

به نام خدا



## تمرین سری اول

درس مبانی هوش محاسباتی

دکتر ناصر مزینی

دستیاران آموزشی:

سینا اسکندری

محمد میرزایی

مهلت تحویل:

۲۱ مهر ۱۴۰۲

## نکات تکمیلی

۱. پاسخ سوالات را به صورت کامل در یک فایل PDF و به همراه کدهای سوالات در فرمت ipynb. در یک فایل فشرده به شکل HW#\_StudentID.zip قرار داده و تا زمان تعیین شده بارگذاری نمایید.

۲. برای پیاده سازی ها زبان پایتون پیشنهاد می شود، لازم به ذکر است توضیح کدها و نتایج بدست آمده، باید در فایل PDF آورده شوند و به کد بدون گزارش نمره ای تعلق نخواهد گرفت.

۳. در مجموع ۱۴۴ ساعت (۶ شبانه روز) برای هر دانشجو تاخیر مجاز لحاظ شده است و برای هر تمرین ۴۸ ساعت زمان مازاد بر مهلت آن، روی سامانه در نظر گرفته خواهد شد. در صورت تجاوز از تاخیر کل، نمره تمرین را از دست خواهید داد.

۴. لطفا برای انجام تمرین زمان مناسب اختصاص داده شود و انجام آن را به روزهای پایانی موکول نکنید.

۵. لطفا منابع استفاده شده در حل هر سوال را ذکر کنید. در صورت عدم ذکر منابع استفاده شده، نمره سوال را از دست خواهید داد.

۶. تمرین ها باید به صورت انفرادی انجام شوند و حل گروهی تمرین مجاز نیست.

۷. ارزیابی تمرین ها بر اساس صحیح بودن راه حل، گزارش های کامل و دقیق، بهینه بودن کدها و کپی نبودن می باشد.

موفق باشید

**در این تمرین می توانید از کتابخانه های numpy، pandas و matplotlib استفاده کنید و فایل نوت بوکی که در اختیار دارید را تکمیل کنید. توضیحات تکمیلی برای سوالات در نوت بوک موجود است.**

## ✓ سوال ۱

تابعی بنویسید که یک مقایسه عنصر به عنصر (بزرگتر، بزرگتر مساوی، کوچکتر و کوچکتر مساوی) از دو آرایه داده شده (هر دو آرایه با هر اندازه‌ی مساوی) انجام دهد. (از کتابخانه‌ی NumPy برای حل این مسئله استفاده شود)

(۱۰ امتیاز)

## ✓ سوال ۲

تابعی طراحی کنید که دو آرایه و یک پارامتر `method` مشخص شده را دریافت کرده و بر اساس `method` مشخص شده، عملیات ضرب عنصر به عنصر یا ضرب ماتریسی را انجام دهد. (از کتابخانه‌ی NumPy برای حل این مسئله استفاده شود)

(۱۰ امتیاز)

## ✓ سوال ۳

تابعی بنویسید که دو آرایه ورودی با ابعاد  $n \times n$  و یک بردار با ابعاد  $n \times 1$  و همچنین یک ورودی `method` بگیرد و بر اساس `method` مشخص شده بردار دوم را به صورت افقی یا عمودی به بردار اول اضافه کند.

(۱۰ امتیاز)

## ✓ سوال ۴

یک ماتریس  $4 \times 4$  به شکل رندوم با مقادیر بین ۱ تا ۱۰ تشکیل دهید، سپس مقادیر آن را نرمال سازی کنید. (پس از نرمال سازی، مقادیر حداقل ۰ و حداکثر ۱ می‌شوند)

(۵ امتیاز)

## ✓ سوال ۵

ابتدا داده‌های موجود در فایل `data.csv` را بخوانید و سپس:

- میزان بازده روزانه را محاسبه کنید: قیمت پایانی روز قبلی از قیمت پایانی روز جاری کم کرده و بر قیمت پایانی روز قبلی تقسیم کنید

- میانگین بازده روزانه را محاسبه کنید.
  - انحراف معیار بازده روزانه را محاسبه کنید.
  - قیمت‌های پایانی روزانه سهام را به مرور زمان نمایش دهید.
  - میزان بازده روزانه را به مرور زمان نمایش دهید.
  - روزهای با بیشترین و کمترین بازده را شناسایی کنید.
  - تاریخ و مقدار بیشترین و کمترین قیمت‌های تاریخی سهام را پیدا کنید.
- (از دو کتابخانه‌ی NumPy و matplotlib استفاده کنید)

(۱۵ امتیاز)

## سوال ۶

در این سوال با مرحله feed forward در شبکه‌های عصبی آشنا می‌شوید. در این مرحله نمونه بردارهای ورودی  $X$  که  $n$  ویژگی دارد در بردار وزن  $W$  به صورت زیر ضرب می‌شوند.

$$f = \sum_{i=1}^n x_i w_i = \vec{x} \cdot \vec{w}$$

که  $x_i$  نمایانگر هر ویژگی در بردار ورودی و  $w_i$  وزن متناظر هر ویژگی است.

به طور مثال یک ورودی  $[1, 2, 3]$  که بردار وزن  $[1, -1, 1]$  دارد، خروجی feed forward برابر است با:

$$(1 \times 1) + (-1 \times 2) + (1 \times 3)$$

فرض کنید در یک شبکه عصبی ۱۰۰۰ نمونه ورودی داریم و به ازای هر نمونه یکبار باید این عملیات محاسبه شود و ۱۰۰۰ خروجی بدست بیاید.

برای پیاده‌سازی این محاسبات دو روش وجود دارد. در روش اول با استفاده از حلقه for برای هر نمونه این مقادیر ضرب و جمع می‌شوند. روش دوم که به روش vectorization معروف است؛ در این روش تمامی نمونه‌ها در کنار هم در یک ماتریس بزرگتر قرار می‌گیرند و این ماتریس بزرگتر در بردار وزن ضرب داخلی می‌شود. به طور مثال برای ۳ نمونه A و B و C بردار وزن W به صورت زیر می‌باشد.

ماتریس

example 1  $\rightarrow$

$$\begin{matrix} & (m, n) & & (n, 1) & & m, 1) \\ \begin{bmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{bmatrix} \cdot \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix} = \begin{bmatrix} f1 \\ f2 \\ f3 \end{bmatrix} \end{matrix}$$

برای این سوال، در نظر داریم که برای ۱۰۰۰ نمونه که هر نمونه ۵۰۰ ویژگی دارد عملیات **feed forward** را انجام دهیم. ابتدا یک ماتریس رندوم با ابعاد ۱۰۰۰×۵۰۰ و یک بردار وزن با ابعاد ۵۰۰×۱ تولید کنید. با استفاده از حلقه‌های **for** و سپس با استفاده از روش **vectorization** محاسبات را انجام دهید. در آخر این دو روش را از نظر سرعت و مدت زمان اجرا مقایسه کنید.

?

(۲۰ امتیاز)

## سوال ۷

تابعی بنویسید که یک آرایه داده شده و یک مقدار مشخص با نام **threshold** را دریافت کند. این تابع باید تمام عناصر آرایه را بررسی کرده و عناصری که بیشتر از مقدار **threshold** هستند را به یک مقدار دلخواه (مثلاً ۱) و عناصری که کمتر یا مساوی **threshold** هستند را به یک مقدار دلخواه دیگر (مثلاً ۰) تغییر دهد. سپس آرایه جدید را با تغییرات اعمال شده برگرداند. (سوال با استفاده از کتابخانه NumPy حل شود و استفاده از حلقه مجاز نیست.) (۱۰ امتیاز)

## سوال ۸

بدون استفاده از کتابخانه **numpy**، کلاسی برای داده ساختار ماتریس، تعریف کنید. این کلاس باید شامل متدهای زیر باشد:

- متدی برای بررسی کردن مساوی بودن ماتریس با ماتریس دیگر. در صورتی که ماتریس‌ها برابر بودند، خروجی **True** و در غیر این صورت **False** باشد.
- متدی برای بررسی بزرگ‌تر یا کوچک‌تر بودن المان‌های ماتریس. خروجی باید ماتریسی با همان ابعاد ماتریس‌های ورودی باشد و هر المان ماتریس بیان‌گر مقایسه بین المان‌های دو ماتریس باشد. به عنوان مثال اگر در ماتریس اول، در اندیس [۲،۱] مقدار ۵ داشته باشد و در ماتریس دوم مقدار این اندیس برابر ۳ باشد، در ماتریس خروجی باید در این اندیس **True** قرار بگیرد.
- متدی برای بررسی زیر مجموعه بودن دو ماتریس.
- متدی برای محاسبه ضرب ۲ ماتریس (شبیه به عملکرد تابع **numpy.dot**)

(۲۰ امتیاز)