

پروژه پایانی جبر خطی

حمیدرضا ربیعی، مریم رضانی

نام و نام خانوادگی: شماره دانشجویی:

چهره‌های ویژه

در الگوریتم‌های تشخیص چهره، درون عکس‌ها به دنبال تشخیص خصوصیت‌های خاص عکس بودند. به طور مثال چشم، بینی و ... که می‌توانست حاکی از وجود چهره در عکس باشند. واضح است که چنین روش‌هایی هزینه‌بر و در موارد زیادی نادقیق هستند.

در این پروژه به دنبال پیاده سازی روشی هستیم که این تشخیص را به گونه‌ای دیگر انجام می‌دهد. در این روش مانند روش قبل به دنبال پیدا کردن مشخصه‌هایی هستیم که یک چهره را تشخیص می‌دهند، اما این مشخصه‌ها را با روش دیگری می‌خواهیم پیاده کنیم.

Load Data (10 points)

قبل از اینکه درباره پیاده‌سازی این الگوریتم صحبت کنیم لازم است داده‌هایی که مورد استفاده قرار می‌دهیم را داشته باشیم. داده‌های مورد نظر در زیپ مربوطه همراه این نوتبوک قرار گرفته‌است. دقت کنید که این دیتاست شامل تصاویر مربوط به چهره 200 فرد است. هر کدام از افراد دو عکس یکی حالت طبیعی و یکی خندان دارند. پس در کل 400 عکس داریم. ابتدا تمام عکس‌ها را لود کنید و داشته باشید که در ادامه به آن‌ها نیاز پیدا می‌کنید.

در نهایت برای این که مطمئن شوید که دیتا را درست لود کرده‌اید، به طور رندوم 5 تا از این چهره‌ها را در دو حالت خندان و عادی نمایش دهید. برای اینکار می‌توانید از تابع `imshow` از لایبرری `matplotlib` استفاده کنید.

حال فعلا تنها با عکس‌های چهره طبیعی کار داریم. 190 تا عکس اول از آن را در نظر بگیرید هر کدام را به صورت یک وکتور Γ_i بعدی Γ_i در بیاورید. در نهایت ماتریس حاصل از داده‌ها که Γ است را تشکیل دهید. دقت کنید که ستون i ام این ماتریس Γ_i است.

In []:

PCA

الگوریتم Principal Component Analysis یا همان PCA یک روش برای کاهش ابعاد داده‌ی موردنظر است که در حوزه یادگیری ماشین و علوم داده کاربرد فراوان دارد. با کاهش ابعاد سادگی را افزایش داده و در عین حال همانگونه که انتظار می‌رود ورودی با کاهش دقت همراه خواهد شد. در الگوریتم PCA تلاش بر این است که در عین کاهش ابعاد، کمترین میزان اطلاعات را از دست بدهیم. پیشنهاد می‌شود قبل از انجام پروژه مطالعات بیشتری در رابطه با این الگوریتم داشته باشید.

در ادامه‌ی بخش‌ها می‌خواهیم PCA را پیاده سازی کنیم. گفتیم که برای تشخیص چهره‌ها به دنبال پیدا کردن مشخصه‌های خاص درون هر عکس هستیم. این مشخصه‌ها ممکن است ترکیبی از چند ویژگی درون عکس باشند، یا حتی ویژگی‌هایی باشند که به طور طبیعی قادر به تشخیص آنها نیستیم. در اینجا با استفاده از الگوریتم PCA در تلاش برای پیدا کردن همچنین ویژگی‌هایی هستیم.

با PCA قرار است کار زیر را انجام دهیم :

فرض کنید هر عکس به صورت یک آرایه دو بعدی $N \times M$ باشد ، بنابراین هر عکس را می‌توان با یک بردار به طول $N \times M$ نشان دهیم، پس در یک فضای $N \times M$ بعدی قرار دارد. فرض کنید W فضایی باشد که تمامی عکس‌های صورت در آن قرار دارند ، اسم آن را Face Space بگذارید. در اینجا به دنبال پیدا کردن K بردار عمود بر هم هستیم که فضای W را Span می‌کنند.

Mean image (5 points)

در این قسمت برای نرمالایز کردن نیاز به mean image داریم. آن را بدست آورید و تصویر آن را نمایش هم دهید.

$$\Psi = \frac{1}{n} \sum_{i=1}^n \Gamma_i$$

در اینجا n تعداد داده های شماست. دقت کنید فعلا کل کار ما بر روی 190 عکس چهره طبیعی است.

In []:

Cov Matrix (15 points)

در این قسمت باید ماتریس کوواریانس را به صورت زیر تعریف کنید :

$$\Phi_i = \Gamma_i - \Psi$$

$$A = [\Phi_1 \Phi_2 \dots \Phi_n]$$

$$C = AA^T$$

حال بردار ویژه های ماتریس C به ترتیب بزرگی مقدار ویژه های متناظرشان ، Principal Components هایی هستند که ما می‌خواهیم.

دقت کنید که ماتریس C ماتریسی

$$(N \times M) \\ \times (N \times M)$$

می‌باشد ، سعی کنید روش بهینه ای برای پیدا کردن بردار ویژه های آن پیاده سازی کنید. به عنوان راهنمایی از $A^T A$ کمک بگیرید. مقدار این مقادیر ویژه را سورت کنید و درون یک نمودار نشان دهید. بعد از سورت کردن مقدار ویژه ها ، بردار ویژه متناظر با 5 مقدار ویژه اول را بکشید. این بردارهای ویژه را eigen face می‌نامیم. در نهایت با کمی مطالعه معیاری را انتخاب کنید و تعداد مورد نیاز از Principal Component ها از نظرتان را مشخص نمایید و آن تعداد بردار ویژه را انتخاب کنید. این عدد را K در نظر بگیرید. همچنین ماتریس شامل PC ها را U در نظر بگیرید.

In []:

Reconstruction images (20 points)

در این قسمت قرار است که تابعی بنویسید که عکسی را به همراه Principal Components های مورد نظر بگیرد و آن را دوباره با آن PC ها بازسازی کند. برای نحوه ی انجام این کار از روابط زیر کمک بگیرید.

$$\begin{aligned} W \\ = U^T \\ (\Gamma \\ - \Psi) \\ \\ rec \\ = UW \\ + \Psi \end{aligned}$$

دقت کنید که Γ این جا عکس ورودی است.

In []:

```
def reconstruct_image(orig, pcs):
    """
    A function to reconstruct the original image with the given PCs
    Arguments
```

```
orig - Original image to perform reconstruction on  
pcs - The principal components to use for the reconstruction
```

Returns

```
recon - The reconstruction of the orig from pcs  
mse - The Mean Squared Error of the recon with orig  
""  
  
recon = None  
mse = None  
  
# todo  
return recon, mse
```

(بخش ب)

حال یکی از عکس‌های دیتاستی که تا الان روی آن کار کردیم (شامل 190 عکس طبیعی) را به صورت رندم انتخاب کنید. برای آن عکس به ازای تمام حالت‌های انتخاب K عملیات بازسازی را انجام دهید. کمترین و بیشترین MSE را به دست آورید. MSE را برای K ای که در قسمت قبل انتخاب کردید محاسبه نمایید. سپس نمودار MSE-number of largest principal components را رسم کنید. در مرحله‌ی بعد برای یک عکس رندم دیگر از دیتاست، عکس اصلی را همراه عکس‌هایی که از بازسازی با در نظر گرفتن 5 انتخاب مختلف برای K که کل محدوده تعداد eigen face ها را پوشش دهد، رسم کنید. نتایج خود را از کارهایی که این بخش انجام دادید و از روی خروجی‌های بدست آمده شرح دهید.

In []:

Smiling images (10 points)

حال این بار تمام کارهایی که در بخش قبل (بخش ب) انجام دادید را تکرار کنید. ولی این بار به جای عکس‌های رندم از چهره‌های عادی، عکس رندم از چهره‌های خندان انتخاب کنید. نتایج خود را از کارهایی که این بخش انجام دادید و از روی خروجی‌های

بدست آمده شرح دهید. به نظرتان دلایل تفاوت خروجی‌ها در هر قسمت نسبت به قسمت قبل چیست؟

In []:

Reconstruction of test set (10 points)

تا به اینجای کار ما با محاسبه MSE و بازسازی دوباره عکس‌ها به شهودی عددی و بصری از دقت فضای جدید یا همان face space خود رسیدیم. اما نمی‌توان از این نکته چشم‌پوشی کرد که ارزیابی‌های ما از فضای جدید با استفاده از عکس‌هایی انجام شده که با آنها face space را ساخته‌ایم. در نتیجه این ارزیابی ما به نوعی biased است. برای ارزیابی بهتر MSE را بر روی تعدادی عکس که الگوریتم آنها را تا به حال ندیده است انجام داده و آنها را بازسازی می‌کنیم. در این قسمت یک عکس از ۱۰ عکس دیده نشده حالت طبیعی را انتخاب نموده و کارهای بخش ب را بر روی آن پیاده کنید. ارزیابی چه تفاوتی کرده است؟ نتیجه‌گیری کنید.

In []:

Reconstruction of non-human image (10 points)

دو عکس غیر از چهره انسان و به دلخواه خود (مثل عکس ماشین یا پرنده) را در نظر گرفته. نخست عملیات‌های لازم مثل سیاه سفید کردن یا تغییر سایز را انجام دهید. سپس با استفاده از تمام PCها سعی بر بازسازی آن داشته باشید و MSE را محاسبه نمایید. آیا الگوریتم به اندازه عکس چهره‌ها موفق عمل کرد؟ چرا؟ نتیجه‌گیری کنید.

In []:

Image Rotation (10 points)

حال یکی از عکس‌های دیتاستی که تا الان روی آن کار کردیم (شامل 190 عکس طبیعی) را به صورت رندم انتخاب کنید. برای آن عکس تمام rotation‌های ممکن (از 0 تا 360) را انجام دهید. حال روی هر یک از عکس‌های بدست آمده با استفاده از تمام PC‌ها عملیات بازسازی و محاسبه‌ی MSE را انجام دهید. نمودار MSE بر حسب درجه‌ی چرخش را رسم کنید. برای 10 درجه به صورت یکنواخت بین 360 و 0، عکس اصلی، عکس بعد از دوران، عکس بازسازی‌شده به کمک الگوریتم را رسم کنید. نتایج به دست آمده را تحلیل کنید. آیا MSE در حالت 360 درجه دوران با حالت 0 برابر است؟ چرا؟

با استفاده از `ndimage` می‌توانید عکس‌های خود را بچرخانید.

In []:

Face Recognition (10 point)

در این قسمت شما باید Face Recognition را تکمیل کنید. به این معنا که کد شما باید این قابلیت را داشته باشد که پس از اجرای PCA، تصویرها را پشت سر هم ورودی بگیرد، و بعد از گرفتن هر تصویر بگوید که آیا این شخص را قبلاً دیده است یا نه.

برای اینکار لازم است به این صورت عمل کنید. در ابتدا 10 تا از بهترین پایه‌های فضا را انتخاب کنید. سپس هر تصویری که به شما داده میشود را میتوانید به صورت یک بردار 10 تایی از ضرایب نشان داد. که برای هر تصویر این ضریب منحصر به فرد میباشد. حال اگر برای 2 بردار 10 تایی خاص، این ضرایب به اندازه کافی به هم نزدیک باشند میتوانیم بگوییم که تصویر متعلق به یک شخص میباشد.

In []: