

۱. (۲۵٪) [پژوهش و محاسبه] برای هر کدام از سوالات زیر پاسخ را بدست آورید.

۱-۱ [۵٪] دو مفهوم Vanishing problem و Exploding problem را توضیح داده و راه‌حلی را برای برطرف کردن این مشکل‌ها پیشنهاد دهید. کدام یک از توابع فعال سازی می‌تواند باعث این دو پدیده شوند؟

۱-۲ [۵٪] در مورد L1 Regularization و L2 Regularization تحقیق کنید و تفاوت میان آن‌ها را با جزئیات شرح دهید و مشخص کنید هر کدام از آنها چگونه بر به‌روزرسانی وزن‌های مدل اثر می‌گذارند. کدام مورد را ترجیح می‌دهید؟ دلیل انتخاب خود را توضیح دهید.

۱-۳ [۵٪] پدیده شکست تقارن (Symmetry breaking) چیست؟ آن را توضیح دهید. برای جلوگیری از آن چه کار(های)ی می‌توان انجام داد؟

۱-۴ [۱۰٪] فرض کنید یک شبکه عصبی تک لایه برای طبقه‌بندی دودویی (Binary classification) دارید. ورودی این شبکه $X \in R^{n \times m}$ و خروجی $\hat{y} \in R^{1 \times m}$ و برچسب واقعی $y \in R^{1 \times m}$ است. فرمول‌های پیش‌خور (Forward propagation) به صورت زیر هستند:

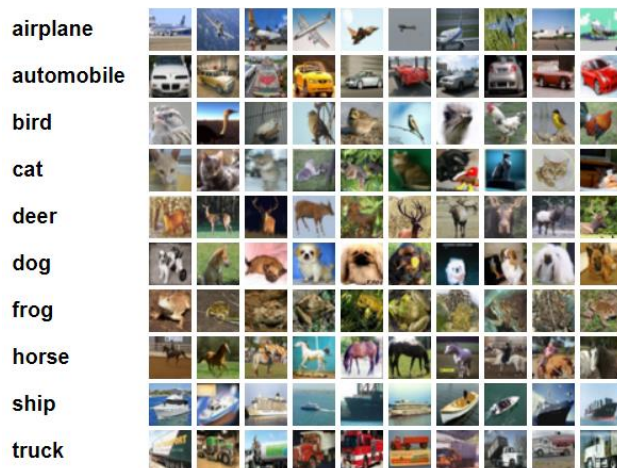
$$\begin{aligned} z^{[1]} &= W^{[1]}X + b^{[1]} \\ a^{[1]} &= \sigma(z^{[1]}) \\ \hat{y} &= a^{[1]} \end{aligned}$$

$$J = - \sum_{i=1}^m y^{(i)} \log(\hat{y}^{[i]}) + (1 - y^{(i)}) \log(1 - \hat{y}^{[i]})$$

عبارت $\frac{\partial J}{\partial W^{[1]}}$ را با استفاده از قانون زنجیر (Chain rule) به دست آورید.

۲. (۷۵٪) [پیاده‌سازی: دسته‌بندی تصاویر با شبکه عصبی پیچشی] در این بخش قرار است که با

استفاده از یک چارچوب برنامه‌نویسی یادگیری عمیق، یک مدل برای دسته‌بندی تصاویر مجموعه داده‌های CIFAR-10 پیاده‌سازی کنیم. این مجموعه داده از ۶۰ هزار تصویر رنگی با ابعاد ۳۲×۳۲ تشکیل شده است (شامل ۵۰ هزار تصویر آموزشی و ۱۰ هزار تصویر آزمایشی). تصاویر این مجموعه داده در ده دسته قرار گرفته‌اند که نمونه‌ای از هر یک از این دسته‌ها در شکل ۱ ملاحظه می‌شود.



شکل ۱ نمونه تصاویری از هریک از برچسب‌های مجموعه داده‌ی CIFAR-10

به صفحه‌ی رسمی این مجموعه داده^۱ مراجعه کنید، آن را دریافت کرده و مجموعه‌های آزمایشی و آموزشی آن را در این بخش از تمرین مورد استفاده قرار دهید. داده‌های آموزشی این پایگاه داده در پنج دسته‌ی جداگانه (هریک با اندازه‌ی ۱۰ هزار تصویر) ارائه شده‌اند. در صورتی که منابع شما اجازه‌ی اجرای الگوریتم بر روی تمامی مجموعه داده را نمی‌دهد، می‌توانید به عنوان داده‌ی آموزشی، تنها دسته اول آن (`data_batch_1`) را مورد استفاده قرار دهید. توجه کنید که در این صورت لازم است این موضوع را در گزارش خود ذکر کنید.

شبکه‌ای پیچشی با معماری زیر در نظر بگیرید:

(۱) لایه‌ی پیچشی: اندازه‌ی کرنل: 4×4 ، (تعداد کانال‌های خروجی: ۶۴ و تابع فعال‌سازی: ReLU)

(۲) لایه‌ی ادغام بیشینه: اندازه‌ی کرنل: 2×2

(۳) لایه‌ی پیچشی: اندازه‌ی کرنل: 4×4 ، (تعداد کانال‌های خروجی: ۳۲ و تابع فعال‌سازی: ReLU)

(۴) لایه‌ی ادغام بیشینه: اندازه‌ی کرنل: 2×2

(۵) لایه خطی با تعداد ۲۵۶ (تعداد دسته‌ها) نرون خروجی

(۶) لایه خطی با تعداد ۱۰ (تعداد دسته‌ها) نرون خروجی

^۱ <https://www.cs.toronto.edu/~kriz/cifar.html>



این شبکه را با بهینه‌ساز Adam، تابع خطای categorical cross entropy و نرخ یادگیری یک هزارم آموزش دهید (با اندازه دسته ۳۲). این معماری و نحوه‌ی آموزش را به عنوان حالت پایه در نظر گرفته. در هریک از بخش‌های الف تا ذ تغییرات گفته شده را بر روی آخرین معماری اعمال کنید.

نکته ۱: قسمتی از داده‌های آموزشی را به عنوان داده‌های validation جدا کرده (می‌توان به عنوان یک پارامتر هنگام فراخوانی تابع آموزش نیز تنظیم شود) و میزان خطای آن را معیاری برای زمان توقف الگوریتم قرار دهید. همچنین از دقت داده‌های آزمایشی به منظور ارزیابی مدل استفاده کنید.

نکته ۲: برای هریک از بخش‌های الف تا ذ، موارد زیر را گزارش کنید:

- نموداری از روند تغییرات خطای شبکه در حین آموزش، بر روی داده‌های train و validation (این دو نمودار را روی یکدیگر قرار دهید)
- دقت مدل بر روی داده‌های آزمایشی
- میانگین مدت زمان اجرای تکرارها
- تحلیل تغییرات «سرعت همگرایی بر اساس تعداد تکرار» و «توان شبکه در یادگیری الگو» در اجراهای مختلف
- بررسی تمام موارد بالا و تحلیل علت تغییرات آنها

الف) نتایج مندرج شده از مشاهدات خود را از مدل پایه بنویسید. (آیا مدل دچار بیش برآزش (Overfitting) شده است؟ اگر پاسخ شما بلی است، راه‌حلی پیشنهاد دهید.)

ب) تعداد کانال‌های خروجی لایه اول را به ۳۲ تغییر دهید و لایه drop-out با احتمال ۳۰ درصد را به مدل اضافه کنید. تغییرات این مدل را با مدل قبلی مقایسه کنید.

پ) تعداد لایه‌های پیچشی را از دو لایه به سه و چهار لایه و اندازه‌ی کرنل لایه‌های پیچشی را به اندازه ۳×۳ تغییر دهید.

ت) لایه‌ی Batch Normalization را بعد از هر لایه پیچشی به مدل اضافه کنید.

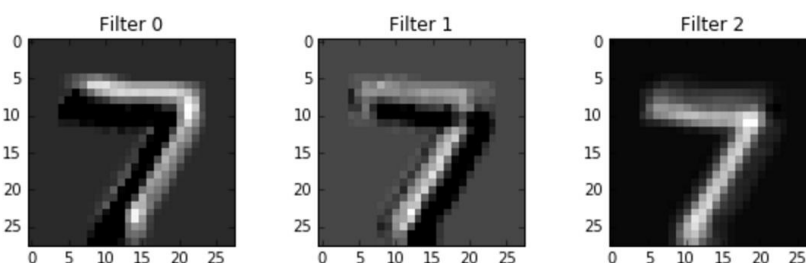
ث) ضریب یادگیری بهینه‌ساز را به 10^{-4} ، 10^{-3} و 10^{-1} تغییر دهید.

ج) تابع فعال‌ساز را به leaky ReLU و Tanh تغییر دهید.

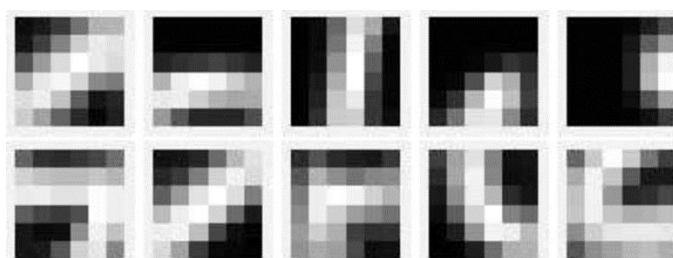
چ) بهینه‌ساز را به SGD و Adam تغییر دهید.

ح) اندازه‌ی batch-size را به ۴ و ۱۲۸ تغییر دهید.

خ) پس از آموزش مدل، یکی از تصاویر آزمایشی را در نظر بگیرید، آن را نمایش دهید و آن را به عنوان ورودی به شبکه بدهید. پس از عبور تصویر از لایه‌ی اول پیچشی، هریک از کانال‌ها تصویری خاکستری^۳ با ویژگی خاصی از تصویر ورودی را نمایش می‌دهد (شبیه به شکل ۲). همچنین وزن‌های مربوط به یکی از کانال‌های کرنل را نیز می‌توان به عنوان یک تصویر خاکستری در نظر گرفت و آن را نمایش داد (شبیه به شکل ۳). حال شما تصویر کانال‌های مختلف لایه اول پیچشی را در کنار تصویر وزن‌های کرنل‌های همان کانال نمایش دهید و ارتباط آن‌ها را توضیح دهید.



شکل ۲ نمونه سه تصویر که توسط سه فیلتر مختلف در لایه‌های میانی شبکه‌ی عصبی تولید شده است



شکل ۳ نمونه ۱۰ تصویر از وزن‌های کرنل (فیلتر) با اندازه‌ی ۶×۶

^۳ gray-scale



د) تعدادی از داده‌های تست را انتخاب کنید و نتایج پیش‌بینی شده توسط مدل را با نتایج واقعی مدل نمایش دهید.

ذ) ماتریس درهم‌ریختگی (confusion matrix) را نمایش دهید.

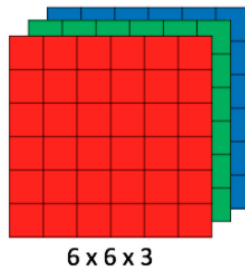
ر) معماری VGG19 و ResNet50 را در نظر بگیرید. در این بخش، با به‌کارگیری نمونه‌ی پیش‌آموزش دیده‌شده‌ی این معماری‌ها بر روی مجموعه‌داده‌ی ImageNet^۴ (که به صورت آماده در چارچوب‌های برنامه‌نویسی tensorflow و pytorch قرار داده شده است)، می‌خواهیم مدلی برای دسته‌بندی مجموعه‌ی داده‌ی CIFAR-10 آموزش دهیم. برای این منظور لازم است به انتهای بخش استخراج ویژگی^۵ این شبکه، با وزن‌های آموزش دیده‌شده، لایه‌های خطی (با تعداد خروجی‌های ۱۰۲۴، ۵۱۲، ۲۵۶ و در نهایت ۱۰) اضافه شود. سرعت همگرایی این مدل بر حسب تکرار و دقت آن بر روی داده‌های آزمایشی را با حالت پایه مقایسه، علت تفاوت‌ها را بیان کرده و نکات مثبت و منفی استفاده از مدل پیش‌آموزش دیده‌شده را بررسی کنید.

^۴ <http://image-net.org/>

^۵ Feature extraction

پیوست: مروری بر لایه‌ی پیچشی^۶ در شبکه‌های عصبی عمیق

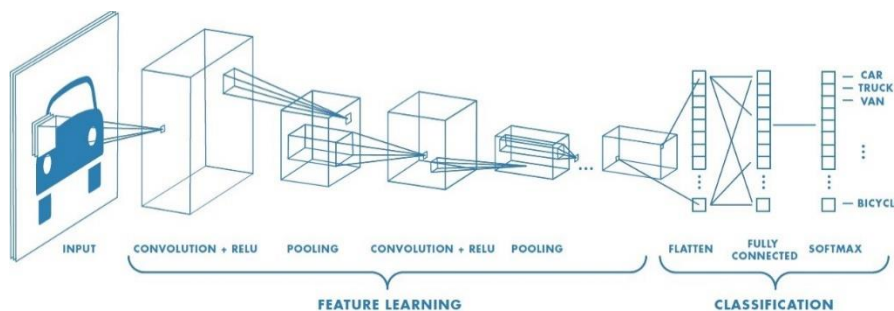
در شبکه‌های عصبی عمیق، لایه‌های پیچشی، نقشی کلیدی در بازشناسی الگو در تصاویر دارد. این لایه تقریباً در تمامی معماری‌های شبکه‌ی عصبی برای کاربردهای پردازش تصویر به کار می‌رود. حتی در کاربردهای غیر تصویری نیز گاهی پس از تبدیل داده‌ی ورودی به یک تصویر، از لایه‌ی پیچشی برای بازشناسی الگوی آن استفاده می‌شود. به عنوان مثال در کاربردهایی نظیر شناسایی اشیاء، تشخیص چهره و تشخیص گفتار (با تبدیل صوت به تصویر)، لایه‌ی پیچشی به طور گسترده مورد استفاده قرار می‌گیرد. در کاربرد «دسته‌بندی تصاویر» یک تصویر ورودی، معمولاً یک تصویر رنگی است به صورت یک ماتریس سه‌بعدی در قالبی مانند شکل ۴ ذخیره می‌شود. در بعد سوم این ماتریس، کانال‌های رنگی تصویر قرار می‌گیرد که به ترتیب شامل مقادیر روشنایی برای نورهای قرمز، سبز و آبی است.



شکل ۴ نمایی از ماتریس سه‌بعدی یک تصویر رنگی با ابعاد ۶×۶

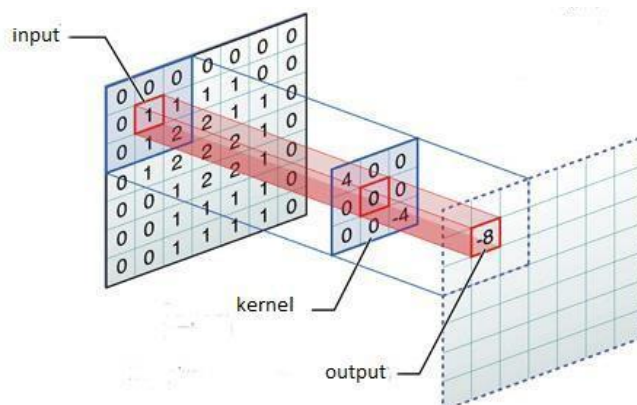
یک شبکه‌ی عصبی عمیق شامل لایه‌ی پیچشی، معمولاً شامل لایه‌های ادغام و خطی (fully-connected) نیز می‌شود. در ادامه توضیح مختصری در مورد لایه‌ی پیچشی و ادغام آورده شده است. در شکل ۵ یک نمونه شبکه‌ی عصبی معمول که به وسیله‌ی آن دسته‌بندی تصاویر صورت می‌گیرد، ملاحظه می‌شود.

^۶ Convolutional





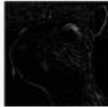




شکل ۵ نمونه معماری یک شبکه‌ی عصبی برای دسته‌بندی تصاویر

لایه‌ی پیچشی: معمولا به عنوان اولین لایه‌ها برای استخراج ویژگی از تصویر در نظر گرفته می‌شود. این لایه، در طول آموزش یاد می‌گیرد که چه ویژگی‌های محلی را از تصویر ورودی باید استخراج کند. تک تک پیکسل‌های خروجی این لایه با استفاده از یک فیلتر و حرکت کردن آن بر روی تصویر ورودی مانند شکل ۶ محاسبه می‌شود.



شکل ۶ نحوه‌ی اعمال فیلتر بر روی تصویر در لایه‌ی پیچشی

اعمال فیلترهای مختلف بر روی تصویر ورودی می‌تواند ویژگی‌های مختلفی از تصویر ورودی را استخراج کند. در شکل ۷ نمونه‌هایی از این ویژگی‌ها و فیلترهای متناظر آن‌ها دیده می‌شود.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

شکل ۷ چند نمونه از فیلترهای پرکاربرد و نتیجه‌ی اعمال هریک از فیلترها بر روی تصویر ورودی (سطر اول)

لایه‌ی ادغام^۷: این لایه برای کاهش ابعاد تصویر در شبکه استفاده می‌شود. در این لایه هدف این است که با حفظ اطلاعات مهم، تعداد پارامترها، کاهش یابد. ادغام بیشینه^۸ یکی از انواع ادغام است که در آن در هر بخش از ورودی تنها مقدار بیشینه نگه داشته شده و سایر مقادیر دور ریخته می‌شود. در شکل ۸ نمونه‌ای از ادغام بیشینه دیده می‌شود.

⁷ Pooling

⁸ Max-Pooling

بر نام خدا

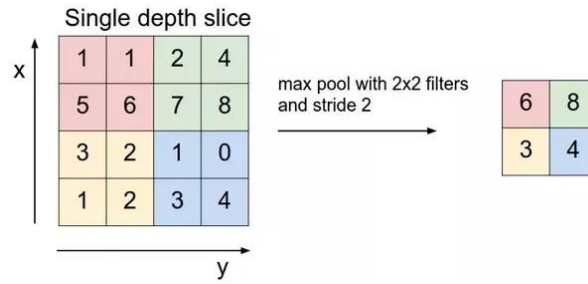
مبانی محاسبات (رایانش) نرم (۸۳-۰۵-۰۳۹-۰۱)
نیمسال اول ۱۴۰۲-۱۴۰۳



دانشکده علوم و فنون نوین

تاریخ تحویل: ۱۴۰۲/۰۸/۲۷

تمرین شماره ۳



شکل ۸ نمونه‌ی ادغام بیشینه