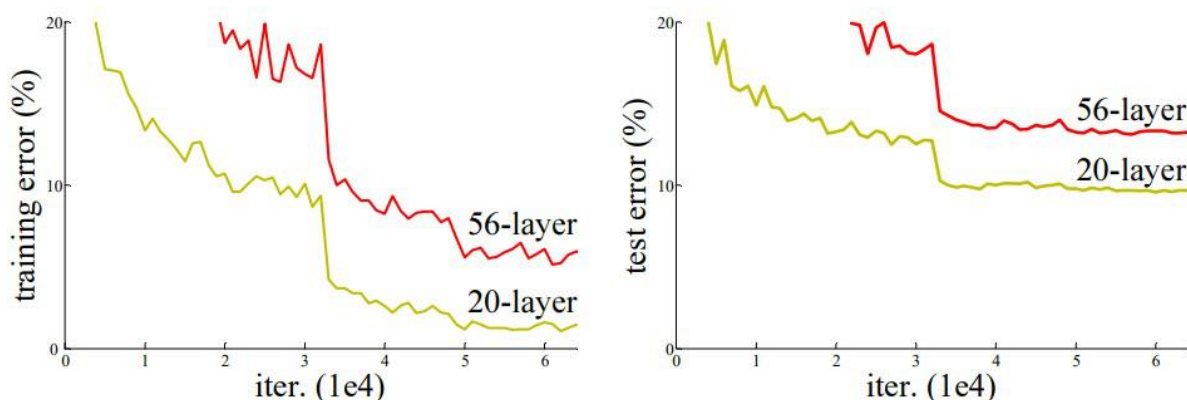


پروژه سوم – شبکه Residual Network

بعد از اولین معماری مبتنی بر شبکه عصبی کانولوشنی، الکس نت (AlexNet)، که برنده مسابقه ImageNet 2012 بود، سایر معماری های برنده از لایه های بیشتری استفاده می کند تا میزان خطا را کاهش دهد. با افزایش تعداد لایه ها در شبکه های عصبی، یک مشکل مشترک در یادگیری عمیق وجود دارد که به آن ناپدید شدن یا انفجار گرادیان (Vanishing/Exploding Gradient) می گویند. این باعث می شود که گرادیان خیلی نزدیک به صفر یا خیلی بزرگ شود. بنابراین وقتی تعداد لایه ها را افزایش می دهیم، میزان خطای آموزش و آزمون نیز افزایش می یابد.

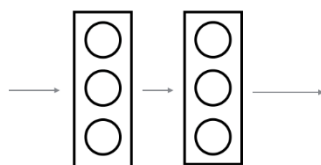


در نمودار فوق، می توان مشاهده کرد که شبکه کانولوشنی با ۵۶ لایه نسبت به معماری ۲۰ لایه CNN میزان خطای بیشتری را در هر دو مجموعه داده آموزش و آزمون ارائه می دهد. اگر این مساله نتیجه بیش برآزش بود، باید خطای آموزش کمتری در شبکه ۵۶ لایه داشته باشیم اما خطای آموزش نیز بالاتر است. ResNet که در سال ۲۰۱۵ توسط محققان Microsoft Research ارائه شد، معماری جدیدی را با نام Residual Network معرفی کرد.

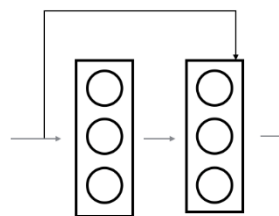
بلوک Residual

به منظور حل مشکل شیب ناپدید شدن و انفجار گرادیان، این معماری مفهومی را به نام شبکه باقیمانده (Residual Network) معرفی کرد. در این شبکه ما از تکنیکی به نام اتصال جست و خیز (Skip Connections) استفاده می کنیم. اتصال جست و خیز از چند لایه پرش می کند و مستقیماً به خروجی متصل می شود. مزیت اضافه کردن این نوع اتصال جست و خیز این است که اگر هر لایه به عملکرد معماری آسیب برساند، با قاعده گذاری از آن عبور می شود. بنابراین، این امر منجر به آموزش شبکه عصبی بسیار عمیق و بدون مشکلات ناشی از ناپدید شدن / انفجار شیب می شود.

without skip connection



with skip connection



تصویر سمت چپ "مسیر اصلی" را از طریق شبکه نشان می دهد. تصویر سمت راست یک میانبر به مسیر اصلی اضافه می کند. با قرار دادن این بلوک ها روی هم ، می توانید یک شبکه بسیار عمیق تشکیل دهید.

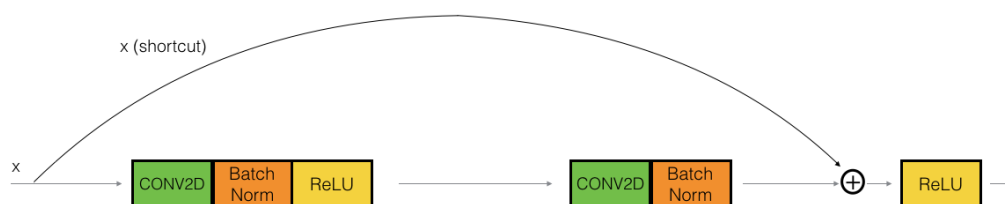
در ویدیوهای درس گفته شد که داشتن بلوک های ResNet با مسیر میانبر یادگیری تابع همانی (Identity Function) را برای بلوک ها بسیار آسان می کند. در واقع شما می توانید بلوکهای اضافی ResNet را با کمترین خطر برای آموزش شبکه، در شبکه قرار دهید. همچنین شواهدی وجود دارد که نشان می دهد این مساله در برطرف کردن محو شدن یا انفجار گرادیان موثر تر از اتصالات میانبر است.

دو نوع بلوک اصلی در ResNet مورد استفاده قرار می گیرند که استفاده از آنها به یکسان بودن یا نبودن ابعاد ورودی و خروجی بستگی دارد. هدف این پروژه پیاده سازی این دو بلوک است: "بلوک همانی" و "بلوک کانولوشن".

توجه کنید که پیش از شروع تمرین ها محتوای پوشه Requirements را در مسیر پروژه کپی کنید.

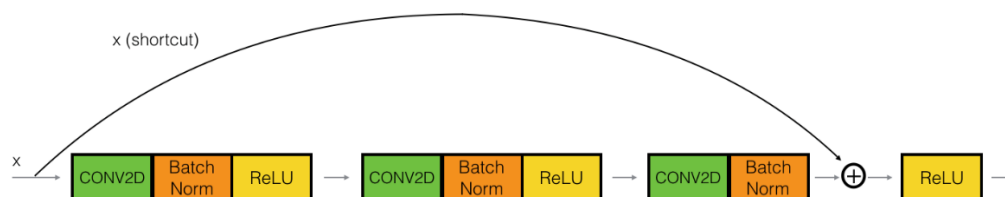
تمرین اول: بلوک همانی (Identity Block)

بلوک همانی در شبکه ResNet هنگامی مورد استفاده قرار میگیرد که ابعاد ورودی و خروجی یکسان باشند. مراحل مختلف این بلوک در نمودار زیر آمده است:



مسیر فوقانی "مسیر میانبر" است. مسیر پایینی "مسیر اصلی" است. در این نمودار ، به مراحل Conv2D و ReLU در هر لایه توجه کنید. برای سرعت بخشیدن به آموزش ، مرحله BatchNorm اضافه شده است.

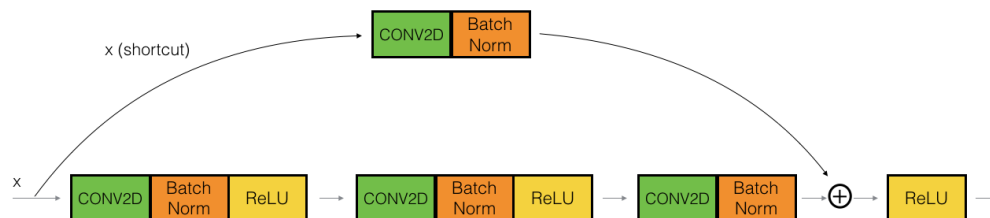
در این تمرین ، شما در واقع یک نسخه قدرتمندتر از این بلوک را پیاده سازی می کنید ، که در آن اتصال پرش به جای ۲ لایه ، از ۳ لایه عبور می کند:



- اولین مولفه مسیر اصلی:
 - لایه Conv2D با فیلتر F1 به ابعاد (۱,۱) و stride برابر با (۱,۱) و padding با مقدار valid و Kernel_initializer = initializer(seed=0) و تابع فعال ساز 'relu'
 - لایه Batch_normalization با axis برابر با کانال های تصویر
- دومین مولفه مسیر اصلی:
 - لایه Conv2D با فیلتر F2 به ابعاد (۱,۱) و stride برابر با (۱,۱) و padding با مقدار same و Kernel_initializer = initializer(seed=0) و تابع فعال ساز 'relu'
 - لایه Batch_normalization با axis برابر با کانال های تصویر
- سومین مولفه مسیر اصلی:
 - لایه Conv2D با فیلتر F3 به ابعاد (۱,۱) و stride برابر با (۱,۱) و padding با مقدار valid و Kernel_initializer = initializer(seed=0) و تابع فعال ساز 'relu'
 - لایه Batch_normalization با axis برابر با کانال های تصویر
- مرحله آخر:
 - مقدار خروجی و X_shortcut به یکدیگر اضافه میشوند.
 - راهنمایی: کد آن به صورت Add()([var1, var2]) است. تابع فعالساز آن تابع 'relu' میباشد.

تمرین دوم: بلوک کانولوشن (Convolutional Block)

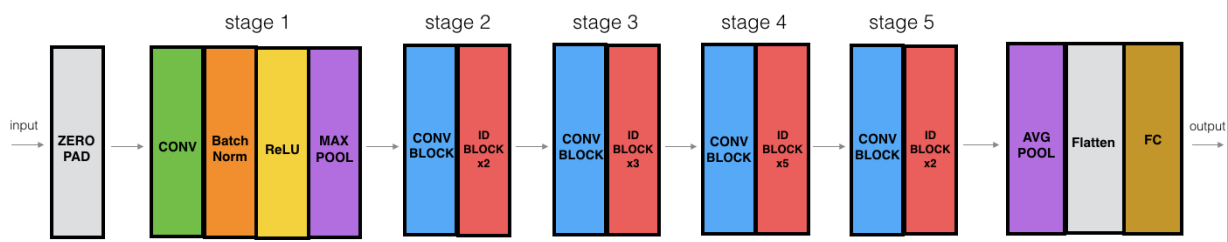
هنگامی که ابعاد ورودی و خروجی با هم مطابقت ندارند ، می توانید از این نوع بلوک استفاده کنید. تفاوت با بلوک هویت این است که یک لایه Conv2D در مسیر میانبر وجود دارد:



- اولین مولفه مسیر اصلی:
 - لایه Conv2D با فیلتر F1 به ابعاد (۱,۱) و stride برابر با (s,s) و padding با مقدار valid
 - Kernel_initializer = glorot_uniform(seed=0) و تابع فعال ساز 'relu'
 - لایه Batch_normalization با axis برابر با کانال های تصویر
- دومین مولفه مسیر اصلی:
 - لایه Conv2D با فیلتر F2 به ابعاد (f,f) و stride برابر با (۱,۱) و padding با مقدار same
 - Kernel_initializer = glorot_uniform(seed=0) و تابع فعال ساز 'relu'
 - لایه Batch_normalization با axis برابر با کانال های تصویر
- سومین مولفه مسیر اصلی:
 - لایه Conv2D با فیلتر F3 به ابعاد (۱,۱) و stride برابر با (۱,۱) و padding با مقدار valid
 - Kernel_initializer = glorot_uniform(seed=0)
 - لایه Batch_normalization با axis برابر با کانال های تصویر
- مرحله آخر:
 - مقدار خروجی و X_shortcut به یکدیگر اضافه میشوند.
 - راهنمایی: کد آن به صورت Add()(var1, var2) است. تابع فعالساز آن تابع 'relu' میباشد.

تمرین سوم: شبکه ResNet50

اکنون بلوک های لازم را برای ساخت یک ResNet بسیار عمیق در اختیار دارید. شکل زیر به طور مفصل معماری این شبکه عصبی را توصیف می کند. ID BLOCK در نمودار مخفف "بلوک همانی" است و ID BLOCK x3 به این معنی است که شما باید ۳ بلوک همانی را در کنار هم قرار دهید.



جزئیات ResNet50 به صورت زیر است:

- اضافه کردن zero_padding با اندازه ۳
- مرحله اول:
 - لایه Conv2D با ۶۴ فیلتر با اندازه (۷,۷) و مقدار stride برابر با (۲,۲)
 - لایه Batch_normalization با axis برابر با کانال های تصویر
 - لایه MaxPooling با اندازه (۳,۳) و stride برابر با (۲,۲)
- مرحله دوم:
 - بلوک کانولوشنی با سه فیلتر با اندازه (۶۴,۶۴,۲۵۶)، مقدار f برابر با ۳ و مقدار s برابر با ۱ است.

- دو بلوک همانی با سه فیلتر با اندازه (۶۴،۶۴،۲۵۶)، مقدار f برابر با ۳
 - مرحله سوم:
 - بلوک کانولوشنی با سه فیلتر با اندازه (۱۲۸،۱۲۸،۵۱۲)، مقدار f برابر با ۳ و مقدار s برابر با ۲ است.
 - سه بلوک همانی با سه فیلتر با اندازه (۱۲۸،۱۲۸،۵۱۲)، مقدار f برابر با ۳
 - مرحله چهارم:
 - بلوک کانولوشنی با سه فیلتر با اندازه (۲۵۶،۲۵۶،۱۰۲۴) مقدار f برابر با ۳ و مقدار s برابر با ۲ است.
 - پنج بلوک همانی با سه فیلتر با اندازه (۲۵۶،۲۵۶،۱۰۲۴)، مقدار f برابر با ۳
 - مرحله پنجم:
 - بلوک کانولوشنی با سه فیلتر با اندازه (۵۱۲،۵۱۲،۲۰۴۸) مقدار f برابر با ۳ و مقدار s برابر با ۲ است.
 - دو بلوک همانی با سه فیلتر با اندازه (۵۱۲،۵۱۲،۲۰۴۸)، مقدار f برابر با ۳
 - لایه Average Pooling با اندازه (۲،۲)
 - لایه Flatten
 - لایه Fully Connected با تعداد نورون هایی به اندازه تعداد کلاس ها و تابع فعالساز softmax
- پس از ایجاد معماری شبکه ResNet50 از پیش آموزش داده شده را با داده های جدید train_signs و test_signs آموزش داده و دقت و عملکرد آن را بررسی کنید.