بسمه تعالی

دانشکده مهندسی کامپیوتر

بینایی کامپیوتر

نام استاد: دکتر محمدی

تمرین دوازدهم

نام دانشجو: آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

دی ۱۴۰۱

فهرست

پاسخ سوال اول

علل underfitting:

- آموزش مدل با دیتای کم
- مدت زمان یا epoch کم برای آموزش مدل در نظر گرفته شده
- ساده بودن مدل نسبا به دیتاست هدف
- ویژگی های دیتاست برای هدف مدل مناسب نیستند

در واقع در این حالت مدل دقت خوبی حتی موقع آموزش هم ندارد. راه درست کردنش استفاده از مدل پیچیده تر، افزایش دفعات آموزش و یا تلاش برای استخراج ویژگی های مناسب تر برای آموزش مدل است.

اما overfitting یعنی مدل ما داده های آموشی را حفظ کرده است و دقت عالی روی آن ها و دقت ضعیف بر روی داده های تست یا ارزیابی دارد. علل مختلف این پدیده:

- پیچیدگی بیش از حد مدل
- بالانس نبودن دیتاست بین کلاس های مختلف و یا شامل نشدن برخی حالات
- به طور کلی کمبود داده ها
- آموزش بیش از حد مدل

برای جلوگیری میتوانیم data augmentation کنیم یا مدل را ساده کنیم و یا آموزش را در جایی که دقت روی داده ارزیابی حداکثر است، متوقف کنیم. همچنین از regularizer ها استفاده کنیم.

پاسخ سوال دوم و سوم

کد مربوط به این دو سوال در نوتبوک HW12 پیوست شده است و خروجی ها مشخص هستند. برخی توضیحات لازم هم بین سلول های نوتبوک داده ام.

پاسخ سوال چهارم

۱. اگر مقدار padding برابر با same باشد بعد از عمل کانولوشن ابعاد خروجی برابر با ابعاد ورودی خواهد بود اما اگر مقدارش برابر با valid باشد یعنی هیچ padding ای اعمال نشده است.

۲. تابع فعال سازی کمک می کند تا مدل بتواند روابط غیر خطی را یاد بگیرد. برای پیچیده شدن روابط مورد یادگیری مدل با عمیق شدن آن، باید این را داشته باشیم زیرا وگرنه درجه ورودی همچنان ۱ باقی می ماند.

۳. پارامتر kernel_initializer به ما این امکان را می دهد تا با الگوریتم دلخواه بتوانیم وزن های اولیه را تعریف کنیم. مثلا Glorot initializer.

۴. Conv2D ابعاد تصویر را کم می کند ولی Conv2Dtranspose ابعاد تصویر را بزرگتر می کند به این عمل upsampling نیز گفته می شود. لایه کانولوشنال Conv2DTranspose یا transpose پیچیده تر از یک لایه upsampling ساده است. هم عملیات نمونه‌برداری را انجام می‌دهد و هم داده‌های ورودی درشت را برای پر کردن جزئیات در حین نمونه‌برداری تفسیر می‌کند.

۵. تابع block_conv_double دو تا لایه کانولوشنی را روی ورودی اعمال میکند. در هرکدام از این لایه ها هم ابعاد ورودی و خروجی یکسان است چونpadding=same و همچنین stride هم یک است. پس در این تابع فقط تعداد کانالهای ورودی تغییر میکند و در خروجی ظاهر میشود.

در تابع downsample_block فقط از تابع double_conv_block استفاده کرده و در ادامه با لایه maxpooling2D ابعاد تصویر را نصف می کند. در تابع upsample_block، ابتدا ابعاد ورودی x را بیشتر می کنیم با استفاده از Conv2Dtranspose و در ادامه خروجی را با فیچر های کانولوشنی مجاور concat می کنیم و در نهایت خروجی نهایی را دوباره به تابع double_conv_block می‌دهیم.

۶. که loss کاهش پیدا کند و وزن های ما در جهت درست (با توجه به گرادیان) تغییر کنند و به تدریج دقت بالا رود.

۷. برای چسباندن optimizer و function loss و تمام پارامترهایی که شبکه با آن config میشود به هم استفاده میشود.

۸. برای مسائل classification این تابع ضرر انتخای اول است چون به خوبی loss را برای موارد خیلی غلط، خیلی زیاد میکند.

۹. برای اینکه از overfitting جلو گیری کنیم و همچنین در وقت صرفه جویی کنیم، از این تابع استفاده می کنیم. در این تابع اگر مدل در تعداد epoch مشخصی بهبود نیابد، آموزش متوقف می شود. در واقع val_loss را مانیتور میکند و اگر به جای کاهشی، افزایشی شود آموزش را متوقف میکند.

۱۰. fit آموزش ر ا شروع میکند و با پارامتر هایی که در compile برای مدل مشخص کردیم مدل را بهبود میبخشد. اما compile فقط مدل را آماده میکند. در تابع کامپایل optimizer و loss function را مشخص می کنیم اما در تابع فیت، دیتای آموزش و آزمون و تعداد epoch و همچنین batch size و ... را مشخص می کنیم.

۱۱. در هر epoch ما روی چندین batch آموزش را انجام میدهیم، Batchها قسمت هایی با تعداد مشخصی از داده آموزشی هستند که در هر epoch پخش شده اند اما در نهایت در یک epoch ما مدل را روی کل داده ها آموزش میدهیم. که درواقع تعداد زیادی batch می شود.

پاسخ سوال پنجم

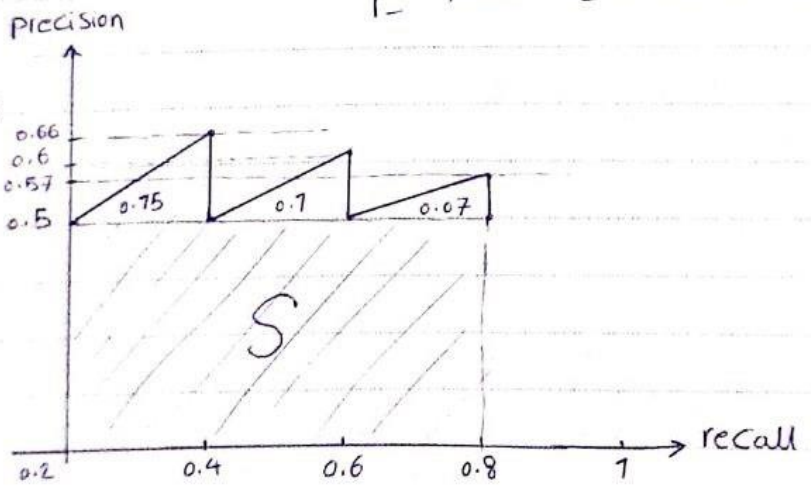
Subject:
Date

به ترتیب score آن ها را Sort می‌کنیم :

Rank	score	P_{25}	R_{25}	P_{50}	R_{50}	P_{75}	R_{75}
1	0.96	1	0.2	1	0.2	0	0
2	0.89	0.5	0.2	0.5	0.2	0	0
3	0.84	0.66	0.4	0.66	0.4	0.33	0.2
4	0.79	0.5	0.4	0.5	0.4	0.25	0.2
5	0.74	0.6	0.6	0.6	0.6	0.4	0.4
6	0.47	0.5	0.6	0.5	0.6	0.33	0.4
7	0.39	0.57	0.8	0.43	0.6	0.28	0.4
8	0.29	0.5	0.8	0.375	0.6	0.25	0.4

حالا طبق precision و recall های به دست آمده در جدول بالا می‌توانیم نمودارها را رسم کرده و مساحت هرکدام را حساب کنیم.

AP_{25} :

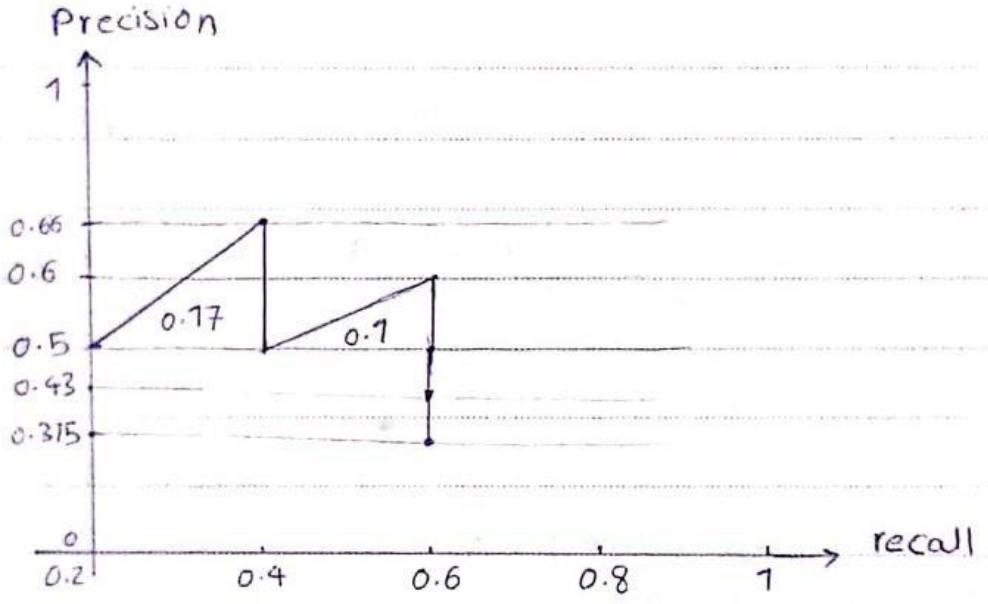


$S_T = 0.17 + 0.1 + 0.07 + 0.3 = 0.64 = \boxed{64\cdot/\cdot}$

PAPCO

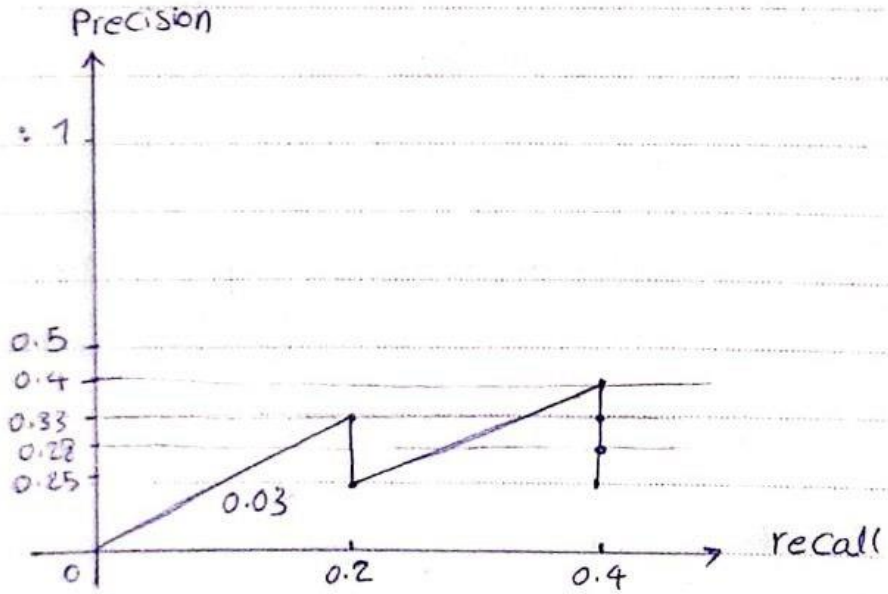
Subject
Date

AP_{50} :



$S_T = 0.17 + 0.9 + 0.2 = 0.47 = 47\%$

AP_{75} :



$S_T = 0.03 + 0.025 + 0.015 = 0.07 = 7\%$