Review Questions 6
ID2223
Group: Nicolas Oulianitski Essipova, Peter Lakatos, Marios Chatiras

1. Modern ML applications consist of large models that can use an immense amount of training data. A technique to improve and accelerate the process is to employ more than one system during training and spread the workload across multiple devices. The two main approaches are model and data parallelization. In model parallelization, the model is divided across multiple machines, and every system is responsible for the training of a subset of the model. In the end, these subsets are combined, and the model is whole. In data parallelization, the entity of the model is on every device, and the training data are split into mini-batches. During training, the machines are communicating to synchronize the parameters across all instances.

2. There are primarily two distribution strategies to consider.

   First one to consider is the *MultiWorkerMirroredStrategy*, whose main weakness is that it requires the model to fit into the RAM on every server and/or device. In a nutshell, you first replicate the model across all servers and then supply each one with different data batches for training, thus each model will compute its own gradients. After all computation is completed; the mean is calculated which is then used across all the models to perform gradient descent.

   Second to consider is called the *ParameterServerStrategy*, which performs asynchronous data parallelism (unlike previous one, which performed mirror data parallelism). This strategy is often slower and more difficult to deploy – but it is of interest if your model is too huge to fit into RAM. What it does is that it replicates the model across all devices on all workers where they have their own training loop that runs asynchronously with the others. With the parameters also sharded across all parameter servers, they all get their own data batches and fetch the latest parameters from the parameter servers, then computes the gradients with these before sending the new parameters back to the parameter servers. The parameter servers then perform gradient descent with these new gradients.

3. Gradient quantization means you reduce the number of bits that represent a value, such as in 16-bit (FP16) or even 8-bit (INT8). Research has shown that this does not incur a significant loss in accuracy. Even lower bits are an active field of research.

   Gradient sparsification is a technique wherein you drop some of the coordinates of the gradient while amplifying the remaining coordinates to ensure the sparsified gradient is unbiased.

4.

| | | | | |
|---|---|---|---|---|
| Worker A | 17 | 11 | 1 | 9 |
| Worker B | 5 | 13 | 23 | 14 |
| Worker C | 3 | 6 | 10 | 8 |
| Worker D | 12 | 7 | 2 | 12 |

Step 1

| | | | | |
|---|---|---|---|---|
| Worker A | | 11 | 1 | 21 |
| Worker B | 22 | | 23 | 14 |
| Worker C | 3 | 19 | | 8 |
| Worker D | 12 | 7 | 12 | |

Step 2

| | | | | |
|---|---|---|---|---|
| Worker A | | 11 | 13 | |
| Worker B | | | 23 | 35 |
| Worker C | 25 | | | 8 |
| Worker D | 12 | 26 | | |

Step 3

| | | | | |
|---|---|---|---|---|
| Worker A | | 37 | | |
| Worker B | | | 36 | |
| Worker C | | | | 43 |
| Worker D | 37 | | | |

Step 4

| | | | | |
|---|---|---|---|---|
| Worker A | 37 | 37 | | |
| Worker B | | 37 | 36 | |
| Worker C | | | 36 | 43 |
| Worker D | 37 | | | 43 |

Step 5

| | | | | |
|---|---|---|---|---|
| Worker A | 37 | 37 | | 43 |
| Worker B | 37 | 37 | 36 | |
| Worker C | | 37 | 36 | 43 |
| Worker D | 37 | | 36 | 43 |

Step 6

| | | | | |
|---|---|---|---|---|
| Worker A | 37 | 37 | 36 | 43 |
| Worker B | 37 | 37 | 36 | 43 |
| Worker C | 37 | 37 | 36 | 43 |
| Worker D | 37 | 37 | 36 | 43 |