Likability Classification - A not so Deep Neural Network Approach

Raymond Brueckner[1,2]*, Björn Schuller*[2]

[1]Institute for Human-Machine Communication, Technische Universität München, Germany
[2]Nuance Communications Inc., Aachen, Germany
raymond.brueckner@web.de, schuller@tum.de

Abstract

This papers presents results on the application of restricted Boltzmann machines (RBM) and deep belief networks (DBN) on the Likability Sub-Challenge of the Interspeech 2012 Speaker Trait Challenge [1]. RBMs are a particular form of log-linear Markov Random Fields and generative models which try to model the probability distribution of the underlying input data which can be trained in an unsupervised fashion. DBNs can be constructed by stacking RBMs and are known to yield an increasingly complex representation of the input data as the number of layers increases. Our results show that the Likability Sub-Challenge classification task does not benefit from the modeling power of DBN, but that the use of an RBM as the first stage of a two-layer neural network with subsequent fine-tuning improves the baseline result of 59.0 % to 64.0 %, i.e., a relative 8.5 % improvement of the unweighted average evaluation measure.

Index Terms: Likability, speaker trait challenge, restricted Boltzmann machines, deep belief networks

1. Introduction

The recent surge of interest in deep belief networks (DBN) and their constituent components have led to remarkable advances in many machine learning and pattern recognition problems and significant improvements have been reported lately in the speech recognition community [2] [3]. These networks have also been successfully applied to emotion recognition [4].

In the Likability Sub-Challenge of the INTERSPEECH 2012 Speaker Trait Challenge [1], the task is to automatically predict the likability of a users voice from the speech signal applying a pre-defined acoustic feature set. This might be of interest in various applications in human-machine and human-human communication, such as voice portals or social networks.

We have investigated the applicability of deep neural networks and Restricted Boltzmann machines (RBM) on this task and we will show that indeed the baseline approaches using Support Vector Machines and Random Forests can be outperformed.

The authors of the present paper are all affiliated with and partly identical with the organizers of the Challenge. Therefore, we do not participate in the Challenge. To ensure comparability of the results, we strictly follow the procedures defined in [1] and we neither use any data or information that were not available to all competitors nor more result trials on the test data.

The structure of this paper is as follows: in section 2 we will describe RBMs for binary input and a variant thereof, the Gaussian-Bernoulli RBM (GBRBM) for real-valued input data. We will further show how deep belief networks (DBN) can be constructed from these RBMs and how the resulting deep networks can be further trained to improve their discriminative performance. In section 3 we will present the experimental setup and the results obtiained on the Likability Sub-Challenge. Finally, in section 4 conclusions will be drawn.

2. Deep Belief Networks

The idea of using deep multilayer neural networks is not new, but traditionally it has been difficult to train these models successfully: with large initial weights, the typically adopted back-propagation algorithm converges towards poor local minima; with small initial weights, on the other hand, the gradients in the lower layers become tiny making it infeasible to train networks with many hidden layers. Furthermore, if there are many hidden layers in the neural network with many hidden units in each layer, it is easy for the network to overfit.

In order to overcome these problems DBNs have been proposed. Deep belief networks are probabilistic generative models that are composed of multiple layers of stochastic, latent variables or hidden units, which typically have binary values. In generative models the goal is to learn the distribution of the input data $p(data)$, instead of $p(labels|data)$, as is common in training a discriminative model.

Hinton et al [5] proposed an efficient, greedy algorithm that allows to learn one layer at a time in an unsupervised fashion using an undirected graphical model called a RBM.

2.1. Restricted Boltzmann Machine

RBMs are the building blocks of DBNs, and are undirected graphical models with a layer of observed, or visible, variables and a layer of latent or hidden variables, with each layer forming one part of a bipartite graph; i.e., each visible unit (node) is connected to each hidden unit, but there are no intra-visible or intra-hidden connections. The graph of a RBM is depicted in Fig. 1.

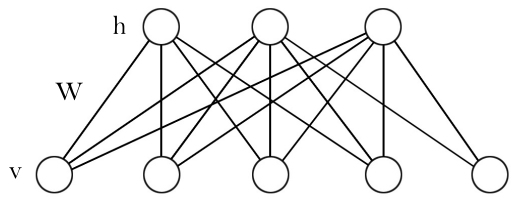

Figure 1: *Restricted Boltzmann Machine graph.*

A RBM assigns an energy to every configuration of visible and hidden state vectors, denoted v and h respectively. For binary visible units, an RBM with V visible units and H hidden units is governed by the following energy function:

$$E(v,h) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ij} - \sum_{i=1}^{V} v_i b_i^v - \sum_{j=1}^{H} h_j b_j^h \quad (1)$$

where v_i and h_j are the binary states of visible unit i and hidden unit j, b_i^v and b_j^h are their biases, and w_{ij} is the weight between them.

Under the this energy function, the conditional probabilities for each visible and hidden unit given the others are

$$p(h_j = 1|\mathbf{v}) = \mathbf{g}\left(\mathbf{b_j^h} + \sum_{\mathbf{i}} \mathbf{v_i w_{ij}}\right) \quad (2)$$

$$p(v_i = 1|\mathbf{h}) = \mathbf{g}\left(\mathbf{b_i^v} + \sum_{\mathbf{j}} \mathbf{h_j w_{ij}}\right) \quad (3)$$

where

$$g(x) = \frac{1}{1+e^{-x}} \quad (4)$$

is the logistic or sigmoid function.

The network assigns a probability to every possible joint configuration (v, h) via the energy function as:

$$p(v,h) = \frac{e^{-E(v,h)}}{Z} = \frac{e^{-E(v,h)}}{\sum_{u,g} e^{-E(u,g)}} \quad (5)$$

where Z is called the partition function. The marginal distribution of the visible units is then given as

$$p(v) = \sum_{h} p(v,h) \quad (6)$$

and the gradient of the average log-likelihood is

$$\frac{\partial \log p(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_\infty \quad (7)$$

The average $\langle\cdot\rangle_0$ can be readily computed using the sample data v, but the average $\langle\cdot\rangle_\infty$ involves the normalisation constant Z, which cannot generally be computed efficiently (being a sum of an exponential number of terms).

To avoid the difficulty in computing the log-likelihood gradient, Hinton [6] proposed the *Contrastive Divergence* (CD) algorithm which approximately follows the gradient of the difference of two divergences:

$$\frac{\partial \log p(v)}{\partial w_{ij}} \approx \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_k \quad (8)$$

The expectation $\langle\cdot\rangle_k$ represents a distribution from running a Gibbs sampler (eqs. 2, 3) initialized at the data for k full steps. This process is shown in Fig. 2. In practice, we typically choose $k = 1$. This is a rather crude approximation of the true log maximum likelihood gradient, but it works well in practice.

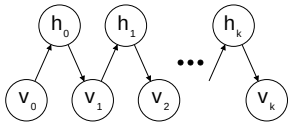

Figure 2: *Illustration of k-step Gibbs sampling.*

2.2. Gaussian-Bernoulli Restricted Boltzmann Machine

To deal with real-valued input data, we use an RBM with Gaussian visible units and binary hidden units yielding a so-called GBRBM, where we use the modified energy function proposed in [7]:

$$E(v,h) = \sum_{i=1}^{V} \frac{(v_i - b_i^v)^2}{2\sigma_i^2} - \sum_{i=1}^{V}\sum_{j=1}^{H} \frac{v_i}{\sigma_i^2} h_j w_{ij} - \sum_{j=1}^{H} h_j b_j^h \quad (9)$$

Under the modified energy function, the conditional probabilities for each visible and hidden unit given the others are

$$p(v_i = v|\mathbf{h}) = \mathcal{N}\left(v|b_i^v + \sum_j h_j w_{ij}, \sigma_i^2\right) \quad (10)$$

$$p(h_j = 1|\mathbf{v}) = g\left(b_j^h + \sum_i \frac{v_i}{\sigma^2} w_{ij}\right) \quad (11)$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ denotes the Gaussian probability density function with mean μ and variance σ.

In our approach we also learn the parameter σ. Note, however, that this σ^2 is not necessarily equivalent to the variance of the input data.

2.3. Constructing a Deep Belief Network

Once the weights of an RBM have been learned, the outputs of the hidden nodes can be used as input data for training another RBM that learns a more complex representation of the input data. Proceeding this way a deep belief network (DBN) can be constructed stacking one RBM on top of the preceding layer. Building a deep generative model one layer at a time is much more efficient than trying to learn all of the layers at once.

The parameters of this stacked network model are often close to an optimum and hence gradient descent techniques can be used to fine-tune the DBN. The limited information contained in the labels is then used to only slightly adjust the pretrained weights in order to improve the discriminative power of the generative model.

One important property of DBNs is that their hidden states can be inferred very efficiently by a single bottom-up pass in which the top-down generative weights are used in the reverse direction. Another important property is that each time an extra layer of learned features is added to a DBN, the new DBN has a variational lower bound on the log probability of the training data that is better than the variational bound for the previous DBN, provided the extra layer is learned in the right way [5]. The weights and biases of a DBN can be used to initialize the hidden layers of a feedforward neural network which is given an additional output layer. For a wide variety of tasks, discriminative fine-tuning of a DBN-initialized neural network gives much better performance than the same neural network initialized with small random weights [8]. Many of the generatively

learned features may be irrelevant for the discrimination task, but those that are relevant are usually much more useful than the input features because they capture the complex higher-order statistical structure that is present in the input data.

It has been shown in [8] that greedy layer-wise unsupervised pre-training is crucial in deep learning by introducing a useful prior to the supervised fine-tuning training procedure. The regularization effect is claimed to be a consequence of the pre-training procedure establishing an initialization point of the fine-tuning procedure inside a region of parameter space in which the parameters are henceforth restricted. Furthermore, overfitting can be substantially reduced if a generative model is used to find sensible features without making any use of the labels.

3. Experiments and Results

The results presented in this section were obtained by carrying out experiments on the *Likability Sub-Challenge* of the Interspeech 2012 Speaker Trait Challenge, which uses the "Speaker Likability Database" [9]. The speech is recorded over fixed and mobile telephone lines at a sample rate of 8 kHz. The feature set contains 6125 features based on low-level descriptors and functionals applied thereon. For details about the Challenge and the underlying feature set refer to [1].

3.1. Feature Preprocessing

We trained and evaluted the networks on the complete Challenge feature set comprising all the 6125 acoustic features. As the feature set exposes an extremely high variance and as RBMs and other neural network architectures are not scale-invariant, all input features were normalized to zero mean and unit variance which often is beneficial or even necessary to obtain good results with neural networks [10]. Means and standard deviations for normalization were computed from the training set. It is interesting to note that the feature set was approximately Gaussian distributed after this preprocessing. This turns out to be beneficial for using GRBMs, as these models work very well in this case.

3.2. Training Setup

3.2.1. Restricted Boltzmann Machine

Training RBMs, although conceptually simple, requires many parameters to be adjusted carefully in order to be successful; we used the suggestions in [11] as a starting point. We applied stochastic gradient descent (SGD) run on mini-batches. The minibatch size trades off noisy gradients with slow convergence. We obtained best results with a mini-batch size of 20. Gradient descent in mini-batches works best if the data is presented in random order, hence we shuffled the input data accordingly.

We trained each RBM for 50 epochs using contrastive divergence with one Gibbs step per mini-batch. For the pre-training, we used a learning rate of 10^{-3} for the weights and biases and a learning rate of 10^{-6} for the σ parameter of the GBRBM. Often a momentum term is used to smooth the gradients, but it was not helpful for pre-training the RBMs, so we set it to 0. As a further regularization we applied a small weightcost parameter. This prevents the weights becoming too big, which typically leads to bad generalization performance.

We also found out that the performance of the RBM in the first layer can be improved by moderate sparsification of the weight matrices. Enforcing weight sparseness is equivalent to incorporating L0 and approximate L1 regularizations to the training criterion. We reached an optimum in performance for a sparsity threshold of 0.08 and after each mini-batch simply set all weights assuming a value below this threshold to 0.

As the first layer of our final network we used a GRBM which is a good model of the approximately Gaussian distributed input data (after preprocessing). Any subsequent RBM layer was then a standard RBM with binary visible and binary hidden units. The training parameters were the same for both type of models, except that in the binary-binary RBM obviously there is no σ parameter to be modeled.

3.2.2. Full Network

Following section 2.3, a deep belief network was constructed by stacking the pre-trained RBMs and adding an additional layer on top. For the Likability Sub-Challenge we have binary target labels (likable / not likable) and thus for the top-layer we used a logistic regression network with one output node.

For the fine-tuning phase, the full network was then trained using SGD based on a crossentropy cost function. Even though SGD is known to be a rather limited approximator with respect to the full batch gradient descent we obtained best results for small batch sizes of 2, with the input data being shuffled randomly.

Network training was run until the unweighted accuracy on the development set reached a minimum. This is referred to as *early stopping* in the literature [10]. Several parameter combinations have been tried out, but we obtained the best results using a learning rate of 0.05, no momentum term, and a sparsity threshold of 0.

To enforce additional regularization, we constrained the square of the L2 norm to be small using a regularization coefficient of 10^{-4}.

3.3. Results

Table 1 shows the results obtained on the Likability Sub-Challenge when applying the different neural network architectures. Preliminary experiments have shown that for the task at hand a constant layer size of 2048 proved to be most effective.

The reported evaluation measure is the unweighted accuracy (UA). In the given case of two classes (L and NL), it is calculated as (Recall(L)+Recall(NL))/2, i. e., the number of instances per class is ignored by intention, because the UA is also meaningful for highly unbalanced distributions of instances among classes.

The results show that this classification task does not benefit from the modeling power of DBNs, but that the use of a GBRBM as the first stage of a two-layer neural network with subsequent fine-tuning improves the baseline result of 59.0 % UA to 64.0 % UA, i.e., a relative 8.5 % improvement. As more layers are added, the performance decreases, and a two-layer DBN already has lower performance than a standard one-hidden layer multi-layer perceptron (MLP).

We also tried architecture topologies with decreasing layer sizes which are equivalent to the well-known bottleneck networks. Such models have been successfully applied in previous challenges, for example on tasks of the Interspeech 2010 Paralinguistic Challenge [12]. A fundamental difference is that the approach to create classic bottleneck networks (such as e.g. autoencoder networks) is to decrease the layer sizes towards the innermost hidden layer and to increase them towards the top layer, then to train the network, and finally to take the output of

Table 1: *Results on the Likability Sub-Challenge. MLP refers to a standard one-hidden layer neural network with random initialization of its weight parameters. Test results are reported for the experiments that were submitted to the Challenge site and which were returned by the Challenge organizers. Participants were allowed to submit only five uploads of their predictions on unlabeled test data.*

	UA Test / UA Devel
Baseline (random forests) [1]	59.0 / 57.6
MLP (random initialization)	60.9 / 56.4
DBN (1 layer)	64.0 / 57.2
DBN (2 layers)	62.9 / 56.2
DBN (3 layers)	62.2 / 56.0
DBN (4 layers)	– / 54.1

the smallest, innermost layer, hence discarding the upper layers. Here we construct bottleneck networks by stacking RBMs trained with the contrastive divergence algorithm and with decreasing layer size towards the output layer. The results obtained on various bottleneck topologies are depicted inTable 2.

Table 2: *Results on the Likability Sub-Challenge for bottleneck architectures. Test results are reported for the experiments that were submitted to the Challenge site and which were returned by the Challenge organizers.*

	UA Test / UA Devel
Baseline (random forests) [1]	59.0 / 57.6
DBN (6125-2048-1024-256)	60.2 / 60.3
DBN (6125-2048-1024-256-32)	– / 59.1
DBN (6125-1024-256-32)	– / 53.6
DBN (6125-1024-256-32-8)	– / 56.2

Even though bottleneck architectures have been used successfully on many different tasks, the results show that the Likability Sub-Challenge task does not benefit from their application. The first two bottleneck topologies actually beat the baseline results on the development set; a result which is confirmed on the test set for the first bottleneck architecture. Nonetheless, the test result on the first topology still is considerably below the best test result obtained with a one-layer DBN.

4. Conclusion

We investigated the applicability of deep belief networks on the Likability Sub-Challenge of the Interspeech 2012 Speaker Trait Challenge. For this particular task we were not able to leverage the power of DBNs to model the complex probability distribution of the supplied full feature set. Neither bottleneck topologies nor the standard topology of stacking equally sized layers were as efficient as a standard MLP whose first layer consists of a GBRBM trained in an completely unsupervised manner. With this architecture we were able to improve the baseline results of 59.0 % UA to 64.0 % UA which constitutes a relative improvement of 8.5 % UA. In our opinion this clearly shows the potential that lies in unsupervised methods for the paralinguistic domain.

With respect to further optimization of our system, we think about considering second-order optimization methods for the fine-tuning stage of the neural network training such as Conjugate Gradient Descent or Quasi-Newton methods, which are known to be more efficient on small-scale problems. Furthermore, given the approximately Gaussian distribution of the mean- and variance-normalized feature set it might be interesting to compare GBRBMs to radial basis funcions in the first hidden layer. More importantly, we plan to examine more thoroughly the potential of semi-supervised techniques in the context of emotion and speaker trait recognition.

5. Acknowledgements

This work was conducted while the first author was working at Nuance Communications Inc., Aachen, Germany.

6. References

[1] B. Schuller, S. Steidl, A. Batliner, E. Nöth, A. Vinciarelli, F. Burkhardt, R. van Son, F. Weninger, F. Eyben, T. Bocklet, G. Mohammadi, and B. Weiss, "The Interspeech 2012 Speaker Trait Challenge," in *Proceedings of the Interspeech 2012*. Portland, OR, USA: ISCA, Sep 2012.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Large-Vocabulary Continuous Speech Recognition with Context-Dependent DBN-HMMs," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May 2011, pp. 4688–4691.

[3] M. Abdel-Rahman, G. Dahl, and G. E. Hinton, "Acoustic Modeling using Deep Belief Networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[4] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller, "Deep Neural Networks for Acoustic Emotion Recognition: Raising the Benchmarks," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP) 2011*. Prague, Czech Republic: IEEE, May 2011, pp. 5688–5691.

[5] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul 2006.

[6] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, Aug 2002.

[7] K. Cho, A. Ilin, and T. Raiko, "Improved Learning of Gaussian-Bernoulli Restricted Boltzmann Machines," in *Proceedings of the International Conference on Artificial Neural Networks*, Espoo, Finland, Jun 2011, pp. 10–17.

[8] D. Erhan, Y. Bengio, A. Courville, P.-A. M. P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?" *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, Mar 2010.

[9] F. Burkhardt, B. Schuller, B. Weiss, and F. Weninger, "'Would You Buy A Car From Me?' - On the Likability of Telephone Voices," in *Proceedings of the Interspeech 2011*. Florence, Italy: ISCA, Aug 2011, pp. 1557–1560.

[10] Y. LeCun, L. Bottou, G. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the trade*, G. Orr and K.-R. Müller, Eds. Springer, 1998, ch. 2, pp. 9–50.

[11] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," Machine Learning Group, University of Toronto, Technical Report 2010-003, 2010.

[12] M. Wöllmer, F. Weninger, F. Eyben, and B. Schuller, "Acoustic-Linguistic Recognition of Interest in Speech with Bottleneck-BLSTM Nets," in *Proceedings of the Interspeech 2011*. Florence, Italy: ISCA, Aug 2011, pp. 77–80.