بسم الله الرحمن الرحیم

محمد عرفان زارع زردینی

۹۸۴۱۱۱۴۳۲

تمرین سری پنجم درس بینایی ماشین

1)

Refrences:

[The Vanishing/Exploding Gradient Problem in Deep Neural Networks | by Kurtis Pykes | Towards Data Science]

https://cafetadris.com/blog/%D8%B1%D8%B2%D9%86%D8%AA-resnet/

https://datagen.tech/guides/computer-vision/resnet/#:~:text=What%20Is%20Residual%20Neural%20Network,or%20thousands%20of%20convolutional%20layers.

الف)

این از مشکلاتی است که هنگام آموزش شبکه عمیق با آن برخورد میکنیم. در شبکه ای دارای n لایه پنهان ، n مشتق با هم ضرب می شوند.

گرادیان انفجاری:

اگر مشتق ها بزرگ باشند، هنگام انتشار شیب به شکل تصاعدی زیاد میشه و در نهایت مشکل گرادیان انفجاری رخ می دهد . پیرامون آن ، انباشت مشتق های بزرگ سبب ناپایداری مدل شده و در کاهش یادگیری مدل ،موثر هست. همچنین تغییر زیاد در وزن مدل ها ، سبب ایجاد شبکه ای ناپایدار شده که که مقادیر وزن ها را به قدری زیاد میکند که سرریز رخ داده و مقادیر وزن NaN شده و دیگر نمیتوان بروزشان کرد.

مدل دارای این مشکل روی داده های train ، چیز زیادی یاد نمیگیرد، درنتیجه loss ضعیفی دارد.همچنین به دلیل ناپایداری مدل ها هنگام بروز رسانی، تغییرات زیادی در loss دارد.

از علائم این مشکل:1– افزایش تصاعدی وزن مدل و بزرگ شدن مدل هنگام آموزش 2–وزن مدل ها هنگام آموزش به NaN تبدیل می شود. 3– مشتق ها عدد ثابت(CONSTANT) اند.

ناپدید شدن گرادیان:

حال اگر مشتق ها کوچک باشند، درحالی که مدل انتشار میابد، شیب به شکل تصاعدی کم شده تا درنهایت ناپدید شده و همان مشکل ناپدیید شدن گرادیان رخ می دهد.

در واقع انباشت شیب های کوچک، سبب مدلی شده که توانایی یادگیری مدل مغناداری را ندارد، زیرا وزن ها و سوگیری های لایه های اولیه که تمایل به یادگیری ویژگی های اصلی از داده های ورودی دارند، به روز نمی شوند. در بدترین حالت گرادیان صفر خواهد بود که سبب متوقف شدن شبکه می شود و آموزش بیشتر را متوقف می کند.این مدل در مرحله train بسیار آهسته بهبود میابد و ممکن است آموزش خیلی زود متوقف شود .یعنی آموزش های بعدی باعث بهبود مدل نمی شود. وزن های نزدیکتر به لایه خروجی مدل شاهد تغییرات بیشتری خواهند بود در حالی که لایه هایی که نزدیکتر به لایه ورودی اند تغییر زیادی ندارند. وزن های مدل هنگام آموزش در این فرمت به شکل تصاعدی کاهش میابد و در نهایت بسیار کوچک می شود.

ب)

Resnet معماری شبکه عصبی عمیق است که برای پوشش مشکل ناپدید شدن گرادیان که در شبکه های بسیار عمیق رخ می دهد، ایجاد شده است .این مشکل زمانی رخ می دهد که گرادیان ها(مقادیری که هر نورون نشان می دهد که در طول train چقدر باید تنظیم شود) با انتشار به سمت عقب ، در شبکه بسیار کوچک می شود.بدین سان وزن ها در لایه های قبلی شبکه ممکن است به شکل موثر بروز نشوند و در نتیجه شبکه داده های ورودی را به درستی نمایش و یاد نگیرد. حال resnet با معرفی نوع جدیدی از بلوک باقی مانده، به شبکه دسترسی می دهد تا به جای توابع کامل، توابع باقی مانده را بیاموزد و مشکل بدین گونه حل می شود.در یک بلوک باقی مانده ، ورودی به جای آنکهمستقیم به لایه کانلوشن متصل شود ،به خروجی مجموعه ای از لایه های کانولوشن اضافه می شود. بدین سان شبکه امکان تشخیص تابع باقیمانده(تفاوت بین ورودی و خروجی) را راحت تر یاد بگیرد ، میتواند به جلوگیری از مشکل گرادیان ناپدید شدن کمک کند.

این معماری با مجموعه ای از لایه های کانولوشن شروع شده که وظیفه استخراج ویژگی ها را روی تصویر ورودی دارد. این لایه توسط یک سری بلوک باقیمانده(residental block) دنبال می شوند که به شبکه اجازه اموختن ویژگی های پیچیده تر را می دهد. در پایان خروجی بلوک باقی مانده از میان مجموعه ای از لایه های کاملا متصل عبور داده می شود تا خروجی نهایی تولید شود. Resnet امکان ساخت شبکه های غصبی بسیار عمیق را فراهم می نماید که سبب عملکرد بهتری در طیف وسیعی از طبقه بندی تصویر دست پیدا کند.rexnet در واقع با یادگیری residental function توانست بر مشکل پراکندگی غلبه و آموزش موثر تر شبکه های بسیار عمیق را امکان پذیر نماید.

2)

Refrences:

الف)

پارامترهای محاسبه شده:

برای هر بلوک از لایه اول،(سه تا بلوک است) ، 32*3*1=96 پارامتر برای هرکانال داریم و درنتیجه برای هر بلاک در مجموع: 36*96=3072 پارامتر داریم. مجموع پارامتر این سه بلوک متصل به ورودی :3*3072=9216 است.

برای هر بلوک لایه دوم ، برای هرکانال 32*32*3*3=9216 .برای هر بلوک هم در مجموع 32*9216=294912 پارامتر است که در نتیجه برای مجموع لایه دوم(دو بلوک است) 294912*2=589824 پارامتر است.

برای بلاک تنها لایه شماره 3: برای هر کانال 32*32*3*3=9216 و برای بلوک مان، 32*9216=294912 پارامتر است.

برای تک بلوک لایه آخر: برای هر کانال 256*1*1*96=24576 و برای این لایه256*24576=6291456 پارامتر داریم.

درنتیجه مجموع همه پارامتر ها می شود:

7274288=9216+589824+294912+6291456

میدان تاثیر :

میدان تاثیر براساس عمق بلوک در شبکه است و برای هر بلوک داریم:

لایه ورودی: 1*1

سه بلوک لایه اول:1*1

دو بلوک لایه دوم:3❁3

یک بلوک لایه سوم: 5❁5

یک بلوک لایه چهارم:4❁4

ب)

هر دو حالت مان میدان تاثیر برابر است و 3❁3 است.(مطابق قسمت قبل)

برای پارامتر:

قسمت اول:448 برای لایه اول و برای لایه دوم 4640 پس تعداد پارامتر ها می شود:5088

قسمت دوم: برای لایه اول 448 و برای لایه دوم 14656 می باشدد که در مجموع 15104 است.

نکته: استفاده از لایه دوبعدی locallyconnected سبب تولید پارامتر بیشتری می شود و درنتیجه حافظه زیادی مصرف می شود.

در هر دو مدل ، دو لایه از فیلتر های 3❁3 وجود دارد ، پس برای محاسبه مقدار هر نورون یا پیکسل در لایه خروجی، به شکل غیر مستقیم و با استفاده از لایه اول و دوم ، از یک ناحیه 7❁7 از لایه ورودی استفاده می شود.

3)

Refrences:

https://keras.io/examples/vision/mnist_convnet/

https://keras.io/guides/transfer_learning/

https://www.tensorflow.org/tutorials/images/data_augmentation

الف)

ابتدا داده های train و تست را ازز دیتاست گفته شده خوانده(لود میکنیم)، همچنین در کد تعدادی از آن ها را نمایش دادیم (هرکدام نماینده یک کلاس از 10 کلاس اند) سپس مدل شبکه ای با سه بلوک با فرمت (cnn2d,cnn2d,maxpool2d,batch normalization) تشکیل شده و دو لایه denseهم در نهایت دارد.شبکه را با تابع ضرر CategoricalCrossentropy و بهینه ساز adam کامپایل نموده و آن را طبق خواسته سوال در 10 epoch با بچ سایز دلخواه(من 256 گذاشتم) برروی داده های train آموزش میدهیم. نتایج در کد موجود است.

ب)

برای داده افزایی ازز دوروش وارون عمودی و چرخاندن تصادفی استفاده و عملیات داده افزایی را مانند یک لایه متوالی تعریف و آن را در مدل بعد از ورودی قرار میدهیم.(داده فقط هنگام اموزش روی ورودی ها اعمال می شود ولی در validation استفاده نمیشود.) نتایج اموزش را در کد میتوانید مشاهده نمایید.

ج)

در هردو بخش ، مقادیر loss وaccuracy برای داده های validation تغییرات زیادی دارد و همچنین با مقادیر داده آموزشی هم فاصله دارد که نشانه overfit شدن شبکه است.بدون داده افزایی ، شبکه روی داده های آموزشی به دقت حدود 0.59 رسیده است وvalidation آن نیز به دقت حداکثر 0.5 دست یافته. درحالی که با داده افزایی دقت داده های تست پایین امده و خیلی نوسانی شده است. دقت داده های آموزشی حدود0.15 کاهش یافته و میشود اشاره نمود که underfittingدر مدل مشاهده می شود.

د)

مدل رزنت خواسته شده با وزن ها تعریفی روی شبکه تصاویر لود شده و مدل جدیدی تعریف می شود که رزنت ابتدای آن بوده و سپس لایه cnn2d برای کم کردن ابعاد و لایه flatten,dense برای کلاس بندی قرار گرفته. نتایج اموزش این مدل نیز در کد موجوده.(از batchnormalizationهم برای نرمالایز کردن استفاده نمودم.)

در خروجی ، نمودار داده های valid ، کم نوسان تر شده و خطی تر است ولی همچنان overfitting و مقداری underfitting قابل مشاهده است که از دلایل آن گستردگی نوع داده ها نسبت به تعداد داده ها

است که سبب می شود مدل در پیدا کردن ویژگی های قابل تعمیم دچار چالش و مشکل بشود و سفت بتواند پیدا نماید.

البته در برخی قسمت ها دقت داده های آموزشی و تست به هم رسیده ولی هم گذر نکرده که این از نکات قابل توجه مدل بررسی شده است .

4)

Refrences:

https://wandb.ai/ayush-thakur/dl-question-bank/reports/What-s-the-Difference-Between-Strided-Convolution-and-Pooling---VmlldzoyMDE5Mjc

https://medium.com/analytics-vidhya/convolution-padding-stride-and-pooling-in-cnn-13dc1f3ada26

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network#:~:text=A%20CNN%20is%20a%20kind,the%20network%20architecture%20of%20choice.

https://medium.com/@datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2

https://towardsdatascience.com/what-is-wrong-with-convolutional-neural-networks-75c2ba8fbd6f

https://iq.opengenus.org/disadvantages-of-cnn/

الف)

در شبکه های عصبی کانولوشنال یا همانCNN ، گام به اندازه ای اشاره دارد که هسته کانولوشنال مان هنگام لغزش روی دیتا های ورودی انجام می دهد. هنگامی که گام بزرگتر از 1 باشد، هسته برخی از داده های ورودی را ignoreمی کند و در نتیجه حجم خروجی کمتری داریم.

مفهوم گام با ادغام(pooling) متفاوت است ، ادغام اندازه فضایی حجم خروجی را نیز کاهش می دهد ولی این کار را با کاهش نمونه گیری داده های ورودی بوسیله عملیات ثابت مانند ادغام حداکثر یا متوسط(max pooling or average pooling) .ادغام همچنین بعد از لایه های کانولوشنی اعمال می شود، در حالیکه گام پارامتری از خود لایه کانولوشنی است.

تاثیرات گام (stride) بر عملکرد CNN به معماری و وظیفه آن وابسته است . یک گام بزرگتر اگر استفاده شود، وضوح فضایی خروجی را کاهش می دهد که برای برخی از کارها که جزئیات کوچک مهم نیست ،همچون طبقه بندی تصاویر مفید باشد. با این حال استفاده از گامی بزرگتر، میتواند سبب از دست رفتن داده ها و کاهش دقت مدل شود.همچنین استفاده از گامی کوچک تر،اطلاعات بیشتری را حفظ می کند و برای کارهایی همچون تشخیص اشیا و تقسیم بندی ضروری است. البته استفاده از گام کوچک ، می تواند سبب افزایش پیچیدگی محاسباتی و سبب کاهش سرعت آموزش و استنتاج شود. پس، انتخاب مقدار گام باید با دقت بر اساس نیاز های خاص و محاسباتی مسئله انجام شود.

ب–1)

برای لای های میانی یکcnn که برای طبقه بندی تصویر کاربرد دارد، توابع فعال سازی ReLU(واحد خطی اصلاح شده) و انواع آن مانند leaky reluو elu اند. این توابع فعالسازی در cnn های عمیق به خوبی عمل کرده و نشان داده شده است که با کاهش مشکل ناپدید شدن گرادیان که ممکنه در هنگام استفاده از توابع فعالسازی همچون simoidیاtanh رخ دهد، اموزش را سرعت بخشد.

برای لایه آخر ، یک cnn برای طبقه بندی تصویر استفاده می شود، تابع فعالساز آن نیز براساس تعداد کلاس های سوال است. اگر دو کلاس تنها موجود بود(معیوب و سالم)، از sigmoid و اگر بیشتر بود از softmax

استفاده می شود که خروجی cnn را برای نمایش احتمالات کلاس نرمالایز می نماید. دلیل استفاده از این تابع فعالساز، این است که یک تفسیر احتمالی از خروجی شبکه ارائه می دهد و امکان مقایسه بین احتمالات پیش بینی شده کلاس مختلف را فراهم مینماید.

در کل برای طبقه بندی تصاویر معیوب از سالم، از relu یا انواع آن برای لایه های میانی cnn ویک تابع sigmoidیاsoftmax برای آخرین لایه بسته به تعداد کلاس های سوال ، مورد استفاده است.

ب–2)

برای مسئله زبقه بندی تصاویر محصولات معیوب از سالم، یک تابع خطا مناسب binary cross entropy

است. این تابع خطابرای مسائل طبقه بندی باینری مانند این مورد استفاده می شود و تفاوت میان احتمالات کلاس پیش بینی شده و برچسب کلاس های واقعی را اندازه گیری میکند و مدل را اگر پیش بینی نادرست انجام داده باشد ، برایش penalty در نظر میگیرد. تابع خطای آن نیز مانند زیر تعریف می شود:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

در آن y لیبل درست است (0برای سالم و 1 برای معیوب)، ^y احتمال پیش بینی شده معیوب بودن است. زیان زمانی به حداقل می رسد که احتمالات پیش بینی شده با برچسب های واقعی مطابقت داشته باشند.

استفاده از این تابع برای این مسئله مناسب است چون تابع خطا به خوبی منتشر شده است و برای طبقه بندی باینری کاربردی و موثر است. در cnn ها کاربرد دارد و از نظر محاسباتی برای مناسبه طول آموزش کارآمد است. علاوه بر این ، معیاری واضح وقابل تفسیر برای ارزیابی عملکرد مدلمان ارائهمی نماید چون که تفاوت میان احتمالات پیش بینی شده وبرچسب واقعی اندازه گیری می نماید.

ب–3)

برای بالا بردن دقت تشخیص محصولات سالم از معیوب، باید منفی های کاذب را به حداقل رسانیم(در اینجا محصولات معیوبی است که به اشتباه سالم پنداشته شده) و در نتیجه برای بررسی بیشتر یا حذف از خط تولید، علامت نمیخورند. در نتیجه در صورتی که تعداد زیادی منفی کاذب را علامت گذاری نشوند، بدین شکل محصولات معیوب زیادی به دست مشتری میرسند.برای حداقل کردن منفی های کاذب ، باید روی ماکزیمم کردن recall تمرکز شود.که به عنوان نسبت های مثبت های واقعی بدرستی شناسایی شده اند. به عبارت دیگر اندازه گیری recall ، توانایی مدل در شناسایی تمامی محصولات معیوب در خط تولید است . مقدار recall بالا ، بدین معنا است که مدل در تشخیص محصولات معیوب موثر است، که به نوبه خود تعداد منفی های کاذب را به حداقل می رساند.

البته، ما باید precision را نیز به عنوان نسبت مثبت های پیش بینی شده (محصولات معیوب) در نظر بگیریم. یک مقدار precision بالا بدین معناست که مدل در پیدا کردن محصولات معیوب دقیق است.این کار برای به حداقل رساندن مثبت های کاذب(محصولات سالم که به اشتباه معیوب تشخیص داده شده اند) مهم است. با توجه به اهمیت مینیمایز کردن منفی های کاذب در این بخش، ما باید به حداکثر رساندن recall اولویت دهیم، در حالی که از سطح معقولی از precision نیز اطمینان داشته باشیم. با تنظیم استانه تصمیم مدل(که استانه احتمالی را که بالاتر از آن مدل رودی را معیوب تشخیص می دهد) ،میتوانیم میان recallو precision بدست آوریم. با تنظیم آستانه میتوانیم میان recall و precision ، trade-off داشته باشیم.البته ، مقدار آستانه خاصی که با آن به سطح مطلوب عملکرد دست میابیم، به محدودیت ها و لزومات خاص و هزینه های مربوط به مثبت و منفی کاذبی که تعریف نموده ایم وابسته و بستگی دارد.

ج1)

مناسب نیست.زیرا این شبکه عمدتا برای کارهای تجزیه و تحلیل تصویر و ویدئو طراحی شده است و برای پردازش داده های متنی مناسب نیست. به جای آن، روش های پردازش زبان طبیعی(NLP) ، همچون شبکه عصبی تکراری (RNN) ، ویا شبکه عصبی کوتاه مدت(LSTM) یا ترانسفورماتورها، برای کارهای طبقه بندی متن مناسب ترند. این مدل ها برای پردازش داده های متوالی مانند متن ،طراحی شده اند و وابستگی بین کلمات و عبارات مختلف را میتواند در یک جمله ثبت کند.(که برای طبقه بندی بسیار ضروری است.)

ج2)

ممکن است مناسب نباشد و در شناسایی گوینده از روی صدا موفق نباشد. چون همانطور که گفته شد،CNN برای تجزیه و تحلیل تصاویر و ویدئو کاربرد دارد و برای داده های صوتی استفاده نمی شود. برای شناسایی گوینده از روی صدا ، انواع دیگر شبکه های عصبی مانند RNN یا شبکه عصبی کوتاه–بلند مدت (LSTM) ممکن است مناسب باشند. این شبکه ها برای تجزیه و تحلیل متوالی داده ها ،همچون سیگنال های صوتی طراحی شده اند و میتوانند وابستگی های زمانی میان نمونه های صوتی را ضبط کنند .

ج3)

شبکه مدنظر **ممکن** است برای تحلیل جدول مربوط به مشتریان و پیش بینی رفتار بعدی شان موفق نباشد. چون همانطور که در دو قسمت قبل گفتیم، CNN عمدتا برای تجزیه و تحلیل تصویر و ویدئو طراحی شده و برای پردازش داده های جدولی کاربردی ندارد و مناسب نیست . به جای آن می توان از انواع دیگر مدل های یادگیری ماشین همچون ،درخت تصمیم ،gradient boosting machine، random forest

استفاده نمود که برای این نوع تحلیل مناسب تر اند.این مدل ها میتوانند به شکل پیچیده تری روابط میان ویژگی های مختلف را به تصویر بکشند و پیش بینی های دقیق تری را ارائه دهند.

د)

مدلCNN مدل محبوب در یادگیری عمیق است که معمولا برای تشخیص و انالیز ویدئو وتصویر کاربرد دارد. از مشکلات آن میتوان به موارد زیر اشاره کرد:

1: تطبیق بیش از حد (overfitting) : CNN ها مستعد برازش بیش از حد و overfitting اند.این اتفاق زمانی رخ می دهد که مدل بیش از حد پیچیده می شود و به جای الگویی اساسی ، شروع به حفظ داده های آموزشی می کند . overfitting می تواند منجر به عملکرد ضعیف در داده های جدید و دیده نشده شود .

2:limited translation invariance: در حالی که مدل cnn به گونه ای طاحی شده است که translation invariant باشد بدین معنا که می تواند اشیا را بدون توجه به موقعیت آنها در تصویر

تشخیص دهد، این درحالی است که کاملا invariant نبوده و میتواند تحت تاثیر چرخش های جزئی شی قرار گیرند. این میتواند در برخی موارد سبب کاهش دقت شود.

3:توانایی محدود در رسیدگی به اشیا کوچک: مدل cnn میتوانند اشیا کوچک یا جزئیات رو در تصویر تشخیص بدهند، بویژه اگر با نویز یا در تصویر ترکیب شده باشد و در پس زمینه احاطه شده باشد که این مورد در مواردی که اجسام کوچک مهم اند مانند تصویر برداری پزشکی یا میکروسکوپی، مشکل ساز باشد .

4: توانایی محدود در رسیدگی به انسداد: انسداد ها زمانی رخ می دهند که یک شی کاملا توسط شی دیگر مبهم ، محدود می شوند .این میتواند تشخیص شی را برای شبکه دشوار کرده و سبب طبقه بندی اشتباه شود.

5:مشکل در مدیریت ورودی هایی با اندازه متغیر: مدلcnn معمولا برای کار با ورودی های با اندازه ثابت طراحی شده اند ، که در نتیجه می تواند در برخی از برنامه ها که اندازه ورودی ممکن است تفاوت کند ، محدودیت باشد. تکنیک های متفاوتی برای مدیریت این موضوع وجود دارد ولی سبب اضافه شدن پیچیدگی به مدل شوند.

6:ادغام لایه ها(pooling layer) : این کار سبب ازبین رفتن اطلاعات ارزشمندی می شود و رابطه بین جز و کل را نادیده می گیرد. و در برخی قسمت ها مانند تشخیص چهره، ممکن است با استنباطی بر این اساس دو ورودی متفاوت را مشابه تشخیص دهد و لیبل بزند که درست نیست.

این ها برخی از مشکلاتی بود که این مدل میتواند داشته باشد و البته این مدل توسط دانشمندان هرسال بروز و تکمیل تر می شود تا به مدلی بهتر دست یابیم.

5)

Refrences:

most of link in code question

ابتدا دیتاست رو دانلود و از فرمت زیپ خارج میکنیم، عکس هایی ورودی و segmentشده را در فولدر های جدا قرار داده و آدرس آن را در دیتافریم ذخیره می کنیم. سپس دیتا فریمماتن به tensorflowد تبدیل و

عمس های ورودی و خروجی خوانده شده ، تغییر سایز داده و نرمالایز می شود .دیتاست نیز به داده های اموزش وتست تقسیم و بچ می شوند.(داده اموزشی shuffle هم میکنیم.)

برای ساختن unet ابتدا شبکه mobilenetv2 را لود و سپس یک مدل backbone تعریف نموده که ورودی آ« خروجی لایه ورودی mobilenet و خروجی آن لیست خروجی های لایه قبل maxpool شبکه موبایل نت است . شبکه upsample وظیفه افزایش ابعاد داده ورودی را دارد . در نهایت هم هر backbone را به یک upsample نظیر و مدل خواست هشده را برای خروجی اعمال و داده ها را به ابعاد اولیه تصاویر میبریم. حال مدلی براساس آنها تعریف وکامپایل کرده و برروی داده اموزش میدیم. نمونه هایذ خروجی هم در شکل موجوده