Regularizing and Optimizing LSTM Language Models

Stephen Merity, Nitish Shirish Keskar & Richard Socher

Language modeling and AWD-LSTM-LM

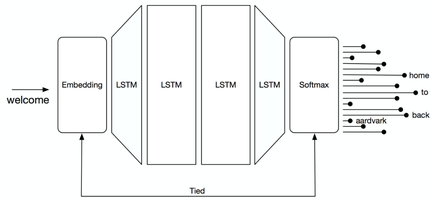
Motivation:
- Language modeling (or predicting the next word/s given previous context) is useful for pre-training decoders in Seq2Seq architectures, as feature extractors and for assigning probability estimates in speech recognition and natural language generation.
- Custom architectures often proposed for language modeling. We revisit the need for such architectures.

Proposal:
- Appropriately regularized and trained LSTMs can efficiently attain state-of-the-art (SOTA) performance for many tasks.
- Proposed strategies are also applicable to Quasi-Recurrent Neural Networks (QRNNs) which enable similar SOTA performance at a fraction of the cost.

Results:
- SOTA on Penn Treebank and WikiText-2 datasets.
- Open source implementation available @ github.com/salesforce/awd-lstm-lm

Regularization

Weight Drop:
- DropConnect the hidden-to-hidden weight matrix.
- Randomly sample a mask for each forward pass.
- Efficient. Can still use cuDNN LSTM.

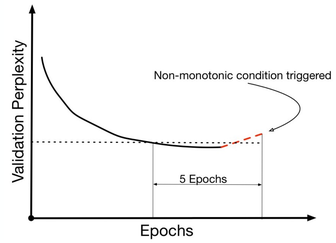


Random BPTT length:
- Prevents the model from seeing the exact same batches. Regularizing effect, especially on rarer words.

Others:
- Embedding and LSTM dropout.
- Activity Regularization (AR), Temporal Activation Regularization (TAR).
- Weight Tying
- L2-norm decay

Optimization

Observations:
- SGD outperformed adaptive methods like Adam or RMSprop.
- Performance of SGD was sensitive to learning rate decay schedule.



Proposal:
- Use Averaged SGD (AvSGD) instead of SGD or adaptive methods.
- Use a non-monotonic trigger on the validation perplexity. If perplexity fails to improve for 5 epochs, start maintaining a running average of iterates.
- Validation-based triggering allows for automatic determination; non-monotonicity allows for conservatism.

Experimental Setup

- LSTMs/QRNNs and embeddings were randomly initialized.
- Coarsely tuned regularization and optimization hyperparameters.
- Experiments were conducted on an NVIDIA Quadro GP100 GPU.
- Each epoch of PTB took ~65 seconds on LSTM and ~28 seconds on QRNN. Each epoch of WT2 took ~180 seconds on LSTM and ~90 seconds on QRNN.

	Penn Treebank			WikiText-2		
	Train	Valid	Test	Train	Valid	Test
Docs	-	-	-	600	60	60
Word	0.8M	70k	79k	2.0M	218k	246k
Vocab	10,000			33,278		
OoV	4.8%			2.6%		

Dataset	Model	Parameters	Val	Test
Zoph & Le, 2016	NAS Cell (tied)	25M	-	64.0
Melis et al., 2017	4-layer LSTM (tied)	24M	60.9	58.3
Ours	3-layer LSTM (tied)	24M	60.0	57.3
Ours	4 layer QRNN (tied)	24M	59.1	56.7

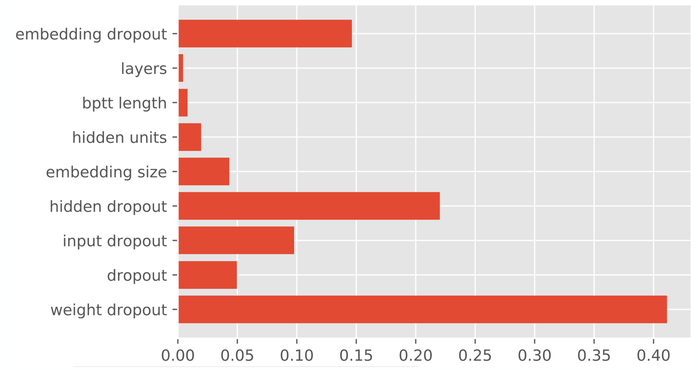
Language modeling results over PTB

Dataset	Model	Parameters	Val	Test
Inan et al., 2016	Variational LSTM (tied)	28M	91.5	87.0
Melis et al., 2017	2-layer LSTM (tied)	24M	69.1	65.9
Ours	3-layer LSTM (tied)	33M	68.6	65.8
Ours	4 layer QRNN (tied)	33M	68.5	65.9

Language modeling results over WikiText-2

Hyperparameter Importance

- Train 200 random bounded hyperparameters on WT2. Use RandomForest regression for parameter importance.



Discussion

- Effectively regularized and trained LSTM language models can achieve state-of-the-art perplexity.
- Weight dropout is simple, effective, and does not require abandoning cuDNN libraries.
- Averaged SGD outperforms SGD, and helps attain robust and superior performance.
- On a state-of-the-art model, neural cache can significantly improve performance further. Points to an inherent deficiency of current neural language models.
- Same strategies work well even for QRNNs and can be much faster.

Recent Work

- (Merity et al., 2018): Regularized LSTMs are sufficient for character-level and large-vocabulary language modeling as well.
- (Melis et al., 2017): Concurrent work also showing that well-tuned LSTMs deliver competitive performance.
- (Krause et al., 2017): Dynamic evaluation can further improve such language models during inference time.
- (Yang et al., 2017): A low-rank softmax matrix limits performance of language modeling. Using multiple softmaxes helps break this barrier further improving state-of-the-art perplexity.

Website:
- github.com/salesforce/awd-lstm-lm

Contact:
- Nitish Shirish Keskar (nkeskar@salesforce.com)

Check out more of our work at https://einstein.ai/research

Selected References:
- *Stephen Merity, Nitish Shirish Keskar, Richard Socher.* Regularizing and Optimizing LSTM Language Models.
- *Stephen Merity, Nitish Shirish Keskar, Richard Socher* An Analysis of Neural Language Modeling at Multiple Scales.
- *James Bradbury, Stephen Merity, Caiming Xiong, Richard Socher.* Quasi-Recurrent Neural Networks.
- *Yann Dauphin, Angela Fan, Michael Auli, David Grangier.* Language Modeling with Gated Convolutional Networks.
- *Hakan Inan, Khashayar Khosravi, Richard Socher.* Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling.

salesforce