Distributed HyperParameter Tuning

High Performance Computing Semester Project

Team AVAAB

Team Details

S.No	Name	Enrollment No
1	Anshuman Aggarwal	E18CSE022
2	Aditya Varshney	E18CSE010
3	Arnav Bansal	E18CSE028

Problem Statement

In order to balance overfit and underfit for Machine Learning and Deep Learning tasks, an optimal set of Hyper-Parameters is chosen via Hyper-Parameter Tuning. This optimization of Hyper-Parameters gets complicated as the number of parameters involved in training increases, sometimes parameter tuning itself takes more time than training.

Objectives

- To inculcate high-performance computing to perform Hyper-parameter optimization
- To accelerate the process of Hyper-parameter optimization by using High-performance containerized systems like docker

Method

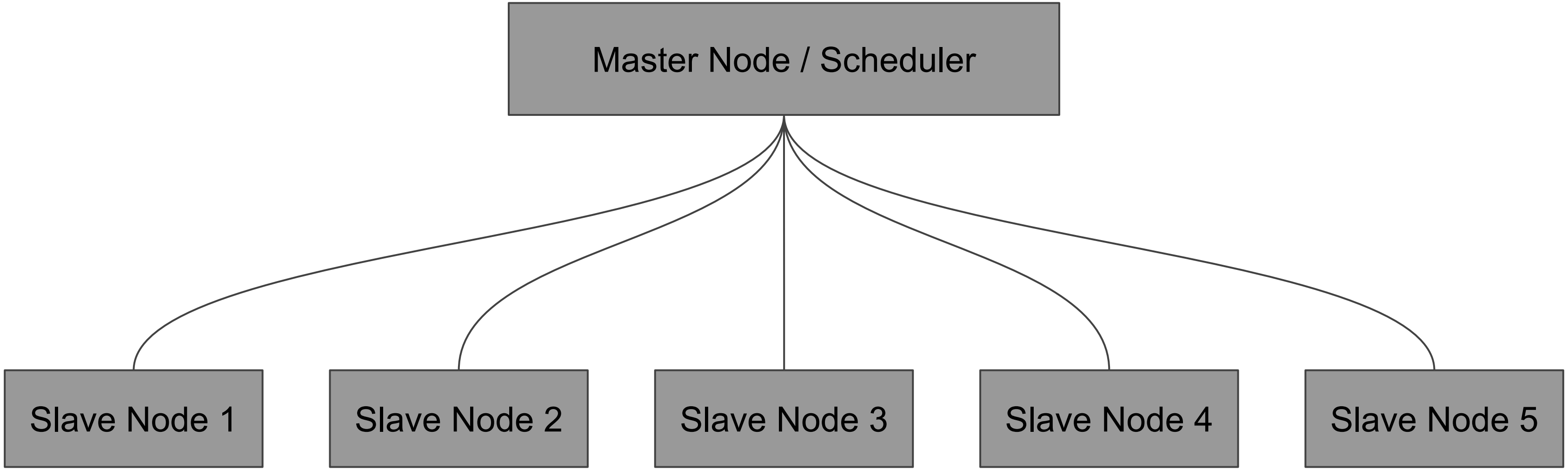
We propose to create a Master-Slave Paradigm where we will have 2 types of nodes, namely

- **Master Node:** Responsible for creating all possible combinations of the hyperparameters we want to work with and schedule instructions for all of its Slave Nodes
- **Slave Node:** Responsible for running the Deep Learning Model using the configuration sent to it by the Master Node and sending the performance metrics back to the Master

Method

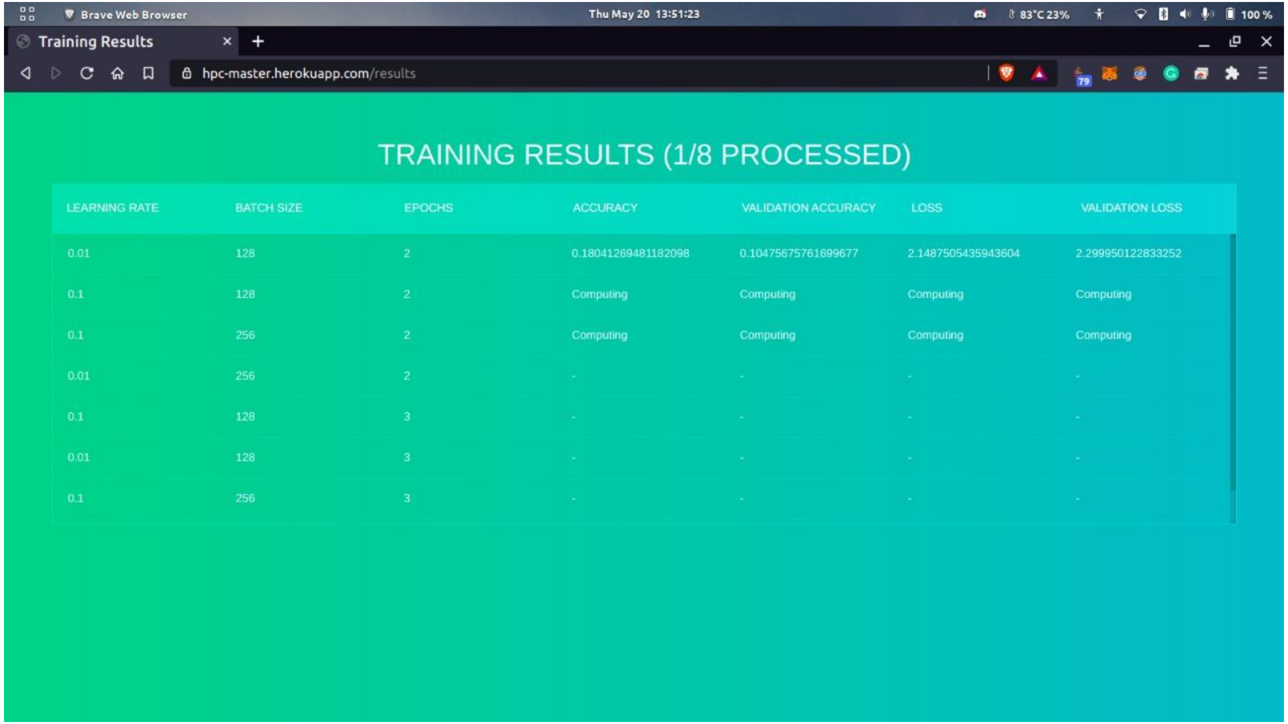
- Compute all possible combinations from the provided set of hyperparameters
- Create JSON files for each of the combination which would act as messengers between the Master and Slave Nodes
- Spawn multiple Slave/worker nodes on different Compute Instances and connect them to a Master Node over the TCP/IP protocol.
- Distribute and Schedule the workload across multiple Slave/Worker nodes using the Master Node
- Slave/worker node runs the training instance using the configuration sent over using the JSON file and returns the training accuracy as a result.
- Once All Configurations are completed, we can easily obtain the best set of hyperparameters for our model.

Method


Master Node / Scheduler
Slave Node 1
Slave Node 2
Slave Node 3
Slave Node 4
Slave Node 5

Technologies Used

Results



TRAINING RESULTS (1/8 PROCESSED)

LEARNING RATE	BATCH SIZE	EPOCHS	ACCURACY	VALIDATION ACCURACY	LOSS	VALIDATION LOSS
0.01	128	2	0.18041269481182098	0.10475675761699677	2.1487505435943604	2.299950122833252
0.1	128	2	Computing	Computing	Computing	Computing
0.1	256	2	Computing	Computing	Computing	Computing
0.01	256	2	-	-	-	-
0.1	128	3	-	-	-	-
0.01	128	3	-	-	-	-
0.1	256	3	-	-	-	-

Conclusion

We were able to use High-performance Docker containers to successfully implement the task of Hyper-parameter tuning using the Master Slave paradigm.

Instead of a serial Implementation where each combination was run sequentially, we were able to spawn ‘n’ slave nodes which could compute hyper-parameter combinations in a parallel manner, as a result reducing the time of execution of the total program.

References

- MobileNetV2 - https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV2
- CIFAR10 dataset - https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz
- Docker - https://docs.docker.com/get-started/
- Master Slave using Docker- https://devopscube.com/docker-containers-as-build-slaves-jenkins/
- Master Slave Paradigm using Docker - https://devopscube.com/jenkins-master-build-slaves-docker-container/

Thank You