Deep Neural Language Models for Machine Translation

Minh-Thang Luong, Michael Kayser, Christopher D. Manning
http://stanford.edu/~lmthang/nlm

1 Neural Language Models in Machine Translation

Source context

She jogs along the bank of the river in the morning

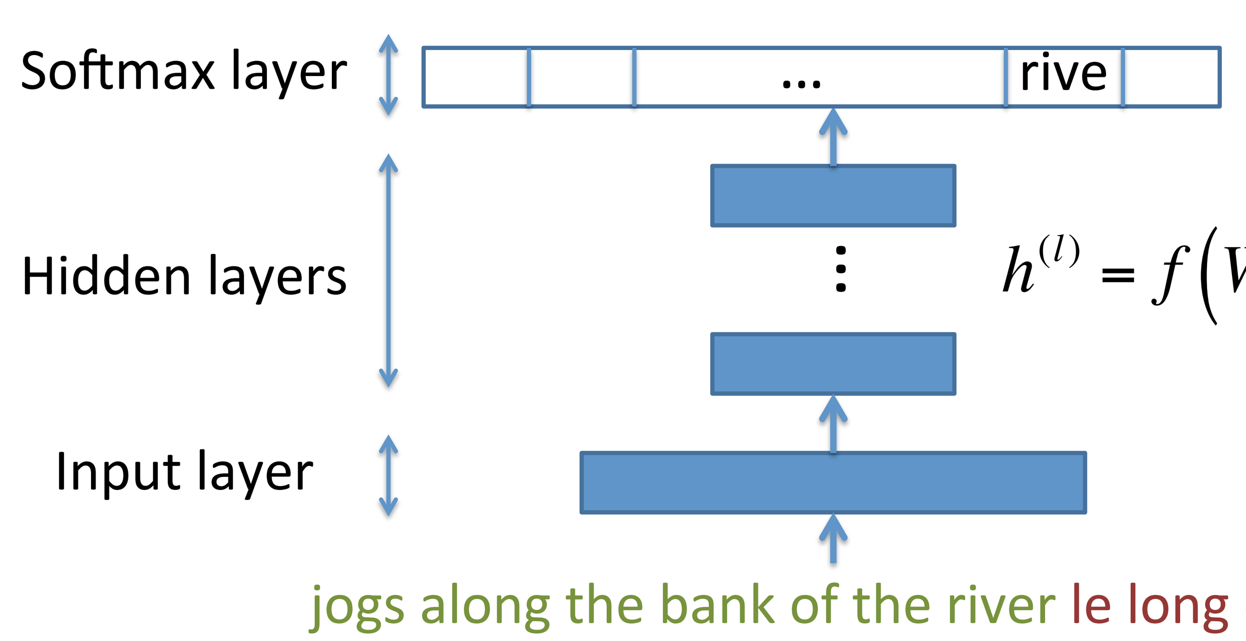
Elle fait du jogging le long de la rive …

Target context

- Neural Language Models (NLMs): able to model *long* contexts of
  - Target words: **standard** NLMs (Bengio et al., 2003).
  - Source + target words: **joint** NLMs (Devlin et al., 2014).
- Only 1- or 2-hidden-layer NLMs have been used in MT.
  - (Schwenk, 2010), (Vaswani et al., 2013): no effect on layers.
  - (Schwenk et al., 2012), (Devlin et al., 2014): a small gain with 2 layers.

No clear results on whether deeper models are better!

2 Deep Neural Language Models



$$h^{(l)} = f\left(W^{(l)} h^{(l-1)} + b^{(l)}\right)$$

- Train self-normalized joint NLMs similar to (Devlin et al., 2014).

$$\sum_{(c,w)\in T} -\log p(w \mid c) + \alpha \log^2(Z_c)$$

- Details (to successfully train deep NLMs):
  - Use *relu* instead of *tanh*: (Vaswani et al., 2013).
  - Simple learning rate schedule: (Sutskever et al., 2014, Luong et al., 2015).
  - Remove sentences with length <= 2 (no gradient clipping).

Train deep NLMs with wisdom from past works.

3 Chinese→English MT Experiments

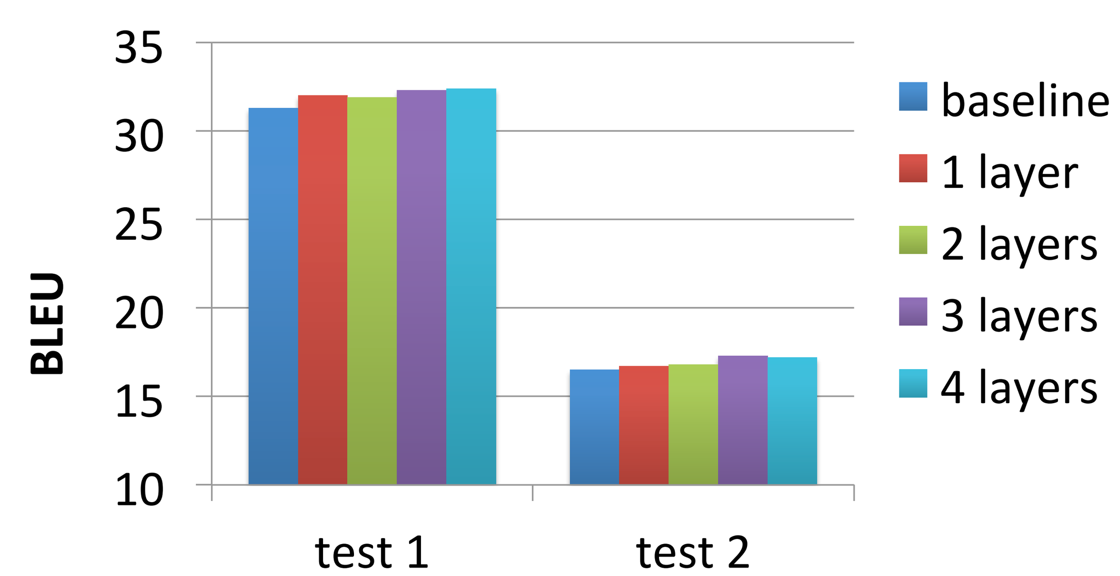
- **Data**: bitext from the DARPA BOLT program.
  - 11.2M sent pairs (281M *Chinese* and 307M *English* words).
- **NLM Training**: 11-gram src context, 4-gram tgt history
  - Top 40K most frequent words.
  - 256-dim embeddings, 512-dim hidden layers.
  - Train 4 epochs: 10-14 days on Tesla K40 (1000 target words/s).
- **Task:** use NLMs to rerank n-best lists of phrasal MT
  - *Strong baseline*: similar to (Green et al., 2014)
    - Dense features + sparse features for rules, word pairs/classes.
    - 3 LMs over English bitext, 16.3B word corpus (word/class).
  - *Discriminative reranker*: on 1000-best output
    - MERT on all dense features, the decoder score, an NLM score.

4 Results

- **NLMs**: more layers, better perplexities / normalization.
  - Self-norm weight α=0.1, validation: 585 sents, test: 1124 sents.

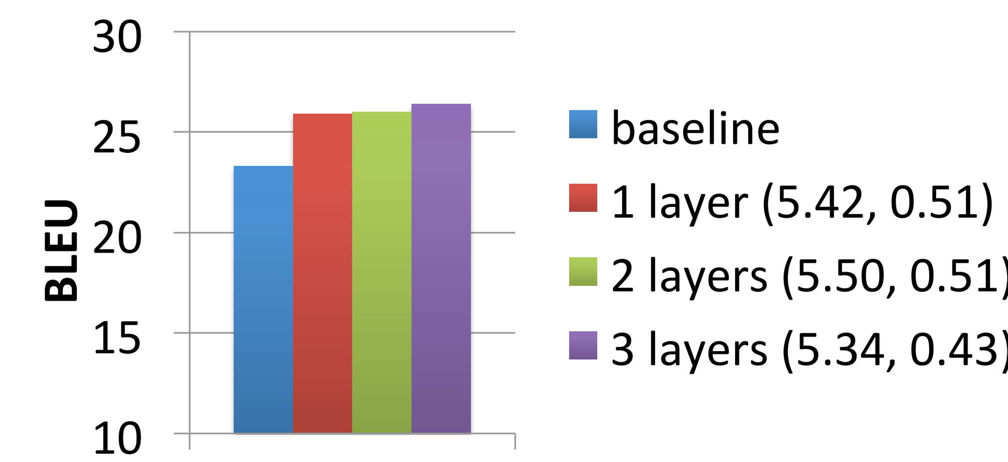
Models	Perplexity Valid	Perplexity Test	Normalization |log Z|
1 layer	9.39	8.99	0.51
2 layers	9.20	8.96	0.50
3 layers	8.64	8.13	0.43
4 layers	**8.10**	**7.71**	**0.35**

- **Reranking**: deeper models give bigger gains
  - 3/4 layers: +0.9 BLEU over baseline, +0.5 BLEU over 1/2 layers.
  - test1: dev10wb syscomtune, test2: p1r6_dev.



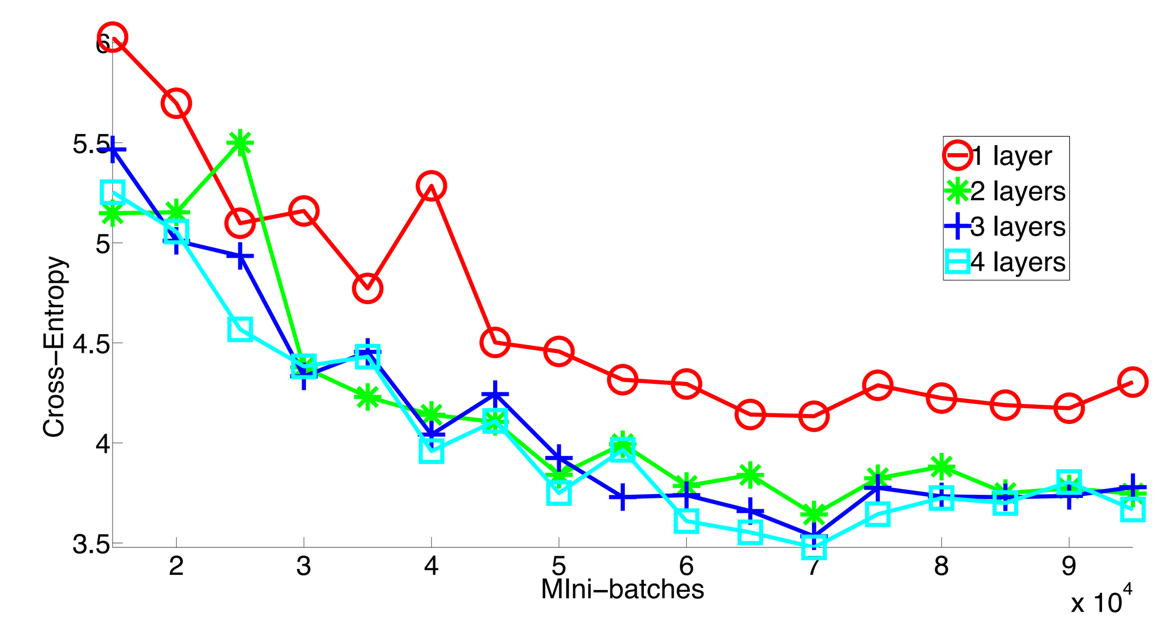
5 Domain Adaptation

- Adapt from web-forum domain to sms-chat:
  - Use existing models: finetune on out-of-domain data.
  - Sms-chat corpus: 146K sent pairs.
- Similar trends observed:
  - 3 layers: +3.1 BLEU over baseline, +0.5 BLEU over 1/2 layers.
  - 4 layers: overfit training data.
  - test: p2r2smscht syscomtune



6 Analysis

- Learning curve: deeper NLMs are better than 1-layer NLMs.
  - Gaps between models towards the end are 40.1, 1.1, 2.0 (in perplexities).



7 Conclusion

- Bridge the gap from past work
  - Deep NLMs (>2 layers) do improve the translation quality, but gains are increasingly modest.
- Demonstrate how to adapt NLMs to out-domain conditions:
  - Deep NLMs with 3 layers yield substantial gains.
  - 4-layer models unfortunately overfit. Future: more regularization.