

image.png
PNG 152.64KB

For real-world models trained on lots of data, the training of neural networks is parallelized and accelerated by running workers on distributed resources, such as clusters of GPUs. In this question, we will explore three popular distributed training paradigms:

All-to-All Communication: Each worker maintains a copy of the model parameters (weights) and processes a subset of the training data. After each iteration, each worker communicates with every other worker and updates its local weights by averaging the gradients from all workers.

Parameter Server: A dedicated server, called the parameter server, stores the global model parameters.

The workers compute gradients for a subset of the training data and send these gradients to the parameter server. The server then updates the global model parameters and sends the updated weights back to the workers.

Ring All-Reduce: Arranges n workers in a logical ring and updates the model parameters by passing messages in a circular fashion. Each worker computes gradients for a subset of the training data, splits the gradients into nequally sized chunks and sends a chunk of the gradients to their neighbors in the ring. Each worker receives the gradient chunks from its neighbors, updates its local parameters, and passes the updated gradient chunks along the ring. After $n-1$ passes, all gradient chunks have been aggregated across workers, and the aggregated chunks are passed along to all workers in the next $n-1$ steps. This is illustrated in Fig. 1 (attached image).

For each of the distributed training paradigms, fill in the total number of messages sent and the size of each message. Assume that there are n workers and the model has p parameters, with p divisible by n . Fill in the three unknown BLANKS 1, 2, and 3:

	Number of Messages Sent	Size of each message
All-to-All	BLANK_1	p
Parameter Server	2n	BLANK_2
Ring All-Reduce	$n(2(n-1))$	BLANK_3

I paste in the question context and part (a) directly from the pdf of the homework. I make no adjustments to weird formatting and potentially ambiguous notation. I also don't give any context, such as "this is a homework for CS 182 Deep Learning class at Berkeley", and I don't write including any interesting prompting diction that papers like Large Language Models as Optimizers (<https://arxiv.org/abs/2309.03409>) mention, for example "take a deep breath".

I also include Figure 1. "Example of Ring All-Reduce in a 3 worker setup" as an image input. However, Deepseek mentions that it is only able to extract text from images, so it is reasonable to believe it would've performed equally well with just the text prompt.

I do a quick txt formatting to create a make-shift table. This is more for viewer pleasure than anything.

Thought for 38 seconds

- First, I need to fill in the blanks for the table comparing three distributed training paradigms:

All-to-All Communication, Parameter Server, and Ring All-Reduce. The table has columns for "Number of Messages Sent" and "Size of each message". There are n workers, and the model has p parameters, with p divisible by n .

Let's start with All-to-All Communication.

From the description: "each worker communicates with every other worker". So, for n workers, each worker needs to send messages to every other worker. That means each worker sends messages to $n-1$ other workers. **Since there are n workers, the total number of messages sent would be $n * (n-1)$.** But in message counting, we might consider each pairwise communication as a message, so if worker A sends to worker B, that's one message, and if B sends to A, that's another, so indeed, for all-to-all, each pair has two messages, but since we're counting messages sent, each worker sends $n-1$ messages, so total messages sent is $n*(n-1)$.

The table says for All-to-All, Number of Messages Sent is BLANK_1, and Size of each message is p .

In all-to-all, when each worker communicates with every other worker, what are they sending? They are sending gradients or averaged gradients? From the description: "each worker communicates with every other worker and updates its local weights by averaging

the gradients from all workers." So, each worker needs to send its gradients to every other worker. Since each worker has computed gradients based on its subset of data, and they need to average all gradients, so each worker must send its entire gradient vector to every other worker. **The gradient vector has p parameters, so each message has size p .**

And since each worker sends to $n-1$ others, and there are n workers, total messages sent are $n \cdot (n-1)$. So BLANK_1 should be $n(n-1)$.

But let's confirm the table. For Parameter Server, it says Number of Messages Sent is $2n$, which makes sense because each worker sends gradients to the server and receives updated weights from the server, so for n workers, that's n sends and n receives, so $2n$ messages. And size is BLANK_2, which we'll get to.

Now for Ring All-Reduce, it says Number of Messages Sent is $n(2(n-1))$, which seems correct from the description. In ring all-reduce, there are $2(n-1)$ steps, and each step, each worker sends a message, so total messages $n \cdot 2(n-1) = 2n(n-1)$. But the table says $n(2(n-1))$ which is the same as $2n(n-1)$.

Now, for the size of each message.

First, for All-to-All, size is p , as given.

For Parameter Server, what is the size of each message? From the description: workers compute gradients and send them to the parameter server. The gradients have size p , since there are p parameters. Then the server sends back the updated weights, which also have size p . So each message, whether sent from worker to server or server to worker, has size p . So BLANK_2 should be p .

But let's see the table: for Parameter Server, Size of each message is BLANK_2. Since all messages are of size p , BLANK_2 should be p .

Now for Ring All-Reduce, Size of each message is BLANK_3.

From the description: "each worker splits the gradients into n equally sized chunks". **Since there are n workers, and p parameters, and p divisible by n , each chunk has size p/n . Then, in the ring, each worker sends a chunk of gradients to its neighbor. So each message has size p/n .**

But in the process, after $n-1$ passes, all gradient chunks are aggregated, and then in the next $n-1$ steps, the aggregated chunks are passed along. So throughout, each message contains a chunk of size p/n .

So BLANK_3 should be p/n .

Now, let's verify the number of messages for Ring All-Reduce. The table says $n(2(n-1))$, which is $2n(n-1)$. From the description, there are $2(n-1)$ steps, and each step, each worker sends one message, so yes, total messages $n \cdot 2(n-1) = 2n(n-1)$.

So, to summarize:

BLANK_1: $n(n-1)$

BLANK_2: p

BLANK_3: p/n

But let's write it properly.

In the table, for All-to-All, Number of Messages Sent is BLANK_1, which should be $n(n-1)$.

For Parameter Server, Size of each message is BLANK_2, which should be p .

For Ring All-Reduce, Size of each message is BLANK_3, which should be p/n .

I think that's correct.

Valid observation here but technically not necessary to fill in blank A. The reasoning on the previous page about n into $(n-1)$ is all that's needed. However, again, since we have "deep think" mode on, Deepseek operates on a better safe than sorry philosophy regarding answer accuracy and quality.

Nice and cohesive reasoning paragraphs, could almost be plucked out into an answer key in my opinion.

As we can tell from the short and cohesive thinking trace (i.e., absence of "oh wait", "but what if", "this may not be right", "let me re-read", etc.), this question is very easy for Deepseek. And will probably be for most frontier models today since the deep think trace didn't reveal anything special about the thinking process. Deepseek basically just read the prompt and interpreted it a bit to arrive at the answer.

Based on the descriptions of the distributed training paradigms and the given assumptions (n workers, p parameters with p divisible by n), the blanks in the table can be filled as follows:

- For **All-to-All Communication**, each worker sends its gradients to every other worker, resulting in $n(n-1)$ messages sent, each of size p . Thus, BLANK_1 is $n(n-1)$.
- For **Parameter Server**, each worker sends gradients to the server and receives updated weights from the server, resulting in $2n$ messages sent. Each message (gradients or weights) has size p , so BLANK_2 is p .
- For **Ring All-Reduce**, each worker splits gradients into n chunks of size p/n , and each message in the ring has this size. The total number of messages sent is $n \cdot 2(n-1)$, and BLANK_3 is p/n .

The completed table is:

	Number of Messages Sent	Size of each message
All-to-All	$n(n - 1)$	p
Parameter Server	$2n$	p
Ring All-Reduce	$n - 2(n - 1)$	p/n