Reading: Overview of Data Loading

Estimated time needed: 5 minutes

Learning objectives:

- Explain how the bulk loading and sequential loading methods influence memory usage, training speed, and a project's overall scalability
- Identify the advantages and disadvantages of the bulk loading and sequential loading methods

Introduction

Data loading is a fundamental step in any data-related workflow. To extract any value from data, be it for analysis, visualization, or building sophisticated deep learning models, it must be made available in the system for inspection. Did you know a poor data pipeline can cost up to 80% of a data scientist's time?

In deep learning, a model's ability to learn, generalize, and perform relies heavily on the quality and quantity of the data it's trained on.

But long before a single gradient is calculated or a neuron is updated, data scientists face a critical yet often overlooked challenge of efficiently loading and managing massive datasets. The two main methods you can use to handle this are bulk loading and sequential loading.

Let's begin by understanding how these methods influence memory usage, training speed, and a project's overall scalability.

Bulk loading method

Bulk loading is a simple, intuitive, and commonly used method, especially among developers just beginning to work on data-intensive tasks. In this method, the entire dataset is read from its storage location, including local disk, network drive, and cloud. The data is then loaded into the system's RAM at once. The program typically loops through each file path in the dataset, opens the image files one by one, converts them into arrays, and stores them in a master list. Once the process is complete, the full dataset is held in memory and ready for immediate access.

Advantages of the bulk loading method

The main advantage of bulk loading is the speed of repeated access it offers after the initial load is finished.

Since all the data is stored in high-speed RAM, subsequent operations like fetching a random batch of images or shuffling the entire dataset become extremely efficient. This eliminates the need for repeated disk I/O, which is comparatively slow and can hinder performance.

Disadvantages of the bulk loading method

The major drawback of bulk loading is its extreme memory usage. The amount of RAM required scales linearly with the size of the dataset. Consider a dataset of 10,000 images, at 256x256 resolution, requiring roughly 2 GB of memory. While this may be manageable for small datasets, real-world applications such as medical imaging or autonomous driving often involve millions of high-resolution images, requiring terabytes of RAM.

This lack of scalability makes bulk loading difficult for production environments and limits its practicality to small-scale or research use cases.

Sequential loading method

In direct contrast to the bulk loading method, sequential data retrieval from the storage is performed on per batch basis, and on demand instead of preloading the entire dataset. The first step in this method involves compiling a list of the file paths to all the data samples. This list, containing only strings, requires a small amount of memory, even for millions of files.

In sequential, or lazy, loading, the data is not read from the disk until it is needed within the training loop. For example, to prepare a batch for training, a subset of file paths is selected, and each file is loaded and transformed, which could be resized or normalized and then collated into a batch. At any given time, only the current batch resides in RAM. After the batch has been used in a training step, its memory can be released, allowing the system to load the next batch and enabling efficient memory usage throughout training.

Advantages of the sequential loading method

The key benefit of sequential loading is its exceptional memory efficiency, which enables high scalability. Whether the dataset has hundreds or hundreds of millions of images, this approach allows training models on datasets far exceeding the system's RAM capacity.

Additionally, since there's no large, initial loading phase, the application can start processing almost immediately.

Disadvantages of the sequential loading method

The primary drawback of naive sequential loading is the repeated disk I/O, which is significantly slower than accessing data from RAM. During training, a graphics processing unit, or GPU, can process data much faster than a single central processing unit, or CPU, thread can read it from the disk. This leads to an I/O bottleneck where the GPU remains idle, waiting for the next batch.

This issue is usually resolved through parallel data loading and is implemented efficiently in most deep learning frameworks such as Keras and PyTorch.

Summary

In this reading, you've learned about the features, advantages, and disadvantages of two data loading methods: bulk loading and sequential loading.

- The bulk loading method offers an easy and intuitive way for data handling.
- The entire dataset is read from the storage location and then loaded into the system's RAM in one go in case of the bulk loading method.
- The main advantage of bulk loading is the speed of repeated access it offers after the initial load is finished.
- The severe memory limitations faced by the bulk loading method make it unusable for serious deep learning implementations.
- The sequential loading method retrieves data from storage one small batch at a time, and on demand, instead of preloading the entire dataset.
- The key benefit of sequential loading is its exceptional memory efficiency, which enables high scalability.
- The primary drawback of the sequential loading method is the repeated disk I/O, which is significantly slower than accessing data from RAM.