INFORMATION NETWORKS WITH MODULAR EXPERTS

J.P. Thivierge
Department of Psychology
McGill University
Montreal, QC Canada
H3A 1B1
management@neurostate.com

T. R. Shultz
Department of Psychology and School of Computer Science
McGill University
Montreal, QC Canada
H3A 1B1
thomas.shultz@mcgill.ca

Abstract

Information networks learn by adjusting the amount of information fed to the hidden units. This technique can be expanded to manipulate the amount of information fed to modular experts in a network's architecture. After generating experts that vary in relevance, we show that competition among them can be obtained by information maximization. After generating equally relevant but diverging experts using AdaBoost, collaboration among them can be obtained by constrained information maximization. By controlling the amount of information fed to the experts, we can outperform a number of other mixture models on real world data.

Key Words

neural networks, information theory, ensemble learning, mixture-of-experts, AdaBoost.

1. Introduction

Researching ways to generate, select, map, and combine available experts for problem solving is a growing area of interest in machine learning. The benefits of using such experts can extend to faster learning times, less necessary computation, and increased training and generalization accuracy. A number of techniques address certain aspects of this problem. Algorithms such as bagging [1] and boosting [2] offer ways to generate experts later to be used by a classifier. Networks such as Knowledge-based Cascade-correlation [3] address the problem of knowledge selection. Networks such as mixture-of-experts [4] can combine many experts to solve a given task. A number of other algorithms have also been devised to make use of previously acquired knowledge, including discriminability-based transfer [5], multi-task learning [6], explanation-based neural networks [7], and knowledge-based artificial neural networks [8].

In the current article, we propose a new technique based on information theory to address the selection, mapping, and combination of expert knowledge. Knowledge selection, in particular, is a problem often ignored in knowledge-based systems. Information theory can offer a helpful framework for performing knowledge transfer.

Information theoretic approaches have been introduced in various ways into neural computing, including maximum information preservation [9], minimum redundancy [10], spatially coherent feature detection [11], and identification of independent input subsets [12]. These applications of information theory to neural networks have lead to improved generalization, and more easily interpretable solutions. The *modus operandi* of these algorithms is to control the amount of information from the environment absorbed by the hidden units. This technique can be expanded to control the amount of information in pre-acquired experts. By concentrating information in a small number of experts, they are forced to compete for information content. Conversely, by distributing information across the experts, they can collaborate towards a solution. The acronym MINEKA (Mixture of Information Networks with Expert Knowledge Attribution) describes the general idea behind these networks.

2. Competition Among Experts

In this section, we apply information maximization and minimization [13] to the input-hidden connections of MINEKA networks. The goal here is to select a winning expert that best corresponds to the task being learned.

2.1 Description of the Algorithm

The general goal of MINEKA networks is to decrease the uncertainty [14] of relevant experts in classifying the input patterns. Maximum uncertainty is present when experts output a value at mid-level of their full activation [15][16], and minimum uncertainty is present when experts output a value at either full or zero activation. If an expert is considered to be important, information in it should be increased. On the other hand, unnecessary experts should not contain information on the input patterns.

In order for experts to compete, we assume they are of varying relevance in solving the target task. The main

network must solve the task by first selecting the best expert, and then mapping it correctly. Thus experts compete to find out which offers the most appropriate solution to the task.

Let Y denote a set of experts $Y = \{y_1, \ldots, y_{\Re}\}$. The probability of occurrence of the j^{th} expert y_j is given by a probability $p(y_j)$. The conditional probability given the s^{th} input pattern of a set of input patterns $X = \{x1, \ldots, x_S\}$ is $p(y_j \mid x_s)$, The average uncertainty of the experts Y and the input patterns X is represented by $H(Y)$ and $H(X)$ respectively. The conditional uncertainty of Y, given X, is represented by $H(Y \mid X)$. The information content of the experts Y, given input patterns X, is defined as:

$$\begin{aligned} I(Y|X) &= -\sum_{j=1}^{Q} p(y_j)\log p(y_j) \\ &\quad + \sum_{s=1}^{S} p(x_s)\sum_{j=1}^{Q} p(y_j|x_s)\log p(y_j|x_s) \\ &\approx \log Q + \sum_{s=1}^{S}\frac{1}{S}\sum_{j=1}^{Q} p_j^s \log p_j^s \end{aligned} \tag{1}$$

where Q denotes the maximum uncertainty, and p_j^s is a normalized output of the j^{th} expert:

$$\vec{p}_j^{\,s} = \frac{\vec{v}_j^{\,s}}{\sum_{m=1}^{M}\vec{v}_m^{\,s}} \tag{2}$$

where M is the total number of experts. Information in the experts can be approximated by:

$$I_j(Y|X) = \log 2 + \frac{1}{S}\sum_{s=1}^{S}\left(\vec{v}_j^{\,s}\log\vec{v}_j^{\,s} + \bar{\vec{v}}_j^{\,s}\log\vec{v}_j^{\,s}\right) \tag{3}$$

where log2 is the maximum uncertainty, and $\vec{v}$ is the vector of activations coming out of expert j for a pattern s. Occurrence of input patterns is considered to be equi-probable, namely, $1/S$. Hence, the true entropy is estimated by a cross-entropy of the error signal as:

$$G = \sum_{i=1}^{N}\left[\frac{1}{S}\sum_{s=1}^{S}\left\{\varsigma_i^s\log\frac{\varsigma_i^s}{O_i^s} + \bar{\varsigma}_i^s\log\frac{\bar{\varsigma}_i^s}{O_i^s}\right\}\right] \tag{4}$$

where ς_i^s is a target for the output O_i^s from the i^{th} output unit, N is the number of output units, S is the number of input patterns, $\bar{O}_i^s = 1 - O_i^s$ and $\bar{\varsigma}_i^s = 1 - \varsigma_i^s$. The network outputs a value i for pattern s as:

$$O_i^s = f\left(\sum_{j=0}^{M}\vec{W}_{ij}\vec{v}_j^{\,s}\right) \tag{5}$$

where $\vec{W}$ represents a vector of hidden-output connections from the j^{th} expert to the i^{th} output unit. The quadratic error function is:

$$E = \sum_{i=1}^{N}\frac{1}{S}\sum_{s=1}^{S}\left(\varsigma_i^s - O_i^s\right) \tag{6}$$

The weight update rule for information maximization is obtained by differentiating the error function E with respect to the information I and the cross-entropy G:

$$\Delta w_{jk} = \beta\sum_{s=1}^{S}\left(\log p_j^s - \sum_{m=1}^{M} p_m^s\log p_m^s\right)p_j^s\vec{v}_j^{\,s}\xi_j^s + \eta\sum_{s=1}^{S}\delta_j^s\xi_k^s \tag{7}$$

where β and η are learning rate parameters, and where

$$\delta_j^s = \sum_{i=1}^{N}\left(\varsigma_i^s - O_i^s\right)\vec{W}_{ij}\vec{v}_j^{\,s}\vec{v}_j^{\,s} \tag{8}$$

Information minimization is the reverse process of maximization. In this case, we want to increase uncertainty across the available resources, in order increase generalization [13]. The update rule in (7) can be modified for this purpose:

$$\Delta w_{jk} = -\beta\sum_{s=1}^{S}u_j^s\vec{v}_j^{\,s}\vec{v}_j^{\,s}\xi_k^s + \eta\sum_{s=1}^{S}\delta_j^s\xi_k^s \tag{9}$$

where

$$\delta_j^s = \sum_{i=1}^{N}\left(\varsigma_i^s - O_i^s\right)O_i^s\bar{O}_i^s\vec{W}_{ij}v_j^s\vec{v}_j^{\,s} \tag{10}$$

where N is the number of output units, and u is an input into the j^{th} expert:

$$u_j^s = \sum_{k=0}^{L}\vec{W}_{jk}\xi_k^s \tag{11}$$

where ξ represents the k^{th} element of the s^{th} input pattern, and L is the number of input units.

Information maximization and minimization are applied to the training of competitive MINEKA networks in the following way (see Figure 1):

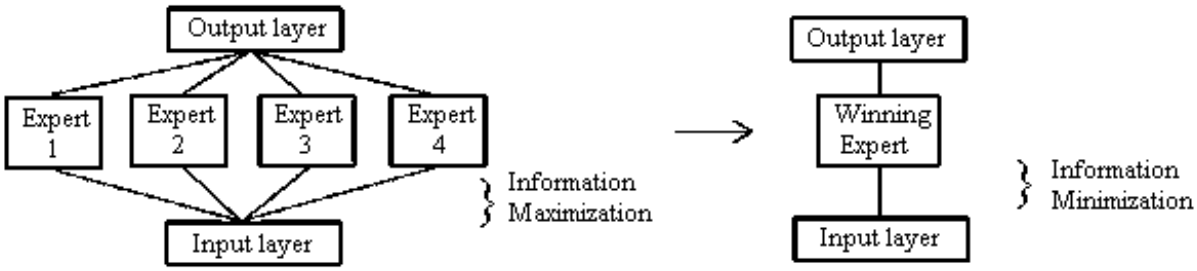


Figure 1. Architecture of the Mixture-of-expert networks employed in the simulations.

1) Define an initial network by connecting every modular expert to both the input and output layers.
2) Train using information maximization (equation 7) on the input-hidden connections.
3) Re-consolidate the network by retaining only the expert with the highest information content.
4) Apply information minimization (equation 9) to the input-hidden connections in order to re-distribute information across the input nodes of the winning expert.

2.2 Experiments

Experiments were performed using the glass database from the PROBEN1 repository [17]. Values from this problem were first normalized, then divided into a training set and a 10-fold cross-validation set used for testing. In order to assess resistance to noise of MINEKA, four different train sets were generated by removing none, 30%, 50%, and 70% of the data respectively. This was performed by randomly replacing values by the normalized average of a given dataset, thus turning them into "unknown" values.

Separate MINEKA networks were trained on each of the cross-validation folds, and on each of the four impoverished train sets, totaling 40 networks. For comparison purposes, backpropagating networks with no manipulation of information content ("EXPERT_BP") were also used. All networks were fitted with a total of four modular experts. In order to vary the information content of these experts, each was trained on a different amount of data. One expert was trained on the full problem, another on a 10% impoverished set, yet another on a 30% impoverished set, and finally another on a 70% impoverished set. In order to assess early use of expert knowledge, the maximum number of training epochs was set to 100.

Regardless of the impoverishment of the target task, MINEKA always attained higher performance for both training (Table 1) and generalization (Table 2). Figure 2 compares the information content of experts in EXPERT_BP and MINEKA networks on the glass problem with no impoverishment. After training, information in the EXPERT_BP was distributed among the experts. MINEKA, however, concentrated information in the expert that best solved the task, and all other experts received virtually no information.

Table 1. Average training mean squared error of networks with competing experts

IMPOVERISHMENT	MINEKA	EXPERT_BP
none	58.27	712.71
10%	66.26	740.74
30%	236.12	772.46
50%	297.82	748.9
70%	95.79	750.06

Table 2. Average generalization mean squared error of networks with competing experts

IMPOVERISHMENT	MINEKA	EXPERT_BP
none	65.61	732.57
10%	69.02	741.96
30%	235.19	769.86
50%	304.58	738.93
70%	108.93	754.11

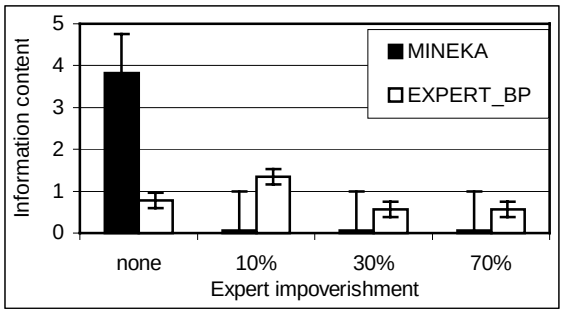
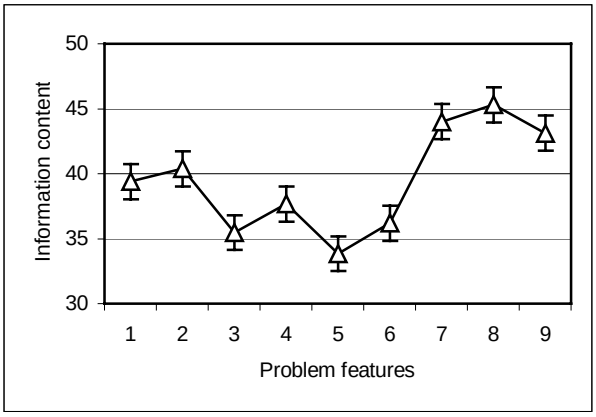


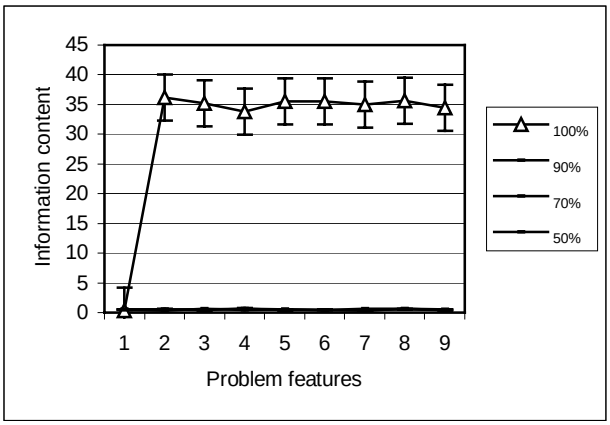
Figure 2. Information content of the various experts of trained MINEKA and EXPERT_BP networks.

These results were found regardless of the impoverishment of the target task. This means that even if networks drew on limited information to select an appropriate expert, they still effectively did so. One further observation is that information did not get distributed across the experts according to the rank of their relevance. Rather the networks selected a single winning expert and disregarded all others. Figure 3 shows the effect of information maximization and

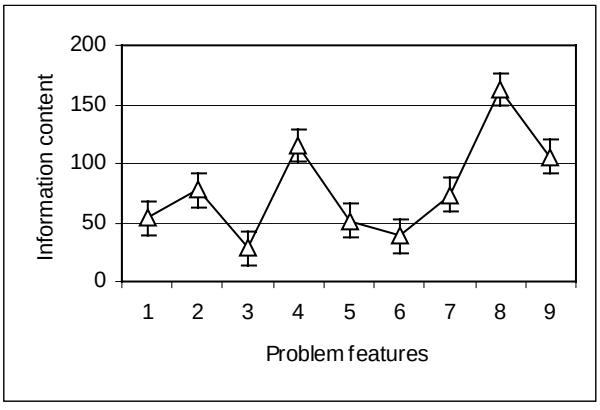
minimization on information in MINEKA. By maximization, variance among the features of the problem was lost for the winning expert (3b). In 3c, minimization restored some variance originally found among the problem's features (3a). In summary, manipulation of information content was found to improve training and testing performances. In addition, competitive MINEKA was able to select the best expert regardless of the amount of noise present in training. The question to be answered next is whether performance can be further improved by having experts collaborate rather than compete.



(a)



(b)



(c)

Figure 3. Average information across features of the glass database. (a) prior to maximization; (b) MINEKA after maximization for the various experts; (c) MINEKA after minimization.

3. Collaboration Among Experts

(b)

In this section, we first propose a way to generate experts that can collaborate in a MINEKA network. Then, we describe a training procedure that allows collaboration instead of competition. Nonlinear collaboration among neural networks has been performed with information theory in some models [18]. Our approach differs in that we control the amount of information fed to the experts rather than the way in which they ultimately combine. In addition, our technique forces the concentration of information in experts rather than just assessing it.

3.1 Description of the Algorithm

The technique used for generating experts is AdaBoost [2], which aims at producing correct experts that diverge as much as possible from one another. AdaBoost works by assigning weight values to each example of a training set. Initially all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the expert is forced to focus on the hard examples in the training set. Adapted to MINEKA, AdaBoost works by first initializing a distribution D over the training set as $D_y(s) = 1/S$. Then for every expert $y,...,Y$, a base classifier c_y is trained using distribution D_y. From one expert to the other, D gets updated as:

$$D_{y+1}(i) = \frac{D_y(i)\exp(-\alpha_y \varsigma_i c_y(x_i))}{Z_y} \tag{12}$$

where Z_y is a normalization factor chosen such that D_{y+1} is a distribution, and $\alpha_y \in \Re$ is the classification accuracy of c_y. The quadratic error function of MINEKA described in (6) is adapted to take D into account by considering the weight of each training example when computing its classification error:

$$E = \sum_{i=1}^{N} \frac{1}{S} \sum_{s=1}^{S} D_{y+1}(i)\left(\varsigma_i^s - O_i^s\right) \tag{13}$$

In standard AdaBoost, the final classifier combines the experts according to their respective alpha values. In MINEKA networks, however, we let the network find the best way to use experts by adjusting the input-hidden weights. Thus, the final classifier that combines the c_y experts is obtained as:

$$C(x) = sign\left(\sum_{j=1}^{M}\sum_{k=1}^{K}\sum_{y=1}^{Y} w_{jk} h_y(x)\right) \quad (14)$$

Because our goal is to maximize information in a number of experts that will collaborate with each other and share the total information, we did not want one expert to hog the information content of a data set. To this purpose, we employed constrained information maximization [19] by forcing the sum of activations coming out of experts to equal a fixed value (theta):

$$\sum_{m=1}^{M} v_m^s = \theta \quad (15)$$

This constraint prohibits information from concentrating in a small portion of the experts, thus granting all experts a chance to gain some information content. Given this constraint, the delta for input-hidden weights can be derived as:

$$\begin{aligned}\Delta w_{jk} = {} & \beta\sum_{s=1}^{S}\left(\log p_j^s - \sum_{r}^{M} p_r^s \log p_r^s\right) p_j^s \bar{v}_i^s \xi_k^s \\ & - \gamma\sum_{s=1}^{S}\left(\sum_{m=1}^{M} v_m^s - \theta\right) v_j^s \bar{v}_i^s \xi_k^s \\ & + \eta\sum_{s=1}^{S}\left\{\sum_{i=1}^{N}\left(\varsigma_i^s - O_i^s\right) W_{ij}\right\} v_j^s \bar{v}_j^s \xi_k^s \end{aligned} \quad (16)$$

where γ, β, and η are learning rate parameters.

In summary, collaborative MINEKA is generated by first training a series of experts using AdaBoost. Then, these experts are incorporated in MINEKA on a single hidden layer, as in the previous section. Finally, the network is trained using constrained maximization.

3.2 Experiments

Experiments were carried out using the diabetes, cancer, and horse problems from the PROBEN1 repository. As in the previous section, the data sets were normalized and divided in a training and test set according to a 10-fold cross-validation. Comparisons were made between collaborative MINEKA ("M_COLL"), competitive MINEKA ("M_COMP"), Backpropagation ("EXPERT_BP"), and a batch version of the popular mixture-of-experts algorithm ("MIX") [4]. Ten of each type of networks were ran. All networks received three experts generated by AdaBoost, except for competitive MINEKA, where four experts were generated with impoverished sets as in the previous section. A limit of 100 training epochs was again imposed.

Results are reported in Tables 3-4. Collaborative MINEKA networks outperformed competitive MINEKA, EXPERT_BP, and mixture-of-experts on both training and generalization accuracy. The fact that collaborative MINEKA outperformed competitive MINEKA is congruent with empirical demonstrations that subdividing a task in ensemble learning can improve performance [2].

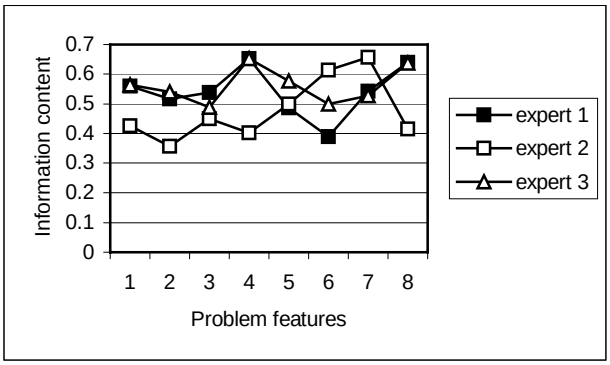
Table 3. Average training mean squared error of networks with collaborating experts

	M_COLL	M_COMP	EXPERT_BP	MIX
Diabetes	49.73	67.52	124.21	68.47
Cancer	44.3	113.81	110.74	166.44
Horse	65.08	85.4	92.7	97.1

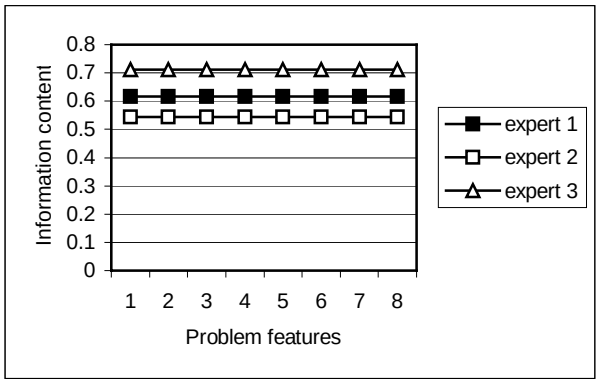
Table 4. Average cross-validation mean squared error of networks with collaborating experts

	M_COLL	M_COMP	EXPERT_BP	MIX
Diabetes	48.09	57.6	109.24	65.37
Cancer	45.52	103.42	111.86	152.66
Horse	67.14	68.61	87.67	93.1

Figure 4 depicts the effect of constrained maximization on



(a)



(b)

Figure 4. Effect of constrained maximization on the information content of experts. (a) before mazimization; (b) after maximization.

the information content of experts for a collaborative MINEKA network on the diabetes problem. As for competition, collaborative maximization distributes variance from the input evenly across the experts. However, collaboration differs in that all experts retain some information content on the given task.

In summary, collaborative MINEKA outperformed competitive MINEKA, mixture-of-experts networks, and backpropagating networks in both training and generalization accuracy. MINEKA made use of all experts present by maximizing information among them.

4. Conclusion

We investigated ways to adapt information theory to the training of networks that combine modular experts. A first algorithm enabled competition among experts by maximizing information in one of the experts and suppressing it in the others. A second algorithm enabled collaboration among the experts by spreading information among the experts. Results of the two algorithms show that manipulating the information content of networks improved their training and generalization accuracy. The performance of collaborative MINEKA networks surpassed that of mixture-of-experts [4] and competitive MINEKA networks in all the databases tested.

One area of future exploration consists in findings ways to determine *a priori* if a technique of knowledge competition or collaboration should be applied to solving a particular task given a number of experts. Also, a new version of MINEKA could be devised where the experts learn to specialize in solving different regions of the error surface, as in mixture-of-experts networks [4].

Due to their use of higher-order statistics, information-based networks have the potential to detect intricate relations between experts in a learned solution. Information theory thus offers a promising approach to the problems of knowledge transfer, selection, and mapping in neural networks.

5. Acknowledgements

This research was supported scholarships to J.P.T. from FCAR (Québec) and Tomlinson (McGill University), and grants to T.R.S. from FCAR (Québec) and NSERC (Canada).

6. References

[1] Linsker, R. (1992). Local synaptic rules suffice to maximize mutual information in a linear network. *Neural Computation, 4*, 691-702.

[2] Atick, J.J., & Redlich, A.N. (1990). Toward a theory of early visual processing. *Neural Computation, 2*, 308-320.

[3] Becker, S. (1996). Mutual information maximization: models of cortical self-organization. *Neural Computation, 7*, 7-31.

[4] Sridhar, D.V., Bartlett, E.B., & Seagrave, R.C. (1998). Information theoretic subset selection for neural network models. *Comput. Chem. Engng., 22*, 613-626.

[5] Breiman, L. (1994). Bagging predictors. *Technical Report No. 421*. University of California.

[6] Shapire, R.E. (2002). The boosting approach to machine learning: An overview. *MSRI Workshop on Nonlinear Estimation and Classification*.

[7] Shultz, T. R., & Rivest, F. (2001). Knowledge-based cascade-correlation: Using knowledge to speed learning. *Connection Science, 13*, 43-72.

[8] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., & Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*, 79, 87.

[9] Pratt, Y.L. (1993). Discriminability-based transfer between neural networks. In S.J. Hanson, C.L. Giles, and J.D. Cowan (Eds.). *Advances in Neural Information Processing Systems 5*. (pp. 204-211). Morgan Kaufmann.

[10] Silver, D. & Mercer, R. (1996). The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. In L. Pratt (Ed.). *Connection Science Special Issue: Transfer in Inductive Systems*. (pp. 277-294). Carfax Publishing Company.

[11] Mitchell, T.M., & Thrun, S.B. (1993). Explanation-based neural network learning for robot control. *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann (pp.287-294). San Mateo, CA.

[12] Shavlik, J.W. (1994). A framework for combining symbolic and neural learning. *Machine Learning, 14*, 321-331.

[13] Kamimura, R., Takagi, T., & Nakanishi, S. (1995). Improving generalization performance by information minimization. *IEICE Transactions on Information and Systems, E78-D*, 163-173.

[14] Gatlin, L.L. (1972). *Information Theory and Living System*. New York: Columbia University Press.

[15] Bridle, J., MacKay, D., & Heading, A. (1994). Unsupervised classifier, mutual information and phantom targets. *Neural Information Processing Systems, 4*, (pp. 1096-1101. Morgan Kaufmann Publishers, San Mateo: CA,

[16] Bruce, C., Desimonde, R., & Gross, C.G. (1981). Visual properties of neurons in a polysensory area in superior temporal sulcus of the macaque. *Journal of Neurophysiology, 46*, 369-384.

[17] Prechelt, L. (1994). PROBEN1 - A set of benchmarks and benchmarking rules for neural network training algorithms. *Technical report 21/94*, Fakultät für Informatik, Universität Karlsruhe.

[18] Sridar, D.V., Barlett, E.B., & Seagrave, R.C. (1999). An information theoretic approach for combining neural network process models. *Neural Networks, 12*, 915-926.

[19] Kamimura, R. (2002). *Controlling Entropy with Neural Network Detectors* (first edition). World Scientific Pub Co.