

Analysis of SGD variants

All the code is been written to be sort of readable, and it's commented, in order to make it easier to comprehend.

For the algorithms, the same structure is used over all of them:

1. a fist part initializes a bunch of arrays for tracking various things, used then for the plots
2. a second part, which is the "loops", that controls the flow of the algorithms (epochs, iterations, minibatches)
3. the algorithm itself
4. the tracking part, where we update the arrays that traks grad, points, and so on

All fo them are distinguishable because separated by a newline in all of the algorithms (and therefore they are pretty redundant)

How to load datasets

The code provided in the Jupyter notebook loads the datasets from the main memory.

In order to so, it's required that they are in specific folders with specific names, however the convention is equal to all of them.

1. Create a folder called `datasets` in the same folder as the Jupyter notebook
2. Create the following folders inside the new `datasets` folder:
 - `bio-or-not` for the "Bio or not" dataset
 - `fish-or-not` for the "Fish or not" dataset
 - `cat-vs-dog` for the "Cats vs Dogs" dataset
3. inside the folder created at point 2, create 2 folders, which names are the first and the last word of the folder name.

The final result should be the following:

```
path/to/project
├── datasets
│   ├── cat-vs-dog
│   │   ├── cat
│   │   └── dog
│   ├── bio-or-not
│   │   ├── bio
│   │   └── not
│   └── fish-or-not
│       ├── fish
│       └── not
└── notebook.ipynb
```

4. inside the "leaf" folders, insert the desired images

5. inside the first snippet, set `DOWNLOAD_AND_STORE = True` (it will tell the code to look in the filesystem for the images, process that will take 2/3 minutes at least, and at the end will store in the dataset folder a serialized and lighter version, which will be loaded all the following times, after having set back `DOWNLOAD_AND_STORE = False`, that will take just a second to be loaded)

WARNING: please pay attention because some images are corrupted, and the code might not run in that case (however, it will display in the output the name of the file that is causing problems)

Libraries

In order to run the full project, some "not too common" libraries are required, mainly to have nice plots and animations:

- `plotly`: a installation procedure can be found here <https://plotly.com/python/getting-started/>, however it's used only for 3D animations and contour plots, so if not required, it's enough to comment out the code that uses it
- `PIL`: a installation procedure can be found here <https://pillow.readthedocs.io/en/stable/installation.html>, it's used to process images from the filesystem to convert them to numpy arrays
- `json`: used only with `plotly`, so if you don't want 3D plots, you can remove it
- `Dash`: used only with `plotly` to create a server on which others can check the 3D animations, an installation procedure can be found here <https://dash.plotly.com/installation>, the code that uses it is already commented, so you might even want to ignore it

Others libraries required are:

- `tensorflow`, to process images
- `matplotlib`, for all the plots
- `numpy`
- `time`, to time the algorithms
- `sklearn`, to process images