تفسیرپذیری در یادگیری ماشین

سجاد سبزی
محمدرضا احمدی تشنیزی

۲۵ آبان ۱۴۰۲

۱ مقدمه

تفسیرپذیری در یادگیری ماشین به توانایی درک و توضیح تصمیمات و پیش‌بینی‌هایی که توسط یک مدل انجام می‌شود، اشاره دارد. این نقش حیاتی را در برنامه‌های مختلفی ایفا می‌کند که شفافیت و اعتماد امور حیاتی هستند.

۲ اهمیت تفسیرپذیری

۱. اعتماد و پذیرش: کاربران احتمالاً بیشتر به مدل‌های یادگیری ماشین اعتماد و آن‌ها را پذیرفته‌تر می‌کنند اگر بتوانند دلیل پیش‌بینی‌ها را درک کنند.

۲. ملاحظات اخلاقی: تفسیرپذیری برای اطمینان از اینکه مدل‌ها تعصبی نشان ندهند یا تصمیمات ناعادلانه اتخاذ نکنند، بسیار حیاتی است و توسعه هوش مصنوعی اخلاقی را ترویج می‌کند.

۳. اشکال‌زدایی و بهبود: مدل‌های قابل تفسیر، اشکالات راحت‌تری دارند و امکان بهبود عملکرد را فراهم می‌کنند.

۴. تطابق با مقررات: در صنایع تنظیم‌شده، تفسیرپذیری ممکن است برای رفع مقررات قانونی لازم باشد.

۳ جایگاه تفسیرپذیری

- مراقبت‌های بهداشتی: توجیه‌پذیری در برنامه‌های پزشکی بسیار حیاتی است تا توصیه‌ها یا درمان‌ها را توضیح دهد.
- مالی: برای توضیح امتیاز اعتبار، شناسایی تقلب و توصیه‌های سرمایه‌گذاری حیاتی است.
- عدالت کیفری: برای شفافیت و عدالت در مدل‌های پیش‌بینی پلیس‌گری و ارزیابی خطر لازم است.
- وسایل نقلیه خودران: برای ایمنی حیاتی است تا تصمیماتی مانند ترمیم یا تغییر خطوط را توضیح دهد.

۴ مزایای تفسیرپذیری

۱. توضیح‌پذیری: کاربران می‌توانند پیش‌بینی‌ها را درک کرده و به آن‌ها اعتماد کنند که تصمیم‌گیری بهتری را تسهیل می‌کند.

۲. اشکال‌زدایی: شناسایی و حل مشکلات مدل به‌صورت آسان‌تر.

۳. عدالت: کمک به شناسایی و کاهش تعصب‌ها و ترویج عدالت در پیش‌بینی‌ها.

۴. نظارت انسانی: به افراد متخصص اجازه می‌دهد تا در صورت لزوم پیش‌بینی‌های ماشینی را تغییر یا اصلاح کنند.

۵ چالش‌ها و نگرانی‌ها

۱. مدل‌های پیچیده: ممکن است مدل‌های بسیار پیچیده به دلیل ساختارهای داخلی پیچیده خود تفسیرپذیری را از دست بدهند.

۲. تعادل با عملکرد: ممکن است یک تعادل بین تفسیرپذیری مدل و عملکرد پیش‌بینی وجود داشته باشد.

۳. حساسیت به سیاق: تفسیرهای مدل ممکن است بر اساس سیاق داده‌ها متغیر باشد.

۴. فهم کاربر: کاربران نمی‌توانند همیشه تخصص لازم را برای درک توضیحات پیچیده مدل داشته باشند.

۵. نگرانی‌های امنیتی و حریم خصوصی: مدل‌های تفسیرپذیر ممکن است اطلاعات حساسی را درباره داده‌های آموزش فاش کنند.

Grad-CAM (نگاه به نقشه فعال‌سازی با وزن‌های گرادیان)

مزایا:

۱. دقت مکان‌یابی: Grad-CAM تصویرسازی‌های با دقت بالا ارائه می‌دهد که ناحیه دقیق در یک تصویر ورودی که به پیش‌بینی مدل کمک می‌کند، را نشان می‌دهد. این امر باعث می‌شود که برای درک اینکه مدل دقیقاً کجا تمرکز کرده است، ارزشمند باشد.

۲. مستقل از مدل: Grad-CAM به یک معماری خاص محدود نیست و می‌تواند بر روی هر مدل قابل تفکیک کاربرد داشته باشد، که آن را چندکاره برای تفسیر یک مجموعه وسیع از شبکه‌های عصبی می‌سازد.

۳. انتگراسیون آسان: این تکنیک می‌تواند به راحتی به معماری‌های موجود شبکه‌های عصبی کانولوشنی متصل شود بدون انجام تغییرات قابل توجه، که به پژوهشگران و کارشناسان این امکان را می‌دهد که آن را به مدل‌های خود برای تفسیر قرار دهند.

چالش‌ها:

۱. ثبت جزئیات: Grad-CAM ممکن است در ثبت جزئیات پیچیده در تصویر ورودی مشکل داشته باشد، به ویژه در مواردی که مدل برای پیش‌بینی‌های خود از ویژگی‌های ظریف استفاده می‌کند. این محدودیت ممکن است بر تفسیر مدل‌های پیچیده تأثیر بگذارد.

۲. وابستگی به :Grad-CAM CNN به خصوص برای شبکه‌های عصبی کانولوشنی (CNNs) طراحی شده است. در حالی که هاCNN به طور معمول در وظایف مربوط به تصاویر استفاده می‌شوند، انواع دیگری از شبکه‌های عصبی ممکن است به تکنیک‌های دیگری برای تفسیر نیاز داشته باشند.

موارد استفاده:

۱. تصویربرداری پزشکی: Grad-CAM می‌تواند در تجزیه و تحلیل تصاویر پزشکی مورد استفاده قرار گیرد تا درک شود کدام ناحیه از یک تصویر برای تشخیص مدل حیاتی است. این می‌تواند تفسیرپذیری مدل‌های استفاده شده در حوزه بهداشت را افزایش دهد.

۲. تشخیص اشیاء: در وظایف بینایی ماشین، Grad-CAM می‌تواند به تصویرسازی ناحیه‌های مسئول برای دسته‌بندی اشیاء کمک کند و در اشکال‌زدایی و بهبود مدل کمک کند.

۳. اشکال‌زدایی مدل: Grad-CAM ابزار مهمی برای اشکال‌زدایی مدل‌های شبکه‌های عصبی است. با تصویرسازی ناحیه‌های فعال‌سازی، پژوهشگران می‌توانند تشخیص دهند که مدل به کدام قسمت‌های غیرمرتبط یا غیرمنتظر ورودی تمرکز کرده است.

مثال پیاده‌سازی:

پیاده‌سازی Grad-CAM در PyTorch - `https://github.com/jacobgil/pytorch-grad-cam`

LIME (توضیحات قابل تفسیر محلی و بی‌توجه به مدل)

مزایا:

۱. مستقل از مدل: LIME یک تکنیک مستقل از مدل است، که این امکان را فراهم می‌کند تا توضیحات محلی و قابل اعتماد برای پیش‌بینی‌های هر مدل یادگیری ماشینی فراهم کند. این باعث می‌شود که قابل استفاده برای یک طیف وسیع از مدل‌ها باشد.

۲. تفسیر محلی: LIME بر روی تولید توضیحات قابل اعتماد محلی تمرکز دارد، با اختلال در داده‌های ورودی و مشاهده تغییرات در پیش‌بینی‌های مدل. این باعث می‌شود که در فرآیند تصمیم‌گیری برای هر نمونه، بینش فراهم شود.

۳. چندکاره: LIME به معماری‌ها یا وظایف خاص محدود نیست و این امکان را دارد که یک ابزار چندکاره برای تفسیر انواع مختلف مدل‌های یادگیری ماشین باشد.

چالش‌ها:

۱. عدم تطابق جهانی در مقابل تطابق محلی: توضیحات LIME محلی هستند و ممکن است توانایی کامل در درک رفتار جهانی یک مدل پیچیده را نداشته باشند. توضیحات قابل اعتماد محلی ممکن است به دقت استراتژی تصمیم‌گیری کلی مدل را نشان ندهد.

۲. حساسیت به اختلالات: کیفیت توضیحات LIME می‌تواند به انتخاب روش‌ها و پارامترهای اختلال حساس باشد. انتخاب راهکارهای اختلال مناسب برای نتایج قابل اعتماد حیاتی است.

موارد استفاده:

۱. تفسیر مدل‌های جعبه‌سیاه: LIME به خصوص برای تفسیر تصمیمات مدل‌های جعبه‌سیاه که عملکرد داخلی آن‌ها شفاف نیست، مفید است. این امکان را فراهم می‌کند که درک شود مدل بر روی نمونه‌های خاص چگونه عمل می‌کند.

۲. اشکال‌زدایی و اعتماد: LIME می‌تواند در اشکال‌زدایی مدل کمک کند با برجسته‌سازی ویژگی‌های مهم برای پیش‌بینی‌های خاص. این به ساخت اعتماد به تصمیمات مدل کمک می‌کند و اطمینان حاصل می‌شود که مدل با انتظارات انسانی هماهنگ است.

۳. تجزیه و تحلیل عدالت و تعصب: LIME می‌تواند برای بررسی و حل مسائل عدالت و تعصب در مدل‌های یادگیری ماشین با بازرسی تأثیر ویژگی‌های ورودی بر پیش‌بینی‌ها بکار رود.

مثال پیاده‌سازی:

پیاده‌سازی LIME در PyTorch - https://github.com/jacobgil/pytorch-grad-cam

SHAP $_sHapleyAdditiveexPlanations$

مزایا:

۱. اندازه‌گیری یکپارچه اهمیت ویژگی: مقادیر SHAP ارزیابی یکپارچه اهمیت ویژگی‌ها بر اساس تئوری بازی همکاری ارائه می‌دهند. آن‌ها یک رویکرد پایه‌ای و از نظر نظری به منظور تفسیر تأثیر هر ویژگی بر خروجی مدل ارائه می‌دهند.

۲. مستقل از مدل: SHAP یک تکنیک مستقل از مدل است، که این امکان را فراهم می‌کند که بر روی انواع مختلف مدل‌های یادگیری ماشین بدون وابستگی به معماری مدل اعمال شود. این چندکاره بودن آن را برای یک طیف وسیع از مدل‌ها مناسب می‌سازد.

۳. تفسیرپذیری جهانی و محلی: مقادیر SHAP می‌توانند هم تفسیرپذیری جهانی و هم تفسیرپذیری محلی فراهم کنند. آن‌ها بینش‌هایی را در مورد تأثیر کلیه ویژگی‌ها در کل مجموعه داده فراهم می‌کنند و همچنین می‌توانند توضیحاتی درباره پیش‌بینی‌های خاص به صورت نمونه به نمونه ارائه دهند.

چالش‌ها:

۱. پیچیدگی محاسباتی: محاسبه مقادیر SHAP ممکن است هزینه محاسباتی داشته باشد، به ویژه برای مدل‌های بزرگ و پیچیده. اغلب بهبودها و بهینه‌سازی‌های کارا برای انجام محاسبات ضروری است.

۲. تعادل تفسیرپذیری: در حالی که مقادیر SHAP یک اندازه جامع اهمیت ویژگی ارائه می‌دهند، تفسیر این مقادیر ممکن است چالش برانگیز باشد. درک اثرات جهانی و محلی ویژگی‌ها در یک فضای بعد بالا ممکن است نیاز به تصویرسازی و تجزیه و تحلیل اضافی داشته باشد.

موارد استفاده:

۱. تحلیل اهمیت ویژگی: مقادیر SHAP به عنوان یک ابزار وسیع برای درک اهمیت ویژگی‌های مختلف در پیش‌بینی‌های یک مدل استفاده می‌شوند. این اهمیتی برای انتخاب ویژگی، بهبود مدل و درک عوامل تأثیرگذار در تصمیمات مدل است.

۲. توضیح در مدل‌های پیچیده: مقادیر SHAP به خصوص برای توضیح پیش‌بینی‌های مدل‌های پیچیده یادگیری ماشین، از جمله مدل‌های ترکیبی، شبکه‌های عصبی عمیق و مدل‌های با تعامل بین ویژگی‌ها، مفید هستند.

۳. تحلیل عدالت: مقادیر SHAP می‌توانند برای تجزیه و تحلیل و حل مسائل عدالت در مدل‌های یادگیری ماشین با ارزیابی مشارکت هر ویژگی در پیش‌بینی‌ها در گروه‌های دموگرافی مختلف استفاده شوند.

مثال پیاده‌سازی:

پیاده‌سازی SHAP در PyTorch - https://github.com/slundberg/shap

SmoothGrad

مزایا:

۱. کاهش نویز: SmoothGrad به منظور تسطیح نقشه‌های مهمی با افزودن نویز تصادفی به تصویر ورودی طراحی شده است. این کمک می‌کند تا حساسیت به تغییرات ورودی کاهش یابد و تفسیرهای قوی‌تری ایجاد شود.

۲. تقویت تفسیرپذیری: با کاهش تأثیر نویز در نقشه‌های مهم، SmoothGrad تفسیرپذیری تصاویر را بهبود می‌بخشد. این به تأمین بینش‌های واضح درباره نواحی تمرکز مدل کمک می‌کند و احتمال تفسیرهای گمراه‌کننده به دلیل نویز را کاهش می‌دهد.

۳. قابل اعمال بر روی مدل‌های مختلف: SmoothGrad یک تکنیک چندکاره است که می‌تواند بر روی انواع مختلف مدل‌های یادگیری ماشین و روش‌های تفسیرپذیری، از جمله نقشه‌های مهم و تصویرسازی‌های مبتنی بر گرادیان، قابل اعمال باشد.

چالش‌ها:

۱. تنظیم پارامترهای نویز: کارایی SmoothGrad به تنظیم صحیح پارامترهای نویز وابسته است. انتخاب سطوح مناسب نویز برای دستیابی به تعادل مطلوب بین کاهش نویز و حفظ اطلاعات مرتبط ضروری است.

۲. تأثیر بر تفسیر: در حالی که کاهش نویز مزیت دارد، تسطیح افراز بیش از اندازه ممکن است منجر به ساده‌سازی و از دست رفتن جزئیات حیاتی در نقشه‌های مهم شود. در نظر گرفتن دقیق سطوح نویز برای جلوگیری از اشتباه تفسیر حیاتی است.

موارد استفاده:

۱. تفسیرهای مقاوم به نویز: SmoothGrad در حالت‌هایی که داده ورودی ممکن است حاوی نویز یا اختلالات باشد، مفید است. این کمک می‌کند تا تفسیرهای پایدارتر و قابل اعتمادتر ایجاد شود، به خصوص زمانی که با داده‌های واقعی و نویزی سر و کار داریم.

۲. تفهیم بهتر مدل: با کاهش تأثیر نویز، SmoothGrad تفهیم بهتری از رویکرد تصمیم‌گیری مدل فراهم می‌کند. این به ویژه در شرایطی که ویژگی‌های ورودی نویزی ممکن است درک صحیح از اطلاعات مهم مدل را ایجاد کنند، مفید است.

۳. توضیحات بصری بهبود یافته: SmoothGrad به تولید توضیحات بصری و اطلاعاتی که در برابر تغییرات تصادفی در داده‌های ورودی مقاوم‌تر هستند، کمک می‌کند.

مثال پیاده‌سازی:

پیاده‌سازی SmoothGrad در PyTorch - `https://github.com/pkmr06/pytorch-smoothgrad`

نقشه‌های نمایشی $saliencyMaps$

مزایا:

۱. سادگی: نقشه‌های نمایشی یک روش ساده و شفاف برای نمایش قسمت‌های برجسته تر یک تصویر ورودی فراهم می‌کنند. سادگی این روش آن را برای اهداف تفسیری قابل دسترس می‌سازد.

۲. محاسبه آسان: محاسبه نقشه‌های نمایشی به طور کلی ساده است و می‌توان آن را بر روی انواع معماری شبکه‌های عصبی بدون تغییرات مهم اعمال کرد. این شامل محاسبه گرادیان نسبت به ورودی است.

۳. تصویرسازی های قابل فهم: نقشه‌های نمایشی تصاویری تولید می‌کنند که به طور مستقیم نواحی یک تصویر ورودی را که بر تصمیمات مدل تأثیر می‌گذارند، مشخص می‌کنند. این نمایش مفید در درک تمرکز توجه مدل است.

چالش‌ها:

۱. نتایج نویزی: نقشه‌های نمایشی ممکن است نتایج نویزی ایجاد کنند، به ویژه در حضور معماری‌های پیچیده مدل یا هنگام مواجهه با داده ورودی حاوی الگوهای ظریف. این می‌تواند به دقت تفسیر اثر بگذارد.

۲. محدودیت در زمینه: نقشه‌های نمایشی تفسیراسیون را پیکسل به پیکسل ارائه می‌دهند اما ممکن است اطلاعات متنوع در مورد نحوه تعامل ویژگی‌ها را ارائه ندهند. درک زمینه گسترده اهمیت ویژگی‌ها برای تفسیر جامع مدل مهم است.

موارد استفاده:

۱. اشکال زدایی مدل: نقشه‌های نمایشی برای اشکال زدایی مدل‌های شبکه‌های عصبی ارزشمند هستند، با شناسایی قسمت‌هایی از ورودی که بیشترین اثر را در پیش‌بینی‌های خاص دارند. این به درک و بهبود رفتار مدل کمک می‌کند.

۲. تصویرسازی توجه: نقشه‌های نمایشی به تصویرسازی مکانیزم توجه مدل کمک می‌کنند، نشان می‌دهند کدام قسمت‌های داده ورودی برای انجام پیش‌بینی‌ها به عنوان مهم ترین در نظر گرفته می‌شوند. این به ویژه در پردازش زبان طبیعی و وظایف مرتبط با تصویر مفید است.

۳. آموزش و ارتباط: نقشه‌های نمایشی به عنوان ابزارهای آموزشی برای توضیح تصمیمات مدل یادگیری ماشین به صورت دسترسی بصری خدمت می‌کنند. آن‌ها در ارتباط بین افراد فنی و غیرفنی کمک می‌کنند.

مثال پیاده‌سازی:

پیاده‌سازی نقشه‌های نمایشی در PyTorch - `https://github.com/pytorch/captum`