Static Gesture Recognition with Restricted Boltzmann Machines

Peter O'Donovan
Department of Computer Science, University of Toronto
6 Kings College Rd, M5S 3G4, Canada
odonovan@dgp.toronto.edu

Abstract

In this paper I investigate a new technique for the recognition of static gestures (poses) from laptop camera images. I apply Restricted Boltzmann Machines (RBMs) to model the manifold of 3 human gestures: pointing, thumbs up, fingers spread, as well as the default no-gesture case. The generative RBM model performs significantly better than other classification techniques including classical discriminative neural networks, and k-Nearest Neighbors on dimensionality reduced images. The natural extension of RBMs into time-series data also suggests that RBMs may be a powerful new tool for this difficult object recognition task.

1 Introduction and Related Work

Gesture recognition has long been an important area for researchers from many fields, including HCI, computer vision, and machine learning, and holds the promise of allowing more intuitive means of computer interaction. The current default interface of keyboard and mouse can present difficulties for new computer users, for navigation through 3D environments, or for the disabled who may have difficulty with fine motor control. Recently, the addition of web cameras to many laptop computers has lowered technological boundaries significantly and provided a wealth of opportunities for more expressive interfaces. The Toshiba Qosmio laptops released this year uses simple gesture recognition to start/stop playback of media, or to control the cursor.

In this project, I explore the use of several machine learning techniques to the classification of three right-handed static gestures: thumbs up, pointing, and fingers spread, as well as the default no-gesture case of the user working with their computer normally. Figure 1 shows examples of the 4 cases, where the first row corresponds to an average case, and the lower three rows show some exaggerated gestures from the training data. Details on the gesture data and pre-processing can be found in Section 3.

Static gesture (pose) recognition is a much simpler problem than dynamic gestures, and one that is often solved using template matching, neural networks, or other simple machine learning techniques [1]. One common approach for pre-processing hand images is an extraction of skin-tone blobs from the image [2], which are then compared to previous templates. Many systems have onerous requirements such as multiple cameras, special user requirements such as hand tracking gloves or markers, or non-real time performance.

For this project, I propose a simple classification approach where a reduced grayscale image from a single laptop camera is directly used for classification, with only edge detection for pre-processing. This allows an extremely quick and simple classifier which is suitable for real-time applications, as well as an easy means of evaluating various techniques for this object recognition task. Another reason to work directly on camera images is it allows an easier extension to more complex recognition

Figure 1: Example images of the 3 gestures as well as the non-gesture case. The first row shows the average gesture, the lower 3 show more exaggerated gestures.

tasks. More complete modeling of the laptop user is necessary for recognition of arbitrary motions of the head or body, or facial detection.

Restricted Boltzmann Machines (RBMs) are a recently developed generative model which have been used for object recognition [3], dimensionality reduction [3], and modeling time series data [4]. RBMs are stochastic multi-layer neural networks where layers are learned greedily and stacked to create a hierarchy of features in an undirected graph. I evaluate RBMs, k-Nearest Neighbors on dimensionality reduced data, and a classical discriminative neural networks, and show that RBMs perform significantly better than the other models. Furthermore, classification rates are quite good even under the very simplistic assumptions of the experiments. The models are described in more detail in Section 2, and the evaluation results are described in Section 3. Possible future extensions are discussed in Section 4.

2 Classification Methods

2.1 Restricted Boltzmann Machines

Unlike classical neural networks which learn weighted connections between layers based on back-propagated error derivatives, RBMs are stochastic neural networks where layers are trained greedily and stacked together in a hierarchy.

In its simplest form, an RBM is a layer of binary stochastic visible units fully connected by weighted links to a layer of binary stochastic hidden units. The probability that a unit activates is given by the logistic function with the sum of weighted inputs of the other layer, as well as a bias term. Once a single layer has been trained, the hidden units of one layer are used as the visible units of the next layer, creating a hierarchy of feature detectors. RBMs are trained using contrastive divergence, a method which quickly approximates the derivative of the probability the model can generate a training example given the weights. For a more complete description of RBMs, please see [3].

In this paper, the architecture of the RBM is 625 visible units connected to two layers of 500 hidden units, then connected to a 2000 hidden units layer with labels attached as a separate unit.

While real-valued visible units are possible for Restricted Boltzmann machines, binary visible and hidden units are used here for speed and simplicity. In this work, the visible units are set to be the edge magnitude of the image and the features the RBMs learns to detect are based on the location of the edgels (see Fig. 3). A further motivation for this approach is that users may wear clothes of

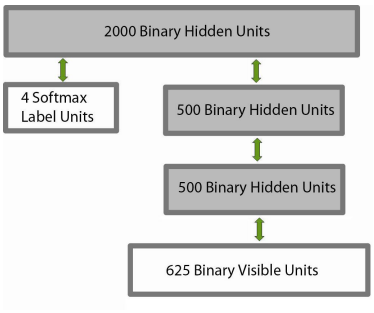

Figure 2: RBM Architecture

different color or have different skin color. An edge-based approach should hopefully be less prone to these variations.

Lastly, backpropagation is applied to the final network using the 4 labels from the training data, fine-tuning the network for improved discrimination. This corresponds to a gradient descent to a local minima after the greedy layer-by-layer training has initialized the weights of the network to a reasonable set of features. The greedy features give a good starting point for the local search of backpropagation and results in good classification rates.

2.2 k-Nearest Neighbors With PCA

One of the simplest classification methods is k-Nearest Neighbors (k-NN), where the current feature vector is compared with its k nearest neighbors from the training data to determine the class. To break ties, the nearest neighbors have weighted votes based on their ordering, that is, the kth nearest neighbor votes:

$$\frac{1}{1+2^k} \tag{1}$$

However, k-NN is a very slow process for even small images, so a dimensionality reduction technique is required. PCA is a popular technique for object recognition in the computer vision community and has been applied to many recognition tasks including gestures [5]. PCA was used to extract the first 20 eigenvectors from the covariance matrix of the training data. The training data is then reduced to 20D and stored along with the mean. During classification, the current image is subtracted by the mean, then reduced to 20D using the principal components and compared with the training data using k-NN as described above.

2.3 Neural Network

To compare the performance with a discriminative classifier, a neural network with a single layer of 200 logistic hidden units was trained with a softmax output layer. The network was training for 50 epochs, and the conjugate gradient method was applied for optimization at each epoch.

3 Experimental Results

For all three gestures, the default case was the gesture relatively centered in the image, with some translation and scale invariance as the user moved the hand around. These correspond to the first row of Fig. 1. For all 4 cases, a variety of possible poses were captured. For the gestures, this included moving the hand around the screen and at different distances to the camera for a variety of scales (see lower rows of Fig. 1) For the no-gesture case, a variety of normal images of the user moving around the desktop, including turning, leaving the screen, and leaning forward, were taken.

Input images were captured at resolution of 640x480, then downsampled to 48x48 greyscale initially. The edge magnitude was calculated using a Gaussian kernel followed by a x/y Sobel edge detector.

Figure 3: Gesture images and associated 25x25 feature vector

Two representations of the image were tested. The first was a banded downsampling which used the centre 32x32 pixels of the image, then 8 surrounding pixels downsampled by a half, then the remaining 8 pixels downsampled again. This results in a 1024 + 144 + 44 = 1212 pixel image. The second representation was a smaller 25x25=625 image downsampled directly from the 48x48 image.

Interestingly, the banded image performed roughly the same for RBMs when trained on validation data as the 25x25 image (2.92% for banded vs 2.84% for the 25x25). The RBM architecture for the banded images was 1000-1000-4000, with 70 epochs of training. Because of this, and the obvious increase in speed at 25x25, the smaller image was used for all tests below. However, higher resolution images are probably necessary for finer gestures than the one captured here.

For training, two videos of the author were taken on different days for a total of 42697 frames. Frames were considered to be i.i.d, and were randomly ordered before training. There were approximately 18000 no-gesture frames and 9000 for each gesture, giving 45000 training samples. Some gestures had less training examples (7000-8000). These were simply added again to reach 9000. Third and fourth videos were taken for validation and final test data with 11606 and 7099 frames respectively.

The assumption made in this work is that all frames are i.i.d. However, this is obviously a simplification since the frames are from sequences taken at different times. This may produce problems in classification since the clothes, background, lighting, etc, may be different day-to-day. For this reason, both the validation and test data are reported below, as they were from two separate videos taken on different days. Similar movement of a variety of hand locations and scales were taken in both training, validation, and testing data. [1]

Classification Method	**Validation Error**	**Test Error**
RBM (500-500-2000)	2.84%	5.96%
Neural Network (200 Hidden Units)	6.20%	38.2%
PCA with k-NN (20 eigenvectors, k=10)	10.2%	66.0%

Table 1: Misclassification Rates

As we can see in Table 1, RBMs perform significantly better than the other methods for both the validation and test data. For the validation set, the classification rate is only 2.84% , well below that of the regular neural network or k-NN. For the test set, the RBMs is again significantly lower than the other methods. There is also a significant decline in performance for all methods from the validation to testing data. One of the two training videos and the validation video were taken on the same day. However, the test data was taken on a separate day. I believe this decline may be due to slightly darker lighting conditions in the test data, a darker shirt, and possibly minor differences in the background between the training and test data.

This suggests that gesture recognition systems need to be fairly robust at capturing these minor changes, and/or significant pre-processing is required. Most striking is how well RBMs deals with

[1] The videos can be viewed at http://www.cs.toronto.edu/t̃onovan/gesture/

these changes compared to the other approaches. Whereas the other methods decline by a factor of roughly 6, RBMs decline much less, indicating the RBMs are the most effective at generalizing.

Gestures also appear to be fairly equally mislabeled. In the test set, misclassifications were almost equal between classes. For the validation set, the misclassifications occurred twice more often for pointing and thumbs-up than for the default case or fingers spread.

One important question for a gesture recognition system is how well the system generalizes based on the size of training data. While the user can always add more training data, it can be tiring to move a hand for an extended period of time. Furthermore, the user will have to train several gestures, possibly for both left and right hands, so determining a minimum number of frames per gesture is an important consideration. Fig. 4 shows the validation error for several sizes of training data. As we can see, the error rates seem to be plateauing as the number of training frames approach 45000, reaching a misclassification rate of 2.65% on the validation data.

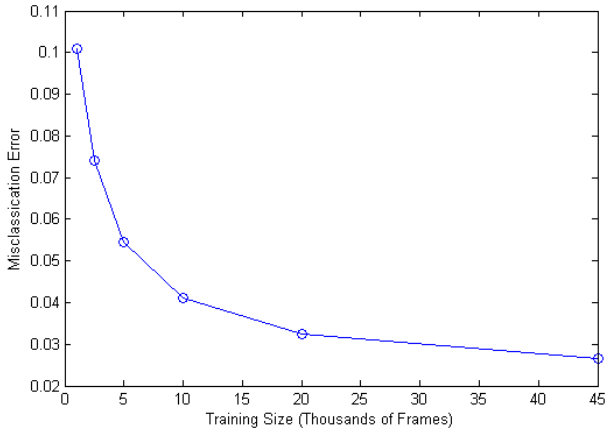

Figure 4: Validation Error vs Training Size

Another aspect of using RBMs is the relatively long training time required and the number of training epochs is a major factor of this cost. Fig. 5 shows the validation error for several training epochs. All stacked RBMs and the final backpropagation fine-tuning were all done with this number of epochs. As expected, as the number of epochs increases, the validation error decreases and then plateaus between 20 and 50 epochs with a misclassification rate of 2.9%. Due to this, the number of epochs was set to 35 for all other experiments.

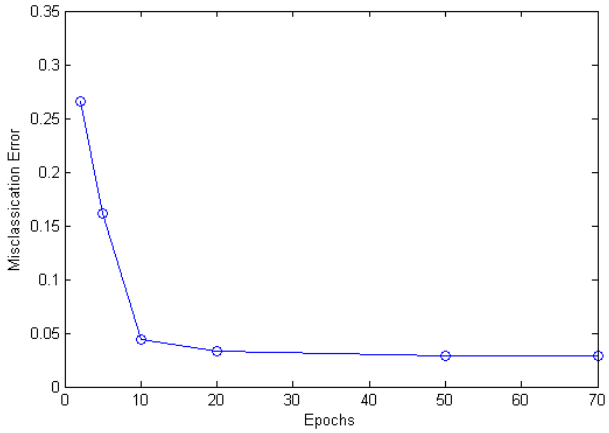

Figure 5: Validation Error vs Training Time

4 Future Extensions

The obvious extension for this project is to model gestures with a temporal component, such as sign language words. Currently, one of the main tools for recognizing temporal gestures are Hidden Markov Models [1]. However, conditional RBMs provide a more natural means of modeling these gestures as they can be directly applied to image data without the need for separate feature detection and tracking. Conditional RBMs are also more powerful than the discrete states of HMMs, and may potentially have improved classifications rates.

Another possible extension is to construct a more interactive system to fully model the images of a laptop user, including gestures, head movements, facial detection and recognition for a single or multiple users, etc. Many existing gesture recognition systems (including the one presented here), force the user to assume certain set gestures. However, particular gestures may be uncomfortable for the user, especially if often repeated. A more interesting approach is to let the user define their own gestures or head movements and map those to particular actions the user also specifies.

While RBMs have significant overhead in training time, it may be possible to construct an online version which trains more slowly, but trains with current labeled data to learn new gestures, and stored data to remember previous gestures. Getting correct labeled data is less difficult than other machine learning systems since the user can easily hold a gesture for a minute or so and move to different locations and scales. While such a system could be bootstrapped, users may also be more tolerant of a system they train themselves, and can view improving over time. Previously seen gestures images could be mixed with the new training data to maintain correct recognition of old gestures.

5 Summary and Discussion

Gesture recognition remains an important area of research for the HCI, computer vision, and machine learning communities. While there exists a great deal of research on gestures, tools for users using a simple webcam are only beginning to enter the mainstream, and there remains significant hurdles to achieving successful recognition of arbitrary gestures.

RBMs present an exciting and powerful tool for object recognition in general, and may have possible uses for gesture recognition in particular. In this paper, it's been shown that RBMs perform significantly better than other classification techniques such as k-NN on dimensionality reduced data, or classical neural networks at recognizing three distinct gestures. Furthermore, the attraction of using RBMs is that they can more fully model the manifold of images of human sitting before a laptop, allowing more general recognition tasks such as facial or movement detection. They also allow an easy extension to temporal gestures using conditional RBMs, which may provide a more natural tool for recognition than HMMs, the standard method currently employed.

Acknowledgments

Thanks to Ruslan Salakhutdinov and Carl Rasmussen for their RBM and conjugate gradient code respectively.

References

[1] Mitra, S., & Acharya, T. (2007) Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 37 (3) pp. 311-324.

[2] Starner, T & Pentland, A. (1998) Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video *IEEE Transactions on Pattern Analysis and Machine Learning* Vol 20 (12) pp. 1371-1375.

[3] Hinton, G. E. & Salakhutdinov, R. R. (2006) Reducing the dimensionality of data with neural networks. *Science* Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

[4] Taylor, G. W., Hinton, G. E. & Roweis, S. (1995) Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems 19*, MIT Press, Cambridge, MA

[5] Black, M. & Jepson, A. (1996) Eigentracking: Robust matching and tracking of articulated objects using a view-based representation *International Journal of Computer Vision* pp. 329-342.