Boltzman Machine

Review and Prospects

Hooman Zolfaghari
Ramtin Moslem
Borna (lastname)
Houman (lastname)

Energy Based Models

Botzman Machine

Introduction

- Idea was taken from statistical physics and formulated by cognitive scientists.
- Energy function:

$$E(x;\theta) = -(\mathbf{x^T W x} + \mathbf{b^T x})$$

- Parameters same as in *Hopfield* networks and *Ising* models

Problem

- Hard to train (due to the partition function)
- Solution:
  - Introducing latent variables
  - Restricting connections among observables.

Restricting BMs

- Consider: binary observable variables $\mathbf{x} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{D}}$ and binary latent (hidden) variables $\mathbf{z} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{M}}$
- The relationships among variables specified through this *energy function*:

$$E(\mathbf{x}, \mathbf{z}; \theta) = -\mathbf{x}^{\mathbf{T}}\mathbf{W}\mathbf{z} - \mathbf{b}^{\mathbf{T}}\mathbf{x} - \mathbf{c}^{\mathbf{T}}\mathbf{x} \tag{1}$$

- For this EF, the RBM is defined by the *Gibbs* distribution:

$$p(\mathbf{x}, \mathbf{z}; \theta) = \frac{\mathbf{1}}{\mathbf{Z}_{\theta}} \exp(-\mathbf{E}(\mathbf{x}, \mathbf{z}; \theta)) \tag{2}$$

Restricting BMs

- the *partition function*:

$$Z_\theta = \sum_{\mathbf{x}} \sum_{\mathbf{z}} \exp\big(-E(\mathbf{x},\mathbf{z};\theta)\big) \tag{3}$$

- The marginal probability over observables (the likelihood of observation):

$$p(\mathbf{x}|\theta) = \frac{1}{Z_\theta} \exp\big(-F(\mathbf{x};\theta)\big) \tag{4}$$

- where $F(\cdot)$ is the *free energy*:

$$F(\mathbf{x};\theta) = -\mathbf{b}^\top\mathbf{x} - \sum_j \log\big(1 + \exp(c_j + (\mathbf{W}_{\cdot j})^\top \mathbf{x})\big). \tag{5}$$

RBM

- The presented model is called a restricted Boltzmann machine (RBM).
- Useful property: the conditional distribution over the hidden variables factorizes given the observable variables and vice versa:

$$p(z_m = 1|\mathbf{x}, \theta) = \mathrm{sigm}\big(c_m + (\mathbf{W}_{\cdot m})^\top \mathbf{x}\big), \tag{6}$$

$$p(x_d = 1|\mathbf{z}, \theta) = \mathrm{sigm}\big(b_d + \mathbf{W}_{d\cdot} \mathbf{z}\big). \tag{7}$$

Learning RBMs

- For given data $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}$
- Train an RBM using the maximum likelihood

$$\ell(\theta) = \frac{1}{N} \sum_{\mathbf{x}_n \in \mathcal{X}} \log p(\mathbf{x}_n \mid \theta) \tag{8}$$

- The gradient with respect to θ:

$$\nabla_\theta \ell(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \left(\nabla_\theta F(\mathbf{x}_n; \theta) - \sum_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}}|\theta) \nabla_\theta F(\hat{\mathbf{x}}; \theta) \right) \tag{9}$$

- Cannot be computed analytically because the second term requires summing over all configurations of observables.

Learning RBMs

- One way to sidestep this: Stochastic approximation
- Replacing the expectation under $p(\mathbf{x}|\theta)$ by a sum over S samples $\{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_S\}$ drawn according to $p(\mathbf{x}|\theta)$:

$$\nabla_\theta \ell(\theta) \approx -\frac{1}{N} \sum_{n=1}^{N} \nabla_\theta F(\mathbf{x}_n; \theta) - \frac{1}{S} \sum_{s=1}^{S} \nabla_\theta F(\hat{\mathbf{x}}_s; \theta). \tag{10}$$

Learning RBMs

- A different approach: *contrastive divergence*
- Approximates the expectation under $p(\mathbf{x}|\theta)$ by a sum over samples $\tilde{\mathbf{x}}_n$ drawn from a distribution obtained by applying K steps of the block Gibbs sampling procedure:

$$\nabla_\theta \ell(\theta) \approx -\frac{1}{N} \sum_{n=1}^{N} \big(\nabla_\theta F(\mathbf{x}_n; \theta) - \nabla_\theta F(\tilde{\mathbf{x}}_n; \theta) \big). \tag{11}$$

Learning RBMs

- The original CD used K steps of the Gibbs chain, starting and is restarted after every parameter update.
- An alternative approach, Persistent Contrastive Divergence (PCD) does not restart the chain after each update typically resulting in a slower convergence rate but eventually better performance

Higher-Order Relationships

- The energy function allows the modeling of higher-order dependencies among variables.
- For instance: Third-order multiplicative interactions by introducing two kinds of hidden variables
  - **Subspace Units:** Reflect feature variations, robust to invariances:
  - **Gate Units:** Activate subspace units, pool subspace features.

Higher-Order Relationships

Random Variables for SubspaceRBM

- Observables: $\mathbf{x} \in \{0,1\}^D$.
- Gate Units: $\mathbf{h} \in \{0,1\}^M$.
- Subspace Units: $\mathbf{S} \in \{0,1\}^{M \times K}$.
- **Connections:** x_i, h_j, and s_{jk}.

Higher-Order Relationships

Energy Function for SubspaceRBM

$$E(\mathbf{x}, \mathbf{h}, \mathbf{S}; \theta) = -\sum_{i=1}^{D}\sum_{j=1}^{M}\sum_{k=1}^{K} W_{ijk} x_i h_j s_{jk} - \sum_{i=1}^{D} b_i x_i - \sum_{j=1}^{M} c_j h_j - \sum_{j=1}^{M} h_j \sum_{k=1}^{K} D_{jk} s_{jk}, \tag{12}$$

$$\theta = \{W, \mathbf{b}, \mathbf{c}, \mathbf{D}\}, \quad W \in \mathbb{R}^{D\times M\times K}, \quad \mathbf{b} \in \mathbb{R}^{D}, \quad \mathbf{c} \in \mathbb{R}^{M}, \quad \mathbf{D} \in \mathbb{R}^{M\times K}.$$

Higher-Order Relationships

Conditional Distributions in SubspaceRBM

$$p(x_i = 1|\mathbf{h}, \mathbf{S}) = \mathrm{sigm}\left(\sum_j \sum_k W_{ijk} h_j s_{jk} + b_i\right) \tag{13}$$

$$p(s_{jk} = 1|\mathbf{x}, h_j) = \mathrm{sigm}\left(\sum_i W_{ijk} x_i h_j + h_j D_{jk}\right) \tag{14}$$

$$p(h_j = 1|\mathbf{x}) = \mathrm{sigm}\left(-K \log 2 + c_j + \sum_{k=1}^{K} \mathrm{softplus}\left(\sum_i W_{ijk} x_i + D_{jk}\right)\right) \tag{15}$$

Research Prospect

Template Things

Single Column

For 4-by-3 aspect ratio slides, specify `standard` as an option to the document class. Write your presentation like a normal LaTeX file with a `\maketitle` command and `\chapter` and `\section` headings. The `\maketitle` contents are defined by the following macros:

`\pretitle`	`\author`
`\title`	`\subtitle`

The `\chapter` heading creates a slide with just the chapter name, and the `\section` heading sets the title of a new slide. However, if no text follows the section, no slide will be created. Text which does not fit on one slide will flow onto the next slide automatically.

Double Column

Use the `\twocolumn` and `\onecolumn` commands right after the section heading to control the number of columns. Text will flow from the left column to the right.

- Point one
- Point two
- Point three
- Point four
- Point five
- Point six
- Point seven
- Point eight
- Point nine
- Point ten
- Point eleven
- Point twelve

You can use `\pagebreak` to force text onto the next column.

Table of Stuff

You can create any variety of subdivisions on your slide by using the `tabular` environment.

Primary	Secondary	Tertiary
First	Second	Third
One	Two	Three

Alpha	Beta	Gamma
Green	Blue	Red
Cyan	Yellow	Magenta

The `\cellcolor` command sets the background color of a table cell.

Centering

Use the `Center` environment
to center horizontally *and* vertically.

Explicit Code

Python

Use the `python` environment for Python code.

```python
def write_list(fid, x, level):
    ind = '  '*level
    xs = '0' if abs(x[0]) < 1e-3 else "%.3f"
    txt = '\n%svalues=\"%s' % (ind, xs)
    for n in range(1, len(x)):
        xs = '0' if abs(x[n]) < 1e-3 else "%.3f"
        if len(txt) + 3 + len(xs) >= 80:
            fid.write(txt + ';\n')
            txt = ind + '  ' + xs
        else:
            txt += '; ' + xs
    fid.write(txt + '\"')
```

Python

You can use the `\HL` command to highlight a line of code.

```python
def write_list(fid, x, level):
    ind = '  '*level
    xs = '0' if abs(x[0]) < 1e-3 else "%.3f"
    txt = '\n%svalues=\"%s' % (ind, xs)
    for n in range(1, len(x)):
        xs = '0' if abs(x[n]) < 1e-3 else "%.3f"
        if len(txt) + 3 + len(xs) >= 80:
            fid.write(txt + ';\n')
            txt = ind + '  ' + xs
        else:
            txt += '; ' + xs
    fid.write(txt + '\"')
```

MATLAB

Use the `matlab` environment for MATLAB code.

```matlab
function savepdf(name, width, height)
    % name is the file name including ".pdf".
    % Both width and height are in (cm).
    set(gcf, 'units', 'centimeters', ...
        'position', [0, 0, width, height])
    set(gca, 'FontSize', 9);
    set(gca, 'FontName', 'Times New Roman');
    exportgraphics(gcf, name, ...
        'ContentType', 'vector');
end
```

R Language

Use the `rlang` environment for R code.

```r
factorial <- function(n) {
    if (n == 0 || n == 1) {
        return(1)
    } else {
        return(n * factorial(n - 1))
    }
}
```

Blank pages with the “`\blankpage`” command.

Arbitrary positioning with the “`\pos`” command.