File Formats, Compression

Common Data Formats

- vision: JPEG, MPEG, PNG, TIFF, tensors
- speech: MP3, FLAC, tensors
- NLP: unicode strings
- other: CSV, HDF5, text files, protobuf, ...

Common Storage Formats

- individual files on file systems
- file-systems-in-files
- databases (HDF5, LMDB, sqlite, SQL, NoSQL, etc.)
- row stores, record-sequential formats (TFRecord, POSIX tar, etc.)
- column stores (Parquet, etc.)
- globally compressed or individually compressed

Common Storage Locations

- rotational drives
- solid state drives
- network file systems
- distributed file systems
- web servers

Streaming vs Block Compression

assume we have a 1 Gbyte stream of data

1. compress all at once, require all-at-once decompression (e.g. reordering)
2. compress chunk-wise, decompression of each chunk independently
3. streaming compression, require fully sequential decompression

These are in order of increasing compression efficiency.

Storage / Bandwidth / Computation Tradeoffs

More Compression means:

- more costly data preparation
- more costly data decompression
- less costly storage
- less costly bandwidth usage

There is no optimal level of compression; it's application dependent.

Example: Image Compression

- TIFF, PNG, JPEG, JPEG2000, webm
- mature compression, both lossy and lossless
- intended for general photographic images, documents
- DCT, Wavelets
- Huffman/arithmetic
- designed for sequential processors with limited memory
- often a bottleneck in deep learning pipelines (!)
- support on GPUs for some formats

Example: Tensor Storage

- direct storage of multidimensional numerical data
- limited compression possible (usually 20-40%)
- 10-100x bigger than compressed images/videos
- HDF5 (random access), torch/Python save, TFRecord/tf.Example
- needed for GPUDirect from disk or network

PyTorch FileDataSet

Common format used in PyTorch:

- reads a list of image files and classes from a CSV/Matlab file
- stores them in an array, e.g. `dataset[index].fname`, `dataset[index].cls`
- upon access, returns `imread(dataset[index].fname), dataset[index].cls`

Note:

- provides random access to data
- results in a lot of seeks on disk
- diassociates image data from classifications

TFRecord + tf.Example

Common format used in Tensorflow:

- inspired by Google's large scale web indexing pipeline
- TFRecord is a binary sequential record format
- tf.Example is a protobuf encoding of data structures
- special conventions for representing tensors etc.

Note:

- associates all related data in each record (row store)
- works great with Google's internal map-reduce pipelines
- few generally available tools
- AV data not well represented, ad hoc compression choices

WebDataset = POSIX tar + files

Common format used for archival storage, WebDataset:

- POSIX tar is a sequential storage container for files
- content is stored in its original file based format

Notes:

- allows bit-identical storage of images, videos, audio, etc.
- widely available tools for reading archives and files in archives
- can cover serialized data structure use case by embedding protobuf, msgpack
- nearly drop-in replacement for PyTorch `DataSet`

WebDataset

(notebook)