Using reticulate to read and write NumPy files

Dirk Eddelbuettel[a]

[a]http://dirk.eddelbuettel.com

This version was compiled on March 22, 2018

This vignette shows how to use the reticulate package for to directly access the NumPy module for Python.

Motivation

The RcppCNPy package by Eddelbuettel and Wu (2016) provides a simple and reliable access to NumPy files. It does not require Python as it relies on the CNPy library which is connected to R with the help of Rcpp. Now, thanks to the reticulate package by Allaire *et al.* (2018), we can also consider an alternative which does not require CNPy–but which requires Python. Thanks to reticulate, we can (on a correctly set up machine, how to do that is beyond the scope of this note but described in the reticulate documentation) use Python to read NumPy data. And reticulate will faithfully transfer the data for us.

Simple Examples

```
### load reticulate and use it to load numpy
library(reticulate)
np <- import("numpy")

## data reading
(mat <- np$load("fmat.npy"))
```

```
#       [,1] [,2] [,3] [,4]
#  [1,]  0.0  1.1  2.2  3.3
#  [2,]  4.4  5.5  6.6  7.7
#  [3,]  8.8  9.9 11.0 12.1
```

```
(vec <- np$load("fvec.npy"))
```

```
#  [1] 0.0 1.1 2.2 3.3 4.4
```

Integer data can be read the same way:

```
(imat <- np$load("imat.npy"))
```

```
#       [,1] [,2] [,3] [,4]
#  [1,]    0    1    2    3
#  [2,]    4    5    6    7
#  [3,]    8    9   10   11
```

Compressed Files

The gzip Python module allows us to access compressed files.

```
## compressed data: import gzip
gz <- import("gzip")

## use it to create handle to uncompressed file
(mat2 <- np$load(gz$GzipFile("fmat.npy.gz","r")))
```

```
#       [,1] [,2] [,3] [,4]
#  [1,]  0.0  1.1  2.2  3.3
#  [2,]  4.4  5.5  6.6  7.7
#  [3,]  8.8  9.9 11.0 12.1
```

Saving Files

Similarly, files can be saved via reticulate access to NumPy.

```
tfile <- tempfile(fileext=".npy")

set.seed(42)
(m <- matrix(sort(rnorm(6)), 3, 2))
```

```
#            [,1]     [,2]
#  [1,] -0.564698 0.404268
#  [2,] -0.106125 0.632863
#  [3,]  0.363128 1.370958
```

```
np$save(tfile, m)

(m2 <- np$load(tfile))
```

```
#            [,1]     [,2]
#  [1,] -0.564698 0.404268
#  [2,] -0.106125 0.632863
#  [3,]  0.363128 1.370958
```

```
all.equal(m, m2)
```

```
#  [1] TRUE
```

Savez Array Files

We can also access `savez` files. First we save two vectors two different ways:

```
x <- seq(1, 10)
y <- sin(x)
np$savez("file1.npz", x, y)
np$savez("file2.npz", x=x, y=y)
```

We can access these files with and without names:

```
npz1 <- np$load("file1.npz")
npz1$files
```

```
#  [1] "arr_1" "arr_0"
```

```
npz1$f[["arr_0"]]
```

```
#   [1]  1  2  3  4  5  6  7  8  9 10
```

```
npz1$f[["arr_1"]]
```

```
#   [1]  0.841471  0.909297  0.141120 -0.756802
#   [5] -0.958924 -0.279415  0.656987  0.989358
#   [9]  0.412118 -0.544021
```

```
npz2 <- np$load("file2.npz")
npz2$files
```

```
#  [1] "y" "x"
```

```
npz2$f[["x"]]
```

```
#   [1]  1  2  3  4  5  6  7  8  9 10
```

```
npz2$f[["y"]]
```

```
#   [1]  0.841471  0.909297  0.141120 -0.756802
#   [5] -0.958924 -0.279415  0.656987  0.989358
#   [9]  0.412118 -0.544021
```

References

Allaire JJ, Ushey K, RStudio, Tang Y, Eddelbuettel D, Lewis B, Geelnard M (2018). *reticulate: Interface to Python*. R package version 1.5, URL http://CRAN.R-Project.org/package=reticulate.

Eddelbuettel D, Wu W (2016). "RcppCNPy: Read-Write Support for NumPy Files in R." *The Journal of Open Source Software*, **1**(5). . URL https://doi.org/10.21105/joss.00055.