Sample Complexity Bounds for RNNs with Application to Combinatorial Graph Problems (Student Abstract)

Nil-Jana Akpinar,[1] Bernhard Kratzwald,[2] Stefan Feuerriegel[2]
[1]Department of Statistics and Data Science, Carnegie Mellon University, Pittsburgh, USA
[2]Chair of Management Information Systems, ETH Zurich, Switzerland
nakpinar@andrew.cmu.edu, {bkratzwald, sfeuerriegel}@ethz.ch

Abstract

Learning to predict solutions to real-valued combinatorial graph problems promises efficient approximations. As demonstrated based on the NP-hard edge clique cover number, recurrent neural networks (RNNs) are particularly suited for this task and can even outperform state-of-the-art heuristics. However, the theoretical framework for estimating real-valued RNNs is understood only poorly. As our primary contribution, this is the first work that upper bounds the sample complexity for learning real-valued RNNs. While such derivations have been made earlier for feed-forward and convolutional neural networks, our work presents the first such attempt for recurrent neural networks. Given a single-layer RNN with a rectified linear units and input of length b, we show that a population prediction error of ε can be realized with at most $\tilde{\mathcal{O}}(a^4 b/\varepsilon^2)$ samples.[1] We further derive comparable results for multi-layer RNNs. Accordingly, a size-adaptive RNN fed with graphs of at most n vertices can be learned in $\tilde{\mathcal{O}}(n^6/\varepsilon^2)$, i. e., with only a polynomial number of samples. For combinatorial graph problems, this provides a theoretical foundation that renders RNNs competitive.

Introduction

Combinatorial optimization constitutes a fundamental problem at the intersection of applied mathematics and computer science (Bello et al. 2017). Most problems of this type can be formulated as minimization or maximization tasks over finite graph structures. These optimization problems arise directly from real-world applications, such as in transportation, communication networks, or scheduling. Such problems are particularly suited for recurrent neural networks (RNNs) as we will demonstrate on the NP-hard edge clique cover number problem. The recursive processing inside RNNs allows them to naturally handle graphs of varying size and, they can numerically outperform feed-forward architectures as well as state-of-the-art heuristics.

However the theoretical understanding, especially with respect to generalization bounds for recurrent neural networks over graphs present an active field of research. A widespread approach to statistical generalization is based on the sample complexity of learning algorithms, which refers to the minimum number of samples required in order to learn a close-to-optimal model configuration. Because of that, considerable effort has been spent on deriving sample complexity bounds for various neural networks. Recent examples of bounding sample complexities involve, for instance, binary feed-forward neural networks (FNNs) (Harvey, Liaw, and Mehrabian 2017) and convolutional neural networks (Du et al. 2018). However, similar results are lacking for real-valued recurrent neural networks (RNNs). Hence, this dearth of theoretical findings motivates our research questions: *What is the sample complexity of learning recurrent neural networks? How can upper bounding the sample complexity of RNNs be applied in practical use cases?*

The edge clique cover number

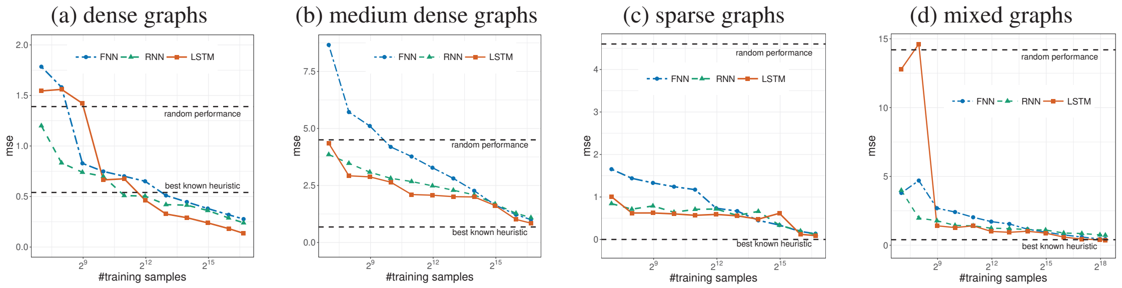
We have specifically chosen the edge clique cover number (ECCN) problem. The ECCN of a graph refers to the minimum number of cliques, i. e., fully connected sub-graphs, required to cover all edges (see our Appendix for more details). The reasons are three-fold: (1) ECCN is NP-hard and thus computationally challenging. (2) It is relevant to a variety of practical applications, including computational geometry, compiler optimization, computational statistics, protein interaction networks, and real-world network analysis. (3) Known heuristics are scarce, with the Kellerman heuristic being the notable exception (Kellerman 1973; Kou, Stockmeyer, and Wong 1978).

Performance across training sample size

We numerically evaluate how the number of training samples affects the out-of-sample performance in predicting solutions of the combinatorial problem. Results are depicted in Fig. 1. The findings justify our derivations for RNNs (as opposed to FNNs) as these appear largely superior. With sufficient training data, neural learning over graphs is on par with the Kellerman heuristic and often even outperforms it. Notably, a reasonable performance can often be achieved with as little as 4,000 training samples. This contributes to our claim of inherent approximation capabilities of the proposed neu-

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] The $\tilde{\mathcal{O}}(\cdot)$ notation indicates that poly-logarithmic factors are ignored.


(a) dense graphs

(b) medium dense graphs

(c) sparse graphs

(d) mixed graphs

Figure 1: Out-of-sample performance in predicting edge clique cover numbers. The mean squared error (mse) is reported as a function of the number of training samples (log-scale). With sufficient training samples, RNN predictions can successfully outperform a naïve baseline (i. e., majority vote) and, in some scenarios, even the state-of-the-art heuristic (Kellerman heuristic).

ral network models. In fact, a satisfactory performance in practical applications seems to require considerably fewer training samples than the derived upper bound to the sample complexity suggests.

The upper bound for RNN sample complexity

In the following, we derive an explicit sample complexity bound for learning real-valued RNNs. The results are based on a novel upper bound on the pseudo-dimension of such networks. A multi-layer version of the Theorem as well as detailed proofs of the results are given in the Appendix.

Theorem 1. *A recurrent neural network with (1) a single recurrent layer of width* a*, (2) rectified linear units, (3) input of maximal length* b*, and (4) one real-valued output unit is learnable with sample complexity that is bounded by*

$$M_L(\varepsilon,\delta) \leq \frac{128}{\varepsilon^2}\Bigg[\ln\left(\frac{16}{\delta}\right) + \ln\left(\frac{34}{\varepsilon}\right)$$
$$\times\, 4(a^2+3a+3)(2b(2a^2+4a)+4a+10+\log_2(8e))\Bigg].$$

Proof. This first requires a novel bound to $\mathrm{Pdim}(\mathcal{F})$; see supplementary materials for details. □

The previous theorem has important implications: (1) A population prediction error of ε can be obtained with at most $\tilde{\mathcal{O}}\left(a^4 b/\varepsilon^2\right)$ samples. (2) The number of required samples grows linearly with the maximum length of the input vector. (3) The sample complexity grows at most polynomially with the number of recurrent units.

Application to combinatorial graph problems

Combinatorial graph problems are particularly suited for learning with RNNs: on the one hand, the recursive processing inside RNNs allows them to naturally handle graphs of varying size and, on the other hand, RNNs can numerically outperform feed-forward architectures as we demonstrated in our numerical experiments.

Let $\mathcal{G}_n$ denote the set of undirected and unweighted graphs without self-loops and n vertices, and let $\mathcal{G}_{\leq n} = \cup_{i=1}^{n}\mathcal{G}_i$ be the respective set of graphs with up to n vertices. We aim at learning a graph problem $\eta : \mathcal{G}_{\leq n} \to \mathbb{R}$ with real-valued output based on a training data sample $\{(x_i, \eta(x_i))\}_{i=1}^{N}$.

We consider a RNN architecture with a single hidden layer and n rectified linear units. A multi-layer version of this Theorem is discussed in the Appendix. Then, based on Theorem 1, the sample complexity of the model is as follows.

Theorem 2. *A size-adaptive RNN with ReLUs in the hidden layer has at most sample complexity* $\tilde{\mathcal{O}}(n^6/\varepsilon^2)$.

Accordingly, when learning an optimal network configuration , the sample complexity grows polynomially with the size of the graphs. More precisely, this growth is in $\mathcal{O}(n^6)$.

Conclusion

Our analysis derive a novel sample complexity bound for single and multi-layer RNNs with RELUs in the hidden layers. For application to combinatorial graph problems, we propose a size-adaptive RNN that obtains a sample complexity bounded by $\tilde{\mathcal{O}}(n^6/\varepsilon^2)$, i. e. polynomially in the size of the graph. Our work is accompanied with numerical experiments that study the approximation performance: based on the example of the NP-hard edge clique cover number problem, neural learning demonstrates competitive results and frequently outperforms the best-known heuristic.

References

Bello, I.; Pham, H.; Le, Q.; Norouzi, M.; and Bengio, S. 2017. *Neural Combinatorial Optimization with Reinforcement Learning*. arXiv: 1611.09940.

Du, S. S.; Wang, Y.; Zhai, X.; Balakrishnan, S.; Salakhutdinov, R. R.; and Singh, A. 2018. How many samples are needed to estimate a convolutional neural network? In *Advances in Neural Information Processing Systems (NeurIPS)*.

Harvey, N.; Liaw, C.; and Mehrabian, A. 2017. Nearly-tight VC-dimension bounds for piecewise linear neural networks. *Proceedings of Machine Learning Research (PMLR)*.

Kellerman, E. 1973. Determination of keyword conflict. *IBM Tech. Disclosure Bull.* 16(2):544–546.

Kou, L. T.; Stockmeyer, L. J.; and Wong, C. K. 1978. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM* 21(2):135–139.