Towards Deep Learning for Dialogue State Tracking Using Restricted Bolzman Machines and Pretraining

Callum Main, Zhuoran Wang, and Verena Rieser
Interaction Lab
Heriot-Watt University
Edinburgh, UK
www.macs.hw.ac.uk/InteractionLab

Abstract

Dialogue state tracking aims to estimate the user's goal over the course of a dialogue. Recently, deep neural networks have shown to be successful in this task, especially for generalising to unseen states. In this research, we investigate an alternative deep learning framework, using Restricted Bolzman Machines with pretraining. We aim to show that, by adding a pre-training phase which allows to initialise learning from unlabelled data, leads to significant improvements in terms of accuracy over a baseline using Deep Neural Networks with shared initialisation.

1 Introduction

Statistical spoken dialogue systems maintain a distribution over possible dialogue states ("belief state") in order to correctly estimate the user's true goal, while communicating with a user, from a noisy and often ambiguous input signal. This process is called *dialogue state tracking* (DST).

Deep neural networks (DNN) have shown to be successful in capturing error correlations for improving Automatic Speech Recognition (ASR) systems, e.g. (Deng et al., 2013), and have recently shown successful for dialogue state tracking (Henderson et al., 2013; Henderson et al., 2014). In this research we extend this previous work by (Henderson et al., 2013) in using Restricted Bolzman Machines (RBMs) with pretraining for initialisation (Hinton and Osindero, 2006), which allows us to utilise an additional data set of 10,619 unlabelled calls to capture correlated hypotheses.

2 Related Work

A recent paper by (Henderson et al., 2013) describes a DNN approach to DST using a simple feed-forward architecture with three layers, see Figure 1. The input consists of feature functions, $f_i(t, v)$ which extract information related to the SLU hypotheses, as well as the machine acts at a particularly turn t. The input layer then consists of the feature functions being summed for turns $t-T$ where T is the window size that has been chosen. This input layer is then combined with three hidden layers, each consisting of a weight matrix $\mathbf{W}_i$ and a bias vector $\mathbf{b}_i$, these layers are then reduced to a single node $E(t, v)$. The overall distribution of the tracker is then given by:

$$P(s = v) = e^{E(t,v)}/Z;$$
$$P(s \notin S_{t,s}) = e^{B}/Z;$$
$$Z = e^{B} + \sum_{v' \in S_{t,s}} e^{E(t,v')};$$

The model was then trained using Stochastic Gradient Descent (with mini-batches to speed up computation) using three different initialisation schemes: training a model for each slot, training a model independent of slot, and training a slot independent model for a few epochs before switching to separate models for each slot. An experimental evaluation revealed that shared initialisation, where a single model is trained first before training separate models for each slot, performed the best. This result leads to using more efficient deep learning techniques such as stacked RBMs using pre-training (Hinton and Osindero, 2006) as a promising direction for research.

3 RBM with Pretraining

Training DNNs is very computationally expensive. As such, recent work uses more efficient models such as deep belief networks (DBNs). A DBN can be efficiently trained in an unsupervised, layer-by-layer manner where the layers are typically made of restricted Boltzmann machines (RBMs). RBMs are stochastic generative artificial neural networks which learn a relationship between a set of visible input units and hidden units

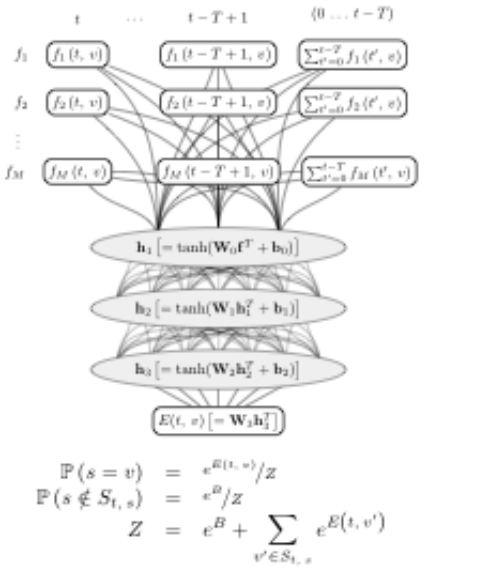

Figure 1: Neural Network structure for computing $E(t, v)$ from (Henderson et al., 2013)

in the form of weighted connections. We can then use an energy based probability model to define a probably distribution that is given by:

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_i h_j - \sum_{i,j} v_i h_j w_{ij};$$

where v_i, v_j are the states of the hidden unit i and hidden unit j, a_i, b_j are their biases and w_{ij} is the difference between them. The probability for each possible pair of hidden and visable units is then given by:

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

Where Z is the partition function given by summing over all of the possible visable-hidden pairs:

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})};$$

We can then find the probability of assigning to a visible vector by computing the marginal probability by summing over all of the possible hidden vectors:

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})};$$

The derivative of the log likelihood with respect to can then be written as:

$$\frac{\partial P(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model};$$

(Hinton and Osindero, 2006) discovered that RBMs can be trained layer-wise using unsupervised pretraining and then stacked together to great a deep neural network. The pretraining algorithm is then:
(1) Train an RBM using the input X as the visible layer to be used as the first layer.
(2) Transform X using the first layer to obtain data for second layer by either sampling or computing mean activation of hidden units.
(3) Repeat steps 2 and 3 for the desired number of layers.

4 Data

We train and evaluate our approach on data available as part of the first Dialogue State Tracking Challenge (DSTC1) (Williams et al., 2013), see table 4. Results from DSTC1 show that Henderson et al.'s (2013) model outperforms most other models on test set 4, showing its ability to generalise to unseen states. We aim to improve over these results by taking advantage of training set 1b and 1 c, which contain 10,619 unlabelled calls for pretraining.

Set	**no. calls**	**Notes**
train 1a	1013	Labelled training data.
train1b&c	10619	Some SDS as train 1a, but **unlabelled**.
train2	678	Similar to train1.
train3	779	Different SDS to other data sets.
test1	765	Very similar to train1 and train2.
test2	983	Somewhat similar to train1 and train2.
test3	1037	Very similar to train3.
test4	451	unseen Spoken Dialogue Systems.

Table 1: Data released through DSTC1 (Williams et al., 2013)

5 Summary and Future Work

This paper describes work in progress towards a more efficient model for training neural networks for dialogue state tracking using Restricted Bolzman Machines with pretraining, following (Hinton and Osindero, 2006). We compare this model against Deep Neural Networks with shared initialisation as proposed by (Henderson et al., 2013). Full results will be presented at the poster session of SemDial 2014. Having a more efficient way of training while making use of unlabelled data, will allow us to investigate more complex models, for example to directly estimate the dialogue state based on ASR output rather than SLU hypothesis (Henderson et al., 2014).

References

Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero. 2013. Recent advances in deep learning for speech research at microsoft. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

M. Henderson, B. Thomson, and S. J. Young. 2013. Deep Neural Network Approach for the Dialog State Tracking Challenge. In *Proc. of SIGdial Workshop on Discourse and Dialogue*.

M. Henderson, B. Thomson, and S. J. Young. 2014. Word-based State Tracking with Recurrent Neural Networks. In *Proceedings of SIGdial*.

Geoffrey E. Hinton and Simon Osindero. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006.

Angeliki Metallinou, Dan Bohus, and Jason D. Williams. 2013. Discriminative state tracking for spoken dialog systems. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.