Restricted Boltzmann Machines

Sara Cocomello, Paolo Da Rold, Elena Rivaroli

Probabilistic Machine Learning

A.Y. 2022-2023

Model Structure

Model structure

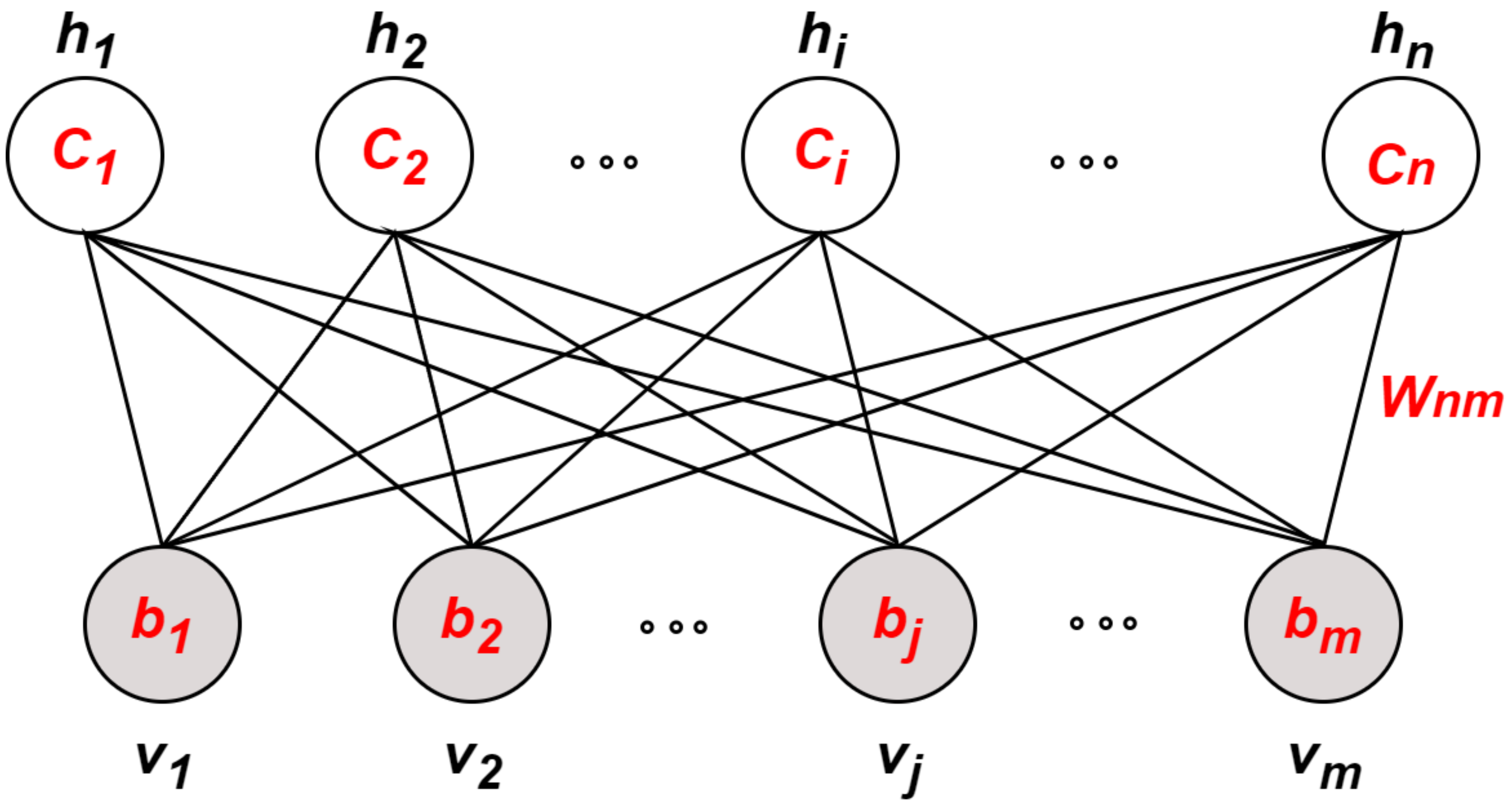


Fig. 1: The undirected graph of an RBM with *n* hidden and *m* visible variables

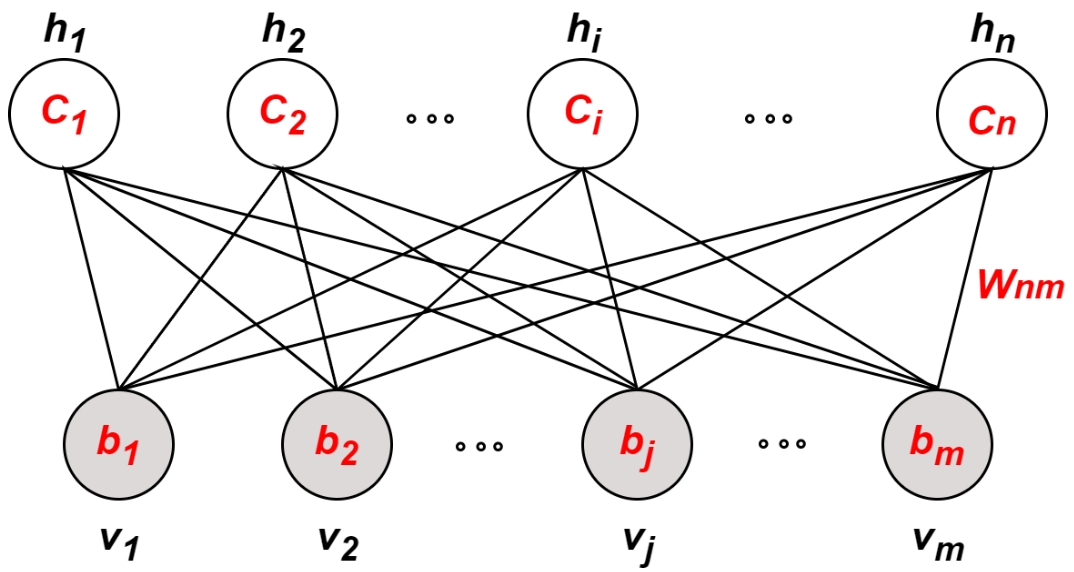
$$(\boldsymbol{v}, \boldsymbol{h}) \in \{0,1\}^{m+n}$$

$$h_i \perp h_j | \mathbf{v}$$

$$v_i \perp v_j | \mathbf{h}$$

Gibbs distribution

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



Energy function

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

Factorization

$$p(\boldsymbol{h} \mid \boldsymbol{v}) = \prod_{i=1}^{n} p(h_i \mid \boldsymbol{v}) \qquad p(H_i = 1 \mid \boldsymbol{v}) = \sigma\left(\sum_{j=1}^{m} w_{ij} v_j + c_i\right)$$

$$p(\boldsymbol{v} \mid \boldsymbol{h}) = \prod_{i=1}^{m} p(v_i \mid \boldsymbol{h}) \qquad p(V_j = 1 \mid \boldsymbol{h}) = \sigma\left(\sum_{i=1}^{n} w_{ij} h_i + b_j\right)$$

Gibbs sampling

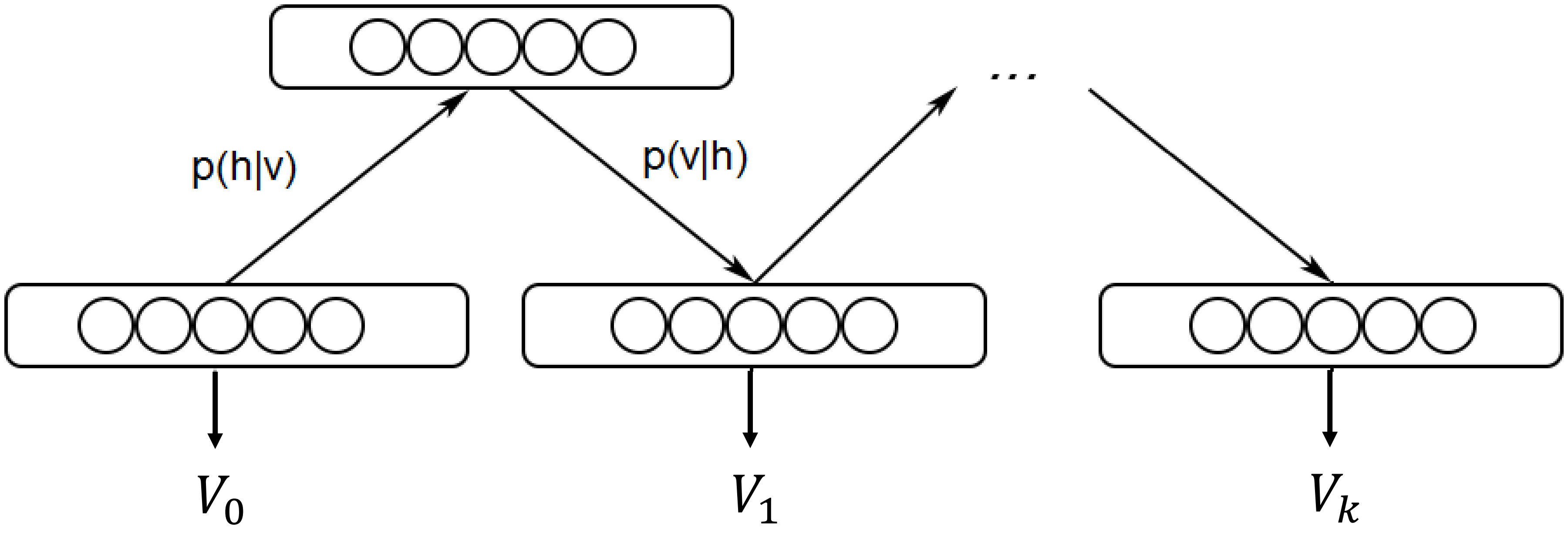


Fig. 2: Block Gibbs Sampling

RBM Training

Maximum Likelihood

The Log-Likelihood of a general MRF with latent variables is given by:

$$\ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v}) = \ln p(\boldsymbol{v} \mid \boldsymbol{\theta}) = \ln \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} - \ln \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

The gradient w.r.t. the parameters is:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v})}{\partial \boldsymbol{\theta}} = -\sum_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{v}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

Maximum Likelihood for RBM

The explicit derivative w.r.t the weights w_{ij} is:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial w_{ij}} = -\sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}} + \sum_{\boldsymbol{v}, \boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}}$$

Computing the derivatives and using a factorization trick we obtain:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial w_{ij}} = \underbrace{\boxed{p(H_i = 1 |\, \boldsymbol{v}) v_j}}_{\text{OK}} - \underbrace{\boxed{\sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 |\, \boldsymbol{v}) v_j}}_{\text{NOT OK}}$$

Contrastive Divergence

To solve the problem we use the contrastive divergence technique:

$$\mathrm{CD}_k(\boldsymbol{\theta}, \boldsymbol{v}^{(0)}) = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(0)}) \frac{\partial E(\boldsymbol{v}^{(0)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(k)}) \frac{\partial E(\boldsymbol{v}^{(k)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

Usually
$k = 1$

Maximum Likelihood: $\longrightarrow$ Minimize $\left[\mathrm{KL}(q|p)\right]$

Contrastive divergence: $\longrightarrow$ Minimize $\left[\mathrm{KL}(q|p) - \mathrm{KL}(p_k|p)\right]$

Pseudocode of Contrastive Divergence

Input: RBM, training batch S
Output: gradient approximations

Init $\Delta \boldsymbol{w}$ = $\Delta \boldsymbol{b} = \Delta \boldsymbol{c} = \mathbf{0}$
forall v in a batch
    $v^{(0)} \leftarrow v$
    for t=0 step (k-1)
        $v^{(k)} \leftarrow$ sample $h^{(t)}$ from p(h|$v^{(t)}$) and subsequently $v^{(t+1)}$ from p(h|$v^{(t)}$)
        update $\Delta \boldsymbol{w}, \Delta \boldsymbol{b}, \Delta \boldsymbol{c}$ using CD rules

Code 1: k steps contrastive divergence

Application

Classification on MNIST dataset

Fig. 3: Sample from MNIST dataset 28x28 images

- **PREPROCESSING** : scaling the pixel values in [0,1] with grayscale
- **TRAINING** : defining the RMB architecture by setting the number of hidden units and train on MNIST
- **FEATURE EXTRACTION**: transform the data with the features extracted by RBM
- **CLASSIFICATION**: training general classifiers with extracted features

Features extraction and classification

MODEL	ACCURACY RBM	ACCURACY RAW
Logistic	0.94	0.92
KNN	0.97	0.98
SVM	0.97	0.98

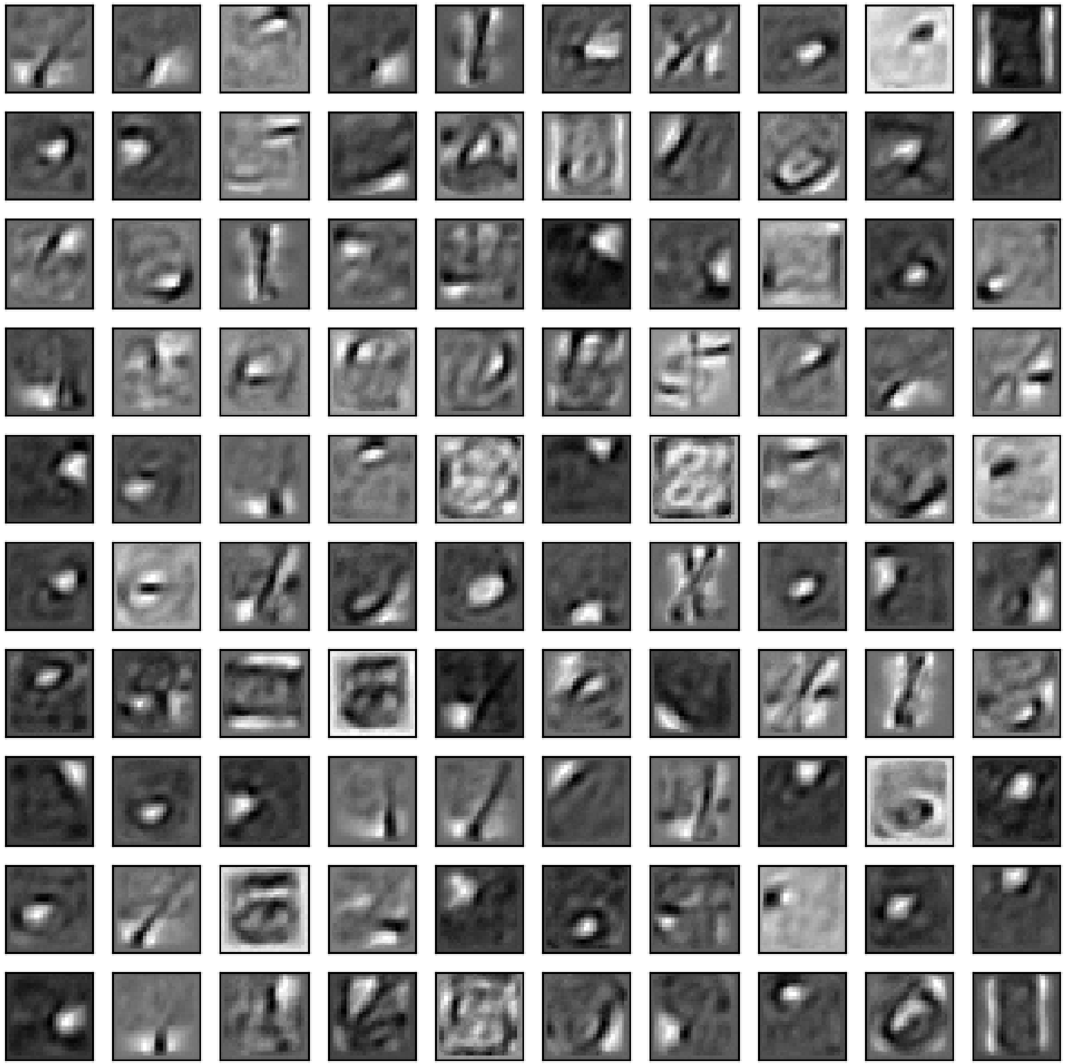
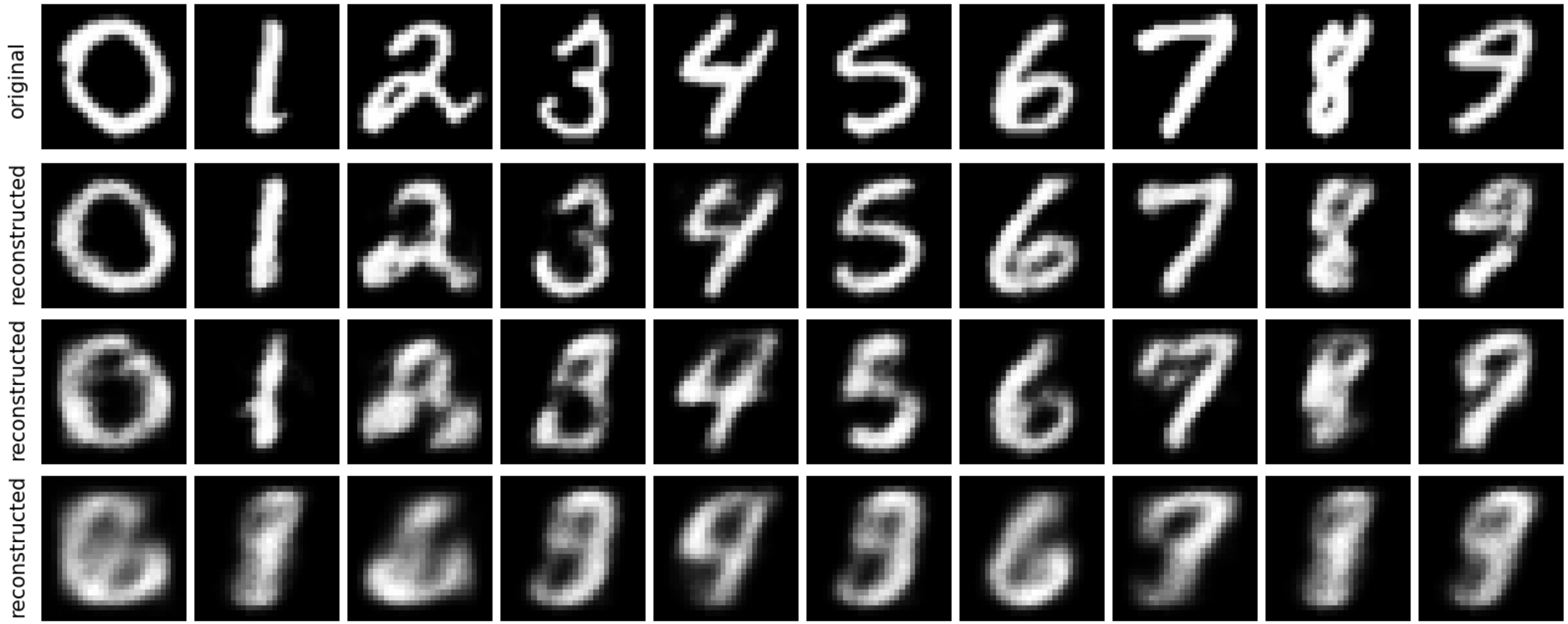


Fig. 4: 100 components

Evaluation



RECONSTRUCTION ERROR

`0.024 (100 hidden)`

`0.038 (25 hidden)`

`0.063 (5 hidden)`

Fig. 5: Comparison of reconstruction with different numbers of hidden variables

Conclusions

RBM vs raw pixels for classification

- Feature extraction
- Dimensionality reduction
- Unupervised learning

Current application

- Using RBM as building blocks of Deep neural networks

References

[1] A. Fischer and C. Igel, “An introduction to restricted Boltzmann machines”. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer, Berlin Heidelberg, 2012.

[2] Carreira-Perpiñán, M.A., Hinton, G.E.: On contrastive divergence learning. In: 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005), pp. 59–66 (2005)

[3] Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14, 1771–1800 (2002)

Thank you for the attention!

Appendix

Factorization trick

$$\sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}} = \sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) h_i v_j$$

$$= \sum_{\boldsymbol{h}} \prod_{k=1}^{n} p(h_k \,|\, \boldsymbol{v}) h_i v_j = \sum_{h_i} \sum_{\boldsymbol{h}_{-i}} p(h_i \,|\, \boldsymbol{v}) p(\boldsymbol{h}_{-i} \,|\, \boldsymbol{v}) h_i v_j$$

$$= \sum_{h_i} p(h_i \,|\, \boldsymbol{v}) h_i v_j \underbrace{\sum_{\boldsymbol{h}_{-i}} p(\boldsymbol{h}_{-i} | \boldsymbol{v})}_{=1} = p(H_i = 1 | \boldsymbol{v}) v_j = \sigma\left(\sum_{j=1}^{m} w_{ij} v_j + c_i \right) v_j$$