Restricted Recurrent Neural Networks

Enmao Diao[1] Jie Ding[2] Vahid Tarokh[1]

[1]Electrical and Computer Engineering
Duke University

[2]Statistics
University of Minnesota Twin Cities

December 11, 2019
2019 IEEE International Conference on Big Data

Overview

1. Motivation
2. Restricted Recurrent Neural Networks (RRNN)
3. Generalized RRNN for LSTM and GRU
4. Experiments
5. Future work and Conclusion

Overview

Motivation

- Can we reduce the number of parameters of Recurrent Neural Networks (RNN) while still maintain comparable performance?
- The number of parameters of RNN and its variations.
  - d: output channel size, k: input channel size
  - Fully Connected: $W^{d \times k}$, $b^{d \times 1}$

$$h_t = \tanh(Wx_t + b)$$

  - Vanilla RNN: 2x parameters

$$h_t = \tanh(W_{xh}x_t + b_{xh} + W_{hh}h_{t-1} + b_{hh})$$

Motivation

- The number of parameters of RNN and its variations.
  - Gated Recurrent Unit (GRU): 6x parameters

$$
\begin{aligned}
r_t &= \sigma(W_{xr}x_t + b_{xr} + W_{hr}h_{t-1} + b_{hr}) \\
z_t &= \sigma(W_{xz}x_t + b_{xz} + W_{hz}h_{t-1} + b_{hz}) \\
n_t &= \tanh(W_{xn}x_t + b_{xn} + r_t * (W_{hn}h_{t-1} + b_{hn})) \\
h_t &= (1 - z_t) * n_t + z_t * h_{t-1}
\end{aligned}
$$

  - Long short-term memory (LSTM): 8x parameters

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + b_{xi} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{xf}x_t + b_{xf} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{xg}x_t + b_{xg} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{xo}x_t + b_{xo} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t * c_{t-1} + i_t * g_t \\
h_t &= o_t * \tanh(c_t)
\end{aligned}
$$

Motivation

- Classical model compression
  - Parameter pruning and quantization
  - Low-rank factorization
  - Retraining from a pre-trained model
  - Fine-tuning regularization parameters

- We propose Restricted RNN specifically taking advantage of the recurrent structures of RNNs
  - One-time training on compressed models
  - Controllable compression rates
  - Restricted LSTM outperform vanilla RNN with even less number of parameters ($<$ 2x parameters)

Overview

Restricted Recurrent Neural Networks (RRNN)

- Why do we model x_t and h_t separately ?
  - x_t and h_t have different 'meanings'
  - x_t and h_t are dependent. We propose sharing model parameters for x_t and h_t with sharing rate $r \in [0, 1]$
  - Two extreme cases:
    - $r = 1$, $W = W_{xh} = W_{hh}, b = b_{xh} = b_{hh}$ (Fully Connected)

$$h_t = \tanh(W(x_t + h_{t-1}) + b)$$

    - $r = 0$, $W_{xh} \neq W_{hh}, b_{xh} \neq b_{hh}$ (Vanilla RNN)

$$h_t = \tanh(W_{xh}x_t + b_{ixh} + W_{hh}h_{t-1} + b_{hh})$$

Restricted Recurrent Neural Networks (RRNN)

- Let output and input channel size for x_t and h_t be d_{xh}, d_{hh} and k_{xh}, k_{hh}. Let sharing rate for x_t and h_t be r_{xh} and r_{hh}.
- shared output channel size s_{xh}, s_{hh}

$$s_{xh} = \text{Round}(r_{xh} \times d_{xh}),\ s_{hh} = \text{Round}(r_{hh} \times d_{hh})$$

- non-shared output channel size q_{xh}, q_{hh}

$$q_{xh} = d_{xh} - s_{xh},\ q_{hh} = d_{hh} - s_{hh},$$

- parameter pool W, b

$$\begin{aligned} s_r &= max(s_{xh}, s_{hh}),\ k_r = max(k_{xh}, k_{hh}) \\ d_r &= s_r + q_{xh} + q_{hh} \\ W &\sim (d_r,\ k_r),\ b \sim (d_r,) \end{aligned}$$

Restricted Recurrent Neural Networks (RRNN)

- Let restricted weight matrix and bias for x_t and h_t be W^r_{xh}, W^r_{hh} and b^r_{xh}, b^r_{hh}.

$$W^r_{xh} = \begin{pmatrix} W[: s_{xh}, : k_{xh}] \\ W[s_r : s_r + q_{xh}, : k_{xh}] \end{pmatrix}$$

$$b^r_{xh} = \begin{pmatrix} b[: s_{xh}] \\ b[s_r : s_r + q_{xh}] \end{pmatrix}$$

$$W^r_{hh} = \begin{pmatrix} W[: s_{hh}, : k_{hh}] \\ W[s_r + q_{xh} : s_r + q_{xh} + q_{hh}, : k_{hh}] \end{pmatrix}$$

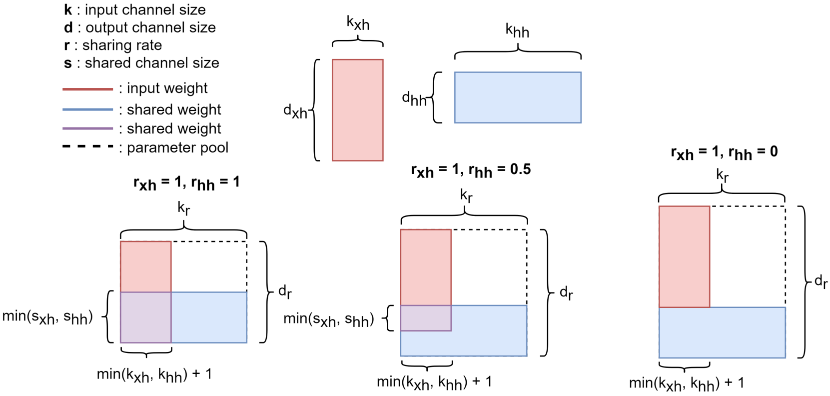
$$b^r_{hh} = \begin{pmatrix} b[: s_{hh}] \\ b[s_r + q_{xh} : s_r + q_{xh} + q_{hh}] \end{pmatrix}$$

- The formulation of Restricted RNN

$$h_t = \tanh(W^r_{xh} x_t + b^r_{xh} + W^r_{hh} h_{t-1} + b^r_{hh})$$

Restricted Recurrent Neural Networks (RRNN)

- Illustration of parameter restriction in RRNN.



- Assuming the common practice that $d_{xh} = d_{hh}, k_{xh} = k_{hh}, r_{xh} = r_{hh}$ compression rate $C = \frac{2d-s}{2d} = \frac{2-r}{2}$

Overview

Generalized RRNN for LSTM and GRU

- Multiple parameter matrices associated with x_t and h_t.
  - Given m inputs, n outputs and sharing rate matrix $r_{m \times n}$, we can generalize our formulation.
  - GRU $(m = 2, n = 3)$, LSTM $(m = 2, n = 4)$
  - shared output channel size $s_{m \times n}$

$$s_{m \times n} = \text{Round}(r_{m \times n} \times d_{m \times n})$$

  - non-shared output channel size $q_{m \times n}$

$$q_{m \times n} = d_{m \times n} - s_{m \times n}$$

  - parameter pool W, b

$$s_r = max(s_{m \times n}),\ k_r = max(k_{1:m})$$

$$d_r = s_r + \sum_{i=1}^{m} \sum_{j=1}^{n} q_{ij}$$

$$W \sim (d_r,\ k_r),\ b \sim (d_r,)$$

Generalized RRNN for LSTM and GRU

- Let restricted weight matrix and bias be W^r_{mn} and b^r_{mn}.

$$
\begin{aligned}
W &\sim (d_r,\, k_r),\; b \sim (d_r,) \\
W^r_{mn} &= \begin{pmatrix} W[: s_{mn}, : k_m] \\ W[s_r + \sum_{\substack{1\le i\le m \\ 1\le j\le n-1}} q_{ij} : s_r + \sum_{\substack{1\le i\le m \\ 1\le j\le n}} q_{ij}, : k_m] \end{pmatrix} \\
b^r_{mn} &= \begin{pmatrix} b[: s_{mn}] \\ b[s_r + \sum_{\substack{1\le i\le m \\ 1\le j\le n-1}} q_{ij} : s_r + \sum_{\substack{1\le i\le m \\ 1\le j\le n}} q_{ij}] \end{pmatrix} \\
y^n_t &= f_n(\sum_{i=1}^{m} W^r_{mn} x^m_t + b^r_{mn})
\end{aligned}
$$

- Assuming all sharing rates, input and output channel size are the same. We have compression rate C as follows.

$$
C = \frac{P_r}{P} = \frac{P - S_r}{P} = \frac{mnd - (mn-1)s}{mnd} \approx 1 - \frac{s}{d} = 1 - r
$$

Overview

Experiments

- Language modeling on the Penn Treebank (PTB) dataset and the WikiText-2 (WT2) dataset [6, 4].
- Language modeling: make prediction of the next word based on the previous text.
- Network Architecture: three recurrent layers with 200 hidden units and 200 embedding size.
- Metric: Perplexity vs. the number of free model parameters
- High perplexity means that the model produces near-uniform random predictions, and thus is undesirable.

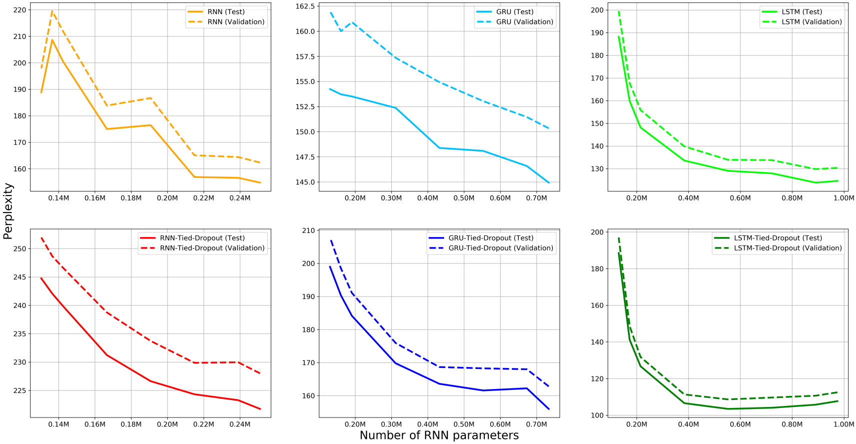
Experiments

- Comparison with state-of-the-art architectures in terms of Test Perplexity on Penn Treebank dataset.

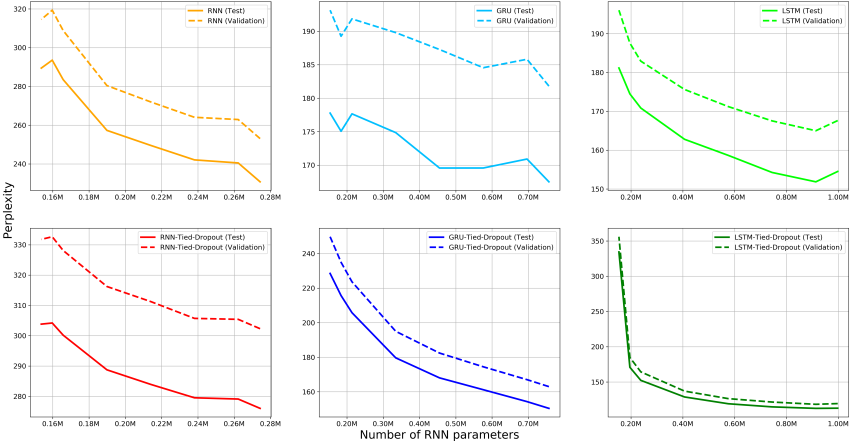
Model	Model parameters (M)	Test Perplexity
LR LSTM 200-200[2]	0.928	136.115
LSTM-SparseVD-VOC[1]	1.672	120.2
KN5 + cache[5]	2	125.7
LR LSTM 400-400[2]	3.28	106.623
LSTM-SparseVD[1]	3.312	109.2
RNN-LDA + KN-5 + cache[5]	9	92
AWD-LSTM[3]	22	55.97
RLSTM-Tied-Dropout (r=0.5)	2 (Embedding) + 0.553 (RNN)	103.5

Experiments

- Perplexity vs. Number of RNN parameters for Penn Treebank dataset.



- Perplexity vs. Number of RNN parameters for WikiText2 dataset.



Overview

Future work

- Optimized parameter restriction
  - Joint optimize sharing rate $r_{m \times n}$
  - Determine shared channel based upon network representation
- Restricted modeling for multi-modal data and multi-task application
  - We cannot retrain a new model for every new data and task.
  - Taxonomic parameter restriction based on hierarchical dependent structure of data and task
  - Restricted multi-modal autoencoder vs. conditional autoencoder

Conclusion

- We propose a novel model compression methodology called Restricted Recurrent Neural Networks (RRNN).

- Our method explicitly takes the advantage of the recurrent structures and does not require pre-training and fine-tuning of pre-trained models.

- Both extreme cases of sharing all and none parameters are not the optimal solution to model multiple dependent data. Sharing partial parameters can exploit the dependencies among inputs and greatly reduce the number of model parameters.

References

[1] Nadezhda Chirkova, Ekaterina Lobacheva, and Dmitry Vetrov. "Bayesian compression for natural language processing". In: *arXiv preprint arXiv:1810.10927* (2018).

[2] Artem M Grachev, Dmitry I Ignatov, and Andrey V Savchenko. "Compression of recurrent neural networks for efficient language modeling". In: *Applied Soft Computing* 79 (2019), pp. 354–362.

[3] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. "Regularizing and optimizing LSTM language models". In: *arXiv preprint arXiv:1708.02182* (2017).

[4] Stephen Merity et al. "Pointer sentinel mixture models". In: *arXiv preprint arXiv:1609.07843* (2016).

[5] Tomas Mikolov and Geoffrey Zweig. "Context dependent recurrent neural network language model". In: *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2012, pp. 234–239.

[6] Tomáš Mikolov et al. "Recurrent neural network based language model". In: *Eleventh annual conference of the international speech communication association*. 2010.