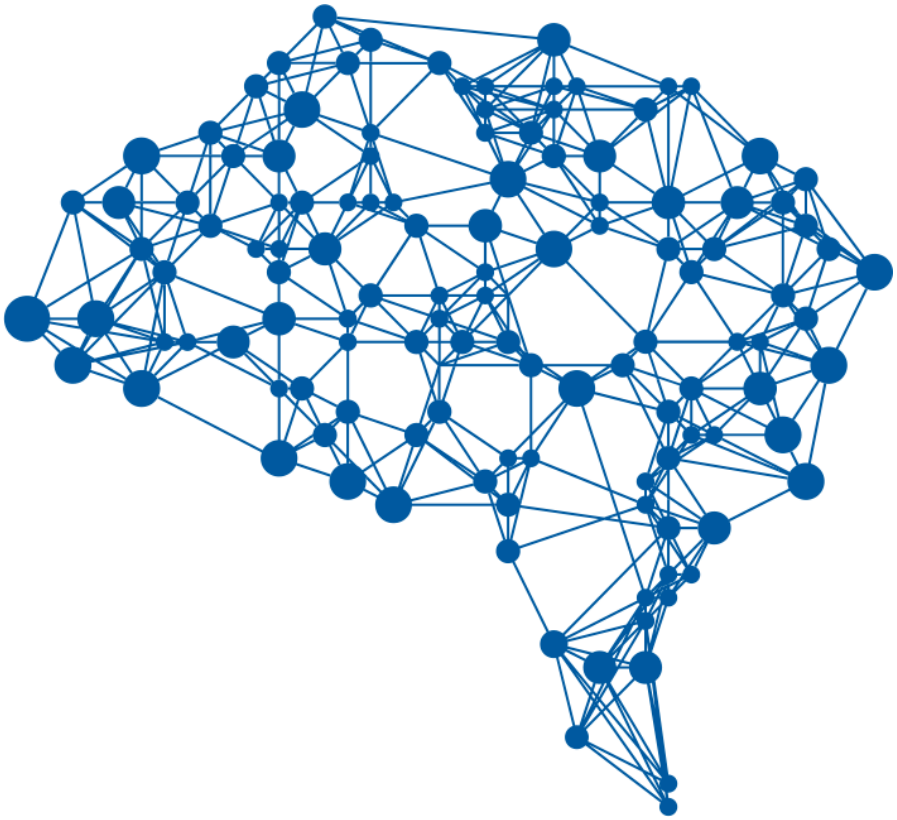
图像处理中的数学方法第一次作业报告



安捷 1601210097

2017 年 3 月 17 日

Split Bregman (ADMM) Algorithm

(i) Set initial guesses d_0 and b_0.

(ii) For $k = 0, 1, \ldots,$ perform the following iteratio

$$u_{k+1} = (A^\top A + \mu W^\top W)^{-1}(A^\top f + \mu W^\top($$
$$d_{k+1} = \mathcal{T}_{\boldsymbol{\lambda}/\mu}(W u_{k+1} + b_k),$$
$$b_{k+1} = b_k + \delta(W u_{k+1} - d_{k+1}).$$

(iii) Stop the iterations of (ii) when

$$\frac{\|W u_{k+1} - d_{k+1}\|_2}{\|f\|_2} < \text{tol}.$$

图 1: 算法流程

为了实现本次作业的需求，我基于 MATLAB，实现了 ADMM 算法，利用迭代方法实现了去模糊 + 去噪的图像重建任务。下面从算法简述，参数设置，数值实验结果，结论四个方面总结本次作业。

1 算法简述

其中，主要问题在于第一步，在这次作业的开始，我尝试使用线性化的思路来实现本次作业，即，将 A 算子与 W 算子实现的操作转化为一个矩阵，经过线性化之后，上述流程中的操作全部转化为矩阵与向量的操作，这一操作可行，但是存在很严重的两个问题：

1. 存储困难，线性化的操作使得矩阵 A 与 W 的维数是原图像的平方，即 128*128 的图像需要的双精度存储量为约 1200MB，512*512 的图像需要的双精度存储量为约 80GB，如果不采取稀疏矩阵存储方式，几乎不可存储；

2. 计算耗时长，在我使用的基于 NVIDIA CUDA 的 GPU 加速技术之后，依然速度很慢，主要原因在于矩阵求逆操作到目前为止没有效率很高的快速计算方法，即使采用 GPU 加速技术；

基于以上两点原因，我放弃了这一方法，转为采用老师上课提到的利用 Fourier Transform 求解 Possion 方程的算法来求解 ADMM 算法的第一步。下面将算法简述如下：

1.