گزارش تمرین دوم - شبکه عصبی

زهرا سالاریان

بخش پیش پردازش:

در این مرحله ابتدا از آنجایی که می خواهیم از مدل MLP استفاده کنیم باید عکس ها را به یک بعد ببریم و همچنین برای y ها نیز عمل one hot را انجام دهیم.
سپس برای عکس ها، مقادیر را از int به 32float تبدیل می کنیم تا در هنگام انجام عملیات به صورت اعشاری باشند و بتوانند دقت بیشتری را کسب کنند. در نهایت نیز برای نرمال کردن، آنها را تقسیم بر 255.0 می کنیم تا همه ی پیکسل ها در بازه ی 0 تا 1 قرار گیرند.
در این مرحله نمونه ای سیاه و سفید از عکس ها را نیز تهیه می کنیم تا بتوانیم بعد از ساخت مدل، آنها را با هم مقایسه کنیم و ببینیم آیا سیاه و سفید کردن و حذف اطلاعاتی که مربوط به رنگ است باعث از بین رفتن اطلاعات ضروری و مفید می شود و یا این که به ما در رسیدن به دقت بیشتر کمک می کند.
همچنین در کار با عکس ها، ویژگی هایی را نداریم که بتوانیم وابستگی آنها با هم را بررسی و برخی از آنها را حذف کنیم و بنابراین این مرحله قابل انجام نمی باشد.

بخش انتخاب hyperparameterها:

در این مرحله از بین 5 batchای که در train set وجود دارد اولین آنها را انتخاب می کنیم تا در کار تعیین hyperparameterها به ما کمک کند. سپس در این یک batch که شامل 10000 نمونه عکس است، 20 درصد آنها را به داده های train و بقیه را به validation اختصاص می دهیم.
در ادامه قدم به قدم تمامی مراحل انتخاب تک تک hyperparameterها به طور عملی و با کد و نتایج آنها بررسی شده و در پایان هر قدم نتیجه نیز در نوت بوک ذکر شده است.
یکی از موارد قابل توجه در این مراحل این است که اگر اندازه ی learning rate بیش از حد بزرگ باشد باعث underfit شدن مدل و عدم یادگیری آن می شود و بنابراین باید مقدار مناسب تنظیم شود.
همچنین در هر قدم نمودارهای accuracy و loss برای داده های train و validation رسم شده است و با مقایسه ی آنها می توانیم میزان overfit و underfit شدن مدل را متوجه شویم. در جایی که فاصله ی بین خطوط train و val در نمودار loss از هم زیاد می شود میدانیم که مدل overfit شده است و کارهایی از قبیل کم کردن تعداد epochها و یا عمل Drop را می توانیم انجام دهیم. کاری که عمل Drop انجام می دهد این است که در هر لایه ای که تعبیه شود برای نودهای

آن لایه با یک احتمال مشخص که خودمان مشخص می کنیم می تواند تعیین کند که آیا آن نود بی تاثیر شود یا خیر. به طور مثال اگر در لایه ی دوم Drop(0.2) را تعبیه کنیم آنگاه هر یک از نودهای آن لایه با احتمال 0.2 امکان دارد که بی تاثیر شوند و انگار که اندازه ی آنها برابر با صفر است. البته این نودها به طور کامل حذف نمی شوند و در پیمایش بعدی نودهای دیگری ممکن برای بی تاثیر شدن انتخاب شوند. اما فایده ی این کار این است که باعث می شوند کمی از اطلاعاتی که مدل تنها مختص داده های train کسب کرده است و وزن هایی که در حال اختصاصی شدن برای داده های train هستند کمتر شوند و در واقع مدل general تر شوند و بتواند بهتر به داده های غیر از train پاسخ دهد و در واقع از overfit شدن مدل جلوگیری شود.

بخش train و test روی تمامی 5 batch:

Static:

در این بخش همه ی 5 batch را برای آموزش مدلی که از مرحله ی قبل به آن رسیده ایم را در نظر گرفته و 20 درصد آن را به عنوان داده های validation در نظر می گیریم. پس از آموزش مدل آن را بر روی داده های test مورد تست قرار می دهیم و به دقت حدود 0.51 می رسیم.

Dynamic:

برای این حالت از روش k-fold cross validation استفاده کرده ایم و در آن مقدار k را برابر با 5 قرار داده ایم. روش کار در این حالت نیز مشخص است و داده های train به 5 فولد تقسیم شده و در هر مرحله یکی از آنها validation و بقیه ی آنها train هستند. پس از آموزش و تست دقتی که از این حالت بدست آمده است برابر با 0.51 می باشد.
با مقایسه ی نتایج گرفته شده میبینیم که دو روش چندان تفاوتی با یکدیگر نداشته اند

در مرحله ی بعدی confusion matrix را محاسبه و رسم می کنیم و همانطور که انتظار داشتیم اعدادی که روی قطر اصلی قرار دارند بزرگتر هستند و این یعنی مدل تعداد زیادی از موارد را درست پیش بینی کرده است اما به طور مثال می توان از نزدیک بودن اعدادی که در ستون dog وجود دارد نتیجه گرفت که از بین کل مواردی که dog بوده اند مدل 379 عدد از آنها را dog پیش بینی کرده است اما 218 تا از آنها را به اشتباه cat دسته بندی کرده است. این اشتباه و اشتباهات مشابه آن، با توجه به شباهت سگ و گربه قابل درک است و انتظار می رفته است که مدل MLP همچین اشتباهی را مرتکب شوند به دلیل این که نمیتواند از موقعیت مکانی پیکسل ها و غیره بهره برده و قوی تر شکل را تحلیل کند.

در قسمت بعد نیز accuracy, precision, recall و F1 score برای هر کلاس محاسبه شده است.
پس از آن تعدادی از عکس ها را به همراه label آنها و labelای که مدل پیش بینی کرده است رسم کرده ایم تا بتوانیم واضح تر نتایج و اشتباهات مدل را مشاهده کنیم.

Grayscale:

در آخرین گام نیز تصاویر سیاه و سفید شده را train کرده ایم که با این که عدد learning rate را بسیار کمتر از قبل، تعداد epoch ها را دو برابر، Drop ها را حذف و batch size را نیز دو برابر کرده ایم باز هم نتیجه و دقتی که بدست آورده ایم بسیار بدتر از قبل می باشد و این بدان معنا است که وجود رنگ برای تصاویر داده ای مفید و تعیین کننده بوده است و با حذف آن به نتیجه ی حاصله لطمه وارد کرده ایم.

جمع بندی:

به طور کلی دقتی که مدل MLP ما در نهایت داشته است 0.5 درصد است که مناسب نیست و دلیل آن هم این است که مدل MLP به طور کلی برای کار با تصاویر مناسب نیست به دلیل این که باید عکس ها را به یک بعد ببریم و تمام اطلاعات شامل مکان پیکسل ها و نزدیکی آنها به یکدیگر و خیلی چیزهای دیگر را از دست می دهیم. به همین دلیل است که از مدل هایی که توانایی پردازش ابعاد بیشتر دارند مانند convolutional neural netwok برای کار با تصاویر استفاده می شود.