Determining the Neural Network Topology: A Review

Muh. Ibnu Choldun R.
School of Electrical Engineering and Informatics , Institut Teknologi Bandung
Jl. Ganecha 10, Bandung
Indonesia
+6281321859867
ibnucholdun@poltekpos.ac.id

Judhi Santoso
School of Electrical Engineering and Informatics, Institut Teknologi Bandung
Jl. Ganecha 10, Bandung
Indonesia
+6281572283045
judhi@stei.itb.ac.id

Kridanto Surendro
School of Electrical Engineering and Informatics, Institut Teknologi Bandung
Jl. Ganecha 10, Bandung
Indonesia
+628122107224
endro@informatika.org

ABSTRACT
One of the challenges in the successful implementation of deep neural network (DNN) is setting the value for various hyper-parameters, one of which is the network topology, which is closely related to the number of hidden layers and the number of hidden neurons. Determining the number of hidden layers and the number of neurons is very important and has a large influence on DNN performance. Determining these two numbers manually (usually through trial and error methods) to find fairly optimal arrangement is a time-consuming process, while the automatic approach is divided into two, they are a model-based approach and a non-model based approach. The non-model-based approach, for example, is grid search and random search, whereas model-based approaches, for example, are using particle swarm optimization (PSO) algorithms. In some researches, how to determine the number of hidden layers or number of neurons, often the guidelines are unclear, even the roles and functions of both are explained minimally. Although it is still a difficult area of research, research to determine the number of hidden layers and the number of neurons must continue to be carried out, because these two numbers will greatly determine the performance of DNN.

CCS Concepts
• **Computing methodologies→ Neural networks**

Keywords
Neural network; number; hidden layer; hidden neuron

1. INTRODUCTION
Neural networks have been successfully applied in a variety of problem areas as diverse as computer science, finance, medicine, engineering, physics, etc. The main reason is that neural networks have the ability to approach arbitrary functions. Over the past 30 years, a number of results have been published which show that artificial neural networks called feedforward networks with one hidden layer can approach all functions arbitrarily. One important aspect for designing neural networks is architecture or topology because it is closely related to generalization capabilities. In some cases using a fully connected network, and given a set of predetermined inputs and outputs, the topology will be determined by how many hidden layers and how many neurons for each hidden layer. The majority of the literature discusses how to determine the number of hidden neurons (assuming only using one hidden layer), but very rarely discusses how to determine the optimal number of hidden layers. This is due to the opinion that networks with only one hidden layer are sufficient to approximate universally almost all functions [1][2]. However, several researches show that the use of two hidden layers provides better performance than if only using one hidden layer in some cases [2][3]. Since the increase in computer capabilities, the use of a neural network that has more than one hidden layer has become one of the attractions for researchers, especially since the use of deep neural networks to solve problems in the real world. Deep neural network learning can be interpreted as a technique that uses neural networks for learning that utilize many hidden layers between input and output layers [4]. One of the challenges in the successful implementation of deep neural networks is setting values for various hyperparameters, one of which is the network topology, which is closely related to the number of hidden layers and the number of hidden neurons. The determination of the number of hidden layers and the number of neurons is very important and has a large influence on the performance of deep neural networks [5][6]. The determination of these two numbers manually (usually through the 'trial and error' method) to find a fairly optimal topology is a time-consuming process.

Some researches on neural network topology have focused on determining the number of neurons because they only use one hidden layer, some focuses on the number of hidden layers (two or one hidden layer), some also determine the number of hidden layers and the number of neurons in each hidden layer. Research on determining the number of neurons hidden has been going on since the 1990s and is still an interesting topic for researchers [7][8][9]. Research comparing the performance of one or two hidden layers is still an interesting topic to date [3][10], while research which also calculates the number of hidden layers and the number of new neurons has been done in recent years, since the emergence of deep learning [4][5][10]. The determination of the right number of neurons is important to avoid under-fitting or over-fitting, while increasing the level of accuracy of the neural network. The choice of the number of hidden layers and the right number of neurons together aims to reduce the complexity of processing time while simultaneously maintaining the accuracy of the neural network [5]. Hidden layers in deep neural networks represent increasingly complex features progressively [11]. Some approaches to calculating optimal number of hidden neurons are: 'trial and error' [7], rule of thumb method, simple method, and two phase method [12]. Madhiarasan and Deepa (2017) propose a new

ICSCA '19, February 19–21, 2019, Penang, Malaysia

ACM ISBN 978-1-4503-6573-4/19/02…$15.00
https://doi.org/10.1145/3316615.3316697

method for calculating the number of neurons by registering as many as 151 functions of convergent functions and then simulating each of these functions to get the number of neurons that will produce a minimum error. Of the 151 functions, a function is finally chosen which produces a minimum error [13]. The hyper-parameter search approach for searching the number of hidden layers and the number of neurons on the deep neural network can be done manually or automatically. The manual approach is done by trial and error or rule of thumb, while the automatic approach is divided into two, they are a model-based approach and a non-model based approach [14]. Non-model based approaches, for example, are grid search and random search, whereas approaches that are model based, for example, are using particle swarm optimization (PSO) algorithms.

Deep neural networks have achieved great success in practice and have a great influence on the machine learning literature and artificial intelligence. Although it has achieved success on a practical level, the theoretical characteristics of determining the topology of a neural network are still being investigated. For example, methods for finding the number of hidden layers and the number of neurons in deep learning have been done, but until now it has not been based on established theories [5][15][16][17]. Many methods have been conducted to calculate the number of hidden layers and the number of neurons, for example: model-based automatic methods using PSO [4][14][18], automatic methods without models using grid search (GS) or random search [19][20]. Some manual methods are also proposed, for example: the number of hidden neurons is 2/3 of the number of inputs plus the number of outputs, the number of hidden neurons per hidden layer follows the rules of the pyramid geometry, mathematical approach [21][22], trial and error/rule of thumb[23], the relationship between the number of hidden layers and the number of hidden layers is logarithmic [24]. In some of the above literature, how to determine the number of hidden layers or optimal number of neurons, often the guidelines are not clear, even the roles and functions of both are explained minimally. Some literature proposes methods or ways to determine the number of hidden layers or the number of neurons, but it does not generally apply, it depends on the type of input data and also depends on the output. Research in this area still leaves difficult research tasks [17]. Some of the methods mentioned, apparently cannot be applied to different types of data. Each researcher determines the number of hidden layers or the number of neurons based on past experience, while for beginner researchers even do it with 'trial and error'. Although it is still a difficult area of research, research to determine the number of hidden layers and the number of neurons must continue to be carried out, because these two numbers will greatly determine the deep neural network learning performance.

2. LITERATURE REVIEW

2.1 Determining the Number of Hidden Neurons

Summary and critics from researches calculating hidden neuron can be seen in table 1, which was presented chronologically from the 1990s to the present.

Table 1. Research determines the number of hidden neurons

No	Researcher	Function	Method	Critics
1	Sartori and Antsaklis(1991)[25]	Classification	the number of neurons is p-1, where p is the number of input features	No correlation checks were carried out between input features
2	Arai(1993)[26]	Classification of two categories	the number of neurons between I-1 to I / 3, where I is the number of input features	Only for classification of two categories
3	Ji Yan Li(1995)[27]	Time Series	m hidden neurons where the amount of data is (m + 1)m/2	Just pay attention to the amount of data, variations in data are not noticed
4	Hagiwara(1994)[28]	Classification: Determines a 4x4 pixel image whether symmetry	Reducing the number of neurons and weights gradually	There is no guarantee that the number of neurons will get the right weight
5	Tamura and Tateishi(1997)[29]	Is not explained specifically	For one hidden layer: the number of neurons is N-1, for two hidden layers the number of neurons is N/2 + 3 where N is the number of inputs minus output	There is no empirical test Ignoring correlation, variations between input features, and between input features and output features
6	Fujita(1998)[30]	Not specifically explained	Add one by one the number of neurons, where the number of neurons is proportional to the number of data sets	Ignoring correlation, variations between input features
7	Keeni et al.(1999)[31]	Classification	Reducing the number of neurons little by little	Ignoring correlation, variations between input features

No	Researcher	Function	Method	Critics
8	Onoda(2001)[32]	Prediction	Using the Neural Network lnformation Criterion(NNIC)	Less practical
9	Islam and Murase(2001)[33]	Not specifically explained	Cascade Neural Network Design Algorithm(CNNDA)	Less practical
10	Zhang et al.(2003)[34]	Not specifically explained	Set Covering Algoritm(SCA)	Less practical
11	Choi(2008)[35]	Not specifically explained	An algorithm that separates training for input and output	Less practical
12	Shibata and Ikeda(2009)[36]	Not specifically explained	$N^{(h)} = N^{(i)}N^{(o)}$) and $\eta = 32/(N^{(i)}N^{(o)})^{1/2}$	Do not see variations and correlations
13	Doukim et al.(2010) [37]	Pre-processing for feature extraction	Searching algorithm	Ignoring correlation, variations between input features
14	Panchal et al.(2011) [38]	Not specifically explained	The number of hidden neurons is inversely proportional to MSE	The number of neurons is known after learning
15	Hunter et al.(2012)[39]	Not specifically explained	$N^h=2^n-1$	Do not see variations and correlations
16	Sheela and Deepa(2013)[7]	Predictive	$(4n^2+3)/(n^2-8)$	Do not see variations and correlations
17	Madhiarasan and Deepa(2016)[8]	Predictive	$(8n-4) / (n-4)$	Do not see variations and correlations
18	Madhiarasan and Deepa(2017)[13]	Predictive	$u_n = (4n-2) / (n-3)$	Do not see variations and correlations

2.2 Comparison Using One or Two Hidden Layers

Research that compares the use of one or two hidden layers is performed on univariate and multivariate functions, which can be briefly read in table 2.

Table 2. Research compares one or two hidden layer

No	Researcher	Function	Method	Critics
1	Nakama (2011) [40]	Linear and quadratic	Comparing one and two hidden layers with other parameters together: one hidden layer reaches convergence faster	More than two hidden layers are not compared
2	Thomas et.al (2016) [10], Thomas et.al (2017) [2]	Predictive	Compare one and two hidden layers: two better hidden layers	More than two hidden layers are not compared
3	Guliyev and Ismailov (2018) [1]	Multivariate functions	Comparing one and two hidden layers: one hidden layer is less able to approach multivariate functions	More than two hidden layers are not compared

2.3 Determining The Number of Hidden Layers and The Number of Neurons

To determine the number of hidden layers and the number of neurons together broadly divided into two, they are the determination of automatically and manually determination [18]. Determination automatically can use the model, or non-model. There are manual determinations that use certain formulas; some are done by trial and error. Automatic determination without a model consists of grid search (GS) and random search (RS). GS is generally used to optimize DNN parameters only if the number is very low. First, users choose a range of values to explore. Then, DNN is trained for each specification along with other parameters. Usually, GS is carried out for the steps of a logarithmic scale, where the best combination is estimated, progressively in conducting a search [19]. The obvious disadvantage of GS is its time complexity - for parameters k to take different values for each (assuming that n is the same for all parameters), complexity grows exponentially at level $O(n^k)$. Random search (RS) is an alternative to GS, which integrates faster into parametricization. RS is not adaptive, meaning it is not dynamically updated during the experiment (solutions that have been found do not affect search). There is also a hybrid algorithm that combines RS with other techniques to improve its performance, for example, manual

updates provided by experts [19][20].Model-based algorithms build replacement models of hyper-parameter spaces. Then, the hyper-parameter is set using this model. Most of these techniques use the Bayesian regression model that turns this problem into a trade-off between exploration, and exploitation. One model-based algorithm is using particle swarm optimization (PSO), an optimization technique by continuously calculating potential solutions using a quality reference. This algorithm optimizes problems by moving particles/prospective solutions in the problem space using certain functions for the position and velocity of the particles. The movement of particles is affected by the best solution of the particle, and the best solution is generally obtained from other particles. The use of more than one hidden layer is usually accompanied by feature selection or feature extraction [41]. Summary and research criticism that calculates hidden layers and hidden neurons can be seen in table 3.

Table 3. Research determines the number of hidden neurons

No	Researcher	Function	Method	Critics
1	Karisma and Widyantoro (2016)[23]	Predictive	Rule of thumb	Difficult to apply to other cases
2	Koutsoukas et al. (2017)[6]	Predictive	Trial and error	Difficult to apply to other cases
3	Qolomany et al. (2017)[4]	Predictive	Particle Swarm Optimization	Need a long time. Architecture exists through a training process
4	Tej and Holban (2018)[42]	Classification	Clustering and Regression	Clustering is more related to variation and regression with correlation

3. CONCLUSION

Research that counts only the number of neurons is carried out assuming that the use of one hidden layer is able to approach almost all functions. Using more than one hidden layer will add to the burden of computer computing. But with the advancement of computer computing capabilities, the use of more than one hidden layer is worth considering. In addition, the majority of researches in this group do not consider the characteristics of the input feature, but only pay attention to the number of features or amount of data. By pre-processing processing input features, for example by looking at variations and correlations between or intra input features, it is hoped that better results will be obtained. Research that compares the use of one or two hidden layers, in addition to not paying attention to the characteristics of the data also does not compare the possibility of using more than two hidden layers. Research only focuses on performance comparisons between the use of one and two hidden layers. The research focuses on determining the number of hidden layers and the number of hidden neurons at once, the majority of research is conducted by trial and error or using rule of thumb for experienced researchers. The trial and error method can consume a lot of research resources with results that are not necessarily as expected. While using references from the results of experienced researchers also cannot be used as a handle, because basically the rule of thumb only applies to certain datasets or certain functions. So this group does not pay attention to the characteristics of the input feature to determine the network topology. The input feature character that can be considered is by calculating the correlation or variation between or input features. In future research the purpose of determining neural network topology is to minimize errors, should pay attention to the correlation and variance of features to achieve better performance. Besides minimizing errors, time complexity must also be considered due to the chosen neural network topology.

4. REFERENCES

[1] Guliyev, N.J. and Ismailov, V.E. 2018. On the approximation by single hidden layer feedforward neural networks with fixed weights. *Neural Networks*, vol. 98, pp. 296–304.

[2] Thomas, A. J. , Petridis,M., Walters, S.D., Gheytassi, S. M., and Morgan, R. E. 2017. Two Hidden Layers are Usually Better than One. In *Engineering Applications of Neural Networks(EANN)*, Boracchi G., Iliadis L., Jayne C., and Likas A. (eds). Communications in Computer and Information Science, vol 744. Springer, Cham.

[3] Thomas, A. J. , Walters, S. D. , Gheytassi, S. M., Morgan, R. E, and Petridis,M. 2016. On the Optimal Node Ratio between Hidden Layers: A Probabilistic Study. *International Journal of Machine Learning and Computing*, vol. 6, no. 5, pp. 241–247.

[4] Qolomany, B., Maabreh, M. , Al-Fuqaha, A., Gupta,A., and Benhaddou, D. 2017. Parameters optimization of deep learning models using particle swarm optimization. In *13th Int. Wirel. Commun. Mob. Comput. Conf.*, pp. 1285–1290.

[5] Nitta, T. 2017. Resolution of Singularities Introduced by Hierarchical Structure in Deep Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2282–2293.

[6] Koutsoukas, A., Monaghan,K.J., Li, X., and Huan, J. 2017. Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *J. Cheminform.*, vol. 9, no. 1, pp. 1–13.

[7] Sheela, K.G. and Deepa,S.N. 2013. Review on methods to fix number of hidden neurons in neural networks. *Math. Probl. Eng.*, vol. 2013, no. 4, pp. 449–471.

[8] Madhiarasan, M., and Deepa, S. N. 2016. A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting. *Applied Intelligence*, 44(4), 878–893. https://doi.org/10.1007/s10489-015-0737-z.

[9] Vujičić, T. and Matijevi, T. 2016. Comparative Analysis of Methods for Determining Number of Hidden Neurons in Artificial Neural Network. In *Cent. Eur. Conf. Inf. Intell. Syst.*, p. 219.

[10] Thomas, L., Manoj Kumar, M. V., and B. Annappa, B. 2016. Discovery of optimal neurons and hidden layers in

feed-forward Neural Network. In *IEEE Int. Conf. Emerg. Technol. Innov. Bus. Pract. Transform. Soc. EmergiTech 2016*.

[11] Gao, T., Jojic, V., and Hill, C. 2017. Sample Importance in Deep Neural Network. In *ICLR 2017*.

[12] Karsoliya, S. 2012. Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. *Int. J. Eng. Trends Technol*., vol. 3, no. 6, pp. 714–717.

[13] Madhiarasan, M. and Deepa, S.N. 2017. Comparative analysis on hidden neurons estimation in multi layer perceptron neural networks for wind speed forecasting. *Artif. Intell. Rev.*, vol. 48, no. 4, pp. 449–471..

[14] Lorenzo, P.R., Nalepa, J., Kawulok, M., Ramos,L.S., and Pastor, J.R. 2017. Particle swarm optimization for hyper-parameter selection in deep neural networks. In *Proc. 2017 Genet. Evol. Comput. Conf. - GECCO '17*, vol. 8, pp. 481–488.

[15] Kawaguchi, K., Kaelbling, L.P., and Bengio, Y. 2017. Generalization in Deep Learning. https://arxiv.org/abs/1710.05468

[16] Lee, S., Zokhirova, M., Nguyen, T.T. and Lee, J. 2017. Effect of Hyper-Parameters on Deep Learning Networks in Structural Engineering. In *Proceedings of the International Conference on Advances in Computational Mechanics 2017* pp 537-544

[17] Lee, S., Ha ,J., Zokhirova, M., Moon, H., and Lee, J. 2018. Background Information of Deep Learning for Structural Engineering. *Arch. Comput. Methods Eng*., vol. 25, no. 1, pp. 121–129.

[18] Lorenzo, P.R. and Ramos, L.S. 2017. Hyper-Parameter Selection in Deep Neural Networks Using Parallel Particle Swarm Optimization. In *GECCO '17 Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Pages 1864-1871.

[19] Bergstra, J. and Bengio, Y. 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, vol. 13, pp. 281–305.

[20] Larochelle, H., Courville, A., and Bergstra, J. 2007. An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation. In *ICML '07 Proceedings of the 24th international conference on Machine learning*, Pages 473-480.

[21] Tang, X., Ding, Y., and Hao, K. 2018. A Novel Method Based on Line-Segment Visualizations for Hyper-Parameter Optimization in Deep Networks. *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 3.

[22] Diaz, G., Fokoue, A., Nannicini, G., and Samulowitz, H. 2017. An effective algorithm for hyperparameter optimization of neural networks. *IBM Journal of Research and Development* , vol. 61, no. 4, pp. 1–11.

[23] Karisma, H. and Widyantoro, D.H. 2016. Comparison study of neural network and deep neural network on repricing GAP prediction in Indonesian conventional public bank. In *Proc. 2016 6th Int. Conf. Syst. Eng. Technol. ICSET 2016*, pp. 116–122.

[24] Grabusts, P. and Zorins, A. 2015. The Influence of Hidden Neurons Factor on Neural Network Training Quality Assurance. In *Proceedings of the 10th International Scientific and Practical Conference*, vol. 3, pp. 76–81.

[25] Sartori, M.A. and Antsaklis, P.J. 1991. A simple method to derive bounds on the size and to train multilayer neural networks. *IEEE Transactions on Neural Networks*,vol.2,no. 4, pp.467– 471.

[26] Arai, M. 1993. Bounds on the Number of Hidden Units in Binary-Valued Three-Layer Neural Networks. *Neural Networks*, vol 6, 855–860.

[27] Li, J-Y, Chow T.W.S, and Yu, Y-L. 1995. The estimation theory and optimization algorithm for the number of hidden units in the higher-order feed forward neural network. In *Proceeding IEEE international conference on neural networks*, vol 3, pp 1229–1233.

[28] Hagiwara, M. 1994. A simple and effective method for removal of hidden units and weights. *Neurocomputing*, vol 6, 207–218.

[29] Tamura, S., and Tateishi, M. 1997. Capabilities of a Four-Layered Feedforward Neural Network : Four Layers Versus Three. *IEEE Transactions on Neural Networks*, vol 8(2), 251–255..

[30] Fujita, O. 1998 . Statistical estimation of the number of hidden units for feedforward neural networks. *Neural Networks : The Official Journal of the International Neural Network Society*, 11, 851–859. https://doi.org/10.1016/S0893-6080(98)00043-4.

[31] Keeni, K. 1999. Estimation of Initial Weights and Hidden Units for Fast Learning of Multi-layer Neural Networks for Pattern Classification. In *IJCNN'99. International Joint Conference on Neural Networks*.

[32] Onoda, T. 1995. Neural network information criterion for the optimal number of hidden units. In *Proceeding IEEE international conference on neural networks*, vol 1, pp 275–280.

[33] Islam, M., and Murase, K. 2001. A new algorithm to design compact two-hidden-layer artificial neural networks. *Neural Network*, vol 14(9).

[34] Zhang, Z., Ma, X., and Yang, Y. 2003. Bounds on the number of hidden neurons in three-layer binary neural networks. *Neural Networks*, vol 16(7), 995–1002. https://doi.org/10.1016/S0893-6080(03)00006-6.

[35] Choi, B., Lee, J., and Kim, D. 2008. Solving local minima problem with large number of hidden nodes on two-layered feed-forward artificial neural networks. *Neurocomputing*, vol 71, 3640–3643.

[36] Shibata, K. and Ikeda. 2009 . Effect of number of hidden neurons on learning in large-scale layered neural networks. In *2009 ICCAS-SICE*.

[37] Doukim, C. A., Dargham, J. A., and Chekima, A. 2010. Finding the Number of Hidden Neurons for an MLP Neural Network Using Coarse to Fine Search Technique. In *10th International Conference on Information Science, Signal Processing and their Applications*.

[38] Panchal G, Ganatra A, Kosta Y.P., Panchal, G., Ganatra, A., Kosta, Y. P., and Panchal, D. 2011. Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers. *International Journal of Computer Theory and Engineering*, Vol 3(2), 332–337.

[39] Hunter, D., Yu, H., Pukish, M. S., Kolbusz, J., and Wilamowski, B. M. 2012 . Selection of Proper Neural Network Sizes and Architectures — A Comparative Study. *IEEE Transactions on Industrial Informatics*, 8(2), 228–240 https://doi.org/10.1109/TII.2012.2187914.

[40] Nakama, T. 2011. Comparisons of Single- and Multiple-Hidden-Layer Neural Networks. In *International Symposium on Neural Networks 2011* , 270–279.

[41] Abroudi, A., Shokouhifar, M., and Farokhi, F. 2013. Improving the Performance of Artificial Neural Networks via Instance Selection and Feature Dimensionality Reduction. *International Journal of Machine Learning and Computing* vol. 3, no. 2, pp. 176-180.

[42] Tej, M. L., and Holban, S. 2018 . Determining Optimal Neural Network Architecture Using Regression Methods. In *2018 International Conference on Development and Application Systems (DAS)*, 180–189.