

Data Fitting & Denoising

در این سری از تمرین قرار است با ۲ موضوع **Data Fitting** و **Denoising** بیشتر آشنا شوید و با استفاده از متد ها و فرمول های بیان شده در کلاس و ذکر شده در منبع اصلی درس، آزمایش هایی را در جهت یادگیری بیشتر مفاهیم ذکر شده انجام دهید.

برای انجام این سری از تمرین پیشنهاد میشود از قالبی که در اختیارتون قرار میگیره استفاده کنید، که در آن برای هر بخش عنوان قرار گرفته شده است و همین برای راحتی دانشجویان طور چند تابع کمکی از قبل داخل آن قرار گرفته است که در ادامه بیشتر به آنها میپردازیم.

استفاده از این قالب اختیاری است اما به صورت کلی باید فایل نهایی شما در فرمت ipynb باشد و همین طور شیوه نامگذاری فایل ها نیز به صورت زیر باشد:

StudentName_StudentID_HW1.ipynb

StudentName_StudentID_HW1.zip

همین طور برای هر سوال و هر بخش نیاز است عنوان مشخص باشد تا در روند بررسی پاسخ ها مشکلی نباشد.

هر یک از سوالات و بخش های آنان دارای نمره خاص خود است و مقدار بارم هر یک کنار آن ذکر شده است. برای مثال:

پیاده سازی تابع برای Denoising (۵ نمره)

همین طوز بعضی از بخش ها دارای نمره امتیازی هستند که نمره آن ها به شکل زیر نشان داده میشود و پاسخ به آن ها اختیاری خواهد بود:

پیدا کردن راهی برای بازیابی جزئیات عکس در پس از از بین بردن نویز (۵+ نمره)

از طریق این لینک میتوانید فایل های مربوطه به این تمرین و همین طور تمپلیت پیشنهادی پاسخگویی را دریافت نمایید.

سوال اول

مسئله Denoising از دسته مسائلی هست که به علت کاربرد گسترده آن، همواره سعی بر پیدا کردن راه های بهینه و همین طور بهبود روش های گذشته بوده است. از جمله کاربرد های آن میتوان به کاربرد در بهبود تصاویر و صدا و همین طور انتقال امواج میان فرستنده ها اشاره کرد. در این بخش قرار است با استفاده از مفاهیم نظری ذکر شده در کلاس به از بین بردن انواع نویز در عکس ها بپردازید.

- برای باز کردن عکس ها به صورت یک ماتریس میتوانید از تابع `load_image` استفاده کنید که در قالب آماده شده نیز آماده است.

```
1 import cv2
2 def load_image(path,size = (256,256)):
3     img = cv2.imread(path,0)
4     img = cv2.resize(img1, size)
5     return img
```

قرار است به بررسی ۳ نوع نویز بپردازیم که شامل موارد زیر میشود:

- Photo with vertical noise
- Photo with horizontal noise
- Photo with random noise



نمونه عکس با نویز عمودی



نمونه عکس با نویز افقی



نمونه عکس با نویز تصادفی

دو نمونه ساده عکس با تعداد مختلف نویز های عمودی در فایل های `img1.png` , `img2.png` , و ۲ نمونه با نوز افقی در فایل های `img3.png` , `img4.png` و ۲ نمونه دارای نویز تصادفی در فایل های `img5.png` , `img6.png` و میبایست این آزمایش را با مقادیر مختلف λ (شدت اعمال حذف نویز) تست کنید و بهترین نتیجه به دست آمده را نشان دهید را نشان دهید.

برای نشان دادن عکس های به دست آمده میتوانید از تابع زیر استفاده کنید:

```

1 | import matplotlib.pyplot as plt
2 | def show_image(image):
3 |     plt.imshow(image, cmap='gray')

```

بارم دهی و زیرمسئله های این سوال به شرح زیر میباشد:

(۱.۱) پیاده سازی تابع برای Denoising توالی از اعداد (۱۵ نمره)

(۱.۲) اجرای فرایند بر روی هر یک از عکس ها و نشان دادن بهترین نتیجه به دست آمده (مجموعاً ۲۰ نمره)

پرسش ها:

(۱.۳) با توجه به نتایج به دست آمده، آیا فکر میکنید این روش برای از بین بردن نویز در عکس ها مناسب

است؟ ایراد استفاده از این روش چیست؟ در کدام نوع نتایج بهتری را دریافت کردید. (۱۰ نمره)

(۱.۴) آیا اعمال کردن فرایند Denoising بر روی یک توالی (یا عکس) با پارامتر $\lambda = \alpha$ معادل با اعمال کردن این

فرایند α بار با پارامتر $\lambda = 1$ خواهد بود؟ با یک نمونه توضیح دهید. (۵ نمره)

(۱.۵) راهی برای بازیابی جزئیات از دست رفته در عکس ها بعد از فرایند Denoising پیشنهاد دهید و بر روی

یک مورد تست کنید. (۱۰+ نمره)

سوال دوم

در این قسمت قرار است به پیاده سازی متد معرفی شده تحت عنوان **Data Fitting** با استفاده از روش **Least Squares** بپردازیم. ابتدا نیاز است یک تابع برای این کار بر طبق فرمول بیان شده در کلاس پیاده سازی کنید و سپس این مورد را روی دیتای موجود در فایل `data1.txt` و `data2.txt` اجرا کرده و چند جمله ای حاصل را رسم کنید. سپس در ادامه با استفاده از تابعی که برای از بین بردن نویز در سوال قبلی نوشتید، ابتدا دیتای موجود در فایل های `data3.txt` و `data4.txt` را **Denoise** کرده و سپس یک چند جمله ای با درجه n به آنها فیت کنید. $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ در مرحله بعد باید تابعی تعریف کنید برای محاسبه میزان خطای هر یک از منحنی های به دست آمده نسبت به نقطه های هر یک از داده ها و این تابع به صورت زیر تعریف میشود:

$$error(f) = \sum_{i=0}^n |f(x_i) - y_i|$$

سپس باید تناسب میان λ و $error(f)$ را نشان دهید و برای این کار میبایست برای λ مقداری میان ۰ تا ۳۰ در نظر گرفته ($\lambda \in [0, 30]$) و آزمایش را انجام دهید و سپس $error(f)$ را به ازای هر یک محاسبه کنید و در آخر نمودار تناسب میان خطار و ضریب نویز را رسم کنید. (برای فایل های data3.txt و data4.txt)

**** توجه داشته باشید که خطای تعریف شده بر حسب اختلاف نقاط پس از حذف نویز با نقاط چند جمله ای حاصل خواهد بود ****

برای خواندن از هر یک از فایل ها می توانید از تابع کمکی زیر استفاده کنید:

```
1 def load_data(path):
2     file = open(path).read().split('\n')
3     lines = [line.split() for line in file]
4     return [(eval(x[0]),eval(x[1])) for x in lines ]
```

بارم دهی و زیرمسئله های این سوال به شرح زیر میباشد:

- ۲.۱ پیاده سازی تابع Data Fitting در جهت پیدا کردن ضرایب چند جمله ای (۱۵ نمره)
- ۲.۲ اجرای تابع بر روی فایل های مربوطه (حدس درجه مناسب برای تطبیق حداکثری) و رسم نمودار برای خود داده ها و چند جمله ای به دست آمده برای هر یک (۱۵ نمره)
- ۲.۳ رسم نمودار خطا بر حسب پارامتر λ به ازای بازه ۰ تا ۳۰ (درجه چند جمله ای را ثابت و برابر با ۱۰ در نظر بگیرید) (۱۰ نمره)

پرسش ها:

- ۲.۴ آیا همواره میتوان یک چند جمله ای مناسب را برای تخمین یک دیتاست مشخص به دست آورد؟ برداشت شما از شرایط لازم برای وجود چنین شرایطی چیست؟ (۵ نمره)
- ۲.۵ تاثیر پارامتر حذف نویز چگونه است؟ آیا پیوسته سازی باعث افزایش دقت خواهد شد؟ برداشت خود را توضیح دهید. (۵ نمره)