

به نام خدا



درس مبانی بینایی کامپیوتر

تمرین سری دوازدهم

مدرس درس:

جناب آقای دکتر محمدی

تهیه شده توسط:

الناز رضایی ۹۸۴۱۱۳۸۷

تاریخ ارسال: ۱۴۰۱/۱۰/۱۹

سوال ۱:

علت به وجود آمدن underfitting و overfitting چیست؟ هر کدام را توضیح دهید و بگویید چگونه می‌توان این موارد را رفع کرد؟ (۱۵ نمره)

پاسخ ۱:

overfitting به حالتی گفته می‌شود که مدل بر روی داده‌های train دقیق شده و جزئیات آن را یاد گرفته است و بنابراین قابلیت generalization آن کم می‌شود و عملکرد خوبی روی داده‌های تست ندارد؛ در صورتی که دقت بالا و loss کمی روی داده‌های train دارد. بنابراین overfitting زمانی اتفاق می‌افتد که مدل نتواند تعمیم بدهد. از علل دیگر overfitting، به کم بودن سائز دیتاست، نداشتن مجموعه داده آموزشی که نماینده خوبی از تمامی حالت‌های ورودی باشد، زیاد بودن پارامترهای شبکه و بالا بودن ظرفیت مدل اشاره کرد. به بیان دیگر، overfitting استفاده بیش از حد از پیچیدگی و قدرت برای حل یک مسئله است. برای رفع overfitting، می‌توان از روش‌های زیر استفاده کرد:

- افزایش مجموعه داده یا داده‌افزایی: در این روش با انجام یک‌سری تبدیلات هندسی نظیر چرخش، شیف، scale کردن و ...، می‌توان از یک مجموعه داده کوچک، داده‌های بیشتری تولید کرد. این کار باعث افزایش قابلیت generalization می‌شود.
- استفاده از Dropout: باعث می‌شود بخشی از نوروها در یک epoch غیرفعال شوند و این کار به صورت تصادفی با احتمال انجام می‌شود. به این ترتیب همه نوروها به طور همزمان یاد نمی‌گیرند، بنابراین ظرفیت کلی شبکه کاهش می‌یابد و احتمال کمتری دارد که مدل overfit شود.
- کاهش ظرفیت شبکه: مانند کاهش تعداد لایه‌ها، تعداد نوروها در لایه‌ها و ... این باعث می‌شود که مدل از قدرت کمتری برخوردار باشد، احتمال کمتری دارد که روی داده‌ها overfit شود.
- روش‌های منظم‌سازی (Regularization): هنگام استفاده از Regularization، جریمه

ضرر با افزایش پیچیدگی مدل افزایش می‌یابد، بنابراین شبکه از یادگیری یک مدل پیچیده دلسرد می‌شود.

● Cross Validation: برای اندازه‌گیری تعمیم مدل در هر epoch آموزشی استفاده می‌شود و می‌تواند برای تشخیص و جلوگیری از overfitting استفاده شود.

underfitting به حالتی گفته می‌شود که مدل نه روی داده‌های train عملکرد خوبی دارد و نه روی داده‌های test؛ بنابراین قابلیت generalization هم ندارد. از علل به وجود آمدن underfitting می‌توان به بالا بودن بایاس مدل، کافی نبودن سائز دیتاست، ساده بودن مدل و clean نبودن دیتا و حاوی نویز بودن آن اشاره کرد. مدل وقتی underfit است، دقت آن روی تمامی داده‌ها کم و loss آن زیر است. استفاده نکردن از optimizer مناسب نیز می‌تواند باعث underfitting شود. مشکل underfitting را می‌توان با استفاده از یک شبکه بزرگتر (nodeهای پنهان بیشتر) حل کرد. تعداد زیادی از nodeها قادر خواهند بود مقدار خوبی از ویژگی‌ها را استخراج کنند و بدین ترتیب داده‌ها را کارآمدتر درک کنند. همچنین، می‌توانیم مدل خود را برای مدت طولانی‌تری آموزش دهید تا از مشکل عدم تناسب جلوگیری کنیم.

سوال ۲:

(محاسبه IoU نواحی پیشنهادی) تصویر ضمیمه Melli.jpg را در نظر بگیرید. با استفاده از ابزارهای برچسب (Label) زنی مانند Labelme یا Labeling و ... ابتدا صورت بازیکنان را به صورت دستی Annotation گذاری کنید. سپس با اجرای الگوریتم پنجره‌ی لغزان ناحیه‌های پیشنهادی را به دست آورید. حتماً نوع پنجره‌(های) پیشنهادی را گزارش نمایید. سپس IoU پنجره‌ها را به دست آورید و با توجه به آستانه‌ای که انتخاب کردید، تعیین کنید که پنجره جزو پس زمینه حساب می‌شود یا صورت بازیکنی در آن هست. توجه داشته باشید که صرفاً محاسبه IoU پروپوزال‌ها در این سوال به چالش کشیده شده است. در آخر بعد از اینکه کلاس پنجره‌ها تعیین شد، به صورت رندوم ۵ تا از پنجره‌ها را به همراه نوع کلاس آنها نمایش دهید. (۶۰ نمره)

برای مثال:



Class: Face



class: Background

پاسخ ۲:

سوال ۴:

در نوت بوک پیوست شده (Unet.ipynb) به سوالات داخل نوت بوک پاسخ دهید. (۳۰ نمره)

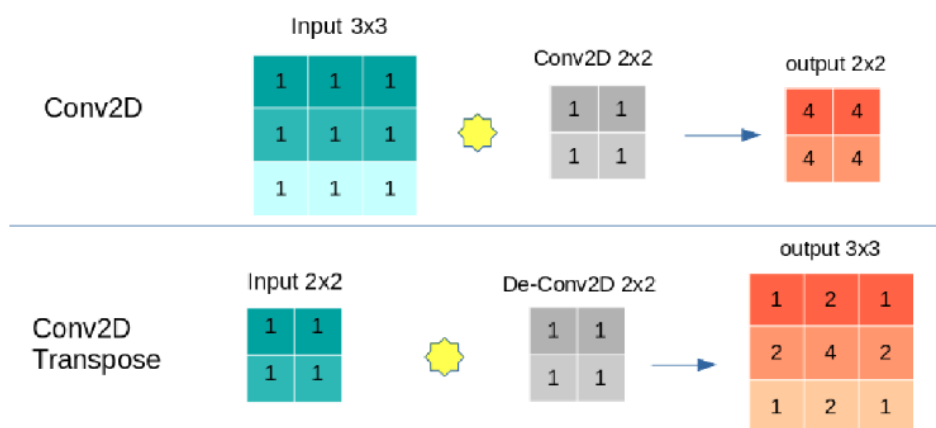
پاسخ ۴:

۱- برای padding دو حالت داریم: same و valid. در صورتی که padding برابر با same باشد، یعنی zero padding داریم و سائز خروجی و ورودی برابر است. در غیر این صورت، یعنی zero padding نداریم و سائز ورودی پس از اعمال لایه کانولوشنی کاهش می‌یابد.

۲- تابع فعال‌سازی با محاسبه مجموع وزن‌ها و اضافه کردن بایاس به آن تصمیم می‌گیرد که آیا یک نورون باید فعال شود یا نه. توابع فعال‌سازی معادلات ریاضی هستند که خروجی یک مدل شبکه عصبی را تعیین می‌کنند. توابع فعال‌سازی همچنین تأثیر عمده‌ای بر توانایی شبکه عصبی برای همگرایی و سرعت همگرایی دارند، یا در برخی موارد، توابع فعال‌سازی ممکن است در وهله اول از همگرایی شبکه‌های عصبی جلوگیری کنند. تابع فعال‌سازی همچنین به normalize کردن خروجی هر ورودی در محدوده بین ۱- تا ۱ یا ۰ تا ۱ کمک می‌کند. بنابراین تابع فعال‌سازی بخش مهمی از یک شبکه عصبی مصنوعی است. آنها تصمیم می‌گیرند که آیا یک نورون باید فعال شود یا نه و این یک تبدیل غیرخطی است که می‌تواند روی ورودی قبل از ارسال آن به لایه بعدی نورون‌ها یا نهایی کردن خروجی انجام شود.

۳- kernel_initializer روش مقداردهی اولیه مورد استفاده برای مقداردهی اولیه در کلاس Conv2D را قبل از آموزش واقعی شبکه کنترل می‌کند. این initializer برای ماتریس وزن kernel است.

۴- Conv2D برای detect کردن featureها معمولاً استفاده می‌شود و ممکن است ورودی را کوچک کند. برعکس، Conv2DTranspose برای ایجاد ویژگی به کار می‌رود. به طور خلاصه، Conv2D برای تشخیص ویژگی‌ها است و ممکن است ورودی را کاهش دهد؛ درحالی‌که Conv2DTranspose ورودی را بزرگ کرده و برای ایجاد ویژگی است. در شکل زیر، عملکرد هر یک از این موارد آورده شده است.



۵- U-Net مانند یک حرف U با encoder، bottleneck و decoder بین آنها شکل می‌گیرد. هر یک از این توابع، نقش یکی از موارد ذکر شده در بالا را دارد. تابع double_conv_block دارای دو لایه کانولوشنی متوالی است که هم در encoder و هم در bottleneck استفاده می‌شود. تابع downsample_block برای downsampling یا استخراج ویژگی تعریف می‌شود تا در encoder استفاده شود. این تابع، تابع double_conv_block را فراخوانی می‌کند تا دو لایه کانولوشنی بزند و سپس با استفاده از MaxPool و Dropout، از تعداد پارامترها و حجم محاسبات کاسته، و از overfitting جلوگیری می‌کند. در نهایت، تابع upsample_block را برای decoder تعریف می‌کنیم. در این تابع نیز ابتدا Conv2DTranspose زده شد و سپس feature کانولوشن و x پیوند داده شد و مانند تابع قبل، یک Dropout برای کاهش حجم محاسبات و جلوگیری از overfitting زدیم و در انتها دو لایه کانولوشنی قرار دادیم. عملکرد این تابع، تقریباً عکس عملکرد تابع قبل می‌باشد.

۶- بهینه‌سازها الگوریتم‌ها یا روش‌هایی هستند که برای به حداقل رساندن یک تابع خطا (loss function) یا به حداکثر رساندن کارایی تولید استفاده می‌شوند. بهینه‌سازها توابع ریاضی هستند که به پارامترهای قابل یادگیری مدل مانند وزن و بایاس وابسته هستند. بهینه‌سازها به دانستن نحوه تغییر وزن و learning rate شبکه عصبی برای کاهش تلفات کمک می‌کنند.

۷- compile معماری مدل مانند loss function، optimizer و metrics را تعیین می‌کند و هیچ کاری به وزن‌ها ندارد و می‌توانیم مدل را هر چند بار که خواستیم compile کنیم بدون آنکه مشکلی برای وزن‌های pretrained به وجود آورد.

۸- زیرا مسئله در اینجا چند کلاس است و در مسائل چند کلاسه، categorical_crossentropy عملکرد بهتری دارد.

۹- کاربرد early stopping، برای این است که اگر metric‌های مورد ارزیابی ما بهبود پیدا نکردند، training متوقف شود. مشکل شبکه‌های عصبی آموزش در انتخاب تعداد epoch‌های آموزشی برای استفاده است. epoch بسیار زیاد می‌تواند منجر به overfitting مجموعه داده آموزشی شود، در حالی که تعداد بسیار کمی ممکن است منجر به یک مدل underfit شود. early stopping روشی است که به ما امکان می‌دهد تعداد زیادی epoch آموزشی دلخواه را مشخص کنید و زمانی که عملکرد مدل در یک مجموعه داده validation متوقف شد، آموزش را متوقف کنید.

۱۰- fit در واقع مدل را طوری آموزش می‌دهد که تمام پارامترها را به مقدار صحیح برساند تا ورودی‌های خود را به خروجی‌های خود نگاشت کنیم. از تفاوت‌های compile و fit، می‌توان به این مورد اشاره کرد که compile تنها معماری مدل را مشخص می‌کند و به پارامترهای آن کاری ندارد؛ اما با fit کردن، training مدل شروع می‌شود و پارامترهای آن update می‌شوند.

۱۱- batch size یک hyperparameter است که تعداد نمونه‌هایی را که باید قبل از به روز رسانی پارامترهای داخلی مدل کار کنند، تعیین می‌کند. تعداد epoch، یک hyperparameter است که تعداد دفعاتی را که الگوریتم یادگیری در کل مجموعه داده آموزشی کار می‌کند را تعیین می‌کند. از تفاوت‌های این دو، می‌توان به موارد زیر اشاره کرد.

- batch size تعداد نمونه‌هایی است که قبل از به روز رسانی مدل پردازش شده اند و epoch تعداد pass‌های کامل از مجموعه داده آموزشی است.

- اندازه یک batch، باید بیشتر یا مساوی یک و کمتر یا مساوی تعداد نمونه‌های مجموعه داده

آموزشی باشد؛ در حالیکه تعداد epoch را می توان به یک مقدار صحیح بین یک و بی نهایت تنظیم کرد.

لینک های مورد استفاده:

<https://www.mygreatlearning.com/blog/activation-functions/>
<https://stackoverflow.com/questions/68976745/in-keras-what-is-the-difference-between-conv2dtranspose-and-conv2d>
<https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/#:~:text=An%20optimizer%20is%20a%20function,loss%20and%20improve%20the%20accuracy.>
<https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>
<https://pyimagesearch.com/2022/02/21/u-net-image-segmentation-in-keras/>