



- هدف از طراحی و انجام این تمرین، آموزش و ارزیابی شبکه های عصبی به صورت توزیع شده (Multi Node) می باشد، پیشنهاد می شود قبل از انجام تمرین، مبحث درسی مربوطه را یکبار مرور کنید.
- تمرین باید روی کلاستر درس (کلاستر ایجاد شده از برد های رزبری پای) انجام شود و آدرس IP های جدید در زیر نوشته شده است:

raspberrypi-dml0 (172.18.35.204)

raspberrypi-dml1 (172.18.35.205)

raspberrypi-dml2 (172.18.35.206)

raspberrypi-dml3 (172.18.35.203)

- برای پیاده سازی از زبان Python و کتابخانه PyTorch distributed استفاده کنید.
- دقت کنید که منابع کلاستر بین همه ی دانشجویان مشترک است. به همین دلیل سعی کنید بیشتر از مقدار مورد نیاز از آن استفاده نکنید.
- سوالات خود را می توانید از طریق ایمیل، گروه تلگرام درس یا تلگرام مسئول تمرین مطرح کنید. کد یا پاسخ سوالات را در گروه یا با سایر دانشجویان به اشتراک نگذارید.

سوال ۱

با استفاده از مجموعه داده Mnist که لینک استفاده آن به کمک پکیج Torchvision در [اینجا](#) گذاشته شده است، یک شبکه عصبی کانولوشنی با تعداد لایه کانولوشنی دلخواه پیاده سازی کنید. (می توانید از شبکه های آماده که در کتابخانه [Torchvision](#) وجود دارند نیز استفاده کنید) سعی کنید تعداد لایه ها به اندازه ای باشد که شبکه دقت مناسبی پیدا کند. استفاده از لایه های دیگر مانند لایه های Pooling و نیز توابع فعال ساز، تابع بهینه سازی، تابع خطا و سایر هاپیرپارامتر های شبکه دلخواه است.

توجه کنید که برای اجرای برنامه های نوشته شده باید از Slurm استفاده کنید (می توانید یک فایل script نوشته که با اجرای آن پارامتر های مورد نظر به فایل پایتون داده شود و آن را اجرا کند) و script های نوشته شده را در کنار برنامه های پایتون در گزارش پیوست کنید.

الف) ساختار شبکه عصبی کانولوشنی خود را در گزارش قید کرده، شبکه پیاده سازی شده روی

(a) یک ماشین و یک هسته پردازشی

(b) یک ماشین و دو هسته پردازشی

آموزش داده و با استفاده از داده های تست آن را ارزیابی کنید، در نهایت دقت مدل روی داده های آموزشی و تست را گزارش کرده و نمودار خطای داده های آموزش و تست را رسم کنید. زمان اجرا را نیز قید کنید.

ب) یکی از مولفه های Pytorch Distributed مولفه Distributed Data Parallel نام دارد، که اجازه می دهد تا آموزش مدل به صورت توزیع شده را انجام دهیم، هدف این است که داده آموزشی را به چند بخش شکسته و بین Node ها توزیع کنیم، هر یک از Node ها باید مدل یکسانی در ابتدا داشته باشند، شبکه عصبی قسمت الف را به صورت توزیع شده روی

(a) دو ماشین هر کدام با یک هسته پردازشی

(b) دو ماشین هر کدام با دو هسته پردازشی

آموزش داده و موارد خواسته شده در قسمت الف را برای آن گزارش کنید.

ج) نتایج قسمت الف و ب را با هم مقایسه و تفسیر کنید.

سوال ۲

الف) در سوال 1 بخش ب، آموزش مدل به صورت توزیع شده روی چند گره را انجام دادید، در این قسمت می خواهیم همان آزمایش را تکرار کنیم با این تفاوت که از روش های Quantization استفاده و محاسبات را به کمک int8 انجام دهیم، یک روش برای انجام این کار استفاده از [qnnpack](#) می باشد. مدل هایی که روی دو گره آموزش و ارزیابی کرده بودید را یکبار دیگر به کمک روش فوق آموزش و تست کنید، نمودار های خطا را رسم و زمان آموزش را نیز محاسبه و در نهایت با نتایج بخش ب سوال 1 مقایسه کنید.

ب) یکی از متد های torch.distributed متدی بنام init_process_group می باشد که برای communication استفاده می شود، پارامتر [backend](#) این متد می تواند مقادیر مختلفی از جمله MPI, GLOO و غیره دریافت کند، در شبکه های عصبی مربوط به مدل هایی که روی دو ماشین اجرا می شوند، اثر تفاوت استفاده از MPI و GLOO را مقایسه کنید.

نحوه تحویل تمرین

فایل‌هایتان را به صورت زیر نام‌گذاری کنید:

۱. گزارش: report.pdf (دقت کنید که فقط فرمت PDF برای گزارش پذیرفته می‌شود).

۲. نام‌گذاری کدها بر اساس جدول زیر باشد.

شماره سوال	بخش	نوع کد	نام فایل‌های پایتون
۱	الف	Python, script	SMSC.py (for single machine and single core) SMMC.py (for single machine and Multiple cores)
	ب		MMSC.py (for Multiple machines and single core) MMMC.py (for Multiple machines and Multiple cores)
2	الف	Python, script	MMSC_int8.py MMMC_int8.py
3	الف	Python, script	MMSC_MPI.py MMSC_Gloo.py MMMC_MPI.py MMMC_Gloo.py

در انتها، همه‌ی فایل‌ها را در پوشه‌ای قرارداده و نام آن را شماره‌ی دانشجویی خود قرار دهید. پوشه را zip کرده و آنرا در سایت elearn آپلود کنید.