Restricted Boltzmann Machine(RBM)

Relevant Concepts and Basic Properties

Learning Algorithms

Different types of unit

Training RBM

Yunfei Wang

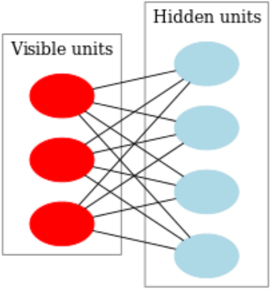
Department of Computer Science & Technology
Huazhong University of Science & Technology

March 20, 2013

Table of contents

1. Restricted Boltzmann Machine(RBM)
    - Relevant Concepts and Basic Properties
    - Learning Algorithms
    - Different types of unit

RBM I



Undirected bipartite graph

Visible units $v \in \{0,1\}^D$
Feature of inputs

Hidden units $h \in \{0,1\}^F$
Feature detectors

The energy of joint distribution:

$$E(v,h;\theta) = -v^T W h - a^T v - b^T h \tag{1}$$

$\theta = W, a, b$ are the model parameters.

RBM II

The probability of each pair of a visible and a hidden vector:

$$P(v,h;\theta) = \frac{1}{Z(\theta)} \exp(-E(v,h;\theta)) \tag{2}$$

Partition function(Normalizing Constant):

$$Z(\theta) = \sum_{v,h} \exp(-E(v,h;\theta)) \tag{3}$$

RBM III

Marginal distribution:

$$\begin{aligned}
P(v;\theta) &= \sum_h P(v,h;\theta) \\
&= \frac{1}{Z(\theta)} \sum_h \exp(v^T W h + a^T v + b^T h) \\
&= \frac{1}{Z(\theta)} \exp(a^T v) \sum_h \exp(v^T W h + b^T h) \\
&= \frac{1}{Z(\theta)} \exp(a^T v) \prod_{j=1}^{F} \sum_{h_j \in \{0,1\}} \exp(\sum_{i=1}^{D} v_i W_{ij} h_j + b_j h_j) \\
&= \frac{1}{Z(\theta)} \exp(a^T v) \prod_{j=1}^{F} (1 + \exp(\sum_{i=1}^{D} v_i W_{ij} + b_j))
\end{aligned} \tag{4}$$

RBM IV

Conditional distribution over visible units can be figured out using Bayesian formula:

$$\begin{aligned} P(h|v;\theta) &= \frac{P(v,h;\theta)}{P(v;\theta)} \\ &= \frac{\exp(v^T W h + b^T h)}{\prod_{j=1}^{F}(1+\exp(\sum_{i=1}^{D} v_i W_{ij} + b_j))} \\ &= \prod_{j=1}^{F} \frac{\exp(\sum_{i=1}^{D} v_i W_{ij} h_j + b_j h_j)}{1+\exp(\sum_{i=1}^{D} v_i W_{ij} + b_j)} \end{aligned} \tag{5}$$

The hidden units are conditionally independent:

$$P(h|v;\theta) = \prod_{j=1}^{F} P(h_j|v;\theta) = \prod_{j=1}^{F} \frac{exp(\sum_{i=1}^{D} v_i W_{ij} h_j + b_j h_j)}{1+exp(\sum_{i=1}^{D} v_i W_{ij} + b_j)} \tag{6}$$

RBM V

It's easy to derive the following conclusions:

$$P(h_j|v) = \frac{\exp(\sum_{i=1}^{D} v_i W_{ij} h_j + b_j h_j)}{1 + \exp(\sum_{i=1}^{D} v_i W_{ij} + b_j)} \tag{7}$$

$$\begin{aligned} P(h_j = 1|v) &= \frac{\exp(\sum_{i=1}^{D} v_i W_{ij} + b_j)}{1 + \exp(\sum_{i=1}^{D} v_i W_{ij} + b_j)} \\ &= \frac{1}{1 + \exp(-\sum_{i=1}^{D} v_i W_{ij} - b_j)} \\ &= g(\sum_{i=1}^{D} v_i W_{ij} + b_j) \end{aligned} \tag{8}$$

where $g(x) = \frac{1}{1+\exp(-x)}$ is the logistic function.

RBM VI

Similarly:

$$P(v|h;\theta) = \prod_{i=1}^{F} P(v_i|h;\theta) \tag{9}$$

$$P(v_i = 1|h) = g(\sum_{j=1}^{F} W_{ij}h_j + a_i) \tag{10}$$

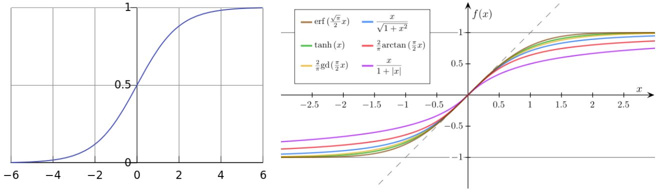


Figure: Sigmoid function:bounded, differentiable, real-valued

RBM VII

Log-likelihood:

$$\log P(v;\theta) = \log \sum_h \exp(-E(v,h;\theta)) - \log Z \tag{11}$$

Derivative of the log-likelihood with respect to parameters θ:

$$\begin{aligned}\frac{\partial P(v;\theta)}{\partial W} &= \frac{\sum_h vh^T \exp(-E(v,h;\theta))}{\sum_h \exp(-E(v,h;\theta))} - \frac{\sum_{v,h} vh^T \exp(-E(v,h;\theta}{Z(\theta)} \\ &= \frac{\sum_h vh^T P(v,h;\theta)}{P(v;\theta)} - \sum_{v,h} vh^T P(v,h;\theta) \\ &= \sum_h vh^T P(h|v;\theta) - \sum_{v,h} vh^T P(v,h;\theta) \\ &= E_{data}(vh^T) - E_{model}(vh^T)\end{aligned} \tag{12}$$

RBM VIII

Similarly:

$$\frac{\partial P(v;\theta)}{\partial a} = E_{data}(v) - E_{model}(v) \tag{13}$$

$$\frac{\partial P(v;\theta)}{\partial b} = E_{data}(h) - E_{model}(h) \tag{14}$$

where $E_{data}(\bullet)$ and $E_{model}(\bullet)$ denote expectations under the distributions specified by data and model respectively. According to *Gradient Descent*,this leads to a learning rule:

$$\triangle W = \alpha(\underbrace{E_{data}(vh^T)}_{tractable} - \underbrace{E_{model}(vh^T)}_{intractable}) \tag{15}$$

where α is the learning rate.

Learning Algorithms I

Approximating the gradient of a different objective function,called "Contractive Divergence":

$$\triangle W = \alpha(E_{data}(vh^T) - E_T(vh^T)) \tag{16}$$

where $E_T(\bullet)$ denote expectation under the distributions defined by running Gibbs Sampling for T full steps,starting from a training example:

What's Gibbs Sampling?

Gibbs sampling is applicable when the joint distribution is not known explicitly, but the conditional distribution of each variable is known and is easy to sample from.
Gibbs sampling algorithm generates an instance from the distribution of each variable in turn, conditional on the current values of the other variables.

Learning Algorithms II

Implementation of Gibbs Sampling

Obtain samples of $X = \{x_1, x_2, \cdots, x_n\}$ from a joint distribution $P(x_1, x_2, \cdots, x_n)$.Denote the i-th sample by $X^i = \{x_1^i, x_2^i, \cdots, x_n^i\}$.

$$\begin{aligned} x_1^{i+1} &\sim P(x_1|x_2^i, x_3^i, \cdots, x_n^i) \\ x_2^{i+1} &\sim P(x_2|x_1^{i+1}, x_3^i, \cdots, x_n^i) \\ x_3^{i+1} &\sim P(x_3|x_1^{i+1}, x_2^{i+1}, \cdots, x_n^i) \\ . \\ . \\ x_n^{i+1} &\sim P(x_n|x_1^{i+1}, x_2^{i+1}, \cdots, x_{n-1}^{i+1}) \end{aligned} \tag{17}$$

A new sample $X^{i+1} = \{x_1^{i+1}, x_2^{i+1}, \cdots, x_n^{i+1}\}$ is produced here.

Learning Algorithms III

Gibbs Sampling in RBMs

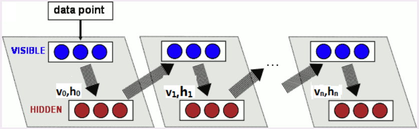


Figure: Gibbs Sampling

Start with a training vector on visible units.
Update all hidden units in parallel.
Update all visible units in parallel to get a "reconstruction".
Update the hidden units again.

The change in weight is then given by:

$$\triangle W = \alpha(E_{data}(vh^T) - E_{recon}(vh^T)) \tag{18}$$

Different types of unit I

Softmax units

For a binary unit,the probability of turning on:

$$P(x) = \frac{1}{1+\exp(-x)} = \frac{\exp(x)}{\exp(x)+\exp(0)} \propto \exp(x) \tag{19}$$

Generalize the binary states to K alternative states:

$$P(x_i) = \frac{\exp(x_i)}{\sum_{k=1}^{K} \exp(x_k)} \tag{20}$$

Softmax unit:States are mutually constrained;Only one unit has value 1,the rest have value 0;View it as a set of K binary units.

Different types of unit II

Multinomial units

Useful for modelling sparse count data,such as word count vectors in a document.Visible units $v \in \mathbb{N}^D$,hidden units $h \in \{0,1\}^F$.The energy function is defined as follows:

$$E(v,h;\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F} v_i W_{ij} h_j - \sum_{i=1}^{D} a_i v_i - M\sum_{j=1}^{F} b_j h_j \quad (21)$$

where v_i is the frequency of word i in a document,D is vocabulary size,$M = \sum_{i=1}^{D} v_i$ is total number of words in the document. This leads to the following conditional distribution:

$$P(v_i = 1|h;\theta) = \frac{\exp(-a_i + \sum_{j=1}^{F} W_{ij} h_j)}{\sum_{i=1}^{D} \exp(-a_i + \sum_{j=1}^{F} W_{ij} h_j)} \quad (22)$$

Different types of unit III

Gaussian visible units

Solution to a representation for images or speech using logistic units: replace binary visible units by linear units with independent Gaussian noise.Visible units $v \in \mathbb{R}^D$,hidden units $h \in \{0,1\}^F$.The energy function is defined as follows:

$$E(v,h;\theta) = \sum_{i=1}^{D} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{F} b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j W_{ij} \qquad (23)$$

where $\theta = \{a, b, W, \sigma\}$ are the model parameters. This leads to the following conditional distribution:

$$P(v_i, h; \theta) = \mathcal{N}(a_i + \sigma_i \sum_{j=1}^{F} W_{ij} h_j, \sigma_i^2) \qquad (24)$$

Different types of unit IV

Gaussian visible and hidden units

Both the visible and hidden units are Gaussian,then the energy function becomes:

$$E(v,h) = \sum_{i \in vis} \frac{(v_i - a_i)^2}{2\sigma_i^2} + \sum_{j \in hid} \frac{(h_j - b_j)^2}{2\sigma_j^2} - \sum_{i,j} \frac{v_i}{\sigma_i} \frac{h_j}{\sigma_j} W_{ij} \tag{25}$$

With a sufficiently small learning rate, CD_1 can learn an undirected version of a factor analysis model using Gaussian units,which is harder than using EM to learn a directed model.