5/15/2024



محمدحسین شفیعی زادگان

99104781

محمدجواد نوروزی

99102434

آز پردازش سیگنال و تصاویر پزشکی

گزارش آزمایش هشتم

**فهرست مطالب**

[اثبات رابطه Gradient Descent 2](#_Toc168485272)

[سوال1) 2](#_Toc168485273)

[سوال2) 4](#_Toc168485274)

[سوال3) 5](#_Toc168485275)

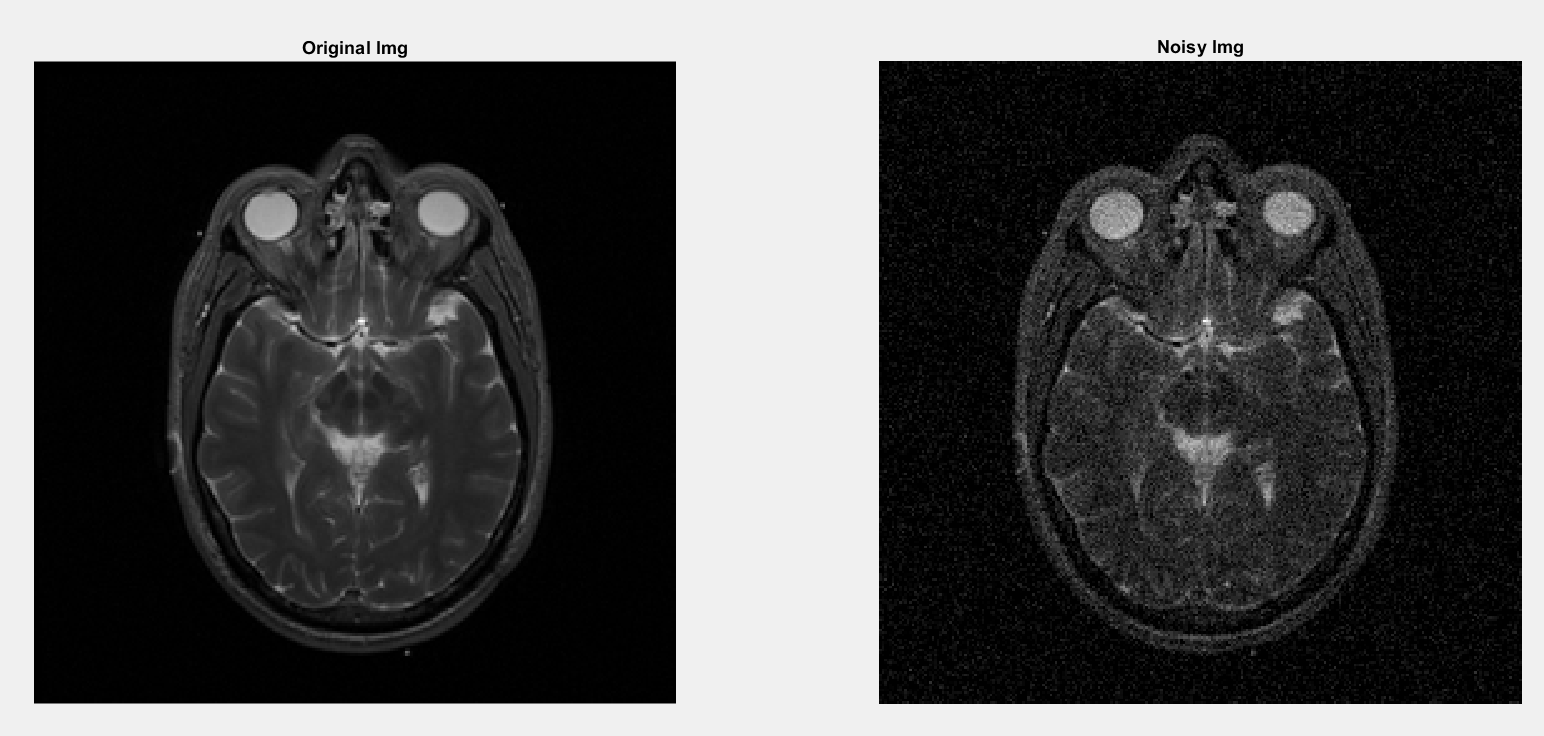
[سوال4) 6](#_Toc168485276)

[سوال5) 7](#_Toc168485277)

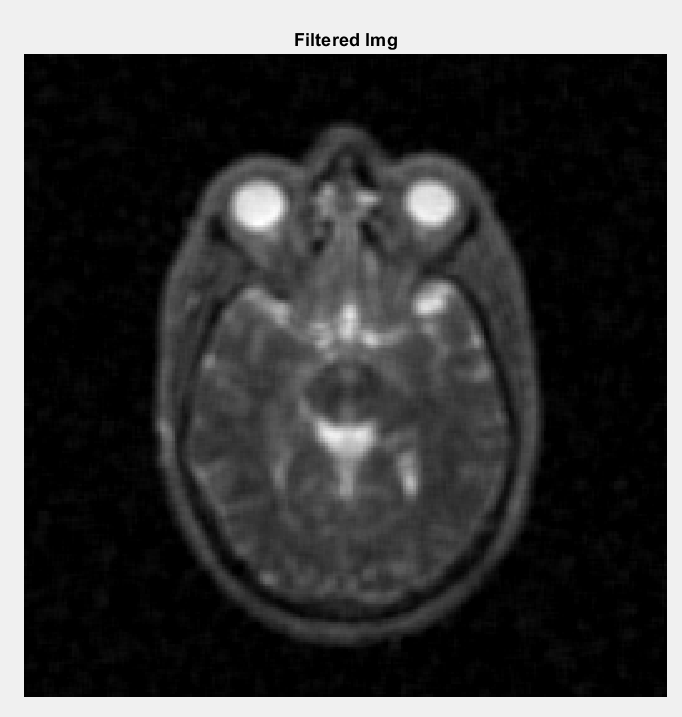
# اثبات رابطه Gradient Descent

# سوال1)

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

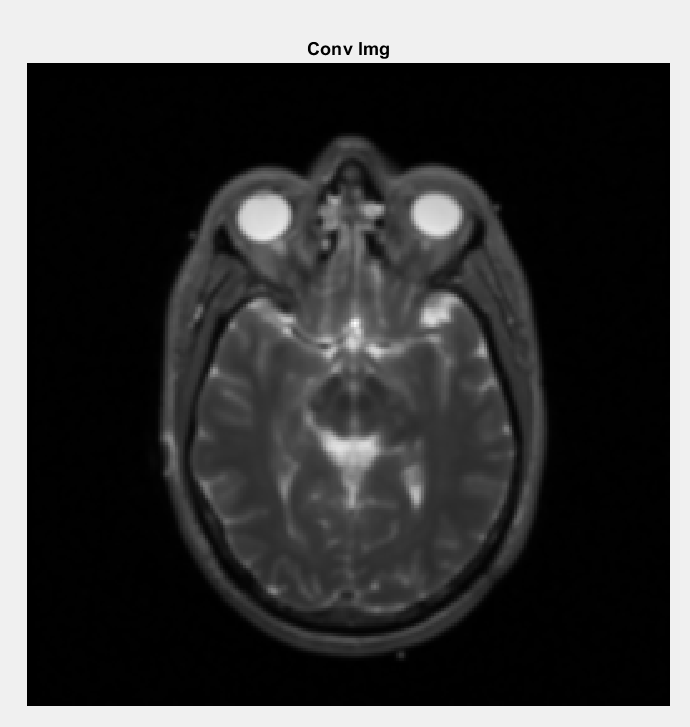


در ادامه، کرنل مربعی نرمالایز شده با سایز 4 در 4 را ساخته و به حوزه فرکانس می بریم و در تبدیل فوریه تصویر نویزی ضرب می کنیم. در آخر تنیجه را به حوزه مکان برمی گردانیم. نتیجه نهایی در شکل زیر مشاهده می شود.



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.

# سوال2)

تصویر اصلی، تصویر محو شده توسط فیلتر گاوسی با واریانس 0.6 و تصویر بازسازی شدۀ 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 می‌شود تصویر به خوبی بازسازی نشده.

# سوال3)

مطابق خواستۀ دستورکار ماتریس D را می‌سازیم:

for r = 1:size(K, 1)

Kp = circshift(K,r-1,1);

for c = 1:size(K, 1)

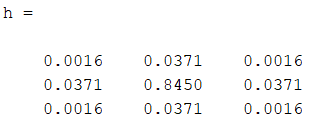
Kz = circshift(Kp,c-1,2);

D((r-1) \* length(K) + c, :) = reshape(Kz, [1, length(K).^2]);

end

end

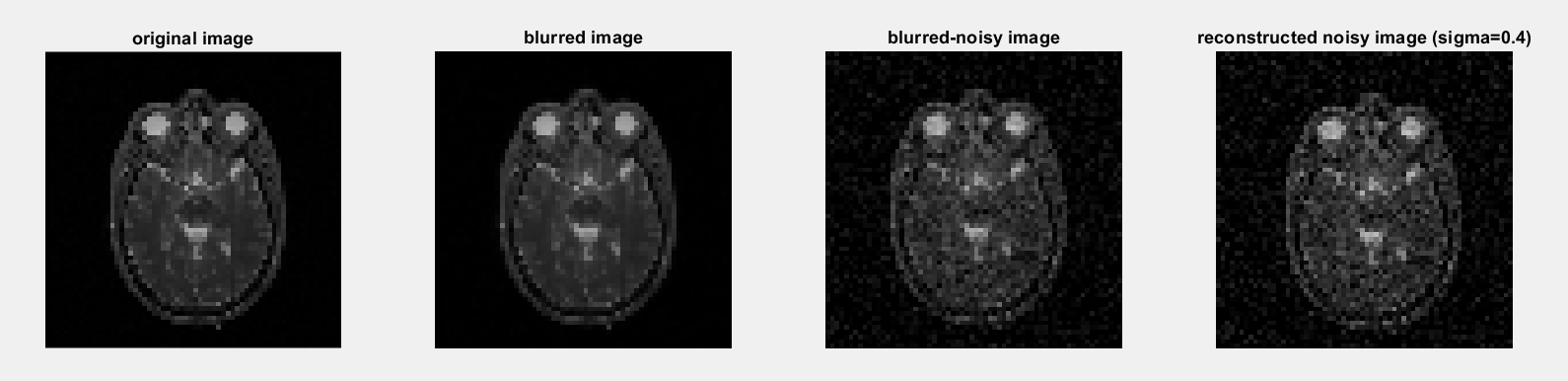
ماتریس h را نیز با استفاده از تابع Gaussian که در اختیار ما قرار داده شده است می‌سازیم و آنرا نرمالیایز میکنیم. مقدار آن در زیر آمده است:



ضرب را میتوان به صورت زیر بازنویسی کرد:

همانطور که میدانیم رابطۀ فوق نشان دهندۀ رابطۀ کانولوشن ماتریس G در h می‌باشد.

تصویر اصلی (دان سمپل شده)، تصویر محو شده، تصویر محو شدۀ 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:



# سوال4)

الگوریتم گرادیان کاهشی را بصورت زیر پیاده سازی می کنیم:

g\_resh = reshape(g\_noisy, [size(g, 1) \* size(g, 2), 1]);

f\_reconst = zeros(size(g\_resh));

for i = 1:numiter

f\_reconst = f\_reconst + beta \* D' \* (g\_resh - D\*f\_reconst);

err = norm(g\_resh-D\*f\_reconst);

fprintf('iter= %d, error= %d \n', i, err)

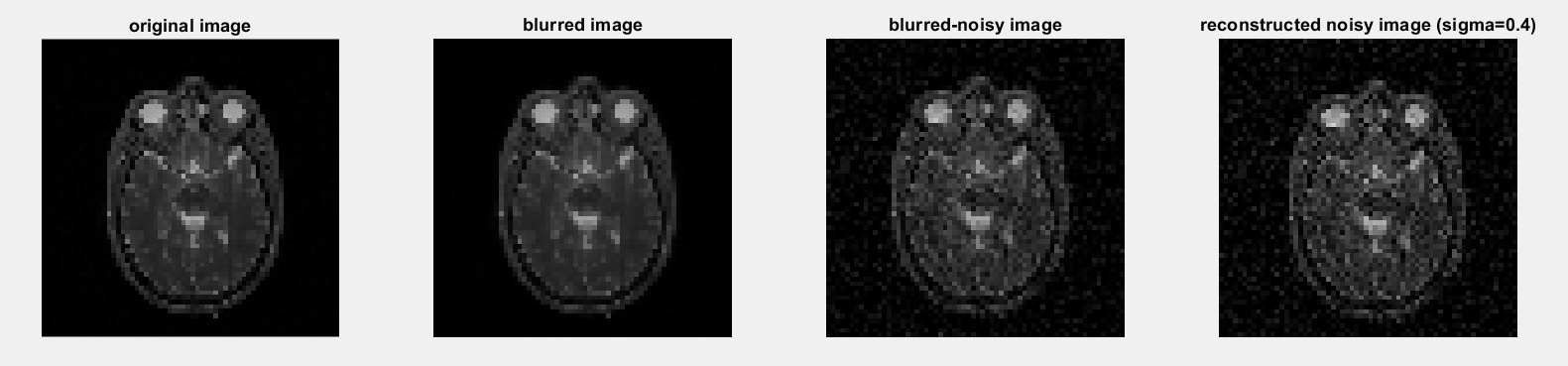
if err < epsilon

break

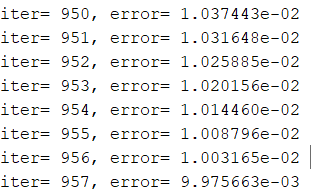
end

end

نتیجه به صورت زیر می‌شود:



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



# سوال5)

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 روش‌های سنتی فیلتر خطی، که تمایل به محو کردن لبه‌ها 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:

که در آن شدت تصویر و 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 گرادیان تصویر، بزرگی گرادیان 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 می‌یابد، و تصویر را با هموارسازی در مناطق بهبود می‌بخشد و در عین حال مرزهای قابل توجهی مانند لبه‌ها را حفظ می‌کند..

خروجی کد به صورت زیر است.

