به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس شبکه‌های عصبی

استاد صفابخش

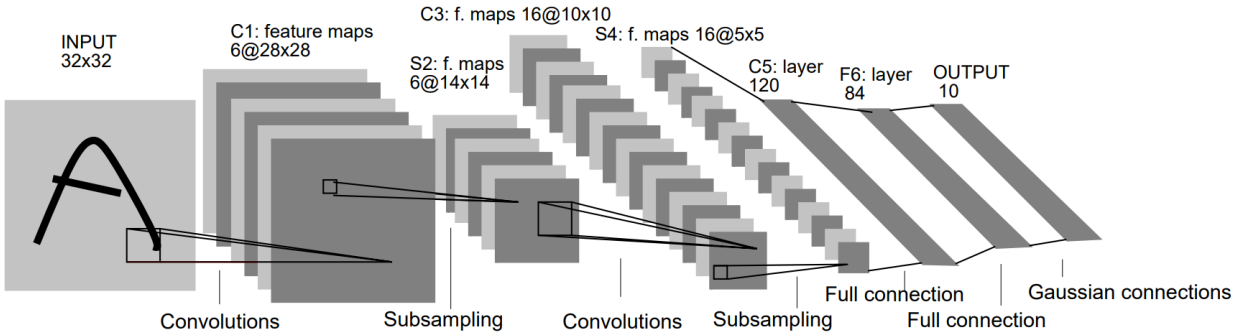
تمرین چهارم

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

---

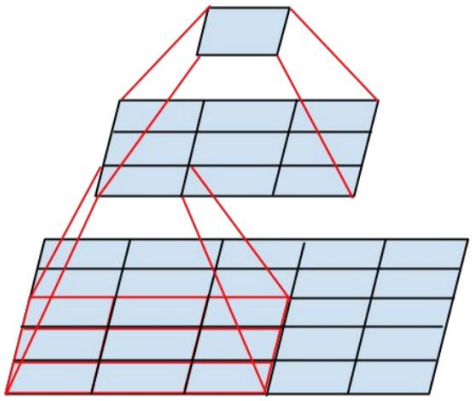
[1] https://towardsdatascience.com/understanding-and-implementing-lenet-5-cnn-architecture-deep-learning-a2d531ebc342

[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.

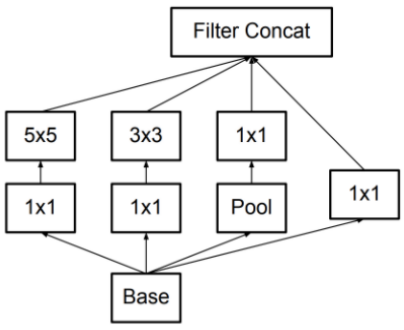
شبکه Inception-v3 تمرکز ویژه‌ای روی کاهش بار محساباتی داشته است. برای این امر از پنج تکنیک مهم استفاده کرده است که در ادامه آن را بررسی می‌کنم:

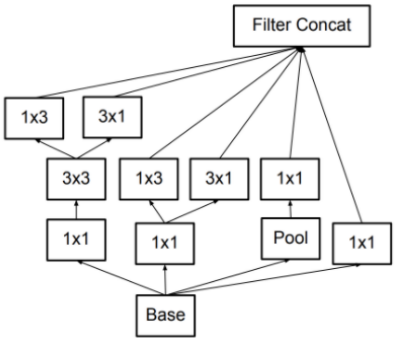
۱) کانوولوشن‌های تجزیه‌شده (Factorized Convolutions): با تجزیه کردن کانوولوشن‌ها پارامترها کاهش می‌یابد و کارایی حفظ می‌شود.



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.

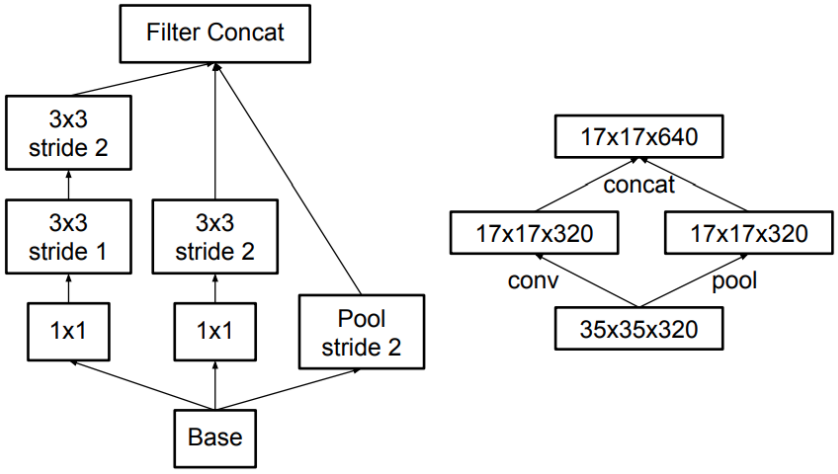
۳) کانوولوشن‌های نامتقارن (Asymmetric Convolutions): در Inception-v3 از کانوولوشن‌های نامتقارن کمک گرفته می‌شود که زمان آموزش را کاهش می‌دهد. یک کانوولوشن ۳×۳ را اگر بخواهیم با تکنیک کانوولوشن‌های کوچک‌تر بشکنیم نیاز به دو کانوولوشن ۲×۲ خواهیم داشت که چندان کاهش پارامتر ندارد ولی اگر از یک کانوولوشن 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.



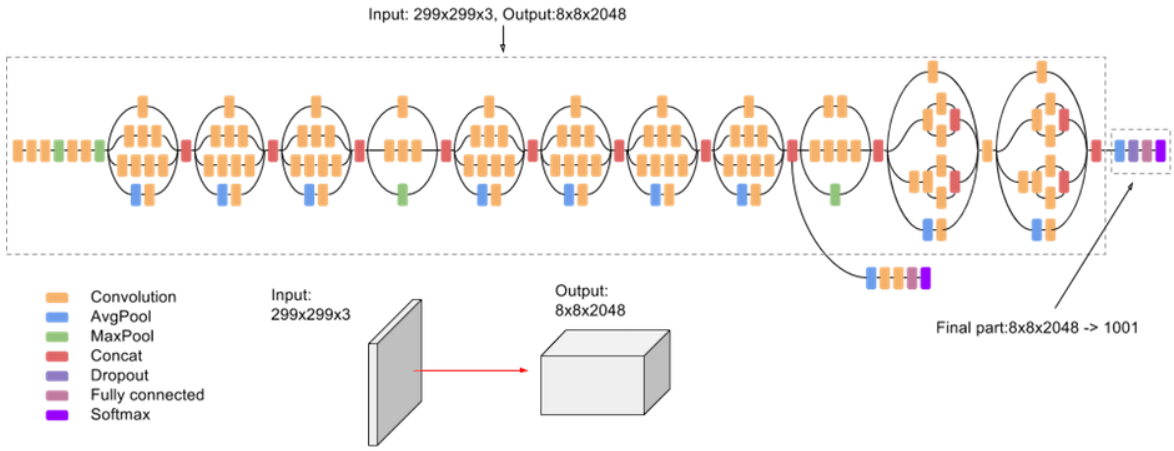


۴) دسته‌بند کمکی: دسته‌بند کمکی یک شبکه کانوولوشنی کوچک است که در میان لایه‌های شبکه اصلی در حین آموزش قرار می‌گیرد. تابع هزینه در زمان آموزش شامل تابع هزینه این زیرشبکه‌های کمکی هم می‌شود. این دسته‌بند کمکی در نقش یک منظم‌ساز برای شبکه عمل می‌کند. بدیهی است که چنین چیزی در LeNET-5 وجود نداشته است.

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

معماری شبکه در این قسمت بدین شکل است که ابتدا یک مدل Inception از پیش آموزش‌یافته درنظر گرفته می‌شود و بالاترین لایه آن حذف و به جای آن یک لایه Dropout برای جلوگیری از بیش‌برازش و یک لایه GlobalMaxPooling2D برای کاهش ویژگی‌ها و نهایتا یک لایه Dense با سه نورون متناسب با سه کلاس مسئله درنظر گرفته شده است.

برای آموزش هم در دو فاز کار پیش می‌رود. ابتدا کل لایه‌های Inception فریز می‌شود تا لایه‌های جدید فرصت آموزش داشته باشند و وزن‌های مناسبی پیدا کنند و سپس تعدادی از لایه‌های بالایی مدل Inception قابل آموزش می‌شود و لایه‌های پایینی همچنان فریز باقی می‌مانند.

در مجموع دو فاز آموزش حداکثر ۳۰ گام برای یادگیری درنظر گرفته شده است و با یک کالبک از نوع Early Stopping مشابه قسمت اول تمرین اگر خطای اعتبارسنجی در چهار گام بهتر نشود آموزش خاتمه می‌یابد.

برای انتخاب تعداد لایه فریزشده بهینه مقادیر ۰ تا ۳۱۱ (که برابر با کل تعداد لایه‌های Inception است) را در نظر گرفتیم. طبیعتا فاز اول آموزش برای تمام تنظیمات یکسان است ولی در عمل نتایج متفاوتی برای فاز اول بدست آمده است!

تعداد لایه بهینه فریزشده

در دو جدول زیر به ترتیب تعداد گام مورد نیاز برای آموزش در هر فاز و نتایج صحت بر روی مجموعه‌داده‌ها پس از هر فاز برای تنظیمات مختلف آورده شده است. با توجه به یکسان بودن شرایط برای فاز اول، میانگین پارامترهای حساب‌شده برای فاز اول در هر دو جدول آورده شده است:

تعداد لایه فریزشده	تعداد گام آموزش اولیه	تعداد گام آموزش ثانویه	مجموع تعداد گام
۰	۲۲	۸	۳۰ (بیشینه گام)
۵۰	۱۲	۵	۱۷
۱۰۰	۱۹	۵	۲۴
۲۰۰	۹	۵	۱۴
۳۱۱	۵	۰	۵
میانگین	۱۳/۴	-	-

تعداد لایه فریزشده	صحت مدل بعد از آموزش اولیه			صحت مدل بعد از آموزش ثانویه		
	آموزش	اعتبارسنجی	آزمون	آموزش	اعتبارسنجی	آزمون
۰	۷۳/۶۹٪	۵۵/۶۴٪	۵۸/۵۹٪	۵۱/۱۶٪	۴۵/۸۶٪	۴۹/۲۲٪
۵۰	۶۳/۶۴٪	۵۴/۱۴٪	۵۹/۳۸٪	۶۱/۹۹٪	۴۸/۸۷٪	۵۱/۵۶٪
۱۰۰	۶۹/۰۵٪	۵۶/۳۹٪	۶۳/۲۸٪	۷۲/۱۵٪	۵۷/۸۹٪	۶۸/۷۵٪
۲۰۰	۵۷/۷۴٪	۵۱/۱۳٪	۵۷/۰۳٪	۶۴/۲۲٪	۵۴/۱۴٪	۵۹/۳۸٪
۳۱۱	۴۲/۲۶٪	۴۴/۳۶٪	۳۵/۱۶٪	-		
میانگین	۶۱/۲۷٪	۵۲/۳۳٪	۵۴/۶۹٪	-		

در مجموع این جداول به نظر می‌رسد ۱۰۰ لایه فریزشده با صحت آزمون نهایی ۶۸/۷۵٪ و تعداد گام مجموع ۲۴ مناسب‌ترین تنظیم باشد. در همین جا باید توجه کرد که این نتایج تنها یک بار بدست آمده است و در همین یک بار می‌بینیم که در فاز اول آموزش مدل با ۱۰۰ لایه فریزشده با اینکه شرایط یکسانی با بقیه داشته است اما بهترین دقت آزمون یعنی ۶۳/۲۸٪ را کسب کرده است. این وزن‌های اولیه می‌تواند نقش سازنده‌ای در آموزش ثانویه داشته باشد و این احتمال وجود دارد که اگر وزن‌های اولیه تنظیمات مختلف یکسان باشد این مدل عقب بیافتد؛ با تمام این‌ها چون هدف یک بار اجرا بیشتر نیست تحلیل‌ها را با این فرض که نتایج در اجراهای بعد تغییر نمی‌کند، ادامه می‌دهیم.

در بررسی‌ای دیگر می‌بینیم که اگر تعداد لایه‌های فریزشده کم باشد، یعنی در فاز دوم آموزش هیچ لایه و یا ۵۰ لایه را فریز نگهداریم، مدل نمی‌تواند یادگیری مناسبی داشته باشد و صحت آن افت پیدا می‌کند؛ اما اگر تعداد لایه‌های فریزشده زیاد باشد، شبکه فرصت دارد تا بخشی از لایه‌های بالاتر را بهبود بدهد؛ این نکته را می‌توان از بهبود معیارهای دقت روی تمام مجموعه‌ها به ازای ۱۰۰ و ۲۰۰ لایه فریزشده دید. نهایتا اگر قرار باشد هیچ لایه‌ای را فریز نکنیم، نیاز به فاز دوم آموزش نیست و صحت‌های فاز اول را باید گزارش کرد. برای این حالت صحت خوب نیست. حتی میانگین صحت آزمایش‌های مختلف هم صحت بهتری از صحت نهایی دو تنظیم ۱۰۰ و ۲۰۰ لایه فریز حاصل نمی‌کند.

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
LeNet	۷۹/۲۱٪	۷۲/۱۸٪	۷۵/۷۸٪	۳۰
Inception	۷۲/۱۵٪	۵۷/۸۹٪	۶۸/۷۵٪	۲۴

با توجه به نتایج پیاده‌سازی من از دو مدل و برای این مسئله می‌توان گفت که LeNet به مراتب بهتر از Inception بوده است؛ صحت مدل LeNet از صحت Inception با اختلاف خوبی بیشتر است. نه تنها بهترین تنظیم که غالب تنظیم‌هایی که در LeNet بررسی شد از صحت بهترین تنظیم Inception بیشتر شده است؛ بدین ترتیب به نظر نمی‌رسد که از روی تصادف برتری دقت رخ داده باشد.

از نظر سرعت همگرایی اگرچه Inception از بهترین تنظیم LeNet تعداد گام کم‌تری داشته است ولی باید توجه کرد که LeNet در بهترین تنظیم و در گام اول به صحت ۷۰/۶۸٪ روی مجموعه اعتبارسنجی رسیده است ولی مدل ترجیح داده است گام‌های بیشتری برای بهبود کم را ادامه دهد. به علاوه آنکه تنظیمات متعددی از LeNet

در قسمت قبل ارائه شده است که با تعداد گام کم‌تر به دقت مناسب رسیده است. همه این‌ها را در کنار این نکته بگذارید که مدل Inception به دلیل شبکه سنگینی که دارد زمان اجرای زیادی را به ازای هر گام خود مصرف می‌کند؛ به طوری که من برای بدست آوردن نتایج شبکه LeNet از CPU و برای بدست آوردن نتایج شبکه Inception از GPU کمک گرفتم.

از نظر میزان تعمیم‌پذیری باتوجه به تکنیک‌های به کار برده‌شده نظیر لایه‌های کالبک روی مجموعه‌داده اعتبارسنجی و لایه‌های Dropout هر دو مدل مناسب است و صحت آزمون تفاوت جدی‌ای با صحت آموزش ندارد.

حال باید به دنبال علت بگردیم؛ در ابتدا من انتظار داشتم Inception حداقل از نظر صحت بهتر باشد ولی نهایتا مدل LeNet من بهتر بود. به نظر می‌رسد برای مجموعه‌داده موجود یک شبکه نسبتا سبک برای استخراج ویژگی‌ها و دسته‌بندی کافی بوده است. در این شرایط شبکه LeNet به خوبی توانسته است بخش اصلی دانش موجود در مجموعه‌داده را استخراج کند اما Inception برای این مجموعه‌داده سنگین بوده است. برای همین هم آموزش زمانبر شد و هم آنکه استعداد مدل برای بیش‌برازش زیاد بود و مجبور به استفاده از مکانیسم‌های متعدد برای جلوگیری از بیش‌برازش شدیم.

در کنار آن باید توجه کرد که ممکن است ویژگی‌های استخراج‌شده توسط Inception تطبیق دقیق با ویژگی‌های مورد استفاده برای این مسئله نداشته باشد. مدل جدید بر پایه Inception تنها می‌تواند با کمک یک لایه اصلی جدید خروجی متناسب با مسئله جدید را تولید کند که ممکن است کافی نباشد. این در حالی است که در LeNet تمام ویژگی‌ها، هرچند ساده، کاملا منطبق با مسئله تولید می‌شود. برای Inception من از دو فاز آموزش استفاده کردم با این امید که در فاز دوم وزن‌های مدل پایه Inception بهبود پیدا کند که می‌کند ولی این بهبود چندان جدی نیست و به سرعت بیش‌برازش رخ می‌دهد؛ تعداد گام‌های فاز دوم هم این مورد را تایید می‌کند.