Sparse Autoencoder Based Semi-Supervised Learning for Phone Classification with Limited Annotations

Akash Kumar Dhaka and Giampiero Salvi

KTH Royal Institute of Technology,
School of Computer Science and Communication,
Dept. for Speech, Music and Hearing, Stockholm, Sweden

{akashd, giampi}@kth.se

Abstract

We propose the application of a semi-supervised learning method to improve the performance of acoustic modelling for automatic speech recognition with limited linguistically annotated material. Our method combines sparse autoencoders with feed-forward networks, thus taking advantage of both unlabelled and labelled data simultaneously through mini-batch stochastic gradient descent. We tested the method with varying proportions of labelled vs unlabelled observations in frame-based phoneme classification on the TIMIT database. Our experiments show that the method outperforms standard supervised models of similar complexity for an equal amount of labelled data and provides competitive error rates compared to state-of-the-art graph-based semi-supervised learning techniques.

Index Terms: automatic speech recognition, deep learning, semi-supervised learning, autoencoders, sparsity

1. Introduction

Deep Learning has revolutionised research in Automatic Speech Recognition (ASR) as well as many other fields of application of machine learning (see [1, 2] for extensive reviews). Despite, the recent significant improvements made in word error rates (WERs), most of the experiments have been reported on large fully-labelled data sets. The initial paradigm, where unsupervised initialisation of the network weights was followed by supervised fine-tuning of the parameters [3, 4], was abandoned in favour of fully supervised methods with more efficient models (e.g. [5]). However, for under-resources languages, where large amounts of labelled data are not available, non fully supervised learning techniques are still relevant.

Unsupervised learning in the context of ASR has been an active topic for many years in the attempt to reduce the amount of handcrafted information needed to build the systems. Many of the studies focus on finding linguistic information from speech in a completely unsupervised way (e.g. [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17], resulting in a recent attempt to standardise this effort in a challenge [18].

In the context of deep neural networks, unsupervised learning has been mainly applied to the task of initialising the network weight of a model that was, otherwise, fully specified. The limit of this approach was that the resulting initial weights are not specifically optimised for the problem at hand. As an example, we would find the same representations for speech or speaker recognition which have orthogonal objectives.

An alternative learning paradigm, that has recently been applied in the field of computer vision as well as ASR, is semi-supervised learning where labelled and unlabelled observations are used jointly [19, 20, 21, 22]. Here the goal is not to discover linguistic information in an unsupervised way, but, rather, to reduce the need for annotated material.

Semi-supervised learning using neural networks has also been explored in [23, 24, 25, 26], by means of a self-training scheme. The self-training scheme is, however, based on heuristics and prone to reinforcing poor predictions.

In [20, 21], the authors propose a number of algorithms employing graph based learning (GBL-SSL), and obtain better WERs over a baseline neural network. In [22] the authors extend the initial results from frame based phone classification to large vocabulary ASR. Graph based learning is, however, computationally intensive, and the addition of a new data point requires the reevaluation of the graph laplacian.

In [27], Ranzato and Szummer propose a semi-supervised learning method based on linearly combining the supervised cost function of a deep classifier with the unsupervised cost function of a deep autoencoder and minimising the combination of costs through mini-batch stochastic gradient descent via standard backpropagation. The authors apply their method to finding representations of text documents for information retrieval and classification.

We propose to use a similar approach to frame-based phone recognition in ASR. Although our objective function is the same as the one proposed in [27], our setup is different in a number of ways. Firstly, instead of the compact and lower dimensional encoding used in [27], we employ sparse encoding. Secondly, instead of stacking a number of encoders, decoders and classifiers in a deep architecture as in [27], we use a single layer model. This is motivated by work in [19], where the authors analyse the effect of several model parameters in unsupervised learning of neural networks on computer vision benchmark data sets such as CIFAR-10 and NORB. They conclude that state-of-the-art results can be achieved with single layer networks regardless of the learning method, if an optimal model setup is chosen.

The method bears some similarities to the so called Ladder Networks, introduced in [28] and used in the context of semi-supervised learning in [29]. However, our method explicitly optimises the combined supervised and unsupervised criteria for each layer of the network independently, whereas, in Ladder Networks the optimisation is performed for all layers conjunctively. This makes Ladder Networks much more complex to optimise, albeit very powerful models. Finally, Rasmus *et al.* in [29] use batch-normalisation which was not tested in this study.

In order to test the method, we perform phone recognition on the TIMIT data set and compare the performance of our model with the results obtained with standard supervised learning models of similar complexity. We also compare our results with the computationally more expensive GBL methods that are

10.21437/GLU.2017-5

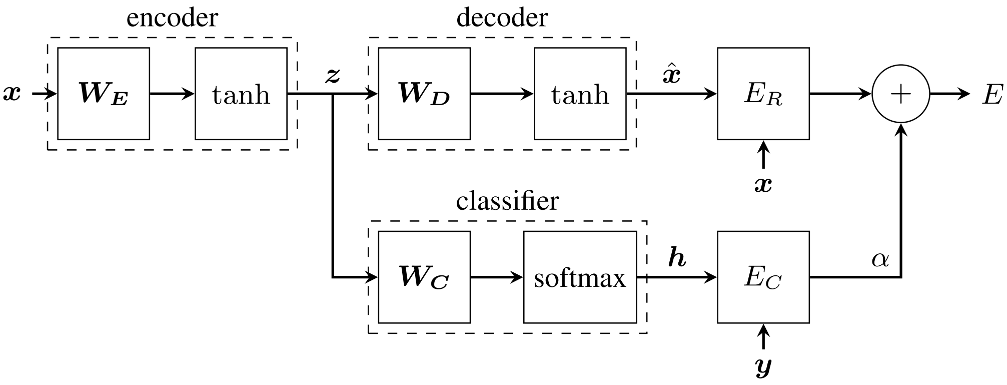

Figure 1: *Flow chart for the cost calculation in a single layer of the network. Three components are considered: encoder, decoder, and classifier. The loss is a weighted sum of cross-entropy E_C and reconstruction loss E_R. If several layers are stacked together, only the encoder is retained after training.*

state-of-the-art in semi-supervised learning.

Note that the goal of this work is not to improve the absolute state-of-the-art of phone recognition, but to provide alternative methods for reducing the amount of hand-crafted annotations that are required to build speech recognisers. However, the current state-of-the-art results in phone classification are also reported for reference.

The paper is organised as follows: Section 2 describes the method. Section 3 reports details on the experimental setup. Section 4 reports the results and, finally, Section 5 concludes the paper.

2. Method

The architecture of a single layer of our model is depicted in Figure 1. If we remove the bottom path, this is equivalent to an autoencoder with a set of encoding weights, a logistic layer, a subsequent set of decoding weights and a new non-linearity. In our model, the representation z obtained by the encoder is also fed to a classifier in parallel with the regular decoder. The aim of combining unsupervised and supervised cost functions is to use both the unlabelled and labelled data in an efficient way in order to obtain good representations of the input as well as good prediction and discriminative abilities from our network.

Although the figure depicts a single layer, in [27] it was shown that a stack of such elements can be trained layer-by-layer in a greedy way. In our experiments, only single layer models were considered.

The model is trained optimising the combined cost of the reconstruction error E_R and the classification errors E_C given respectively by the autoencoder and the classification network. The combination is linear and defined as:

$$E = E_R + \alpha E_C \tag{1}$$

where α is a hyper-parameter controlling the proportion of the two costs in the objective function. α is optimised on a validation set that is independent from the training set. Its optimal value depends in general on the proportion between labelled and unlabelled examples in the training set, as will also be shown in Section 4.

In the supervised setting, the cost function is the cross-entropy logloss given by:

$$E_C = -\sum_{i=1}^{N_C} y_i \log h_i \tag{2}$$

$$h_j = \frac{\exp((\boldsymbol{W_C})_j \boldsymbol{z} + b_{Cj})}{\sum_i \exp((\boldsymbol{W_C})_i \boldsymbol{z} + b_{Ci})}, \tag{3}$$

where, for the classification network: $\boldsymbol{h}$ denotes softmax output, $(\boldsymbol{W_C})_j$ is the jth row of the weight matrix $\boldsymbol{W_C}$, $\boldsymbol{b_C}$ is the set of biases and N_C is the number of output classes.

The variable $\boldsymbol{z}$ denotes the output of the encoder network and is defined as:

$$\boldsymbol{z} = \tanh(\boldsymbol{W_E}\boldsymbol{x} + \boldsymbol{b_E}), \tag{4}$$

Where $\boldsymbol{W_E}$ and $\boldsymbol{b_E}$ are the weights and biases of the encoder network, and $\boldsymbol{x}$ is the input to the entire model.

In the unsupervised path through the model, the decoder attempts to reconstruct the input $\boldsymbol{x}$ from the encoded vector $\boldsymbol{z}$ using the set of weights $\boldsymbol{W_D}$ and set of biases $\boldsymbol{b_D}$. The decoder output $\hat{\boldsymbol{x}}$ is computed as

$$\hat{\boldsymbol{x}} = \tanh(\boldsymbol{W_D}\boldsymbol{z} + \boldsymbol{b_D}). \tag{5}$$

The cost function, in this case, is the L_2 norm of the reconstruction error, that is, the difference between original input $\boldsymbol{x}$ and the reconstructed input $\hat{\boldsymbol{x}}$:

$$E_R = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2. \tag{6}$$

This a standard cost function for an auto-encoder. In practice, we compute the cost E_R averaged over a batch of p points, through an optimisation called mini-batch Stochastic Gradient Descent (SGD). We apply a process called "corruption", that is we randomly set to zero some elements of the input vector. This has been found to help the network learn better representations of the input data [30].

When the input datapoint is not accompanied by a label, the classifier part of the layer is not updated, and the loss function simply reduces to E_R. This model can be iteratively applied to several layers. It is important to note that the update of encoder weights $\boldsymbol{W_E}$ is dependent both on the decoder weights $\boldsymbol{W_D}$ and on the classifier weights $\boldsymbol{W_C}$, and the delta propagated in the backpropagation algorithm will be a linear combination of

Table 1: *Results on frame-based phone classification on the test and validation sets on the TIMIT material. Our method (SSSAE) is compared to a neural network trained with supervised backpropagation with the same amount of labelled data. The total number of training frames is 1068818. The value of α is optimised on the validation set as the proportion of labelled examples is varied.*

Results on TIMIT						
Labelled Observations		Neural Network		SSSAE		
%	#	valid. acc. (%)	test acc. (%)	valid acc. (%)	test acc. (%)	α
1	10688	57.46	57.93	59.65	59.84	100
3	32065	61.71	61.31	64.12	64.20	150
5	53441	63.20	63.30	65.44	65.71	150
10	106881	65.78	65.82	66.96	67.03	400
20	213763	68.02	67.80	69.31	69.18	600
30	320644	69.08	68.83	69.80	69.65	900

the deltas calculated in both parts. We used an adaptive learning rate scheme in which the learning rate decays linearly after a certain number of epochs.

The size of the hidden representation $\boldsymbol{z}$ is larger than the input size in our experiments. Consequently, we promote sparsity in our feature representation. In autoencoders, encoding and decoding weights are often tied, which means that the decoder weight matrix is the transpose of the encoder weight matrix: $\boldsymbol{W_D} = \boldsymbol{W_E}'$. This reduces the amount of free parameters available, but also the expressive power of the model. In our experiments, instead, we optimise $\boldsymbol{W_D}$ and $\boldsymbol{W_E}$ independently. This makes our model more expressive at the cost of more computational overhead and possible delayed convergence. Another aspect that increases the computational cost of our model is the use of sparse autoencoders as opposed to autoencoders with bottleneck architecture which have fewer nodes in hidden layer and, consequently, reduced memory and computational complexity. However, the computational cost is linear in the number of training samples, and thus it is more efficient than graph based semi-supervised learning algorithms which have cubic complexity $O(N^3)$ where N is the number of data points.

3. Experiments

3.1. Experimental Setup

We performed our experiments on the standard TIMIT data set [31] for frame-based phone classification. We used the standard core test set of 192 sentences, and a development/validation set of 184 sentences. For training, we had 3512 sentences. Similarly as a part of standard procedure of experiments on TIMIT, glottal stop segments are excluded. The data is created with the help of standard recipes given in [32, 33]. The input to our network was created by first extracting a 39-dimensional feature vectors for each frame. The feature vector is made of 12 MFCC coefficients computed at a rate of 10 ms with an overlapping window of 20 ms, 1 energy coefficient, deltas and delta-deltas. For each time step, the features obtained 5 frames to the left to 5 frames to the right are concatenated together to form a final vector with dimensionality of $11 \times 39 = 429$ coefficients as in [34]. Speaker-dependent mean and variance normalisation was also applied.

The total number of frames in the training set is 1068816. The validation set has 56005 frames in total, and the test set has 57919 frames. These counts are in line with the experiments of [20, 35]. For training, we used the standard phone set of 48 phones, collapsed into 39 phones for evaluation as in [36]. This means, the output layer will have 48 nodes, but at the time of evaluation, the 48 phonemes will be reduced to 39 phonemes. This procedure has also been used in [20]. Although it is more common to use senones as the target labels for the classification network as in [34], the output of our classification network was based on phonemes in order to be able to compare with other studies on semi-supervised learning for speech.

To simulate the effect of missing labels during training, the training set was divided into a labelled portion and an unlabelled portion of data set. The percentage of labelled frames in the training set was varied from 1% to 30% with intermediate steps: 3%, 5%, 10%, 20%. For each of these conditions, we optimised the hyper-parameter α on the validation set. All the accuracy results are reported for the optimal value of α. Finally the number of nodes in the encoder network was also optimised on the validation set resulting in an optimal value of 10000 nodes.

As a baseline, we compare the results obtained with our method with those obtained with a similar neural network trained with supervised backpropagation, on the same amount of labelled examples. The network contained a single hidden layer of 2000 units as in [20, 35]. Finally, we compare our results with those obtained in the literature on semi-supervised learning.

3.2. Practical Setup

We used Kaldi [32] and PDNN [33] for feature extraction, Theano [37] for symbolic algebra and GPU computing. The experiments were run on a Titan X card installed on a Ubuntu 14.04 based machine.

4. Results

Figure 2 illustrates the results that are also detailed in Table 1. We report the frame-level classification accuracy rates for the neural network trained in a supervised way (NN) and the proposed single layer semi-supervised sparse auto-encoder (SSSAE) for varying percentage of labelled data. Both test set and validation set accuracy are reported. Table 1 also reports the number of labelled frames (observations) and the value of the hyper-parameter α that was optimised independently for each case on the validation set.

The results for the supervised model are similar to those obtained with an equivalent model in [20, 35].

Our results show that the proposed semi-supervised method always outperforms the supervised baseline by as much as 2.9% absolute improvement. As expected this advantage decreases when the proportion of labelled training examples is increased. The validation and test errors are always very close, indicating that the parameters optimised on the validation set generalise well to the test set and both models avoid over-fitting.

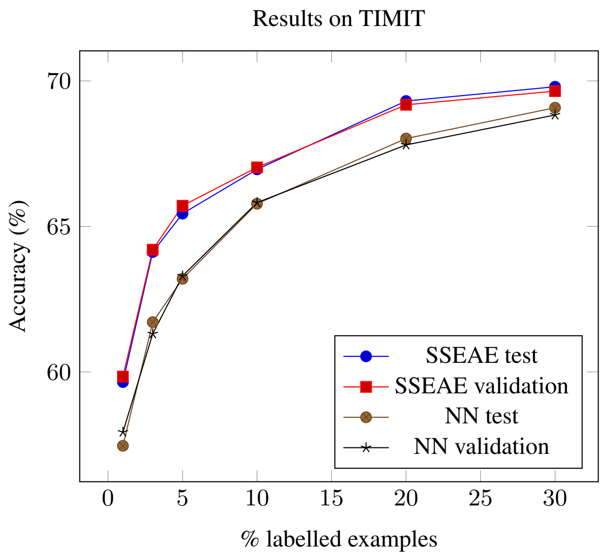

Figure 2: *Frame-based phone recognition accuracy (%) versus percentage of labelled training examples on the TIMIT database. NN: neural network trained with supervised back-propagation. SSSAE: our method. Both validation and test accuracy rates are shown. See Table 1 for the corresponding numerical values.*

Table 2: *Accuracy rates (%) for frame-based phone classification on TIMIT for the baseline (NN), the four different algorithms in GBL-SSL [20] and our model, SSSAE*

Comparison with other methods			
		10% labelled	30% labelled
Method	Reference	Test accuracy (%)	
NN	this work	65.94	69.24
LP	[20]	65.47	69.24
MP	[20]	65.48	69.24
MAD	[20]	66.53	70.25
pMP	[20]	67.22	71.06
SSSAE	this work	67.03	69.65

As expected, the optimal value for α is strongly dependent on the proportion of labelled material. The higher the proportion the more weight the algorithm gives to the classification error, compared to the unsupervised reconstruction error.

In Table 2 we compare the performance of our system to the results obtained with graph based semi-supervised learning methods published in [20] on 10% and 30% labelled data. We observe that our system performs better than all the techniques mentioned except the Prior Regularised Measure Propagation (pMP) algorithm.

Finally, it is interesting to consider the current state-of-the-art performance in phone classification on the TIMIT database as an upper bound to the results that can be obtained on that data set. To our knowledge, the best performing method is based on hierarchical convolutional deep maxout networks and achieves 16.5% Phone Error Rate (or, equivalently, 83.5% accuracy) [38].

5. Conclusions

We reported results on frame based phone classification on the TIMIT database using semi-supervised learning based on sparse autoencoders. We observe that our method outperforms a neural network of similar complexity trained with supervised back-propagation on the same amount of labelled training data in all experimental conditions. Our results also outperform many of the semi-supervised learning methods proposed in the literature for a similar task, with the exception of Prior-Regularised Measure Propagation (pMP) method. As expected, the advantage of using our method decreases when the proportion of labelled training observations is increased. However, we can argue that in realistic situations we will always find an abundance of unlabelled data as compared to data that was carefully annotated. As a consequence, it becomes more important for us to investigate our model when the percentage of labelled data is low.

The results we obtain, although comparable with the state-of-the-art in semi-supervised learning, are not comparable with the current state-of-the-art in phone recognition on the TIMIT database which is 16.5% Phone Error Rate (or, equivalently, 83.5% accuracy) [38]. The reason for this is twofold: Firstly, the above results only use a fraction of the labels provided by the TIMIT database for training. Secondly, the goal of the work is to compare semi-supervised and supervised learning in similar settings, and, therefore, many parameters that could be optimised have been left out in this study. For example, state-of-the-art ASR methods take advantage of the ability of neural networks to deal with correlated inputs. Filterbank features have been found to perform better with these models than the previously popular MFCCs. The reason for using MFCCs in this study was to allow for comparison with previous results in the literature that also used the same features.

In spite of the promising results, in order to draw general conclusions on ASR, we would need to test our method on a word recognition task, and, in particular, on large-vocabulary ASR. However, the improvements we see in frame-level phone classification are an incentive to continue work in this direction. Possible improvements may be obtained by using alternative features (e.g. filterbank features), testing the effect of adding depth to the model by stacking several blocks of autoencoders/classifiers, and, finally experimenting with techniques such as batch-normalisation.

6. Acknowledgements

The GeForce GTX TITAN X used for this research were donated by the NVIDIA Corporation. This research was partly supported by the CHIST-ERA project IGLU.

7. References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[3] D. Erhan, Y. Bengio, A. Courville, P.-A. Mansagol, and P. Vincent, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[4] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. of ICML*, 2013.

[5] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton,

"On rectified linear units for speech processing," in *Proc. ICASSP*, 2013.

[6] C.-H. Lee, F. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," vol. 1, 1988, pp. 501–504.

[7] T. Svendsen, K. Paliwal, E. Harborg, and P. Husoy, "An improved sub-word based speech recognizer," vol. 1, 1989, pp. 108–111.

[8] M. Bacchiani, M. Ostendorf, Y. Sagisaka, and K. Paliwal, "Design of a speech recognition system based on acoustically derived segmental units," vol. 1, 1996, pp. 443–446.

[9] M. Huijbregts, M. McLaren, and D. Van Leeuwen, "Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection." IEEE, 2011, pp. 4436–4439.

[10] P. O'Grady, "Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint," *Neurocomputing*, vol. 72, no. 1-3, pp. 88–101, 2008.

[11] O. Räsänen, "A computational model of word segmentation from continuous speech using transitional probabilities of atomic acoustic events," *Cognition*, vol. 120, no. 2, pp. 149 – 176, 2011.

[12] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," vol. 16, no. 1, pp. 186–197, 2008.

[13] G. Aimetti, R. K. Moore, and L. ten Bosch, "Discovering an optimal set of minimally contrasting acoustic speech units: A point of focus for whole-word pattern matching," 2010, pp. 310 – 313.

[14] V. Stouten, K. Demuynck, and H. van Hamme, "Discovering phone patterns in spoken utterances by non-negative matrix factorization," *IEEE Signal Processing Lett.*, vol. 15, pp. 131–134, 2008.

[15] J. Driesen, L. ten Bosch, and H. van Hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," 2009.

[16] N. Vanhainen and G. Salvi, "Word discovery with beta process factor analysis," Portland, OR, USA, Sep. 2012.

[17] ——, "Pattern discovery in continuous speech using block diagonal infinite hmm," 2014.

[18] M. Versteegh, R. Thiolliere, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The Zero Resource Speech Challenge 2015," 2015.

[19] A. Coates, H. Lee, and A. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, 2011, pp. 215–223.

[20] Y. Liu and K. Kirchhoff, "Graph-based semi-supervised learning for phone and segment classification." in *INTERSPEECH*, 2013, pp. 1840–1843.

[21] ——, "Graph-based semi-supervised acoustic modeling in DNN-based speech recognition," in *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 177–182.

[22] ——, "Graph-based semisupervised learning for acoustic modeling in automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1946–1956, Nov 2016.

[23] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks." in *Proceedings of IEEE Conference on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 267–272.

[24] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semisupervised GMM and DNN acoustic model training with multisystem combination and confidence re-calibration," 2013.

[25] J. Ma and R. Schwartz, "Unsupervised versus supervised training of acoustic models," 2008.

[26] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, no. 1, pp. 115–129, 2002.

[27] M. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks." in *ICML*, ser. ACM International Conference Proceeding Series, W. W. Cohen, A. McCallum, and S. T. Roweis, Eds., vol. 307. ACM, 2008, pp. 792–799.

[28] H. Valpola, "From neural pca to deep unsupervised learning," in *Adv. in Independent Component Analysis and Learning Machines*. Elsevier, 2015, p. 143–171, arXiv:1411.7783.

[29] A. Rasmus, H. Valpola, and M. Honkala, "Semi-supervised learning with ladder networks," in *NIPS*, 2015, arXiv:1507.02672.

[30] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 153–160.

[31] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA speech recognition research database: Specifications and status," in *Proceedings of DARPA Workshop on Speech Recognition*, 1986, pp. 93–99.

[32] D. Povey, A. Ghoshal, G. Boulianne, and Burget, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.

[33] Y. Miao, "Kaldi+pdnn: Building dnn-based asr systems with kaldi and pdnn," *ArXiv e-prints*, Jan. 2014.

[34] A.-r. Mohamed, T. N. Sainath, G. E. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition." in *ICASSP*. IEEE, 2011, pp. 5060–5063.

[35] J. Labiak and K. Livescu, "Nearest neighbors with learned distances for phonetic frame classification," in *Interspeech*, 2011.

[36] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, Nov 1989.

[37] P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," in *Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2012.

[38] L. Tóth, "Phone recognition with hierarchical convolutional deep maxout networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 25, 2015, dOI: 10.1186/s13636-015-0068-3.