DEEP ORDER STATISTIC NETWORKS

Steven J. Rennie, Vaibhava Goel, and Samuel Thomas

IBM Thomas J. Watson Research Center

{sjrennie, vgoel, sthomas}@us.ibm.com

ABSTRACT

Recently, Maxout networks have demonstrated state-of-the-art performance on several machne learning tasks, which has fueled aggressive research on Maxout networks and generalizations thereof. In this work, we propose the utilization of order statistics as a generalization of the max non-linearity. A particularly general example of an order-statistic non-linearity is the "sortout" non-linearity, which outputs all input activations, but in sorted order. Such Order-statistic networks (OSNs), in contrast with other recently proposed generalizations of Maxout networks, leave the determination of the interpolation weights on the activations to the network, and remain conditionally linear given the input, and so are well suited for powerful model aggregation techniques such as dropout, drop connect, and annealed dropout. Experimental results demonstrate that the use of order statistics rather than Maxout networks can lead to substantial improvements in the word error rate (WER) performance of automatic speech recognition systems.

***Index Terms*—** Order Statistic Networks, Maxout Networks, Rectified Linear Units, Deep Neural Networks, Multi-Layer Perceptrons.

1. INTRODUCTION

Recently, Maxout Networks [1] have demonstrated state-of-the-art performance on several machine learning tasks [1–3]. These networks abandon traditional network non-linearities and generalize rectified-linear networks [4] by utilizing units that are the maximum over a set of affine functions of the input. Maxout networks are conditionally linear given an input and so well suited for model aggregation techniques such as dropout [5] and drop-connect [6] which discourage co-adaptation of feature detectors. Their recent success has fueled aggressive research on maxout networks, and generalizations [3, 7].

Recently published generalizations involve the utilization of the log-sum function as a non-linearity, which is continuously differentiable, and closely approximates the max function [3], and LP norm-based non-linearities such as the L2 norm [3, 7], which has deep connections with independent component analysis (ICA), and sparse coding [8, 9]. Such generalizations have the property that, for a given input, multiple activations explain the generated output. Such networks can interpolate between 'modes' of the detector in the sense that multple high activations can produce a stronger response that would be output by the max non-linearity, but the interpolation weights are pre-determined by the non-linearity.

In this work, we propose the utilization of order statistics as a generalization of the max non-linearity. A particularly general example of an order-statistic non-linearity is the "Sortout" non-linearity, which outputs all input activations, but in sorted order. Such networks leave the determination of the interpolation weights on the activations to the network and are a strict generalization of Maxout networks. Importantly, these networks remain conditionally linear given the input, which makes them ideally suited for powerful model aggregation techniques such as dropout [5], drop-connect [6], and annealed dropout [10]. In practise, order statistics in the context of detection can be of diminishing returns, but when utilized in deep neural networks, this "detector" is part of a complex classification (or regression) task, and order statistics beyond the max can be utilized to improve classification (or regression) performance. We demonstrate that Order Statistic Networks (OSNs) perform on-par with Maxout networks on both Aurora 4, a small scale, medium vocabulary automatic speech recognition task, and in the context of a larger scale internal open voice search (OVS) task. Furthermore, preliminary investigations suggests that by regularizing the weights of OSNs, they can outperform Maxout networks. Our best OSNs, which are trained using annealed dropout [10] outperform the best published WER results on the Aurora 4 database that we are aware of [11] by 10% relative. OSNs, like standard deep neural networks, are applicable to any task (e.g. classification or regression) that involves mapping inputs to target outputs.

2. DEEP ORDER STATISTIC NETWORKS

Maxout networks [1] have non-linearities of the form:

$$s_j = \max_{i \in C(j)} a_i \tag{1}$$

where the activations, a_i, are typically based on inner products with an input feature:

$$a_i = \sum_k w_{ik} x_k + b_i \tag{2}$$

In the case of activations with unconstrained weights, the sets $C(j) \forall j$ are generally disjoint [1] . Such "pooling" can of course also be overlapping, as is the case for Maxout CNNs [1] and networks layers constrained to have local receptive fields (LRFs) [7], where pooling is done over spatially "local" activations.

In this work we propose Deep Order-statistic Networks (DONs), which utilize non-linearities of the form:

$$\mathbf{s}_j = \mathbf{O}_j(a_i : i \in C(j)) \tag{3}$$

where $s_j[k] = O_j[k]$ is defined as the kth largest value in $a_i \in C(j)$. Note that the output for a given detector is *vector-valued*. Note also that the term "order statistic" is generally utilized in the context of a statistical sample. In this sense we treat the input activations to an order statistic non-linearity as a samples over detector activity level. Figure 2 depicts a plot that compares the non-linearities utilized by DONs to those of Maxout networks and traditional neural networks. While traditional networks apply non-linearities such as the sigmoid

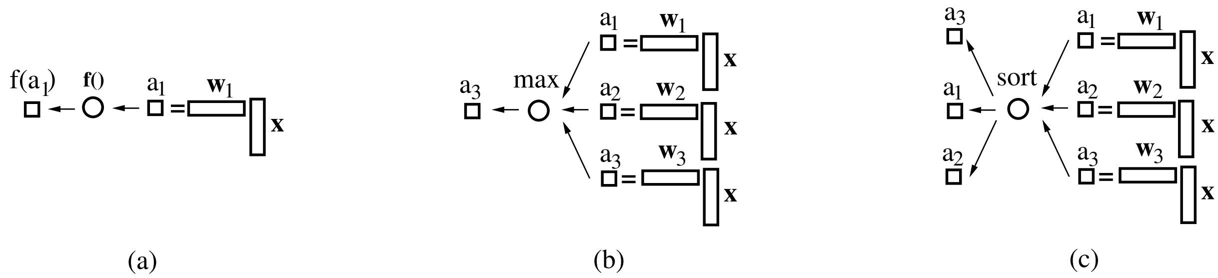

Fig. 1. Traditional units (a) apply a non-linear function independently to each input activation, whereas Maxout units (b) implement a detector with multiple modes. Order statistic networks generalize Maxout units by "ordering" their inputs, and then outputing all input activations, so that the detectors in the next layer can interpolate over them.

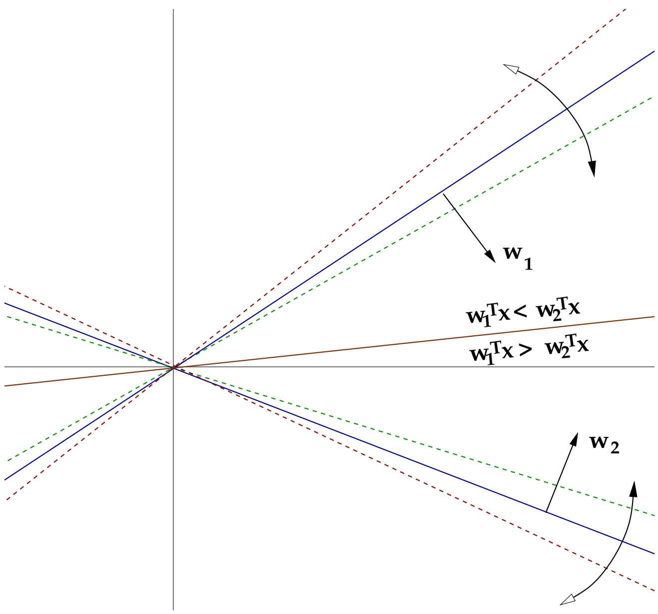

Fig. 2. The "sortout" non-linearity viewed as a customizable maxout unit for the case of two linear filters. Units in the next layer have access to both the maximum and minimum outputs, and so can form a weighted sum of these outputs to form an equivalent maxout projection with higher (red) or lower (green) intensity response levels.

function to each individual linear projection independently, Maxout networks utilize "clusters" of linear projections that jointly form a non-linear detector with multiple modes, and output the maximum detection result. DONs in contrast, output a more general set (e.g. all for the "Sortout" network depicted) of order statistics. This allows detectors in the subsequent layer of the network to linearly interpolate between these linear projections based on their "rank" ordering for the current input, effectively allowing for *customization* of the lower-level detector by the higher-level detectors that utilize it. For example, for the case of $F = 2$ linear projections being combined by a Sortout non-linearity (the case we will focus on in this paper), the activation produced by a given projection is given by:

$$a_i = \sum_j a_{ij} \tag{4}$$

where a_{ij} is the activation due to a given input.
For Sortout networks with $F = 2$:

$$\begin{aligned} a_{ij} &= \alpha_{ij} \max(w_{j1}^T x + b_{j1}, w_{j2}^T x + b_{j2}) + \\ &\quad \beta_{ij} \min(w_{j1}^T x + b_{j1}, w_{j2}^T x + b_{j2}) \\ &= (\alpha_{ij} w_{jm} + \beta_{ij} w_{j\bar{m}})^T x + (\alpha_{ij} b_{jm} + \beta_{ij} b_{j\bar{m}}) \\ &= \tilde{w}_{jm}^T x + \tilde{b}_{jm} \end{aligned} \tag{5}$$

where m and $\bar{m}$ encode the maximizing and minimizing arguments, respectively. This shows that detectors in the next layer can construct *customized* equivalents to Maxout units from a single Sortout unit, in the sense that the intensity of the response as a function of the input to the layer below their input can be modulated as depicted in fig. 2.

3. EXPERIMENTS ON AURORA 4

3.1. Task

The Aurora 4 task is small scale (10 hour), medium vocabulary noise and channel ASR robustness task based on the Wall Street Journal corpus [12]. All ASR models were trained using the task's multi-condition training set, which consists of 7137 base utterances (10 hours of data) sampled at 16kHz from 83 speakers. One half of the training utterances was recorded with a primary Sennheiser microphone, and the other half was collected using one of 18 other secondary microphones. Both sections of the training data contain both clean and noisy speech utterances. The noisy utterances are corrupted with one of six different noise types (airport, babble, car, restaurant, street traffic and train station) at 10-20 dB SNR.
The standard Aurora 4 test set was utilized, which consists of 330 base utterances from 8 speakers, which are used to generate 14 test conditions (330x14=4620 utterances in total). As with the training set, the test set was also recorded using two microphones–a primary microphone and a secondary microphone, where the secondary microphone is different than the secondary mic. used in the training set). The same six noise types used during training are used to create noisy test utterances with SNRs ranging from 5-15dB SNR, resulting in a total of 14 test sets. These test sets are commonly grouped into 4 subsets–clean (1 test case, group A), noisy (6 test cases, group B), clean with channel distortion (1 test case, group C) and noisy

with channel distortion (6 test cases, group D).

3.2. Baseline ASR systems

Before building deep neural network (DNN) baselines for multi-condition training, an initial set of HMM-GMM models was trained to produce alignments. Unlike the baseline systems that will be described momentarily, these models are built on the corresponding clean training (7137 utterances) set of the Aurora 4 task in speaker-dependent fashion. Starting with 39-dimentional VTL-warped PLP features and speaker-based cepstral mean/variance normalization, an ML system with FMLLR based speaker adaptation and 2000 context-dependent HMM states is trained. The alignments produced by this system were further refined using a DNN system also trained on the clean training set with FMLLR based features.

Three sets of neural network based system baselines were built for the multi-condition task. The first set are unconstrained deep neural networks and include models that utilize rectified linear (ReLU), and Maxout non-linearites with 2 filters/unit. Corresponding networks with constrained feature extraction layers–both convolutional networks, CNNs [13], and networks that utilize local receptive fields, LRFS [7]–were also trained. All the systems were trained on 40 dimensional *log-mel* spectra augmented with Δ and $\Delta\Delta$ features based on a cross-entropy criterion, using stochastic gradient decent (SGD), and a mini-batch size of 256. The *log-mel* spectra were extracted by first applying *mel* scale integrators on power spectral estimates taken over short analysis windows (25 ms). Each frame of speech was appended with a context of ± 5 frames after applying speaker independent global mean and variance normalization. After training, the Aurora 4 test set is decoded with the trained acoustic model and the task-standard WSJ0 bigram language model using the Attila dynamic decoder [14], and then scored using scoring scripts from the Kaldi toolkit [15].

3.2.1. *DNN Systems*

All DNN systems estimate the posteriors of 2000 output targets using networks with 7 hidden layers and a varied number of hidden units. Note that, because of differences in the semantics of traditional, Maxout, and Sortout deep networks, the number of hidden units and number of parameters per layer are not in 1-1 correspondence. For example, a maxout network with 1K inputs, 1K outputs, and 2 linear projections (i.e. filters) per output unit has 2M parameters (ignoring biases), whereas a ReLU network with 2M pars/layer has $\sqrt{2M} \approx \underline{1414}$ hidden units/layer, and a Sortout unit with 2 filters/unit has $\sqrt{2M}/2 \approx 707$ units per hidden layer. For the DNN systems that utilize ReLU non-linearities, we utilized a fixed dropout rate of 50% on layers 4-6–we found that this was most effective dropout training strategy for ReLU networks. All Maxout and OSN networks were trained using annealed dropout [10], by annealing the dropout rate from 0.5 to zero linearly over 30 iterations, using a fixed learning rate decay rate, selecting the iteration with the best performance, and then performing additional iterations with the identified fixed dropout rate. We have found that annealed dropout is a much more effective for training Maxout and OSN networks than any fixed dropout rate scheme. Note that in the case of OSNs the entire set of outputs for a given unit should be jointly dropped out.

3.2.2. *CNN Systems*

All CNN baselines use two convolutive layers with 256 feature maps each, followed by five fully connected layers with 2 million parameters/layer, as for the DNN systems. The feature maps in the first layer utilize 9×9 filters that are convolved with the input *log-mel* representations. The feature maps in the second layer are applied after 3×1 (freq. x time) pooling and utilize 3×4 filters. Please consult [16,17] for further details on how the layers are combined. Similar to the DNN baselines, separate CNN baseline systems with ReLU non-linearities are also trained to estimate posterior probabilities of 2000 output targets. When ReLU non-linearities are used, a fixed dropout rate of 50% is applied to layers 4 and 5. Both the CNNs and DNNs are (layer-wise) discriminatively pre-trained before being fully trained to convergence, using the cross-entropy training criterion.

Table 1. ASR performance on the Aurora 4 task as a function of network type (WER%) for unconstrained DNNs. All networks utilize 7 hidden layers. The number of units per hidden layer are given following the non-linearity type. Networks depicted in the same color have the same number of parameters per hidden layer (ignoring unit biases, a negligible difference).

Network	A	B	C	D	AVG
ReLU, 1024	4.9	8.5	8.3	17.2	11.9
ReLU, 1414	4.9	8.7	8.2	16.9	11.9
ReLU, 2048	5.0	8.6	8.1	17.0	11.9
Maxout , 1024	4.3	7.7	7.0	15.6	10.8
OSN, 707	4.0	7.8	7.6	16.0	11.0
OSN, 1024	4.4	7.8	7.3	15.6	10.8

3.2.3. *LRF DNN Systems*

All LRF DNNs baseline models utilize an initial feature extraction layer with 40 feature maps based on 9×9 filters, with all weights *untied*, so that more complex invariances than translation can be learned.

3.3. Results

Table 1 summarizes the word error rate (WER) performance of ASR systems based on various DNN acoustic models. The number of units per hidden layer is given for each network, and the networks are color-coded according to the number of parameters per hidden layer. Annealed dropout [10] was used to train both the Maxout and OSN (Sortout) networks. For the case of unconstrained DNNs, our initial experiments suggest that OSNs slightly lag the performance of Maxout networks on a parameter for parameter basis, although our training procedures are more optimized for Maxout networks. Further regularization of the weights on higher order (here just min) outputs appears to be necessary.

Table 2 summarizes the word error rate (WER) performance of ASR systems based on various DNN acoustic models that utilize local receptive fields (LRFs) in thier initial layer. As before, the networks are color-coded according to number of pars. per hidden layer, and the number of hidden units per layer for each network is given. Again, annealed dropout was used to train both the Maxout and OSN (Sortout) LRF networks. Here the OSN networks outperform Maxout networks on a per-parameter basis, and the best network (WER10.0%) outperforms the best previous result we are aware of on Aurora 4 (posterior average of multiple ReLU networks, each dropout-trained on different noise aware features [11]) by 1.1% absolute, or 10% relative. The next best result we are aware of (sig-

Table 2. ASR performance on the Aurora 4 task as a function of network type (WER%) for DNNs that utilize local receptive fields (LRFS) in their first layer (9x9 patches,40 nodes per position). All networks utilize 7 hidden layers. The number of units per hidden layer are given following the non-linearity type. Networks depicted in the same color have the same number of parameters per hidden layer (ignoring unit biases, a negligible difference).

Network	A	B	C	D	AVG
ReLU LRF, 1414	4.7	8.3	7.5	16.1	11.3
Maxout LRF, 1024	4.1	7.6	6.7	15.1	10.5
Maxout LRF, 1414	4.2	7.4	6.5	14.8	10.3
OSN LRF, 707	3.8	7.4	6.6	15.1	10.4
OSN LRF, 1024	3.9	7.2	6.2	14.7	10.1
OSN LRF 1414	4.0	7.2	6.4	14.5	10.0

Table 3. ASR performance on the Aurora 4 task as a function of non-linearity (WER%) for CNNs. All networks utilize 7 hidden layers (initial 2 convolutional). The number of units per unconstrained hidden layer are given following the non-linearity type. Networks depicted in the same color have the same number of parameters per hidden layer (ignoring unit biases, a negligible difference).

Network	A	B	C	D	AVG
Relu CNN, 1024	4.8	8.4	7.4	16.0	11.3
ReLU CNN, 1414	4.9	8.1	7.3	15.5	11.0
ReLU CNN, 2048	5.1	9.0	8.3	16.5	11.9
Maxout CNN, 1024	4.0	7.8	6.7	14.9	10.5
Maxout CNN, 1414	4.0	7.6	6.4	14.6	10.3
OSN CNN, 707	4.3	7.8	7.0	14.8	10.5
OSN CNN, 1024	4.2	7.6	6.6	14.3	10.3

moid, dropout-trained, noise aware training [18]) is outperformed by 2.3% absolute, or 19% relative. Note that here we have not attempted to optimize the input features for noise and channel robustness, which should result in further gains. Table 3 shows that parameter for parameter OSN CNNs perform on par with Maxout CNNs, which significantly outperform the ReLU CNNs that we tested on Aurora 4.

3.3.1. Regularized OSNs

An OSN layer has roughly 4 times as many parameters as a Maxout layer with the same number of hidden units. However, it is natural to expect diminishing returns from higher order statistics in a dectection scenario, and to constrain the weights associated with the later order statistics (here just the minimum activation) to be sparse. To begin to explore the effects of contraining the weights of later order outputs we first experimented with varying the relative magnitude that the weights are initialized to, which is a very simple form of regularization. Table 4 depicts the results. The weights of the minimum projections are clearly less important to network performance than those of the maximum projections, as expected. However, there is also evidence that overly aggressive regularization of the minimum weights can hurt performance. We are currently experimenting with L1 and group (L1L2) regularization of the columns of the network matrices to improve the efficiency of inference in OSNs.

	WER (%)				
α	A	B	C	D	AVG
0	4.1	7.3	6.9	14.9	10.3
0.1	3.9	7.2	6.2	14.7	10.1
0.2	4.3	7.2	6.4	14.6	10.1
1.0	4.0	7.4	6.7	14.9	10.1

Table 4. Word error rate (WER) of OSN (Sortout) networks on the Aurora 4 task as a function of the relative initialization scale of the "min" outputs relative to the "max" outputs of the previous "sortout" layer, α. Interestingly, performance is not highly sensitive to α. All networks consist of 7 hidden layers with 1024 sortout units, and 2 linear filters/unit.

4. EXPERIMENTS - OVS

To begin to investigate how relevant OSNs are in data plenty scenarios we have conducted some preliminary experiments on 100 hours of internal open voice search data. Table 5 summarize the results we have gathered so far. Note that all networks were trained using the cross-entropy objective, based on allignments generated from a system trained on much more data, and that all networks have roughly the same number of parameters. As with the Auora 4 systems, all Maxout and OSN networks utilize annealed dropout (annealed to zero from 0.5) [10] during system training. This boosts WER performance substantially. Note that it was necessary to increase the size of the "pinch" layer to make maxout and OSN networks more effective, whereas for the baseline sigmoid acoustic model, small pinch layers do not negatively affect performance. Looking at the results, we can see that the OSN LRF with 1K hidden units per layer, which has the same number of parameters as the 1.4K Maxout and 2K Sigmoid based system, outperforms the baseline 2K Sigmoid system, and performs on-par with the 1.4K Maxout LRF in terms of word error rate (WER). The 1.4K hidden unit OSN LRF system is able to improve slightly on this result.

#H x #L + P	Network	WER(%)
2K x 5 + 100 (lin.)	Sigmoid	13.0
1.4K x 4 + 512	AD Maxout	12.6
1.4K x 4 + 512	AD Maxout+ LRF	12.5
1K x 4 + 512	AD Sortout+ LRF	12.5
1.4K x 4 + 512	AD Sortout+ LRF	12.4

Table 5. Word error rate (WER) as a function of model and dropout rate when trained/tested on 100/7 hours of (internal) open voice search (OVS) data. All maxout networks have two linear filters per maxout unit. For each model, the number of hidden layers (#L), number of units per hidden layer (#H), and the size of the 'pinch' layer (P) immediately before the output layer are specified. During training of the annealed dropout (AD) models, the dropout rate was linearly decayed to zero. All networks depicted in the same color have roughly the same number of parameters. All models were trained using a cross-entropy based criterion.

5. DISCUSSION AND CONCLUDING REMARKS

In this paper we have introduced a new type of deep network architecture, order statistic networks (OSNs). On the Aurora 4 task, OSNs far outperform the best published results on the task, and perform similarly to Maxout networks. Preliminary results on 100 hours of open voice search data are also promising. Several important questions remain. In this paper, we have focused on OSNs that utilize 2 linear filter per unit. Even in this scenario, OSNs are more computationally intensive on a per hidden unit basis than Maxout networks, and we are currently investigating how to regularize them, given the intuition and preliminary evidence that the weights of "min" outputs can be highly constrained. Similarly, networks that can efficiently utilize "sortout" units with more filters via careful regularization towards sparse solutions are important research direction. The intuition that sortout units implement customizable maxout units may be able to be leveraged to efficiently cluster the weights acting upon higher order statistics. Perhaps the most pressing set of investigation remaining is to explore OSNs in "big data" regimes, using the best training criterion available. The results presented here on Aurora 4 (10 hours of data) and open voice search (100 hours of data) using cross-entropy trained models are encouraging, but the performs of OSNs in the scenario of thousands of hours of available training data and sequence training criterion has yet to be explored. So far indications suggest that OSNs are a fruitful generalization of Maxout networks.

6. REFERENCES

[1] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[2] Yajie Miao, Florian Metze, and Shourabh Rawat, "Deep maxout networks for low-resource speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013.

[3] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," *ICASSP*, 2014.

[4] 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.

[5] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[6] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.

[7] Quoc V Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8595–8598.

[8] Urs Köster and Aapo Hyvärinen, "A two-layer ICA-like model estimated by score matching," in *Artificial Neural Networks–ICANN 2007*, pp. 798–807. Springer, 2007.

[9] Aapo Hyvärinen, Jarmo Hurri, and Patrik O Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, vol. 39, Springer, 2009.

[10] Steven Rennie, Vaibhava Goel, and Samuel Thomas, "Annealed dropout training of deep networks," in *Spoken Language Technology (SLT), IEEE Workshop on*. IEEE, 2014.

[11] Arun Narayanan and DeLiang Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2014.

[12] N. Parihar and J. Picone, "Aurora working group: DSP frontend and LVCSR evaluation au/384/02," *Tech. Rep., Inst. for Signal and Information Processing,Mississippi State University*, 2002.

[13] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[14] Hagen Soltau, George Saon, and Brian Kingsbury, "The IBM Attila speech recognition toolkit," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 97–102.

[15] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[16] Hagen Soltau, Hong-Kwang Kuo, Lidia Mangu, George Saon, and Tomás Beran, "Neural network acoustic models for the DARPA RATS program.," in *INTERSPEECH*, 2013, pp. 3092–3096.

[17] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[18] 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.