



(گزارش بخش عملی پروژه)

آشنایی با ماشین لرنینگ

محمدعلی هاشمی فر

99106758

مبین خطیب

99106114



Image1 :

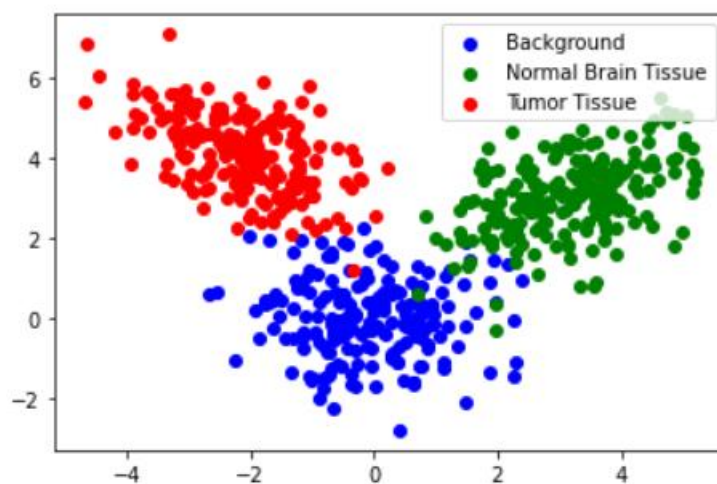
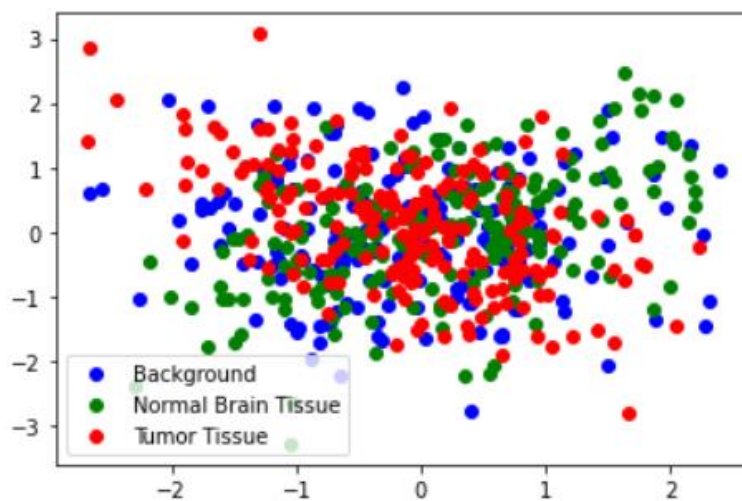


Image2 :



سوال (2)

به تابع e-step خود بایستی 2 ورودی بدهیم :

ورودی 1: 'data' که یک آرایه 2×600 می باشد که درواقع همان نقاط مشخص شده روی صفحه هستند.

ورودی 2: 'means' که یک آرایه 3×2 شامل میانگین هر 3 توزیع نرمال می باشد.

در ابتدا فاصله ی اقلیدسی بین تمام نقطه ها و میانگین هر **distribution** پیدا میکنیم و آن را در **distances** می ریزیم. سپس **index** کمترین فاصله را برای هر نقطه پیدا میکنیم تا آن نقطه را به آن **distribution** اختصاص دهیم.

در نهایت هم ماتریس $R_{3 \times 600}$ را *return* میکنیم.

R برای 50 ستون اول :

Image1 :

[illegible]

Image2 :

[illegible]

سوال 3) در ابتدای امر دو آرایه خالی می سازیم تا مقادیر جدید ماتریس های **mean** و **covariance** را حفظ

کنیم. سپس روی هر کدام از **gaussian distribution** ها **loop** می زنیم تا تعداد داده های اختصاص داده شده به هر کدام از **distribution** ها را بفهمیم. پس از آن مقادیر میانگین و واریانس جدید را محاسبه می کنیم

Image1 :

image2:

Updated Means:

```
[[ -0.00424818  0.0766469 ]
 [  3.2552938  3.11237787]
 [ -2.12150568  4.10651646]]
```

Updated Covariances:

```
[[[ 0.9910141  0.09622623]
 [ 0.09622623  1.01683264]]

 [[ 0.8528404  0.30503126]
 [ 0.30503126  0.79892933]]

 [[ 0.84024096 -0.41831694]
 [-0.41831694  0.90470716]]]
```

Updated Means:

```
[[ -0.01487545 -0.00685517]
 [  1.86129727  2.00036125]
 [ -1.9766022  2.07971658]]
```

Updated Covariances:

```
[[[ 0.89197566  0.00889288]
 [ 0.00889288  0.84081304]]

 [[ 0.03454337 -0.05185701]
 [-0.05185701  0.12077032]]

 [[ 0.26091642  0.04990456]
 [ 0.04990456  0.26473843]]]
```

سوال 4)

در ابتدا ماکزیمم مقدار **iteration** را مشخص می کنیم مثلاً ما آن را 100 گذاشتیم.

سپس **convergence tolerance** را مشخص می کنیم که به نظر مقدار $1e-4$ برای آن مناسب باشد.

سپس در حلقه ای که بر روی ماکزیمم مقدار **iteration** زدیم این مراحل را به ترتیب انجام می دهیم:

1) از **e-step** استفاده می کنیم تا هر کدام از نقاط را با توجه به فاصله ی اقلیدسی آن به یکی از توزیع ها

نسبت دهیم. و سپس ماتریس **R** را بسازیم.

2) از **m-step** استفاده می کنیم تا میانگین و واریانس هر توزیع را **update** کنیم .

3) حال **convergence** را بررسی میکنیم به این معنی که **log-likelihood** داده ها در هر مرحله نسبت به

مرحله قبل کمتر از **tol** تغییرات داشته یا خیر.

4) **Log-likelihood** را آپدیت می کنیم.

Image1 :

converged in step : 2

Final Means:

```
[[ -0.07057305  0.02795606]
 [  3.18586944  3.06155446]
 [ -2.13711914  4.12519435]]
```

Final Covariances:

```
[[[  0.92055847 -0.01603312]
   [-0.01603312  0.95829743]]
```

```
[[  0.93854291  0.36795759]
 [  0.36795759  0.83508943]]
```

```
[[  0.82050513 -0.3950643 ]
 [-0.3950643   0.87780739]]]
```

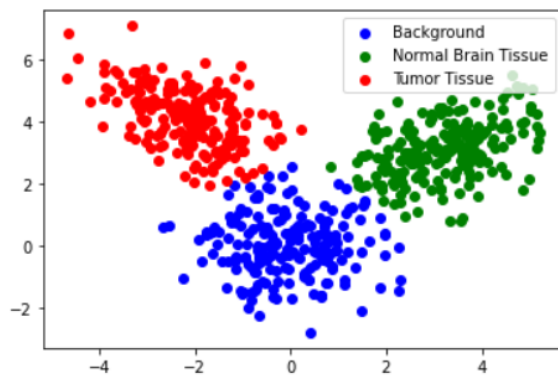


Image2 :

converged in step : 14

Final Means:

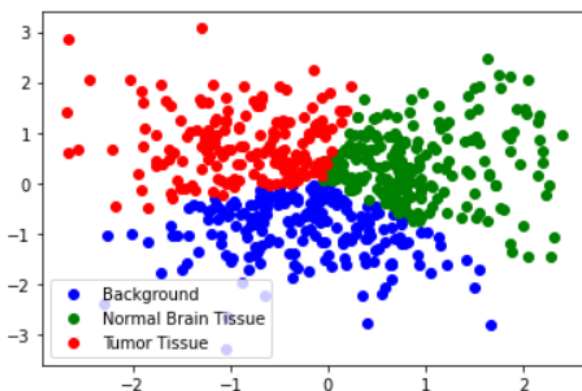
```
[[ -0.26849419 -0.92604895]
 [  0.90646733  0.36807678]
 [-0.84263741  0.75906057]]
```

Final Covariances:

```
[[[  0.53189564 -0.06893279]
   [-0.06893279  0.34537548]]
```

```
[[  0.36150855  0.01552642]
 [  0.01552642  0.49135215]]
```

```
[[  0.41415562 -0.0223189 ]
 [-0.0223189   0.4009015 ]]]
```



سوال 5)

همانطور که مشاهده می کنیم از آنجایی که در image1 داده ها از همان ابتدا تقریباً 3 خوشه جدا از هم دیگر را به ترتیب تشکیل می دادند ، بعد از iteration 2 به شکل نهایی رسیدیم و مقادیر نهایی خیلی تغییری نکرد. اما در image2 پس از اختصاص دادن نقاط به همان ترتیب 200 تایی پشت سرهم به هر کدام از توزیع ها، تفاوت خوشه ها از همدیگر مشخص نبود . از این رو هم iteration بالا تری داریم هم مقادیر نهایی جابجایی بیشتری دارند.