Utilização da GPU em Data Science

Higor de Oliveira Chaves
Universidade Federal
do Mato Grosso do Sul
Coxim, MS 79400–000
Telefone: (67) 99896-4797
Email: higor.chaves@aluno.ufms.br

Rafael Gonçalves de Oliveira Viana
Universidade Federal
do Mato Grosso do Sul
Coxim, MS 79400–000
Telefone: (67) 99950-7979
Email: rafael.viana@aluno.ufms.br

Ramon da Silva V. dos Santos
Universidade Federal
do Mato Grosso do Sul
Coxim, MS 79400–000
Telefone: (67) 98114–5649
Email: ramon.santos@aluno.ufms.br

***Resumo*—Existe atualmente um grande volume de dados sendo transmitidos pelos mais diversos dispositivos disseminados na sociedade mundial. A tendência do mundo moderno, é que esses dados tornem-se um recurso natural, que como qualquer outro, necessita ser refinado para que possa ser bem utilizado. Para que essa grande massa de dados se torne informação útil e relevante porém, algumas técnicas precisam ser aplicadas, técnicas estas que necessitam de um grande poder computacional. Neste artigo, será demonstrado a diferença de perfomance entre CPU e GPU em algoritmos que necessitam de muito processamento. Para tal objetivo foram utilizadas as bibliotecas TensorFlow e cuDDN juntamente com a plataforma CUDA.**

***Index Terms*—Data Science, GPU, CUDA, TensorFlow.**

I. Introdução

Com a chegada do conceito de *BigData* e com a atual tendência de sensoriar tudo e todos, a rede mundial de computadores tem se tornado uma grande e complexa mina de ouro. Esta mina porém, não contém apenas dados bons e relevantes, para chegar a essas características, deve-se passar por um processo de mineração, onde as informações inúteis são descartadas e as relevantes conservadas. À descoberta de conhecimento, extração e preservação apenas do que é útil, dá-se o nome de *Data Science*.

Assim como na mineração de recursos naturais, a mineração de dados necessita de ferramentas e técnicas robustas para que seja possível realizar um maior e melhor processamento. Estas técnicas no entanto, demandam muito poder de máquina e por conta disso são extremamente custosas. Além disso, a mineração é apenas uma das etapas de um grande e complexo sistema de análise e refinamento de dados.

Para obter informações valiosas, como por exemplo a probabilidade de um produto ser vendido em determinada região, ou a predição de ações na bolsa de valores, os dados são tratados utilizando diversas técnicas de *Data Science* - agrupamento, filtro, classificação, entre outras, que assim como a mineração, demandam muito processamento.

Afim de amenizar a lentidão causada por algoritmos que necessitam de muito recurso computacional, será apresentado neste artigo como a GPU pode se tornar um grande auxiliar no processamento massivo de dados, paralelizando as tarefas e tornando assim o processo como um todo mais veloz.

II. Fundamentação Teórica

A. GPGPU

Além de trabalhos de artistas e desenvolvedores de jogos, trabalhos inovadores com a tecnologia começaram a sugir, assim ouve o nascimento do movimento da GPU de Propósito Geral - *GPGPU*.

B. KDD

Knowledge Discovery in Database ou Descoberta de Conhecimento em Bando de Dados, tem uma função muito importante na produção de conhecimento neste artigo, pois a extração de conhecimento que é usada por esta técnica, não é simplesmente coletar os dados, e sim um conjunto de etapas, para chegar em um resultado refinado.

C. CUDA

CUDA é uma plataforma de computação paralela inventada pela NVIDIA. Ela é capaz de aumentar significativamente a performance computacional ao utilizar e aproveitar a potência da unidade de processamento gráfico GPU.

O CUDA Toolkit conta com um compilador, bibliotecas de matemáticas e ferramentas para depuração e otimização da performance de seus aplicativos, para ficar melhor ele é distribuído gratuitamente, além de sua documentação e manutenção fornecidos pela NVIDIA.

D. cuDDN

É uma biblioteca para o NVIDIA CUDA, para rede neurais profunda que fornece implementações altamente sintonizadas para rotinas padrão, como convolução para trás (propagação), agrupamento, normalização e camadas de ativação. Diversos softwares de redes profundas dependem do cuDDN (Caffe2, MatLab, Microsofot Cognitive Toolkit, TensorFlow, Theano, PyTorch) para aceleracção GPU de alto desempenho, permitindo se concentrar no treinamento de rede neurais e no desenvolvimento de aplicações em vez de gastar tempo no ajuste de desempenho de GPU de baixo nível.

E. TensorFlow

O TensorFlow desenvolvida pela Google é uma biblioteca de técnicas em Inteligência Artificial inovadora e conta com uma comunidade ativa. O TensorFlow pode ser usado por indivíduos em busca de pesquisas ou mesmo grandes empresas que precisam implementar estratégias.

III. Recurso Computacional

Para a realização do teste será utilizado um computador com as seguintes configurações demostrada na Tabela I.

Tabela I
Configuração do Hardware

Hardware	Descrição
SO	Debian 9,Stretch Linux Kernel 4.9
CPU	Intel i5-7200U (3.1 GHz, Cache de 3 MB)
GPU	NVIDIA® GeForce® 940MX de 4GB GDDR5
RAM	8GB, DDR4, 2400MHz

IV. Análise de Técnicas

Existem cálculos computacionais como os utilizados na engenharia, mídia digital e aplicações científicas que exigem grande poder de processamento, onde a CPU não consegue um bom desempenho.

As GPUs atuam como um co-processador e podem acelerar as aplicações devido ao seu poder de processamento massivamente paralelo em relação ao design dos vários núcleos das CPUs.

Nesta sessão iremos apresentar a técnica de multiplicação de Matrizes, onde a mesma e utilzada por diversas outras técnicas [4].

A. Multiplicação de Matrizes

Esse é um exemplo trivial da computação pelo seu grande gasto computacional, um exemplo prático séria algumas redes neurais convolucional de classificação de imagem, que utiliza o produto das matrizes para criação de filtros.

Um exemplo prático pode ser observado na Tabela II, onde foram apresentados três testes de 150^2, 1500^2, 15000^2, todos com valores internos randômicos, a GPU somente foi superior no teste quando o nível de complexidade relativamente grande.

Tabela II
Demonstração

Tamanho da Matriz	Tempo na CPU ms	Tempo na GPU ms
150 X 150	0:00.137	0:00.668
1500 X 1500	0:00.318	0:00.661
15000 X 15000	2:36.198	0:10.329

1) Camada Convolucional: A convolução é uma operação matemática utilizada processamento único para filtrar sinais, identificar padrões em sinais. Em uma camada convolucional, todos os neurônios aplicam operação de convolução às entradas, portanto, são chamados de neurônios convolucionais. O parâmetro mais importante em um neurônio convolucional é o tamanho do filtro, digamos que temos uma camada com tamanho de filtro 5 * 5 * 3. Além disso, suponha que a entrada que é alimentada ao neurônio convolucional é uma imagem de entrada de tamanho de 32 * 32 com 3 canais, como na Figura abaixo [1].

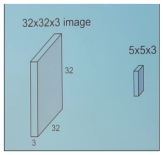

Figura 1. Criação de um filtro com uma parcela da imagen.

Após algumas técnicas divididas em camadas de neurônios (como a camada de agrupamento e a totalmente conectada), o algoritmo converge reconhecendo os padrões através dos pixels multiplicados. O resultado desse processo e a Figura 2, que contém um vetor de padrões reconhecidos [4].

Figura 2. Foto com vetor de padrões reconhecidos

Conclusão

O processamento paralelo em GPU tem grande difirença quando os dados computados são de grande complexidade, o que torna essa tarefa muito custosa para a CPU, tornando assim a GPU uma otima opção ao aproveitar sua paralelização nesses cálculos, porém para cálculos computacionais de baixa complexidade a CPU, se mostrou superior na velocidade do processamento.

Referências

[1] Campos V., Sastre F., Maurici Y., Jordi Torres and Xavier G. *Scaling a Convolutional Neural Network for classification of Adjective Noun Pairs with TensorFlow on GPU Clusters* Barcelona Supercomputing Center - Centro Nacional de Supercomputación (BSC), 2017.

[2] Monard, Maria Carolina and Baranauskas, José Augusto *Conceitos sobre aprendizado de máquina* Sistemas Inteligentes-Fundamentos e Aplicações, 2003.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, ...*TensorFlow: A System for Large-Scale Machine Learning* 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), 2016.

[4] Marcelo H. and Santos S., R. Leal, José Alberto and KIM, HaeYong *Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenhos* Simpósio Brasileiro de Sensoriamento Remoto, 2005.

[5] R. Sujith, *ProjectionNet: Learning Efficient On-Device Deep Networks Using Neural Projections* Google Research, 2017.

[6] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, Fernanda B. Viégas, and Martin W.*Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow* 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2017.

[7] Young J., J. Kim , Jong-Kook K. , A. Mohaisen, and Woojoo L.*Performance of Deep Learning Computation with TensorFlow Software Library in GPU-Capable Multi-Core Computing Platforms* School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea, 2017.