Advanced Institute for Artificial Intelligence

Máquinas de Boltzmann Restritas - RBM

Advanced Institute for Artificial Intelligence – AI2

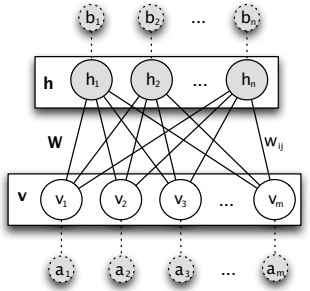
https://advancedinstitute.ai

Introdução

As Máquinas de Boltzmann Restritas, do inglês *Restricted Boltzmann Machines* (RBMs) são técnicas baseadas no aprendizado de modelos de **energia**. Elas podem atuar nas etapas de:

- Redução de dimensionalidade.
- Aprendizado de características.
- Classificação.
- Reconstrução.
- Remoção de ruídos (*denoising*).
- Pré-treinamento de redes neurais MLP.
- Pré-treinamento de redes neurais convolucionais.

RBMs, de maneira geral, são modelos probabilísticos compostos por duas camadas: visível $\boldsymbol{v} \in \{0,1\}^m$ (entrada) e escondida $\boldsymbol{h} \in \{0,1\}^n$. Enquanto que a camada visível é responsável pela leitura dos dados, a camada escondida mapeia sua distribuição por meio de unidades escondidas (binárias, inicialmente) h_j e uma matriz de peso $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ que conecta essas duas camadas. Adicionalmente, temos as camadas de bias (viés) $\boldsymbol{a} \in \mathbb{R}^m$ e $\boldsymbol{b} \in \mathbb{R}^n$ conectadas às camadas visível e escondida, respectivamente.



A **energia** de uma RBM é dada por:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_j w_{ij} - \sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j, \tag{1}$$

em que a probabilidade de uma dada configuração $(\mathbf{v}, \mathbf{h})$ é calculada como segue:

$$P(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{Z}, \tag{2}$$

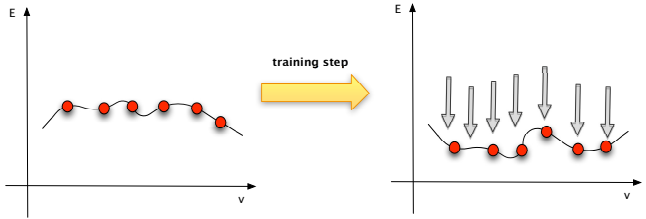
onde Z denota a chamada **constante de normalização/função de partição**, a qual pode ser calculada da seguinte forma:

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \tag{3}$$

A probabilidade de uma amostra do conjunto de dados $\boldsymbol{v}$ (unidade visível) é definida como segue:

$$P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{Z}. \tag{4}$$

Seja $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_z\}$ um conjunto de treinamento. A etapa de aprendizado de uma RBM, de maneira geral, objetiva diminuir a energia de cada amostra de treinamento $\boldsymbol{x}_i$ visando aumentar a sua probabilidade no conjunto.



A verossimilhança do conjunto de dados considerando apenas uma amostra de treinamento $\boldsymbol{v} \leftarrow \boldsymbol{x} \in \mathcal{X}$ é calculada como segue:

$$\phi = \log P(\mathbf{v}) = \phi^{+} - \phi^{-}, \tag{5}$$

em que

$$\phi^{+} = \log \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{6}$$

e

$$\phi^{-} = \log Z = \log \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \tag{7}$$

Agora, a questão é: Como podemos treinar uma RBM?

Basicamente, a etapa de treinamento consiste em atualizar o conjunto de pesos $\boldsymbol{W}$ no sentido de maximizar o logaritmo da verossimilhança do conjunto de dados de treinamento até algum critério de convergência ter sido atingido. Usualmente, utilizamos o gradiente descendente estocástico para tal propósito:

$$\mathbf{W}^{t+1} \leftarrow \mathbf{W}^t + \eta \left(\frac{\partial \phi^+}{\partial \mathbf{W}} - \frac{\partial \phi^-}{\partial \mathbf{W}} \right), \tag{8}$$

em que o **gradiente positivo** é dado por (fácil de ser calculado):

$$\frac{\partial \phi^+}{\partial \mathbf{W}} = \mathbf{v}^T P(\mathbf{h}|\mathbf{v}). \tag{9}$$

O lado direito da Equação 9 pode ser calculado como segue:

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{n} P(h_j = 1|\mathbf{v}), \tag{10}$$

e

$$P(h_j = 1|\mathbf{v}) = \sigma\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right). \tag{11}$$

Neste caso, $\sigma(c) = 1/(1 + \exp(-c))$.

Entretanto, o grande problema consiste no **gradiente negativo**, o qual é dado por:

$$\frac{\partial \phi^-}{\partial \mathbf{W}} = \mathbf{v}^T P(\mathbf{v}|\mathbf{h}), \tag{12}$$

em que $\mathbf{v}$ denota a estimativa (model) dos dados de entrada $\mathbf{v}$, e $P(\mathbf{v}|\mathbf{h})$ é dado por:

$$P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{m} P(v_i = 1|\mathbf{h}), \tag{13}$$

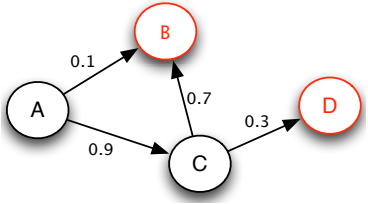
e

$$P(v_i = 1|\mathbf{h}) = \sigma\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right). \tag{14}$$

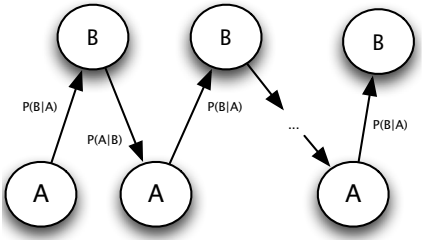
O problema é obter uma boa aproximação do modelos, ou seja, $\frac{\partial \phi^-}{\partial \mathbf{W}}$, o qual requer um grande número de iterações.

Usualmente, podemos modelar a tarefa de estimar a probabilidade condicional por meio de uma Cadeia de Markov, do inglês *Markov Chain Monte Carlo* (MCMC), a qual modela cada passo na direção da aproximação do dados reais como sendo uma cadeia de Markov.

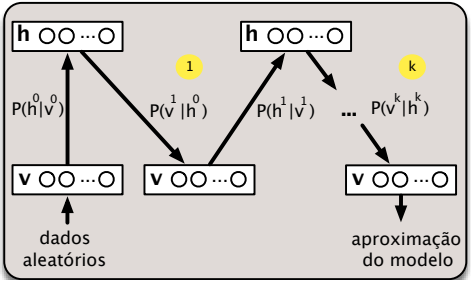
Uma cadeia de Markov é, basicamente, um grafo direcionado e ponderado que obedece algumas propriedades (Teorema Ergódico):



Uma das abordagens mais famosas para amostragem em cadeias de Markov é a conhecida por **amostragem de Gibbs**, a qual aproxima a solução da verossimilhança quando $k \to \infty$, em que k corresponde ao número de iterações.

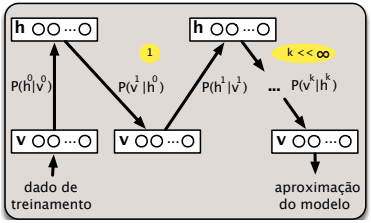


Como podemos utilizar a amostragem de Gibbs para treinar RBMs? Suponha que tenhamos uma cadeia de Markov $\mathcal{C} = \{\mathbf{v}, \tilde{\mathbf{v}}^1, \tilde{\mathbf{v}}^2, \ldots, \tilde{\mathbf{v}}^k\}$ composta pelo dado de entrada (estado inicial) $\mathbf{v}$ e sua reconstrução no tempo t dada por $\tilde{\mathbf{v}}^t$.



Problema? Alto custo computacional dado que precisamos de $k \to \infty$ para uma boa aproximação do modelo.

Hinton (2002)[1] propuseram a Divergência Contrastiva, do inglês *Contrasttive Divergence* (CD), a qual é uma alternativa mais eficiente à amostragem de Gibbs.



Usualmente, $k = 1$. Problema? Modelos estimados tendem a se tornar muito parecidos com as amostras de treinamento.

[1] Hinton, G. E. "Training products of experts by minimizing contrastive divergence", *Neural Computation*, 14(8), 1771-1800, 2002.

Posteriormente, novos parâmetros foram introduzidos para melhorar o processo de treinamento: decaimento de peso λ e momento α. Desta forma, temos novas equações de atualização:

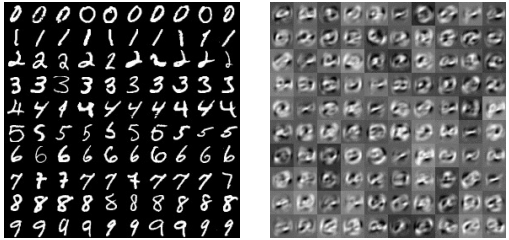
$$\mathbf{W}^{t+1} = \mathbf{W}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{v})\mathbf{v}^T - P(\tilde{\mathbf{v}}|\tilde{\mathbf{h}})\tilde{\mathbf{v}}^T) - \lambda\mathbf{W}^T + \alpha\Delta\mathbf{W}^{t-1}}_{=\Delta\mathbf{W}^t}, \tag{15}$$

$$\mathbf{a}^{t+1} = \mathbf{a}^t + \underbrace{\eta(\mathbf{v} - \tilde{\mathbf{v}}) + \alpha\Delta\mathbf{a}^{t-1}}_{=\Delta\mathbf{a}^t} \tag{16}$$

e

$$\mathbf{b}^{t+1} = \mathbf{b}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{v}) - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})) + \alpha\Delta\mathbf{b}^{t-1}}_{=\Delta\mathbf{b}^t}. \tag{17}$$

A seguir, mostraremos um exemplo do uso de uma RBM para reconstrução de imagens binárias. O critério de minimização foi o do erro médio quadrático, do inglês *Mean Squared Error* (MSE), entre os pixels da imagem de entrada e da imagem reconstruída. A base de dados a ser adotada é a MNIST, que consiste em imagens de dígitos manuscritos.



(a) (b)

Figura: Exemplos de imagens base MNIST: (a) imagens originais e (b) imagens dos pesos da rede.

Exemplo de uma RBM treinada apenas com dígitos ‘0’.

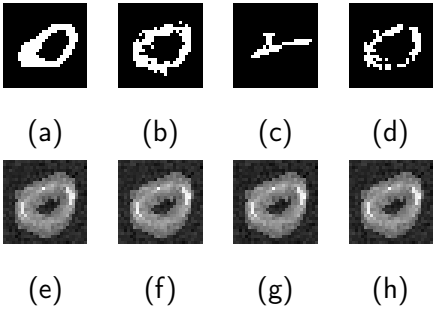


Figura: (a) Dígito original ‘0’ e (b) sua reconstrução, (c) dígito ‘1’ e (d) sua reconstrução. Alguns pesos da RBM estão mostrados em (e)-(h).

Como podemos, então, utilizar RBMs para redução de dimensionalidade, aprendizado não supervisionado de características e ainda como pré-treinamento de redes neurais MLP?

- Note que os valores presentes nas unidades escondidas $h_i \in \{0, 1\}/$ ou ainda $P(h_i = 1|\boldsymbol{v})$ podem ser utilizados como características e, quando $n < m$, temos uma redução do espaço original.
- Suponha que você tenha um conjunto grande de dados de treinamento, mas que somente uma pequena parte dele é rotulada. Você pode usar todo o conjunto de treinamento para o aprendizado de uma RBM (**não supervisionado**) e, em seguida, adicionar uma camada de saída à RBM e utilizar as imagens rotuladas para treinar uma MLP. Note que os pesos já estarão inicialmente aprendidos por meio do algoritmo de treinamento da RBM.

Máquinas de Bolztmann Discriminativas

Como funcionam as RBMs no contexto de classificação? Para este tipo de fim, uma alternativa proposta corresponde às Máquinas de Boltzmann Restritas Discriminativas, do inglês *Discriminative Restricted Boltzmann Machines* (DRBMs).

De maneira geral, a ideia consiste em fazer uso do rótulo de cada amostra como entrada para a DRBM. A classificação é então realizada tentando "reconstruir" o rótulo.

Temos, agora, uma unidade visível denominada $\boldsymbol{y} \in \{0,1\}^k$ e uma matriz de pesos $\boldsymbol{U} \in \mathbb{R}^{k \times n}$ que conecta $\boldsymbol{y}$ à camada escondida $\boldsymbol{h}$.

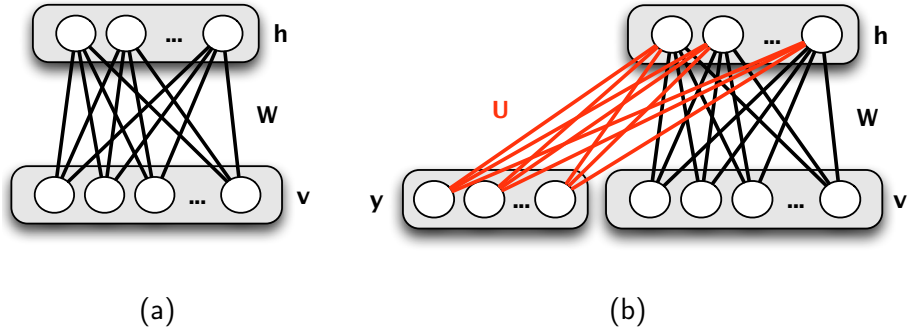


Figura: Arquitetura: (a) RBM e (b) DRBM.

Em uma DRBM, a camada $\boldsymbol{y}$ também assume valores binários para codificar a classe (*one-hot encoding*). A probabilidade de **configuração conjunta** $P(\mathbf{y}, \mathbf{v}, \mathbf{h})$ pode ser expressa da seguinte forma:

$$P(\mathbf{y}, \mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{y}, \mathbf{v}, \mathbf{h})}}{\sum_{\mathbf{y}, \mathbf{v}, \mathbf{h}} e^{-E(\mathbf{y}, \mathbf{v}, \mathbf{h})}}, \tag{18}$$

em que a função de energia $E(\mathbf{y}, \mathbf{v}, \mathbf{h})$ é dada por:

$$E(\mathbf{y}, \mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} v_i h_j - \sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{z=1}^{k} c_z y_z - \sum_{z=1}^{k}\sum_{j=1}^{n} u_{zj} h_j y_z, \tag{19}$$

em que $\boldsymbol{c} \in \mathbb{R}^z$ corresponde à camada de bias da unidade de rótulos $\boldsymbol{y}$.

Assim sendo, a etapa de treinamento de uma RDBM objetiva encontrar o conjunto de parâmetros $\theta = \{\mathbf{W}, \mathbf{U}, \mathbf{a}, \mathbf{b}, \mathbf{c}\}$ que minimiza algum critério de erro como, por exemplo, o MSE da camada de rótulo reconstruída.

O procedimento de aprendizado por CD para encontrar θ consiste em calcular $P(\mathbf{h}|\mathbf{y}, \mathbf{v})$, o qual pode ser realizado para cada unidade h_j, $j = 1, 2, \ldots, n$ como segue:

$$P(h_j = 1|\mathbf{y}, \mathbf{v}) = \phi\left(b_j + \sum_{z=1}^{k} u_{zj} y_z + \sum_{i=1}^{m} w_{ij} v_i\right), \tag{20}$$

em que $\phi(\cdot)$ corresponde à função logística.

Em seguida, podemos calcular a probabilidade condicional de obter **v** dada cada camada escondida **h** para cada unidade visível:

$$P(v_i = 1|\mathbf{h}) = \phi\left(a_i + \sum_{j=1}^{n} w_{ij} h_j\right), \tag{21}$$

bem como a probabilidade de obter **y** dada a unidade escondida **h** para cada unidade da camada de rótulo, como segue:

$$P(y_z = 1|\mathbf{h}) = \frac{\exp\left(c_z + \sum_{j=1}^{n} u_{zj} h_j\right)}{\sum_{l=1}^{k} \exp\left(c_l + \sum_{j=1}^{n} u_{lj} h_j\right)}. \tag{22}$$

O passo final da técnica CD consiste em calcular a Equação 20 mais uma vez para obtermos uma versão atualizada da camada escondida **h**:

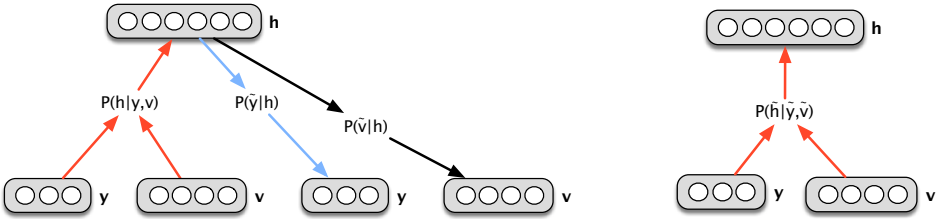


Figura: Aprendizado por CD: (a) calcular as probabilidades condicionais para as unidades escondida (“vermelho"), rótulo (“azul") e visível (“preto"), e (b) e calculando a probabilidade condicional da camada escondida mais uma vez.

De maneira geral, o algoritmo de treinamento da DRBM consiste em estimar cada parâmetro em θ até algum critério de convergência ter sido atingido. Assim sendo, as seguintes equações de atualização podem ser aplicada no instante de tempo $t+1$:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{y},\mathbf{v})(\mathbf{v})^T - P(\tilde{\mathbf{h}}|\tilde{\mathbf{y}},\tilde{\mathbf{v}})(\tilde{\mathbf{v}})^T) - \lambda\mathbf{W}^t + \alpha\Delta\mathbf{W}^{t-1}}_{=\Delta\mathbf{W}^t}, \tag{23}$$

$$\mathbf{U}^{t+1} = \mathbf{U}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{y},\mathbf{v})(\mathbf{y})^T - P(\tilde{\mathbf{h}}|\tilde{\mathbf{y}},\tilde{\mathbf{v}})(\tilde{\mathbf{y}})^T) - \lambda\mathbf{U}^t + \alpha\Delta\mathbf{U}^{t-1}}_{=\Delta\mathbf{U}^t}, \tag{24}$$

$$\mathbf{a}^{t+1} = \mathbf{a}^t + \underbrace{\eta(\mathbf{v}-\tilde{\mathbf{v}}) + \alpha\Delta\mathbf{a}^{t-1}}_{=\Delta\mathbf{a}^t}, \tag{25}$$

$$\mathbf{b}^{t+1} = \mathbf{b}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{yv}) - P(\tilde{\mathbf{h}}|\tilde{\mathbf{y}},\tilde{\mathbf{v}})) + \alpha\Delta\mathbf{b}^{t-1}}_{=\Delta\mathbf{b}^t}, \tag{26}$$

e

$$\mathbf{c}^{t+1} = \mathbf{c}^t + \underbrace{\eta(\mathbf{y} - \tilde{\mathbf{y}}) + \alpha \Delta \mathbf{c}^{t-1}}_{=\Delta \mathbf{c}^t}. \tag{27}$$

A etapa de classificação de uma DRBM pode ser realizada associando à uma dada amostra de teste $\mathbf{x}$ a classe e, $e = 1, 2, \ldots, k$, que maximiza a equação abaixo:

$$p(y_e|\mathbf{x}) = \frac{\exp\{-F(\mathbf{x}, y_e)\}}{\sum_{y^* \in \{1,2,\ldots,k\}} \exp\{-F(\mathbf{x}, \mathbf{y}^*)\}}, \tag{28}$$

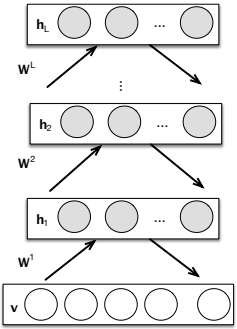
em que $F(\mathbf{x}, y_e)$ corresponde à conhecida "energia livre", que pode ser calculada como segue:

$$F(\mathbf{x}, y_e) = c_e + \sum_{j=1}^{n} \varphi \left(\sum_{i=1}^{m} w_{ij} + u_{yj} + b_j) \right), \tag{29}$$

em que $\varphi(z)$ corresponde à função *softplus*, isto é, $\varphi(z) = \log(1 + \exp(z))$. É importante lembrar que a unidade de rótulo **y** é um vetor binário em que somente a posição e está ligada quando **y** representa a classe e.

Redes de Crença em Profundidade

As Redes de Crença em Profundidade, do inglês *Deep Belief Networks* (DBNs), podem ser entendidas, de maneira geral, como sendo diversas RBMs empilhadas.



O seu treinamento é realizado de maneira **gulosa**, ou seja, cada RBM é treinada de maneira independente, em que a saída de um modelo alimenta a entrada do outro. Podem ser utilizadas para inicializar redes MLP com diversas camadas escondidas. Já nas Máquinas de Boltzmann em Profundidade, do inglês *Deep Boltzmann Machines* (DBMs), as inferências de cada camada escondida levam em consideração as suas camadas acima e abaixo.