



## تمرین امتیازی ۱: ریاضیات و جبرخطی

### چکیده و موعد تحویل تمرین

- در این تمرین، در بخش اول با بردارها و مقادیر ویژه و همچنین ارتباط آن‌ها با ماتریس‌های مثبت معین آشنا می‌شویم. در ادامه علاوه بر بررسی و تحلیل مفاهیم پایه‌ای جبرخطی و تجزیه مقادیر منفرد، به مباحثی هم‌چون پیاده‌سازی برداری نیز پرداخته خواهد شد.
- موعد تحویل این تمرین، ساعت ۱۸:۰۰ روز جمعه مورخ ۱۴۰۲/۱۲/۱۸ است.
- استفاده از دستیارهای هوشمند (مانند ChatGPT) آزاد است؛ اما حتماً باید برنامه‌ها و جزئیات پروژه‌های تحویلی خود را فهمیده باشید.

۱

ثابت کنید اگر یک ماتریس مثبت معین باشد، تمامی مقادیر ویژه آن نیز مثبت است و بالعکس (راهنمایی: ممکن است نیاز به تجزیه ویژه ماتریس‌ها داشته باشید).

۲

جفت‌های مقدار و بردارهای ویژه را برای ماتریس دوران بدست بیاورید (می‌توانید مسأله را صرفاً برای یک زاویه خاص و ماتریس دوران دو بعدی تحلیل کنید).

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

۳

اگر ماتریس مُدال (که شامل بردارهای ویژه است) یک ماتریس مربعی با ابعاد  $n \times n$  باشد، با استدلال مقدار دترمینان آن را بدست بیاورید.

۴

پی‌شتر با روش کمینه مربعات به‌عنوان راهکاری برای یافتن مجهولات معادلات بیش تعیین شده (over-determined) آشنا شده‌اید با جستجو در اینترنت راهکاری برای حل این نوع معادلات با کمک تجزیه مقادیر منفرد پیشنهاد دهید.

۵

ماتریس‌هایی به فرم زیر را ماتریس‌های بالا مثلثی می‌نامند. آیا چنین ماتریس‌هایی غیرمنفرد اند؟ دلیل خود را بیان کنید و با کمک پایتون استدلال خود را تأیید کنید.

$$A = \begin{pmatrix} p_1 & p_2 & p_3 & \cdots & p_n \\ 0 & p_4 & p_5 & \cdots & p_n \\ 0 & 0 & p_6 & \cdots & p_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p_n \end{pmatrix}$$

۶

اکثر کتابخانه‌های متداول هوش مصنوعی از پیاده‌سازی برداری پشتیبانی می‌کنند، به‌طوری‌که عدم بهره‌گیری از آن‌ها و استفاده مستقیم از حلقه‌های تودرتو، پیاده‌سازی بسیاری از الگوریتم‌های یادگیری ماشین را ناممکن می‌کند. در این بخش به مقایسه زمانی اجرای الگوریتم‌ها توسط حلقه‌های for پرداخته خواهد شد.

در ابتدا با کمک توابع موجود در کتابخانه Numpy الگوریتم زیر را بصورت برداری پیاده‌سازی کنید، سپس با کمک توابع np.allclose() و timeit.timeit() به مقایسه آن‌ها بپردازید.

```
import numpy as np

m, n = 20, 10
A = np.random.rand(m, n)
k = np.random.rand(n)
p = np.zeros(m)

for i in range(n):
    p += A[:, i] * k[i]
```

شکل ۱: کد نوشته‌شده توسط حلقه‌های آشکار