گزارش پروژه

کسری خوشجو_۹۸۱۰۰۴۲۹ رضا پیشکو_۹۸۱۰۰۳۶۷

۱ پیشگفتار

فضای نظریه اطلاعات در سال ۱۹۴۸ توسط شانون معرفی شد که یکی از موضوعات مهم آن انتقال اطلاعات است. به این معنی که اطلاعات از یک منبع به گونه ای فرستاده شود که گیرنده بتواند آن را بازیابی کند. از طرفی یادگیری ماشین را هم می‌توان به عنوان علم رمزگشایی کردن پارامتر های مدل حقیقی که نمونه های تصادفی از آن مدل تولید شده اند در نظر گرفت.به این ترتیب می‌توان چرایی اینکه این دو حیطه در طلاقی با یکدیگر هستند را دید.

در دو دهه اخیر مدل های یادگیری عمیق موفقیت قابل توجهی در حل مسائل یادگیری ماشین داشته اند. اما این موفقیت در عمل با پیشرفت در حوزه تئوری همراه نبوده و علت موفقیت مدل های یادگیری عمیق تا حدی زیادی همچنان ناشناخته باقی مانده است. به همین دلیل است که محققان تلاش می کنند به کمک ابزار های تئوری در حوزه نظریه اطلاعات به فهم بهتری از نحوه عملکرد Deep Neural Network ها دست پیدا کنند.

در بخش دوم به بیان تعاریف و قضایای مربوط به روش IB می پردازیم و ارتباط آن با مساله های یادگیری ماشین را بررسی می کنیم و روش تکرار شونده ای برای حل مساله IB در حالت گسسته را بیان می کنیم. در بخش سه نیز به بیان ایده هایی برای تحلیل رفتار شبکه های عصبی عمیق با استفاده از IB و ابزار های نظریه اطلاعاتی می پردازیم. در انتها در بخش چهار به یک ایده مشابه روش IB برای مطالعه مدل های یادگیری عمیق اشاره می کنیم.

۲ مقدمه

۱.۲ تعاریف

مدل سازی مساله یادگیری ماشین:

فرض کنیم $X \in \mathcal{X}$ و $Y \in \mathcal{Y}$ متغییر های تصادفی با تابع توزیع توام $p(x,y)$ باشند که $\mathcal{X}$ و $\mathcal{Y}$ به ترتیب فضا های ورودی و خروجی هستند. همچنین x و y برای نمایش نمونه از متغییر های تصادفی X,Y استفاده می‌شوند. همچنین معمولا فرض می کنیم که به توزیع توام دسترسی نداریم اما N مثال با توزیع i.i.d داریم که به فرم $\{(x^{(i)},y^{(i)})\}_{i=1}^{N}$ نمایش می‌دهیم.

تعریف: آنتروپی X به شرط Y به صورت $H(X|Y)$ نمایش داده شده و برابر است با $\mathbb{E}\left[-\log p(X,Y)\right]$.

تعریف : و همچنین اطلاعات متقابل X,Y به صورت $I(X;Y)$ نمایش داده شده و برابر است با $\mathbb{E}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right]$.

۲.۲ Relevant Information

یک مفهوم مشترک بین نظریه اطلاعات و آمار و یادگیری ماشین تعریف کردن و استخراج اطلاعات مفید درباره متغییر هدف از روی مشاهده ورودی ها است که یکی از راه های که می توان این اطلاعات را فرمول بندی کرد به وسیله مفهموم Sufficient Statistic است.

تعریف :

فرض کنید $Y \in \mathcal{Y}$ یک پارامتر ناشناخته باشد و $X \in \mathcal{X}$ یک متغییر تصادفی با تابع توزیع شرطی $p(x|y)$ باشد. تابع $f : \mathcal{X} \to \mathcal{S}$ را در نظر بگیرید. در اینصورت متغییر تصادفی $S = f(X)$ یک Sufficient Stastic برای Y است اگر و تنها اگر :

$$\forall x \in \mathcal{X}, y \in \mathcal{Y} : \mathbb{P}\left[X = x | Y = y, S = s\right] = \mathbb{P}\left[X = x | S = s\right]$$

به طور معادل دنباله $Y \to S \to X$ یک زنجیر مارکوفی است.
قضیه(۱): فرض کنید S یک تابع تصادفی از X باشد در اینصورت S یک SS برای Y است اگر و تنها اگر :

$$I(S;Y) = I(X;Y)$$

از آنجایی که S یک تابع تصادفی از X است و X نیز از توزیع شرطی نسبت به Y بدست آمده، $Y \to X \to S$ تشکیل یک زنجیر مارکوفی می‌دهد.از طرفی طبق شرط SS بودن، $Y \to S \to X$ نیز یک زنجیر مارکوفی می‌دهد. بنابراین طبق قاعده پردازش داده ها می توان دو نامساوی زیر را با توجه به زنجیر های مارکوفی ذکر شده نتیجه گرفت.

$$\left.\begin{array}{l} I(X;Y) \leq I(S;Y) \\ I(S;Y) \leq I(X;Y) \end{array}\right\} \Rightarrow I(X;Y) = I(S;Y)$$

مفهوم SS به این معنی است که همه اطلاعات مفید در مورد Y که در X هستند توسط پردازش انجام شده حفظ می‌شوند.
به وضوح می توان دید که یک مثال بدیهی از SS ها متغییر $S \triangleq X$ است که عملا بی فایده است چرا که کپی کردن ورودی را نمی توان به عنوان استخراج اطلاعات در نظر گرفت. برای جلوگیری از لحاظ کردن چنین جواب های نامطلوبی به تعریف دیگری نیاز داریم:

تعریف:
به یک SS مانند S مینیمال گفته می شود اگر :

$$\forall T \neq S, T \text{is an SS} : \exists g : S = g(T)$$

قضیه(۲):
فرض کنید X یک نمونه گرفته شد از یک تابع توزیع باشد که بر حسب متغییر تصادفی Y تعیین شده. به آماره S ، یک Minimal Sufficient Statistic برای Y گوییم اگر و تنها اگر پاسخی برای مساله بهینه سازی زیر باشد:

$$\min_{T \text{is an SS}} I(X;T)$$

که با استفاده از قضیه (۱) می‌توان شرط مساله بهینه سازی بالا را به صورت زیر در نظر گرفت:

$$\min_{T : I(T;Y) = I(X;Y)} I(X;T)$$

این قضیه ارتباط بین MSS و MI را نشان می‌دهد. به عبارتی یک MSS کمترین اطلاعات از X را حفظ می‌کند به طوری که اطلاعاتی از Y که در X وجود داشته از بین نرود. در واقع MSS بهترین فشرده سازی X است.
با توجه به تعریف، یک MSS تابعی از هر SS دیگری است و از طرفی می دانیم که خود SS هم تابعی از Y است. بنابراین ، $Y \to SS \to MSS$ تشکیل یک زنجیر مارکوفی می دهد و با توجه به پردازش داده ها، می‌توان نتیجه گرفت که

$$\forall SS : I(MSS;X) \leq I(SS;X)$$

که این درستی قضیه (۲) را نشان می دهد.
SS ها به نسبت محدود هستند به اینصورت که بعد آنها همواره به اندازه نمونه بستگی دارد مگر اینکه داده ها از یک خانواده نمایی تولید شده باشند. در نتیجه خوب است که مساله ریلکس شده تری را در نظر بگیریم که یک نمونه آن مساله Information Bottleneck است.

۳.۲ Information Bottleneck

در روش IB برای استخراج کردن اطلاعات مرتبط از یک متغییر تصادفی X از فضای $\mathcal{X}$ درباره Y به این صورت تعریف می شود:
فرض کنید که $P_{S|X}$ یک کرنل تبدیل از فضای $\mathcal{X}$ به فضای $\mathcal{S}$ باشد. سه تایی $Y \to X \to S$ با توجه به تابع توزیع چگالی $P_{X,Y,S} = P_{X,Y} \cdot P_{S|X}$ تشکیل یک زنجیر مارکوف می دهد.
حال به دنبال کرنل $P_{S|X}$ ای هستیم که اطلاعات Y را به خوبی حفظ کند یعنی $I(Y;S)$ زیاد باشد و به صورت همزمان X را حتی المکان فشرده کند به این معنا که $I(X;S)$ کم باشد. از طرفی با توجه به قاعده پردازش داده ها S نمی تواند اطلاعات بیشتری در مورد Y نسبت به X داشته باشد. و همین موضوع باعث بوجود آمدن یک "trade-off" بین فشرده کردن X و حفظ اطلاعات معنادار می‌شود که می توان آنرا با یک پارامتر کنترل کرد. در نهایت اگر مساله IB را به صورت زیر فرمولبندی کنیم:

$$\inf_{I(Y;S)\geq\alpha} I(X;S)$$

α عضو $\mathbf{R}_{\geq 0}$ پارامتری است که این "trade-off" را کنترل می کند.
در واقع به دنبال S هستیم که کمترین اطلاعات از X را داشته باشد در حالی که حداقل α بیت از اطلاعات مفید در مورد Y را منتقل می کند.
مد نظر داشته باشیم که مساله Information Bottleneck محدب نیست ولی می‌توان آنرا به یک مساله "rate-distortion" محدب تبدیل کرد: که با تعریف ضریب لاگرانژ β و در نظر گرفتن تابع زیر :

$$\mathcal{L}_\beta(P_{S|X}) \triangleq I(X;S) - \beta I(S;Y)$$

مساله IB تبدیل به مینیمم سازی $\mathcal{L}_\beta(P_{S|X})$ برای یک β ثابت روی همه $P_{S|X}$ ها می شود. دقت کنید که با توجه به زنجیر مارکوفی $Y \to X \to S$ و طبق قاعده پردازش داده ها داریم:

$$I(X;S) \leq I(S;Y) = ((1-\beta)+\beta)I(S;Y)$$

در نتیجه مینیمم مساله بالا از $(1-\beta)I(S;Y)$ بیشتر مساوی است و در حالت $\beta \leq 1$ مینیمم کردن $\mathcal{L}_\beta(P_{S|X})$ معادل با صفر کردن $I(S;Y)$ می شود که با صفر شدن $I(S;X)$ محقق می شود.در نتیجه جواب بهینه وقتی رخ می دهد که X, S از هم مستقل باشند.
در حالت کلی کوچک بودن β باعث فشرده سازی بیشتر و از دست رفتن اطلاعات بیشتری می شود؛در مقابل، بزرگ بودن β باعث فشرده سازی کمتر و حفظ اطلاعات بیشتر می شود. بنابراین با تغییر β می‌توانیم "trade-off" بین حفظ اطلاعات مفید و فشرده سازی را تنظیم کنیم.

در نتیجه روش IB تعبیر MSS را از دو منظر ریلکس می‌کند.اول اینکه ،MSS به عنوان یک تابع غیر تصادفی از X تعریف می‌شود اما جواب های مساله IB نگاشت های تصادفی هستند و می‌تواند به مقدار های کمتری برای $\mathcal{L}_\beta(P_{T|X})$ برسد. دومین تفاوت این است که روش IB می‌تواند با تنظیم مقدار اطلاعاتی که از Y در S نگه می‌دارد به تقریب هایی از MSS ها برسد.

برای هر $\beta \in [0, +\infty)$ نقطه بهینه $\mathcal{L}_\beta(P_{S|X})$ را می توان با یک سری معادله خود سازگار بیان کرد که در ادامه به آن می پردازم.

۴.۲ معادلات خود سازگار

جواب بهینه مسئله کمینه سازی $\mathcal{L}_\beta(P_{S|X})$ باید در شرط زیر صدق کند :

$$p(s|x) = \frac{p(s)}{Z(x,\beta)} \cdot exp\left(-\beta \sum_y p(y|x) \log \frac{p(y|x)}{p(y|s)}\right)$$

که مقدار $p(y|s)$ با توجه به قانون بیز و زنجیره مارکوفی $Y \to X \to S$ به صورت زیر قابل بیان است :

$$p(y|s) = \frac{1}{p(s)} \sum_x p(y|x) \cdot p(s|x) \cdot p(x)$$

با تعریف ضرایب لاگرانژ β به عنوان ثابت اطلاعات و $\lambda(x)$ برای نرمال کردن توزیع شرطی $p(s|x)$ برای هر x معادله بهینه سازی ما تبدیل می شود به :

$$\mathcal{L} = I(X;S) - \beta I(S;Y) - \sum_{s,x} \lambda(x) p(s|x)$$

که با محاسبه $\frac{\partial \mathcal{L}}{\partial p(s|x)}$ و تعریف $Z(x,\beta)$ به صورت زیر نهایتا به رابطه ذکر شده در بالا میرسیم.

$$Z(x,\beta) = \sum_{s} p(s) \cdot exp(\beta D_{KL}[p(y|x)||p(y|s)]))$$

۵.۲ الگوریتم تکرار شونده IB

معادلات خودسازگار زیر را در نظر بگیرید :

$$p(y|s) = \sum_{x \in X} p(y|x) p(x|s)$$
$$p(s) = \sum_{x} p(s|x) p(x)$$
$$p(s|x) = \frac{p(s)}{Z(x,\beta)} \cdot exp\left(-\beta \sum_{y} p(y|x) \log \frac{p(y|x)}{p(y|s)}\right)$$

این معدله ها همزمان در مینیمم تابع زیر محقق می شوند :

$$\mathcal{F}\left[p(s|x); p(s); p(y|s)\right] = - < logZ(x,\beta) >_{p(x)} = I(X,S) + \beta < D_{KL}[p(y|x)||p(y|s)] >_{p(x,s)}$$

که این مینیمم سازی به طور مستقل روی مجموعه های محدب توزیع نرمال های $\{p(s)\}, \{p(s|x)\}, \{p(y|s)\}$ انجام می شود. یعنی :

$$\min_{p(y|s)} \min_{p(s)} \min_{p(s|x)} \mathcal{F}\left[p(s|x); p(s); p(y|s)\right].$$

که این کمینه سازی توسط تکرارهای متناوب همگرا انجام می شود :

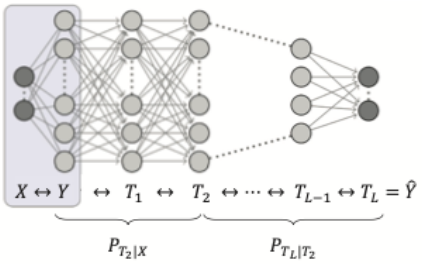
$$p_i(s|x) = \frac{p_i(s)}{Z_i(x,\beta)} \cdot exp\left(\beta d(x,s)\right)$$
$$p_{i+1}(s) = \sum_{x} p(x) p_i(s|x)$$
$$p_{i+1}(y|s) = \sum_{y} p(y|x) p_i(x|s)$$

که i نشان دهنده هر قدم است و $Z_i(x,\beta)$ هم در مرحله محاسبه می شود.

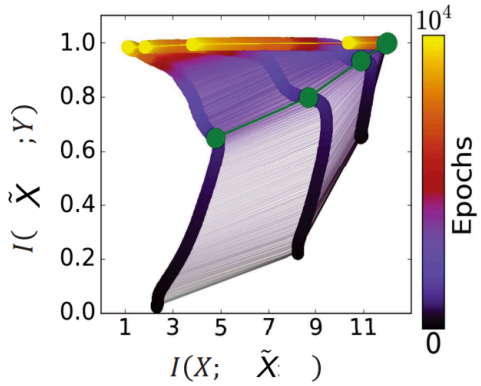
۳ Deep Neural Networks

در این بخش به کاربرد های روش IB در تحلیل عملکر شبکه های عصبی می‌پردازیم. طی سال های گذشته مدل های یادگیری عمیق توانایی زیادی برای حل مساله های یادگیری از خود نشان داده اند اما علت این توانایی چندان شناخته شده نیست.
یکی از ابزار های نظریه اطلاعاتی ای که برای تحلیل مدل های یادگیری عمیق مطرح شد روش IB است که توسط آقای Tishby و همکارانشان برای اولین بار مطرح شد، در ادامه به توضیح مدلسازی انجام شده در این مقاله و ادعا های مطرح شده می پردازیم.

Information Plane ۱.۳



یک مدل یادگیری عمیق را دنباله ای از متغییر های تصادفی در نظر می گیریم که هر متغییر نماینده خروجی هر لایه از این شبکه است. بدین ترتیب هر متغییر برحسب متغییر قبلی دنباله تحت یک تابع(تصادفی) محاسبه می شود. این بدین معناست که شبکه ما در حقیقت یک زنجیر مارکوفی است و خروجی آن تخمین Y با داشتن $X \sim P_{X|Y}$ است.



در این مدل تمایل داریم اطلاعات متقابل بین هر T_i و Y بیشینه باشد چرا که با داشتن این خاصیت می‌توان انتظار داشت که $H(\hat{Y}|Y)$ کاهش یابد و در نتیجه با احتمال بالایی $\hat{Y} = Y$. در مقاله آقای Tishby ابزار مهم دیگری که مطرح شد Information Plane است. در حقیقت نموداری است که در آن به ازا هر قدم از بهینه سازی یادگیری عمیق یک دنباله از نقطه ها در نظر می گیریم که هر نقطه نماینده یک لایه از شبکه عصبی یا متغییر متناظر آن، T_i، است و مختصات آن به ترتیب $I(X;T_i)$و $I(Y;T_i)$ است.

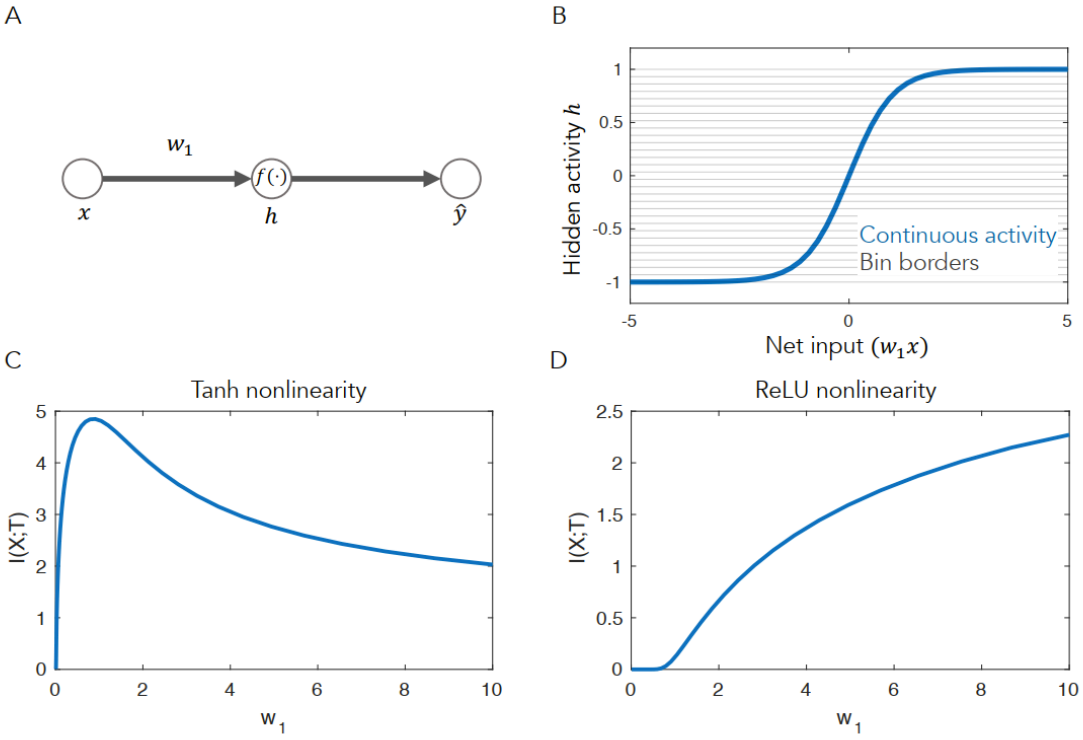
انتظار داریم در این مدل لایه ها در حال بهینه سازی تابع $\mathcal{L}_\beta(P_{T_i|X})$ باشند به گونه ای که لایه آخرین کمترین اطلاعات متقابل با X را داشته باشد و شاهد کاهش نرخ کاهش $I(T_i;Y)$ در زنجیر مارکوفمان با هر مرحله پیشروی در الگوریتم بهینه سازی هستیم.

IB and SGD ۲.۳

از آنجایی که اطلاعات متقابل بین ورودی یک تابع غیر تصادفی با خروجی پیوسته و خروجی آن نا متناهی است، برای اینکه در مدلسازیمان بتوانیم از Information Plane استفاده کنیم نیازمند تعریفی نزدیک به اطلاعات مقابل هستیم، بدین منظور در این مقاله از binning استفاده شد به این معنا که خروجی ها در بازه هایی گسسته سازی شدند.

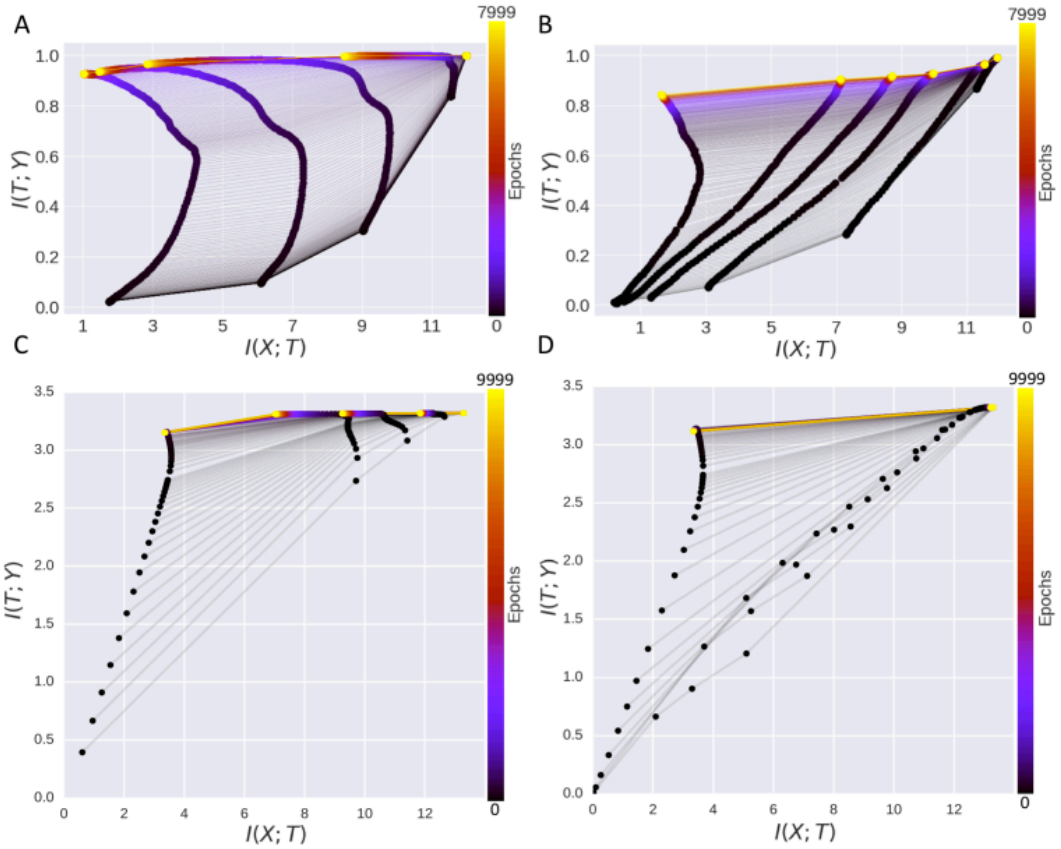
در این مقاله نتیجه ای که ارائه شد این بود شبکه های عصبی در تلاش برای بهینه سازی "trade-off" بین فشرده سازی و فیت شدن به داده ها هستند. ضمنا در حین یادگیری، دو فاز وجود دارد (مطابق شکل)، در فاز اول هر دو مختصه مربوط به لایه های شبکه افزایش پیدا می کنند و شبکه به داده ها فیت می شود، سپس شبکه وارد فاز دوم(بالای مسیر سبز رنگ) می‌شود که در آن اطلاعات متقابل نسبت به ورودی کاهش پیدا می کند که این فاز فشرده سازی یا compression نامیده می شود.

در مقاله ای که توسط آقای Saxe در سال ۲۰۱۸ ارائه شد این ادعای آخر نقض شد، آزمایش ها نشان داد که نتایج بدست آمده در مقاله آقای Tishby و همکارانشان تحت تاثیر استفاده از توابع activation ای است که saturating هستند و همچنین از binning برای ارائه تقریبی از اطلاعات متقابل استفاده شده.



از آنجایی که حین آموزش اندازه مقادیر وزن های شبکه افزایش پیدا می کنند، خروجی لایه های شبکه به یک و منفی یک نزدیک می‌شوند. همچنین binning اعدادی که به یک و منفی یک نزدیک هستند را کمتر تمایز می دهد (شکل B) و به همین دلیل اطلاعات متقابل ورودی و خروجی یک تابع مانند tanh با مدل داده شده ابتدا افزایش و سپس به مرور کاهش می‌یابد(شکل C). این درحالی است که این اتفاق در activation function هایی مانند ReLU دارای این ویژگی نیستند(شکل D).

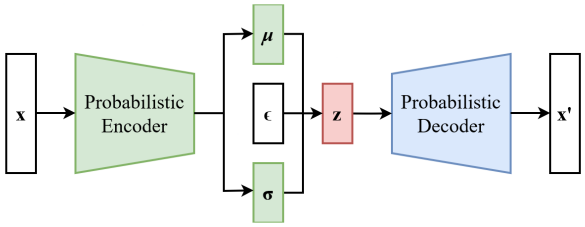
به همین دلیل وقتی آزمایش ها را با توابع غیر saturating تکرار می کنیم می بینیم که "phase transition" دیگر رخ نمی هد.



یکی از مشکلات مدلسازی ای که مطرح شد، نامتناهی بودن خود MI است. همچنین دیدیم که binning روش مناسبی برای جایگزینی آن نیست. از این جایگزین انتظار برقرار نگه داشتن قاعده پردازش داده هارا هم نمی توان داشت. چرا که binning پردازشی است که خارج از زنجیر مارکوف انجام می شود. یک ایده برای این مساله اعمال binning و ورودی دادن آن به لایه بعدی است. به عبارتی گسسته سازی خروجی ها را بخشی از زنجیر مارکوف در نظر بگیریم. روش دیگری که توسط آقای Amjed و همکارانشان در مقاله ای در سال ۲۰۱۸ عنوان شد استفاده از توابع تصادفی در شبکه عصبی است که منجر به متناهی شدن اطلاعات متقابل می‌شود.

IB and VAEs ۳.۳

یکی از ویژگی های جالب IB شباهت تابع هدف آن به تابع هدف مدل های $Variational\ Auto\ Encoder$ ها است. ابتدا به معرفی VAE ها می پردازیم:



در یک مدل VAE به دنبال رسیدن به دو پردازش $P_{Z|X}$ و $P_{X'|Z}$ هستیم به گونه ای که ترکیب توزیع P_X با پردازش اول به توزیعی مشخصی ($P_0(Z)$) نزدیک باشد. به عبارتی نمونه گیری از Z از مسیر $X \overset{P_{Z|X}}{\rightarrow} Z$ به نمونه ای از توزیع $P_0(Z)$ منجر شود. همچنین پردازش دوم دارای این خاصیت باشد که احتمال برگشتن به همان نمونه xای که به پردازش اول داده شده به خروجی ای نزدیک به همان x منجر شود.
بنابراین به طور خلاصه برای ما مطلوب است:

$$\tilde{X} \sim p_\phi(\tilde{X}|x)p(x), X' \sim p_\theta(x'|x)p_0(z)$$
$$\tilde{X} \rightarrow Z, X' \rightarrow X$$

که در اینجا ϕ, θپارامتر های مدل هایی هستند که این پردازش آنها را محدود به این مدل ها در نظر می‌گیریم. منظور از $\rightarrow$ نزدیکی توزیع متغییر ها است. تابع هدفی که برای VAE می توان در نظر گرفت به صورت زیر است:

$$\mathcal{L}(\theta,\phi) \triangleq \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{E}_{p_\theta(\tilde{x}|x^{(i)})}\left[-\log p_\phi(x^{(i)}|\tilde{x})\right]}_{\text{marginal likelihood of data point}} + \underbrace{D_{KL}(P_\theta(\tilde{x}|x)||P_0(\tilde{x}))}_{\text{distance from the prior dist.}})$$

حال اگر $\mathcal{L}_\beta(P_{\tilde{X}|X})$ را به کمک فرمول بندی ای از MI که براساس انتروپی است بازنویسی کنیم:

$$\mathcal{L}_\beta(P_{\tilde{X}|X}) = I(\tilde{X}|X) + \beta H(Y|\tilde{X})$$

که می توان با داشتن داده ها تخمینی برای آن درنظر گرفت:

$$\mathcal{L}_{approx.} \triangleq \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{E}_{p(\tilde{x}|x^{(i)})}\left[-\log \mathbb{P}\left[y^{(i)}|\tilde{x}\right]\right]}_{\text{Loss Entropy Cross}} + \beta \underbrace{D_{\mathrm{KL}}(P(\tilde{x}|x^{(i)})||P(\tilde{x}))}_{regularization}$$

حال با داشتن سه شرط، $\mathcal{L}_{\text{approx.}}$ با $\mathcal{L}(\theta,\phi)$ برابر می‌شود.

- $\beta = 1$
- $Y = X$
- داشتن توزیع پیشین برای $\tilde{X}$

این تمایز ها بین VAE و روش IB می تواند راهنمایی برای بهبود مدل های VAE باشد. در روش IB ،مقدار $\beta^* > 1$ وجود دارد که بهترین مقدار برای "trade-off" بین bias,variance است. این درحالی است که در روش VAE تابع هدفی که تعریف شده دارای پارامتر $\beta = 1$است. در کار آقای Higgins در سال ۲۰۱۶ ، مدل beta-VAE ارائه شد که در آن $\beta > 1$به نتایج بهتری منجر می شد.
نمود دیگر این تمایز ها مستقل بودن $p_\theta(\tilde{x}|x), p_\phi(x|\tilde{x})$ است. در روش IB توزیع $p(\tilde{x}|x), p(x|\tilde{x})$ وابسته هستند و امکان تغییر آنها به طور مستقل وجود ندارد. یکی از مشکلات VAE ها این است که با داشتن مدل های با ظرفیت بالا، امکان این وجود دارد که decoder مستقل از Encoder روشی برای نزدیک شدن به توزیع X پیدا کند که این باعث می شود که در مسائل feature extraction نتوان از چنین مدل هایی استفاده کرد.

۴ ابزار های دیگر

یک مدل که در کنار روش IB مطرح بوده و ممکن است بتوان از آن نتایج خوبی در مورد رفتار مدل های یادگیری عمیق گرفت بررسی اطلاعات متقابل کل داده ها و وزن های خروجی داده شده از الگوریتم یادگیری است. به عبارتی $I(S;\mathcal{A}(S))$ در مقابل $I(T_i;Y), I(T_i;X)$. در اینجا منظور از $\mathcal{A}$الگوریتم (تصادفی) یادگیری است. یک مثال از خواص خوب این معیار، باند بالا برای خطای تعمیم است:

$$\mathbb{P}\left[|err_{train} - err_{test}| > \epsilon\right] < O\left(\frac{I(S;\mathcal{A}(S))}{n\epsilon^2}\right)$$

به عبارتی هرچه خروجی الگوریتم ارتباط کمتری با نمونه های گرفته شده از توزیع اصلی داشته باشد خطای تعمیم کمتر خواهد بود. در عین حال تعداد نمونه ها در کاهش خطای تعمیم موثر است.

۵ منابع

[Hafez-Kolahi, 2019] Hassan Hafez-Kolahi and Shohreh Kasaei (2019) Information Bottleneck and its Applications in Deep Learning *Journal of Information Systems and Telecommunication*

[Goldfeld, 2020] Ziv Goldfeld and Yury Polyanskiy (2020) The Information Bottleneck Problem and Its Applications in Machine Learning *CoRR*

[Tishby, 2000] Naftali Tishby, Fernando C. Pereira and William Bialek (2000) The information bottleneck method *arXiv*

[Andrew Michael Saxe, 2018] Andrew Michael Saxe and Yamini Bansal and Joel Dapello and Madhu Advani and Artemy Kolchinsky and Brendan Daniel Tracey and David Daniel Cox (2018) On the Information Bottleneck Theory of Deep Learning *International Conference on Learning Representations*

[Rana Ali Amjad, 2018] Rana Ali Amjad and Bernhard C. Geiger (2018) How (Not) To Train Your Neural Network Using the Information Bottleneck Principle *CoRR*