



تحلیل هوشمند تصاویر زیست پزشکی

سوال اول:

i.

• Two-stage object detectors:

آشکارسازهای دو مرحله ای وظیفه تشخیص شی را به دو مرحله تقسیم می کنند:

۱. Region Proposal:

این اولین مرحله ای است که مدل مناطق مورد نظر (RoIs) را در تصویر پیشنهاد می کند. این را می توان توسط یک شبکه پیشنهادی منطقه (RPN) انجام داد، که به سرعت و کارآمدی هر مکان در تصویر را اسکن می کند تا ارزیابی کند که آیا پردازش بیشتر در یک منطقه خاص باید انجام شود یا خیر. RPN پیشنهادهای جعبه مرزی را خروجی می دهد، که هر کدام با دو امتیاز نشان دهنده احتمال وجود یا عدم وجود یک شی در هر مکان است. این مرحله با مشخص کردن اینکه چه ناحیه ای باید بیشتر مورد توجه قرار بگیرد، به منظور کاهش الزامات محاسباتی فرآیند کلی استنتاج مورد استفاده قرار می گیرد.

۲. Classification and Localization:

در مرحله دوم، مدل از RoIs برای طبقه بندی محتوا در کادر محدود (یا دور انداختن آن، با استفاده از «پس زمینه» به عنوان برچسب) استفاده می کند و مختصات جعبه مرزی را تنظیم می کند تا بهتر با شی مطابقت داشته باشد. این مرحله را می توان به عنوان یک CNN معمولی در نظر گرفت که بر روی مناطق پیشنهاد شده کار می کند

• Single-stage object detectors:

آشکارسازهای شی تک مرحله ای دسته ای از معماری های تشخیص اشیا هستند که تشخیص اشیا را به عنوان یک مشکل رگرسیون ساده در نظر می گیرند. آنها به طور مستقیم جعبه های محدود کننده شی را برای یک تصویر به صورت یک مرحله ای پیش بینی می کنند. به عبارت دیگر، هیچ وظیفه میانی وجود ندارد که باید برای تولید خروجی انجام شود. این منجر به یک معماری مدل ساده تر و سریع تر می شود، اگرچه گاهی اوقات ممکن است برای سازگاری با وظایف دلخواه (مانند پیش بینی ماسک) به اندازه کافی انعطاف پذیر باشد. این مدل ها معمولاً استنتاج سریع تری دارند (احتمالاً به قیمت عملکرد). آنها از مرحله پیشنهاد منطقه مدل های دو مرحله ای پرش می کنند و تشخیص را مستقیماً روی نمونه گیری متراکم از مکان ها اجرا می کنند.

• تفاوت ها:

آشکارسازهای شی تک مرحله ای تشخیص شی را به عنوان یک مشکل رگرسیون مستقیم در نظر می گیرند.

آنها به طور مستقیم کلاس و اندازه جعبه مرزی را برای هر سلول شبکه در یک تصویر پیش بینی می کنند.

آنها از یک مرحله میانی برای پیشنهاد مناطق نامزد استفاده نمی کنند.

آنها به طور کلی سریع تر و کارآمدتر هستند و آنها را برای برنامه های بلادرنگ مناسب می کنند.

آشکارسازهای شی دو مرحله ای:

آنها وظیفه تشخیص شی را به دو مرحله مجزا تقسیم می کنند: پیشنهاد منطقه و طبقه بندی.

در مرحله اول، آنها پیشنهادات منطقه ای (جعبه های محدود کننده نامزد) را تولید می کنند.

در مرحله دوم، طبقه بندی و رگرسیون جعبه مرزی را بر روی این مناطق پیشنهادی انجام می دهند.

آنها به طور کلی دقیق تر و قوی تر هستند، به خصوص برای تشخیص اجسام کوچک.

• کاربرد:

○ Single-stage object detectors:

آشکارسازهای شی تک مرحله ای مانند YOLO و SSD در زمینه های پزشکی کاربرد دارند. آنها را می توان برای تشخیص ضایعه در رادیولوژی، تشخیص سلول در پاتولوژی، محلی سازی ابزار پزشکی، تقسیم بندی اندام، کشف دارو، کمک آندوسکوپی، نظارت بر بیمار و کنترل کیفیت در تصویربرداری پزشکی استفاده کرد. این آشکارسازها پردازش و کارایی در زمان واقعی را ارائه می کنند و به کارهایی مانند تشخیص زودهنگام، کمک جراحی و توسعه دارو کمک می کنند. با این حال، بررسی دقیق انتخاب مدل، دقت و ملاحظات اخلاقی در کاربرد پزشکی این فناوریها ضروری است.

○ Two-Stage Object Detectors:

آشکارسازهای شی دو مرحله ای، مانند Faster R-CNN و Mask R-CNN، به دلیل دقت بالا و قابلیت تقسیم بندی دقیق، در کاربردهای پزشکی متنوعی مورد استفاده قرار می گیرند. آنها در وظایفی مانند تشخیص دقیق ضایعه در تصویربرداری پزشکی، تقسیم بندی اندام های ریز دانه، پوشش نمونه برای اطلاعات مکانی دقیق، تقسیم هسته سلولی در آسیب شناسی، و تشخیص ابزارهای جراحی در طول روش ها برتر هستند. این آشکارسازها همچنین در نظارت بر سیستم تحویل دارو، تجزیه و تحلیل تصویربرداری پزشکی سه بعدی، تشخیص ناهنجاری در رادیولوژی و کارهای مختلف تصویربرداری تحقیقات زیست پزشکی قابل استفاده هستند و به بهبود تشخیص، برنامه ریزی درمان و تحقیقات در زمینه پزشکی کمک می کنند.

.ii

معماری R-CNN (Region-based Convolutional Neural Network) شامل یک فرآیند دو مرحله ای برای تشخیص شی است. با تولید پیشنهادهای منطقه ای با استفاده از جستجوی انتخابی شروع می شود و به دنبال آن هر پیشنهاد از طریق یک شبکه عصبی کانولوشنال (CNN) از پیش آموزش دیده برای استخراج ویژگی منتقل می شود. Region of Interest (RoI) pooling layer پیشنهادات با اندازه متغیر را با یک نقشه ویژگی با اندازه ثابت تطبیق می دهد. لایه های fully connected طبقه بندی و رگرسیون جعبه مرزی را کنترل می کنند.

۱. R-CNN (Regions with Convolutional Neural Networks): برای مقابله با مشکل تشخیص

کارآمد اشیاء در تشخیص اشیاء ارائه شد. این الگوریتم جستجوی انتخابی را برای پیشنهاد حدود ۲۰۰۰ منطقه در هر تصویر استفاده می کند. این پیشنهادات سپس به یک مدل CNN در این مورد AlexNet استفاده می شود (منتقل می شوند که یک بردار ویژگی از هر پیشنهاد منطقه را خروجی می دهد. این بردار سپس به یک مدل SVM برای طبقه بندی شی و

یک بازگرداننده جعبه محدوده برای مکان یابی منتقل می شود. با این حال، R-CNN دارای برخی از نقاط ضعف است. آموزش شبکه زمان زیادی می برد زیرا هر تصویر باید ۲۰۰۰ پیشنهاد منطقه را طبقه بندی کند. همچنین نیاز به فضای دیسک زیادی برای ذخیره نقشه ویژگی پیشنهاد منطقه دارد.

۲. **Fast R-CNN**: با استفاده از یک لایه کانولوشنی مشترک برای پردازش تصویر کل و تمام مناطق مورد علاقه (ROIs)، به جای پردازش هر ROI به طور مستقل، نسبت به R-CNN بهبود می بخشد. این بدان معنی است که Fast R-CNN ابتدا CNN را پیاده سازی می کند و سپس پیشنهادات منطقه را تولید می کند. این نسبت به R-CNN، که ابتدا پیشنهادات منطقه را تولید می کند و سپس CNN را پیاده سازی می کند، بهبود قابل توجهی است.

۳. **Faster R-CNN**: یک مدل تشخیص اشیاء است که با استفاده از یک شبکه پیشنهاد منطقه (RPN) با مدل CNN، نسبت به Fast R-CNN بهبود می بخشد. RPN ویژگی های کانولوشنی تصویر کامل را با شبکه تشخیص به اشتراک می گذارد، که باعث می شود پیشنهادات منطقه تقریباً بدون هزینه باشد. این یک شبکه کاملاً کانولوشنی است که به طور همزمان مرزهای شی و امتیازات شی در هر موقعیت را پیش بینی می کند.

RPN به صورت end-to-end آموزش داده شده است تا پیشنهادات منطقه با کیفیت بالا ایجاد کند، که توسط Fast R-CNN برای تشخیص استفاده می شود. RPN و Fast R-CNN با به اشتراک گذاشتن ویژگی های کانولوشنی خود، به یک شبکه یکپارچه تبدیل می شوند. کامپوننت RPN به شبکه یکپارچه می گوید که کجا بنگرد. در کل، Faster R-CNN شامل دو ماژول است:

۱. ماژول اول یک شبکه کانولوشنی عمیق کامل است که مناطق را پیشنهاد می دهد.

۲. ماژول دوم تشخیص دهنده Fast R-CNN است که از مناطق پیشنهادی استفاده می کند.

این معماری به Faster R-CNN امکان می دهد تا از هر دو R-CNN و Fast R-CNN در زمان تشخیص سریع تر باشد، در حالی که دقت میانگین بیشتری (mAP) نسبت به هر دو مورد قبلی دارد.

در خلاصه، Faster R-CNN با استفاده از یک شبکه پیشنهاد منطقه (RPN) برای تولید ROIs، که بسیار سریع تر از الگوریتم جستجوی انتخابی استفاده شده در R-CNN و Fast R-CNN است، نسبت به Fast R-CNN بهبود می بخشد. شبکه از نقشه های ویژگی CNN برای پیش بینی مرزهای شی استفاده می کند.

سوال دوم:

i.

(۱) فرضیه اول (Saliency maps are truthful):

بر اساس این فرضیه، پیکسل‌های خاصی که به‌طور تصادفی در سراسر تصویر پراکنده شده‌اند، در نحوه تصمیم‌گیری شبکه نقش محوری دارند و منجر به ایجاد نویز در saliency maps می‌شوند. به این معنی که نویز موجود در نقشه‌های برجسته ممکن است در واقع برای فرآیند تصمیم‌گیری شبکه عصبی معنی دار و ضروری باشد.

(۲) فرضیه دوم (Gradients are discontinuous):

این فرضیه نشان می‌دهد که ناپیوستگی گرادیان‌ها در شبکه‌های عصبی عمیق (DNNs) به نویز در saliency maps کمک می‌کند. DNN ها اغلب از توابع خطی تکه ای (piecewise-linear functions) مانند فعال سازی ReLU و max-pooling استفاده می‌کنند که می‌تواند باعث جهش ناگهانی در امتیاز اهمیت نسبت به تغییرات بی‌نهایت کوچک در ورودی شود.

(۳) فرضیه سوم (Saturating activations):

با توجه به این فرضیه، یک ویژگی ممکن است اثر قوی در سطح global داشته باشد اما یک مشتق کوچک در سطح local. این اشباع می‌تواند منجر به ایجاد نویز در saliency maps شود.

دو نوع روش برای مقابله با این مشکل ارائه شده:

(۱) Gradient-based Methods

یکی از روش‌های بهبود saliency maps، افزودن نویز به تصویر ورودی و استفاده از گرادیان‌های تصویر ورودی perturbed با توجه به خروجی است. این روش که به نام SmoothGrad شناخته می‌شود، با ایجاد اختلال در ورودی و استفاده از گرادیان‌های تصویر ورودی perturbed، هدف آن حذف نویز است.

(۲) Backprop-based Methods

ایجاد تغییر در روش backprop برای بهبود عملکرد مدل در آموزش و حذف نویز

ii.

یکی از روش‌های ارزیابی attribute methods در selectivity، interpretability است که به توانایی یک سیستم یا مدل برای پاسخ دادن خاص به ورودی‌ها یا محرک‌های خاص در حالی که دیگران را نادیده می‌گیرد، اشاره دارد. در زمینه شبکه‌های عصبی عمیق و یادگیری ماشین، انتخاب‌پذیری اغلب به توانایی یک مدل برای شناسایی دقیق و متمایز و پاسخگویی به کلاس‌ها یا دسته‌های مختلف داده‌های ورودی اشاره دارد. در این روش از الگوریتم‌های متفاوتی استفاده می‌شود که یکی از آن‌ها الگوریتم pixel-flipping است. در زمینه selectivity و interpretability شبکه‌های عصبی عمیق، pixel-flipping تکنیکی است که برای ارزیابی faithfulness روش‌های توضیح برای پیش‌بینی‌های مدل استفاده می‌شود. این شامل perturbing یا flipping سیستماتیک پیکسل‌های منفرد در یک تصویر ورودی و مشاهده چگونگی تأثیر این perturbations بر امتیاز خروجی مدل است. با تجزیه و تحلیل تأثیر perturbations پیکسل بر پیش‌بینی‌های مدل، می‌توان گزینش‌پذیری مدل را از نظر توانایی آن در حفظ پیش‌بینی‌های دقیق و خاص در هنگام تغییر

پیکسل‌های خاص ارزیابی کرد. این تکنیک بینش‌هایی در مورد استحکام و حساسیت فرآیند تصمیم‌گیری مدل ارائه می‌کند و به ارزیابی گزینش پذیری و تفسیرپذیری آن کمک می‌کند.

نمودار، تکنیک pixel-flipping را برای سه روش توضیحی در نظر گرفته شده (Occlusion, Integrated Gradients, and Layer-wise Relevance Propagation) و دو مدل (VGG-16 و ResNet-50) اعمال می‌کند. شکل، نتایج آزمایش pixel-flipping را نشان می‌دهد، که تأثیر حذف ویژگی‌های مربوطه را بر امتیازات خروجی شبکه عصبی نشان می‌دهد. این نمودار شامل دو بخش است که هر کدام مربوط به مدل متفاوتی است (VGG-16 و ResNet-50). محور x نشان‌دهنده درصد پیکسل‌های perturb شده (مثلاً حذف ویژگی‌های مرتبط) است و محور y نشان‌دهنده میانگین امتیاز خروجی شبکه عصبی با توجه به این تغییرات در ورودی است.

سوال سوم:

(۱)

الگوریتم یادگیری ماشین توزیع شده یک سیستم چند گره‌ای است که مدل‌های آموزشی را با آموزش مستقل بر روی گره‌های مختلف می‌سازد. داشتن یک سیستم آموزشی توزیع شده، آموزش بر روی حجم عظیمی از داده‌ها را تسریع می‌کند. هنگام کار با داده‌های بزرگ، زمان آموزش به طور تصاعدی افزایش می‌یابد که باعث مقیاس پذیری و آموزش مجدد آنلاین می‌شود.

در مقایسه با یادگیری توزیع شده، الگوریتم‌های یادگیری فدرال اساساً متفاوت هستند و در درجه اول برای پرداختن به حریم خصوصی داده‌ها هستند. در یک خط لوله علوم داده سنتی، داده‌ها در یک سرور جمع‌آوری شده و برای ساخت و آموزش یک مدل متمرکز استفاده می‌شود. در واقع، یادگیری فدرال دارای یک مدل متمرکز با استفاده از آموزش مدل غیر متمرکز است. در سیستم‌های یادگیری فدرال، یک مجموعه پارامتر seed به گره‌های مستقل حاوی داده ارسال می‌شود و مدل‌ها بر روی گره‌های محلی با استفاده از داده‌های ذخیره شده در این گره‌های مربوطه آموزش داده می‌شوند. هنگامی که مدل به طور مستقل آموزش داده شد، هر یک از این وزن‌های مدل به روز شده به سرور مرکزی بازگردانده می‌شوند و در آنجا ترکیب می‌شوند تا یک مدل بسیار کارآمد ایجاد کنند. استفاده از آموزش داده‌های جهانی کارایی مدل را با یک عامل بزرگ بهبود می‌بخشد. این همچنین تضمین می‌کند که داده‌های هر گره به خط مشی‌های حفظ حریم خصوصی داده‌ها پایبند هستند و از هرگونه نشت / نقض داده محافظت می‌کند.

می‌توان گفت که یادگیری توزیع شده در مورد داشتن داده‌های متمرکز است اما آموزش مدل را به گره‌های مختلف توزیع می‌کند، در حالی که یادگیری فدرال شامل داشتن داده‌ها و آموزش غیر متمرکز و در واقع داشتن یک مدل مرکزی است.

در یادگیری توزیع شده، داده‌ها به صورت مرکزی ذخیره می‌شوند (به عنوان مثال، در یک مرکز داده). هدف اصلی فقط تمرین سریعتر است. ما نحوه توزیع داده‌ها در بین کارگران را کنترل می‌کنیم: معمولاً به صورت تصادفی با توزیع uniform توزیع می‌شود. در یادگیری فدرال، داده‌ها به طور طبیعی به صورت محلی توزیع و تولید می‌شوند. داده‌ها مستقل نیستند و به طور یکسان توزیع نمی‌شوند و imbalanced هستند.

در آموزش فدرال، مجموعه‌ای از کاربران هر کدام به داده‌های محلی خود دسترسی دارند و هدف آموزش مدلی بر روی تمام نقاط داده در شبکه بدون مبادله داده‌های محلی خود با سایر کاربران یا گره مرکزی به دلیل مسائل مربوط به حریم خصوصی یا محدودیت های ارتباطی است.

یک رویکرد مرسوم برای دستیابی به این هدف، به حداقل رساندن مجموع توابع محلی است که یک خروجی مشترک برای همه کاربران ایجاد می کند. با این حال، این طرح فقط یک مدل مشترک برای همه کاربران ایجاد می کند و بنابراین، مدل را برای هر کاربر تطبیق نمی دهد.

برای غلبه بر این مسئله، مقاله فرمول اصلاح شده ای از مسئله یادگیری فدرال را پیشنهاد می کند که شخصی سازی را در بر می گیرد. این فرمول اصلاح شده بر اساس فرمول مسئله Model-Agnostic Meta-Learning (MAML) است. هدف این فرمول جدید یافتن یک نقطه اولیه مشترک بین همه کاربران است که احتمالاً با انجام چند مرحله از روش مبتنی بر گرادیان پس از اینکه هر کاربر آن را با توجه به loss function خود به روزرسانی می کند، عملکرد خوبی دارد. به عبارت دیگر، مدل اشتراک گذاری شده به عنوان نقطه شروع استفاده می شود و هر کاربر آن را بر اساس داده ها و loss خود به روزرسانی می کند. این اجازه می دهد تا یک مدل شخصی تر برای هر کاربر به جای یک مدل واحد که ممکن است برای همه کاربران خوب عمل نکند.

این مقاله یک نوع شخصی سازی شده از مسئله یادگیری فدرال (FL) را پیشنهاد می کند، که هدف آن یافتن یک مدل اولیه سازی مناسب برای کاربران است که می تواند به سرعت با داده های محلی هر کاربر پس از مرحله آموزش سازگار شود. رویکرد پیشنهادی بر اساس فرمول مسئله Model-Agnostic Meta-Learning (MAML) است و Personalized FedAvg (Per-FedAvg) نامیده می شود.

Per-FedAvg برای یافتن راه حل بهینه برای مشکل FL شخصی سازی شده طراحی شده است که منجر به مدل شخصی تر برای هر کاربر در مقایسه با روش استاندارد FedAvg می شود. در این الگوریتم کسری از کاربران توسط سرور انتخاب میشوند، سپس مدل فعلی خود را به این کاربران ارسال میکند و هر کاربر منتخب مدل دریافتی را بر اساس تابع loss خود و با اجرای تعداد معینی از مراحل stochastic gradient descent به روزرسانی می کند. سپس مدل های به روز شده به سرور ارسال می شوند، که میانگین مدل های دریافتی از کاربران انتخاب شده را برای به روزرسانی مدل global محاسبه می کند.

همانطور که در مقاله بیان شده، f_i ، loss متناظر با کاربر i ام است و هدف در مسئله حل معادله $\min_{\omega \in \mathbb{R}^d} f(\omega) := \frac{1}{n} \sum_{i=1}^n f_i(\omega)$ است. به طور خاص با در نظر گرفتن ساختار supervised learning، f_i معادل امید ریاضی تابع loss بر روی توزیع داده مربوط به کاربر i ام است:

$$f_i(\omega) := \mathbb{E}_{(x,y) \sim p_i} [l_i(\omega; x, y)]$$

که در آن l_i خطای مدل ω در پیشبینی پاسخ و p_i توزیع تجربی کاربر i ام است. بنابراین میتوان آن را بصورت زیر بازنویسی کرد که در آن \mathcal{S}_i داده های کاربر i ام است:

$$f_i(\omega) := \sum_{(x,y) \sim p_i \in \mathcal{S}_i} p_i(x, y) l_i(\omega; x, y)$$

در روش MAML، ما به دنبال مقدار دهی اولیه ای هستیم که بعد از بروزرسانی با توجه به task جدید، عملکرد خوبی داشته باشد. اگر فرض کنیم هر کاربر با در نظر گرفتن یک نقطه شروع مقدار دهی و با توجه به loss خود بروزرسانی میشود، مسئله به شکل زیر تغییر میکند که در آن α مقدار stepsize است:

$$\min_{\omega \in \mathbb{R}^d} F(\omega) := \frac{1}{n} \sum_{i=1}^n f_i(\omega - \alpha \nabla f_i(\omega))$$

در روش Per-FedAvg نیز سعی میشود $F_i := f_i(\omega - \alpha \nabla f_i(\omega))$ که meta function کاربر نام است کمینه شود. به همین دلیل نیازمند محاسبه مشتق F_i است:

$$\nabla F_i(\omega) = (I - \alpha \nabla^2 f_i(\omega)) \nabla f_i(\omega - \alpha \nabla f_i(\omega))$$

در نتیجه مشتق دوم در معادله ظاهر میشود.