Future Word Contexts in Neural Network Language Models

X. Chen, X. Liu, A. Ragni, Y. Wang, M.J.F. Gales
{xc257,ar527,yw396,mjfg}@eng.cam.ac.uk

UNIVERSITY OF CAMBRIDGE

ALTA Institute / Department of Engineering, University of Cambridge

1. Introduction

- Language model aims to compute probability of sentence $\mathbf{w}_1^L$

$$P(\mathbf{w}_1^L) = P(w_1, w_2, w_3, ..., w_L)$$

- Unidirectional language model (uni-LM)

  The Cat Sat On The Mat

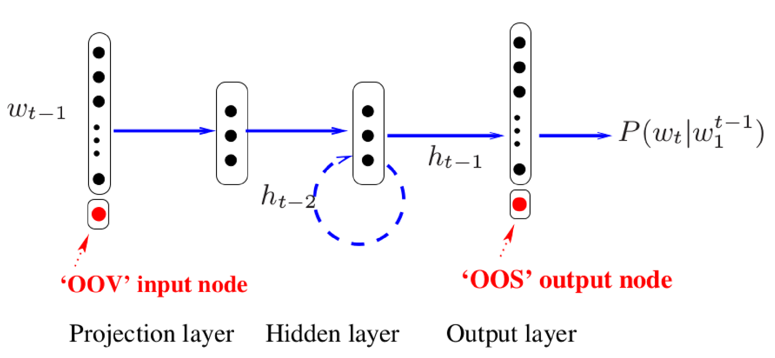
  - Estimate $P(\text{Sat}|\text{The Cat})$
  - Only history information used
- Bidirectional language model (bi-LM)

  The Cat Sat On The Mat

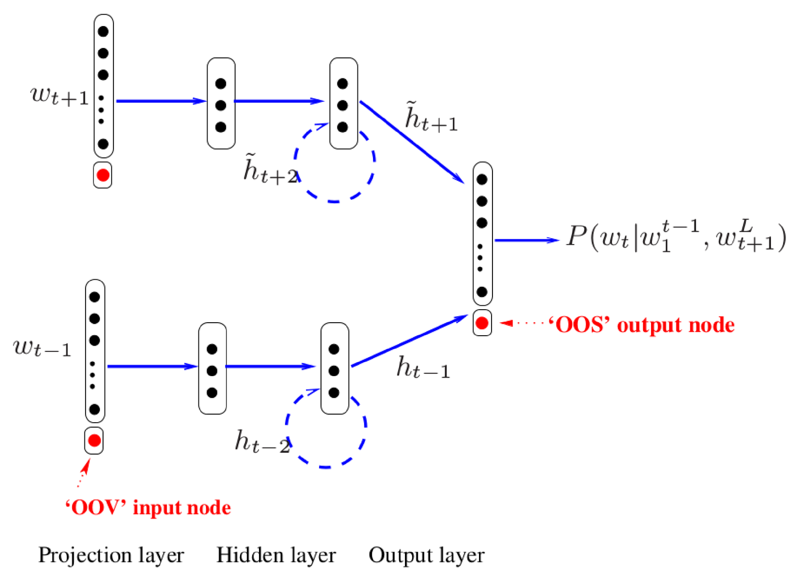
  - Estimate $P(\text{Sat}|\text{The Cat, On The Mat})$
  - Future word context also used
- Recently, bi-RNNLM outperform uni-RNNLM. However, bi-RNNLM
  - Difficult to implement and slow to train
  - Difficult for lattice rescoring, n-best rescoring was used
- In this work, su-RNNLM proposed to address these two issues

2. Unidirectional and bidirectional RNNLM

Unidirectional RNNLM



Bidirectional RNNLM



$\mathbf{h}_{t-1}$: model past history $\mathbf{w}_1^{t-1}$
Sigmoid, GRU and LSTM can be used as recurrent units

$$P(\mathbf{w}_1^L) = \prod_{t=1}^{L} P(w_t|\mathbf{w}_1^{t-1}) \approx \prod_{t=1}^{L} P(w_t|\mathbf{h}_{t-1})$$

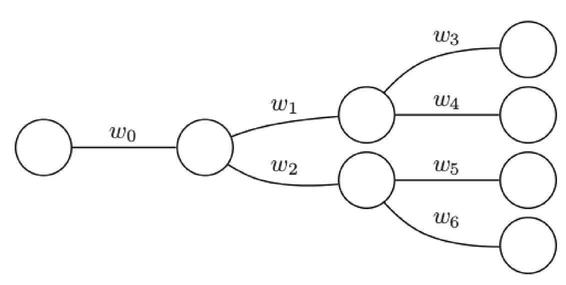
$\tilde{\mathbf{h}}_{t+1}$: model future context w_{t+1}^L

$Z = \sum_{\mathbf{w}_1^L \in \Theta} \prod_{t=1}^{L} P(w_t|\mathbf{h}_{t-1}, \tilde{\mathbf{h}}_{t+1})$

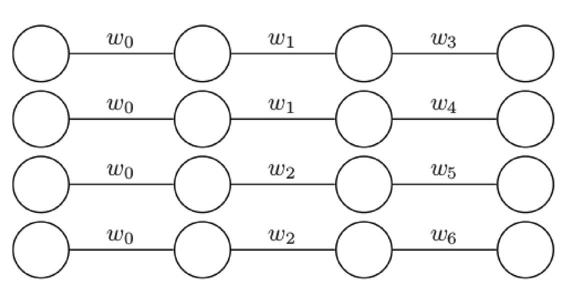
- infeasible to compute

$$P(\mathbf{w}_1^L) \approx \frac{1}{Z} \prod_{t=1}^{L} P(w_t|\mathbf{h}_{t-1}, \tilde{\mathbf{h}}_{t+1})$$

- Unidirectional RNNLM is correct only if
  - a) infinite data, perfect training
  - b) correct history representation
- Bidirectional RNNLM
  - a) product of expert framework
  - b) "optimal" reverse RNNLM
- But, bidirectional RNNLM awkward
  - a) train
  - b) lattice rescoring $\longrightarrow$ n-best rescoring used instead

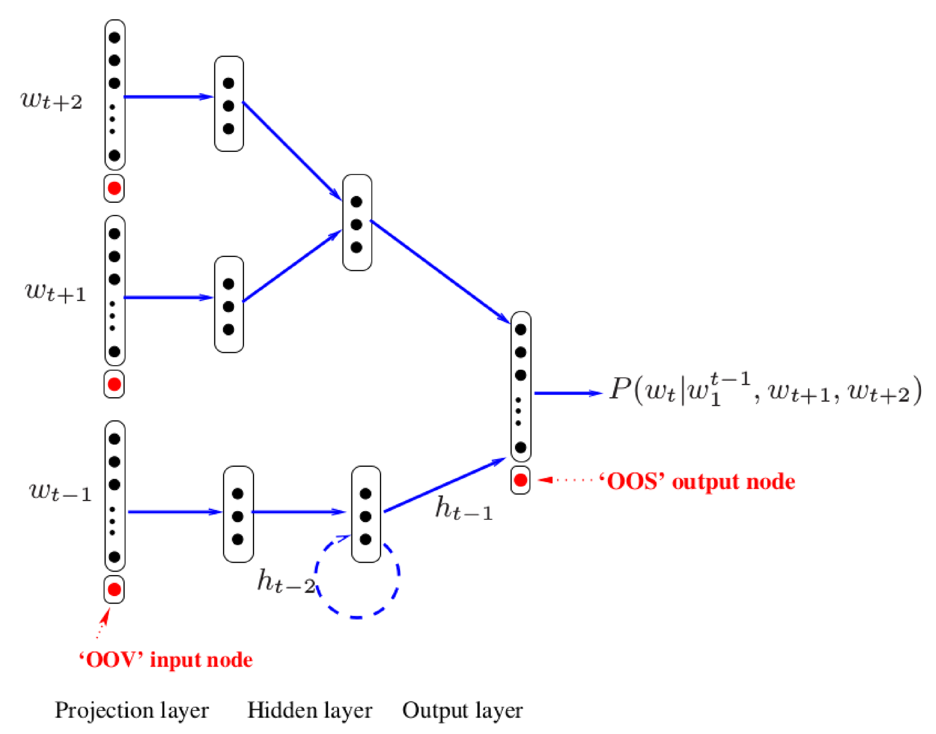


prefix tree



n-best list

4. RNNLM with Succeeding Words (su-RNNLM)

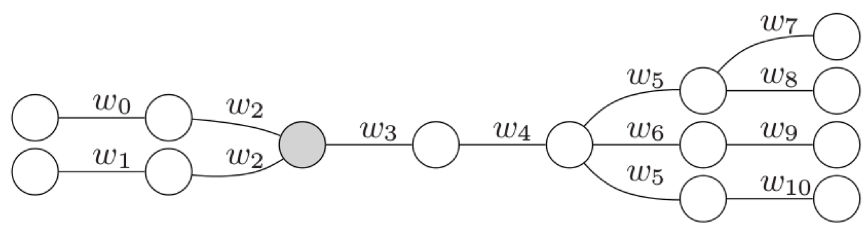


$$P(\mathbf{w}_1^L) = \frac{1}{Z} \prod_{t=1}^{L} P(w_k|\mathbf{w}_1^{t-1}, \mathbf{w}_{t+1}^{t+2}) \approx \frac{1}{Z} \prod_{t=1}^{L} P(w_t|\mathbf{h}_{t-1}, \mathbf{w}_{t+1}^{t+2})$$

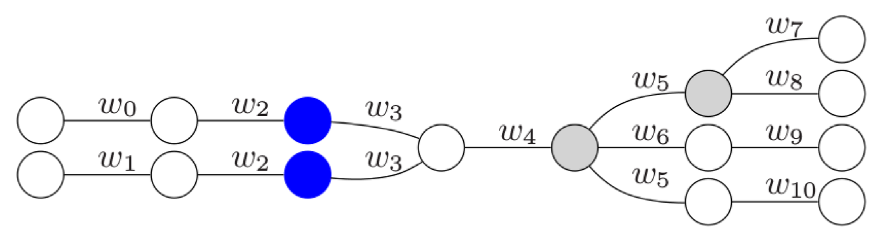
- Recurrent net used for complete history information
- Feedforward net to model a fixed and finite number of succeeding words

5. Lattice Rescoring of su-RNNLMs

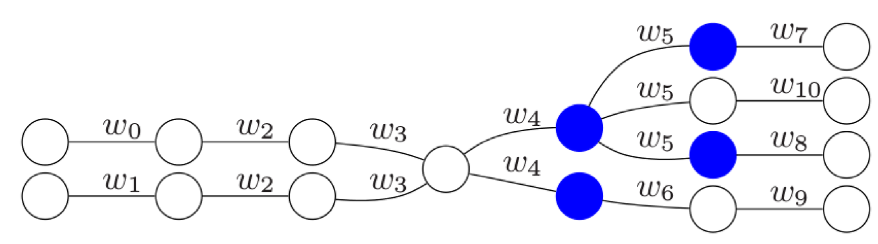
- Lattice generated by 2-gram LM



- Lattice rescored by uni-RNNLM with 3-gram aprox.



- Lattice rescored by su-RNNLM with 3-gram aprox. and 1 succeeding word



3. Interpolation of LMs

1) Uni-LMs interpolation - linear interpolation

$$P_{uni}(w_k|w_1^{k-1}) = \lambda_1 P_{ng}(w_k|w_1^{k-1}) + (1-\lambda_1) P_{rnn}(w_k|w_1^{k-1})$$

2) Bi/Uni-LMs interpolation - log-linear interpolation

$$P(\mathbf{w}_1^L) \propto P_{uni}(\mathbf{w}_1^L)^{\lambda_2} P_{bi}(\mathbf{w}_1^L)^{1-\lambda_2}$$

$$\propto \prod_{k=1}^{L} P_{uni}(w_t|\mathbf{w}_1^{t-1})^{\lambda_2} P_{bi}(w_k|\mathbf{w}_1^{k-1}, \mathbf{w}_{t+1}^L)^{1-\lambda_2}$$

6. Experimental Results

- Setup
  - AMI IHM corpus
  - Kaldi recipe for acoustic model construction
  - 14M words for all LM training (4-gram LM, RNNLMs)
- WERs of uni-, bi, and su-RNNLMs with 100-best rescoring.

LM	#succ words	train speed (w/s)	(pseudo) PPL	dev	eval
ng4	-	-	80.4	23.8	24.2
+uni-rnn	-	4.5K	66.8	21.7	22.1
+su-rnn	0	4.5K	66.8	21.7	22.1
	1	4.5K	25.5	21.5	21.8
	3	3.9K	21.5	21.3	21.6
	5	3.8K	21.3	21.3	21.6
	7	3.8K	21.3	21.4	21.6
	∞	0.8K	22.4	21.2	21.4

- Training of su-RNNLM is much faster than bi-RNNLM (∞)
- su-RNNLM outperform uni-RNNLM (0.4%-0.5%)
- su-RNNLM slightly worse than bi-RNNLM (0.1%-0.2%)

- WERs of uni- and su-RNNLMs with lattice rescoring

LM	#succ words	dev		eval	
		Vit	CN	Vit	CN
ng4	-	23.8	23.5	24.2	23.9
+uni-rnn	-	21.7	21.5	21.9	21.7
+su-rnn	1	21.6	21.3	21.6	21.5
	3	21.3	21.0	21.4	21.1

- Lattice rescoring can be applied on su-RNNLMs
- Consistent improvement obtained from confusion network decoding
- su-RNNLM with 3 succeeding words gave 0.5%-0.6% WER reduction

7. Conclusions

- Future information is useful for language modeling
- Proposed su-RNNLM is easy to implement and fast to train
- Su-RNNLMs suitable for lattice rescoring, consistent WER gain obtained
- Accepted ASRU2017 paper can be found: https://arxiv.org/abs/1708.05592