DATA PROCESSING USING BOLTZMANN MACHINES

Mark Polyak

Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia

markpolyak@gmail.com

Abstract

This article mainly reviews the Boltzmann machine and its application to data processing tasks such as classification, recognition, filtering, etc. First part of this survey describes different types of Boltzmann machines and deep networks along with their main properties. Second part is dedicated to the brief review of different applications of Boltzmann machine found in scientific literature.

I. INTRODUCTION

The recurrent neural networks (RNNs) are a general case of artificial neural networks where the connections are not feed-forward ones only. In RNNs, connections between units form directed cycles, providing an implicit internal memory. Those RNNs are adapted to problems dealing with signals evolving through time. Their internal memory gives them the ability to naturally take time into account. The presence of feedback between units in RNNs allows them to model complex temporal data. Valuable approximation results that have been obtained for dynamical systems are gathered in [53].

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. Unsupervised neural networks can also be used to learn representations of the input that capture the salient characteristics of the input distribution, e.g., the Boltzmann machine (BM), and more recently, deep learning algorithms, which can implicitly learn the distribution function of the observed data.

Restricted Boltzmann Machine (RBM) is a type of stochastic RNNs. RBMs have been successfully applied in collaborative filtering [13], information and image retrieval [51], time series modeling [28, 52] and many other areas.

II. BOLTZMANN MACHINES

A neural network called Boltzmann Machine was invented by Geoffrey Hinton and Terrence Sejnowski [1 – 3] in 1980s. The name comes after the Boltzmann distribution in statistical mechanics, which is used in the sampling function of the network. The Boltzmann Machine is a Monte Carlo version of the Hopfield network, a form of recurrent neural network proposed by John Hopfield in [4]. Like a Hopfield net, a Boltzmann Machine is a network of binary units with an energy term defined for the network. In a Boltzmann Machine the global energy E is identical in form to the energy of a Hopfield network:

$$E = -\sum_{i,j} w_{ij} s_i s_j + \sum_i b_i s_i \ , \qquad (1)$$

where w_{ij} is the connection weight between unit i and j ; s_i is the state of unit i , $s_i \in \{-1, 1\}$; b_i is the bias of unit i .

The main difference between BM and Hopfield network is that BM uses stochastic units while Hopfield net is based on traditional McCullough-Pitts artificial neuron model. In a BM the decision for a unit to switch its state from –1 to +1 (or backwards) is probabilistic [5]:

$$s_i = \begin{cases} +1, & \text{with probability } p(+1) \\ -1, & \text{with probability } 1 - p(+1) \end{cases}, \qquad (2)$$

where $p(+1)$ is the probability of unit s_i switching its state from –1 (“off”) to +1 (“on”).

The original updating rule for Hopfield net forces each unit to switch into whichever of its states makes the total energy of the system lower. As all connections between units are symmetric (this is true for both Hopfield nets and BMs), it is possible for a stochastic unit to make the decision to be “on” of “off” locally by computing the energy difference ΔE between it being active and inactive. For the unit i this can be written as follows:

$$\Delta E_i = E_i(-1) - E_i(+1) = \sum_j w_{ij} s_j - b_i \ . \qquad (3)$$

Equations (1) and (3) for simplicity may be rewritten without the bias b_i by adding a new unit $s_0 = +1 = const$ with a weight $w_{0i} = b_i$:

$$E = -\sum_{i,j} w_{ij} s_i s_j \ , \qquad (4)$$

$$\Delta E_i = \sum_j w_{ij} s_j \ . \qquad (5)$$

A property of Boltzmann distribution called Boltzmann factor runs that the energy of a state is proportional to the negative log probability of that state. By applying this property to the first part of equation (3), rearranging the terms and taking the exponent to get rid of the logarithm, we get the following expression for the probability that i -th unit is active:

$$p_i(+1) = \frac{1}{1+e^{\Delta E_i / T}} \ , \qquad (6)$$

where T is a parameter that describes the “temperature” of the network. The same way as in statistical physics, units will usually go into the state which reduces the system energy, but occasionally they will go into the state which increases the energy, just as physical systems sometimes (not often) visit higher energy states. The higher is the temperature the higher is the possibility for a unit to go into a higher energy state instead of a state with a lower energy.

The network is run by repeatedly choosing a random unit and setting its state according to the formula (6). If the average activation $\langle s_i \rangle$ of unit i stops changing over time, the network is said to have reached a thermal equilibrium. It can be proved that any stochastic network which always goes downhill in some Lyapunov function (the energy term (1) is such a function) is guaranteed to reach a thermal equilibrium [5].

Unlike the Hopfield net, stochastic networks such as BM move from state to state without settling down into a stable configuration [6]. This means that by simply measuring the fraction of the time a BM spent in each of its states when it reached thermal equilibrium, we could use BM to generate probability distributions over various states. This probability distribution heavily depends on the connection weights of the network. So by carefully adjusting the weights we can force the network’s equilibrium distribution to be similar to the world distribution. Thus, BMs can be used to model data.

Unlike a Hopfield net, BM has two different layers: a layer of visible units and a layer of hidden units, see figure 1. Visible units reflect the state of BM, visible to the public. The network is presented data by setting output values of visible units. The same units determine network’s output after the network has reached a thermal equilibrium.

Hidden units are used to capture higher order regularities in the data distribution. Indeed, the energy function (1) includes only terms involving the activation states of pairs of units $s_i s_j$. Thus, hidden units are needed to be able to capture any structure in the world probability distribution that is higher than second order.

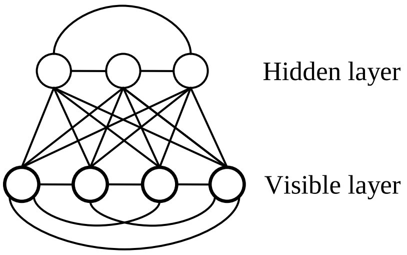

Fig. 1. Graphical model of Boltzmann Machine with 4 visible and 3 hidden units

Unfortunately hidden units make training the network more complicated. Training data (patterns we want the network to reproduce) can be used to learn the probability distribution of the states of the visible units. But there is no data to train the probability of hidden units on. The idea is that the network should discover how to use the hidden units to best represent the structure of probability distribution of patterns.

III. LEARNING PROCEDURE

The learning procedure for the BM has two phases [6]. In positive phase (Phase+), the visible units are clamped to the value of particular pattern, and the network is run until it reaches a thermal equilibrium. Then Hebbian learning is used: the weight between any two units that are both on is incremented. This phase is repeated many times, with each pattern clamped with a frequency corresponding to the world probability we would like to model.

In negative phase (Phase–) the network is run freely, without any units clamped. When it reaches a low temperature equilibrium the activities of all the units are sampled. Enough samples should be taken to obtain reliable averages of $s_i s_j$. Then the so called *unlearning* procedure is performed: the weight between any two units which are both on is decremented.

By alternating between both phases with approximately equal frequency this learning procedure will on average reduce the cross-entropy between the network's free-running distribution and the target distribution.

It is obvious that the efficiency of BM depends on equilibrium being reached fairly rapidly. A simulated annealing algorithm may be used to gradually reduce the temperature as the network runs to achieve low temperature equilibrium fairly fast.

IV. RESTRICTED BOLTZMANN MACHINES AND PRODUCTS OF EXPERTS

The main problem with Boltzmann machines is that they are difficult to learn. Even with simulated annealing it takes quite some time to reach a thermal equilibrium. And this equilibrium must be reached many times for each training example during each training epoch. This problem can be solved by adding some restrictions on the general BM structure described earlier to form a new network called Restricted Boltzmann Machine (RBM).

RBM is a bipartite undirected graphical model with a two-layer architecture [7, 8]. Unlike a general Boltzmann Machine, RBM doesn't allow any visible-visible or hidden-hidden connections between units from the same layer. Only visible-hidden connections are present, see figure 2.

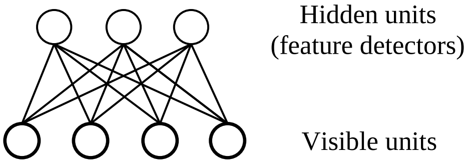

Fig. 2. Graphical model of Restricted Boltzmann Machine with 4 visible and 3 hidden units

RBM is sometimes called a Product of Experts model (PoE) [9] as it combines a number of individual component models (the experts) by taking their product. Binary hidden units are such experts, and may be also thought of as feature detectors. Such an expert (feature detector) fires a +1 to its output with probability

$$\mathrm{P}(h_j = +1 \mid \mathbf{v}) = \frac{1}{1+\exp\left(\sum_i w_{ij} v_i + a_j\right)}, \quad (7)$$

where $h_j = s_j$ is the state of hidden unit j, $h_j \in \{0,1\}$; $\mathbf{v}$ is the binary vector representing the states of all visible units; $v_i = s_i$ is the state of visible unit i, $v_i \in \{0,1\}$; a_j is the bias of hidden unit j.

Similarly, the probability that a visible unit's state is +1 will be

$$\mathrm{P}(v_i = +1 \mid \mathbf{h}) = \frac{1}{1+\exp\left(\sum_j w_{ij} h_j + b_i\right)}, \quad (8)$$

where $\mathbf{h}$ is the binary vector representing the states of all hidden units and b_i is the bias of visible unit i.

Finally conditional distributions over visible units $\mathbf{v}$ and hidden vector $\mathbf{h}$ are as follows:

$$p(\mathbf{v} \mid \mathbf{h}; W) = \prod_i \mathrm{P}(v_i \mid \mathbf{h}), \quad (9)$$

$$p(\mathbf{h} \mid \mathbf{v}; W) = \prod_j \mathrm{P}(h_j \mid \mathbf{v}), \quad (10)$$

where W is the matrix of connection weights w_{ij} between all units. Equation (10) clearly shows why RBM is called a product of experts: an RBM is a PoE with one expert per hidden unit.

PoEs may be trained by maximizing the log likelihood of the data being generated (reconstructed) by the model. Unfortunately, it is a difficult task [9] since some of the calculations involved are intractable.

The process of reconstruction of the data by the model uses a Markov Chain Monte Carlo (MCMC) algorithm called Gibbs sampling. A visualization of alternating Gibbs sampling is presented on figure 3. At time 0, the visible variables denoted by $\mathbf{v}$ represent a data vector or, simply put, the input of the neural network. Then the hidden variables (experts) $\mathbf{h}$ are updated in parallel with samples from their posterior distribution given the visible variables. At time 1, the visible variables are all updated to produce a reconstruction of the original data vector from the hidden variables, and then the hidden variables are updated in parallel again. By repeating this process long enough, it is possible to get arbitrarily close to the equilibrium distribution. The correlations $\langle v_i h_j \rangle$ shown on the connections between visible and hidden variables are the statistics used for learning in RBMs.

Hinton [9] has shown that RBMs can be efficiently trained by minimizing Contrastive Divergence. Contrastive Divergence (CD) estimates energy function's gradient, given a set of model parameters (synaptic weights w_{ij}), and training data [10]. The time shown on figure 3 can be thought of as a number of steps of CD algorithm. CD_0 is the source data at time 0, CD_1 is the first reconstruction and CD_n is the reconstruction at step n. The simple expression for RBM weight updates based on CD leaning is

$$\Delta w_{ij} \approx \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_1 . \quad (11)$$

V. DEEP NETWORKS

The RBM can serve as a building block for more complex and powerful models. Such models include Deep Belief Networks and Deep Boltzmann Machines.

Deep Belief Network (DBN) is a probabilistic generative model composed of many layers of hidden variables (figure 4, left). Each layer captures high-order correlations between the activities of hidden features in the layer below. The top two layers of the DBN form an undirected bipartite graph with the lower layers forming a directed sigmoid belief network [7], as shown in figure 4.

The learning algorithm for DBN [7, 11] uses a stack of RBMs (see figure 4, right) and proceeds as follows. First the bottom RBM is trained with parameters W^1. Then the weights of a 2nd layer RBM are initialized to $W^2 = \left(W^1\right)^T$, which ensures that the two-hidden layer DBN is at least as good as the original one-layer RBM. After that the DBN's fit to the training data may be improved by modifying W^2. Samples $\mathbf{h^1}$ from the layer of hidden features of the first RBM are used as the training data for the 2nd layer RBM. The same steps are repeated recursively for all RBMs in the stack, with samples of hidden units from one layer used as source data for training the next layer.

Unlike a DBN, Deep Boltzmann Machine (DBM) is a graphical model [7, 12] where all connections between layers are undirected (see figure 4, center). DBMs are interesting for several reasons. First, like deep belief networks, DBMs have the potential of learning internal representations that become increasingly complex, which is considered to be a promising way of solving object and speech recognition problems. Second, high-level representations can be built from a large supply of unlabeled sensory inputs and very limited labeled data can then be used to only slightly fine-tune the model for a specific task at hand. Finally, unlike DBNs, the approximate inference procedure, in addition to an initial bottomup pass, can incorporate top-down feedback, allowing DBMs to deal more robustly with ambiguous inputs.

Training algorithm for DBM is similar to the one for DBN and is described in [12].

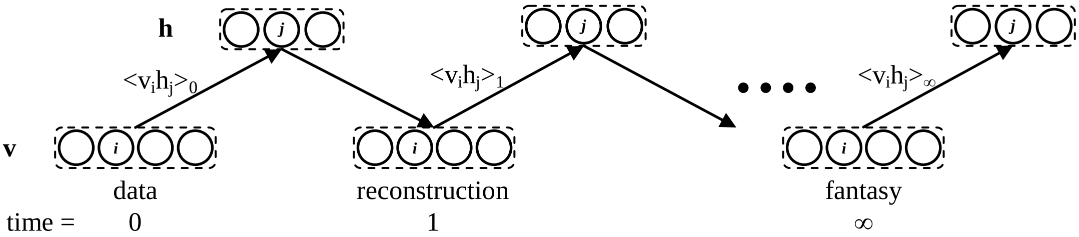

Fig. 3. Visualization of Gibbs sampling in a RBM

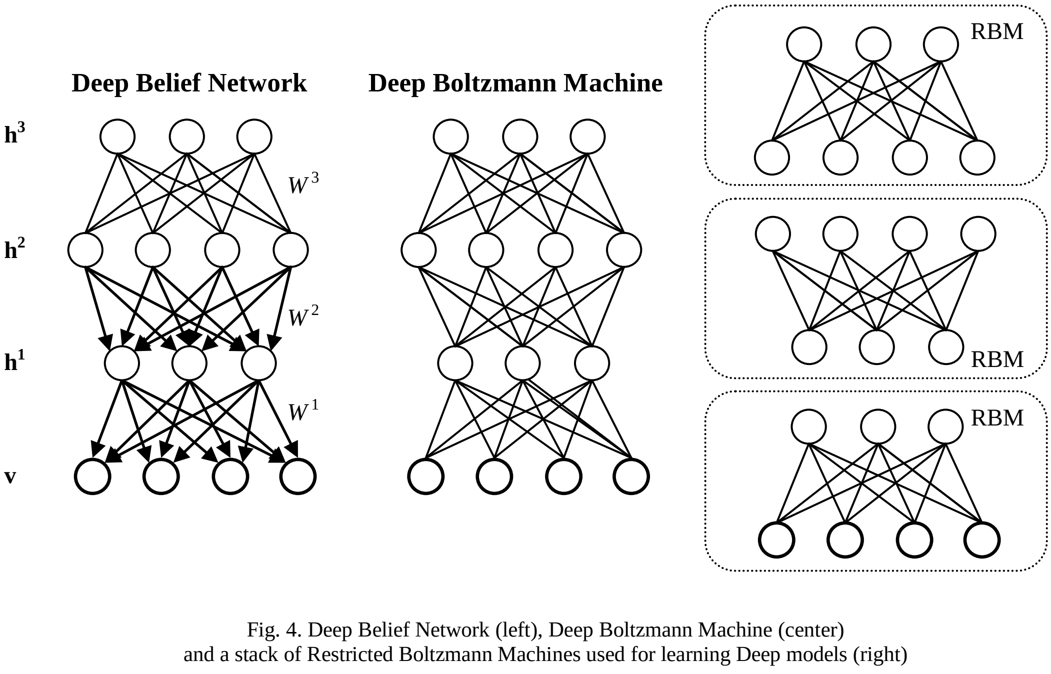

Fig. 4. Deep Belief Network (left), Deep Boltzmann Machine (center) and a stack of Restricted Boltzmann Machines used for learning Deep models (right)

VI. RBMs for collaborative filtering

Collaborative filtering is a technique used by many recommendation systems for information filtering and prediction. In 2006 a US online DVD-rental company Netflix started a competition for the best collaborative filtering algorithm to predict user ratings for films based on previous ratings made by different users. This competition, called Netflix Prize, gave rise to many new recommendation algorithms among which was RBM-based collaborative filtering algorithm [13].

The use of RBM as a collaborative filtering algorithm became very popular [14 – 17]. The reason behind this popularity is the following. RBM is a really good feature extractor, it essentially performs a binary version of factor analysis. Some comparison results suggest that RBM can outperform traditional factor analysis methods such as Singular Value Decomposition (SVD) [13]. Nevertheless experiments have shown [16, 17] that for the best results RBMs must be used in conjunction with other prediction algorithms in a blended solution [18, 19]. All teams with high scores in Netflix competition used RBMs for collaborative filtering in their blends.

A good overview of collaborative filtering algorithms based on RBMs is presented in [14]. Application of RBMs to Netflix prize is described in detail in [13, 18].

VII. MODELING DATA USING RBMs

RBMs and DBNs are very good universal approximators. It can be shown [20], that any distribution p on the set $\{0,1\}^n$ of binary vectors of length n can be arbitrarily well approximated by an RBM with $k-1$ hidden units, where k is the minimal number of pairs of binary vectors differing in only one entry such that their union contains the support set of p. If contrastive divergence (11) is run for a long time, the resulting visible vector $\mathbf{v}$ represents some fantasy data, sampled from the distribution p learned by RBM (see figure 3). By running the Gibbs sampling in RBM for some time it is possible to generate a random approximation to the original data used to initialize the Markov chain. So RBM may be used for modeling complex multidimensional data, especially when it is difficult to learn the analytical form of data distribution function.

A good example of modeling images of handwritten digits with RBM is described by Hinton et al in [21 – 24]. The images are taken from MNIST database [25]. The network learns a generative model (DBN) with several layers and has the structure shown on figure 5.

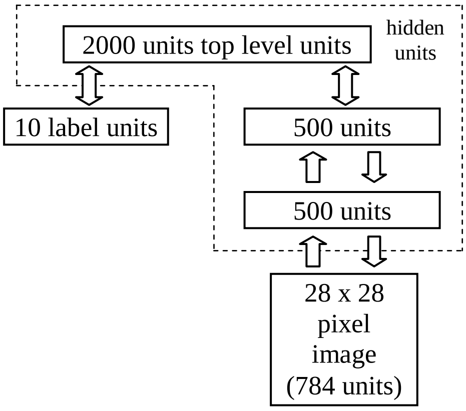

Fig. 5. Deep Belief Network used for generating images of handwritten digits and for classification

A modified version of RBM called Continuous Restricted Boltzmann Machine (CRBM) can perform reconstruction of continuous data [26, 27]. It has good results on nonlinear data, both artificial and real. In [26] an application of CRBM to ECG modeling and analysis is shown. The hidden units of CRBM can underpin a simple novelty detector such as a single layer perceptron. Another application of CRBM to real data from a pH sensor is demonstrated in [27]. CRBM is shown to perform better than multilayer perceptron and self organizing maps.

A generative model for human motion is introduced in [28]. It is based on the idea that local constraints and global dynamics can be learned efficiently by a conditional Restricted Boltzmann Machine. Once trained, such models are able to efficiently capture complex non-linearities in the data without sophisticated pre-processing or dimensionality reduction. The model has been designed with human motion in mind, but should lend itself well to other high-dimensional time series.

Another expansion of RBM called Recurrent Temporal Restricted Boltzmann Machine (RTRBM) is shown to generate videos at pixel level [29]. It is also shown to be good at generating motion capture.

Yet another RBM modification called Implicit Mixture of Conditional Restricted Boltzmann Machines (imCRBM) is used in human pose tracking [30]. It allows one to learn models from many different types of motion and subjects using the same set of latent variables. The imCRBM is suggested to be useful for time series analysis beyond the tracking domain.

VIII. CLASSIFICATION AND RECOGNITION WITH RBMs

Application of RBMs to classification tasks is as wide as its application to data modeling. In fact if it is possible to build a model consisting of one or more levels of hidden variables and that model is good at reconstructing data from visible units, than it will be possible to use those features from latent variables for classification.

In [27] a simple single layer perceptron is appended to the layer of hidden units in CRBM for classification. A DBN model for classification and modeling of images of hand-written written digits proposed by Hinton et al [21] is shown on figure 5.

DBN can also be applied to speech recognition problems [31 – 33]. Image recognition with DBNs is not limited by MNIST database, but includes a NORB dataset for 3D object recognition [12, 34], a FERET database for face recognition [44] and a CIFAR dataset of tiny colour images for classification [35, 36]. Other uses of RBMs and DBNs as generative models and classifiers include document retrieval and text processing [22, 37].

IX. CONCLUSION

Many variants of RBMs, DBNs and DBMs and their generalizations to exponential family models [50] have been successfully applied not only for classification tasks [11, 38, 39], but also regression tasks [40], visual object recognition [41 – 44], dimensionality reduction [22, 23], information retrieval [37, 45, 46], modeling image patches [47], extracting optical flow [48], and robotics [49]. Research on neural network models with deep architectures such as Boltzmann machine is making progress all the time.

REFERENCES

[1] Hinton, G. E. and Sejnowski, T. J. Optimal perceptual inference. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, Washington DC, 1983

[2] Hinton, G. E., Sejnowski, T. J., and Ackley, D. H. Boltzmann Machines: Constraint satisfaction networks that learn. Technical Report CMU-CS-84-119, Carnegie-Mellon University. 1984

[3] Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. A learning algorithm for Boltzmann machines. Cognitive Science, vol. 9, pp. 147-169, 1985

[4] Hopfield J. J. Neural networks and physical systems with emergent collective computational properties, Proceedings of the National Academy of Sciences of the USA, vol. 79, no. 8, pp. 2554–2558, April 1982.

[5] Haykin, S. Neural Networks: A Comprehensive Foundation. 2nd edition. Macmillan, New York, 1998

[6] Roweis, S. Boltzmann Machines. Lecture Notes, 1995

[7] Salakhutdinov, R. R. Learning Deep Generative Models. Ph.D. Thesis, Dept. of Computer Science, University of Toronto, Sep 2009

[8] Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, vol. 1, ch. 6, pp. 194-281. MIT Press, Cambridge, 1986

[9] Hinton, G. E. Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation, vol. 14, pp. 1771-1800, 2002

[10] Woodford, O. Notes on Contrastive Divergence, 2006. http://www.robots.ox.ac.uk/~ojw/files/NotesOnCD.pdf (retrieved on 25 March 2012)

[11] Hinton, G. E., Osindero, S. and Teh, Y. A fast learning algorithm for deep belief nets. Neural Computation vol. 18, pp. 1527-1554, 2006

[12] Salakhutdinov, R. R. and Hinton, G. E. Deep Boltzmann Machines. Artificial Intelligence and Statistics, 2009

[13] Salakhutdinov, R. R., Mnih, A., and Hinton, G. E.. Restricted Boltzmann machines for collaborative filtering. In Zoubin Ghahramani, editor, *Proceedings of the International Conference on Machine Learning*, vol. 24, pp. 791-798. ACM, 2007.

[14] Louppe, G. Collaborative filtering. Scalable approaches using restricted Boltzmann machines. M. Sc. Thesis, Department of Electrical Engineering and Computer Science, University of Liege, 2010

[15] Introduction to Restricted Boltzmann Machines, blog, 2011. http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines/ (retrieved on 25 March 2012)

[16] Töscher, A. and Jahrer, M. Collaborative Filtering Ensemble for Ranking. Track 2 KDD CUP 2011, Yahoo, 2011

[17] Töscher, A. and Jahrer, M. Collaborative Filtering Ensemble. Track 1 KDD CUP 2011, Yahoo, 2011

[18] Bell, R. M., Koren, Y. and Volinsky, C. The BellKor solution to the Netflix prize. Technical report, AT&T Labs - Research, October 2007

[19] Jahrer, M., Töscher, A. and Legenstein, R. Combining predictions for accurate recommender systems. In KDD: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010

[20] Montufar, G., Ay, N. Refinements of Universal Approximation Results for DBNs and RBMs. Neural Computation, vol. 23, no. 5, pp. 1306-1319, 2011

[21] Hinton, G. E., Osindero, S. and Teh, Y. A fast learning algorithm for deep belief nets. Neural Computation, vol. 18, pp. 1527-1554, 2006

[22] Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. Science, vol. 313, no. 5786, pp. 504-507, 28 July 2006

[23] Salakhutdinov, R. R., and Hinton, G. E. Learning a non-linear embedding by preserving class neighborhood structure. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 11, 2007

[24] Hinton, G. E. To recognize shapes, first learn to generate images. In P. Cisek, T. Drew and J. Kalaska (Eds.) Computational Neuroscience: Theoretical Insights into Brain Function. Elsevier, 2007

[25] The MNIST Database of handwritten digits by Yann LeCun http://yann.lecun.com/exdb/mnist/ (retrieved on 25 March 2012)

[26] Chen, H. and Murray, A. F. A Continuous Restricted Boltzmann Machine with an Implementable Training Algorithm, IEE Proceedings of Vision, Image and Signal Processing, vol. 150, no. 3, p. 153-158, 2003

[27] Tang, T. B., Murray, A. F. Adaptive sensor modelling and classification using a continuous restricted Boltzmann machine (CRBM). Neurocomputing, vol. 70, pp. 1198-1206, 2007

[28] Taylor, G. W., Hinton, G. E., and Roweis, S. T. Modeling human motion using binary latent variables. Advances in Neural Information Processing Systems, vol. 19, pp. 1345-1352, 2007

[29] Sutskever, I., Hinton, G. E. and Taylor, G. W. The Recurrent Temporal Restricted Boltzmann Machine. Advances in Neural Information Processing Systems, vol. 21, MIT Press, Cambridge, MA, 2009

[30] Taylor, G. W., Sigal, L., Fleet, D. and Hinton, G. E. Dynamic binary latent variable models for 3D human pose tracking. IEEE Conference on Computer Vision and Pattern Recognition, 2010

[31] Dahl, G., Yu, D., Deng, L. and Acero, A. Context-Dependent Pre-Trained Deep Neural Networks for Large Vocabulary Speech Recognition, IEEE Transactions on Audio, Speech, and Language Processing, 2011

[32] Mohamed, A. and Hinton, G. E. Phone Recognition using Restricted Boltzmann Machines. Proceedings of the 35th International Conference on Acoustics, Speech, and Signal Processing, pp. 4354-4357, Dallas, Texas, 2010

[33] Chang J. Speech Recognition Leaps Forward, Microsoft Research Blog entry on 29 August 2011. http://research.microsoft.com/en-us/news/features/speechrecognition-082911.aspx (retrieved on 25 March 2012)

[34] The NORB dataset for 3D generic recognition. http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/ (retrieved on 25 March 2012)

[35] Krizhevsky, A.. Learning multiple layers of features from Tiny Images. Master's thesis, Dept. of Computer Science, University of Toronto, 2009

[36] CIFAR dataset. http://www.cs.toronto.edu/~kriz/cifar.html (retrieved on 25 March 2012)

[37] Salakhutdinov R. R, and Hinton, G. E. Semantic Hashing. Proceedings of the SIGIR Workshop on Information Retrieval and Applications of Graphical Models, Amsterdam, 2007

[38] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. Greedy layer-wise training of deep networks. Bernhard Scholkopf, John C. Platt, and Thomas Hoffman, editors, Advances in Neural Information Processing Systems, pp. 153-160. MIT Press, 2007

[39] Larochelle, H., Bengio, Y., Louradour, J. and Lamblin, P. Exploring strategies for training deep neural networks. Journal of Machine Learning Research, vol. 10, pp. 1-40, 2009

[40] Salakhutdinov, R. R. and Hinton, G. E. Using deep belief nets to learn covariance kernels for Gaussian processes. In Advances in Neural Information Processing Systems, vol. 20, 2008

[41] Ranzato, M. A., Huang, F., Boureau, Y. and LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2007

[42] Ranzato, M. A., Boureau, Y. and LeCun, Y. Sparse feature learning for deep belief networks. Advances in Neural Information Processing Systems, 2008

[43] Bengio, Y. and LeCun, Y. Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, Large-Scale Kernel Machines. MIT Press, 2007

[44] Teh, Y.-W., Hinton, G. E. Rate-coded Restricted Boltzmann Machines for Face Recognition. Advances in Neural Information Processing Systems, vol. 13, MIT Press, Cambridge, MA, 2001

[45] Ranzato, M. A. and Szummer, M. Semi-supervised learning of compact document representations with deep networks. In Proceedings of the International Conference on Machine Learning, vol. 25, pp. 792-799, 2008

[46] Torralba, A., Fergus, R. and Weiss, Y. Small codes and large image databases for recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008

[47] Osindero, S., and Hinton, G. E. Modeling image patches with a directed hierarchy of Markov random fields. Advances in Neural Information Processing Systems, Cambridge, MA, 2008

[48] Memisevic, R., and Hinton, G. E. Unsupervised learning of image transformation. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007

[49] Hadsell, R., Erkan, A., Sermanet, P., Scoffier, M., Muller, U. and LeCun, Y. Deep belief net learning in a long-range vision system for autonomous off-road driving. In IROS, pp. 628-633. IEEE, 2008

[50] Welling, M., Rosen-Zvi, M. and Hinton, G. E. Exponential Family Harmoniums with an Application to Information Retrieval. Advances in Neural Information Processing Systems, 17, MIT Press, Cambridge, MA, 2005

[51] Gehler, P., Holub, A., and Welling, M.. The Rate Adapting Poisson (RAP) model for information retrieval and object recognition. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006

[52] Sutskever, I. and Hinton, G. E. Learning multilevel distributed representations for high-dimensional sequences. Technical Report UTML TR 2006-003, Dept. of Computer Science, University of Toronto, 2006

[53] Cardot, H. and Boné, R. (editors). Recurrent Neural Networks for Temporal Data Processing. InTech, February, 2011. ISBN 978-953-307-685-0