گزارش پروژه: تفسیرپذیری در شبکه‌های عمیق

آرمان ملک‌زاده لشکریانی
دانشکده مهدسی کامپیوتر
دانشگاه صنعتی شریف
malekzaadeh@ce.sharif.edu

فهرست مطالب

چکیده

امروزه در بسیاری از حوزه‌ها از جمله پردازش تصویر و متن، از شبکه‌های عصبی عمیق استفاده می‌شود. علیرغم عملکرد دقیق این مدل‌ها، توانایی درک آن‌ها توسط یک انسان و استدلال بر مبنای آن‌ها یک چالش محسوب می‌شود. برای حل این چالش که ما از آن اصطلاحا با نام «تفسیرپذیری» یاد می‌کنیم، روش‌های مختلفی پیشنهاد شده است. در این نوشتار، ما نتایج استفاده از سه روش تفسیرپذیری LIME، Kernel SHAP و Layer-wise Relevance Propagation را برای دو مجموعه داده MNIST و CIFAR10 مورد بررسی قرار می‌دهیم. هدف از این بررسی، مقایسه این روش‌ها و دست‌یابی به درکی از نحوه عملکرد شبکه‌های عصبی عمیق آموزش‌دیده است.

۱ مقدمه

در سال‌های اخیر، پیشرفت‌های عظیمی به کمک یادگیری عمیق صورت گرفته است و از مدل‌هایی که بر این پایه طراحی می‌شوند، در حوزه‌های بسیاری از جمله پردازش متن و تصاویر استفاده می‌شود. با این حال، عدم درک صحیحی از نحوه عملکرد این سیستم‌ها، به عنوان مانعی جدی برای به کارگیری آن‌ها در زمینه‌هایی به شمار می‌رود که عواقب یک تصمیم نادرست می‌تواند هزینه‌های سنگین و گاه غیرقابل جبرانی را به مسئولین یا استفاده‌کنندگان تحمیل کند. از جمله این زمینه‌ها، حوزه‌هایی مانند پزشکی و روانشناسی است که مدل باید به تصمیم‌گیری درباره یک انسان و یا وضعیت جسمی و روانی او کمک کند. از همین رو، روش‌های تفسیر شبکه‌های عصبی به تازگی محبوبیت فراوانی بدست آورده‌اند و تحقیقات گسترده‌ای در این زمینه در حال انجام است.

با درک عمیق‌تر نحوه عملکرد یک مدل، قادر خواهیم بود مانند یک بیمار با او برخورد کنیم و نقاط ضعف او را مورد بررسی قرار دهیم. در نهایت ممکن است با برطرف کردن کاستی‌های مدل، دقت عملکرد آن افزایش پیدا کند و مدلی با قابلیت اطمینان بیشتر در اختیار داشته باشیم. به طور مثال، اگر بیماری یک فرد با استفاده از یک سیستم بر مبنای هوش مصنوعی تشخیص داده شود و دلایل این تشخیص نیز ذکر گردد، پزشک می‌تواند بر اساس دانشی که دارد، این تصمیم را بپذیرد و یا رد نماید. در این مثال، در واقع سیستم به عنوان دستیار پزشک عمل کرده است؛ با این تفاوت که این دستیار، احتمالا پرونده‌های پزشکی بسیار زیادی را قبلا مطالعه نموده و دانش نسبتا بالایی کسب کرده است.

در یک نگاه، روش‌های تفسیر شبکه‌های عصبی به دو نوع کلی تقسیم‌بندی می‌شوند:

- روش‌های post-hoc: در این گونه روش‌ها، مدلی که آموزش داده‌شده و موجود است، مورد تفسیر قرار می‌گیرد. از جمله این روش‌ها می‌توان به «اهمیت ویژگی جایگشت‌یافته» [1] اشاره کرد. در این روش، مقادیر مربوط به یک ویژگی، جایگشت داده می‌شوند؛ به طوری که مثلا ممکن است مقدار متناظر آن ویژگی در نمونه i –ام با همان مقدار از نمونه j– ام جایگزین گردد. پس از این جایگزینی، افت دقت مدل اندازه‌گیری می‌شود و اگر آن ویژگی تاثیر زیادی در پیش‌بینی مدل داشته باشد، انتظار می‌رود که افت چشم‌گیری در دقت مدل مشاهده شود.
- روش‌های ad-hoc: در این روش‌ها، پس از آموزش مدل، از روی آن یک مدل تفسیرپذیر آموزش داده می‌شود. از آنجایی که مدل اصلی معمولا به ندرت قابل تفسیر می‌باشد، مدل تفسیرپذیری که بدست‌آمده، می‌تواند مثلا به صورت محلی رفتار مدل اصلی را توضیح دهد. از جمله این روش‌ها، روش LIME است که در بخش ۱.۲ توضیح داده می‌شود.

ذکر این نکته حائز اهمیت است که میان دانشمندان مختلف، توافقی روی مفهوم تفسیرپذیری صورت نگرفته است و تعریف یکتایی از آن موجود نیست. اما به صورت کلی می‌توان تفسیرپذیری را به عنوان محدوده توانایی انسان برای درک یک مدل و استدلال بر مبنای آن در نظر گرفت.

مفهوم تفسیرپذیری را می‌توان در سه سطح مورد بررسی قرار داد:

- قابلیت شبیه‌سازی [2]: به معنای درک ما از کل مدل است و مربوط به فهم مکانیزم عملکرد مدل در سطح بالا می‌شود. از همین رو، هر چه مدل ساده‌تری داشته باشیم، قابلیت شبیه‌سازی آن بیشتر خواهد بود. برای مثال، یک دسته‌بند خطی کاملا قابل فهم و شبیه‌سازی است. در بررسی این قابلیت، می‌توانیم به این فکر کنیم که انسان تا چه اندازه توانایی شبیه‌سازی یک مدل را دارد و از خود بپرسیم «آیا یک انسان می‌تواند اتفاقاتی که در مدل می‌افتند را در ذهن خود به آسانی تصور کند؟».

[1] Permutation Feature Importance
[2] Simulatability

- تجزیه‌پذیری [۳]: مربوط به درک ما از اجزای یک مدل است. به طور مثال در یک شبکه عصبی، درک نورون‌ها، لایه‌ها، بلوک‌ها مد نظر است. به طور مثال، یک درخت تصمیم، از تجزیه‌پذیری بالایی برخوردار است؛ زیرا به طور مشخص می‌دانیم که هر گره مربوط به چه تصمیمی است و در نهایت برگ‌ها چگونه با استفاده از تجمیع نتایج گره‌های قبلی، تصمیم نهایی را نمایندگی می‌کنند. به طور خاص در شبکه‌های عصبی، تجزیه‌پذیری و دانستن نقش هر یک از اجزاء در کل مدل، می‌تواند به بهینه‌سازی آن کمک کند.
- شفافیت الگوریتمی [۴]: مربوط به درک پروسه آموزش مدل و دینامیک آن می‌باشد. گاهی اوقات تابع هزینه در یک شبکه عصبی بسیار نامحدب است. در این شرایط، مدل عمیق ما جواب یکتا نخواهد داشت و این امر از شفافیت الگوریتمی آن می‌کاهد. در حال حاضر بسیاری از روش‌های آموزش مدل‌ها که عموما بر مبنای محاسبه گرادیان و حرکت در جهت کاهش آن عمل می‌کنند، منجر به عملکرد بسیار خوب شبکه‌های عمیق شده‌اند. درک این که چرا این الگوریتم‌های یادگیری، تا این حد خوب عمل می‌کنند، می‌تواند باعث ایجاد زمینه‌های تحقیقاتی جدیدی در مباحث یادگیری عمیق گردد و پیشرفت‌های چشم‌گیری را رقم بزند [۱].

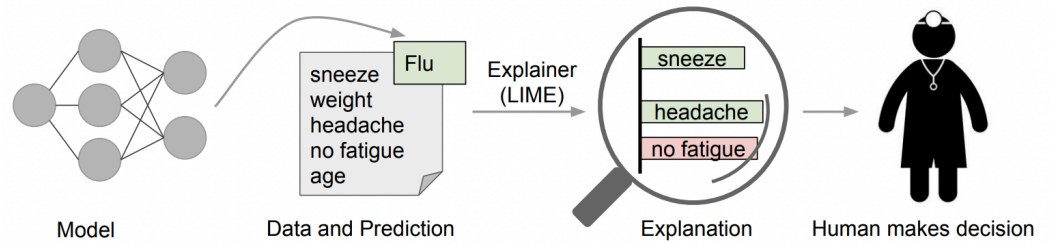
موضوع دیگری که باید مورد توجه قرار گیرد، تعدد روش‌های تفسیر شبکه‌های عصبی است. می‌دانیم هیچ یک از این روش‌ها به صورت کامل قابل اطمینان نیستند و هر یک نقص‌هایی دارند. هر روش، ویژگی‌های متفاوتی را به عنوان ویژگی مهم تشخیص می‌دهد و به همین دلیل، نیاز است معیاری وجود داشته باشد که امکان محاسبه و مقایسه دقت و کارایی روش‌های تفسیر را فراهم آورد. یکی از استراتژی‌های معمول برای انجام این مقایسه، حذف ویژگی‌های مهم (از دیدگاه یک روش تفسیر) از ورودی و محاسبه افت دقت مدل می‌باشد. انتظار داریم که اگر واقعا یک روش توانسته باشد مهم‌ترین ویژگی‌ها را تشخیص بدهد، با حذف آن‌ها، خطای محسوسی در عملکرد مدل مشاهده شود. اما این روش یک مشکل اساسی نیز دارد. با حذف ویژگی‌هایی که مهم تشخیص داده شده‌اند، به یک مجموعه داده آزمایشی [۵] جدید می‌رسیم که توزیع آن با توزیع مجموعه داده آموزشی [۶] یکسان نیست و این خلاف فرض اساسی ما در یادگیری ماشین است. از همین جهت، نمی‌توان با اطمینان تشخیص داد که افت دقت مدل، به علت تشخیص صحیح ویژگی‌های مهم بوده و یا تغییر در توزیع داده‌ها سبب این افت عملکرد شده است [۲].

۲ روش‌های منتخب

۱.۲ LIME

یکی از تکنیک‌هایی است که به منظور توضیح‌دادن پیش‌بینی‌های یک دسته‌بند به کار می‌رود؛ به گونه‌ای که یک مدل تفسیرپذیر در اطراف پیش‌بینی یاد گرفته می‌شود و از آن استفاده می‌گردد. مسئله‌ای که LIME به آن پاسخ می‌دهد، مسئله‌ی اعتماد به یک پیش‌بینی است. با انتخاب چندین نمونه پیش‌بینی و تفسیر آن‌ها، عملا همین روش به گونه‌ای گسترش داده می‌شود که به مسئله‌ی اعتماد به مدل پاسخ دهد. Ribeiro و همکاران علاوه بر پیشنهاد LIME همچنین نشان دادند که از این روش می‌توان برای مهندسی ویژگی‌ها به منظور بهبود عملکرد مدل‌های یادگیری ماشین نیز استفاده نمود [۳].

منظور از توضیح یک پیش‌بینی، ارائه شواهد متنی یا بصری است که به درک کیفی ارتباط میان اجزای نمونه‌ها (مثلا کلمات در یک متن یا بخش‌های یک تصویر) و پیش‌بینی مدل کمک کنند. شکل ۱ نحوه تفسیر نتایج پیش‌بینی مربوط به یک بیمار توسط این روش را نشان می‌دهد.



شکل ۱: یک مدل تشخیص داده است که بیمار آنفولانزا دارد و LIME علائمی در تاریخچه بیماری او را که به این پیش‌بینی منجر شده، برجسته کرده است. همانطور که مشاهده می‌شود، عطسه و سردرد به عنوان شواهدی برای تایید این پیش‌بینی و عدم خستگی به عنوان شواهدی بر رد آن توسط مدل در نظر گرفته شده‌اند.

[3] Decomposability
[4] Algorithmic Transparency
[5] Testing Dataset
[6] Traning Dataset

از جمله دلایلی که باعث شده محققان به جای تلاش برای توضیح عملکرد کلی مدل، به سراغ توضیح تک تک پیش‌بینی‌های آن بروند، این است که ارزیابی روی داده‌های تست و یا اعتبارسنجی، می‌تواند منجر به نتیجه‌گیری غلط درباره عملکرد مدل روی داده‌های واقعی بشود. معمولا پژوهشگران و مهندسان، دقت مدل‌های خود را بیشتر از آنچه واقعیت دارد ارزیابی می‌کنند. همچنین، ممکن است داده‌های مورد استفاده برای تست، تفاوت‌های چشم‌گیری با داده‌های آموزشی داشته باشند و از همین رو، نتایج بدست‌آمده قابلیت اطمینان خود را از دست بدهند.

از جمله معیارهای یک توضیح خوب برای عملکرد مدل، تفسیرپذیری آن است؛ به عبارت دیگر، باید بتوان درکی کیفی از ارتباط میان متغیرهای ورودی و پاسخ مدل بدست آورد. در راه رسیدن به این درک، همواره باید محدودیت‌های کاربر در نظر گرفته شوند. مثلا ممکن است یک بردار گرادیان بسیار طولانی، برای خیلی از کاربران قابل درک نباشد. همچنین مدل باید حداقل به صورت محلی قابل اطمینان باشد؛ بدین معنا که در همسایگی نمونه‌ای که خروجی آن بدست آمده، رفتارش قابل پیش‌بینی باشد. البته باید به این نکته توجه داشت که پیش‌بینی رفتار مدل حول یک نقطه، به معنای دانستن نحوه رفتار کلی آن نیست. اما این به هر حال می‌تواند معیار مناسبی برای ارزیابی مدل باشد؛ خصوصا زمانی که ارزیابی کلی به علت پیچیدگی بیش از حد مدل قابل انجام نیست. همچنین، یک توضیح مناسب باید مستقل از مدل عمل کند؛ بدین معنا که بتوان رفتار هر مدلی را با استفاده از آن درک نمود.

در روش LIME، علاوه بر توجه به این معیارها، به تفسیرپذیری بازنمایی ورودی نیز توجه می‌شود. در این راستا، باید در نظر داشته باشیم که ویژگی‌هایی که به عنوان ورودی به مدل داده می‌شوند، ممکن است برای یک انسان عادی قابل درک نباشند. به طور مثال، تصاویر ممکن است در یک شبکه عصبی عمیق، به شبکه تنسورهای سه-بعدی نمایش داده شوند که هر بعد آن‌ها، میزان شدت یکی از نورهای سبز، قرمز و یا آبی را نشان می‌دهد. در این حوزه، یک بازنمایی تفسیرپذیر از همان تصویر، می‌تواند یک بردار متشکل از تعدادی صفر و یک باشد که هر کدام از ابعاد آن، حضور یا عدم حضور بخشی از تصویر در ورودی شبکه را مشخص می‌کنند.

۱.۱.۲ نمادگذاری و تعریف دقیق مسئله

فرض کنید $x \in \mathbb{R}^d$ بازنمایی اصلی یک شیء در کامپیوتر و $x' \in \{0,1\}^{d'}$ بازنمایی تفسیرپذیر آن باشد. همچنین در نظر بگیرید که G یک کلاس از مدل‌ها باشد و $g \in G$ یکی از اعضای آن باشد. دامنه‌ی g، $\{0,1\}^{d'}$ است که حضور یا عدم حضور عناصر تفسیرپذیر در ورودی است. ما همچنین به یک معیار سنجش پیچیدگی هستیم. این معیار را تابعی مانند $\Omega(g)$ در نظر بگیرید. مثلا در مورد یک درخت تصمیم، این تابع می‌تواند عمق درخت را اندازه بگیرد و یا درباره یک مدل خطی، وزن‌های غیر صفر آن توسط این تابع شمرده می‌شوند.

معمولا در شبکه‌های عصبی که برای دسته‌بندی به کار می‌روند، برای مدل‌سازی احتمال تعلق یک نمونه به هر کلاس، یک نورون در آخرین لایه شبکه در نظر گرفته می‌شود. فرض کنید $f : \mathbb{R}^d \to \mathbb{R}$ تابعی باشد که ورودی اصلی شبکه را دریافت می‌کند و احتمال تعلق آن به یک کلاس خاص را تحویل می‌دهد. برای هر نمونه مانند x و هر نمونه دیگر مانند z میزان نزدیکی z به x را با تابع $\pi_x(z)$ مدل می‌کنیم. از این تابع در واقع برای تعریف همسایگی محلی حول x استفاده می‌شود. اگر $L(f, g, \pi_x)$ تابعی باشد که نشان‌دهنده میزان غیرقابل اعتماد بودن g در تخمین f حول همسایگی x با معیار π_x می‌باشد، ما برای تضمین تفسیرپذیری در عین قابل اطمینان بودن به صورت محلی، باید ضمن پایین نگه‌داشتن $\Omega(g)$ به نحوی که g برای انسان قابل تفسیر باشد، $L(f, g, \pi_x)$ را نیز کمینه کنیم. به عبارت دیگر در روش LIME ما به یک مدل تفسیرپذیر مانند $\epsilon(x)$ می‌رسیم که طبق رابطه ۱ تعریف می‌گردد.

$$\epsilon(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \qquad (۱)$$

برای یادگیری نحوه رفتار محلی f زمانی که ورودی‌های تفسیرپذیر تغییر می‌کنند، ما $L(f, g, \pi_x)$ را با نمونه‌گیری و وزن‌دهی آن نمونه‌ها بر حسب میزان نزدیکی به x تخمین می‌زنیم. برای هر نمونه تفسیرپذیر مانند $x' \in \{0,1\}^{d'}$ ، ما یک یا تعدادی نمونه تغییریافته مانند $z' \in \{0,1\}^{d'}$ می‌سازیم که فقط بخشی از عناصر غیر صفر x' را داراست. برای هر نمونه ساختگی مانند z' ، بازنمایی اصلی آن یعنی $z \in \mathbb{R}^d$ را بازیابی می‌کنیم و $f(z)$ را بدست می‌آوریم. از $f(z)$ به عنوان برچسب این نمونه استفاده می‌کنیم تا به یک مدل توضیح‌دهنده برسیم.

پس ما برای هر نمونه ساختگی، یک برچسب هم داریم. حالا کل این مجموعه داده ساختگی را Z بنامید. با استفاده از همین مجموعه داده، رابطه ۱ را حل می‌کنیم و $\epsilon(x)$ را بدست می‌آوریم که همان مدل قابل تفسیر مورد نظر ما است. در روش LIME ما کلاس G را مجموعه مدل‌های خطی در نظر می‌گیریم؛ به طوری که داریم:

$$g(z') = w_g z' \qquad (۲)$$

به طور خاص، در این روش توابع نزدیکی نمونه‌ها و خطای اطمینان نیز از روابط ۳ و ۴ بدست می‌آیند.

$$\pi_x(z) = \exp(-D(x,z)^2/\sigma^2) \tag{۳}$$

$$L(f,g,\pi_x) = \sum_{z,z' \in Z} \pi_x(z)(f(x) - g(z'))^2 \tag{۴}$$

برای ارائه درکی بهتر از این نمادها، فرض کنید ورودی‌های شبکه، بازنمایی‌های برداری کلمات هستند. اگر هر کلمه را با یک بردار ۳۰۰-بعدی نشان دهیم، آنگاه خواهیم داشت $x \in \mathbb{R}^{300}$ و این بردار طولانی برای انسان قابل درک نیست. مثلا اگر یک جمله ۱۰ کلمه‌ای داشته باشیم، شخص باید با ۱۰ بردار ۳۰۰-تایی (به عبارت ۳۰۰۰ ویژگی) روبرو شود که معنایشان را نمی‌داند. به جای این کار می‌توانیم یک نمایش ساده‌تر برای جمله را در نظر بگیریم؛ به نحوی که برای حضور یا عدم حضور هر کلمه، یک ۰ یا ۱ در بردار قرار داده شود. اگر از میان کل کلمات زبان، فقط 50 تای پرتکرار آن را در نظر بگیریم، برای نمایش هر جمله تنها به یک بردار تفسیرپذیر مانند $x' \in \{0,1\}^{50}$ نیاز خواهیم داشت که بعد i ام آن، نشان‌دهنده حضور یا عدم حضور کلمه i ام زبان در آن جمله است. با حذف و یا افزودن هر کلمه به جمله واقعی، یک بردار جدید مانند $z' \in \mathbb{R}^{50}$ بدست می‌آوریم. در زمینه پردازش تصاویر، به جای اینکه هر بُعد بردار نشانه حضور یک کلمه باشد، حضور یا عدم حضور بخشی از تصویر در ورودی را نشان می‌دهد. حداکثر تعداد مولفه‌هایی از ورودی که می‌توانند در این بردار در نظر گرفته شوند، می‌تواند معیاری برای تعریف تابع $\Omega(g)$ باشد.

نهایتا پس از یافتن تابع g که از حل معادله ۱ بدست می‌آید، ما به هر بخش قابل تفسیر ورودی، یک وزن نسبت می‌دهیم و کلیه وزن‌ها در برداری مانند w_g ذخیره خواهند شد. بر اساس این وزن‌ها می‌توانیم متوجه شویم که مدل به کدام بخش ورودی اهمیت بیشتری داده است. شکل ۲ یک نمونه از استفاده LIME در حوزه پردازش تصویر را نشان می‌دهد.

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

شکل ۲: توضیح دسته‌بندی نسبت‌داده‌شده به یک تصویر که توسط شبکه عصبی Inception گوگل انجام شده است. بیشترین امتیازها مربوط به گیتار الکتریک، گیتار آکوستیک و نوعی سگ به نام لابرادور بوده است که به ترتیب در شکل‌های b,c,d قسمت‌های مهم تصویر که منجر به تشخیص آن‌ها شده، برجسته شده است.

۲.۲ Kernel SHAP

در این روش که توسط Lundberg و Lee [۴] معرفی شد، به هر ویژگی یک سهم نسبت داده می‌شود که آن را اصطلاحا Shapley value می‌نامیم. این پژوهشگران مفهوم مدل توضیح‌دهنده [۷] را مطرح نمودند که به هر تقریب تفسیرپذیری از مدل اصلی گفته می‌شود. مانند بخش ۱.۱.۲ ، فرض کنید f مدل اصلی مورد استفاده برای پیش‌بینی و g یک مدل توضیح‌دهنده باشد. همچنین، در نظر بگیرید که برای هر نمونه ورودی مدل اصلی مانند x ، تابعی مانند h_x وجود دارد که اگر هر نمونه‌ی ساده‌سازی شده از x مانند $x' \in \{0,1\}^M$ را دریافت کند، می‌تواند از روی آن به بازسازی x بپردازد. به عبارت دیگر فرض کنید رابطه ۵ برقرار است.

$$x = h_x(x') \tag{۵}$$

در این صورت، یک مدل تفسیرپذیر محلی، به دنبال آن خواهد بود که برای هر $z' \approx x'$ خروجی را به نحوی تولید کند که بسیار نزدیک به خروجی مدل اصلی در صورت بازسازی z از روی z' توسط تابع h_x باشد. به عبارت دقیق‌تر، مدل تفسیرپذیر (توضیح‌دهنده) g زمانی توضیح مناسبی به ما می‌دهد که رابطه ۶ برقرار باشد.

$$z' \approx x' \implies g(z') \approx f(h_x(z')) \tag{۶}$$

[7] Explanation Model

در روش Shapley کلاسیک، اگر ویژگی‌های مجموعه داده اصلی را F بنامیم، مدل باید روی تمامی زیرمجموعه‌های آن مانند $S \subseteq F$ آموزش داده شود. این روش به هر ویژگی یک مقدار اهمیت نسبت می‌دهد که نشان‌دهنده تاثیر آن در پیش‌بینی مدل است. این مقدار اهمیت را برای ویژگی i –ام با ϕ_i نشان می‌دهیم. برای محاسبه این تاثیر، یک مدل مانند $f_{S\cup\{i\}}$ روی داده‌هایی که شامل ویژگی i –ام هستند آموزش داده می‌شود و مدلی دیگر مانند f_S بدون حضور این ویژگی در داده‌ها آموزش می‌بینند. سپس، خروجی‌های دو مدل روی همین ورودی با و بدون حضور ویژگی i –ام مقایسه می‌شوند و اختلاف آن‌ها که با $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$ نشان داده می‌شود، بدست می‌آید. توجه کنید که در اینجا، همه زیرمجموعه‌های ممکن مانند $S \subseteq F\backslash\{i\}$ در نظر گرفته می‌شوند. نهایتا مقادیر تاثیر مربوط به ویژگی‌ها از رابطه ۷ محاسبه می‌گردد که به نوعی میانگین وزن‌دار تمامی حالت‌های ممکن است.

$$\phi_i = \sum_{S\subseteq F\backslash\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{۷}$$

با داشتن تاثیر هر ویژگی، می‌توانیم مدل توضیح‌دهنده را طبق رابطه ۸ تعریف کنیم

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{۸}$$

که در آن z'_i ویژگی i–ام موجود در z' را نشان می‌دهد.

اثبات شده است که مقادیر Shapley ویژگی‌های زیر را دارند:

- دقت محلی: طبق این ویژگی، وقتی مدل اصلی – یعنی f را روی یک ورودی خاص مانند x تخمین می‌زنیم، مدل توضیح‌دهنده باید حداقل روی ورودی ساده‌سازی شده – یعنی x' همان خروجی را تولید کند که f برای x تولید می‌کرده است. رابطه ۹ این ویژگی را فرموله می‌کند.
$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{۹}$$
- فقدان: طبق این ویژگی، اگر ورودیِ ساده‌سازی‌شده – یعنی x' عدم حضور یک ویژگی را نشان بدهد، آنگاه تاثیر آن ویژگی باید صفر باشد. به بیان دقیق‌تر باید رابطه ۱۰ برقرار شود.
$$x'_i = 0 \implies \phi_0 = 0 \tag{۱۰}$$
- سازگاری: طبق این ویژگی، اگر یک مدل به گونه‌ای تغییر کند که سهم یک ویژگی از ورودیِ ساده‌سازی‌شده افزایش یابد و یا صرف نظر از بقیه ویژگی‌ها، ثابت بماند، تاثیر آن ورودی نباید کاهش یابد. به بیان دقیق‌تر، فرض کنید $f_x(z') = f(h_x(z'))$ و $z'\backslash i$ نشان‌دهنده صفر بودن ویژگی i–ام در z' باشد. برای هر دو مدل مانند f و f' اگر داشته باشیم:
$$\forall z' \in \{0,1\}^M \quad f'_x(z') - f'_x(z'\backslash i) \geq f_x(z') - f_x(z'\backslash i) \tag{۱۱}$$
آنگاه خواهیم داشت:
$$\phi_i(f', x) \geq \phi_i(f, x) \tag{۱۲}$$
در رابطه ۱۲ منظور از $\phi_i(f, x)$ تاثیر ویژگی i –ام در زمانی است که مدل f ورودی x را دریافت کند.

نشان داده شده که اگر توابع خاصی را در روش LIME در نظر بگیریم، تابع g که در LIME بدست می‌آید نیز همین ویژگی‌ها را خواهد داشت. از طرفی، در قضیه‌ای دیگر اثبات شده است که فقط ۱ تابع همه این ویژگی‌ها را دارد. لذا از این دو حقیقت، نتیجه می‌گیریم که می‌توان مقادیر Shapley را توسط روش LIME تخمین زد. برای این کار، کافیست در روش LIME توابع $\pi_{x'}, L, \Omega$ را به شکل زیر تعریف نماییم:

$$\Omega(g) = 0$$
$$\pi_{x'}(z') = \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)}$$
$$L(f, g, \pi_{x'}) = \sum_{z'\in Z}[f(h_x^{-1}(z')) - g(z')]^2 \pi_{x'}(z')$$

در روابط فوق، منظور از $|z'|$ تعداد عناصر غیر صفر موجود در z' می‌باشد.

۳.۲ Layer-wise Relevance Propagation

یکی دیگر از تکنیک‌هایی است که برای تفسیر شبکه‌های عصبی عمیق به کار می‌رود [۵]. روش این تکنیک، از انتشار [۸] پیش‌بینی مدل - همان $f(x)$ - به سمت عقب با قواعدی خاص پیروی می‌کند. ویژگی مهمی که LRP از آن تبعیت می‌کند، حفاظت [۹] نام دارد. طبق این ویژگی، هر آنچه توسط یک نورون دریافت شده است، باید به همان مقدار میان نورون‌های لایه قبلی (پایینی) توزیع شود.

فرض کنید j, k دو نورون در لایه‌های متوالی یک شبکه عصبی باشند. در این صورت، انتشار امتیاز مرتبط‌بودن [۱۰] از نورون k به نورون j که در لایه قبلی آن قرار دارد، طبق رابطه ۱۳ اتفاق می‌افتد.

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \tag{۱۳}$$

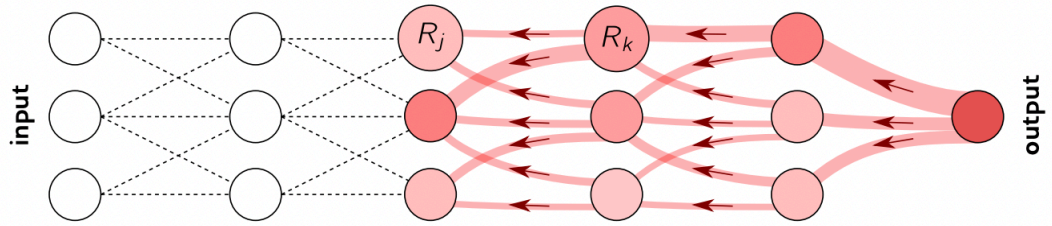
در این رابطه، z_{jk} میزان نقش‌داشتن نورون j در مرتبط‌ساختن نورون k را مدل می‌کند. مخرج کسر، برقراری ویژگی حفاظت را تضمین می‌کند. این انتشار زمانی متوقف می‌شود که به لایه ورودی برسیم. اگر همین قاعده را برای تمام نورون‌های شبکه اعمال کنیم، به سادگی می‌توان دید که ویژگی حفاظت، برای هر لایه نیز برقرار است. به عبارت دیگر داریم:

$$\sum_j R_j = \sum_k R_k \tag{۱۴}$$

با تعمیم این ویژگی، به صورت سراسری نیز ویژگی حفاظت برقرار خواهد بود و خواهیم داشت:

$$\sum_i R_i = f(x) \tag{۱۵}$$

شکل ۳ نحوه عملکرد این روش را نشان می‌دهد [۶].



شکل ۳: مصورسازی روش LRP. هر نورون آنچه را که از لایه بالایی خود دریافت کرده، میان نورون‌های لایه پایینی توزیع می‌کند.

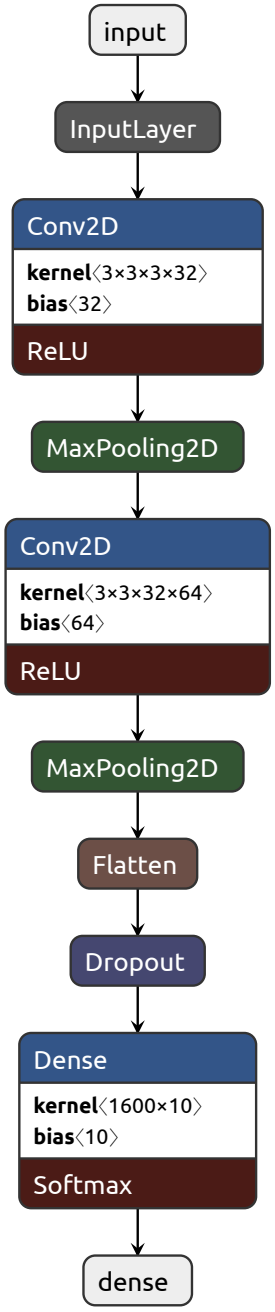
۳ شبکه

۱.۳ معماری مدل مجموعه داده MNIST

مجموعه داده MNIST[11] شامل تصاویری سیاه و سفید با ابعاد $(28, 28, 1)$ است. ما در ابتدا این تصاویر را به تصاویر RGB تبدیل می‌کنیم تا ابعاد آن‌ها به صورت $(28, 28, 3)$ در آید. این تغییر از آن جهت لازم است که در کتابخانه پایتون مربوط به روش تفسیر LIME [۱۲] تنها تصاویر از این نوع را می‌پذیرد. پس از این تغییر، هر تصویر وارد یک لایه پیچشی [۱۳] می‌شود تا ویژگی‌هایی از آن استخراج گردد. از میان این ویژگی‌ها، بزرگترین مقادیر که نماینده مهم‌ترین ویژگی‌ها می‌باشند، توسط یک لایه Max Pooling استخراج می‌گردد. سپس دوباره همین عمل تکرار می‌شود تا از میان ویژگی‌های باقیمانده نیز، با اهمیت‌ترین آن‌ها استخراج و انتخاب گردد. پس از این عملیات، کل ویژگی‌های استخراج‌شده، توسط یک لایه Flatten به صورت یک

[8] Propagation
[9] Conservation
[10] Relevance Score
[11] http://yann.lecun.com/exdb/mnist/
[12] https://github.com/marcotcr/lime
[13] Convolutional

بردار در می‌آیند. به منظور جلوگیری از رخ‌دادن پدیده‌ی بیش‌برازش [14] ، از یک لایه Dropout پس از این عملیات استفاده می‌شود. نهایتا با استفاده از یک لایه تماما متصل [15] از این بردار استخراج ویژگی صورت می‌گیرد و توسط یک لایه دیگر از همین نوع، ویژگی‌ها به یک بردار ۱۰ تایی نگاشت می‌شوند. از آنجایی که تابع فعال‌سازی این لایه Softmax می‌باشد، هر مولفه این بردار ۱۰ تایی، یک عدد بین ۰ و ۱ است و مجموع همه این مولفه‌ها ۱ می‌باشد. هر مولفه را متناظر یک دسته و مقدار آن را به عنوان احتمال تعلق ورودی به آن دسته در نظر می‌گیریم. شکل ۴ معماری مدل به کار رفته برای دسته‌بندی تصاویر مجموعه داده MNIST را نشان می‌دهد.



شکل ۴: معماری مدل به کار رفته برای دسته‌بندی تصاویر مجموعه داده MNIST

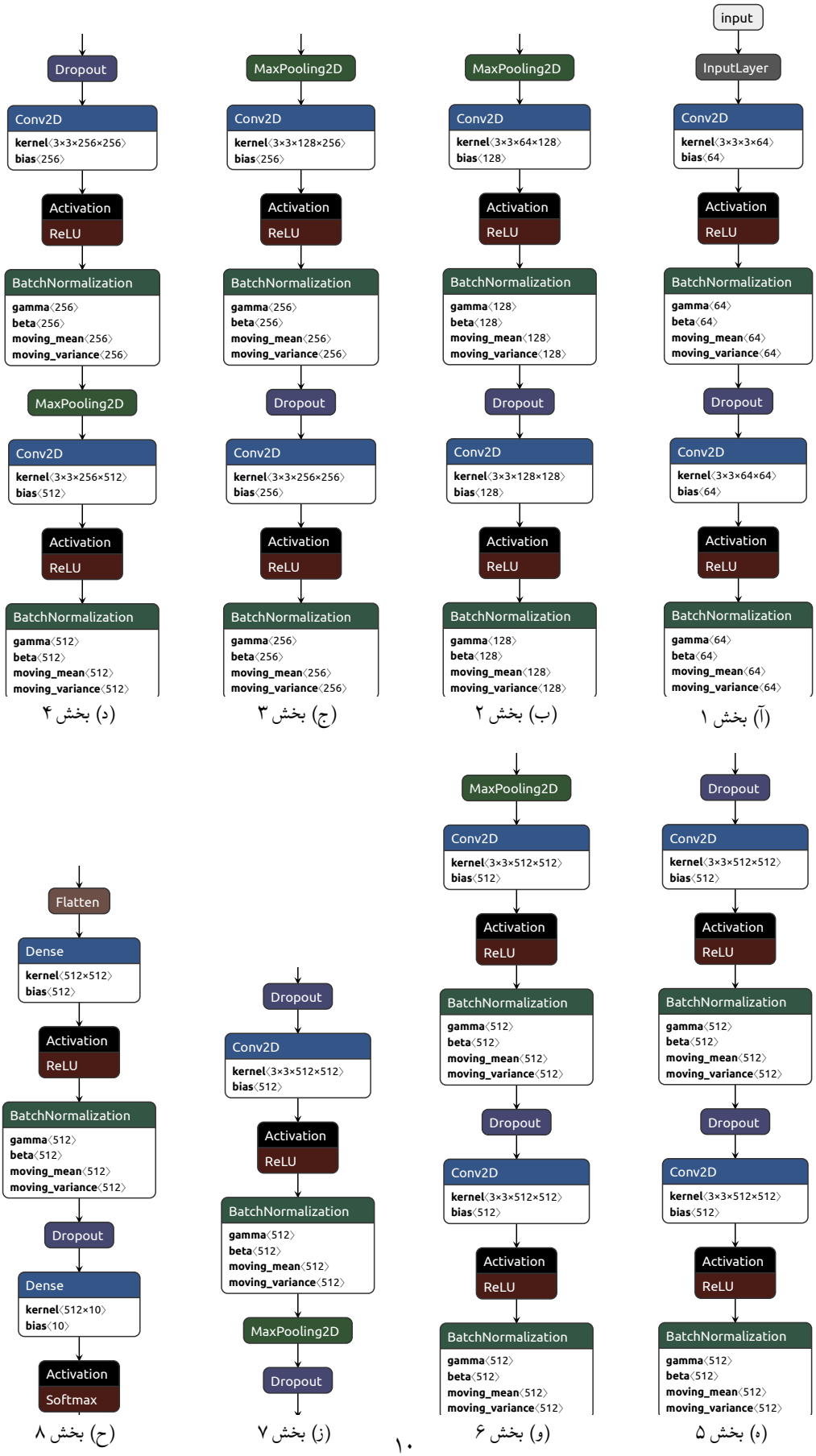
[14] Overfit
[15] Dense (Fully-Connected)

۲.۳ معماری مدل مجموعه داده CIFAR10

مجموعه داده CIFAR10[16] شامل تصاویری با ابعاد $(32, 32, 3)$ از نوع RGB می‌باشد؛ به طوری که هر تصویر متعلق به یکی از دسته‌های «هواپیما، اتومبیل، پرنده، گربه، گوزن، سگ، قورباغه، اسب، کشتی و کامیون» است. برای دسته‌بندی این تصاویر، ما از شبکه‌ای بسیار عمیق متشکل از لایه‌های پیچشی، Max Pooling ، Dropout ، Flatten و Batch Normalization استفاده می‌کنیم که به ترتیب برای انتخاب مهم‌ترین ویژگی(ها)، جلوگیری از بیش‌برازش، تبدیل تنسور به بردار و نرمال‌سازی هر Batch از تصاویر به کار می‌روند. لازم به ذکر است در کلیه استفاده‌های ما از لایه پیچشی (Conv2D) ، از ReLU به عنوان تابع فعال‌ساز بهره گرفته شده است.

فرض کنید یک Batch جدید به شبکه وارد شده است. ابتدا از هر نمونه استخراج ویژگی توسط لایه پیچشی صورت می‌گیرد. سپس کل Batch نرمال می‌شود. آنگاه یک لایه Dropout روی آن اثر می‌کند. فرایندی که تا به اینجای کار توضیح داده شد را از این پس «استخراج ویژگی نوع ۱» بنامید. پس از استخراج ویژگی نوع ۱، یک بار دیگر لایه پیچشی روی داده‌ها اثر می‌کند و دوباره روی Batch نرمال‌سازی صورت می‌گیرد. این بار پس از نرمال‌سازی، یک لایه Max Pooling مهم‌ترین ویژگی را از هر نمونه استخراج می‌کند. سپس یک لایه پیچشی دیگر روی داده‌ها اثر کرده، دوباره کل Batch نرمال می‌شود و سپس لایه Dropout بخشی از داده‌ها را به صورت تصادفی به صفر تبدیل می‌کند و مانع تمرکز شبکه روی یک ویژگی خاص می‌شود. این فرایند را نیز «استخراج ویژگی نوع ۲» بنامید. استخراج ویژگی نوع ۲ بار دیگر روی داده‌ها اعمال می‌شود. سپس استخراج ویژگی نوع ۱ و پس از آن دوباره نوع ۲ اعمال می‌گردد. عملیات استخراج ویژگی نوع ۱ و ۲ به ترتیب دو بار دیگر روی داده‌ها پیاده می‌شود و پس از آن، ویژگی‌های بدست‌آمده، با استفاده از Flatten به یک بردار تبدیل می‌گردند. از این بردار توسط یک لایه تماما متصل حاوی تابع فعال‌سازی ReLU ویژگی‌های جدیدی بدست می‌آید و دوباره کل Batch نرمال می‌شود. سپس یک لایه Dropout دیگر به کار گرفته می‌شود و نهایتا یک لایه تماما متصل با تابع فعال‌سازی Softmax ویژگی‌ها را به یک فضای احتمال نگاشت می‌کند. خروجی شبکه مانند قبل، یک بردار است که هر مولفه آن، احتمال تعلق نمونه موردنظر به یک دسته خاص را نشان می‌دهد. شکل ۵ معماری مدل به کار گرفته‌شده برای دسته‌بندی تصاویر مجموعه داده CIFAR10 را نشان می‌دهد.

[16] https://www.cs.toronto.edu/ kriz/cifar.html



(آ) بخش ۱ (ب) بخش ۲ (ج) بخش ۳ (د) بخش ۴

(ه) بخش ۵ (و) بخش ۶ (ز) بخش ۷ (ح) بخش ۸

شکل ۵: معماری مدل به کار رفته برای دسته‌بندی مجموعه داده cifar10

۴ دقت بر روی داده آموزشی و تست

پس از آموزش مدل‌هایی که معماری آن‌ها در اشکال ۴ و ۵ آمده است، دقت هر یک از مدل‌ها روی دو مجموعه داده MNIST و CIFAR10 محاسبه شد. جدول ۱ دقت مدل‌های مذکور در بخش ۳ روی داده‌های آموزشی و تست را نشان می‌دهد.

جدول ۱: دقت مدل‌های آموزش‌دیده روی دادگان

مجموعه داده	دقت روی دادگان آموزشی	دقت روی دادگان تست
MNIST	98.9%	99.3%
CIFAR10	98.5%	93.5%

۵ شیوه پیاده‌سازی هر روش

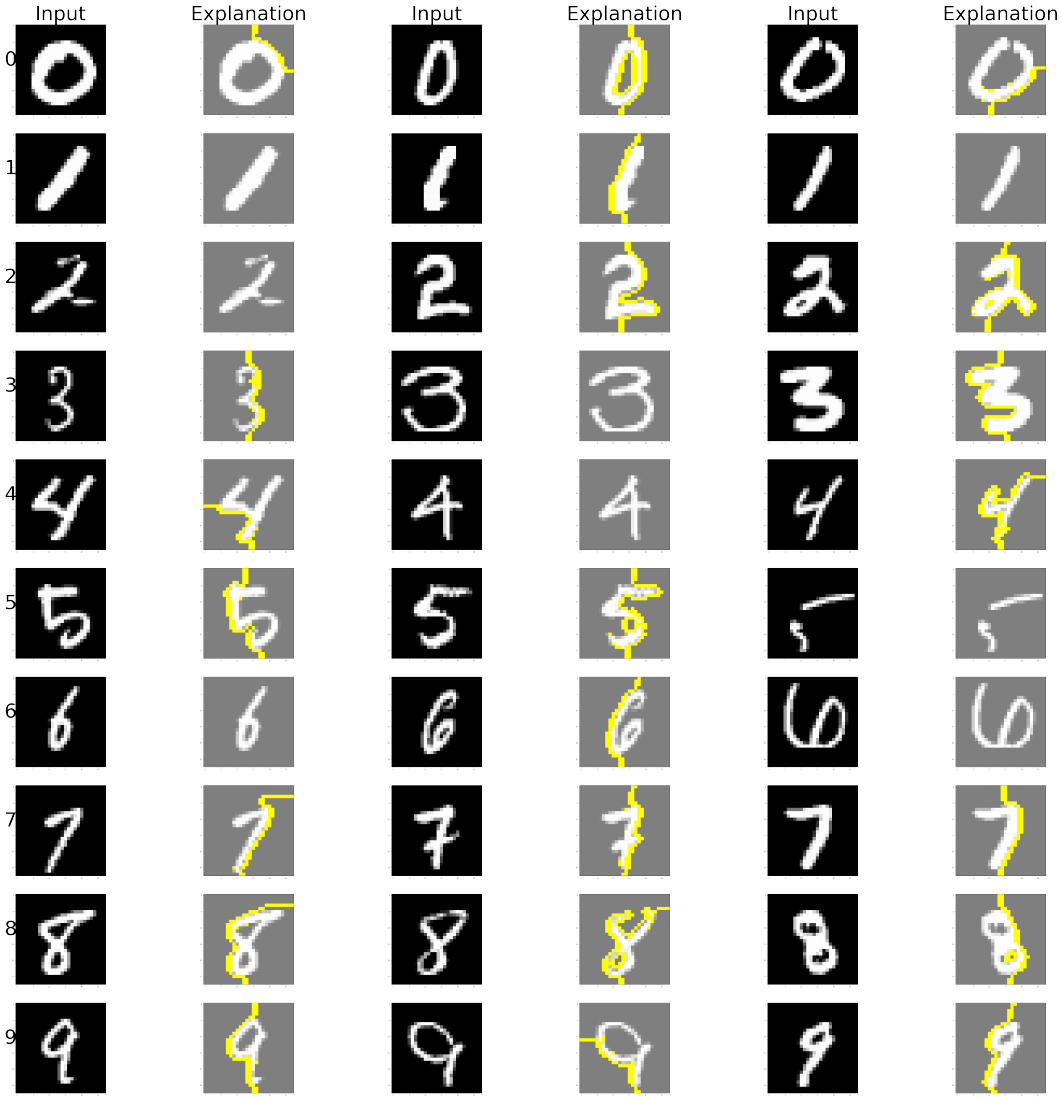
در این پروژه کلیه پیاده‌سازی‌ها با استفاده از زبان برنامه‌نویسی پایتون انجام شده است. مشخصات کلیه پکیج‌های به کار رفته به شرح زیر است:

- برای پیاده‌سازی روش تفسیرپذیری LIME: پکیج lime[۱۷]
- برای پیاده‌سازی روش تفسیرپذیری Kernel SHAP: پکیج shap[۱۸]
- برای پیاده‌سازی روش تفسیرپذیری Layer-wise Relevance Propagation روی مجموعه داده MNIST: پکیج iNNvestigate[۱۹]
- برای پیاده‌سازی روش تفسیرپذیری Layer-wise Relevance Propagation روی مجموعه داده CIFAR10: پکیج deepexplain[۲۰]

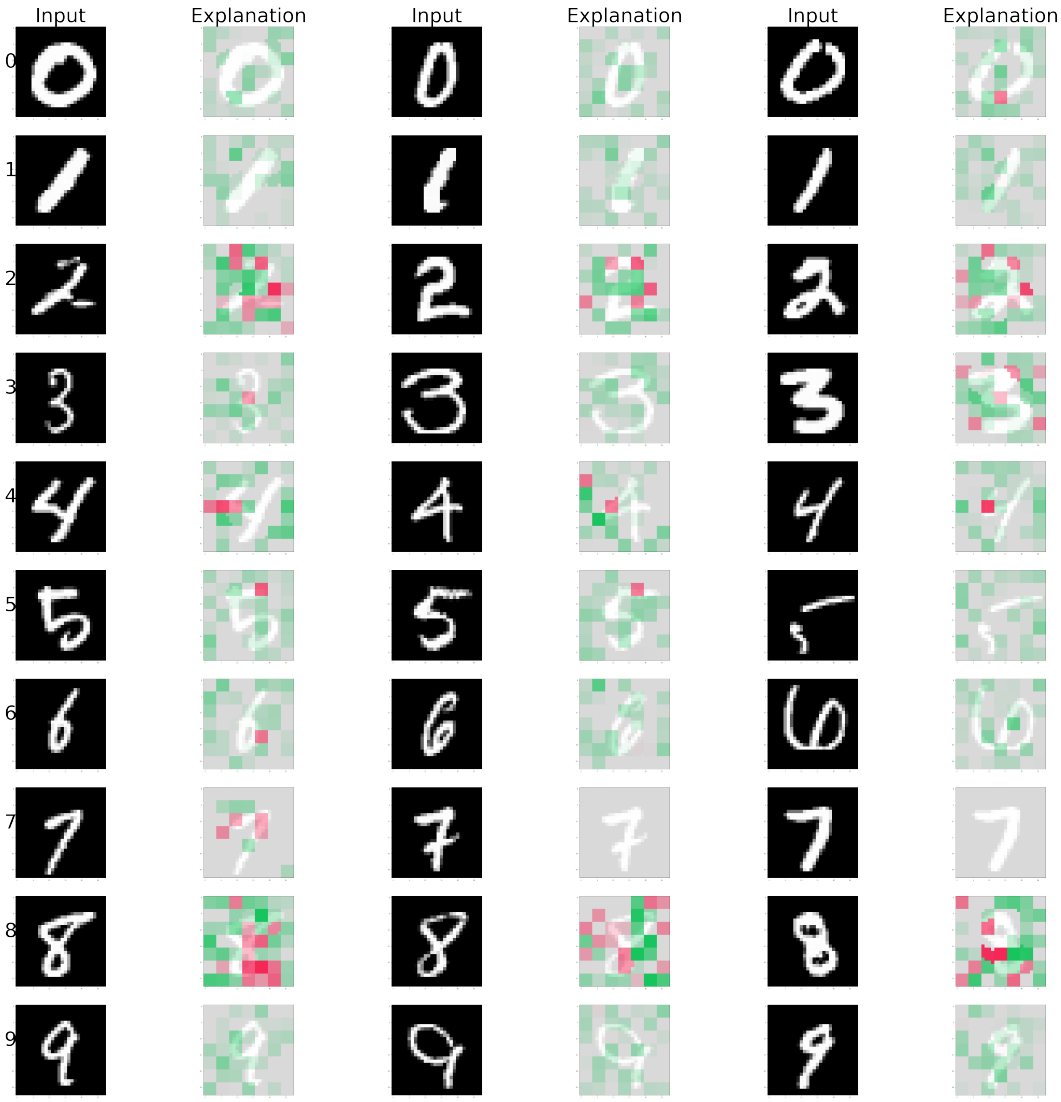
۶ مقایسه با استفاده از مصورسازی

برای هر یک از دو مجموعه داده MNIST و CIFAR10 که شامل ۱۰ دسته از تصاویر هستند، از هر دسته، سه تصویر به صورت تصادفی انتخاب نمودیم. با این روند، از هر یک از این دو مجموعه داده، ۳۰ تصویر انتخاب شد. برای هر یک از این تصاویر، نتایج تفسیرپذیری بر اساس روش‌های LIME ، Kernel SHAP و LRP را مصورسازی نمودیم. اشکال ۶ ، ۷ و ۸ خروجی این مصورسازی برای دادگان MNIST نشان می‌دهند و اشکال ۹ ، ۱۰ و ۱۱ مربوط به همین مصورسازی‌ها برای دادگان CIFAR10 هستند.

[17] https://github.com/marcotcr/lime/
[18] https://github.com/slundberg/shap
[19] https://github.com/albermax/innvestigate
[20] https://github.com/marcoancona/DeepExplain



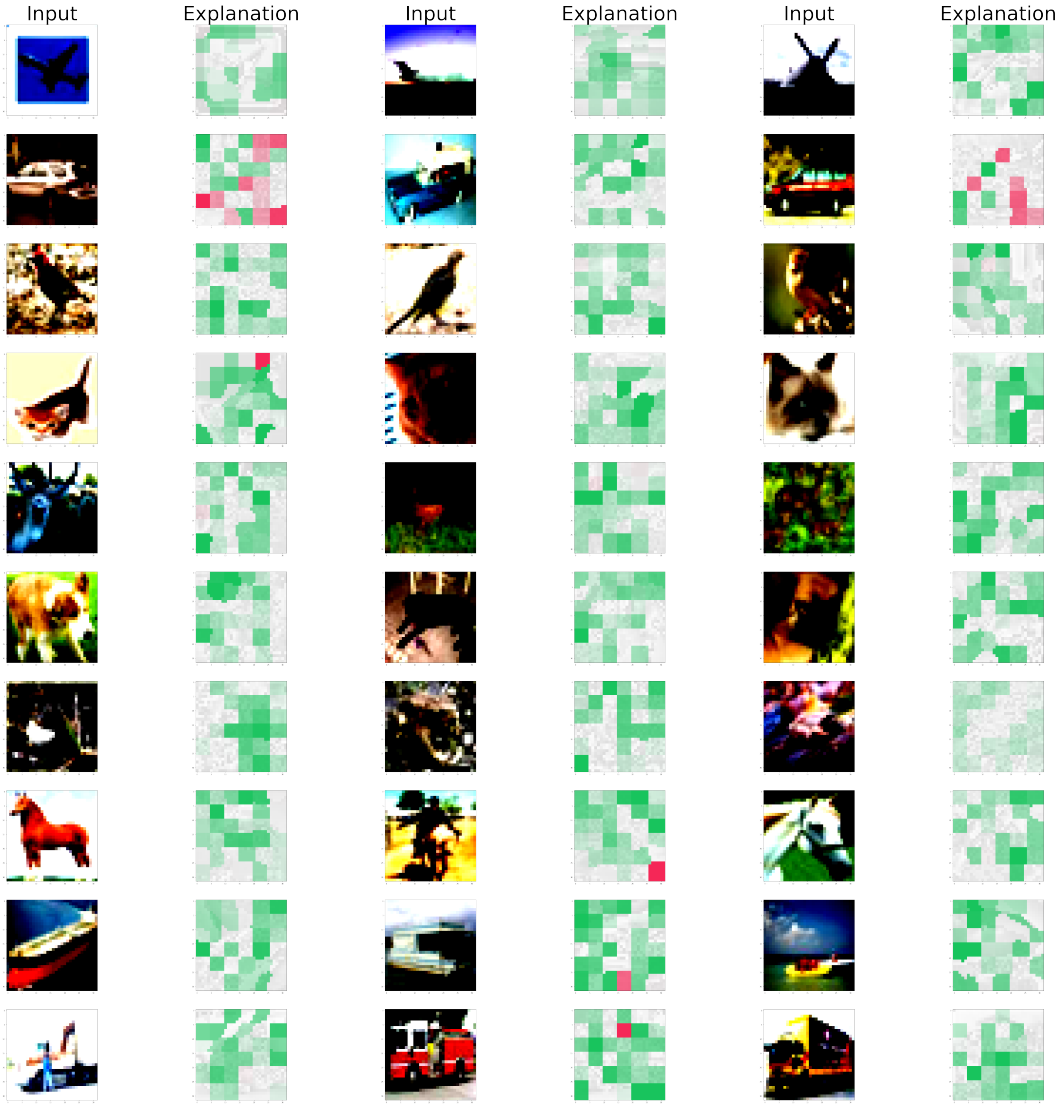
شکل ۶: مصورسازی روش LIME روی مجموعه داده MNIST



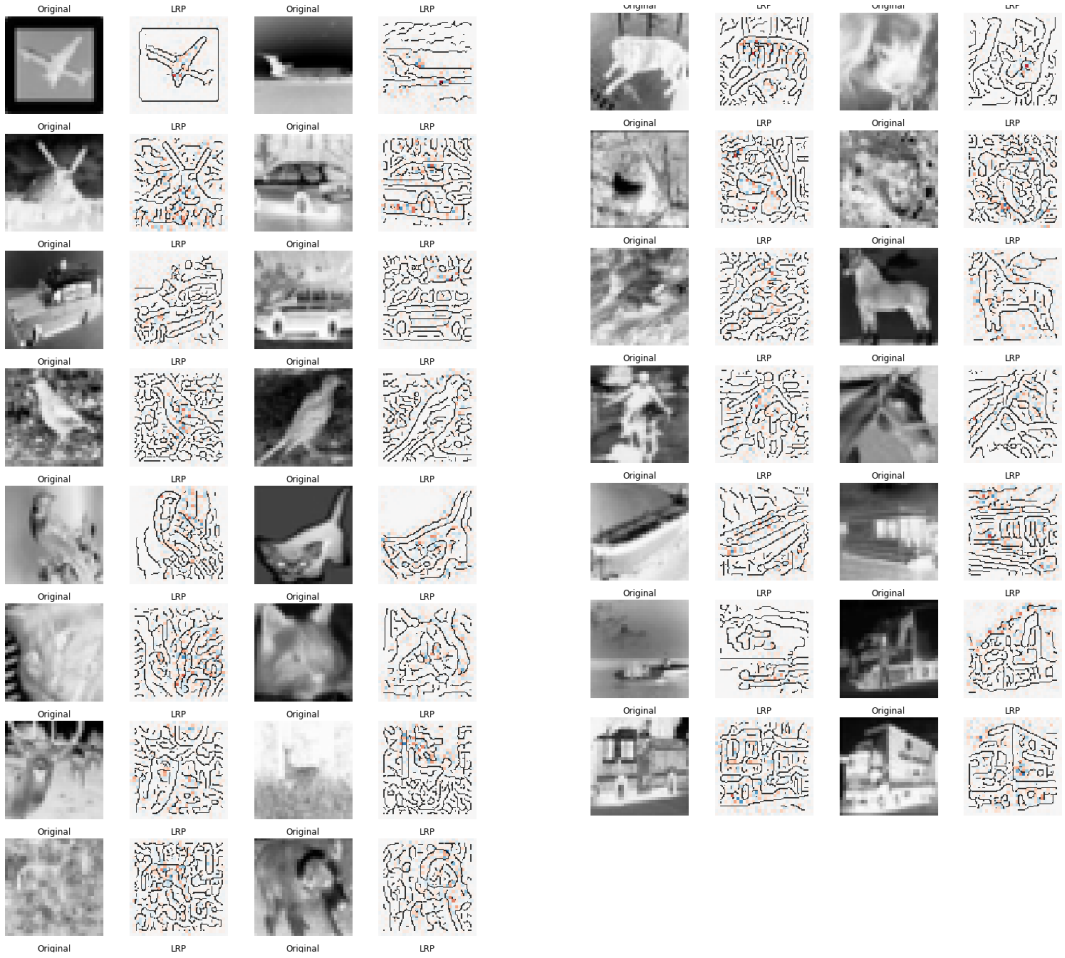
شکل ۷: مصورسازی روش Kernel SHAP روی مجموعه داده MNIST

شکل ۸: مصورسازی روش LRP روی مجموعه داده MNIST

شکل ۹: مصورسازی روش LIME روی مجموعه داده CIFAR10



شکل ۱۰: مصورسازی روش Kernel SHAP روی مجموعه داده CIFAR10



شکل ۱۱: مصورسازی روش LRP روی مجموعه داده CIFAR10

دقت کنید که در شکل ۶ پیکسل‌های خاکستری در واقع بخشی هستند که از نظر روش LIME غیر مهم شناخته شده‌اند و پیکسل‌های زرد رنگ نشان‌دهنده تمایز میان بخش‌هایی هستند که کمک به پیش‌بینی صحیح می‌کنند و یا مدل را از آن دور می‌کنند. همانطور که مشاهده می‌شود، روش LIME به خوبی پیکسل‌های مهم تصویر را شناسایی کرده و پس زمینه اعداد را کاملا کنار گذاشته است. شکل ۹ نتایج همین روش برای مجموعه داده CIFAR10 را نشان می‌دهد. این بار این روش در بسیاری از موارد، کل تصویر را به عنوان قسمت مهم در نظر گرفته و به ندرت بخش‌هایی از آن را خاکستری کرده است. این بدان معناست که از دید این روش، مدل ما در واقع تقریبا کل تصویر را می‌بیند. البته لازم به ذکر است که در برخی موارد، روش LIME توانسته است مرز اشیای درون تصویر را به خوبی تشخیص دهد.

در شکل ۷ نتایج مربوط به اعمال روش Kernel SHAP روی مجموعه داده MNIST مشاهده می‌شود. در این شکل، قسمت‌های خاکستری، بخش‌هایی از تصویر هستند که طبق دیدگاه روش KSHAP، مدل از آن‌ها برای پیش‌بینی استفاده خاصی نمی‌کند. در مقابل قسمت‌های سبز رنگ، بخش‌هایی هستند که در پیش‌بینی صحیحی مدل نقش مثبتی دارند و قسمت‌های قرمز رنگ، مانند ادله‌ای برای نفی آن پیش‌بینی توسط مدل شناخته می‌شوند. همانطور که در این شکل دیده می‌شود، بر اساس این روش هم مدل تا حد خوبی قسمت‌های غیر مهم را شناسایی کرده اما در بسیاری از موارد قادر به تشخیص قسمت‌های مهم نمی‌باشد. مثلا اگر به دسته مربوط به عدد ۵ توجه کنید، اکثرا قسمت‌های پس‌زمینه با رنگ سبز نشان داده شده که حاکی از اهمیت مدل به این قسمت‌ها است. در دسته مربوط به عدد ۷، دو مورد از تصاویر به گونه‌ای مصورسازی‌شده که نشان می‌دهد بر اساس روش KSHAP، مدل اصلا توجهی به قسمت خاصی از این تصاویر ندارد. این نتیجه که از این روش تفسیرپذیری بدست آمده، بسیار عجیب است و می‌توان آن را نشانه‌ای از تفسیر اشتباه مدل توسط این روش دانست. زیرا مدل دارای دقت بالایی روی همین تصاویر است و همه را به صورت صحیح دسته‌بندی می‌کند. با وجود این دسته‌بندی صحیح، غیر معقول است که به هیچ قسمتی از این دو تصویر مربوط به عدد ۷، توجه نکرده باشد.

در مقابل، طبق شکل ۸ که نتایج روش تفسیرپذیری Layer-wise Relevance Propagation را نشان می‌دهد، مدلی که معماری آن در شکل ۴ آمده، به خوبی قادر به جداسازی پس‌زمینه و تشخیص اعداد می‌باشد. همچنین، نتایج مربوط به اعمال همین روش روی مجموعه داده CIFAR10 که در شکل ۱۱ آمده نشان می‌دهند که مدل به خوبی لبه‌های اشیاء را در تصاویر شناسایی می‌کند. نتایج این روش با دقت مدل روی دادگان آموزشی و آزمایشی سازگاری بیشتری دارد.

لذا در مجموع به نظر می‌رسد طبق مقایسه با استفاده از مصورسازی، بهترین روش LRP است. پس از آن، روش LIME نتایج بهتری را نسبت به KSHAP ارائه می‌کند. دلیل برتری روش LRP نسبت به دو روش دیگر می‌تواند مربوط به این باشد که در این روش، علاوه بر ورودی و خروجی، روند رسیدن به خروجی از ورودی نیز در نظر گرفته می‌شود. در واقع اصلا این روش مخصوص شبکه‌های عصبی عمیق طراحی شده است و بر مبنای ایده‌ی backpropagation عمل می‌کند؛ در حالی که دو روش دیگر کلی‌تر هستند و برای هر ابزار دسته‌بندی قابل استفاده می‌باشند.

همچنین در هر دو روش LIME و KSHAP، از مفهوم سوپرپیکسل (مجموعه‌ای از پیکسل‌ها) برای تفسیر استفاده می‌شود. در واقع در این دو روش، ابتدا تصویر به تعداد مشخصی سوپرپیکسل تقسیم می‌شود و سپس اهمیت هر سوپر پیکسل در تغییر خروجی محاسبه می‌گردد. برای درک یکی از مشکلات احتمالی این روش، به این نکته توجه کنید که ممکن است یک سوپرپیکسل، دارای پیکسل‌های مهم و غیر مهم به صورت همزمان باشد. این امر مخصوصا در لبه‌های اشیاء بسیار محتمل است. مثلا در دادگان MNIST سوپر پیکسلی را در نظر بگیرید که شامل لبه‌ی یک عدد و بخشی از پیکسل‌های مشکی پس‌زمینه باشد. تغییراتی که در چنین سوپرپیکسلی پدید می‌آید، می‌تواند منجر به ایجاد یک الگوی جدید در تصویر شود و تفسیر را با مشکل مواجه کند.

همچنین برای درک عملکرد ضعیف‌تر روش KSHAP نسبت به LIME می‌توان به این نکته توجه نمود که در روش LIME سعی می‌شود نمونه‌های ساختگی تغییریافته، در همسایگی نمونه اولیه قرار گیرند و توسط یک معیار همسایگی، هر چه نمونه ساختگی به نمونه‌های واقعی نزدیک‌تر باشد، وزن بیشتری می‌گیرد. در حالی که در روش KSHAP همه ویژگی‌ها یک دور از داده‌ها حذف می‌شوند و در این فرایند، نزدیکی و دوری به توزیع اولیه نقشی ایفا نمی‌کند. همین امر ممکن است منجر به تغییر شگرفی در توزیع نمونه‌های ساختگی نسبت به توزیع اولیه داده‌ها شده و باعث این عملکرد ضعیف در روش KSHAP باشد.

۷ مقایسه با استفاده از مدل

در این مرحله، برای هر یک از روش‌های تفسیرپذیری و هر یک از مجموعه داده‌های MNIST و CIFAR10 یک مجموعه داده آزمایشی جدید ساخته شد. نحوه ساختن هر داده آزمایشی جدید، به شرح زیر است:

۱. ابتدا تعدادی از داده‌های آزمایشی انتخاب شد. (در این پروژه به علت زمان بالای اجرا، ۱۰۰۰ نمونه انتخاب گردیده است)

۲. برای داده‌های منتخب، مهم‌ترین پیکسل‌های هر تصویر به کمک روش‌های تفسیرپذیری محاسبه شد.

۳. برای هر نمونه، ۲۰ درصد از مهم‌ترین پیکسل‌های تصویر بدون تغییر رها شده و باقی پیکسل‌ها با صفر جایگزین شد.

پس از انجام موارد فوق، مجموعه داده آزمایشی جدید به مدل‌ها داده شد و دقت آن‌ها روی این داده‌های ساختگی محاسبه گردید. جدول ۲ نتایج این مرحله را نشان می‌دهد.

جدول ۲: میزان افت مدل‌ها پس از تخریب ۸۰ درصد پیکسل‌های غیرمهم تصویر و نگهداری تنها ۲۰ درصد از مهم‌ترین پیکسل‌ها. به دلیل زمان اجرای زیاد، تنها هزار نمونه از مجموعه آزمایشی در نظر گرفته شده است.

مجموعه داده	روش تفسیرپذیری	دقت روی دادگان آزمایشی جدید
MNIST	LIME	41.6%
MNIST	KSHAP	23.2%
MNIST	LRP	14.5%
CIFAR10	LIME	19.8%
CIFAR10	KSHAP	16.2%
CIFAR10	LRP	22.9%

طبق بخش ۶ انتظار ما این بود که روش LRP از دو روش دیگر بهتر عمل کند. اگر این امر درست باشد، انتظار داریم با حذف پیکسل‌هایی از تصویر که بر اساس روش LRP مهم هستند، افت دقت بیشتری را نسبت به دو روش LIME و KSHAP مشاهده کنیم. این امر درباره مجموعه داده MNIST صحت دارد و همانطور که در جدول ۲ مشاهده می‌شود، کمترین دقت

پس از تغییر دادگان آزمایشی، برابر ۱۴.۵ است که مربوط به روش LRP می‌باشد. طبق نتایج بخش ۶ انتظار داشتیم که روش LIME بهتر از KSHAP عمل کرده باشد. لذا در این بخش انتظار افت دقت بیشتری را برای LIME نسبت به KSHAP داشتیم که این انتظار ما در هیچ کدام از مجموعه داده‌های MNIST و CIFAR10 صدق نمی‌کند. این امر می‌تواند به این دلیل باشد که با تغییراتی که روش‌های تفسیرپذیری مذکور در داده‌های آزمایشی پدید آورده‌اند، توزیع آن‌ها بیش از حد تغییر کرده و لذا آنچه گمان می‌رود که درباره مدل اصلی آموخته شده، ممکن است اشتباه بوده باشد. البته تفاوت دقت این دو روش در مجموعه داده CIFAR10 تنها ۳ درصد است. در بخش ۶ نیز این دو روش نسبتا عملکرد مشابهی داشتند و تفاوت میان آن‌ها بارز نبود.

در مجموع، طبق جدول ۲ می‌توانیم نتیجه بگیریم که میان این سه روش تفسیرپذیری، روش LIME احتمالا مناسب مجموعه داده MNIST نمی‌باشد؛ زیرا هم بر اساس مصورسازی ضعیف عمل کرده و هم بر اساس نتایج بخش ۷ نتوانسته به خوبی پیکسل‌های مهم را شناسایی کند (توجه کنید که اگر جز این بود، باید شاهد افت دقت بیشتری می‌بودیم). نتایجی که از این بخش بدست آمده، نشان می‌دهد که از میان دو روش KSHAP و LRP نمی‌توان به صورت واضح یکی را بر دیگری ترجیح داد.

۸ تفسیر در شبکه‌های بیزین

بیشتر شبکه‌های عصبی که با آن‌ها سر و کار داریم، در حوزه مسائل دسته‌بندی [۲۱] یک ایراد اساسی دارند. این شبکه‌ها صرف نظر از اینکه ورودی ممکن است چقدر بی‌ربط به داده‌های آموزشی باشد، یک خروجی را به ازای آن تحویل می‌دهد. مثلا اگر یک شبکه به منظور تشخیص اخبار سیاسی و اقتصادی آموزش بدهیم و به شبکه یک خبر علمی را به عنوان ورودی بدهیم که هیچ ارتباطی به سیاست یا اقتصاد ندارد، خروجی در هر صورت «سیاسی» یا «اقتصادی» خواهد بود. اما در این شرایط به شبکه‌ای نیازمندیم که بتواند این عدم ارتباط را تشخیص داده و از خروجی‌دادن امتناع کند. در واقع ما باید معیاری را در شبکه تعبیه کنیم که مقدار اطمینان [۲۲] و قطعیت مربوط به هر پیش‌بینی را نیز در نظر بگیرد.

برای حل این مشکل، ما در این بخش به نگاه Bayesian رجوع می‌کنیم. در این نگاه، به هر متغیر در شبکه عصبی به چشم یک متغیر تصادفی می‌نگریم و به این متغیر تصادفی یک توزیع پیشین نسبت می‌دهیم. لذا بر خلاف شبکه‌های عصبی معمول، هر بار که به یک متغیر مراجعه کنیم، مقدار متفاوتی خواهد داشت. این امر باعث می‌شود برای یک ورودی که به شبکه داده می‌شود، بتوانیم چندین خروجی بدست آوریم. برای هر خروجی، یک مقدار قطعیت نیز محاسبه می‌شود.

برای پیاده‌سازی این شبکه Bayesian، از پکیج Pyro استفاده می‌کنیم که به ما امکان برنامه‌نویسی احتمالاتی روی فریم‌ورک آموزش شبکه‌های عصبی Pytorch را می‌دهد. لازم به ذکر است شبکه عصبی که در اینجا از آن استفاده می‌کنیم، شبکه‌ای تماما متصل و شامل تنها یک لایه مخفی با ۱۰۲۴ نورون است.

طبق قضیه بیز داریم:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{۱۶}$$

در مسئله ما، در رابطه ۱۶، جمله $P(A)$ احتمال پیشینی است که به وزن‌ها و بایاس‌های شبکه نسبت داده می‌شود و خود شبکه عصبی جمله‌ی $P(B|A)$ را نمایندگی می‌کند. همچنین، B در واقع همان داده آموزشی است که شامل تصاویر و برچسب متناظر آن‌ها می‌شود. در واقع $P(B)$ احتمال مشاهده داده‌ها تحت همه مقادیر ممکن پارامترها و احتمال وقوع آن‌ها می‌باشد. به طور دقیق‌تر داریم:

$$P(B) = \sum_j P(B|A_j)P(A_j) \tag{۱۷}$$

در آموزش این شبکه عصبی، ما به دنبال یافتن توزیع بروزرسانی‌شده‌ی وزن‌ها و بایاس‌های شبکه - همان $P(A|B)$ هستیم که احتمال پسین نامیده می‌شود. در ابتدا توزیع پارامترها (احتمال پیشین) را به صورت رندوم نسبت می‌دهیم و چون شبکه هنوز آموزش داده نشده، این مقداردهی نمی‌تواند دقت خوبی در دسته‌بندی داشته باشد. اما با بروزرسانی شبکه (مثلا میانگین و واریانس توزیع)، به توزیع دقیق‌تری می‌رسیم که قابلیت تعمیم‌پذیری بسیار خوبی را خواهد داشت. به طور خاص ما در اینجا توزیع پیشین مربوط به وزن‌های شبکه را از نوع نرمال در نظر می‌گیریم. آموزش این شبکه شامل یادگیری توزیع‌های احتمال مربوط به وزن‌ها و بایاس‌ها است؛ به گونه‌ای که likelihood انتساب یک احتمال زیاد به یک زوج مرتب (عکس و برچسب) واقعی بیشینه شود. از طرفی باید توجه کرد که $P(B)$ به طور دقیق قابل محاسبه نیست و به همین دلیل باید برای یافتن احتمال پسین

[21] Classification
[22] Confidence

نیز معیاری از نزدیکی هر توزیع دیگر به توزیع واقعی پیدا کنیم. به طور دقیق‌تر، ما Evidence Lower Bound (ELBO) را کمینه می‌کنیم که معادل بیشینه‌کردن likelihood مذکور است. به این عملیات اصطلاحا Variational Inference گفته می‌شود.

در نهایت نحوه پیش‌بینی با استفاده از شبکه بیزی آموزش‌دیده بدین شکل است:

- ابتدا برای هر ورودی داده‌شده، صد بار شبکه فراخوانی می‌شود و صد خروجی مختلف بدست می‌آید. هر خروجی به شکل یک بردار با ۱۰ مولفه است که متناظر هر یک از اعداد ۰ تا ۹ می‌باشند. پس ما ۱۰۰ بردار ۱۰-تایی خواهیم داشت.
- برای هر یک از ابعاد این بردار (هر کدام از احتمالات) میانه مقادیر را در نظر می‌گیریم.
- اگر میانه این مولفه، بزرگتر از 0.2 باشد، گوییم ورودی شبکه، به این کلاس تعلق دارد. توجه کنید با این روند ممکن است یک ورودی جزء هیچ دسته‌ای قرار نگیرد و یا به چند دسته تعلق گیرد.

پس از آموزش شبکه روی دادگان MNIST، مشابه بخش ۷، به ترتیب، ۱۰، ۲۰، ۳۰، ۴۰ و ۵۰ درصد از مهم‌ترین پیکسل‌های تصاویر تست را - که توسط روش KSHAP محاسبه شد - به شبکه دادیم و دقت آن روی این دادگان ساختگی جدید را محاسبه نمودیم. جدول ۳ نتایج این مرحله نشان می‌دهد.

جدول ۳: میزان تغییر دقت مدل و قطعیت آن در پیش‌بینی، پس از تخریب بخشی از پیکسل‌ها و حفظ درصدی از مهم‌ترین اطلاعات تصویر. به دلیل زمان بالای اجرا، تنها ۱۰۰ نمونه برای انجام آزمایش انتخاب شده‌اند

درصد پیکسل‌های مهم حفظ‌شده	میزان دقت مدل پس از تخریب تصاویر	کمینه قطعیت روی داده‌ها	میانگین قطعیت روی داده‌ها	بیشینه قطعیت روی داده‌ها
10	5%	0.50	0.95	1.0
20	1%	0.58	0.97	1.0
30	3%	0.57	0.99	1.0
40	5%	0.51	0.98	1.0
50	11%	0.51	0.97	1.0

همانطور که مشاهده می‌شود، هرچه درصد بیشتری از پیکسل‌های مهم حفظ شوند، دقت بیشتری را از مدل خواهیم گرفت. البته این نتیجه در مورد تفاوت میان ۱۰ و ۲۰ درصد از پیکسل‌ها صدق نمی‌کند. اما روند کلی به این ترتیب است. مشاهده می‌کنیم که با تغییر درصد پیکسل‌های مهم. حفظ‌شده از ۴۰ به ۵۰ درصد، از لحاظ میزان دقت، ۶ درصد بهبود خواهیم داشت. اما از نظر قطعیت مدل روی داده‌ها، تقریبا تفاوت چشم‌گیری حاصل نمی‌شود و کمینه و بیشینه قطعیت مدل یکسان خواهد بود. همچنین به طور کلی باید توجه داشت که در همه این حالات، حداقل ۵۰ درصد پیکسل‌های مهم تخریب شده‌اند و این تخریب منجر به ایجاد افت چشم‌گیری در عملکرد مدل شده است. در واقع مدل پیش از این دارای دقت ۹۸ درصد بوده و پس از اعمال این تغییرات، در بهترین حالت می‌توان مشاهده کرد که ۸۷ درصد افت دقت داشته است.

همچنین توجه کنید که مدل اولیه در اغلب موارد قطعیتی بین ۰.۹ تا ۱ داشته است و این قطعیت پس از تخریب پیکسل‌ها تا ۰.۵ هم کاهش یافته است. همانطور که در جدول ۳ مشاهده می‌شود، بیشترین کاهش قطعیت مربوط به حالتی است که تنها ۱۰ درصد از مهم‌ترین پیکسل‌های تصاویر حفظ شوند. این نتیجه کاملا مورد انتظار ماست. اما پس از آن، در حالاتی که ۴۰ یا ۵۰ درصد از مهم‌ترین پیکسل‌ها حفظ بشوند، بیشترین کاهش قطعیت را مشاهده خواهیم کرد که نهایتا تا ۰.۵۱ هم می‌رسد. این در حالی است که به شکل منطقی می‌توان انتظار داشت که کاهش قطعیت برای حالات ۲۰ و ۳۰ درصد، بیش از ۴۰ و ۵۰ درصد باشد؛ زیرا در این حالات، پیکسل‌های مهم بیشتری تخریب شده‌اند.

۹ جمع‌بندی

ما سه روش تفسیرپذیری LIME، Kernel SHAP و Layer-wise Relevance Propagation را روی دو مجموعه داده MNIST و CIFAR10 بررسی نمودیم. مقایسه میان این روش‌ها به دو طریق انجام شد: مصورسازی و حذف برخی اطلاعات تصاویر. بر اساس نتایج مصورسازی، روش LRP نسبت به دو روش دیگر برتری قابل توجهی دارد. اما نتایج روش حذف اطلاعات از تصویر، لزوما این گزاره را تایید نمی‌کند.

علاوه بر این، یک شبکه عصبی با رویکرد بیزین نیز روی مجموعه داده MNIST آموزش داده شد. مانند قبل، روش حذف اطلاعات روی این شبکه نیز بررسی شد. بر اساس نتایج به نظر می‌رسد هرچه اطلاعات بیشتری حذف شود، میزان قطعیت مدل در پیش‌بینی‌ها کاهش می‌یابد. همچنین هرچه پیکسل‌های بیشتری از تصاویر حذف شود، عملکرد مدل به سمت عملکردی تصادفی گرایش پیدا می‌کند. این را می‌توان در نتایج مربوط به حذف ۵۰ درصد از پیکسل‌های مهم تصاویر برداشت کرد که پس

از آن، دقت مدل به ۱۱ درصد رسیده است (توجه کنید که دقت تصادفی ۱۰ درصد می‌باشد). در ادامه این پژوهش می‌توان روش‌های بهبود شبکه‌های عصبی با توجه به نتایج روش‌های تفسیرپذیری را بررسی نمود.

منابع

[1] Fan, F. L., Xiong, J., Li, M., & Wang, G. (2021). On interpretability of artificial neural networks: A survey. IEEE Transactions on Radiation and Plasma Medical Sciences.

[2] Hooker, S., Erhan, D., Kindermans, P., & Kim, B. (2019). A Benchmark for Interpretability Methods in Deep Neural Networks. NeurIPS.

[3] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

[4] Lundberg, S. M., & Lee, S. I. (2017, December). A unified approach to interpreting model predictions. In Proceedings of the 31st international conference on neural information processing systems (pp. 4768-4777).

[5] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7), e0130140.

[6] Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K. R. (2019). Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning, 193-209.