Project details and tips

You have been given a brain image that has certain noise like artifacts. These artifacts have been generated by randomly throwing away some samples from its 2D Fourier data. The **mask** matrix is a logical matrix that is the sampling pattern used in sampling 2D Fourier data of the brain. Please take a look at the **data** matrix to confirm the sampling pattern.

Project task: Design and test an algorithm for restoring the brain image.

Step 1:

The general framework for finding a solution to this problem is the following:

$$\mathbf{v}_{\mathbf{reg}} = \underset{\mathbf{v}}{\mathbf{argmin}}(\|\mathbf{Ev} - \mathbf{d}\|_2^2 + \lambda\varphi(\mathbf{v})) \qquad [1]$$

The function $\varphi(\mathbf{v})$ is a penalty function that constrains the number of possible solutions for eqn.1. The coefficient $\boldsymbol{\lambda}$ determines the extent of the constraint imposed by $\varphi(\mathbf{v})$ on the solution. In your case, the matrix **E** is an encoding matrix that generated the data "**d**" in the **data** matrix as an output after the operation $\mathbf{Ev_{orig}=d}$ **.**

Tip 1: In this problem, the encoding matrix **E** is equivalent to taking an image and (1) performing a 2D FFT operation on it followed by (2) performing a point by point multiplication of the 2D Fourier data matrix with "**mask**" matrix.

You will need to implement these operations when computing $\|\mathbf{Ev}-\mathbf{d}\|_2^2$ and its gradients in the iterative reconstruction algorithm.

Tip 2: It is easy to compute $\|\mathbf{Ev}-\mathbf{d}\|_2^2$. I have already provided with the estimate of **v** that I call **res** in the demo_Brain_2D_edit code. Inside fnlcg code, you must simply implement (**Ex-d)'*(Ex-d) and its gradient.** In both these steps, you will need to implement E as described in Tip1. The iterative steps will keep refining **x** to ultimately provide you with the ideal solution.

Step 2: Use the research as a reference to determine **ϕ(v)** and these are many **ϕ(v)** and their associated λ that you can use as listed in the paper to improve on your image estimates.

Caution:

The variable param (stands for parameters) supplied to fnlcg has not been specified. You need to decide what to supply to the function. TVWeight is the λ for the TV penalty function. I can assure you that you need to, at the very least, use the TV function.

Please state any doubts etc in class so that I can regularly clarify the same. If you struggle in the implementation, you need to visit me in my office and I can help in the implementation.