The Journal of Open Source Software

mmappickle: Python 3 module to store memory-mapped numpy array in pickle format

Laurent Fasnacht[1]

1 University of Neuchâtel

DOI: 10.21105/joss.00651

Software

- Review
- Repository
- Archive

Submitted: 12 March 2018
Published: 18 June 2018

Summary

Mmappickle is a Python 3 library which enables storing large numpy arrays into a file, along with the associated metadata, and to retrieve it in such a way that the numpy array are memory-mapped (numpy.memmap) instead of copied into the system memory.

This library allows working on matrices bigger than the size of the RAM, and allows concurrent read/write access from multiple processes, while having data stored in a structure behaving like a normal Python dictionary. This structure is mutable, as keys can be added and removed without having to rewrite the entire file. Moreover, the file complies with the Python Pickle format, and can be loaded on computers which don't have mmappickle installed (provided enough RAM is available to load all the matrices).

Mmappickle is designed to be used for the development of parallel algorithms, using for example the Python multiprocessing module ("multiprocessing — Process-based parallelism — Python 3.6.4 documentation," n.d.). It also allows unstructured parallel access, such as multiple different programs accessing the same file.

Alternative solutions exist, for example:

- The various bindings for HDF5, like pytables (Alted, Vilata, and others 2002) or h5py (Collette 2013). However, they have severe limitations for concurrent access, and the API adds complexity to the source code. It also adds additional binary dependencies (and therefore requires either a compiler or binary packages).
- Joblib ("Joblib: running Python functions as pipeline jobs," n.d.) provides advanced features, such as providing memory-mapped access to arrays embedded in complex structures, compression, and features to enable caching with minimal code modifications. However, the files it produce are not mutable (it is not possible to add or remove entries from a structure), and are not compatible with the Python Pickle protocol.
- The direct use of numpy.memmap ("numpy.memmap - NumPy v1.14 Manual," n.d.). Unfortunately, numpy.memmap is only able to store the matrix data into a file, but not its shape (i.e. its dimensionality) or datatype. Moreover, numpy.memmap is difficult to use in practice because one needs to work on multiple matrices simultaneously (and even additional metadata) and the direct use of numpy.memmap is particularly challenging in this situation.
- The use of numpy.lib.format.open_memmap ("Numpy source code: numpy/lib/format.py," n.d.), which is not documented in the numpy main documentation. It is a wrapper to numpy.memmap, which uses the standard npy file format (hence storing the shape and the datatype). It still has the limitation that only one array can be stored per file, and it requires specifying an explicit file access mode. This is an inconvenience, because a wrong choice of access mode can result in overwriting the file, resulting in data loss.

Mmappickle exhibits similar performance as these alternative solutions, as the underlying array access technique is similar. Mmappickle is also the only approach capable of handling arrays with masked values (i.e. missing data).

This library is currently used for storing and processing hyperspectral imaging data. For such application, the ability to modify the dictionary to add data is crucial: as an example, it enables to store a timelapse incrementally to a file (adding new images as new dictionary entries), while simultaneously monitoring the file from a viewer program.

Limitation and further work

As this library relies on Python Pickle format, which is not secure ("pickle — Python object serialization - Python 3.6.4 documentation," n.d.), it should not be used with files from untrusted sources.

Further work is ongoing to allow it to load data directly when needed from a (trusted) HTTP server, in order to simplify data distribution.

References

Alted, Francesc, Ivan Vilata, and others. 2002. "PyTables: Hierarchical Datasets in Python." http://www.pytables.org/.

Collette, Andrew. 2013. *Python and Hdf5*. O'Reilly.

"Joblib: running Python functions as pipeline jobs." n.d. https://pythonhosted.org/joblib/.

"multiprocessing — Process-based parallelism — Python 3.6.4 documentation." n.d. https://docs.python.org/3/library/multiprocessing.html.

"Numpy source code: numpy/lib/format.py." n.d. https://github.com/numpy/numpy/blob/8d5bdd1/numpy/lib/format.py#L696.

"numpy.memmap - NumPy v1.14 Manual." n.d. https://docs.scipy.org/doc/numpy/reference/generated/numpy.memmap.html.

"pickle — Python object serialization - Python 3.6.4 documentation." n.d. https://docs.python.org/3/library/pickle.html.