MethodsX 8 (2021) 101477

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

Machine learning performance validation and training using a 'perfect' expert system

Jeremy Straub

Department of Computer Science, North Dakota State University

A B S T R A C T

A method is proposed for generating application domain agnostic data for training and evaluating machine learning systems. The proposed method randomly generates an expert system network based upon user specified parameters. This expert system serves as a generic model of an unspecified phenomena. The expert system is run to determine the ideal output from a set of random inputs. These inputs and ideal output are used for training and testing a machine learning system. This allows a machine learning technology to be developed and tested without requiring compatible test data to be collected or before data collection as a proof-of-concept validation of system operations. It also allows systems to be tested without data error noise or with known levels of noise and with other perturbations, to facilitate analysis. It may also facilitate testing system security, adversarial attacks and conducting other types of research into machine learning systems.

- Provides an application domain agnostic way to test machine learning technologies and facilitates the generalization of results.
- Allows technologies to be tested with data with different characteristics without having to locate datasets that have these characteristics.
- Utilizes randomly generated network to represent non-specific phenomena which can be used for training and testing machine learning techniques.

A R T I C L E I N F O

Method name: 'Perfect' Model Training and Validation of Machine Learning Technologies
Keywords: Machine learning, Learning model, Knowledge engineering, System performance evaluation
Article history: Received 6 July 2021; Accepted 2 August 2021; Available online 2 August 2021

E-mail address: jeremy.straub@ndsu.edu

https://doi.org/10.1016/j.mex.2021.101477

Specifications table

Subject Area:	Computer Science
More specific subject area:	*Expert Systems and Neural Networks*
Method name:	*'Perfect' Model Training and Validation of Machine Learning Technologies*
Name and reference of original method:	*J. Straub, Expert system gradient descent style training: Development of a defensible artificial intelligence technique, Knowledge-Based Systems (2021), doi: https://doi.org/10.1016/j.knosys.2021.107275*
Resource availability:	*N/A*

Introduction

Artificial intelligence (AI) and machine learning (ML) technologies are widely used and provide significant benefits [1] to modern society. AI and ML can command robots [2], play games [3], find software bugs [4], help those with handicaps [5], facilitate student learning [6] and even detect hackers [7].

AI systems take a number of forms ranging from largely human-directed systems, where humans define operating rules and protocols for the system to follow, to systems that adaptively learn on their own or through a training process. ML systems fall into this later group. A number of different ML techniques have been developed. Some learn by being supplied input and desired output data [8]. Other systems provide 'rewards' to encourage system behaviors [9] or are developed to identify association between different types of data [10]. Some ML systems are designed to be human-understandable, while others are more opaque requiring human users to rely upon the functionality of the system, sight unseen [11,12].

As ML systems are developed, a variety of techniques are used to train them. When developing systems for particular application areas, domain specific data is used to train them and test their efficacy for the particular application [13]. However, Roh, Heo and Whang [13] note that a lack of data often hinders development processes. This issue would inherently be more pronounced for demonstrating the efficacy of techniques, where multiple data sets might be required to demonstrate system efficacy and performance.

In [14], the use of gradient descent training was proposed for use with expert systems to create a hybrid system with known-meaning nodes (facts) and associations (rules). A 'perfect' expert system network was used for training and evaluating the performance of the learning system. Inputs to this system would serve as inputs to the system being trained and tested and the output of this system would be used as the desired output for training or testing. This application agnostic and highly configurable training and evaluation method is presented herein.

Method details

The 'perfect' model training approach can be used with both the expert system machine learning approach it was presented with in [14] (as well as other machine learning approaches which might be developed in the future for use with expert system networks) and neural networks. A key area of future work will be assessing the approach's efficacy for other types of artificial intelligence systems.

Figures 1 shows how the technique can be used with neural networks. Figure 2 shows how it has been used with expert system machine learning.

Use with Neural Networks

With a neural network, input values from the expert system are also used as input values to the neural network. A designated node (or nodes) from the expert system is (are) compared to the output (or outputs) of the neural network. For training, the expert system output (or outputs) is (are) used as the target output (or outputs) by the training process. For evaluation, the neural network output (or outputs) is (are) compared to the expert system output (or outputs) for purposes of system efficacy assessment. Because a neural network is a densely connected network system (and nodes are

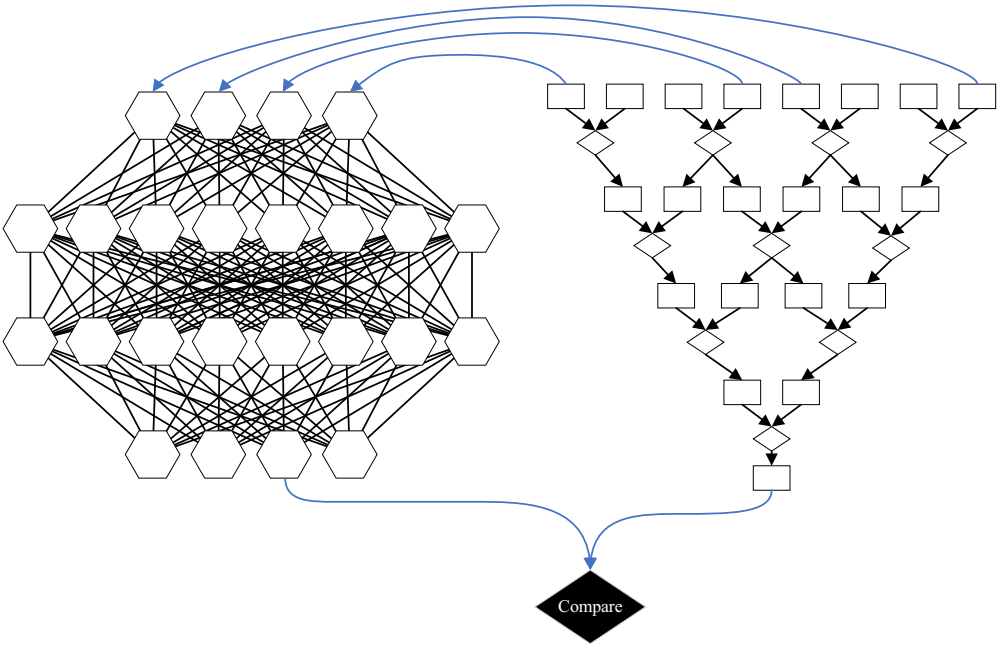

Fig. 1. Use of expert system network to train neural network.

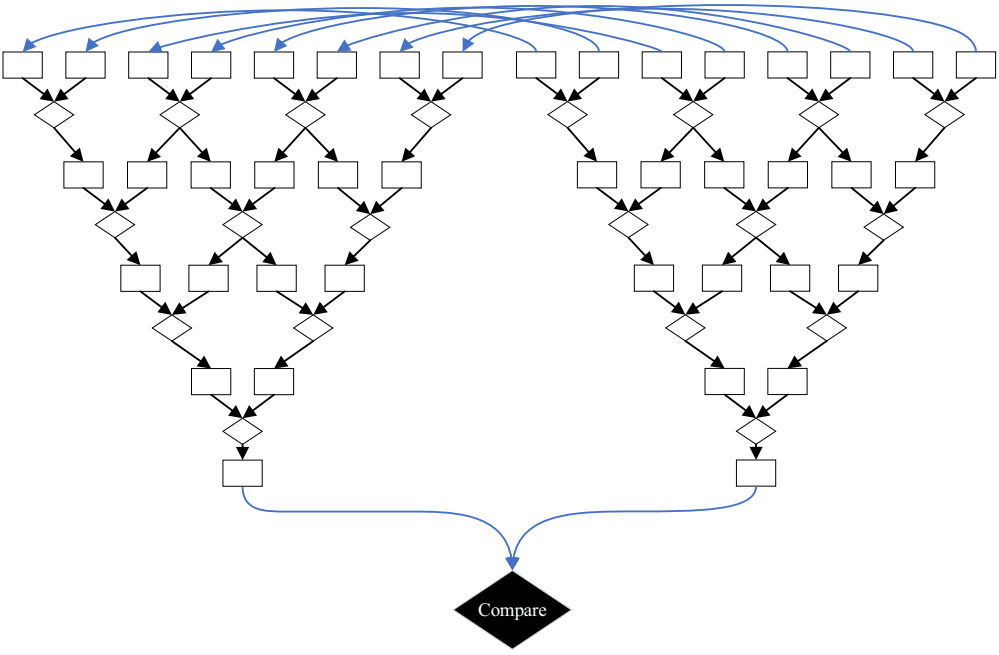

Fig. 2. Use of expert system network to train gradient descent expert system.

effectively homogeneous), the inputs and outputs of the expert system which are selected for training can be paired with any input or output, respectively, of the neural network as long is this is done consistently for training and data presentation. It is not necessary to pair all inputs or outputs of the expert system with a neural network input or output (just as not all fields of a data set would necessarily need to be used for an application domain training process). The number and selection of inputs and outputs could, in fact, become an experimental variable for some types of system testing.

Data from the 'perfect' model expert system can be used as inputs and outputs to the neural network in a manner similar to collected data for a particular application area. Once the system has been trained and 'perfect' model input data has been presented to collect post-training system results, these results can be analyzed in the same way as result data from other neural network experiments is analyzed.

Use with Expert System Machine Learning

With the expert system machine learning system, a very similar technique is used; however, for the system presented in [14], only known-meaning associations (rules) are created between nodes (facts). While the 'perfect' model training approach inherently abstracts the known-meaning concept somewhat (as none of the nodes necessarily have value in the real world – their meaning is only within their construct), the fundamental concept is that a node (fact) represents a specific identified piece of data. Given this, node inputs should only be connected to matching nodes in the other network. Connecting non-matching nodes would be inherently illogical: it would be the equivalent of taking the answer supplied for one question and providing it as the answer for an alternate arbitrary question. However, like with the use of this method for neural networks, it is not necessary to supply a value for every node of the network under training. Components of the network can, prospectively, be trained without training the entire network. Also, while the associations (rules) would ideally be the same between networks (for performance), this is not required. In fact, [14] assessed system performance under a variety of perturbations of the network and more analysis is planned, as future work, in this area. Other manipulations of the 'perfect' network data are also possible to simulate noise, input error and other phenomena that may be of interest in testing a technique.

Once the 'perfect' network has been used for input and output data, like any other data source would be used, any conventional assessment techniques for neural networks and other machine learning techniques can be used, as experimentally relevant, to assess and characterize system performance.

Extended Uses

Beyond the simple use of training and testing a machine learning system with perfect data, this method can be used in a number of ways that may be beneficial to some experiments. Because the initial state of the data is known to be error-free, within the artificial construct of its creation, different perturbations can be added to the data to study their impact without fear of confoundment with other variables which may be present in real world data. For example, a study could focus on the resilience of a machine learning system to input or collection error and artificially introduce known amounts of error to facilitate this assessment. Alternately, studies could introduce different types of bias in input or output data (or both) by applying relevant modifications to the 'perfect' data to study the impact of this.

In addition to manipulating the data from the 'perfect' network, the network itself (rules) and node (fact) values can also be modified to simulate different types of phenomena. For example, a study could compare different levels of complexity of the 'perfect' expert system network (representing the actual real-world complexity of a decision-making process) with different complexity levels of an abstracting machine learning expert system or different numbers of layers of a neural network. Data patterns could be, similarly, studied by controlling the creation of the data that is used in the 'perfect' expert system.

Generalization Benefits

A key benefit provided by this approach is as generalizability. Because the data can be generated randomly (or randomly, based on known characteristics or with known perturbations), the results of testing a system can be far more generalizable than would be possible with a given set of domain-specific test data. With domain-specific test data, questions of inherent biases and confoundment within the data may be unresolvable or may be difficult to resolve. Using random generation techniques, independence, dependence or interrelationships between nodes and rules are inherently known, as they were deliberately created. This allows focus to be placed on the technology being developed or evaluated as opposed to necessarily being split between the technology and the application domain. In fact, some studies may benefit from first assessing a technology generally, then assessing it with generated domain-specific characteristic data (i.e., data that has the characteristics that the domain data is believed to have), before supplying it with actual data. This would allow the logical / domain-theoretical characteristics of the system to be assessed (and the system to be, perhaps, modified as needed) before introducing the complexities of real-world data. Additionally, if the system performed as expected with the simulated data and differently with the actual data, this may indicate an error in the model or that additional aspects of a phenomena being studied are not modeled. This could help improve researchers' understanding of the phenomena itself. The comparison of the simulated data system and the real-world data system could even, in some cases, be used as part of the validation of the accuracy of the model implemented itself.

Summary

The 'perfect' model training and validation method can be used with neural networks and machine learning trained expert systems to assess their performance. It can be used to demonstrate and assess base functionality as well as to assess performance under different scenarios which can be simulated through data or 'perfect' model rule-fact network perturbation. Additionally, it can potentially be used to assess the differences between an ideal model of a phenomena and data collected from the real-world, to assess model accuracy. The method may have application to additional styles of machine learning, beyond neural networks and machine learning trained expert systems. This remains an area of potential future work and analysis.

Declaration of Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Figures 1 and 2 incorporate network depictions, as component parts of each figure, that were initially made and presented in [14].

References

[1] M. Nadimpalli, Artificial Intelligence Risks and Benefits, Int. J. Innov. Res. Sci. Eng. Technol. (2007) 3297 https://www.researchgate.net/publication/319321806. (accessed June 25, 2020).
[2] S.C. Jacobsen, M. Olivier, F.M. Smith, D.F. Knutti, R.T. Johnson, G.E. Colvin, W.B. Scroggin, Research Robots for Applications in Artificial Intelligence, Teleoperation and Entertainment, Int. J. Rob. Res. 23 (2004) 319–330, doi:10.1177/0278364904042198.
[3] S. He, Y. Wang, F. Xie, J. Meng, H. Chen, S. Luo, Z. Liu, Q. Zhu, Game player strategy pattern recognition and how UCT algorithms apply pre-knowledge of player's strategy to improve opponent AI, in: 2008 Int. Conf. Comput. Intell. Model. Control Autom. CIMCA 2008, 2008, pp. 1177–1181, doi:10.1109/CIMCA.2008.82.
[4] A. Tosun, A. Bener, R. Kale, AI-Based Software Defect Predictors: Applications and Benefits in a Case Study, 2010. https://www.aaai.org/ocs/index.php/IAAI/IAAI10/paper/view/1561 (accessed June 25, 2020).
[5] H.A. Yanco, J. Gips, Preliminary investigation of a Semi-Autonomous Robotic Wheelchair Directed Through Electrodes, in: Proc. Rehabilition Eng. Soc. North Am. Annu. Conf., RESNA Press, Pittsburgh, PA, USA, 1997, pp. 414–416. www.cs.bc.edu/~gips/EagleEyes.

[6] K. Holstein, B.M. McLaren, V. Aleven, Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer Verlag, 2018, pp. 154–168, doi:10.1007/978-3-319-93843-1_12.
[7] Z.A. Baig, M. Baqer, A.I. Khan, A pattern recognition scheme for Distributed Denial of Service (DDoS) attacks in wireless sensor networks, in: Proc. - Int. Conf. Pattern Recognit, 2006, pp. 1050–1054, doi:10.1109/ICPR.2006.147.
[8] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: ACM Int. Conf. Proceeding Ser, New York, New York, USA, ACM Press, 2006, pp. 161–168, doi:10.1145/1143844.1143865.
[9] Y. Duan, X. Chen, R. Houthooft, J. Schulman, P. Abbeel, Benchmarking Deep Reinforcement Learning for Continuous Control, in: Proc. 33 Rd Int. Conf. Mach. Learn, 2016.
[10] G. Paliouras, C. Papatheodorou, V. Karkaletsis, C.. Spyropoulos, Discovering user communities on the Internet using unsupervised machine learning techniques, Interact. Comput. 14 (2002) 761–791, doi:10.1016/S0953-5438(02)00015-2.
[11] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.Z. Yang, XAI-Explainable artificial intelligence, Sci. Robot. 4 (2019), doi:10.1126/scirobotics.aay7120.
[12] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer, 2019, pp. 563–574, doi:10.1007/978-3-030-32236-6_51.
[13] Y. Roh, G. Heo, S.E. Whang, A Survey on Data Collection for Machine Learning: A Big Data-AI Integration Perspective, IEEE Trans. Knowl. Data Eng. 33 (2021) 1328–1347, doi:10.1109/TKDE.2019.2946162.
[14] J. Straub, Expert system gradient descent style training: Development of a defensible artificial intelligence technique, Knowledge-Based Syst. (2021) 107275, doi:10.1016/j.knosys.2021.107275.