به نام خدا

در اینجا ما از یک مدل مبتنی بر attention استفاده میکنیم که این قابلیت را به ما میدهد تا ببینیم مدل روی چه قسمت هایی از تصویر متمرکز میشود در همان حالی که دارد کپشن را تولید میکند.

مجموعه داده:

در اینجا از دیتاست MS-COCO dataset استفاده میکنیم تا مدلمان را آموزش دهیم. این دیتاست از 82 هزار تصویر تشکیل شده است که هر کدام از تصاویر حداقل 5 کپشن دارد. با کدی که اجرا میکنیم تصاویر دانلود و اکسترکت میشوند. برای آموزش مدل حدودا 30 هزار کپشن و تصاویر متناظر آن استفاده میشود.

پیش پردازش:

1- ریسایز تصاویر به 229px در 229px
2- پیش پردازش تصاویر که از تابع preprocess_input استفاده میکند تا تصاویر را طوری نرمال کند که در رنج 1- تا 1+ قرار بگیرند و برای آموزش InceptionV3 مناسب شوند.

استخراج ویژگی(استفاده از InceptionV3):

ما از وزن های پیش آموخته روی imagenet مدل InceptionV3 برای استخراج ویژگی ها از تصاویر استفاده میکنیم. با قسمت کلسیفایر آن کاری نداریم پس لایه آخر را کنار میگذاریم و از آخرین لایه کانولوشنی استفاده میکنیم. در اینجا ما هر تصویر را به شبکه میدهیم و خروجی آن یک بردار از ویژگی هست که در دیکشنری ذخیره میشود. بعد از اینکه تمام تصاویر به شبکه داده شدند، دیکشنری را در دیسک کش میکنیم.

Tokenization:

در ابتدا caption ها را tokenize میکنیم به صورتی که واژگان تمام کلمات منحصر به فرد را در داده ها به دست آوریم. از بین این واژگان 5000 کلمه که بیشترین تکرار را داشته اند(based on word frequency) را نگه میداریم و بقیه کلمات را با unknown جایگزین میکنیم.

یک مپینگ word-to-index و index-to-word درست میکنیم و در نهایت طبق بلندترین جمله بقیه جملات را pad میکنیم.

مدل و معماری شبکه:

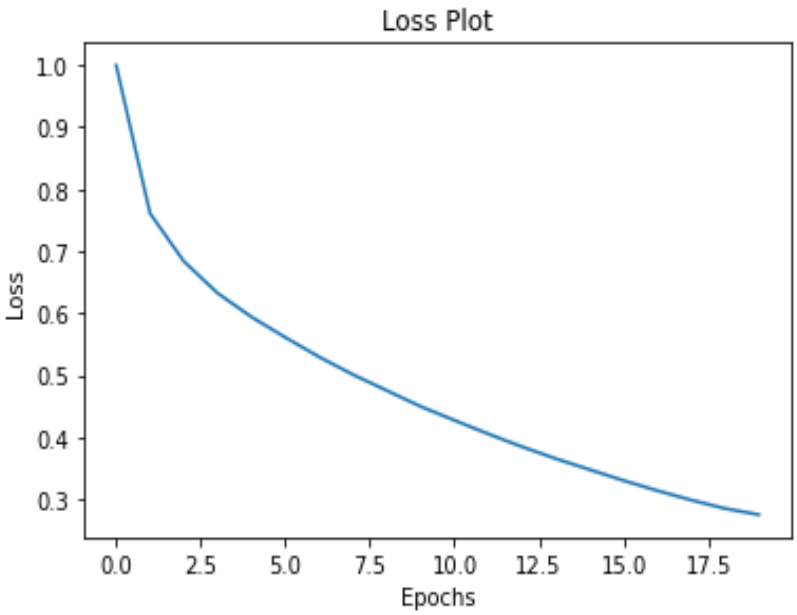
مدل از دو قسمت CNN Encoder و RNN Decoder تشکیل شده است. این معماری از مقاله معرفی شده در صورت سوال الهام گرفته است. در این مثال ما ویژگی ها را از InceptionV3 استخراج میکنیم و یک بردار به shape (8,8,2048) به دست می آوریم. این بردار را squash میکنیم تا (64,2048) شود سپس این بردار به CNN Encoder پاس داده میشود که یک لایه کاملا متصل دارد. در RNN که اینجا GRU است(در مقاله از LSTM استفاده شده است) کلمه بعدی را از روی تصویر پیش بینی میکند.

بررسی مکانیزم attention:

مکانیسم اتنشنی که در این کد استفاده میشود BahdanauAttention است که دو پارامتر برای محاسبه attention weight و context vector میگیرد. اولین پارامتر features است که همان ویژگی های استخراج شده از تصویر میباشد و دومین پارامتر hidden است که همان hidden state های decoder میباشند. این دو پارامتر را به صورت موازی به یک لایه dense با تعداد unit ای که برای شبکه decoder هم استفاده میکند میدهد و سپس این دو را کانکت میکند. بعد از آن tanh activation میزند تا بازنمایی غیرخطی به دست آورد. خروجی این قسمت را به یک لایه dense با تعداد unit برابر 1 میدهد و خروجی آن score میباشد. با اعمال softmax جمع امتیاز ها برابر 1 میشود و attention weights به دست می آید. این وزن ها در فیچر ها ضرب میشوند تا context vector به دست آید که بردار با لایه embedding کانکت میشود و به لایه gru داده میشود.

نتایج:

بعد از آموزش در اخرین ایپاک مقدار loss به 0.275 رسید و نمودار آن به شکل زیر است:



همانطور که در شکل زیر معلوم است برای مثال هنگام پیش بینی کلمه "man" قسمت های مربوط به مردِ روی تصویر فعال شده اند. یا برای "wave" قسمت های موج دریا فعال شده اند.