Preliminary To-Dos:

- Keep a daily log of the research you work on – tasks completed, progress made, etc. When you learn to write in latex, you will submit this report compiled in latex. For starting out, .txt is fine.
- Acquire a method to turn .tex files to PDF; (e.g., Overleaf, pdfLaTeX, TeXShop, TeXStudio, LaTeX $\to$ dvips $\to$ ps2pdf, etc. If you use vim, I can recommend vim-latex.)
- Install Julia; familiarize yourself with Julia by running a basic experiment with the theoretical tools below (e.g., implement a few basic proximity operators from here `http://proximity-operator.net/indicatorfunctions.html` and compare with the premade MATLAB/Octave scripts available in their repo)
- **Theory and Vocabulary Homework** (if a question is not provided, please define the word **and** provide an intuitive description/example.):
  - What is the definition of a convex subset of $\mathbb{R}$? What about a convex subset of $\mathbb{R}^n$?
  - What does it mean for a function $f\colon \mathbb{R}^n \to \mathbb{R}$ to be convex?
  - *Proximity operator* (of a convex function): (hint: `http://proximity-operator.net/proximityoperator.html`)
  - *Projection operator* (of a convex set): (hint: `https://en.wikipedia.org/wiki/Hilbert_projection_theorem`)
  - An *indicator function* of a closed convex set $C \subset \mathbb{R}^n$ is given by

    $$\iota_C\colon \mathbb{R}^n \to \mathbb{R}\colon x \mapsto \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{otherwise.} \end{cases} \tag{1}$$

    Prove that the proximity operator of ι_C is the projection operator of C.
  - Define the *Gradient* of a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$.
  - Prove or disprove (via counterexample): The gradient of a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ always exists.
  - What is the difference between a gradient and a proximity operator?
  - What "direction" does the gradient point towards?
  - Provide a geometric description of where the vector $x - \nabla f(x)$ points.

  One of our main tasks is to program asynchronous algorithms which involve gradients, projections, and proximity operators. These algorithms theoretically are proven to work in certain situations; however, they have not been asynchronously implemented to-date. A big research goal is to determine hyperparameter selection strategies for each algorithm, and write up a comparison of the available algorithms. Optimization algorithms – which are the backbone of ML – arise as a special case of these algorithms.

  Many of the papers provided in the Literature folder involve *firmly nonexpansive operators* – this is a class which includes proximity operators, projections, and rescaled gradient operators (we must rescale by $1/L$ where L is known as the "Lipschitz constant" of the gradient. Don't worry too much about where L comes from for now).

Computational goals:

- Implement the main algorithms in Glau20 and Comb18 with the ability to activate the operators $(T_i)_{i \in I}$ asynchronously.
- Initial testing: implement algorithms for $(T_i)_{i \in I}$ being projections onto simple shapes in 2D (boxes, balls, half-planes, subspaces). Construct visualizations.

Literature:

- Glaudin/PLC - Asynchronous implementation has not been studied
- onto simple shapes in 2D (boxes, balls, half-planes, subspaces). ow-rank + sparse decomposition – smooth, although there are likely problem-specific algos to compare.

Contact people

Mathieu Besançon (research) Heike Balliunet (General admin questions)