به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس شبکه‌های عصبی

استاد صفابخش

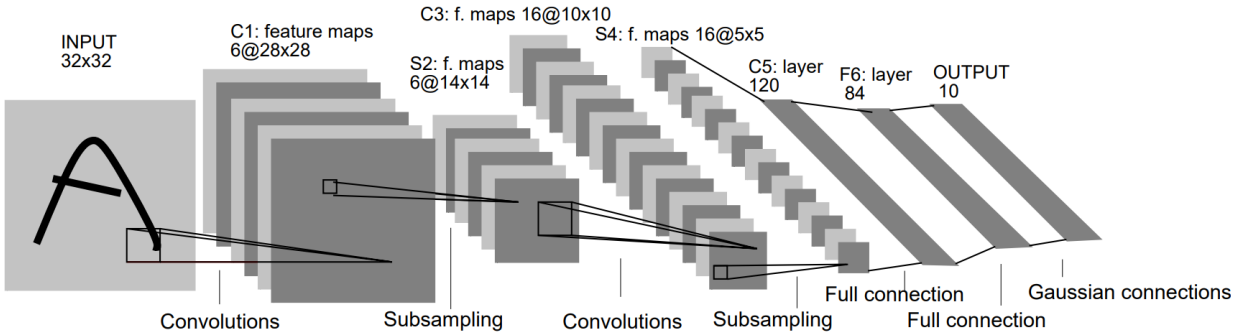
تمرین چهارم

علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

سوال ۱[۱][۲]

شبکه LeNET-5 دارای هفت لایه است: سه لایه کانوولوشنی، دو لایه نمونه‌برداری (Subsampling) و دو لایه از نوع تماما متصل (Fully Connected). این معماری در تصویر زیر قابل مشاهده است:



در معماری پیشنهادی آن‌ها ابعاد ورودی ۳۲×۳۲ است. کرنل استفاده‌شده در کلیه‌ی لایه‌های کانوولوشنی ۵×۵ به همراه تابع فعال‌سازی تانژانت هایپربولیک و در لایه‌های نمونه‌برداری ۲×۲ با یک تابع فعال‌سازی سیگموید است پس از اولین لایه کانوولوشنی تعداد ۶ نقشه ویژگی با ابعاد ۲۸×۲۸ تشکیل می‌شود. لایه نمونه‌برداری بعد از آن ابعاد هر نقشه ویژگی را به ۱۴×۱۴ کاهش می‌دهد. دومین لایه کانوولوشنی ۱۶ نقشه ویژگی با ابعاد ۱۰×۱۰ ایجاد می‌کند که لایه نمونه‌برداری بعد از آن ابعاد را به ۵×۵ کاهش می‌دهد. نهایتا آخرین لایه کانوولوشنی ۱۲۰ نقشه ویژگی با ابعاد ۱×۱ می‌سازد. در این مرحله عملا داده ورودی به یک داده یک بعدی با ۱۲۰ ویژگی تبدیل می‌شود. سپس با دو لایه تماما متصل ابعاد به ترتیب به ۸۴ و ۱۰ کاهش پیدا می‌کند.

شبکه‌ای که آن‌ها پیشنهاد داده‌اند برای تشخیص اعداد دست‌نویس انگلیسی است و لذا خروجی نهایی مشخص می‌کند که داده ورودی با چه احتمالی به کدام

---

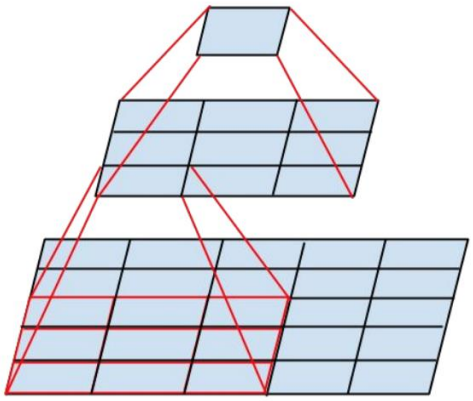
[1] https://towardsdatascience.com/understanding-and-implementing-lenet-5-cnn-architecture-deep-learning-a2d531ebc342

[2] https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/

کلاس متعلق است. همچنین داده ورودی در اصل از نوع ۲۸×۲۸ است که با حاشیه‌گذاری (Padding) به ۳۲×۳۲ تبدیل می‌شود.

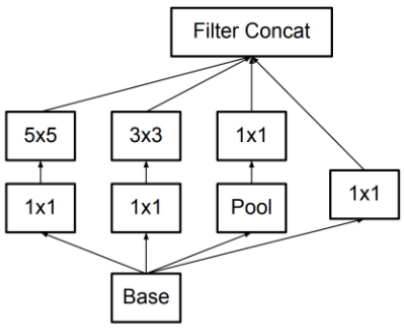
شبکه Inception-v3 تمرکز ویژه‌ای روی کاهش بار محساباتی داشته است. برای این امر از پنج تکنیک مهم استفاده کرده است که در ادامه آن را بررسی می‌کنم:

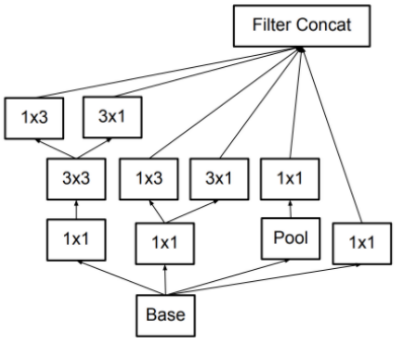
۱) کانوولوشن‌های تجزیه‌شده (Factorized Convolutions): با تجزیه کردن کانوولوشن‌ها پارامترها کاهش می‌یابد و کارایی حفظ می‌شود.



۲) کانوولوشن‌های کوچک‌تر: استفاده از کانوولوشن‌های کوچکتر تعداد پارامتر کمتری دارد و محاسبات کمتری را رقم خواهد زد. در LeNET-5 از کانوولوشن‌های ۵×۵ استفاده شده است که ۲۵ پارامتر دارد. در Inception-v3 با جایگزین کردن دو کانولوشن ۳×۳ با حفظ کارایی تعداد پرامترها را به ۱۸ رسانده‌اند. در تصویر روبرو می‌توان دید که چگونه یک کانوولوشن ۵×۵ با دو مرحله کانوولوشن ۳×۳ جایگزین شده است.

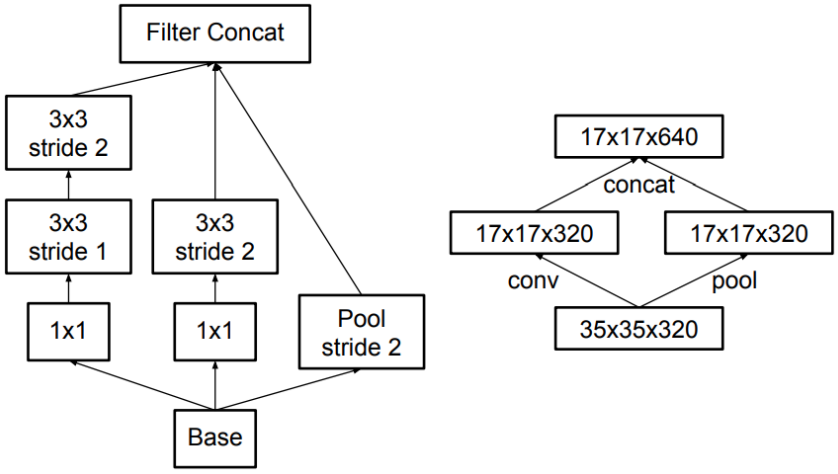
۳) کانوولوشن‌های نامتقارن (Asymmetric Convolutions): در Inception-v3 از کانوولوشن‌های نامتقارن کمک گرفته می‌شود که زمان آموزش را کاهش می‌دهد. یک کانوولوشن ۳×۳ را اگر بخواهیم با تکنیک کانوولوشن‌های کوچک‌تر بشکنیم نیاز به دو کانوولوشن ۲×۲ خواهیم داشت که چندان کاهش پارامتر ندارد ولی اگر از یک کانوولوشن ۳×۱ و یک کانوولوشن ۱×۳ استفاده کنیم مفید است. معماری تصویر راست جایگزین‌شده معماری تصویر چپ با بهره‌گیری از این تکنیک و تکنیک قبلی است.



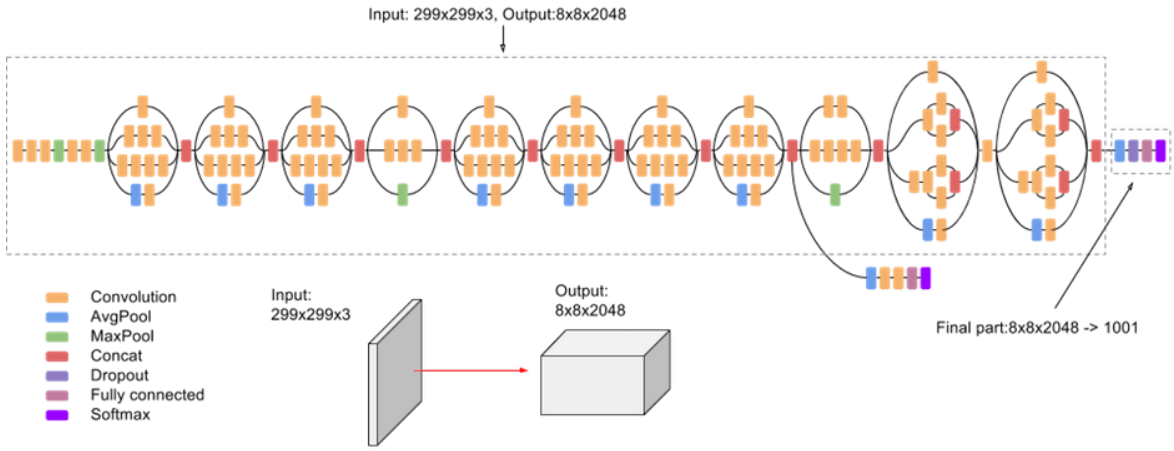


۴) دسته‌بند کمکی: دسته‌بند کمکی یک شبکه کانوولوشنی کوچک است که در میان لایه‌های شبکه اصلی در حین آموزش قرار می‌گیرد. تابع هزینه در زمان آموزش شامل تابع هزینه این زیرشبکه‌های کمکی هم می‌شود. این دسته‌بند کمکی در نقش یک منظم‌ساز برای شبکه عمل می‌کند. بدیهی است که چنین چیزی در LeNET-5 وجود نداشته است.

۵) کاهش ابعاد (Grid Size Reduction) بهینه: در Inception-v3 برای کاهش ابعاد ورودی از نحوه دیگری از ترکیب لایه‌ها استفاده کرده است که در مجموع تعداد محاسبات کمتری را خواهد داشت. در تصویر زیر معماری مربوط به این تکنیک آورده شده است.



در تصویر زیر معماری نهایی مربوط به Inception-v3 آورده شده است:



سوال ۲

لازم است ابتدا نکات کلی راجع به پیاده‌سازی خودم بیان کنم:

- در شبکه اصلی ارائه‌شده برای مقاله داده‌ها دارای ابعاد (۱و۳۲و۳۲) است درحالی‌که داده‌های این سوال از نوع (۳و۵۰۰و۵۰۰). برای آنکه داده‌ها ۵۰۰×۵۰۰ پس از سه لایه کانوولوشن و دو لایه نمونه‌برداری به ۱×۱ برسد به ناچار مقادیر strides در لایه‌های کانوولوشن و pool_size در لایه‌های نمونه‌برداری را بیشتر از حالت اصلی قرار دادم. بدین شکل شبکه با کمترین تغییرات مناسب مسئله جدید می‌شود. به عنوان راه جایگزین می‌توانستیم ابعاد ورودی را پیش از دادن به شبکه کم کنیم و تصویر را سیاه و سفید کنیم ولی طبیعی است که در این راه بخشی از اطلاعات از بین می‌رود و به دقت پایین‌تری می‌رسدیم.
- مقادیر kernel_size در لایه‌های کانوولوشن به طور پیش‌فرض برابر ۵ و تعداد کرنل‌های هر لایه و تعداد واحدهای لایه‌های تماما متصل مانند حالت اصلی در نظر گرفته شده است.
- برای منظم‌سازی از مقادیر پیش‌فرض هر یک از منظم‌سازی‌های استفاده کردم.
- برای جلوگیری از بیش‌برازش از یک کالبک EarlyStopping استفاده کردم و تعداد گام حداکثر برابر با ۳۰ تعبین شده است. بدین ترتیب ممکن است تعداد گام آموزش برای تنظیمات مختلف متفاوت باشد.

تاثیر منظم‌سازی

تنظیمات	صحت آموزش	صحت اعتبارسنجی	صحت آزمون	تعداد گام
بدون منظم‌سازی	۷۲٫۴۴٪	۷۱٫۱۴٪	۶۸٫۷۵٪	۱۶
منظم‌سازی Dropout	۷۵٫۲۴٪	۷۵٫۹۴٪	۷۱٫۰۹٪	۲۶
منظم‌سازی L1	۶۴٫۶۰٪	۶۴٫۶۶٪	۶۴٫۸۴٪	۲۸
منظم‌سازی L2	۶۸٫۱۸٪	۶۹٫۹۲٪	۶۷٫۹۷٪	۱۱

به طور کلی استفاده از منظم‌سازی‌ها به جز L2 باعث افزایش تعداد گام آموزش شده است. طبیعی است که اگر از کالبک مذکور استفاده نمی‌شد، مدل بدون منظم‌سازی با ۳۰ گام دچار بیش‌برازش می‌شد. ولی با شرایط فعلی استفاده از این منظم‌سازها به جز L2 همگرایی را کند کرده است. از نظر دقت منظم‌سازهای به غیر از Dropout دقت را کاهش داده‌اند.

با شرایط فعلی به نظر نمی‌رسد که منظم‌سازی‌ها با پارامترهای پیش‌فرض در پیاده‌سازی من چندان مفید باشند. از آنجایی که زمان آموزش به طور کلی پایین است شاید استفاده از Dropout بد نباشد.

تاثیر تعداد کرنل

تعداد کرنل	صحت آموزش	صحت اعتبارسنجی	صحت آزمون	تعداد گام
۳-۱۰-۴۰	۶۷/۵۰٪	۶۶/۹۲٪	۶۹/۵۳٪	۷
۶-۱۶-۱۲۰	۷۲/۳۴٪	۷۲/۱۸٪	۶۷/۹۷٪	۱۹
۱۰-۴۰-۳۰۰	۸۰/۳۷٪	۷۹/۷۰٪	۷۵/۰۰٪	۱۹

استفاده از تعداد کرنل بیشتر باعث می‌شود که همگرایی کند شود. در کنار آن باید توجه کرد هر گام با زمان بیشتری طول خواهد کشید؛ اما به طور کلی دقت افزایش پیدا کرده است. با تنظیماتی که آزمایش شده است به نظر می‌رسد استفاده از به ترتیب ۱۰، ۴۰ و ۳۰۰ کرنل در سه لایه کانوولوشنی به نتایج بهتری از حالت پیش‌فرض منجر شود.

تاثیر اندازه کرنل

اندازه کرنل	صحت آموزش	صحت اعتبارسنجی	صحت آزمون	تعداد گام
۵×۵	۷۴/۸۵٪	۷۰/۶۸٪	۷۰/۳۱٪	۱۶
۷×۷	۶۸/۴۷٪	۶۶/۱۷٪	۷۰/۳۱٪	۱۱
۹×۹	۶۸/۵۷٪	۶۷/۶۷٪	۶۵/۶۲٪	۸

کرنل‌های با اندازه بیشتر با تعداد گام کمتری به همگرایی می‌رسند که از این بابت خوب است ولی تعداد پارامتر به مراتب بیشتر خواهند داشت؛ مثلا یک کرنل ۹×۹ بیش از سه برابر یک کرنل ۵×۵ پارامتر دارد. از منظر دقت هم به نظر می‌رسد کرنل‌های کوچک‌تر نتایج بهتری داشته باشند. پس با این شرایط مزیت خاصی را نمی‌توان برای کرنل‌های بزرگ برای این مسئله و برای پیاده‌سازی من درنظر گرفت.

بهترین تنظیم

شاید جالب باشد که ببینیم ترکیب بهترین تنظیم از هر یک از سه بررسی قبل به چه نتیجه‌ای ختم می‌شود. برای این کار از منظم‌سازی Dropout با ۱۰، ۴۰ و ۳۰۰ کرنل ۵×۵ در سه لایه کانوولوشنی استفاده کردم. نتیجه در جدول زیر آورده شده است:

صحت آموزش	صحت اعتبارسنجی	صحت آزمون	تعداد گام
۷۹٫۲۱٪	۷۲٫۱۸٪	۷۵٫۷۸٪	۳۰ (بیشینه گام)

از نظر صحت آزمون به بهترین نتایج رسیدیم که تقریبا مورد انتظار بود (اگرچه باتوجه به آزمایشات جداگانه ممکن بود خلافش پیش بیاید). استفاده همزمان از Dropout و تعداد کرنل زیاد باعث شد آموزش تا ۳۰ گام طول بکشد و اگر تعداد گام بیشتر داشتیم باز هم آموزش ادامه پیدا می‌کرد که از این لحاظ همگرایی کند و سرعت آموزش پایین نسبت به مدل‌های بررسی‌شده اتفاق می‌افتد.

سوال ۳

انتقال یادگیری یک تکنیک یادگیری ماشین است که یک مدل آموزش دیده در یک وظیفه قابل استفاده در یک وظیفه مشابه دیگر می‌شود. این کار باعث می‌شود که کمبود داده در مسئله اصلی تاحدی جبران شود و سرعت آموزش تسریع پیدا کند و به دقت‌های بالاتری دست پیدا کنیم.

روال کلی کار به این شکل است که ابتدا یک مدل آماده که با داده‌های کافی برای یک وظیفه مشابه آموزش داده شده است درنظر گرفته می‌شود (گام انتخاب مدل). وزن‌های این مدل پیش آموزش‌داده‌شده به عنوان وزن‌های شروع مدل جدید برای وظیفه اصلی در نظر گرفته می‌شود. طبیعتا ممکن است بخشی از شبکه مدل اولیه استفاده شود و یا آنکه نیاز به افزودن لایه یا تغییراتی نسبتا جزئی در مدل جدید وجود داشته باشد (گام استفاده مجدد مدل). نهایتا باید با آموزش مدل جدید با داده‌های مسئله مدل تنظیم دقیق شوند (گام تنظیم مدل).

سوال ۴

برای این سوال هم مشابه شبکه قبلی، حداکثر ۳۰ گام آموزش درنظر گرفتم و با یک کالبک Early Stopping و یک لایه Dropout جلوی بیش‌برازش احتمالی را گرفتم. همچنین برای شبکه این سوال از مقدار یادگیری ۰/۰۰۰۱ در بهینه‌ساز Adam استفاده کردم؛ چراکه مقدار پیش‌فرض یعنی ۰/۰۰۱ منجر به نتایج بدتری می‌شد.

تعداد لایه بهینه فریزشده

در جدول زیر نتایج صحت روی مجموعه‌داده‌های مختلف و تعداد گام مورد نیاز برای آموزش مدل آورده شده است:

تعداد لایه فریزشده	صحت آموزش	صحت اعتبارسنجی	صحت آزمون	تعداد گام
۰	۹۹/۵۲٪	۹۳/۹۸٪	۹۲/۹۷٪	۱۲
۵۰	۹۹/۹۰٪	۷۵/۹۴٪	۸۲/۸۱٪	۱۵
۱۰۰	۹۵/۹۴٪	۶۹/۱۷٪	۶۹/۵۳٪	۱۳
۲۰۰	۷۲/۲۴٪	۶۳/۱۶٪	۵۶/۲۵٪	۸
۳۱۱	۴۴/۲۹٪	۳۰/۸۳٪	۴۱/۴۱٪	۱۴

با توجه به نتایج حاصل‌شده تعداد لایه فریزشده بهینه برابر با صفر است؛ یعنی بهتر است هنگام آموزش هیچ لایه‌ای را فریز نکنیم.

بر اساس نتایج حاصل‌شده هر چه تعداد لایه فریزشده کمتر باشد، صحت آزمون بیشتری خواهیم داشت. در عین حال باید توجه کرد که هر چه تعداد لایه فریزشده کمتر باشد، آموزش یک گام به دلیل بروز شدن وزن‌های بیشتر، کندتر خواهد بود؛ اما باتوجه به اختلاف شدید صحت روی این مجموعه‌داده و متناسب با پیاده‌سازی من کاملا به صرفه است که تعداد لایه فریزشده را کمترین مقدار ممکن لحاظ کنیم.

مقایسه دو مدل

برای مقایسه دو مدل بهترین تنظیم از هر کدام را در نظر می‌گیریم. در جدول زیر نتایج نهایی مربوط به این دو آمده است:

مدل	صحت آموزش	صحت اعتبارسنجی	صحت آزمون	تعداد گام
Inception	۹۹٫۵۲٪	۹۳٫۹۸٪	۹۲٫۹۷٪	۱۲
LeNet	۷۹٫۲۱٪	۷۲٫۱۸٪	۷۵٫۷۸٪	۳۰

کاملا مشخص است که شبکه Inception صحت به مراتب بهتری داشته است. چنین چیزی مطابق با انتظار ماست؛ چراکه شبکه Inception پیش‌آموزش‌یافته قادر است ویژگی‌های بسیار مناسبی که از روی حجم عظیمی از داده‌های دیگر آموخته است را از روی داده‌های مسئله استخراج کند. طبیعتا در این شرایط و در اولین گام آموزش مدل می‌تواند با سرعت بیشتری به صحت‌های مناسبی برسد. در عین حال باید توجه کرد که اگر تعداد لایه‌های فریزشده زیاد باشد، صحت Inception از صحت LeNet بدتر می‌شود؛ این امر احتمالا به این دلیل است که باید به شبکه جدید Inception اجازه داد تا ویژگی‌ها را متناسب با مسئله تنظیم دقیق کند و ویژگی‌های اولیه کاملا منطبق بر مسئله نیست.

در مورد همگرایی می‌بینیم که مدل Inception با تعداد گام کمتری نسبت به LeNet همگرا شده است. این موضوع هم مورد انتظار بود؛ چراکه در LeNet به تعداد گام بیشتری نیاز داریم تا شبکه بتواند ویژگی‌های مناسب را پیدا کند؛ این درحالی است که شبکه Inception از ابتدا ویژگی‌هایی دارد که تا حد زیادی مناسب مسئله است. شایان ذکر است که لزوما تعداد گام کمتر Inception به معنای آموزش سریع‌تر از نظر زمان اجر نیست! مدل LeNet به دلیل شبکه بسیار سبک‌تری‌ای که دارد در زمان کمتری یک گام از داده را پردازش می‌کند. این موضوع به حدی بود که برای LeNet من از CPU و برای Inception از GPU استفاده کردم.

آخرین چیزی که باید بررسی شود، تعمیم‌پذیری است. با مکانیسم‌های مختلف Dropout و کالبک عملا هر دو مدل تعمیم‌پذیری مناسبی دارند. اما به نظرم شبکه Inception پتانسیل بیشتری برای بیش‌برازش دارد و اگر تدابیر مذکور نبود، احتمالا کمتر قابل تعمیم بود.