Deep Belief Networks for Real-Time Extraction of Tongue Contours from Ultrasound During Speech

Ian Fasel & Jeff Berry
University of Arizona
ianfasel@cs.arizona.edu, jjberry@email.arizona.edu

Abstract

Ultrasound has become a useful tool for speech scientists studying mechanisms of language sound production. State-of-the-art methods for extracting tongue contours from ultrasound images of the mouth, typically based on active contour snakes [3], require considerable manual interaction by an expert linguist. In this paper we describe a novel method for fully automatic extraction of tongue contours based on a hierarchy of restricted Boltzmann machines (RBMs), i.e. deep belief networks (DBNs) [2]. Usually, DBNs are first trained generatively on sensor data, then discriminatively to predict human-provided labels of the data. In this paper we introduce the translational RBM (tRBM), which allows the DBN to make use of both human labels and raw sensor data at all stages of learning. This method achieves performance comparable to human labelers, without any temporal smoothing or human intervention required at runtime.

1. Introduction

Ultrasound imaging of the tongue is a popular method for studying articulation of speech, due to its inexpensive and non-invasive nature. Typically, ultrasound is used to view the midsagittal tongue surface contour in real time by placing the transducer beneath the chin during speech (see Figure 3). A bottleneck for these studies arises from the time-consuming process of extracting the tongue contours from the ultrasound images. Most labeling tools make use of active contour snakes [3]. For instance, in EdgeTrak [5], the most widely used software for this task, contours are initially given by the user for each sequence, then active contour energy functions are successively minimized, with each frame using the preceding contour for initialization.

While snake-based methods have proven useful for data extraction, they are still not fully automatic. The software requires manual initialization, and must be monitored by an expert since the snake often attaches to a noise artifact or loses the contour during rapid tongue movements, and must be reinitialized.

We propose an alternative method based on deep belief networks (DBNs), which are probabilistic generative models composed of multiple layers of stochastic binary units. Typically DBNs are trained in two stages: first, a latent representation of the input data is learned by a greedy layer-wise unsupervised learning algorithm for restricted Boltzmann machines (RBMs). Then a discriminative “fine-tuning” algorithm is used to optimize the network for some classification or regression task.

Here we use the human labels in both generative and discriminative learning stages. This allows the generative model to construct a shared representation of both input data (in this case, ultrasound images) and human labels (tongue contour traces) in all parts of the model before the discriminative fine-tuning. Previous work has jointly learned sensors and labels in single RBMs [4, 6] and in the top-level RBM of a DBN [1]. However in these methods inference of labels given only the sensor inputs require variational approximations, sampling, or potentially expensive marginalizations. We therefore present a new method that allows inference to be performed in a single feedforward pass through the network. This is done by learning a new RBM that transforms raw sensor inputs into a hidden representation trained to model the sensors and labels jointly during unsupervised learning.

We test our technique on extracting tongue contours from ultrasound without any human supervision. We show that the new technique performs as well as human labelers and snake-based methods, with the advantage of being able to run at video frame-rates on a desktop computer without any manual intervention.

2 DBNs as Autoencoders

We begin by describing the use of deep belief networks as autoencoders, largely recapitulating the de-
Figure 1. First, a deep net is learned on concatenated sensor and label input vectors (e.g., ultrasound and contours) to make an autoencoder (left). The first layer is then replaced with a tRBM trained to transform sensor-only inputs into the same hidden representation as the original network (right). Finally, discriminative training is used to minimize the reconstruction error of labels (e.g., contours) from sensor inputs (e.g., ultrasound).

An autoencoder is a system which takes an input vector \( x \) and maps it to a hidden representation \( h \) using an encoder network, followed by a decoder network which reconstructs the input from the hidden representation. A common way of constructing an autoencoder is through restricted Boltzmann machines (RBMs). An RBM is a specific type of Markov random field in which the \( V \) dimensional input vector \( x \) and \( H \) dimensional hidden feature vector \( h \) are modeled by products of conditional Bernoulli distributions:

\[
f_{\phi,j}(x) = p(h_j = 1 \mid x; \phi) = \sigma(\sum_{i=1}^{V} W_{i,j} x_i + a_j)
\]

\[
f'_{\phi,i}(h) = p(x_i = 1 \mid h; \phi) = \sigma(\sum_{j=1}^{H} W_{i,j} h_j + b_i)
\]

where \( \sigma(u) = (1 + e^{-u})^{-1} \), and \( \phi = (W, a, b) \). Thus when used as an autoencoder, \( f_{\phi} \) is the encoder network and \( f'_{\phi} \) is the decoder network, using shared parameters \( \phi \). The marginal distribution over visible vector \( x \) is:

\[
p(x; \phi) = \frac{\sum_h e^{-E(x,h; \phi)}}{\sum_u \sum_h e^{-E(u,h; \phi)}} \tag{1}
\]

where \( E \) is the energy of the joint configuration \( (x,h) \):

\[
E(x, h; \phi) = \sum_{i=1}^{V} \sum_{j=1}^{H} W_{i,j} x_i h_j + \sum_{i=1}^{V} b_i x_i + \sum_{j=1}^{H} a_j h_j
\]

Maximum likelihood parameters \( \phi \) can be found by taking gradient steps determined by the contrastive divergence (CD) rule \([1]\), which gives parameter updates:

\[
\Delta w_{i,j} = \langle x_i h_j \rangle_{\text{data}} - \langle \hat{x}_i \hat{h}_j \rangle \tag{2}
\]

where expectation \( \langle x_i h_j \rangle_{\text{data}} \) is the frequency that \( x_i \) and \( v_i \) are on together in the training set, and \( \langle \hat{x}_i \hat{h}_j \rangle \) is the frequency that \( \hat{x}_i \) and \( \hat{h}_j \), which represent samples of \( x_i \) and \( y_j \) obtained by running the Gibbs sampler initialized at the data for one step, are on together. Thus one gradient update involves first sampling \( h \) from \( f_{\phi}(x) \), then generating the “reconstructions” \( \hat{x} \) by sampling from \( f'_{\phi}(h) \), and finally encoding new features \( \hat{h} \) by sampling from \( f_{\phi}(\hat{x}) \), giving us all the terms needed to compute (2).

When inputs are real-valued instead of binary, Gaussian units can be used instead. In this case the only change needed to the above equations is:

\[
f'_{\phi,i}(h) = p(x_i = x_i \mid h; \phi) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i - b_i - \sum_{j=1}^{H} W_{i,j} h_j)^2}{2}}
\]

Recently it has been shown \([1]\) that higher order correlations in input can be captured by training a sequence of \( N \) RBMs, where the hidden layer of each lower-level RBM is treated as the input to the next higher-level RBM. At runtime data is then run through each encoder function \( f_{\phi_0} \) in sequence, and then is decoded by running through the decoder functions \( f'_{\phi} \) in reverse order, in what is called a deep autoencoder. On the left of Figure 1, we show a deep autoencoder where each \( \phi_i \) represents the parameters of the \( i \)th RBM.

### 2.1 Jointly modeling sensors and labels

A deep autoencoder can be used to learn joint relationships in both sensor data \( v \) that will be visible at runtime in an application and human provided labels \( l \) that are only available during training time, by training a deep autoencoder in which the inputs \( x \) to the first layer RBM consist of sensor data and human labels concatenated, i.e., \( x = (v, l) \). The hidden features learned by the deep net thus represent relationships between both the sensor and the label variables.

Unfortunately, provided a deep autoencoder trained with joint sensor and label data, it is not straightforward to compute a reconstruction of only the labels \( l \) given just the sensor data \( v \). For instance, in \([6]\), when
RMs were used to jointly model images and text annotations, a variational approximation was required to estimate text given images. [4] developed a discriminative RBM that uses both sensors and labels, and [1] trained the top-layer RBM of a DBN to jointly model labels and the inputs from the lower level RBM. Both cases used the fact that if there are a reasonably small number of discrete labels, it is possible to compute $p(l|v; \phi)$ exactly for a single RBM by marginalizing out $h$. However for deep autoencoders, or even a single-layer RBM with continuous valued labels, an exact computation is generally not feasible.

2.2 Translational RBMs

We now propose a new method for training a deep belief network that can produce an estimate of the label data given only the sensor data. We start with the intuition that a deep network trained on concatenated sensor-label inputs learns a representation that captures hidden relationships between both types of data, allowing the decoder stage to accurately reconstruct both sensors and labels given these shared features. We therefore hypothesize that if a new encoder can be trained to produce hidden features from sensor-only inputs that are identical to those produced by the original network when given sensors concatenated with labels, then the decoder from the original network can be used to reconstruct labels from the hidden codes produced by the new, sensor-only encoder.

We test this hypothesis by introducing the “translational” RBM (tRBM), which “translates” raw sensor inputs into a hidden representation that was originally learned to encode sensors and labels jointly. A tRBM is identical to a standard RBM, but training is done in two stages. In the first stage, a full deep autoencoder is trained on concatenated sensor-label data $x = (v, l)$, with bottom-layer RBM having parameters $\phi_1$. We then create a new set of parameters $\theta$ for an RBM which takes as input sensor-only data $v$, and has the same number of hidden units as the RBM with parameters $\phi_1$. This is the tRBM. For clarity, we will refer to the encoder and decoder functions for the tRBM as $g_\theta$ and $g_\theta'$. The tRBM is now trained with CD, but with a twist. For each gradient step, instead of sampling the term $h$ on the left hand side of eq. (2) from $g_\theta(v)$, we instead sample from $f_{\phi_1}(v, 1)$. Then sensor reconstructions $\hat{v}$ are sampled from $g_\theta'(h)$, and finally $\hat{h}$ are sampled from $g_\theta(\hat{v})$. This is illustrated in Fig. 2. Learning the tRBM is otherwise identical to normal RBM learning with CD.

We can now reconstruct labels from sensor inputs by re-using the original deep autoencoder, but with the first layer encoder stage replaced by the encoder in the tRBM. If the tRBM is successful at making sensor inputs have the same hidden activation that they had in the original network (when their corresponding labels were available), then the higher-order parameters of the deep network will still be able to generate both the sensors and labels even though the labels are missing from the input. Moreover, if we are ultimately interested only in the labels given sensor inputs, then we can delete the top layer weights that only reconstruct sensor data.

As in other work on deep belief nets, we finally use this generatively trained “translational” deep belief net (tDBN) as the initial parameterization of a discriminative neural network. We “fine-tune” this network by using conjugate gradient descent to minimize the squared euclidean distance between reconstructed versus true (human-labeled) outputs given sensor inputs.

3 Tongue Contour Extraction with tDBNs

We now turn to using tDBNs for extracting tongue contours from ultrasound images during speech. In this setting, the ultrasound images are the visible sensor data $v$, while the human traces of the tongue contour are the labels $l$. We crop a central $180 \times 330$ pixel region from each ultrasound image and scale to $19 \times 34$ pixels. Contours are represented as real valued images of the same size$^1$. To convert the tDBN (image-like) outputs into a list of $(\hat{x}, \hat{y})$ contour coordinates, for each $j$th column of the reconstructed contour image of height $M$, we set $\hat{x}_j = j$, and let

$$\hat{y}_j = \frac{1}{M} \sum_{i=1}^{M} \hat{\ell}_{i,j}/\tau$$

(3)

where $\hat{\ell}_{i,j}$ represents the pixel at row $i$ and column $j$ of the reconstruction, and $\tau = 0.05$. The resulting set of coordinates are then smoothed in the horizontal direction using local linear regression (Gaussian kernel, standard deviation 1 pixel), and the results are finally scaled up to original size. Fig. 3 shows examples of the full process.

---

$^1$Representing the labels as images instead of just a list of $(x, y)$ coordinates in a contour makes it possible for the tDBN to entertain multiple hypotheses about the location of the tongue at the same time.
4 Experiments and Results

Our data consisted of 7 subjects pronouncing a variety of words with the letter /l/ in frame sentences, such as bell, belling, bowl, fell, left, lemma, loaf, etc., resulting in 8640 images. The wordlist was constructed for a separate study on articulations of /l/. Each image was manually labeled by an expert linguist. We then trained a tDBN using the architecture shown in Fig. 2 (Gaussian inputs, hidden layers of 1142, 1142, and 3230 Bernoulli units). To compare the automatic curves to the hand-labeled curves for validation, we use mean sum of distances (MSD) from [5], where, given two curves \( U = (u_1, \ldots, u_n) \) and \( V = (v_1, \ldots, v_n) \),

\[
MSD(U, V) = \frac{1}{2n} \sum_{i=1}^{n} \min_j |v_i - u_j| + \sum_{i=1}^{n} \min_j |u_i - v_j|
\]

We performed five-fold cross validation and found the tDBN achieved an average MSD of 2.5443 ± 0.056 pixels (1 pixel = .295mm) which is identical both to the inter-labeler agreement and the snake based results reported in [5]. Some example results are shown in Fig. 3.

5 Discussion

Examining the results, the tDBN was surprisingly good at reconstructing contours even when shadows in the ultrasound created large gaps in the visible contours. The tDBN rarely was distracted by artifacts, and occasionally even predicted contours which extended farther than the human-labeled contour. Because the run-time algorithm requires only a single feedforward pass through the network, processing one image takes about 0.03s in MATLAB, allowing real-time labeling.

Due to limited space we will only briefly summarize results with different architectures. First, we found that generative pre-training with ultrasound only (thus no tRBM stage), using discriminative fine-tuning to predict contours from the encoder layers only (thus skipping the decoder layers), and standard neural networks without generative pre-training, all resulted in contours that were extremely poor and unusable by linguists. Second, training a tDBN with only one hidden layer also gave very poor performance. Thus apparently both the joint training method and the use of many hidden layers in an encoder-decoder arrangement is important.

Our results in this domain suggest that the tDBN method may be useful for other medical imaging applications, such as segmentation of cancerous structures in breast and prostate, cardiology, or vascular disease. Moreover, the tDBN method is general enough to be applied to a variety of other pattern recognition problems. In addition to combining sensors and labels at all stages of learning, it may provide benefit in any situation that multiple sources of training information are available.

References