Comp. Sys. Perf. Analysis

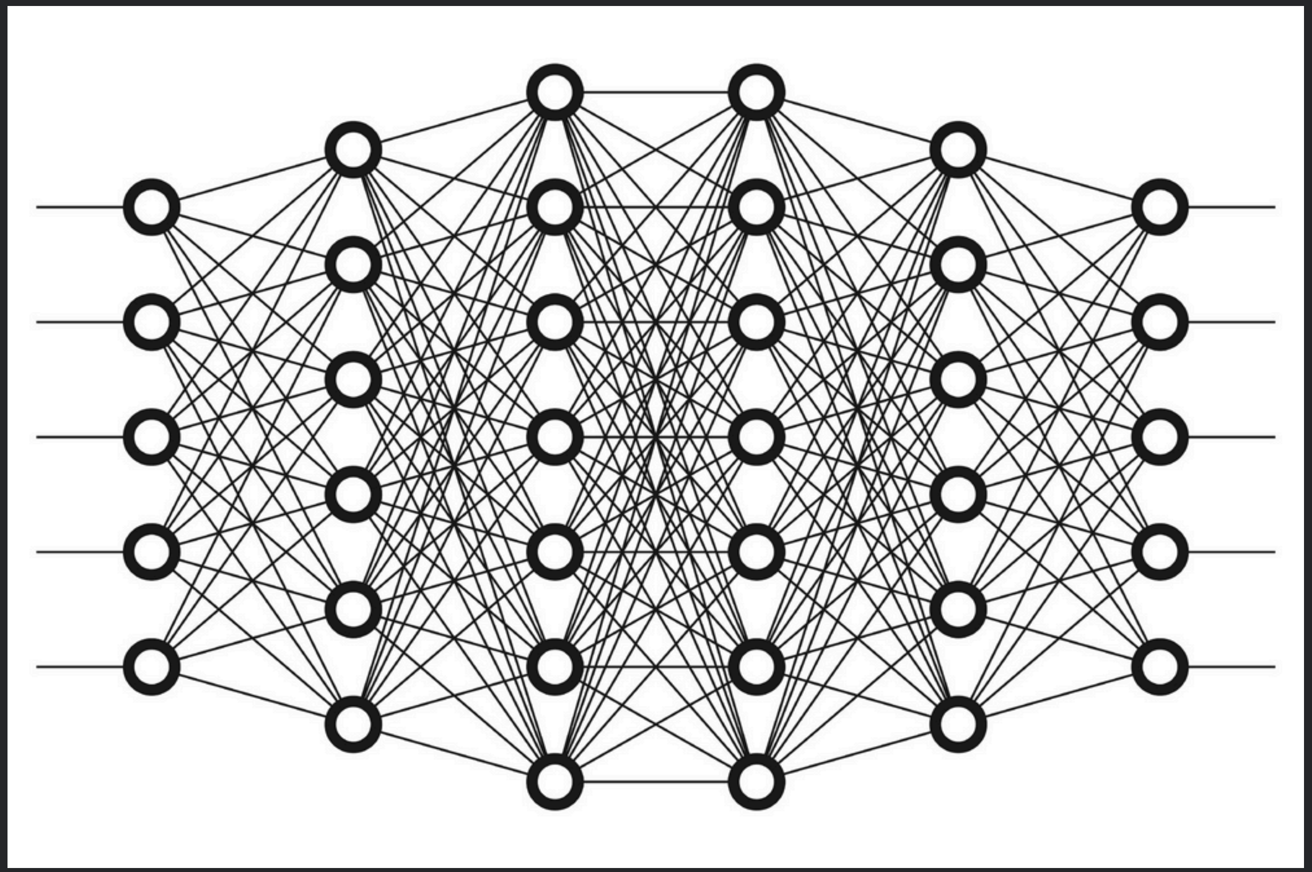
Rayan Raddatz, Kenichi Brumati and Marcelo Gulart

2025/2

Outline

- Computational object.
- Analysis methods and tools.
- Metrics.
- Preliminary Schedule.

Neural Networks and Deep Learning



Deep learning (DL) is neural networks (NN) with multiple hidden layers. Training a DL model can be a computationally demanding task that needs a lot of time and energy to accomplish. With the popularity of AI in recent times, larger DL models have emerged, bringing with it the need for more efficient ways to train them. Nowadays several parallelization techniques are employed to solve the problem of the time required to train these models. This means that any new technique that makes it possible to spend less time or energy when training NN is extremely valuable.

Main goal

Analyze the performance of training neural network model using parallel methods from two common machine learning platforms: **PyTorch** and **TensorFlow**

Computational Object

Chosen Model: ResNet50

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [40] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test *set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.*

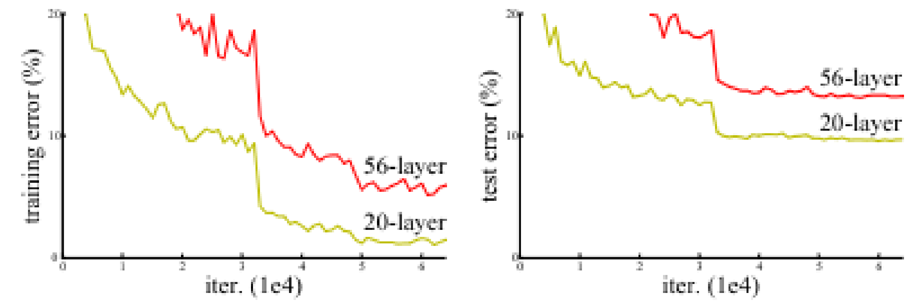
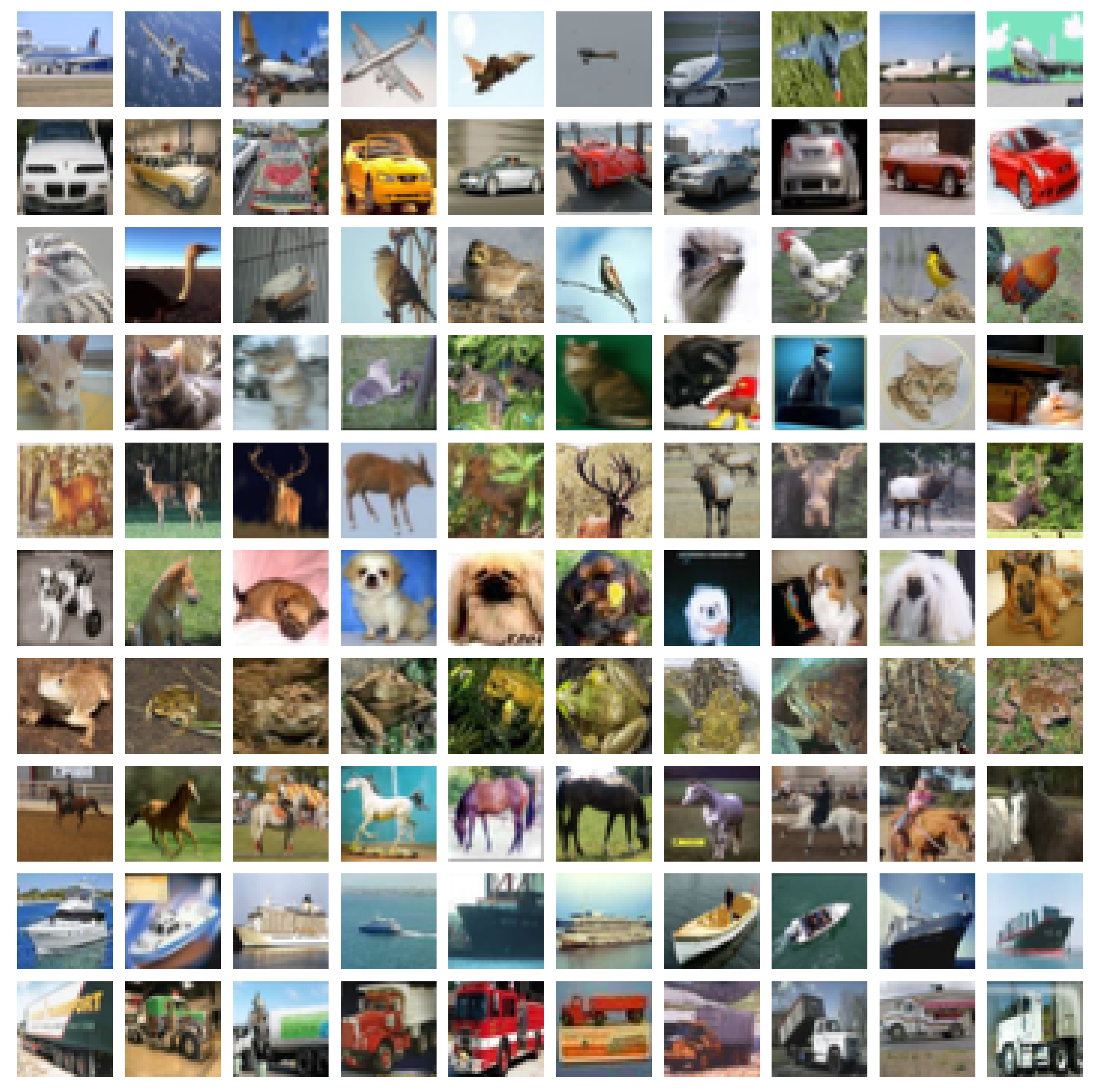


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

Driven by the significance of depth, a question arises: *Is learning better networks as easy as stacking more layers?* An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [14, 1, 8], which



Training Datasets

- **CIFAR-100** (60000 32x32 colour images and 100 classes)
- **IMAGENET** (1.281.167 training images and 1000 classes)

Tools for training:

As said, we will use the PyTorch e TensorFlow (Python) opensource platforms for AI (focused in deep learning).

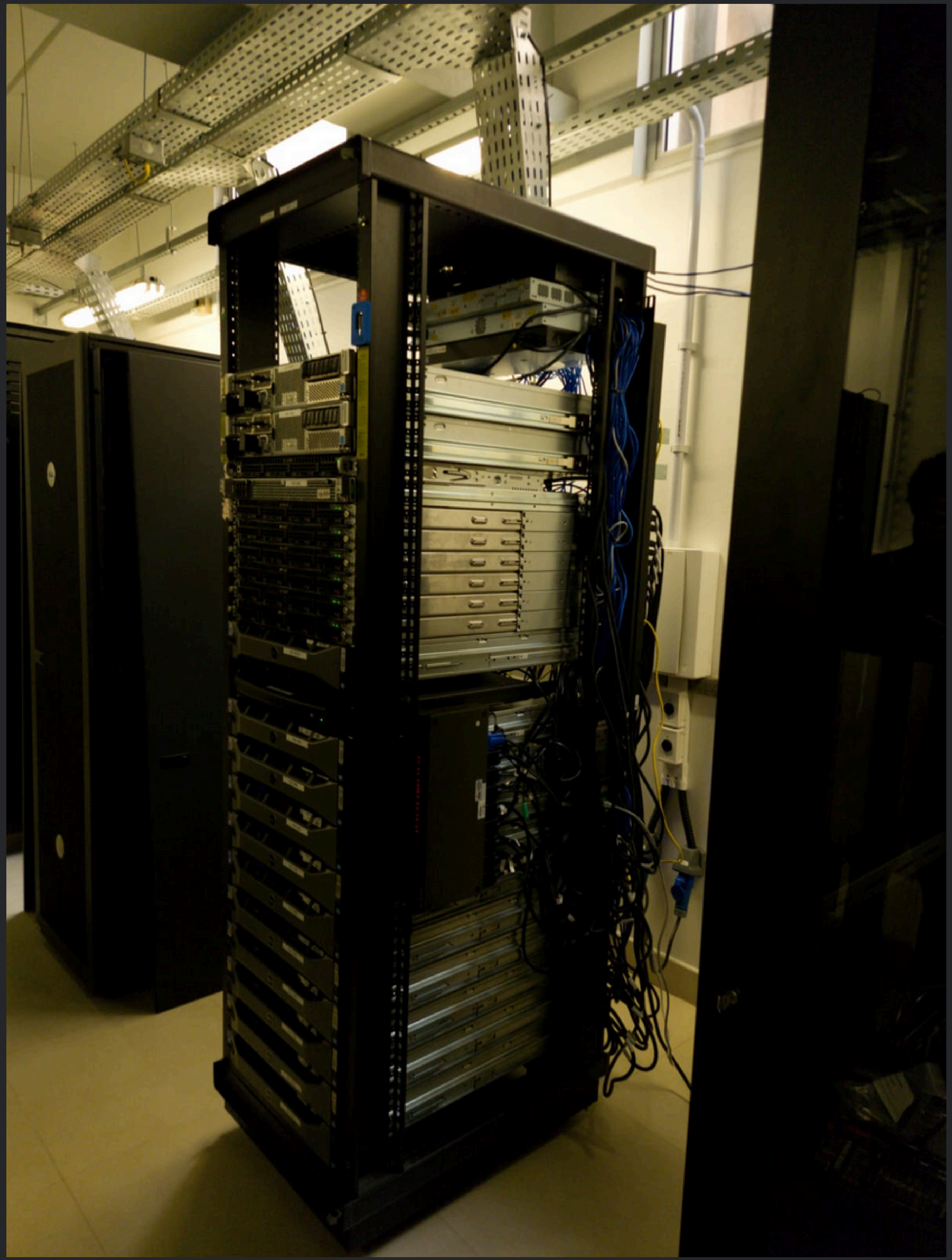
We will use the data parallelism strategies present in this platforms such as DistributedDataParallel for PyTorch and distribute.MirroredStrategy for TensorFlow for training our chosen model.

We also plain to evaluate CPU+GPU strategies that uses CPU to load the data used for training the model in the GPU.

Hardware Configuration:

We intend to use the PCAD infrastructure to execute our experiments. We will execute the training with one up to four computational nodes from the Tupi partition (tupi[3,4,5,6]). Each node has the following hardware configuration:

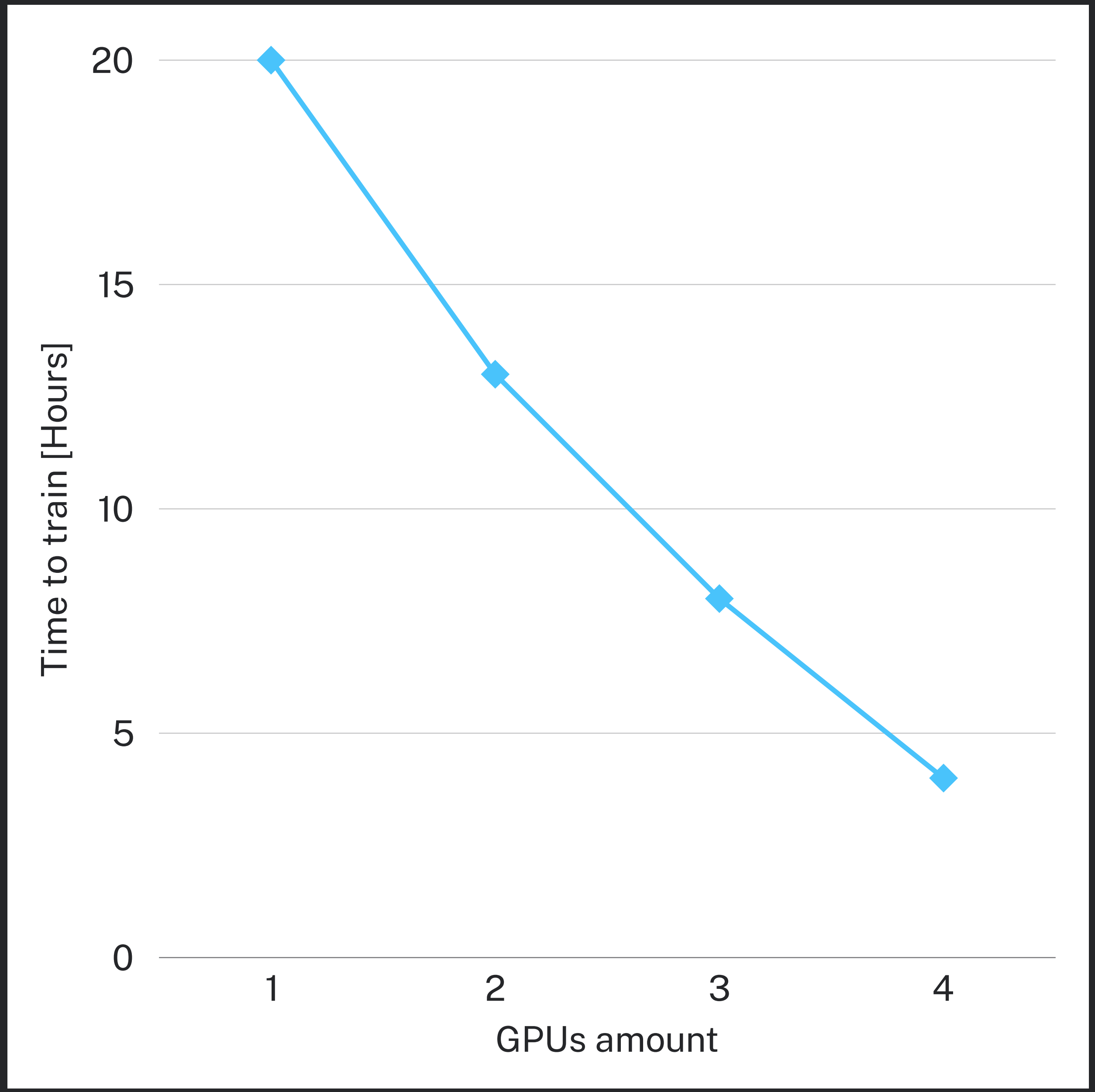
CPU: Intel(R) Core(TM) i9-14900KF, 3.20 GHz, 32 threads, 24 cores
RAM: 128 GB DDR5 RAM
GPU: RTX 4090 (24GB)
Motherboard: Gigabyte Technology Co., Ltd. Z790 UD AX



PCAD Picture

Metrics*:

- Time to train the model
- Model accuracy
- GPU and CPU usage



*For illustrative purposes only

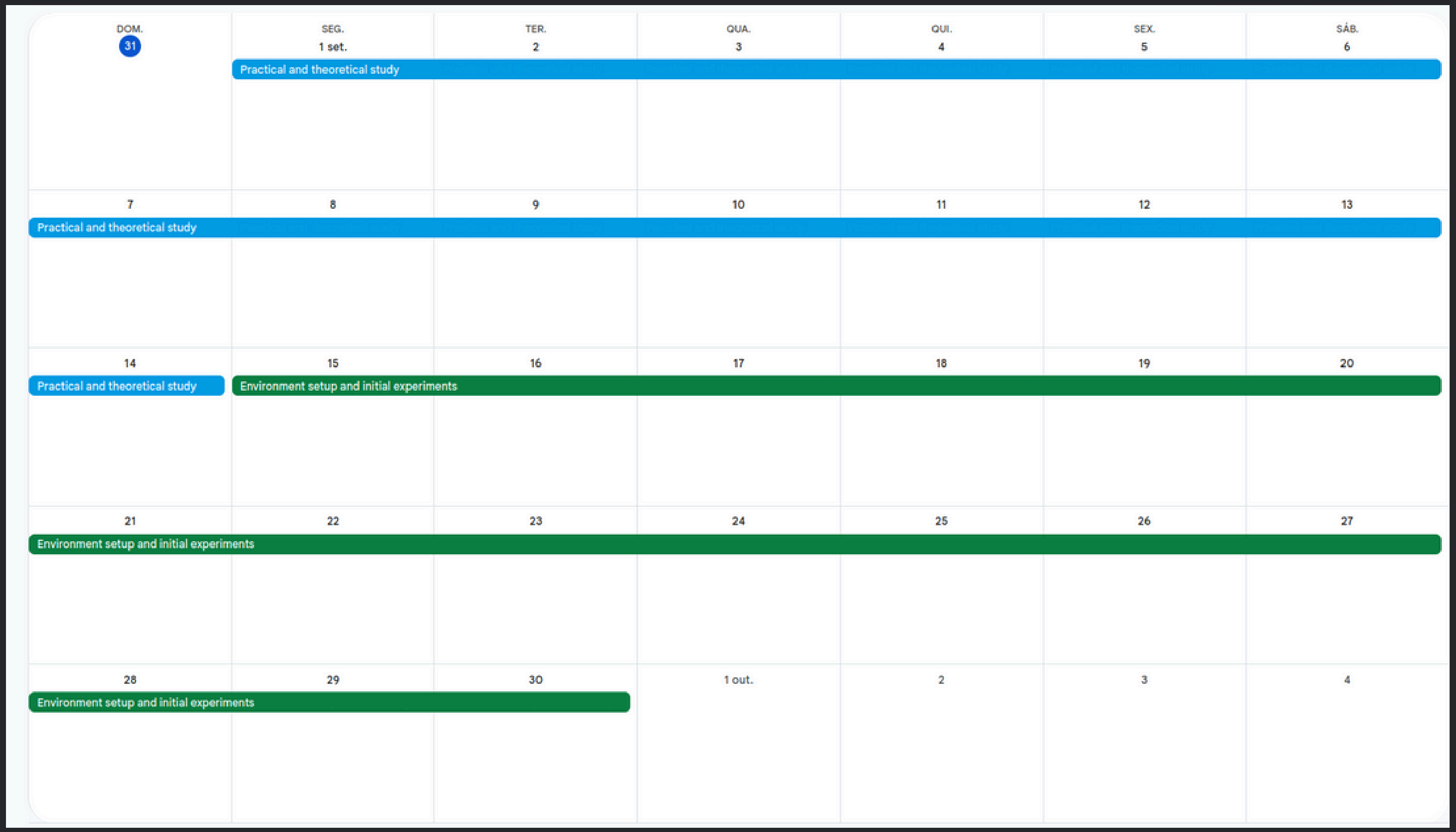
*If possible, we want to evaluate energy used or potency in the training process.

Preliminary Schedule.

Cronogram*

Our goal is to split the project into two-week "sprints". We will organize our workflow using .org files and maintain a public github repository with our ongoing work.

- **01/09 - 15/09:** Practical and theoretical study.
- **16/09 - 30/09:** Environment setup and initial experiments.
- **1/10 - 14/10:** Verification of initial results and report writing.



*This is a preliminary schedule and can be changed anytime. If we go ahead of schedule we can evaluate other methods.

References:

Russell, Stuart J. (Stuart Jonathan), 1962-. Artificial Intelligence : a Modern Approach. Upper Saddle River, N.J. :Prentice Hall, 2010.

He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition.

PyTorch official documentation: https://pytorch.org/docs/stable/index.html

TensorFlow Official Guide: https://www.tensorflow.org/guide/

Huber, Philipp et al. “Energy Consumption in Parallel Neural Network Training.” (2025).

Official LPPD website: https://pcad.inf.ufrgs.br