1) الف)

ناپدید شدن گرادیان: هرگاه گرادیان ها در مشتق خیلی کوچک شوند و در backpropagation مقدارشان کمتر هم میشود و وزن هارا نمیتوانند به لایه های پایین انتقال دهند و در واقع یادگیری نداریم. تشخیص این مشکل :

- لایه های نزدیک خروجی بهتر تغییر می کنند نسبت به لایه های نزدیک به ورودی
- کند بودن بهتر شدن مدل با آموزش بیشتر آن
- وزن ها به صورت نمایی تغییر کرده و در آموزش به سمت صفر میروند

برای حل این مشکل، روش‌هایی مانند استفاده از توابع فعال‌سازی جدید مانندReLU استفاده از روش‌های نرمال‌سازی وزن‌ها (مانند نرمال‌سازی چرخشی وزن‌ها) مورد استفاده قرار می‌گیرند.

انفجار گرادیان: هرگاه گرادیان های در مشتق خیلی بزرگ شوند و به سمت بی نهایت بروند به وجود می آید. از دلایل ایجاد این مشکل میتوان به زیاد بودن مشتق گیری یا مقادیر بزرگ در داده ها نام برد. این اتفاق باعث تغییرات بزرگ در وزن ها و همگرا نبودن آموزش می شود. تشخیص این مشکل :

- با هر بار آموزش، تغییرات بزرگی در خطا ایجاد می شود.
- خطای NaN خواهیم داشت.
- مدل خوب آموزش نمیبیند و نتایج تغییر نمیکند.

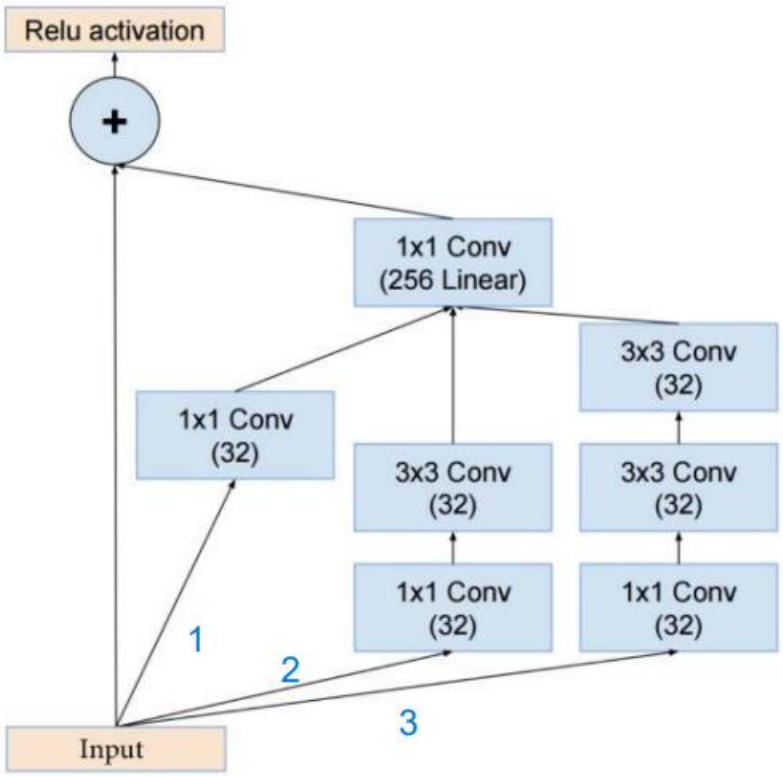
برای حل این مشکل، می‌توان از روش‌هایی مانند کاهش نرخ یادگیری (learning rate) و استفاده از روش‌های کاهش گرادیان مانند کاهش گرادیان جزئی یا Partial Gradient استفاده کرد.

ب)

با افزایش لایه های مدل بهینه سازی دچار مشکل و سخت شده و مدل آموزش و عملکرد خوبی نخواهد داشت. این موضوع باعث ایجاد ناپدید شدن گرادیان میشود.

برای حل این مشکل، این شبکه از چند بلوک متوالی residual تشکیل شده که هر کدام تعدادی لایه دارند. در این بلوک ها هربار ورودی هر بلوک بعلاوه ی خروجی آن به تابع فعال سازی داده شده و با این کار اگر ضرایب و بایاس‌ها در لایه‌های میانی به صفر برسند، با اضافه کردن مقدار قبلی به لایه‌های بعدی، از ناپدید شدن گرادیان جلوگیری می‌شود. بنابراین مشکل ناپدید شدن گرادیان حل می شود.

2. الف)



شاخه شماره یک: شامل یک فیلتر 1 × 1 است پس هر پیکسل خروجی فقط از پیکسل متناظر خود در ورودی تأثیر می‌پذیرد و فیلد آن برابر با 1 × 1 است.

شاخه شماره دو: شامل یک فیلتر 1×1 و یک فیلتر 3 × 3 است. فیلتر 1 × 1 تأثیری روی روی میدان نمیگذارد. اما فیلتر 3 × 3 باعث می‌شود میدان تأثیر این شاخه به ابعاد 3 × 3 برسد.

شاخه شماره سه: شامل دو فیلتر 3 × 3 است. استفاده از دو کانولوشن 3 × 3 باعث می‌شود میدان تأثیر این شاخه به ابعاد 5 × 5 برسد.

میدان تأثیر کلی ماژول : 5 × 5

تعداد پارامتر های هر لایه با استفاده از فرمول :

$((m \times n \times d) + 1) \times k$

- لایه کانولوشن 1 در 1 با عمق 32 : 128
- لایه کانولوشن 3 در 3 با عمق 32 : 9248
- لایه کانولوشن 3 در 3 با عمق 32 : 9248
- لایه کانولوشن 1 در 1 با عمق 96 : 128

- لایه کانولوشن 3 در 3 با عمق 32 : 9248
- لایه کانولوشن 1 در 1 با عمق 96 : 128
- لایه کانولوشن 1 در 1 با عمق 256 : 24,832

تعداد کل پارامترهای قابل آموزش در بلوک: 52960

ب)

A :

تعداد پارامترها = (تعداد کانال‌های ورودی) × (تعداد فیلترها) × (سایز کرنل) + (تعداد فیلترها)

اولین Conve2D :

تعداد پارامترها : 3 × 16 × (3 × 3) + 16 = 448

زمینه اثر : 3 در 3

دومین Conve2D :

تعداد پارامترها : 16 × 32 × (3 × 3) + 32 = 4640

زمینه اثر : 3 در 3

تعداد پارامترها در حالت A : 5088

زمینه اثر حالت A : 3 در 3

B :

تعداد پارامترها = (تعداد کانال‌های ورودی) × (تعداد فیلترها) × (اندازه هسته) × (اندازه هسته) + (تعداد فیلترها)

دومین LocallyConnected2D :

تعداد پارامترها = 16 × 32 × (3 × 3 × 3) + 32 = 13856

زمینه اثر : 3 در 3

تعداد پارامترها در حالت B : 13856

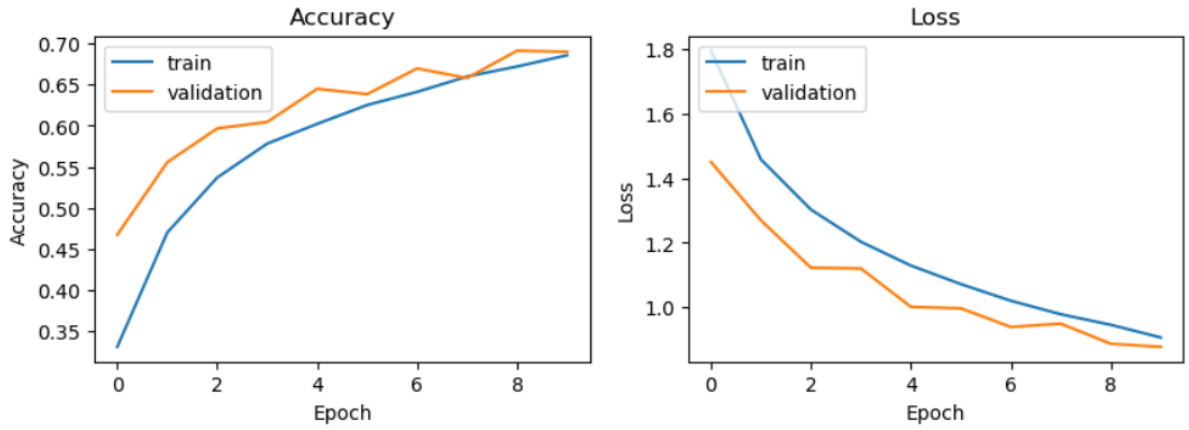
زمینه اثر حالت B : 3 در 3

برای تعیین زمینه تأثیر در هر یک از حالت‌ها، باید سایز کرنل و پدینگ لایه‌های کانولوشنال را در نظر بگیریم. در دو حالت A وB ، اندازه هسته (3، 3) است. یعنی هر پیکسل خروجی توسط یک همسایگی 3 در 3 پیکسلی از تصویر ورودی تأثیر می‌گیرند. و چون هر دو حالت از valid استفاده میکنند پس به تصویر پدینگ اضافه نشده و فقط پیکسل های داخل تصویر را در نظر میگیرد و زمینه تاثیر کوچک تر میشود.

3)

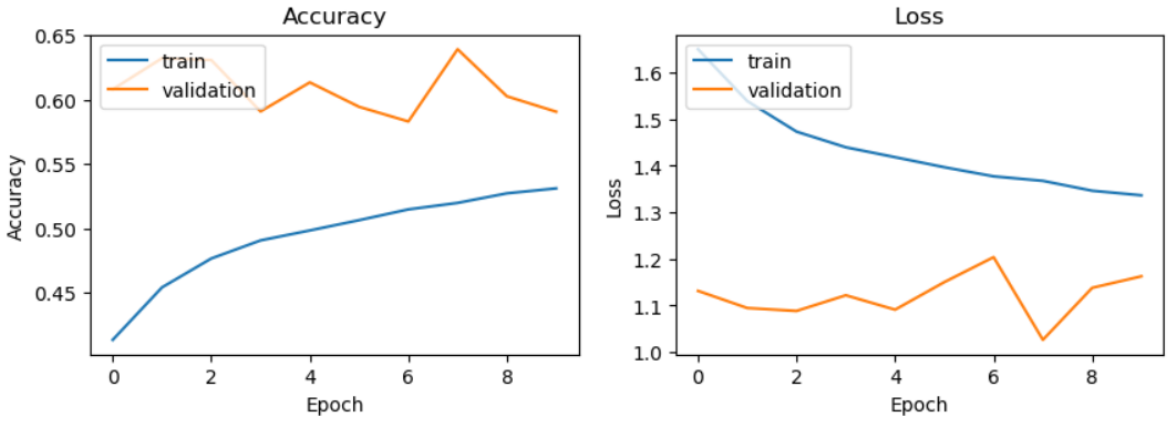
الف)

در این قسمت بعد از لود کردن داده مورد نظر آن هارا نرمال سازی کرده و بعد به صورت one hot encoding تبدیل میکنیم. در ادامه مدل را تعریف میکنیم و آن را کامپایل و train میکنیم. در خروجی مدل میتوان مشاهده کرد که accuracy داده validation از train بیشتر است که یعنی مدل overfit شده.



ب)

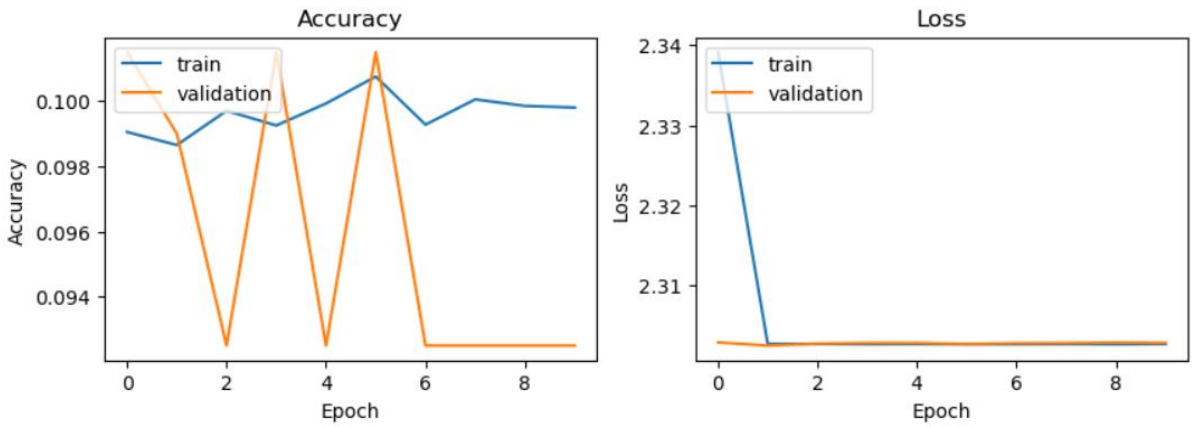
در این قسمت با استفاده از دو لایه RandomFlip و RandomRotation که باعث چرخش و fllip تصویر میشوند، مدل جدیدی میسازیم که این دولایه در ابتدای آن قرار دارند و باعث میشوند مدل داده های جدیدی ببیند. در ادامه مدل را تعریف میکنیم و آن را کامپایل و train میکنیم. در خروجی مدل میتوان مشاهده کرد که accuracy داده validation از train بیشتر است که یعنی مدل overfit شده.



ج)

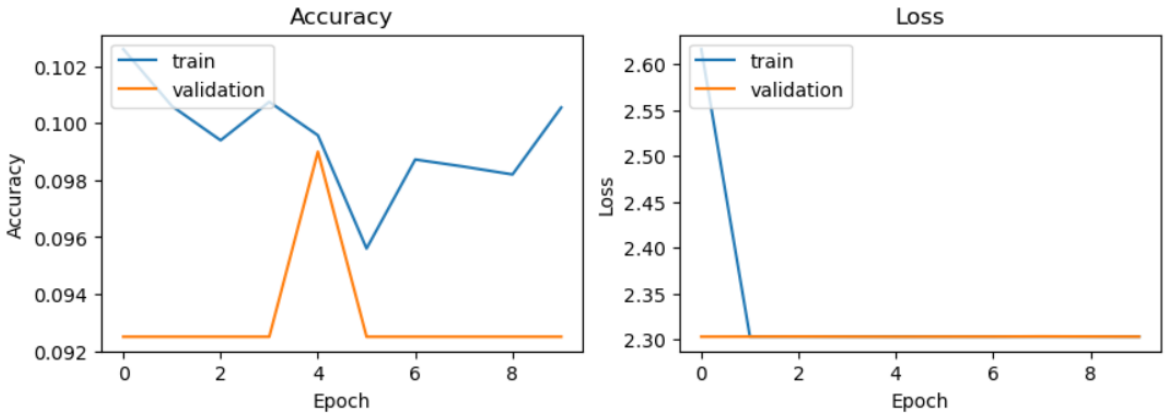
در هر دو نمودار overfitting رخ داده که در حالت ب بسیار بیشتر است و با سرعت کمتری مقدار loss داده train کم میشود.

د) برای انجام این کار از ImageDataGenerator استفاده کردیم زیرا روش های دیگر به دلیل بالا بودن حجم دیتا کرش میکرد. این روش یک instance از داده هارا میسازد. در ادامه مدل ResNet50 را لود میکنیم. برای ساخت مدل ابتدا ورودی آن و بعد لایه های مدل را قرار میدهیم و در ادامه از GlobalAveragePooling2D و یک لایه classifier قرار میدهیم و مدل را میسازیم. در ادامه مدل را تعریف میکنیم و آن را کامپایل و train میکنیم. در خروجی مبینیم که underfitting داریم چون خطای مدل روی داده‌های آموزش به طور پیوسته کاهش می‌یابد اما سپس خطای آن روی داده‌های validation تغییر زیادی نمیکند. که ممکن است مشکل از ناپدید شدن گرادیان در مدل باشد.



ه)

برای این قسمت مانند قسمت ج ابتدا داده هارا تبدیل به 224 در 224 میکنیم سپس با استفاده از layers سه لایه ی ابتدا داده را لوود میکنیم و مانند قسمت بالا مدل را تعریف میکنیم و آن را کامپایل و train میکنیم. در اینجا چون loss روی داده های val تغییر زیادی نمیکند احتمالا مشکل ناپدید شدن گرادیان را داریم و درواقع مدل overfit شده .



4) الف)

گام تعداد پیکسل‌هایی است که فیلتر در هر مرحله به طول و عرض ورودی حرکت می‌کند. در واقع، گام نشان می‌دهد فاصله بین محاسبات کانولوشنی چقدر است.

تفاوت stride و pooling :

در روش Stride، اطلاعات بین هر مرحله از کانولوشن منتقل نمی‌شوند و تنها محاسبات کانولوشنی انجام می‌شود.در روش Pooling، مقادیر میانگین یا حداکثر در هر مرحله از کانولوشن گرفته می‌شوند و اطلاعات فشرده‌تری از تصویر حاصل می‌شود. با افزایش مقدار stride، ابعاد خروجی کانولوشن و تعداد پارامترها نیز کاهش می‌یابد. که باعث افزایش سرعت آموزش مدل و کاهش پیچیدگی محاسبات میشود. البته با افزایشstride ، اطلاعات موجود در تصویر کاهش می‌یابد و ممکن است بخشی از اطلاعات مهم را از دست بدهیم. اما استفاده ازpooling ، به کاهش ابعاد تصویر و تجمیع اطلاعات کمک می‌کند و باعث می‌شود مدل به تغییرات محلی مقاومت بیشتری داشته باشد. و همچنین تعداد پارامترها را تغییر نمی‌دهد.

ب)**1)از** ReLU استفاده میکنیم چون برای لایه‌های میانی ساده است **و فقط مقادیر مثبت را انتقال می‌دهد و مقادیر منفی را به صفر تبدیل می‌کند،** و استفاده از Softmax در لایه آخر برای classification مناسب است.

ب)**2**)) از Cross Entropy استفاده میکنیم زیرا این تابع برای مسائلclassification مناسب است. **با استفاده از این تابع، می‌توانیم اختلاف بین توزیع احتمالی خروجی شبکه و توزیع احتمالی مورد انتظار (برچسب واقعی) را اندازه‌گیری کنیم**. این تابع احتمال درست را تشدید میکند و بعث بهتر شدن عملکرد شبکه میشود. همچنین این تابع تفاوت در توزیع داده های سالم و معیوب را در نظر میگیرد و نیازی به وزن دادن به داده ها ندارد.

ب)3)

دقت (Precision):

دقت نسبت مثبت‌های واقعی را به مجموع مثبت‌های پیش‌بینی شده محاسبه می‌کند. به عبارت دیگر، دقت نشان می‌دهد که از مثبت‌های پیش‌بینی شده توسط مدل، چه تعداد واقعاً مثبت هستند.

دقت = تعداد مثبت‌های واقعی / (تعداد مثبت‌های واقعی + تعداد مثبت‌های غلط)

بازخوانی (Recall):

بازخوانی نسبت مثبت‌های واقعی را به مجموع مثبت‌های واقعی و منفی‌های غلط محاسبه می‌کند. به عبارت دیگر، بازخوانی نشان می‌دهد که چه تعداد از مثبت‌های واقعی به درستی توسط مدل شناسایی شده‌اند.
بازخوانی = تعداد مثبت‌های واقعی / (تعداد مثبت‌های واقعی + تعداد منفی‌های غلط)

بهتر است از precision استفاده کنیم چون **Precision نسبت معیوب‌های شناسایی شده درست به کل محصولاتی است که مدل به عنوان معیوب شناسایی کرده است. با افزایش آن، تعداد محصولات سالمی که اشتباهاً به عنوان معیوب شناسایی می‌شوند کاهش می‌یابد. همچنین میتوان تعداد محصولات سالمی که به اشتباه به عنوان معیوب شناسایی می‌شوند را قبول کنیم تا از رسیدن محصولات معیوب جلوگیری شود. در این صورت، با افزایشPrecision ، احتمال رسیدن محصولات معیوب به مشتری کاهش می‌یابد.**

ج)1) مناسب نیستند چون متن ها میتوانند اندازه های متفاوتی داشته باشند و CNN ها برای ورودی با سایز های محدود طراحی شده اند، شبکه برای پردازش کلمات و جملات مشکل پیدا میکند.

همچنین در این شبکه ها مسئله ی معنای کلمه نسبت به جمله و موقعیت مکانی واژه، مورد توجه قرار نمیگیرد.

ج)2) می توان در تشخیص گوینده از روی صوت، شبکه‌های عصبی کانولوشنی با موفقیت استفاده شوند. با تبدیل صدا به شکل‌های خاصی مانند مجموعه ویژگی‌های تبدیل فرکانس-فرکانس اطلاعات صوتی به صورت یک بردار دو بعدی تبدیل می‌شوند که می‌تواند به عنوان ورودی برای مدل CNN استفاده شود.

ج)3) مناسب نیست چون رابطه فضایی بین ورودی‌ها نداریم. زیرا در این مورد هیچ ارث‌بری وجود ندارد، و عملکرد آن‌ها به صورت کلی الگوی خاصی را به وجود نمی‌آورد.

1. **پارامترهای زیاد: شبکه‌های کانولوشنی به پارامترهای قابل تنظیمی دارند که نیاز به تنظیم و آموزش مناسب دارند. مدیریت و بهینه‌سازی این پارامترها ممکن است زمان‌بر و پیچیده باشد.**
2. **زمان و هزینه محاسباتی: انجام کانولوشن در شبکه‌های کانولوشنی زمان بر و پرهزینه است.**
3. **نیاز به حجم بالای داده: برای آموزش این شبکه ها، نیاز به حجم بالایی از داده‌های آموزشی داریم. و در غیر این صورت مدل عملکرد خوبی نخواهد داشت.**
4. نیاز به منابع پردازشی قوی: شبکه‌های کانولوشنی به منابع پردازشی قوی نیاز دارند، به خصوص اگر ساختار شبکه بسیار پیچیده و عمیق باشد.
5. مشکل در تشخیص وابستگی‌های طولانی: برخی متون، به خصوص متون طولانی، وابستگی‌های طولانی بین کلمات را دارند. شبکه‌های کانولوشنی معمولاً به‌خوبی نمی‌توانند این نوع وابستگی‌ها را درک کنند و از اطلاعات کلمات دور در جمله یا متن کمتر بهره‌برداری کنند.

5)ب)

فاوت اصلی بین دو تابع Loss به نام‌های "Binary Cross Entropy (BCE)" و "Intersection over Union (IoU)"در روش اندازه‌گیری خطا و هدفی که به آن‌ها می‌دهند است. البته این تفاوت در مسائلی که این توابع برای آن‌ها استفاده می‌شوند نیز نقش مهمی دارد.

BCE Loss یک روش رایج در مسائل دسته‌بندی دودویی (binary classification) است. این تابع معمولاً برای مسائلی که هدف آن‌ها تشخیص دسته‌بندی صحیح برچسب یک نمونه استفاده می‌شود Loss BCE .بر اساس تابع log-loss عمل می‌کند و احتمالاتی که توسط مدل برای هر دسته تولید می‌شوند را با برچسب واقعی مقایسه می‌کند. هدف این تابع کاهش خطا و افزایش صحت دسته‌بندی است.

IoU Loss در مسائل مربوط به تشخیص و دسته‌بندی اشیاء در تصاویر استفاده می‌شود. معمولاً در مسائل شناسایی شی (object detection) و تشخیص ماسک یا سگمانتاسیون تصاویر مورد استفاده قرار می‌گیرد .
IoUمقدار همپوشانی (overlap) بین منطقه پیش‌بینی شده توسط مدل و منطقه واقعی شی را اندازه‌گیری می‌کند.تلاش می‌کند مقدار همپوشانی را افزایش داده و از این طریق بهبود نتایج دسته‌بندی و تشخیص شی در تصویر را دست‌یابی کند.
به طور خلاصه، تفاوت اصلی میان Loss BCE و Loss IoU در روش اندازه‌گیری خطا و هدفی که به آن‌ها می‌دهند قرار دارد. Loss BCE بر روی احتمالات دسته‌بندی تمرکز می‌کند و سعی می‌کند صحت دسته‌بندی را افزایش دهد، در حالی که Loss IoU بر روی همپوشانی منطقه‌ها تمرکز می‌کند و بهبود دقت تشخیص و تمایز شی را هدف می‌گیرد.