بسم الله الرحمن الرحیم

محمد عرفان زارع زردینی

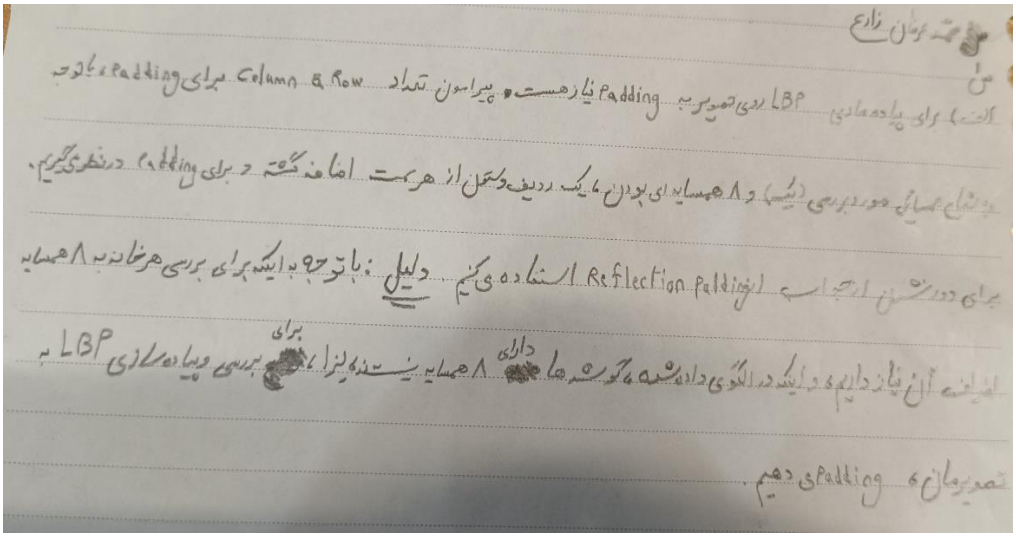
۹۸۴۱۱۱۴۳۲

تمرین سری پنجم درس بینایی ماشین

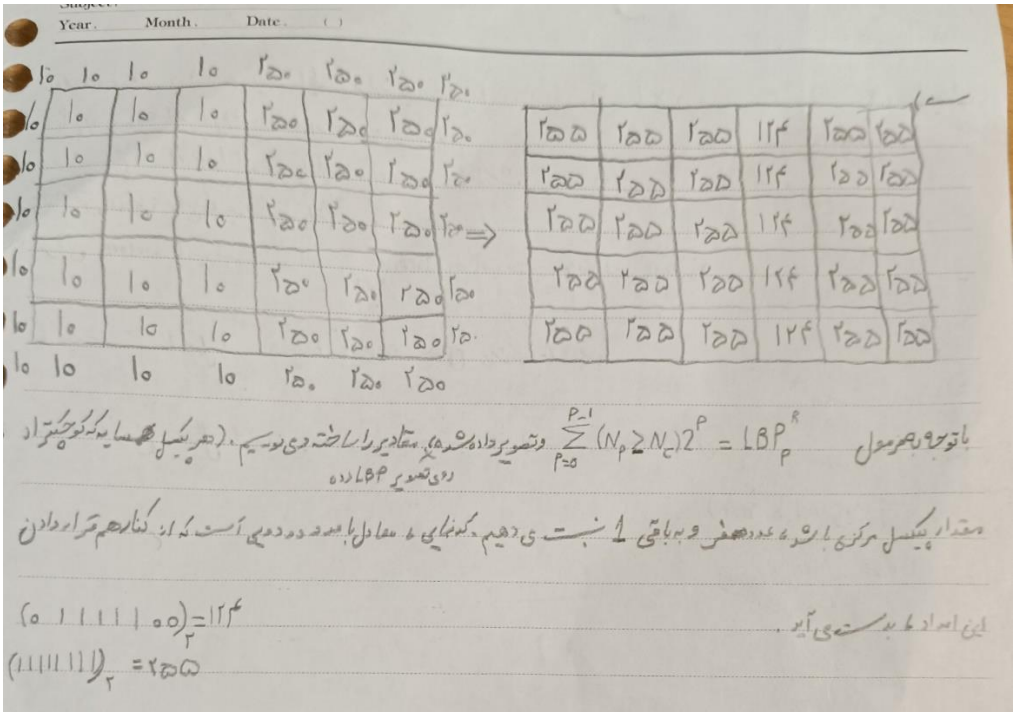
سوال 1)

Refrences: Slides class

الف)



ب)



سوال 2)

Refrences:

https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html

https://medium.com/analytics-vidhya/contours-and-convex-hull-in-opencv-python-d7503f6651bc

https://docs.opencv.org/3.4/de/d62/tutorial_bounding_rotated_ellipses.html

برای یافتن ویژگیcompactness شکل، محیط و مساحت را میابیم. مساحت با تابع contourarea بدست آورده می شود. محیط نیز با محاسبه فاصله نقاط مجاور شکل و جمع آنها یافته می شود. مساحت را در 4p ضرب و در مربع محیط تقسیم میکنیم.

برای یافتن ویژگی solidity ،مساحت را مانند قسمت قبل یافته،محیط را با دستور convexhull که محیط شکل محدب را میابیم.مساحت را بر مساحت محدب محیط، تقسیم و خروجی میدهیم.

برای یافتن طول محور اصلی و فرعی در تابع axis_find،از تابع fitellipse استفاده می شود، همچنین جهت شکل را هم میدهد.چون ورودی تابع شامل مینیمم5 نقطه

است، نقاط مرزی را با ترکیب خطی نقاط کانتور پیدا و به fitellipse میدهیم.مقدار خروجی این تابع از حاصل تقسیم طول محور فرعی بر اصلی بدست می آید.

تابع info_contour ، یک کانتور را دریافت و مشخصه های آن را با توابع بالا یافته و در قالب یک ndarray خروجی میدهد.

تابع dist_era ،فاصله بین دوبردار ویژگی با روش اقلیدس را بوسیله linalg.norm محاسبه می کنیم. مقادیر اختلاف ویژگی های دوبردار را در ضرایبی ضرب کرده تا گروه بندی بهتر انجام شود.

حال برای یافتن و نمایش ویژگی، با توابع بالا به فرم یک ماتریس n*m تبدیل کرده که n تعداد شکل و m ویژگی است.درنهایت هم با توابع نشان شده عمل دسته بندی را انجام داده و به هر دسته مشخصه ای رنگی معین میدهیم.

سوال 3)

Refrences:

https://machinelearningmastery.com/using-activation-functions-in-neural-networks/#:~:text=Activation%20functions%20play%20an%20integral,a%20simple%20linear%20regression%20model.

https://himanshuxd.medium.com/activation-functions-sigmoid-relu-leaky-relu-and-softmax-basics-for-neural-networks-and-deep-8d9c70eed91e

https://www.aitude.com/comparison-of-sigmoid-tanh-and-relu-activation-functions/

توابع فعالساز، در شبکه عصبی برای معرفی غیرخطی بودن خروجی هر نورون استفاده می شود. هدف اصلی استفاده از توابع فعالساز، تبدیل خروجی یک نورون به فرم مناسبی است که بتوان از آن به عناون ورودی لایه بعدی شبکه عصبی استفاده نمود.بدون توابع فعالساز، شبکه عصبی مانند یک مدل رگرسیون خطی است که قادر به آموزش الگوهای غیرخطی پیچیده در دیتاها نیست.

چندین توابع فعالساز مانند آن هایی که در صورت سوال بیان شده داریم که انتخاب تابع فعالساز براساس تسک خاص و ماهیت داده مورد استفاده، بستگی دارد.

تابع فعالساز sigmoid تابعی رایج است که بیشتر در لایه خروجی یک شبکه عصبی برای طبقه بندی باینری(binary classification) استفاده می شود. مقادیر ورودی در محدوده میان 0 و1 است که میتوان آن را به عنوان احتمال یک طبقه بندی باینری بیان نمود. این درحالیست که تابع sigmoid یک مشکل به صفر رسیدن گرادیان (vanishing) که سبب همگرایی کُند، در مدت آموزش می شود و میتواند یادگیری الگوهای پیچیده را برای مدل دشوار نماید.

تابع Tanh مشابه تابع sigmoid است، اما مقادیر ورودی را در محدوده ای بین 1- و 1 نسبت بندی میکند. این تابع فعالسازی میتواند سبب کاهش مشکل به صفر رسیدن گرادیان می شود اما باز هم همان مشکل را دارد.

تابع ReLU(واحد خطی اصلاح شده) تابع فعالسازی است که به دلیل سادگی و کارآمدی کاربرد بیشتری پیداکرده است و بدین گونه است که تمام ورودی های منفی را صفر کرده،مقادیر ورودی مثبت را نیز بدون تغییر میگذارد. این به شبکه امکان می دهد که سریعتر اموزش بیند .به این دلیل که از مشکل صفر شدن گرادیان جلوگیری می نماید و سبب نمایش کمتر دیتا ها می شود. با این حال این تابع میتواند مشکل dying ReLU را داشته باشد.این مشکل سبب می شود بخش بزرگی از شبکه، غیرفعال باشد و یادگیری را متوقف می کند.

تابع PreLUگونه ای از تابع ReLU است که به جای صفر کردن آن ها، پارامتری قابل یادگیری را به مقادیر ورودی منفی معرفی می نماید.این کار سبب کاهش dying ReLU می شود و عملکرد شبکه بهبود می یابد.

در کل توابع sigmoidو Tanh برای طبقه بندی باینری مناسب اند، توابع ReLU و PreLU برای آموزش شبکه عصبی عمیق کاربردی تر است و سبب همگرایی سریعتر و عملکرد بهتر میشود.

سوال 4)

Refrences:ta’s classes code

Code question

ابتدا دیتاست mnist را بارگزاری میکنیم و داده های تست و داده تمرینی را ذخیره میکنیم. بعد داده های تمرینی را به همراه خروجی بررسی و چاپ میکنیم.(در کد هست)

سپس چون داده هایمان بین 0تا 255 پیسکل اند ، ان را به بازه 0تا1 میبریم.همچنین تمام خروجی ها توسط تابع زیرمجموعه keras به فرم one_hot میبریم.

در دو روش توالی و توابعی، ما لایه های یگسان را می سازیم. لایه هایمان شامل ورودی ،flatten(ورودی تعریفی از لایه قبلی را از دوبعدی خارج می کند.توسط لای های fully connected قابل دریافته .)،لایه پنهانی کاملا متصل شده با 100 نورون،تابع فعالساز relu ، و لایه خروجی کاملا متصل شده که شامل 10کلاس است و درنهایت با تابع نهایی نرمالایز میکنیم که جمع برابر 1 شه.

در روش توالی یک شی ساخته و لایه ها را افزوده و مدل را کامپایل میکنیم و داده تمرینی را با fit اموزش میدهیم.همچنین epoch و ارگومان validation_split را مشخص مینماییم.(این ارگومان مقدار برابری که برایش گذاشته است جدا میکند)

خروجی داده ها در کد هست. مدلمان دچار overfit شده، دلیل اختلاف میان دقت داده تمرینی و validation ، سادگی داده نسبتا کوچک هست.

در روش تابعی،تابعی را تعریف کرده و داده های ورودی توسط لایه ای ذخیره شذه و لایه های بعدی به شکل تابعی برروی متغیر اعمال می شود.مزیت این روش نسبت به روش دیگر ،امکان اتصال لایه ها با اوالویت(ترتیب) غیرترتیبی است.سپس تابع را فراخوانده و مدلی ساخته و مانند مدل ترتیبی کامپایل کرده و برروی داده تمرینی اموزش داده و روی داده تست امتحان میکنیم.

دیده می شود به دلیل استفاده از لایه های یکسان ، دو مدل تقریبا عملکردی یکسان دارند.

سوال 5)

Refrences:

https://www.analyticsvidhya.com/blog/2021/07/understanding-sequential-vs-functional-api-in-keras/

https://medium.com/analytics-vidhya/keras-model-sequential-api-vs-functional-api-fc1439a6fb10

نه. همه شبکه هایی که با متد functional میتوان پیاده سازی کرد ،نمی شود با متد sequential پیاده سازی نمود. دلیل آن این است که متد sequential یک فرم محدود از متد functional است که تنها به معماری های خطی stack مانند، کاربرد دارد . در واقع لایه ها در یک مدل متوالی به ترتیب پس هم اضافه شده و هر لایه از خروجی لایه قبلی ورودی می گیرد . همچنین متد functional انعطاف پذیری بیشتری را در تعریف ساختار معماری شبکه را امکان سازی میکند و مدل های پیچیده تر و غیرخطی را امکان مدل سازی میدهد. همچنین این امکان را فراهم میکند که خروجی یک لایه بتواند به عنوان ورودی چندین لایه بدان متصل شود. این در حالی است که این رخداد در

مدل ترتیبی ممکن نیست و فقط خروجی هر لایه به ورودی لایه بغد در stack متصل می شود.

بنابر این دلایل که بیان شد، هر شبکه ای که میتوان با متد ترتیبی(sequential) پیاده سازی شود، می تواند با متد functional نیز پیاده سازی شود، ولی لزوما برعکس این موضوع صادق نیست . زیرا برخی از شبکه ها برای پیاده سازی کارآمد،به انعطاف پذیری و وضوح(expressiveness) متد functional نیاز دارند .

سوال 6)

Refrences: slides class

الف)

اگر یک کرنل 7❁7 را با تصویرمان که یک تصویر 7❁7 با stride برابر1 است استفاده و کانوالو نماییم، خروجی یک ماتریس 1❁1❁3 است.بدلیل اینکه چه در محور افقی و چه در محور عمودی تنها یک بار میتوان لغزاند.

ب)

اگر تصویر را با سه کرنل 3❁3 در سه مرحله کانوالو کنیم، ابعاد خروجی بسته به تعداد فیلترهای استفاده شده در هر مرحله دارد. با فرض یکسان بودن تعداد فیلترها در هر مرحله،خروجی یک tensor سه بعدی با ابعاد 3*5*5 است.

ج)

استفاده از سه کرنل 3❁3 در سه مرحله بهتر است. بدین دلیل که این روش به شبکه های عمیق و غیرخطی ، امکان یادگرفتن ویژگی های پیچیده تر را می دهد . به علاوه، استفاده از چند کرنل کوچک، سبب ایجاد پارامترهای کمتری نسبت به استفاده از یک کرنل بزرگ در شبکه می شود . در این بخش، استفاده از یک کرنل 7❁7 سبب ایجاد 147 (7در7 در 3) پارامتر می شود .این در حالی است که برای بخش دوم(ب)، تعداد پارامتر برابر 84 پارامتر می باشد. پس استفاده از کرنل کوچک، سبب ایجاد شبکه با ساختاری موثر تر وبا کیفیت شده و ویژگی های استخراجی بهتر می باشد .