Learning Class-Specific Image Transformations with Higher-Order Boltzmann Machines

Gary B. Huang Erik Learned-Miller
University of Massachusetts Amherst
Amherst, MA
{gbhuang,elm}@cs.umass.edu

Abstract

In this paper, we examine the problem of learning a representation of image transformations specific to a complex object class, such as faces. Learning such a representation for a specific object class would allow us to perform improved, pose-invariant visual verification, such as unconstrained face verification. We build off of the method of using factored higher-order Boltzmann machines to model such image transformations. Using this approach will potentially enable us to use the model as one component of a larger deep architecture. This will allow us to use the feature information in an ordinary deep network to perform better modeling of transformations, and to infer pose estimates from the hidden representation.

We focus on applying these higher-order Boltzmann machines to the NORB 3D objects data set and the Labeled Faces in the Wild face data set. We first show two different approaches to using this method on these object classes, demonstrating that while some useful transformation information can be extracted, ultimately the simple direct application of these models to higher-resolution, complex object classes is insufficient to achieve improved visual verification performance. Instead, we believe that this method should be integrated into a larger deep architecture, and show initial results using the higher-order Boltzmann machine as the second layer of a deep architecture, above a first layer convolutional RBM.

1. Introduction

The visual verification task can be defined as the problem of determining, given two images, whether the images are of the same object class or not. This task is one way to generalize the problem of object recognition, removing the assumption that there are a fixed number of object classes or that we have seen instances, at training time, of each object class. Verification can be done at different granularities, such as determining if two images are of the same category (*e.g.* airplane, car, bike) or determining if two images are of the same instance within a category (*e.g.* two views of the same model of car). One important instance of visual verification is face verification, where, given two face images, determine whether the images are of the same person or not.

Often in visual verification, we cannot assume that we have seen prior training instances of each class. For instance, in a photo album application, a user may tag one person's face with its corresponding identity, and desire for all other instances of that same person's face in the album to be automatically tagged. In general, it would be highly unlikely that the recognizer would have been trained with an example of that person's face. This set-up is referred to as visual identification of never seen objects [12] or the unseen pair match problem [5], and is closely related to learning from one example [11].

Since the images presented in the test pairs may be of classes not represented in the training set, it is necessary to learn the manner in which an arbitrary object from the set of classes being considered can be transformed from one image to another, due to factors such as viewpoint, background, and occlusions. The large amount of intra-class variability makes the problem of visual identification of never seen objects especially difficult. In particular, the variability in face appearance due to the pose of the head is an extremely challenging aspect of unconstrained face recognition, making it difficult to determine cases of one face in two different poses and two different faces in the same pose [4].

1.1. Prior Work

Some existing methods for this task have had some success by learning some variation of a Mahalanobis metric-based similarity function on the image pairs [2, 12]. These methods are trained to give a higher similarity score to image pairs of the same object and lower scores to image pairs of different objects. Intra-class variability is handled

through methods such as combining many descriptors that are roughly pose-invariant, such as SIFT [8] and searching a small window in the second image to find a rough correspondence with the first.

Another method that has achieved success on visual identification of unseen face images is based on computing one-shot similarity scores [14]. For a pair of test images, a model is learned that separates each image in the pair from a set of negative examples, and is then applied to the other image in the pair, giving a final score that is the average of the two scores. The authors found that unconstrained face images contain a strong bias toward pose, meaning pose similarities outweigh subject identity similarities. Additional pose-invariance and improved accuracy was therefore achieved by first roughly aligning the faces and using, as negative examples, faces from approximately the same pose, obtained through clustering.

We believe that further improvements can be obtained by explicitly modeling the image transformations that capture how two images of the same object are related. Recent work has shown how image transformations such as translation and scaling can be learned using higher-order Boltzmann machines with multiplicative interactions [9, 10]. These transformations can be learned from moving dot patterns or video, and used to create a transformation-invariant metric that leads to improved performance on tasks such as digit recognition. We would like to take these methods and extend them to learn representations of image transformations for more complex object classes such as faces, and at a higher resolution suitable for use in visual verification.

A related line of work has been in learning deep architectures for traditional object classification. In particular, Convolution Deep Belief Networks have been applied to larger-sized images and obtained competitive state-of-the-art results on data sets such as Caltech-101 without using specialized hand-engineered features [7]. At the upper levels of the deep network, the learned filters capture higher-order class-specific features, such as eyes or portions of faces when trained on face images. Interestingly, these features have been found to be more invariant to transformations such as out of plane rotation than filters in the lower levels of the network [1].

It seems natural, then, to apply these deep networks to the problem of visual verification. However, these methods rely on having training data of the particular class, and hence are not directly appropriate for the unseen object task. For example, a network trained on face images would be useful for the problem of face detection, but the learned filters would not necessarily be useful in discriminating between faces of different people.

We propose to combine the two above lines of work, ideally using the higher-order Boltmann machines to learn image transformations, as one component in a larger deep architecture. We believe the two components would be mutually beneficial: the filters learned by the deep architecture at each level would provide useful information in modeling the transformations in object classes, and the transformation information would be useful in producing features appropriate for visual verification and give, as a byproduct, transformation-specific information such as pose.

In this paper, we give initial results on this task, applying these ideas to two data sets of complex objects, the NORB (small) data set [6], and the unconstrained face images data set Labeled Faces in the Wild (LFW) [5].

2. Background

A standard restricted Boltzmann machine (RBM) is an undirected graphical model over a set of visible binary random variables $\mathbf{v}$ and a set of hidden binary random variables $\mathbf{h}$ with edges forming a (generally complete) bipartite graph, connecting visible variables to hidden variables. The energy function defining the RBM is

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j c_j h_j$$

where the parameters of the RBM are the weights W_{ij} over edges and bias terms b_i and c_j. The joint distribution over all variables is given by

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h}))$$

where

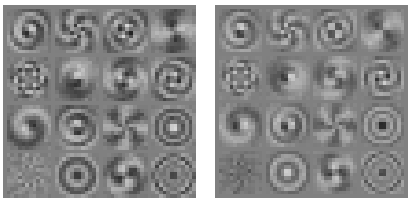
$$Z = \sum_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$$

is the partition function, normalizing the distribution. This can be extended to real-valued Gaussian visible variables by adding a v_i^2 term to the energy.

The hidden units $\mathbf{h}$ are conditionally independent given the visible units $\mathbf{v}$, and vice-versa, and so each set can be inferred exactly given the other set. Learning an RBM by maximizing the log-likelihood of the training data involves computing the expectation over the data distribution, which can be done exactly, and the expectation over the model distribution, which is intractable. This term is generally approximated using contrastive divergence, where a fixed number (generally one) of blocked Gibbs sampling iterations is performed to estimate the expectation [3].

In [9], a three-way multiplicative interaction term was added to the energy function to model the transformation from an input image $\mathbf{x}$ to an output image $\mathbf{y}$, giving a new energy term

$$E(\mathbf{y}, \mathbf{h}; \mathbf{x}) = -\sum_{ijk} W_{ijk} x_i y_j h_k$$



(a) w^x (b) w^y

Figure 1. Subset of weights in the factored RBM model learned on rotated random dot patterns.

along with lower-order bias terms if desired.

To scale this model to handle larger image patches, in [10], the three-way interaction term is factored as

$$E(\mathbf{y}, \mathbf{h}; \mathbf{x}) = -\sum_f \sum_{ijk} x_i y_j h_k w^x_{if} w^y_{jf} w^h_{kf},$$

or equivalently, by the distributive law, as

$$E(\mathbf{y}, \mathbf{h}; \mathbf{x}) = -\sum_f \left(\sum_i x_i w^x_{if}\right)\left(\sum_j y_j w^y_{jf}\right)\left(\sum_k h_k w^h_{kf}\right).$$

This reduces the number of factors from $O(N^3)$ in the original formulation to $O(N^2)$ in the factored model. Figure 1 shows a subset of the weights w^x and w^y learned on rotated random dot patterns using this model.

Once the parameters of the factored RBM model have been learned, given two input images $\mathbf{x}$ and $\mathbf{y}$, the hidden variables $\mathbf{h}$ that encode the transformation from $\mathbf{x}$ to $\mathbf{y}$ can be inferred as

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}, \mathbf{y}).$$

This $\hat{\mathbf{h}}$ can then be applied to a new image $\mathbf{x}'$ to perform an "image analogy", applying the same transformation that produced $\mathbf{y}$ from $\mathbf{x}$ to $\mathbf{x}'$.

This can also be used to construct an image metric invariant to the transformations learned by the factored RBM. A reconstructed version of the output is formed by

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{h}}),$$

producing a metric defined as

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \hat{\mathbf{y}}\|.$$

In a standard RBM, each of the hidden units is fully connected to all of the visible units, making it difficult to scale to large (*e.g.* 150x150) images. In [7], scaling to large images was achieved by using a set of fixed-size weight matrices and convolving over the entire image. The energy function of the Convolutional Deep Belief Net (CDBN) is now

$$E(\mathbf{v}, \mathbf{h}) = -\sum_k \sum_{i,j} \sum_{r,s} h^k_{ij} W^k_{rs} v_{i+r-1, j+s-1}$$

where W^k are the fixed-size weight matrices and h^k are the hidden variables associated with W^k. In additional, probabilistic max-pooling is performed, forcing at most one unit in a small neighborhood to be activated. This pooling over neighborhoods allows for invariance to small translations and reduced computational complexity at the deeper layers of the network.

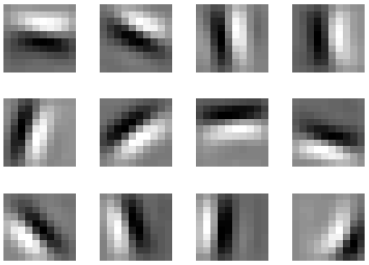


Figure 2. Weights learned on natural images in first layer of a CDBN. (Ordering done for visualization.)

Figure 2 shows the weight matrices learned in the first layer on the Kyoto natural image data set[1] using 12 hidden groups.

3. Methods and Experiments

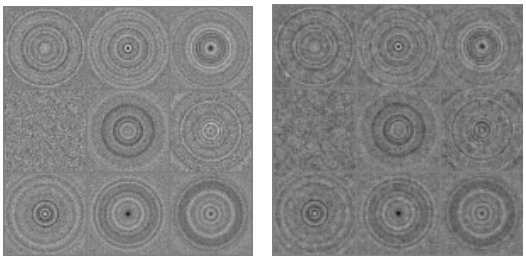
Experiments were carried out using the small (96x96) images from the NORB data set, down-sampled by one-half in each dimension, and on the LFW face images, cropping a smaller window about the head and down-sampling to 50x50 or smaller. All images were first zero-meaned and whitened by filtering with a circularly symmetric whitening filter [13].

3.1. Direct Pixel-Level Modeling

As a first step, we tried applying the factored RBM model directly to learning the image transformations present in the two object classes. We did this in two ways.

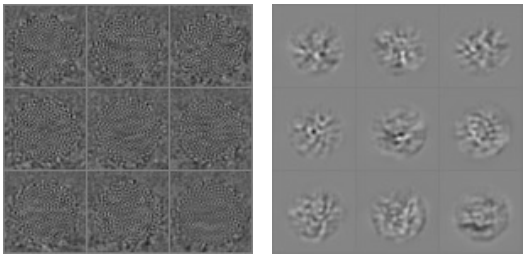
Transfer learning from random dot patterns: First we learned filters on random dot patterns that have been transformed by the types of transformations we expect to see in the object classes. For instance, in the NORB toy figures data set, the objects are centered but captured at different

[1] http://www.cnbc.cmu.edu/cplab/data_kyoto.html



(a) w^x (b) w^y

Figure 3. Subset of weights in the factored RBM model learned on rotated 48x48 random real-valued dot patterns.



(a) w^x (b) w^y

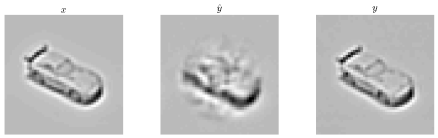
Figure 4. Subset of weights in the factored RBM model learned on NORB stereo image pairs.

rotations and elevations. Figure 3 shows a subset of the filters that were learned on rotated 48x48 random real-valued dot patterns.

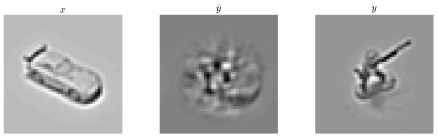
While these learned filters are themselves rotationally invariant, they do not contain the same spiral patterns as those learned on small, binary dot patterns, and do not perform well in terms of reconstructing the output (and so are not useful in constructing an invariant metric). This is most likely due to the much larger image patch size and the use of continuous real-values, both of which are necessary to apply the model to complex object classes for visual verification. Filters learned on translations and scaling were similarly not as successful as on smaller, binary dot patterns, and we were not able to learn filters that combine multiple types of transformations, such as both rotation and scaling.

Learning on image pixel values: Rather than learning on random dot patterns and transferring the model to the object classes, we next learned the transformations directly on the whitened images of the object classes themselves. Each instance in the NORB data set is given as a stereo pair, so we first learned transformations that would produce the right image from the left. Figure 4 shows a subset of the weights that were learned on the stereo image pairs, and Figure 5 shows an example of a matching stereo pair with the reconstruction of the right image in the center, and an example with a non-matching stereo pair.

In both cases, the Euclidian distance between the reconstructed image and right image was less than the distance between the left and right images, but in the matching case, the distance was reduced more, and the reconstructed image more closely matched the right image.

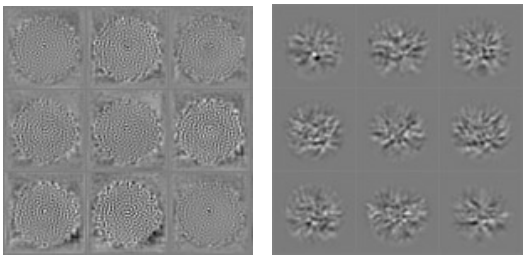


(a) matching stereo pair



(b) non-matching stereo pair

Figure 5. Examples of (left) left image of a stereo pair, (middle) reconstruction of right image given left image, and (right) right image of a stereo pair.



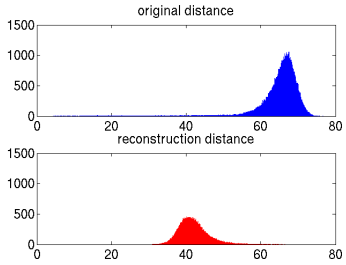
(a) w^x (b) w^y

Figure 6. Subset of weights in the factored RBM model learned on NORB airplane images.

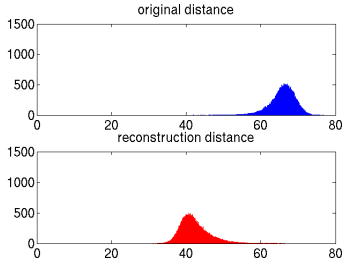
With this validation, we next tried modeling the transformations of objects due to different viewpoints. We used as training data the images of airplanes from NORB, with each input image being of a specific airplane from one viewpoint, and the output image being of the same airplane from a different viewpoint. Figure 6 shows a subset of the learned weights.

We examined whether the invariant metric defined by the learned transformations would be useful in visual verification. The transformations were learned on the images of airplanes in the training set of NORB. We constructed histograms of distances between pairs of images from the test set of NORB, and the reconstruction of the right test image with the original right test image. This was done on three sets of test cases: images of the same airplane, images of two different airplanes, and images of airplanes and a different class, in this case human figures. Figure 7 gives the three histograms of Euclidian distances, and Figure 8 shows examples of reconstructions for each of the three test cases.

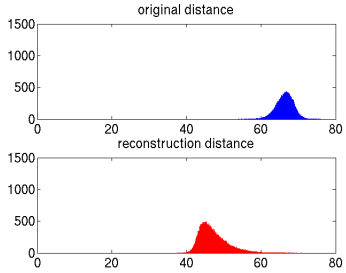
The mean distances went from 64.84, 65.88, and 66.37



(a) same airplane

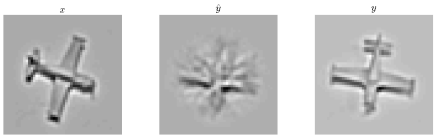


(b) two different airplanes

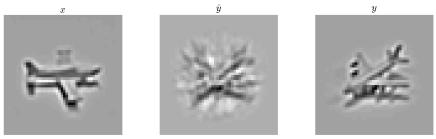


(c) airplane and human

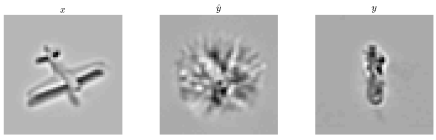
Figure 7. Histogram of Euclidian distances for three test cases, of original distance between image pairs and distance of reconstructed image to right image.



(a) same airplane

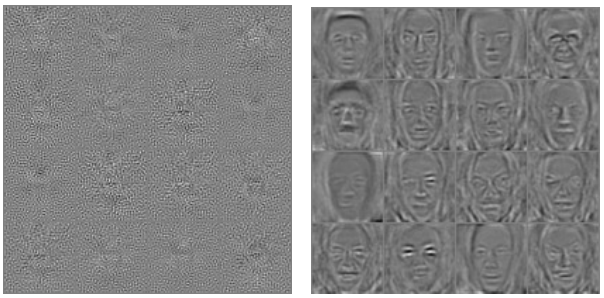


(b) two different airplanes



(c) airplane and human

Figure 8. Examples of (left) input image, (middle) reconstruction of output image given input image, and (right) output image, for three test cases.



(a) w^x (b) w^y

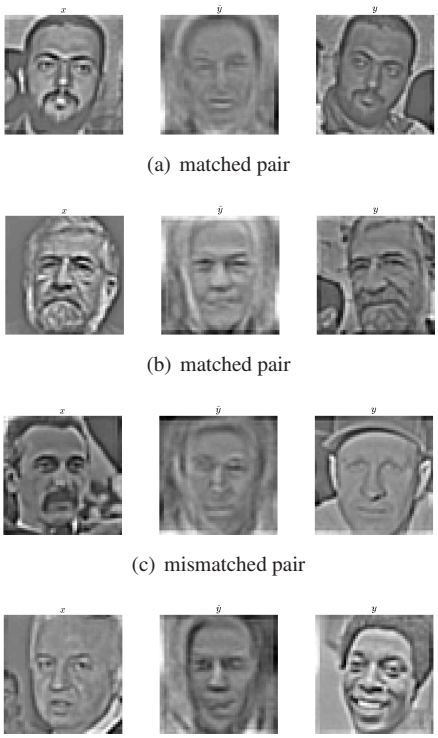
Figure 9. Subset of weights in the factored RBM model learned on LFW face images.

for the three test cases respectively, to 41.85, 42.18, and 47.09. This suggests that the model is learning some information about how airplanes in general are transformed from one image to another, but this learned information is not sufficiently based on the appearance of the input image to generate output images capable of discriminating between different types of airplanes. Thus, while this model could potentially help in traditional object recognition, it would not necessarily be appropriate for visual verification where we want to distinguish between different types of airplanes that we may not have seen before. Furthermore, the reconstructed images are very noisy, and much of the details needed for discriminating between classes is blurred away.

Finally, we applied our model to LFW face images. The learned weights are shown in Figure 9. Here it is clearer that the model is primarily learning to represent only the output face appearances, and that not much information is obtained by conditioning on the input faces. Unsurprisingly, the invariant metric derived from these filters was unable to distinguish between pairs of images of the same person and pairs of images of two different people. Examples of matched and mismatched pairs, along with reconstructions of the right image, are given in Figure 10. As with the NORB images, the reconstructions of the faces are blurry compared with the original images.

3.2. Combining Factored RBM with CDBN

In directly applying the factored RBM model to more complex object classes, we ran into problems due to the larger resolution necessary for visual verification, the need to use real-valued variables for the image data, and the amount of intra-class variation that arises from these two factors. To ameliorate this, we decided to apply the factored RBM on the first layer representation learned from a



(a) matched pair

(b) matched pair

(c) mismatched pair

(d) mismatched pair

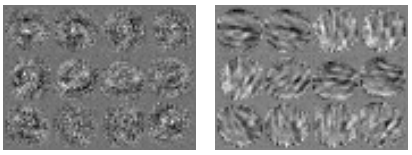
Figure 10. Examples of (left) input image, (middle) reconstruction of output image given input image, and (right) output image, for four face pairs.

CDBN with 12 hidden groups, shown in Figure 2. We use the weights in the CDBN learned from natural images, although we also obtained similar weights when learning the CDBN on images from NORB and LFW.

We believe this offers several benefits. The first layer representation is now binary, and we can work with a smaller image patch size due to max-pooling. Moreover, rather than trying to learn how arbitrary real-valued pixels are transformed from one image of an object to another, we are constraining the factored RBM to learn how the edges in the image, where presumably most of the important information resides, transform from one image to another.

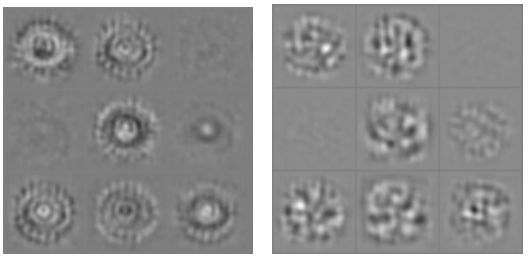
It could be argued that we do not need to bring in the machinery of CDBNs, and rather simply use some set of edge filter banks combined with down-sampling, and achieve the same representation. However, we believe that using the first layer representation is merely the first step in combining factored RBMs with CDBNs. Ideally, we would like to use the representations learned at multiple levels of the CDBN to model the transformations in the object class, and also have an interplay between the feature learning and transformation learning.

We learned the same factored RBM on NORB airplane images as above, except on the first layer representation obtained from the CDBN. Figure 11 shows one learned factor, in the first layer representation space.



(a) w^x (b) w^y

Figure 11. Example of one learned factor on airplanes in first layer representation space.



(a) w^x (b) w^y

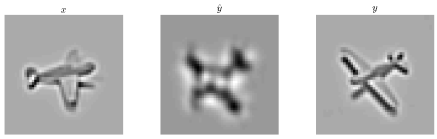
Figure 12. Subset of weights in the factored RBM model learned on NORB airplane images using CDBN first layer representation.

For visualization, these factors can be projected back onto the original image space using the CDBN weights. A subset of the learned factors projected onto the original space are shown in Figure 12.

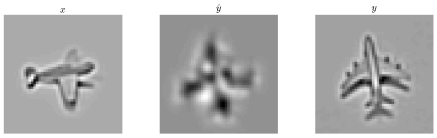
As before, the model was learning on images from the training set of NORB, and we computed histograms of distances between pairs of images in the test set under three sets of test cases: images of the same airplane, images of two different airplanes, and images of airplanes and human figures. This time, the mean distances went from 32.15, 32.65, and 31.81 between the original images to 22.41, 22.73, and 22.78. Like in the case of the model learned directly on the pixel values, the invariant metric defined by the transformations represented in this model does not seem suitable for use in visual verification. However, the reconstruction images produced by this model are noticeably better, in terms of matching the output image and preserving discriminative details, as can be seen in the examples in Figure 13.

Given these reconstruction results, we believe that this method has a lot of potential. Two simple ideas which may improve the results substantially are to use higher resolution images (for instance the original 96x96 images), which will contain more edge structure, and to learn the CDBN directly on the NORB images and learn multiple layers of features.

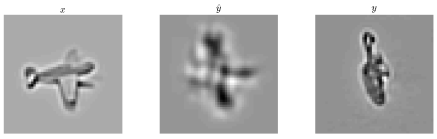
We also learned the factored RBM model on the first representation of LFW faces. Figure 14 shows a subset of learned weights projected to the original image space. In contrast to the weights learned directly on the pixel val-



(a) same airplane

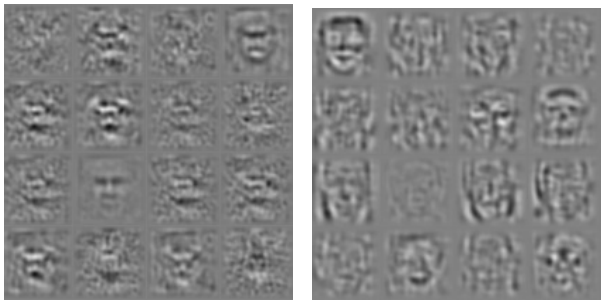


(b) two different airplanes



(c) airplane and human

Figure 13. Examples of (left) input image, (middle) reconstruction of output image given input image, and (right) output image, for three test cases.



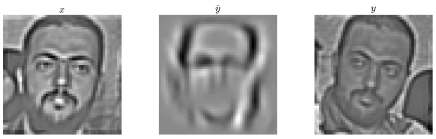
(a) w^x (b) w^y

Figure 14. Subset of weights in the factored RBM model learned on first layer representation of LFW face images.

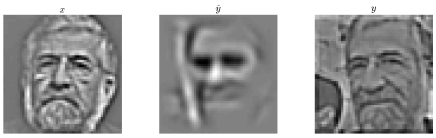
ues, there seems to be some learned structure in the input weights, rather than entirely modeling the faces using the output weights.

We learned the model using matching image pairs of people in the first 9 folds of LFW. Using images in the 10th fold, we computed distances between images in each of the 300 matched and mismatched pairs, and distances between the reconstruction of the right image and the original right image. The mean distance went from 34.55 to 26.67 for the match pairs and 36.36 to 27.31 for the mismatched pairs.

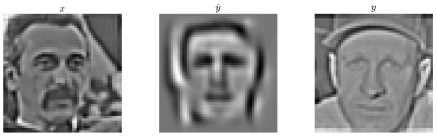
Examples of reconstructed images are given in Figure 15, for the same face pairs as above in Figure 10. Compared with the reconstructions produced by the model learned directly on pixel values, these reconstructions are superior in several respects. They are sharper, able to preserve discriminative details such as the beard in the top row and large nose in the second row. In addition, they better



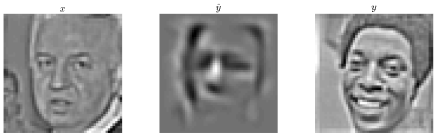
(a) matched pair



(b) matched pair



(c) mismatched pair



(d) mismatched pair

Figure 15. Examples of (left) input image, (middle) reconstruction of output image given input image, and (right) output image, for four face pairs.

capture the pose of the right image, such as the profile view in the second row and the rotation in the fourth row. Since details specific to person identity are preserved while differences due to pose are lessened, the reconstruction step could be used as the first stage in a recognition pipeline, reducing the amount of variability due to pose the recognizer must contend with. Thus, although the Euclidian distance of reconstructions to original images is not significantly less for matched pairs than mismatched pairs, the reconstructions themselves may still be useful as input for learning a face recognizer.

4. Conclusions and Future Work

In this paper, we have shown some initial work on learning a representation of class-specific image transformations for complex object classes. Direct application of existing models of factored RBMs to images of objects from classes such as faces are able to learn some information, but are insufficient to learn a transformation-invariant metric useful for visual verification.

Instead, we believe that these higher-order Boltzmann machines should be a component in a larger deep architecture, for instance integrating these models into a Convolutional Deep Belief Net. We demonstrated one example

of combining these methods by learning the higher-order Boltzmann machine on the first layer representation given by a CDBN. This allows us to work with larger image sizes, work with binary variables, and to focus on learning the transformations of edges, to preserve the discriminative information contained in the edges rather than creating blurry reconstructions.

We believe there are several promising directions for future work on these lines. The first would be learning a deep, convolutional version of the original three-way interaction RBM or factored RBM. We could model small, local transformations in the first layer, and learn how these local transformations combine to form more global transformations in the upper layers. For example, in faces, we might learn that a series of small translations in the bottom layer are actually caused by a global rotation of the head, or that small translations in the bottom half of the image are caused by a particular facial expression in the mouth.

Another direction is to learn transformations on representations given by more than just the first layer of the CDBN. Upper layer features such as corners and contours in the second layer and class-specific parts in the deeper layers provide important cues that would aid in learning transformations of objects in different images. One important aspect here would be in dealing with the increase in computational complexity as more features are being considered; the deep, convolutional version of the three-way RBM, mentioned above, may help here as well.

A more distant goal is to create a deep architecture capable of simultaneously learning both features and transformations, so that the two components can interact and benefit from one another. For instance, when learning a CDBN on faces, we may learn filters at deeper levels that correspond to eyes. However, these filters may not be able to discriminate between different types of eyes, which would be the information we want to have available when performing visual verification. If the deep architecture is able to learn both one particular type of eye, and how that one eye can be transformed from one image to another, then the network will be more likely to learn different types of eyes that would be discriminative in determining if two images are of the same person or two different people.

Our end goal is to develop a deep architecture suitable for the visual verification task. Such a network would take in a pair of images as input, and at the top-most layer produce a binary decision indicating whether the pair of images are of the same object class or two different object classes. As a byproduct, intermediary hidden units would encode transformation information, and so this model would also be useful for tasks such as pose estimation.

References

[1] I. J. Goodfellow, Q. V. Le, A. M. Saxe, H. Lee, and A. Y. Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems*, volume 22, 2009. 2

[2] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *International Conference on Computer Vision*, 2009. 1

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006. 2

[4] G. B. Huang, M. Narayana, and E. Learned-Miller. Towards unconstrained face recognition. In *Sixth IEEE Computer Society Workshop on Perceptual Organization in Computer Vision IEEE CVPR*, 2008. 1

[5] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 1, 2

[6] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition*, 2004. 2

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Twenth-Sixth International Conference on Machine Learning*, 2009. 2, 3

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 20:91–110, 2003. 2

[9] R. Memisevic and G. E. Hinton. Unsupervised learning of image transformations. In *Computer Vision and Pattern Recognition*, 2007. 2

[10] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Computation*, 2010. 2, 3

[11] E. G. Miller. *Learning from one example in machine vision by sharing probability densities*. PhD thesis, Massachusetts Institute of Technology, 2002. 1

[12] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *CVPR*, 2007. 1

[13] 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. 3

[14] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *British Machine Vision Conference*, 2009. 2