org`. Accessed: 2020-11-14.

Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley New York.

Knüpfer, A. et al. (2012). Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope,Scalasca, TAU, and Vampir. In Brunst, H., Müller, M. S., Nagel, W. E., and Resch, M. M., editors, *Tools for High Performance Computing 2011*, pages 79–91, Berlin, Heidelberg. Springer Berlin Heidelberg.

Laine, A. (2003). *Neural Networks*, page 1233–1239. John Wiley and Sons Ltd., GBR.

Sergeev, A. and Balso, M. D. (2018). Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv preprint arXiv:1802.05799*.

Van Essen, B., Kim, H., Pearce, R., Boakye, K., and Chen, B. (2015). LBANN: Livermore big artificial neural network hpc toolkit. In *Proceedings of the Workshop on Machine Learning in HPC Environments*, MLHPC '15, New York, NY, USA. ACM.

[1] http://gppd-hpc.inf.ufrgs.br/

[2] https://www.tidyverse.org/