به نام خدا

سامانه‌های یادگیری ماشین توزیع‌شده (پاییز ۱۴۰۴)

تمرین کامپیوتری ۲

مهلت ارسال: ۰۵ / ۰۹ / ۱۴۰۴

استاد درس: دکتر دوستی

دستیار طراح: علی رمضانی

قوانین و ملاحظات

نحوه ارسال تمرین:

- تمامی فایل‌ها باید در یک فایل فشرده با نام DMLS-CA2-StudentID ارسال شوند.
- کدهای مربوط به هر بخش را بر اساس جدول انتهای همین فایل ذخیره کرده و همراه گزارش ارسال کنید.
- تمامی کدهای ارسال‌شده باید امکان اجرای مجدد داشته باشند. اگر تنظیمات خاصی برای اجرا نیاز است، آن را ذکر کنید.
- کدهای ارسال‌شده باید توسط خودتان اجرا شده باشند و نتایج اجرا در فایل ارسالی مشخص باشد.

رعایت اصول آکادمیک و صداقت علمی:

- این تمرین باید به‌صورت **فردی** انجام شود. هرگونه همکاری یا نوشتن تمرین به‌صورت گروهی ممنوع است.
- در صورت مشاهده تشابه در پاسخ، تمامی افراد درگیر نمره صفر دریافت خواهند کرد و موضوع به استاد گزارش خواهد شد.

استفاده از ابزارهای هوش مصنوعی:

استفاده از ابزارهایی مانند ChatGPT، Gemini، Copilot و موارد مشابه مجاز است، اما تحت شرایط زیر:

- نحوه استفاده از این ابزارها را در گزارش خود توضیح دهید (ابزارهای استفاده‌شده، کاربردهای مشخص و موارد مرتبط).
- تمامی پرامپت‌ها و لینک‌های استفاده‌شده را در انتهای گزارش قرار دهید.
- عدم ارائه این اطلاعات به منزلهٔ سرقت علمی محسوب شده و منجر به نمره صفر خواهد شد.

مهلت ارسال و جریمه تأخیر:

- امکان ارسال تمرین با تأخیر تا ۲ روز وجود دارد. برای هر روز تأخیر ۱۰ درصد جریمه اعمال می‌شود.
- تأخیر به‌صورت ساعتی محاسبه شده و پس از دو روز تأخیر، تمرین پذیرفته نخواهد شد.

ارزیابی حضوری:

- ارزیابی تمرین به‌صورت حضوری انجام خواهد شد.
- محل ارزیابی: آزمایشگاه NLP، طبقه منفی یک، دانشکده مهندسی برق و کامپیوتر ساختمان شماره ۲.

توضیحات تمرین

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

- برای سؤال اول می‌توانید از Google Colab یا Kaggle نیز استفاده کنید و نیازی به استفاده از سرور ندارید.
- برای سؤال دوم حتماً باید از سرور سبلان استفاده کنید.
- ویدئوهای CUDA Programming و PyTorch DDP را قبل از انجام تمرین مشاهده نمایید.
- برای راحتی در توسعه و تست کد، از ماشین مجازی لینوکس خود استفاده نمایید تا ترافیک کلاستر (به‌خصوص در ساعات آخر مهلت تمرین) افزایش نیابد. پس از اطمینان از عملکرد کد، می‌توانید آن را روی کلاستر اجرا کنید.
- سؤالات خود را در گروه تلگرام درس مطرح نمایید. به هیچ وجه کد یا پاسخ سؤالات را در گروه به اشتراک نگذارید.
- هدف از انجام این تمرین، آشنایی بیشتر با PyTorch DDP و CUDA Programming و بررسی پارامترهای مختلف بر تغییر سرعت و حافظه‌ی مصرفی در آموزش مدل‌های شبکه‌ی عصبی است.
- در سؤال اول این تمرین با دیتاست MNIST و در سؤال دوم با دیتاست STL-10 کار می‌کنیم. دیتاست دوم در محل /storage/dmls/stl10_data در سرور ذخیره شده است.
- برای در دسترس بودن کتابخانه‌های لازم، لازم است ابتدا محیط مجازی زیر را فعال کنید:

```
source /home/dmls/pytorch/bin/activate
```

- هر جایی که از شما خواسته شده، باید کدها را در گزارش نیز توضیح دهید.
- می‌توانید از طریق ایمیل زیر در ارتباط باشید:

```
ali.ramezani.96@ut.ac.ir
```

- برای انجام این تمرین به سرور زیر متصل شوید:

```
Sabalan 1:  172.18.32.194
```

سؤال ۱. آشنایی با Kernel CUDA و پیاده‌سازی Kernel Fused ReLU + Convolution

۵۰ نمره

در این تمرین قرار است با مفاهیم پایه و کاربردی CUDA Kernel و نقش آن در تسریع عملیات یادگیری عمیق آشنا شوید. هدف، نوشتن یک کرنل Fused برای ترکیب دو عملیات Convolution و ReLU است، ثبت این کرنل به‌عنوان یک لایه در PyTorch، و بررسی عملی سرعت و دقت آن نسبت به پیاده‌سازی‌های استاندارد است.

اهداف این تمرین:

- توانایی نوشتن و درک عملکرد CUDA Kernel اختصاصی
- امکان استفاده از این کرنل‌ها در شبکه‌های عصبی سفارشی
- بنچمارک و تحلیل سرعت و کارایی کرنل اختصاصی در مقایسه با PyTorch و cuDNN

- تشخیص اینکه چه زمانی پیاده‌سازی کرنل اختصاصی منطقی و مؤثر است

شما برای این تمرین یک ویدئوی آموزشی و یک نوت‌بوک راهنما دریافت کرده‌اید؛ بسیاری از بخش‌های کدنویسی تکمیل شده و تنها بخش‌هایی کلیدی را باید کامل کنید تا بتوانید روی درک و تحلیل تمرکز بیشتری داشته باشید.

۱.۱ مفاهیم اولیه (۵ نمره)

۱. CUDA Kernel چیست و چه کاری انجام می‌دهد؟

۲. cuDNN چیست؟ چرا PyTorch و سایر فریم‌ورک‌ها از این کتابخانه استفاده می‌کنند؟ مزیت‌های cuDNN چیست و چه مشکلی را حل می‌کند؟ نسبت cuDNN و cuBLAS چطور است؟

۲.۱ پیاده‌سازی کرنل Fused Convolution + ReLU (۱۵ نمره)

۱. یک کرنل پایتونی بنویسید که دو عملیات Convolution و ReLU را همزمان برای یک تصویر تک‌کاناله انجام دهد. (کافیست توابع conv_relu_kernel_py و conv_relu_py را کامل کنید).

ویژگی‌های کانوولوشن مد نظر:

```
C_in = 1, C_out = 1
kernel_size = 3x3
padding = 1, stride = 1
```

```
x:  [N, 1, H, W]
w:  [1, 1, 3, 3]
b:  [1]
out:  [N, 1, H, W]
```

۲. کرنل بالا را به زبان CUDA/C++ بنویسید. قسمت‌های خالی رشته‌ی cuda_src را کامل کنید. دقت کنید ایندکس‌دهی، Padding و سایر جزئیات باید دقیقاً مطابق نسخه‌ی پایتونی باشد.

۳.۱ ثبت کرنل جدید به‌عنوان یک لایه در پایتورچ (۵ نمره)

در این بخش ما این کرنل را به‌عنوان یک لایه جدید در پایتورچ ثبت کرده‌ایم. شما کافی است تابع کلاس FusedConvReLUFn را کامل توضیح دهید.

۴.۱ بنچمارک کردن لایه جدید ایجاد شده (۵ نمره)

با استفاده از داده‌ی MNIST، دو شبکه کانولوشنی با ۲ و ۱۰ لایه convolution تعریف کنید و در هر کدام سناریوهای زیر را تست کنید:

۱. شبکه با لایه اصلی روی GPU

۲. شبکه با لایه جدید (کرنل Fused) روی GPU

۳. شبکه با لایه اصلی روی GPU با غیرفعال بودن cuDNN

۴. شبکه با لایه اصلی روی CPU

در هر سناریو، زمان اجرای ۵ ایپاک (Epoch) را برای کل فرآیند، همچنین فقط بخش forward و فقط بخش backward به‌طور جداگانه محاسبه و گزارش کنید. دقت نهایی شبکه (Test Accuracy) را نیز برای هر مورد بنویسید.

۵.۱ تحلیل و بررسی (۲۰ نمره)

۱. نکات جالب توجه، اختلاف‌های زمانی و دقت را توضیح دهید.

۲. در چه مواقعی پیاده‌سازی کرنل اختصاصی (Custom Kernel) می‌تواند مفید باشد و چه موقع بهتر است به پیاده‌سازی‌های داخلی PyTorch و cuDNN اعتماد کنیم؟

سؤال ۲. بررسی تغییر سرعت آموزش در شرایط استفاده از چند GPU — ۳۰ نمره

در این سؤال، قصد بررسی تغییر سرعت آموزش در شرایط استفاده از چند GPU را داریم.

۱.۲ طراحی و آموزش روی یک GPU (۱۰ نمره)

یک شبکه‌ی کانولوشنال طراحی کنید که بتواند به دقت بالای ۵۰ درصد برای طبقه‌بندی داده‌های تست برسد. این شبکه را بر روی یک GPU آموزش دهید. از بهینه‌ساز Adam و اندازه‌ی Batch برابر با ۳۲ استفاده نمایید.

۲.۲ پیاده‌سازی موازی روی دو GPU (۱۰ نمره)

کد بخش قبل را به گونه‌ای تغییر دهید که بتوان آن را با کمک دستور python و با استفاده از PyTorch Multiprocessing بر روی دو GPU اجرا کرد.

۳.۲ مقایسه و تحلیل (۱۰ نمره)

در هر آزمایش (تک GPU و دو GPU)، دقت مدل بر روی داده‌های تست، زمان اجرای آموزش و میزان حافظه‌ی مصرفی GPU را به دست آورده، با یکدیگر مقایسه کرده و تحلیل نمایید.

سؤال ۳. بررسی تأثیر تغییر اندازه‌ی Batch — ۱۰ نمره

در این سؤال، قصد بررسی تأثیر تغییر اندازه‌ی Batch در سرعت آموزش را داریم. بخش دوم سؤال دوم (آموزش بر روی دو GPU) را با اندازه‌های Batch برابر با ۱۶، ۳۲، ۶۴ و ۱۲۸ اجرا نمایید. در هر آزمایش، زمان اجرای آموزش، حافظه‌ی مصرفی GPU و دقت نهایی مدل را به دست آورده، جدول تغییر زمان آموزش، دقت مدل و حافظه‌ی مصرفی GPU را بر حسب تغییر اندازه‌ی Batch رسم نمایید و نتایج را تحلیل کنید.

سؤال ۴. بررسی های Backend مختلف communication در DDP PyTorch — ۱۰ نمره

در این سؤال، قصد بررسی استفاده از Backendهای مختلف communication در PyTorch DDP (Distributed Data Parallel) را داریم. سؤال را با بهینه‌ساز Adam و با اندازه‌های Batch برابر با ۳۲ و ۱۲۸ و با استفاده از backendهای nccl و gloo اجرا کنید و نتایج را از نظر حافظه‌ی مصرفی GPU و زمان آموزش مقایسه نمایید.

نحوه تحویل پروژه

۱. گزارش پروژه را در قالب فایل PDF تهیه کنید.

۲. تمام فایل‌های مربوط به تمرین را طبق ساختار جدول زیر در پوشه‌های مشخص‌شده قرار داده و در انتها با فرمت zip در سامانه elearn بارگذاری کنید.

سؤال	بخش	نام فایل
۱	بخش ۲	CUDA.ipynb
۲	بخش ۱	Classifier.py
	بخش ۲	Classifier_mp.py

توزیع نمرات

سؤال	بخش	نمره	توضیح
۱ (۵۰ نمره)	۱.۱ (بخش الف)	۲	تعریف و توضیح CUDA Kernel
	۲.۱ (بخش ب)	۳	توضیح cuDNN، مزایا و ارتباط با cuBLAS
	۳.۱ (بخش ج)	۱۵	پیاده‌سازی کرنل Fused Convolution + ReLU در پایتون و CUDA
	۴.۱ (بخش د)	۵	توضیح دقیق کلاس FusedConvReLUFn و تعامل با autograd
	۵.۱ (بخش ه)	۵	پیاده‌سازی و اجرای آزمایش‌های MNIST در سناریوهای مختلف
	۶.۱ (بخش ی)	۲۰	تحلیل اختلاف‌های زمانی و دقت، نمودارها و بحث در مورد کرنل اختصاصی
۲ (۳۰ نمره)	۱.۲ (بخش الف)	۱۰	طراحی شبکه و آموزش روی تک GPU با دقت بالاتر از ۵۰٪
	۲.۲ (بخش ب)	۱۰	پیاده‌سازی آموزش توزیع‌شده روی دو GPU با DDP
	۳.۲ (بخش ج)	۱۰	مقایسه‌ی سرعت، دقت و حافظه‌ی مصرفی میان تک GPU و دو GPU
۳ (۱۰ نمره)	سؤال سوم	۱۰	اجرای آزمایش‌های Batch Size مختلف، رسم جدول/نمودار و تحلیل نتایج
۴ (۱۰ نمره)	سؤال چهارم	۱۰	مقایسه‌ی Backendهای nccl و gloo از نظر زمان و حافظه و تحلیل نهایی