A Exploration of Neural Network Development Methodologies

Ki-Dong Lee† · Peter Meso††

ABSTRACT

We examined current publications on artificial neural network development with a view to identifying the methodologies that are being used to develop these networks, how extensive these methodologies are, the categorization of these methodologies, if these methodologies demonstrate a common underlying and generic (standard) methodology for the development of artificial neural networks, and how closely these methodologies (and the underlying genetic methodology, if established) relate to the conventional systems development methodologies.

Key words : Artificial Neutral Network, System Development Methodology, Artificial Intelligence, Conventional Systems Development Methodologies

인공지능 네트워크의 Methodology 개발 상호비교

이기동† · Peter Meso††

요 약

본 연구는 인공지능 네트워크 시스템의 개발을 위해 존재하는 방법론들이 어떠한 것이 있는지를 연구하고자 한다. 인공지능 네트워크 개발에 대해 현재 발표된 것과 방법론을 명확히 하기위한 관점들을 측정하였으며 그것은 이러한 네트워크 개발에 이용되었다. 광범위한 이러한 방법론들을 어떻게 범주화하고, 만약 이런 방법론들이 인공지능 네트워크 개발에 대한 일반적, 근본적이고 포괄적인 방법론으로 증명할 수 있는지, 그리고 이런 방법론들이 기존 시스템 개발 방법론들과 어떻게 다른지 본 연구를 통해 시험했다.

주제어 : 인공지능 네트워크, 시스템 개발 방법론, 인공지능, 전통적 시스템 개발 방법론

---

† Professor, Graduate School of Business, Univ. of Incheon.
†† Professor, corresponding author. School of Science and Technology, Georgia Gwinnett College.(Corresponding Author)
Received : 2011-05-23, Amended : 2011-07-15, Accepted : 2011-07-25
* This work was supported by the University of Incheon Research Grant in 2010

1. INTRODUCTION

A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by the network structure, the connection strengths (or weights), and the processing performed at the computing elements (or nodes). Neural network architectures are inspired by the architecture of biological systems. The characteristics of biological neural networks that artificial neural network (ANN) models hope to provide include, fault tolerance to loss of small numbers of computational elements, insensitivity to small variations between computational elements, the need for primarily local connectivity and local learning rules, real time and parallelism[14].

Further, these methodologies require that the information system being developed be atrue model of the real life problem that it is modeling. Hence the Data elements (entities, relationships, and attributes) need to be deterministic and fully definable, and the functions or processes need to be similar to the functions of the real life problem [1][19].

By their nature, artificial neural networks are stochastic systems [2][9]. Therefore they have attributes that make conventional systems development methodologies ill suited for their development. Unlike conventional information processing systems, artificial neural networks thrive under noisy data. The relationships between the various processing units that make up the artificial neural network are not discrete and the flow of data through the system is not deterministic in nature. The sequence of the functions or processes in the network is not well established, and can change as data flows through the network. Therefore whereas conventional information systems are deterministic in orientation, artificial neural networks are largely stochastic. Further whereas conventional information systems lend themselves to a deterministic development approach, the development of a neural network may not be fully deterministic[14].

Because of these reasons it is logical to presume that greater good is derived form employing a systems development methodology to the development of an Artificial Neural Network Information Processing System. Therefore the key question becomes: What methodologies exist for the development of artificial neural network systems?

We examined current publications on artificial neural network development with a view to identifying the methodologies that are being used to develop these networks, how extensive these methodologies are (if they address all aspects of the systems development effort), the categorization of these methodologies (if they can be categorized as weak methodologies or strong methodologies), if these methodologies demonstrate a common underlying and generic (standard) methodology for the development of artificial neural networks, and how closely these methodologies (and the underlying genetic methodology, if established) relate to the conventional systems development methodologies.

2. LITTERATURE REVIEW

The origin of today's neural networks is from the seminal work of McCulloch-Pitts[5]. MuCulloch and Pitts describe a model of a neuron as a binary threshold device. As the neuron receives a weighted sum of inputs from connected units, it produces one of two outputs: one if the sum exceeds a threshold, zero otherwise. Rosenblatt[6] develops a single layer neuron device called the perceptron. Widrow and Hoff do research on perceptron-like systems called ADLINE (ADAptive

Linear Neuron). Their learning algorithm for the model is known as the Widrow-Hoff delta learning rule which is a key foundation of the widely-used backpropagation learning algorithm.

The resurgence of interest in this area was as a result of the invention of efficient training algorithms for multi-layer perceptrons suggested by Werbs, Parker, Rumelhart, and others, owing to developments in analog VLSI implementation techniques[8][10][11][20]. The multi-layer perceptron or back propagation employs a hidden layer between the input and output layers. The back propagation learning algorithm is a systematic method for training multi-layer feed forward neural network. It is the most popular learning algorithm in neural network applications. Other neural network models that have received widespread recognition are the Hopfield networks, and Kohonen's self organizing networks. Hopefield's recurrent network works as an associative memory. The associative memory may recall an example from a partial or distorted version. Hopfield nets are different from back propagation in that they are non-layered with complete connectivity between nodes. Kohonen's feature maps[5][18] are motivated by the self-organizing behavior of the human brain. The learning process for these types of networks is unsupervised which is useful for clustering problem.

Papers directly relating to the methodologies for developing neural networks are, not only few, but fairly recent. The emergence of these papers however, is an indication that neural networks have matured to the point where they are now being perceived as full-fledged information systems. Acknowledging the difficulty inherent in developing neural networks via a structured scientific approach, Murphy, Koehler and Folger[12] present a heuristic approach to fitting neural net models to business applications. Primathu, Ragavan and Shaw[13][22] present a methodology for measuring the learning difficulty inherent in neural network design. Ingwersen[17] proposes that Ranganathan's approach to knowledge organization may find useful application in neural network modeling. Papers providing a broader coverage of neural network development methodologies[15][16][9] are also published within the same time frame.

3. RESEARCH DESIGN

In order to identify the existing methodologies for developing artificial neural networks, we conducted a search for all resources that contained the key words "methodology" and "neural network". This search was conducted on the Ohiolinkdatabase, an online educational catalog of all library resources in Northeastern Ohio, using the search engine provided by that database. The search yielded a listing of about 84 sources. We physically reviewed the abstracts of each to identify those thatspecifically addressed the development of Neural Networks via some articulated methodology. The result was the three methodologies presented in this paper. We then reviewed a wide cross section of literature on neural networks, systems theory, Informationsystems and conventional systems development methodologies in a bid to identify whether the existing methodologies exhibited a generic underlying methodology for the development of any type of artificial neural network system. Our review supported the existence of such a methodology but noted that the methodology seems not to have been formally articulated or documented in any of the past literature. Therefore the generic methodology for the development of neural networks documented in this paper is the result of this extensive review. Finally, we performed a cross comparison of the three methodologies identified in artificial neural network development literature and the generic methodology that we derived. This cross comparison, articulated in <Table 5>, allowed us to draw inferences on the nature of the methodologies currently being employed to develop artificial neural networks. It also provided for a structured comparison of these methodologies to the conventional systems development life cycle methodology.

4. RESEARCH FINDINGS

We identified the following methodologies, as labeled by the names of their authors, in the artificial neural network publications:

- The Vico and Sandoval Methodology[3]
- The Glorfeld and Hardgrave Methodology [16], and
- The Marwah, Li and Mahajan Methodology[22].

Each of these methodologies is summarized in <Tables 1 to 3>. For each, we present the main phases, the activities conducted in each phase and the techniques used to effect each phase.

<Table 1> The Vico and Sandoval Methodology[9]

PHASE	ACTIVITIES	TECHNIQUES &TOOLS
Specification Of The Problem	- Input Set - Output Set - Input/Output Relations - Performance Required	
Random Generation Of Genetic Codes	- specify the number of input cells and that of output cells - specify the number of nodes - Generate the nodal functions	cellular bipartition gradual specialization
Development Of Each Code To Obtain The Neural Networks	- randomly develop a large number of initial networks by linking the input, nodes, and output cells in different ways.	
Gross Adjustment Of The Network In The Phase Of Synaptic Plasticity	- optimize the networks by eliminating the misused connections	synaptic plasticity
Weights Adjustment In A Non Supervised Way	- Adjust the weights allocated to each node until optimal output is achieved from each alternative network	The Hebian rule (non-supervised learning)
Fitness Evaluation Of Each Network With The Controlled Sample	- Using simulation, evaluate the alternative networks developed. Transform the data structure of each alternative network into the subroutines of a simulator - assess the level of fit of each. alternative network by comparing the output of the network with the desired output pattern as generated by the simulator.	Rochester Connectionist Simulator
Stop If We Got A Network With The Required Performance	- Identify all the stable alternative networks - If there are stable alternative networks select the best of these as the final network and terminate the development process. - If none is available proceed to the next phase	
Obtain The Next Codes From The Best Networks That We Have Got	- Generate new nodal functions or revise the nodal functions - Return to the third Phase	

<Table 2> The Glorfeld and Hardgrave Methodology[16]

PHASE	ACTIVITIES	TECHNIQUES &TOOLS
Model Selection	- Split the sample data into V-training sets and V-validation sets (Usually V=10) - Train and Test single cell ADALINE model - Train and test linear PAWR model - Train and test BP network model - add hidden layer elements until no further improvement - Select the simplest model with the highest cross-validated objective function.	V-fold cross validation
Variable Selection	- Develop network model using full set of input variables (n) - Drop variable with smallest weight which makes least contribution - Repeat for each (n-1) variable sets until n=1	Garson's Variable importance measure
Selecting the Final Model	- Select model with the fewest number of variables that has the highest V-fold cross-validated objective function	

<Table 3> The Marwah, Li and Mahajan Methodology[22]

PHASE	ACTIVITIES	TECHNIQUES &TOOLS
User Supplied data	- Determine data availability - If sufficient data is available proceed to next phase - Otherwise use Design of Experiment approach to "conduct carefully designed experiments over the entire input domain such that the general behavior of the process can be captured. The data points so generated are used" in the next phase	
Split data into training and testing set	- Divide the data into two sets: a training data set (three quarters of the data available) and a testing data set (one quarter of the data available).	
Transform data	- Using the training set 1. determine number of training points 2. pick extreme output points 3. determine points per input 4. sort data with respect to inputs	CU-ANN
Choose initial simple structure, default parameters	- select a simple network structure consisting of two neurons in a single hidden layer - Randomly initialize all the network weights to small values around zero	CU-ANN

train	1. Train this network using a number of randomized initial weights (usually three). 2. If learning is unacceptably slow terminate training and proceed to step 3 below. Otherwise proceed to step 7 below. 3. Increase its complexity by using an additional neuron 4. Measure the R-squared and mean squared error of the network. 5. Measure the rate of decrease of mean squared error of the improved network to determine the rate of learning	CU-ANN
train and test	1. Compute the R-squared and the mean squared error of the network after each epoch. Adjust the weights of the network and re-train it until R-squared reaches the desired threshold value (0.8). 2. If the R-squared value reaches a desired threshold begin simultaneous testing and training of the network 3. Simultaneously train and test the network, calculating the network test values - R-squared and mean squared error for each epoch, but do not re-adjust the weights. 4. Compare the calculated R-squared and mean squared error results to the desired results to determine if the network has been successfully trained.	CU-ANN
Network trained	- Implement the trained network	CU-ANN

An analysis of the three methodologies identified in the literature reveals that neural network development methodologies tend to be highly specialized (strong) methodologies. Further, none seem to address all the phases of a typical systems development project. An analysis of the classification of documented neural network development methodologies reveals that these methodologies focus on the latter phases of the development of a neural network. When perceived in light of the conventional systems development paradigm, methodologies for developing neural networks can be seen as focusing on the program development or low-end functions of the systems development process.

Both the Vico[4] and Sandoval and the Glorfeld[6][7] methodology seem to address very specific types of neural networks. Although the Glorfeld and Hardgrave, and the Marwah, Li and Mahajan Methodologies[12] are much broader in scope than the other two methodologies, they remains specific to neural networks in that they are not synchronous to the development of other types of information processing systems. However, these methodologies provided some significant insights regarding the need for a structured methodology for the development of neural networks. The next section presents the generic methodology that resulted from this detailed review.

<Table 4> The Generic Artificial Neural Network Development Methodology

PHASE	ACTIVITIES
Analysis of the Problem	a) articulated and formulated the problem, b) determine the scope of the problem c) identify the stakeholders or entities included or affected by the problem and its solution d) undertake an initial assessment of the resources immediately available to solve the problem e) assess the nature, quality and quantity of data to be used in solving the problem f) study and document the traditional or existing approach to solving the problem.
Classification of the Problem	a) The problem is evaluated to identify if it is structured or if it can be solved by proven mathematical, statistical, conventional information systems development, or operations research methods. b) If neither of these are appropriate then a neural network alternative is evaluated. c) Alternately, if the neural network approach promises to yield a better solution, neural networks may be suggested even where the problem can be solved via an existing method.
Selection of the Training Method	Based on the following factors: -The type of data available -The type of solution desired Select the most appropriate training method from the following: a) supervised learning, b) unsupervised learning,
Preparation and Representation of Data	a) Establish the important parameters of the problem. b) For each parameter collect or prepare an adequate amount of appropriate data. c) Perform data representation.
Selection of the Neural Network Architecture	a) select an appropriate generic network architecture b) Determine the style of computation to be used by the network. c) Determine the topology of the network d) Determines the type of software development language (or CASE tool) or the type of neural network application software to purchase.
Selection of Feasible Models	a) Identify all existing models that support the selected architecture b) Enumerate the technical advantages and disadvantages of each and perform a corresponding cost benefit analysis. c) Assess each model based on experience and published research literature d) Select all models that appear to contribute to the solution. This selection is conducted in light of the technical, operational, and economic feasibility of each model appropriate neural network application package. 3.The network is tested and optimized using the testing data. The optimal performance of this network is then documented 4.One additional layer is added to the initial network via programming or adequate definitions in the application package.
Selection of Optimal Parameters.	a) Determine the contribution of each parameter to the final solution, b) Drop the parameter that makes the least contribution to the output, c) Re-Optimize the network using testing data, d) Validate the network using the validation data e) Record the performance of the new network, f) Repeat steps a) to e) until there is only one parameter left in the network.

5. COMPARISON OF THE ARTIFICIAL NEURAL NETWORK DEVELOPMENT METHODOLOGIES TO CONVENTIONAL SYSTEMS DEVELOPMENT METHODOLOGIES

<Table 5> outlines how each of the four methodologies that we identified in our survey fit into this genetic methodology. An analysis of the classification of documented neural network development methodologies reveals that these methodologies focus on the latter phases of the development of a neural network. When perceived in light of the conventional systems development paradigm, methodologies for developing neural networks can be seen as focusing on the program development or low-end functions of the systems development process. This may explain why systems development methodologies for developing neural networks were not derived until fairly recently -- when the study of neural nets is beginning to mature.

A comparison of Neural Network Development methodologies to conventional systems development methodologies revealed that these methodologies were not synchronous with the traditional systems development life cycle methodologies. This result was expected since neural nets are non-linear in orientation while the SDLC methodologies are largely linear in orientation. All the Neural network development methodologies, however, employed aspects of Prototyping and Rapid Application Development. The iterative nature of these techniques lends them to the development of non-linear systems such as the neural networks.

We also identified that in the development of neural networks, the analysis and design activities are so intertwined that it is difficult to differentiate them. Further, there appears to be dismal logical design stage in the development of neural networks. Whereas the logical design of the system is usually fully developed before physical design begins in conventional information systems design, this is not the case in the development of neural network systems. The development of neural network systems, being largely experimental in nature, placed more emphasis on the physical design of a large number of feasible network models. It was thus apparent that, given the experimental nature of developing neural network systems, the design of the final system could not be predetermined. This brought out another key difference between the development of conventional systems and the development of neural network systems: The neural network system emerges as the development process proceeds and can only be accurately articulated and documented at the end of the development process

<Table 5> A Comparison of Documented Methodologies To The Generic Methodology

The Generic Methodolog y	The Vico and Sandoval Methodology	The Glorfeld Methodolo gy	The Glorfeld and Hardgrave Methodolog y
Analysis of the Problem	1.Specification of the problem		
Classificatio n ofthe Problem			
Selection of the Training Method			
Preparation and Representati on of Data			
Selection of the Neural Network Architectur e		1.develop initial network 2.Determi ne network weights.	1.Model Selection

Selection of Feasible Models	2.Random Generation of Genetic Code 3.Development of each code to obtain the neural networks	3.Variable selection	2.Variable selection
Selection of Optimal Parameters.	4.Gross adjustment of the network in the phase of synaptic plasticity. 5.Weight Adjustment in a non supervised way. 6.Fitness evaluation of each network with the controlled sample 7.Stop if we got a network with the required performance. 8. Obtain the next codes from the best networks that we have got	4.Select final model	3. Selecting the Final Model

6. CONCLUSION

Like all other information processing systems artificial neural networks benefit from the use of a methodology. With this realization, researchers and practitioners are beginning to develop methodologies specific to the development of artificial neural networks. The methodologies currently in existence tend to address the lower end of the systems development phases -- phases dealing with the detailed design and implementation of artificial neural networks -- and hardly any objectively address the initial phases of the development effort. However, a review of artificial neural network literature suggests that there are elaborate phases that can be used to guide a systems developer through the initial as well as the latter phases of developing an artificial neural network. These phases, when ordered together provide a structured, generic methodology by which any type of artificial neural network can be developed.

The generic methodology for developing artificial neural networks, though having analysis,design, and implementation components as found in the traditional systems development methodologies, differs from the traditional systems development life cycle methodologies is certain specific ways. First, the nature of artificial neural networks is such that their development requires a lot of experimentation and iterative activity. Second, the analysis, design and implementation components of the generic artificial neural network development methodology are so intertwined as to be difficult to uniquely identify. Third, while the traditional systems development methodologies focus on the real life problem that is to be solved, the artificial neural network development methodologies focus on the data that models the problem that is to be solved. This renders the traditional systems development life cycle methodologies unsuitable for the development of artificial neural networks. It is our view that as artificial neural networks become an integral part of corporate information systems, more elaborate and standardized development methodologies for use in developing these kinds of systems will emerge.

References

[1] Al-Ghanim, Amjed (1997) An Unsupervised Learning Algorithm for Identifying Process Behavior on Control Charts and a Comparison With Supervised Learning Approaches. Computers & Industrial Engineering, 32(3), 627-639.

[2] Anjum, M. Farooq, Tasadduq, Imran and Al-Sultan, Khaled (1997) Response Surface Methodology: A Neural Network Approach. European Journal of Operations research, 101, 65-73

[3] Bailey, Alan (1998) Uh-Oh. It's a Computer Systems Project…É.: A Project Management

Approach Designed to Sidestep The High IT Project Failure Rate. PM Network, August, 29-34

[4] Bellman, Kristie L. and Landauer, Christopher (1995) Designing Testable, Heterogeneous Software Environments. Journal of Systems Software, 29, 199-217

[5] Ben-David, Arie and Pao, Yoh-Han (1992) Self-Improving Expert Systems: An Architecture and Implementation. Information & Management, 22(6), 323-331

[6] Debackere, K., and Clarysse, B. (1997) Advanced Bibliometric Methods to Model the Relationship Between Entry Behavior and Networking Emerging Technological Communities. Journal of the American Society for Information Science, 49, 49-58

[7] Debackere, K., and Rappa, M. A. (1994) Institutional Variations in Problem Choice and Persistence among Scientists in an Emerging Field. Research Policy, 23, 425-441.

[8] Diamond J, Pedrycz W, and McLeod D (1994) Fuzzy JK Flip Flops as Computational Structures - Design and Implementation. IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing, (41:3), 215-226

[9] Chen, S. K., Mangiameli, P., West, D. (1995) The Comparative Ability of Self-Organizing Neural Networks to Define Cluster Structure. Omega, 23(3), 217-279

[10] Cortez, Edwin M, Park, Sang C., Kim, Seonghee (1995) The Hybrid Application of an Inductive Learning Method and a Neural Network for Intelligent Information Retrieval. Information Processing and Management, 31(6), 789-813

[11] Denuth, H. and Beale M. (1994) Matlab: Neural Network Toolbox, Computation, Visualization, Programming: User's guide. The Math Works, Inc.

[12] Fish, Kelly E., Barnes, James H., and Aiken, Milam W. (1995) Artificial Neural Networks: A New Methodology for Industrial Market Segmentation. Industrial Marketing Management, 24(5), 431-438

[13] Freeman, Paul and Rennolis, Keith (1994) Modeling Methodology: Basics to Neural Nets - A return to Ignorance? Market Research Society. Journal of the Market Research Society, 36(1), 69-77

[14] Gallant, Stephen I. (1993) Neural Network Learning and Expert Systems. MIT Press, Cambridge, Massachusetts

[15] Glorfeld, Louis W. (1996) A Methodology for Simplification and Interpretation of Backpropagation-Based Neural Network Models. Expert Systems With Applications, 10(1), 37-54

[16] Glorfeld, Louis W., and Bill C. Hardgrave (1996) An Improved Method for Developing Neural Networks: The Case of Evaluating Commercial Loan Creditworthiness. Computers in Operations Research, 23(10), 933-944

[17] Ingwersen, P., and Wormelli, I. (1992) Ranganathan In the Perspective of Advanced Information-Retrieval. Libri, 42(3), 1992.

[18] Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. Biol. Cybernet, 43, 59-69

[19] Laudon, Kenneth C., and Jane P. Laudon (1998) Management Information Systems: New Approaches to Organization & Technology (Fifth Edition). Prentice Hall, Upper Saddle River, New Jersey

[20] Lippmann, R. P. (1987) An introduction to computing with neural nets. IEEE ASSP

Magazine, April, 4-22.
[21] Mangiameli, Paul, Chen, Shaw K., West, David (1996) A Comparison of SOM Neural Network and Hierarchical Clustering Methods. European Journal of Operations Research, 93(2), 402-417
[22] Marwah, Manish, Li, Yuan and Mahajan, Roop L. (1996) Integrated Neural Network Modeling For Electronic Manufacturing. Journal of Electronic Manufacturing, 6(2), 79-91

Ki-Dong Lee

1985 서강대학교 경영학과 (경영학사)
1992 University of Maine (경영학석사)
2001 Kent State University(경영학 박사)
2005~2007 디지털정책연구, 편집위원장
2008~현재 디지털정책학회 경영분과위원
2002~현재 인천대학교 경영학부 교수
관심분야: MIS, 인공지능, Social Network e-Commerce, e-Government
E-Mail: kdlee@incheon.ac.kr

Peter Meso

BS: United States Int. Univ.
MA: United States Int. Univ.
Ph.D. Kent State University
관심분야: Information Systems Research, Journal of the Association for Information Systems (JAIS), Information Systems Journal, European Journal of Information Systems, Communications of the ACM, Journal of Systems and Software, Information Systems Management,
E-Mail: pmeso@ggc.edu