High Performance Machine Learning
Lab 5

Paweł Rościszewski
Department of Computer Architecture
Faculty of Electronics, Telecommunications and Informatics
Gdańsk University of Technology

Introduction

In this lab we will execute the training in a distributed setup and examine the training performance. Due to limited computing resources, this lab should be done in groups. In order for a group to pass the task, the multi-node training should involve as many computing nodes as group members, not less than two. If someone was absent during the previous lab or does not have access to a working code with tf.data.Dataset provider, they should collaborate with others.
Hint: For the sake of performance it is recommended to use the version with GPU device placement.

1 The task: Data parallel training (10 points)

For distributing the trainings across multiple GPUs, multiple machines or TPUs, TensorFlow provides the *tf.distribute.Strategy* API[1]. Various strategies are available, depending on the target hardware environment, parallelization scheme and topology. Let's use the MultiWorkerMirroredStrategy[2] which allows ditributing the trainings over a set of nodes. The strategy object has to be set up in the estimator's RunConfig, through the *train_distribute* parameter.

In the case of using the MultiWorkerMirroredStrategy, multiple distributed worker processes can be executed as parts of the training program. In order to make the workers know about each other and provide connectivity, the cluster specification has to be configured by setting the *TF_CONFIG* environment variable in the following format:

```
TF_CONFIG='{"cluster":
{"worker":[HOSTNAME1:␣PORT1,␣HOSTNAME2,␣PORT2....␣HOSTNAMEN,␣PORTN]},
"task":{"type":␣"worker",␣"index":␣MYINDEX}}'
```

[1] https://www.tensorflow.org/guide/distributed_training

[2] https://www.tensorflow.org/api_docs/python/tf/distribute/experimental/MultiWorkerMirroredStrategy

The task for today is to implement data parallel training. For this, we will need more than one worker process that will perform the computations. The process specifications need to be included in the TF_CONFIG environment variable.
Hint: The MultiWorkerMirroredStrategy will try to place all operations on the GPU. However, some operations have no implementations supporting GPU computation. Setting the allow_soft_placement parameter of session config to True will allow TensorFlow to place operations on the CPU in such situations.

To pass the task: show that the training is working on GPUs that belong to multiple nodes. Compare training performance depending on the number of used nodes.