

سامانه‌های یادگیری ماشین توزیع شده

نمونه امتحان میان ترم

مدت امتحان: ۹۰ دقیقه

۱. (۱۰ نمره) فرض کنید ماتریس مربعی A را از روی دیسک می‌خوانید و می‌خواهید A^2 را محاسبه کنید. ماتریس به صورت زیر روی دیسک ذخیره شده است. آیا پاسخ این محاسبه به فرمت ذخیره‌سازی وابسته است؟ چرا؟ نتیجه یا نتایج محاسبه را ذکر کنید.

[1 2 3 2 4 5 3 5 7]

پاسخ: (۳ نمره ساختن ماتریس، ۵ نمره توضیح اینکه پاسخ تفاوتی نمی‌کند و ۲ نمره نتیجه ضرب) چون ماتریس متقارن است، نمایش row-major و column-major آن یکسان خواهد بود. در نتیجه حاصل A^2 به طور یکتا تعیین می‌گردد.

$$A^2 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 7 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 14 & 25 & 34 \\ 25 & 45 & 61 \\ 34 & 61 & 83 \end{bmatrix}$$

۲. (۱۵ نمره) فرض کنید در روز اول معرفی Apple MacBook Pro موفق شدید لپتاپ جدید اپل را خریداری کنید تا بتوانید پروژه‌ی درس سامانه‌های یادگیری ماشین توزیع شده را روی آن راحت‌تر انجام دهید. وقتی شروع به استفاده از لپتاپ می‌کنید، متوجه می‌شوید لپتاپ به جای پردازنده‌ی اینتل، از پردازنده‌ی M1 ساخت شرکت اپل استفاده می‌کند که هیچ شباهتی (از لحاظ معماری و ساختار داخلی) به پردازنده اینتل ندارد. به همین دلیل امیدی به استفاده از Intel MKL برای بهبود سرعت محاسبات ریاضی ندارید. با مراجعه به سایت اپل متوجه می‌شوید که هنوز این شرکت کتابخانه Accelerate را نیز برای پردازنده M1 خود ارائه نداده است. چه راهکاری برای افزایش سرعت آموزش یا پیش‌بینی مدل‌های یادگیری ماشین به کار می‌بندید؟

پاسخ: از پیاده‌سازی‌های منبع باز BLAS نظیر OpenBLAS می‌توان استفاده کرد و به کمک کامپایلر M1 کد نسبتاً مناسبی برای استفاده روی M1 تولید کرد. دقت کنید که چنین کدی به کیفیت کتابخانه تولید شده

توسط شرکت نمی‌باشد ولی از پیاده‌سازی مجدد و احتمالا نامناسب الگوریتم‌های جبرخطی توسط کاربر/برنامه‌نویس جلوگیری می‌کند.

پاسخ امتیازی (+۵ نمره): کتابخانه‌های منبع باز نظیر **OpenBLAS** شامل تست‌های متعددی هستند که با اجرای آن روی سخت‌افزار، می‌توانند بین چندین پیاده‌سازی، پیاده‌سازی بهتر را روی سخت‌افزار داده شده انتخاب کنند.

۳. (۱۵ نمره) در اسلایدهای درس، برای اجرای برنامه‌ی **MPI** از دستور **mpirun** یا **mpiexec** استفاده شد ولی در تمرین کامپیوتری اول به جای این دستورات، یک فایل **batch** برای **Slurm** نوشتید و از آن طریق به کمک دستور **srun** برنامه‌ی خود را اجرا کردید. چه تفاوتی در این دو روش وجود دارد و چرا در پروژه از **Slurm** استفاده شد؟

پاسخ: برای استفاده از **mpiexec** با **mpirun** می‌بایست آدرس **node**های کلاستر به طور دستی مشخص می‌کردید. به کمک **slurm** مشخصات **node**های کلاستر یکبار مشخص شده و **mpi** از این تنظیمات می‌تواند بهره‌برد. به علاوه **Slurm** اجازه اعمال محدودیت به ازای هر کاربر و مدیریت **workload** های را به طور مرکزی فراهم می‌نماید.

۴. (۱۰ نمره) همانطور که در درس دیدیم، وزن‌های یک مدل یادگیری ماشین را معمولا می‌توان با دقت نسبتاً پایین ۱۶ بیت و یا حتی کمتر نمایش داد بدون اینکه دقت مدل افت چندانی بکند. از طرف دیگر دیدیم که اینتل در طراحی افزونه‌های **SIMD** پردازنده‌های خود، از سال ۱۹۹۷ به بعد، برخلاف این جهت حرکت کرده و طول رجیسترها را از ۶۴ بیت به ۵۱۲ بیت رسانده است. آیا این روند کمکی به یادگیری یا پیش‌بینی مدل‌های یادگیری ماشین می‌کند؟ چرا؟

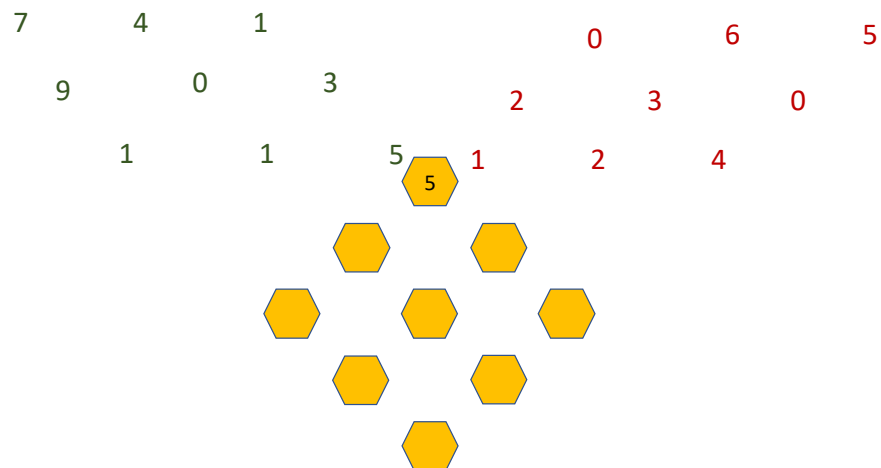
پاسخ: (۳ نمره جواب مثبت، ۷ نمره توضیح) بله کمک می‌کند، زیرا اینتل اجازه کنارهم‌گذاری مقادیر را می‌دهد. به عبارتی، به کمک یک رجیستر ۵۱۲ بیتی، می‌توان ۳۲ مقدار ۱۶ بیتی را ذخیره کرده و روی آن به کمک یک عملگر، عمل محاسباتی انجام داد. این اساس محاسبات **SIMD** می‌باشد.

۵. (۲۰ نمره) فرض کنید دو ماتریس زیر داده شده‌اند. یک آرایه‌ی سیستم‌تولیک مناسب برای ضرب این دو ماتریس طراحی کرده و مراحل ضرب آن‌ها به کمک این آرایه را گام به گام نشان دهید.

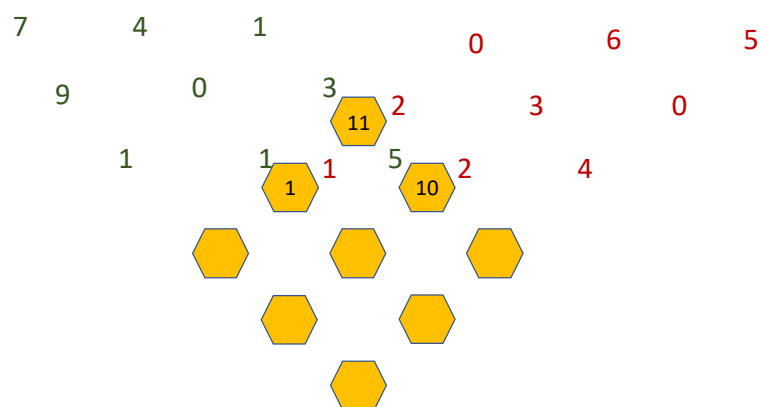
$$A = \begin{bmatrix} 5 & 3 & 1 \\ 1 & 0 & 4 \\ 1 & 9 & 7 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 0 \\ 0 & 6 & 5 \end{bmatrix}$$

پاسخ: (۴ نمره معماری صحیح و ۲ نمره آرایش درست اعداد در ورودی، هر مرحله ۲ نمره) برای جزئیات طراحی به اسلایدها مراجعه کنید.

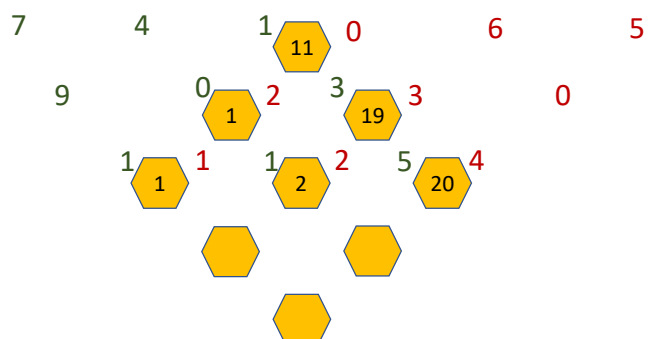
مرحله ۱:



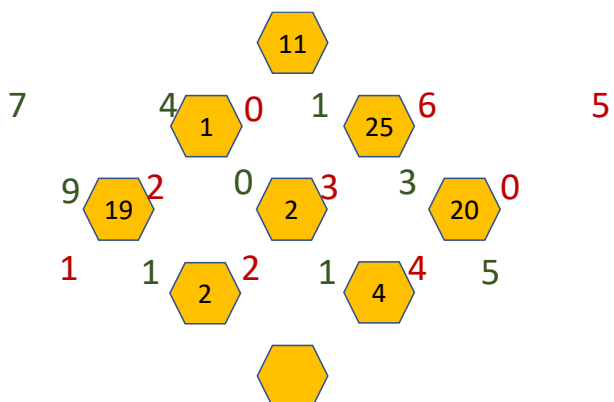
مرحله ۲:



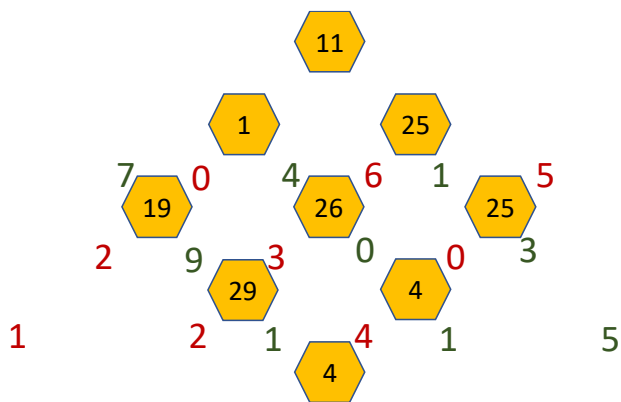
مرحله ۳:



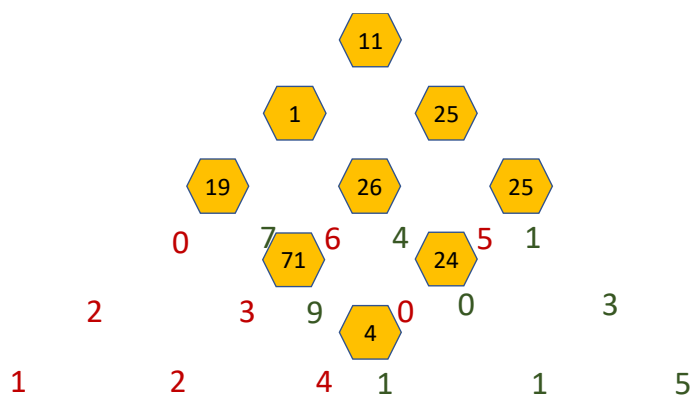
مرحله ۴:



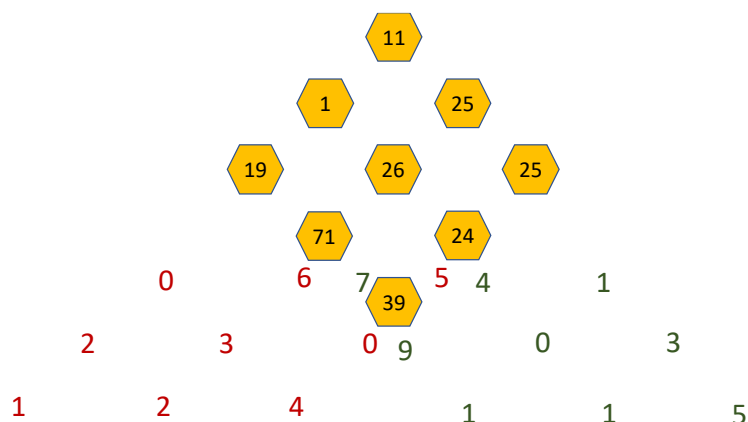
مرحله ۵:



مرحله ۶:



مرحله ۷:



۶. (۱۵ نمره) فرض کنید یک گروه سه نفره برای انجام پروژه درس تشکیل شده است. نفر اول و دوم گروه تنها به کامپیوترهایی با پردازنده‌ی اینتل و نفر سوم به کامپیوترهایی با GPU ی انویدیا دسترسی دارد. نفر اول کار طراحی مدل با PyTorch را انجام می‌دهد. وزن‌های مدل ۳۲ بیتی هستند. نفر دوم برای تسریع مدل، وزن‌ها را به ۱۶ بیت تبدیل می‌کند. نفر سوم، از افت کیفیت راضی نیست و وزن‌ها را به ۳۲ بیت باز می‌گرداند. بهترین کتابخانه‌ی برقراری ارتباط بین node های شبکه برای هر فرد کدام است؟ توضیح دهید.

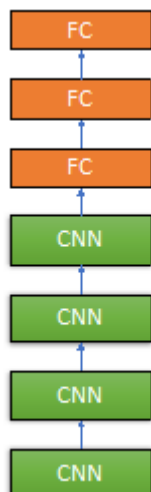
پاسخ: (هر مورد با توضیح صحیح ۵ نمره)

• نفر اول: MPI به خصوص اگر اندازه تنسورها کوچک باشد.

• نفر دوم: Gloo

• نفر سوم: NVIDIA NCCL

برای توضیح، به اسلایدهای horizontal scaling مراجعه کنید.



۷. (۲۵ نمره) فرض کنید مدل مقابل را می‌خواهید به طور توزیع شده روی n کامپیوتر و داده‌هایی به حجم d آموزش دهید. بهترین معماری که برای این کار می‌شناسید توضیح داده و مراحل انتشار رو به جلو و رو به عقب (forward & backward propagation) را در این معماری شرح دهید. FC مخفف fully connected و CNN مخفف convolutional neural network است.

پاسخ: (معماری: ۵ نمره، ۱۰ نمره توضیح forward prop. و ۱۰ نمره توضیح back prop)

معماری استفاده شده: بر اساس مقاله [One weird trick for parallelizing convolutional neural networks](#). بخش CNN به صورت داده-موازی (data parallelism) و بخش FC به صورت مدل-موازی (model parallelism) آموزش داده می‌شوند.

مرحله انتشار رو به جلو: هر کدام از n کامپیوتر، d/n داده را دریافت و محاسبات مربوط به CNN را روی آن داده انجام می‌دهند. سپس هر کامپیوتر، $1/n$ خروجی آخرین لایه را به بقیه کامپیوترها ارسال می‌کند. در ادامه کامپیوترها محاسبات را برای FC با این داده‌ها ادامه داده تا به آخرین لایه برسند و آماده مرحله انتشار رو به عقب شوند. در این لحظه، همه کامپیوترها مجدداً $1/n$ داده‌ی خود را به اشتراک می‌گذارند.

مرحله انتشار رو به عقب: مشابه قسمت قبل، کامپیوترها محاسبات FC را انجام می‌دهند. در نهایت گرادیان هر قسمت از $1/n$ داده به کامپیوتر مربوطه ارسال می‌شود (برعکس مرحله انتشار رو به جلو) این کار تا زمانی ادامه می‌یابد که همه گرادیان‌ها به کامپیوتر مربوط به خود برسند. در همین حین، هر کامپیوتر که بخشی از گرادیان را دریافت کرد، مرحله انتشار رو به عقب را برای CNN انجام می‌دهد.