تمرین سری پنجم تحلیل تصاویر پزشکی — همفکری با سید امیرکسائی

امید محمد عزتی 402212269

سوال ① object detection

i) two-stage detector ها در یک مرحله با استفاده از یک شبکه region proposal های عکس را استخراج می‌کند و سپس در مرحله بعد ROI pooling، classification و مقادیر تصحیح شده bounding box را بدست می‌آورد.

امّا Single-stage detector ها دو مرحله بالا را با یک forward pass با استفاده از یک شبکه انجام می‌دهند.

- تفاوت اساسی‌شان در نحوه عملکردشان است که باعث می‌شود، سرعت Single-stage detector ها نسبت به two-stage detector ها بیشتر باشد.
امّا دقت two-stage بیشتر از Sing-stage ها است.

- موارد استفاده: Single-stage ها بدلیل سرعت بیشتر در کاربردهای real time پزشکی مثل عکسبرداری ultra sound برای تشخیص تومور یا سلامت کردن سلول‌ها استفاده می‌شود. two-stage ها چون دارای دقت بیشتر هستند ولی کندتر عمل می‌کنند در عکسبرداری‌های مثل CT-scan یا MRI استفاده می‌شود.
مثالی از تسک‌های مورد استفاده، تشخیص pneumonia یا lung nodule است.

ii) معماری R-CNN

R-CNN ابتدا به کمک proposal method، حدود 2000 تا region پیدا کرده و 2000 تا forward pass روی شبکه ConvNet به ازای هر region می‌زند و bounding box reg و classification را انجام می‌دهد. چون 2k جدا forward pass می‌زند، بسیار کند است.

شبکه fast R-CNN به جای 2K forward pass جداگانه، ابتدا یک ConvNet
می زند که feature map را بدست بیاورد، سپس ROI هر region proposal
در feature map بدست می آورد. سپس به ازای هر region با یک شبکه CNN
box offset و class را بدست می آورد. (چون در این روش ROI را به شبکه CNN می دهیم
و یک بار forward pass روی ConvNet داریم، سرعت بیشتر خواهد بود)

شبکه faster R-CNN از یک شبکه Region proposal برای تولید
region proposal ها به جای استفاده از proposal method استفاده میکند. سپس در
مرحله بعد box offset و class را تخمین می زند. برای این سریع تر از
دو روش قبلی عمل می کند.

سوال (2) :

ب) فرض اول : نویز موجود در Saliency map درست و صادقانه است.

ممکن است برای تشخیص کلاس یک object فقط نگاه کردن به pixel های همان object کافی نباشد و نیاز به توجه به پیکسل های کناری و اطراف object نیز باشد که در اینصورت، نویز موجود در Saliency map درست است.

فرض دوم : گرادیان ها غیر پیوسته هستند.

بدلیل اینکه شبکه DNN از تابع های خطی piecewise مثل ReLU activation یا max pooling استفاده می کند، گرادیان همواره پیوسته ندارد و این گرادیان غیر پیوسته باعث می شود نویز در map داشته باشیم.

jump های ناگهانی در تابع score با تغییر جزئی ورودی داریم و این نشان می دهد که شبکه مان به راحتی مستعد پذیرش است.

فرض سوم : تابع $f(x)$ ، Saturated شده است. (اشباع)

یعنی ممکن است یک feature بصورت عمومی تاثیر زیادی داشته باشد اما مشتق محلی ناچیزی داشته باشد. اگر مثلا $f(x)$ به مقدار ماکزیمم و نهایی خودش رسیده باشد آنگاه این مشتق توانایی ندارد که آن جنبه خاص را در هر پیکسل در Saliency map ، smooth تری تولید کند.

- دو راه حل برای بهبود Saliency map وجود دارد:

۱) روش براساس گرادیان : در این روش به داده ورودی نویز اضافه می کنیم تا تغییری در آن ایجاد کنیم سپس گرادیان خروجی برحسب داده تغییر یافته را حساب می کنیم $\nabla_{x^*} f(x^*)$.

در نهایت میانگین مجموعه ای از داده های perturbed شده که نزدیک ورودی اصلی

اولیه ما هستند، خروجی ای با نویز کمتر خواهد داشت. یعنی activation داده‌های perturbed شده مختلف را حساب می‌کنیم (که برخی نویز دارند و برخی ندارند) سپس با میانگین گرفتن از این مجموعه داده‌ها که نزدیک ورودی original هستند، Saliency map با نویز کمتر بدست می‌آید.

2) روش مبتنی بر back-prop

در این روش الگوریتم back propagation همان را برای محاسبه گرادیان تغییر می‌دهیم تا بهبود در نویز خروجی با گرفتن گرادیان‌های Smooth تر داشته باشیم.

ii) در روش‌های مبتنی بر attribution یک نوع ارزیابی بصورت Selectivity است در این ارزیابی سعی می‌کنیم روش بهتر را گزینش کنیم. یکی از الگوریتم‌ها برای گزینش الگوریتم Pixel-flipping است. این الگوریتم به این صورت است که هر مرحله پیکسل‌هایی که توسط روش explanation، اهمیت می‌شوند را حذف می‌کند. (حذف با الگوریتم ساده تقاشی صورت می‌گیرد)

انتظار می‌رود با حذف این پیکسل‌ها در هر مرحله، Score پیش‌بینی شده و شواهد کلاس مورد نظر در هر مرحله کاهش یابد. یعنی عملکرد روش explanation ما بهتر از یک روش explanation تصادفی باشد. (هر مرحله پیکسل‌های مرتبط‌تری را برای استدلال پیدا کند و با حذف آن‌ها Class evidence مربوطه با سرعت زیادی کاهش بیابد)

در نمودار آمده شده، محور افقی نشان دهنده درصد پیکسل‌های perturbed شده (حذف شده) و محور عمودی نشان دهنده میانگین Score خروجی مدل است که با افزایش درصد پیکسل‌های حذف شده باید کاهش زیادی داشته باشد.

این نمودار عملکرد مدل VGG-16 و ResNet را با هم مقایسه می‌کند که در مدل VGG-16 روش‌های explanation را روی بطور کلی عملکرد بهتری دارند.

سوال ③

ا) یادگیری distributed یک سیستم ماشین چندین node ای است که عملکرد و دقت و scalability بالایی ارائه می‌کند. این نوع یادگیری برای مدیریت و scale شدن داده های بسیار عظیم بوجود آمده اند. چون هندل کردن داده ها با مقیاس بزرگ یک چالش اساسی الگوریتم های ماشین لرنینگ است و یادگیری distributed الگوریتم scalable و efficient برای این چالش ارائه می‌کند.

یادگیری federated یک نوع روش یادگیری ماشین distributed است که دارای چندین user می باشد که بصورت مشترک و همکارانه یک مدل را آموزش می دهند. در این حالت داده های خام بین user ها توزیع شده است بدون اینکه به یک data center انتقال یافته باشد. مدل اولیه به همه گره ها ارسال می‌شود و گره ها، مدل را روی داده های خود آموزش می دهند و یک نسخه محلی از مدل دارند. هدف اصلی federated حفظ حریم خصوصی و داده های محرمانه هر user است.

در واقع در یادگیری distributed، داده های متمرکز داریم امّا توزیع آموزش مدل را به node های مختلف می سپاریم. ولی در federated هم داده ها و هم آموزش مدل بصورت غیر متمرکز است و هدف یادگیری داشتن یک مدل مرکزی است.

نکات تفاوت کلیدی:

- FL امکان ارتباط مستقیم با داده های خام را نمی دهد distributed محدودیتی ندارد.
- FL از منابع محاسباتی توزیع شده چندین سازمان استفاده می‌کند. امّا DL از یک سرور یا یک cluster از نود ها که متعلق به یک سازمان واحد هستند استفاده می‌کند.
- FL از رمزنگاری و تکنیک های privacy برای تضمین محرمانه بودن و امنیت داده ها استفاده می‌کند امّا در DL چنین چیزی زیاد مورد توجه نیست.

سوال ③ :

ii) در این مقاله برای personalized کردن مدل‌ها از فرمول روش MAML استفاده می‌شود. هدف اصلی این است که یک نقطه اولیه درست پیدا شود برای همه user ها، که بتواند انجام update هر user روی loss function خودش به نتیجه خوبی برسیم.

روش MAML به این صورت است که مجموعه‌ای از task های مختلف در اختیار دارد و هدف این است که مدل (یا نقطه‌ای) پیدا شود که روی همه task ها عملکرد خوبی داشته باشد. در واقع نقطه‌ای که اگر یک یا چند Step ، گرادیان کاهشی روی آن بزنیم بتوانیم آن را Customized شبک جدید خودمان بکنیم.

$$\min F(w) := \frac{1}{n} \sum_{i=1}^{n} f_i \left(w - \alpha \nabla f_i(w)\right)$$

مدل ارائه‌شده در مقاله Per-FedAvg نام دارد و از الگوریتم FedAvg الهام گرفته است. هربار سرور تعدادی از user ها را انتخاب کرده و مدل فعلی (Meta model) را به آنها می‌فرستد و هر user با توجه به loss خودش با اجرای 1 یا چند Stochastic gradient descent ، مدل را update می‌کند.
سرور، مدل‌های update شده را از کاربران انتخابی می‌گیرد و میانگین آنها را محاسبه می‌کند.

meta function مرتبط با user i به صورت زیر تعریف می‌شود:

$$F_i(w) := f_i \left(w - \alpha \nabla f_i(w)\right)$$

مدل نهایی میانگین meta function های $F_1, F_2, \ldots, F_n$ است.

اگر مراحل حل مسئله را با توجه به الگوریتم FedAvg پیش برویم باید گرادیان تابع های local را محاسبه کنیم یعنی $\nabla F_i(w)$:

$$\nabla F_i(w) = \left(I - \alpha \nabla^2 f_i(w)\right) \nabla f_i\left(w - \alpha \nabla f_i(w)\right)$$

همانطور که می بینیم در محاسبه گرادیان تابع های محلی به مشتق مرتبه دوم (Hessian) برخورد می کنیم.