Project Proposal

07-300

Justin Zhang
https://www.andrew.cmu.edu/user/justinz/

1 Project Description

I will be working with Professor Rashmi Vinayak of the CS department and TheSys group on adapting current matrix approximation techniques and applying them to Deep Learning Recommendation Machine (DLRM) systems. Recommendation systems are pivotal to a user's experience on many web applications such as social media and online marketplaces. Conventionally, recommendation systems have been modeled as click-through rate (CTR) prediction, but recently work has shown improved recommendations through a shift in architecture towards neural networks. These improvements intuitively stem from a recommendation system's need to place a user into their "best" category based on their preferences and the inherent abundancy of categorical data. A deep learning, neural network approach addresses both of these characteristics by having recommendation systems further learn from previous user behaviors and modeling categorical data using embedding techniques, which are much more rich representations than traditional hot vectors.

The current state of the art in neural network archicture based recommendation systems is DLRM. Developed and open-sourced by Facebook, the DLRM model handles categorical features with embeddings and processes continuous features with a multilayer perceptron (MLP). At a high level, these features are then fed to a second level where interaction features are learned and finally fed to the top level MLP.

As with many deep learning models, DLRM suffers from memory usage and speed performance. A current solution to this problem is parallelism; data and model parallelism provide significant speedup, but does not address memory usage. Thus, to address both of these issues, we examine matrix approximations as an alternative solution. There has been much recent work

in reducing parameter size through matrix approximations in other architectures such that the reduction in accuracy is not too much. An often occurrence in matrix approximation is that reduction will lead to small training error increase but a significant test error increase. Thus, many of these matrix reduction methods make insights about their specific architecture to make sizeable parameter reduction while keeping predictive accuracy high.

While the computer architecture for these matrix approximation techniques are studied extensively on CPUs and GPUs, we aim to optimize approximation on Cloud Tensor Processing Units (TPUs), which are processing units specialized for TensorFlow application. Our objective is to start with adapting current state of the art approximation techniques for TPUs to provide insight on future work and optimizations. If we are successful, we will have significantly improved the downsides of the DLRM model. Memory and model processing is expensive, so reducing these metrics will allow for much more general use of a powerful recommender model.

Our work will be primarily done through TensorFlow and Google Cloud. Specifically, we will be working with the TensorFlow implemented, open source DLRM model.

Major challenges in this project are implementing other matrix approximations for our specific architecture and eventually, an approximation specialized for DLRM on TPU architecture. It will be theoretically challenging to discover and understand the right approximation for novel architecture. Inversely, it may be equally as challenging to understand why certain approximations may not work well and whether the reason is the TPU architecture or the inherent recommender system.

2 Project Goals

2.1 75 % Project Goal

1. Determine single TPU specific optimizations for embedding lookup and store runtime
2. Analyze metrics across different architectures (GPU/CPU) and setups

2.2 100 % Project Goal

1. Determine single/accelerator setup TPU specific optimizations for embedding lookup and store runtime
2. Analyze metrics across different architectures (GPU/CPU) and setups

3. Apply multiple different matrix approximation techniques and gain insight on how they scale

2.3 125 % Project Goal

1. Determine single/accelerator setup TPU specific optimizations for embedding lookup and store runtime
2. Analyze metrics across different architectures (GPU/CPU) and setups
3. Apply multiple different matrix approximation techniques and gain insight on how they scale
4. Use recently developed theory of TPU storage optimization to draw more theoretical claims about further optimizations.

3 Project Milestones

3.1 First Technical Milestone

By the first milestone, I hope to have a better understanding of current matrix approximation techniques like random projections. Furthermore, I would like to perform a deeper literature review in TPU and DLRM.

3.2 First Biweekly Milestone: February 1st

I will have developed the basic framework in which to run TPU experiments on. This includes setting up google cloud, learning enough tensorflow, and implementing the DLRM.

3.3 Second Biweekly Milestone: February 15st

I hope to have determined the appropriate dataset to test metrics of my initial model on. Furthermore, I hope to implement a matrix approximation or find a library which does it for me. If time permits, I will have run some test runs.

3.4 Third Biweekly Milestone: March 1st

At this point, I want to have baseline results for TPU and CPU/GPU. I want to learn more advanced matrix approximation techniques while applying basic ones such as SVD and collecting

the results.

3.5 Fourth Biweekly Milestone: March 15th

I will have decided which scenarios in which to test the randomized algorithm against the optimal algorithm. I will have started running simulations of these scenarios.

3.6 Fifth Biweekly Milestone: March 29th

At this point, I should have most of my analysis done! I will start drafting a paper detailing significant results. This will give me insight for possible future work (I believe there is a lot of development to be done here especially since the research on TPU architecure is so sparse).

3.7 Six Biweekly Milestone: April 12th

I should have the draft done by now. I will continue writing my paper and consolidating results into a poster.

3.8 Seven Biweekly Milestone: April 26th

The paper will be finalized and ready for publication. Work for a follow up paper will hopefully be started from the insights gained from this project.

4 Literature Search

There has been recent interesting work about transparency in social choice theory [**?**]. While kidney exchange is a rather distinct problem, the motivation for transparency in a social choice setting is rather similar to the need for transparency in kidney exchange. Moreover, there is rich literature which has explored fairness, which I have mostly completed in an initial literature review [**?**] [**?**] [**?**][1] [**?**]. I will have to gain more background on the graph theory techniques used in the papers' work cited sections [**?**][**?**][**?**].

5 Resources Needed

We will be using google cloud, jupyter notebook, and tensorflow. The google cloud credits should come from the google team that Professor Vinayak is in collaboration with.

References

[1] Pradeep Singh, Pijush Dutta Pramanik, Avick Dey, and Prasenjit Choudhury. Recommender systems: An overview, research trends, and future directions. *International Journal of Business and Systems Research*, 15:14–52, 01 2021.