Deep Residual Output Layers for Neural Language Generation

Nikolaos Pappas, James Henderson

Idiap Research Institute, Martigny, Switzerland

https://github.com/idiap/drill

Motivation

Many tasks such as *zero-shot classification* and *structured prediction* benefit from learning the output space structure. Typical output layers for *neural language generation*:

- Indirectly capture the similarity structure of the output space
- Have limited expressivity and are prone to overfitting
- Increasing their power comes with a high overhead

Problem: Neural Language Generation

The probability distribution at time t conditioned on $\mathbf{y}_1^{t-1}$ encoded in a vector $\mathbf{h_t} \in \mathbb{R}^d$ is modeled by linear unit $\mathbf{W} \in \mathbb{R}^{d\times|\mathcal{V}|}$, $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$:

$$p(\mathbf{y_t}|\mathbf{y}_1^{t-1}) \propto \texttt{exp}(\mathbf{W}^T\mathbf{h_t} + \mathbf{b})$$

- Parameterisation depends on the vocabulary size $|\mathcal{V}|$
- Power depends on the classifier rank d aka "softmax bottleneck"

Previous Work

Weight tying [PW17] matrix $\mathbf{W}$ with the word embedding $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}|\times d}$ helps but still lacks parameter sharing across outputs:

$$\mathbf{E}\mathbf{h_t} + \mathbf{b}$$

Bilinear mapping [G18] explicitly shares parameters across outputs through matrix $\mathbf{W}_l$:

$$\mathbf{E}\mathbf{W}_l\mathbf{h_t} + \mathbf{b}$$

Dual nonlinear mapping [P18] shares parameters across outputs and contexts through a nonlinear joint space:

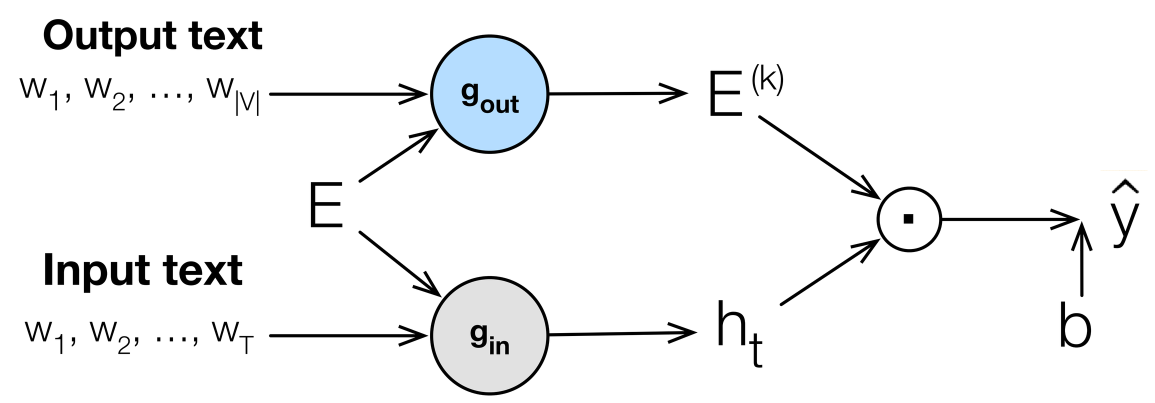
$$g_{out}(\mathbf{E})g_{in}(\mathbf{h_t}) + \mathbf{b}$$

Limitations:

$$|\Theta_{tied}| < |\Theta_{bilinear}| \leq |\Theta_{dual}| \leq |\Theta_{base}|$$

- *Shallow output space modeling*: power depends on the rank d
- *Tendency to overfit*: increased power leads to undesired effects

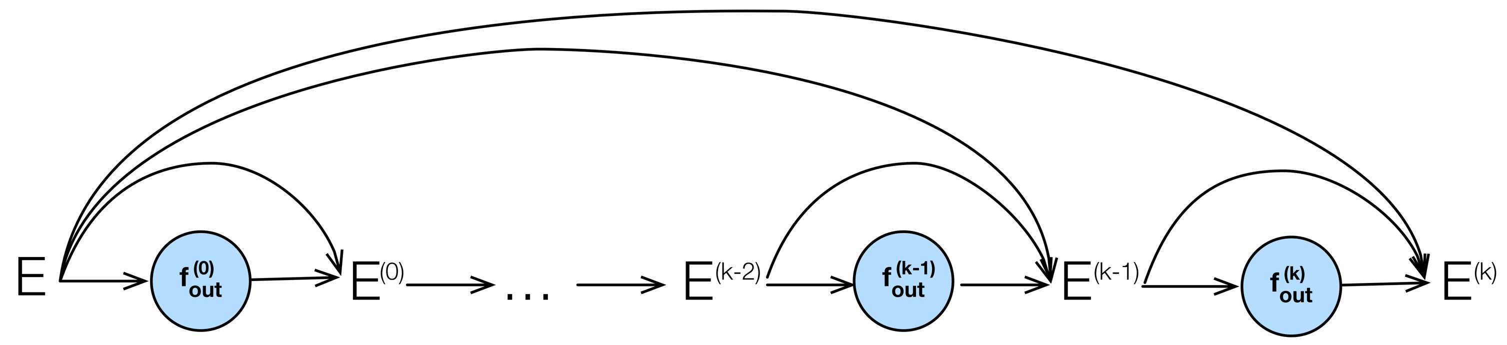
Proposed Architecture Overview



We propose a deep output layer architecture based on the general form and the basic principles of previous work, the power of which no longer depends on the classifier rank d:

$$p(\mathbf{y_t}|\mathbf{y}_1^{t-1}) \propto \texttt{exp}(\mathbf{E}^{(k)}\mathbf{h_t} + \mathbf{b})$$

Label Encoder Network



Shares parameters across outputs through a deep residual output mapping with depth k while keeping the rank d fixed:

$$\mathbf{E^{(k)}} = f_{out}^{(k)}(\mathbf{E^{(k-1)}}) = \sigma(\mathbf{E^{(i-1)}}\mathbf{U^{(i)}} + \mathbf{b_u^{(i)}})$$

Properties

Preserving information with residual connections to the word embedding and, optionally, to the outputs of previous layers:

$$\mathbf{E^{(k)}} = f_{out}^{(k)}(\mathbf{E^{(k-1)}}) + \mathbf{E^{(k-1)}} + \mathbf{E}$$

Controlling power by increasing the projection depth k:

$$|\Theta_{drill}| \approx k \times (d \times d)$$

Avoiding overfitting with standard or variational dropout in between each of the k projection layers:

$$f_{out}'^{(i)}(\mathbf{E^{(i-1)}}) = \delta(f_{out}^{(i)}(\mathbf{E^{(i-1)}})) \odot f_{out}^{(i)}(\mathbf{E^{(i-1)}})$$

References

[PW17] Ofir Press and Lior Wolf. Using the Output Embedding to Improve Language Models. *EACL*. Valencia, Spain, 2017

[G18] Kristina Gulordava et al. Improving tied architectures for language modelling. *EMNLP*, Brussels, Belgium, 2018.

[P18] Nikolaos Pappas et al. Learning Joint Input-Output Embeddings for Neural Machine Translation. *WMT*, Brussels, Belgium, 2018.

[M18] Stephen Merity et al. Regularizing and Optimizing LSTM Language Models. *ICLR*, Vancouver, Canada, 2018

[Y18] Zhilin Yang et al. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model. *ICLR*, Vancouver, Canada, 2018

[V17] Ashish Vaswani et al.. Attention is All you Need. *Advances in Neural Information Processing Systems*, 2017.

Evaluation

We evaluate on two language generation tasks using state-of-the-art architectures, namely AWD-LSTM [M18] and Transformer [V18].

Language Modeling

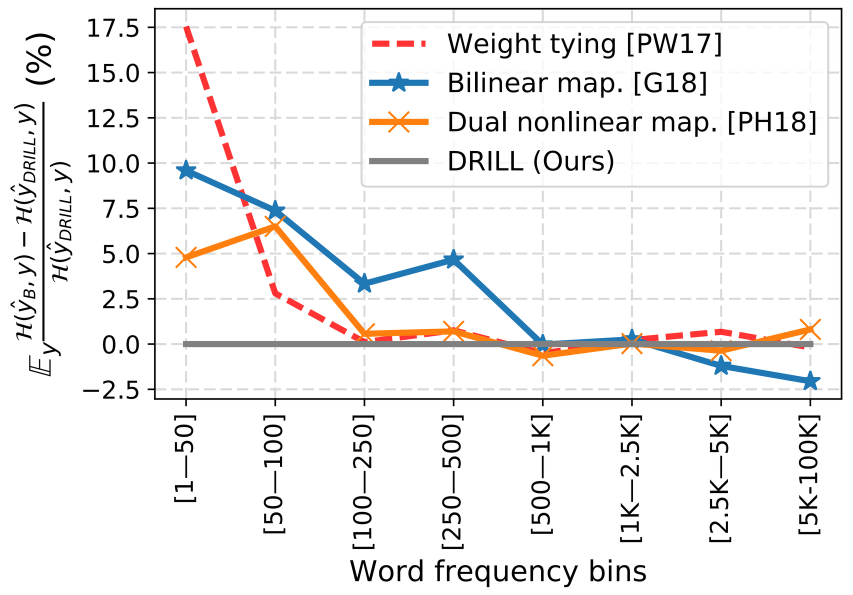
	PennTreebank		WikiText-2	
Model	**ppl**	**sec/ep**	**ppl**	**sec/ep**
AWD-LSTM [M18]	57.3	47 (1.0×)	65.8	89 (1.0×)
AWD-LSTM-DRILL	55.7	53 (1.1×)	61.9	106 (1.2×)
AWD-LSTM-MoS [Y18]	54.44	139 (3.0×)	61.45	862 (9.7×)

Machine Translation

	En→De (32K)	
Model	**bleu**	**min/ep**
Transformer (base) [V17]	27.3	111 (1.0×)
Transformer-DRILL (base)	28.1	189 (1.7×)
Transformer (big) [V17]	28.4	779 (7.0×)

Ablation Analysis

Output Layer	#Param	Penn
Full softmax	43.8M	66.8
Weight tying [PW17]	24.2M	57.3
Bilinear map. [G18]	24.3M	58.5
Dual map. [P18]	24.5M	56.4
DRILL 1-layer	24.3M	56.2
DRILL 2-layers	24.5M	56.0
DRILL 3-layers	24.7M	55.9
DRILL 4-layers	24.8M	**55.7**



Conclusion

Deeper output mappings for neural language generation:

- Improve recurrent or self-attentional architectures without increasing their rank which often leads to high overhead
- Lead to better transfer across the output labels, especially the low-resource ones

Future work: Explore other generation tasks, learn elaborate/multi-level descriptions, investigate transferability