



Department of
Computer Engineering

پروژه شبکه های عصبی

دکتر محمدمهدی عبادزاده

بهار 1403



+

نکاتی در مورد این پروژه که نیاز به توجه دارد:

1- لطفاً از کپی کردن پروژه از دیگران و یا چت بات ها خودداری شود و برای تحویل پروژه آمادگی حداکثری داشته باشید.

2- آخرین مهلت ارسال پروژه، ساعت 23:59 روز 21 اردیبهشت 1403 می باشد. این زمان با توجه به زمانی که مبحث از طرف استاد به صورت کامل تدریس شود تمدید خواهد شد و از این بابت نگرانی نداشته باشید.

3- دوستان فایل ارسالی خود را به صورت فشرده با فرمت zip و به صورت PR2_stdNum.zip اپلود کنید. مثال: PR2_40031000.zip

4- بخش های امتیازی در متن با رنگ سبز مشخص شده اند.

5- در صورت هرگونه سوال یا مشکل می توانید با تدریسار های پروژه در ارتباط باشید:

پویا محمدی

محمد خلیل پور

علی بازشوستری

فاز دوم: پیاده سازی MyTorch

در فاز اول پروژه شما با پیاده سازی شبکه های عصبی به کمک کتابخانه PyTorch آشنا شده اید. حالا وقتش رسیده است که عمیق تر وارد شویم و خودمان یک کتابخانه برای انجام این کار پیاده سازی کنیم. در ادامه شما را با بخش های مختلف این پروژه آشنا می کنیم.

کلاس Tensor

این کلاس اساس کار شما می باشد و تمامی محاسبات پایه ای مورد نیاز را برای شما انجام می دهد. یکی از فیلد های مهم آن نیز data می باشد که از نوع `numpy.ndarray` است، به عبارتی شما عملیات ماتریسی خود را با این نوع از داده انجام می دهید و همچنین برای پیاده سازی بقیه محاسبات ریاضی نیز لازم است از کتابخانه `Numpy` کمک بگیرید. در فایل `tensor.py` با بررسی کد و خواندن کامنت ها بخش هایی که کامل نشده است را پیاده سازی کنید.

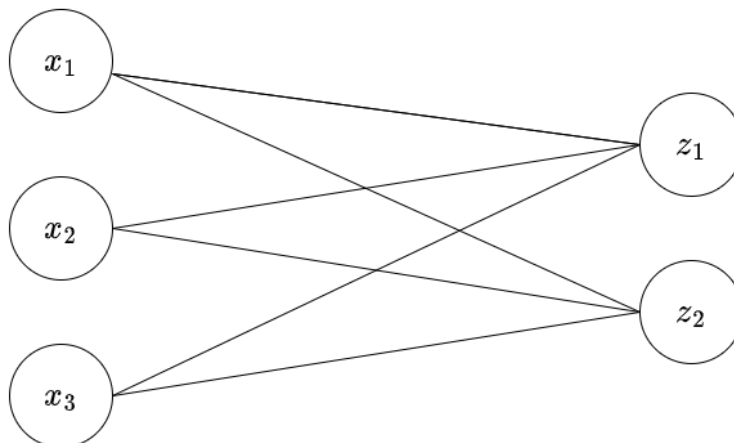
کلاس Model

یک کلاس `abstract` برای مدل هایی که بعدا تعریف می کنید می باشد. مدل شما باید در این چهار چوب پیاده سازی شود.

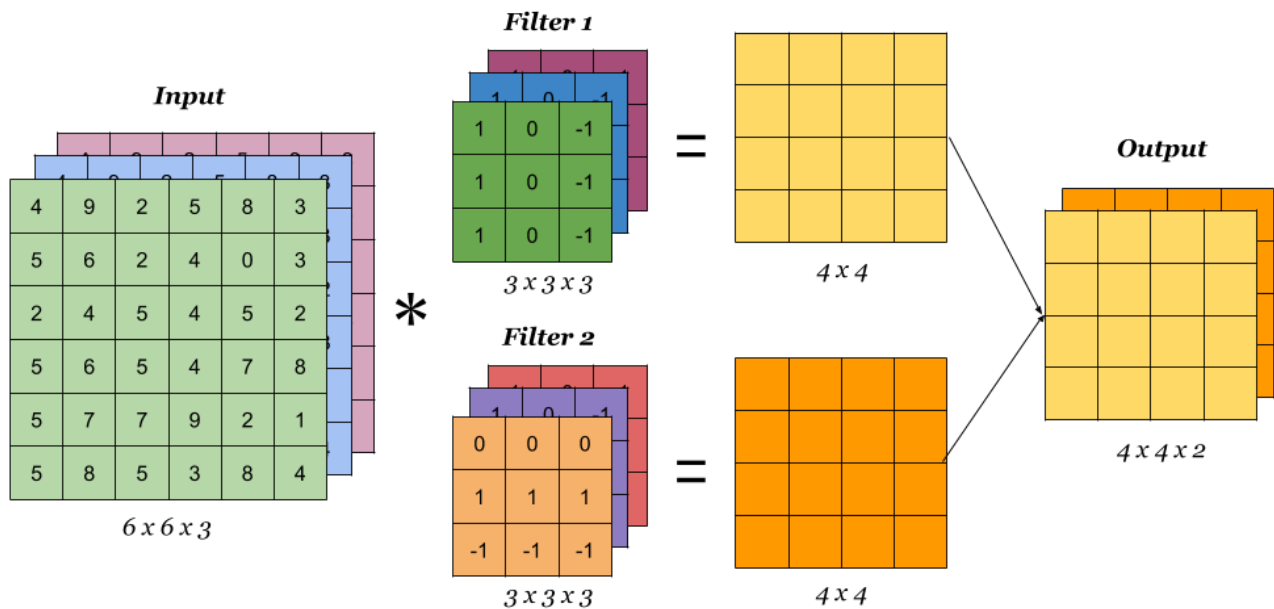
ماژول Layer

در این بخش چهار نوع از انواع لایه هایی را که در فصل اول از درس یادگرفته اید باید پیاده سازی کنید. به ترتیب لایه های `fully-connected layer` , `convolutional layer` , `average pooling` و `max pooling` می باشد.

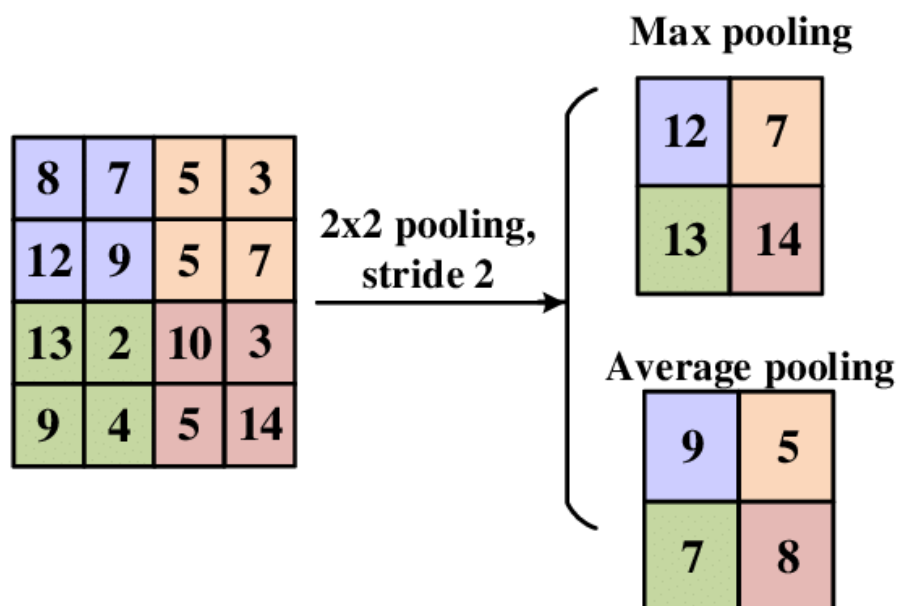
لایه ای `fully-connected` یک لایه متراکم می باشد که ورودی های آن در وزن ها ضرب شده و جمع آنها به خروجی ها می رود. شکل زیر ایده این پیاده سازی را به شما می دهد. برای پیاده سازی این قسمت فایل `linear.py` را کامل کنید.



شما با عملیات ریاضی لایه کانولوشنالی در درس آشنا شده اید. از این لایه برای استخراج ویژگی ها در تصاویر استفاده می شود. برای پیاده سازی این کلاس و بخصوص تعریف ساختار filter (همان kernel) به تصویر زیر توجه کنید. کانال های ورودی تعداد لایه های ورودی است (برای مثال در یک تصویر RGB شما 3 کانال ورودی دارید) و تعداد کانال های خروجی با توجه به نیاز شما تعریف می شود (هر مقداری می تواند باشد ولی باید یک مقدار مناسب انتخاب کنید). برای پیاده سازی این لایه فایل conv2d.py را کامل کنید.



لایه های max pooling و average pooling در واقع دو عملیات در شبکه های عصبی کانولوشنالی هستند که در عین کاهش ابعاد ویژگی های مهم را برای شما نگه می دارند. در این لایه ها عملیات مورد نظر بر هر کدام از کانال های ورودی انجام داده می شود. این لایه ها هیچ وزن و بایاسی ندارند. برای max pool باید عملیات بیشینه گیری را در هر kernel انجام دهید و برای average pool در هر kernel میانگین اعضا را بگیرید و خروجی بدهید. برای پیاده سازی این بخش فایل های max_pool2d.py و avg_pool2d.py را کامل کنید.



ماژول Activation

در این بخش شما باید توابع فعال سازی را پیاده سازی کنید. تابع `step` پیاده سازی شده و سایر توابع نیاز به کامل شدن دارند. توابع `sigmoid`, `relu`, `leaky relu` و `softmax` اجباری می باشند و تابع `tanh` امتیازی است. شما با توابع مذکور در درس آشنا شده‌اید.

ماژول Loss

در این بخش شما باید توابع هزینه را پیاده سازی کنید. از شما خواسته شده می شود که دو تابع CE^2 و MSE^1 را تکمیل کنید.

ماژول Optimizer

در این بخش شما باید بهینه ساز ها را پیاده سازی کنید. در فصل اول درس شما با روش گرادیان کاهشی آشنا شده‌اید و باید پیاده سازی آن را در `sgd.py` کامل کنید. دقت داشته باشید شما در هر گام باید پارامتر های مدل را با انجام عملیات ریاضی مربوط به آن الگوریتم بروزرسانی کنید. پیاده سازی **مابقی بهینه ساز ها** امتیازی می باشد و برای پیاده سازی آنها می توانید تحقیق کنید.

تسک اول: پیاده سازی شبکه ساده

در این قسمت شما باید یک شبکه خطی ساده را پیاده سازی کرده و آن را آموزش دهید. به شما یک معادله ریاضی ساده داده شده است و ضرایب ورودی آن را می توانید به صورت دلخواه انتخاب کنید. در ادامه شما با تکمیل بخش های مورد نیاز باید طوری مدل را آموزش دهید که وزن های آن به ضرایب شما میل کند.

مجموعه داده MNIST

این مجموعه داده شامل تصاویر اعداد 0 تا 9 می باشد. این تصاویر رزولوشن پایین سیاه و سفید هستند و داده ورودی شما به صورت 28×28 می باشد. یک نسخه سبک تر از این مجموعه داده در اختیار شما قرار داده شده است و می توانید به کمک کلاس `DataLoader` این داده ها را به صورت مناسبی در پروژه خود `load` کنید.

شما در فاز اول پروژه با مجموعه داده `CIFAR10` آشنا شده اید و به کمک `PyTorch` مدل هایی با معماری MLP^3 و CNN^4 پیاده سازی کرده اید. حال در این فاز باید این دو معماری را با `MyTorch` پیاده سازی کرده و برای مجموعه داده `MNIST` مدل خود را آموزش دهید.

¹ Mean Squared Error

² Cross Entropy

³ Multi-Layer Perceptron

⁴ Convolutional Neural Network

تسک دوم: آموزش مدل با MNIST و معماری MLP

شما باید یک مدل با معماری MLP پیاده سازی کنید و آن را آموزش دهید. مقدار دقت نهایی شما باید حداقل 75 درصد باشد و **دقت بالای 85 نمره** امتیازی دارد.

تسک سوم: آموزش مدل با MNIST و معماری CNN

شما باید یک مدل با معماری CNN پیاده سازی کنید و آن را آموزش دهید. مقدار دقت نهایی شما باید حداقل 80 درصد باشد و **دقت بالای 90 نمره** امتیازی دارد.

نکات پیاده سازی:

فایل های پروژه بر روی کورسز آپلود شده اند. در پیاده سازی این پروژه به نکات زیر توجه کنید:

- ابتدا سعی کنید ماژول MyTorch را کامل کنید.
- برای انجام تسک ها به تکمیل قسمت های TODO در فایل های نوت بوکی که در اختیار شما قرار داده شده است بپردازید.
- یک نسخه سبک تر از مجموعه داده MNIST در اختیار شما قرار داده شده است که بایستی از آن استفاده کنید.
- لازم به ذکر است که در صورت نیاز به تغییر سایر بخش ها باید به صورت خلاصه دلیل خود را برای تغییر آن قسمت توضیح دهید.
- **تکمیل کردن تمیز پیاده سازی ها** نمره امتیازی دارد.