Hyperparameters of Multilayer Perceptron with Normal Distributed Weights

Y. Karaki[a,]* **and N. Ivanov**[b,]**

[a] *Department of Computer Science, Faculty of Sciences and Fine Arts,*
Arts Sciences and Technology University in Lebanon, Cola, Beirut, 14-6495 Lebanon

[b] *Department of Computing Machinery, Faculty of Computing Systems and Networks,*
Belarusian State University of Informatics and Radioelectronics, Minsk, 220013 Republic of Belarus

** e-mail: youmna_karaki@yahoo.com*

*** e-mail: ivanovnn@gmail.com*

Abstract—Multilayer Perceptrons, Recurrent neural networks, Convolutional networks, and others types of neural networks are widespread nowadays. Neural Networks have hyperparameters like number of hidden layers, number of units for each hidden layer, learning rate, and activation function. Bayesian Optimization is one of the methods used for tuning hyperparameters. Usually this technique treats values of neurons in network as stochastic Gaussian processes. This article reports experimental results on multivariate normality test and proves that the neuron vectors are considerably far from Gaussian distribution.

Key words: neural networks, hyperparameters, Gaussian distribution, Bayesian optimization

DOI: 10.1134/S1054661820020054

INTRODUCTION

Neural Network has been a booming field recently as many industries have been disrupted by its influx. In the last few years, Neural Networks have created a significant impact in many areas such as autonomous driving, pattern recognition, big data classification, and computer vision.

In deep learning projects, tuning hyperparameters is the key to reduce computing time providing reasonable error. Hyperparameters include the number of network layers, nodes in each layer, the activation function, and other characteristics for specific neural networks. In general, hyperparameters determine the structure of neural network and how it is trained. The problem of hyperparameters optimization arose together with first perceptron; for instance, a monograph was published in 1996 [1]. There are several common approaches for statement formalization; in fact they do not provide a priori error. Mathematical model for both deterministic and stochastic nonlinear optimization problem with constrains may be applied to hyperparameters estimation. The formal model is formulated as [2]:

$$\arg\min_{x}\{f(x,D)\,|\,x\in X\},\tag{1}$$

where X is a feasible set of hyperparameters, D is a learning set, $f(x, D)$) is the estimated performance of x over set D.

Mathematical model (1) is abstract, useless, and doesn't ensure solution for real problem. Unfortunately, this optimization problem includes implicit function $f(x, D)$ and cannot be solved with ordinary methods.

HYPERPARAMETERS OPTIMIZATION

Contemporary methods for solving hyperparameters optimization problem are grid search, random search, gradient approach, evolutionary algorithms, and Bayesian optimization. Grid search [3] is naïve method checking all feasible arguments on selected grid. Objective function is estimated for each sample and optimal value is selected. If several kinds of parameters are taken into consideration, then the grid is a set of multidimensional vectors. Computational complexity of grid search depends exponentially upon dimension of vectors.

Random search [4, 5] for estimation of neural network hyperparameters is an extension of the grid search. A statistical distribution for argument x from formula (1) has to be estimated. In comparison with grid search, the random search has much better convergence due to focusing upon more important hyperparameters. But for the random search, preliminary statistical investigation for hyperparameters must be implemented.

Evolutionary algorithms are optimization methods that originated from genetic science. They exploit two main genetic concepts, namely crossing and mutation. Evolutionary algorithm, on the base of two (or more)

Received January 8, 2020; revised January 15, 2020; accepted January 16, 2020

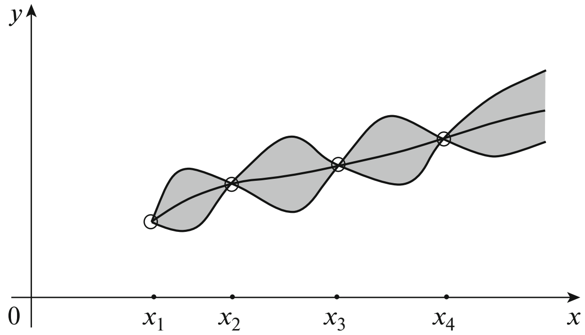


Fig. 1. Estimating an objective function $f(x, D)$ with a 1-dimensional continuous parameter x.

feasible solutions of the problem under consideration, constructs new solution improving target function. First, these two solutions are crossing, and second, this new solution is mutated**.** The process of generating new solution is similar to operation of crossover and mutation in genetics. Evolutionary hyperparameter search follows the biological concept. Initial set that named initial population contains neural networks with random generated hyperparameters [6, 7]. Algorithm checks fitness of each element of population and replaces the worst elements with new ones. New sample is NN generated through evaluation; that is crossover and mutation operations of arguments. These arguments are the components of vector x from model (1), e.g. number of layers and number of nodes in each layer. Target function is a percentage of correct classified samples. The algorithm terminates when the learning process of new sample does not yield improvement with preset error.

Bayesian optimization [8, 9] is applied to find parameters of derivative-free stochastic function. It consists of two main components: an algorithm for stochastic parameters, usually Gaussian regression; and an acquisition function for finding argument for additional sample that has to improve prediction based on regression. Usual mathematical model of the method is Gaussian process with posterior Bayesian probability formula estimating expecting values for $f(x, D)$ in model (1) at a desired argument x. When a new point is observed, the parameters of posterior distribution are refreshed. An acquisition function is used for measuring the value that would be generated by evaluation of the objective function at a new point x, taking into consideration the current distribution over $f(x, D)$. Figure 1 illustrates some kind of confidence intervals for values $f(x, D)$ at any point x. Argument x consists of NN hyperparameters, axe y may depict percentage of correct recognized instances by NN. Points x^i are ones for which objective function of the problem (1) was obtained with experiment, an interval estimating procedure is applied for function y at other points x. Extrapolated confidence interval for $f(x, D)$ values where $x > x^4$ provides predicting result for NN performance.

Table 1. Datasets for computing experiment

No.	Name	No. Instances	No. Attributes	No. Classes
1	Adult	4884	14	8
2	Arrhythmia	452	9	5
3	Dermatology	366	33	7
4	Glass identification	214	10	7
5	Hepatitis	155	19	5
6	Lymphography	148	18	6
7	Student performance (3 targets)	649	33	8
8	Wine	178	13	5

TEST FOR VECTORS ON NEURAL NETWORK LAYERS

It is well known that normal distribution is widespread in engineering data. Normal distribution and Gaussian stochastic process had been scrutinized in XIX century. Gaussian density function has just two parameters. Normal distributed dataset is investigated with deep elaborated mathematical theory. The aim of this experiment is testing vectors on NN layers on multivariate normality.

Recognized repository of University of California Irvine [10] was selected for our experiment with deep learning perceptron. Namely 10 classification problems were under investigation. Nonnumeric coding of attributes was changed with nonnegative numbers. Omitted attribute parameters were interpolated. Dataset list is available in Table 1.

Let n be number of NN layers, L_1 is the input layer, L_n the output one, and $L_2, ..., L_{n-1}$ are the hidden layers. The question that imposes itself: Are the vectors of learning dataset on hidden layers normal distributed?

If the answer to this question is positive, then transformation of initial input set of vectors into output resulting vector of NN can be fully defined as discrete stochastic Gaussian process realized by sequence of normal vectors:

$$x^1, \ldots, x^i, x^{i+1}, \ldots, x^n. \quad (2)$$

Initial data for NN may consist of real, integer, categorical values that definitely cannot have normal distribution. Following layers of NN are a mixture of these values. Distributions of vectors x^i were compared with Gaussian ones. Henze multivariate normality test with confidence level 0.95 was applied. None of datasets successfully passed test on Gaussian distribution. To sum it up, multilayer perceptron didn't fit Gaussian distribution at any of the hidden layers.

In statistics, normality tests are used to verify whether data sets can be considered as Gaussian distributed samples. The tests are a base for a model construction, and can be explained by several ways, depending on the problem of interest. Pearson chi-square or Kolmogorov–Smirnov tests are exploited for single dimension Gaussian distribution validity. These tests are hardly modified for multidimensional variate testing. It seems that a simple normality test for random vector may be accomplished by linearly transforming normal vector into independent one-dimensional random values. Then independent values have to be tested by standard procedures. However, this transformation can distort initial vector; and the result may be false.

Multivariate normality statistical tests had been developed in the last four decades [11–14]. L. Baringhaus and N. Henze [15] proposed to estimate the difference between the sample characteristic function and the theoretical one, which is Gaussian. In contradiction with one dimensional, K.V. Mardia [11, 12] tests the multivariate normality by estimating two random characteristics of given sample, namely kurtosis and skewness. Verification tables for these parameters were developed [12]. The statistical test is a function with cumbersome expression containing integration over probabilistic measure. For estimation of this statistic, Monte-Carlo method is exploited.

Learning procedure with feedback error correction had been implemented for each sample; then neural networks with tuned weights were applied to them. A vector set on layers corresponding to each dataset was verified. Standard value 0.05 of significance level was exploited. The verification test produced two statistics, namely kurtosis and skewness measures to check if vector set complies with Gaussian distribution. In fact, it occurs that for all hidden layers, no vector set fits Gaussian distribution.

CONCLUSION AND FUTURE WORK

The carried out experiment suggests that the vectors corresponding to neurons of multilayer perceptron networks hardly fit to such simple structure as Gaussian multivariate normality.

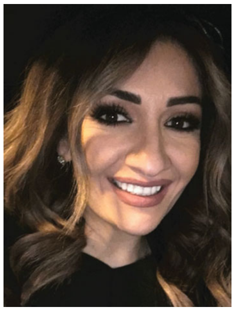
For hyperparameter optimization problem, values of vectors corresponding to layers have specific distributions; their probability density function depends on input dataset and neural network design as well. Predicting hyperparameter values of neural network with Bayesian optimization method may be exploited, but keeping in mind possible discrepancy.

CONFLICT OF INTERESTS

The authors declare that they have no conflicts of interest.

REFERENCES

1. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proc. 2014 IEEE Conf. on Computer Vision and Pattern Recognition* (*CVPR 2014*) (Columbus, OH, USA, 23–28 June 2014), pp. 1725–1732.
2. G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, “An effective algorithm for hyperparameter optimization of neural networks,” IBM J. Res. Dev. **61** (4/5), 9.1 – 9.11 (2017).
3. T. Domhan, J. T. Springenberg, and F. Hutter, “Speeding up automatic hyperparameter optimization of Deep Neural Networks by extrapolation of learning curves,” in *Proc. 24th Int. Joint Conf. on Artificial Intelligence* (*IJCAI 2015*) (Buenos Aires, Argentina, 25–31 July 2015), pp. 3460–3460.
4. J. Bergstra and B. Yoshua, “Random search for hyper-parameter optimization,” J. Mach. Learn. Res. **13** (2), 281–305 (2012).
5. J. Bergstra, D. Yamins, and D. D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures,” in *Proc. 30th Int. Conf. on Machine Learning* (*ICLM’13*) (Atlanta, Georgia, USA. 16–21 June 2013), JMLR: W&CP **28**, I-115–I-123 (2013).
6. D. Orive, G. Sorrosal, C. E. Borges, C. Martin, and A. Alonso-Vicario, “Evolutionary algorithms for hyperparameter tuning on neural networks models,” in *Proc. 26th European Modeling and Simulation Symposium* (*EMSS 2014*) (Bordeaux, France, 10-12 September 2014), pp. 402–410.
7. E. Bochinski, T. Senst, and T. Sikora, “Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms,” in *Proc. 2017 IEEE Int. Conf. on Image Processing* (*ICIP 2017*) (Beijing, China, 17-20 September 2017), pp. 3924–3928.
8. J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” in *Advances in Neural Information Processing Systems 25*: *Proc. Conf. NIPS 2012* (Lake Tahoe, Nevada, USA, 3-6 December 2012), pp. 2951–2959.
9. J. Lampinen and A. Vehtari, “Bayesian approach for neural networks — review and case studies,” Neural Networks **14** (3), 257–274 (2001).
10. Machine Learning Data Sets, University of California, Irvine. https://archive.ics.uci.edu/ml/datasets.php (Retrieved January 6, 2020).
11. K.V. Mardia, “Measures of multivariate skewness and kurtosis with applications,” Biometrika, **57** (3), 519–530 (1970).
12. K.V. Mardia, “Applications of some measures of multivariate skewness and kurtosis for testing normality and robustness studies,” Sankhyā: Indian J. Stat. Ser. B **36** (2), 115–128 (1974).
13. N. Henze. “Invariant tests for multivariate normality: a critical review,” Stat. Pap. **43** (4), 467–506 (2002).
14. J. E. Gentle, *Computational Statistics* (Springer, New York, 2009), pp. 315–316.
15. L. Baringhaus and N. Henze. “A consistent test for multivariate normality based on the empirical characteristic function,” Metrika **35**, 339–348 (1988).



Youmna Karaki (born in 1983) has graduated from Arts, Sciences, and Technology University In Lebanon in 2005 where she got her Masters degree in Computer Science. She is now a PhD student in the field of Artificial Neural Networks at Belarusian State University of Informatics and Radioelectronics, Minsk.

Y. Karaki has published 2 articles for now. She has more than 15 years of teaching experience at different Lebanese Universities. She is currently working as an Instructor at Arts, Sciences, and Technology University in Lebanon.

Nick Ivanov (born in 1949) has graduated from Belarusian State University in 1972; his specialty is applied mathematics. His fields of interest are network security and artificial neural networks.

N. Ivanov has published 1 monograph and more than 70 papers. He works now as an Associate Professor at Belarusian State University of Informatics and Radioelectronics.