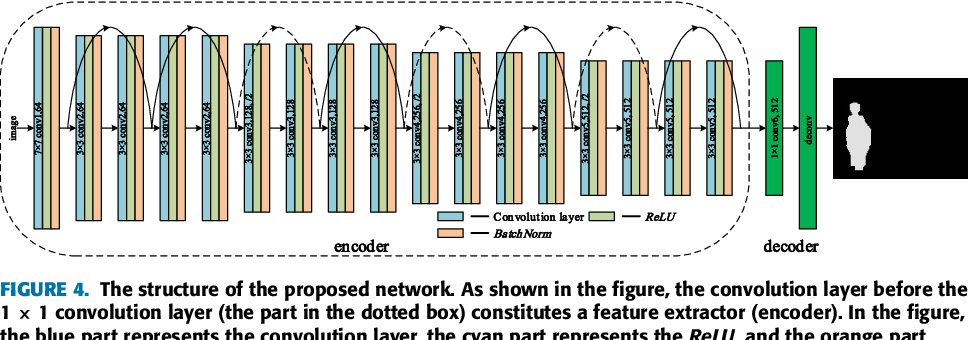
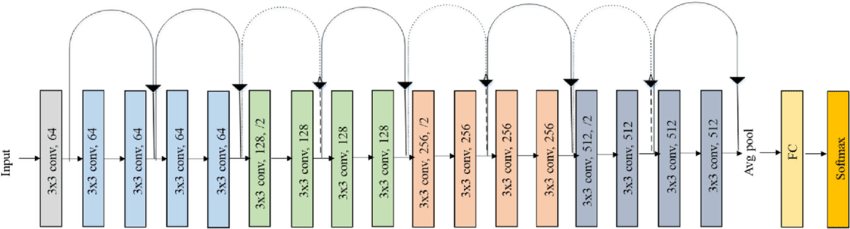
توضیحات کد resnet18 :

رزنت 18 اولین باز سال 2015 توسط محققان مایکروسافت معرفی شد. این مدل pre train شده برای تصاویر اموزش دیده که در قسمت دسته بندی بسیار خوب عمل میکند. این مدل با ساختار ابتکاری پرش از لایه ها میتواند با عمیق شدن از vanishing جلوگیری کند.

حالا من در کدم برای دستی ساختن این مدل از عکس های زیر استفاده کردم.





همانطور که در تصویر میبینیم این مدل دارای 18 لایه است. لایه اول input است و بعد یک conv2d با 64 فیلتر و کرنل 3\*3 که بعد از ان بلاک های رزنت پشت هم می ایند.

بلاک های رزنت برای این ساخته شدند که کار تکراری ما با یک تابع اسان تر شوند. و راحت تر پرش از لایه را بتوانیم پیاده سازی کنیم. بعد لایه دوم بلاک ما صدا زده میشود که ما به ان ورودی و تعداد فیلتر ها و تعداد گام را میدهیم

def block\_resnet(input, filter, stride):

اندازه کرنل ثابت است پس نیازی نیست داده شود و تابع فعال سازی که در کل رزنت استفاده کردم Relu بوده پس باز نیازی نیست داده شود.

در ابتدای بلاک رزنت باید از ورودی یک کپی نگه داریم که بتونیم به عنوان خروجی هم ورودی رو بدیم تا اون پرش از لایه قابل پیاده سازی باشه.

در هر بلاک دوتا conv2d که همراه نرمال سازی و تابع فعال سازی هست میبینیم. ولی بعدش برای اینکه خروجی و ورودی که کپی داشتیم بتونن با هم add بشن باید روی هر دو لایه کانولوژن با کرنل یک بزاریم که اینجوری اندازه های یکسان میگیرن برای اد شدن.

بعد از add کردن هم تابع فعال سازی روشون میزنیم و میفرسیم برگردن به تابع اصلیمون.

# add for skip layers

x = tf.keras.layers.Add()([x\_i, x])

x = Activation(tf.keras.activations.relu)(x)

return x

بعد از ان هم از GlobalAveragePooling2D برای اینکه هم اورفیت نشن هم مناسب دادن به لایه های دنس باشن و حجم محاسبات زیاد نباشه استفاده میکنیم.

بعد با flatten یک بعدی میکنیم و به لایه دنس میدهیم که میخواهد به تعداد کلاس ما دسته بندی کند.

با MODEL میتوان این لایه ها را به هم چسباند و مدلشان کرد. و بعد هم مدل را باز میگردانیم

برای تست این کد از دیتاست cifar10 استفاده کردم که عکس های 32 در 32 دارد و 10 دسته میشوند.

اندازه تصاویر و تعداد دسته هارا به تابع میدهیم و یک ابجکت از ان میسازیم. بعد هم کامپایل و فیت میکنیم و دقت و خطای تست را بدست می اوریم

فقط یک کاری که من کردم در این بین این بود که سرعت train به شدت کم بود پس از chatgpt خواستم کمک کنه سریع تر بشه که اون به کدم from\_tensor\_slices را اضافه کرد که x y را تبدیل به دیتاست تنسورفلو میکند و داده ها مخلوط میشود و داده ها به دسته هایی به تعداد 32 تقسیم میشود. اینجوری سرعت بهتری دارم ولی خب رزنت اینکارو احتمالا نکرده .

با این حال سرعت هنوزم کم بود و برای اینکه اجرای چندباره نکنم به ده ایپاک و فقط داشتن دو دسته داده قناعت کردم و در اخر [0.6560556888580322, 0.7912999987602234] دقت و خطای داده تست من بودن.

کمک من در کل کد : <https://xerac.ir/chat/>

مچکر 