Fuzzy Restricted Boltzmann Machine for the Enhancement of Deep Learning

C. L. Philip Chen, *Fellow, IEEE*, Chun-Yang Zhang, Long Chen, *Member, IEEE*, and Min Gan

***Abstract*—In recent years, deep learning caves out a research wave in machine learning. With outstanding performance, more and more applications of deep learning in pattern recognition, image recognition, speech recognition, and video processing have been developed. Restricted Boltzmann machine (RBM) plays an important role in current deep learning techniques, as most of existing deep networks are based on or related to it. For regular RBM, the relationships between visible units and hidden units are restricted to be constants. This restriction will certainly downgrade the representation capability of the RBM. To avoid this flaw and enhance deep learning capability, the fuzzy restricted Boltzmann machine (FRBM) and its learning algorithm are proposed in this paper, in which the parameters governing the model are replaced by fuzzy numbers. This way, the original RBM becomes a special case in the FRBM, when there is no fuzziness in the FRBM model. In the process of learning FRBM, the fuzzy free energy function is defuzzified before the probability is defined. The experimental results based on bar-and-stripe benchmark inpainting and MNIST handwritten digits classification problems show that the representation capability of FRBM model is significantly better than the traditional RBM. Additionally, the FRBM also reveals better robustness property compared with RBM when the training data are contaminated by noises.**

***Index Terms*—Deep learning, fuzzy deep networks, fuzzy restricted Boltzmann machine, image classification, image inpainting, restricted Boltzmann machine (RBM).**

I. Introduction

RESTRICTED Boltzmann machine (RBM), as illustrated in Fig. 1, is a stochastic graph model that can learn a joint probability distribution over its n visible units $\mathbf{x} = [x_1, \ldots, x_n]$ and m hidden feature units $\mathbf{h} = [h_1, \ldots, h_m]$. The model is governed by parameters $\boldsymbol{\theta}$ denoting connection weights and biases between cross-layer units. The RBM [1] was invented in 1986; however, it received a new birth when G. Hinton and his partners recently proposed several deep networks and corresponding fast learning algorithms, including deep autoencoder [2], deep belief networks [3], and deep Boltzmann machine [4]. The RBM and its deep architectures have a large number of applications [5], such as dimensionality reduction [6], classification [7], collaborative filtering [8], feature learning [9], and topic modeling [10]. A detailed knowledge of the RBM and its deep architectures can be found in [11] and [12].

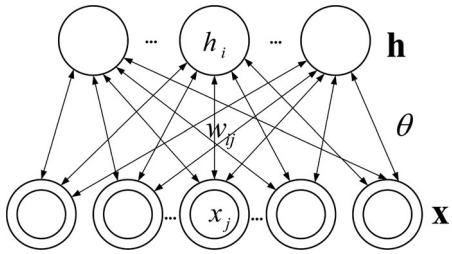


Fig. 1. Restricted Boltzmann Machine (RBM).

Most researchers in deep learning field focus on deep network design and corresponding fast learning algorithms. Some research works try to improve the deep learning technique from the model representation. For example, Gaussian-restricted Boltzmann machines (GRBMs) with Gaussian linear units are proposed to learn representations from real-valued data [13]. It improves the RBM by replacing binary-valued visible units with Gaussian ones. The deep networks based on GRBM are also developed in recent years, such as Gaussian–Bernoulli deep Boltzmann machine [14], [15]. Conditional versions of RBMs have also been developed for collaborative filtering [16], and temporal RBMs and recurrent RBMs are proposed to model high-dimensional sequence data, such as motion capture data [17] and video sequences [18].

For regular RBMs and their existing variants, the parameters that represent the relationships between units in visible and hidden layers are restricted to be constants. This structure design will surely lead to many problems. First, it will constrain the representation capability, since the variables often interact in some uncertain ways. Second, the RBMs are not very robust when the training data samples corrupted by noises. Third, the parameter learning process of the RBMs is confined in a relatively small space. This is contrary with merits of deep learning. All of these constraints will be reflected by the fitness of the target joint probability distribution. To overcome these disadvantages and reduce the inaccuracy and distortion deduced by linearization of the relationship between cross-layer units, this paper proposes the fuzzy restricted Boltzmann machines (FRBM) and corresponding learning algorithm, where the parameters governing the models are all fuzzy numbers. After the vagueness in the relationship between the visible units

Manuscript received August 26, 2014; revised December 14, 2014; accepted January 12, 2015. Date of publication February 24, 2015; date of current version November 25, 2015. This work was supported by the Macau Science and Technology Development Fund under Grant 008/2010/A1, UM Multiyear Research Grants, and the National Natural Science Foundation of China under Grant 61203106. *(Corresponding Author: Chun-Yang Zhang.)*

C. L. P. Chen and L. Chen are with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: philip.chen@ieee.org; longchen@umac.mo).

C.-Y. Zhang was with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China. He is now with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China (e-mail: cyzhangfst@gmail.com).

M. Gan is with the Department of Computer and Information Science, Hefei University of Technology, Hefei 230009, China (e-mail: aganmin@aliyun.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2015.2406889

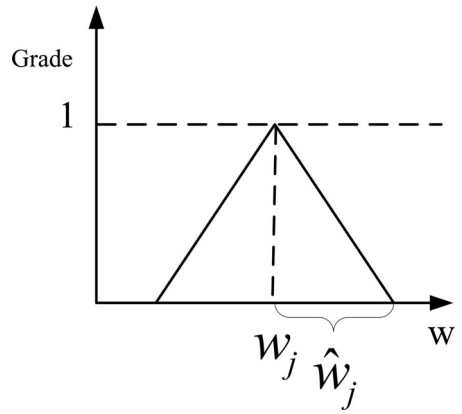


Fig. 2. Symmetric triangular fuzzy number.

and their high-level features is introduced into the model, the fuzzy RBMs demonstrate competitive performances in both data representation capability and robustness to cope with noises. These similar merits also can be initially found in fuzzy regression [19] and fuzzy support vector machine (SVM) [20].

The fuzzy RBMs, which are also designed to boost the development of deep learning from the building component (RBMs) of deep networks, has never been introduced before. The stochastic gradient descent method integrated with Monte Carlo Markov chain (MCMC) approach is employed to train the proposed fuzzy RBMs. This kind of learning methods are commonly used in training RBMs, and proved to be very efficient [12], [21], [22]. Other learning approaches like Bayesian estimation methods have also been developed to learn RBM in [23] and [24]. Gaussian RBMs, conditional RBMs, temporal RBMs, and recurrent RBMs are developed through modifying the structure of RBMs. Alternatively, fuzzy RBMs are proposed from different perspective of extending the relationships between visible and hidden units. Therefore, fuzzy RBMs can also be further developed by taking consideration of others variants of RBMs.

The rest of the paper is organized as follows. In Section II, the preliminaries about fuzzy sets, fuzzy functions, and their notations are presented. The proposed FRBM and its learning algorithm are introduced in Section III. After that, the outstanding performance of the FRBM model is verified by conducting experiments on bar-and-stripe (BAS) benchmark inpainting and MNIST handwritten digits classification in Section IV. Finally, the conclusion and remarks are drawn in Section V.

II. Preliminaries

A. Fuzzy Number

Restricting the discussion on symmetric triangular fuzzy number [25], $\overline{\mathbf{W}} = [\overline{W}_1, \ldots, \overline{W}_n]^T$ is regarded as a vector of them. The membership function for jth fuzzy number $\overline{W}_j$ can be defined as

$$\overline{W}_j(w) = \max\left\{1 - \frac{|w - w_j|}{\widehat{w}_j}, 0\right\} \tag{1}$$

where w_j is the center of the fuzzy number, and $\widehat{w}_j$ is the width of the fuzzy number as illustrated in Fig. 2. A fuzzy set is specified by placing a "bar" over a capital letter, such as $\overline{W}_j$. $\overline{W}_j(w)$ represents the membership value at w.

B. Alpha-Cuts

If $\overline{A}$ is a fuzzy set, the α-cut of $\overline{A}$, denoted by $\overline{A}[\alpha]$, is defined as

$$\overline{A}[\alpha] = \{x \in \Omega | \overline{A}(x) \geq \alpha\} \tag{2}$$

where $0 < \alpha \leq 1$.

C. Fuzzy Function

The *fuzzy function* $\overline{f}$, which is extended from real-value function $f : Y = f(x, \mathbf{W})$, is defined by

$$\overline{Y} = \overline{f}(x, \overline{\mathbf{W}}) \tag{3}$$

where $\overline{Y}$ is the dependent fuzzy output set, $\mathbf{W}$ and $\overline{\mathbf{W}}$ are parameters in the two functions [26].

1) Extension Principle: The membership function deduced from the extension principle can be expressed as

$$\overline{Y}(y) = \sup_{\mathbf{W}}\{\min(\overline{W}_1(W_1), \ldots, \overline{W}_n(W_n)) | f(x, \mathbf{W}) = y\} \tag{4}$$

where $\mathbf{W} = (W_1, \ldots, W_n)^T$, and $\overline{\mathbf{W}} = (\overline{W}_1, \ldots, \overline{W}_n)^T$.

2) Alpha-Cuts of $\overline{Y}$: If f is continuous, the α-cut of $\overline{Y}$, i.e., $\overline{Y}[\alpha] = [\overline{Y}_1(\alpha), \overline{Y}_2(\alpha)]$, has following expression:

$$\begin{cases} \overline{Y}_1(\alpha) = \min\{Y(\mathbf{W}, x) | \mathbf{W} \in \overline{\mathbf{W}}[\alpha]\} \\ \overline{Y}_2(\alpha) = \max\{Y(\mathbf{W}, x) | \mathbf{W} \in \overline{\mathbf{W}}[\alpha]\}. \end{cases} \tag{5}$$

D. Interval Arithmetic

For two intervals $[a, b]$ and $[c, d]$, that are two subsets of the real domain, the fundamental operations of interval arithmetic [27] are defined as follows:

$$\begin{aligned} [a,b] + [c,d] &= [a + c, b + d] \\ [a,b] - [c,d] &= [a - c, b - d] \\ [a,b] \times [c,d] &= [\min(a \times c, a \times d, b \times c, b \times d) \\ &\quad \max(a \times c, a \times d, b \times c, b \times d)] \\ [a,b] \div [c,d] &= [\min(a \div c, a \div d, b \div c, b \div d) \\ &\quad \max(a \div c, a \div d, b \div c, b \div d)], 0 \notin [c,d]. \end{aligned}$$

To calculate the membership function by using extension principle (3) and (4) is infeasible, since it involves the maximization and minimization of the original function. There is another efficient way to extend the function to be the corresponding fuzzy one. That is to use α-cuts and interval arithmetic

$$\overline{Y}[\alpha] = f(x, \overline{\mathbf{W}}[\alpha]) \tag{6}$$

where $\overline{\mathbf{W}}[\alpha]$ are intervals that are easy to calculate. Interval arithmetic then can be employed to finish the computation of membership function of the fuzzy function. However, interval arithmetic may become NP hard problems when f is very complex [28].

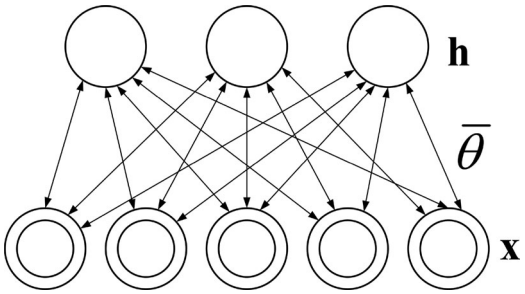


Fig. 3. Fuzzy restricted Boltzmann Machine (FRBM).

III. Fuzzy-Restricted Boltzmann Machine and Its Learning Algorithm

A. Fuzzy-Restricted Boltzmann Machine

The proposed novel FRBM is illustrated in Fig. 3, in which the connection weighs and biases are fuzzy parameters denoted by $\overline{\boldsymbol{\theta}}$. There are several merits of the FRBM model. The first one is that the FRBM has much better representation than the regular RBM in modeling probabilities over visible and hidden units. Specifically, the RBM is only a special case of the FRBM when no fuzziness exists in the FRBM model. The second one is that the robustness of the FRBM model surpasses RBM model. The FRBM shows out more robustness when it comes to the fitting of the model with noisy data. All these advantages spring from the fuzzy extension of the relationships between cross-layer variables, and inherit the characteristics of fuzzy models.

Since the FRBM is an extension of the RBM model; therefore, the discussion starts from a brief introduction on the RBM model. An RBM is an energy-based probabilistic model, in which the probability distribution is defined through an energy function. Its probability is defined as

$$P(\mathbf{x},\mathbf{h},\boldsymbol{\theta}) = \frac{e^{-E(\mathbf{x},\mathbf{h},\boldsymbol{\theta})}}{Z} \tag{7}$$

$$Z = \sum_{\tilde{\mathbf{x}}}\sum_{\tilde{\mathbf{h}}} e^{-E(\tilde{\mathbf{x}},\tilde{\mathbf{h}},\boldsymbol{\theta})} \tag{8}$$

where $E(\mathbf{x},\mathbf{h},\boldsymbol{\theta})$ is the energy function, $\boldsymbol{\theta}$ are the parameters governing the model, Z is the normalizing factor which is called the *partition function*, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{h}}$ are two vector variables representing visible and hidden units that are used to traverse and summarize all the configurations of units on the graph. The energy function for the RBM is defined by

$$E(\mathbf{x},\mathbf{h},\boldsymbol{\theta}) = -\mathbf{b}^T\mathbf{x} - \mathbf{c}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{x} \tag{9}$$

where b_j and c_i are the offsets, and W_{ij} is the connection weight between jth visible unit and ith hidden unit, and $\boldsymbol{\theta} = \{\mathbf{b},\mathbf{c},\mathbf{W}\}$.

To establish FRBM, it is necessary to first define the fuzzy energy function for the model. The fuzzy energy function can be extended from (9) in accordance with extension principle as follows:

$$\overline{E}(\mathbf{x},\mathbf{h},\overline{\boldsymbol{\theta}}) = -\overline{\mathbf{b}}^T\mathbf{x} - \overline{\mathbf{c}}^T\mathbf{h} - \mathbf{h}^T\overline{\mathbf{W}}\mathbf{x} \tag{10}$$

where $\overline{E}(\mathbf{x},\mathbf{h},\overline{\boldsymbol{\theta}})$ is a fuzzified energy function, and $\overline{\boldsymbol{\theta}} = \{\overline{\mathbf{b}},\overline{\mathbf{c}},\overline{\mathbf{W}}\}$ are fuzzy parameters. Correspondingly, the fuzzy free energy $\overline{\mathcal{F}}$, which marginalize hidden units and map (7) into a simpler one, is deduced as

$$\overline{\mathcal{F}}(\mathbf{x},\overline{\boldsymbol{\theta}}) = -\log\sum_{\tilde{\mathbf{h}}} e^{-\overline{E}(\mathbf{x},\tilde{\mathbf{h}},\overline{\boldsymbol{\theta}})} \tag{11}$$

where $\overline{\mathcal{F}}$ is extended from crisp free energy function $\mathcal{F}$

$$\mathcal{F}(\mathbf{x},\boldsymbol{\theta}) = -\log\sum_{\tilde{\mathbf{h}}} e^{-E(\mathbf{x},\tilde{\mathbf{h}},\boldsymbol{\theta})}. \tag{12}$$

If the fuzzy free energy function is directly employed to define the probability, it leads to a fuzzy probability [27]. Finally, the optimization in learning process turns into a fuzzy maximum likelihood problem. However, this kind of problem is quite intractable because the fuzzy objective function is nonlinear and the membership function is difficult to compute, since the computation of its alpha-cuts become NP-hard problems [29]. Therefore, it is necessary to transform the problem into regular maximum likelihood problem by defuzzifying the fuzzy free energy function (11). The center of area (centroid) method [30] is employed to defuzzify the fuzzy free energy function $\overline{\mathcal{F}}(\mathbf{x})$. Then, the likelihood function can be defined by the defuzzified fuzzy free energy function. Consequently, the fuzzy optimization problem becomes real-valued problem, and conventional optimization approaches can be directly applied to find the optimal solutions. The centroid of fuzzy number $\overline{\mathcal{F}}(\mathbf{x})$ is denoted by $\mathcal{F}_c(\mathbf{x})$, and has the following form:

$$\mathcal{F}_c(\mathbf{x},\overline{\boldsymbol{\theta}}) = \frac{\int \boldsymbol{\theta}\mathcal{F}(\mathbf{x},\boldsymbol{\theta})d\boldsymbol{\theta}}{\int \mathcal{F}(\mathbf{x},\boldsymbol{\theta})d\boldsymbol{\theta}}, \boldsymbol{\theta}\in\overline{\boldsymbol{\theta}}. \tag{13}$$

Naturally, after the fuzzy free energy is defuzzified, the probability can be defined as

$$P_c(\mathbf{x},\overline{\boldsymbol{\theta}}) = \frac{e^{-\mathcal{F}_c(\mathbf{x};\overline{\boldsymbol{\theta}})}}{Z}, \qquad Z = \sum_{\tilde{\mathbf{x}}} e^{-\mathcal{F}_c(\tilde{\mathbf{x}},\overline{\boldsymbol{\theta}})}. \tag{14}$$

In the fuzzy RBM model, the objective function is the negative log-likelihood, which is given by

$$\mathcal{L}(\overline{\boldsymbol{\theta}},\mathcal{D}) = -\sum_{\mathbf{x}\in\mathcal{D}}\log P_c(\mathbf{x},\overline{\boldsymbol{\theta}}) \tag{15}$$

where $\mathcal{D}$ is the training dataset.

The learning problem is to find optimal solutions for parameters $\overline{\boldsymbol{\theta}}$ that minimize the objective function $\mathcal{L}(\overline{\boldsymbol{\theta}},\mathcal{D})$, i.e.

$$\min_{\overline{\boldsymbol{\theta}}} \quad \mathcal{L}(\overline{\boldsymbol{\theta}},\mathcal{D}). \tag{16}$$

In the following section, the detailed procedure to address the dual problem of maximum likelihood by utilizing stochastic gradient descent method will be investigated.

B. Fuzzy-Restricted Boltzmann Machines Learning Algorithm

In order to solve the optimization problem (16), it is required to first finish the defuzzification process of the fuzzy free energy function in some viable ways. However, it is infeasible to defuzzify the fuzzy free energy function by using (13) which involves integrals. Alternatively, the centroid is calculated by employing a discrete form, which associates with a number of

alpha-cuts of the fuzzy function. Therefore, the alpha-cuts of the fuzzy free energy function and interval arithmetic are first investigated to obtain an approximation of the centroid.

1) Alpha-Cuts of Fuzzy Free Energy Function: As supposed, $\overline{\boldsymbol{\theta}}$ is a vector of symmetric triangular fuzzy numbers, and its α-cut is $\overline{\boldsymbol{\theta}}[\alpha] = [\boldsymbol{\theta}_L, \boldsymbol{\theta}_R]$, where $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_R$ are lower and upper bounds of the interval with respect to α, respectively. $\overline{\mathcal{F}}(\mathbf{x}, \overline{\boldsymbol{\theta}})$ is often a triangular-shaped fuzzy number for nonlinear functions [27]. However, the fuzzy free energy is monotonic decreasing function with respect to parameters $\overline{\boldsymbol{\theta}}$ when $\mathbf{x}$ and $\mathbf{h}$ are non-negative. Therefore, according to interval arithmetic, the α-cut of $\overline{\mathcal{F}}(\mathbf{x}, \overline{\boldsymbol{\theta}})$ can be given by

$$\begin{aligned}\overline{\mathcal{F}}(\mathbf{x}, \overline{\boldsymbol{\theta}})[\alpha] &= \mathcal{F}(\mathbf{x}, \overline{\boldsymbol{\theta}}[\alpha]) \\ &= [\mathcal{F}(\mathbf{x}, \boldsymbol{\theta}_R), \mathcal{F}(\mathbf{x}, \boldsymbol{\theta}_L)]. \end{aligned} \tag{17}$$

2) Approximation of Centroid: An approximation of the centroid of the fuzzy free energy function is provided by discretizing the fuzzy output $\overline{\mathcal{F}}(\mathbf{x}, \overline{\boldsymbol{\theta}})$ and calculating M number of its α-cuts. By combining with these α-cuts, the approximate centroid is given by

$$\mathcal{F}_c(\mathbf{x}, \overline{\boldsymbol{\theta}}) \approx \frac{\sum_{i=1}^{M} \alpha_i [\mathcal{F}(\mathbf{x}, \boldsymbol{\theta_{iL}}) + \mathcal{F}(\mathbf{x}, \boldsymbol{\theta_{iR}})]}{2 \sum_{i=1}^{M} \alpha_i} \tag{18}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N), \boldsymbol{\alpha} \in [0,1]^N$ and $\overline{\boldsymbol{\theta}}[\alpha_i] = [\boldsymbol{\theta_{iL}}, \ \boldsymbol{\theta_{iR}}]$.

As all the α-cuts are bounded intervals [31], for convenience, we only consider a special case, in which fuzzy numbers are degraded into intervals ($\alpha = 1$). Let $\overline{\boldsymbol{\theta}} = [\boldsymbol{\theta}_L, \boldsymbol{\theta}_R]$. According to (18), the free energy function can be written as

$$\mathcal{F}_c(\mathbf{x}, \overline{\boldsymbol{\theta}}) \approx \frac{1}{2} \left[\mathcal{F}(\mathbf{x}, \boldsymbol{\theta}_L) + \mathcal{F}(\mathbf{x}, \boldsymbol{\theta}_R) \right]. \tag{19}$$

After defuzzifying the fuzzy free energy, the probability defined on it returns to (14). Then, the problem is transformed into regular optimization problem, which can be solved by the gradient descend-based stochastic maximum likelihood method.

3) Gradient Related Optimization: The gradients of negative log-probability with respect to $\boldsymbol{\theta}_L$ then have a particularly form (see Appendix A)

$$-\frac{\partial \log P_c(\mathbf{x}, \overline{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_L} = \frac{\partial \mathcal{F}_c(\mathbf{x}, \boldsymbol{\theta}_L)}{\partial \boldsymbol{\theta}_L} - E_P \left[\frac{\partial \mathcal{F}_c(\mathbf{x}, \boldsymbol{\theta}_L)}{\partial \boldsymbol{\theta}_L} \right] \tag{20}$$

where $E_P(\cdot)$ means the expectation over the target probability distribution P. Similarly, the gradients of (16) with respect to $\boldsymbol{\theta}_R$ is given by

$$-\frac{\partial \log P_c(\mathbf{x}, \overline{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_R} = \frac{\partial \mathcal{F}_c(\mathbf{x}, \boldsymbol{\theta}_R)}{\partial \boldsymbol{\theta}_R} - E_P \left[\frac{\partial \mathcal{F}_c(\mathbf{x}, \boldsymbol{\theta}_R)}{\partial \boldsymbol{\theta}_R} \right]. \tag{21}$$

It is usually difficult to compute these gradients analytically, as it involves the computation of $E_P \left[\frac{\partial \mathcal{F}(\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]$ ($\boldsymbol{\theta} = \boldsymbol{\theta}_L$ or $\boldsymbol{\theta}_R$). This is nothing less than an expectation over all possible configurations of the input $\mathbf{x}$. An estimation of the expectation can be obtained by using a fixed number of model samples. Assume there are $\mathcal{N}$ samples that sampled from approximate distribution, then, we have

$$-\frac{\partial \log P_c(\mathbf{x}, \overline{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} \approx \frac{\partial \mathcal{F}_c(\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \frac{1}{|\mathcal{N}|} \sum_{\tilde{\mathbf{x}} \in \mathcal{N}} \frac{\partial \mathcal{F}_c(\tilde{\mathbf{x}}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{22}$$

4) Conditional Probability: For the RBM, the conditional energy-based probabilities [11] are defined as

$$P(\mathbf{h}|\mathbf{x}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{\sum_{\tilde{\mathbf{h}}} e^{-E(\mathbf{x},\tilde{\mathbf{h}})}} \tag{23}$$

$$P(\mathbf{x}|\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{\sum_{\tilde{\mathbf{x}}} e^{-E(\tilde{\mathbf{x}},\mathbf{h})}}. \tag{24}$$

In the commonly studied case of the RBM with binary units, where x_j and $h_i \in \{0,1\}$, the probabilistic version of the usual neuron activation functions can be derived from conditional probabilities. They have the following affine forms:

$$P(h_i = 1|\mathbf{x}) = \frac{e^{c_i + W_i \mathbf{x}}}{1 + e^{c_i + W_i \mathbf{x}}} = \sigma(c_i + W_i \mathbf{x}) \tag{25}$$

$$P(x_j = 1|\mathbf{h}) = \frac{e^{b_j + W_{\cdot j}^T \mathbf{h}}}{1 + e^{b_j + W_{\cdot j}^T \mathbf{h}}} = \sigma\left(b_j + W_{\cdot j}^T \mathbf{h}\right) \tag{26}$$

where W_i and $W_{\cdot j}$ denote the ith row and jth column of W, respectively, σ is logistic sigmoid function

$$\sigma(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}}.$$

For fuzzy RBM, the conditional probabilities also become fuzzy and can be extended from (25) and (26) as

$$\begin{aligned}\overline{P}(h_i = 1|\mathbf{x}) &= \sigma(\overline{c}_i + \overline{W}_i \mathbf{x}) \\ \overline{P}(x_j = 1|\mathbf{h}) &= \sigma(\overline{b}_j + \overline{W}_{\cdot j}^T \mathbf{h}).\end{aligned}$$

After defuzzifying the objective function, MCMC method can be employed to sample from these conditional distributions. This process is crucial to approximate the objective function due to the difficulties to calculate the expectations. For the predefined α, the α-cut of fuzzy conditional probabilities are consistent with the targeting probability distribution. They are given as follows:

$$\begin{aligned}\overline{P}(h_i = 1|\mathbf{x})[\alpha] &= [P_L(h_i = 1|\mathbf{x}), P_R(h_i = 1|\mathbf{x})] \\ \overline{P}(x_j = 1|\mathbf{h})[\alpha] &= [P_L(x_j = 1|\mathbf{h}), P_R(x_j = 1|\mathbf{h})]\end{aligned}$$

where $P_L(h_i|\mathbf{x})$, $P_R(h_i|\mathbf{x})$, $P_L(x_j|\mathbf{h})$, and $P_R(x_j|\mathbf{h})$ are the conditional probabilities with respect to the lower bounds and upper bounds of the parameters governing the model. They have following forms:

$$\begin{aligned}P_L(h_i|\mathbf{x}) &= P(h_i|\mathbf{x}; \theta_{\mathbf{L}}) = \sigma\left(c_i^L + W_i^L \mathbf{x}\right) \\ P_R(h_i|\mathbf{x}) &= P(h_i|\mathbf{x}; \theta_{\mathbf{R}}) = \sigma\left(c_i^R + W_i^R \mathbf{x}\right)\end{aligned} \tag{27}$$

and

$$\begin{aligned}P_L(x_j|\mathbf{h}) &= P(x_j|\mathbf{h}; \theta_{\mathbf{L}}) = \sigma\left(b_j^L + W_{\cdot j}^L \mathbf{h}\right) \\ P_R(x_j|\mathbf{h}) &= P(h_j|\mathbf{h}; \theta_{\mathbf{R}}) = \sigma\left(b_j^R + W_{\cdot j}^R \mathbf{h}\right).\end{aligned} \tag{28}$$

There are six kinds of parameters for visible unit j and hidden unit i in the FRBM model, i.e., lower bound of connection

weight W_{ij}^L, visible bias b_j^L, hidden bias c_i^L, and their upper bounds W_{ij}^R, b_j^R, and c_i^R. For simplicity, the energy function is denoted by a sum of terms associated with only one hidden unit

$$E(\mathbf{x}, \mathbf{h}) = -\mu(\mathbf{x}) - \sum_{i=1}^{m} \phi_i(\mathbf{x}, h_i) \tag{29}$$

where

$$\mu(\mathbf{x}) = \mathbf{b}^T \mathbf{x}, \qquad \phi_i(\mathbf{x}, h_i) = -h_i(c_i + W_i \mathbf{x}). \tag{30}$$

Then, the free energy of RBM with binary units can be further simplified explicitly to (see Appendix B)

$$\mathcal{F}(x) = -\mathbf{b}^T \mathbf{x} - \sum_{i=1}^{m} \log(1 + e^{(c_i + W_i \mathbf{x})}). \tag{31}$$

The gradients of the free energy can be calculated explicitly when the RBM has binary units and the energy function has form (29).

No matter whether the fuzzy parameters in fuzzy RBM are symmetric or not, their α-cuts are always intervals with lower and upper bounds that need to be learned in the training phase. According to (19)–(21), all the gradients (see details in Appendix C) can be obtained. Associate with (20) and (31), it is easy to get the following negative log-likelihood gradients for the fuzzy RBM with binary units:

$$\begin{aligned}
-\frac{\partial \log P_c(\mathbf{x})}{\partial W_{ij}^L} &= E_P\left[P_L(h_i|\mathbf{x}) \cdot x_j^L\right] - P_L(h_i|\mathbf{x}) \cdot x_j^L \\
-\frac{\partial \log P_c(\mathbf{x})}{\partial c_i^L} &= E_P[P_L(h_i|\mathbf{x})] - P_L(h_i|\mathbf{x}) \\
-\frac{\partial \log P_c(\mathbf{x})}{\partial b_j^L} &= E_P[P_L(x_j|\mathbf{h})] - x_j^L \\
-\frac{\partial \log P_c(\mathbf{x})}{\partial W_{ij}^R} &= E_P[P_R(h_i|\mathbf{x}) \cdot x_j^R] - P_R(h_i|\mathbf{x}) \cdot x_j^R \\
-\frac{\partial \log P_c(\mathbf{x})}{\partial c_i^R} &= E_P[P_R(h_i|\mathbf{x})] - P_R(h_i|\mathbf{x}) \\
-\frac{\partial \log P_c(\mathbf{x})}{\partial b_j^R} &= E_P[P_R(x_j|\mathbf{h})] - x_j^R
\end{aligned}$$

where $P_c(\mathbf{x})$ is the centroid probability defined though (14) combining with (18) and (19).

5) Contrastive Divergence: When to approximate the expectations, samples of $P_L(\mathbf{x})$ and $P_R(\mathbf{x})$ can be obtained by running two Markov chains to convergence, using Gibbs sampling as the transition operator. One is for the lower bounds, and the other is for the upper bounds. *Gibbs sampling* of the joint distribution over N random variables $S = (S_1, ..., S_N)$ is done through a sequence of N sampling substeps of the form $S_i \sim P(S_i|S_{-i})$, where S_{-i} contains the $N-1$ other random variables in S excluding S_i.

For both the RBM and fuzzy RBM model, S consists of the set of visible and hidden units. However, since they are conditionally independent, one can perform block Gibbs sampling. A step in the Markov chain, visible units are sampled given hidden units, hidden units are sampled given visible units, as illustrated in Fig. 4

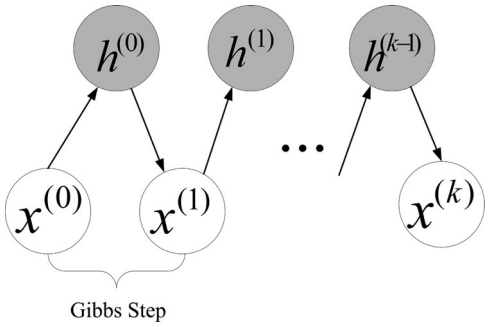


Fig. 4. k-step Gibbs sampling.

$$\mathbf{h}^{(k+1)} \sim P(\mathbf{h}^{(k+1)}|\mathbf{x}^{(k)}) \tag{32}$$

$$\mathbf{x}^{(k+1)} \sim P(\mathbf{x}^{(k+1)}|\mathbf{h}^{(k+1)}). \tag{33}$$

As $k \to \infty$, samples $(\mathbf{x}^{(k)}, \mathbf{h}^{(k)})$ are guaranteed to be accurate samples of $P(\mathbf{x}, \mathbf{h})$. However, Gibbs sampling is very time-consuming as k needs to be large enough. An efficient learning approach, called contrastive divergence (CD) [32] learning was proposed in 2002. It shows that learning process still performs very well, even though only a number of steps are run in Markov chain [33]. The CD learning uses two tricks to speed up the sampling process. The first on it to initialize the Markov chain with a training example, and the second one is to obtain samples after only k-steps of Gibbs sampling. This is regarded as CD-k learning algorithm. A lot of experiments show that the performances of the approximations are still very good when $k = 1$.

CD is a different function compared with Kullback–Leibler divergence to measure the difference between approximated distribution and true distribution. Why is CD learning efficient? It is because that CD learning provides an approximation of the log-likelihood gradient that has been found to be a successful update rule for training probabilistic models. Variational justification can provide a theoretical proof to the convergence of the learning processes [11], [34]. Conducting CD-1 learning by using (25) and (26), namely $\mathbf{x} = \mathbf{x}^{(0)} \to \mathbf{h}^{(0)} \to \mathbf{x}^{(1)} \to \mathbf{h}^{(1)}$, it is easy to get the updating rules for all the parameters ($\theta_{\mathbf{L}}$ and $\theta_{\mathbf{R}}$) in the FRBM model. The pseudocode is demonstrated in Algorithm 1.

IV. Experimental Results

In this section, the representation capabilities of the RBM and FRBM will be examined on two datasets. One is BAS benchmark dataset, and another one is MNIST handwritten digits dataset. The RBM and proposed FRBM will be trained in an unsupervised way on BAS dataset to recovery incomplete images. The training on noisy BAS dataset is also considered to compare the robustness of the two models. To compare the classification performances of the two models in experiments based on MNIST handwritten digits dataset, both RBM and FRBM are trained in a supervised manner.

On account of the partition function Z in (14), it is very tricky to track the RBM and FRBM training process in unsupervised

Algorithm 1 Updating Rules for FRBM

Input: $\mathbf{x}^{(0)}$ is a training sample from the training distribution for the RBM;
ϵ is the learning rate for updating the parameters;
$\mathbf{W}^L$ and $\mathbf{W}^R$ are the visible-hidden connection weight matrix;
$\mathbf{b}^L$ and $\mathbf{b}^R$ are the bias vectors for input units;
$\mathbf{c}^L$ and $\mathbf{c}^R$ are the bias vectors for hidden units;
Output: The updated parameters in the RBM: $\mathbf{W}^L$, $\mathbf{b}^L$, $\mathbf{c}^L$, $\mathbf{W}^R$, $\mathbf{b}^R$, $\mathbf{c}^R$.
for all hidden units i **do**
compute $P_L(h_i^{L(0)} = 1|\mathbf{x}^{(0)})$ and $P_R(h_i^{R(0)} = 1|\mathbf{x}^{(0)})$ by using (27);
sample $h_i^{L(0)} \in \{0, 1\}$ from $P_L(h_i^{L(0)}|\mathbf{x}^{(0)})$;
sample $h_i^{R(0)} \in \{0, 1\}$ from $P_R(h_i^{R(0)}|\mathbf{x}^{(0)})$;
end for
for all visible units j **do**
compute $P_L(x_j^{L(1)} = 1|\mathbf{h}^{L(0)})$ and $P_R(x_j^{R(1)} = 1|\mathbf{h}^{R(0)})$ by using (28);
sample $x_j^{L(1)} \in \{0, 1\}$ from $P_L(x_j^{L(1)}|\mathbf{h}^{L(0)})$;
sample $x_j^{R(1)} \in \{0, 1\}$ from $P_R(x_j^{R(1)}|\mathbf{h}^{R(0)})$;
end for
for all hidden units i **do**
compute $P_L(h_i^{L(1)} = 1|x^{L(1)})$ and $P_R(h_i^{R(1)} = 1|\mathbf{x}^{R(1)})$ by using (27);
sample $h_i^{L(1)} \in \{0, 1\}$ from $P_L(h_i^{L(1)}|\mathbf{x}^{L(1)})$;
sample $h_i^{R(1)} \in \{0, 1\}$ from $P_R(h_i^{R(1)}|\mathbf{x}^{R(1)})$;
end for
$\mathbf{W}^L = \mathbf{W}^L + \epsilon(\mathbf{x}^{(0)} \cdot P_L(\mathbf{h}^{L(0)} = 1|\mathbf{x}^{(0)}) - \mathbf{x}^{L(1)} \cdot P_L(\mathbf{h}^{L(1)} = \mathbf{1}|\mathbf{x}^{L(1)}))$;
$\mathbf{b}^L = \mathbf{b}^L + \epsilon(\mathbf{x}^{(0)} - \mathbf{x}^{L(1)})$;
$\mathbf{c}^L = \mathbf{c}^L + \epsilon(P_L(\mathbf{h}^{L(0)} = \mathbf{1}|\mathbf{x}^{(0)}) - P_L(\mathbf{h}^{L(1)} = \mathbf{1}|\mathbf{x}^{L(1)}))$;
$\mathbf{W}^R = \mathbf{W}^R + \epsilon(\mathbf{x}^{(0)} \cdot P_R(\mathbf{h}^{R(0)} = \mathbf{1}|\mathbf{x}^{(0)}) - \mathbf{x}^{R(1)} \cdot P_R(\mathbf{h}^{R(1)} = 1|\mathbf{x}^{R(1)}))$;
$\mathbf{b}^R = \mathbf{b}^R + \epsilon(\mathbf{x}^{(0)} - \mathbf{x}^{R(1)})$;
$\mathbf{c}^R = \mathbf{c}^R + \epsilon(P_R(\mathbf{h}^{R(0)} = \mathbf{1}|\mathbf{x}^{(0)}) - P_R(\mathbf{h}^{R(1)} = \mathbf{1}|\mathbf{x}^{R(1)}))$;
return $\mathbf{W}^L$, $\mathbf{b}^L$, $\mathbf{c}^L$,$\mathbf{W}^R$, $\mathbf{b}^R$, $\mathbf{c}^R$;

learning. However, one can observe the learning process by inspecting the negative samples in BAS benchmark dataset. For the MNIST handwritten digits recognition problem, the classification error rate of the testing dataset is also one of significant criteria except the observation of the negative samples. All the experiments in this paper are run on MATLAB (R2009a) in 32-b Windows 7 OS. The computational platform has two 3.40-GHz CPUs and 4.00-GB memory. These experimental results are demonstrated in the following.

A. Bar-and-Stripe Benchmark Inpainting

In this part, the practical aspects of training RBM and FRBM based on BAS benchmark dataset [35] will be discussed. There

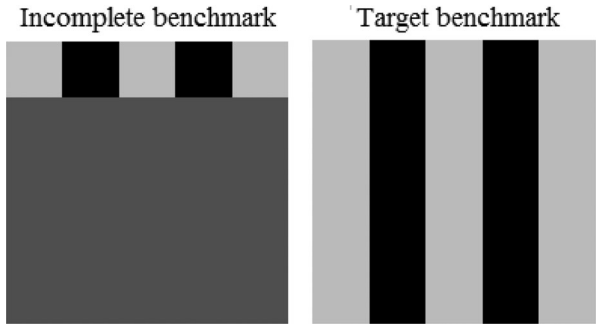


Fig. 5. BAS benchmark.

are 60 000 BAS benchmarks created randomly, and each of them has 5×5 units. Every benchmark is generated by randomly choosing a direction (row or column) with equal probability first, and then set the states for all units of the row or column uniformly at random. Therefore, the dataset consists of 64 different BAS benchmarks. One example for BAS benchmarks is illustrated in right part of Fig. 5. The RBM and FRBM models are used to learn the joint probability over the BAS benchmark.

As generative models, fuzzy RBM and RBM learn the probability distribution over visible variables and hidden features. Once they are well trained, they can generate samples from the probability distribution $P(\mathbf{x})$. For the BAS benchmark inpainting problem, it is supposed that there are l known visible units $x_1, \ldots, x_l$, while $x_{l+1}, \ldots, x_n$ are unknown observations. Then, one can evaluate the unknown observations from conditional probability $P(x_1, \ldots, x_l|x_{l+1}, \ldots, x_n)$. In this process, the inferences can be easily implemented, such as in image classification and inpainting.

For the image inpainting problem, only a part of the image is known. The task is to infer the unknown part. In Fig. 5, the left benchmark can be regarded as an incomplete image in which the pixels in the first row are clamped. The right benchmark is the target image that is required. For a clear illustration, the gray and black squares are used to denote binary pixel values. The objective is to recovery the benchmark by using the FRBM and RBM models. First, there are 60 000 BAS benchmarks generated to train the two models. Then, one can recover the benchmark by conducting Gibbs sampling over $P(x_1, \ldots, x_l|x_{l+1}, \ldots, x_n)$. The unknown observations are initially resigned to be 0.5.

In the following experiments conducted to evaluate FRBM on different energy functions and different types of fuzzy numbers for RBMs' parameters, the energy function defined as (10) is regarded as type-1 energy function (EF-1), and the energy of a joint configuration ignoring biases terms is regarded as type-2 energy function (EF-2). The fuzzy RBM with symmetric triangular fuzzy number is denoted as FRBM-STFN. Accordingly, fuzzy RBM with Gaussian membership function is denoted as FRBM-GMF. In the experiments, the FRBM and RBM model have 25 visible units and 50 hidden units. For the two different models, the minibatch learning scheme (100 samples each batch) and CD-1 learning approach are employed to speed up the training processes. The biases $\mathbf{b}$ and $\mathbf{c}$ are initialized with zero values, and connection weights are generated randomly from standard Gaussian distribution in the initial stages. The

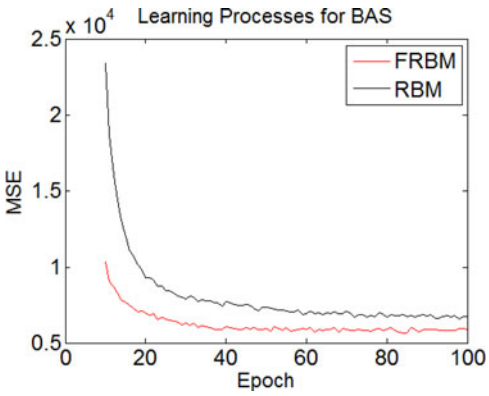


Fig. 6. Learning processes of RBM and FRBM based on BAS dataset.

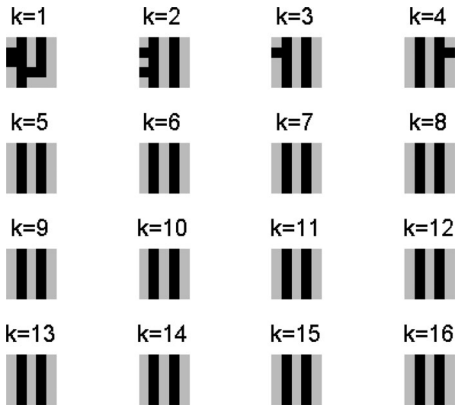


Fig. 7. BAS benchmark inpainting based on RBM.

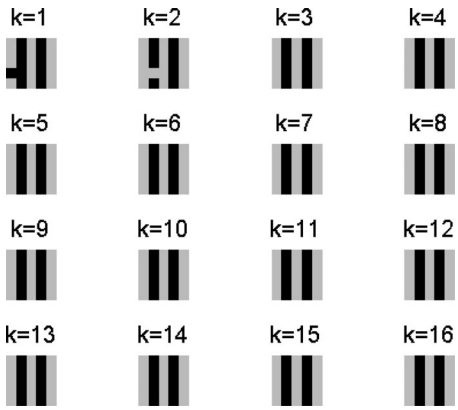


Fig. 8. BAS benchmark inpainting based on FRBM.

learning rates for updating W, $\mathbf{b}$ and $\mathbf{c}$ are set to be 0.05. The MSEs (the summation over reconstruction error of all the $60\,000$ training samples) produced in the two learning phases (50 hidden units) are shown in Fig. 6. It is easy to see that the FRBM model generates less reconstruction errors than the RBM model, which means the FRBM can learn the probability distribution more accurate than the traditional RBM model.

The comparative recovery results for the RBM and FRBM models are demonstrated in Figs. 7 and 8, respectively, where k denotes the Gibbs step in sampling. As the probabilistic model has randomness and uncertainty, two stable results are collected and shown in Figs. 7 and 8. These two figures show that the FRBM model needs fewer Gibbs steps to recover the image than the RBM model, since the distribution learnt by the FRBM is much closer to the real distribution. Otherwise, the FRBM also produces less negative predictions of pixels than the RBM in each Gibbs step. Therefore, one can conclude that the FRBM has stronger representation capability than the RBM. This advantage springs from the fuzzy relationship between the visible and hidden units. Intuitively, the parameters' searching space of RBM can be regarded as a subspace of the FRBM's searching space when the relationships are extended.

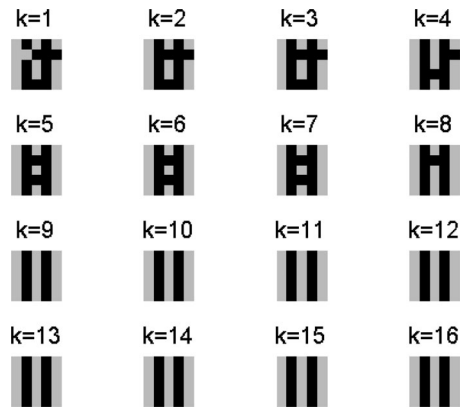


Fig. 9. BAS benchmark inpainting with 20% noises based on RBM.

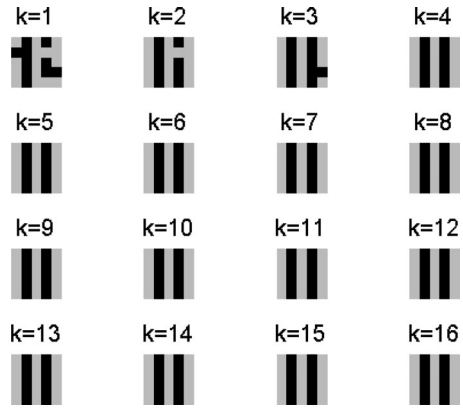


Fig. 10. BAS benchmark inpainting with 20% noises based on FRBM.

B. Noisy Bar-and-Stripe Benchmark Inpainting

To verify the robustness of the FRBM model, there are 10% and 20% training samples replaced with noisy training samples. Each noisy sample is generated by reversing all the pixel values in a row or column. Then, the FRBM and RBM are trained with these noisy samples to investigate the robustness of the two different models. The inpainting results are shown in Figs. 9 and 10. The same conclusion is that the FRBM model learn a more accurate distribution than the regular RBM model.

TABLE I
BAS BENCHMARK INPAINTING IN TESTING PROCESS: MEAN SQUARE ERROR (MSE), RECOVERY ERROR RATE (RER)

EF Type	Items	Noise Level	RBM	FRBM-STFN	FRBM-GMF
EF-1	MSE	0%	1271	256	248
		10%	6060	1944	1859
		20%	8777	4381	4197
	RER	0%	3.18%	0.64%	0.62%
		10%	15.15%	4.86%	4.65%
		20%	21.94%	10.95%	10.49%
	Learning time(s)	-	54	91	117
EF-2	MSE	0%	5921	3451	3293
		10%	8921	5022	4741
		20%	11093	8134	7584
	RER	0%	14.80%	8.63%	8.24%
		10%	22.30%	12.56%	11.85%
		20%	27.73%	20.34%	18.96%
	Learning time(s)	–	45	72	91

As there exists the randomness and uncertainty in the probabilistic models, it is necessary to analyze the statistical property of the MSE and recovery accuracy (error rate). The results are demonstrated in Table I. The MSEs are the error summation over 16 generated patterns (each has 25 pixel values) from 100 times of Gibbs sampling. The recovery error rate is percentage of negative recovery pixel values. The results show that the proposed FRBM makes a significant improvement to the regular RBM, also for the noisy case. Additionally, the results show that the energy function plays an important role in model establishment and should be defined according to the specific graph model. The performance of both RBM and FRBM models are much better when the biases terms are taken into account in energy functions for this case. The energy functions for current variants of RBM are designed by considering the structures of the variants. Otherwise, the membership function of fuzzy parameters also has effect on representation capability of the FRBM model. From the results in Table I, the FRBM model with Gaussian membership function has better representation of BAS dataset than that with symmetric triangular membership function. The learning time for FRBM model raises accordingly as the number of the parameters increases when fuzziness is introduced into the model.

C. Handwritten Digits Recognition

In this experiment, the supervised learning is carried out for the FRBMs and RBMs on MNIST handwritten digits dataset which contains 60 000 training and 10 000 test handwritten digit images [36]. Each image representing 0 to 9 is constructed with 28×28 pixels whose intensities range from 0 to 255. This dataset is useful to examine a large number of learning techniques and pattern recognition methods on real-world data. One fact should be clarified that it is not expected that the performances of the FRBM and RBM in this experiment can outperform other sophisticated machine learning algorithm, such as SVM, because they are not deep networks. However, DBM, DBN, and deep autoencoder have already surpassed the SVM approach in this experiment. Herein, the objective is to evaluate the representation capability of the FRBM model. Its deep architectures and corresponding pretraining and fine-tuning algorithms will be discussed in the future. It is optimistic that fuzzy deep autoencoder, fuzzy deep belief networks, and fuzzy deep Boltzmann machine, that are all consists of the fuzzy RBMs, will have significant improvements of their original versions.

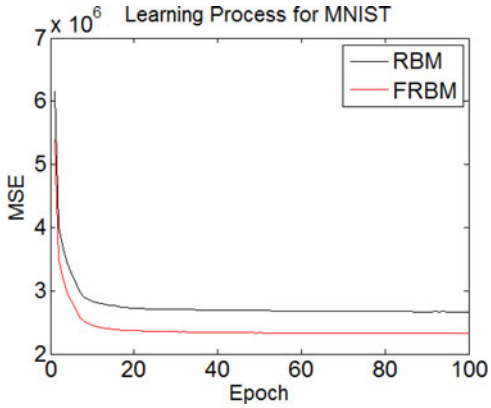


Fig. 11. Learning processes of RBM and FRBM based on MNIST dataset.

Fig. 12. MNIST samples.

The learning processes of RBM and FRBM for MNIST dataset are shown in Fig. 11. There are 60 000 samples in training dataset, and 100 hidden units for both RBM and FRBM. One epoch means one learning process over all training samples that are divided into 600 minibatches, and all the parameters are updated one time after each minibatch is finished. The MSEs in Fig. 11 are the total square reconstruction errors over all the training samples. It is clear that FRBM produces less MSE than RBM model, which means FRBM can learn the representation of the MNIST dataset better than the original RBM model.

In Figs. 12 and 13, 100 handwritten digits and their corresponding reconstructions are demonstrated. The reconstructed samples are generated after 20 epoch learning on the training dataset, where CD-1 learning was employed on the network with 100 hidden units. These two figures show that the two-layers FRBM can learn the probability over all the pixels very well. In other words, each reconstructed image approximates the original image closely.

Fig. 13. Reconstructed MNIST samples after 20 epoch with 100 hidden units.

TABLE II
MNIST CLASSIFICATION: ERROR RATE IS THE PERCENTAGE OF THE NEGATIVE RECOGNITION FOR 10 000 TESTING SAMPLES, m IS THE NUMBER OF HIDDEN UNITS

m	Criterion	RBM	FRBM-STFN	FRBM-GMF
m = 100	Error rate	667/10 000	523/10 000	486/10 000
	Learning time(s)	705	1138	1365
m = 400	Error rate	412/10 000	281/10 000	254/10 000
	Learning time(s)	1655	2896	3470
m = 800	Error rate	367/10 000	249/10 000	236/10 000
	Learning time(s)	2857	5199	6236
m = 1000	Error rate	361/10 000	245/10 000	230/10 000
	Learning time(s)	3635	6725	7734

The error rates of the classification for the two models are shown in Table II. From the results, it is reasonable to conclude that the FRBM outperforms the regular RBM model, as its representation capability to model the probability is increased by relaxing restrictions on the relationships between cross-layer units. When the number of hidden units are 100, the fuzzy RBM improves the recognition accuracy by 1.44%. It is a significant advancement in the MNIST digits recognition, as the state-of-the-art machine learning recognition algorithms only produce 0.5% improvement compared with the previous approaches. As the number of hidden units increasing, the fuzzy RBMs produce less error rate than the RBM but the improvement of the fuzzy RBM decreases to 1.06%. It is because that both the RBMs and fuzzy RBMs reach their capacities to model the data. As every image in MNIST dataset is different and with low resolution (noises already exist, see Fig. 12), it does not need to factitiously add noises into the images.

V. CONCLUSION

In order to improve the representation capability and robustness of the RBM model, a novel FRBM model is proposed, in which the connection weighs and biases between visible and hidden units are fuzzy numbers. It means that the relationship between input and its high level features (hidden units in the graph) are extended. After relaxing the relationship between the random variables by introducing fuzzy parameters, the RBM is only a special case of the FRBM when no fuzziness exists in the fuzzy model. In other words, the parameter searching space of RBM is only a subspace of the FRBM's searching space. Therefore, the FRBM has more powerful representation capability than the regular RBM. On the other hand, the robustness of the FRBM is also more powerful than the RBM model. These merits attribute to the fuzziness of the FRBM model.

Both the RBM and FRBM can be trained in both supervised and unsupervised way, the performances of the FRBM are verified in the experiments conducted on BAS benchmark inpainting problem and MNIST handwritten digits classification problem. The powerful robustness of the FRBM model is also examined on BAS benchmarks with different levels of noises. From those experiments, it is reasonable to conclude that the proposed FRBM has more representation capability to model the probability distribution than the RBM. In the meantime, the FRBM also shows out more powerful robustness that is necessary to address the noisy problem in training datasets and variational real-valued applications.

There still exists many open problems for the RBM and FRBM, such as model selection and developments of deep networks based on them. For the model selection problem, it is very hard to determine how many units in hidden layer, and how many hidden layers in deep architectures. Other setting problems, such as Gibbs step, learning rate, and batch learning, also impact the performance of the models. On the other hand, the computational cost of training their deep architectures is so high that needs to be speed up in some more efficient optimization approaches [37]. These are the common challenges for training deep networks. However, after conforming the improvement of the FRBM, one can have enough confidence to develop its deep architectures and corresponding learning algorithms in the coming future, such as fuzzy deep autoencoder, fuzzy deep belief networks, and fuzzy deep Boltzmann machine.

APPENDIX A

In the following, the detailed process for calculating the gradients of the negative log-likelihood function is presented as

$$\begin{aligned} -\frac{\partial \log P(\mathbf{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial \mathcal{F}_c(\mathbf{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \frac{1}{Z}\sum_{\tilde{\mathbf{x}}} e^{-\mathcal{F}_c(\tilde{\mathbf{x}},\boldsymbol{\theta})}\frac{\partial \mathcal{F}_c(\tilde{\mathbf{x}},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\ &= \frac{\partial \mathcal{F}_c(\mathbf{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \sum_{\tilde{\mathbf{x}}} P(\tilde{\mathbf{x}})\frac{\partial \mathcal{F}_c(\tilde{\mathbf{x}},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\ &= \frac{\partial \mathcal{F}_c(\mathbf{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - E_P\left[\frac{\partial \mathcal{F}_c(\mathbf{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]. \end{aligned} \tag{34}$$

APPENDIX B

As the energy function is denoted by

$$E(\mathbf{x},\mathbf{h}) = -\mu(\mathbf{x}) + \sum_i \phi_i(\mathbf{x},h_i) \tag{35}$$

where

$$\mu(\mathbf{x}) = \mathbf{b}^T\mathbf{x}, \qquad \phi_i(\mathbf{x},h_i) = -h_i(c_i + W_i\mathbf{x}). \tag{36}$$

Then, the likelihood function can be given by

$$\begin{aligned} P(\mathbf{x}) &= \frac{1}{Z}e^{-E(\mathbf{x})} = \frac{1}{Z}\sum_{\tilde{\mathbf{h}}} e^{-E(\mathbf{x},\tilde{\mathbf{h}})} \\ &= \frac{1}{Z}\sum_{h_1}\sum_{h_2}\cdots\sum_{h_m} e^{\mu(\mathbf{x})-\sum_{i=1}^{m}\phi_i(\mathbf{x},h_i)} \\ &= \frac{1}{Z}\sum_{h_1}\sum_{h_2}\cdots\sum_{h_m} e^{\mu(\mathbf{x})}\prod_{i=1}^{m} e^{-\phi_i(\mathbf{x},h_i)} \\ &= \frac{e^{\mu(\mathbf{x})}}{Z}\sum_{h_1} e^{-\phi_1(\mathbf{x},h_1)}\sum_{h_2} e^{-\phi_2(\mathbf{x},h_2)} \\ &\quad \cdots\sum_{h_m} e^{-\phi_m(\mathbf{x},h_m)} \\ &= \frac{e^{\mu(\mathbf{x})}}{Z}\prod_{i=1}^{m} e^{-\phi_i(\mathbf{x},h_i)}. \end{aligned} \tag{37}$$

For the RBM model with binary value units, its free energy can be written as

$$\begin{aligned} \mathcal{F}(\mathbf{x}) &= -\log P(\mathbf{x}) - \log Z \\ &= -\mu(\mathbf{x}) - \sum_{i=1}^{m}\log\sum_{h_i} e^{-\phi_i(\mathbf{x},h_i)} \\ &= -\mathbf{b}^T\mathbf{x} - \sum_{i=1}^{m}\log(1+e^{(c_i+W_i\mathbf{x})}). \end{aligned} \tag{38}$$

APPENDIX C

As the free energy of RBM model is explicitly deduced as

$$\mathcal{F}(\mathbf{x}) = -\mathbf{b}^T\mathbf{x} - \sum_{i=1}^{m}\log(1+e^{(c_i+W_i\mathbf{x})})$$

corresponding to (19) and (30), it is easy to get

$$\begin{aligned} \mathcal{F}_c(\mathbf{x}) &= -\frac{1}{2}\mu(\mathbf{x};\theta_L) - \frac{1}{2}\mu(\mathbf{x};\theta_R) \\ &\quad - \frac{1}{2}\sum_{i=1}^{m}\log(1+e^{\phi_i(\mathbf{x},h_i^L,c_i^L,W_i^L)}) \\ &\quad - \frac{1}{2}\sum_{i=1}^{m}\log(1+e^{\phi_i(\mathbf{x},h_i^R,c_i^R,W_i^R)}). \end{aligned} \tag{39}$$

In term of the following donations:

$$\begin{aligned} P_L(h_i|\mathbf{x}) &= \sigma(c_i^L + W_i^L\mathbf{x})) \\ P_R(h_i|\mathbf{x}) &= \sigma(c_i^R + W_i^R\mathbf{x})) \end{aligned} \tag{40}$$

all the gradients can be expressed as

$$\begin{aligned} \frac{\partial\mathcal{F}_c(\mathbf{x})}{\partial W_{ij}^L} &= -P_L(h_i|\mathbf{x})\cdot x_j^L \\ \frac{\partial\mathcal{F}_c(\mathbf{x})}{\partial c_i^L} &= -P_L(h_i|\mathbf{x}) \\ \frac{\partial\mathcal{F}_c(\mathbf{x})}{\partial b_j^L} &= -x_j^L \\ \frac{\partial\mathcal{F}_c(\mathbf{x})}{\partial W_{ij}^R} &= -P_R(h_i|\mathbf{x})\cdot x_j^R \\ \frac{\partial\mathcal{F}_c(\mathbf{x})}{\partial c_i^R} &= -P_R(h_i|\mathbf{x}) \\ \frac{\partial\mathcal{F}_c(\mathbf{x})}{\partial b_j^R} &= -x_j^R. \end{aligned} \tag{41}$$

REFERENCES

[1] P. Smolensky, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA, USA: MIT Press, 1986, pp. 194–281.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Adv. Neural Inform. Process. Syst.*, vol. 19, pp. 153–160, 2007.

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines," *Neural Comput.*, vol. 24, no. 8, pp. 1967–2006, 2012.

[5] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "Learning with hierarchical-deep models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1958–1971, Aug. 2013.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[7] J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2012, pp. 3642–3649.

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learning Res.*, vol. 11, pp. 3371–3408, 2010.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[10] R. Salakhutdinov and G. Hinton, "Replicated softmax: An undirected topic model," *Adv. Neural Inform. Process. Syst.*, vol. 22, pp. 1607–1614, 2010.

[11] Y. Bengio, "Learning deep architectures for AI," *Foundations Trends Mach. Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[12] G. Hinton, "A practical guide to training restricted Boltzmann machines," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. UTML TR 2010-003, 2010.

[13] N. Wang, J. Melchior, and L. Wiskott, "Gaussian-binary restricted Boltzmann machines on modeling natural image statistics," arXiv preprint arXiv:1401.5900, 2014.

[14] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series," *J. Mach. Learning Res.*, vol. 12, pp. 1025–1068, 2011.

[15] K. H. Cho, T. Raiko, and A. Ilin, "Gaussian–Bernoulli deep Boltzmann machine," in *Proc. Int. Conf. Neural Netw.*, Aug. 2013, pp. 1–7.

[16] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learning*, 2007, pp. 791–798.

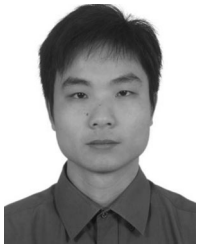
[17] G. W. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2007, pp. 1345–1352.

[18] I. Sutskever and G. E. Hinton, "Learning multilevel distributed representations for high-dimensional sequences," in *Proc. Int. Conf. Artificial Intell. Statist.*, 2007, pp. 548–555.
[19] H.-F. Wang and R.-C. Tsaur, "Insight of a fuzzy regression model," *Fuzzy Sets Syst.*, vol. 112, no. 3, pp. 355–369, 2000.
[20] P.-Y. Hao and J.-H. Chiang, "Fuzzy regression analysis by support vector learning approach," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 2, pp. 428–441, Apr. 2008.
[21] A. Fischer and C. Igel, "Training restricted Boltzmann machines: An introduction," *Pattern Recog.*, vol. 47, no. 1, pp. 25–39, 2014.
[22] C.-Y. Zhang and C. Chen, "An automatic setting for training restricted Boltzmann machine," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2014, pp. 4037–4041.
[23] M. Aoyagi, "Learning coefficient in bayesian estimation of restricted Boltzmann machine," *J. Algebraic Statist.*, vol. 4, no. 1, pp. 31–58, 2013.
[24] M. Aoyagi and K. Nagata, "Learning coefficient of generalization error in Bayesian estimation and Vandermonde matrix-type singularity," *Neural Comput.*, vol. 24, no. 6, pp. 1569–1610, 2012.
[25] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.
[26] S. Dutta and M. Chakraborty, "Fuzzy relation and fuzzy function over fuzzy sets: A retrospective," *Soft Comput.*, vol. 19, no. 1, pp. 99–112, 2014.
[27] J. J. Buckley, *Fuzzy Probabilities: New Approach and Applications*. New York, NY, USA: Springer, 2009.
[28] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of Granular Computing*. New York, NY, USA: Wiley, 2008.
[29] W. A. Lodwick and J. Kacprzyk, *Fuzzy Optimization: Recent Advances and Applications*. New York, NY, USA: Springer, 2010.
[30] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inform. Sci.*, vol. 132, no. 14, pp. 195–220, 2001.
[31] O. Castillo and P. Melin, *Type-2 Fuzzy Logic: Theory and Applications*. New York, NY, USA: Springer, 2008.
[32] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.
[33] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proc. 10th Int. workshop artificial intelligence and statistics*, NP: Society for Artificial Intelligence and Statistics, 2005, pp. 33–40.
[34] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learning*, vol. 37, no. 2, pp. 183–233, 1999.
[35] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[36] G. E. Hinton, "Learning multiple layers of representation," *Trends Cognitive Sci.*, vol. 11, no. 10, pp. 428–434, 2007.
[37] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inform. Sci.*, vol. 275, pp. 314–347, 2014.

C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

After having worked in the U.S. for 23 years as a tenured Professor, as the Department Head and an Associate Dean in two different universities, he is currently the Dean of the Faculty of Science and Technology, University of Macau, Macau, China, and the Chair Professor of the Department of Computer and Information Science. From 2012-2013, he was the IEEE SMC Society President, currently, he is the Editor-in-Chief of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and an Associate Editor of several IEEE Transactions. He is also the Chair of TC 9.1 Economic and Business Systems of IFAC. His research areas are systems, cybernetics, and computational intelligence. In addition, he is a Program Evaluator for Accreditation Board of Engineering and Technology Education in computer engineering, electrical engineering, and software engineering programs.

Dr. Chen is a Fellow of the AAAS.

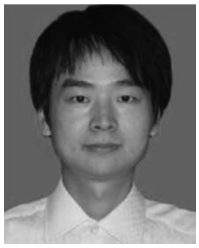


Chun-Yang Zhang received the B.S. degree in mathematics from Beijing Normal University, Zhuhai, China, in 2010, and the M.S. and Ph.D. degrees from the Faculty of Science and Technology, University of Macau, Macau, China, in 2012 and 2015, respectively.

He is currently with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests include computational intelligence, machine learning and Big Data analysis.

Long Chen (M'11) received the B.S. degree in information sciences from Peking University, Beijing, China, in 2000, the M.S.E. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2003, the M.S. degree in computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2005, and the Ph.D. degree in electrical engineering from the University of Texas at San Antonio, San Antonio, TX, USA, in 2010.

From 2010 to 2011, he was a Postdoctoral Fellow with the University of Texas at San Antonio. He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His current research interests include computational intelligence, Bayesian methods, and other machine learning techniques and their applications. He has been working in publication matters for many IEEE conferences and was the Publications Cochair of the IEEE International Conference on Systems, Man and Cybernetics in 2009, 2012, and 2014.



Min Gan received the B.S. degree in computer science and engineering from the Hubei University of Technology, Wuhan, China, in 2004, and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2010.

He is currently an Associate Professor with the School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China. His current research interests include neural networks, system identification, and nonlinear time series analysis.