INVESTIGATING SPARSE DEEP NEURAL NETWORKS FOR SPEECH RECOGNITION

Gueorgui Pironkov, Stéphane Dupont, Thierry Dutoit

TCTS Lab, University of Mons, Belgium

{gueorgui.pironkov, stephane.dupont, thierry.dutoit}@umons.ac.be

ABSTRACT

We propose an organized sparse deep neural network architecture for automatic speech recognition. The proposed method is inspired by the tonotopic organization in the auditory nerve/cortex. The approach consists of limiting the neurons connections between the hidden layers, in a manner that preserves frequency proximity, resulting in a diffuse integration of the spectral information inside the neural network. This method is put in perspective with related work on sparser neural network architectures for speech recognition (tonotopy, convolutional nets, dropout). The model is trained and tested on the TIMIT database, showing encouraging results compared to the traditional fully connected architecture.

Index Terms— Automatic speech recognition, deep neural networks, sparse, TIMIT

1. INTRODUCTION

Deep neural networks (DNN) have proven to be a powerful discriminative modeling tool for automatic speech recognition (ASR) [1]. Thanks to their many levels of non-linearities, they are able to address complex classification tasks. As the DNN features propagate deeper and deeper in the network layers, they become increasingly invariant and discriminative [2]. Deep neural networks where initially inspired from the mammal brain functioning, where the frequency-based information coming from the cochlea is transmitted to the auditory cortex trough the auditory nerve [3]. In the classical configuration of a DNN, the neurons from one layer to another are all interconnected, which means that a neuron in the l^{th} layer will be activated depending of the information it receives from all neurons in the $(l-1)^{\text{th}}$ layer.

In a real mammal brain, the structure is a slightly different. It follows a so-called tonotopic scheme [4, 5]. This organization implies that a particular neuron in the auditory cortex receives an electrical signal from a particular zone of the cochlea, that is, a distinct frequency sub-band. Thus, this neuron will fire depending of the information it receives from a limited number of neurons situated in a peculiar narrow-frequency area, and not from all possible frequencies.

Applying this observation to a classical fully connected DNN leads to limiting the weights connections within the neural network in an ordered manner. By doing this, the number of connections within the network will decrease and thus, the DNN weight matrices will become sparser.

This paper is organized as follows. Section 2 describes related work. In Section 3 we present the sparse neural network architecture. Section 4 presents the experimental setup. The obtained results are presented in Section 5. Finally, we conclude and present future work ideas in Section 6.

2. RELATED WORK

In 1996, [6] proposes to process frequency band features independently of each other, in effect resulting in a sparse connectivity of the first layers of a DNN. [7] introduces for the first time a tonotopic approach for speech recognition one year later. The weights connectivity was limited by using exponentially decaying connectivity probability function, depending of the distance between neurons weights. This method showed interesting results but was tested only on a single-layered MLP, due to hardware limitations.

Similarly to sparse DNN, Convolutional Neural Networks (CNN) tend to limit the weights connectivity, as they parse the spectral information localized-patch by localized-patch into the network [8, 9]. The method we propose differs with the CNN approach as the weights in a sparse DNN are kept independently updated contrarily to CNN, where the same weights are applied across the different patches. To a certain extent, a sparse DNN can be seen as a particular case of CNN with limited weight sharing (LWS) as in [10], where the LWS is set to one connection.

Beside the biological inspiration outlined in the introduction, another motivation to reduce the number of connections is that fully connected neural networks tend to be over-parametrized, leading to a non-negligible redundancy for the model [11]. Several architectures investigates sparser models to address this problem. For instance in [12], the fully connected layer is replaced with global average pooling. Another method used to reduce the DNN complexity is network pruning as in [13]. These methods differ with the sparse DNN we present by their internal architecture.

A parallel between sparse DNN and dropout also exists. Both mechanisms lead to thinned neural nets at training time, with reduced number of connections between layers [14, 15].

This comparison is limited though, as for a sparse DNN the number of units is kept the same as in a fully connected DNN, whereas part of these units are ignored during training time with dropout. Another element of divergence concerns the selection of the suppressed connections. For dropout, this connections are selected randomly in order to decrease the neurons co-adaptation, thus, preventing over-fitting. In our case, we strengthen close frequency band connections, and avoid units being influenced by too spatially-faraway units, following the tonotopic scheme. Despite the divergence of these approaches, some heuristic observations deriving from dropout enabled us to improve sparse DNN (cf. Section 5.3).

3. SPARSE DEEP NEURAL NETWORK ARCHITECTURE

As explained earlier, the sparsity of a network can be obtained by different means. Our interest is in reducing the connections in a controlled manner, following the natural sparseness of mammal brains.

3.1. Physiological approach

The tonotopic architecture is depicted in Figure 1. The vibrations in the air are transmitted through the tympanic membrane to the base of the cochlea. Depending of the sound frequency, a different region of the basilar membrane will be excited. Lower frequencies will excite areas closer to the cochlea base, whereas higher frequencies will be closer to the apex. The basilar membrane stimulation information is then passed throughout the cochlea nerves (a.k.a. auditory nerves) to the neurons. This mechanism implies that only the neurons connected to a specific zone of the basilar membrane will be stimulated, depending of the initial sound frequencies.

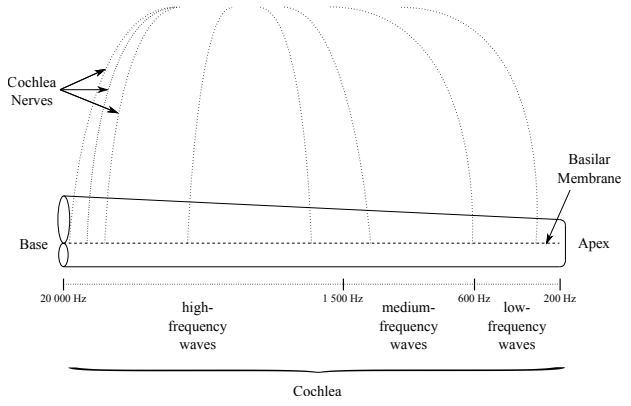

Fig. 1. Tonotopic scheme. The cochlea is represented uncoiled for clarity.

3.2. Sparse DNN

Transposing this mechanism to classical fully connected layers deep neural networks leads to Figure 2. In this example the connections of the first three hidden layers are limited in such way that neurons in the l^{th} layer will receive stimulation information only from close neurons in the $(l-1)^{\text{th}}$ layer.

The aim of sparse layers is to model spectral correlation while reducing spectral variation as well as feature information redundancy. After a few sparse layers, we keep some classical fully connected layers, similarly to CNN training. Keeping fully connected layers will ease accumulating and transmitting the local information up to the softmax layer.

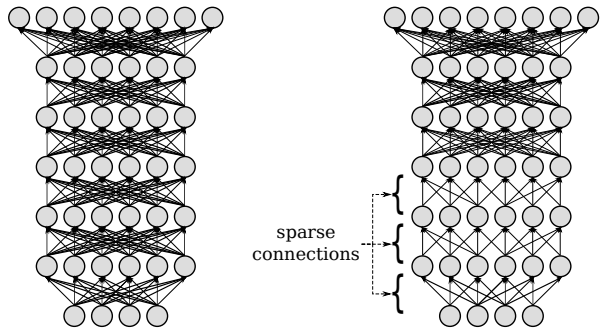

Fig. 2. Example of sparse deep neural network. Here the connections from the input features layer up to the 3rd hidden layer are sparse.

Sparse DNN are also interesting in the manner feature information propagates though the network. In contrast to classical DNN or CNN, here the neurons from the first sparse layers will have access to a limited amount of the audio features. The spectral information from these features will spread in the network as the layers become deeper and deeper. In other words, we can describe the data integration inside the neural network as *layer-wise diffusion*. Suppressing large amounts of weight connections in the sparse layers implies that we will need to go to a deeper layer to see a neuron influenced by all acoustic features. On the contrary, if we keep most of the connections in the sparse layers, the information coming from the initial features will influence neurons in earlier layers. For instance, if we suppress 40% of the weights, each unit in the first layer will be updated by 60% of the features. Following this logic, each neuron in the second hidden layer will encounter 60% of the neurons from the first layer. Thus, each single neuron in the second layer will be influenced by all initial features. The lower the connection percent is, the longer it takes for information from the different frequency channels to diffuse and integrate through the sparse DNN.

We will use the term *connection percent* to describe the proportion of the connections that are kept in the sparse weight matrices.

4. EXPERIMENTAL SETUP

In this section, we will describe the different components of our setup : the database, the initial audio features and a description of our system.

4.1. Database

The proposed sparse DNN approach was investigated on a phone recognition task using the TIMIT Acoustic-Phonetic Continuous Speech Corpus [16].

The database is composed of 630 speakers, each of them reading 10 sentences. Pre-training and training were performed on the standard training set (consisting of 462 speakers). Ten percent of the training data is randomly hold out as part of the development set, first to fine-tune the hyper-parameters, second to avoid over-fitting by performing an early stopping procedure. The 24-speaker standard test set, which is independent from the train end development sets, is used for evaluation. The Phone Error Rate (PER) metric is measured based on the phone label outputs and the supplied phone transcription.

4.2. Input features

The audio features choice is crucial for sparse neural networks. Most speech recognition systems use Mel-Frequency Cepstral Coefficients (MFCC) as input features. This type of features have proven to be very efficient with Gaussian Mixture Models - Hidden Markov Model (GMM - HMM) acoustic models. MFCC are easier to model using GMM as their coefficients tend to be independent. [17] shows that deep networks do not require uncorrelated features. Thus, features as the filter banks, that are strongly correlated, are worth considering. Results in [17] reveal better performances of filter banks compared to MFCC on the TIMIT database.

Another effective feature extraction approach for neural networks are LDA-MLLT features [18]. These features are obtained by splicing successive MFCC vectors. The resulting features are reduced by a LDA projection. Extracting features in such manner would not be desirable for our sparse DNN approach. Using LDA implies a projection of the features into another subspace, meaning that the spectral proximity, similar to the one resulting from the cochlea, will be lost. With sparse DNN we try to boost features spectral closeness.

Here, we are using 23-dimensional filter banks spliced over 11 frames with their associated first- and second-oder temporal differences for a total of 759 dimensions.

4.3. System description

Training and testing are done on the free, open-source Kaldi speech recognition toolkit [19].

4.3.1. Baseline

We use a hybrid DNN - HMM model, with the DNN estimating posterior probabilities of tied HMM states, and the HMM modeling speech temporal nature. Our baseline system is a 6 hidden layers fully connected neural network with 2046 neurons per layer.

The system is pre-trained using greedy unsupervised layer-wise training of stacked restricted Boltzmann machines (RBM) [20]. At training time, the network is trained to minimize the cross entropy error between the expected HMM target states and the softmax output. For pre-training the learning-rate is fixed at 0.4, it decreases to 0.008 during training. We set the mini-batch size to 128.

During decoding, the network's state probabilities are associated with a dictionary and language model to establish the most likely transcription.

4.3.2. Proposed system

In order to introduce sparsity in the model, we are using sparse filters. Each of the concerned weight matrices are multiplied by a sparse filter before the propagation phase. The backpropagation phase is not a problem as only the non-null weights will be updated. As mentioned earlier, we will use *connection percent* to describe the quantity of kept connections. Finally in order to preserve the weights providing information of spectrally-close features, the sparse filter matrix is diagonalized. Thus, depending of the percent of suppressed connections the digonal width will variate. For a higher connection percent, the diagonal of the sparse filter will be thicker than for a lower connection percent.

The sparse neural network that will be investigated is very similar to the one depicted in Figure 3. Each rectangle corresponds to a weight matrix. The white ones are the initial features or neuron vectors, the gray ones are the weight matrices. In this example the network has 6 hidden layers, 3 of them being sparse, and a connection percent of 20%. The diagonal weight matrices can be clearly observed between the 1st and 3rd hidden layers. The weight matrix between the initial audio features and the 1st layer is slightly more specific. Three different diagonals are computed, corresponding to 23-bins filter bank, the delta and the delta delta derivatives. Multiple diagonals in the first weight matrix ensure that units in the first hidden layer receive acoustic information from the same spectral areas. This matrix is simplified in Figure 3 for clarity, as in practice we also add splicing, leading to a total of 33 diagonals during training. After the 3rd hidden layer, all weight matrices are fully connected from one layer to another.

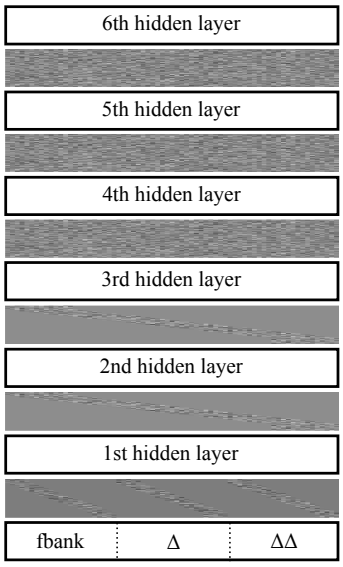

Fig. 3. Matrices of a sparse DNN are schematically represented for a connection percent of 20%. Gray rectangles represent weight matrices. White rectangles represent the initial features or neuron vectors. For clarity, we display no splicing in the initial filter banks (+ delta + delta delta).

The weight matrices are filtered during the RBM unsupervised pre-training and the fine-tunning iterations. During decoding, the weight matrices obtained from training are tested, without any sparse filtering.

5. RESULTS

We investigate sparse neural nets efficiency by varying different parameters of the initial setup. The efficiency is measured with the PER over both development and test sets. The first step is to evaluate the impact of lowering the percent connection of the weight matrices. Then, the number of sparse layers is examined. We also explore some dropout heuristic observation on sparse DNN.

5.1. Connection percent

In this section we are using the 6 hidden layer sparse DNN with 3 layers of sparse connections between the filter bank features and the 3rd hidden layer (as in in Figure 3). The connection percent is varied from 10% to 100%, the fully connected DNN baseline corresponding to 100%. The results are presented in Table 1.

The baseline is fixed at 18.9% PER for the development set and 20.5% for the test set. These PER are higher compared to current existing results on TIMIT. One of the main reasons is features. LDA-MLLT features outperform filter banks in fully connected DNN. However, we have seen in Section 4.2 that LDA features are unadapted for the sparse structure we are proposing.

A slight decrease of the number of connections (90% and 80% connections) seems to lower the PER on both sets (up to 1.6% relative improvement on dev set and 2.9% on test set). The word error rate increases around the baseline values between 70% and 50% connections. The best results are obtained for a connection percent of 40%. The relative improvement for dev set is 3.2% and 3.4% for the test set. Then, the recognition decreases. These results are encouraging and show that improvement can be brought by this approach.

Table 1. Varying the number of connections within the first three weight matrices.

Connection percent	*dev* PER	*test* PER
fully connected	18.9	20.5
.....	.....	.....
90	18.6	20.4
80	18.8	19.9
70	18.9	20.6
60	18.9	20.7
50	18.9	20.4
40	**18.3**	**19.8**
30	18.6	20.4
20	19	20.3
10	19	20.7

5.2. Number of sparse layers

Starting with the best performing setup from the previous section (40% connections in the weight matrices), we try here to evaluate the influence of the number of sparse layers on system performance. Thus, the number of sparse layers ranged from 1 sparse layer to 5, starting by the weight matrix between the filter banks features and the 1st hidden layer. The results in Table 2 demonstrate that the lower PER is obtained for 3 sparse layers, our initial choice.

Table 2. Varying the number of consecutive sparse layers, starting from the weight matrix between the input features and the 1st layer. The connection percent is fixed at 40%.

Sparse layer(s)	*dev* PER	*test* PER
5	19.1	20.7
4	18.6	20.3
3	**18.3**	**19.8**
2	18.4	20.6
1	18.8	20.6

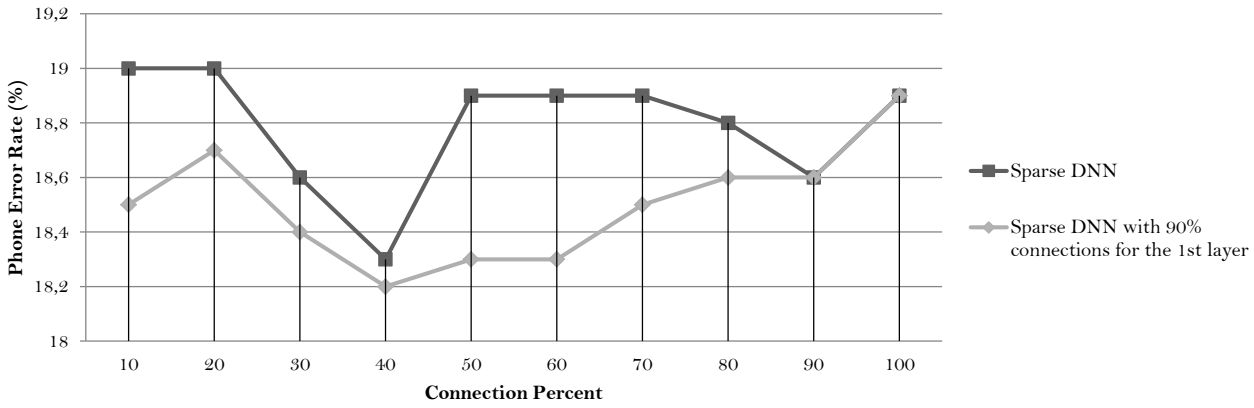

Fig. 4. Improvement of fixing the 1st sparse layer at 90% connection percent on TIMIT *dev set*, when the connectivity of the 2nd and 3rd sparse layer varies.

5.3. Dropout inspired improvement

We have seen in Section 2 that a parallel between sparse DNN and dropout exists. [15] shows that the optimal solution for dropout training on TIMIT consists of using different dropout probabilities of retention depending of the layer. More specifically, [15] suggests that the first layer should have a higher dropout probability of retention than the other layers, meaning to keep more neurons in the 1st hidden layer. We have adapted this observation to sparse DNN by fixing the connection percent of the 1st weight matrix at 90%, the 2nd and 3rd weight matrices varying between 90% and 10%. Table 3 presents the results.

Table 3. Varying the number of connections of the 2nd and 3rd weight matrices of the sparse DNN, when the 1st layer connection percent is fixed at 90%.

Connection percent	*dev* PER	*test* PER
fully connected	18.9	20.5
.....	.....	.....
90	18.6	20.4
80	18.6	20.2
70	18.5	20.5
60	18.3	20.1
50	18.3	20.1
40	**18.2**	20
30	18.4	**19.7**
20	18.7	20.2
10	18.5	19.9

The obtained relative improvements, using this method, reach 3.7% on the dev set for 40% connections and 3.9% on the test set for 30% connections, in comparison with a fully connected neural network.

Results on the *development set* from Table 1 and Table 3 are gathered in Figure 4. We can see the beneficial effect of a less-sparse first layer. Indeed, the PER is reduced when the first weight matrix is fixed at 90% connections, independently of the connection percent of the other sparse weight matrices.

6. CONCLUSIONS

In this paper, we presented a frequency-organized sparse method for deep neural networks architecture. This model is inspired by the biological tonotopic configuration. We discussed the importance of features for this kind of setup, showing why filter banks are the most relevant ones. By diagonalizing the weight matrices we were able to keep close frequency information, while reducing the number of connections in the hidden layers weight matrices. Encouraging results were obtained with the proposed sparse design in comparison to classical fully connected networks. Following recent observation on the best strategies for dropout, we further improved these results by fixing a high connection percent for the first weight matrix.

Future work involves testing this approach on bigger and noisy databases as Aurora 4. We will also investigated how the proposed sparse DNN could reduce training and testing time. We have studied in this paper the frequency-domain influence. Furthermore, we are interested in the time-domain effect on recognition. This can be tested by varying the number of spliced frames.

7. ACKNOWLEDGEMENTS

This work has been partly funded by the European Regional Development Fund (ERDF) through the DigiSTORM project.

8. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[3] Warren S McCulloch and Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[4] Nelson Yuan-Sheng Kiang, "Discharge patterns of single fibers in the cat's auditory nerve.," Tech. Rep., DTIC Document, 1965.

[5] Christo Pantev, Olivier Bertrand, Carsten Eulitz, Chantal Verkindt, S Hampson, Gerhard Schuierer, and Thomas Elbert, "Specific tonotopic organizations of different areas of the human auditory cortex revealed by simultaneous magnetic and electric recordings," *Electroencephalography and clinical neurophysiology*, vol. 94, no. 1, pp. 26–40, 1995.

[6] Hervé Bourlard and Sthéhane Dupont, "A mew asr approach based on independent processing and recombination of partial frequency bands," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. IEEE, 1996, vol. 1, pp. 426–429.

[7] Nikko Ström, "A tonotopic artificial neural network architecture for phoneme probability estimation," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 156–163.

[8] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[9] László Tóth, "Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 190–194.

[10] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, 2014.

[11] Misha Denil, Babak Shakibi, Laurent Dinh, Nando de Freitas, et al., "Predicting parameters in deep learning," in *Advances in Neural Information Processing Systems*, 2013, pp. 2148–2156.

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[13] Babak Hassibi and David G Stork, *Second order derivatives for network pruning: Optimal brain surgeon*, Morgan Kaufmann, 1993.

[14] George E Dahl, Tara N Sainath, and Geoffrey E Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8609–8613.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] John S Garofolo, Linguistic Data Consortium, et al., *TIMIT: acoustic-phonetic continuous speech corpus*, Linguistic Data Consortium, 1993.

[17] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn, "Understanding how deep belief networks perform acoustic modelling," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4273–4276.

[18] George Saon, Mukund Padmanabhan, Ramesh Gopinath, and Scott Chen, "Maximum likelihood discriminant feature spaces," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 2, pp. II1129–II1132.

[19] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," 2011.

[20] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.