Aprimorando o algoritmo SLIDE-GPU para Problemas de Classificação Extrema

Michel B. Cordeiro[1], **Wagner M. Nunan Zola**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)

Resumo. *A Classificação Extrema é uma categoria de problema de aprendizado de máquina supervisionado, no qual muitas classes precisam ser consideradas, podendo chegar a centenas de milhares ou milhões. Este artigo apresenta uma proposta de melhorias para o algoritmo SLIDE-GPU, que tem demonstrado potencial para resolver eficientemente problemas de Classificação Extrema.*

1. Introdução

Tecnologias de aprendizado profundo têm sido aplicadas a uma ampla gama de cenários nos últimos anos. Redes neurais vêm sendo utilizadas com quantidades crescentes de parâmetros, podendo chegar a bilhões. Nesse contexto, problemas de Classificação Extrema podem envolver redes neurais que processam mais de um milhão de classes e entradas com mais de cem mil dimensões. Esses problemas podem ser encontrados em diversas áreas, como modelagem de linguagem, marcação de documentos em PLN, reconhecimento facial, aprendizado de representações universais de características, previsão de função genética em bioinformática, etc. Entretanto, o alto custo computacional necessário pode tornar inviável o processo de treinamento dessas redes. Para superar essa limitação, diversas técnicas foram propostas, como o algoritmo *Sub-Linear Deep Learning Engine* (SLIDE) [Chen et al. 2020]. A implementação desse algoritmo baseia-se na utilização do paralelismo em CPU para construir tabelas *hash* sensíveis à localidade (LSH) e selecionar neurônios com alta ativação em tempo sublinear. Utilizando apenas uma CPU *multicore*, o SLIDE foi capaz de reduzir drasticamente o tempo de execução tanto do treinamento quanto da inferência, superando a implementação otimizada para GPU do TensorFlow. Isso foi possível porque, para problemas de Classificação Extrema, é possível reduzir o tempo de execução selecionando apenas alguns neurônios para ativar durante cada atualização de gradiente no treinamento, sem prejudicar significativamente o treinamento do algoritmo. Aceleração ainda mais expressiva foi alcançada pelo SLIDE-GPU, proposto por [Meyer and Nunan Zola 2023]. Esse algoritmo faz uso da GPU para acelerar a etapa de ativação na rede neural. Além disso, o SLIDE-GPU utiliza algoritmos que realizam a busca aproximada por vizinhos mais próximos, ou *Aproximate Nearest Neighbor* (ANN), para selecionar os neurônios que devem ser ativados. Para isso, os autores utilizaram a biblioteca FAISS [Johnson et al. 2019], amplamente empregada para realizar busca por similaridade em GPU. Embora a fase de backpropagation ainda seja executada em CPU, o algoritmo híbrido (CPU+GPU) desenvolvido foi capaz de alcançar uma aceleração do processo de treinamento de até 268% em relação ao algoritmo puro em CPU proposto por [Chen et al. 2020]. Na etapa de ativação dos neurônios, o SLIDE-GPU alcançou um tempo de execução 28,09 vezes menor em comparação com a implementação em CPU. Isso sugere que uma aceleração adicional pode ser obtida em trabalhos futuros, por meio da aplicação do paralelismo massivo em todo o processo. Nesse cenário, o presente artigo propõe estratégias para acelerar o SLIDE-GPU. Inicialmente

duas estratégias são consideradas: substituir o algoritmo de seleção de neurônios por outro mais eficiente e implementar todas as etapas do novo algoritmo em GPU, incluindo *backpropagation*, produzindo um algoritmo puro em GPU.

2. Proposta

O algoritmo KNN proposto por [Cordeiro and Zola 2023], denominado KNN-PPW, demonstrou potencial para superar dois algoritmos da biblioteca FAISS quando a pesquisa é realizada em pequenos lotes. Como o SLIDE-GPU utiliza o algoritmo de busca por vizinhos da biblioteca FAISS para selecionar neurônios, e como essa seleção é feita em conjuntos esparsos, este trabalho investigará se a adaptação do KNN-PPW para uso nesse contexto poderia proporcionar ganhos de desempenho. Além disso, mesmo que o SLIDE-GPU não realize a construção e atualização de tabelas *hash*, operações que geralmente são computacionalmente custosas, o algoritmo de ANN utilizado ainda faz uso de estruturas de dados que particionam o espaço de busca, as quais também precisam ser atualizadas quando o espaço de busca é modificado. Esse problema poderia ser eliminado com o uso do KNN-PPW evitando manter estruturas de dados que necessitam de atualizações.

Apesar da esparsidade de computações no SLIDE apresentar desafios para o processamento, considera-se que o uso de técnicas centradas em *warps* poderá habilitar a integração da etapa de *backpropagation* em GPU de maneira efetiva. É possível armazenar na memória da GPU os neurônios que foram ativados em cada camada. Assim, o algoritmo proposto consiste em utilizar cada *warp* para atualizar cada neurônio que foi ativado. Essa estratégia facilita o acesso coalescido à memória e evita a divergência de *threads*. A técnica permite o uso de operações para comunicar dados entre *threads* do mesmo *warp* usando registradores. Dessa forma, cada um dos pesos relacionados aos neurônios ativados pode ser atribuído a uma *thread* do *warp* e, utilizando primitivas de comunicação entre *threads* do mesmo *warp*, é possível evitar o acesso à memória global. Além das melhorias citadas, a implementação totalmente acelerada em GPU reduzirá a necessidade de transferências de dados entre CPU e GPU em diferentes etapas do algoritmo, potencializando uma maior aceleração no novo algoritmo.

Agradecimentos

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Programa de Excelência Acadêmica (PROEX)

Referências

Chen, B., Medini, T., Farwell, J., Tai, C., Shrivastava, A., et al. (2020). SLIDE: In defense of smart algorithms over hardware acceleration for large-scale deep learning systems. *Proceedings of Machine Learning and Systems*, 2:291–306.

Cordeiro, M. B. and Zola, W. M. N. (2023). KNN paralelo em GPU para grandes volumes de dados com agregação de consultas. In *Anais do XXIV Simpósio em Sistemas Computacionais de Alto Desempenho, WSCAD'23*, pages 253–264. SBC.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Meyer, B. H. and Nunan Zola, W. M. (2023). Towards a GPU accelerated selective sparsity multilayer perceptron algorithm using K-nearest neighbors search. In *Workshop Proceedings of the 51st International Conference on Parallel Processing*, ICPP W '22.