تمرین سوم شبکه عصبی

محمدرضا صیدگر-401422215

Exercise 1)

بسياري از معماري هاي طراحي شده در راستاي استفاده از توان محاسباتي موجود به صورت بهينه مي باشند.حال مي‌خواهيم بدانيم اگر ميزان توان محاسباتي كاهش يا افزايش پيدا كند يا اگر بخواهيم يك شبكه زودتر آموزش داده شود، به چه طريق مي‌توان شبكه مورد نظر را scale كنيم.

در شبكه هاي كانولوشني سه روش براي افزايش دقت استفاده ميشود:

افزايش عمق شبكه، ارتفاع شبكه و همچنين افزايش رزولوشن ورودي ميباشد كه افزايش هر كدامم از اين ويژگيها ميتواند باعث بهبود عملكرد شبكه شود.

اين سه ويژگي با يكديگر ارتباط مستقيمي دارند، به اين صورت كه با افزايش رزولوشن، ويژگي بيشتري براي بررسي وجود دارد پس شبكه ميتواند عمق بيشتري داشته باشد.مي توان به effecientNet به عنوان يك نوع جستجو براي كارآمد ترين شبكه ي عصبي با توجه به ميزان توان محاسباتي نگاه كرد.

با توجه به اينكه دستگاه هاي مختلف از توان پردازشي متفاوتي بهره مند هستند مي‌خواهيم شيوه اي داشته باشيم كه با توجه به دستگاه در دسترس و توانايي پردازش موجود چگونه يك شبكه را Scale كنيم.اگر بخواهيم شبكه ي ما سريع‌تر آموزش ببيند و كمي كاهش دقت در نتايج شبكه مسئله ي خيلي مهمي نباشد ميتوان از اين روش استفاده نمود.

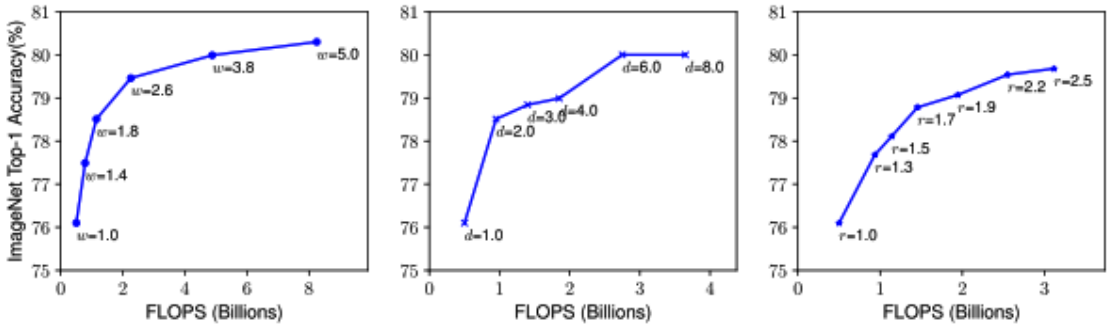
بنابراين Efficient-net يك راهي براي به دست آوردن بهينه ترين ميزان براي scale up كردن با توجه به شرايط موجود مي باشد.

با در نظر گرفتن يك baseline به صورت كلي سه روش براي scaling شبكه براي به دست آوردن دقت بهينه وجود دارد.

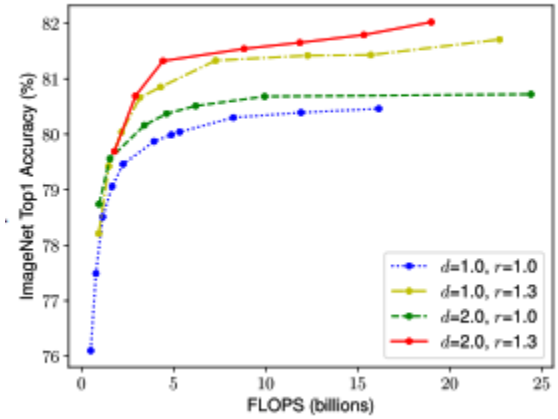
روش اول افزايش عمق: اين روش بيشترين استفاده را در معماري هاي موجود تا كنون داشته است. منظور از افزايش عمق، افزايش تعداد لايه هاي يك شبكه مي باشد.

روش دوم افزايش عرض: از اين روش نسبت به عمق كمتر استفاده مي‌شود. منظور از عرض نيز مقدار كانال هاي يك شبكه مي باشد.

روش سوم افزايش resolution عكس ورودي: از اين روش نيز گاها در مقالات مشاهده شده كه استفاده شده است.
در اينجا روشي معرفي شده است كه ابتدا يك مدل baseline در نظر گرفته و سعي مي كند با افزايش سه بعد طول، عرض و رزولوشن در مدل baseline بهبود ايجاد نمايد.
مسئله كه وجود دارد براي پيدا كردن طول، عرض و رزولوشن مناسب فضايي كه بايد جستجو نماييم بسيار بزرگ ميشود بنابراين ما براي هر لايه در نظر ميگيريم كه به صورت يكنواخت با يك ضريب ثابت رشد كنند.
يكي از مشكلاتي كه scale up كردن شبكه مي تواند ايجاد كند، مشكل vanishing gradient شود كه با skip connection قابل حل مي باشد.



همان طور كه در شكل مشاهده مي‌شود، افزايش هر يك از اين ابعاد باعث افزايش دقت مي‌شود، اما با رسيدن به دقت 80% دقت اشباع شده و افزايشي صورت نمي‌گيرد.



شكل بالا نشان مي‌دهد كه مي‌توان با تركيب اين ابعاد با يكديگر به ميزان دقت بيشتري دست پيدا نمود. به عبارت ديگر مي‌توان بيان نمود كه اين ابعاد ازيكديگر مستقل نيستند.

در ادامه به معرفي compound scaling method مي پردازيم. اين مسئله به صورت يك مسئله بهينه سازي ميباشد، كه به صورت فرمول زير مي‌توان مشاهده نمود. آلفا، بتا و گاما به صورت ثابت در نظر گرفته ميشوند كه با يك grid search مي توان آن ها را به دست آورد. و اگر بتوانيم هزينه محاسباتي را افزايش دهيم با يك ضريب ثابت، مقدار phi را تغيير مي دهيم.

Depth: d = alpha^phi

Width: w = betha^phi

Resolution: r = gama^phi

s.t. alpha.betha^2.gama^2 = 2

alpha>=1, betha>=1, gamma>=1

في نيز مي تواند بسته به مقدار منابع محاسباتي تغيير داده شود.ميزان توان محاسباتي با FLOPS سنجيده مي شود كه نشان دهنده‌ي ميزان تعداد عمليات محاسباتي شناور كه در يك ثانيه قابل انجام است، مي باشد. محدوديت alpha.betha^2.gama^2 = 2 به اين معني مي‌باشد كه ميزان FLOPS ها به ميزان phi^2 مي‌تواند افزايش يابد.
معماري كه به عنوان baseline در نظر گرفته مي شود اهميت بسياري دارد، زيرا efficient net معماري baseline را تغييري نمي دهد و صرفا آن را scale مي‌كند. efficientNet داراي هفت ورژن مختلف مي باشد. كه ابتدا B0 را با استفاده از روش و معماري معرفي شده به دست مي آورده و آن را به عنوان baseline در نظر گرفته و سپس با دو گام ورژن هايي ديگر (B1-B7) را به دست مي آوريم.
گام اول:
ابتدا phi را ثابت و برابر يك در نظر ميگيريم، فرض مي كنيم كه دو برابر توان محاسباتي در اختيار داريم و با استفاده از يك beta ، alpha ، grid search و gamma مناسب را پيدا ميكنيم. با در نظر گرفتن شرط ذكر شده alpha.betha^2.gama^2 = 2 .

گام دوم:

در اين قسمت آلفا، بتا و گاما را ثابت در نظر گرفته و با در نظر گرفتن phi هاي مختلف baseline يعني B0 را scale up مي نماييم.
سوالي كه در اين جا مي توان مطرح نمود اين است كه آيا مي توان بدون در نظر گرفتن baseline و بر روي شبكه هاي بزرگ تر آلفا، بتا و گاماي مناسب را جستجو نمود؟
امكان جستجو وجود دارد، ولي از لحاظ محاسباتي بسيار گران مي شود. و اين مسئله با به دست آوردن مقادير مناسب بر روي baseline و سپس scale كردن آن به مدل هاي بزرگ تر به دست آمده است.

Exercise 2)

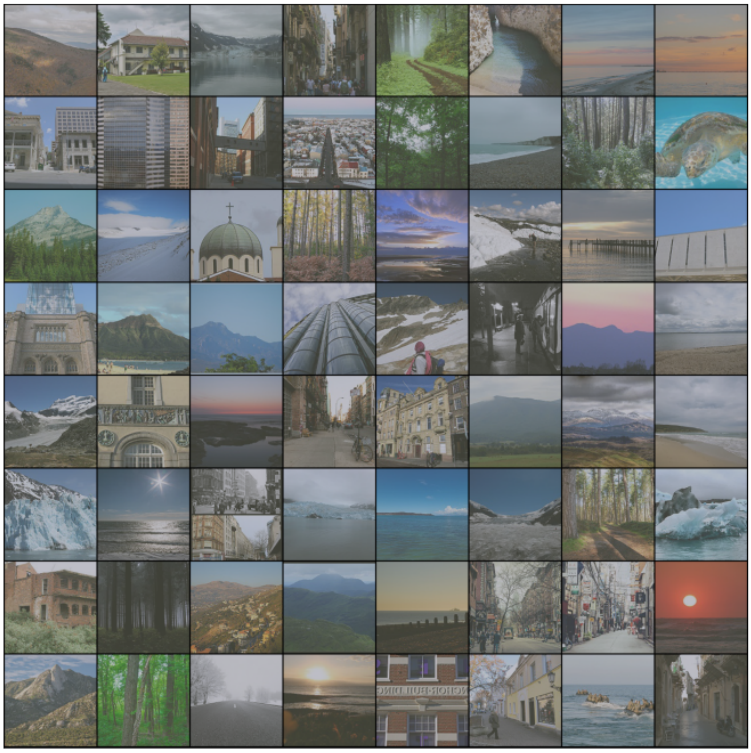
در شبکه های CNN فیلتر های کانولوشن برای استخراج ویژگی های تصویر استفاده می شوند. اندازه فیلترها یکی از عوامل تعیین کننده در ویژگی‌های استخراج شده از تصویر است. پیش از معرفی ماژول های Inception طراحان شبکه باید با توجه به تصویر اندازه فیلترها را تعیین می‌کردند.
ایده ماژول Inception استفاده از فیلترهای با ابعاد مختلف به طور همزمان است. به این صورت که چند فلیتر با ابعاد مختلف (فیلترهای کانولوشنی و pooling) روی ورودی اعمال می‌شوند. سپس خروجی آنها کنار یکدیگر قرار می‌گیرد (concat می شوند). در این حالت feature mapها مشتمل بر ویژگی های مختلف خواهند بود. هر ماژول شامل چند عملیات کانولوشنی و pooling موازی است. شکل زیر به صورت بلوکی یک ماژول Inception دلخواه را نشان می دهد.
مشکل این شبکه ها چه بود؟
در نگاه اول تعداد زیادی عملیات موازی، که یعنی هزینه محاسباتی زیاد! طراحان ماژول پس از این ضمن تلاش برای افزایش دقت، به اصلاحاتی برای کاهش هزینه محاسباتی نیز پرداختند.
نسخه V4 محصول توسعه TensorFlow بوده است. این نسخه تغییری در ساختارهای ماژول ارائه نداده. در واقع قبل از TensorFlowطراحان برای پایداری شبکه و همچنین رعایت قیدهای حافظه محافظه کارانه شبکه را طراحی می کردند با کاهش محدودیتهای طراحی به وسیله TensorFlow طراحان شبکه های عمیق تر با ماژولهایی با فیلترهای بیشتری را پیشنهاد دادند.
Inception-ResNet: Resnet با معرفی Residual blockها تاثیر زیادی بر دنیای شبکه عصبی گذاشت. به همین جهت طراحان Inception نیز سعی کردند از ویژگی مسیر

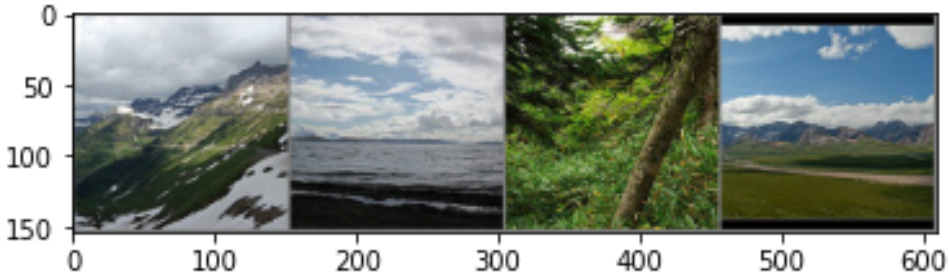
Residual در ساختار خود بهره ببرند. به این صورت که مجموعه فیلترها inceptionی در مسیر Residual قرار خواهند گرفت.
با به وجود آوردن این نوع ماژول مشکلی که در خود شبکه ResNet نیز به آن اشاره شده بود پدید آمد. اگر تعداد فیلترها از ۱۰۰۰ بیشتر می بود خروجی لایه ها بعد از چند هزار تکرار صفر میشد. راه حل پیشنهادی استفاده از Activation scaling در مسیر Residual بوده در واقع ضریبی از بلوک Inception به مسیر مستقیم اضافه شود.
عمق و فیلترهای ماژولها در Inception-ResNet به گونه ای انتخاب شد که از نظر حجم محاسباتی مشابه V4 باشد. هر دو شبکه دقتی مشابه یکدیگر داشتند اما سرعت همگرایی Inception-ResNet به وضوح بالاتر است.

Exercise 3)

گزارش پروژه:

در این پروژه مسئله ما یک مسئله طبقه بندی یا classification است بر روی داده های Intel Image Classification که داده های عکسی هستند با کانال های رنگی rgb.
ما باید این عکس ها را به 6 دسته مختلف اختصاص دهیم که شامل forest ، buildings ، sea ، mountain ، glacier و street است.
بعضی از عکس ها را چاپ کردیم تا ببینیم چه عکسایی هستند:





کلاس های عکسای بالا به ترتیب از راست به چپ glacier,sea,forest,mountain هستند.

برای پیشبینی کلاس های این مدل از داده ها که عکس هستند باید از شبکه های عصبی کانولوشن استفاده کنیم. در صورت سوال گفته شده است که از شبکه های ResNet استفاده کنیم برای طبقه بندی داده ها.

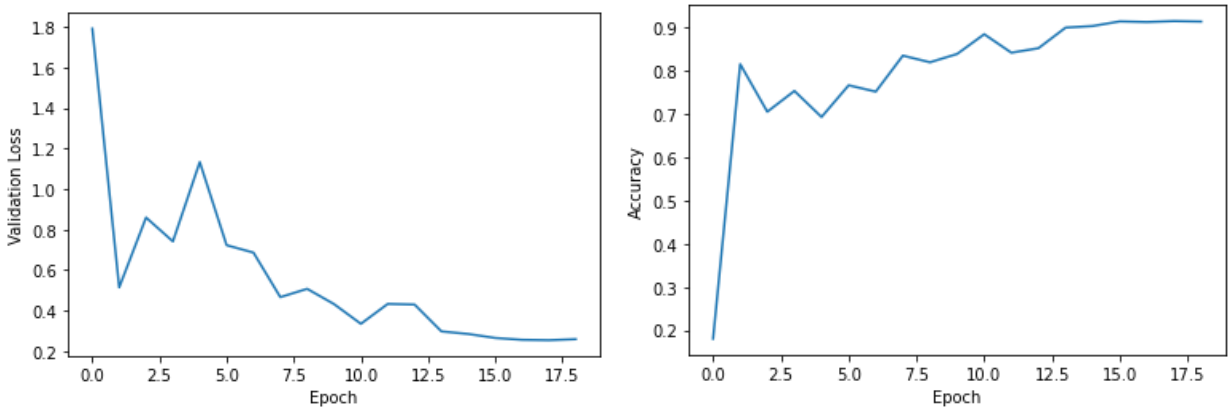
مدل ما دارای بلاک ها کانولوشن است که درون هرکدام از این بلاک ها کانولوشن 2 بعدی است ، لایه batch normalization و یک تابع فعالسازی relu و همینطور در بعضی مواقع در صورت نیاز یک لایه maxpool 2 بعدی هم خواهد داشت.

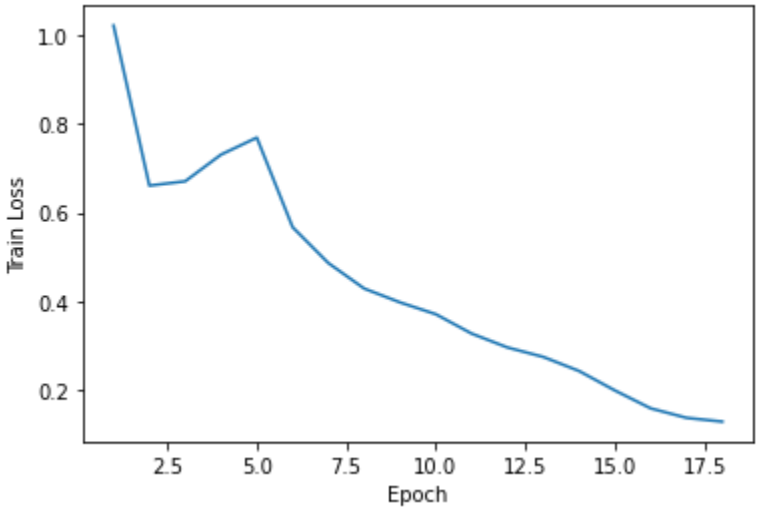
شبکه ما در اول یک بلاک کانولوشن دارد که ورودی 3 کانال رنگی می گیرد و خروجی 64 کانال میدهد سپس یک بلاک دیگر با maxpool که 64 میگیرد و 128 میدهد و لایه بعدی یک لایه resnet است که دارای 2 بلاک کانولوشن با ورودی و خروجی 128 که

خروجی این لایه علاوه بر خروجی این لایه ورودی های این لایه را هم به لایه بعدی انتقال میدهد که این همان چیزی است که در resnet ما داریم. در ادامه دوباره یک بلاک کانولوشن با ورودی 128 و خروجی 256 و یک بلاک کانولوشن دیگر با ورودی 256 و حروجی 512 و لایه بعدی باز یک لایه resnet است با 2 بلاک کانولوشن با ورودی و خروجی 512 و درنهایت یک لایه classifier که ورودی 512 و خروجی ان 6 است که این لایه دارای لایه های maxpool2 و flatten و linear است.
مدل را با optimizer adam و0.01 max learning rate و 18 ایپاک پیش رفتیم و نتایج زیر را روی داده های train , valid داشتیم:

Train loss: 0.12, validation loss: 0.25, validation accuracy: 0.91

نمودار های زیر روند آموزش رو به خوبی در طول این 18 ایپاک نشان می دهند:





در نهایت مدل را روی داده های تست هم ارزیابی کردیم که نتایج زیر را به همراه داشت:

Test loss: 0.27 , Test accuracy: 0.90

که نتایج بسیار خوبی می باشد.