Mesh Tensorflow

Shubham Gupta

January 3, 2020

The original paper can be found here: `https://arxiv.org/abs/1811.02084`

1 Introductionn

- This paper talks about the new tool released by Google to train exceptionally large models(think models with billions of parameters)
- It has been created for distributed deep learning.
- Helps split your data over processors(generally TPU's) for faster computation.
- It's a layer over TF.

1.1 Do you need it?

From the github repo available `https://github.com/tensorflow/mesh`, you need it if:

- The parameters of the model do not fit on one device - e.g. a 5-billion-parameter language model.
- An example is so large that the activations do not fit on one device. - e.g. large 3D image model.
- Lower-latency parallel inference

2 Key Ideas

- Batch splitting:
  - Good because small data batches $\implies$ faster training
  - Bad because cannot train large models, high latency and inefficient at small batch sizes
- **SPMD**: Single-Program-Multiple-Data is a technique to split data and obtain results faster.
- Mesh-TF graph compiles program to SPMD. This helps implement data-parallel, model-parallel implementations of big networks such as Transformer.

3 Hardware Assumptions

- **Cluster**: Collection of identical processors with reliable memory
- **Mesh**: n-d array of processors. Could be local processors or TPU's.

4 SPMD

- The main steps in the SPMD algorithm are as follows:
  - Split data in batches.
  - Send each batch to different processor
  - Each processor has it's own copy of the parameters.
  - Processor computes forward and backward passes on the data. Returns the gradients.
  - Sum the gradients and broadcast results to all processors i.e **AllReduce**
  - Each processor updates it's copy of params
- Generally, SPDM splits data across batch dimension.
- Mesh-TF generalizes and let's split by arbitary dimensions.

5 Mesh TF

5.1 Similarities

- Each tensor is rep by one slice of the it on the processor
- Each op is implemented as one op in the processor.

5.2 Differences

- Named dimensions. Allow dimensions to be split the same way for different tensors and ops.
- Mesh also has n-dims
- **Computation layout**: Mapping from tensor-dim to mesh-dim. Batch splitting will be
- Current implementation needs tensor-dim to be *evenly divisible* with mesh-dim.

6 Functions

- Component wise ops
- Reduction
- Einstein summation(einsum in numpy)
- Reshape(requires network connection)

7 Syntax

- More or less similar to TF.
- Each dimension has a name and shape.

8 Experiments

- Trained multiple models on mesh tf.
- **Transformer**: Trained with layout

```
mesh_shape = [("all", n)]
computation_layout = [("vocab", "all"), ("d_ff", "all"), ("heads", "all")]
```

- Trained for 10 epochs on 512 core TPUv2 cluster. Best performance on billion-word language modelling benchmark(Perplexity = 23.5)
- On WMT14 En-Fr, 3 epochs. Trained on 128 core TPUv2 cluster. Best ever BLEU score 43.9

9 Conclusion

- Overall, this paper has explored the SPMD approach to breaking up datasets in smaller chunks and improving computation.
- The concept of named tensors and splitting these tensors in a similar fashion across all processors is a new and interesting concept for me.
- While the paper has mentioned that this is only for models with billions of parameters, I feel frameworks like these can be used only by the big corps for now. NO ONE can afford a 512 core TPU cluster lol.