

# CS 744 Homework2

Rui Huang, Zhenyu Zou, Zheng Liu

October 6, 2019

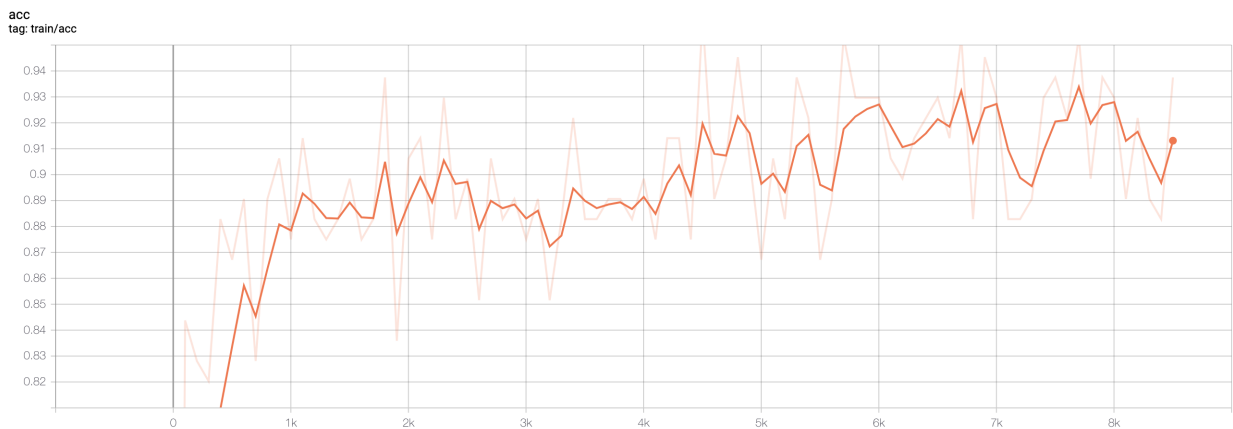
## 1 Part 1

We implemented logistic regression using Tensorflow. We ran it in both single node and cluster mode, using synchronous SGD and asynchronous SGD. We plotted the accuracy and loss curve w.r.t training steps, using tensorboard. We also compared CPU, memory and network usage.

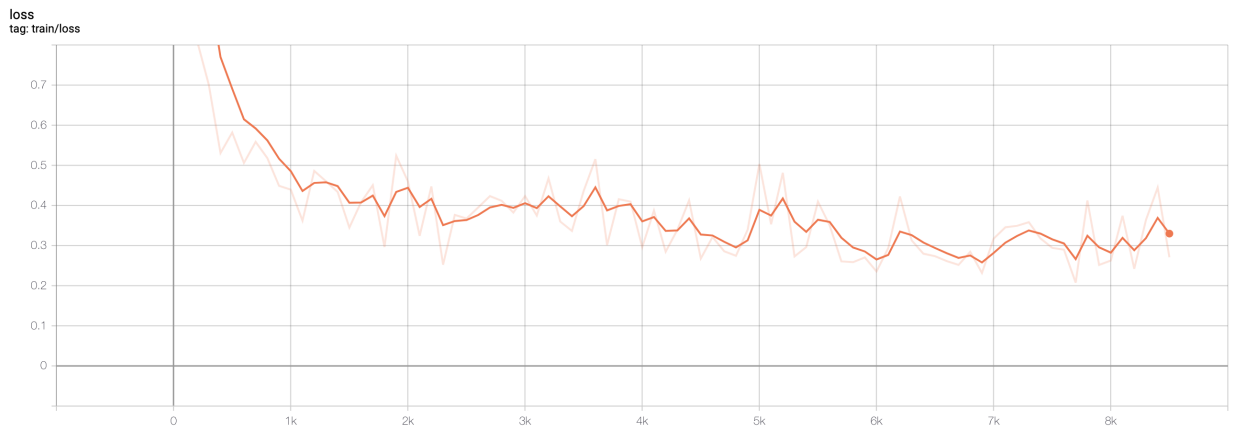
### 1.1 Task 1

There're 60,000 images in the training set and 10,000 images in the test set. We set batch size as 128. And we ran 20 epochs. So there're in total around 8.5K training steps, as shown below. The test accuracy is 91.8%.

Here's the accuracy curve for single node.



Here's the loss curve for single node.

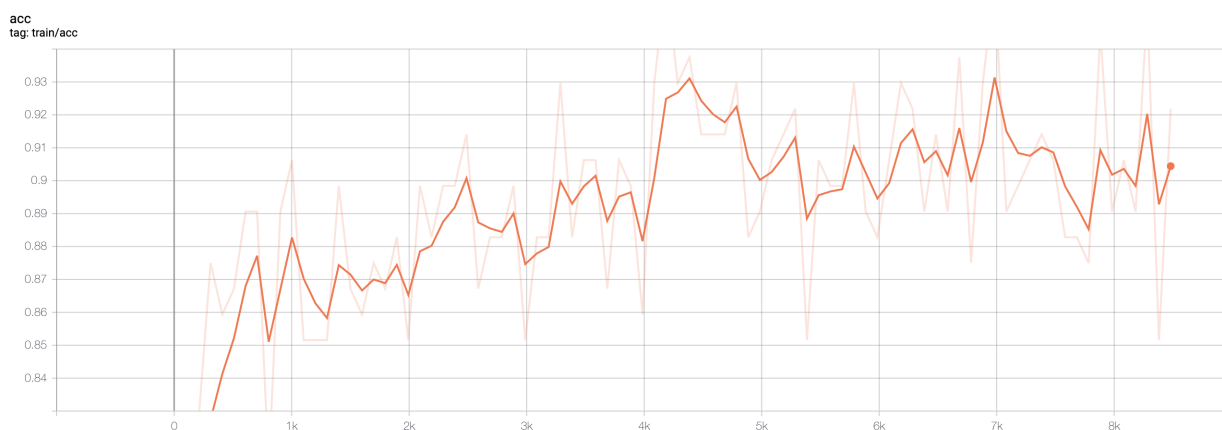


We can see that a simple logistic regression model can get an accuracy of around 91.5%. And the loss is about 0.3. It's hard for it to improve beyond this.

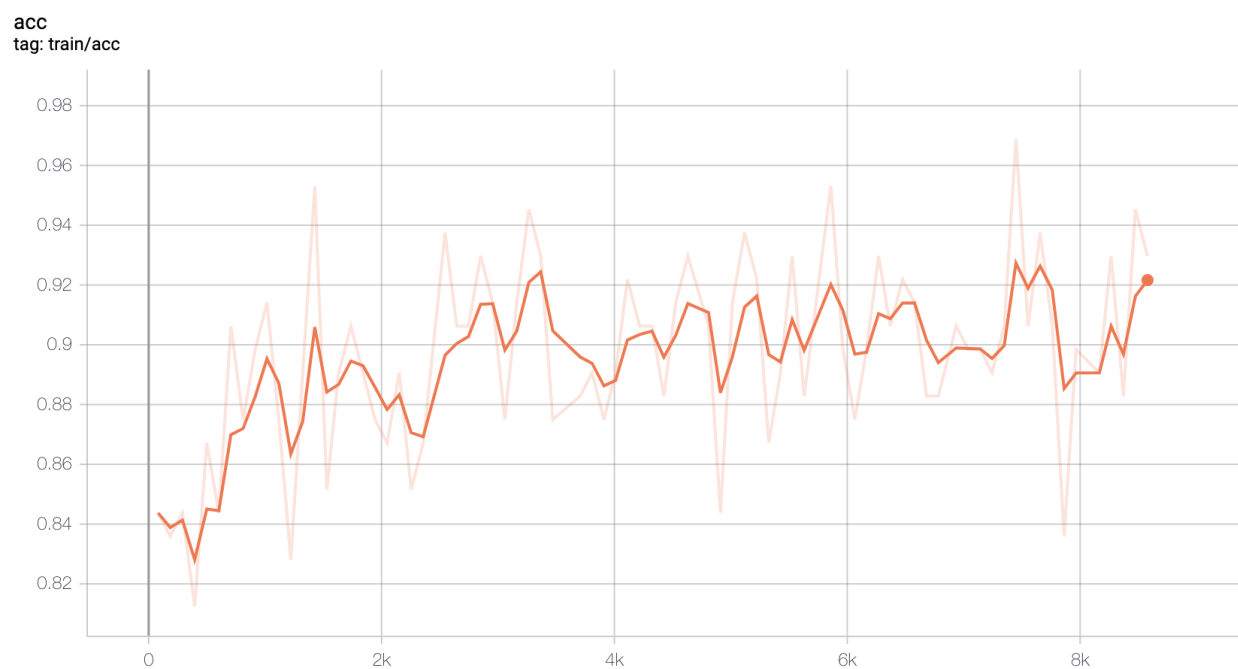
## 1.2 Task 2

We run the model in a cluster of 3 nodes. Same parameter setting as above. The test accuracy for Sync SGD is 91.74%. Async SGD is 91.57%.

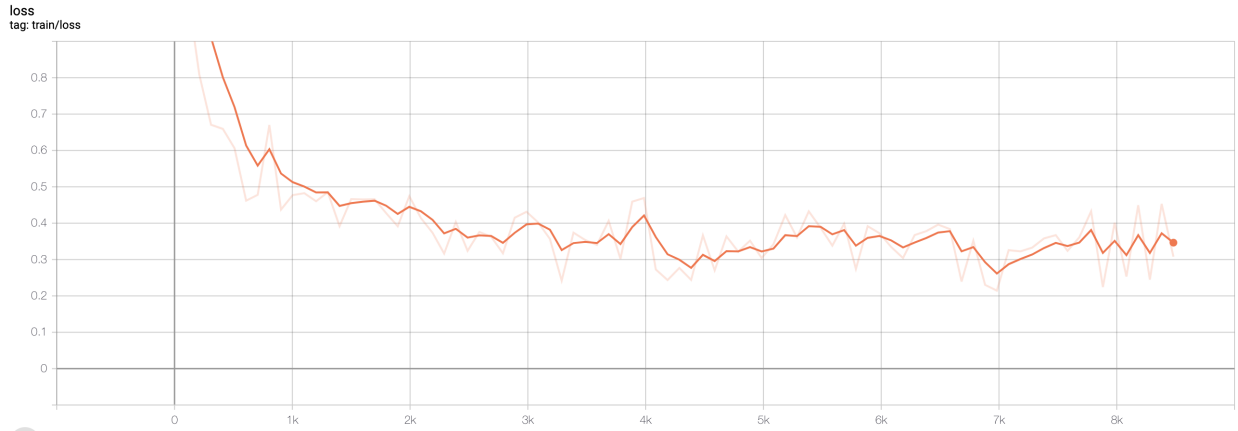
Here's the accuracy curve for synchronous SGD.



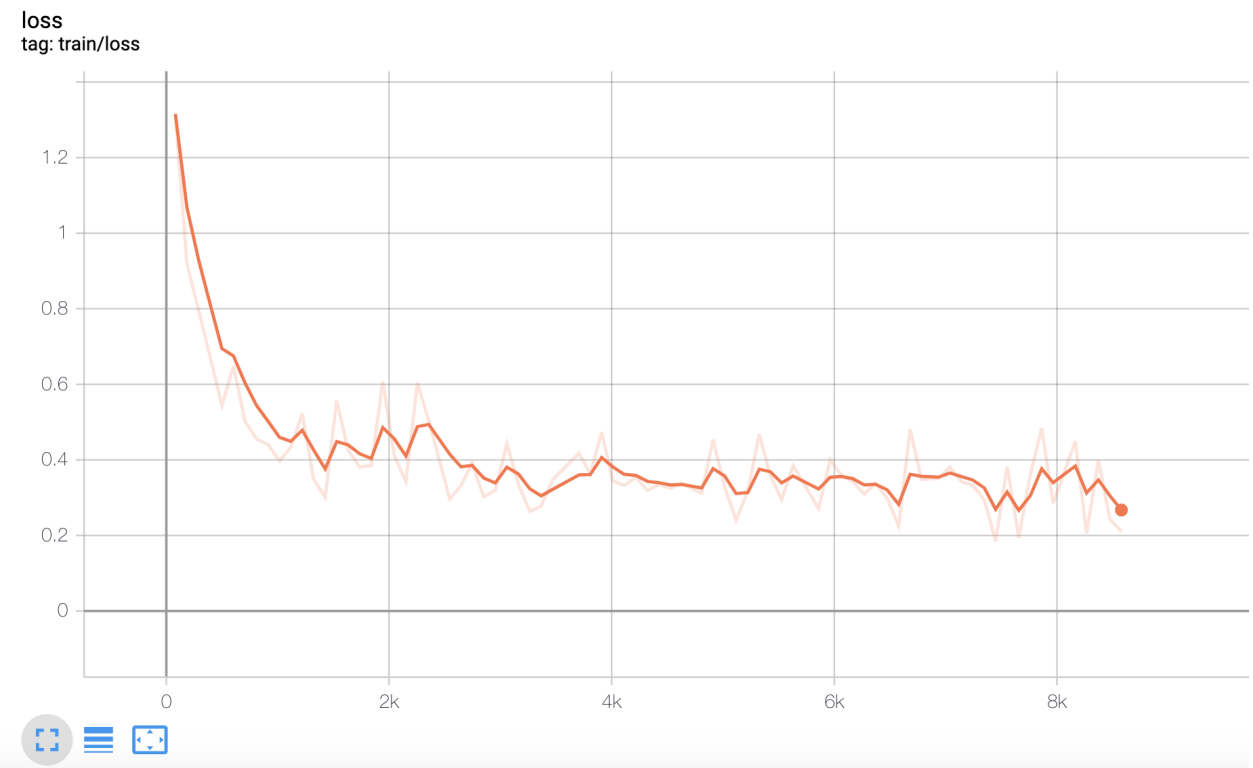
Here's the accuracy curve for asynchronous SGD.



Here's the loss curve for synchronous SGD.



Here's the loss curve for asynchronous SGD.



We can see that the accuracy and loss for synchronous SGD and asynchronous SGD are almost the same, and there seems not to be significant difference.

Here's the CPU, memory and network usage comparison. We use dstat to get the data.

|                       | 1 node | 3 nodes(Sync)     | 3 nodes(Async)    |
|-----------------------|--------|-------------------|-------------------|
| Average CPU usage     | 25%    | 33%, 20%, 20%     | 33%, 20%, 20%     |
| Average memory usage  | 870M   | 1060M, 760M, 680M | 1060M, 760M, 610M |
| Average network usage | 20K    | 8M, 4M, 4M        | 8M, 4M, 4M        |

## 2 Part 2

We implemented LeNet with 7 layers. We used Adam optimizer instead of SGD, which significantly improves convergence. We set batch size as 64. And we ran 20 epochs, achieving test accuracy around 99.0%.

### 2.1 Task 1

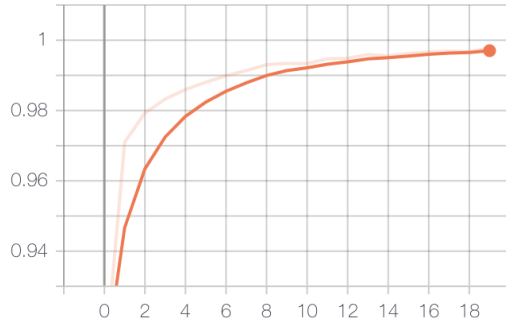
We compared accuracy and loss curve for different number of machines, 1, 2 and 3. The test accuracy for 1 node is 99.08%. 2 nodes 99.13%. 3 nodes 98.9%. The running time for one epoch is 17s, 12s and 12s respectively.

We can see that actually, the accuracy and loss curve are not significantly different.

Here's the CPU, memory and network usage comparison.

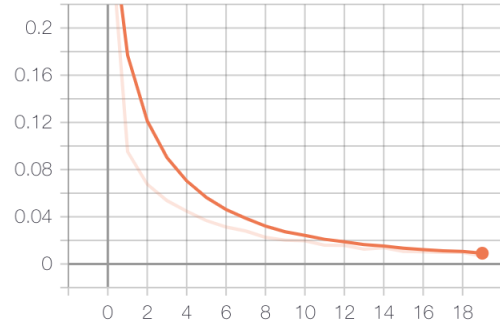
|                       | 1 node | 2 nodes      | 3 nodes             |
|-----------------------|--------|--------------|---------------------|
| Average CPU usage     | 67%    | 58%, 57%     | 51%, 52%, 52%       |
| Average memory usage  | 4989M  | 3110M, 2906M | 2359M, 2250M, 2250M |
| Average network usage | 21K    | 9.2M, 9.5M   | 8M, 8M, 8M          |
| Average epoch time    | 17s    | 12s          | 12s                 |

epoch\_acc



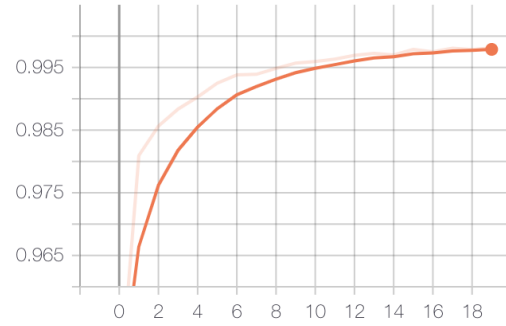
(a) 1 node

epoch\_loss



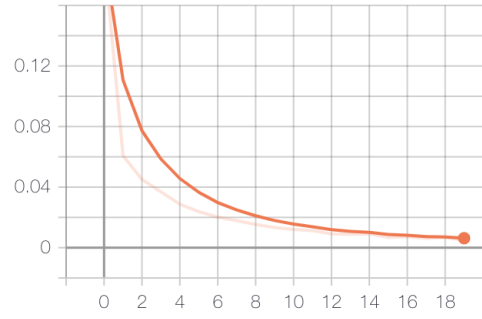
(b) 1 node

epoch\_acc



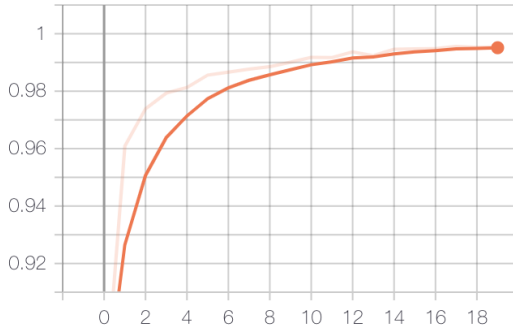
(c) 2 nodes

epoch\_loss



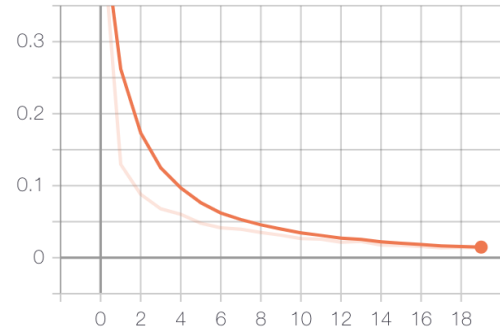
(d) 2 nodes

epoch\_acc



(e) 3 nodes

epoch\_loss



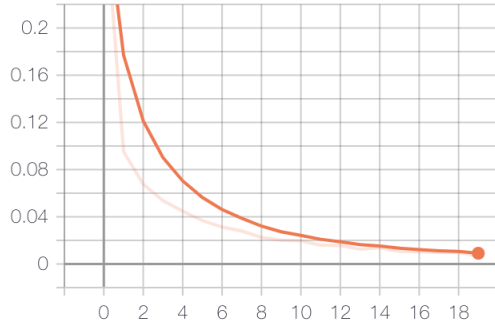
(f) 3 nodes

## 2.2 Task 2

We compared different batch sizes, 8, 64 and 512. The results are as below. The accuracy is 98.9%, 98.9% and 98.0%, respectively. The running time for each epoch is 45s, 12s and 14s, respectively.

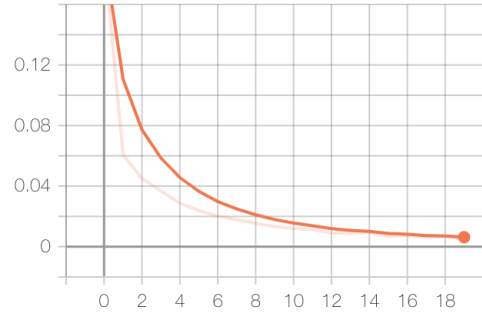
Here's the loss curve for different batch sizes.

epoch\_loss



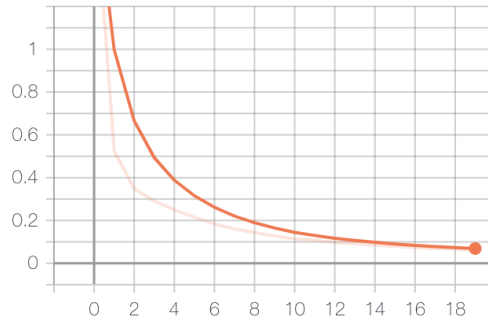
(g) batch size 8

epoch\_loss



(h) batch size 64

epoch\_loss



(i) batch size 512

Here's the CPU, memory and network usage comparison.

|                       | batch size 8        | batch size 64       | batch size 512      |
|-----------------------|---------------------|---------------------|---------------------|
| Average CPU usage     | 39%, 40%, 40%       | 51%, 52%, 52%       | 43%, 44%, 45%       |
| Average memory usage  | 2239M, 2209M, 2042M | 2359M, 2250M, 2250M | 2355M, 2320M, 2000M |
| Average network usage | 17M, 18M, 17M       | 8M, 8M, 8M          | 0.9M, 1M, 1M        |
| Average epoch time    | 45s                 | 12s                 | 14s                 |