Lab 2 CSE 5194.01

Luyu Liu, Subash Chebolu, Zicong Zhang[[1]](#footnote-1)

{liu.6544, chebolu.1, zhang.5157}@osu.edu

Disclaimer: Due to the limit number of group members, after consulting Dr. Panda, we change the number of models from 3 to 2. After the change, it is much more reasonable to distribute and balance the workloads for each group member.

1. Choose two different DNN models available for your chosen dataset. a. AlexNet, ResNet, and MobileNet for ImageNet dataset b. Use any model for Word2Vec that you can find for your assigned DL framework. c. Linear Model, Linear Model with Crosses, and DNN for Criteo Click Logs. (Luyu & Zicong)

For this experiment, we chose the Word2Vec 1- billion words dataset.

1. Choose two different DNN models available for your chosen dataset. a. AlexNet, ResNet, and MobileNet for ImageNet dataset b. Use any model for Word2Vec that you can find for your assigned DL framework. c. Linear Model, Linear Model with Crosses, and DNN for Criteo Click Logs. (Luyu & Zicong)

For the first training model, we still chose the gated convolutional networks [2] as the lab 1: a major contribution of the model is that it was a new attempt to use a non-recurrent approach to attain near performance as recurrent neural networks. The model is based on convolutional networks with a gating mechanism. The code is originated from a GitHub repository [3], which was developed from the methods from the paper and later adjusted by Luyu Liu [4].

The other model we choose is transformer-XL [5], [6], which builds on the original transformer model that has been very popular in the field of natural language processing. Transformer model adds a recurrent layer to in the attention layer to further enhance to the model’s capability of predicting variable length input and long-range dependencies.

1. (20 points) Run the experiments for each DNN using a single node (If you have these numbers from Lab #1, please re-use these to save SUs. Otherwise, you can re-run these experiments.)
2. Run the GPU version

The GPU version is run on the Owens P100 node with one node.

1. Run the CPU version

The CPU version is run on the Owens CPU cluster with 28 cores in one node.

1. Report throughput (training time) in terms of samples/second or images/second for both CPU and GPU
2. Vary the batch size and find out the best batch size that gives you the highest throughput.

**[Gated CNN] (Luyu)**

|  |  |  |
| --- | --- | --- |
|  | accuracy | training time |
| Batch size = 32 | 0.1077 | 538 |
| Batch size = 64 | 0.4328 | 291 |
| Batch size = 80 | 0.4323 | 273 |

Table 1 Average epoch time among all epochs for each GPU experiment

|  |  |  |  |
| --- | --- | --- | --- |
|  | CPU with 7 threads | CPU with 14 threads | CPU with 28 threads |
| Batch size = 32 | 0.4018 | 0.4021 | 0.0898 |
| Batch size = 64 | 0.4297 | 0.4303 | 0.4295 |
| Batch size = 80 | 0.4341 | 0.4303 | 0.4353 |

Table 2 Final accuracy among all epochs for each CPU experiment

|  |  |  |  |
| --- | --- | --- | --- |
|  | CPU with 7 threads | CPU with 14 threads | CPU with 28 threads |
| Batch size = 32 | 4109.6 | 3789.2 | 5235 |
| Batch size = 64 | 3683.4 | 2178.6 | 3185.2 |
| Batch size = 80 | 2044.4 | 1843.6 | 2038.6 |

Table 3 Average epoch time among all epochs for each CPU experiment

The results show that batch size = 80 is the best option. We will continue our training with this batch size.

**[Transformer-XL] (Zicong)**

1. (40 points) After you have the data for the best batch size for a single node, run the experiment with the best batch size for 1, 2, 4, and 8 nodes.
2. Run for CPU only

The GPU version is run on the Owens P100 node with 2-8 node. Considering all the time and computational budget, instead of running the code on one node again, we use the old data in the last lab.

1. Run for GPU only

The CPU version is run on the Owens CPU cluster with 28 cores in 1-8 node.

1. Report the throughput for both versions
2. Report the scalability/speedup for multiple nodes by creating a graph that presents images/second on the y-axis and #nodes on x-axis.

**[Gated CNN] (Luyu)**

|  |  |  |
| --- | --- | --- |
| # Node | accuracy | training time (s) |
| 1 | 0.4114 | 273/278 |
| 2 | 0.4085 | 328 |
| 4 | 0.4042 | 197 |
| 8 | 0.3910 | 101 |

Table 4 benchmark for each multi-node GPU experiment

|  |  |  |
| --- | --- | --- |
| # Node | accuracy | training time (s) |
| 1 | 0.4353 | 2038 |
| 2 |  | 328 |
| 4 |  | 197 |
| 8 | 0.4012 | 1132 |

Table 5 benchmark for each multi-node CPU experiment

1. (20 points) What can you conclude from this study? Write a few paragraphs to explain your results and the insights in-depth.

1. In first name alphabetical ordering [↑](#footnote-ref-1)