Twin Networks: Using the Future as a Regularizer

Dmitriy Serdyuk[1,2], *Rosemary Nan Ke*[1,3,4], *Alessandro Sordoni*[3], *Chris Pal*[1,4], *Yoshua Bengio*[1,2,†]

[1]MILA , [2] University of Montreal
[3] Microsoft Maluuba, [4] Polytechnique Montreal
[†] CIFAR Senior Fellow

serdyuk@iro.umontreal.ca, rosemary.nan.ke@gmail.com

Abstract

Being able to model long-term dependencies in sequential data, such as text, has been among the long-standing challenges of recurrent neural networks (RNNs). This issue is strictly related to the absence of explicit planning in current RNN architectures, more explicitly, the network is trained to predict only the next token given previous ones. In this paper, we introduce a simple way of biasing the RNNs towards planning behavior. Particularly, we introduce an additional neural network which is trained to generate the sequence in reverse order, and we require closeness between the states of the forward RNN and backward RNN that predict the same token. At each step, the states of the forward RNN are required to match the future information contained in the backward states. We hypothesize that the approach eases modeling of long-term dependencies thus helping in generating more globally consistent samples. The model trained with conditional generation achieved 4% relative improvement (CER of 7.3 compared to a baseline of 7.6).

Index Terms: recurrent neural networks, sequence generation, speech recognition, attention model

1. Introduction

Recurrent Neural Networks are the basis of state-of-art models for generative modeling of sequential data, such as speech recognition. For conditional generation, state-of-art models take advantage of content-based soft attention, e.g., for image captioning [1], speech recognition [2, 3], and machine translation [4]. The decoder model for an attention model is a generative recurrent neural network which has as additional input the convex combination of *contexts* (outputs of the encoder), each corresponding to a different focus of attention. The soft attention mechanism assigns different weights to each context vector providing their convex combination to the decoder.

Given a target sequence, RNNs are usually trained with *teacher forcing*: at each time-step, the hidden state of the RNN is trained to predict the next token given all the previously observed tokens. This corresponds to optimizing a one-step ahead prediction. Usually, samples from RNNs usually exhibit local coherence but lacks meaningful global structure [5]. As there is no explicit bias towards planning in the training objective, the model may prefer focusing on few previously generated tokens instead of capturing long-term dependencies in order to ensure global coherence.

The issue of capturing long-term dependencies was raised and explored in several works [6, 7]. Gated architectures such as LSTMs [8] and GRUs [9] have been successful in easing the modeling of long term-dependencies. Recent work explicitly attempted to model planning by using a value function estimator during sequence decoding, i.e. by biasing the decoding towards tokens that maximize the expected "success" at each step of the generation process [10].

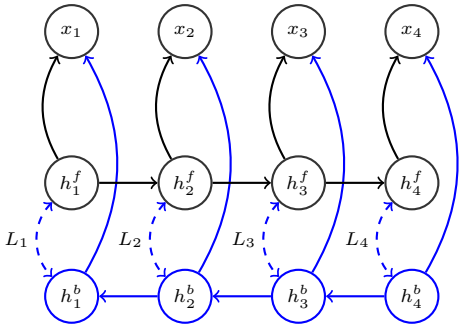

Figure 1: The forward and the backward networks predict the sequence $\{x_1, ..., x_4\}$ independently. The L_2 penalty matches the forward and the backward hidden states. The forward network receives the gradient signal from the log-likelihood objective as well as L_2 between states that predict the same token. The backward network is trained only by maximizing the data log-likelihood. During the evaluation part of the network colored with blue is discarded. Best viewed in color.

In this paper, we propose a simple way of regularizing the recurrent network towards better implicit planning during the training phase. In addition to predict the next token in the sequence, we require the hidden state to contain information about the whole future in the sequence. Succinctly, this is achieved as follows: we run a backward RNN that predicts the sequence in reverse and we encourage the forward hidden states to be close to the backward hidden states that predict the same token, i.e. we force overlap between past and future information about a specific token (Fig. 1). Our model can be generalized to any conditional generative models for sequence-to-sequence tasks. In this paper, we evaluate our model in the setting of conditional generation for speech recognition. We describe the model in details in the next section.

2. Model

Given a dataset $\mathbf{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^n\}$, where each $\mathbf{x} = \{x_1, \dots, x_T\}$ is an observed sequence, an RNN models a density over the space of possible sequences $p(\mathbf{x})$ and is trained to maximize the log-likelihood of the observed data $\mathcal{L} = \sum_{i=1}^{n} \log p(\mathbf{x}^i)$. RNNs factorize the probability of the se-

quence as

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)... = \prod_t p(x_t|x_{<t}), \quad (1)$$

i.e. predict the next element in the sequence given all the previous ones. At each step, the RNN updates a hidden state h_t, which iteratively summarizes the sequence values seen until time t, i.e. $h_t^f = \phi_f(x_{t-1}, h_{t-1}, c)$, where f symbolizes that the network reads the sequence in the forward direction, c symbolizes task-dependent context information, and ϕ_f is typically a non-linear function such as a LSTM [8]. The prediction of x_t is performed using another non-linear transformation on top of h_t^f, i.e. $p_f(x_t|x_{<t}) = \psi_f(h_t^f)$. Therefore, h_t^f summarizes all the information about the past in the sequence. The basic idea of our approach is to promote h_t^f to contain information that is useful to predict x_t but belongs to the future in the sequence. We run another network that predicts the sequence in reverse, i.e. it updates its hidden state according to $h_t^b = \phi_b(x_{t+1}, h_{t+1}, c)$, and predicts $p_b(x_t|x_{>t}) = \psi_b(h_t^b)$ only using information about the future of the sentence. Then, we penalize the forward and backward hidden states to be far away in the Euclidean space (Fig. 1):

$$L_t(\mathbf{x}) = L_2(h_t^f, h_t^b) = ||h_t^f - h_t^b||_2, \quad (2)$$

where the dependence on $\mathbf{x}$ is implicit in the definition of h_t^f, h_t^b. Or using a learned metric, such as:

$$L_t(\mathbf{x}) = L_2(g(h_t^f), h_t^b), \quad (3)$$

where the parametric function g is modelled with an affine transformation. The total loss incurred by the model for a sequence $\mathbf{x}$ is a weighted sum of the forward and backward negative log-likelihoods and the penalty term:

$$L(\mathbf{x}) = -\log p_f(\mathbf{x}) - \log p_b(\mathbf{x}) + \alpha \sum_t L_t(\mathbf{x}), \quad (4)$$

where α controls the importance of the penalty term.

3. Experiments and Results

We apply the model to the task of character-level speech recognition, where the model is trained to convert the speech audio signal to the sequence of characters. The forward and backward RNNs are trained as conditional generative models with soft-attention [2], i.e. the context information c is an encoding of the audio sequence and the target sequence $\mathbf{x}$ is the corresponding character sequence. We evaluate our model on the Wall Street Journal (WSJ) dataset following the setting described in [11]. We use 40 mel-filter bank features with delta and delta-deltas with their energies as the acoustic inputs to the model, these features are generated according to the Kaldi s5 [12] recipe. The resulting input feature dimension is 123.

We observe the Character Error Rate (CER) for our validation set, and we early stop on the best CER observed so far. We report CER for both our validation and test sets. For both the baseline and our model, we pretrain the model for 1 epoch, we then let the context window look freely and perform main training for 15 epochs, we also then train with 2 different annealed learning rate for 3 epochs each. We use the AdaDelta optimizer for training. We weight the L2 norm by 0.5, 0.25, and 0.1 (hyper-parameter α) and select the best one according to the CER on the validation set.

We summarize our initial results in Table 1. We decode from the network without any external language model. Our model shows improvement of 0.2pp comparing to the baseline.

Table 1: Average character error rate (CER%) on WSJ dataset.

Experiment	Beam size	Test, %	Valid, %
Baseline	10	6.8	9.0
+ Twin (L2)	10	6.6	8.7
+ Twin (parametric)	10	**6.2**	8.4
Baseline	1	7.6	8.8
+ Twin (L2)	1	7.3	8.4
+ Twin (parametric)	1	**6.7**	9.2

4. Conclusion and Discussion

We present a generative recurrent model which is regularized to anticipate the future states computed via a second network running in the opposite direction. The experimental results show that this direction is promising and worth exploration.

5. Acknowledgments

The authors would like to acknowledge the support of the following agencies for research funding and computing support: NSERC, Calcul Québec, Compute Canada, the Canada Research Chairs and CIFAR. We would also like to thank the developers of Theano [13] and Blocks and Fuel [14] for developments of great frameworks. We thank Marc Alexandre Coté, Anirudh Goyal, Philemon Brakel and Adam Trischler for useful discussions.

6. References

[1] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[2] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[3] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[5] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues." 2017.

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

[7] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

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

[9] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[10] J. Li, W. Monroe, and D. Jurafsky, "Learning to decode for future success," *arXiv preprint arXiv:1701.06549*, 2017.

[11] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *CoRR*, vol. abs/1508.04395, 2015. [Online]. Available: http://arxiv.org/abs/1508.04395

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[13] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[14] B. van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, and Y. Bengio, "Blocks and fuel: Frameworks for deep learning," *CoRR*, vol. abs/1506.00619, 2015. [Online]. Available: http://arxiv.org/abs/1506.00619