



هدف از طراحی و انجام این تمرین، مروری بر شبکه های عصبی کانولوشنی و آموزش آن به صورت توزیع شده (یک ماشین و چند منبع پردازشی GPU) می باشد، پیشنهاد می شود قبل از انجام تمرین، مبحث درسی مربوطه را یکبار مرور کرده و ویدیوی آموزشی آشنایی با پلتفرم PyTorch Distributed را مشاهده کنید.

## سوال ۱

با استفاده از مجموعه داده Cifar10 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. 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.

**الف)** ساختار شبکه عصبی کانولوشنی خود را در گزارش قید کرده، شبکه پیاده سازی شده روی تنها یک GPU 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. نوع GPU ای که از آن استفاده شده را گزارش کنید و زمان اجرا را نیز قید کنید.

**ب)** یکی از مولفه های Pytorch Distributed مولفه Data Parallel و Distributed Data Parallel نام دارند، این دو اجازه می دهند تا آموزش مدل به صورت توزیع شده را انجام دهیم، هدف این است

که داده آموزشی را به چند بخش شکسته و بین GPU ها توزیع کنیم، هر یک از GPU ها باید مدل یکسانی در ابتدا داشته باشند، شبکه عصبی قسمت الف را به صورت توزیع شده روی دو GPU 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)** یکی از مشکلات شبکه های عصبی که ممکن است با آن مواجه شوند، مساله بیش برازش ([overfitting](#)) می باشند، چگونه می توان متوجه شد که یک شبکه عصبی دچار این مشکل شده است؟ آیا مدل های آموزش داده شده در قسمت های الف و ب سوال 1 دچار این مشکل شده اند؟ چه راه کارهایی برای غلبه بر این مشکل وجود دارد؟

ب) همان طور که می دانید PyTorch به صورت پیش فرض با floating point های 32 بیتی فرآیند آموزش شبکه های عصبی را انجام می دهد، هم چنین روشی بنام [Mixed Precision](#) وجود دارد که بیان می کند نیاز نیست همیشه برای رسیدن به بیشترین دقت از fp32 استفاده کنیم و می توان بخش هایی از فرآیند آموزش را با fp16 انجام داد. مزیت های استفاده از این روش را بیان کرده و دلیل اینکه چرا همواره نمی توان از fp16 استفاده کرد را ذکر کنید، هم چنین شبکه عصبی سوال 1 قسمت ب که روی Multi GPU آموزش داده بودید را به کمک این روش بار دیگر آموزش داده و خروجی های گفته شده را گزارش نمایید و با حالت fp32 مقایسه کنید.

ج) برای افزایش قابلیت تعمیم پذیری شبکه های عصبی (Regularization)، تکنیک های مختلفی وجود دارند، حداقل 5 مورد را به همراه دلیل اثر گذاری شان نام ببرید، و دو مورد را انتخاب کرده و به مدل پیاده سازی شده در سوال 1 قسمت ب، اضافه کنید و اثر آنها را در ارزیابی و نمودار دقت و خطای مدل نشان دهید و تفسیر کنید.

د) 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 `NCCL`, `GLOO` و غیره دریافت کند، در مورد تفاوت مقادیر مختلف بحث کرده و در شبکه عصبی پیاده سازی شده اثر تفاوت استفاده از `NCCL` و `GLOO` را مقایسه کنید.