Using Denoising Autoencoder for Emotion Recognition

Rui Xia, Yang Liu

1 Computer Science Department, The University of Texas at Dallas, Richardson, TX 75080
rx, yangl@hlt.utdallas.edu

Abstract
In this paper, we propose to use the denoising autoencoder to generate robust feature representations for emotion recognition. In our method, the input of the denoising autoencoder is the normalized static feature set (state-of-the-art features for emotion recognition). This input is mapped to two hidden representations: one is to capture the neutral information from the input, and the other one is used to extract emotional information. Model parameters are learned by minimizing the squared error between the original and the reconstructed input. After pre-training and fine-tuning, we use the hidden representation as features in the SVM model for emotion classification. Our experimental results show significant performance improvement compared to using the static features.

Index Terms: Emotion Recognition, autoencoder

1. Introduction
Detecting human emotions from video and audio is an important part in human-computer interaction systems. In recent years, there has been a lot of research interest in automatic emotion recognition, as evidenced by the many challenge tasks such as Interspeech Emotion Challenge 2012 [1] and Audio/Visual Emotion Challenge and Workshop in 2012 [2].

In previous studies, one of the state-of-the-art systems from speech emotion recognition is based on static features, extracted by using lots of functional on low-level descriptors (LLD) including MFCC, prosodic features, etc. SVM classifiers are often used for these features. These have shown competitive performance in many challenge tasks [3]. Besides the static feature set, other acoustic feature representations, such as GMM based model distance features [4], i-vector features [5], have also been shown to provide some performance gain when combined with the static features at the feature level or decision level. In addition, information from other sources, such as lexical information [4] can further improve performance, though extracting lexical features is highly dependent on the hand-crafted sentimental or emotional dictionary and the accuracy of the speech recognition system. Recently, deep neural networks is also applied on emotion recognition and shows significantly improvement [6].

Autoencoder has been successfully applied in many fields. Recursive autoencoders are used in some natural language processing tasks, such as predicting sentiment distributions [7] and paraphrase detection [8]. Denoising autoencoder and its deep structure, stacked denoising autoencoders, are applied successfully to digital image recognition [9], showing competitive results compared with deep belief networks using Restricted Boltzmann Machines as layer component trained with contrastive divergence [10].

One of the reasons that autoencoder achieves good performance in classification tasks is its hidden information representation based on the raw features. In this study we thus investigate the feasibility of using autoencoder for better feature representation for emotion recognition from speech. First, for the traditional denoising autoencoder (DAE), we evaluate using the hidden representation as features in the SVM classifier. Parameters are trained using pre-training to minimize the squared error, following by fine-tuning based on the class information. Second, we propose a modified DAE in order to capture more emotion specific information. In our method, each input is projected to two hidden representations. One of them is designed to capture the hidden neutral information with pre-trained parameters based on a large neutral based corpus. The other aims to extract emotion related hidden information. Based on the assumption that the emotional speech features can be viewed as shifting from the neural space, the new reconstructed input can be represented using a linear combination of two reconstructed inputs from the two hidden representations respectively. The emotion specific hidden representations are used as features in the SVM classifiers for emotion classification. Our experimental results show that using DAE generates better features and improves classification performance. Furthermore, additional gain is observed when adding another projection to the neutral hidden representation, showing that more emotion indicative features are captured in our proposed method.

2. Method
We first briefly review the classical denoising autoencoder (DAE) introduced by Vincent [9] and how it can be used for emotion recognition, and describe our proposed variation of DAE for more discriminative emotion feature representation.

2.1. Basic Denoising Autoencoder for Emotion Recognition
The major difference between denoising autoencoder (DAE) and traditional autoencoders is that DAE is trained to recover from corrupted inputs. The motivation behind this is that using corrupted input allows the model to learn more robust feature representation and thus can better recover the clean input. It can also be viewed as a regularization method such as the weight decayed method. In the following, we describe how DAE is used for emotion recognition.

There are two stages for training DAE, unsupervised pre-training and supervised fine-tuning (the supervision is with respect to the use of the class labels in the emotion recognition corpus). The unsupervised pre-training is designed to make the hidden representation better retain the information of input and better reconstruct the original input by minimizing the error between the reconstructed input and original input. Previous studies have shown that the pre-training stage can provide a better initial value than random initialization [11]. The supervised fine-tuning procedure can further update model parameters based on the labeled training instances.
In the pre-training stage, there are four steps for pre-training one layer denoising autoencoder. First, corrupted input is generated by adding noise on the clean input. Different noise functions, such as masking noise and salt-and-pepper noise introduced in [9], can be applied to make corrupted input. Here, we use Gaussian noise on the clean input \( x \) (raw features with a dimension of \( D \)) to get corrupted input \( \tilde{x} \) as:

\[
\tilde{x} \sim \mathcal{N}(x, \sigma^2 I)
\]

where \( \sigma \) is used to control the corruption level.

The second step and third step are decoding and encoding part. The corrupted input is projected to \( L \) dimensional hidden representation \( y \), and then the reconstructed input \( z \) is mapped back from \( y \).

\[
y = s(W \tilde{x} + b)
\]

\[
z = s(W' y + b')
\]

where \( s \) is the sigmoid function \( s(x) = (1 + \exp(-x))^{-1} \), \( W \) is \( L \times D \) matrix and \( b \) and \( b' \) are bias vectors of dimension \( D \). Here we use tied weights which means \( W' \) is equal to \( W^T \).

After getting reconstructed input \( z \), the squared error function is used as the loss measurement between clean input \( x \) and \( z \) as:

\[
Loss = ||z - x||^2
\]

To learn parameter \( W \), \( b \) and \( b' \), optimization methods, stochastic gradient decedent or L-BFGS, can be applied to minimize the cost function \( Loss \).

Figure 1 shows such a DAE. In our task, we use all the training data together (without using their class labels) in pre-training. The raw features \( x \) are the static features, described in Section 3. Then we use the class information in fine-tuning. Since there are four emotion classes in our task, we use a softmax layer on top of the hidden representation. Fine-tuning updates the model parameters with the goal of optimizing the classification accuracy. After pre-training and fine-tuning, we use the hidden representations \( y \) as the features for the emotion recognition task. A SVM classifier is trained using such features. For a test instance, the raw features are passed through the DAE and its hidden representation \( y \) is computed and used for classification.

It is easy to extend this DAE to deeper structure (i.e., stacked DAE). After pre-training the first layer, the hidden representation can be used as input of the next hidden layer. The training process described above can be applied layer by layer. After pre-training for all the layers, the softmax layer is put on the top of the hidden layer to predict the class distribution in fine-tuning stage.

2.2. Modified Autoencoder

To further remove redundant information and extract robust emotion related hidden feature representation, we propose to project the input to two hidden spaces. One of them is meant to represent the emotional information; whereas the other is used to capture redundant information, specifically the neural information that we hypothesize may be contained in all the emotion speech.

Figure 2 shows the structure of this modified DAE using one hidden layer.

The first step in our method is to train a parameter set, \( W_n, b_n \) and \( b'_n \), which can represent the neutral hidden representation. In this step, to avoid overfitting, instead of using the neutral instances directly from the training set (used for emotion recognition task), additional neutral based corpus is enrolled. We use the Wall Street Journal corpus. To learn the parameter set, traditional denoising autoencoder as described above is applied. Since all the instances are neutral speech, it is not necessary to add fine-tuning step after pre-training.

In the second step, the corrupted input \( \tilde{x} \) from the emotion corpus is projected to two hidden representations. The parameter set of the first projection is the one pre-trained in the first step, \( W_n, b_n \) and \( b'_n \). We expect that this projection can capture the neutral information of the emotional utterance. The parameter set, \( W_n, b_n \) and \( b'_n \), for the other projection is randomly initialized and will be updated during the pre-training process. The encoded neutral hidden representation \( y_n \) and the emotion hidden representation \( y_e \) are obtained as follows:

\[
y_n = s(W_n \tilde{x} + b_n)
\]

\[
y_e = s(W_e \tilde{x} + b_e)
\]

After that, the two hidden representations are decoded to make reconstructed input vector, \( z_n \) and \( z_e \) as follow:

\[
z_n = s(W'_n y_n + b'_n)
\]

\[
z_e = s(W'_e y_e + b'_e)
\]

where tight weights are used, \( W'_n = W_n^T \) and \( W'_e = W_e^T \).

In the last step, \( z_n \) and \( z_e \) are used to form the reconstructed \( z \):

\[
z = \alpha * z_e + (1 - \alpha) * z_n
\]
Note that here we use a linear weighted combination for the emotional speech. The assumption is that the emotional speech (or the features used to represent them) can be thought of as generated by adding some emotional component to neutral ones.

Similar to the traditional DAE, we use squared error between and the clean input and stochastic gradient decedent methods to minimize the cost. Since the parameter set, , and , is pre-trained on a large corpus, we fix these parameters and only update the others, , and . The motivation behind this is that the parameter set of emotion hidden representation is used to train hidden representation in order to capture emotional information to minimize the squared error between and input . After pre-training, we again use the class labels for fine-tuning of the parameters.

For emotion recognition, we use the emotional hidden representation as the features. These are expected to capture the most emotion indicative cues, excluding the neural information. Again, this modified DAE can also be extended to deeper structure with more hidden layers similar to the basic DAE above.

3. Experiments

3.1. Features

The features used as the input of the DAE are static features, which have been successfully applied to the emotion recognition task recently [12]. 1584 features used in the INTERSPEECH 2010 Paralinguistic Challenge [13] are extracted using openSMILE toolkit [14]. Table 1 shows these features. Details about these features can be found in [13]. Note that all the features are linearly normalized to the range of 0 to 1 when used as the input to DAE.

<table>
<thead>
<tr>
<th>Descriptors</th>
<th>Functionals</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM loudness</td>
<td>Position max./min.</td>
</tr>
<tr>
<td>MFCC [0-14]</td>
<td>arith. mean, std. deviation</td>
</tr>
<tr>
<td>log Mel Freq. Band [0-7]</td>
<td>skewness, kurtosis</td>
</tr>
<tr>
<td>LSP Frequency [0-7]</td>
<td>lin. regression coeff. 1/2</td>
</tr>
<tr>
<td>FO</td>
<td>Q/A</td>
</tr>
<tr>
<td>F0 Envelope</td>
<td>quartile 1/2/3</td>
</tr>
<tr>
<td>Voicing Prob.</td>
<td>quartile range 2-1/3-2/3-1</td>
</tr>
<tr>
<td>Jitter local</td>
<td>percentile 1/99</td>
</tr>
<tr>
<td>Jitter consec. frame pairs</td>
<td>percentile range 99-1</td>
</tr>
<tr>
<td>Shimmer local</td>
<td>up-level time 75/90</td>
</tr>
</tbody>
</table>

3.2. Data

Interactive Emotional Dyadic Motion Capture (USC-IEMOCAP) database [15] is used in this study. This corpus has approximately 12 hours of audiovisual data, including video, speech, motion capture of face, text transcriptions [15]. It has 10 professional actors (5 male and 5 female) acting in two different scenarios: scripted play and spontaneous dialog, in their dyadic interaction. Each interaction is around 5 minutes in length, and is segmented into sentences. These sentences are labeled by at least 3 annotators. We use four emotion categories in this study: angry, happy, sad and neutral. Note that we merged Happy and Excited in the original annotation into one class: happy. Only the utterances with the majority agreement are used in the experiments. There are 5,531 utterances in this data set in total. The class distribution is: 20% angry, 29.6% sad, 19.6% happy, and 30.8% neutral.

3.3. Experimental Setup

To pre-train the parameter set, , and , in the proposed modified DAE method, we use the Wall Street Journal corpus, containing about 78K utterances. We treated all of these as neutral speech. The learning rate is set to 0.01. We used 30 training epochs and the number of utterances in each minibatch is 1000. Then to pre-train the emotion representation parameter set, , , and , we use 15 iterations and the learning rate of 0.01. In the fine-tuning stage, we set 8 iterations with 0.002 as learning rate. The weight combination parameter for the two representations is 0.5. For all of the DAE models, we use a corruption level of 0.1 to generate the corrupted signal from the original raw features.

SVMs with RBF kernels are used as the classifier for the new feature representation. We use leave-one-speaker-out cross validation for the emotion recognition experiments. This allows us to better evaluate the system’s speaker independent performance. For each speaker, the training data is from other speakers.

3.4. Results

Table 2 shows the emotion classification results using the new features generated using DAE, traditional one and our proposed one with two projections. These results are the average accuracy for the ten test speakers. For the DAE models, we also varied the number of hidden nodes, using 200, 400, and 800 nodes respectively. For a comparison, the last row shows the performance using the original static feature set, which again represents the state-of-the-art performance. We can see from the table that when the size of the new feature representation based on DAE is proper (400), better performance can be achieved than the original static features. There is additional gain using our proposed DAE compared to the traditional DAE, suggesting that the two projections indeed allow the model to better capture the emotion specific information. Our method has about 3% accuracy increase over the static features, a significant improvement.

Table 2: Emotion classification results (accuracy in %) using new features generated using DAE.

<table>
<thead>
<tr>
<th>DAE hidden nodes</th>
<th>DAE proposed</th>
<th>DAE traditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>57.4</td>
<td>59.4</td>
</tr>
<tr>
<td>400</td>
<td>60.1</td>
<td>61.5</td>
</tr>
<tr>
<td>800</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 shows the accuracy for each emotion class using our proposed DAE, as well as using the static features for the entire data set. We can see that our method performs better for angry, happy and neutral categories, but degraded performance for sad. There are more sad instances that are predicted as neutral in our method. We hypothesize that this may be because in the new method, the added neutral hidden representation makes the sad instances more confused with neutral ones.

We also tried using deep structure for the proposed DAE by adding more hidden layers; however, we found that the performance improves for some test speakers, but decreases for others. On average, there is a slight performance degradation.
using more hidden layers compared to the results above using just one hidden layer. In addition, using deep structure also requires more training time. To better explore the potential power of deep structures, we may need to optimize parameters such like pre-training rate or the minibatch size for each speaker. We will continue to investigate this in the future work.

4. Conclusion and Future Work

In this paper, we evaluate using denoising autoencoder for emotion recognition. We introduced a modified version of denoising autoencoder for better feature representations. Raw feature inputs are projected into two hidden representations. One projection, with parameter set pre-trained on a large neutral corpus, aims to capture neutral information inside the emotional instances. The other one is designed to extract emotional information. The emotion hidden representations are used as input features for a back-end classifier. On a speaker independent evaluation, our experimental results show that our proposed method outperformed using the static features for emotion recognition.

For the future work, we plan to further improve various components in the current framework. First, in the reconstruction step, currently we use a linear combination of neutral and emotional representation. It is worthwhile to investigate other non-linear methods. Second, we will use other features besides static features, such as GMM supervectors.

5. Acknowledgment

This work is supported by U.S. Air Force Award FA9550-10-1-0388. Any opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

6. References


