بسمه تعالی

درس سیستم‌های نهفته مبتنی بر هسته

تکلیف کامپیوتری ۲: آشنایی با برنامه نویسی CUDA و اجرای آن بر روی سرور Google Colab

پردیس دانشکده‌های فنی دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
دکتر احمد شعبانی
نیم‌سال دوم سال تحصیلی ۱۴۰۲-۱۴۰۱
نگارش: امیرمهدی جودی - نگین سفاری

مقدمه:

هدف از انجام این تمرین کامپیوتری آشنایی با برنامه نویسی CUDA و اجرای آن بر روی سرور Google Colab است. بدین منظور در گام نخست تمرین ابزار Google Colaboratory را آماده نمایید و سپس در گام های بعدی مطابق با خواسته های صورت تمرین به بررسی موارد خواسته شده بپردازید.

گام اول:

یکی از ساده ترین روش های اجرای کدهای *CUDA* استفاده از سرور های *Google Colab* است. به منظور استفاده از *Google Colab* وارد *Google Drive* خود شوید. از تب *New* → *more* → *connect more apps* برنامه Google Colaboratory را جستجو کرده و به منوی برنامه هایتان اضافه کنید. سپس برنامه را باز کرده و مراحل زیر را طی کنید (یک فایل ipynb نیز در اختیار شما قرار داده شده است که دستورات زیر را شامل می‌شود):

۱. در ابتدا به آدرس زیر رفته و *GPU* را به برنامه خود اضافه کنید و سپس *Connect* را بزنید.

Runtime → Change runtime type → Hardware accelerator

۲. دستورات زیر را برای نصب nvcc اجرا کنید (اجرای این دستورات ممکن است کمی طولانی باشد!):

```
!apt-get --purge remove cuda nvidia* libnvidia-*

!dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 dpkg --purge

!apt-get remove cuda-*

!apt autoremove

!apt-get update

!wget https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 -O cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb

!dpkg -i cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb

!apt-key add /var/cuda-repo-9-2-local/7fa2af80.pub
```

```
!apt-get update
!apt-get install cuda-9.2
```

۳. به کمک دستور زیر، افزونه مورد نیاز برای نوشتن کدهای *CUDA* را نصب کنید:

```
!pip install git+https://github.com/andreinechaev/nvcc4jupyter.git
```

۴. ورژن nvcc را بررسی کنید تا از نصب کامل برنامه مطمئن شوید:

```
!nvcc --version
```

۵.محیط nvcc را لود کنید:

```
%load_ext nvcc_plugin
```

۶. در ابتدای کدهایی که میخواهید در محیط nvcc اجرا شوند، از %%cu استفاده کنید.

۷. برای پروفایل گیری از فایل های CUDA در ابتدای کد از %%writefile filename.cu استفاده کنید. سپس با دستور !nvcc file_address -o filename -Wno-deprecated-gpu-targets آن را کامپایل کنید. سپس دستور زیر را اجرا کنید.

```
!nvprof ./filename
```

۸. برای گرفتن زمان اجرا هم می‌توانید از دستور زیر استفاده کنید:

```
!time ./filename
```

الگوریتم کانولوشن:

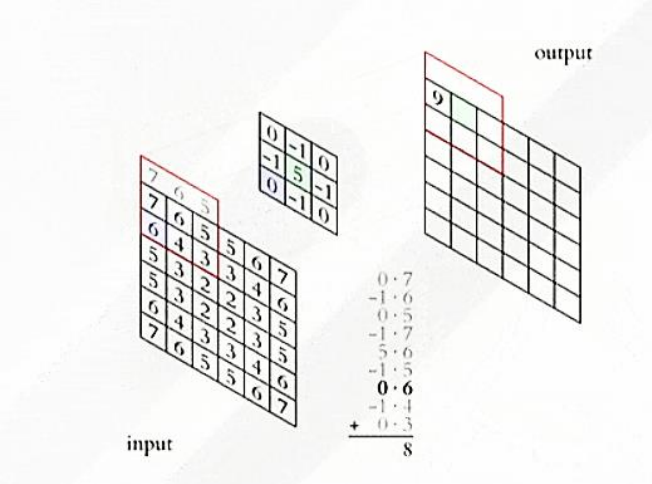
کانولوشن یک عملیات اساسی در یادگیری عمیق، به ویژه در زمینه بینایی کامپیوتر است. برای استخراج ویژگی ها از تصاویر و سایر داده ها با اعمال فیلتر یا هسته به داده های ورودی استفاده می شود. عملیات کانولوشن معمولاً در شبکه‌های عصبی کانولوشن (CNN) استفاده می‌شود که نوعی شبکه عصبی طراحی شده برای پردازش تصویر و ویدئو هستند.

نقش کانولوشن در یادگیری عمیق اسکن تصویر ورودی با یک ماتریس کوچک به نام فیلتر یا کرنل است که به عنوان فیلتر کانولوشن نیز شناخته می شود. این فیلتر برای تشخیص ویژگی های خاص در تصویر ورودی مانند لبه ها، شکل ها و بافت ها استفاده می شود. فیلتر در کل تصویر جابجا می شود و در هر موقعیت، فیلتر بر حسب عنصر با بخش مربوط به تصویر ضرب می شود. سپس نتیجه این ضرب ها برای تولید یک مقدار خروجی خلاصه می شود که به آن نقشه ویژگی می گویند.

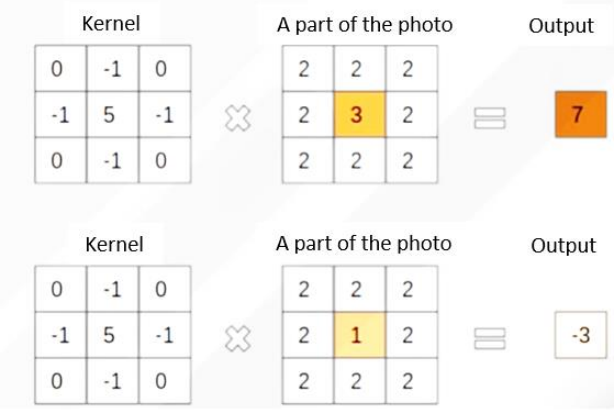
در فرایند کانولوشن از کرنل برای استخراج ویژگی‌های عکس استفاده می‌شود. کرنل ماتریسی است که مانند پنجره‌ای روی عکس ورودی حرکت می‌کند. در هر بار حرکتش مقدار این ماتریس با ماتریس ورودی ضرب می‌شود

تا درنهایت خروجی مدنظر را داشته باشیم. با این کار می‌توانیم اطلاعات یک درایه را با درنظر گرفتن تاثیر همسایه‌های آن، ذخیره کنیم.

کرنل‌ها انواع مختلفی دارند و هر کرنل برای استخراج ویژگی‌های متفاوتی طراحی شده است. در شکل زیر کرنلی برای شارپ‌کردن (Sharpening) عکس استفاده شده است. همان‌طور که می‌بینیم، کرنل روی هر قسمت از ماتریس عکس ورودی که قرار می‌گیرد مقدار هر خانه‌ی آن با مقدار متناظر در ماتریس ورودی ضرب می‌شود و درنهایت همه‌ی این مقادیر با هم جمع می‌شوند و ماتریس خروجی را گام‌به‌گام ایجاد می‌کنند.



مثال دیگری را در شکل بعدی می‌بینیم که زمانی‌که در عکس اول مقادیر کرنل در مقادیر عکس ضرب می‌شوند و درنهایت با هم جمع می‌شوند مقدار خروجی برابر با ۲×۲+۰×۲-۲×۱+۲×۲+۳×۱+۲×۵+۲×۱+۲×۲+۱×۰+۲×۰-۲×۱+۰=۷ می‌شود؛ یعنی درواقع مقدار ۳ که در مرکز عکس قرار دارد به ۷ افزایش یافته است؛ هم‌چنین در عکس دوم خروجی برابر با ۳- شده است؛ یعنی مقدار ۱ که مرکز قرار دارد به ۳- کاهش یافته است. این یعنی تضاد میان ۱ و ۳ به ۷ و ۳- تبدیل شده است و این یعنی کنتراست (Contrast) عکس افزایش می‌یابد و شارپ می‌شود.



در این تمرین کامپیوتری، تابعی بنویسید که فیلتر دو بعدی با ابعاد ۳ در ۳ موجود در تصویر قبل را روی یک تصویر دو بعدی به ابعاد ۵۱۲ در ۵۱۲ حرکت دهد و در هر مرحله به صورت یک به یک مقادیر آن را در هم ضرب و با هم جمع کند و در نهایت مقادیر مربوط به تصویر نهایی را در خروجی قرار دهد. برای این کار می توانید از هر تصویر دلخواهی استفاده کنید یا مقادیر هر پیکسل تصویر ورودی را به صورت رندوم تولید کنید.

خواسته های مسئله:

گام اول: تابع بالا را به صورت سریال اجرا کنید و زمان اجرا تابع را مشخص کنید.

گام دوم: تابع را با *CUDA* نوشته و زمان اجرا را به دست آورید.

گام سوم: برای تعداد ترد ۱۰۲۴، ۵۱۲، ۲۵۶، ۱۲۸و ۳۲ در هربلاک تعداد Grid را محاسبه کرده و زمان اجرای آن ها را به دست آورید.

گام چهارم: در مورد تعداد تردها و تاثیر آن ها روی روند اجرا توضیح دهید و خروجی قسمت های قبل را با هم مقایسه کنید.

گام پنجم: پروفایل برنامه خود را بگیرید و درباره‌ی زمان اجرای هر بخش ار کد توضیح دهید. چه کارهایی می‌توان کرد که این زمان‌ها بهبود یابند؟

نکات قابل توجه:

* انجام تمرین به صورت **فردی** می باشد.
* نمره دهی در این تمرین بر اساس گزارش کار شما صورت می گیرد، بنابراین لطفا گزارش کار خود را به صورت کامل و دقیق نگارش نمایید. (در گزارش کار خود تمامی مراحل انجام تمرین را با قرار دادن تصاویر مناسب و توضیحات کافی شرح دهید.)
* فایل کدهای خود را همراه گزارش قرار دهید.

نمره دهی هر بخش:

گام اول: ۲۰ درصد

گام دوم: ۲۰ درصد

گام سوم: ۲۰ درصد

گام چهارم: ۲۰ درصد

گام پنجم: ۲۰ درصد

تاریخ تحویل: ساعت ۲۳:۵۹ روز ۱۴۰۲/۲/۲۷

موفق و سلامت باشید