Complex Architectures for Neural Language Modelling

What recent research has to tell us

Dr. Giancarlo D. Salton

Applied Intelligence Research Centre & ADAPT Centre
School of Computing

Dublin, 27 August 2018

Neural Language Models

- Component of almost all Natural Language Processing systems
- Provides a probabilistic score for a piece of text
- Reflects how likely that piece of text is to appear in a given language

Example: Machine Translation

- Source:

  er geht ja nicht nach hause

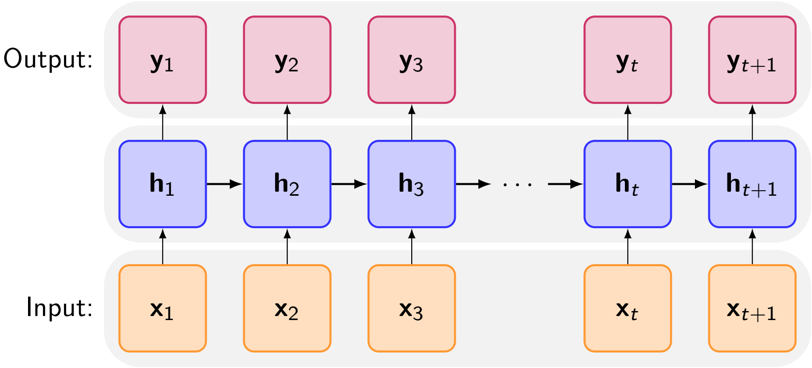
- Candidate translations:

  he is yes not after house

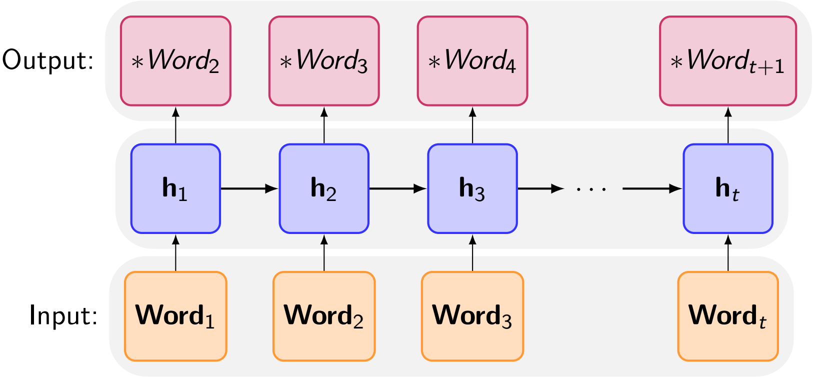
  it are is do not according to home

  – *he does not go home*

Recurrent Neural Network (unrolled through time)



Language Models based on RNNs (Neural Language Models)



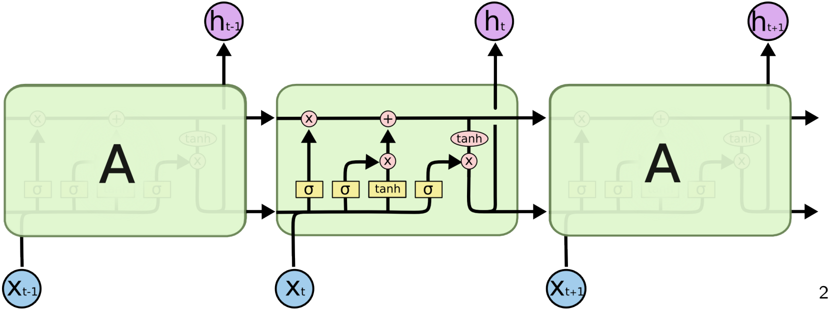
Long Short-Term Memory

LSTM equations[1]

$$
\begin{aligned}
\widetilde{\mathbf{c}}_t =& tanh(\mathbf{W}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{(t-1)} + \mathbf{b}) \\
\mathbf{i}_t =& \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{(t-1)} + \mathbf{b}_i) \\
\mathbf{f}_t =& \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{(t-1)} + \mathbf{b}_f) \\
\mathbf{c}_t =& \mathbf{f}_t \times \mathbf{c}_{(t-1)} + \mathbf{i}_t \times \widetilde{\mathbf{c}}_\mathbf{t} \\
\mathbf{o}_t =& \sigma(\mathbf{W}_{io}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{(t-1)} + \mathbf{b}_o) \\
\mathbf{h}_t =& \mathbf{o}_t \times \tanh(\mathbf{c}_t)
\end{aligned}
$$

[1] Gers, F. A., Schmidhuber, J. A., and Cummins, F. A. (2000). Learning to forget: Continual prediction with lstm. *Neural Comput.*, 12(10):2451–2471

LSTMs



2

[2] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Complex Architectures

- Recurrent Memory Network (Tran et al., 2016)
- Long Short-Term *Memory-Network* (LSTMN) (Cheng et al., 2016)
- *N*-Gram Recurrent Neural Network (Daniluk et al., 2017)
- Attentive Language Model (Salton et al., 2017)

- What all this architectures have in common?
  - Complex models
  - Require additional computer power
  - Difficult to optimise
  - **Minor improvements over baselines**

What Recent Research has to Tell us

Press and Wolf (2016)[3] - Embeddings Weights

- Embeddings learned by the NLM are similar to *word2vec*, *GloVe*, etc
- Using these embeddings in the final linear transformation helps in regularizing the embeddings themselves
- Applying variational dropout to input weights is better than regular dropout
  - Note: the authors refer to *Variational dropout* as *Bayesian dropout*
- Results on PennTree Bank:

Model	Params	Valid. Set	Test Set
Large LSTM	66M	82.2	78.4
Large LSTM + VD + WT	**66M**	**75.8**	**73.2**

[3] Press, O. and Wolf, L. (2016). Using the output embedding to improve language models. *arXiv*

Melis et al. (2017)[5] - Hyperparameter search

- There is a lot of hyperparameters in a NLM
- Fine-tuning those hyperparameters makes a huge difference!
  - Also demonstrated by Howard and Ruder (2018)[4] within text classification context

- Results on PennTree Bank:

Model	Params	Valid. Set	Test Set
Large LSTM + VD + WT	66M	75.8	73.2
Fine-tuned LSTM	**24M**	**60.9**	**58.3**

[4] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339

[5] Melis, G., Dyer, C., and Blunsom, P. (2017). On the state of the art of evaluation in neural language models. *ICLR'2018*, abs/1707.05589

Merity et al. (2017)[6] - Weight Dropping

- Drop Connect Wan et al. (2013) to recurrent weights → Weight Dropping
- Averaged Stochastic Gradient Descent (ASGD) variant → Non-monotonically Triggered ASGD (NT-ASGD)
- Pointer network Merity et al. (2016) (evaluation only)

- Results on PennTree Bank:

Model	**Params**	**Valid. Set**	**Test Set**
Fine-tuned LSTM	24M	60.9	58.3
AWD + WT + Pointer	**24M**	**53.9**	**52.8**

[6] Merity, S., Keskar, N. S., and Socher, R. (2017). Regularizing and Optimizing LSTM Language Models. *arXiv*

Krause et al. (2018)[7] - Dynamic Evaluation

- Update weights during evaluation based on parts of the sequence
- Reset to initial weights at the end of evaluation

- Results on PennTree Bank:

Model	Params	Valid. Set	Test Set
AWD + WT + Pointer	24M	53.9	52.8
AWD + WT + Dynamic	**24M**	**51.6**	**51.1**

[7] Krause, B., Kahembwe, E., Murray, I., and Renals, S. (2018). Dynamic evaluation of neural sequence models. In *Proceedings of the 35th International Conference on Machine Learning*

Yang et al. (2018)[8] - Mixture of Softmaxes

- Replace softmax with a *Mixture of Softmaxes*, *i.e.*, a set of softmaxes are applied to different parts of the prediction vector
- Each part of the prediction vector has its own *mixture weight*
- In the authors' words words:
  - "MoS computes K Softmax distributions and uses a weighted average of them as the next-token probability distribution"

- Results on PennTree Bank:

Model	Params	Valid. Set	Test Set
AWD + WT + Dynamic	24M	51.6	51.1
AWD + WT + MoS + Dynamic	**24M**	**48.3**	**47.6**

[8] Yang, Z., Dai, Z., Salakhutdinov, R., and Cohen, W. W. (2018). Breaking the softmax bottleneck: A high-rank RNN language model.
ICLR'2018

What all these models have in common?

- Tricks to reuse parameters
- Use of dropout on recurrent connections
- Include prior information about the sequences on test/eval. time
- **Great improvement over initial baselines**

- **Use of standard LSTM units!**

Khandelwal et al. (2018)[9] - Use of context by NLMs with LSTM units

- NLMs have an effective context size of about 200 tokens on average
- Infrequent words need more context than frequent words
- Content words matter more than function words
- Local word order only matters for the most recent 20 tokens
- LSTM units can regenerate words seen in nearby context
- Caches help words that can be copied from long-range context the most

[9] Khandelwal, U., He, H., Qi, P., and Jurafsky, D. (2018). Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 284–294

Mahalunkar and Kelleher, (2018)[10] - MI decay of datasets

- Analysis of Mutual Information (MI) of long-distance dependencies:
  - The decay of that MI in benchmark language model datasets is a power-log decay
- Why not explore this information to help the LM to recover the MI from context?

[10] *To appear*

Salton et al., (2018)[12] - Recurrent residual connections

- We revisit recurrent residual connections Wang and Tian (2016)[11]
- Although these are preliminary results, our results are competitive with the SOTA models when not considering dynamic evaluation methods
- We are now:
  - conducting ablation studies similar to Khandelwal et al. (2018) to understand exactly where the model is improving
  - finetuning hyperparameters and analysing the use of dynamic evaluation methods
- Results on PennTree Bank (without dynamic evaluation methods):

Model	Params	Valid. Set	Test Set
AWD + WT	24M	60.0	57.3
AWD + WT + MoS	24M	56.5	54.4
AWD + WT + RIT†	24M	58.7	54.0

†Ours

[11] Wang, Y. and Tian, F. (2016). Recurrent residual learning for sequence classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 938–943. Association for Computational Linguistics

[12] *To appear*

Conclusions

Summary of Published Results

Model	Params	Valid. Set	Test Set
Medium LSTM†	16M	86.2	82.7
Large LSTM†	66M	82.2	78.4
Large LSTM + VD + WT‡	66M	75.8	73.2
Fine-tuned LSTM$^{\diamond}$	24M	60.9	58.3
AWD + WT + Pointer$^{\pm}$	24M	53.9	52.8
AWD + WT + Dynamic§	24M	51.6	51.1
AWD + WT + MoS + Dynamic$^{\mp}$	24M	**48.3**	**47.6**

$\dagger$Zaremba et al. (2015)
$\ddagger$Press and Wolf (2016)
$\diamond$ Melis et al. (2017)
$\pm$ Merity et al. (2017)
$\S$ Krause et al. (2018)
$\mp$ Yang et al. (2018)

Conclusions

- If you do not have a good infrastructure to run complex models, you are better of using carefully tuned standard LSTMs
- In fact, until the next breakthrough, we are all better of using carefully tuned standard LSTMs with some type of caching mechanism after training.

Thank you!

This research was partly funded by the ADAPT Centre. The ADAPT Centre is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.