Highway-LSTM and Recurrent Highway Networks for Speech Recognition

Golan Pundak, Tara N. Sainath

Google Inc., New York, NY, USA

{golan, tsainath}@google.com

Abstract

Recently, very deep networks, with as many as hundreds of layers, have shown great success in image classification tasks. One key component that has enabled such deep models is the use of "skip connections", including either residual or highway connections, to alleviate the vanishing and exploding gradient problems. While these connections have been explored for speech, they have mainly been explored for feed-forward networks. Since recurrent structures, such as LSTMs, have produced state-of-the-art results on many of our Voice Search tasks, the goal of this work is to thoroughly investigate different approaches to adding depth to recurrent structures. Specifically, we experiment with novel Highway-LSTM models with bottlenecks skip connections and show that a 10 layer model can outperform a state-of-the-art 5 layer LSTM model with the same number of parameters by 2% relative WER. In addition, we experiment with Recurrent Highway layers and find these to be on par with Highway-LSTM models, when given sufficient depth.
Index Terms: speech recognition, recurrent neural networks, residual networks, highway networks.

1. Introduction

In recent years Long-Short Term Memory Recurrent Neural Networks (LSTMs), [1] have become a popular alternative to feed-forward deep neural networks (DNNs) [2], producing state-of-the-art results in various large vocabulary continuous speech recognition (LVCSR) tasks [3, 4]. An LSTM network with a single layer models a shallow mapping from the input at time t to the output at time t, where depth is achieved by letting an input influence a future output at time greater than t via state variables that persist from one time step to the next. In contrast, feed-forward DNNs achieve deep input-output mapping by directly stacking feedforward transformations. A Deep LSTM follows this approach and stacks LSTM layers to allow depth in immediate input-output mappings. Deep networks suffer from the vanishing and exploding gradients problems, and there are many approaches to alleviate this problem. The focus of this paper is to explore various techniques to build deep recurrent networks.

One approach to address this gradient issue was proposed in [5], which showed that this problem can be solved by introducing skip connections that provide shorter input-output paths, on which the gradient can propagate. On image recognition tasks, the authors found that these deep networks with skip connections can be trained with hundreds of layers, and also outperformed shallow ones. The skip connections used in [5] are named Highway Connections (HW-SKIP), and they require additional parameters in the model. In contrast a successful parameter-less skip connection, known as residual connection (RES-SKIP), was introduced in [6] for an image recognition task. In [6], the skipped layers are convolutional, and the entire architecture is referred to as "ResNet".

An alternative approach to build deep recurrent networks is to use "Recurrent Highway Networks" (RHW) [7]. RHW is a new type of recurrent layer, that allows a deep input-to-state mapping. The authors show superior performance with RHW networks compared to LSTMs on a language modeling task. One novel addition we explore are HW-RHW networks, which are stacked RHW layers with HW-SKIP connection.

It is important to note that there have been a number of speech-related works using skip connections to train deep models, though most work has focused on feed-forward type networks. Xiong et al [8] used ResNet type convolutional network for an LVCSR task (in fact they follow [9]). Lu et al. [10], [11] have used "thin and deep" HW-DNN networks for small-footprint speech recognition tasks, but have not applied this to recurrent networks like LSTMs. In addition, Hsu et al. [12], and Zhang et al. [13] explored the Highway-LSTM-RNN (HLSTM-RNN) which incorporated the HW-SKIP structure directly into the LSTM cell. Finally, Zhang et al. [14] used RES-SKIP connections, where the residual connection was always applied after a feed-forward bottleneck layer.

Thus, while skip connections have been explored for speech, to our knowledge, there has not been a thorough investigation on adding depth to recurrent structures, which are an integral component of our state-of-the-art systems [3, 4]. The goal of this paper is to compare different approaches to adding depth to recurrent networks, by comparing RES-SKIP, HW-SKIP, RHW, and HW-RHW networks.

Our experiments are conducted on a 10,000 hour English Voice Search task. We compare HW-SKIP, RES-SKIP and models without skip connections and find HW-SKIP to outperform the rest, even when using parameter-saving bottlenecks. We show that a 10 layer HW-LSTM model can outperform a state-of-the-art 5 layer LSTM model with the same number of parameters by 2% relative. We also attempt to use highway gates to guide a layer-pruning procedure for HW-LSTM models, albeit with limited success. In addition, we find that HW-RHW models can achieve results that similar to our best HW-LSTM, thus presenting an attractive modeling alternative.

2. Deep recurrent neural networks with skip connections

A single step of a recurrent layer in a neural network at time t can be described by a *transform function* $h^t = H(x^t, s^{t-1})$, where x^t, h^t, s^t are real valued vectors, and a *transition function* that defines a state update $s^{t-1} \Rightarrow s^t$. In this work we are interested in stacking L such layers in a feed forward manner. The output of layer l can then be computed recursively by:

$$h_l^t = H_l(h_{l-1}^t, s_l^{t-1}) \quad (1)$$

Where $h_0^t = x^t$ is the input to the network. The output of the network will be obtained as $y = softmax(h_L)$. We defer the choice of spcific transfer and transition functions to section 2.2.

2.1. Skip Connections

To be able to train deeper networks, we introduce skip connection $skip(x, h)$ that allow information to pass from the lower layer directly to the upper layers by "skipping" intermediate transformations. Skip connections are added by updating Equation 1 to: $h_l^t = skip(h_{l-1}^t, H_l(h_{l-1}^t, s^{t-1}))$.

One popular skip connection is the *residual connection* [6] (RES-SKIP) defined by:

$$skip(x, h) = x + h \tag{2}$$

This connection type does not add parameters to the model and is easy to implement, thus presenting an attractive modeling option. This definition requires that x and h have the same dimension, and we assume this throughtout.

Another type of skip connection, named *highway connection* [5] is obtained by intoducing two gates: a *carry gate*, $C(x)$, that controls how much information flows from the lower layer, and a *transform gate*, $T(x)$, that controls how much information should be read from current layer. With these, the highway connection is defined by:

$$skip(x, h) = h \cdot T(x) + x \cdot C(x) \tag{3}$$
$$T(x) = \sigma(W_T x + b_T) \tag{4}$$
$$C(x) = \sigma(W_C x + b_C) \tag{5}$$

where "$\cdot$" denotes the element-wise product, σ is the sigmoid function, and W_T, b_T, W_C, b_C, are matrices and bias terms that parametrize the connection. This is in sharp contrast with RES-SKIP connection, that doesn't require any additional parameters. To reduce the number of parameters a common modeling choice is to couple the gates by setting $T(x) = \mathbf{1} - C(x)$, where $\mathbf{1}$ is the vector of 1's. Another option, which we explore in this paper for the first time is to enforce a low-rank structure on W_T and W_C by adding a low rank projection P:

$$W_C = PU_C \tag{6}$$
$$W_T = PU_T \tag{7}$$

Where P projects to a dimension which is lower than that of x.

2.2. Recurrent Layers

We now describe two options for the recurrent layers used in our experiments: the LSTM layer and the RHW layer.

2.2.1. LSTM Layer

One popular recurrent layer is the LSTM layer, which we briefely discuss below. For more details see [3]. For each time step t, the LSTM layer takes as input x^t, previous cell state c^{t-1} and previous output h^{t-1}, all real valued vectors, and computes the new cell state, c^t by:

$$\begin{aligned} \hat{x}^t &= [x^t; h^{t-1}] \\ i^t &= \sigma(W_i \hat{x}^t + b_i) \\ f^t &= \sigma(W_f \hat{x}^t + b_f) \\ c^t &= f^t \cdot c^{t-1} + i^t \cdot tanh(W_c \hat{x}^t + b_c) \end{aligned} \tag{8}$$

where i_t and f_t are called the *input* and *forget* gates. In the second step we transform c^t to an output by passing it though a $tanh$ and multiplying with an *output* gate, o^t.

$$\begin{aligned} o^t &= \sigma(W_o \hat{x}^t + b_o) \\ h^t &= o^t \cdot tanh(c^t) \end{aligned} \tag{9}$$

At this work we use coupled-input-forget-gates, as suggested in [15], i.e. instead of equation (8) we set $f^t = \mathbf{1} - i^t$, where $\mathbf{1}$ is the vector of 1's. This formulation has the advantage of bounding the value of the cell state and reducing the number of parameters by tying the input and forget gates.

It is worth noting that RES-LSTM and HW-LSTM are similar in their ability to control the output of a layer, since a RES-SKIP connection (Eq. 2) will sum in this case two gated LSTM outputs (Eq. 9), which is exactly what at HW-SKIP connection does (Eq. 3). Nevertheless the HW-SKIP connection adds more depth, and thus has a potential advantage.

2.2.2. Recurrent Highway Layer

A *Recurrent Highway layer* (RHW) [7] borrows some of its gating mechanism from the LSTM layer, while allowing an arbitrary depth within a layer, called *recurrence depth* (RD). With RD=M, we have M sub-layers in a single RHW layer. We denote the RHW's output by y^t, and the output of sub-layer m by s_m^t. y^t is read from the last sub-layer, thus, $y^t = s_M^t$, and is also fed into the first layer by letting: $s_0^t = y^{t-1}$. The layer is defined recursively for $l = 1, \ldots, M$ by:

$$\begin{aligned} s_m^t &= h_m^t \cdot T_m^t + s_{m-1}^t \cdot C_m^t \\ h_m^t &= tanh(W_H x^t \delta_{1,m} + R_{H_m} s_{m-1}^t + b_{H_m}) \\ T_m^t &= \sigma(W_T x^t \delta_{1,m} + R_{T_m} s_{m-1}^t + b_{T_m}) \\ C_m^t &= \sigma(W_C x^t \delta_{1,m} + R_{C_m} s_{m-1}^t + b_{C_m}) \end{aligned}$$

Where $\delta_{i,j}$ is the Kronecker delta and W_*, b_* are the matrix and bias terms that parametrize the connection. Thus the output of the first sub-layer, s_1^t is obtained from current input x^t and previous output, y^{t-1}, and the rest of the computation is done in a feedforward manner with internal HW-SKIP connections. This formulation shares an arbitrarily deep mapping for the layers transition and transfer functions.

3. Experiments

3.1. Neural Network Architecture

The input and output of our networks closely resemble those used for the LFR models in [16]. The acoustic features used for all experiments are 128-dimensional log-mel filterbank energies computed using a 32ms window every 10ms. At the current frame t, these features are stacked with $l = 3$ frames to the left and downsampled to 30ms, to produce a 512-dimensional feature $x_{t-l} : x_t$. This is the same feature used for all NN experiments in this paper.

The LSTM models used in this work have coupled-input-forget-gates, and peephole connections [15]. Our baseline models consist of 5 layers, beyond that we found training to be unstable due to vanishing gradients. All acoustic models have 8,192 CD-phone output targets [17]. The HW-Skip connections used for the LSTM models used projection to reduce parameters. The first layer of these models is never skipped since the input features have a different dimension than the internal layers width.

The RHW models used in this work have coupled-carry-transform-gates and we have found that non-projected HW-Skip

connections work best for them. The model weights are initialized with values chosen uniformly at random from the range $[-0.2, 0.2]$.

3.2. Training procedure

The models are trained with CE using asynchronous stochastic gradient descent (ASGD) optimization [18]. The models use an existing forced-alignment generated by an existing CLDNN CD-state model [4]. Features are extracted and stacked, and we keep every n-th feature-frame (e.g. n=3 for 30ms) and drop the rest. The LFR models map the CD-states to CD-Phones and subsample by averaging n 1-hot target labels, producing soft targets. All LFR models are trained with a 1-state HMM. During training, recurrent networks are unrolled for 20 time steps for training with truncated back-propagation through time (BPTT) [19].

3.3. Decoding procedure

All acoustic models are decoded with a single pass WFST-based decoder which uses a 100-million n-gram language model and a vocabulary larger than 5 million words. The decoder performs beam search and only keeps 7,000 active arcs at any point in time.

3.4. Data Sets

For our experiments we use a mutli-style training procedure [20], where we first start with clean 15 million utterances (about 12,500 hrs) which are anonymized and hand-transcribed voice search queries, and are representative of Google's voice search traffic. Next, we create a noisier version by adding varying degrees of noise and reverberation at the utterance level, such that overall SNR is between 5dB and 30dB. Samples of noise were taken from YouTube and daily life noisy environmental recordings. Our evaluation set consists of a separate set of about 13,000 utterances (about 11 hours) with similar noise conditions to the training set.

4. Results

4.1. Skip-LSTM networks

Our first set of experiments is concerned with Skip-LSTM networks. First, to assess the effect of skip connections, we compare LSTM, RES-LSTM and HW-LSTM models with 5 layer in Table 1. The LSTM and RES-LSTM models have the same number of parameters. For the HW-LSTM model to have a comparable number of parameters we used projection to 64 dimensions (as in eq. 6, 7). For these shallow models, we find HW-LSTM to be as effective as LSTM, while RES-LSTM lags behind. We conjecture the reason for this is that HW-LSTM has more control over the skip connection than RES-LSTM, thus allows a learnt, input-driven, balance between skipping a layer and using it.

Model Type	Layers	Units	Parameters	WER (%)
DLSTM	5	512	12M	22.4
RES-LSTM	5	512	12M	23.9
HW-LSTM	5	512	12.6M	**22.2**

Table 1: *Comparison of skip types for a 5 layer LSTM model.*

To illustrate this point, we track the change in the transform gates during training. We plot the *mean normalized transform gates gain*, computed by $\hat{t} = \mathbb{E}_x[\frac{T(x)}{C(x)+T(x)}]$ where C and T are defined in eq. (4)-(5), and the expectation is taken over samples and units. $\hat{t}$ is an indicator for how much are the transform gates used. As can be seen in Figure 1, at the beginning of training the (normalized) transform gates, at all layers, have mean value of 0.5, thus give equal weight to both input of the skip-connections. This is similar to having a fixed RES-SKIP connection. However, as training progresses, some layers reach higher values, thus are using mostly the transformed input. The flexibility to tune the usage of each of the inputs is what gives HW-SKIP an advantage over RES-SKIP connections. Based on these observations, the rest of the experiments in this paper will use HW-SKIP and not RES-SKIP.

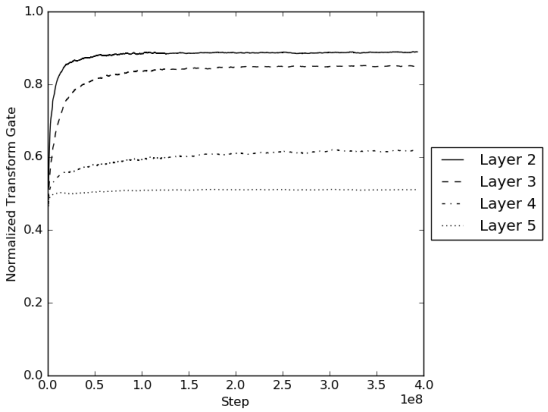

Figure 1: *Mean of normalized transform gate activations of a 5 layer HW-LSTM model thoughout training. Each curve corresponds to a different layer.*

Next, we trained a few models with about 20M parameters and different depths (Table 2). These models were trained with uncoupled HW-SKIP connections with projection as above. When experimenting with LSTMs and have found that it is possible to train up to 5 of the layers without skip connections, and these are the results we report here. When using skip connections (HW-LSTM), we can increase the depth, and the results show that 10 layers is the optimal depth for our task, with a budget of 20M parameters. In addition, note that with increased depth, the HW-LSTM offers improvements over the shallow LSTM with matched parameters.

Model Type	Layers	Units	Parameters	WER (%)
LSTM	5	700	20M	20.4
HW-LSTM	10	512	21.1M	**19.8**
HW-LSTM	14	430	20.4M	20.7
HW-LSTM	18	384	20.7M	21.1

Table 2: *Comparison of models with about 20M parameters.*

To see what happens if we allow more parameters we trained a few models with 36M parameters which is a practical upper bound on the size of the models we serve in our production systems (Table 3). The conclusion remains - up to 10 layers we see gains and beyond that we see degradation. Again, notice the improvement in performance with the HW-LSTM compared to the LSTM with matched parameters.

4.1.1. Pruning skip networks

So far we have found that there is an optimal depth for our task, namely 10 layers. Next, we wanted to check if this depth can be found without an exhaustive search, by pruning a deeper

Model Type	Layers	Units	Parameters	WER (%)
LSTM	5	1,024	36M	18.9
HW-LSTM	10	700	36M	**18.5**
HW-LSTM	14	590	35.6M	19.4
HW-LSTM	18	512	34.5M	19.7

Table 3: *Comparison of models with about 36M parameters.*

model. Our pruning strategy was to remove some layers from a trained model and continue to train the pruned model for additional 50M steps. We tried this with a trained 18 layer model, and found that pruning upto 5 layers produces results which are similar to a model that was trained from scratch, but beyond that results are degraded. For example a 10 layer pruned model was 5% relative worse than a 10 layer model that was trained from scratch. More complex pruning strategies like relaying on gate activations didn't seem to help. We conclude that this method is effective to some extent, but is not reliable enough to detect the optimal depth.

4.2. HW-RHW networks

In our second set of experiments we explore RHW models with coupled-carry-transform gates. For more than a few layers, we found that skip connections are needed, or these models do not train due to instability. We use HW-SKIP connections without projection (i.e. these are HW-RHW models)

Our implementation follows closely that of [7]. As discussed in Section 2.2.2, RHW networks allow depth to be added in two different ways: By adding more recurrent depth (RD) to each layer, or by stacking more layers. Our initial set of experiments, with equal width models, is reported in 4. We see that gains can be obtained by larger RD (up to 16), for a single layer, but stacking 5 layers with RD$=3$ is much better. We attribute this finding to the following observation: A single RHW layer has a single state that captures the temporal dynamics of the speech signal at all the time scales and representation depths. In contrast, multiple RHW layers model a hierarchy of states, each with its own temporal dynamics. We argue this is a better modeling alternative, as "lower" states (e.g. state of background noise) can be separated from high level states (e.g. current cd-phone) and tracking separated states is easier than tracking a single mixed state.

Model Type	Layers	RD	Units	Parameters	WER (%)
RHW	1	4	512	6.8M	31.4
RHW	1	8	512	8.9M	28.2
RHW	1	16	512	13.1M	26.1
RHW	1	20	512	15.2M	26.0
HW-RHW	5	3	512	14.7M	23.5

Table 4: *Comparison of single layer RHW models with different recurrent depth (RD) and a 5 layer HW-RHW model.*

From these initial results we conclude that stacking RHW layers is more effective than larger RD, and in the next section we will explore how to stack the RHW layers.

We used coupled skip-connections without projection and found these train effectively. Attempts to reduce the number of parameters using projections either in the RHW layers or in the skip connections led to unstable training and degraded results. In [7] it is stressed that RHW layers work well due to their flexible Temporal Jacobian. We suspect that adding projection hurts that flexibility.

To further explore the different ways to add depth, we trained a few combinations of width, RD and number of layers, all with the same number of parameters: 33M. Results are presented in Table 5.

We find that RD$=2$ does especially poorly. We attribute this finding to the following observation: A RHW layer with RD$=2$ has the same internal depth as a standard LSTM layer, and roughly the same number of parameters, yet the LSTM layer has a more elaborate gating mechanism. In order to be as effective as an LSTM layer, more internal depth (larger RD) is required. We also find RD$=12$ to be less effective. This strengthens the conclusion from Table 4, which shows that stacking more layers is more effective than large RD.

Model Type	Layers	RD	Units	Parameters	WER (%)
HW-RHW	4	12	520	33M	20.0
HW-RHW	8	2	730	33.2M	21.0
HW-RHW	8	4	580	33.1M	19.1
HW-RHW	8	6	495	32.9M	19.0
HW-RHW	8	8	440	32.7M	19.4
HW-RHW	12	4	475	33M	18.7
HW-RHW	16	4	410	33M	**18.6**
HW-RHW	18	4	388	33M	19.3

Table 5: *Comparison of HW-RHW models with about 33M parameters.*

We conclude this section with a comparison of LSTM, HW-LSTM and HW-RHW models. To have an equal number of parameters (33M) we pruned our best HW-LSTM model to 9 layers as discussed in section 4.1.1 with no lost of accuracy. The two models are compared at Table 6. As can be seen the HW-LSTM-based model outperforms the RHW model, but not by a huge margine. It is worth noting the the HW-LSTM model has been optimized and perfected for more than a decade, and the fact that the RHW model achieves a close result is quite remarkable. More importantly, we can observe that adding depth to the network, either through a HW-LSTM or HW-RHW, is more beneficial than an LSTM with fewer layers and no skip or highway connections.

Model Type	Layers	RD	Units	Parameters	WER (%)
LSTM	5	-	1024	36M	18.9
HW-LSTM	9	-	700	33M	**18.5**
HW-RHW	16	4	410	33M	18.6

Table 6: *Comparison of the best models from each type.*

5. Conclusions

In this paper we have explored different ways to build deep acoustic models (RES-LSTM, HW-LSTM, HW-RHW). We have shown that a HW-LSTM model can be trained with more layers and achieve results which are 2% better than those of plain LSTM models. We also showed that HW-RHW models are only 0.5% off from our best HW-LSTM model, thus present an attractive modeling alternative.

6. Acknowledgements

The authors would like to thank Parisa Haghani, Seungji Lee, Michiel Bacchiani and Erik McDermott for useful discussions.

7. References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

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

[3] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

[4] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*, 2015.

[5] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[7] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, "Recurrent highway networks," *arXiv preprint arXiv:1607.03474*, 2016.

[8] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The microsoft 2016 conversational speech recognition system," *Proc. ICASSP*, 2017.

[9] P. Ghahremani, J. Droppo, and M. L. Seltzer, "Linearly augmented deep neural network," 2016.

[10] L. Lu and S. Renals, "Small-footprint deep neural networks with highway connections for speech recognition," in *Proc. Interspeech*, 2016.

[11] L. Lu, "Sequence training and adaptation of highway deep neural networks," in *Proc. SLT*, 2016.

[12] W. Hsu, Y. Zhang, A. Lee, and J. Glass, "Exploiting depth and highway connections in convolutional recurrent deep neural networks for speech recognition," in *Proc. Interspeech*, 2016.

[13] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory rnns for distant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5755–5759.

[14] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. ICASSP*, 2017.

[15] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, 2015.

[16] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models," in *Proc. Interspeech*, 2016.

[17] A. Senior, H. Sak, and I. Shafran, "Context dependent phone models for LSTM RNN acoustic modelling," 2015, pp. 4585–4589.

[18] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, , and A. Y. Ng, "Large scale distributed deep networks," in *Proc. Advances in NIPS*, 2012.

[19] A. Robinson and F. Fallside, "The utility driven dynamic error propagation network," University of Cambridge, Tech. Rep. CUED/F-INFENG/TR.1, 1987.

[20] R. P. Lippmann, E. A. Martin, and D. B. Paul, "Multi-style training for robust isolated word speech recognition," in *Proc. ICASSP*, 1987.