به نام خدا

گزارش تمرین ۱۲

نیما کمبرانی ۹۸۵۲۱۴۲۳

مبانی بینایی کامپیوتر

استاد: دکتر محمدی

سوال۱)

overfit به معنای تطبیق بیش از حد مدل با داده های آموزش و کاهش قدرت تعمیم دهی مدل بر روی داده های آزمایش است. عواملی که در به وجود آمدن این مشکل نقش دارند عبارتند از، کم بودن داده های آموزش و پوشش ندادن درست از فضای مسئله، آموزش بیش از حد مدل و پیچیده بودن مدل درحال آموزش برای مسئله و یادگیری ویژگی های غیرکاربردی.

برای رفع این مشکل می‌توان تعداد داده های آموزش را افزایش داده تا مدل با حالت های بیشتری آشنا شود یا پیچیدگی مدل را کاهش داد و از مدل ساده تر استفاده کرد یا طول آموزش مدل را کاهش داد تا فرصت برای یادگیری ویژگی های غیر مفید نداشته باشد.

Underfit به معنای کم بودن دقت مدل بر روی داده های تست و آموزش است.

عواملی که باعث ایجاد این مشکل می‌شوند عبارتند از، کم بودن طول آموزش مدل، ساده بودن مدل برای حل مسئله یا پیچیدگی مسئله.

برای رفع این مشکل می‌توان از مدل های پیچیده تر استفاده کرد و یا مدل را برای زمان طولانی‌تری آموزش داد تا فرصت بیشتری برای یادگیری داشته باشد.

سوال۲)

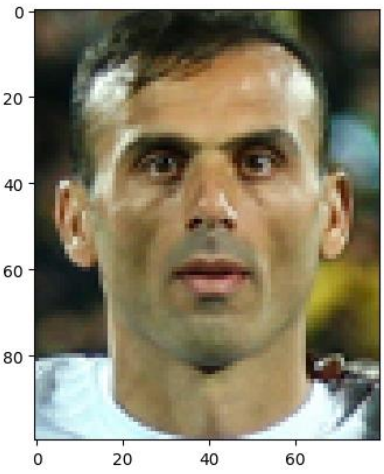
برای لیبل گذاری تصویر داده شده از برنامه labelme استفاده شد. نتیجه لیبل گذاری در شکل۱ قابل مشاهده است.

شکل ۱ . تصویر ورودی پس از لیبل گذاری با استفاده از labelme

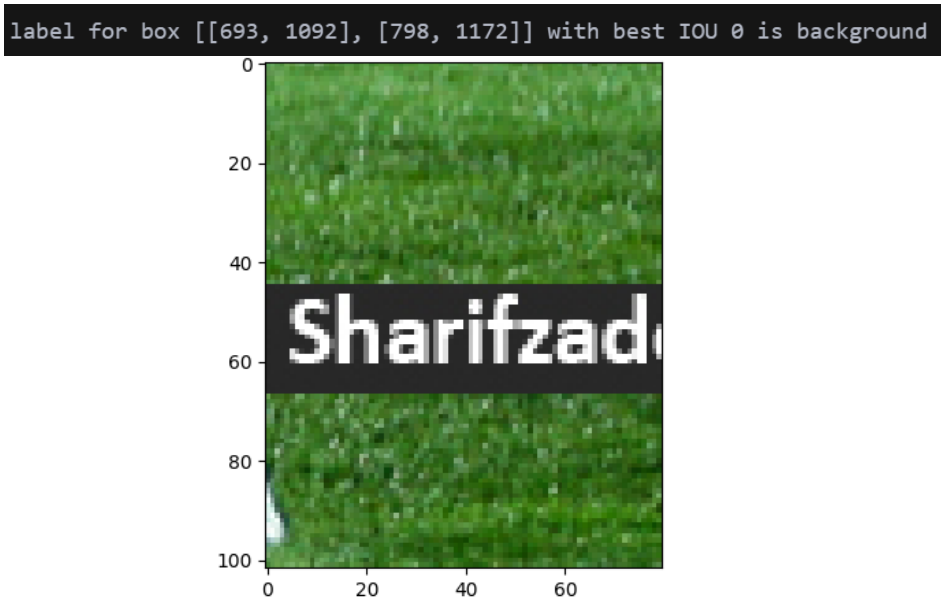
در این برنامه خروجی یک فایل json است که در آن مستطیل های برچسب با مختصات دو نقطه چپ بالا و راست پایین مشخص می شوند. البته ترتیب مختصات نقاط بصورت x,y است که در عکس به هنگام خواندن باید به صورت y,x ایندکس دهی شود.

برای ساختن پنجره های پیشنهادی با توجه به شکل، اکثر برچسب ها بصورت حدودی مستطیل با طول ۸۰ و ارتفاع ۱۰۵ هستند. در نتیجه برای تولید پنجره های پیشنهادی با استفاده از حلقه تو درتو باکس هایی که با دو نقطه گوشه آن مشخص می‌شوند و طول و ارتفاع ۸۰ در ۱۱۰ دارند ایجاد می‌کنیم. فاصله این پنجره ها با استفاده از یک متغیر stride قابل تنظیم است. برای هر یک از پنجره های بدست آمده اشتراک آنها با همه ی برچسب ها محاسبه می‌شود و بزرگترین اشتراک برای آن پیدا می‌شود. اگر این اشتراک بیشتر از ۵۰ درصد بود بعنوان تصویر صورت در نظر گرفته می‌شود.

نمونه هایی از پنجره های برش خورد در زیر آمده است.



شکل ۳ . تصویر پنجره با ابعاد ۱۱۰ در ۸۰ و اشتراک بیشتر از ۹۰ درصد با برچسب صورت



شکل ۲ . شکل بدون اشتراک با هیچ برچسب

سوال ۳)

سوال ۴)

۱. با قرار دادن padding = same تعداد پیکسل های اضافه شده به دور تصویر ورودی لایه، با توجه به اندازه فیلتر به طوری خواهد بود که اندازه طول و عرض تصویر خروجی با تصویر ورودی برابر شود. حالت دیگر padding = valid است که در این حالت هیچ پیکسلی به دور تصویر اضافه نمی‌شود.

۲. تابع فعال سازی باعث می‌شود تا فیلترهای کانولوشنی خاصیت غیر خطی پیدا کنند و در نتیجه بتوانند توابع پیچیده‌تری را بیابند. در اینجا با قرار دادن تابع فعال سازی هر تصویر پس از گذر از فیلتر، از این تابع relu نیز عبور می‌کند.

۳. در ابتدای ساخت مدل نیاز به مقداردهی اولیه وزن های مدل داریم. در اینجا با استفاده از توزیع تصادفی نرمال وزن های اولیه فیلتر ها مقدار دهی می‌شوند.

۴. عمل کانولوشن به طور کلی باعث کاهش ابعاد تصویر ورودی می‌شود اما conv transpose برخلاف آن باعث افزایش ابعاد خروجی نسبت به ورودی می‌شود. در اینجا با توجه به وجود stride =2 ابعاد خروجی نسبت به ورودی ۲ برابر می‌شود.

۵. تابع double_conv_block از دو لایه کانولوشن تشکیل شده است که ابعاد تصویر ورودی را حفظ می‌کنند. تابع downsample در ابتدا با استفاده از تابع double_conv بر روی ورودی ۲ لایه کانولوشن اعمال می‌کند. سپس با استفاده از با استفاده از max pooling ابعاد ورودی را نصف می‌کند و به خروجی می‌دهد. upsample خروجی لایه کانولوشنی با عمق برابر و خروجی لایه upsample قبل را بعنوان ورودی می‌گیرد. سپس ابتدا یک convtranspose بر روی آن اعمال می‌کند تا ابعاد آن دو برابر و برابر با ابعاد ویژگی های کانولوشنی شود. در نهایت ویژگی های کانولوشنی و ورودی لایه قبل را در کنار هم قرار می‌دهد و به دولایه کانولوشنی می‌دهد تا خروجی نهایی تولید شود.

۶. optimizer توابعی هستند که به بهینه سازی وزن های مدل و کاهش تابع هزینه در آن کمک می‌کنند. همچنین باعث می‌شوند تا مدل سریع تر یادگیری را انجام دهد و در نهایت به نتایج بهتری برسد و در حالت هایی مانند شیب کم گرادیان و مینیموم های محلی گیر نکند.

۷. تابع compile بر روی وزن های مدل تاثیری ندارد و تنها باعث می‌شود تا مدل با تابع هزینه و بهینه ساز جفت شود و آماده آموزش شود.

۸. تابع هزینه categorical_crossentropy برای زمانی استفاده می‌شود که خروجی بیشتر از یک کلاس دارد و در هر نمونه تنها یکی از این کلاس های خروجی فعال می‌شوند.

۹. تابع earlystopping برای بررسی روند آموزش مدل استفاده می‌شود. این تابع با بررسی یک متریک (در اینجا validation loss است) آموزش مدل را تا زمانی که این متریک در حال بهبود باشد ادامه می‌دهد. اگر تغییرات و بهبود در چند دور متوالی کمتر از حد مشخصی بود آموزش را متوقف می‌کند.

۱۰. تابع compile تنها تابع های مورد نیاز برای آموزش مدل را با آن جفت می‌کند و تاثیری در وزن های مدل ندارد. تابع fit عملیات آموزش مدل با توجه به توابعی که در بخش compile با آن جفت شده است را انجام می‌دهد.

۱۱. برای اینکه آموزش مدل بهتر و سریعتر و با حافظه کمتری صورت بگیرد، کل داده های آموزش را به تعدادی بخش کوچکتر دسته‌بندی می کنیم که به آن Batch می‌گوییم. به یک دور آموزش بر روی کل دسته های داده epoch می‌گوییم. در نتیجه در هر epoch تعدادی batch بررسی می‌شوند.

سوال۵)

برای بدست آوردن هر یک از مقادیر AP به ازای IOU برابر مقادیر ۰,۲۵ و ۰,۵ و ۰,۷۵ به این صورت عمل می‌کنیم که اگر iou برای یک باکس بیشتر از آستانه بود باید score آن نیز بیشتر از ۰,۵ باشد. در نتیجه نسبت تعداد باکس هایی که درست پیشبینی شده اند به کل باکس هایی که باید پیشبینی می‌شدند بدست می‌آوریم.

$AP = TP/(TP + FP)$

$AP_{25} = 3/5$

$AP_{50} = 3/5$

$AP_{75} = 2/5$