

Learning to Optimize: Training Deep Neural Networks for Interference Management

Conference Paper

Author(s):

Sun, Haoran; Chen, Xiangyi; Shi, Qingjiang; Hong, Mingyi; Fu, Xiao; Sidiropoulos, Nicholas D.

Publication date:

2018-02

Permanent link:

<https://doi.org/10.3929/ethz-b-000245053>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Learning to Optimize: Training Deep Neural Networks for Interference Management

Haoran Sun*, Xiangyi Chen*, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D. Sidiropoulos

I. INTRODUCTION

We consider an interference management problem for an interference channel consisting of K single-antenna transceiver pairs. Let $h_{kk} \in \mathbb{C}$ denote the direct channel between transmitter k and receiver k , and $h_{kj} \in \mathbb{C}$ denote the interference channel from transmitter j to receiver k . Furthermore, we assume that the transmitted symbol of each transmitter k is an independent Gaussian random variable with zero mean and variance p_k (which is also referred to as the transmission power of transmitter k). Then the signal to interference-plus-noise ratio (SINR) for each receiver k is given by

$$\text{sinr}_k \triangleq \frac{|h_{kk}|^2 p_k}{\sum_{j \neq k} |h_{kj}|^2 p_j + \sigma_k^2},$$

where σ_k^2 denotes the noise power at receiver k .

To optimally allocate power for each transmitter so that the weighted system throughput is maximized, the problem can be formulated as the following *nonconvex* problem

$$\begin{aligned} \max_{p := \{p_1, \dots, p_K\}} \quad & \sum_{k=1}^K \alpha_k \log \left(1 + \frac{|h_{kk}|^2 p_k}{\sum_{j \neq k} |h_{kj}|^2 p_j + \sigma_k^2} \right) \\ \text{s.t.} \quad & 0 \leq p_k \leq P_{\max}, \quad \forall k = 1, 2, \dots, K, \end{aligned} \quad (1)$$

where P_{\max} denotes the power budget of each transmitter and α_k denotes the nonnegative weight. Problem (1) is known to be NP-hard [1]. To obtain a good solution for problem (1), many transceiver design algorithms developed in the literature, such as WMMSE [2], SCALE [3], and the pricing algorithm [4]. A particular version of the WMMSE [5, Figure 1] also applied to solve the power control problem (1). However, optimization algorithms often entail considerable complexity, which creates a serious gap between theoretical design/analysis and real-time processing.

II. THE LEARNING TO OPTIMIZE APPROACH

To resolve the computational issues arisen on the above interference management problem with stringent real-time requirements, we design a new ‘learning to optimize’ based framework as shown in Figure 1. The main idea is to treat a given algorithm as a ‘black box’, and try to *learn* its input/output relation by using a *deep neural network* (DNN) [6]. If the nonlinear mapping can be learned accurately by a DNN of moderate size, then the interference management tasks can be performed in almost *real time* – since passing the input through a DNN only requires a small number of simple operations.

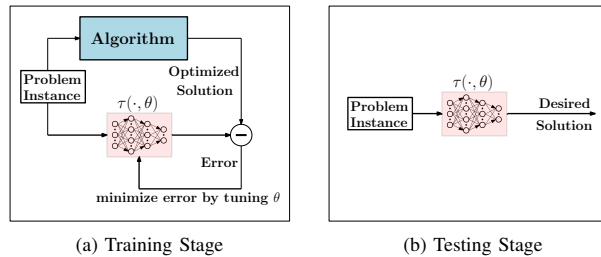


Fig. 1: The Proposed Method. The key idea is to treat the input and output of an algorithm as an unknown nonlinear mapping and use a DNN to approximate it. In the figure $\tau(\cdot, \theta)$ represent a DNN parameterized by θ .

Unlike all existing works on approximating optimization algorithms such as those using unfolding [7]–[10], our approach is justified by rigorous theoretical analysis. We show that there are conditions under which an algorithm is learnable by a DNN [5], and indicate that it is possible to learn a well-defined optimization algorithm very well by using finite-sized deep neural networks. To concisely state the result, let us make the following definitions. Given an input channel vector $h := \{h_{ij}\} \in \mathbb{R}^{K^2}$, let us use $v(h)_i^t$ to denote the variable v_i at t^{th} iteration generated by WMMSE [5] (which basically represents $\sqrt{p_i}$ at t^{th} iteration). Also let $H_{\min}, H_{\max} > 0$ denote the minimum and maximum channel strength and let $V_{\min} > 0$ be a given positive number. Let $NET(x, z)$ represent a neural network with (x, z) as input.

Theorem 1 *Suppose that WMMSE is randomly initialized with $(v_k^0)^2 \leq P_{\max}$, $\sum_{i=1}^K v(h)_i^0 \geq V_{\min}$, and it is executed for T iterations. Define the following set of ‘admissible’ channel realizations*

$$\mathcal{H} := \left\{ h \mid H_{\min} \leq |h_{jk}| \leq H_{\max}, \forall j, k, \sum_{i=1}^K v(h)_i^t \geq V_{\min}, \forall t \right\}.$$

Given $\epsilon > 0$, there exists a neural network with $h \in \mathbb{R}^{K^2}$ and $v^0 \in \mathbb{R}_+^K$ as input and $NET(h, v^0) \in \mathbb{R}_+^K$ as output, with the following number of layers

$$O\left(T^2 \log \left(\max \left(K, P_{\max}, H_{\max}, \frac{1}{\sigma}, \frac{1}{H_{\min}}, \frac{1}{P_{\min}} \right) \right) + T \log \left(\frac{1}{\epsilon} \right)\right)$$

and the following number of ReLUs and binary units

$$O\left(T^2 K^2 \log \left(\max \left(K, P_{\max}, H_{\max}, \frac{1}{\sigma}, \frac{1}{H_{\min}}, \frac{1}{P_{\min}} \right) \right) + TK^2 \log \left(\frac{1}{\epsilon} \right)\right),$$

TABLE I: Sum-Rate and Computational Performance for IMAC

# of base stations and users (N,K)	average sum-rate (bit/sec.)			total CPU time (sec.)			
	DNN	WMMSE	DNN/WMMSE	DNN	WMMSE (MATLAB)	WMMSE(C)	DNN/WMMSE (C)
(3, 12)	17.722	18.028	98.30%	0.021	22.33	0.27	7.78%
(3, 18)	20.080	20.606	97.45%	0.022	42.77	0.48	4.58%
(3, 24)	21.928	22.648	96.82%	0.025	67.59	0.89	2.81%
(7, 28)	33.513	35.453	94.53%	0.038	140.44	2.41	1.58%
(20, 80)	79.357	87.820	90.36%	0.141	890.19	23.0	0.61%

such that the relation below holds true

$$\max_{h \in \mathcal{H}} \max_i |(v(h)_i^T)^2 - NET(h, v^0)_i| \leq \epsilon \quad (2)$$

Remark 1 The bounds in Theorem 1 provide an intuitive understanding of how the size of the network should be dependent on various system parameters. A key observation is that having a neural network with multiple layers is essential in achieving our rate bounds. Another observation is that the effect of the approximation error on the size of the network is rather minor [the dependency is in the order of $\mathcal{O}(\log(1/\epsilon))$]. However, we do want to point out that the numbers predicted by Theorem 1 represent some upper bounds on the size of the network. In practice, much smaller networks are often used to achieve the best tradeoff between computational speed and solution accuracy.

III. NUMERICAL RESULTS

To demonstrate the achievable performance of the proposed approach, a multi-cell interfering multiple Access Channel (IMAC) model is considered with a total of N cells and K users. In each cell, one BS is placed at the center of the cell and the users are randomly and uniformly distributed in the area; The channel between each user and each BS is randomly generated according to a Rayleigh fading distribution; see [5] for more detail. We perform the training and testing following the procedures outlined in Figure 1 and summarize the testing results in TABLE I. It can be seen that the proposed DNN approach can be trained to well-approximate the behavior of the state-of-the-art algorithm WMMSE [2], and achieve relatively high sum-rate performance. It is also shown that DNNs can achieve orders of magnitude speedup in computational time compared to state-of-the-art power allocation algorithms based on optimization.

Note that in the table, WMMSE (C)/(MATLAB) represents the WMMSE algorithm implemented using either C or MATLAB. The proposed DNN approach is implemented in Python 3.6.0 with TensorFlow 1.0.0 on one computer node with two 8-core Intel Haswell processors, two Nvidia K20 Graphical Processing Units (GPUs), and 128 GB of memory. The GPUs are used in the training stage to reduce the training time, but are *not* used in the testing stage.

REFERENCES

[1] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, 2008.
 [2] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[3] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3711–3724, 2009.
 [4] D. Schmidt, C. Shi, R. Berry, M. Honig, and W. Utschick, "Distributed resource allocation schemes," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 53–63, 2009.
 [5] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Signal Processing Advances in Wireless Communications (SPAWC), 2017 IEEE 18th International Workshop on*. IEEE, 2017.
 [6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
 [7] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 399–406.
 [8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Science*, vol. 2, no. 1, pp. 183–202, 2009.
 [9] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.
 [10] P. Sprechmann, R. Litman, T. B. Yakar, A. M. Bronstein, and G. Sapiro, "Supervised sparse analysis and synthesis operators," in *Advances in Neural Information Processing Systems*, 2013, pp. 908–916.