Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images

Marc'Aurelio Ranzato **Alex Krizhevsky** **Geoffrey E. Hinton**
Department of Computer Science - University of Toronto
Toronto, ON M5S 3G4, CANADA

Abstract

Deep belief nets have been successful in modeling handwritten characters, but it has proved more difficult to apply them to real images. The problem lies in the restricted Boltzmann machine (RBM) which is used as a module for learning deep belief nets one layer at a time. The Gaussian-Binary RBMs that have been used to model real-valued data are not a good way to model the covariance structure of natural images. We propose a factored 3-way RBM that uses the states of its hidden units to represent abnormalities in the local covariance structure of an image. This provides a probabilistic framework for the widely used simple/complex cell architecture. Our model learns binary features that work very well for object recognition on the "tiny images" data set. Even better features are obtained by then using standard binary RBM's to learn a deeper model.

1 Introduction

Deep belief nets (DBNs) (Hinton et al., 2006a) are generative models with multiple, densely connected layers of non-linear latent variables. Unlike mixture models, many of the variables in a layer can contribute simultaneously when generating data, so DBN's have exponentially more representational power than mixture models. DBN's are relatively easy to learn from unlabeled data because it is possible to learn one layer at a time and learning each layer is relatively straightforward. After a DBN has been learned it is very easy to sample fairly accurately from the posterior over the latent variables.

The efficient learning, fast inference and high representational power make DBNs very attractive for modeling perceptual processes such as object recognition. The easiest way to use a DBN for classification is to first learn multiple layers of features on unlabeled data and then train a multinomial logistic classifier using the top layer of features as input. After learning on unlabeled data, it is possible to back-propagate through the layers of features (Hinton and Salakhutdinov, 2006), though this can cause overfitting, especially when there is not much labeled data. In this paper we do not use back-propagation to fine tune the features found by unsupervised learning.

DBNs were first developed for binary data using a Restricted Boltzmann Machine (RBM) as the basic module for learning each layer (Hinton et al., 2006a). A standard RBM is an undirected graphical model with one layer of binary visible units for representing a data-point and one layer of binary hidden units that learn to extract stochastic binary features which capture correlations in the data. The hidden and visible biases and the matrix of weights connecting the visible and hidden units are easy to train using contrastive divergence learning which is a crude but efficient approximation to maximum likelihood learning (Carreira-Perpignan and Hinton, 2005). A simple introduction to RBMs can be found at Hinton (2007).

For real-valued data, such as natural image patches, binary units or their mean-field approximations are inadequate but it is possible to make the visible units of an RBM be Gaussian variables (Lee et al., 2009), a model dubbed Gaussian RBM. This is much slower to train (Krizhevsky, 2009) and it is not a good model of the covariance structure of an image because it does not capture the fact that the intensity of a pixel is almost always almost exactly the average of its neighbors. Also, it lacks a type of structure that has proved very effective in vision applications: The outputs of linear filters are passed through a rectifying non-linearity and then similar filters have their outputs pooled by a "complex" cell which exhibits some local invariance due to the pooling (Fukushima and Miyake, 1982, LeCun et al., 1998, Serre et al., 2005). In this paper, we show that RBM's with real-valued visible units and binary hidden units can be modified to incorporate 3-way interactions that allow the

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

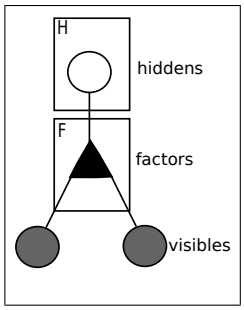

Figure 1: A graphical representation of the factored 3-way RBM. The triangular symbol represents a factor. Two identical copies of the visible units are shown in order to emphasize the relationship to a previous model (Memisevic and Hinton, 2010) that learns to extract motion from a sequential pair of images. Our model is equivalent to constraining the images in the sequence to be identical.

hidden units to control the covariances of the visible units, not just the biases as was done in Osindero and Hinton (2008). To keep the number of parameters under control, it is necessary to factor the 3-way interactions, and the factors turn out to look remarkably like simple cells: They act as linear filters that send their squared outputs to the hidden units, and they learn to be local, oriented edge detectors.

In section 2 we explain factored 3-way interactions assuming that all of the units are binary. In section 3 we describe how the reconstruction phase of the contrastive divergence learning procedure can be modified to deal with real-valued visible units that are not independent given the hidden states. In section 6 we show the filters that are learned on gray-level patches of natural images and also on smaller patches of low-resolution color images harvested from the web. We also show that the binary hidden units learned by our model are good for object recognition and that even better features can be obtained by stacking a number of standard RBM's on top of these binary features to produce a DBN.

2 A more powerful module for deep learning

Restricted Boltzmann machines can be modified to allow the states of the hidden units to modulate pairwise interactions between the visible units. The energy function is redefined in terms of 3-way multiplicative interactions (Sejnowski, 1986) between two visible binary units, v_i, v_j, and one hidden binary unit h_k:

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i,j,k} v_i v_j h_k W_{ijk} \qquad (1)$$

This way of allowing hidden units to modulate interactions between visible units has far too many parameters. For real images we expect the required lateral interactions to have a lot of regular structure. A hidden unit that represents a vertical occluding edge, for example, needs to modulate the lateral interactions so as to eliminate horizontal interpolation of intensities in the region of the edge. This regular structure can be approximated by modeling the three-way weights as a sum of "factors", f, each of which is a three-way outer product $W_{ijk} = \sum_f B_{if} C_{jf} P_{kf}$, where matrices B and C have as many rows as number of pixels and as many columns as number of factors, and P has as many rows as number of hidden units and as many columns as number of factors. Since the factors are connected twice to the same image through matrices B and C, it is natural to tie their weights further reducing the number of parameters, yielding the final parameterization $W_{ijk} = \sum_f C_{if} C_{jf} P_{kf}$. Thus Eq. 1 becomes:

$$\begin{aligned} -E(\mathbf{v},\mathbf{h}) &= \sum_f \left(\sum_i v_i C_{if}\right)\left(\sum_j v_j C_{jf}\right)\left(\sum_k h_k P_{kf}\right) \\ &= \sum_f \left(\sum_i v_i C_{if}\right)^2 \left(\sum_k h_k P_{kf}\right) \end{aligned} \qquad (2)$$

The parameters of the model can be learned by maximizing the log likelihood, whose gradient is given by:

$$\frac{\partial L}{\partial w} = <\frac{\partial E}{\partial w}>_{\text{model}} - <\frac{\partial E}{\partial w}>_{\text{data}} \qquad (3)$$

where w represents a generic parameter in the model and the angle brackets represent expectations under the distribution specified by the subscript. Fortunately, the intractable integral over the model distribution can be approximated by drawing samples from the distribution. We can draw biased samples by running a Markov chain Monte Carlo algorithm for a very short time starting at the data, as proposed by Hinton (2002). This is called "contrastive divergence" learning.

Given the states of the hidden units, the visible units form a Markov Random Field in which the effective pairwise interaction weight between v_i and v_j is $\sum_k \sum_f h_k C_{if} C_{jf} P_{kf}$. The hidden units remain conditionally independent given the states of the visible units and their binary states can be sampled using:

$$p(h_k = 1 \mid \mathbf{v}) = \sigma\left(\sum_f P_{kf}(\sum_i v_i C_{if})^2 + b_k\right) \qquad (4)$$

where σ is a logistic function and b_k is the bias of the k-th hidden unit. Given the hidden states, however, the visible units are no longer independent so it is much more difficult to compute the reconstruction of the data from the hidden states that is required for contrastive divergence learning. However, in the binary case a mean field approximation can be used yielding to a message passing algorithm (Hinton, 2010) on the graph of fig. 1.

3 Producing reconstructions using hybrid Monte Carlo

In *conditional* 3-way factored models (Taylor and Hinton, 2009, Memisevic and Hinton, 2010), one of the two sets of visible variables is held constant during reconstruction. As a result, the visible variables in the other visible set are all conditionally independent given the hidden variables, so they can all be sampled independently from their exact conditional distributions. In our *joint* 3-way model, both copies of the visible variables need to be identically reconstructed, and since the pixels interact it is harder to sample from the exact distribution. It would be possible to invert the inverse covariance matrix specified by the current states of the hidden units and to sample from the true distribution, but this is an expensive operation for the inner loop of a learning algorithm.

Fortunately, the reconstruction does not need to be an exact sample. For contrastive divergence learning to work all that is required is that the reconstruction be closer than the data to the joint distribution of the visibles given the current state of the hiddens. This could be achieved by one or more rounds of sequential Gibbs sampling of the visibles, but it is more efficient to *integrate out the hidden units* and use the hybrid Monte Carlo algorithm (HMC) (Neal, 1996) on the free energy:

$$F(\mathbf{v}) = -\sum_k \log(1+\exp(\sum_f P_{kf}(\sum_i C_{if}v_i)^2 + b_k)) \tag{5}$$

HMC draws a sample by simulating a particle moving on this free energy surface. The particle starts at the data-point and is given an initial random momentum sampled from a spherical Gaussian with unit variance. It then moves over the surface using the gradient of the free energy to determine its acceleration. If the simulation is accurate, the sum of the potential and kinetic energies will not change and the results of the simulation can be accepted. If the total energy rises during the simulation, the result is accepted with a probability equal to the negative exponential of the total energy increase. Numerical errors can be reduced by using "leapfrog" steps (Neal, 1996). After simulating the trajectory for a number of leapfrog steps, the current momentum is discarded and a new momentum is sampled from a spherical Gaussian. This Markov chain will eventually sample from the correct distribution, but we only need to run it for a small fraction of the time this would take.

4 Learning

We learn the factored 3-way model by maximizing the likelihood using stochastic gradient ascent on mini-batches and contrastive divergence to approximate the derivative of the log partition function. Samples are drawn by using HMC as described in the previous section. The algorithm proceeds as follows:
1) compute the derivative of the free energy in Eq. 5 w.r.t. to the parameters (visible-to-factor, factor-to-hidden weights and hidden biases) at the training samples,
2) draw (approximate) samples from the distribution by using contrastive divergence with HMC,
3) compute the derivatives of the free energy w.r.t. the parameters at the samples given by HMC,
4) update the parameters by taking the difference of these derivatives as shown in Eq. 3.

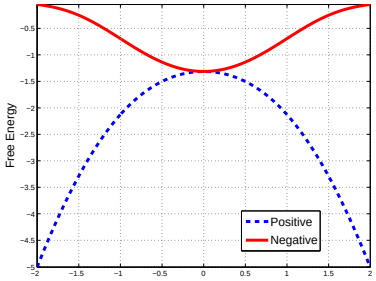

Figure 2: Free energy of a factored 3-way RBM with only one factor and one hidden unit as a function of the factor input, using a positive (dashed line) or negative (continuous line) factor-to-hidden matrix.

5 The sign of the factor-to-hidden weights

Consider a system with one factor, and one hidden unit. If the factor-to-hidden weight is positive, there is no lower bound to the energy that can be achieved by using extreme values for the visible units (see Eq. 5 and the dashed curve in fig. 2). Extreme values can be contained by using a quadratic energy penalty on each visible value, as is done in a Gaussian-binary RBM, but this makes it very hard to model occasional extreme values if the representation is overcomplete. If the factor-to-hidden weights are constrained to be negative, this problem disappears because a hidden unit simply turns off when it receives extremely negative total input (see Eq. 4). To allow a hidden unit to model a constraint that is normally satisfied but occasionally strongly violated, it is necessary to use negative weights and a positive bias. The hidden unit then creates negative energy when the constraint is satisfied and zero energy when it is strongly violated as shown by the continuous line in fig. 2. The similarity of the shape of this energy with that of a heavy-tailed distribution is further investigated in sec. 7.

Our learning algorithm includes the negativity constraint by simply setting positive factor-to-hidden weights to zero after each weight update.

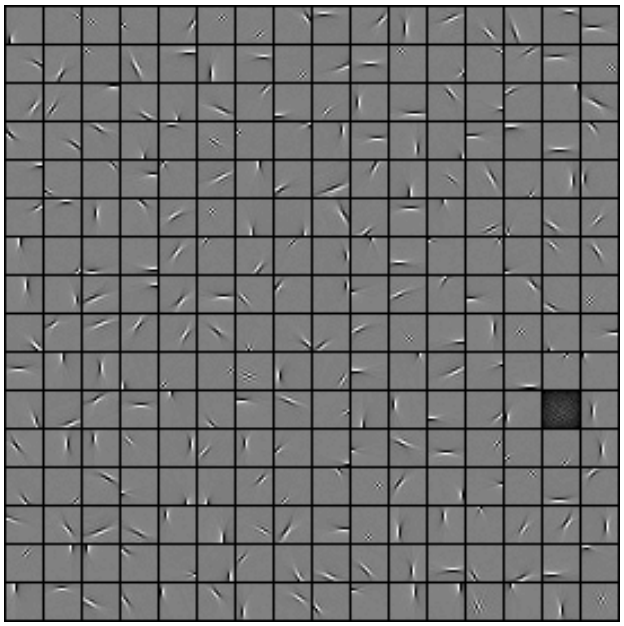

Figure 3: 256 filters of size 16x16 pixels (columns of visible-to-factor matrix) learned on whitened image patches sampled from the Berkeley data set.

Figure 4: Location and size of filters learned on the Berkeley data set.

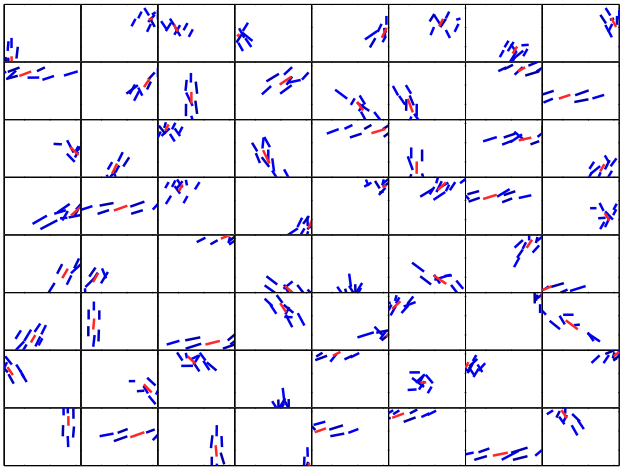

Figure 5: Grouping induced by the factor-to-hidden matrix: visualization of a random subset of the rows of the matrix. The red bar corresponds to the filter with largest weight to a hidden unit, the blue bars are other filters with smaller weight. Each hidden unit *pools* features placed at nearby locations with similar orientations.

6 Experiments

The factored 3-way model is specifically designed for heavy-tailed distributions of the type that arise in low-level vision (Simoncelli and Olshausen, 2001, Olshausen and Field, 1997). We therefore applied our algorithm to a data set of natural image patches to see whether the factors learned filters that exhibited a heavy-tailed output distribution.

One way to evaluate the features learned by the 3-way model is to use them for object recognition. We used the features to classify images in the CIFAR-10 data set (Torralba et al., 2008a, Krizhevsky, 2009) and we achieved significantly higher accuracy than the carefully engineered and widely used GIST descriptors (Oliva and Torralba, 2001). We also achieved slightly higher accuracy than the features learned by a Gaussian-Binary RBM (Krizhevsky, 2009) which has the best published performance for this data set.

6.1 Modelling Natural Image Patches

We randomly sampled 100,000 image patches of size 16x16 pixels from the images of the Berkeley segmentation data set [1]. After gray-scale conversion, the patches were ZCA whitened. Whitening is not necessary but speeds up the convergence of the algorithm. We constrained the factor-to-hidden matrix to be non-positive and with length-normalized columns by projecting after every parameter update. Training is performed by approximating the maximum likelihood gradient with Contrastive Divergence, as described in sec. 2, using the hybrid Monte Carlo method described in section 3 to generate the required reconstructions. The hybrid Monte Carlo started at a data-point, chose a random initial momentum and then ran for 20 leapfrog steps to produce a reconstruction. The step size was dynamically adjusted to keep the rejection rate around 10%.

During the first part of the training we learn only the visible-to-factor matrix keeping the factor-to-hidden matrix fixed at the negative identity. After the filters converged,

[1] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

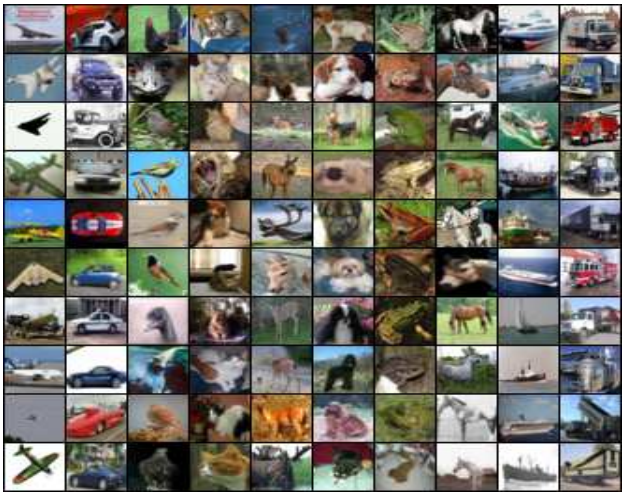

Figure 6: Example of images in the CIFAR-10 data set; each column shows examples belonging to the same class.

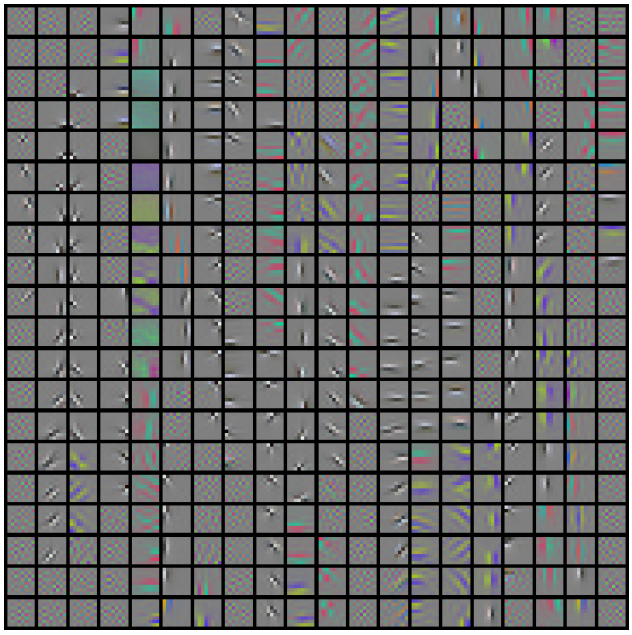

Figure 7: 400 filters in the visible-to-factor matrix learned on 2 million tiny images. The factor-to-hidden matrix was initialized to learn a one-dimensional topography enforcing similarity between nearby filters (scan the figure from bottom to top, and from left to right). This figure is best viewed in color.

we then trained both the visible-to-factor matrix and the factor-to-hidden matrix simultaneously, similarly to Osindero et al. (2006), Koster and Hyvarinen (2007).

After training, each factor can be visualized with the corresponding filter as shown in fig. 3. As expected these filters look like band-pass oriented and localized edge detectors, reminiscent of simple cell receptive fields in area V1 of the visual cortex. These features are strikingly similar to those learned by ICA models (Hyvarinen et al., 2001) and other sparse coding algorithms (Olshausen and Field, 1997, Teh et al., 2003). These features are well described by Gabor wavelets and fig. 4 uses this fit to show that they tile the space nicely.

One way to look at the factor-to-hidden matrix is through the non-zero weights in its rows. Each such weight corresponds to a factor and can be represented by its Gabor fit. Fig. 5 shows that each hidden unit pools features that have similar orientation and position, achieving *de facto* a more invariant representation. This result confirms what was found also by other authors (Osindero et al., 2006, Hinton et al., 2006b, Karklin and Lewicki, 2009, Koster and Hyvarinen, 2007, Kavukcuoglu et al., 2009) using related models.

6.2 Recognition on CIFAR-10

The CIFAR-10 data set (Krizhevsky, 2009) is a hand-labeled subset of a much larger data set of 80 million tiny images (Torralba et al., 2008a), see fig. 6. These images were downloaded from the web and down-sampled to a very low resolution, just 32x32 pixels. The CIFAR-10 subset has ten object categories, namely airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. The training set has 5000 samples per class, the test set has 1000 samples per class. The low resolution and extreme variability make recognition very difficult and a traditional method based on features extracted at interest-points is unlikely to work well. Moreover, extracting features from such images using carefully engineered descriptors like SIFT (Lowe, 2004) or GIST (Oliva and Torralba, 2001) is also likely to be sub-optimal since these descriptors were designed to work well on higher resolution images. Previous work on this data set has used GIST (Torralba et al., 2008b) and Gaussian RBM's (Krizhevsky, 2009).

We follow a very simple protocol. We train the 3-way factor model on small 8x8 color image patches, and then we apply the algorithm to extract features convolutionally over the whole 32x32 image. After extracting the features, we use a multinomial logistic regression classifier to recognize the object category in the image. Training the 3-way factor model does not require labeled data, so we train the model using a set of two million images from the TINY data set that does not overlap with the labeled, CIFAR-10 subset. This procedure prevents over-fitting of models with lots of parameters and improves generalization in data sets with paucity of reliable labeled data (Hinton et al., 2006a, Ranzato et al., 2007, Raina et al., 2007).

In particular, we trained on ZCA whitened images as processed by Krizhevsky (2009). We used the same set up for HMC as described earlier, as well as the same con-

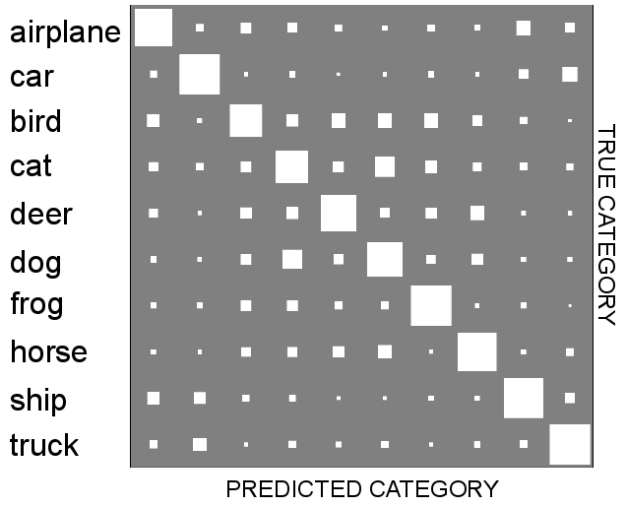

Figure 8: Confusion matrix produced by the third stage 4096 dimensional features on the CIFAR-10 data set.

straints on the factor-to-hidden matrix. In addition to that, we found it beneficial to normalize the length of the input patch as well as the lengths of the filters in the visible-to-factor matrix. The first rescaling is analogous to a local contrast normalization of the input (Pinto et al., 2008) and avoids excessive saturation of hidden units for patches with high contrast. The normalization of the filters also prevents saturation of units and, in addition, it avoids the tendency for some of them to become very small (Osindero et al., 2006, Olshausen and Field, 1997). We trained 400 factors with color filters of size 8x8, and we initialized the factor-to-hidden matrix to form a one-dimensional topography. Initially each hidden unit is connected to 9 factors through a Gaussian weighting. If we imagine laying out the 400 factors on a line, these Gaussian windows are stepped by 2 producing 200 hidden units. We first train the visible-to-factor matrix and subsequently we let the factor-to-hidden matrix adapt as done in the previous section. Fig. 7 shows the learned filters in the visible-to-factor matrix. Although the factor-to-hidden matrix is adapted, much of the 1 dimensional topography is left since nearby filters (by scanning along the columns) tend to be similar. Notice how the model learns a nice mix of high-frequency, gray-scale edge detectors and lower frequency color derivative filters in different orientations and scales. All of the gray-scale detectors were colored earlier in the learning, so they *learned* to be exactly balanced in the RGB channels.

Our model encodes 8x8 color patches into a 200 dimensional feature vector. In order to represent the whole 32x32 image, we apply the feature extractor on a 7x7 grid by stepping every 4 pixels in the horizontal and vertical direction. This produces a 9,800 (200x7x7) dimensional feature vector for each image. By using this representation to train a logistic regression classifier on the CIFAR-10 training set,

Table 1: Recognition accuracy on the CIFAR-10 test and training (in parenthesis) data sets varying depth and feature dimensionality. The second stage is trained on the 9,800 hidden probabilities produced by the first stage. Each subsequent stage is trained on the 4096 hidden probabilities from the previous stage.

Dimen.	1st stage	2nd stage	3rd stage	4th stage
9,800	62.8 (71.2)			
4,096		64.7 (71.3)	**65.3 (69.1)**	63.2 (66.8)
1,024		61.2 (65.5)	62.8 (66.7)	61.9 (64.9)
384		56.8 (58.3)	58.7 (61.3)	58.2 (59.9)

Table 2: Test recognition accuracy on the CIFAR-10 data set produced by different methods. Features are fed to a multinomial logistic regression classifier for recognition. Results marked by (*) are obtained from Krizhevsky (2009).

Method	Accuracy %
384 dimens. GIST	54.7
10,000 lin. random proj.	36.0
10K GRBM(*), 1 layer, ZCA'd images	59.6
10K GRBM(*), 1 layer	63.8
10K GRBM(*), 1 layer with fine-tuning	64.8
10K GRBM(*), 2 layers	56.6
9,800 3-Way, 1 layer, ZCA'd images	**62.8**
4,096 3-Way, 3 layer, ZCA'd images	**65.3**
384 3 -Way, 3 layer, ZCA'd images	**58.7**
19,600 3-Way, 1 layer, ZCA'd images, no pool	62.3

we achieve a recognition rate equal to 62.3%. This result should be compared to 59.6% achieved by a Gaussian RBM trained on the same whitened data and 63.8% achieved by a Gaussian RBM on unprocessed data (Krizhevsky, 2009).

Since the representation produced by the 3-way factor model is binary, it is very easy to build a deep model (Hinton et al., 2006a) by training standard binary RBM's. We have trained up to 4 stages as reported in table 1. The performance on the test set and the generalization improve up to the third stage. The third stage representation achieves the best reported accuracy on this data set while using a lower dimensional representation than Krizhevsky (2009), 65.3% with a 4096-dimensional feature vector. The corresponding confusion matrix is shown in fig. 8. The most common misclassification is between dog and cat, and animal categories tend not to be confused with man-made object categories (except for the category bird and airplane).

Compared to Gaussian RBM's, our model not only achieves better performance but also improves generalization when used as first stage in a deep hierarchical model. This improved generalization is achieved thanks to the invariance that is gained by pooling and rectifying the in-

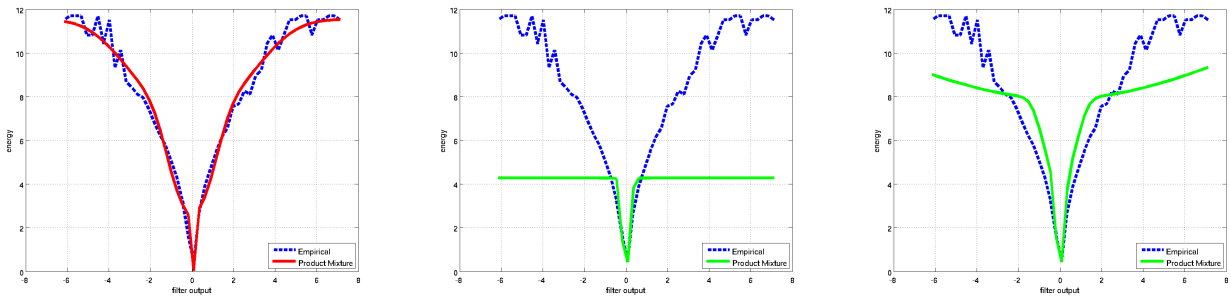

Figure 9: Factored 3-way RBM approximating a heavy-tailed distribution (empirical distribution of a Gabor response to natural images, dashed line) by using hidden units that have different scaling of the factor-hidden matrix and different biases (continuous line). The target distribution is approximated with a product of simpler distributions (a two component mixture whose energy is given in Eq. 5). Left: least square fit to the empirical distribution of a model with replicated weights (three terms as in Eq. 5, each with a different bias and scaling); the fit is not weighted by the number of data points in each bin. Center: *learning* the empirical distribution with a single 3-way factor. Right: learning the empirical distribution with a replicated 3-way factor having three biases and 3 scalings. The poor fit on the tails is due to data scarcity.

put. The importance of pooling can be assessed by setting the factor-to-hidden matrix to negative identity (which eliminates the pooling operated by the factor-to-hidden matrix); as reported in the last row of table 2 the accuracy gets slightly worse even if the feature dimensionality is twice as big. Table 2 shows also that our model consistently outperforms GIST at each layer by quite a large margin.

7 Relation to Previous Work and Extensions

The seminal work of Geman and Geman (1984) and following related methods (Black and Rangarajan, 1996, Hinton and Teh, 2001) inspired this work in the way images are interpreted. Images are assumed to be almost always smooth, but rare violations of this constraint are allowed as well. Like the Gaussian Scale Mixture model (Wainwright and Simoncelli, 2000), our model uses hidden variables that control the modeled covariance between pixels, but our hidden variables are binary and the inference process is simpler. Our model is very similar to the PoT (Product of t-distributions) model (Osindero et al., 2006) but has some important differences. The main advantage of our model is that it uses binary hidden units which are ideal for subsequent stacking of RBM's to form a DBN. The main advantage of the PoT model is that the negative log of the heavy-tailed student t-distribution does not flatten out nearly as quickly as the free energy contributed by one of our hidden units (see central panel in fig.9). As a result, the PoT model can cope better with a wide range of values of the filter output. In fact, for large filter outputs, the derivative of the PoT energy function with respect to a filter weight is invariant to scaling of all the intensities in the image.

It is possible to approximate the energy function of the PoT model by using several hidden units that differ only in their biases and the scale of the weights connecting them to the factors. The scale determines the slope of the energy function when the hidden unit is firmly on and the bias determines the vertical scale of the energy function. Figure 9 shows that hidden units differing only in scale and bias can approximate a heavy-tailed distribution quite well.

8 Conclusion

We proposed a new model, a factored 3-way RBM, for the joint statistics of pairs of input variables. This model maps real-valued images into factor outputs that represent local breakdowns in the normal covariance structure of an image. The model learns how to pool these factor outputs to produce a more invariant representation. Learning requires sampling in order to approximate the derivative of the log partition function, but this is relatively efficient since we can exactly integrate out the hidden variables and use HMC sampling on the free energy. Results demonstrate that the model learns the typical features of simple-complex cell models of early vision, and that the complex features are very suitable for use in DBNs. The learned feature hierarchy achieves state-of-the-art accuracy on the challenging CIFAR-10 data set beating Gaussian RBMs and GIST descriptors.

Future avenues of work include extending this model to capture structure in the mean of the data by adding direct visible to hidden connections; scaling to full resolution images by pooling not only over different factors for the same patch but also over factors for nearby patches; and using 3-way factored binary RBMs for the higher levels of a DBN.

Acknowledgements

The authors thank H. Larochelle for his suggestions and V. Mnih for making available his CUDA package for GPU (Mnih, 2009).

References

M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–92, 1996.

M. A. Carreira-Perpignan and G. E. Hinton. On contrastive divergence learning. *Artificial Intelligence and Statistics*, 2005.

K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 1982.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE PAMI*, 6: 721–741, 1984.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

G. Hinton. Learning to represent visual input. *Phil. Trans. R. Soc.*, 365(1537):177–184, 2010.

G. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

G. Hinton and Y. Teh. Discovering multiple constraints that are frequently approximately satisfied. In *UAI*, 2001.

G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006a.

G. E. Hinton. Boltzmann machine, 2007. Scholarpedia, 2(5):1668.

G. E. Hinton, S. Osindero, M. Welling, and Y. Teh. Unsupervised discovery of non-linear structure using contrastive backpropagation. *Cognitive Science*, 30:725–731, 2006b.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

Y. Karklin and M. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457:83–86, 2009.

K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *CVPR*, 2009.

U. Koster and A. Hyvarinen. A two-layer ica-like model estimated by score matching. In *ICANN*, 2007.

A. Krizhevsky. Learning multiple layers of features from tiny images, 2009. MSc Thesis, Dept. of Comp. Science, Univ. of Toronto.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of IEEE*, 86 (11):2278–2324, 1998.

H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proc. ICML*, 2009.

D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

R. Memisevic and G. Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Comp.*, pages 1–20, 2010.

V. Mnih. Cudamat: a CUDA-based matrix class for python. Technical Report UTML TR 2009-004, Dept. Computer Science, Univ. of Toronto, 2009.

R. Neal. *Bayesian learning for neural networks*. Springer-Verlag, 1996.

A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001.

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

S. Osindero and G. E. Hinton. Modeling image patches with a directed hierarchy of markov random fields. In *NIPS*, 2008.

S. Osindero, M. Welling, and G. E. Hinton. Topographic product models applied to natural scene statistics. *Neural Comp.*, 18: 344–381, 2006.

N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1), 2008.

R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, 2007.

M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

T. Sejnowski. Higher-order boltzmann machines. In *AIP Neural networks for computing*, 1986.

T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005.

E. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24: 1193–1216, 2001.

G. W. Taylor and G. E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *ICML*, 2009.

Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *JMLR*, 4:1235–1260, 2003.

A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *PAMI*, 30:1958–1970, 2008a.

A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *Computer Vision and Pattern Recognition*, 2008b.

M. Wainwright and E. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *NIPS*, 2000.