به نام خدا

گزارش تمرین ۱۱

نیما کمبرانی ۹۸۵۲۱۴۲۳

مبانی بینایی کامپیوتر

استاد: دکتر محمدی

سوال۱)

الف)

در تصاویر پیکسل های کنار هم به یکدیگر وابسته هستند و ممکن است یک الگو در تصاویر مختلف در بخش های مختلفی از تصویر باشد. در لایه های fully connected با توجه به این که این اطلاعات از همسایگی ها در نظر گرفته نمی‌شود با جابه‌جایی چند پیکسلی تصویر یاد گرفته شده دیگر وزن های قبلی مناسب نیستند اما در کانولوشن با توجه به اعمال فیلتر ثابت در همسایگی های مختلف این الگو ها پیدا خواهد شد و جابه جایی آن تاثیر چندانی ندارد.

ب)

برای اینکه طول عرض خروجی تغییر نکند، با توجه به سایز فیلتر های مرحله بعد که ۵ در ۵ هستند نیاز به گسترش مرز به ابعاد ۲ از هر طرف تصویر داریم.

تعداد پارامتر های این لایه نیز برابر ۱۶ فیلتر با ۲۵ پارامتر است که برابر با ۴۰۰ می شود.

پ)

در حالت اول با توجه به تعداد فیلتر ها خروجی ۳ کانال دارد که هر کدام با توجه به اندازه اولیه ۳۲ و اندازه ۵ برای فیلتر دارای ابعاد ۲۸ در ۲۸ هستند. در نتیجه خروجی ۲۸ در ۲۸ در ۳ است.

در حالت دوم هر یک از فیلتر های ۳ در ۳، یه ردیف از اطراف عکس کم می کنند که در نتیجه ابعاد خروجی نهایی با توجه به تعداد فیلتر ها برابر ۲۸ در ۲۸ در ۹ خواهد بود.

ت)

لایه global avg polling در آخر لایه های کانولوشنی قرار می‌گیرد و تمام خروجی‌های هر فیلتر را میانگین می‌گیرد. این لایه برای استخراج ویژگی ها و کاهش ابعاد این ویژگی ها و در نتیجه کاهش هزینه محاسباتی کاربرد دارد. در کاربرد هایی مانند کلاس بندی تصاویر که تنها به وجود یا عدم وجود یک ویژگی در تصویر نیاز داریم این لایه مناسب است. و در جایی که بدنبال مکان یک شی در تصویر هستیم، مناسب نیست.

لایه های max pooling و average در میان شبکه قرار می گیرد و برای کاهش ابعاد و محاسبات استفاده می شوند. همچنین باعث میشوند تا مدل نسبت به جابه‌جایی های کوچک مقاوم تر باشد. Max تنها بزرگترین ویژگی در آن ناحیه را باز می‌گرداند اما average میانگین این مقادیر را باز می گرداند.

ث)

در resnet تنها یک لایه تماما متصل داریم و همچنین در لایه آخر از لایه global average pooling استفاده می کنند که به کاهش محاسبات و تعداد متغیر ها کمک می کنند. علاوه بر این، در این مدل برای مدل های عمیق با استفاده از فیلتر های ۱در۱ تعداد کانال‌های ویژگی ها را کاهش می دهند که باعث کاهش محاسبات می شود.

در vgg ایده جدیدی که استفاده شد، استفاده از فیلتر های کوچک اما با تعداد بیشتر بود که باعث می‌شد با توجه به تعداد بیشتر آن‌ها و وجود تابع غیر خطی در میان آن‌ها نسبت به مدل های قبل از خود نتیجه بهتری کسب کند.

در resnet ایده اصلی، جمع کردن ورودی یک بلاک با خروجی فیلتر های آن بود که باعث شد برای مدل های با عمق بیشتر امکان آموزش فراهم شود و در نتیجه دقت افزایش یابد.

سوال ۲)

الف)

با توجه به نقطه شروع تصادفی برای پارامتر های مدل، در اجرا های مختلف خروجی متفاوت بود و در بعضی حالات به دقت خوبی همگرا نمی شدند. به طور کلی میزان خروجی تابع خطا با توجه به فاصله پیشبینی از مقدار واقعی محاسسبه می‌شود در نتیجه ممکن است در یک مدل با فاصله زیاد اما درست پیشبینی را انجام دهند اما در یک مدل با فاصله کمتر اما به صورت غلط پیشبینی صورت پذیرد و درنتیجه با هزینه کمتر ولی دقت کمتر رو به رو باشیم. این موارد در epoch های مختلف آموزش قابل مشاهده است که با هزینه و خطای کمتر دقت کمتری داشته باشیم.

```
Loss and Accuracy on Test set :
313/313 [==============================] - 1s 4ms/step - loss: 9.4742 - accuracy: 0.1383
```

شکل ۱ . دقت مدل کانولوشنی بر روی داده تست

```
Loss and Accuracy on Test set :
313/313 [==============================] - 1s 3ms/step - loss: 2.4200 - accuracy: 0.1744
```

شکل ۲ . دقت مدل تماما متصل بر روی داده تست

ب)

برای مدل تماما متصل در اجرا بر روی cpu هر epoch در حدود ۱۰ ثانیه و بر روی gpu حدود ۷ ثانیه زمان می برد. برای مدل کانولوشنی اما با توجه به تعداد زیاد محاسبات موازی، بر روی cpu حدود ۶ دقیقه و بر روی gpu حدود ۱۰ ثانیه زمان می برد.

پ)

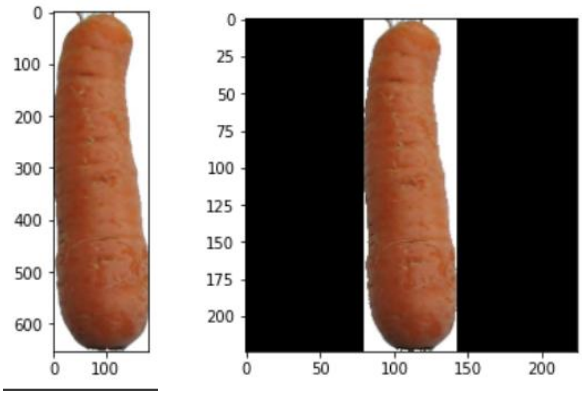
با توجه به اینکه ساختار دو مدل متفاوت است، تعداد پارامتر ها نمی تواند برای مقدار زمانی که برای محاسبات هر مدل برای اموزش مورد نیاز است، استفاده شود. در شبکه تماما متصل محاسبات در هر تصویر با هر پارامتر تنها یک بار انجام می پذیرد. اما در شبکه کانولوشنی هر یک از فیلتر ها با توجه به ابعاد تصویر و اندازه قدم می تواند بر روی بخش های مختلف تصویر اعمال شود و در نتیجه هزینه محاسباتی و تعداد عملیات های موازی افزایش میابد.

سوال ۳)

الف)

برای تغییر اندازه تصاویر داده شده به همراه حفظ مقیاس آن‌ها با توجه به مراحل زیر عمل می‌کنیم:

۱. ابتدا با توجه به ابعاد تصویر، نسبت کوچک کردن تصویر را با توجه به بیشینه طول و عرض تصویر و نسبت آن با اندازه نهایی خواسته شده می‌یابیم.
۲. با توجه به مقیاس بدست آمده ابعاد جدید تصویر را می‌یابیم و با استفاده از تابع resize تصویر را به ابعاد خواسته شده تبدیل می‌کنیم.
۳. برای تبدیل اندازه ضلع کوچکتر به اندازه خواسته شده، یک مربع با طول اضلاع خواسته شده می‌سازیم و عکس تغییر یافته را در وسط آن قرار می دهیم. در شکل ۳ تصویر اولیه و تصویر پس از تغییر ابعاد مشخص شده است.



شکل ۳ . در سمت چپ تصویر اولیه و در سمت راست تصویر نهایی پس از اعمال تغییر سایز آمده است

ب)

در ابتدا با استفاده از مدل از پیش طراحی شده و با شروع وزن های تصادفی مدل را می‌سازیم و به آن یک لایه با ۲۴ نورون برای تعیین خروجی اضافه می کنیم و همه ی پارامتر های مدل را قابل آموزش قرار می‌دهیم.

پ)

در این مرحله مدل resnet از پیش طراحی شده را با وزن های آموزش یافته بر روی دیتاست imagenet مقدار دهی اولیه می‌کنیم و همچنین پارامتر های این مدل را غیر قابل اموزش می کنیم تا در طول آموزش ثابت بمانند. در ادامه به آن یک لایه تماما متصل اضافه می‌کنیم تا وزن های آن را در طول آموزش بدست آوریم.

ت)

آموزش با ۳۰ epoch انجام شد. مدل resnet که از وزن های تصادفی شروع کرده بود در چند دور ابتدایی دقت پایین تری داشت و در ۱۰ دور اول نوسان بیشتری در دقت داشت اما بعد از epoch۲۰ هر دو مدل به دقت بالای ۹۹ درصد همگرا شدند.

در مدل اول که وزن ها در حال آموزش بودند با توجه به تعداد بیشتر پارامتر های در حال آموزش، در هر epoch حدود ۱ ثانیه و برای مدل دوم که وزن ها پیش اموخته شده و ثابت بودند،حدود 0.5 ثانیه بود.

همچنین دقت بر روی داده های تست نیز برای هر دو مدل نزدیک به هم و بیش از ۹۹ درصد بود اما دقت مدل با وزن های از پیش آموزش یافته کمی بهتر بود. با توجه به نتیجه، هر دو مدل توانسته اند با دقت بالا و بدون overfit شدن به نتایج خوبی برسند.

دقت مدل با وزن های تصادفی بر روی داده تست: 99.71 درصد

دقت مدل با وزن های از از پیش آموخته بر روی داده تست: 99.94 درصد