Expert Systems with Applications 37 (2010) 1071–1074

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A novel Bayesian learning method for information aggregation in modular neural networks

Pan Wang [a,b], Lida Xu [a,c,d,*], Shang-Ming Zhou [e], Zhun Fan [f], Youfeng Li [b], Shan Feng [g]

[a] Institute of Systems Science and Engineering, Wuhan University of Technology, Wuhan 430070, China
[b] School of Automation, Wuhan University of Technology, Wuhan 430070, China
[c] School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China
[d] Department of Information Technology and Decision Sciences, Old Dominion University, Norfolk, VA 23529, USA
[e] UKCRC DECIPHer (Development and Evaluation of Complex Interventions for Public Health Improvement) Centre, Health Information Research Unit (HIRU), School of Medicine, University of Wales Swansea, SA2 8PP, UK
[f] Department of Management Engineering, Technical University of Denmark, Lyngby, Denmark
[g] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Keywords:
Bayesian learning
Modular neural network
Information aggregation
Combination
Modularity

ABSTRACT

Modular neural network is a popular neural network model which has many successful applications. In this paper, a sequential Bayesian learning (SBL) is proposed for modular neural networks aiming at efficiently aggregating the outputs of members of the ensemble. The experimental results on eight benchmark problems have demonstrated that the proposed method can perform information aggregation efficiently in data modeling.

1. Introduction

A modular neural network (MNN) is a neural network model that is characterized by a group of independent sub-neural networks moderated by some intermediaries in the entire architecture to perform complex functions (Jordan & Jacobs, 1991a, 1991b). Each independent sub-network serves as a module and operates on separate inputs to accomplish some subtasks of the task that the network is to perform. An additional layer acting as an intermediary takes the outputs of each module and processes them to produce the output of the network as a whole. In contrast to the "global" neural network models such as back-propagation neural networks (Haykin, 1999) and recurrent neural networks (Zhou & Xu, 1999, 2001), an MNN offers some significant advantages (Feng & Wang, 2007; Wang, Feng, & Fan, 2008). First it has strong biological backgrounds. Cognitive, neuropsychological and neurobiological sciences have indicated that biological brain function exhibits the property of modularity. Some neuropsychological experiments show that a circumscribed brain lesion could cause a specific disorder of language while other brain cognitive functions remain intact (Eccles, 1984; Edelman, 1979). The brain also performs the function of dividing the complex task of visual perception into many subtasks (Happel & Murre, 1994). Interestingly, the neuropsychology indicates that the thalamus of the brain is divided into different layers that separately process color and contrast (Hubel & Livingstone, 1990). Hence, according to the cognitive, neuropsychological and neurobiological sciences, the regions of animal and human brains are organized into specially and functionally segregated modules, while the MNN offers a computational intelligence technique for emulating these functions of segmentation and modularization found in the brain. The second advantage of MNN is that as a computing scheme, it provides the capability of dividing a large, unwieldy neural network into smaller, more manageable components. This is very useful in neural network applications since many real-world problems appear intractable for practical purposes by a single neural network as its size increases. If a complex task is further separated into distinct parts, the possible connections each node can make in building up a network are limited, so the sub-tasks will perform more efficiently than trying to tackle the whole task at once by a single network. As a matter of fact, even though a large neural network modelled by a large number of parameters can be constructed to tackle a complex problem, the network training process could suffer from interference when new data turns up, as these new data can dramatically alter existing connections or just serve to confuse. On the hand, if a task is divided into subtasks to be solved by independent sub-neural networks, each sub-neural network can be tailored for its task with its unique training data. In this way, more effective computing performance can be achieved. Currently, MNN has many applications in the domains such as pattern recognition and classification (Drake & Packianalher, 1998; Perez & Galdames, 1998; Zhou, Wu, & Tang,

* Corresponding author. Address: Department of Information Technology and Decision Sciences, Old Dominion University, Norfolk, VA 23529, USA. Tel.: + 1 757 6836138.
E-mail address: lidxu@hotmail.com (L. Xu).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.06.104

2002), image processing (Van Hulle & Tollenacre, 1993; Wang, Der, & Nasrabadi, 1998), system identification (Srinivas, Jeffrey, Huang, Phillips, & Wechsler, 2000), language/speech processing (Chen & Chi, 1999; Sibte & Abidi, 1996), control (Deepak, 1999), prediction/modelling (Pan & Sensen, 2005; Wang, Fan, & Li, 2006), target detection/recognition (Wang et al., 1998), fault diagnosis (Kim & Park, 1993), etc.

In MNN applications, after the ensemble of neural networks are trained, an important issue is how to effectively aggregate the individual neural network outputs. In fact, knowledge and information aggregation has become a subject of intensive research due to its practical and academic significance in many domains such as decision making (Carvalho & Costa, 2007; Li, 1999a, 1999b; Li & Li, 1999; Xu, 1988; Xu, Xu, Liu, & Jones, 2008; Zhao & Li, 2009; Zhou, Chiclana, John, & Garibaldi, 2008), information fusion (Tahani & Keller, 1990), knowledge engineering (Qi, Liu, & Bell, 2007; Xu, Wang, Luo, & Shi, 2006) and system modelling (Zhou & Gan, 2007). In MNN, the objective of aggregation is to combine the component network outputs in an appropriate way so that the final result can take all the individual contributions into account. The primary concerns in choosing an ensemble aggregation method for a MNN are bias and variance in the functional mapping estimates. Currently, many methods have been investigated for MNN to tackle this important issue (Cho & Kim, 1995; Hansen & Salamon, 1990; Hashem, 1997; Perrone & Cooper, 1994; Xu, Krzyzak, & Suen, 1992). Among them the simple averaging (Perrone & Cooper, 1994) and weighted averaging schemes (Hashem, 1997) for aggregating the outputs of the members of the ensemble are commonly used. However, few efforts have been made to address this issue from the perspective of Bayesian reasoning, although some researchers have proposed Bayesian schemes for designing "global" neural networks (Buntine & Weigend, 1991; Lampinen & Vehtari, 2001). The objective of this paper is to create a novel Bayesian approach to combining the outputs of the ensemble members in a MNN.

The paper is organised as follows. Section 2 presents the proposed sequential Bayesian learning algorithm for MNN, experimental results are described in Section 3, and Section 4 provides a conclusion.

2. Sequential Bayesian learning method

Formally, a neural network is considered to be modular (Jordan & Jacobs, 1991a, 1991b) if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other. The outputs of the modules are mediated by an integrating unit that is not permitted to feed information back to the modules. In particular, the integrating unit determines (1) how the modules are combined to form the final output of the system, and (2) which modules should learn which training patterns. The corresponding network architecture is illustrated in Fig. 1, in which $X \in D \subseteq R^n$ is the input vector; y is the output, $\{Net_i\}_{i=1}^K$ represent the individual sub-nets, and $\{w_i\}_{i=1}^K$ are the corresponding combination weights for aggregating the outputs of individual sub-nets to obtain the overall outputs.

$$y = \frac{\sum_{i=1}^K w_i y_i}{\sum_{i=1}^K w_i} \tag{1}$$

where $w_i \geqslant 0$. These weights could be viewed as the measure of "goodness" of the sub network's behavior in the entire system. A larger combination weight indicates the associated ensemble member playing a more important role in the decision making process.

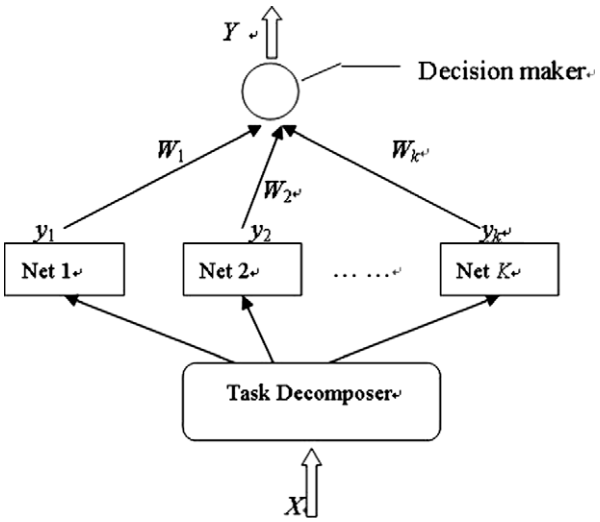

Fig. 1. Modular neural network architecture.

In the following, based on sequential Bayesian decision analysis a Bayesian learning approach to MNN is proposed to combine the output(s) of the members of the ensemble of a modular neural network.

First, if the unknown vector has prior density, the posterior density $p(\theta|x)$ determined by a certain observed vector x is:

$$p(\theta|x) = \frac{\pi(\theta) * p(x|\theta)}{p(x)} \tag{2}$$

If θ is continuous, the marginal density (or predicted function) of the observed vector x is calculated as

$$p(x) = \int_{\vartheta \in \Theta} p(x|\theta)\pi(\theta)d\theta \tag{3}$$

In case of discrete θ,

$$p(x) = \sum_{\vartheta \in \Theta} p(x|\theta)\pi(\theta) \tag{4}$$

where $p(x|\theta)$ is the likelihood function (LF). After the ensemble of sub-networks are trained, the issue of finding effective ways to combine the individual network outputs needs to be resolved.

In this paper, we propose to consider the combination of members of the ensemble as a statistical decision problem, in which each individual network is a decision-maker, the corresponding combination weight acts as the corresponding reliability measure. One of the advantages of Bayesian decision analysis is that it can model uncertain information via Bayesian reasoning process (Lampinen & Vehtari, 2001), which can help the analyst gain more insights into the system to be modeled. The proposed method consists of two steps: the first step is to train the networks of the ensemble; the second step is to learn the inter-connecting combination weights for the individual networks in a sequential fashion. At each stage, the available data samples plus the newly acquired data set are split into training data sets, test data sets, and validation data set. The training data sets are used to train the members of the ensemble, and the validation samples are used to identify the optimal combination weights in the second step. Then the generalization errors of each component network and the likelihood function (LF) value of each component network are calculated on the test samples. Finally the combination weights are adjusted according to Bayesian reasoning, in which the current weights are posterior and the previous weights are prior. Initial prior weights are set to be equal if we know nothing about the initial

weights. This sequential Bayesian learning scheme is summarized as follows.

Given the training set, validation set, the learning rate, the error limit, and the maximal number of iterations, we have the following steps for MNN learning:

Step 1. Train all the component networks until the error limit or the maximal number of iteration is achieved;
Step 2. Split the validation set into S parts, each one denoted as $\{ES_i\}_{i=1}^{S}$;
Step 3. For $i = 1$ to S:
(Step 3a). Compute the LF value $\omega_j^i (j = 1, 2, \ldots, K)$ as

$$\omega_j = \frac{1/sse_j}{\sum_{k=1}^{K} 1/sse_k} \tag{5}$$

in $\{ES_1 \cup ES_2 \cdots \cup ES_i\}$, where the sse_j represents the training error of the jth sub-network.
(Step 3b). Update the combination weights by using Bayesian reasoning:

$$w_j^i = \begin{cases} w_j^i = w_j^{i-1} & \text{if } \sum_{j=1}^{K} w_j^{i-1}\omega_j^i = 0 \\ \frac{w_j^{i-1}\omega_j^i}{\sum_{j=1}^{K} w_j^{i-1}\omega_j^i} & \text{otherwise} \end{cases}$$

From the (Step 3a), it can be seen that the validation data sets are constructed in a sequential way so that each validation set possesses certain property of inheritance. In the process of learning the combination weights, the (intra-network) connection weights of each component network do not change. Hence, in this sequential Bayesian learning algorithm, the global model performance is gradually improved.

Table 3
Architecture of component networks.

Problem	The architecture of each component network
MH2	1-10-1, 1-10-1, 1-12-1, 1-12-1
MH3	2-8-8-1, 2-8-8-1, 2-8-10-1, 2-8-10-1, 2-10-10-1
F1	5-18-1, 5-18-1, 5-20-1, 5-20-1, 5-22-1, 5-22-1, 5-24-1, 5-24-1
F2	4-25-1, 4-25-1, 4-30-1, 4-30-1, 4-35-1, 4-35-1
GB	2-12-1, 2-12-1, 2-14-1, 2-6-8-1, 2-8-8-1, 2-8-8-1
MI	5-10-1, 5-10-1, 5-12-1, 5-12-1, 5-14-1, 5-14-1, 5-16-1, 5-16-1
PL	2-8-1, 2-8-1, 2-10-1, 2-5-5-1, 2-5-5-1
PO	1-10-1, 1-10-1, 1-12-1, 1-12-1

3. Experimental results

In this section, eight benchmark regression examples (Zhou et al., 2002) are used to evaluate the proposed Bayesian learning algorithm for constructing MNNs, and compare with the learning scheme used in (Perrone & Cooper, 1994).

The eight benchmark regression examples are described in Table 1. For each problem, the proposed algorithm is run five times to reduce the potential randomness. The model performance is evaluated by averaging the five run results. First, we need to generate a data set and get the training set, validation set and test set. The training set is used to learn the connection weights of each component network, evaluation set to learn the combination weights, and test set to check the generalization performance of the whole system. The size of each data set and variables' distribution interval are shown in Table 2, where $x - y - z$ represents the size of these sets. Then the architecture of each individual network is constructed. Table 3 shows the architecture of each modular neural network, where $x - y - z$ (or $x - y_1 - y_2 - z$) means the number of units in input, hidden and output layer are x, y (or y_1 and y_2) and z. The activation function of the neurons in the hidden layer(s) is sigmoid function and the one in output layer is linear.

The next step is to train all the ensemble members of each MNN. Firstly, we split the training set into K (the number of the component networks) subsets. Then on each subset a component network is trained. The training parameters are indicated in Table 2 for each

Table 1
Eight regression examples.

Name	Function	Variable (s)
2-D Mexican Hat (MH2)	$y = \frac{\sin\lvert x\rvert}{x}$	$x \sim U[-2\pi, 2\pi]$
3-D Mexican Hat (MH3)	$y = \sin\sqrt{x_1^2 + x_2^2}/\sqrt{x_1^2 + x_2^2}$	$x_i \sim U[-4\pi, 4\pi]$
Friedman (F1)	$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$	$x_i \sim U[0,1]$
Friedman (F2)	$y = \sqrt{x_1^2 + \left(x_2 x_3 - \left(\frac{1}{x_2 x_4}\right)\right)^2}$	$x_1 \sim U[0,100]$, $x_2 \sim U[40\pi, 560\pi]$, $x_3 \sim U[0,1]$, $x_4 \sim U[1,11]$.
Gabor (GB)	$y = \frac{\pi}{2}\exp\left[-2(x_1^2 + x_2^2)\right]\cos[2\pi(x_1 + x_2)]$	$x_i \sim U[0,1]$
Multi-interaction (MI)	$y = 0.79 + 1.27x_1x_2 + 1.56x_1x_4 + 3.42x_2x_3 + 2.06x_3x_4x_5$	$x_i \sim U[0,1]$
Plane (PL)	$y = 0.6x_1 + 0.3x_2$	$x_i \sim U[0,1]$
Polynomial (PO)	$y = 1 + 2x + 3x^2 + 4x^3 + 5x^4$	$x \sim U[0,1]$

Table 2
Generation of data sets

Problem	Size of dataset	Partition of dataset	Number of component networks	Training error limit	Parameters
MH2	3000	1200–1000–800	4	1e−6	5000
MH3	4000	2000–1000–1000	5	1e−5	3000
Fl	5000	3200–1000–800	8	1e−6	5000
F2	21,000	9000–10,000–2000	6	1	3000
GB	3000	1200–1000–800	6	1e−6	5000
MI	3000	1200–1000–800	8	1e−4	5000
PL	3000	1200–1000–800	5	1e−6	5000
PO	3000	1200–1000–800	4	1e−6	5000

Table 4
Generalization performance comparison.

	MH2	MH3	F1	F2	GB	MI	PL	PO
The proposed	3.282e−7	1.717e−5	3.322e−5	3.99178	5.992e−6	0.000474	3.383e−7	5.987e−7
Scheme in (Perrone & Cooper, 1994)	6.734e−6	0.001193	0.357947	1225.76	1.695e−5	0.00053	6.112e−7	5.559e−7

regression problem. Then the proposed sequential Bayesian learning algorithm is used to perform information aggregation. Table 4 shows the generalization performance of the proposed method in comparison with the scheme used in (Perrone & Cooper, 1994). It can be seen that the proposed algorithm outperforms the scheme used in (Perrone & Cooper, 1994) for aggregating the outputs of ensemble members of MNNs on the eight benchmark problems.

4. Conclusion

In this paper, a sequential Bayesian learning method is proposed for modular neural network to combine the outputs of independent component networks. At each stage the likelihood function (LF) value of each component network is computed first, then the combination weights are adjusted according to Bayesian reasoning, in which the current weights are posterior and the previous weights are prior. The results of experiments on the eight benchmark problems show that the proposed method outperforms the widely used scheme for MNN.

Acknowledgements

This research has been supported by the NSFC (National Natural Science Foundation of China) under the Grant 60174039, the research fund from the Institute of Systems Science and Engineering of Wuhan University of Technology, and the research fund from the Changjiang Scholar Program, Chinese Ministry of Education.

References

Buntine, W. L., & Weigend, A. S. (1991). Bayesian back-propagation. *Complex Systems, 5*(6), 603–643.
Carvalho, R. A., & Costa, H. G. (2007). Application of an integrated decision support process for supplier selection. *Enterprise Information Systems, 1*(2), 197–216.
Chen, K., & Chi, H. (1999). A modular neural network architecture for pattern classification based on different feature sets. *International Journal of Neural Systems, 9*(6), 563–581.
Cho, S. B., & Kim, J. H. (1995). Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems Man and Cybernetics, 25*(2), 380–384.
Deepak, S. (1999). Multiple neural-network-based adaptive controller using orthogonal activation function neural networks. *IEEE Transactions on Neural Networks, 10*(6), 1484–1501.
Drake, P., & Packianalher, M. (1998). Decision tree of neural networks for classifying images of wood veneer. *International Journal of Advanced Manufacturing Technology, 14*(4), 280–285.
Eccles, J. C. (1984). The cerebral neocortex: A theory of its operation. In E. G. Jones & A. Peters (Eds.). *Cerebral cortex: Functional properties of cortical cells*. Plenum Press.
Edelman, G. M. (1979). Group selection and phasic reentrant signaling: A theory of higher brain function. In F. O. Schmitt & F. G Worden (Eds.), *The neurosciences: Fourth study program*. Cambridge, MA: MIT Press.
Feng, S., & Wang, P. (2007). A modular neural network simulation system for teaching/research based on .NET framework. In *Proceedings of 1st ISITAE, IEEE, Piscataway, NJ* (pp. 422–427).
Hansen, L. K., & Salamon, P. (1990). Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*, 993–1000.
Happel, B., & Murre, J. (1994). The design and evolution of modular neural network architectures. *Neural Networks, 7*, 985–1004.
Haykin, S. (1999). *Neural networks: A comprehensive foundation*. Prentice Hall.
Hashem, S. (1997). Optimal linear combinations of neural networks. *Neural Networks, 10*(4), 599–614.
Hubel, D. H., & Livingstone, M. (1990). Color and contrast sensitivity in the lateral geniculate body and primary visual cortex of the macaque monkey. *Journal of Neuroscience, 10*, 2223–2237.
Jordan, M. I., & Jacobs, R. (1991a). A competitive modular connectionist architecture. In *Advances in neural information processing systems* (pp. 767–773). San Maeto, CA: Morgan Kaufmann Publisher.
Jordan, M. I., & Jacobs, R. (1991b). Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science, 15*, 219–250.
Kim, K., & Park, J. (1993). Application of hierarchical neural networks to fault diagnosis of power systems. *International Journal on Electrical Power and Energy System, 15*(2), 65–70.
Lampinen, J., & Vehtari, A. (2001). Bayesian approach for neural networks-review and case studies. *Neural Networks, 14*, 257–274.
Li, L. (1999a). Proposing an architectural framework of hybrid knowledge-based system for production rescheduling. *Expert Systems, 16*(4), 273–279.
Li, L. (1999b). Knowledge-based problem solving: An approach to health assessment. *Expert Systems with Applications, 16*(1), 33–42.
Li, H. X., & Li, L. (1999). Representing diverse mathematical problems using neural networks in hybrid intelligent systems. *Expert Systems, 16*(4), 262–272.
Pan, Y., & Sensen, C. (2005). Modular neural networks and their applications in exon prediction. In *Advances in bioinformatics and its applications series in mathematical biology and medicine* (pp. 47–61). Berlin: Springer. Vol. 8.
Perez, C., & Galdames, P. (1998). Improvement on handwritten digit recognition by cooperation of modular neural networks. In *Proceedings of IEEE international conference on systems, man and cybernetics, IEEE, Piscataway, NJ* (pp. 4172–4177).
Perrone, M., & Cooper, L. (1994). When network disagree: Ensemble methods for hybrid neural networks. In R. Mammone (Ed.), *Artificial neural networks for speech and vision* (pp. 126–142). London, UK: Chapman and Hall.
Qi, G., Liu, W., & Bell, D. (2007). Combining multiple prioritized knowledge bases by negotiation. *Fuzzy Sets and Systems, 158*, 2535–2551.
Sibte, S., & Abidi, R. (1996). Neural networks and child language development: A simulation using a modular neural network architecture. In *Proceedings of International Conference on Neural Networks, IEEE, Piscataway, NJ* (pp. 840–845).
Srinivas, G., Jeffrey, R., Huang, J., Phillips, P., & Wechsler, H. (2000). Mixture of experts for classification of gender, ethnic origin, and pose of human faces. *IEEE Transactions on Neural Networks, 11*(4), 948–959.
Tahani, H., & Keller, J. (1990). Information fusion in computer vision using the fuzzy integral. *IEEE Transactions Systems, Man, and Cybernetics, 20*(3), 733–741.
Van Hulle, M. M., & Tollenacre, T. (1993). A modular artificial neural network for texture processing. *Neural Networks, 6*(1), 7–32.
Wang, L., Der, S., & Nasrabadi, N. (1998). Automatic target recognition using a feature-decomposition and data-decomposition modular neural network. *IEEE Transactions on Image Processing, 8*, 1113–1121.
Wang, P., Fan, Z., & Li, Y. F. (2006). Dynamic integration of modular neural network's sub-networks. *Dynamics of Continuous, Discrete and Impulsive Systems Series B, S1*, 2280–2283.
Wang, P., Feng, S., & Fan, Z. (2008). Some issues of the paradigm of multi-learning machine- modular neural networks. In *Proceedings of ISCSN, Salt Lake City, USA* (pp. 388–395).
Xu, L. (1988). A fuzzy multi-objective programming algorithm in decision support systems. *Annals of Operations Research, 12*, 315–320.
Xu, L., Krzyzak, A., & Suen, C. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics, 22*, 418–435.
Xu, L., Wang, C., Luo, X., & Shi, Z. (2006). Integrating knowledge management and ERP in enterprise information systems. *Systems Research and Behavioral Science, 23*(2), 147–156.
Xu, W. X., Xu, L., Liu, X., & Jones, J. (2008). A new approach to decision-making with key constraint and its application in enterprise information systems. *Enterprise Information Systems, 2*(3), 287–308.
Zhao, S., & Li, Z. (2009). Assembly information modelling and sequences generation algorithm of autobody. *Enterprise Information Systems, 3*(2), 159–172.
Zhou, S. M., & Xu, L. (1999). Dynamic recurrent neural networks for a hybrid intelligent decision support system for the metallurgical industry. *Expert Systems, 16*(4), 240–247.
Zhou, S. M., & Xu, L. (2001). A new type of recurrent fuzzy neural network for modeling dynamic systems. *Knowledge-Based Systems, 14*(5), 243–251.
Zhou, S. M., & Gan, J. (2007). Constructing parsimonious fuzzy classifiers based on L2-SVM in high-dimensional space with automatic model selection and fuzzy rule ranking. *IEEE Transactions on Fuzzy Systems, 15*, 398–409.
Zhou, S. M., Chiclana, F., John, R., & Garibaldi, J. (2008). Type-1 OWA operators for aggregating uncertain information with uncertain weights induced by type-2 linguistic quantifiers. *Fuzzy Sets and Systems, 159*(24), 3281–3296.
Zhou, Z., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence, 137*(1-2), 239–263.