

سوال اول

منبع:

<https://becominghuman.ai/transformers-in-vision-e2e87b739feb>

<chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/2105.07197v2.pdf>

الف

گره بودن یا نبودن تصویر:

CNN:

این تصاویر هر دو گره هستند اما چون CNN ها لبه ها و یافت تصویر را هم در لایه های ابتدایی یاد میگیرند، تصویر سیاه و سفید بافت پوست گره را ندارد به همین دلیل جز گره ها دسته بندی نمیشود چرا که یان مدل ها برخلاف انسانها که بیشتر شکل کلی را برای دسته بندی موثر میدانیم بافت تصویر برایش مهم است.

مدل توجه:

مدل های توجه در اینجا میتوانند هر دو را گره تشخیص بدهند چون به شکل قرار گیری اجزای تصویر کنار هم بیشتر توجه میکنند و به بافت که تقریباً در تمام قطعات گره ای یک تصویر یکسان است کمتر توجه میکنند.

ب

انسان بودن یا نبودن تصویر:

CNN:

برای یک CNN، هر دوی این تصاویر تقریباً یکسان هستند. CNN موقعیت نسبی ویژگی های مختلف را رمزگذاری نمی کند. فیلترهای بزرگ برای رمزگذاری ترکیبی از این ویژگی ها مورد نیاز است. به عنوان مثال: چشم چپ و راست، بینی و دهان را جداگانه به خوبی یادگرفته است و همگی در تصویر سمت راستی وجود دارند ولی در جای مناسب قرار ندارند. چون فیلتر مربوط به هر کدام از این اعضا فعال شده و وجودشان در تصویر تایید میشود به عنوان انسان شناخته میشود.

برای در نظر گرفتن وابستگی های دوربرد در یک تصویر، فیلدهای دریافتی بزرگ مورد نیاز است. افزایش اندازه هسته های کانولوشن می تواند ظرفیت بازنمایی شبکه را افزایش دهد اما با انجام این کار کارایی محاسباتی و آماری به دست آمده با استفاده از ساختار کانولوشن محلی را نیز از دست می دهد.

مدل توجه:

مدل های توجه به خود، نوعی مکانیسم توجه، همراه با CNN به مدل سازی وابستگی های دوربرد بدون به خطر انداختن کارایی محاسباتی و آماری کمک می کند. مدل خودتوجهی مکمل کانولوشن ها است و به مدل سازی وابستگی های طولانی و چند سطحی در مناطق تصویر کمک می کند.

سوال دوم:

الف

F N

موقعیتی که مدل به اشتباه یک منفی را تشخیص داده است. یعنی مدل اشتباهاً تشخیص داده که ویژگی مورد نظر وجود ندارد، در حالی که در واقعیت وجود داشته است.

T P

موقعیتی که مدل به درستی یک مثبت را تشخیص داده است. به عبارت دیگر، مدل درستاً تشخیص داده که یک ویژگی مورد نظر وجود دارد.

TN

موقعیتی که مدل به درستی یک منفی را تشخیص داده است. یعنی مدل به درستی تشخیص داده که ویژگی مورد نظر وجود ندارد.

F P

موقعیتی که مدل به اشتباه یک مثبت را تشخیص داده است. به عبارت دیگر، مدل اشتباهاً گزاره ای اعلام کرده است که در واقعیت وجود ندارد.

ب

اشتباه تشخیص ندادن افراد بی گناه معادل کم بودن FP: بهتر است از روش امتناع از تصمیم گیری استفاده کنیم. یعنی پس از اینکه مدل با یک دقتی تعدادی از افراد را به عنوان هکر معرفی کرد یک اپراتور انسانی آن افراد را چک کند و تعدادی که مشخص است مدل اشتباه کرده را خودش چک کند و از لیست حذف کند. لازم است تخمین زده شود که چه قدر از تصمیم مطمئنیم. این روش میزان کار انسانی را کاهش میدهد.

سوال سوم:

الف

در بعضی موارد، اطلاعات جهت می‌توانند تفاوت‌هایی در داده‌های ورودی ایجاد کنند. با تخمین چرخش، می‌توان این اختلافات را کاهش داده و ورودی‌ها را به یک استاندارد چرخش مشخص تبدیل کرد. تخمین چرخش می‌تواند کمک کند تا مدل برای داده‌هایی با چرخش‌های مختلف یاد بگیرد و این باعث افزایش تعمیم‌پذیری مدل می‌شود. به این ترتیب، مدل توانایی بهتری در تشخیص الگوها و اجسام در داده‌های تست را دارا می‌شود. در بعضی حالات، تخمین چرخش می‌تواند از تاثیر تغییرات چرخشی در داده‌ها کاسته و مدل را مقاوم‌تر در برابر این تغییرات کند. اگر در داده‌های ورودی جهت یا چرخش اطلاعات مفیدی داشته باشد و این اطلاعات برای تمیزدهی و تشخیص اجزای مختلف تصویر یا الگوهای خاص مفید باشد، تخمین چرخش می‌تواند بهبود دقت طبقه‌بندی کمک کند.

ب

On-hot vector

برای امبدینگ کلمات و اشیاء استفاده میشود. ابتدا باید تعداد چیزهای یونیک که میخواهیم کد کنیم را داشته باشیم. سپس به هر کدام یک عدد اختصاص میدهیم. به ازای هر شی یک آرایه میسازیم. در آرایه مربوط به هر شی فقط درایه شماره اش یک میشود و مابقی اعضا صفر هستند. مثلا وقتی یک دیکشنری سه کلمه ای از کلمات dog و cat و duck داریم و به ترتیب عدد 0 و 1 و 2 را به هر یک نسبت میدهیم امبدینگ هر یک میشود:

$$\text{Dog}=[1, 0, 0]$$

$$\text{Cat}=[0, 1, 0]$$

$$\text{Duck}=[0, 0, 1]$$

مشکلات استفاده:

تنک بودن یا sparse که باعث میشود در مثال های واقعی فضای زیادی برای ذخیره سازی و پردازش نیاز داشته باشیم. ابعاد بالا: در مثال های واقعی تعداد کلمات خیلی بیشتر است و کار کردن با آرایه های بزرگ و جست و جو را سخت است. هارد کد: قابل آموزش از طریق مدل های شبکه عصبی نیست. روابط معنایی بین کلمات را در نظر نمیگیرد.

ج

Word2Vec یک مدل معروف برای تولید بردارهای واژگانی (word embeddings) است که از الگوریتم های self-supervised برای یادگیری نمایش واژگان استفاده می‌کند. الگوریتم‌های self-supervised یک دسته از الگوریتم‌های یادگیری نظارت ذاتی هستند که بدون نیاز به برچسب‌های دقیق و تفسیری از داده‌های آموزشی عمل می‌کنند Word2Vec. از دو رویکرد معروف در self-supervised learning یعنی Skip-gram و Continuous Bag of Words (CBOW) بهره می‌برد.

Skip-gram:

در رویکرد Skip-gram، مدل سعی می‌کند با دادن یک واژه هدف، واژه‌های همسایه یا مجاور را پیش‌بینی کند. به این ترتیب، مدل سعی می‌کند اطلاعات معنایی و مفهومی اطراف هر واژه را در بردارهای واژگانی ذخیره کند.

Continuous Bag of Words (CBOW):

در رویکرد CBOW، مدل با داشتن واژه‌های مجاور یا همسایه، سعی می‌کند واژه مرکزی را پیش‌بینی کند. این رویکرد نیز به مدل کمک می‌کند تا بردارهای واژگانی را به گونه‌ای بهینه بسازد که در تبیین مفهوم واژه‌ها مؤثر باشد.

Word2Vec با استفاده از این دو روش، بردارهای واژگانی را به گونه‌ای آموزش می‌دهد که ویژگی‌های معنایی و مفهومی واژه‌ها در فضای برداری بازنمایی می‌شوند. این بردارها دارای خاصیت‌هایی مانند معنای مشابه و روابط معنایی هستند. بنابراین، اگر دو واژه در یک متن معنایی مشابه داشته باشند، بردارهای مربوط به آن دو واژه نیز به گونه‌ای مشابه در فضای برداری قرار می‌گیرند.

تا حدی که مدل Word2Vec به صورت self-supervised آموزش می‌بیند، به معنای آن است که از تنها داده‌های خود برای یادگیری بدون نیاز به برچسب استفاده می‌کند. این الگوریتم نشان می‌دهد که یادگیری نمایش واژگانی به صورت self-supervised می‌تواند نمایش‌های با ارزشی از واژگان تولید کرده و در وظایف مختلفی از جمله معنایی گرفته تا ماشین ترجمه و غیره مورد استفاده قرار گیرد.

سوال چهارم:

الف

1. تعریف مساله: در این مرحله، مساله مورد نظر برای یادگیری تقویتی مشخص می‌شود که شامل تعریف وضعیت‌ها، عمل‌ها، پاداش‌ها و ساختار کلی محیط است. علاوه بر این، هدف کلی یادگیری، مشخص کردن محیط، مسیر و راه‌حل‌های بهینه در فرآیند یادگیری بسیار حایز اهمیت است..

2. تعریف وضعیت‌ها و عمل‌ها

وضعیت‌ها ممکن است شامل اطلاعات مشاهده شده از محیط یا وضعیت داخلی عامل باشند. عمل‌ها نیز مجموعه‌ای از اقدامات قابل انجام توسط عامل هستند. تعریف وضعیت‌ها و عمل‌ها در یادگیری تقویتی:

تعریف وضعیت‌ها: وضعیت‌ها نشان‌دهنده شرایط محیطی هستند که عامل در آن قرار می‌گیرد. وضعیت‌ها می‌توانند اطلاعات مشاهده شده از محیط و یا وضعیت داخلی عامل را شامل شوند. به طور کلی، وضعیت‌ها می‌توانند به صورت توصیفی (مشاهده شده) یا به صورت خلاصه (توصیفی) تعریف شوند. برای مثال، در یک بازی شطرنج، وضعیت می‌تواند شامل موقعیت مهره‌ها در صفحه شطرنج، نوبت بازیکن و سایر جزئیات بازی باشد.

تعریف عمل‌ها: عمل‌ها نشان‌دهنده اقدامات قابل انجام توسط عامل در هر وضعیت هستند. معمولاً عمل‌ها در قالب یک فضای عمل تعریف می‌شوند. فضای عمل می‌تواند متناهی یا نامتناهی باشد و شامل تمام عمل‌های مجاز در محیط است. برای مثال، در یک بازی شطرنج، عمل‌ها می‌توانند شامل حرکت مهره‌ها، قرار دادن پیش‌بینی‌ها یا انجام تغییرات دیگر در صفحه بازی باشند.

3. تعریف پاداش

در این مرحله، نحوه ارزیابی عمل‌ها توسط محیط و تعریف پاداش‌ها مشخص می‌شود. پاداش‌ها معمولاً به صورت عددی تعریف می‌شوند و نشان می‌دهند که عملی چقدر مطلوب است. هدف عامل در یادگیری تقویتی این است که مجموع پاداش‌های دریافتی را بیشینه کند. همان‌گونه که اشاره کردیم، پاداش‌ها معمولاً توسط محیط به عامل ارائه می‌شوند و عامل باید با تجربه و تعامل با محیط، سیاست‌های مناسبی را یاد بگیرد تا بتواند پاداش‌های بیشتری دریافت کند.

در برخی موارد، پاداش‌ها می‌توانند به صورت فوری (Immediate Reward) تعریف شوند که به میزان پاداش دریافتی توسط عامل بلافاصله پس از انجام عمل در وضعیت مشخص اشاره دارند. به عنوان مثال، در یک بازی کامپیوتری، پاداش فوری می‌تواند امتیازی باشد که عامل بلافاصله بعد از انجام حرکتی دریافت می‌کند. علاوه بر پاداش‌های فوری، ممکن است پاداش‌های تاخیری (Delayed Reward) نیز در نظر گرفته شوند. پاداش‌های تاخیری به میزان پاداش دریافتی توسط عامل در طول زمان و به دلیل انجام عمل‌های متوالی در وضعیت‌های مختلف اشاره دارند. به عنوان مثال، در یک بازی استراتژیک، پاداش تاخیری می‌تواند امتیازی باشد که عامل پس از گذشت چند مرحله از بازی

دریافت می‌کند. تعریف پاداش در یادگیری تقویتی معمولاً به تفصیل صورت نمی‌گیرد و محتوا و مقدار پاداش‌ها بسته به مساله و برنامه‌ریزی خاص یادگیری تقویتی متفاوت است. در برخی موارد، از توابع ارزیابی پاداش (Reward Shaping Functions) نیز استفاده می‌شود تا پاداش‌های اصلی را با پاداش‌های فرعی یا تعدیل‌شده بهبود داده و فرآیند یادگیری را تسریع کرد.

4. تعریف سیاست

سیاست‌ها در یادگیری تقویتی نشان‌دهنده نحوه تصمیم‌گیری عامل بر اساس وضعیت‌های مشاهده شده است. سیاست می‌تواند به صورت قوانین (قواعد تصمیم‌گیری) یا به صورت تابع (تابع ارزش یا تابع عمل) تعریف شود. هدف عامل در این مرحله یادگیری سیاست بهینه است که مجموع پاداش‌ها را بیشینه کند. در یادگیری تقویتی، سیاست (Policy) به عنوان یک تابع تصمیم‌گیری تعریف می‌شود که نحوه عملکرد عامل در محیط را مشخص می‌کند. سیاست نشان می‌دهد که در هر وضعیت، عامل چه عملی را باید انتخاب کند. به طور ساده‌تر، سیاست مشخص می‌کند که عامل در وضعیت‌های مختلف چه کاری انجام دهد. سیاست می‌تواند به صورت تابعی مستقل از زمان (Time-Independent)، که به عنوان یک تابع ثابت عمل می‌کند یا به صورت تابعی وابسته به زمان (Time-Dependent) که می‌تواند تغییر کند، تعریف شود. در سیاست مستقل از زمان، عامل همیشه با توجه به وضعیت فعلی به یک عمل خاص تصمیم می‌گیرد، اما در سیاست وابسته به زمان، عامل ممکن است توسط تجربه‌های قبلی خود و یا با توجه به تغییرات محیط، سیاست خود را تغییر دهد. سیاست می‌تواند به صورت قاعده‌مند (Deterministic) یا به صورت تصادفی (Stochastic) تعریف شود. در سیاست قاعده‌مند، در هر وضعیت، عامل همیشه یک عمل خاص را انتخاب می‌کند. اما در سیاست تصادفی، عامل با احتمالات مشخصی بین چند عمل ممکن تصمیم می‌گیرد. سیاست می‌تواند به صورت ثابت (Static) تعریف شود که در آن عامل در هر وضعیت یک سیاست ثابت را دنبال می‌کند، یا به صورت دینامیک (Dynamic) که در آن عامل در طول زمان می‌تواند سیاست خود را تغییر دهد. تعریف سیاست معمولاً بر اساس تجربه‌های عامل و هدف یادگیری تقویتی تعیین می‌شود. هدف اصلی در یادگیری تقویتی بهینه‌سازی سیاست است تا عامل بتواند پاداش بیشتری دریافت کند و عملکرد بهتری در محیط داشته باشد.

5. تعامل با محیط

در این مرحله، عامل با محیط تعامل می‌کند و بر اساس سیاست‌های خود عمل‌هایی را انجام می‌دهد. هنگام انجام هر عمل، محیط پاداشی به عامل می‌دهد که نشان‌دهنده عمل مورد انتظار است. عامل بر اساس این بازخورد، سیاست‌ها و روش‌های یادگیری خود را بهبود می‌بخشد. تعامل با محیط در یادگیری تقویتی به صورت تعاملی و دوسویه بین عامل و محیط انجام می‌شود. در هر مرحله از فرآیند یادگیری، عامل با محیط تعامل می‌کند تا وضعیت فعلی را مشاهده کند، عملی را انجام دهد و پاداش مربوط به آن عمل را دریافت کند. این تعامل پیوسته در طول زمان ادامه می‌یابد تا عامل بتواند بر اساس تجربه‌های خود عملکرد خود را بهبود بخشد. فرآیند تعامل با محیط در یادگیری تقویتی عموماً به صورت زیر پیش می‌رود:

عامل وضعیت محیط را مشاهده می‌کند: در هر مرحله، عامل اطلاعات مشاهده شده از محیط را دریافت می‌کند که شامل جزییات وضعیت فعلی محیط و اطلاعات دیگری که ممکن است در تعیین عمل بعدی تاثیرگذار باشند، می‌شود.

عامل عمل را انتخاب می‌کند: بر اساس وضعیت فعلی، عامل بر اساس سیاست خود یک عمل خاص را انتخاب می‌کند. سیاست مشخص می‌کند که در هر وضعیت، عامل باید چه عملی را انجام دهد.

اجرای عمل توسط عامل در محیط: عامل عمل انتخاب شده را در محیط اجرا می‌کند. این عمل ممکن است تغییراتی در محیط ایجاد کند.

عامل پاداش را دریافت می‌کند: بعد از اعمال عمل، عامل پاداش مربوط به عمل انجام شده را از محیط دریافت می‌کند. پاداش می‌تواند از خود محیط دریافت شود و یا به عنوان ورودی به عامل داده شود.

وضعیت جدید را مشاهده می‌کند: پس از دریافت پاداش، عامل وضعیت جدید محیط را مشاهده می‌کند که نتیجه عمل انجام شده و تغییراتی که در محیط ایجاد شده است را نشان می‌دهد.

6. یادگیری و بهبود سیاست

در این مرحله، عامل بر مبنای امتیازاتی که دریافت کرده است، اشتباهات قبلی را برطرف می‌کند تا بتواند عملکرد دقیق‌تری از خود نشان دهد. دو روش مهم برای یادگیری و بهبود سیاست به شرح زیر است:

روش مستقیم: در این روش، عامل مستقیماً سیاست خود را بر اساس تجربه بهبود می‌دهد. یعنی عامل با توجه به پاداش‌های دریافتی از محیط، سعی می‌کند سیاست خود را به گونه‌ای تغییر دهد که پاداش کلی حاصله بیشتر شود. این روش به صورت معمول با استفاده از الگوریتم‌هایی مانند الگوریتم‌های Gradient Ascent، Policy Gradient و REINFORCE انجام می‌شود. در روش مستقیم، عامل پارامترهای سیاست خود را به صورت مستقیم به روزرسانی می‌کند. به این ترتیب، با تکرار تعامل با محیط و به‌روزرسانی سیاست، سیاست بهتری یاد می‌گیرد که منجر به عملکرد بهتر در محیط می‌شود.

روش غیرمستقیم: در این روش، عامل از تابع ارزش یا تابع عملکرد (value function) استفاده می‌کند تا سیاست خود را بهبود بخشد. تابع ارزش، تخمینی از ارزش هر وضعیت یا جفت وضعیت-عمل در محیط است و نشان می‌دهد که وضعیت یا جفت وضعیت-عمل چقدر ارزش دارد. عامل با استفاده از این تخمین، سعی می‌کند عملی را انتخاب کند که ارزش کلی را بیشینه کند. روش‌های غیرمستقیم شامل الگوریتم‌هایی مانند Q-Learning، SARSA و Deep Q-Networks (DQN) هستند. این الگوریتم‌ها با استفاده از تابع ارزش، تخمینی از بهترین عمل در هر وضعیت را محاسبه کرده و بر اساس آن، سیاست خود را بهبود می‌دهند. در روش غیرمستقیم، عامل ابتدا تخمینی از تابع ارزش را به دست می‌آورد و سپس بر اساس این تخمین، سیاست خود را بهبود می‌بخشد. این روش به صورت پیش‌بینی-ارزش‌گذاری (value-based) عمل می‌کند و به عامل امکان می‌دهد تا ارزش هر وضعیت را تخمین بزند.

همچنین، روش‌هایی مانند Actor-Critic و Proximal Policy Optimization (PPO) که ترکیبی از روش‌های مستقیم و غیرمستقیم هستند، نیز برای یادگیری و بهبود سیاست در یادگیری تقویتی استفاده می‌شوند. این الگوریتم‌ها همزمان تلاش می‌کنند تا سیاست را بهبود بخشند و تابع ارزش را بهینه کنند.

در کل، یادگیری و بهبود سیاست در یادگیری تقویتی به وسیله تعامل با محیط، تجربه از پاداش‌های دریافتی و استفاده از الگوریتم‌های مناسب برای بهبود سیاست، انجام می‌شود. هدف این فرآیند، یافتن سیاستی است که عامل را به عملکرد بهتر و بهبود پاداش کلی هدایت کند.

ب

بله، می‌توان از یادگیری تقویتی برای جستجوی ابرپارامترهای مثل اندازه تصویر ورودی و تعداد لایه‌ها در مسئله تشخیص اشیاء استفاده کرد. یادگیری تقویتی به عنوان یک رویکرد خودکار و تجربی، قابلیت یادگیری و بهینه‌سازی ابرپارامترها را از تجربیات مستمر با محیط وظیفه مشخص شده فراهم می‌کند.

در مسئله تشخیص اشیاء، این روش به صورت زیر ممکن است اجرا شود:

- عامل (Agent):** عامل ممکن است مدلی باشد که ابرپارامترهایی مثل اندازه تصویر ورودی، تعداد لایه‌ها و سایر ابرپارامترها را نمایش دهد.
 - عمل (Action):** عملکرد عامل می‌تواند شامل تغییر ابرپارامترها باشد. مثلاً، افزایش یا کاهش اندازه تصویر ورودی یا تعداد لایه‌ها.
 - پاداش (Reward):** پاداش می‌تواند از عملکرد مدل در وظیفه تشخیص اشیاء مشتق شود. به عنوان مثال، افزایش دقت تشخیص به معنای دریافت پاداش بیشتر است.
 - استراتژی (Policy):** استراتژی می‌تواند تعیین کننده این باشد که چگونه عامل تصمیم بگیرد تا ابرپارامترها را تغییر دهد. می‌تواند با استفاده از روش‌های تصادفی یا خودآموزشی مشخص شود.
 - تابع ارزیابی (Value Function):** می‌توان از تابع ارزیابی برای تخمین ارزش هر وضعیت استفاده کرد. این تابع می‌تواند نشان دهد که تغییر یک ابرپارامتر چقدر می‌تواند به بهبود یا بدتر شدن عملکرد منجر شود.
- با اجرای یادگیری تقویتی در مسئله تشخیص اشیاء، عامل می‌تواند به صورت تعاملی و بهینه‌سازی مستمر ابرپارامترها را انجام دهد. این رویکرد می‌تواند به مدل کمک کند تا به طور خودکار و با توجه به ویژگی‌های محیط، بهترین تنظیمات را برای ابرپارامترها انتخاب کند و در نتیجه، به کارایی و عملکرد بهتری برسد.

سوال پنجم:

1. Mode Collapse: در مدل‌های GAN، Mode Collapse ممکن است رخ دهد. این به معنای این است که مولد به جای اینکه تصاویر متنوعی را تولید کند، تمایل دارد تنها یک یا چند تصویر مشخص را تولید کند. حتی اگر تابع ضررها در انتهای دوره‌های آموزش نسبت به اولین دوره تغییر چندانی نکند، این مشکل می‌تواند باعث تفاوت کیفیت تصاویر تولیدی شود.
2. اعتباردهی به مولد و ممیز: ممکن است مقداردهی اولیه به شبکه‌های مولد و ممیز نقش داشته باشد. در حالی که تابع ضررها ممکن است در طول زمان تغییر نکند، اما شروع آموزش و مقداردهی اولیه به وزن‌های شبکه‌ها می‌تواند تأثیر گذار باشد و تفاوت در کیفیت تولید تصاویر را ایجاد کند.
3. اندازه نمونه‌ها و اندازه بردارهای نویز: ممکن است در طول آموزش، اندازه نمونه‌های آموزشی یا اندازه بردارهای نویز تغییر کند. این تغییرات می‌توانند تأثیرات مختلفی بر روی عملکرد مدل داشته باشند.
4. پایداری آموزش: پایداری آموزش GAN یک چالش است. در برخی مواقع، GAN ممکن است به سختی آموزش پیدا کند و ناپایدار شود. در این حالت، ممکن است تصاویر تولیدی به عنوان نتیجه نهایی در انتهای آموزش تفاوت‌هایی با نتایج اولیه داشته باشند.
5. مدل مولد تصویری از توزیع تصاویر آموزشی تولید میکند و این تصویر تولید شده به مرور بهتر میشود.