Dynamic Object Recognition using Sparse Coded Three-Way Conditional Restricted Boltzmann Machines

Wei Zhao Haitham Bou Ammar Nico Roos

Department of Knowledge Engineering, Maastricht University, The Netherlands

Abstract

In this paper a novel framework capable of both accurate predictions and classifications of dynamic images is introduced. The proposed technique makes of use of a novel combination of sparse coding, a feature extraction algorithm, and three-way weight tensor conditional restricted Boltzmann machines, a form of deep learning. Experiments performed on both the prediction and classification of various images show the efficiency, accuracy, and effectiveness of the proposed technique.

1 Introduction

Vision robotics is an important and challenging task. Perception of the world using eyes is one of the most important source of information for humans and many animal species. Robots equipped with cameras should also be able to access this information. Despite a large amount of research in vision and the availability of vision libraries such as OpenCV, human level object recognition using camera images cannot yet be achieved. Nevertheless, for single images good results can sometimes be achieved.

In robot vision the task of identifying objects becomes even more challenging. Firstly, the robot's perception is changing because of its actions, such as moving around in the world. Secondly, the environment in which a robots operates is often dynamic. Thirdly, robots often have only limited processing power available for a timely process of a stream of images. To address these issues, we propose to use a sequence of images instead of single images as the basis robot vision. We investigate an architecture based on a *Three-Way Factored Conditional Restricted Boltzmann Machine* [6] which we feed with a sequence of the last n camera images. The Boltzmann machine is used for three tasks:

1. To predict the next image in the sequence of image.
2. To classify the object in the image.
3. To predict the position and size of the object in the next image.

To improve the quality of the results, we investigate the method of preprocessing the images using Sparse Coding [10, 12]. Sparse Coding enables the extraction of the main features of an image.

The proposed architecture as been evaluated through two series of experiments. First, we investigate the use of Sparse Coding. Next we investigated the use of a Three-Way Factored Conditional Restricted Boltzmann Machine. The Boltzmann machine uses a history of images as input and produces as output (1) a prediction of the next image, (2) a classification of the object in the image, and (3) a predication of the position and size of the object. Based on the results of the first series of experiments, the sparse coding of the images in the history, instead of the images themselves, are as inputs of the Boltzmann machine.

Outline The next section starts with providing the necessary background for the remainder of the paper. Section 3 describes the architecture that we propose. Section 4 describes the experiments used to evaluate the performance of the proposed architecture, and discusses the experimental results. Conclusions and future work are described in Section 5.

2 Background

This section provides background knowledge needed by the reader to understand the remainder of the paper.

2.1 Sparse Coding

In most real-world applications data availability is problematic. Therefore, if decisions are to be made autonomously, these have to be based on a small amount of data. To remedy this problem, a relatively new trend in machine learning has been sought. More precisely, the focus has shifted from collecting more data to the generation and discovery of more informative features [12, 13]. Sparse coding is an unsupervised algorithm for finding a succinct representation of an unlabeled data set. Given only unlabeled input data, it learns basis functions that capture high-level features in the inputs [10, 12]. Formally, given a data set $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$, the question is to represent $\mathbf{x}^{(i)}$ as a combination of basis vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_l$, where $\mathbf{b}_j \in \mathbb{R}^d$, and sparse activations $\mathbf{a}^{(i)} \in \mathbb{R}^l$ such that $\mathbf{x}^{(i)} \approx \sum_{j=1}^{l} \mathbf{b}_j a_j^{(i)}$. The basis set can be overcomplete, where $l > d$, and therefore can capture a large number of patterns in the input data. Sparse coding aims at solving the following optimization problem:

$$\min_{\{\mathbf{b}_j\},\{\mathbf{a}_j^{(i)}\}} \sum_{i=1}^{n} \frac{1}{2\sigma^2} \left|\left| \mathbf{x}^{(i)} - \sum_{j=1}^{l} \mathbf{b}_j a_j^{(i)} \right|\right|_2^2 + \beta \sum_{i=1}^{n} \sum_{j=1}^{l} \Lambda\left(a_j^{(i)}\right)$$

$$\text{subject to} \quad ||\mathbf{b}_j||_2^2 \leq c,\ \forall j \in \{1, 2, \ldots, l\} \tag{1}$$

where, the first term; i.e., $\left|\left|\mathbf{x}^{(i)} - \sum_{j=1}^{l} \mathbf{b}_j a_j^{(i)}\right|\right|_2^2$, is the reconstruction error, and the second; i.e., $\beta \sum_{i=1}^{n} \sum_{j=1}^{l} \Lambda\left(a_j^{(i)}\right)$, is the sparsity regularization, with $\Lambda(\cdot)$ being the penalty function, and $\beta \in \mathbb{R}$ is a constant. Different, sparsity functions (i.e., Λ) can be used. The one used in this paper is the L_1 norm, which has been shown to induce sparse activations and is robust to irrelevant features [12].

Assuming the usage of the L_1 penalty as the sparsity function, the optimization problem is convex in the bases, while holding the activations fixed, and convex in the activations when holding the bases fixed. However, the problem is not convex in both the activations and bases simultaneously [10, 13]. Different methods for solving the optimization problem have been proposed. In this paper the algorithms proposed in [10] are adopted. The solution of the above problem is twofold. Firstly, the bases are fixed, and the activations are determined. Given the activation, the second step commences to solve for the bases. More information about solving the optimization problem can be found in [10].

2.2 Restricted Boltzmann Machine

Restricted Boltzmann Machines (RBM) [1] are energy-based models for unsupervised learning. These models are stochastic with stochastic nodes and layers, making them less vulnerable to local minima [14]. Further, due to their multiple layers and their neural configurations, RBMs possess excellent generalization capabilities [3].

Formally, an RBM consists of visible and hidden binary layers. The visible layer represents the data, while the hidden layers increases the learning capacity by enlarging the class of distributions that can be represented to an arbitrary complexity [14]. This paper follows standard notation where i represents the indices of the visible layer, j represents those of the hidden layer, and $w_{i,j}$ denotes the weight connection between the i^{th} visible and j^{th} hidden unit. Further, v_i and h_j denote the state of the i^{th} visible and j^{th}

hidden unit, respectively. According to the above definitions, the energy function of an RBM is given by:

$$E(v, h) = -\sum_{i,j} v_i h_j w_{ij} - \sum_i v_i a_i - \sum_j h_j b_j \tag{2}$$

where a_i and b_j represent the biases of the visible and hidden layers, respectively. Because of the specific structure of RBMs, visible and hidden units are conditionally independent given one-another. The joint probability of a state of the hidden and visible layers can be given: $P(v, h) = \exp(-E(v, h))/Z$ with $Z = \sum_{x,y} \exp(-E(x, y))$. To determine the probability of a data point represented by a state v, the marginal probability is used. This is determined by summing out the state of the hidden layer as: $p(v) = \sum_h P(v, h) = \sum_h \left(\exp\left(-\sum_{i,j} v_i h_j w_{ij} - \sum_i v_i a_i - \sum_j h_j b_j\right)\right)/Z$. Parameters are fitted by maximizing the likelihood function. Many algorithms can be used to learn the parameters of RBM, such as Contrastive Divergence, maximum pseudo-likelihood, ratio matching. Interested readers are referred to [8, 9] for a more comprehensive discussion of learning algorithms for RBM. .

3 Proposed Architecture

In this section the proposed architecture is discussed in detail. The architecture aims at

1. predicting the next image in a consistent sequence of images,
2. classifying the object in the image,
3. predicting the position and size of the object in the next image.

The idea behind the proposed architecture is to use a sequence of images instead of a single image. A sequence of images contains more information because each image is slightly different. We hope that this additional information improve the quality of the classifications. A sequence of image also contains information about the movement of the object, its change in size in the next image, and the face of object that will be visible in the next image.

To realize the above mentioned objectives a Three-Way Factored Conditional Restricted Boltzmann Machine (3BM), similar to that proposed in [6], is used. The energy function is modified by adding a new components including a three-way weight tensor to capture the correlations among the input, output, and hidden variables. But this three-way tensor increase the parameters to cubic, thus motivates the use of factors to decrease the parameters. The the three-way weight tensors are divided into three types of pairwise interactions: the connections between output units and factors, connections between hidden units and factors, and connections between input units and factors. The 3BM is a modified version of a FCRBM [11], which aims to make use of a factored three-way weight tensor to reason about the relations between the provided images. Figure 1 gives an illustration of the 3BM we are using.

Formally, define $\mathcal{V}_{<t} = [v_{<t}^{(i)}, \ldots, v_{<t}^{(n_1)}]$, with n_1 being the number of units in the history layer. Further, define $\mathcal{H}_t = [h_t^{(1)}, \ldots, h_t^{(n_2)}]$, with n_2 being the number of nodes in the hidden layer. Finally, define $\mathcal{V}_t = [v_t^{(1)}, \ldots, v_t^{(n_3)}]$, with n_3 being the number of units in the present layer. In the history and present layers, a Gaussian distribution is adopted, with a sigmoid distribution for the hidden. The energy function is given by:

$$E(\mathcal{V}_t, \mathcal{H}_t | \mathcal{V}_{<t}, \mathbf{W}) = -\sum_i \frac{\left(v_t^{(i)} - a^{(i)}\right)}{2\sigma_i^2} - \sum_j h_t^{(j)} b^{(j)} - \sum_f \left(\sum_i \mathbf{W}_{if}^{\mathcal{V}_t} \frac{v_t^{(i)}}{\sigma_i} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{<t}} \right) \tag{3}$$

where, f is the number of factors used for factoring the three-way weight tensor among the layers, and σ_i is the variance of the Gaussian distribution in the history layer. Furthermore, $\mathbf{W}_{if}^{\mathcal{V}_t}$, $\mathbf{W}_{jf}^{\mathcal{H}}$, and $\mathbf{W}_{kf}^{\mathcal{V}_{<t}}$ are the factored tensor weights of the history, hidden, and present layer, respectively. Finally, $a^{(i)}$ and $b^{(j)}$ are the biases of the history and hidden layers, respectively.

Parameters of the model can be learned by maximizing the log likelihood, which is given by 4, while the derivative of log likelihood at parameter ω is given by 5.

$$\omega^* = argmaxL(\omega) \tag{4}$$

$$\frac{\partial L}{\partial \omega} = \left\langle \frac{\partial E}{\partial \omega} \right\rangle_{data} - \left\langle \frac{\partial E}{\partial \omega} \right\rangle_{model} \tag{5}$$

The angled brackets in Equation 5 represent expectations under the distribution specified by the subscripts, and ω is a generic parameter in the model. To learn the parameters, a Contrastive Divergence discussed in [8, 9] method is used.

The quality of the results returned but the 3BM depends of course on the quality of the input data; i.e., the history of images. To improve the quality of the input data, we investigate the use of Sparse Coding (SC) as a preprocessing step. SC [2, 10] has been proven to be a powerful form of feature extraction. It is able to learn the most informative features of a set of images, and can subsequently describe new images in terms of the learned feature space. As we will demonstrate in a classification experiment, see the Subsection 4.1, SC is able to learn features that improve the quality of classification task.

Figure 2 show the whole architecture. Images are fed into the sparse coding module, which extracts the relevant features. The history of features of images are stored in the history layer of the 3BM. The position, and the size of the object is also provided as inputs. After training the 3BM using the features of the current images, including position and size, and also including a classification of the object in the images, the 3BM can be used to predict the features of the next image, the class of the object and the position and size of the object.

The proposed architecture is not able to handle more than one object at the time. Therefore, we introduced an image segmentation step [4] that selects a sub-image containing the object to be identified by the propose architecture. Of course, the history of sub-images should all contain the same object. The proposed architecture can help in selecting the correct sub-image using the predicted position and size of the object in the whole image.

4 Experiments

To test the efficiency and effectiveness of the proposed architecture two series of experiments have been performed. These involved both classification and prediction of different objects in a robotics environment. Data collection and image segmentation was performed in *V-rep*, a robotics simulator[1]. Firstly, a set consisting of sequences of images for several objects were collected. Each sequence shows an object from different angles. In this experiment, there were 8 classes of objects, including 3 kind of chairs, 3 kind of tables, and 2 kind of shelves. About 200 images are captured for each class: 50 of them are reshuffled as the training data for computing bases of sparse coding, from which informative features using sparse coding were discovered. The others are left for the classification in 4.1, and recognition task in 4.2 by keeping their original orders. The benefits of using the sparse coded images was investigated in the first series of experiments. In the second series of experiments, a history of informative features was first used to train the 3BM and was subsequently evaluate the predictions and the classification of the 3BM.

4.1 The benefits of using sparse coding

To evaluate the benefits of extracting informative feature using sparse coding, a classification experiment using a Support Vector Machine (SVM) named C-support vector classification (C-SVC) [5] with a linear kernel was carried out. To solve the quadratic minimization problems, an SMO-type decomposition method proposed in [7] was taken. The classification by the SVM also provides a benchmark to compare the classification results of the proposed architecture.

In the experiment, eight different kinds of objects were present in the robotic environment. 50 images of each category of objects were used as training data for sparse coding. About 2000 images were split into

[1] http://www.coppeliarobotics.com/

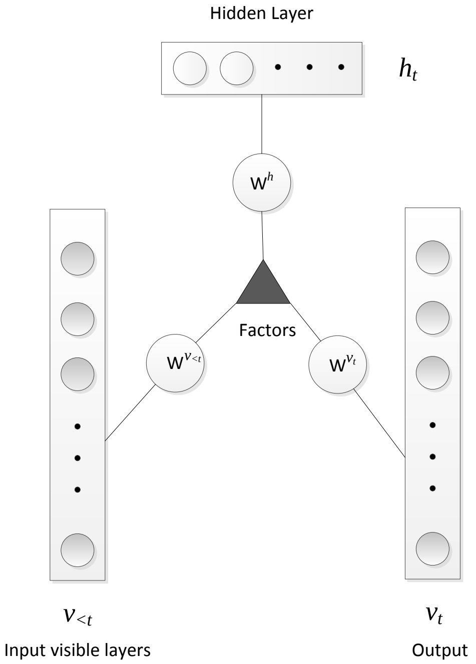

Figure 1: Three-Way Factored Conditional restricted Boltzmann machine in its full form. Three layers: (1) history layer, $v_{<t}$, (2) present layer v_t, and (3) hidden layer h are shown. Each of the layers have a certain number of nodes that are connected via a three dimensional bidirectional weight tensor.

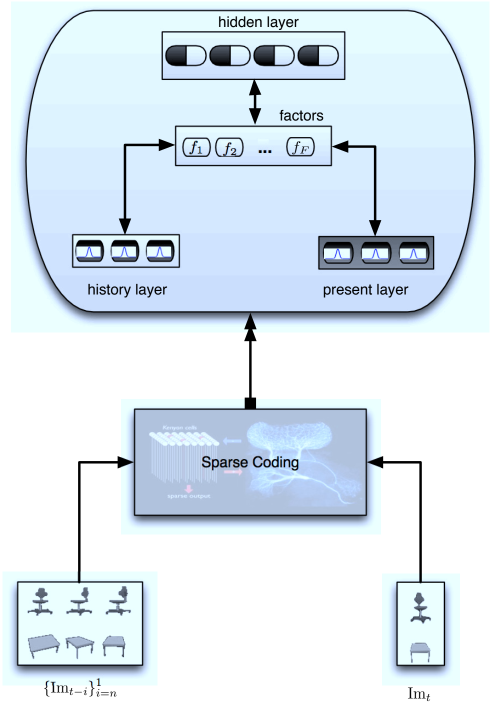

Figure 2: The overall framework of the proposed prediction technique. Given a history of images (i.e., $\{\text{Im}_{t-n}, \text{Im}_{t-n-1}, \ldots, \text{Im}_{t-1}\}$) and a present image (i.e., Im_t), sparse coding is first performed to discover high-level and informative features. These features are then passed to the proposed Boltzmann machine for prediction and classification.

a training set and a test set for classification. Different amounts of training images are selected to train a classifier, then the classification accuracy were computed by applying it on the test set. We compared the classification results when using original images and sparse coded images respectively. Figure 3 shows the classification results of the SVM with and without first extracting informative feature using sparse coding. Figure 4 presents a comparison between the training time of the SVM when using the original images and the training time when using the sparse coded representation.

From Figures 3 and 4 the following three conclusions can be arrived at:

Conclusion I The usage of SC representation of an image leads to better classification results than the usage of the original images.

Conclusion II The accuracy of classification using SC representation decline smoothly when the number of training data decreases, while it drops significantly with the usage of the original images.

Conclusion III The time need to train a classifier using SC representations of image is more or less constant in the number of images, and outperforms the usage of the original images. The time to train a classifier using the original images increases linearly in the number of images.

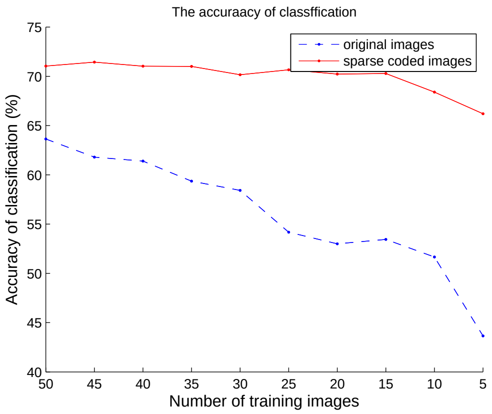

Figure 3: The accuracy of classification results by using the original images and the sparse coded representation.

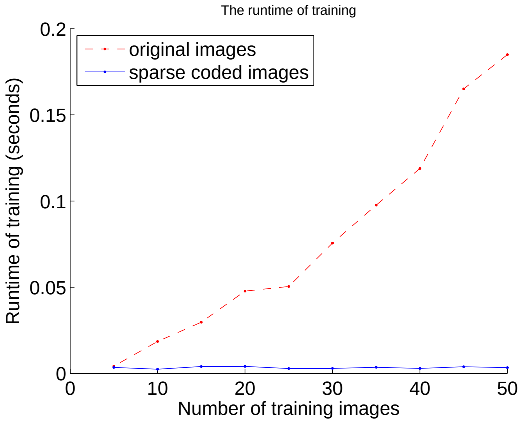

Figure 4: The time of training classifier.

4.2 Evaluation of proposed architecture

To evaluate the benefits of architecture proposed in Section 3, the SVM in the first series of experiments was replaced with the 3BM. Instead of individual images, the data set used here are several sequences of successive images. Each sequences recorded a period of an object in transformation. Figure 5 shows two instances of sequence with length of 5 frames. The features provided by the sparse coding were extended with information coming from the image segmentation about the position and size of the object. Here the position represents the center coordinate of the object in a single image, while the size is a vector denoting the width and length of the object. A difference between the SVM an the the 3BM is that the 3BM will do the classification and the prediction the same time. That means, we don't need to train a large number of labeled data to gain a classier. First we used 10 to 50 sequences of length n as the training data. For the labeled n images, directed connections between the history (the first $n-1$ frames in the sequence) and current units (the last frame in the sequence) are computed. Then it is used to predict the next frames of 10 to 50 test sequences. All the images have a original resolution of $64 * 64$.

After training the 3BM, we evaluated the proposed architecture with respect to (1) prediction the features of the next image, (2) classifying the object in the image, and (3) predicting the position and size of the object in the next image. We evaluated the output of the architecture based on different sequence lengths n.

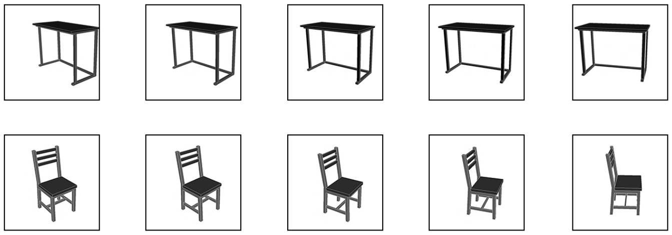

Figure 5: Two images sequences for training and test with the 3BM.

Table 1 and Table 2 shows the results of our experiments with and without preprocessing by using sparse coding respectively, where the accuracy of predicted features, positions and sizes are measured by Euclidean distance in pixels. Table 3 shows the comparison of classification accuracy between SVM and the 3BM (by making use of sparse coding).

The experimental results show that the proposed architecture performs very well on all aspects. There-

Experiment with sparse coding					
n	50	30	20	10	5
Classification	98%	95%	95%	96%	94%
Features	1.02	0.91	0.92	0.80	0.74
Position	0.11	0.16	0.29	0.36	0.30
Size	0.06	0.07	0.08	0.09	0.06
Training time(seconds)	5.75	3.47	2.31	1.16	1.09

Table 1: The accuracy of the predicted features of the next image, accuracy of the classification of the object in the image, the accuracy of the predicted position of the object, and the accuracy of the predicted size.

Experiment without sparse coding					
n	50	30	20	10	5
Classification	99%	99%	99%	98%	98%
Features	0.83	0.97	0.65	0.61	0.87
Position	0.13	0.19	0.22	0.21	0.28
Size	0.04	0.08	0.07	0.04	0.06
Training time(seconds)	3.01×10^3	2.16×10^3	1.46×10^3	0.89×10^3	0.72×10^3

Table 2: The experiment results by using the original images (without progressing the images by sparse coding)

Comparison of classification accuracy					
n	500	200	100	50	10
SVM	97%	96%	89%	72%	68%
3BM	98%	95%	96%	94%	94%

Table 3: The accuracy of classification by using SVM and the 3BM based on different number of training images(with spars coding)

fore the following conclusions can be drawn:

Conclusion IV The proposed architecture is capable of correctly learning the relations between the images in a sequence.

Conclusion V The proposed architecture is capable of making accurate predictions.

Conclusion VI The proposed architecture is capable of making very good classification. Compare to the SVM, whose performance shows downtrend as the amount of training samples decreases, the 3BM shows a relatively stable tendency.

Conclusion VII The experiments show that the 3BM without preprocessing the images using sparse coding preforms slightly better than with using sparse coding. However, the use of sparse coding significantly reduces the training time. This reduction in the training time can be attributed to the lower dimension of input data of the 3BM.

5 Conclusions and Future Work

In this work a new architecture capable of classifying objects as well as predicting the next image of a sequence of images, has been presented. The approach made use of sparse coding, and a Three-Way Factored

Conditional Restricted Boltzmann Machine, a form of deep learning, to achieve the results. Experiments performed on both the use of sparse coding and the Three-Way Factored Conditional Restricted Boltzmann Machine show the effectiveness of proposed architecture.

There are many interesting directions for future research. Implementing the proposed method on real-world robotics constitute an interesting future research direction. Furthermore, other forms of deep learning techniques, such as deep belief networks can be introduced. Finally, image segmentation in an environment containing many object has to be investigated.

References

[1] H. Ackley, E. Hinton, and J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, pages 147–169, 1985.

[2] Samy Bengio, Fernando Pereira, Yoram Singer, and Dennis Strelow. Group sparse coding. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 82–89. NIPS, 2009.

[3] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009. Also published as a book. Now Publishers, 2009.

[4] Tony F Chan and Luminita A Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

[5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[6] Siqi Chen, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. Using conditional restricted boltzmann machine for highly competitive negotiation tasks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Beijing, China, 2013.

[7] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training support vector machines. *The Journal of Machine Learning Research*, 6:1889–1918, 2005.

[8] Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800, August 2002.

[9] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[10] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.

[11] Volodymyr Mnih, Hugo Larochelle, and Geoffrey Hinton. Conditional restricted boltzmann machines for structured output prediction. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2011.

[12] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *International Conference on Machine Learning*, 2004.

[13] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the Twenty-fourth International Conference on Machine Learning*, 2007.

[14] Nicolas Le Roux and Yoshua Bengio. Representational power of restricted boltzmann machines and deep belief networks. Technical report, Dept. IRO, Université de Montréal, 2007.