INVESTIGATION OF MAXOUT NETWORKS FOR SPEECH RECOGNITION

*Pawel Swietojanski**

Centre for Speech Technology Research
University of Edinburgh, UK
p.swietojanski@ed.ac.uk

Jinyu Li, Jui-Ting Huang

Microsoft Corporation
One Microsoft Way, Redmond, WA 98052
{jinyli,jthuang}@microsoft.com

ABSTRACT

We explore the use of maxout neuron in various aspects of acoustic modelling for large vocabulary speech recognition systems; including low-resource scenario and multilingual knowledge transfers. Through the experiments on voice search and short message dictation datasets, we found that maxout networks are around three times faster to train and offer lower or comparable word error rates on several tasks, when compared to the networks with logistic nonlinearity. We also present a detailed study of the maxout unit internal behaviour suggesting the use of different nonlinearities in different layers.

Index Terms— deep neural networks, maxout networks, multi-task learning, low-resource speech recognition

1. INTRODUCTION

Neural network acoustic models (AMs) [1, 2] and its recent resurgence in the form of deep neural networks (DNNs) [3] have been successfully applied across a range of different automatic speech recognition (ASR) tasks. Some contemporary examples include i) initial work on phone classification [4], ii) conversational-style large vocabulary speech recognition (LVSR) systems [5, 6, 7, 8], iii) noise robust applications [9], iv) various aspects of multi– and cross– lingual learning schemes [10, 11, 12, 13, 14] and v) distant and multi-channel LVSR of meetings [15].

All of the above examples share similar feed-forward multi-layer network architectures where each hidden layer implements a linear affine operation followed by an element-wise logistic non-linearity. Indeed, smooth and continuously differentiable activation functions were considered to be a crucial component of training DNNs allowing for a seamless flow of back-propagated gradients and the discovery of highly non-linear features. Recently however, it has been shown experimentally that semi-hard functions which break many of these conventional design mainstays can be not only very accurate but also easy and fast to learn. An example of such activation functions are rectified linear units (ReLU) [16, 17] implementing the lower bounded operation $max(0, x)$. Such a piece-wise linear function causes the network to saturate at 0 only (for $x \leq 0$) enforcing sparse activations and preserving sufficient gradients dynamics on the positive slope to mitigate the vanishing gradients problem in deeper networks [18], and improving convergence.

In the prior work ReLUs in the context of acoustic models have been already studied in several papers. Toth [19] applied recitifying units to phone classification, Dahl et al. [20] investigated ReLUs combined with dropout [21] and a modified variant of restricted Boltzmann machine pre-training on a 50 hour broadcast news transcription task. Zeiler et al. [22] found ReLUs to be useful in training acoustic models on hundreds of hours of speech for very deep networks while Maas et al. [23] proposed a leaky variant of the rectified non-linearity where a small portion of the gradient is allowed to flow through negative activations making them more likely to become active again.

Inspired by the promising application of semi-hard nonlinearities for LVSR in this work we extend the study to *maxout neural networks* (MNN) proposed recently by Goodfellow et al. [24] and evaluated on image processing tasks. MNNs, instead of making a prior assumption about parametric form of non-linearity, try to build it automatically from a number of linear components. While this work was under review two additional papers were published on maxout activation for ASR [25, 26]. As a result, contributions of this work overlap to some extent with one or the other and we will refer to those in text when necessary. An explicit added value of this paper is a detailed analysis of internal maxout behaviour. The remainder of this paper presents the study on using MNNs from an acoustic modelling perspective.

2. (MAXOUT) NEURAL NETWORKS FOR ASR

Context-dependent deep neural network hidden Markov model (CD-DNN-HMM) systems use DNNs to classify the input acoustics into classes corresponding to the HMM tied states. After training, the output of the DNN is an estimate of the posterior probability $P(s|\mathbf{o}_t)$ of each state s given the acoustic observations $\mathbf{o}_t$ at time t. The computation performed in a forward pass of the conventional L-layer network may be summarized as:

$$\begin{aligned} \mathbf{z}^l &= \mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l, \\ \mathbf{h}^l &= f(\mathbf{z}^l), \end{aligned} \quad \text{for } 1 \leq l < L \tag{1}$$

$$\mathbf{v}^L = \mathbf{W}^L \mathbf{h}^{L-1} + \mathbf{b}^L, \tag{2}$$

$$P(s|\mathbf{o}_t) = \frac{\exp\{v^L(s)\}}{\sum_{s'} \exp\{v^L(s')\}}, \tag{3}$$

where $\mathbf{h}^l$ is the input to the $l+1$-th layer, with $\mathbf{h}^0 = \mathbf{o}_t$; $\mathbf{W}^l$ is the matrix of connection weights between $l-1$-th and l-th layers; $\mathbf{b}^l$ is the additive bias vector at the l-th layer; $\mathbf{v}^L$ is the activation at the output layer; and $f(x)$ is an activation function which most often is chosen to be either sigmoid $f(x) = 1/(1+\exp(-x))$, hyperbolic tangent $f(x) = \tanh(x)$ or ReLU $f(x) = \max(0, x)$.

The recogniser uses a pseudo log-likelihood of state s given observation $\mathbf{o}_t$; $\log p(\mathbf{o}_t|s) \propto \log P(s|\mathbf{o}_t) - \log P(s)$, where

*Work performed while interning Microsoft Corporation. The first author would like to thank Yifan Gong for an opportunity and mentoring the internship, Frank Seide for help with DNN training tool and Arnab Ghoshal of University of Edinburgh for discussion on 'Inside a maxout layer' section.

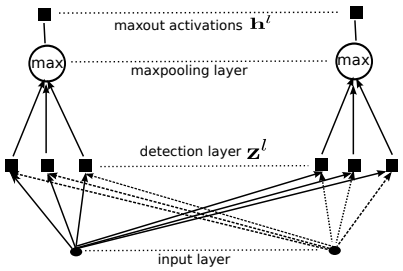

Fig. 1. A scheme of a single maxout layer with pool size $K = 3$. Layers can be stacked on each other to form deeper structures.

$P(s)$ is the prior probability of state s calculated from the training data [1]. We use stochastic gradient descent (SGD) to train DNNs, minimising a negative log posterior probability cost function over the set of training examples $\mathcal{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$; $\theta^* = \arg\min_\theta - \sum_{t=1}^{T} \log P(s_t|\mathbf{o}_t; \theta)$, where s_t is the most likely state at time t obtained by a forced-alignment of the acoustics with the transcript and $\theta = \{\mathbf{W}_1, \ldots, \mathbf{W}_L, \mathbf{b}_1, \ldots, \mathbf{b}_L\}$ is the set of parameters of the network.

The *maxout* network [24] rather than applying any explicit form of non-linearity $f(\cdot)$ as in equation (1), groups the linear activations in detection layer $\mathbf{z}^l$ and passes forward the maximum value within each group (Figure 1) – we will refer to this operation as maxpooling [27]. In mathematical terms, having $\mathbf{z}^l = [z_0^l, z_1^l, \ldots, z_j^l, \ldots, z_{M\times K-1}^l]$ and assuming non-overlapping pools of size K, an i-th maxout unit can be computed by the following formula

$$h_i^l = \max_{k=0}^{K-1}(z_{j+k}^l), \quad \text{where } j = i \cdot K \tag{4}$$

Note, in contrast to standard element-wise non-linearities as in eq. (1), for MNNs when $\mathbf{h}^l$ is in $\mathbb{R}^M$, the underneath $\mathbf{z}^l \in \mathbb{R}^{M\times K}$ and accordingly the trainable layer's parameters are $\mathbf{W}^l \in \mathbb{R}^{M\times K\times D}$ and $\mathbf{b}^l \in \mathbb{R}^{M\times K}$, where D is an input layer dimensionality.

The architecture of a maxout network allows a single neuron to automatically learn a piecewise linear approximation of any convex function, while the groups of maxout neurons can then approximate any continuous function [24]. The activations produced by the neurons are unbounded (unless the neuron learned to do it) so the optimisation process does not suffer from vanishing gradients. Additionally, larger gradients than in sigmoid networks complement dropout training regime [21], where each sub-model selected by different dropout masks contributes more towards the final solution, which may have a positive effect in building acoustic models with limited data. Finally, the maxpooling mechanism sparsifies the gradients which, as argued by Bengio in [28], may be a desirable property from an optimization standpoint – since SGD relies on an invalid assumption that one can modify the parameter θ_i in gradient direction $\frac{\partial C}{\partial \theta_i}$ ignoring contributions introduced to $\frac{\partial C}{\partial \theta_i}$ when other parameters θ_j change, sparse gradients – by zeroing many such θ_j – mitigate this effect.[1]

[1]In maxout networks the partial differential $\frac{\partial h_i}{\partial z_\cdot} = 1$ so gradients flow through all units h regardless of their actual value. This is different from other non-linearities in which the gradient is coupled with the value of h_i by differential form of $f(\cdot)$ i.e. for sigmoid the error signal is multiplied by $h_i(1-h_i)$. Essentially, when h_i fires at 0 (or has been deliberately dropped) the gradient in this unit will be zero. For MNNs we found that better values of optimised cross-entropy objective function were obtained when the gradients for dropped units were 'manually' set to zero. Although it remains unclear whether it is the case of optimisation or more effective model-averaging.

Table 1. WERs (%) for the VS task. The VS test set consists of 31830 words. Notation 1536/2x4 means four maxout layers each having 1536 maxout units and pool size 2. Note: 1 epoch constitutes to one third of the full-sweep over dataset.

System	Test-VS	Convergence
Sigmoid (+RBM)	31.8	21 epochs
Sigmoid (rand)	32.9	23 epochs
Maxout 1536/2x4 + dropout	32.2	12 epochs
Maxout 1536/2x4	32.1	6 epochs
Maxout x3 + Sigmoid x2	31.6	6 epochs

Table 2. WERs (%) for different maxout architectures on VS task.

#maxouts / #pool size	#Hidden Layers			
	2	3	4	5
1536 / 2	33.2	32.3	32.1	32.8
1024 / 3	33.6	32.4	32.3	-
768 / 4	34.4	32.8	32.4	32.3

3. EXPERIMENTS

In this paper all models were trained on 52 dimensional mel frequency cepstral coefficients features (statics + up to third derivatives) which were presented to the network's input in 11-frame context windows (central frame $\pm$ 5 context frames). Unless explicitly stated otherwise, the baseline sigmoid networks were pre-trained in stacked restricted Boltzmann machine fashion (RBM) [29] and had 5 hidden layers with 2048 units in each layer. On the other side maxout models were always randomly initialized and, to make training stable, were further regularized by imposing a maximum norm on each weight vector [30]. Both networks were finetuned with exponentially decaying learning rates controlled by accuracy on a held-out cross-validation set. Dropout (when used) was configured to disable hidden units with probability 0.2. A similar configuration of dropout has been recently found to give good results for speech applications [14]. The number of tied-states is similar across the tasks and is around 1800.

3.1. Maxout networks and LVSR

To evaluate how well MNNs deal with LVSR tasks we use a Microsoft internal American English voice search (VS) dataset comprising around 72 hours of transcribed audio for training purposes. More details regarding VS or baseline systems can be found in [31].

Table 1 presents the results of various DNN AMs. The maxout network performs better than a comparable randomly initialized sigmoid network (0.75% absolute WER reduction). Moreover, it requires less than a third of time to converge (6 epochs versus 23 epochs). As a comparison, the sigmoid network after 6 epochs (2 full sweeps) scored 34.13% WER, which is around absolute 2% worse than the maxout net at the same epoch and in the same ballpark as a discriminatively trained Gaussian mixture model HMM system [31]. Also as expected for larger amounts of training material and model sizes we worked with, the use of dropout did not bring further improvements and MNNs with and without dropout got similar WERs. Note we do not use dropout with sigmoids for larger amounts of data since it has been already reported to be unnecessary [20, 14]. On the other side, the sigmoid model still benefits from RBM pre-training what suggests applying similar technique for MNNs. Since MNNs cannot be initialised using probabilistic models one may use stacked

Table 3. WERs (%) for different mono— and multi— lingual models trained on limited 30-hour partitions of the original training sets. *MT* prefix denotes the models trained in multi-task fashion jointly on all four languages. Results for networks trained on whole partitions are given for comparison. The number of words in the test sets is: DEU 40k, FRA 37k, ITA 31k and ESP 18k.

System	Test-DEU	Test-FRA	Test-ITA	Test-ESP
Sigmoid (all data) [10]	24.08 (195h)	27.16 (138h)	23.66 (93h)	27.97 (63h)
Sigmoid	27.45	29.88	25.58	28.77
Maxout 1536/2x4 + dropout	27.04	29.19	24.78	27.56
MT.Sigmoid	26.12	28.30	24.23	27.81
MT.Maxout 1536/2x4 + dropout	26.07	28.11	24.41	27.71

auto encoders [22, 14]. Finally, we also trained a mixed-type layer variant where we put three maxout layers on the bottom and two sigmoid layers on the top. The major motivation was the network will not suffer from vanishing gradients in lower layers while the logistic non-linearity in top layers will help to break symmetry while training. Interestingly, such a network converged as fast as a pure MNN giving at the same time the best result. This suggests the reason of slow convergence in sigmoid networks is mainly caused by poor learning dynamics in bottom layers.

Table 2 gives more insight into how the architecture of maxout network affects the quality of AMs. We compare the MNNs with different numbers of maxout units and pool sizes keeping the number of detection activations constant. First of all, deeper MNN structures benefit from stacking additional hidden layers while the number of components building a maxout unit seem to be less crucial. Also, one could hope the fact that MNNs are able to better approximate continuous functions will enable shallow networks with less parameters to give good results. However, in a control experiment a 2 hidden layer pre-trained sigmoid network scored 33.7% WER which is similar to the results obtained with MNNs. Another observation is that the gain of adding more layers is more obvious for units of larger pool size, and this aspect will be further investigated in Section 4.

3.2. Multi-task learning and low-resource conditions

The experiments within this section made use of the multilingual short message dictation (SMD) Microsoft dataset. In particular, following [10] we used Spanish (ESP), French (FRA), Italian (ITA), German (DEU) and American English (ENU). The first four languages were used to train the models in multi-task (MT) fashion and due to time constraints were limited to have 30 hours each giving in total 120 hours of speech. The ENU partition, for the purpose of evaluating MNNs in under-resourced and transfer learning conditions, was artificially limited to have 3 hours of training material.

The results of MT experiments are presented in Table 3 where the first row contains the reference WER results on full partitions (the amount of training data ranges from 63 to 195 hours) as reported in [10]. Then in two successive rows we present the corresponding baselines for sigmoid– and maxout– based models trained on 30-hour subsets. Similar to the VS task sigmoid models were initialized with RBMs while maxout networks started from random parameters and MNNs were trained with dropout since in a control experiment on Spanish we were able to obtain a small gain i.e. 27.56% versus 27.74% WER. For all four languages MNN models gave superior results, in particular for ESP data MNNs (27.56%) outperformed an RBM initialized sigmoid network trained on full-partition with twice as much data (27.97%). The last two rows present similar numbers for the models trained in multi-task fashion where the hidden part of the networks is shared and collaboratively learned with other languages. For multi-task learning, maxout and sigmoid networks perform similarly.

Table 4. WERs for 3-hour under-resourced training conditions and multi-lingual knowledge transfers on SMD ENU task. The SMD ENU test set consists of 18k words. Top #L - # of adapted top layers.

System	Test-ENU
Sigmoid	35.87
Sigmoid + dropout	34.43
Maxout 1536/2 + dropout	34.32
SHL Sigmoid + Top 1L	29.87
SHL Sigmoid + Top 2L	29.46
SHL Maxout 1536/2x4 + Top 1L	29.74
SHL Maxout 1536/2x4 + Top 2L	29.33

Table 4 presents the results for 3-hour under-resourced training conditions and multi-lingual knowledge transfers on SMD ENU task. First of all, we can see dropout working for both sigmoid[2] and maxout models reducing WER by more than 1.5% absolute for 3 hours case. Second, MNNs with random initialization gave similar results to sigmoid networks with RBM-style pre-training. Given the observations that dropping neurons and pre-training are complementary to each other under low-resource conditions for both sigmoid [14] and maxout [26] nonlinearities, it suggests the potential of further improvement once the maxout model is pretrained. Finally, we were interested in how well maxout shared hidden layers (SHL) can transfer the knowledge across languages. To investigate this scenario we extracted SHL part of the MT models (Tab. 3), stacked a new logistic regression layer on top and finetuned either 1 or 2 topmost layers using the target low-resource ENU data. Note, since MNNs in MT experiments were finetuned with dropout while DNNs not, to make a fair comparison, both maxout and sigmoid SHL models in table 4 are adapted to the low-resource task without dropout[3]. For both models SHL gave large gains in accuracy when compared to in-domain-only models and, as expected, adapting an additional top hidden layer brought further improvements. We did not retrain all layers since that was found harmful in a recent study [10] and WER as a function of adapted top #layers is expected to be U-shaped.

4. INSIDE A MAXOUT LAYER

It would be interesting to see what happens in the maxout unit internally. For example, by looking at some pool-related statistics we

[2]Note, to make dropout effective for sigmoid we had to increase learning rate 6-fold – from an initial 0.08 to 0.5

[3]Prior to this operation MNNs were transformed to mean models by multiplying the relevant weight matrices $\mathbf{W}$ by $(1 - dropout\ rate)$.

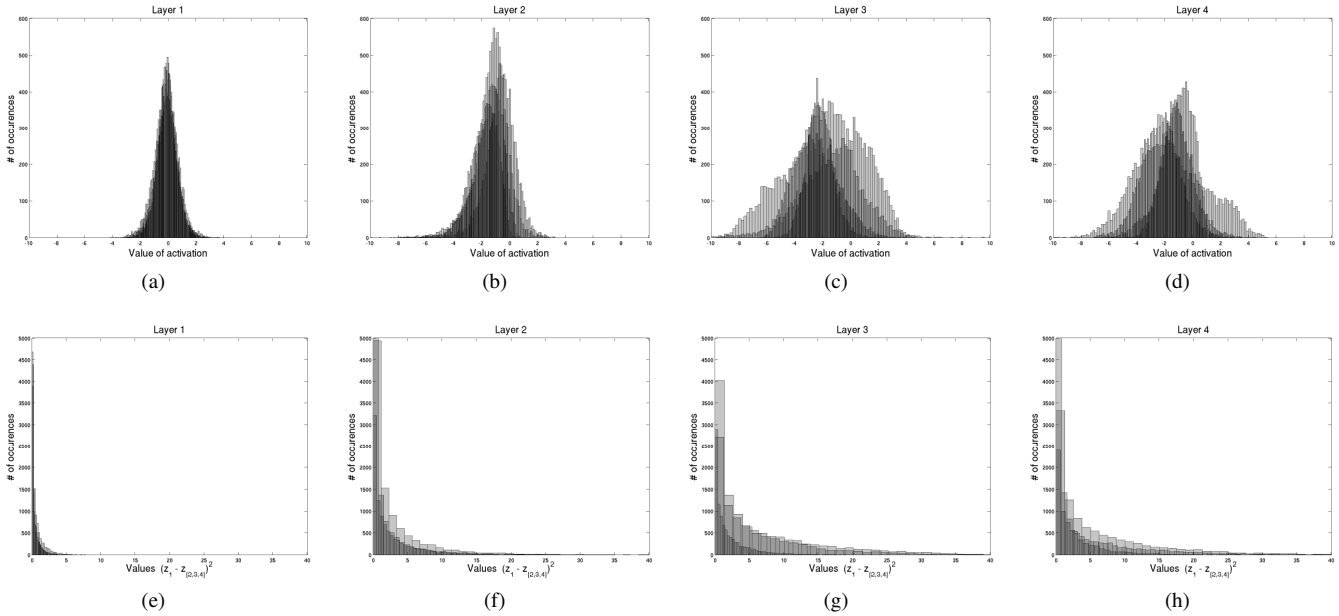

Fig. 2. (Top) Histograms of activations of detection linear units z_1 to z_4 building the first maxout unit. (Bottom) histograms of mean square errors between the first and remaining components within a pool, $(z_1 - z_2)^2$, $(z_1 - z_3)^2$ and $(z_1 - z_4)^2$, obtained in an example-wise fashion.

could say how well the model uses its parameters and if one of detection units $z_{1\ldots K}$ dominates the pool producing the maximum output all the time for any input, an additional parametrisation of $K - 1$ units remains highly redundant. The effect of wasting parameters would also take place if the units within the given pool for the same input computed similar outputs - which means the angle between linear components building a pool is small.

To investigate the above issues we collected the required pool statistics on 10k frames from the VS development set using one of the models (4 hidden layers, 1024 maxout units per layer, pool size K=4) trained on the VS task. Figure 3 shows maxout unit activity (i.e. how often each of the detection units produced the maximum output) in different layers. We can see the bottom layer (Layer 1) distribution is nearly uniform suggesting all its parametrisation was active while in higher layers the units tend to specialize and dominate each other within a pool. However, another question is whether these units approximate sufficiently different functions so that they become invariant to small perturbations in the input. To answer this question we plot another figure 2 (a-d) showing the output value distributions produced by each of the four detection components $z_.$ of the first maxout unit (from the figure 3). The intuition is that the less overlapping distributions the more orthogonal to each other the units z are. And as is clearly visible, in the first layer units produce very similar outputs which become more spread in higher layers. Since these plots show histograms for all 10k frames without focusing on what happens in the pool for the same inputs we draw another set of figures 2 (e-h) of mean square errors between the first component and all remaining ones in a frame-wise fashion. In principle, if components produce similar outputs for the given speech frame the angle between learned functions is small thus the parametrization remains redundant.

The above analyses show that the bottom layers seem to waste a large portion of the additional parametrisation (figure 2 (a,e)) thus could be replaced, for example, by smaller ReLU layers. Similarly, maxout units in higher layers seem to use piecewise-linear components in a more active way suggesting the use of larger pools.

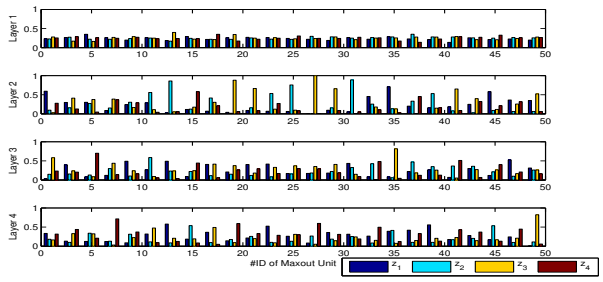

Fig. 3. Normalized histograms of how often each detection unit $z_1, \ldots, z_4$ produced the max output for the first 50 pools (maxout units) in each layer. We draw every 2nd pool to make the plot clearer.

5. CONCLUSIONS AND FUTURE WORK

We studied properties of maxout networks in acoustic modelling applications. We found it superior or equal to the networks with logistic non-linearity in terms of WERs. Moreover, its convergence time is three times faster. It is also worth mentioning the optimised cost and frame accuracies were always better for maxout neurons which suggests superior optimisation properties and encourages the use of sequence training criterion [32]. As expected, maxout units work well with dropout for low-resource conditions. However, dropout is less crucial when the amount of training data increases and the maxout models can be efficiently trained without dropout. Future work could follow the exploration of auto-encoder pre-training schemes for low resource-applications which could provide even bigger gains over pre-trained sigmoid networks. Another interesting direction would be to apply Bayesian hyper-parameters [33] search, for example, to select the right maxout structures (number of maxout units, pool sizes) and possibly tune layer-specific dropout rates [20]. Also, since pooling remains a crucial component of maxout network it would be interesting to see whether differential pooling mechanism, where we can learn pool weights, can bring any gains.

6. REFERENCES

[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[2] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, 1994.

[3] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A-R Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[4] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[5] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *In Proc. IEEE ASRU*, 2011, pp. 24–29.

[6] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012.

[7] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. Interspeech*, 2012.

[8] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[9] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *In Proc. ICASSP*, 2013.

[10] J-T Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, 2013.

[11] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *In Proc. IEEE SLT*, Miami, 2012.

[12] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *In Proc. ICASSP*, 2013.

[13] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *In Proc. ICASSP*, 2013.

[14] Y. Miao and F. Metze, "Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training," in *Interspeech*, 2013.

[15] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Proc. IEEE ASRU*, Dec. 2013.

[16] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *In Proc. ICML*, 2010, pp. 131–136.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Journal of Machine Learning Research*, vol. 15, pp. 315323, 2011.

[18] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[19] L. Toth, "Phone recognition with deep sparse rectifier neural networks," in *In Proc. ICASSP*, 2013.

[20] G.E. Dahl, T.N. Sainath, and G.E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. ICASSP*, 2013.

[21] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[22] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G.E. Hinton, "On rectified linear units for speech processing," in *In Proc. ICASSP*, 2013.

[23] A.L. Maas, A.Y. Hannun, and A.Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013.

[24] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv:1302.4389*, 2013.

[25] M. Cai, Y. Shi, and J. Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*, Dec 2013, pp. 291–296.

[26] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recognition," in *Proc. ASRU*, 2013.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] Y. Bengio, "Deep learning of representations: Looking forward," *CoRR*, vol. abs/1305.0445, 2013.

[29] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[30] T. Mikolov, *Statistical Language Models based on Neural Networks*, Ph.D. thesis, Brno University of Technology, 2012.

[31] J. Li, D. Yu, J. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *In Proc. IEEE SLT*, 2012, pp. 131–136.

[32] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *In Proc. ICASSP*, 2009, pp. 3761–3764.

[33] J. Snoek, H. Larochelle, and R.P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Neural Information Processing Systems*, 2012.