A Study of the Recurrent Neural Network Encoder-Decoder for Large Vocabulary Speech Recognition

Liang Lu[1]*, Xingxing Zhang*[2]*, Kyunghyun Cho*[3]*, and Steve Renals*[1]

[1]Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
[2]Institute for Language, Cognition and Computation, University of Edinburgh, Edinburgh, UK
[3]Montreal Institute for Learning Algorithms, University of Montreal, Montreal, Canada

{liang.lu, x.zhang, s.renals}@ed.ac.uk, kyunghyun.cho@umontreal.ca

Abstract

Deep neural networks have advanced the state-of-the-art in automatic speech recognition, when combined with hidden Markov models (HMMs). Recently there has been interest in using systems based on recurrent neural networks (RNNs) to perform sequence modelling directly, without the requirement of an HMM superstructure. In this paper, we study the RNN encoder-decoder approach for large vocabulary end-to-end speech recognition, whereby an encoder transforms a sequence of acoustic vectors into a sequence of feature representations, from which a decoder recovers a sequence of words. We investigated this approach on the Switchboard corpus using a training set of around 300 hours of transcribed audio data. Without the use of an explicit language model or pronunciation lexicon, we achieved promising recognition accuracy, demonstrating that this approach warrants further investigation.

Index Terms: end-to-end speech recognition, deep neural networks, recurrent neural networks, encoder-decoder.

1. Introduction

Neural network (NN) based acoustic models have significantly improved the accuracy of automatic speech recognition (ASR) systems [1]. In this framework, a neural network is used to estimate the posterior probabilities of hidden Markov model (HMM) states, which may be transformed into (scaled) likelihoods, replacing the conventional Gaussian mixture model (GMM) state-based pdfs – the hybrid NN/HMM system [2, 3, 4]. These systems have been scaled-up to be both deeper (many hidden layers) and wider (large number of context-dependent outputs), through the availability of increased computational power, in particular general purpose graphical computational units (GPGPUs) [1, 5, 6].

Hybrid NN/HMM approaches have been largely based on feed-forward networks which carry out limited temporal modelling through a finite input context, typically $\pm$3–7 frames. The main sequential modelling is carried out by the HMM superstructure of context-dependent phone models, pronunciation models, and language models. Richer temporal models have been employed in this framework. Feed-forward networks have been replaced by recurrent neural networks (RNNs) which offer potentially infinite context. RNN acoustic models were first developed by Robinson [7], and further developed as a hybrid RNN/HMM system [8, 9]. In such a hybrid system, the RNN is used to provide richer local temporal context – possibly bidirectional [8, 10] – with the temporal structure at the phone level and above provided by the HMM. More recently, state-of-the-art performance was achieved using a long short-term memory (LSTM) RNN/HMM hybrid system [11].

HMMs have been the cornerstone of ASR for several decades [12, 13, 14, 15], providing a coherent temporal modelling approach that is well-matched to the problem of sequential modelling of speech signals. The HMM framework scales elegantly to model speech units at different levels of granularity and can take practical advantage of the underlying theoretical framework of finite automata [16]. Although the HMM itself is a generative model, discriminative training has been developed and used extensively and successfully, regardless of the choice of acoustic model [17, 18, 19, 11]. Nevertheless, HMMs have a number of limitations including the first-order Markov and the conditional independence assumptions, and the difficulty to globally optimise all the system components. Furthermore, HMM-based models usually rely on the availability of pronunciation lexicons for high accuracy, which may cause the development of ASR systems for new languages to be costly.

To address these limitations, a number of alternatives to the HMM sequence model have been investigated [20, 21, 22, 23, 24]; however these approaches did not improve speech recognition accuracy. Recently there has been an interest in the use of RNNs as the main sequence model for speech recognition, enabling direct end-to-end training of an ASR system [25]. This idea has been applied to machine translation [26, 27, 28] and image caption generation [29, 30], as well as to speech recognition [31, 32, 33, 34]. Graves [31] proposed the use of an RNN with a connectionist temporal classifier (CTC) output layer for end-to-end speech recognition. CTC does not rely on a prior alignment between input and output sequences, but integrates over all possible alignment during the model training – this can be interpreted as a specific HMM on the output layer; however the bulk of the sequence modelling is performed by the RNN. This approach was successfully scaled to large vocabulary ASR [34].

In this paper, we follow a similar approach using the RNN encoder-decoder model recently introduced for machine translation [27, 28]. In this approach, an encoder transforms a sequence of input vectors into a sequence of feature representations using RNNs, from which a decoder recovers the corresponding output sequence. Chorowski et al. [33] studied this approach for phoneme recognition using TIMIT database. In this paper, we focus on the large vocabulary case using Switchboard. Unlike [33], we directly use words as output units, removing the dependence on the pronunciation lexicon, and not requiring an implicit HMM at the output layer.

L. Lu and S. Renals are funded by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

2. RNN encoder-decoder

2.1. The decoder

The RNN encoder-decoder is a neural network model that directly computes the conditional probability of the output sequence given the input sequence without assuming a fixed alignment, i.e. $P(y_1, \ldots, y_O|\mathbf{x}_1, \ldots, \mathbf{x}_T)$ where the lengths of the output and the input, O and T respectively, may be different. For speech recognition, the input is usually a sequence of acoustic feature vectors, while the output is usually a sequences of class indices corresponding to units such as phonemes, letters, HMM states, or words. The idea of the encoder-decoder approach is that for each output y_o, the encoder maps the input sequence into a fixed length hidden representation $\mathbf{c}_o$, to which we refer as *context vector* (cf. Section 2.2). From the previous output symbols and the context vector, the decoder computes

$$P(y_1, \ldots, y_O|\mathbf{x}_1, \ldots, \mathbf{x}_T) = \prod_{o=1}^{O} P(y_o|y_1, \ldots, y_{o-1}, \mathbf{c}_o).$$

Since the probability $P(y_o|y_1, \ldots, y_{o-1}, \mathbf{c}_o)$ is conditioned on the previous outputs as well as the context vector, an RNN can be used to compute this probability which implicitly remembers the history using a recurrent layer.

Let $\mathbf{y}_o$ be a vector representation of the output symbol y_o, where $\mathbf{y}_o$ is an one-hot vector indicating one of the words in the vocabulary followed by a neural projection layer for dimension reduction. The posterior probability of y_o is computed as

$$P(y_o|y_1, \ldots, y_{o-1}, \mathbf{c}_o) = g(\mathbf{y}_{o-1}, \mathbf{s}_o, \mathbf{c}_o) \quad (1)$$

$$\mathbf{s}_o = f(\mathbf{y}_{o-1}, \mathbf{s}_{o-1}, \mathbf{c}_o), \quad (2)$$

where $\mathbf{s}_o$ denotes the output of a recurrent hidden layer $f(\cdot)$ with inputs $\mathbf{y}_{o-1}, \mathbf{s}_{o-1}$, and $\mathbf{c}_o$. $g(\cdot)$ is a *softmax* function with inputs $\mathbf{y}_{o-1}, \mathbf{s}_o$, and $\mathbf{c}_o$. We condition both $f(\cdot)$ and $g(\cdot)$ on the context vector to encourage the decoder to be heavily reliant on the context from the encoder. The previous output $\mathbf{y}_{o-1}$ is also fed to the softmax function $g(\cdot)$ to capture the bigram dependency between consecutive words [28]. We have also investigated a simpler output function without the dependence on the previous output $\mathbf{y}_{o-1}$, i.e. $P(y_o|y_1, \ldots, y_{o-1}, \mathbf{c}_o) = g(\mathbf{s}_o, \mathbf{c}_o)$. See Section 4 for further discussion.

2.2. The encoder

As discussed above, the computation of the conditional probability relies on the availability of the context vector $\mathbf{c}_o$ for each output y_o. The context vector is obtained from the encoder which reads the input sequence and generates a continuous-space representation. In [28], the context vector $\mathbf{c}_o$ is obtained by the weighted average of all the hidden representations of a bidirectional RNN (BiRNN) [10]:

$$\mathbf{c}_o = \sum_t \alpha_{ot} \mathbf{h}_t \quad (3)$$

where $\alpha_{ot} \in [0, 1]$ and $\sum_t \alpha_{ot} = 1$; $\mathbf{h}_t = \left(\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t}\right)$ and $\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t}$ denote the hidden representations of $\mathbf{x}_t$ from the forward and backward RNNs respectively. In [26, 27], the context vector $\mathbf{c}_o$ is global, for instance, $\mathbf{c}_o = \mathbf{h}_T$. This means the context vector does not depend on the index o, meaning that the whole input sequence is encoded into a fixed vector representation. This approach has produced state-of-the-art results in machine translation when the dimension of the vector is relatively large [26]. When the model size is relatively small, however, the use of a dynamic context vector – as in Eq. (3) – has been found to be superior, especially for long input sequences [28].

In Eq. (3), the weight α_{ot} is computed by a learned alignment model for each $\mathbf{c}_o$, which is implemented as a neural network such that

$$\alpha_{ot} = \frac{\exp(e_{ot})}{\sum_{t'} \exp(e_{ot'})} \quad (4)$$

$$e_{ot} = a(\mathbf{s}_{o-1}, \mathbf{h}_t) \quad (5)$$

where $a(\cdot)$ is a feedforward neural network that computes the relevance of each hidden representation $\mathbf{h}_t$ with respect to the previous hidden state of RNN decoder $\mathbf{s}_{o-1}$. As in [28], the alignment model is a single-hidden-layer neural network:

$$a(\mathbf{s}_{o-1}, \mathbf{h}_t) = \mathbf{v}^\top \tanh(\mathbf{W}\mathbf{s}_{o-1} + \mathbf{U}\mathbf{h}_t), \quad (6)$$

where $\mathbf{W}$ and $\mathbf{U}$ are weight matrices, and $\mathbf{v}$ is a vector so that the output of $a(\cdot)$ is a scalar. More hidden layers can be used in the alignment model, however, this proved not to be helpful in our experiments (cf. section 4.3).

In the case of using a fixed context vector [26, 27], using an RNN to map the whole input sequence into the context vector is necessary because this vector must represent all the relevant information in the input sequence. In this study, we used the dynamic context vector approach Eq. (3), which averages over all the hidden representations from the input sequence to generate the context vector. In Section 4 we inverstigated whether the encoder RNN could be replaced by a feed-forward network in this case.

2.3. Model training

Given a set of input and output sequence pairs, the model can be trained by maximising the average conditional log-likelihood over all the training set as

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \frac{1}{N} \sum_{n=1}^{N} \log P(y_1^n, \ldots, y_O^n|\mathbf{x}_1^n, \ldots, \mathbf{x}_T^n, \mathcal{M})$$

where $\mathcal{M}$ denotes the set of model parameters, and N is the number of training utterances. Since all the functions used in the encoder and decoder are differentiable, we can use stochastic gradient decent (SGD) to train the model. As in [28], we used the Adadelta algorithm [35] to automatically estimate the learning rates, instead of empirical tuning them. In our internal experiments on a small scale problem, we found that Adadelta performed comparable to the well tuned exponential scheduling approach [36] with momentum. However, the latter is more cumbersome in tuning the hyperparamters. Refer to [28] for more details on the model training of an encoder-decoder.

3. Experimental Setup

3.1. Encoder types

We have investigated three different types of encoder for speech recognition (Figure 1). The first approach is referred to the tied BiRNN encoder, in which the forward and backward RNNs read the same feature representations from a single feedforward neural network. In the untied BiRNN encoder approach, the forward and backward RNNs use separate feedforward neural networks which are expected to learn complimentary feature extractions. In both cases, we use bi-directional RNNs which have

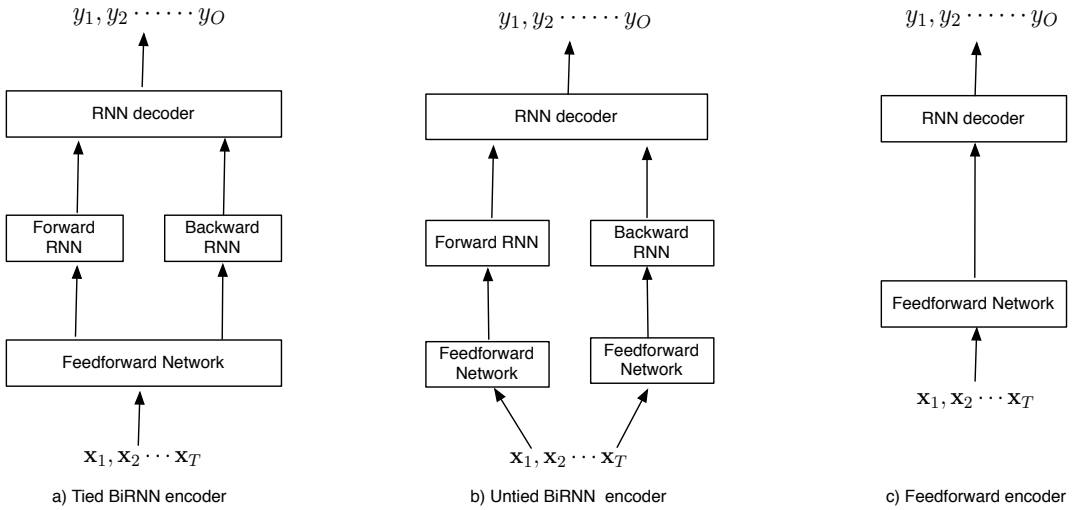

Figure 1: The three types of encoders studied in this paper. a) Tied BiRNN encoder: the forward and backward RNN use the same feedforward network for feature extraction. b) Untied BiRNN encoder: the forward and backward RNN use separate feedforward network for feature extraction. c) Feedforward encoder: the RNN is not used in the encoder, and decoder directly reads features from the feedforward network.

been shown to be superior to a unidirectional RNN [32]. Finally, we explore a feedforward encoder approach, which does not use an encoder RNN, with the decoder directly reading the features from the feedforward neural network. As discussed above, the aim is to validate whether using a RNN in the encoder is necessary if we use the dynamic context vector approach. It is difficult for conventional RNNs with a simple activation function, such as `tanh`, to capture long-term dependencies in sequences due to the problem of vanishing and exploding gradient [37]. In this work, we have used the gated recurrent unit [27] to mitigate this problem, which is similar to the long short-term memory unit [38].

3.2. Downsampling

Unlike machine translation where the input sequences are usually relative short (less than one hundred words), speech recognition uses input sequences of acoustic vectors, whose length can be hundreds or thousands of frames. Modelling such long sequences is challenging to RNNs even with gated recurrent units, especially due to high computational requirement. We explore an approach of downsampling the input sequence at the low-level acoustic features $\mathbf{x}_t$ and at the level of encoder representations $\mathbf{h}_t$. The latter approach is similar to the "striding" approach in [34]. The advantage of downsampling the output of the recurrent hidden layer, is that the history learned by the recurrent layer is not down sampled, and thus reducing the information loss.

Downsampling input sequences can also significantly speedup model training since the unrolled RNNs will be less deep, and it reduces the cost for the alignment model to compute the context vector and the relevance scores as Eq. (3) - (5). In addition, downsampling significantly reduces the memory requirement, as the number of encoded representations $\mathbf{h}_t$ is directly proportional to the length of the input sequence. This is especially important when we use GPUs which often have limited on-board memory. In our experiments, training without downsampling resulted in a small maximum usable minibatch size (10), while downsampling enables the use of a larger minibatch size to speedup the training. For example, training our model without downsampling for 15 epochs took more than 2 weeks, where as downsampling by a factor of 3 reduced the training time to be less than 5 days.

Table 1: WERs (%) of using the three encoder types. `#layers` indicates the number of hidden layers in the feedforward neural network. Using more hidden layers for feedforward encoder increased WER, so the results are not presented here.

Encoder	#layers	CHM	SWB	Avg
Tied BiRNN	1	70.5	55.0	62.8
Tied BiRNN	2	67.3	51.3	59.3
Tied BiRNN	3	68.1	54.0	61.1
Untied BiRNN	1	61.3	40.8	51.1
Untied BiRNN	2	60.5	41.2	50.9
Untied BiRNN	3	67.7	46.2	57.0
FeedForward	1	93.2	86.5	89.9

3.3. System details

We report results using the Switchboard corpus of [39], with a 300 hour training set. We show separate results for the Callhome (CHM) and Switchboard (SWB) evaluation sets. We used 39 dimensional mel frequency cepstral coefficient (MFCC) acoustic features, with cepstral mean normalisation (CMN) on per utterance basis. We then spliced the MFCCs using a left and right context window of 5 frames (i.e. ± 5) unless specified otherwise. As mentioned before, for the output representation $\mathbf{y}_o$, we mapped each word into a one hot vector whose dimension is the size of the vocabulary (29,877 including the unknown and end of sentence symbols). This high-dimensional, but sparse vector is linearly projected into a 500-dimensional continuous vector, and the projection matrix (or embedding matrix) is estimated along with all the other parameters, starting from random initialisation. Using words as output units can remove the dependency on the pronunciation lexicon, but can increase the problem of out-of-vocabulary words, as the number of words in the training set is more limited. The dimension of the hidden layers in the feedforward neural network as well as each encoder RNN is 1000, and therefore, the dimension of $\mathbf{h}_t$ is 2000

Table 2: The results of using different downsampling steps. `step=1` means no downsampling, while `step=2` means we take 1 frame in every 2 frames.

Step	Splicing	Space	CHM	SWB	Avg
1	± 5	feature	62.7	47.6	55.2
2	± 5	feature	61.3	40.8	51.1
3	± 5	feature	59.9	**38.8**	**49.4**
4	± 5	feature	60.2	41.7	51.0
1	± 7	feature	65.5	47.6	56.6
2	± 7	feature	59.9	41.7	50.9
3	± 7	feature	59.8	40.3	50.1
4	± 7	feature	60.0	43.0	51.6
3	± 5	hidden	60.7	42.3	51.5
3	± 5	hidden	**58.9**	41.7	50.3

for the BiRNN encoders. The number of training utterances in Switchboard is 192,701, and training each model until convergence took approximately 15 epochs.

4. Results and Discussion

4.1. Results of different encoders

Table 1 shows the word error rates (WERs) of three different types of encoders. In these experiments, we downsampled the input sequence at the feature level $\mathbf{x}_t$ by the factor of 2. In other words, we took only one frame in every two frames (cf. section 4.2). Our results indicate that using RNN in the encoder is essential for speech recognition, even with the dynamic context vector approach. This may be explained that since the encoder-decoder approach does not use an explicit alignment between the input and output sequences, the hidden representation $\mathbf{h}_t$ from an RNN is more robust to the time space distortions because it can capture the information of the long context — unlike the feedforward encoder. We also observed that the untied BiRNN encoder that use separate feedforward neural network for feature extraction achieved much low WERs compared to the tied BiRNN encoder. This means that using separate feedforward networks can learn complimentary feature extractions for BiRNNs. However, using *deep* feedforward neural network did not make much difference for both tied and untied BiRNN encoders. This may be due to a larger parameter space of the model which makes learning more difficult, or due to the fact that the encoder-decoder model already performs sophisticated feature extraction with deep temporal processing.

4.2. Results of downsampling

Table 2 shows the results of using different downsampling steps. In these experiments, we only used one hidden layer in the feedforward neural networks, and we used the untied BiRNN encoder. We see that discarding frames in feature space can in fact improve recognition accuracy. This may mean that using short sequences can improve the modelling accuracy of RNNs, which is more important than providing all the information from all the frames. We also compare downsampling in the feature space of $\mathbf{x}_t$ to the hidden representation space of $\mathbf{h}_t$, expecting that downsampling $\mathbf{h}_t$ would work better because the RNNs have seen all the frames and, therefore, they can minimise the loss of information from downsampling. However, our results show that the two approaches achieve comparable results. In the future, we shall investigate more intelligent approaches to reduce the length of input sequences, for instance, by taking convolutions in time.

Table 3: Results of using deep alignment and implicit bigram model. `#layers` indicates the layers in the alignment model in Eq. (6).

#layers	bigram	CHM	SWB	Avg
1	$\surd$	59.9	38.8	49.4
2	$\surd$	60.6	40.8	50.8
1	$\times$	59.0	40.4	49.7

4.3. Deep alignment and implicit bigram models

In the previous experiments, we have used just one hidden layer in the alignment model $a(\cdot)$ as in Eq. (6). We tried to use more hidden layers in $a(\cdot)$ to make the alignment model deeper and more powerful. However, this did not improve recognition accuracy in our experiments as shown in Table 3. In [33], the authors introduced a gate function in the alignment model which learns a soft window to encourage the alignment model to search nearby along the previous hidden state. Our preliminary experiments using this approach have not been successful, which calls for further investigation. As mentioned in section 2.1, the output softmax function $g(\cdot)$ takes the previous output representation $\mathbf{y}_{o-1}$ to predict $\mathbf{y}_o$, to which we refer as the implicit bigram model. In our experiments, we found that this is not necessary. Using a simpler output function $g(\mathbf{s}_o, \mathbf{c}_o)$ instead of $g(\mathbf{y}_{o-1}, \mathbf{s}_o, \mathbf{c}_o)$ achieves comparable results as shown in Table 3, and even slightly speed up the model training.

4.4. Future work

In the future, we shall investigate the use of language model to improve the recognition accuracy. In the CTC-based end-to-end speech recognition system [34], the authors obtained much better results (CHM: 31.8, SWB: 20% and Avg: 25.9%) on the Switchboard task using 4-gram language model trained on both Fisher and Switchboard transcriptions, while on the WSJ corpus, the WER obtained without language model is much higher than that using a bigram language (35.8% vs. 14.1%) [32]. One potential approach to incorporating a language model into the encoder-decoder model was recently proposed in [40] in the context of machine translation. Furthermore, we shall also study improved approaches to shorten the long input sequences rather than the simple downsampling approach, such as temporal convolutional neural networks (related to time-delay neural networks [41]). Finally, regularisation including dropout shall also be investigated, especially when training large models.

5. Conclusions

In this paper, we study the application of an RNN encoder-decoder model for large vocabulary end-to-end speech recognition. Without using any language model or pronunciation lexicon, we have obtained encouraging recognition accuracy on the Switchboard corpus. Our experiments show that using RNNs in the encoder is essential for the success of this model architecture, and using separate feedforward neural networks for feature extraction in the encoder can reduce the word error rate. Reducing the lengths of input sequences can make the RNN training easier which improves the recognition accuracy, and is important to speedup the training process. Future work includes the use of language models, other model architectures of the encoder-decoder, and exploring other machine learning algorithms to map the long sequences into short ones, as well as dropout for model regularisation.

6. References

[1] 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," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[2] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994, vol. 247.

[3] N. Morgan and H. A. Bourlard, "Neural networks for statistical recognition of continuous speech," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 742–772, 1995.

[4] 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.

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

[6] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Interspeech*, 2011, pp. 437–440.

[7] T. Robinson and F. Fallside, "A recurrent error propagation network speech recognition system," *Computer Speech and Language*, vol. 5, pp. 259–274, 1991.

[8] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent networks in continuous speech recognition," in *Automatic Speech and Speaker Recognition – Advanced Topics*, C. Lee, K. Paliwal, and F. Soong, Eds. Kluwer Academic Publishers, 1996, pp. 233–258.

[9] A. Robinson, G. Cook, D. Ellis, E. Fosler-Lussier, S. Renals, and D. Williams, "Connectionist speech recognition of broadcast news," *Speech Communication*, vol. 37, pp. 27–45, 2002.

[10] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 2673–2681, 1997.

[11] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Proc. INTERSPEECH*, 2014.

[12] J. Baker, "The DRAGON system – an overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, pp. 24–29, 1975.

[13] L. Bahl, F. Jelinek, and R. Mercer, "A maximum likelihood approach to speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 179–190, 1983.

[14] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[15] J. A. Bilmes, "What HMMs can do," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 3, pp. 869–891, 2006.

[16] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, pp. 69–88, 2002.

[17] D. Povey, "Discriminative training for large vocabulary speech recognition," *Ph.D thesis Cambridge, UK: Cambridge University*, 2004.

[18] 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 *INTERSPEECH*, 2012.

[19] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013.

[20] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Transactions on Speech and Audio Processing*, pp. 360–378, 1996.

[21] N. Smith and M. Gales, "Speech recognition using SVMs," in *Advances in neural information processing systems*, 2001, pp. 1197–1204.

[22] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification." in *INTERSPEECH*, 2005, pp. 1117–1120.

[23] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template-based continuous speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1377–1390, 2007.

[24] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 2, pp. 354–365, 2009.

[25] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Pro. EMNLP*, 2014.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[29] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *arXiv preprint arXiv:1411.4555*, 2014.

[30] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, 2015.

[31] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[32] A. L. Maas, A. Y. Hannun, D. Jurafsky, and A. Y. Ng, "First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs," *arXiv preprint arXiv:1408.2873*, 2014.

[33] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results," *arXiv preprint arXiv:1412.1602*, 2014.

[34] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger *et al.*, "Deep Speech: Scaling up end-to-end speech recognition," in *arXiv preprint arXiv:1412.5567*, 2014.

[35] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[36] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 6724–6728.

[37] 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.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*. IEEE, 1992, pp. 517–520.

[40] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[41] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, 1989.