باسمه تعالی

تمرین سری چهارم درس یادگیری ماشین

استاد درس: جناب آقای دکتر باباعلی

نام دانشجو: ایمان کیانیان

شماره دانشجویی: ۶۱۰۳۰۰۲۰۳

۱- مقدمه

در این تمرین قرار است چند بخش که در صورت سوال آمده است را بررسی کنیم. مراحل محاسبه PC ها را بررسی میکنیم. با استفاده از LDA و PCA دسته بندی انجام داده و نتایج آن را تحلیل و مقایسه میکنیم. فشرده سازی تصویر را انجام میدهیم. آنها را بر میگردانیم به حالت اولیه و سپس تعداد PC بهینه را برای آن مسئله پیدا میکنیم.

۲- سوال اول

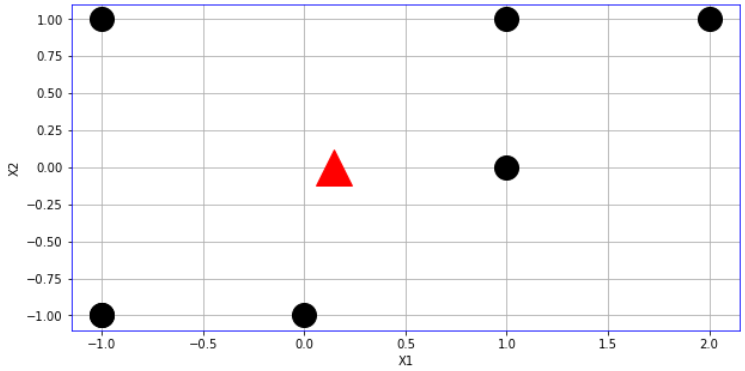
در سوال اول ۷ داده دو بعدی به شکل زیر داریم:

$$\left\{\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}2\\1\end{pmatrix},\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix}\right\}$$

ابتدا میانگین این داده ها را بدست می آوریم:

$$\mu = \begin{pmatrix}0.14285714\\0\end{pmatrix}$$

رسم داده ها و میانگین :



حال با استفاده از میانگین و داده های اصلی، داده ها را منهای میانگین میکنیم :

$$\tilde{X} = \left\{\begin{pmatrix}-0.14285714\\-1\end{pmatrix}.\begin{pmatrix}0.85714286\\0\end{pmatrix}.\begin{pmatrix}1.85714286\\1\end{pmatrix}.\begin{pmatrix}0.85714286\\1\end{pmatrix}.\begin{pmatrix}-1.14285714\\1\end{pmatrix}.\begin{pmatrix}-1.14285714\\-1\end{pmatrix}.\begin{pmatrix}-1.14285714\\-1\end{pmatrix}\right\}$$

سپس Covariance (کواریانس) داده های بالا را بدست می آوریم:

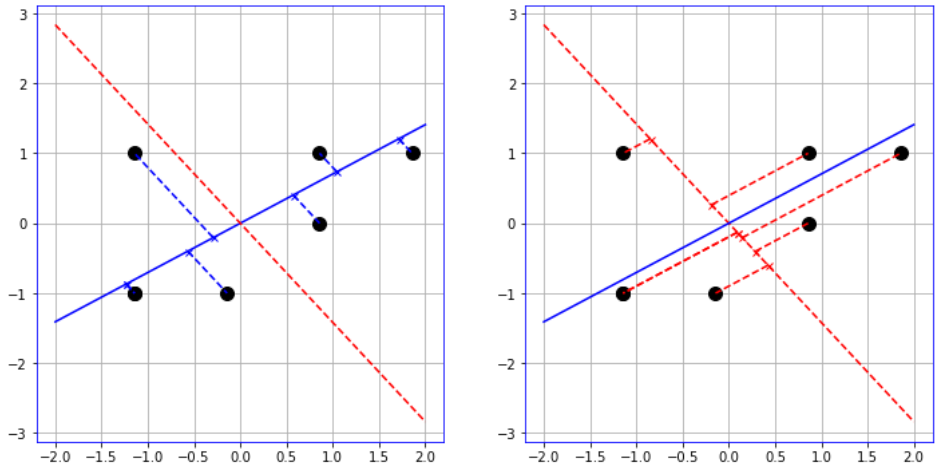
```
[1.47619048, 0.66666667],
[0.66666667, 1.        ]
```

در مرحله ی بعد باید مقادیر ویژه و بردار های ویژه این ماتریس کواریانس را بنویسیم:

$$\text{Eigen Values} = 1.94600327,\ 0.5301872$$

$$\text{Eigen Vectors} = \begin{pmatrix}0.81741556\\0.57604844\end{pmatrix}, \begin{pmatrix}-0.57604844\\0.81741556\end{pmatrix}$$

سپس بردار ویژه متناظر با بزرگترین مقدار ویژه که همان $\begin{pmatrix}0.81741556\\0.57604844\end{pmatrix}$ است را انتخاب میکنیم. PC اول برابر اولین بردار و PC دوم به عنوان دومین بردار انتخاب می شود . که به شکل زیر است:



همانطور که از ظاهر هم پیداست خط آبی متناظر با PC اصلی ماست چون بیشترین پراکندگی را داده روی آن دارد. خط عمود بر آن نیز PC دوم است که قرمز است. ما از اینجا به بعد با خط آبی یا PC اول و یا $\begin{pmatrix}0.81741556\\0.57604844\end{pmatrix}$ کار داریم. سپس با استفاده از ضرب زیر میتوانیم داده هامان را به یک بعد ببریم:

$$X' = \tilde{X} \cdot \begin{pmatrix} 0.81741556 \\ 0.57604844 \end{pmatrix}$$

که داده های جدید به صورت زیر خواهند بود:

$$X' = \{-0.69282209 \ . \ 0.70064191 \ . \ 2.09410591 \ . \ 1.27669035 \ . \ -0.35814078 \ . \ -1.51023765 \ . \ -1.51023765\}$$

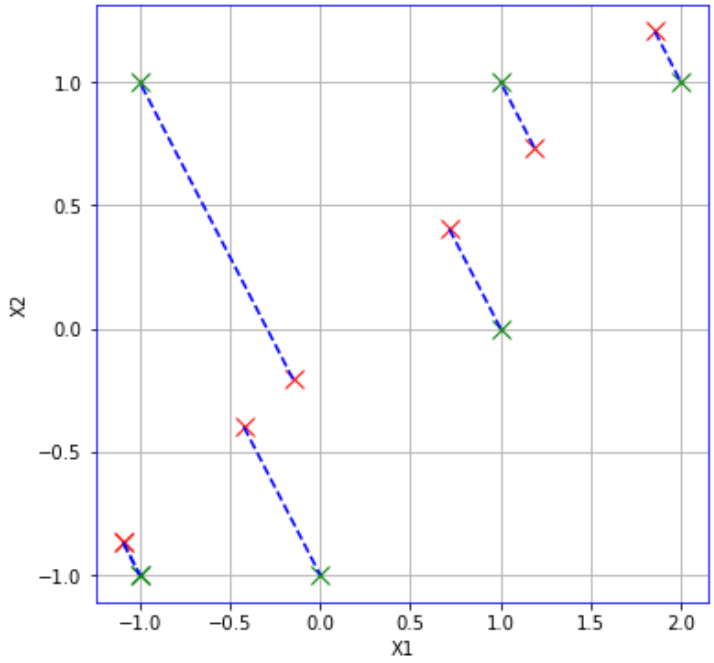
سپس با استفاده از فرمول زیر داده ها را دوباره بر میگردانیم:

$$X_{new} = \mu + \begin{pmatrix} 0.81741556 \\ 0.57604844 \end{pmatrix} \cdot X'$$

که چون X' یک ماتریس ۷ در ۱ است و $\begin{pmatrix} 0.81741556 \\ 0.57604844 \end{pmatrix}$ یک ماتریس ۲ در ۱ است پس حاصل کل یک ماتریس ۲ در ۷ همانند داده های اصلی خواهد بود. مقادیر زیر مقادیر بازگردانی شده از داده های اصلی هستند که میبینیم کمی فرق دارند و این اطلاعاتی است که در PCA از بین رفته است.

```
array([[-0.42346641, -0.39909908],
       [ 0.71557274,  0.40360368],
       [ 1.8546119 ,  1.20630643],
       [ 1.1864437 ,  0.73543548],
       [-0.1498927 , -0.20630643],
       [-1.09163461, -0.86997004],
      [-1.09163461, -0.86997004]])
```

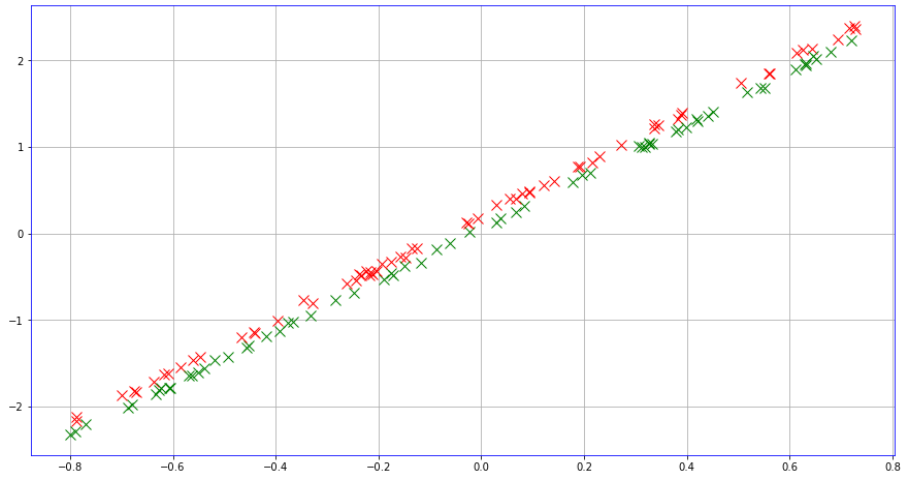
به شکل زیر توجه کنید:



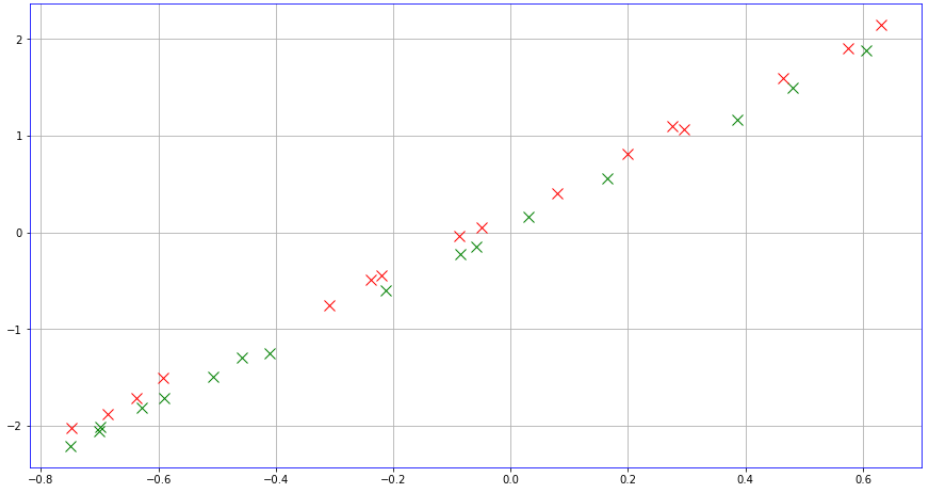
نقاط سبز رنگ داده های اصلی هستند و متناظر آنها نقطه ی قرمز رنگی کشیده شده است که در شکل با خط آبی بهم وصل شده اند. نقطه ی قرمز رنگ مقدار حال از بازگرداندن اطلاعات با روش PCA است. همانطور که میبینیم نتوانسته ایم واریانس داده ها را با یک مولفه ذخیره کنیم . درصد بالایی از واریانس مربوط به وجود مولفه دوم است (با توجه به مقادیر ویژه چون مقدار ویژه دوم خیلی کوچک نیست – اگر مقدار ویژه دوم خیلی کوچک باشد میتوان گفت که اولین PC واریانس قابل قبولی از داده را پوشش میدهد) پس نمیتوان گفت که بردار ویژه مربوط به مولفه اول ، ۱۰۰ درصد دقت دارد (۱۰۰ درصد واریانس را پوشش میدهد) و مقدار بالایی در این نمونه خطا وجود دارد.

۳- سوال دو

در این سوال از ما خواسته شده ابتدا داده های آموزشی و تست را در فضای دو بعدی نمایش دهیم. داده های آموزشی به صورت زیر است:



و داده تست به صورت زیر است :



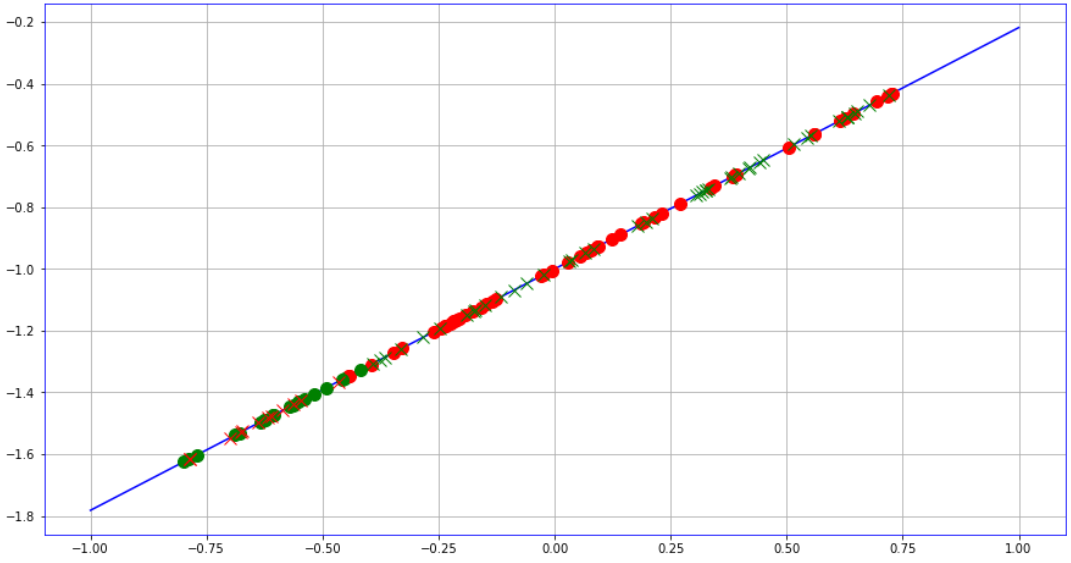
سپس از ما خواسته شده با استفاده از LDA ، داده ها را به فضای یک بعدی ببریم. سپس با استفاده از پرسپترون داده های تولید شده را آموزش دهیم و دقت را برای دسته بندی در این حالت بگیریم.

LDA را توسط داده های آموزشی ، آموزش دادیم و سپس با استفاده از transform داده های آموزشی و تست را تبدیل به داده های با یک فیچر کردیم. سپس این داده ها به همراه لیبل هایشان به یک دسته بند perceptron خطی دادیم با پارامتر تلرانس 10^{-10} . دقت روی این دسته بندی ، برای داده های آموزشی و تست ۱۰۰ درصد است.

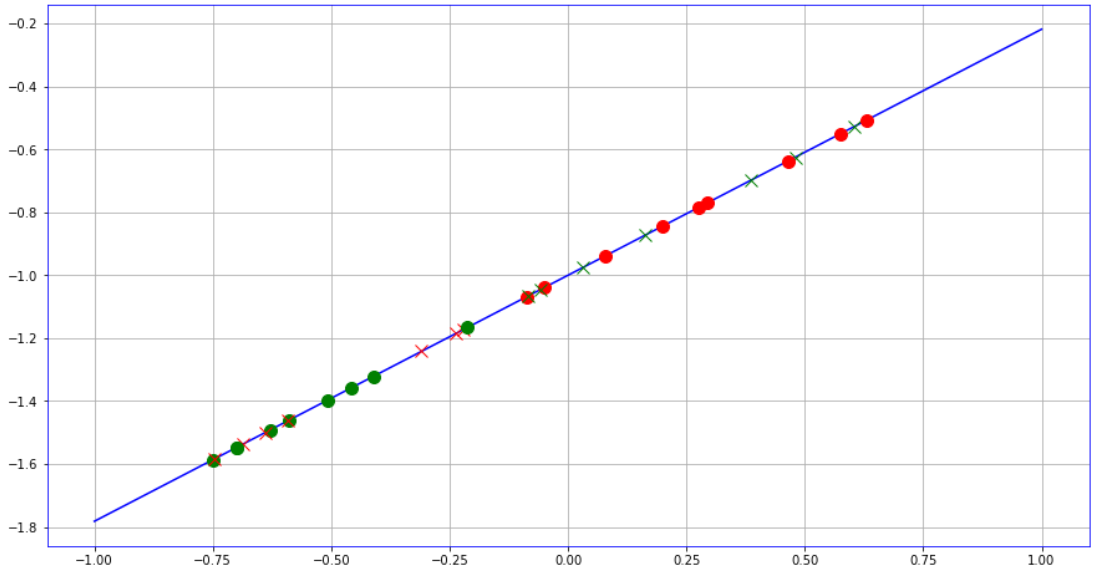
سپس با استفاده از PCA یک فیچر را حذف کرده (PCA را با داده های آموزشی فیت میکنیم سپس داده های آموزشی و تست را transform میکنیم و به حالت یک فیچره تبدیل میکنیم). سپس perceptron را روی این داده ها اعمال میکنیم. دقت روی داده ی آموزشی ۴۵ درصد و روی داده های تست ۳۸ درصد است.

	Accuracy on Training Data	**Accuracy on Test Data**
LDA	100	100
PCA	45	38

چون دقت دسته بندی روی داده های آموزشی و تست LDA ، ۱۰۰ درصد است پس از کشیدن شکل داده های misclassify شده در این حالت صرف نظر میکنیم چون اصلا چیزی نداریم. شکل داده های misclassify شده در حالت PCA برای داده های آموزشی به صورت زیر است:

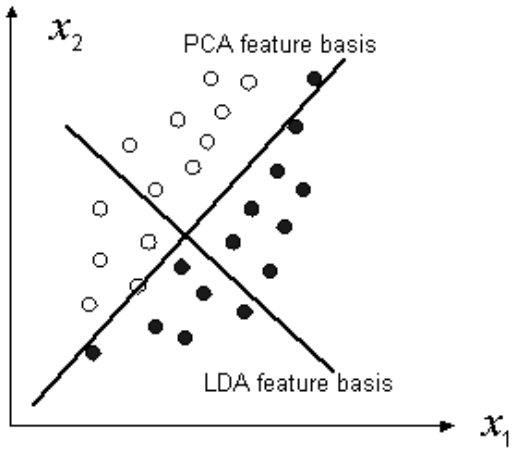


و روی داده های تست:

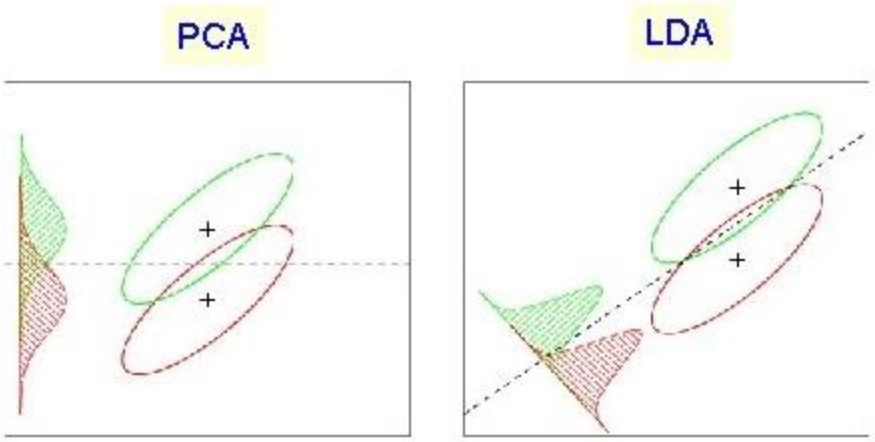


که دایره های سبز داده های misclassify شده از کلاس دوم ، ضربدر های سبز داده های درست دسته بندی شده از کلاس دوم هستند . همینطور دایره های قرمز داده های misclassify شده از کلاس اول و ضربدر های قرمز داده های درست دسته بندی شده از کلاس اول هستند.

میبینیم که عملکرد PCA در کاهش بعد برای مسئله استفاده در دسته بندی داده های چند کلاسه مانند LDA مناسب نیست. LDA سعی کرده کلاس ها را جدا کند یعنی داده های درون کلاسی کمترین واریانس و برون کلاسی بیشترین واریانس یا پراکندگی را داشته باشند. از این نظر LDA برای دسته بندی بسیار مناسب است. اما PCA سعی میکند خطی در جهت بیشترین واریانس یا پراکندگی روی داده ها رسم کند و لزوما این عمل برای دسته بندی مناسب نیست. بنابراین برای عمل دسته بندی بهتر است از LDA استفاده کنیم. همانطور که در شکل زیر میبینیم به وضوح LDA مناسب تر است چون PCA قادر به حفظ تفاوت داده های کلاس اول و دوم نیست:



یا در این شکل تفاوت این دو فشرده ساز برای کاری که ما میخواهیم انجام دهیم بیشتر مشخص خواهد شد:



همانطور که میبینیم توزیع بین کلاسی در PCA جدا نشده اند ولی در LDA این اتفاق رخ داده است.

۴- سوال سوم

از ما خواسته شده است با استفاده از تصاویری که در faces.mat به ما داده شده است ، ابتدا داده ها را به صورت ۷۰ به ۳۰ برای آموزش و تست تقسیم بندی کنیم. سپس با ۴۵ مولفه اصلی و ۷۰ درصد داده آموزشی PCA ای فیت کنیم. سپس RMSE را برای داده های آموزشی و تست بدست آوریم.

ابتدا ۴ عکس به طور رندوم از داده های آموزشی نمایش دادیم:

سپس با استفاده از ۴۵ مولفه اصلی و PCA ، داده های آموزشی را transform کردیم و برای اینکه ببینیم چقدر PCA موثر بوده دوباره به حالت قبل بر میگردانیم و عکس های جدید به صورت زیر هستند:

همینطور که مشاهده میشود، مقدار زیادی از جزئیات عکس از بین رفته است. بنابراین ۴۵ مولفه اصلی ، ۱۰۰ درصد واریانس داده ی ما را شامل نمیشود و تنها بخشی از واریانس داده ها را شامل میشود.

حال ۴ عکس رندوم از داده های تست گرفتیم و نمایش دادیم:

بعد از transform داده های های تست و برگرداندن آن توسط PCA ای که توسط داده های آموزشی فیت شده بود ، چنین تصاویری بازیابی کردیم:

که باز هم مشاهده میشود دقت بسیار بالایی نگرفتیم چون جزئیات زیادی از تصاویر را از دست داده ایم.
در این حالت ارور RMSE روی داده های آموزشی 11.562185547599679 و روی داده ی تست 14.77984864667843 بود.

حال میخواهیم تعداد مناسبی از PC ها را انتخاب کنیم تا عکس باز تولید شده شبیه به عکس اصلی باشد و تقریبا جزئیات یکسانی داشته باشند. میدانیم هر چه PC ها را بیشتر کنیم درصد واریانس پوشش داده شده از داده اصلی بیشتر میشود و این را میدانیم که هر چه PC ها را بیشتر کنیم میزان رشد درصد پوشش واریانس کمتر میشود چون بزرگترین PC ها را اول به ترتیب صعودی به نزولی انتخاب کردیم.

تعداد مولفه های اصلی مختلفی را در نظر گرفتیم اگر ۹۸ درصد واریانس داده ی اصلی پوشش داده شد ، آنگاه آن تعداد مولفه اصلی احتمالا تصاویر بازگردانی شده مشابه با تصاویر اصلی خواهند بود.

با استفاده از برنامه نویسی، میزان مولفه های اصلی مناسب برای دریافت ۹۸ درصد واریانس را برابر ۱۶۲ پیدا کردیم. حال با استفاده از ۱۶۲ مولفه اصلی عکس های آموزشی را با PCA آموزش داده و سپس عکس های آموزشی را `transform` کرده و باز میگردانیم. ۴ عکس آموزشی به صورت رندوم نمایش دادیم:

بعد از اعمال PCA و بازگردانی مجدد عکس ها با عکس های زیر برخورد میکنیم:

که تقریبا تمام جزئیات را در بر میگیرد و با چشم نمیتوان متوجه تفاوت ها شد. پس ۱۶۲ مولفه اصلی میتواند عکس ها را به خوبی فشرده کند. البته میتوانستیم میزان واریانس ۹۸ درصد را کمتر کنیم و شاید در حالت ۹۵ درصد نیز عکس های خوبی را دریافت میکردیم (تست شد و خیلی هم عالی نبود).

خطای RMSE برای داده های آموزشی با استفاده از PCA دارای ۱۶۲ مولفه اصلی برابر `4.5385394478753` که مشخصا کاهش چشمگیری در مقایسه با ۴۵ مولفه اصلی داشته است.

سپس ۴ داده تست به صورت تصادفی نمایش دادیم:

با استفاده از همان PCA با ۱۶۲ مولفه اصلی که با داده های آموزشی فیت کرده بودیم ، عکسهای تست را نیز transform میکنیم و سپس بازگردانی انجام میدهیم. تصاویر بازگردانده شده به صورت زیر هستند:

که مشاهده میشود نتوانستیم خیلی زیاد دقت بالایی بگیری با اینکه جزئیات خوبی را داریم. انتظار داریم خطای RMSE کاهش چشمگیری نکرده باشد. همینطور هم شد. مقدار خطای RMSE روی داده تست برابر 11.495316989498226 است که پیشرفت چشمگیری نداشته است.

	RMSE on Training Data	RMSE on Test Data
45 PC	11.562185547599679	14.77984864667843
162 PC	4.5385394478753	11.495316989498226

که مشاهده میشود خطای RMSE روی داده آموزشی خیلی کم شده است ولی روی داده تست کاهش چشمگیری نداشته است.

۵- سوال چهارم (اختیاری).

فرض کنید $\{x_n\}$ مجموعه ای از مشاهدات (داده ها) باشد بطوری که $n = 1.2.\cdots.N$ و $x_n \in R^D$. میخواهیم با توجه به فرمول بندی واریانس ، تصویر عمود بر x_n روی یک فضا با بعد $M < D$ را پیدا کنیم.

در ساده ترین حالت فرض میکنیم $M = 1$ باشد . بردار $w_1 \in R^D$ را یک بردار در جهت فضای با بعد کمتر تعریف میکنیم. از آنجایی که برای ما جهت مهم است و اندازه ی بردار مهم نیست بردار w_1 را بردار یکه در نظر میگیریم یعنی:

$$\mathbf{w}_1^T \mathbf{w}_1 = 1$$

حال میتوانیم داده های x_n را روی بردار w_1 (در فضای جدید) به صورت زیر تصویر کنیم:

$$\hat{\mathbf{x}}_n = \mathbf{w}_1^T \mathbf{x}_n$$

اگر $\bar{x}$ میانگین داده ها در فضای اصلی باشد ، آنگاه میانگین داده ها در فضای جدید برابر است با :

$$\hat{\bar{\mathbf{x}}} = \mathbf{w}_1^T \bar{\mathbf{x}}$$

ادامه صفحه بعد.

حال میتوانیم واریانس داده های تصویر شده را به صورت زیر بنویسیم:

$$\begin{aligned}
\sigma^2(\hat{\mathbf{x}}) &= \frac{1}{N}\sum_{n=1}^{N}(\hat{\mathbf{x}}_n - \hat{\bar{\mathbf{x}}})^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})^T \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})(\mathbf{x}_n^T\mathbf{w}_1 - \bar{\mathbf{x}}^T\mathbf{w}_1) \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{w}_1^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n^T - \bar{\mathbf{x}}^T)\mathbf{w}_1 \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{w}_1^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\mathbf{w}_1 \\
&= \mathbf{w}_1^T\mathbf{S}\mathbf{w}_1
\end{aligned}$$

لازم به ذکر است ماتریس S ماتریس کواریانس مربوط به داده های اصلی (قبل از تصویر روی فضای جدید) هستند که به صورت زیر میتوان نوشت:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$$

باید واریانس داده ها در فضای جدید را به حداکثر برسانیم. البته طبق فرمول بالا اگر $\|w_1\| \to \infty$ این امر محقق خواهد شد اما هدف ما این نیست و ما نمیخواهیم این اتفاق رخ دهد. در بالا هم گفتیم w_1 بردار یکه است.

چون میخواهیم ماکزیمم سازی با یک قید $\|w_1\| = 1$ را داشته باشیم، پس میدانیم باید از بسط لاگرانژ استفاده کنیم. فرض کنید λ_1 ضریب لاگرانژ باشد. هدف بهینه سازی ما (تابع هزینه ما) به صورت زیر خواهد بود :

$$J(\mathbf{w}_1) = \mathbf{w}_1^T\mathbf{S}\mathbf{w}_1 + \lambda_1(1 - \mathbf{w}_1^T\mathbf{w}_1)$$

که یک پنالتی برای زمانی در نظر میگیریم که w_1 بردار یکه نباشد . حال برای اینکه این را ماکزیمم کنیم باید تلاش کنیم w_1 برداری یکه باشد.

حال از فرمول بالا نسبت به w_1 مشتق میگیریم . داریم:

$$\frac{\partial J(\mathbf{w}_1)}{\partial \mathbf{w}_1} = 2\mathbf{S}\mathbf{w}_1 - 2\lambda_1\mathbf{w}_1 = 0$$

$$\mathbf{S}\mathbf{w}_1 = \lambda_1\mathbf{w}_1$$

که این یعنی w_1 بردار ویژه ماتریس کواریانس S و λ_1 مقدار ویژه متناظر با آن است. با ضرب w_1^T از سمت چپ در این فرمول داریم:

$$\begin{aligned}
&\mathbf{w}_1^T\mathbf{S}\mathbf{w}_1 = \mathbf{w}_1^T\lambda_1\mathbf{w}_1 \\
&\mathbf{w}_1^T\mathbf{S}\mathbf{w}_1 = \lambda_1\mathbf{w}_1^T\mathbf{w}_1 \quad (\lambda_1 \text{ is a scalar.}) \\
&\mathbf{w}_1^T\mathbf{S}\mathbf{w}_1 = \lambda_1
\end{aligned}$$

با توجه به فرمول های بالا ، یعنی راستایی که بزرگترین مقدار واریانس را در فضای با بعد پایینتر دارد مربوط به بردار ویژهی متناظر با مقدار ویژه λ_1 از ماتریس کواریانس است که همان w_1 است.