Sound Signal Processing with Seq2Tree Network

Weicheng Ma†, Kai Cao‡, Zhaoheng Ni◇, Peter Chin†, Xiang Li‡
†Boston University
‡Cambia Health Solutions
◇City University of New York
†{wcma, spchin}@bu.edu, ◇zni@gradcenter.cuny.edu
‡{kai.cao, xiang.li}@cambiahealth.com

Abstract

Long Short-Term Memory (LSTM) and its variants have been the standard solution to sequential data processing tasks because of their ability to preserve previous information weighted on distance. This feature provides the LSTM family with additional information in predictions, compared to regular Recurrent Neural Networks (RNNs) and Bag-of-Words (BOW) models. In other words, LSTM networks assume the data to be chain-structured. The longer the distance between two data points, the less related the data points are. However, this is usually not the case for real multimedia signals including text, sound and music. In real data, this chain-structured dependency exists only across meaningful groups of data units but not over single units directly. For example, in a prediction task over sound signals, a meaningful word could give a strong hint to its following word as a whole but not the first phoneme of that word. This undermines the ability of LSTM networks in modeling multimedia data, which is pattern-rich. In this paper we take advantage of Seq2Tree network, a dynamically extensible tree-structured neural network architecture which helps solve the problem LSTM networks face in sound signal processing tasks—the unbalanced connections among data units inside and outside semantic groups. Experiments show that Seq2Tree network outperforms the state-of-the-art Bidirectional LSTM (BLSTM) model on a signal and noise separation task (CHiME Speech Separation and Recognition Challenge).

Keywords: Deep Learning, Tree Model, Dynamic Neural Network

1. Introduction

Recent RNN-based approaches are achieving high performance in speech processing tasks, including but not limited to the signal and noise separation task (Erdogan et al., 2015; Zhu and Vogel-Heuser, 2014; Wu et al., 2015; Barker et al., 2015). The underlying hypothesis is that the energy in each frequency bin over a period of time is continuous and predictable. However, in real life scenes noises can break in at any time and intertwine with the sound signal with no predictable pattern, which undermines these models' ability to predict the distribution of noise over frequency bins.

To address the problem of finding correct boundaries of noises, some variants of the original LSTM network are used. The current state-of-the-art system on this task applies BLSTM, which tries to bound noises by foreseeing future information (Erdogan et al., 2015; Weninger et al., 2015; Grais et al., 2014). Nevertheless, information from the future also contains more distant sound signals, which does not solve the signal superposition problem. Furthermore, we believe the future for speech processing should be dominated by real-time speech processing techniques, which BLSTM models are not able to handle.

What's more important, phonemes in sound signals make no sense if not combined into "words", which are not found in noise signals. So, the sound waveforms should not be understood as a chain of phonemes, but rather on a "word" level. This leads to a natural choice of tree structured modeling of the waveforms.

In this paper we introduce two variants of Seq2Tree (Ma et al., 2018b; Ma et al., 2018a), a novel architecture which extends LSTM networks to be able to parse sequential input into a tree structure and show its superiority in decomposing sound and noise signals. Seq2tree network architecture differs from the standard LSTM since each node inherits the hidden state not from the previous state in time sequence but from its parent in the tree structure, based on its position predicted by the network itself.

Our evaluation demonstrates the advancement of Seq2Tree network compared to the BLSTM baseline on the signal and noise separation task (Barker et al., 2015). Experiments show that our system shows comparable performance to the BLSTM implementation, while outperforming it in more complex scenarios. Further optimization and adjustment to this task will follow.

2. LSTM Network

RNNs have the advantage of processing input sequences regardless of their lengths. The network reads an element in a sequence at a time and passes it to an activation function recursively together with the current state of this network. The sequence and elements can be of arbitrary types—for example, phonemes in a piece of sound when it comes to the task of speech processing.

Generally the input elements are represented as vectors, and the state at a certain time t is a distributed representation with a preset dimension d. Based on the recurrent nature of RNNs, the state at time t stores the information from all the states before time t. The commonly accepted activation functions in RNNs are often an affine transformation of the previous state h_{t-1} and the current input x_t combined with a non-linear function σ:

$$h_t = \sigma(Wx_t + Uh_{t-1} + b). \qquad (1)$$

Though RNNs are designed to store previous information, they are easily trapped by the explosive growth or rapid vanishment of the gradient over long distances (Hochreiter, 1998; Hochreiter et al., 2001). This makes it difficult for RNNs to represent long-term information.

The LSTM network is introduced to deal with this problem (Hochreiter, 1998; Hochreiter et al., 2001; Zaremba and Sutskever, 2014; Zaremba et al., 2014). Different from directly passing the previous state and the current input to the transition function on which the gradient is calculated, LSTM uses a memory cell to preserve the longer-term information. Using the settings in (Zaremba and Sutskever, 2014; Zaremba et al., 2014), the LSTM transition functions are as follows:

$$\begin{aligned} i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}), \\ f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}), \\ o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}), \\ u_t &= tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}), \\ c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\ h_t &= u_t \odot tanh(c_t) \end{aligned} \tag{2}$$

where i_t, f_t, o_t, c_t are the input gate, forget gate, output gate and the memory cell, respectively, and $\odot$ refers to element-wise multiplication. In the equations, the input gate decides how much information from the new input will be added to the memory cell. Similarly, the forget gate f controls how much information to forget from the previous states, and the output gate limits the amount of information to expose. By balancing the incoming and outgoing information amount, LSTM is able to prevent the gradient vanishment and explosion problems.

Ordinary LSTM is based on chain-structured sequences. There exist two common variants of LSTM networks by structure, namely BLSTM and Multilayer LSTM, which combines multiple LSTM networks together to provide additional information in the prediction at each time step. Tree LSTM (Tai et al., 2015) could be regarded as one variation of Multilayer LSTM with the dependency relation reversed.

3. Seq2Tree Network

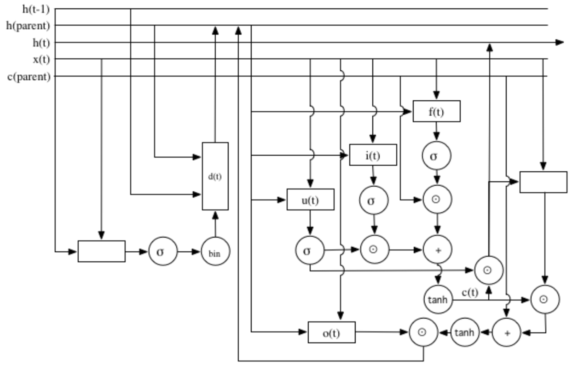

Figure 1: Seq2Tree Network Architecture

The LSTM architectures described in the previous section all have limits in constructing a tree structure from sequential input. Though Multilayer LSTM and Tree LSTM networks are able to maintain multilevel dependencies, Multilayer LSTM exposes children cells to all the other units, and Tree LSTM requires tree-structured input. These characteristics limit their use in speech processing tasks where no reliable parser exists, especially in the case of online speech processing. Hence we in this paper apply Seq2Tree network, a dynamic tree-structured neural network architecture we developed, on sound signal processing tasks. Because of the structural characteristics of sound signal data, we in this paper introduce two variants to the general idea of Seq2Tree network—Single Level Seq2Tree and Multilayer Seq2Tree architectures. Both variants are able to find dependencies from adjacent signals, while the Multilayer Seq2Tree architecture catches deeper, weaker-bounded correlations.

Similar to original LSTM networks, at each time step our Seq2Tree architecture passes information from a preceding state with a hidden unit h_t, accepts new information from the input x_t gated by an input gate i_t, controls the output by an output gate o_t, drops unimportant data in an amount decided by the forget gate f_t, and keeps long-term information from the beginning of the input sequence in a memory cell c_t. The difference is that instead of taking the previous state as the preceding state, Seq2Tree networks use one additional direction gate d_t to choose the direction to go at time step t. The path selection gate is implemented differently in Single Level Seq2Tree and Multilayer Seq2Tree architectures.

3.1. Single Level Seq2Tree

The Single Level Seq2Tree architecture allows at most a depth of 1 for all the nodes in the generated tree structure. It is based on a simplified hypothesis that children states under a parent state do not affect outer units. On speech processing tasks, for example, this means no two signals overlap each other if they are not within the exact same phase.

Since the height of the tree is limited to 2, at each step there exist only 2 possible directions to go: up and down. The strategy is that if the predicted direction to go is “up”, the parent node’s hidden state becomes the input hidden state h_{t-1} and the parent hidden state is assigned the hidden state of the current input after processing it. If the direction is “down” while the previous state is already a child node, the new node becomes the sibling of the former one and inherits the hidden state from the previous state. Otherwise, the new unit takes the hidden state from its previous neighbor and becomes the child of its preceding state.

After processing each state, its parent node’s information is updated. The forget gate of the child state f_t controls the amount of change to give its parent state. This mechanism is inspired by the Tree LSTM networks. The transition functions of the Single Level Seq2Tree network are as follows:

$$\begin{aligned} d_t &= bin(\sigma(W^{(d)}x_t + U^{(d)}h_{t-1} + b^{(d)})), \\ h_{parent} &= d_t \begin{pmatrix} h_{parent} \\ h_{t-1} \end{pmatrix}, \\ i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{parent} + b^{(i)}), \end{aligned}$$

$$
\begin{aligned}
f_t &= \sigma(W^{(f)}x_t + U^{(f)}(d_t \begin{pmatrix} \mathbf{0} \\ h_{parent} \end{pmatrix}) + b^{(f)}), \\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{parent} + b^{(o)}), \\
u_t &= tanh(W^{(u)}x_t + U^{(u)}h_{parent} + b^{(u)}), \\
c_t &= i_t \odot u_t + f_t \odot c_{parent}, \\
h_t &= u_t \odot tanh(c_t), \\
\Delta f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_t + b^{(f)}), \\
c_{parent} &= c_{parent} + \Delta f_t \odot c_t, \\
h_{parent} &= o_{parent} \odot tanh(c_{parent}).
\end{aligned} \tag{3}
$$

where bin denotes a binary threshold activation function, σ represents the sigmoid function, and $\odot$ is element-wise multiplication.

3.2. Multilayer Seq2Tree

The Single Level Seq2Tree architecture can perfectly model the superposition of signals without an overlap of three or more signals with different phases. However, in speech processing tasks the boundaries of noise signals are not necessarily distinct from each other.

To model the more complicated scenarios, a deeper tree structure is needed so that when noises overlap with each other, the layer $l+1$ represents phonemes which come before the $l-th$ sound waveform ends. This architecture is an extension to the Single Level Seq2Tree architecture at the point that at each time step, there exist three directions instead of two. Multiple jumps towards the root in one time step is also allowed. Moreover, at each jump an update gate is used to control the amount of change on a parent layer, and the remainder is passed to even higher states in the tree if there are further jumps. The transition functions differ from those of Single Level Seq2Tree on parent state selection and parent state update mechanisms:

$$
\begin{aligned}
d_{kt} &= bin(\sigma(W^{(d)}x_t + U^{(d)}h_k + b^{(d)})), \\
h_{parent} &= \prod_{k=0}^{d_{kt} \neq \binom{0}{0}} d_{kt} \begin{pmatrix} h_{t-k-1} \\ h_{t-k} \end{pmatrix}, \\
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{parent} + b^{(i)}), \\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}(d_{0t} \begin{pmatrix} \mathbf{0} \\ h_{parent} \end{pmatrix}) + b^{(f)}), \\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{parent} + b^{(o)}), \\
u_t &= tanh(W^{(u)}x_t + U^{(u)}h_{parent} + b^{(u)}), \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\
h_t &= u_t \odot tanh(c_t), \\
\Delta f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_t + b^{(f)}), \\
\Delta c_t &= \Delta f_t \odot c_t, \\
c_{kt} &= c_{kt} + \Delta c_t - \sum_{i=0}^{k-1} u_i \Delta c_t, \\
h_{kt} &= o_{kt} \odot tanh(c_{kt}).
\end{aligned} \tag{4}
$$

where kt indexes the parent nodes in the path until the root from the node at time step t, and d_{0t} represents the gate d at the current time step under the selected parent node.

4. Task and Model

4.1. Signal and Noise Separation Task

We test our Seq2Tree architecture on the signal and noise separation task, the goal of which is to predict a mask which weakens the energy of noise when applied to the input sound. The task is defined in the Second CHiME Speech Separation and Recognition Challenge (Vincent et al., 2013).

4.2. Tree2Seq Signal-Masking Model

For this task, at each time step t we want to predict a mask over all frequency bins. We achieve this by training a softmax regression matrix which takes the current hidden state:

$$mask = softmax(U^{(R)}h)$$

where $U^{(R)}$ is the regression matrix.

We train our signal-masking model in two stages using two different loss calculations, as is suggested in (Weninger et al., 2014; Erdogan et al., 2015). The two losses we applied are:

$$
\begin{aligned}
J_1(t) &= -\frac{1}{c}\sum_{i=1}^{c}(mask_i - label_i)^2 \\
J_2(t) &= -\frac{1}{c}\sum_{i=1}^{c}(\|x_t\|(mask_i - label_i))^2
\end{aligned}
$$

where c is the number of frequency bins, $mask_i$ is the predicted mask at time t for bin i and $label_i$ is the labeled mask on bin i at time t.

5. Experiments

We evaluate our Seq2Tree architecture on the signal and noise separation task. The data is a fraction of 1500 audio files from the CHiME dataset (Vincent et al., 2016), in which 10% is used for test and the rest for training. Each input file is Fourier Transformed and fed to the models. Every model predicts a mask given the input matrix. The quality of the mask is evaluated in terms of Overall Perceptual Score (OPS) by applying the mask onto the source waveform, given the noise-removed audio gold standard (Emiya et al., 2011). In our experiment, the shape of the training data is 50×513, representing the energy at 50 time steps in 513 frequency bins. The test data has variable length over time steps, taking advantage of LSTM models' ability to deal with variable length inputs.

We compare the results generated by our Single Level Seq2Tree with those output by the BLSTM baseline. The hidden layer size for our Seq2Tree network is set to 1024, and we list the results with different numbers of iterations. The BLSTM baseline applies a 256 hidden layer size, and is trained for 30 epochs. Due to long training time cost, our Multilayer Seq2Tree model for this task is only trained for 20 runs with the same parameter settings as our Single Level Seq2Tree model. Best and worst scores of our Single Level model are also included.

As is shown in the results table, our Single Level Seq2Tree model has comparable performance to the BLSTM implementation. The accuracy increases with more training iterations, indicating a preference of more training data and

Implementation	OPS(dB)
BLSTM	25.01
Single Level Seq2Tree	25.13
Single Level Seq2Tree (Worst Case)	23.87
Single Level Seq2Tree (Best Case)	27.96
Multilayer Seq2Tree	24.41

Table 1: Evaluation Results.

more training epochs. Also when looked into the specific WAV files, in more complex cases where noises overlap with each other, our Multilayer Seq2Tree model largely outperformed the other models, which agrees with our estimation. Further experiments are needed to demonstrate the effectiveness of the Multilayer Seq2Tree architecture on the noise separation task.

6. Conclusion & Future Work

In this paper, we introduced a generative tree-structured LSTM network architecture. The Seq2Tree architecture can be applied to arbitrary sequential input with potential local dependencies among nodes. We demonstrated its effectiveness by evaluating two Seq2Tree-based models on a signal and noise separation task. However, due to time constraints we are only able to thoroughly study the performance of the Single Level Seq2Tree architecture. Our results are comparable to the current state-of-the-art model in this task, though leaving some minor errors indicating a preference to the multilayer tree structure and the need for more careful parameter tuning. We will keep refining our model to fit the noise separation task, and we will try to expand the use of our Seq2Tree architecture to other tasks. Since deep learning model have widely been used in AI tasks, we propose that the seq2tree model can be used in different NLP tasks, such as NLP tasks and multimedia tasks. Syntactic structures have been implemented with deep neural networks and applied to build tree-strutured LSTMs, however tree-structured LSTMs have not been applied to syntactic parsers. In the future, we are going to build a Seq2Tree based dependency parser. Dependency parsers have been utilized in quite a few NLP tasks such as Relation Extraction and Event Extraction systems. For example, (Cao et al., 2015) and (Cao et al., 2016) includes syntactic relations with dependency regularizations in event detection systems. Deep neural networks have also applied in semantic relations such as Abstract Meaning Representation parsers. The Seq2Tree structure can also be applied in AMR parsing because the AMR semantic structure is also a tree. AMR parser is widely explored with different NLP tasks such as event detection (Li et al., 2015) and natural language generation (Flanigan et al., 2016).

7. Bibliographical References

Barker, J., Marxer, R., Vincent, E., and Watanabe, S. (2015). The third ‘chime’ speech separation and recognition challenge: Dataset, task and baselines. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 504–511. IEEE.

Cao, K., Li, X., and Grishman, R. (2015). Improving event detection with dependency regularization. In *Proceedings of RANLP*.

Cao, K., Li, X., and Grishman, R. (2016). Leveraging dependency regularization for event extraction. In *Proceedings of FLAIRS*.

Emiya, V., Vincent, E., Harlander, N., and Hohmann, V. (2011). Subjective and objective quality assessment of audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2046–2057.

Erdogan, H., Hershey, J. R., Watanabe, S., and Le Roux, J. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 708–712. IEEE.

Flanigan, J., Dyer, C., and Smith, Noah A.and Carbonell, J. (2016). Generation from abstract meaning representation using tree transducers. In *HLT-NAACL*.

Grais, E. M., Sen, M. U., and Erdogan, H. (2014). Deep neural networks for single channel source separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3734–3738. IEEE.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Li, X., Nguyen, T. H., Cao, K., and Grishman, R. (2015). Improving event detection with abstract meaning representation. In *Proceedings of ACL-IJCNLP Workshop on Computing News Storylines (CNewS 2015)*.

Ma, W., Cao, K., Ni, Z., Li, X., and Chin, P. (2018a). Tree structured multimedia signal modeling. In *The Florida Artificial Intelligence Research Society*.

Ma, W., Cao, K., Ni, Z., Ni, X., and Chin, P. (2018b). Sound signal processing based on seq2tree networks. In *Interspeech workshop on Vocal Interactivity in-and-between Humans, Animals and Robots*.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., and Matassoni, M. (2013). The second ?chime?speech separation and recognition challenge: An overview of challenge systems and outcomes. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 162–167. IEEE.

Vincent, E., Watanabe, S., Nugraha, A. A., Barker, J., and Marxer, R. (2016). An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech & Language*.

Weninger, F., Hershey, J. R., Le Roux, J., and Schuller, B. (2014). Discriminatively trained recurrent neural networks for single-channel speech separation. In *Sig-*

nal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on, pages 577–581. IEEE.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., and Schuller, B. (2015). Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 91–99. Springer.

Wu, M., Liu, Z., Xu, L., and Chen, D. (2015). Accurate and cost-effective technique for jitter and noise separation based on single-frequency measurement. *Electronics Letters*, 52(2):106–107.

Zaremba, W. and Sutskever, I. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zhu, K. and Vogel-Heuser, B. (2014). Sparse representation and its applications in micro-milling condition monitoring: noise separation and tool condition monitoring. *The International Journal of Advanced Manufacturing Technology*, 70(1-4):185–199.