ME759 Final Project Proposal

Project Title: Implement einsum using CUDA in PyTorch

Link to git repo for project: https://github.com/ShawnZhong/pytorch-einsum-cuda

Problem statement: explain in clear terms what you want to accomplish.

We want to use parallel computing techniques to implement a CUDA version of einsum function in PyTorch.

Motivation/Rationale: explain why you chose to work on this project.

We found that using einsum to perform tensor operation is much slower than manually performing tensor operation. The expected behavior is that einsum should not be slower than calling einsum. Also, we found that the einsum function is not implemented in CUDA version, so we would like to implement einsum in the CUDA version and see the improvements in performance.

```
[28]  1  import torch
      2  a = torch.rand(20000000, 1, device = "cuda")
      3  b = torch.rand(20000000, 1, device = "cuda")
```

```
[29]  1  %time (a * b).sum(dim=1); torch.cuda.synchronize()
```

```
CPU times: user 2.69 ms, sys: 792 µs, total: 3.48 ms
Wall time: 3.18 ms
```

```
[30]  1  %time torch.einsum("ij,ij->i", a, b); torch.cuda.synchronize()
```

```
CPU times: user 180 ms, sys: 114 ms, total: 294 ms
Wall time: 295 ms
```

Explain how you contemplate going about it: indicate if you'll use GPU/OpenMP/MPI parallel computing, what libraries, etc. Indicate what algorithms/approaches you are considering.

We will use GPU, especially CUDA, to implement einsum for PyTorch. Since it is about calculations on sequential data, we will also try to utilize cache to improve the performance.

ME759 aspects the proposed work draws on: bulleted list, be brief

- We will use CUDA we learned from the class to implement the function
- We would utilize what we learned in class the strategies to improve CUDA execution time.

Team member[s]: (if more students, list *alphabetically* according to last name)

- **Name**: Yuhan Liu, Ziyi Zhang, Shawn Zhong
- **Email: yliu738@wisc.edu, zzhang765@wisc.edu, shawn.zhong@wisc.edu**
- **Home department + advisor**: We are all CS undergrads, so none of us has an advisor

- **Student's role in the project:**
    - Yuhan Liu: Profiling and benchmark
    - Ziyi Zhang: Implementation of einsum in CUDA
    - Shawn Zhong: Further optimization and report

Deliverables: what you expect to deliver on 05/06/2020, 7:45 am: code, input files, tech report, etc.

We would like to open a Pull Request in PyTorch and implement the einsum in CUDA. We will submit the code we implemented, tech report that includes the profiling before and after using the einsum we implemented, and input data we used to test the function.

How you will demonstrate what you accomplished: this is particularly important if what you do is a small piece of a bigger project that you will continue to pursue after wrapping up ME759.

We will benchmark both the original implementation and different possible optimizations of it and compare the running time of all.

Milestone: indicate what will be accomplished by April 23 milestone (9 pm).

1. Profile the current implementation of einsum in PyTorch on different sizes of the matrix and different types of operations
2. Set up the development environment for PyTorch
3. Get familiar with the codebase
4. Propose different possible schemes for optimizations of the original code
5. Try a few of those schemes

Other remarks: say here anything else that you think Dan should be aware of and doesn't fall within any other category above.