System Programming 2019 Fall

Programming Assignment # 4

Due: 23:00 Sun, Jan 5, 2020

1.Problem Description

In assignment 4, you are required to implement a multiclass classifier with thread. You should train a model to classify handwritten digits from MNIST dataset. And the most important part is that you have to accelerate the matrix multiplication (let it parallelly) by thread.

In training, you should decide how many iterations you train your classifier on your own.
For each iteration, you update your classifier with：

$$\hat{y} = \text{softmax}(\text{X} * \text{W}) \qquad (1)$$

$$\text{softmax}(\text{z}_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad (2)$$

$$\text{w} = \text{w} - \text{lr} * \text{w}_{\text{grad}} \qquad (3)$$

$$\text{w}_{\text{grad}} = X^T(\hat{y} - y) \qquad (4)$$

X：training data matrix (60000 * 784)
W：weight matrix (784 * 10)
*(If you consider adding bias to your classifier, you can let W to be 785*10, and add a column with 1's to X.)*
y_hat：predicted label (60000 * 10)
y：true label, you need to transform each label to one-hot. (60000 * 10)
(if label = 2 => [0, 0, 1, 0, 0, 0, 0, 0, 0, 0])
lr：learning rate (scalar)

You need to create threads to accelerate the matrix multiplication in (1). We will tell you how many threads you should create, and each thread should calculate [60000 / thread_num] rows multiplication.
(For example, if thread_num = 1000, each thread should be responsible for "60" rows ([60*784] * [784*10]) multiplication in (1).)

To evaluate the accuracy, you may choose the label with largest probability from 10 classes for each image.

2. Format of Inputs & Outputs

Input：MNIST dataset (including 4 files)

X_train：60000 images, 784 pixels for each image, value：0~255

y_train：60000 labels, value：0~9

X_test：10000 images

y_test：10000 labels

(Note: You can use X_test and y_test to check your classifier's accuracy, but don't use them to train your classifier.)

data link : **https://drive.google.com/drive/folders/1wips8uJtKFllnXVzu2fDC_SRjbxaxiD2?usp=sharing**

Output：result.csv

format：

id	label
0	6
1	0
2	7
3	8
4	1
5	1
6	5
7	3
8	2
9	9
10	2

3.Sample Execution

./hw4 [X_train] [y_train] [X_test] [number of threads]

(compile：gcc hw4.c -lm -lpthread -O3 -o hw4)
(When we mark your assignment, we will use our private testing dataset (size = 10000), your program should output the result.csv file according to the testing data we specified.)

4.Grading

There are 4 (+2) subtasks in this assignment. You can get 7 (+5) points if you finish all of them. We will test your code on csie workstation.

1. (2 points) Your classifier can finish the training on time and have at least 0.80 accuracy in testing data.[# thread = 1]

We would run your program with the same training data and our private testing data. Your program have to finish your training in **35 sec**.

2. (2 points) Your classifier can finish the training on time and have at least 0.80 accuracy in testing data. [# thread = 100]

We would run your program with the same training data and our private testing data. Your program have to finish your training in **25 sec**.

3. (Report, 2 points) Compare the execution time with different thread number

You should find out the relation between execution time and thread number, you may use line chart or form to describe the relation.

4. (Report, 1 points) Compare the instructions number with different threads numbers

You should find out the relation between instruction number and thread number, you may use line chart or form to describe the relation.

[how to get the instruction number]

You can use Perf to get the instruction count, the tool has already installed in all workstations.

Usage：

$ perf stat -e instructions:u -v ./hw4 ……

5. (Bonus, 3 points) Any other parallel method ?

You could choose another parallel method with thread instead of row parallel. Please describe your method, performance, bonus code file name, and how to execute it in report.

6. (Bonus, 2 points) More accuracy?

With the time limit in 1. and 2., 10 students who have the highest (top 10) accuracy performance can get 2 extra points as bonus.

6.Submission

Your assignment should be submitted to CEIBA before the deadline, or you will get 0. At least 3 files should be included:

1. hw4.c
2. Makefile (as well as other *.c files)
3. report.pdf
4. bonus code (optional, for Bonus)

7.Reminder

1. Plagiarism is **STRICTLY** prohibited.
2. Different from the previous assignment, we do not accept the late submission.
3. If you have any question, please feel free to contact us via email *ntucsiesp@gmail.com* or come to R302 during TA hours.
4. Please start your work ASAP and do not leave it until the last day!
5. **Good luck!**

7.Reference

https://stats.stackexchange.com/questions/235528/backpropagation-with-softmax-cross-entropy?fbclid=IwAR2ysI7rg59ozHVNlo0eUdzx0G33LI_bqrrmgatIknbX-Ib26orl86fgJlQ