Parameters Tuning Experience for NCS

Project 2 Report of

CS303 Artificial intelligence

Department of Computer Science and Engineering

BY

Ting Sun

11710108

Southern University of Science and Technology

December 2019

Contents

Main Idea of NCS

As said in the paper of Tang et al. [1], the core idea of NCS is a new model for implementing the cooperation between individuals in a population, which was inspired by an interpretation of cooperation in human behaviors. That is, when a team of people is tackling a complex task, members of the team tend to work cooperatively by handling different parts of the task and communicate to avoid multiple members working on the same part. Analogously, NCS comprises multiple search processes. The search processes are run in parallel and strive to find better candidate solutions, while information is shared to explicitly encourage each search process to emphasize the regions that are not covered by others [1].

Applications of NCS

The applications of NCS are very comprehensive, for NCS is designed for general solution search in the solution space and NCS can be used in both discrete form and differentiable form. Therefore, we can use NCS in big data, machine learning and so on. For instance, it can be used in minimizing the Symbol-Error-Rate for Amplify-and-Forward Relaying Systems [1]. NCS algorithm is very promising.

Main Idea of OLMP

OLMP is derived from LMP [2][3], and LMP is derived from MP [4]. The idea of MP is that cutting off some weight-smaller-than-threshold edges in the model can compress the trained model and hopefully it will do little harm to the accuracy of the model [4]. However, LMP says that thresholds should be selected lawyer by lawyer, rather than uniformly [2][3]. And typically LMP thresholds are selected manually.

The main idea of OLMP is that selecting thresholds manually is not easy for a green hand and is really labor-consuming [5]. Thus, thresholds can be selected by using some advanced algorithms, such as NCS. Also, OLMP introduces some approaches like error-correction for concrete problems during the detailed processing [5].

Applications of OLMP

As mentioned before, OLMP can cut off edges in the trained model automatically and thus cutting off edges from the model will not be be a disaster for a green hand. And after this pruning, the size of models will be considerably decreased. In addition, empirical verification shows that pruning model in this way can be efficient and the accuracy loss of the model is bearable [5]. Therefore, a model which is be pruned by OLMP can be used in many AI-model-needed occasions which are limited by memory.

Also, the thought of OLMP can be used in many fields, such as the control theory, the information theory, and so on. For example, in the view of control theory, it is often a disaster if the population is too large. However, we can use the brilliant thought of OLMP and decrease some less important population automatically in appropriate approaches similar with approaches described in OLMP, which guarantees both efficiency and correctness of the control model.

Parameter Description

Final Values & Results

Parameter	Best submission of F6
lambda	1
r	0.888
epoch	10
n	3
Final Result	390.00023976219467
Running Time/s	50.31319522857666

Parameter	Best submission of F12
lambda	1.1414696798881185
r	0.6816493207741794
epoch	251
n	3
Final Result	-459.9999284265068
Running Time/s	56.86451506614685

Parameter	Best submission of F29
lambda	1.0000102
r	1
epoch	10
n	92
Final Result	0.9889914106747684
Running Time/s	60.56613612174988

Roles of the Parameters

Parameter	meaning
lambda	Balance factor of exploration and exploitation. It determines the relative weights of good performance and new area exploration.
r	Converge factor. It determines the rising speed of the difficulty of new parameters substitution during the search.
epoch	Period of RLS correction. It determines the period interval length of RLS correction.
n	Number of searching individuals. It determines the number of individuals working for the search.

effect about different values of the parameter

Parameter	effect
lambda	We can achieve good performance even we just always set lambda as 1.
r	Usually the more significant figure r has, the longer time the algorithm needs.
epoch	Usually it is better to set epoch more than 10.
n	Usually the bigger n is, the more time the NCS will cost.

Best Range

It seems that in F6, F12, all the parameters change very fast. They are coupling tightly and even a little bit change will result in a totally different performance. In addition, in F6 and F12, you can get similar results even you choose totally different 4 parameters. Still, it seems that when r belongs to [0.8,0.99], F6 can achieves good performance. My best submission is just one of the good parameters and other parameter may achieve similar or better performance than mine, even they are far away from mine. Also, I found several pairs such parameters myself during the project.

However, things are different in F29, where experiments show that only n can determine the result and the other three parameters can only influence the running time. And after tried all the numbers from 1 to 100 for n, I found that when n=92, it could achieve best performance.

Tuning Procedure

Basically, in the F6, F12 and F29, I used multi-parametel estimation of waterquality model by network search method (aka grid search) and random algorithm.

F6

At the beginning, I chose parameters manually and just wanted to find some details about this data set.

During this time, I found that the search space is too large to do traversal and I had to choose some promising domains to further search. Therefore, I set some step lengths for r and lambda, trying to pseudo-iterate the promising search space which is chosen in the early time.

After a whole day, I used my PC and got some parameters which achieved good performance.

Then I chose the neighborhood of these parameters and set a smaller step. After I did this several times recursively, I finally got my best result.

F12

In F12, at the beginning I just did as what I did in F6 and I did get some good parameters.

However, some day I heard that random algorithm in precise domain can be helpful in F12. Thus, I stopped to use random algorithm in the neighborhood of these selected domains and achieved better performance.

F29

Things were totally different in F29, where after I tried some parameters manually, I realized that only n mattered in F29.

After that, I tried from n=1 to n=100 and found that I could achieved the best performance when n=92.

In this way, I ranked top20 at that time, which made me believed that this is the case and I should get a rest, rather than bother to install Ubuntu operating system and something else in my poor PC and adjust these parameters again and again.

Reference

[1] K.Tang, P. Yang and X. Yao, "Negatively Correlated Search," *IEEE Journal on Selected Areas in Communications*, 34(3): 1-9, March 2016.

[2] Y.-W. Guo, A.-B. Yao, and Y.-R. Chen. Dynamic network surgery for efficient dnns. In *Advances in Neural Information Processing Systems 29 (NIPS'16)*, pages 1379–1387, Barcelona, Spain, 2016.

[3] S.Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems 28 (NIPS'15),* pages 1135–1143, Montreal, Canada, 2015

[4] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems 2 (NIPS'89)*, pages 598–605, Denver, CO, 1989

[5] Guiying Li, Chao Qian, Chunhui Jiang, Xiaofen Lu, and Ke Tang. Optimization based Layer-wise Magnitude-based Pruning for DNN Compression. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, Stockholm, Sweden, 2018, pp.2383-2389.