

# شبکه‌های عصبی مصنوعی

## شبکه‌های باور عمیق

هادی ویسی

[h.veisi@ut.ac.ir](mailto:h.veisi@ut.ac.ir)

دانشگاه تهران - دانشکده علوم و فنون نوین



## ○ شبکه باور عمیق

- ساختار
- آموزش
- کاربرد

## ○ ماشین بولتزمن

## ○ ماشین بولتزمن محدود

- (ساختار، آموزش، مثال و آزمون)

## ○ ماشین بولتزمن محدود گاوسی

- (ساختار، آموزش، مثال و آزمون)

## ○ استخراج ویژگی با DBN

- روش، مثال

## ○ جعبه ابزار DeeBNet Toolbox

## شبکه باور عمیق ...

### ○ شبکه باور عمیق (DBN: Deep Belief Network)

- متشکل از چند لایه ماشین بولتزمن محدود

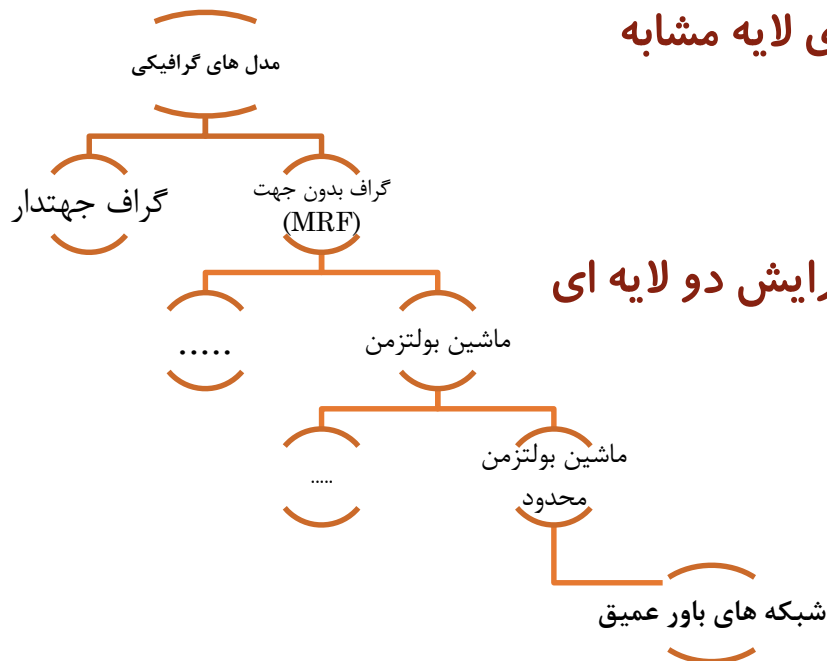
### ○ ماشین بولتزمن محدود (RBM: Restricted Boltzmann Machine)

- نوعی ماشین بولتزمن بدون اتصال واحدهای لایه مشابه

### ○ ماشین بولتزمن

- مدل گرافیکی بدون جهت

- مدل کردن وابستگی بین متغیرها با یک آرایش دو لایه ای



## کاربرد شبکه باور عمیق ...

○ بیشترین کاربرد این شبکه در دسته بندی و استخراج ویژگی است

• کاربردها

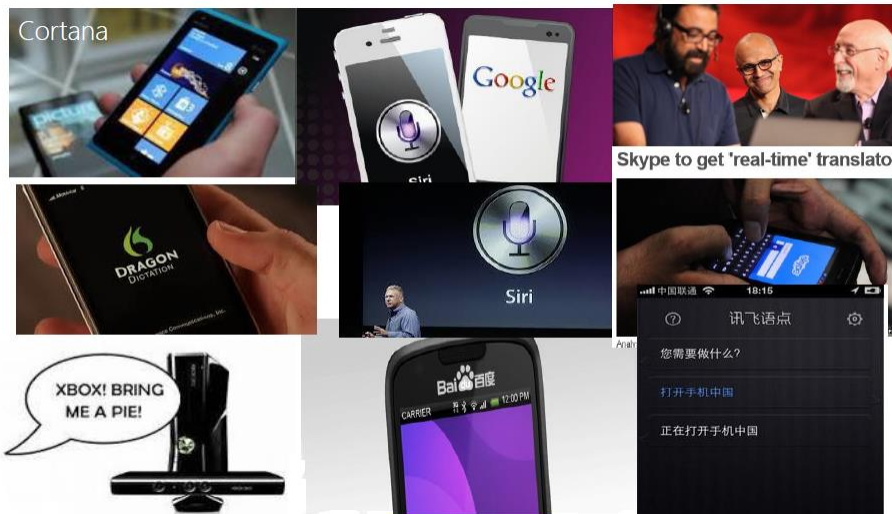
○ تشخیص نوع موسیقی

○ بازشناسی گفتار

○ مدل سازی آکوستیکی

○ نظارت ویدئویی هوشمند

○ بازشناسی تصویر





## مشکل آموزش شبکه‌های عمیق ...

### ○ محو شدن گرادیان (Gradient Vanishing)

- مشکل عمده آموزش شبکه‌های با ساختار عمیق است
  - در شبکه‌هایی که از الگوریتم‌های مبتنی بر گرادیان استفاده می‌کنند
  - مانند الگوریتم پس انتشار خطا (Back Propagation)
  - افزایش تعداد لایه‌ها باعث بیشتر شدن مشکل می‌گردد
- آموزش و تعیین پارامترها در لایه‌های اولیه شبکه به خوبی انجام نمی‌گیرد
- عمدتاً به دلیل نوع تابع فعالسازی می‌باشد
  - اکثر فعالسازها دامنه بزرگی از مقادیر ورودی را به دامنه کوچکی نگاشت می‌کنند مانند سیگموید
  - تشدید روند نگاشت به مقادیر کوچکتر در لایه‌های بعدی
  - تغییرات بزرگ در مقادیر پارامترها در لایه‌های اولیه موجب تغییرات کوچک در خروجی می‌شود
  - گرادیان کوچک خواهد بود

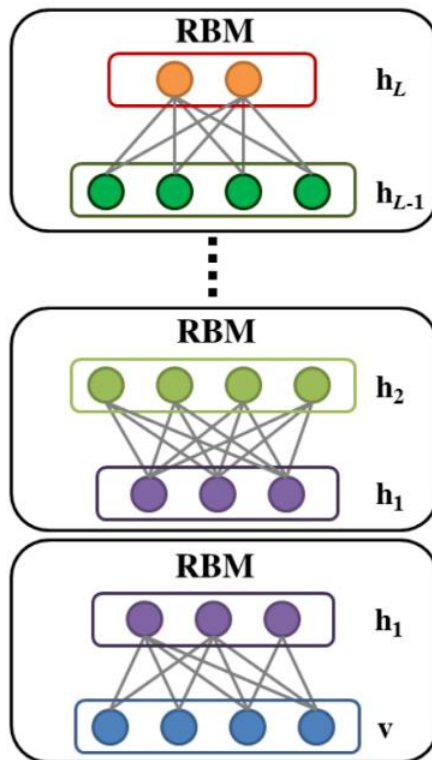
### ○ راه حل

- آموزش شبکه به صورت لایه به لایه
  - مانند آموزش شبکه باور عمیق

## شبکه باور عمیق ...

### ○ ساختار DBN

- یک شبکه عصبی چند لایه
- از پشته کردن تعدادی RBM حاصل می‌شود
- هر لایه آن یک RBM



## آموزش شبکه باور عمیق ...

### • آموزش شبکه شامل دو مرحله است

#### • پیش آموزش (Pre Training)

• آموزش RBM ها به صورت لایه به لایه با روش زیر انجام می‌شود:

• آموزش RBM اول

• استفاده از خروجی RBM اول (پس از آموزش) جهت آموزش RBM دوم

• در واقع ویژگی‌ها استخراج می‌شود

• ادامه این روند تا آموزش آخرین RBM

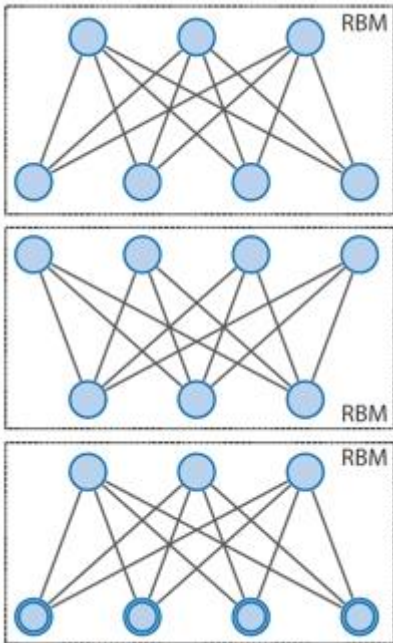
• حل شدن مشکل Vanishing Gradient آموزش شبکه‌های عمیق

• یک روش حریصانه (Greedy) است

#### • پس انتشار خطا (Back Propagation)

• استفاده از وزن‌های به دست آمده در مرحله pre-training

به عنوان وزن‌های اولیه جهت آموزش شبکه با الگوریتم پس انتشار خطا





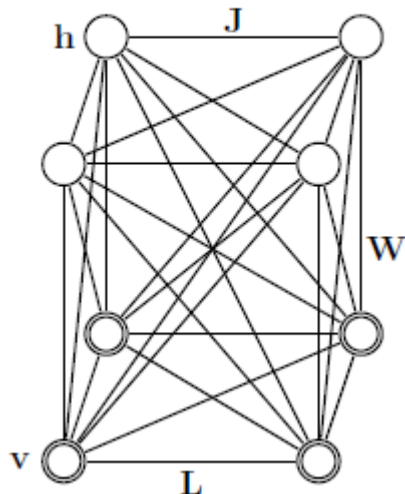
Geoff Hinton

## ماشین بولتزمن...

- اولین بار توسط Geoffrey Hinton در سال ۱۹۸۵ معرفی شد
- شبکه دولایه است
- وزن‌های بین این دو لایه (W) متقارن است
- شامل مجموعه‌ای از واحدهای قابل مشاهده **دودویی** و مجموعه‌ای از واحدهای مخفی **دودویی**

- وجود اتصالات بین نرون‌های لایه‌های مشابه (ماتریس‌های وزن J و L)
- انرژی حالت  $\{v, h\}$  در ماشین بولتزمن به صورت زیر است

General Boltzmann Machine



$$E(v, h) = -\frac{1}{2}v^T L v - \frac{1}{2}h^T J h - v^T W h$$

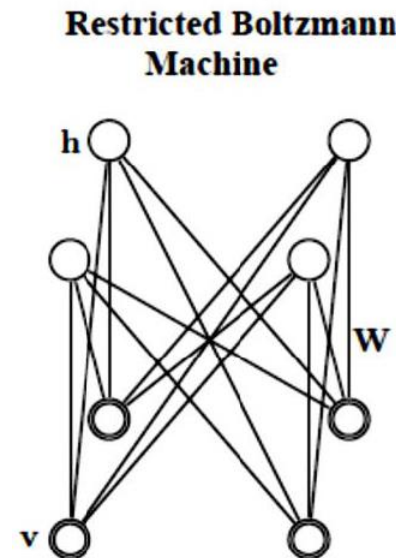
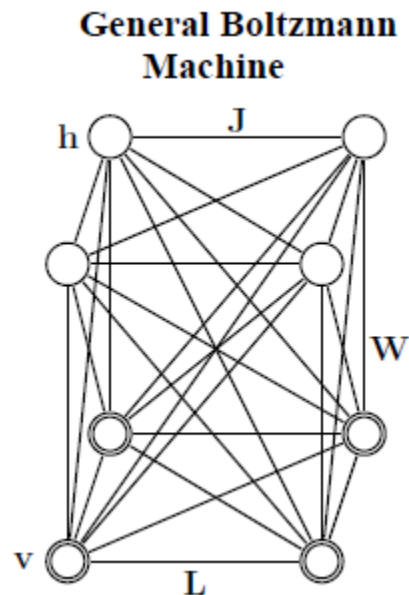
- به دلیل اتصالات زیاد عملاً نمی‌توان از این شبکه‌ها استفاده کرد

$$W_{ij} = W_{ji}$$



## ماشین بولتزمن محدود (RBM) ...

- نوعی ماشین بولتزمن که اتصالات بین واحدهای مشاهده‌پذیر و واحدهای پنهان قطع شده  
○  $J=0$  و  $L=0$
- شامل: لایه مشاهده‌پذیر ( $v$ ) و لایه پنهان ( $h$ )
- واحدهای مشاهده‌پذیر و واحدهای مخفی هر دو از نوع **دودویی** هستند
- وزن‌های بین این دو لایه متقارن می‌باشد





## ماشین بولتزمن محدود (RBM)...

### آموزش RBM

- انرژی حالت  $\{v, h\}$  در ماشین بولتزمن محدود به صورت زیر است

$$E(v, h) = - \sum_{i=1}^{g_v} \sum_{j=1}^{g_h} W_{ij} v_i h_j - \sum_{i=1}^{g_v} a_i v_i - \sum_{j=1}^{g_h} b_j h_j$$

تعداد واحدهای مشاهده پذیر  $g_v$

تعداد واحدهای پنهان  $g_h$

بایاس واحد مشاهده پذیر  $i$   $a_i$

بایاس واحد پنهان  $j$   $b_j$

- شبکه به هر حالت ممکن مقادیر بردارهای مشاهده‌پذیر و مخفی با تابع انرژی، یک مقدار احتمال نسبت می‌دهد

$$P(v, h) = \frac{1}{Z} \exp(-E(v, h))$$

- انرژی کمتر = احتمال بیشتر

$$Z = \sum_v \sum_h \exp(-E(v, h))$$

ثابت نرمال‌سازی

## ماشین بولتزمن محدود (RBM)...

- احتمالی که مدل به بردار قابل مشاهده ( $v$ ) نسبت می‌دهد برابر است با

$$P(v) = \sum_h P(v, h) = \frac{1}{Z} \sum_h \exp(-E(v, h))$$

جمع تمام حالات ممکن بردارهای مخفی

- تابع هدف به صورت زیر تعریف می‌شود:

جهت رسیدن به انرژی کمتر برای داده‌های آموزشی و انرژی بیشتر برای سایر داده‌ها

$$\text{maximize}_{\{w_{ij}, a_i, b_j\}} \frac{1}{m} \sum_{l=1}^m \log \left( \sum_h P(v^{(l)}, h^{(l)}) \right)$$

$i$  امین بعد داده آموزشی  $l$

تعداد نمونه های آموزشی

$$\frac{\partial}{\partial w_{ij}} \left( \frac{1}{m} \sum_{l=1}^m \log \left( \sum_h P(v^{(l)}, h^{(l)}) \right) \right) = \frac{1}{m} \sum_{l=1}^m \sum_h x_{il} h_j P(h|v=x) - \sum_{v'} \sum_{h'} v'_i h'_j P(v', h')$$

غیر قابل محاسبه است



## ماشین بولتزمن محدود (RBM)...

- با توجه به غیر قابل محاسبه بودن مشتق تابع هدف، این مقدار به صورت زیر تخمین زده می‌شود

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

فاز مثبت

فاز منفی

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

- قانون اصلاح وزن‌ها به صورت زیر محاسبه می‌گردد

$$\Delta w_{ij} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$$

$$\Delta a_i = \epsilon (\langle v_i \rangle_{data} - \langle v_i \rangle_{model})$$

$$\Delta b_j = \epsilon (\langle h_j \rangle_{data} - \langle h_j \rangle_{model})$$

## ماشین بولتزمن محدود (RBM)...

محاسبه  $\langle v_i h_j \rangle_{data}$

- حالت دودویی هر واحد مخفی با شرط داشتن واحدهای مشاهده پذیر به احتمال زیر یک می شود. در صورتیکه این مقدار از یک عدد تصادفی در بازه  $[0-1]$  بزرگتر بود برابر یک و در غیر این صورت برابر صفر است.

تابع سیگموئید

$$P(h_j = 1 | \mathbf{v}) = \varphi \left( b_j + \sum_i v_i w_{ij} \right)$$

به دلیل نبود اتصال بین واحدهای مخفی این واحدها به شرط واحد مشاهده پذیر مستقل هستند

- حالت دودویی هر واحد مخفی با شرط داشتن واحدهای مشاهده پذیر به احتمال زیر یک می شود. در صورتیکه این مقدار از یک عدد تصادفی در بازه  $[0-1]$  بزرگتر بود برابر یک و در غیر این صورت برابر صفر است.

$$P(v_i = 1 | \mathbf{h}) = \varphi \left( a_i + \sum_j h_j w_{ij} \right)$$

به دلیل نبود اتصال بین واحدهای مشاهده پذیر این واحدها به شرط واحد پنهان مستقل هستند

- با ضرب این دو مقدار  $\langle v_i h_j \rangle_{data}$  محاسبه می گردد.



## ماشین بولتزمن محدود (RBM)...

محاسبه  $\langle v_i h_j \rangle_{model}$

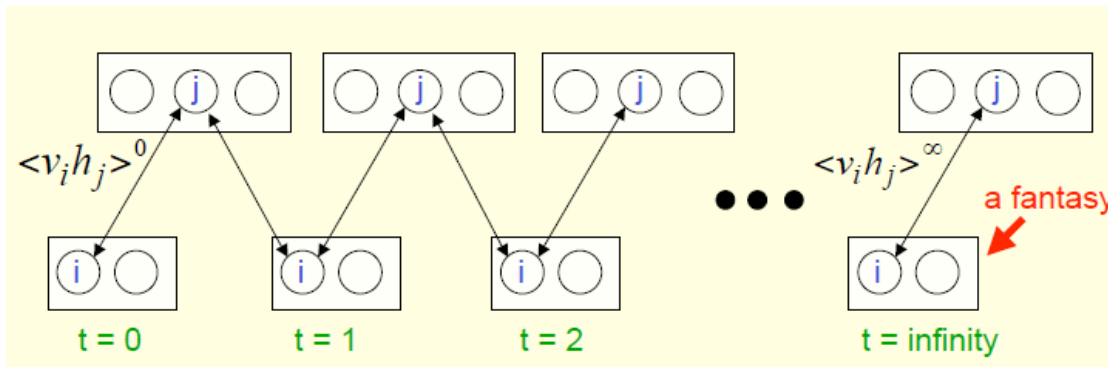
- جهت محاسبه این مقدار می توان از روش های نمونه برداری استفاده کرد
- نمونه برداری گیبز
- سرعت پایین و عملا غیر ممکن
- روش واگرایی متقابل
- دقیقا معادل گرادیان نیست ولی دقت قابل قبولی دارد

## ماشین بولتزمن محدود (RBM)...

### روش نمونه برداری گیبز (Gibbs Sampling)

- ابتدا قرار دادن یک بردار از داده آموزشی در لایه مشاهده پذیر
- تکرار مراحل زیر به صورت نامحدود

- بروز رسانی تمام واحدهای مخفی به صورت موازی
  - بروز رسانی تمام واحدهای مشاهده پذیر به صورت موازی
  - دوباره بروز رسانی واحدهای مخفی
- یک مرحله از نمونه برداری گیبز



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

مرحله صفر از نمونه برداری گیبز و معادل امید ریاضی ضرب  $v_i$  و  $h_j$  می باشد

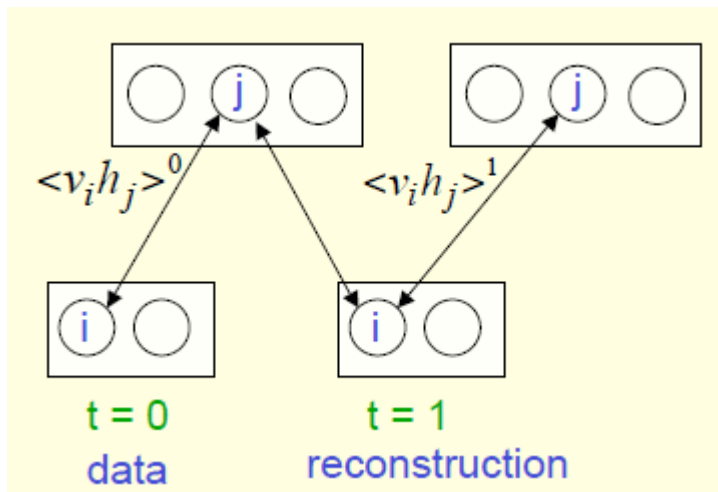
## ماشین بولتزمن محدود (RBM)...

### روش واگرایی متقابل (Contrastive Divergence)

اجرای نمونه برداری گیبز تا گام زمانی  $t=1$

- قرار دادن یک بردار از ورودی در لایه مشاهده پذیر
- بروز رسانی تمام واحدهای مخفی به صورت موازی
- بروز رسانی تمام واحدهای مشاهده پذیر به صورت موازی
- دوباره بروز رسانی واحدهای مخفی
- و در نهایت اصلاح وزن ها:

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



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

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

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





## ماشین بولتزمن محدود (RBM)...

### ○ الگوریتم آموزش واگرایی متقابل

- گام ۰ - مقداردهی اولیه وزن‌ها به صورت تصادفی و صفر قرار دادن مقادیر بایاس
- گام ۱ - برای هر بردار ورودی گام‌های ۲ تا ۹ را تکرار کنید
- گام ۲ - یک بردار ورودی ( $v$ ) را در لایه مشاهده پذیر قرار دهید.
- گام ۳ - احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

تابع سیگموید

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

$Positive_{hidden}$

- گام ۴ - مقدار واحدهای 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  $h$ )
- گام ۵ - احتمال فعال شدن واحدهای مشاهده‌پذیر را به شرط بردار  $h$  محاسبه نمایید

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



## ماشین بولتزمن محدود (RBM)...

### الگوریتم آموزش (ادامه...)

- گام ۶- در صورتی که احتمال به دست آمده بزرگتر از یک مقدار تصادفی در بازه  $[0,1]$  بود، مقدار واحدهای مشاهده‌پذیر را برابر یک و در غیر اینصورت برابر صفر قرار دهید. (بردار  $v'$ )

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

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

*Negative<sub>hidden</sub>*

- گام ۸- اگر نرخ یادگیری برابر  $\alpha$  باشد تغییرات وزن را به کمک رابطه‌های زیر محاسبه کنید.

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

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

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



## ماشین بولتزمن محدود (RBM)...

○ الگوریتم آموزش (ادامه...)

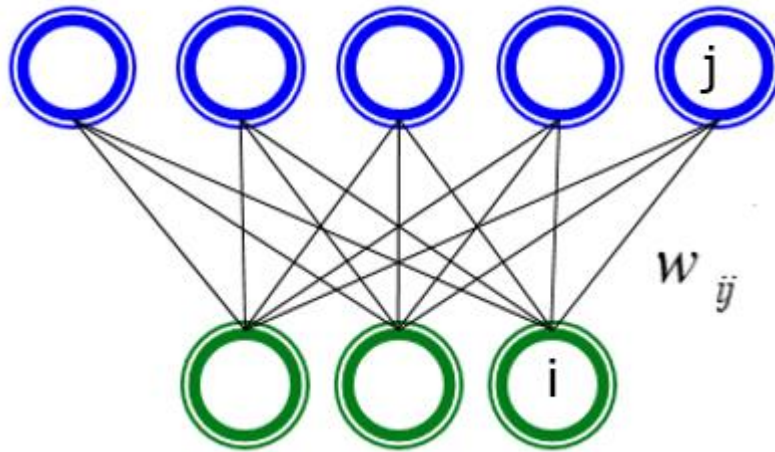
• گام ۹- وزن‌ها را بروز رسانی کنید.

$$W = W + \Delta W$$

$$b = b + \Delta b$$

$$a = a + \Delta a$$

## ماشین بولتزمن محدود (مثال)...



### ○ ساختار شبکه RBM

- تعداد نرون‌های لایه مشاهده پذیر: ۵
  - از نوع دودویی
- تعداد نرون‌های لایه پنهان: ۷
  - از نوع دودویی
- نرخ یادگیری: ۰.۱
- وزن‌های اولیه: مقادیر کوچک تصادفی از یک گوسی
  - میانگین صفر و انحراف معیار ۰.۰۱
- مقادیر اولیه بایاس: صفر



## ماشین بولتزمن محدود (مثال)...

- گام ۰ - بردار ورودی را به صورت زیر در نظر بگیرید

1	0	1	1	0
---	---	---	---	---

- گام ۱ - وزن‌ها و بایاس را مقدار دهی اولیه کنید

$W =$

	1	2	3	4	5	6	7
1	-0.0226	0.1101	-0.1062	0.0889	-0.0177	0.1588	0.0216
2	0.1117	0.1544	0.2350	-0.0765	-0.0196	-0.0804	-0.1166
3	-0.1089	0.0086	-0.0616	-0.1402	0.1419	0.0697	-0.1148
4	0.0033	-0.1492	0.0748	-0.1422	0.0292	0.0835	0.0105
5	0.0553	-0.0742	-0.0192	0.0488	0.0198	-0.0244	0.0722

$a =$

1	2	3	4	5
0	0	0	0	0

$b =$

1	2	3	4	5	6	7
0	0	0	0	0	0	0

*Positive<sub>hidden</sub>*

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

## ماشین بولتز من محدود (مثال)...

- گام ۲- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

$$P(h_j = 1|v) =$$

	1	2	3	4	5	6	7
	0.4680	0.4924	0.4768	0.4517	0.5383	0.5774	0.4793

$$Positive_{Hidden} = \sigma [ ((-0.0226*1)+0) + ((0.1117*0)+0) + ((-0.1089*1)+0) + ((0.0033*1)+0) + ((0.0553*0)+0) ] = \sigma (-0.1282) = 0.4680$$

- گام ۳- مقدار واحدهای پنهان را در صورتی که احتمال به دست آمده بزرگتر از یک تصادفی در بازه  $[0,1]$  بود برابر یک و در غیر اینصورت برابر صفر قرار دهید (بردار  $h$ )

$$P(h_j = 1|v)$$

1	2	3	4	5	6	7
0.4680	0.4924	0.4768	0.4517	0.5383	0.5774	0.4793

Λ V Λ Λ V V V

1	2	3	4	5	6	7
0.9340	0.1299	0.5688	0.4694	0.0119	0.3371	0.1622

بردار  $h$

1	2	3	4	5	6	7
0	1	0	0	1	1	1

بردار تصادفی



## ماشین بولتز من محدود (مثال)...

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

- گام ۴- احتمال فعال شدن واحدهای مشاهده‌پذیر را به شرط بردار  $h$  محاسبه نمایید

$P(v_i = 1 h)$	1	2	3	4	5
	0.5677	0.4845	0.5263	0.4935	0.4984

$$\sigma [ ((-0.0226*0)+0) + ((0.1101*1)+0) + ((-0.1062*0)+0) + ((0.0889*0)+0) + ((-0.0177*1)+0) + ((0.1588*1)+0) + ((0.0216*1)+0) ] = \sigma (0.2728) = 0.5677$$

- گام ۵- در صورتی که احتمال به‌دست آمده بزرگتر از یک مقدار تصادفی در بازه  $[0,1]$  بود، مقدار واحدهای مشاهده‌پذیر را برابر یک و در غیر اینصورت برابر صفر قرار دهید. (بردار  $v'$ )

$$P(v_i = 1|h)$$

1	2	3	4	5
0.5677	0.4845	0.5263	0.4935	0.4984
$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$
1	2	3	4	5
0.7943	0.3112	0.5285	0.1656	0.6020

1	2	3	4	5
0	1	0	1	0

بردار تصادفی



## ماشین بولتزمن محدود (مثال)...

$Negative_{hidden}$

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

- گام ۶- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v'$  محاسبه نمایید.

$$P(h'_j = 1|v') =$$

	1	2	3	4	5	6	7
	0.5287	0.5013	0.5768	0.4455	0.5024	0.5008	0.4735

$$Negative_{Hidden} = \sigma [ ((-0.0226*0)+0) + ((0.1117*1)+0) + ((-0.1089*0)+0) + ((0.0033*1)+0) + ((0.0553*0)+0) ] = \sigma (0.115) = 0.5287$$

- گام ۷- اگر نرخ یادگیری برابر  $\alpha$  باشد تغییرات وزن را به کمک رابطه‌های زیر محاسبه کنید.

$$\Delta W =$$

	1	2	3	4	5	6	7
1	0.0468	0.0492	0.0477	0.0452	0.0538	0.0577	0.0479
2	-0.0529	-0.0501	-0.0577	-0.0446	-0.0502	-0.0501	-0.0474
3	0.0468	0.0492	0.0477	0.0452	0.0538	0.0577	0.0479
4	-0.0061	-8.9412e-04	-0.0100	6.2134e-04	0.0036	0.0077	5.8260e-04
5	0	0	0	0	0	0	0

$$\Delta W_{11} = 0.1 * [(1 * 0.4680) - (0 * 0.5287)] = 0.0468$$

$$\Delta W = \alpha (v^T * Positive_{Hidden} - v'^T * Negative_{Hidden})$$





## ماشین بولتزمن محدود (مثال)...

$$\Delta a_1 = 0.1 * [1 - 0] = 0.1$$

$$\Delta a =$$

1	2	3	4	5
0.1000	-0.1000	0.1000	0	0

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

$$\Delta b = \alpha(Positive_{Hidden} - Negative_{Hidden})$$

$$\Delta b =$$

1	2	3	4	5	6	7
-0.0061	-8.9412e-04	-0.0100	6.2134e-04	0.0036	0.0077	5.8260e-04

$$\Delta b_1 = 0.1 * [0.4680 - 0.5287] = -0.0061$$

- گام ۸- وزن‌ها را بروز رسانی کنید.

$$W = W + \Delta W$$

$$b = b + \Delta b$$

$$a = a + \Delta a$$



## ماشین بولتزمن محدود (آزمون) ...

### ○ الگوریتم آزمون (کاربرد)

- گام ۱- برای هر بردار ورودی آزمون گام‌های ۲ تا ۴ را تکرار کنید
- گام ۲- یک بردار ورودی ( $v$ ) را در لایه مشاهده پذیر قرار دهید.
- گام ۳- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

تابع سیگموئید

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

$Positive_{hidden}$

- گام ۴- مقدار واحدهای 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

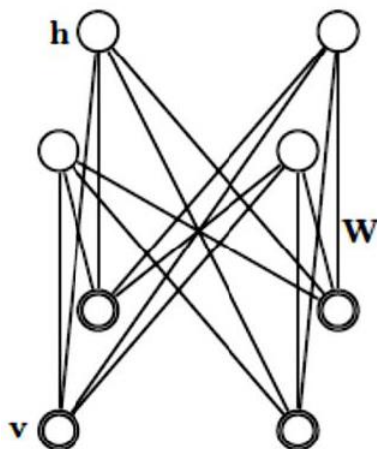
## ماشین بولتزمن محدود گاوسی (GRBM)...

- نوعی ماشین بولتزمن که اتصالات بین واحدهای مشاهده‌پذیر و واحدهای پنهان قطع شده
- رایج‌ترین رویکرد برای مدل کردن مشاهدات با **مقادیر حقیقی**، در قالب ماشین بولتزمن محدود، نوع گاوسی آن است

○ کاربرد در صوت و تصویر

- واحدهای مشاهده‌پذیر از نوع **حقیقی** و واحدهای مخفی از نوع **دودویی** هستند
- شامل دو لایه مشاهده‌پذیر و لایه پنهان
- وزن‌های بین این دو لایه متقارن می‌باشد

Restricted Boltzmann Machine



$$W_{ij} = W_{ji}$$



# ماشین بولتزمن محدود گاوسی (آموزش)...

## آموزش...

- انرژی حالت  $\{v, h\}$  در ماشین بولتزمن محدود گاوسی به صورت زیر است

$$E(v, h) = - \sum_{i=1}^{g_v} \sum_{j=1}^{g_h} W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{i=1}^{g_v} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{g_h} b_j h_j$$

تعداد واحدهای مشاهده پذیر

انحراف معیار مولفه  $i$  بردار داده‌ها

بایاس واحد مشاهده پذیر  $i$

تعداد واحدهای پنهان

بایاس واحد پنهان  $j$

- شبکه به هر حالت ممکن مقادیر بردارهای مشاهده‌پذیر و مخفی با تابع انرژی، یک مقدار احتمال نسبت می‌دهد

$$P(v, h) = \frac{1}{Z} \exp(-E(v, h))$$

• انرژی کمتر = احتمال بیشتر

$$Z = \sum_v \sum_h \exp(-E(v, h))$$

ثابت نرمال‌سازی



## ماشین بولتزمن محدود گاوسی (آموزش)...

- توزیع حاشیه ای بر روی بردار مشاهدات به صورت زیر خواهد بود

○ مشابه RBM

$$P(v) = \sum_h P(v, h) = \frac{1}{Z} \sum_h \exp(-E(v, h))$$

- ثابت نرمالسازی به دلیل پیوسته بودن مقادیر مشاهدات به رابطه زیر تغییر می‌کند

$$Z = \int_v \sum_h \exp(-E(v, h)) dv'$$

- احتمال شرطی واحدهای مشاهده پذیر و مخفی به صورت زیر تغییر می‌کند

○ هر واحد مشاهده پذیر با یک گاوسی مدل می‌شود

$$p(v_i = x_i | h) = \frac{1}{\sqrt{2\pi} \sigma_i} \text{Exp}\left(-\frac{(x_i - a_i - \sigma_i \sum_j h_j w_{ij})^2}{2\sigma_i^2}\right), \quad p(h_j = 1 | v) = g\left(b_j + \sum_i w_{ij} \frac{v_i}{\sigma_i}\right)$$



## ماشین بولتزمن محدود گاوسی (آموزش)...

- مشتق احتمال بردار مشاهده پذیر نسبت به بردار وزن به صورت زیر تخمین زده می‌شود

فاز مثبت

فاز منفی

$$\frac{\partial \log P(v)}{\partial w_{ij}} = \left\langle \frac{1}{\sigma_i} v_i h_j \right\rangle_{data} - \left\langle \frac{1}{\sigma_i} v_i h_j \right\rangle_{model}$$

<>: امید ریاضی بر روی ضرب مقادیر مشاهده پذیر و مخفی

- $\sigma_i$  یک عدد ثابت در نظر گرفته می‌شود و بر روی آن آموزش صورت نمی‌گیرد
- اگر  $\sigma_i$  معادل یک در نظر گرفته شود روش اصلاح وزن‌ها تغییری نمی‌کند
- (با نرمالسازی داده ورودی این مقدار معادل یک می‌شود)

- قانون اصلاح وزن‌ها به صورت زیر محاسبه می‌گردد

$$\Delta w_{ij} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$$

$$\Delta a_i = \epsilon (\langle v_i \rangle_{data} - \langle v_i \rangle_{model})$$

$$\Delta b_j = \epsilon (\langle h_j \rangle_{data} - \langle h_j \rangle_{model})$$

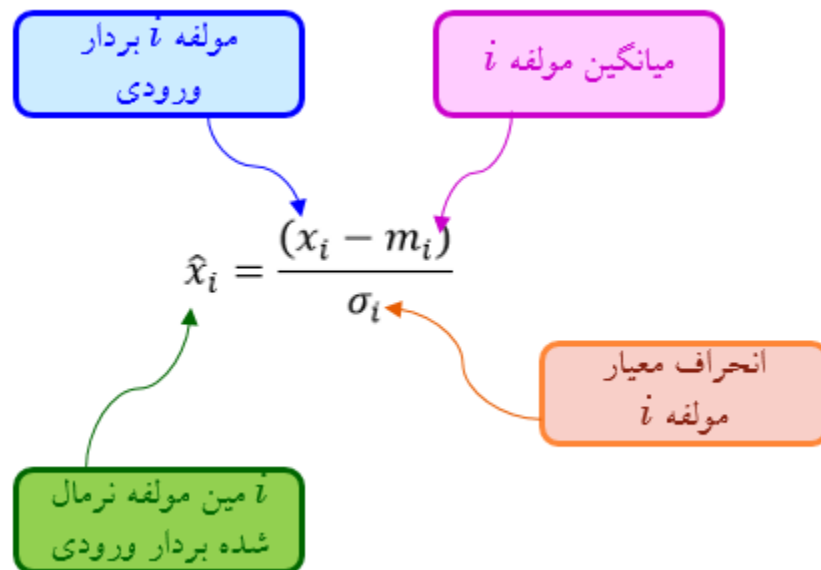
# ماشین بولتزمن محدود گاوسی (آموزش)...

## الگوریتم آموزش

### گام ۰ - نرمال سازی داده ورودی

هر مولفه ورودی به برداری با میانگین صفر و انحراف معیار یک تبدیل می شود

1. ابتدا میانگین هر مولفه بردار ورودی را محاسبه کنید.
2. سپس انحراف معیار هر مولفه بردار ورودی را محاسبه کنید
3. هر مولفه را به کمک رابطه زیر نرمال کنید





# ماشین بولتزمن محدود گاوسی (آموزش)...

## الگوریتم آموزش

- گام ۱ - مقداردهی اولیه وزن‌ها به صورت تصادفی و صفر قرار دادن مقادیر بایاس
- گام ۲ - برای هر بردار ورودی گام‌های ۲ تا ۹ را تکرار کنید
- گام ۳ - یک بردار ورودی ( $v$ ) را در لایه مشاهده پذیر قرار دهید.
- گام ۴ - احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

تابع سیگموئید

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

$Positive_{hidden}$

- گام ۵ - مقدار واحدهای 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  $h$ )

- گام ۶ - احتمال فعال شدن واحدهای مشاهده‌پذیر را به شرط بردار  $h$  محاسبه نمایید و بردار حاصل را ( $v'$ ) بنامید

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

بردار  $v'$





## ماشین بولتزمن محدود گاوسی (آموزش)...

### ○ الگوریتم آموزش (ادامه...)

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

$$P(h_j = 1 | v') = \sigma(b_j + \sum_{i=1}^m w_{ij} v'_i)$$

Negative<sub>hidden</sub>

- گام ۸- اگر نرخ یادگیری برابر  $\alpha$  باشد تغییرات وزن را به کمک رابطه‌های زیر محاسبه کنید.

$$\Delta W = \alpha(v^T * Positive_{Hidden} - v'^T * Negative_{Hidden})$$

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

$$\Delta b = \alpha(Positive_{Hidden} - Negative_{Hidden})$$

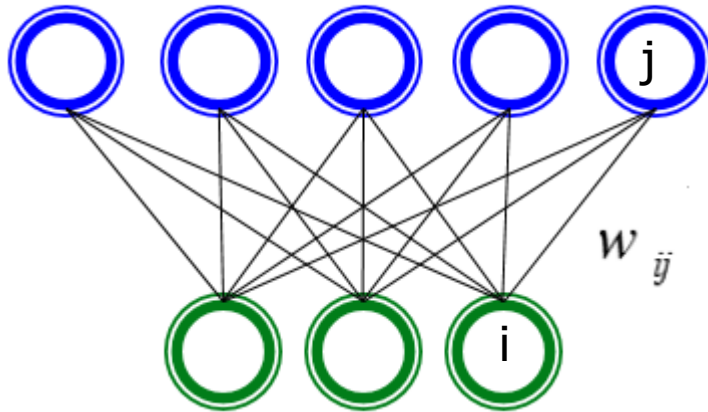
- گام ۹- وزن‌ها را بروز رسانی کنید.

$$W = W + \Delta W$$

$$b = b + \Delta b$$

$$a = a + \Delta a$$

## ماشین بولتزمن محدود (مثال)...



### ○ ساختار شبکه GRBM

- تعداد نرون‌های لایه مشاهده پذیر: ۳
  - از نوع حقیقی
- تعداد نرون‌های لایه پنهان: ۵
  - از نوع دودویی
- نرخ یادگیری: ۰.۰۰۱
- وزن‌های اولیه: مقادیر کوچک تصادفی از یک گوسی
  - میانگین صفر و انحراف معیار ۰.۰۱
- مقادیر اولیه بایاس: صفر



# ماشین بولتزمن محدود گاوسی (مثال)...

○ مثال...

• گام ۰ - نرمال‌سازی داده ورودی

$$\frac{3.43 - (-3.2368)}{3.9384} = 1.6928$$

	1	2	3
1	-0.7379	-0.0963	1.6928
2	1.8242	0.9224	0.2515
3	-1.2349	-1.9055	-0.3282
4	0.9710	0.3600	0.5710
5	-0.1694	0.6708	-1.9009
6	0.1641	0.7021	0.6457
7	0.3885	0.7274	-0.1718
8	-1.2056	-1.3810	-0.7601

داده نرمال شده

کل داده ورودی

	1	2	3
1	-36.2809	-5.1105	3.4300
2	-31.3580	-2.9480	-2.2462
3	-37.2360	-8.9512	-4.5293
4	-32.9975	-4.1419	-0.9878
5	-35.1886	-3.4821	-10.7233
6	-34.5479	-3.4156	-0.6938
7	-34.1167	-3.3621	-3.9135
8	-37.1795	-7.8377	-6.2302

بردار ورودی

بردار میانگین مولفه‌ها

1	2	3
-34.8631	-4.9062	-3.2368

بردار انحراف معیار مولفه‌ها

1	2	3
1.9214	2.1228	3.9384



## ماشین بولتز من محدود گاوسی (مثال)...

- گام ۰ - بردار ورودی را به صورت زیر در نظر بگیرید

1	2	3
-0.7379	-0.0963	1.6928

- گام ۱ - وزن‌ها و بایاس را مقدار دهی اولیه کنید

$W =$

	1	2	3	4	5
1	0.1355	0.0124	-0.0198	0.0825	-0.0469
2	-0.1072	0.1437	-0.1208	0.1379	-0.0272
3	0.0961	-0.1961	0.2908	-0.1058	0.1098

$a =$

	1	2	3
1	0	0	0

$b =$

	1	2	3	4	5
1	0	0	0	0	0



## ماشین بولتز من محدود گاوسی (مثال)...

**Positive<sub>hidden</sub>**

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

- گام ۲- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

$$P(h_j = 1|v) =$$

	1	2	3	4	5
1	0.5183	0.4122	0.6268	0.4370	0.5556

$$Positive_{Hidden} = \sigma [ ((0.1355 * (-0.7379)) + 0) + ((-0.1072 * (-0.0963)) + 0) + ((0.0961 * 1.6928) + 0) = \sigma(0.073) = 0.5183$$

- گام ۳- مقدار واحدهای 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  $h$ )

$$P(h_j = 1|v)$$

	1	2	3	4	5
1	0.5183	0.4122	0.6268	0.4370	0.5556

V      ^      V      ^      V

	1	2	3	4	5
1	0.4314	0.5499	0.2399	0.4909	0.2417

بردار  $h$

	1	2	3	4	5
1	1	0	1	0	1

بردار تصادفی



## ماشین بولتز من محدود گاوسی (مثال)...

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

- گام ۴- احتمال فعال شدن واحدهای مشاهده‌پذیر را به شرط بردار  $h$  محاسبه نمایید و بردار حاصل را  $(v')$  بنامید

$$P(v_i = 1|h) =$$

	1	2	3
1	0.0688	-0.2552	0.4967

$$[(0.1355*1)+0] + [(0.0124*0)+0] + [(-0.0198*1)+0] + [(0.0825*0)+0] + [(-0.0469*1)+0] = 0.688$$

- گام ۵- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v'$  محاسبه نمایید.

$$P(h'_j = 1|v') =$$

	1	2	3	4	5
1	0.5211	0.4667	0.5434	0.4795	0.5146

$$Negative_{Hidden} = \sigma [ ((0.1355*(0.0688))+0) + ((-0.1072*(-0.2552))+0) + ((0.0961*0.4967)+0) ] = \sigma (0.0844) = 0.5211$$

$Negative_{hidden}$

$$P(h_j = 1|v') = \sigma(b_j + \sum_{i=1}^m w_{ij} v'_i)$$



## ماشین بولتزمن محدود گاوسی (مثال)...

- گام ۷- اگر نرخ یادگیری برابر  $\alpha$  باشد تغییرات وزن را به کمک رابطه‌های زیر محاسبه کنید.

$\Delta W =$

	1	2	3	4	5
1	-4.1827e-04	-3.3627e-04	-4.9989e-04	-3.5547e-04	-4.4535e-04
2	8.3119e-05	7.9456e-05	7.8357e-05	8.0320e-05	7.7863e-05
3	6.1842e-04	4.6589e-04	7.9109e-04	5.0157e-04	6.8481e-04

$$\Delta W_{11} = 0.001 * [(-0.7379 * 0.5183) - (0.0688 * 0.5211)] = -4.18e-04$$

$\Delta a =$

	1	2	3
1	-8.0671e-04	1.5898e-04	0.0012

$$\Delta a_1 = 0.001 * [(-0.7379) - (0.0688)] = -8.0671e-04$$

$$\Delta W = \alpha (v^T * Positive_{Hidden} - v'^T * Negative_{Hidden})$$

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

$$\Delta b = \alpha (Positive_{Hidden} - Negative_{Hidden})$$

$\Delta b =$

	1	2	3	4	5
1	-2.8431e-06	-5.4552e-05	8.3419e-05	-4.2477e-05	4.0985e-05

$$\Delta b_1 = 0.001 * [(0.5183) - (0.5211)] = -2.8431e-06$$



## ماشین بولتزمن محدود گاوسی (مثال)...

- گام ۸- وزن‌ها را بروز رسانی کنید.

$$W = W + \Delta W$$

$$b = b + \Delta b$$

$$a = a + \Delta a$$





## ماشین بولتزمن محدود گاوسی (آزمون)...

### ○ الگوریتم آزمون (کاربرد)

- گام ۱- برای هر بردار ورودی آزمون گام‌های ۲ تا ۴ را تکرار کنید
- گام ۲- یک بردار ورودی ( $v$ ) را در لایه مشاهده پذیر قرار دهید.
- گام ۳- احتمال فعال شدن واحدهای پنهان را به شرط بردار  $v$  محاسبه نمایید

تابع سیگموئید

$Positive_{hidden}$

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

- گام ۴- مقدار واحدهای 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

## شبکه باور عمیق خود رمز گذار...

### ○ استخراج ویژگی با DBN Autoencoder

- در این Autoencoder هدف کاهش غیر خطی بُعد داده‌ها است

○ آموزش به صورت Generative است

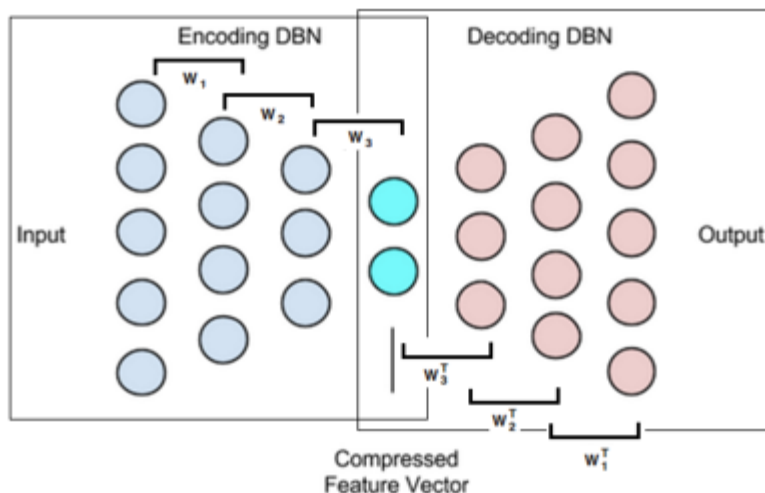
○ بدون برچسب گذاری

- شبکه شامل دو بخش: رمز گذار (Encoding) و رمز گشا (Decoding)

○ رمز گذار: کد کردن داده ورودی

○ رمز گشا: ساخت دوباره داده ورودی به کمک داده کد شده

- تعداد نرون‌های لایه ورودی با تعداد نرون‌های لایه خروجی برابر است



## شبکه باور عمیق خود رمز گذار

### الگوریتم استخراج ویژگی...

- آموزش بخش رمز گذار (Encoding)

- آموزش با الگوریتم حریصانه شبکه باور عمیق

- به صورت لایه به لایه

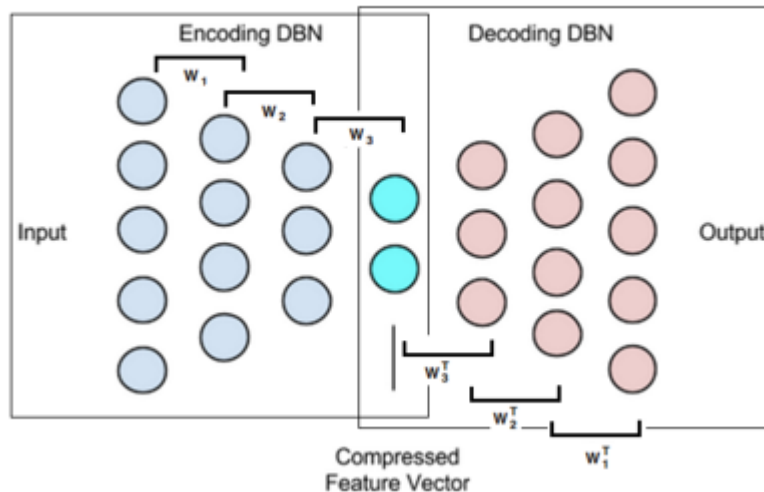
- قرار دادن ترانهاده وزن‌های بخش رمز گذار به عنوان وزن بخش رمز گشا (Decoding)

- آموزش کل شبکه خود رمز گذار (شامل بخش رمز گذار و رمز گشا)

- با استفاده از الگوریتم پس انتشار خطا استاندارد

- ارائه داده ورودی به بخش رمز گذار شبکه

- ویژگی‌های استخراج شده از این بخش ویژگی‌های مورد نظر می‌باشد





# استخراج ویژگی با DBN Autoencoder

## مجموعه دادگان

- فارس دات

## روش استخراج ویژگی

- شامل دو مرحله:

○ استخراج ویژگی‌های MFCC از هر فریم

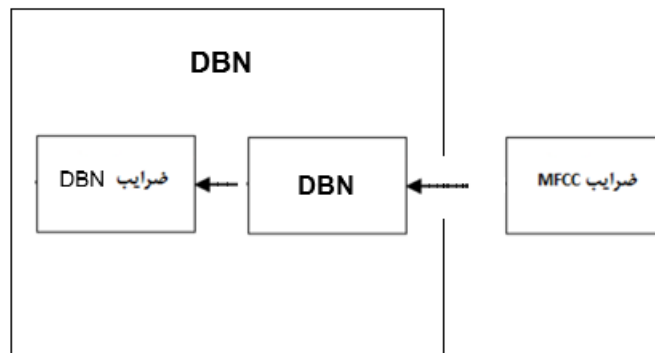
○ از هر فریم ۳۹ ویژگی استخراج می شود

○ استفاده از DBN Autoencoder

○ ورودی: ویژگی های MFCC هر فریم به همراه ۴ فریم بعدی

○ شامل ۴ ماشین بولتزمن محدود

○ خروجی: ۳۹ ویژگی





# استخراج ویژگی با DBN Autoencoder

- استخراج ویژگی‌های MFCC از هر فریم

طول فریم	۱۶ میلی ثانیه
تعداد فیلترهای مل	۲۶
نوع پنجره گذاری	همینگ
میزان هم پوشانی فریم ها	۸ میلی ثانیه
تعداد ویژگی های MFCC	۳۹

- استفاده از DBN Autoencoder

پارامتر	مقدار
داده ورودی DBN	ویژگی‌های MFCC هر فریم به همراه ۴ فریم بعدی
تعداد نرون‌های لایه ورودی	۱۹۵
تعداد نرون‌های RBM اول	۱۰۲۴
تعداد نرون‌های RBM دوم	۵۱۲
تعداد نرون‌های RBM سوم	۲۵۶
تعداد نرون‌های RBM چهارم	۳۹
تعداد ویژگی‌های استخراج شده	۳۹



# استخراج ویژگی با DBN Autoencoder

## ○ نتایج به دست آمده

○ استفاده از DBN جهت استخراج ویژگی موجب بهبود دقت شده است

روش استخراج ویژگی	تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
MFCC	۲۵۰	۰.۰۰۰۳	۵۶	۷۷
DBN	۲۵۰	۰.۰۰۰۳	۵۰	<u>۷۸</u>



## جعبه ابزار DeeBNet Toolbox

- جهت استفاده از شبکه DBN جعبه ابزار DeeBNet موجود می‌باشد

○ دانلود جعبه ابزار و راهنما

- <http://ceit.aut.ac.ir/~keyvanrad/DeeBNet%20Toolbox.html>

- جعبه ابزار شامل:

○ تعدادی نمونه کد آماده جهت آشنایی با روش استفاده از جعبه ابزار

○ فایل راهنمای استفاده

○ تعدادی کلاس و تابع برای کار با داده‌ها و نمونه برداری

○ تعدادی کلاس برای تعریف RBM و DBN



## معرفی برخی کلاس‌های مهم DeeBNet Toolbox

### ○ کلاس `ValueType`

• نوع واحدهای موجود در یک لایه را مشخص می‌کند

- `ValueType.binary` واحدها مقادیر ۰ و ۱ را می‌تواند بگیرد %
- `ValueType.probability` هر مقدار حقیقی بین ۰ و ۱ %
- `ValueType.Gaussian` هر مقدار حقیقی با میانگین ۰ و واریانس ۱ %

○ مثال

- `rbmParams=RbmParameters(Hid,ValueType.binary);`





## معرفی برخی کلاس‌های مهم DeeBNet Toolbox

### ○ کلاس RbmType

- نوع RBM را مشخص می‌کند

○ مثال: جهت استخراج ویژگی

○ `rbmParams.rbmType=RbmType.generative;`

- طریقه‌ی تعریف برخی پارامترهای مهم

○ نرخ یادگیری

○ `rbmParameters.learningRate=0.07;`

○ تعداد تکرارهای آموزش

○ `rbmParams.maxEpoch=50;`

○ تعیین روش نمونه برداری

○ `rbmParams.samplingMethodType=SamplingClasses.SamplingMethodType.CD;`



## معرفی برخی کلاس‌های مهم DeeBNet Toolbox

### ○ کلاس rbmParams

- همه ی پارامترهای RBM در این کلاس تعریف می گردد

#### RbmParameters

- +weight
- +visBias
- +hidBias
- +numHid
- +moment
- +learningRate
- +batchSize
- +maxEpoch
- +epochNumber
- +avgLast
- +penalty
- +samplingMethodType
- +kSamplingIteration
- +rbmType



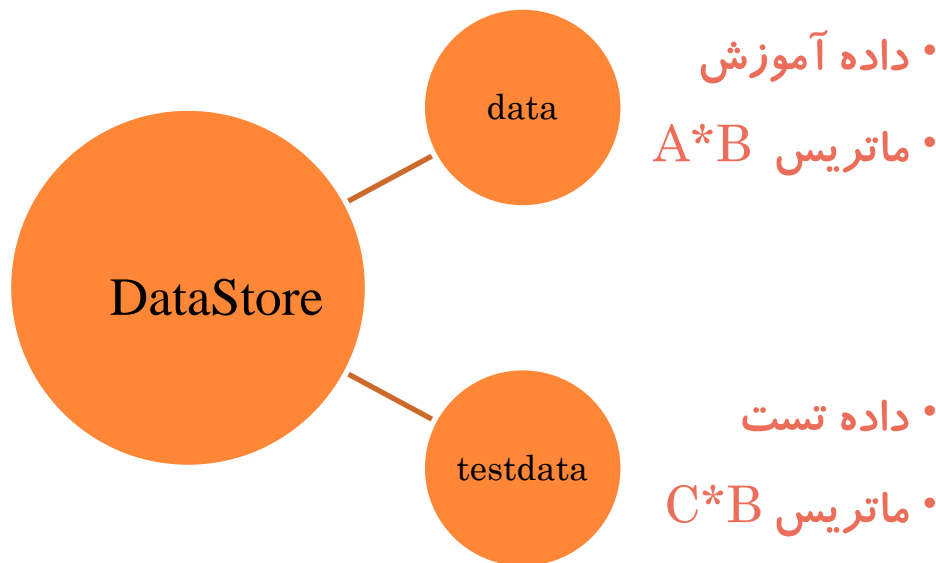
## معرفی برخی کلاس‌های مهم DeeBNet Toolbox

### ○ نرمال سازی داده های ورودی

- Meanvar: نرمال سازی داده ها با میانگین صفر و واریانس ۱
- valueType از نوع gaussian می شود

### ○ کلاس DataStore

- برای راحتی کار با این جعبه ابزار داده های خود را به صورت نمودار مقابل مرتب کنید و به صورت یک فایل .mat ذخیره کنید

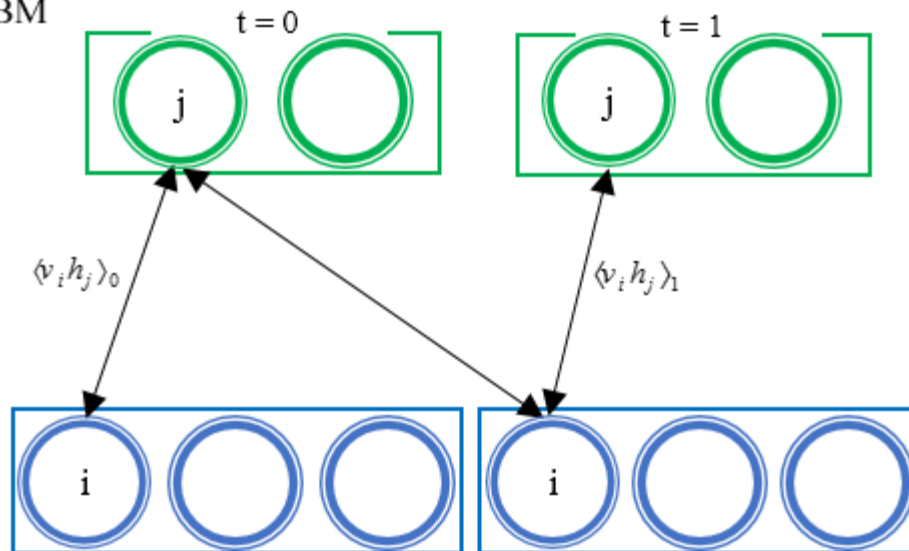


# معرفی برخی کلاس‌های مهم DeeBNet Toolbox

## ○ کلاس Sampling

- یک کلاس واسطه برای استفاده از کلاس‌های نمونه برداری است

RBM



Sampling	
+posHid	$h_{j0}$
+negVis	$v_{i1}$
+negHid	$h_{j1}$
-methodObj	
+Sampling()	
+run()	
+up()	مقدار نمونه‌های مخفی و احتمال فعال شدن آنها
+down()	مقدار نمونه‌های مشاهده پذیر و احتمال فعال شدن آنها



## معرفی برخی کلاس‌های مهم DeeBNet Toolbox

### ○ کلاس DBN

- ساخت شبکه DBN با پشته کردن مدل‌های RBM

○ ساخت DBN از نوع Autoencoder

- `dbn.dbnType='autoEncoder';`

○ پشته کردن RBM ها

- `dbn.addRBM(rbmParams);`

○ تبدیل DBN به شبکه عصبی جهت پس انتشار خطا

- `Function net=DBNtoNN(obj,data)`



## معرفی برخی کلاس‌های مهم DeeBN toolbox

- نمونه کد جهت ساخت Autoencoder برای استخراج ویژگی

◦ تابع برای گرفتن خروجی از سیستم:

◦ `dbn.getFeature(.)`

```
data = MNIST.prepareMNIST_Small('+MNIST\');
data.normalize('meanvar');
data.validationData=data.testData;
data.validationLabels=data.testLabels;

dbn=DBN();
dbn.dbnType='autoEncoder';
% RBM1
rbmParams=RbmParameters(1000,ValueType.binary);
rbmParams.maxEpoch=50;
rbmParams.samplingMethodType=SamplingClasses.SamplingMethodType.CD;
dbn.addRBM(rbmParams);

% RBM2
rbmParams=RbmParameters(500,ValueType.binary);
rbmParams.maxEpoch=100;
rbmParams.samplingMethodType=SamplingClasses.SamplingMethodType.CD;
dbn.addRBM(rbmParams);

% RBM3
rbmParams=RbmParameters(39,ValueType.binary);
rbmParams.maxEpoch=50;
rbmParams.samplingMethodType=SamplingClasses.SamplingMethodType.CD;
dbn.addRBM(rbmParams);

dbn.train(data);
save('dbn.mat','dbn');
dbn.backpropagation(data);
save('dbn+BP.mat','dbn');
```