

## گزارشکار (Restricted) Boltzmann Machine (RBM)

استاد : جناب دکتر طالبی

اعضای گروه :

محمدحسین هادیان

ملیکا صالحیان

محمدادیب جهان تاب

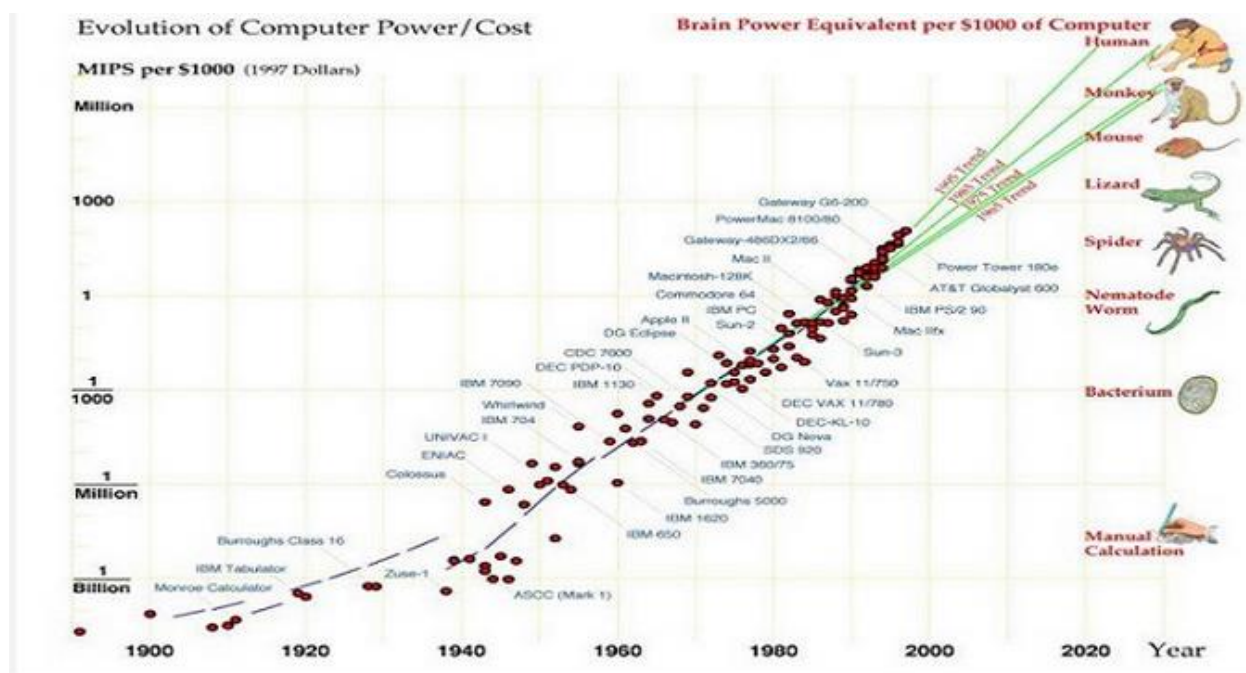
محمدعرفان شیخ محمدی

## فهرست

۱-۱ تاریخچه	۳
۲-۱ معرفی پایه ای ماشین بولتزمن	۴
۱-۲-۱ مارکو چین	۴
۲-۲-۱ مدل گرافیکی	۵
۳-۱ ماشین بولتزمن محدود	۷
۴-۱ قانون های وزن و یادگیری شبکه	۸
۵-۱ آموزش RBM	۱۱
۵-۱-۱ روش نمونه برداری گیبز (Sampling Gibbs)	۱۳
۵-۱-۲ روش واگرایی متقابل (Contrastive Divergence)	۱۳
۶-۱ نتیجه گیری:	۱۵
۷-۱ خروجی کد آزمایشی:	۱۶
۷-۱ منابع	۱۷

## ۱-۱ تاریخچه

در دهه ۱۹۷۰، شبکه‌های عصبی اولیه می‌توانستند تعداد بسیار محدودی از نورون‌ها را همزمان شبیه‌سازی کنند، بنابراین نمی‌توانستند الگوهای پیچیدگی زیادی را تشخیص دهند. در اواسط دهه ۱۹۸۰، جفری هینتون و دیگران به بهبود شبکه‌های عصبی با مدل‌های به اصطلاح عمیق کمک کردند که باعث استفاده بهتر از لایه‌های نورون‌های نرم افزاری شد. اما این تکنیک همچنان به مشارکت شدید انسانی نیاز داشت زیرا برنامه‌نویسان مجبور بودند داده‌ها را قبل از تغذیه به شبکه برچسب گذاری کنند و تشخیص پیچیده گفتار/تصویر به قدرت کامپیوتری بیشتری نسبت به آنچه که در دسترس بود نیاز داشت.



شکل ۱: تکامل قدرت کامپیوتر و رشد هزینه‌های آن

با این حال، در سال ۲۰۰۶، هینتون روش کارآمدتری برای آموزش لایه‌های جداگانه نورون‌ها ایجاد کرد که در آن لایه اول ویژگی‌های ابتدایی، مانند لبه در یک تصویر یا کوچک‌ترین واحد صدای گفتاری را با یافتن ترکیبی از پیکسل‌های دیجیتالی یا امواج صوتی که اغلب رخ می‌دهند، یاد می‌گیرد. هنگامی که آن لایه به طور دقیق آن ویژگی‌ها را تشخیص داد، به لایه بعدی هدایت می‌شود، که خود را برای تشخیص ویژگی‌های پیچیده‌تر، مانند گوشه یا ترکیبی از صداها یا گفتار، آموزش می‌دهد. این فرآیند در لایه‌های متوالی تکرار می‌شود تا زمانی که سیستم بتواند واج‌ها یا اشیاء را به‌طور قابل اعتماد تشخیص دهد و این همان چیزی است که پایه مدل‌های یادگیری عمیق نظارت شده مانند شبکه‌های عصبی مصنوعی/پیوندی/تکراری را تشکیل می‌دهد. حتی قبل از آن، هینتون به همراه تری سجنوفسکی در سال ۱۹۸۵ یک مدل یادگیری عمیق بدون نظارت به نام ماشین بولتزمن را اختراع کردند. این مدل بر اساس توزیع بولتزمن (همچنین به عنوان توزیع گیبس شناخته می‌شود) است که بخشی جدایی‌ناپذیر از مکانیک آماری است و به ما کمک می‌کند تا تأثیر پارامترهایی مانند آنتروپی و دما را بر حالات کوانتومی در ترمودینامیک درک کنیم.

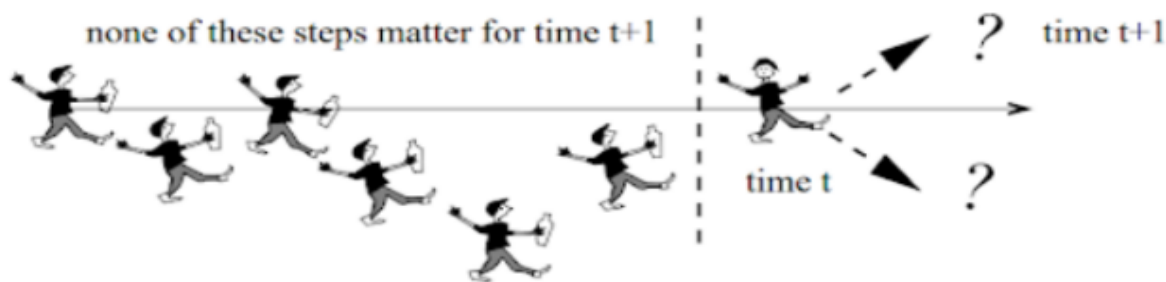
هیئتون زمانی به تصویرسازی نیروگاه هسته ای به عنوان مثالی برای درک ماشین های بولتزمن اشاره کرد. این یک موضوع پیچیده است، بنابراین ما به آرامی برای درک شهود پشت هر مفهوم، با حداقل مقدار ریاضی و فیزیک پیش می رویم. بنابراین در ساده ترین ماشین های بولتزمن به دو دسته اصلی ۱-مدل های مبتنی بر انرژی ۲-مدل های مبتنی بر ماشین بولتزمن محدود تقسیم میشوند. که در صورت چیده شدن RBM روی هم به شبکه های دیپ بلیف Deep belief network (DBN) میرسیم.

## ۲-۱ معرفی پایه ای ماشین بولتزمن

ماشین های بولتزمن خود (که توسط قانون هب آموزش داده می شوند) از لحاظ نظری به دلیل محلی بودن و ماهیت های الگوریتم آموزشی خود، و به دلیل موازی بودن و شباهت دینامیک آنها به فرآیندهای فیزیکی ساده، جذاب هستند. ماشین های بولتزمن با اتصال نامحدود برای مشکلات عملی در ماشین لرنینگ یا استنتاج مفید نبوده اند، اما اگر اتصال به درستی محدود شود، یادگیری می تواند آنقدر کارآمد شود که برای مشکلات عملی مفید باشد. حال در ابتدا به بررسی ماشین بولتزمن معمولی میپردازیم و بعد به علت مشکلاتش که می خواهیم در ادامه ذکر کنیم به سراغ نوع محدود (restricted) میرویم. برای درک ماشین بولتزمن نیاز است که در ابتدا با مدل مارکو چین (Markov chain) و مدل گرافیکی آشنا شویم.

### ۱-۲-۱ مارکو چین

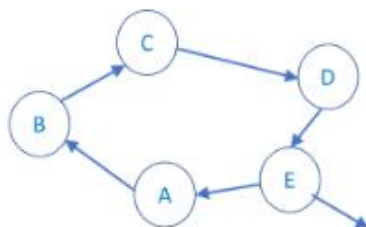
زنجیره مارکوف یک مدل احتمالی است که برای تخمین دنباله ای از رویدادهای احتمالی استفاده می شود که در آن احتمال هر رویداد فقط به وضعیت به دست آمده در رویداد قبلی بستگی دارد. در زنجیره مارکوف، وضعیت آینده فقط به وضعیت حال بستگی دارد و نه به حالات گذشته. نمونه ای از فرآیند مارکوف در شکل زیر نشان داده شده است. (موقعیت فردی که به طور تصادفی راه می رود) این رفتار به عنوان ویژگی مارکوف شناخته میشود.



شکل ۲: حرکت فرد فقط به مکان لحظه قبل بستگی دارد

## ۲-۲-۱ مدل گرافیکی

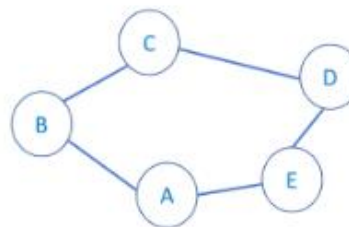
یک مدل احتمالی گرافیکی یک نمایش گرافیکی است که برای بیان وابستگی شرطی بین متغیرهای تصادفی استفاده می شود. یک مدل گرافیکی دارای دو جزء در خود است. رئوس و لبه ها. رئوس حالت متغیر تصادفی و یال نشان دهنده جهت تبدیل است. شکل زیر نمایش دهنده دو نوع اصلی مدل های گرافیکی است.



Vertices (V): {A,B,C,D,E}  
Edges (E): { (A,B), (B,C), (C,D), (D,E), (E,A) }

Directed graph

شکل ۴: گراف با جهت



Vertices (V): {A,B,C,D,E}  
Edges (E): { (A,B), (B,A), (B,C), (C,B), (C,D), (D,C), (D,E), (E,D), (E,A), (A,E) }

Undirected graph

شکل ۳: گراف بدون جهت

تفاوت میان گراف جهت دار و بدون جهت :

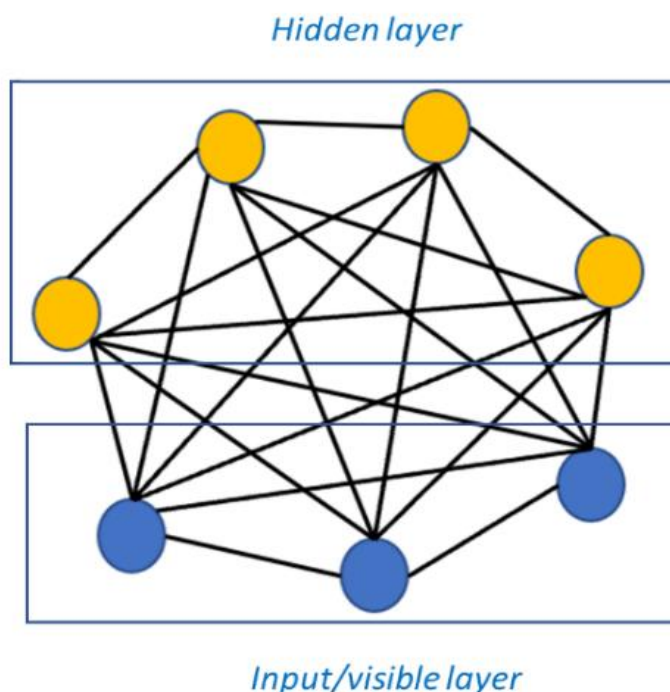
در گراف جهت دار، وضعیت متغیر می تواند در یک جهت تبدیل شود. در نمودار جهت دار در شکل ۴، وضعیت متغیر می تواند از A-B or C-D از طریق جهت فلش ها منتقل شوند ولی برای مثال امکان حرکت از D-C or B-A وجود ندارد. این در حالی است که در صورتی که گراف بدون جهت را بررسی کنیم دیگر محدودیت حرکت نخواهیم داشت. از هر استیت به هر استیت دیگر و در هر جهتی می توانیم حرکت کنیم.

شکل ۳ یک مدل گرافیکی بدون جهت از فرآیند مارکوف است که عادت رژیم غذایی نوزاد را نشان می دهد. مدل نمودار برای نشان دادن انتخاب کودک برای وعده غذایی بعدی با احتمالات مرتبط استفاده می شود. انتخاب وعده غذایی بعدی کودک صرفاً به آنچه در حال حاضر می خورد بستگی دارد نه آنچه که قبلاً می خورد. احتمال انتخاب یک غذای خاص برای وعده غذایی بعدی بر اساس مشاهدات محاسبه می شود.

مجموعه ای از متغیرهای تصادفی که دارای ویژگی مارکوف هستند و توسط یک گراف غیر جهت دار توصیف می شوند، به عنوان شبکه مارکوف نامیده می شوند.

ما اکنون به برخی از مفاهیم اساسی برای درک (BM) پی برده ایم. ماشین بولتزمن یک مدل گراف بدون جهت مولد احتمالی که ویژگی های مارکو را برآورده میکند. ماشین بولتزمن چگالی احتمال را از داده های ورودی یاد می گیرد تا به تولید داده های جدید از همین توزیع ها بپردازد. دارای یک لایه ورودی یا قابل مشاهده و یک یا چند لایه پنهان است. هیچ لایه خروجی وجود ندارد.

شکل ۵ یک نمونه از ماشین بولتزمن با یک لایه ی پنهان را نشان میدهد. همانطور که از شکل دریافت میشود مدل ماشین بولتزمن همانند شبکه هاپفیلد است و از بالا به پایین است.

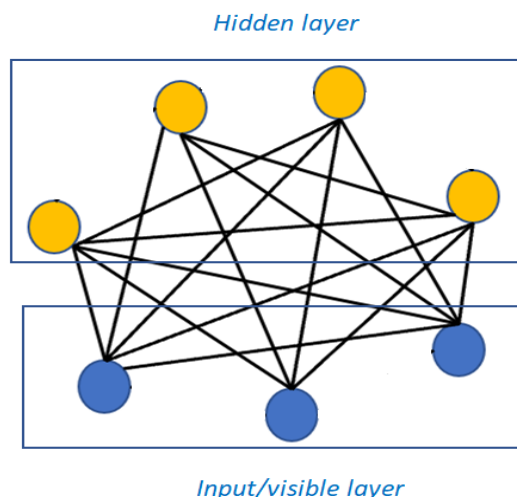


شکل ۵: مدل ساده ای از ماشین بولتزمن نامحدود با یک لایه ی پنهان

نرون های شبکه که بر اساس داده هایی که در طول آموزش به شبکه داده می شوند یاد می گیرند که تصمیم گیری تصادفی در مورد خاموش و روشن شدن نرون ها انجام بدهند. این امر به ماشین بولتزمن کمک میکند تا الگو های پیچیده درون داده ها را کشف و مدلسازی کند یک تفاوت حیاتی بین ماشین بولتزمن و شبکه های محبوب دیگر این است که در ماشین بولتزمن نرون ها علاوه بر اینکه به نرون های لایه های دیگر وصل هستند نرون های هر لایه نیز به هم وصل هستند بنابراین بدون قید و شرط میتوان گفت تمامی نرون ها به هم متصل هستند و دقیقا به همین علت آموزش این شبکه با دشواری های زیادی روبه رو است. به همین علت کاربرد کمتری دارد و در عوض نسخه محدود آن یعنی (restricted boltzmann machine(RBM)) کاربرد های بیشتری دارد.

ماشین بولتزمن عادی که در واقع همان نسخه محدود آن است در اثر حذف ارتباط های نرون در یک لایه به وجود آمده است(که در اثر این حذف دیگر نرون ها در لایه پنهان با هم ارتباط برقرار نمی کنند و هر گره در یک لایه محاسبات خود را انجام می دهد. پس از انجام محاسبات، گره تصمیمی تصادفی در مورد اینکه آیا آن را به لایه بعدی منتقل کند یا خیر می گیرد).

حال در ادامه به بررسی دقیق نوع محدود ماشین بولتزمن و کاربرد های آن خواهیم پرداخت.



شکل ۶: مدل ساده ای از ماشین بولتزمن محدود با یک لایه ی پنهان

### ۳-۱ ماشین بولتزمن محدود

به علت دلایل ذکر شده در بالاتر که از مهمترین آن ها میتوان به سختی در پیاده کردن ماشین بولتزمن اکثراً از نوع محدود آن استفاده میکنند . ماشین بولتزمن محدود یک مدل مولد پارامتری شده است که توزیع احتمالی را نشان می دهد که برای مقایسه احتمال مشاهدات (دیده نشده) و نمونه برداری از توزیع آموخته شده، به ویژه از توزیع های حاشیه ای مورد علاقه استفاده می شود. ایده این است که نرون های پنهان ویژگی های مربوطه را از مشاهدات استخراج می کنند که به عنوان ورودی برای آر بی ام بعدی که در بالای آن انباشته شده است، استفاده می کنند و یک شبکه عصبی پیش خورنده قطعی را تشکیل می دهند.

ورودی های ما با وزن های تعمیم یافته مقداردهی اولیه می شوند و به گره های مخفی منتقل می شوند، که به نوبه خود گره های ورودی ما را بازسازی می کنند، و این گره های بازسازی شده هرگز با گره های قابل مشاهده ما اصلی ما یکسان نیستند. این تفاوت به این دلیل است که همانطور که قبلاً گفته شد، گره های قابل مشاهده ما هرگز به هم متصل نبودند، بنابراین نمی توانستند یکدیگر را مشاهده کنند و از یکدیگر یاد بگیرند.

این توالی بازسازی با واگرایی متضاد<sup>۱</sup> تا رسیدن به حداقل انرژی گلوبال ادامه می یابد و به نمونه گیری گیبس معروف است. آر بی ام را می توان به عنوان یک شبکه عصبی تصادفی تفسیر کرد که در آن گره ها و لبه ها به ترتیب با نرون ها و اتصالات سیناپسی مطابقت دارند احتمال شرطی بودن یک متغیر واحد را می توان به عنوان نرخ شلیک یک نرون (تصادفی) با تابع فعال سازی سیگموئید تفسیر کرد.

استقلال بین متغیرها در یک لایه، نمونه برداری گیبس را بسیار آسان می کند، زیرا به جای نمونه برداری از مقادیر جدید بنابراین، برای همه متغیرها، می توان حالت های همه متغیرها در یک لایه را به طور مشترک نمونه برداری کرد.

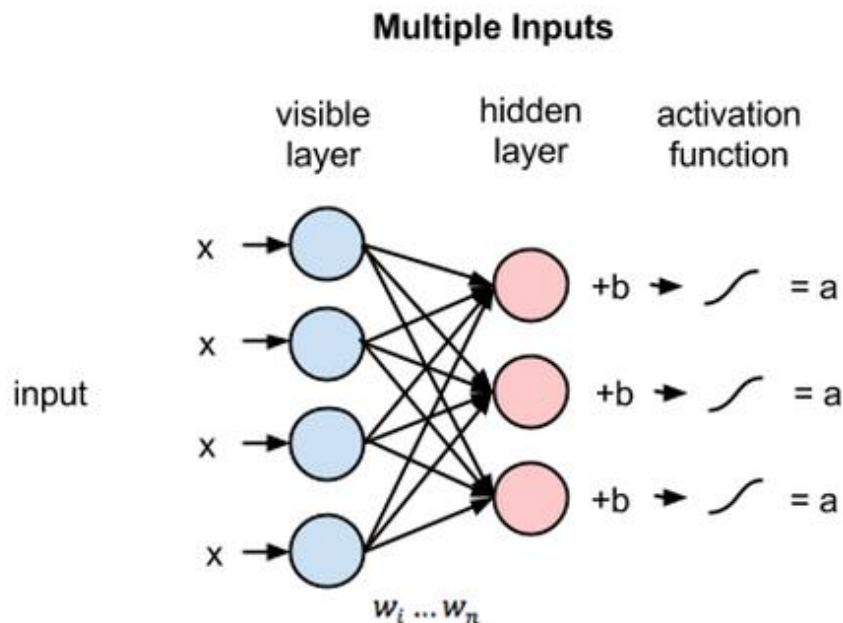
نمونه برداری گیبس را میتوان در دو مرحله فرعی انجام داد:

۱- نمونه برداری از حالت جدید اچ<sup>۱</sup> برای نرون های پنهان بر اساس  $p(h|v)$

۲- نمونه برداری از وضعیت وی برای لایه ی مرئی بر اساس  $p(v|h)$

به این نمونه برداری بلوکی گیبس نیز گفته می شود.

<sup>1</sup> Contrastive Divergence



شکل ۷: لایه های پنهانی و معلوم ماشین بولتزمن

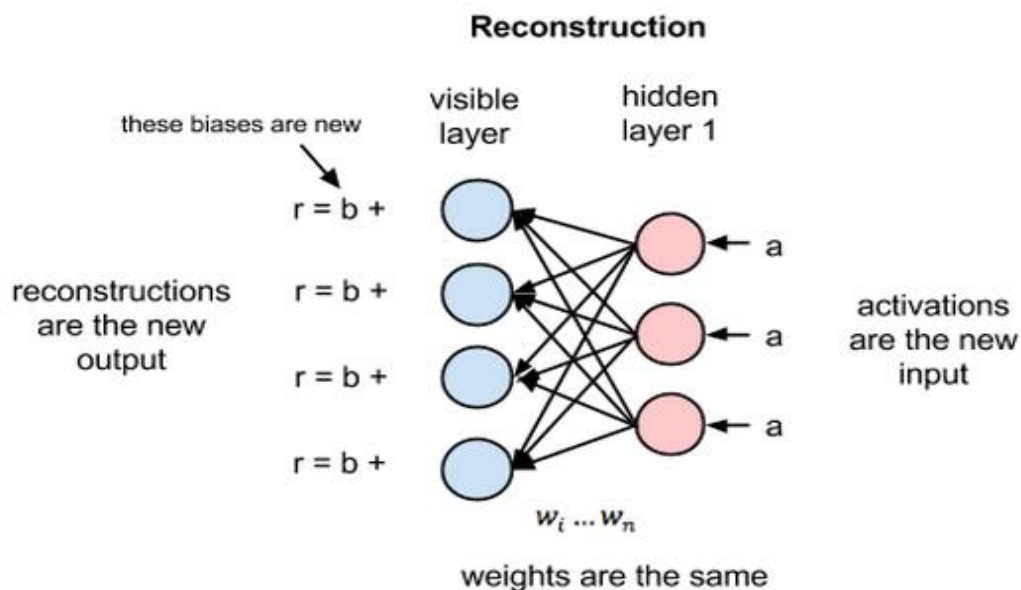
شکل بالا اولین مرحله از آموزش شبکه آر بی ام را نشان میدهد ورودی ها در وزن ها ضرب می شوند و سپس به بایاس اضافه می شوند. سپس نتیجه از طریق یک تابع فعال سازی سیگموئید منتقل می شود و خروجی تعیین می کند که آیا حالت پنهان فعال می شود یا خیر. وزن ها یک ماتریس با تعداد گره های ورودی به عنوان تعداد ردیف ها و تعداد گره های پنهان به عنوان تعداد ستون ها خواهد بود. اولین گره پنهان، ضرب برداری بردار ورودی ها را در ستون اول وزن ها قبل از اینکه عبارت بایاس مربوطه به آن اضافه شود، دریافت می کند. آنچه که در مرحله اول بدست می آید در پایین نشان داده شده است.

$$\mathbf{h}^{(1)} = S(\mathbf{v}^{(0)T} \mathbf{W} + \mathbf{a})$$

که در آن بردار های متناظر  $\mathbf{h}^{(1)}, \mathbf{v}^{(0)}$  بردار های متناظر (ماتریس های ستونی) برای لایه های قابل مشاهده و پنهان هستند که روی آن تکرار می شود.  $\mathbf{V}^{(0)}$  به معنای ورودی است که به شبکه ارایه میکنیم و  $\mathbf{a}$  بردار بایاس آن است.

(توجه داشته باشید که در اینجا با بردار ها و ماتریس ها سروکار داریم و نه مقادیر تک بعدی).





شکل ۸: مرحله ی دوم در ترین

حال این تصویر فاز معکوس یا مرحله بازسازی را نشان می دهد. شبیه فاز اول اما در جهت مخالف است. معادله به دست می آید: که در آن  $H(1), v(1)$  های متناظر (ماتریس های ستونی) برای لایه های مرئی و پنهان با بالانویس به عنوان تکرار و  $b$  بایاس لایه پنهانی است.

حال تفاوت  $v(0)-v(1)$  را میتوان به عنوان خطای بازسازی در نظر گرفت که باید در مراحل بعدی فرآیند آموزش آن را کاهش داد. بنابراین وزن ها در هر تکرار به گونه ای تنظیم می شوند که این خطا به حداقل برسد و این همان چیزی است که فرآیند یادگیری اساساً انجام می شود. حال، بیایید سعی کنیم این فرآیند را بدون اینکه عمیقاً در ریاضیات وارد شویم، در اصطلاحات ریاضی درک کنیم. در حرکت رو به جلو، احتمال خروجی  $h(1)$  را باتوجه به ورودی  $matrix$   $v(0)$  محاسبه کنیم.

$$p(\mathbf{h}^{(1)} \mid \mathbf{v}^{(0)}; W)$$

و در گذر به عقب، در حین بازسازی ورودی، احتمال خروجی  $v(1)$  را در حرکت به عقب برای بازسازی ورودی احتمال خروجی  $v(1)$  را با توجه به ورودی و ماتریس وزن  $W$ ،  $H(1)$  محاسبه می کنیم.

$$p(\mathbf{v}^{(1)} \mid \mathbf{h}^{(1)}; W)$$

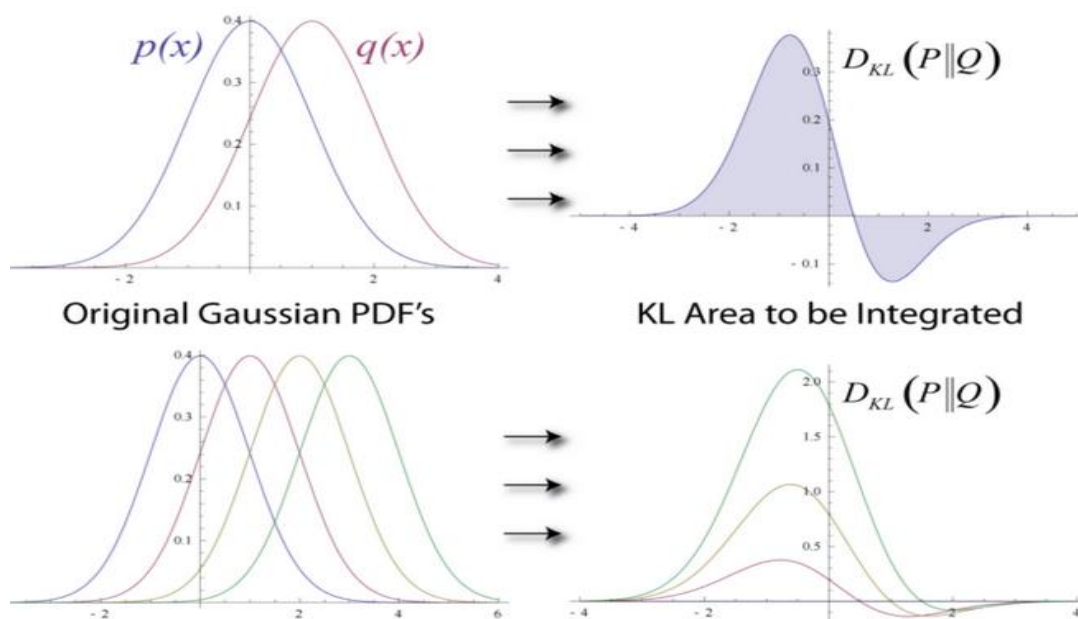
وزن های استفاده شده در هر دو حرکت جلو و عقب یکسان است. این دو احتمال مشروط با هم ما را به توزیع مشترک ورودی ها و فعال سازی ها هدایت می کنند.

$$p(\mathbf{v}, \mathbf{h})$$

بازسازی با رگرسیون یا طبقه‌بندی متفاوت است زیرا به جای مرتبط کردن یک مقدار پیوسته/گسسته به یک مثال ورودی، توزیع احتمال ورودی اصلی را تخمین می‌زند. این بدان معنی است که سعی دارد چندین مقدار را به طور همزمان حدس بزند.

حال در ادامه بررسی خواهیم کرد که الگوریتم چگونه ارور را کاهش می‌دهد.

فرض کنید که دو توزیع نرمال داریم، یکی از داده‌های ورودی (که با  $P(x)$  نشان داده می‌شود) و دیگری تقریبی از ورودی (که با  $Q(x)$  نمایش داده می‌شود) تفاوت بین این دو توزیع، خطای ما در مفهوم گرافیکی است. هدف ما به حداقل رساندن آن است یعنی نمودارها را تا حد امکان به هم نزدیک کنیم. این ایده با اصطلاحی به نام واگرایی<sup>۲</sup> نشان داده می‌شود. واگرایی نواحی غیر همپوشانی را در زیر دو نمودار اندازه‌گیری می‌کند و الگوریتم بهینه‌سازی را بر بی ام<sup>۳</sup> سعی می‌کند این تفاوت را با تغییر وزن‌ها به حداقل برساند تا بازسازی بسیار شبیه ورودی باشد. نمودارهای سمت راست ادغام تفاوت در مناطق منحنی سمت چپ را نشان می‌دهد.



شکل ۹: نمونه ای از نحوه ی عملکرد RBM

این به ما شهودی در مورد اصطلاح خطای خود می‌دهد. اکنون، برای اینکه ببینیم واقعاً چگونه این کار برای ماشین بولتزمن انجام می‌شود. همانند بقیه الگوریتم‌های آموزشی در این الگوریتم همه به محاسبه loss میپردازیم و در جهت گرادیان دیسنت آن حرکت می‌کنیم.

<sup>۲</sup> Kullback-Leibler

<sup>۳</sup> RBM

## ۱-۵ آموزش RBM

ماشین های بولتزمن (RMB) مدل های مبتنی بر انرژی هستند و یک پیکربندی مشترک،  $(v, h)$  واحدهای مرئی و پنهان دارای انرژی است که توسط:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij}$$

$v_i, h_i$  حالت های باینری واحد مرئی ii و واحد پنهان j ،  $a_i, b_i$  بایاس ها هستند و  $w_{ij}$  وزن بین آنها است.

احتمالی که شبکه به یک بردار مرئی نسبت داده شده،  $V$  ، با جمع کردن تمام بردارهای پنهان ممکن به دست می آید:

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}$$

$Z$  در اینجا ثابت نرمال سازی است که به صورت زیر به دست می آید:

$$Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}$$

بدین ترتیب داریم:

$$p(\mathbf{v}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}}{\sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}}$$

مشتق احتمال لگاریتم بردار training با توجه به وزن به صورت زیر تخمین می زنیم:

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}$$

( $\langle \rangle$ : امید ریاضی بر روی ضرب مقادیر مشاهده پذیر و مخفی)

قانون اصلاح وزن ها به صورت زیر محاسبه می گردد:

$$\Delta w_{ij} = \epsilon (< v_i h_j >_{data} - < v_i h_j >_{model})$$

$$\Delta a_i = \epsilon (< v_i >_{data} - < v_i >_{model})$$

$$\Delta b_j = \epsilon (< h_j >_{data} - < h_j >_{model})$$

ε نرخ یادگیری است.

نکته مهمی که باید در نظر گرفته شود این است که به خاطر این که هیچ ارتباط مستقیمی بین واحد های پنهان در یک RBM وجود ندارد به راحتی می شود که در نمونه گیری unbiased از  $< v_i, h_i >_{data}$  دریافت کرد. یک تصویر آموزشی به صورت تصادفی انتخاب شده است (v)، حالت باینری  $(h_i)$ ، از هر واحد پنهان j با احتمال ۱ تنظیم شده است.

$$p(h_j = 1 | \mathbf{v}) = \sigma(b_j + \sum_i v_i w_{ij})$$

$\sigma(x)$  در اینجا تابع سیگموئید لجستیک است:

$$1/(1 + \exp(-x))$$

زیرا هیچ ارتباط مستقیمی بین واحدهای قابل مشاهده در یک RBM وجود ندارد به راحتی می شود که در نمونه گیری unbiased از حالت یک واحد مرئی دریافت کرد. یک بردار پنهان داده شده است.

$$p(v_i = 1 | \mathbf{h}) = \sigma(a_i + \sum_j h_j w_{ij})$$

با ضرب این دو مقدار  $< v_i, h_i >_{data}$  به دست میاید.

حال باید  $< v_i, h_i >_{model}$  را محاسبه کنیم.

۱- نمونه برداری گیبز

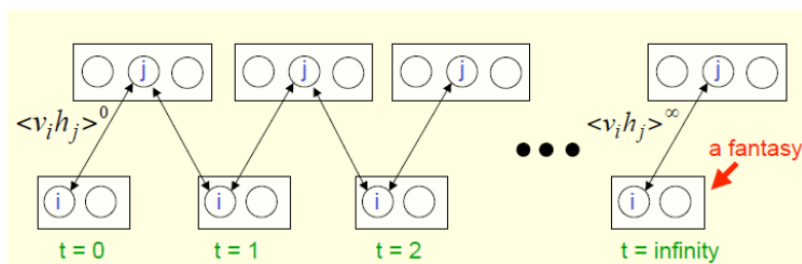
۲- روش واگرایی متقابل

## ۱-۵-۱ روش نمونه برداری گیبز (Sampling Gibbs)

این روش سرعت پایین و عمل غیرممکن و ابتدا قرار دادن یک بردار از داده آموزشی در لایه مشاهده پذیر

سپس مراحل زیر را به صورت نامحدود تکرار می کنیم:

- بروزرسانی تمام واحدهای پنهان به صورت موازی
- بروزرسانی تمام واحدهای مشاهده پذیر به صورت موازی
- دوباره بروزرسانی واحدهای مخفی



و داریم:

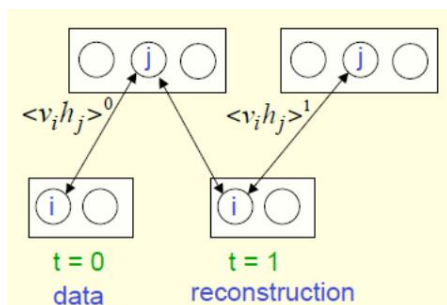
$$\frac{\partial \log p(v)}{\partial W_{ij}} = \langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^\infty$$

مرحله صفر از نمونه برداری گیبز و معادل امید ریاضی ضرب  $h_i, v_i$  می باشد.

## ۱-۵-۲ روش واگرایی متقابل (Contrastive Divergence)

- اجرای نمونه برداری گیبز تا گام زمانی  $t=1$
- قراردادن یک بردار از ورودی در لایه مشاهده پذیر
- بروزرسانی تمام واحدهای مخفی به صورت موازی
- بروزرسانی تمام واحدهای مشاهده پذیر به صورت موازی
- دوباره بروزرسانی واحدهای مخفی
- و در نهایت اصلاح وزن ها

$$\frac{\partial \log p(v)}{\partial W_{ij}} = \langle v_i h_j \rangle^0 - \langle v_i h_j^1 \rangle$$



$$\Delta w_{ij} = \alpha (\langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^1)$$

$$\Delta a_i = \alpha (\langle v_i \rangle^0 - \langle v_i \rangle^1)$$

$$\Delta b_j = \alpha (\langle h_j \rangle^0 - \langle h_j \rangle^1)$$

## الگوریتم آموزش واگرایی متقابل

گام ها :

۰- مقداردهی اولیه وزن ها به صورت تصادفی و صفر قرار دادن مقادیر بایاس

۱- برای هر بردار ورودی گام های ۲ تا ۹ را تکرار کنید

۲- یک بردار ورودی ( $v$ ) را در لایه مشاهده پذیر قرار دهید

۳- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

$$\text{positive}_{hidden} \rightarrow P(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^m w_{ij} v_i)$$

۴- مقدار واحد های پنهان را در صورتی که احتمال به دست آمده بزرگتر از یک مقدار تصادفی در بازه  $[0,1]$  بود

برابر یک و در غیر اینصورت برابر صفر قرار دهید (بردار  $h$ )

۵- احتمال فعال شدن واحدهای مشاهده پذیر را به شرط بردار  $h$  محاسبه نمایید

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^n w_{ij} h_j)$$

۶- در صورتی که احتمال به دست آمده بزرگتر از یک مقدار تصادفی در بازه  $[0,1]$  بود، مقدار واحدهای مشاهده

پذیر را برابر یک و در غیر اینصورت برابر صفر قرار دهید. (بردار  $v'$ )

۷- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v'$  محاسبه نمایید.

$$\text{negative}_{hidden} \rightarrow P(h'_j = 1|v') = \sigma(b_j + \sum_{i=1}^m w_{ij} v'_i)$$

۸- تغییرات وزن را به کمک رابطهای زیر محاسبه کنید.

$$\Delta W = \alpha(v^T * \text{Positive}_{Hidden} - v'^T * \text{Negative}_{Hidden})$$

$$\Delta a = \alpha(v - v')$$

$$\Delta b = \alpha(\text{Positive}_{Hidden} - \text{Negative}_{Hidden})$$

۹- وزن ها را بروز رسانی کنید.

$$W = W + \Delta W$$

$$b = b + \Delta b$$

$$a = a + \Delta a$$

#### ۱-۶ نتیجه گیری:

هدف ماشین بولتزمن محدود شده (RBM) این است که ورودی ها را در حد امکان دقیق بازسازی کند. در گذر از جلو ، ورودی ها توسط وزن ها و بایاس ها اصلاح می شوند و برای فعال کردن لایه پنهان استفاده می شوند. در مرحله ی بعدی ، فعال سازی های موجود از لایه پنهان توسط وزن ها و بایاس ها اصلاح می شوند و برای فعال سازی به لایه ورودی ارسال می شوند. در لایه ورودی، فعال سازی های اصلاح شده به عنوان بازسازی ورودی مشاهده می شوند و با ورودی اصلی مقایسه می شوند. از معیارهایی به نام KL Divergence برای تحلیل دقیق شبکه استفاده می شود. روند آموزش شامل افزایش مداوم وزن ها و بایاس ها در هر دو گذر است تا ورودی بازسازی شده تا حد ممکن به ورودی اصلی نزدیک شود.

## ۷-۱ خروجی کد آزمایشی:

```
[BernoulliRBM] Iteration 1, pseudo-likelihood = -25.57, time = 0.12s
[BernoulliRBM] Iteration 2, pseudo-likelihood = -23.68, time = 0.23s
[BernoulliRBM] Iteration 3, pseudo-likelihood = -22.88, time = 0.23s
[BernoulliRBM] Iteration 4, pseudo-likelihood = -21.91, time = 0.22s
[BernoulliRBM] Iteration 5, pseudo-likelihood = -21.79, time = 0.23s
[BernoulliRBM] Iteration 6, pseudo-likelihood = -20.96, time = 0.22s
[BernoulliRBM] Iteration 7, pseudo-likelihood = -20.80, time = 0.22s
[BernoulliRBM] Iteration 8, pseudo-likelihood = -20.63, time = 0.23s
[BernoulliRBM] Iteration 9, pseudo-likelihood = -20.38, time = 0.23s
[BernoulliRBM] Iteration 10, pseudo-likelihood = -20.19, time = 0.22s
```

```
Logistic regression using RBM features:
precision    recall  f1-score   support

0           0.98      0.98      0.98       174
1           0.90      0.91      0.91       184
2           0.92      0.95      0.93       166
3           0.95      0.89      0.92       194
4           0.95      0.94      0.94       186
5           0.93      0.92      0.93       181
6           0.98      0.96      0.97       207
7           0.94      0.99      0.97       154
8           0.89      0.87      0.88       182
9           0.88      0.90      0.89       169

accuracy          0.93       1797
macro avg         0.93      0.93      0.93       1797
weighted avg      0.93      0.93      0.93       1797
```

```
Logistic regression using raw pixel features:
precision    recall  f1-score   support

0           0.90      0.92      0.91       174
1           0.60      0.59      0.60       184
2           0.75      0.85      0.80       166
3           0.78      0.78      0.78       194
4           0.81      0.84      0.83       186
5           0.77      0.77      0.77       181
6           0.91      0.87      0.89       207
7           0.86      0.88      0.87       154
8           0.67      0.58      0.62       182
9           0.75      0.76      0.76       169

accuracy          0.78       1797
macro avg         0.78      0.78      0.78       1797
weighted avg      0.78      0.78      0.78       1797
```



همان طور که در شکل بالا خروجی کد را می بینیم یکی از کاربرد های RBM است که Feature extraction می باشد.

در اینجا ۱۰۰ مؤلفه ورودی با RBM تشخیص داده شده است.



## ۱-۷ منابع

- 1- <https://www.mygreatlearning.com/blog/understanding-boltzmann-machines/>
- 2- <https://towardsdatascience.com/restricted-boltzmann-machines-simplified-eab1e5878976>
- 3- <https://wiki.pathmind.com/restricted-boltzmann-machine>
- 4- <https://arxiv.org/pdf/2107.12521.pdf>
- 5- [http://www.scholarpedia.org/article/Boltzmann\\_machine](http://www.scholarpedia.org/article/Boltzmann_machine)
- 6- <https://www.geeksforgeeks.org/types-of-boltzmann-machines/>
- 7- <https://ofai.at/papers/oefai-tr-2014-13.pdf>
- 8- <https://www.youtube.com/watch?v=i64Kpxyalpo>
- 9- <https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf>
- 10- [https://www.gsd.inesc-id.pt/~mpc/pubs/RBM\\_applied\\_no\\_NIDS-final.pdf](https://www.gsd.inesc-id.pt/~mpc/pubs/RBM_applied_no_NIDS-final.pdf)
- 11- [https://www.gsd.inesc-id.pt/~mpc/pubs/RBM\\_applied\\_no\\_NIDS-final.pdf](https://www.gsd.inesc-id.pt/~mpc/pubs/RBM_applied_no_NIDS-final.pdf)
- 12- <https://analyticsindiamag.com/beginners-guide-to-boltzmann-machines/>
- 13- [https://en.wikipedia.org/wiki/Boltzmann\\_machine](https://en.wikipedia.org/wiki/Boltzmann_machine)
- 14- <https://medium.datadriveninvestor.com/an-intuitive-introduction-of-boltzmann-machine-8ec54980d789>
- 15- [ANN-Lecture4-DBN.pdf](#)

۱۶- بررسی الگوریتمهای فرا ابتکاری تنظیمکننده پارامترهای ماشین بولتزمن \*محمد رحیمی

۱۷- روشی جدید برای پیش تعلیم شبکه عصبی بر مبنای ماشین بولتزمن عمیق به منظور افزایش سرعت آموزش و دقت در بازشناسی واج 1 تکتم ذوقی 2 و محمد مهدی همایون پ