Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

First attempt of Boltzmann Machines for Speaker Verification

Mohammed Senoussaoui[1,2], Najim Dehak[3], Patrick Kenny[1] , Réda Dehak[4]

and Pierre Dumouchel[1,2]

[1]École de technologie supérieure (ÉTS), Montréal
[2]Centre de recherche informatique de Montréal (CRIM), Montréal
[3]MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA
[4]Laboratoire de recherche et de développement de l'EPITA (LRDE), Paris
{mohammed.senoussaoui, patrick.kenny, pierre.dumouchel}@crim.ca, najim@csail.mit.edu, reda@lrde.epita.fr

Abstract

Frequently organized by NIST[1], Speaker Recognition evaluations (SRE) show high accuracy rates. This demonstrates that this field of research is mature. The latest progresses came from the proposition of low dimensional i-vectors representation and new classifiers such as Probabilistic Linear Discriminant Analysis (PLDA) or Cosine Distance classifier. In this paper, we study some variants of Boltzmann Machines (BM). BM is used in image processing but still unexplored in Speaker Verification (SR). Given two utterances, the SR task consists to decide whether they come from the same speaker or not. Based on this definition, we can illustrate SR as two-classes (same vs. different speakers classes) classification problem. Our first attempt of using BM is to model each class with one generative Restricted Boltzmann Machine (RBM) with symmetric Log-Likelihood Ratio on both models as decision score. This new approach achieved an Equal Error Rate (EER) of 7% and a minimum Detection Cost Function (DCF) of 0.035 on the female content of the NIST SRE 2008. The objective of this research is mainly to explore a new paradigm i.e. BM without necessarily obtaining better performance than the state-of-the-art system.

1. Introduction

Over the last five years, we have seen a huge improvement in term of performances. The greatest improvements result from the proposition of the Joint Factor Analysis (JFA) [1] and recently, the introduction of i-vector representation [2]. The i-vector has the advantage of modeling the speaker useful information in a low-dimensional space. These low dimensional i-vectors are generally given as inputs to another classifier such as Probabilistic Linear Discriminant Analysis [3] or simple Cosine Distance classifier [2]. By applying these methods, we are achieving about 2% EER. Since then, it seems that performances have reached a plateau. This finding motivated us to explore new approaches inspired by other application areas.

PLDA and Cosine Distance techniques are based on strength assumption that data follow a Gaussian distribution. This assumption is not always correct [3]. Introducing latent variables in Boltzmann Machines and learning a deep architecture enable to model distribution with a high level of complexity [4]. BM is definitely not a new research [5]. However, it has recently seen progress in term of robustness in learning algorithm. BM has been mainly used in image processing problems. Recently, many successful attempts of using these approaches for speech recognition have been reported [6][7][8].

The rest of this paper is organized as follows. In Section 2, we present the general case of Boltzmann Machines. Sections 3 and 4 are dedicated to a particular case of BM called Restricted Boltzmann Machine (RBM). In Section 5, we show how to apply RBM's in speaker recognition. In Section 6 we present some preliminary results on NIST SRE2008 SRE telephone speech and then conclude.

2. Boltzmann Machines

A Boltzmann machine [5] is a stochastic neural network with symmetric connections between units and no connection in the same unit. In this model, the probability distribution of a binary observable inputs **v** is expressed as follows:

$$\mathrm{P}(\mathbf{v};\theta) = \frac{1}{\mathrm{Z}}\mathrm{e}^{-\mathrm{E}(\mathbf{v};\theta)} \quad (1)$$

where:

- $\theta = \{\mathbf{W}, \mathbf{b}\}$ are model parameters, **W** is the synaptic symmetric weight matrix with diagonal elements set to zero and **b** is a biases vector.
- $E(\mathbf{v};\theta)$ is the energy function of the state vector x. It is given by:

$$\mathrm{E}(\mathbf{v};\theta) = -\sum_{\mathrm{i}} \mathbf{v}_{\mathrm{i}}\mathbf{b}_{\mathrm{i}} - \sum_{\mathrm{i}<\mathrm{j}} \mathbf{v}_{\mathrm{i}}\mathbf{W}_{\mathrm{ij}}\mathbf{v}_{\mathrm{j}} \quad (2)$$

- Z is a normalizing constant called partition function computes as follows

$$Z = \sum_{\mathbf{v}} e^{-E(\mathbf{v};\theta)} \quad (3)$$

[1] http://www.itl.nist.gov/iad/mig/tests/spk/

Usually the estimation of the partition function Z is intractable and it becomes exponentially hard when the complexity of the model increases. However, the good news is that for the verification task we don't need to evaluate since it is constant for all the trials.

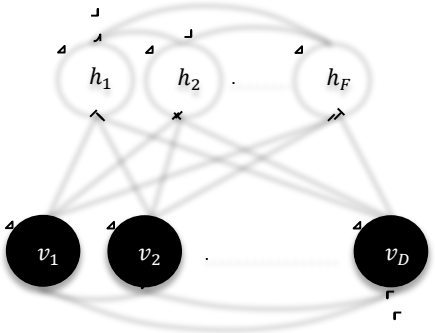

Figure 1: Boltzmann machine with D visible units $\mathbf{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_D\}$ (black states) and F hidden units $\mathbf{h} = \{\boldsymbol{h}_1, \boldsymbol{h}_2 \dots \boldsymbol{h}_F\}$ (white states).

The original version of Boltzmann Machine used only visible units. The introduction of hidden variables in the model (Figure 1) increases certainly its ability of modeling more complex pattern for a given data, even though this data is not fully observed.

The training of such a model consists of estimating the weight matrix and the biases vector. Unfortunately, it is very expensive in term of time and resources since it is based on minimization of gradient. Recently, many robust algorithms that approximate the gradient are developed such as Contrastive Divergence (CD) [9], Persistent Contrastive Divergence (PCD) [10] and variational approximations [11].

3. Restricted Boltzmann Machines

Restricted Boltzmann machines are a particular case of the Boltzmann machine. The architecture of RBM has two layers without any interaction between units in the same layer, i.e. between visible-visible or hidden-hidden (Figure 2).

This restriction is useful for several reasons. First of all, it makes training easier and faster because of exact analytic solution for mathematic derivations of training formulas. Second, with this architecture, RBM becomes the basic brick in order to build more complex models such as Deep Belief Networks (DBN) or Deep Boltzmann Machines (DBM).

Given an RBM with D binary input states $\{\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_\mathrm{D}\}$, and F hidden variables $\{\boldsymbol{h}_1, \boldsymbol{h}_2 \dots \boldsymbol{h}_\mathrm{F}\}$, the energy function of the RBM is given by the following formula:

$$\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta) = -\sum_\mathrm{i} \mathbf{v}_\mathrm{i}\mathbf{b}_\mathrm{i} - \sum_\mathrm{ij} \mathbf{v}_\mathrm{i}\mathbf{W}_\mathrm{ij}\mathbf{h}_\mathrm{j} - \sum_\mathrm{j} \mathbf{h}_\mathrm{j}\mathbf{a}_\mathrm{j} \quad (4)$$

where $\mathbf{a}$ is the biases vector of the hidden variables.

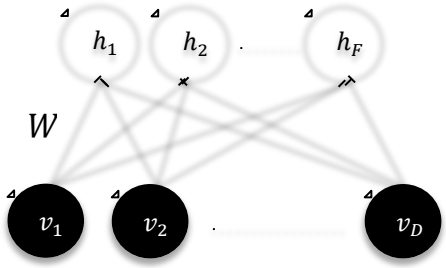

Figure 2: Restricted Boltzmann machine with D visible units $\mathbf{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_D\}$ (black states) and F hidden units $\mathbf{h} = \{\boldsymbol{h}_1, \boldsymbol{h}_2 \dots \boldsymbol{h}_F\}$ (white states).

By analogy with the general form of the probability distribution of a BM given in equation (1), we can easily derive the joint distribution of visible and hidden variable as follows:

$$\mathrm{P}(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z}\exp\big(-E(\mathbf{v}, \mathbf{h}; \theta)\big) \quad (5)$$

with the partition function $\mathrm{Z} = \sum_v \sum_h exp\big(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)\big)$

For a given visible vector $\boldsymbol{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_D\}$, the probability $P(\boldsymbol{v}; \theta)$ assigned to $\boldsymbol{v}$ can be evaluated by marginalizing out the hidden variable $\boldsymbol{h}$:

$$\mathrm{P}(\mathbf{v}; \theta) = \frac{1}{\mathrm{Z}}\sum_\mathbf{h} \mathrm{e}^{(-\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta))} \quad (6)$$

After some mathematical simplification, we obtain:

$$\mathrm{P}(\mathbf{v}; \theta) = \frac{1}{\mathrm{Z}}\exp(\mathbf{b}'\mathbf{v}) \prod_{j=1}^{F} \left(1 + \exp\big(\mathbf{a}_j + \sum_{i=1}^{D} \mathbf{W}_{ij}\, \mathbf{v}_i\big)\right) \quad (7)$$

From (5), we can also derive the factorized conditional distribution for both the visible and the hidden units:

$$\mathrm{P}(\mathbf{v}|\, \mathbf{h}; \theta) = \prod_{i=1}^{D} P(\mathbf{v}_i|\mathbf{h}; \theta) \quad (8)$$

$$P(\mathbf{h}|\, \mathbf{v}; \theta) = \prod_{j=1}^{F} P\big(\mathbf{h}_j|\mathbf{v}; \theta\big) \quad (9)$$

Both of the probabilities, $P(\mathbf{v}_i = 1|\mathbf{h}; \theta)$ (that a visible unit $\mathbf{v}_i$ is active observing all the hidden units $\mathbf{h}$), and $P\big(\mathbf{h}_j = 1|\mathbf{v}; \theta\big)$ (that a hidden $\mathbf{h}_j$ is active observing all the visible units $\mathbf{v}$), are given by *sigmoid* functions as follow:

$$P(\mathbf{v}_i = 1|\mathbf{h}; \theta) = sigm\big(\textstyle\sum_j \mathbf{W}_{ij}\boldsymbol{h}_j + \mathbf{b}_i\big) \quad (10)$$

$$P\big(\mathbf{h}_j = 1|\mathbf{v}; \theta\big) = sigm\big(\textstyle\sum_j \mathbf{W}_{ij}\mathbf{v}_i + \mathbf{a}_j\big) \quad (11)$$

3.1. Model training

3.1.1. Derivation of a learning rule

We can set a learning rule by taking derivatives of *log-likelihood* of data based on the model parameters as follows:

$$\frac{\partial \log P(\mathbf{v};\theta)}{\partial \mathbf{W}} = \langle \mathbf{v}\mathbf{h}' \rangle_{data} - \langle \mathbf{v}\mathbf{h}' \rangle_{model} \quad (12)$$

$$\frac{\partial \log P(\mathbf{v};\theta)}{\partial \mathbf{b}} = \langle \mathbf{v} \rangle_{data} - \langle \mathbf{v} \rangle_{model} \quad (13)$$

$$\frac{\partial \log P(\mathbf{v};\theta)}{\partial \mathbf{a}} = \langle \mathbf{h} \rangle_{data} - \langle \mathbf{h} \rangle_{model} \quad (14)$$

where $\langle \, . \, . \rangle_{data}$ is an expectation operator calculated on the data distribution and $\langle \, . \, . \rangle_{model}$ is also an expectation calculated with respect to the model distribution.

3.1.2. Gibbs sampling

During the RBM training, it is very complex to estimate the expectation regarding the model $\langle \, . \, . \rangle_{model}$ as defined in the previous section. For this reason it is fundamental to sample from an RBM distribution in order to estimate model parameters or to measure how well the model captures the irregularities in the training data.

The sampling method used most in the RBM framework is the *Gibbs Sampling* method. Starting from the visible data v, the *Gibbs Sampler* of m-steps is given as follows:

$$\begin{aligned} \mathbf{v}_0 &\sim P(\mathbf{v}) \\ \mathbf{h}_0 &\sim P(\mathbf{h} \mid \mathbf{v}_0) \\ \mathbf{v}_1 &\sim P(\mathbf{v} \mid \mathbf{h}_0) \\ \mathbf{h}_1 &\sim P(\mathbf{h} \mid \mathbf{v}_1) \\ &\vdots \\ \mathbf{v}_m &\sim P(\mathbf{v} \mid \mathbf{h}_{m-1}) \end{aligned}$$

In practice, *Gibbs Sampling* is doing surprisingly well with only 1-step.

3.1.3. Contrastive Divergence CD

Contrastive Divergence is an approximation of the stochastic gradient that is widely used to train the RBM. Based on the accumulated *model* and *data* statistics in (12), (13) and (14), the update rule of parameters is expressed as follows:

$$\mathbf{W}^{(k)} = \mathbf{W}^{(k-1)} + \sigma_{\mathbf{w}}(\langle \mathbf{v}\mathbf{h}' \rangle_{data} - \langle \mathbf{v}\mathbf{h}' \rangle_{model}) \quad (15)$$

$$\mathbf{b}^{(k)} = \mathbf{b}^{(k-1)} + \sigma_{\mathbf{b}}(\langle \mathbf{v} \rangle_{data} - \langle \mathbf{v} \rangle_{model}) \quad (16)$$

$$\mathbf{h}^{(k)} = \mathbf{h}^{(k-1)} + \sigma_{\mathbf{h}}(\langle \mathbf{h} \rangle_{data} - \langle \mathbf{h} \rangle_{model}) \quad (17)$$

where $\sigma_{\mathbf{w}}$, $\sigma_{\mathbf{b}}, \sigma_{\mathbf{h}}$ are learning rates.

4. RBM for continuous data

Binary representation of real complex data is not obvious in the common real problems. Therefore, many extended versions of RBM working on real-valued data have been proposed [11][12]. The most interesting proposed version is the *Gaussian-Bernoulli* RBM (GB-RBM). The visible units of GB-RBM are continuous *Gaussian* data and the hidden units are either *Bernoulli* or binary data.

Given a vector $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2 \ldots \mathbf{v}_D\} \in \mathbb{R}$ of real-valued states. The *energy function* of a GB-RBM is given by:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_i \frac{(\mathbf{v}_i - \mathbf{b}_i)^2}{2\sigma_i^2} - \sum_{ij} \frac{\mathbf{v}_i}{\sigma_i} \mathbf{W}_{ij} \mathbf{h}_j - \sum_j \mathbf{h}_j \mathbf{a}_j \quad (18)$$

where σ is the standard deviation.

In analogy with the binary RBM we can also easily derive the marginal probability function as follows:

$$P(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right) \quad (19)$$

with the partition function $Z = \int \sum_h exp\left(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)\right) dv$

$$P(\mathbf{v}; \theta) = \frac{1}{Z} \exp\left(\frac{(\mathbf{b}-\mathbf{v})'(\mathbf{b}-\mathbf{v})}{2\sigma^2}\right) \prod_{j=1}^{F} \left(1 + \exp\left(\mathbf{a}_j + \sum_{i=1}^{D} \mathbf{W}_{ij} \frac{\mathbf{v}_i}{\sigma_i}\right)\right) \quad (20)$$

Also conditional probability functions are given by the following formulas:

$$P(\mathbf{v}_i = x | h; \theta) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{\left(x - \mathbf{b} - \sigma_i \sum_j \mathbf{W}_{ij} \mathbf{h}_j\right)^2}{2\sigma_i^2}\right) \quad (21)$$

$$P(\mathbf{h}_j = 1 | \mathbf{v}; \theta) = sigm\left(\sum_j \mathbf{W}_{ij} \frac{v_i}{\sigma_i} + \mathbf{a}_j\right) \quad (22)$$

where x is real number.

In practice, we perform whitening as a preprocessing of the data before modeling it with GB-RBM. This whitening ensures the same $\sigma \approx 1$ for all visible states and simplifies the implementation of the model.

5. RBM for Speaker Verification

The problem of Speaker Verification can be implemented as follows. Given two i-vectors: $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2 \ldots \mathbf{e}_D)$ and $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2 \ldots \mathbf{t}_D)$ ($\mathbf{e}$ for *enrolment* and $\mathbf{t}$ for *test*). The question that arises is whether these two recordings are belonging to the same speaker (*Target* class) or to different speakers (*Non-Target* class).

Based on this implementation of speaker verification problem, we propose to model *Target* and *Non-Target* classes by two different GB-RBM's[1]. Each RBM has as input a concatenation of two i-vectors $(\mathbf{e}, \mathbf{t})$ (Figure 3).

We refer RBM-T (with parameters θ^T) as the RBM of the *target* class. RBM-T is trained on a set of target i-

[1] For simplicity we will use shortly RBM to refer to GB-RBM for the rest of this paper.

vector couples $(\mathbf{e}, \mathbf{t})$. By analogy, we define RBM-N (with parameters θ^N) as the RBM for *non-target* class. TBM-N is trained on non-target data only.

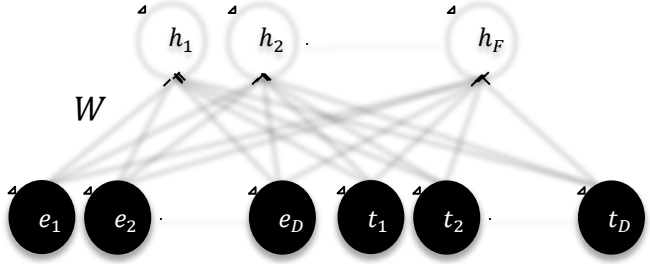

Figure 3: Restricted Boltzmann machine with $2\times D$ visible units $\boldsymbol{e} = \{\boldsymbol{e}_1, \boldsymbol{e}_2 \ldots \boldsymbol{e}_D\}$ for speaker *enrolment* utterance (left), $\boldsymbol{t} = \{\boldsymbol{t}_1, \boldsymbol{t}_2 \ldots \boldsymbol{t}_D\}$ for *test* utterance (right) and F hidden units $\boldsymbol{h} = \{\boldsymbol{h}_1, \boldsymbol{h}_2 \ldots \boldsymbol{h}_F\}$.

5.1. Model Scoring

The Log Likelihood Ratio (LLR) is traditionally used in order to compute the decision scores on speaker verification system. In our modeling, the decision score is evaluated using LLR as described in the following equation

$$S(\mathbf{e}, \mathbf{t}) = \log \frac{P(\mathbf{e}, \mathbf{t}; \theta^T)}{P(\mathbf{e}, \mathbf{t}; \theta^N)}$$
$$= \log P(\mathbf{e}, \mathbf{t}; \theta^T) - P(\mathbf{e}, \mathbf{t}; \theta^N) \quad (23)$$

It is clear that our approach will not provide symmetrical scores since $P(\mathbf{e}, \mathbf{t}; RBM) \neq P(\mathbf{t}, \mathbf{e}; RBM)$. In order to resolve this problem, we propose a symmetric version of the scoring defined as:

$$\mathrm{SymS}(\mathbf{e}, \mathbf{t}) = \frac{1}{2}(\mathrm{S}(\mathbf{e}, \mathbf{t}) + \mathrm{S}(\mathbf{t}, \mathbf{e})) \quad (24)$$

6. Experiments

We performed experiments on the short2-short3 condition of the NIST SRE 2008 FEMALE part. We use the Equal Error Rate (EER) and the old minimum Detection Cost Function (DCF) of NIST as metrics to report the results.

6.1. Feature extraction

6.1.1. Universal Background Model

We use a gender dependent Universal Background Model (UBM) containing 2048 Gaussians. This UBM is trained with the LDC releases of Switchboard II, Phases 2 and 3; Switchboard Cellular, Parts 1 and 2; and NIST 2004–2005 SRE. It was trained on 60-dimensional vector of Mel Frequency Cepstral Coefficients (MFCC) with their first and second derivatives.

6.1.2. i-vector extractor

We use a gender dependent i-vector extractor of dimension 600. Its parameters are estimated on LDC releases of Switchboard II, Phases 1, 2 and 3; Switchboard Cellular, Parts 1 and 2; Fisher data and NIST 2004, 2005 and 2006 SRE. In order to train the i-vector extractor, we performed minimum divergence training algorithm [1] at the last step of the training process in order to make all the i-vectors having zero mean and variance equals to Identity which is a crucial assumption on the training on GB-RBM, $\sigma = 1$.

6.1.3. Results

We carried out two experiments. The first one used i-vectors without any post-processing. In the second experiment, we normalized the length of the i-vector before as input to the two RBMs. This processing is based on the work presented in [13] which proves that the length normalization makes the distribution of the i-vectors more Gaussian. Results are reported on Table 1.

Table 1: The results obtained with the generative RBMs based on both raw and length normalized i-vectors. The experiments are carried out on NIST 2008 SRE (Det7).

	Non-sym (**e**,**t**)		Non-sym (**t**,**e**)		Symmetrical scoring	
	EER	DCF	EER	DCF	EER	DCF
Raw i-vector	8.6%	0.044	9.6%	0.048	7.1%	0.037
Length normalization	8.7%	0.041	9.4%	0.044	7.0%	0.035

Results shows that length-normalized i-vectors produce slightly better performance compared to the raw ones. We also note from Table 1 that the symmetrical scoring always outperforms the non-symmetrical ones. However, the performances are definitely worst compared to the ones obtained by the PLDA [3] and the cosine distance classifier [2].

7. Conclusions

In this work we presented a new paradigm for speaker verification. Despite its lower performance than the state-of-the-art systems (Probabilistic Linear Discriminant Analysis and the Cosine Distance classifier), we believe that BM could open new horizons in the speaker verification area. From complexity point of view, the proposed model is quite simple. We believe that these results can be improved with the use of Deep Boltzmann machines rather than a single layer of restricted Boltzmann machines.

8. Acknowledgements

We would like to thank Ruslan Salakhutdinov who kindly provide his Matlab codes of RBM.

9. References

[1] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification", *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 16, no. 5, pp. 980–988, July 2008.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front- end factor analysis for speaker verification", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[3] P. Kenny, "Bayesian speaker verification with heavy tailed priors", *in Proc. Odyssey 2010: The speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010.

[4] Y.Bengio, "Learning deep architectures for AI", *Foundations and Trends in Machine Learning*, vol.

[5] G. E. Hinton and T. J. Sejnowski,. "*Optimal Perceptual Inference*". *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 448-453 Washington DC, 1983.

[6] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing (Special Issue on Deep Learning for Speech and Language Processing)*, Jan. 2012.

[7] A. Mohammed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny, "Deep belief networks using discriminative features for phone recognition", *in Proc. ICASSP*, 2011.

[8] G. Dahl and G. Hinton, "Phone recognition with the mean-covariance restricted Boltzmann machine", *in Advances in Neural Information Processing 23*, 2010.

[9] G. E. Hinton. "Training products of experts by minimizing contrastive divergence". *Neural Computation*, 14(8):1711-1800 2002.

[10] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient". *Proceedings of the 25th international conference on Machine learning* (pp. 1064–1071), 2008.

[11] S. Ruslan "Learning Deep Generative Models". *PhD Thesis*, University of Toronto, 2009.

[12] M. Maxwelling. "Exponential family harmoniums with an application to information retrieval". *In Advances in Neural Information Processing Systems*, pages 1481–1488, Cambridge, MA, 2005. MIT Press.

[13] D. Garcia-Romero and C. Y. Espy-Wilso, "Analysis of i-vector length normalization in speaker recognition systems," *in Proceedings of Interspeech*, Florence, Italy, Aug. 2011.