Switching between CPU and GPU on-the-fly using CuPy

Stéphan Tulkens

SOME CONTEXT: GPU

- Graphics Cards (GPU) are able to process lots of different things in parallel
  - Required for rendering “video games”

SOME CONTEXT: GPU

- Graphics Cards (GPU) are able to process lots of different things in parallel
  - Required for rendering “video games”

- If running things on CPU is like stuffing all your operations in one big pipe, running on a GPU is like all kinds of stuff through an array of pipes.

SOME CONTEXT: GPU

- Graphics Cards (GPU) are able to process lots of different things in parallel
  - Required for rendering “video games”

- If running things on CPU is like stuffing all your operations in one big pipe, running on a GPU is like all kinds of stuff through an array of pipes.
- **Result: SPEED!**

SOME CONTEXT: GPU

- Graphics Cards (GPU) are able to process lots of different things in parallel
  - Required for rendering “video games”

- If running things on CPU is like stuffing all your operations in one big pipe, running on a GPU is like all kinds of stuff through an array of pipes.
- **Result: SPEED!**
- One of the drivers of recent “Deep Learning” technology

Drawback

- Writing code for the GPU is difficult
  - Usually done in CUDA
  - Not portable

- Leads to premature optimization, or rewrites
  - You implement on GPU, might be too much for the job
  - You implement on CPU and rewrite for GPU because CPU is too slow.

Cupy

- Cupy is a python module which dynamically decides to run operations on the CPU or GPU, depending on whether the data lives on the CPU or GPU.

```
def agnostic_sigmoid(x):
    xp = cupy.get_array_module(x)
    e = xp.exp(x)
    return e / e.sum()
```

Links

- **Cupy:**
- https://github.com/cupy/cupy

- **Cupy developers:** preferred networks
- https://www.preferred-networks.jp/en/