به نام خدا

سینا علی نژاد

۹۹۵۲۱۴۶۹

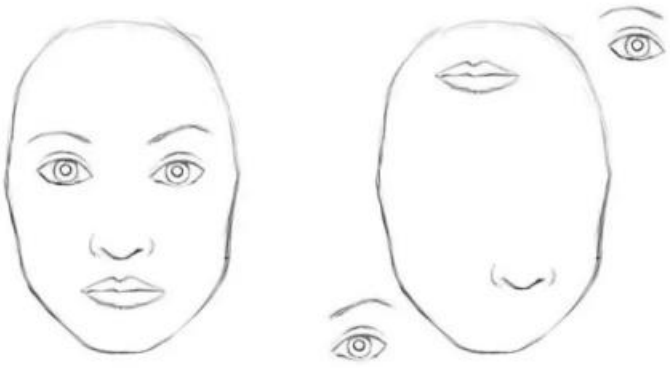
تمرین سری ششم

درس مبانی یادگیری عمیق

سوال ۱-

(آ) مسئله گربه بودن یا نبودن

برای عکس بالا، یک شبکه CNN عکس سمت چپ را به عنوان گربه خواهد شناخت اما برای عکس سمت راست نمیتواند تشخیص دهد و دلیل آن اهمیت دادن به بافت تصویر است که در اینجا یک گربه در بافت پوست فیل است.



(ب) مسئله انسان بودن یا نبودن

برای عکس بالا، یک شبکه CNN هر دو را به عنوان چهره انسان تشخیص میدهد در حالیکه عکس سمت راست نباید چنین تشخیصی برای آن داده شود. دلیل اینکه CNN آن را به عنوان چهره تشخیص میدهد، این است که فقط وجود برخی عناصر را بررسی میکند و نحوه قرارگیری آنها را در نظر نمیگیرد. این بدلیل لایه GAP در آخر کار میتواند باشد که تنها وجود یک ویژگی در کل تصویر را بررسی میکند. برای مثال وجود چشم در تصویر.

در جهت مقابل، با استفاده از مکانیزم توجه، در تصویر (آ) میتوان عکس سمت راست را به عنوان گربه تشخیص داد. زیرا در مکانیزم توجه، به جای توجه به بافت، به نحوه قرارگیری عناصر در کنار یکدیگر توجه میشود.

همچنین در تصویر (ب) عکس سمت راست را به عنوان چهره انسان تشخیص نخواهد داد زیرا قرارگیری عناصر در کنار یکدیگر نیز اهمیت دارد.

۲-

الف)

TP: مواردی که مدل به عنوان نمونه مثبت پیش بینی کرده و به درستی پیشبینی کرده.

TN: مواردی که مدل به عنوان نمونه منفی پیش بینی کرده و به درستی پیشبینی کرده.

FP: مواردی که مدل به عنوان نمونه مثبت پیش بینی کرده و به اشتباه پیشبینی کرده.

FN: مواردی که مدل به عنوان نمونه منفی پیش بینی کرده و به اشتباه پیشبینی کرده.

ب)

در این مسئله، با توجه به اهمیت گناهکار شناخته نشدن مردم عادی، باید از بین مواردی که مدل ما به عنوان نمونه مثبت تشخیص داده، اکثرشان واقعا مثبت باشند. بنابراین معیار precision برای ما مهمتر خواهد بود. از طرفی بخاطر اهمیت امنیت مردم، باید بتوانیم اکثر مجرمان را شناسایی کنیم که در نتیجه معیار recall مهمتر خواهد بود. برای حل این مشکل از ترکیب این دو استفاده میکنیم. برای مثال میتوان از F-Score استفاده کردم.

در یادگیری عمیق، دقت و یادآوری دو معیار مهمی هستند که برای ارزیابی عملکرد یک مدل طبقه‌بندی استفاده می‌شوند.

دقت کسری از مثبت واقعی (نمونه های مثبت پیش بینی شده صحیح) در بین تمام نمونه های مثبت پیش بینی شده است.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

یادآوری کسری از مثبت های واقعی در بین تمام نمونه های مثبت واقعی است. به صورت زیر محاسبه می شود:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

به عبارت دیگر، دقت اندازه‌گیری می‌کند که چه تعداد از نمونه‌های مثبت پیش‌بینی‌شده واقعاً مثبت هستند، در حالی که یادآوری اندازه‌گیری می‌کند که چه تعداد از نمونه‌های مثبت واقعی به درستی مثبت پیش‌بینی شده‌اند.

به عنوان مثال، یک مشکل طبقه بندی باینری را در نظر بگیرید که در آن می خواهیم پیش بینی کنیم که آیا ایمیل اسپم است یا خیر. دقت بالا به این معنی است که اکثر ایمیل های پیش بینی شده به عنوان هرزنامه در واقع هرزنامه هستند، در حالی که فراخوانی بالا به این معنی است که بیشتر ایمیل های هرزنامه واقعی به درستی به عنوان هرزنامه پیش بینی می شوند.

امتیاز F1 اندازه‌گیری دقت مدل طبقه‌بندی باینری است که هم دقت و هم یادآوری را در یک متریک واحد ترکیب می‌کند.

دقت کسری از مثبت های واقعی در بین تمام نمونه های مثبت پیش بینی شده است، در حالی که یادآوری کسری از مثبت های واقعی در بین تمام نمونه های مثبت واقعی است.

امتیاز F1 میانگین هارمونیک دقت و یادآوری است و از ۰ تا ۱ متغیر است و مقادیر بالاتر نشان دهنده عملکرد بهتر است. فرمول محاسبه امتیاز F1 به این صورت است:

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

امتیاز F1 زمانی که توزیع کلاس نامتعادل باشد، یک معیار مفید است، به این معنی که یک کلاس نمونه‌های بیشتری نسبت به ۲ کلاس دیگر دارد. در چنین مواردی، دقت به تنهایی می‌تواند گمراه‌کننده باشد، زیرا مدلی که همیشه پیش‌بینی می‌کند کلاس اکثریت دقت بالایی خواهد داشت. اما دقت و یادآوری کم. امتیاز F1 هم دقت و هم یادآوری را در نظر می گیرد و آن را به معیار قابل اعتمادتری برای عملکرد تبدیل می کند.

-۳

الف) در یادگیری خود نظارتی، تخمین چرخش تکنیکی است که برای یادگیری نمایش های مفید داده ها بدون نیاز به حاشیه نویسی انسانی استفاده می شود. ایده این است که یک مدل را آموزش دهیم تا زاویه چرخش یک تصویر را با توجه به جهت اصلی آن پیش بینی کند. این کار به خودی خود نظارت می شود زیرا برچسب ها به طور خودکار از خود داده ها و بدون هیچ گونه دخالت انسانی تولید می شوند.

شهود پشت استفاده از تخمین چرخش برای طبقه‌بندی این است که مدل را تشویق می‌کند تا ویژگی‌هایی را بیاموزد که نسبت به چرخش‌ها تغییر نمی‌کنند. به عبارت دیگر، اگر مدل بتواند زاویه چرخش یک تصویر را به درستی پیش‌بینی کند، باید ویژگی‌های مفیدی را یاد گرفته باشد که تحت تأثیر چرخش قرار نمی‌گیرند. این ویژگی‌ها سپس می‌توانند برای کارهای دیگر، مانند طبقه‌بندی، که در آن تغییر ناپذیری چرخش مطلوب است، استفاده شوند.

به عنوان مثال، مجموعه داده ای از تصاویر ارقام دست نویس را در نظر بگیرید. با آموزش مدلی برای پیش بینی زاویه چرخش این تصاویر، می توانیم ویژگی هایی را یاد بگیریم که نسبت به چرخش ها تغییر نمی کنند. سپس می توان از این ویژگی ها برای طبقه بندی ارقام استفاده کرد، حتی اگر آنها چرخیده یا کج شده باشند.

ب) در پردازش زبان طبیعی، بردارهای تک داغ تکنیکی است که برای نمایش کلمات به عنوان بردار استفاده می شود. در این تکنیک، هر کلمه به عنوان بردار ۰ و ۱ نشان داده می شود که ابعاد بردار برابر با اندازه واژگان است و موقعیت ۱ مطابق با شاخص کلمه در واژگان است. به عنوان مثال، واژگانی از ۴ کلمه را در نظر بگیرید: "گربه"، "سگ"، "ماهی" و "پرنده". بردارهای یک داغ برای این کلمات خواهد بود:

```
cat:  [1, 0, 0, 0]
dog:  [0, 1, 0, 0]
fish: [0, 0, 1, 0]
bird: [0, 0, 0, 1]
```

مشکل استفاده از بردارهای تک داغ برای نمایش کلمات این است که ابعاد بسیار بالایی دارند و بیشتر ابعاد آنها صفر است. این باعث می شود آنها بسیار پراکنده باشند، که می تواند منجر به ناکارآمدی محاسباتی و بیش از

حد برازش شود. علاوه بر این، بردارهای یک داغ هیچ اطلاعات معنایی در مورد کلمات را نمی گیرند و همه کلمات را به یک اندازه از یکدیگر دور می کنند، که در بیشتر موارد درست نیست.

برای غلبه بر این مشکلات، از جاسازی کلمات استفاده می شود که کلمات را به عنوان بردارهای متراکم اعداد واقعی نشان می دهد، که در آن هر بعد از بردار جنبه متفاوتی از معنای کلمه را به تصویر می کشد. جاسازی های کلمه از مقادیر زیادی داده متنی با استفاده از تکنیک هایی مانند word2vec و GloVe آموخته می شوند و نشان داده شده است که در بسیاری از کارهای پردازش زبان طبیعی بسیار موثر هستند.

ج) Word2vec تکنیکی است که برای یادگیری جاسازی کلمات استفاده می‌شود، که نمایش‌های برداری متراکمی از کلمات هستند که معنای معنایی آنها را به تصویر می‌کشند. Word2vec یک الگوریتم خود نظارت است زیرا بدون نیاز به حاشیه نویسی انسانی از داده ها یاد می گیرد.

در word2vec، مدل برای پیش‌بینی متن یک کلمه با توجه به کلمات همسایه آن (یا برعکس) آموزش داده می‌شود. این کار به خودی خود نظارت می شود زیرا برچسب ها به طور خودکار از خود داده ها و بدون هیچ گونه دخالت انسانی تولید می شوند. شهود پشت این رویکرد این است که کلماتی که در زمینه های مشابه ظاهر می شوند احتمالاً معانی مشابهی دارند.

وزنهای کلمات آموخته شده را می‌توان برای انواع وظایف پردازش زبان طبیعی، مانند تجزیه و تحلیل احساسات، ترجمه ماشینی و طبقه‌بندی متن استفاده کرد. با استفاده از جاسازی‌های کلمه، می‌توانیم کلمات را به‌عنوان بردارهای متراکم اعداد واقعی نشان دهیم که از نظر محاسباتی کارآمدتر و از نظر معنایی معنادارتر از بردارهای تک داغ هستند.

نکته: البته در نسخه اصلاحی word2vec به جای ورودی context و پیشبینی target که هزینه محاسباتی سنگین به دلیل لایه softmax آخر دارد، به گونه دیگری عمل میکنیم. بدین صورت که context و target را به عنوان ورودی میدهیم و خروجی، صفر و یک است. یک یعنی این دو کلمه نمونه مشابه هستند، صفر یعنی نمونه منفی یا غیرمشابه هستند.

سوال ۴-

الف) یادگیری تقویتی نوعی از یادگیری ماشینی است که شامل آموزش یک عامل برای تصمیم گیری در یک محیط با به حداکثر رساندن سیگنال پاداش است. در زمینه جستجوی ساختار شبکه، یادگیری تقویتی می‌تواند برای تولید خودکار معماری‌های شبکه با کارایی بالا برای یک کار یادگیری خاص استفاده شود.

ایده آموزش یک عامل برای انتخاب متوالی لایه‌های شبکه با استفاده از Q-Learning با یک استراتژی کاوش ϵ-greedy و تکرار تجربه انتخاب کند. عامل فضای بزرگ اما محدودی از معماری‌های ممکن را کاوش می‌کند و به طور مکرر طرح‌هایی را با عملکرد بهبود یافته در کار یادگیری کشف می‌کند. . سیگنال پاداش را می توان به روش های مختلفی تعریف کرد، مانند دقت مدل در مجموعه اعتبارسنجی یا سرعت همگرایی در طول آموزش.

با استفاده از یادگیری تقویتی، می‌توانیم فرآیند جستجوی ساختار شبکه را خودکار کنیم و معماری‌هایی را کشف کنیم که برای یک کار یادگیری خاص بهینه شده‌اند. این می تواند در مقایسه با طراحی دستی در زمان و تلاش زیادی صرفه جویی کند، به خصوص برای کارهای پیچیده ای مانند تشخیص تصویر یا پردازش زبان طبیعی.

ب) در اینجا بحثی در مورد امکان استفاده از رویکرد جستجوی یادگیری تقویتی (RL) برای بهینه سازی هایپرپارامترها در تشخیص اشیا، به ویژه تمرکز بر ویژگی های تصویر ورودی و تعداد لایه ها وجود دارد:

۱. درک نقش فراپارامترها:

تصویر ورودی: وضوح، تکنیک های پیش پردازش، و روش های تقویت به طور قابل توجهی بر عملکرد مدل تاثیر می گذارد.

تعداد لایه‌ها: عمق یک شبکه عصبی بر توانایی آن در استخراج ویژگی‌های پیچیده تأثیر می‌گذارد، اما لایه‌های بیش از حد می‌تواند منجر به تطبیق بیش از حد یا ناپدید شدن گرادیان شود.

۲. چالش های تنظیم هایپرپارامتر سنتی:

جستجوی دستی: پر زحمت و اغلب نابهینه.

جستجوی شبکه ای و جستجوی تصادفی: می تواند در فضاهای جستجوی بزرگ زمان بر و ناکارآمد باشد.

۳. جستجوی یادگیری تقویتی:

ایده کلیدی: تنظیم هایپرپارامتر را به عنوان یک مسئله RL فرموله کنید که در آن یک عامل با محیط تعامل دارد (فرایند آموزش مدل) تا بهترین پیکربندی هایپرپارامتر را یاد بگیرد.

عامل: می آموزد که مقادیر فراپارامتر را از طریق آزمون و خطا انتخاب کند، که توسط پاداش بر اساس عملکرد مدل هدایت می شود.

۴. آدرس دهی تصویر ورودی و تعداد لایه ها:

تصویر ورودی: عامل می تواند وضوح تصویر مختلف، تکنیک های پیش پردازش و استراتژی های تقویت را برای به حداکثر رساندن دقت یا به حداقل رساندن مصرف منابع بررسی کند.

تعداد لایه‌ها: عامل می‌تواند با عمق‌های مختلف شبکه آزمایش کند، و معاوضه بین دقت و پیچیدگی را مشاهده کند.

۵. مزایای جستجوی یادگیری تقویتی:

کاوش خودکار: به طور موثر فضاهای فراپارامتری بزرگ و پیچیده را هدایت می کند.

سازگاری: از تجربه یاد می گیرد و استراتژی ها را تنظیم می کند، به طور بالقوه سناریوهای نادیده قبلی را مدیریت می کند.

پتانسیل برای بهبود مستمر: می توان از آن برای اصلاح فراپارامترها با ظهور داده ها یا معماری های مدل جدید استفاده کرد.

۶. ملاحظات و چالش ها:

هزینه محاسباتی: جستجوی RL خود می تواند از نظر محاسباتی گران باشد، به خصوص برای معماری های مدل پیچیده.

طراحی عملکرد پاداش: تعریف یک تابع پاداش موثر که عملکرد مدل و محدودیت‌های منابع را به تصویر می‌کشد، می‌تواند چالش برانگیز باشد.

مقیاس‌پذیری: مقیاس‌بندی جستجوی RL به فضاهای فراپارامتری با ابعاد بالا با وابستگی‌های زیاد، یک حوزه تحقیقاتی فعال است.

۷. نتیجه گیری:

جستجوی یادگیری تقویتی یک رویکرد امیدوارکننده برای خودکارسازی و بهبود بالقوه تنظیم هایپرپارامتر برای تشخیص اشیا ارائه می دهد. توانایی آن در کاوش تطبیقی و یادگیری از تجربه، آن را برای مدیریت پیچیدگی‌های بهینه‌سازی فراپارامتر، از جمله چالش‌های بهینه‌سازی تصویر ورودی و طراحی معماری شبکه، مناسب می‌سازد. در حالی که طراحی تابع پاداش و هزینه محاسباتی همچنان چالش‌هایی است، تحقیقات در حال انجام به این مسائل می‌پردازد، و جستجوی RL را به ابزاری مناسب برای بهینه‌سازی فراپارامتر در تشخیص اشیا و سایر حوزه‌های یادگیری ماشین تبدیل می‌کند.

سوال ۵-

شبکه مولد سعی میکند عبارت زیر را مینیمم کند و شبکه ممیز سعی بر بیشینه شدن آن دارد. نکته ای که اینجا هست، این است که همزمان که مدل مولد عکسهای بهتری تولید میکند، مدل ممیز هم با عکسهای مصنوعی بهتر آموزش میبیند و قوی تر میشود. در واقع مانند یک بازی دونفره است که هر دو بازیکن هرچه از بازی پیش میرود، قوی و قویتر میشوند و باعث تقویت یکدیگر میشوند. بنابراین، پس از ۱۰۰ ایپاک همچنان ممکن است مقدار ضرر مدل مولد با ایپاک های اولیه تفاوتی نکند و معمولا باید همینطور باشد.

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

جواب استنفورد برای این سوال:

Solution: You should not necessarily expect them to be the same since the losses are with respect to different quality models over time. That is, the loss of the generator at epochs 1 and 100 are with respect to a discriminator which might have significantly improved, and the same follows for the loss of the discriminator.

که به همان موارد بالا که گفتم اشاره دارد.