Data processing pipeline

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — 861fb9d0

0.1 Context

0.2 Learned in this study

0.3 Things to explore

1 Overview

Processing data is one of the core activities of a program. There are many ways to write how to process a given set of data, however the concept of pipes and streams has been a popular one for many years.

In this article, we look into a potential implementation that would allow us to deal with data processing in a generic fashion.

2 Requirements

- Plug-and-play addition of new processing units
- The ability to replay already processed data on newer processing units only
- Processing units have an identifier, a list of processing units it depends on (dependencies) and a processing function
- Processing units, like pipes, can be connected to one another

3 Conception

3.1 Terminology

- **Graph** Directed graph of operations to be executed
- **Feed/Placeholder** Indicate where data can be fed into the graph
- **Operation** Operation executed on data provided as input
- **Fetch/Output** The result of an operation
- **Session** A context within which a set of computation is executed

3.1.1 Alternative nomenclature

- **Stream** set of data
- **Kernel functions** operations

4 See also

5 References

- http://c2.com/cgi/wiki?DataflowProgramming

  - http://c2.com/cgi/wiki?FlowBasedProgramming
  - http://c2.com/cgi/wiki?PipesAndFilters
- https://en.wikipedia.org/wiki/Dataflow_programming
- https://en.wikipedia.org/wiki/Flow-based_programming
- https://en.wikipedia.org/wiki/Stream_processing
- https://en.wikipedia.org/wiki/Pipeline_(software)
- https://www.tensorflow.org/versions/0.6.0/get_started/basic_usage.html#the-computation-graph