



سامانه‌های یادگیری ماشین توزیع شده

تمرین کامپیوتری ۱

موعد تحویل: ۲ آبان ۱۴۰۱، ساعت ۲۳:۵۵

لطفا پیش از شروع کار بر روی تمرین، به نکات زیر توجه فرمایید .

- حتما ویدئوی راه اندازی کلاستر را به دقت مشاهده کنید و مطمئن شوید به کلاستر درس دسترسی دارید.
- برای راحتی در توسعه و تست کد، از ماشین مجازی لینوکس خود استفاده نمایید تا ترافیک کلاستر (به خصوص در ساعات آخر مهلت تمرین) افزایش نیابد. پس از اطمینان از عملکرد کد، می‌توانید آن را روی کلاستر اجرا کنید.
- توصیه می‌شود از زبان Python به همراه با کتابخانه mpi4py برای کدنویسی استفاده کنید. در آدرس زیر، راهنمای استفاده از این کتابخانه آمده است :

<https://mpi4py.readthedocs.io/en/stable/tutorial.html>

- سوالات خود را در گروه تلگرام درس مطرح نمایید. به هیچ وجه کد یا پاسخ سوالات را در گروه به اشتراک نگذارید.
- میزان تاخیر، تا دو روز مجاز است. تاخیر به صورت ساعتی محاسبه شده و هر روز ۱۰ درصد نمره کم می‌شود. تحویل تمرین پس از دو روز تاخیر امکان پذیر نخواهد بود.

۱. (۵۰ نمره) در این قسمت می‌خواهیم با روش‌های مختلف پیاده سازی ضرب ماتریس آشنا شویم. فایل‌های ورودی، ماتریس‌هایی است که مقدار هر اندیس ماتریس را در یک خط نمایش می‌دهند که در آدرس زیر روی کلاستر در دسترس می‌باشد:

```
/home/shared_files/CA1/A_matrix.txt  
/home/shared_files/CA1/B_matrix.txt  
/home/shared_files/CA1/A_matrix_min.txt  
/home/shared_files/CA1/B_matrix_min.txt
```

خواندن ماتریس به صورت row major می‌باشد و خروجی ضرب ماتریس را به صورت column major همانند فایل ورودی با نام C_matrix.txt ذخیره کنید.

الف) (۵ نمره) ضرب ماتریس را به صورت سریال و با استفاده از سه حلقه for پیاده سازی نمایید. برای این قسمت از ورودی‌های با پسوند min استفاده نمایید.

ب) (۱۰ نمره) ضرب ماتریس را به صورت سریال با استفاده از تابع matmul در کتابخانه numpy پیاده سازی نمایید. (برای هر دو نوع ورودی عملیات ضرب ماتریس محاسبه شود)

ج) (۱۵ نمره) ضرب ماتریس را با استفاده از کتابخانه mpi4py به صورت موازی پیاده سازی نمایید. برای این کار از یک node و ۴ هسته استفاده نمایید. (برای هر دو نوع ورودی عملیات ضرب ماتریس محاسبه شود)

د) (۸ نمره) کد قسمت ج را با استفاده از ۲ node و ۲ هسته بر روی هر node اجرا نمایید.

ه) (۱۲ نمره) زمان اجرای بخش‌های الف، ب، ج و د را با هم مقایسه و تحلیل نمایید.

۲. (۵۰ نمره) در این قسمت می خواهیم مقدار عدد π را با استفاده از روش آماری مونت کارلو محاسبه نماییم. در این روش دو عدد به صورت تصادفی بین مقادیر ۱ و ۱- انتخاب می شود و اگر مجموع مربع دو عدد کوچکتر یا مساوی یک شد مقدار کانتر (counter) را یکی اضافه می کنیم و این کار را به اندازه i بار انجام می دهیم و در نهایت با استفاده از فرمول زیر مقدار عدد π را تخمین می زنیم:

$$\pi = 4 * \text{counter} / i$$

الف) (۵ نمره) کدی سریال بنویسید که مقدار عدد π را بر حسب توضیحات بالا برای $i = 40000000$ محاسبه نمایید. (مقدار عدد π و زمان اجرا را در خروجی نمایش دهید)

ب) (۲۰ نمره) قسمت الف را به صورت موازی با استفاده از کتابخانه mpi4py پیاده سازی نمایید. این کد را بر روی ۴ هسته بروی یک node محاسبه نمایید.

ج) (۵ نمره) قسمت ب را بر روی ۲ node و هر ۲ هسته اجرا نمایید.

د) (۵ نمره) قسمت ب را بر روی ۲ node و هر ۴ هسته اجرا نمایید.

ه) (۱۵ نمره) زمان اجرای بخش های الف، ب، ج و د را با هم مقایسه و تحلیل نمایید.

• دقت کنید برای محاسبه عدد π به جای استفاده از نوع داده float از ماژول decimal استفاده نمایید.

نحوه تحویل پروژه

فایل ها را به صورت زیر نام گذاری کنید و در آخر همه را در یک فایل zip در سامانه ارسال کنید:

۱- گزارش: report.pdf (دقت نمایید فرمت فایل حتما pdf باشد)

۲- نام گذاری کدها براساس جدول زیر انجام دهید:

شماره سوال	بخش	نوع کد	نام فایل
۱	الف	Python source	matrix_multiplication_simple.py C_matrix_simple.txt
	ب	Python source	Matrix_multiplication_numpy.py C_matrix_numpy.txt
	ج	Python source Slurm batch script	matrix_multiplication_parallel_1n4c.py Matrix_multiplication_parallel_1n4c.sh C_matrix_parallel_1n4c.txt
	د	Python source Slurm batch script	matrix_multiplication_parallel_2n2c.py Matrix_multiplication_parallel_2n2c.sh C_matrix_parallel_2n2c.txt
۲	الف	Python source	pi_serial.py
	ب	Python source Slurm batch script	pi_parallel_1n4c.py pi_parallel_1n4c.sh

pi_parallel_2n2c.py pi_parallel_2n2c.sh	Python source Slurm batch script	2	
pi_parallel_2n4c.py pi_parallel_2n4c.sh	Python source Slurm batch script	4	