

5/15/2024

آز پردازش سیگنال و تصاویر

پزشکی

گزارش آزمایش هشتم

محمدحسین شفیعی زادگان

99104781

محمدجواد نوروزی

99102434

2	اثبات رابطه Gradient Descent .....
2	سوال (1) .....
4	سوال (2) .....
5	سوال (3) .....
6	سوال (4) .....
7	سوال (5) .....

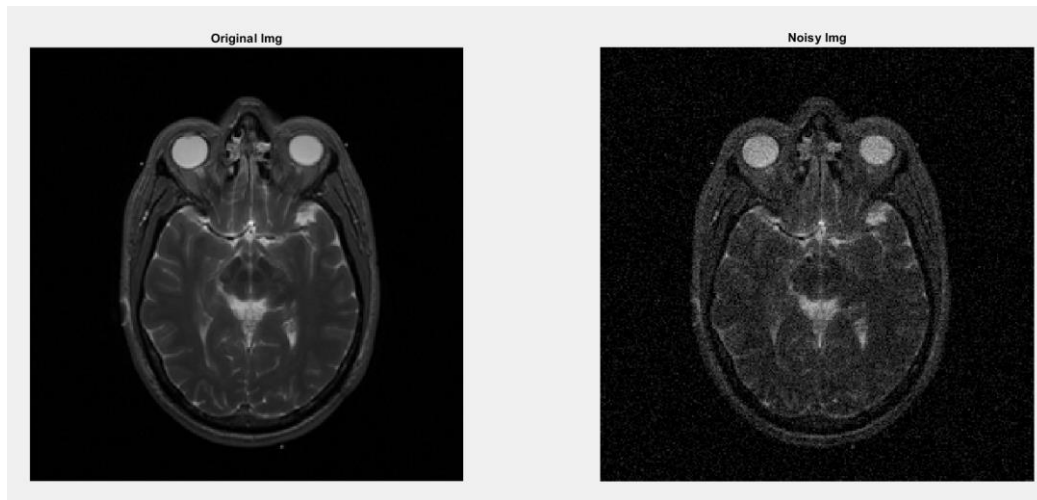
$$f = \operatorname{argmin} \|g - Df\|^2 \Rightarrow$$

$$\frac{\partial \|g - Df\|^2}{\partial f} = -2D^T(g - Df) = 0 \Rightarrow$$

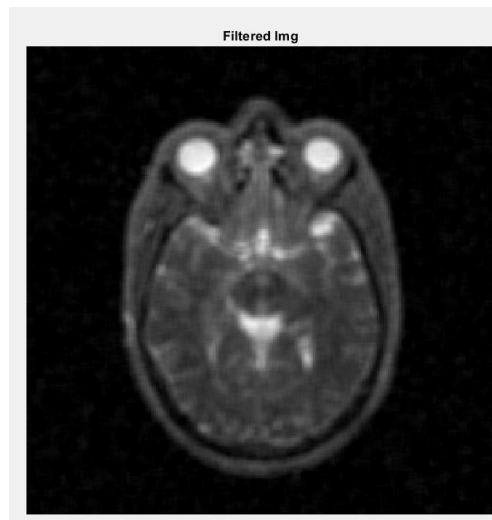
$$f_{k+1} - f_k = -2D^T(g - Df) \Rightarrow f_{k+1} = f_k + \beta D^T(g - Df)$$

### سوال 1)

در شکل زیر، تصویر اصلی و تصویر نویزی تولید شده را مشاهده می کنیم.

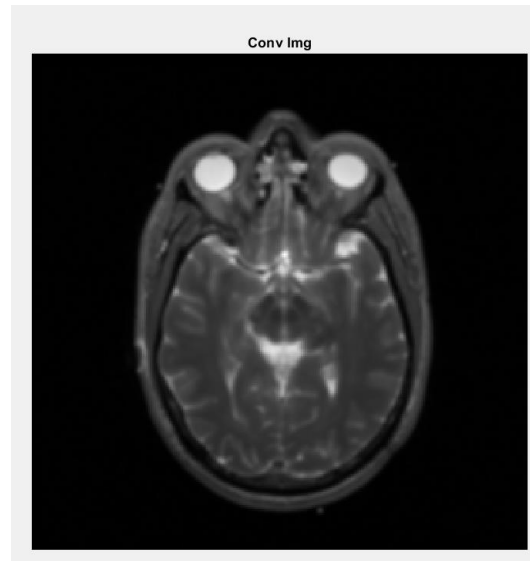


در ادامه، کرنل مربعی نرمالایز شده با سایز 4 در 4 را ساخته و به حوزه فرکانس می بریم و در تبدیل فوریه تصویر نویزی ضرب می کنیم. در آخر نتیجه را به حوزه مکان برمی گردانیم. نتیجه نهایی در شکل زیر مشاهده می شود.



با مقایسه آن با تصویر اصلی و نویزی، به خوبی مشاهده می شود که نویز گوسی از بین رفته و کیفیت تصویر بهبود داشته است اما نسبت به تصویر اصلی به اصطلاح blur شده است.

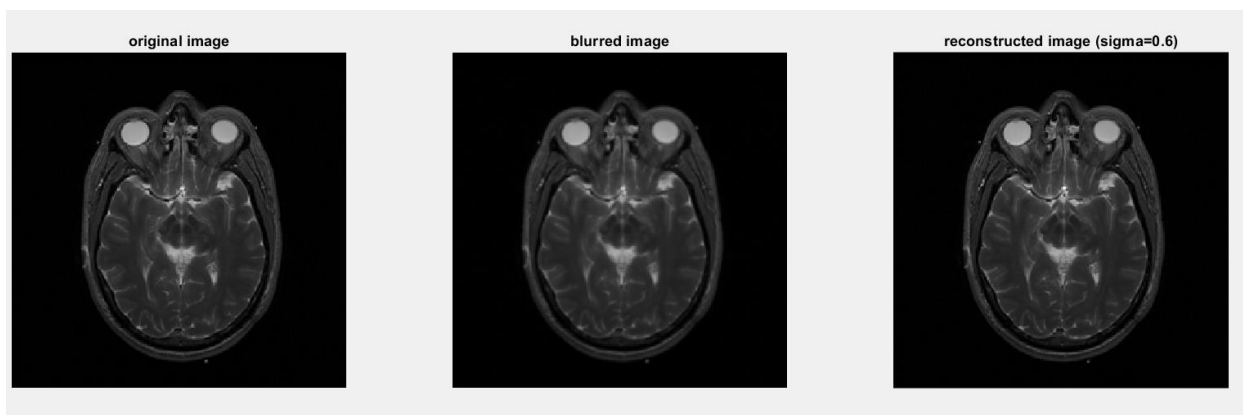
حال از دستور `imgaussfilt` استفاده می کنیم. نتیجه به صورت زیر خواهد شد.



مشاهده می شود که نویز گوسی به خوبی فیلتر و حذف شده است. نتیجه تا حدی شبیه با اعمال کرنل مربعی است اما دارای `sharpness` بیشتری است و نتیجه کمتر `blur` شده است.

## سوال 2)

تصویر اصلی، تصویر محو شده توسط فیلتر گاوسی با واریانس 0.6 و تصویر بازسازی شده آن در شکل زیر آمده اند:



همانطور که مشاهده می کنیم تا حد خوبی تصویر بازسازی شده است. حال نویز موردنظر را به تصویر اضافه می کنیم:



همانطور که مشاهده می شود تصویر به خوبی بازسازی نشده.

### سوال 3)

مطابق خواسته دستور کار ماتریس D را می‌سازیم:

```
for r = 1:size(K, 1)
    Kp = circshift(K,r-1,1);
    for c = 1:size(K, 1)
        Kz = circshift(Kp,c-1,2);
        D((r-1) * length(K) + c, :) = reshape(Kz, [1, length(K).^2]);
    end
end
```

ماتریس h را نیز با استفاده از تابع Gaussian که در اختیار ما قرار داده شده است می‌سازیم و آنرا نرمالایز می‌کنیم. مقدار آن در زیر آمده است:

h =

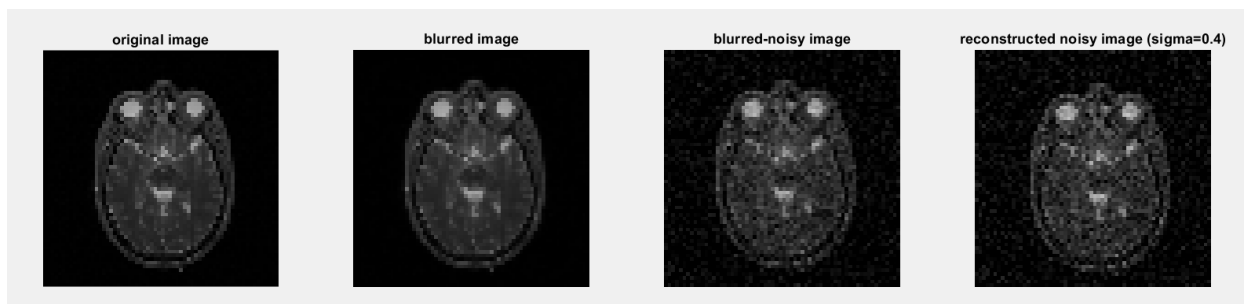
```
0.0016    0.0371    0.0016
0.0371    0.8450    0.0371
0.0016    0.0371    0.0016
```

ضرب  $\hat{f} = D^+g$  را میتوان به صورت زیر بازنویسی کرد:

$$\hat{f}[c + rN] = \sum_{k=0}^{N^2} D[(c + rN) \times N + k].g[k] = K_{c,r}.G = \sum_{i=0}^3 \sum_{j=0}^3 h[i][j].G[r + i][c + j]$$

همانطور که میدانیم رابطه فوق نشان دهنده رابطه کانولوشن ماتریس G در h می‌باشد.

تصویر اصلی (دان سمپل شده)، تصویر محو شده، تصویر محو شده نویزی و در نهایت تصویر بازسازی شده در زیر آمده اند:



همانطور که مشاهده می‌شود، فیلتر کردن باعث کاهش نویزها و پدیدارتر شدن لبه ها شده است اما تاثیر آن بساز ناچیز است.

خطای بازسازی:

```
reconstruction error= 1.832144e+00
```

## سوال 4)

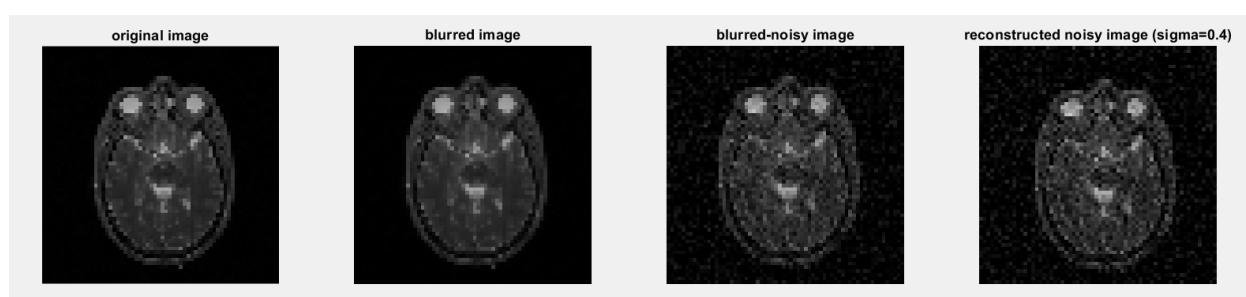
الگوریتم گرادیان کاهشی را بصورت زیر پیاده سازی می کنیم:

```
g_res = reshape(g_noisy, [size(g, 1) * size(g, 2), 1]);
f_reconst = zeros(size(g_res));

for i = 1:numiter
    f_reconst = f_reconst + beta * D' * (g_res - D*f_reconst);

    err = norm(g_res-D*f_reconst);
    fprintf('iter= %d, error= %d \n', i, err)
    if err < epsilon
        break
    end
end
```

نتیجه به صورت زیر می شود:



و خطای بازسازی هم در زیر آمده است (حلقه 957 بار اجرا شده است تا به خطای کمتر از یک درصد برسد):

```
iter= 950, error= 1.037443e-02
iter= 951, error= 1.031648e-02
iter= 952, error= 1.025885e-02
iter= 953, error= 1.020156e-02
iter= 954, error= 1.014460e-02
iter= 955, error= 1.008796e-02
iter= 956, error= 1.003165e-02 |
iter= 957, error= 9.975663e-03
```

## سوال 5)

روش Anisotropic Diffusion filtering تکنیکی است که برای کاهش نویز در تصاویر و در عین حال حفظ جزئیات مهم مانند لبه ها، خطوط و ساختارهای دیگر استفاده می شود. این یک تغییر شکل غیرخطی و فضایی از تصویر اصلی است که با روش های سنتی فیلتر خطی، که تمایل به محو کردن لبه ها و جزئیات ظریف دارند، متفاوت است.

این فرآیند شامل یک معادله انتشار است که بر اساس محتوای محلی تصویر اقتباس شده است. این بدان معنی است که فیلتر در مناطقی با اطلاعات کمتر مهم (مانند مناطق صاف) و با شدت کمتری در لبه ها و جزئیاتی که باید حفظ شوند، اعمال می شود. نتیجه تصویری است که عناصر ساختاری حیاتی خود را حفظ می کند و در عین حال نویز را کاهش می دهد.

معادله انتشار عمومی مورد استفاده در این روش به صورت زیر است:

$$\frac{\partial I}{\partial t} = \nabla \cdot (c(x, y, t) \nabla I)$$

که در آن  $I$  شدت تصویر و  $c(x, y, t)$  ضریب انتشار است که تابعی از مکان و زمان است.

در کد ارائه شده، این روش پیاده سازی شده است. در ادامه کد را توضیح می دهیم.

این کد ابتدا پارامترهای مربوط به گام زمانی، مدت زمان کل و مقیاس بندی انتشار را مقداردهی اولیه می کند، سپس به صورت iterative گرادینان تصویر، بزرگی گرادینان و دیورژانس را محاسبه می کند. یک تابع توقف لبه برای کنترل نرخ انتشار، با شرایط برای محو کردن ساده یا یک مدل پیچیده تر، اعمال می شود. تصویر در هر مرحله زمانی به روز می شود و انتشار را شبیه سازی می کند و به صورت دوره ای برای مقایسه تصویر اصلی و پردازش شده تصویرسازی می شود. این فرآیند تا زمان پایان مشخص شده ادامه می یابد، و تصویر را با هموارسازی در مناطق بهبود می بخشد و در عین حال مرزهای قابل توجهی مانند لبه ها را حفظ می کند..

خروجی کد به صورت زیر است.

