



طراحی کامپیوتری سیستم‌های دیجیتال

دانشکده مهندسی برق و کامپیوتر

پروژه سوم: شبکه عصبی کانولوشنی

دستیاران آموزشی:

محبوبه فخریه، مهدی محمدی نسب، پرمیدا ضرغامی، مونا مجتهدی

توضیحات پروژه

شبکه‌های عصبی کانولوشنی^۱ یا به اختصار CNN، یکی از محبوب‌ترین شبکه‌های عصبی عمیق هستند که برای پردازش تصویر و شناسایی الگوها استفاده می‌شوند. این شبکه‌ها ابتدا تصویر ورودی را دریافت می‌کنند، سپس با استفاده از لایه‌های مختلف کانولوشن، اطلاعات مهم تصویر را استخراج و تحلیل می‌کنند. در یک شبکه کانولوشنی، ورودی تصویر از طریق لایه کانولوشن به چندین قسمت تقسیم شده و هر قسمت با استفاده از یک فیلتر یا کرنل^۲ به یک نقطه در خروجی تبدیل می‌شود. سپس با استفاده از لایه فشرده‌سازی^۳، اندازه تصویر کاهش داده می‌شود. این فرایند تکرار می‌شود تا اطلاعات مهم تصویر به صورت خلاصه شده استخراج شود. در نهایت، با استفاده از لایه تمام متصل^۴، ویژگی‌های استخراج شده از تصویر به یک یا چند خروجی نهایی متصل می‌شوند که ممکن است برای شناسایی الگوهای مختلف در تصویر مورد استفاده قرار گیرند.

Convolutional Neural Networks^۱

Kernel^۲

Pooling^۳

Fully Connected^۴

به منظور پیاده‌سازی یک شبکه کانولوشنی، این تمرین در دو فاز انجام می‌شود:

۱- فاز اول: پیاده‌سازی یک لایه کانولوشنی که این بخش را انجام دادید.

۲- فاز دوم: اتصال چند لایه کانولوشنی

در این بخش دو لایه کانولوشنی را به هم متصل خواهید کرد. این کار را مطابق شکل ۱ انجام دهید. در فاز قبل Layer1 را پیاده‌سازی کردید (ورودی $16 \times 16 \times 1$) که برای هر PE یک حافظه جداگانه برای نوشتن خروجی (OFM) در آن، در نظر گرفتید. درواقع از این لایه در نهایت یک خروجی $13 \times 13 \times 4$ (هر واحد عملیاتی یک 13×13 تولید کرده است) ایجاد می‌شود که ورودی لایه بعد خواهد بود. در این فاز، Layer 2 را پیاده‌سازی خواهید کرد. به نحوی که:

۱- این لایه نیز شامل ۴ فیلتر (متفاوت از لایه قبل) بوده که باید بین PE ها پخش شوند. فیلترها در این لایه $4 \times 4 \times 4$ هستند؛

زیرا بعد سوم آن باید مطابق با تعداد OFM های لایه‌ی قبل باشد.

۲- برای سادگی پیاده‌سازی، اجرای لایه دوم را پس از اتمام کامل لایه اول شروع کنید.

۳- حافظه‌های خروجی لایه قبل، به عنوان حافظه ورودی لایه بعد استفاده خواهند شد (بنابراین آدرس شروع ورودی‌ها برای

لایه‌ی دوم Z خواهد بود). هر حافظه خارجی که با outmem1 تا outmem4 در شکل ۱ مشخص شده‌اند، شامل فیلتر

نظیر خود و یک OFM از لایه قبل است؛ بنابراین هر حافظه خروجی را در ابتدای اجرا به نحوی مقداردی‌های اولیه کنید که

فیلتر نظیر آن از آدرس M شروع شود.

۴- توجه کنید که تمامی OFM های تولید شده در لایه‌ی اول باید به عنوان ورودی PE های لایه‌ی دوم در نظر گرفته شوند.

پیشنهاد می‌شود که برای سادگی پیاده‌سازی، ورودی هر PE آرایه‌ای از اندازه فیلتر باشد تا OFM هایی که از حافظه‌های

مختلف به PE وارد می‌شوند هر یک در ایندکس نظیر خود قرار گیرند.

۵- اندازه بافر وزن و بافر ورودی داخل هر PE برابر سایز کرنل آن لایه است. یعنی برای لایه اول، با توجه به اینکه سایز فیلتر

4×4 بود، بافر داخلی PE برای وزن‌ها و ورودی به اندازه‌ی 4×4 در نظر گرفته شده بود. بنابراین برای لایه دوم که

سایز فیلتر $4 \times 4 \times 4$ است، سایز بافرهای داخلی PE نیز باید به همین اندازه در نظر گرفته شود. جهت تداخل نداشتن با

PE های لایه اول، پیشنهاد می‌شود که اندازه این بافرها را به صورت پارامتری در PE خود تنظیم کنید.

۶- وجود حافظه میانی 8×8 در این فاز اجباری نیست.

۷- خروجی‌های لایه دوم را نیز مانند لایه اول، در چهار حافظه خروجی جدا برای هر PE ذخیره کنید. هر خروجی $4 \times 10 \times 10$

است.

نکات درباره فایل‌های داده‌ها:

۱- در همه فایل‌ها، هر عدد ۸ بیتی بوده و هر ۴ عدد در یک خط قرار می‌گیرند.

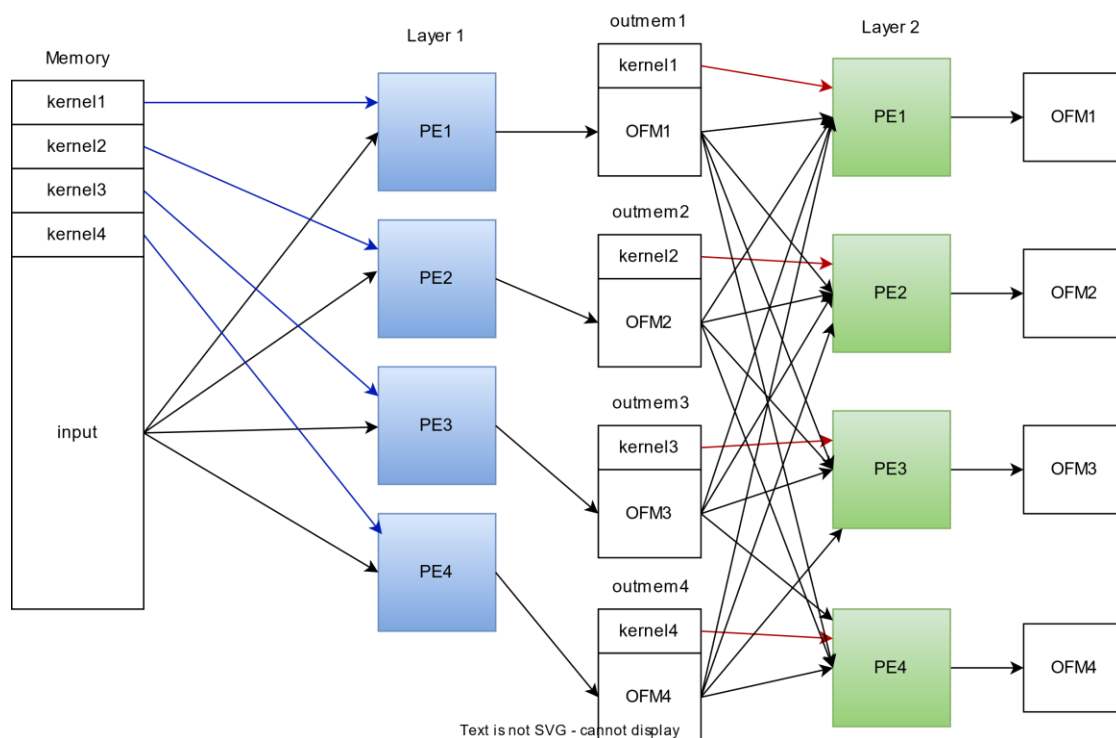
۲- فایل ورودی مانند فاز اول است که ۴ فیلتر 4×4 در آن با شروع از آدرس Y قرار دارند و از آدرس شروع X اعداد ورودی

(16×16) قرار می‌گیرند.

۳- چهار فایل فیلتر $4 \times 4 \times 4$ که هر کدام ۱۶ خط هستند، در فایل‌های جداگانه برای لایه دوم در اختیار شما قرار داده می‌شوند. شما باید از این فایل‌ها برای مقداردهی اولیه‌ی حافظه‌های `outmem1` تا `outmem4` استفاده کنید. به گونه‌ای که آدرس شروع کرنل‌ها در تمامی این حافظه‌ها `M` باشد.

۴- چهار فایل خروجی برای صحت سنجی خروجی لایه اول به شما داده می‌شود که هر کدام ۴۳ خط دارد.

۵- چهار فایل خروجی برای صحت سنجی خروجی لایه دوم به شما داده می‌شود که هر کدام ۲۵ خط دارد.



شکل ۱- شمای پیاده سازی

مواردی که در حین پیاده سازی باید در نظر بگیرید:

- برای یادآوری و درک بهتر پروژه، توصیه می‌شود که توضیحات مربوط به بخش Take home امتحان میانترم را مرور کنید و از عملکرد صحیح طراحی و کد خود اطمینان حاصل کنید.
- در این پروژه هر واحد MAC دارای حافظه خروجی مجزایی بوده و تمام MAC ها مستقل از دیگری اجرا می‌شوند. البته توجه کنید که این واحدها کاملاً سنکرون هستند.
- تعیین تعداد MAC ها به صورت پارامتری در نظر گرفته شود.

مواردی که باید در این فاز تحویل دهید:

- لازم است که **DataPath** و **Controller** این طراحی کشیده شده و در گزارش آورده شود.
- در بخش پیاده‌سازی کدی بنویسید که بتواند دولایه کانوولوشنی را پشت سر هم قرار داده و اجرا کند و نتایج نهایی در حافظه‌های نظیر هر PE در لایه دوم ذخیره شود.

توجه: انجام این تمرین به صورت گروه‌های دونفره خواهد بود.

نکات پایانی

- برای فاز دوم تمرین، لازم است فایل‌های HDL و testbench خود را مطابق ساختار توضیح داده شده در trunk/doc در subdirectory های trunk آپلود کنید. همچنین، اطمینان حاصل کنید که با اجرای trunk/sim/sim_top.tcl تست بنچ شما اجرا می‌شود. برای اجرای این اسکریپت می‌توانید از دستور زیر در Modelsim استفاده کنید:

```
>> do <sim_file>
```

- لازم است فرمت خروجی مدار شما دقیقاً مطابق ساختار مطرح شده برای ورودی باشد. توجه کنید که صحت کارکرد مدار شما با تست‌های آماده بررسی خواهد شد.
- فایل‌ها و گزارش خود را تا قبل از موعد تحویل هر فاز، با نام‌های CAD_HW3_P1_<SID>.zip و CAD_HW3_P2_<SID>.zip به ترتیب در محل‌های مربوطه برای فاز اول و دوم در صفحه درس آپلود کنید.
- هدف از این تمرین، یادگیری شماست! در صورت کشف تقلب، مطابق با قوانین درس برخورد خواهد شد.
- در صورت داشتن هرگونه سوال یا ابهام از طریق ایمیل‌های زیر با دستیاران آموزشی در ارتباط باشید.

m.fakhire17@gmail.com

mahdimn2011@yahoo.com

