Automatic Modularization of ANNs Using Adaptive Critic Method*

RUDOLF JAKŠA†
Kyushu Institute of Design
4-9-1 Shiobaru, Minami-ku, Fukuoka, 815-8540
JAPAN

Abstract: - We propose automatic modularization method for artificial neural networks. Adaptive critic approach is used for structure optimization of the network during the learning, where modular structure is defined to be optimal. The idea is to start with plain unstructured topology of the network and to finish learning with modular neural network. This approach not only learns to map inputs to outputs, but it also tries to discover structure of knowledge represented by training patterns.

Key-Words: - structured (or modular) neural networks, structure optimization, modularization, adaptive critic method

1 Introduction

Structured, or modular, neural networks (ANNs) are common type of ANNs, however structure of these networks is usually fixed. Various methods exist for optimization of structure (topology) for unstructured (or plain) neural networks, this include pruning algorithms and GA (genetic algorithm) evolved neural networks with optimization of topology. However, these algorithms do not produce modular or hierarchical structure of network, they only reduce every-to-every connection scheme to more sparse network topology.

Structured neural networks are usually of NARA[1] type or "Mixture of Experts"[2] type. Both these types of ANN structures have similar topology. This topology consist of several parallel modules specialized for particular subtasks of whole task and single antecedent module. This antecedent module control participation of particular modules on final output of the network according to position of actual input in input space. Described modular structure is easy to follow for humans and allows understanding of functionality of such a ANN or embedding of apriori knowledge into such a network. This type of structure is even used for rough modeling of brain functionality, where several areas of cortex are responsible for several different tasks.

Proposed method is inspired by NARA modular structure of network. Automatic building-up of this structure is treated as optimization task. Therefore, optimization criteria is defined for the modularization and adaptive critic method is used for this optimization. Note that other optimization methods can be used here as well (for instance evolutionary computation).

2 Automatic Modularization

Classical modular networks (of the NARA type) consist of several independent modules, where nodes in one module are not connected to nodes in other modules. In the plain ANN with every-to-every connection scheme every node depends on all the others (or an all nodes in previous layer).

In process of evolving modular network from plain network, it is logical to go through interim state, where nodes inside particular module are connected tightly and nodes from two different modules depend only a little bit.

Let define the dependency of two nodes to be corresponding to strength of the connection between them. This connection can be direct or indirect using intermediate nodes. In simplified case, two nodes with

*This work is supported by Japan Society for the Promotion of Science (JSPS).

†Author is on leaving from Technical University of Košice.

small weight on direct connection are less dependent as those with high weight on this connection. Using this assumption, module is a group of nodes with big weights on connections inside module (intramodule connections) and small weights on connections to other modules (intermodule connections). Given definition of "soft module" is illustrated on figure 1 by "structured plain" structure (dotted lines represent links with small weights).

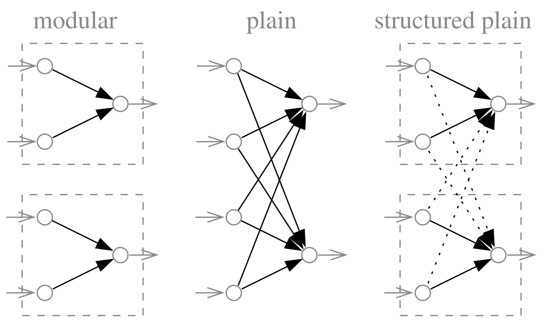

Fig.1: Types of ANN modularity.

To discover modular structure of ANN two basic approaches can be used:

passive: after learning of the network, analysis is done to find modular structure,

active: during learning, modularization algorithm is run and modular structure is developed iteratively.

Passive approach can be based on ANN analysis algorithms (for instance [5]) and is similar to analysis of source code in reverse-engineering field [4]. Discovered structure cannot be further optimized using passive approach, while active approach allows real structure optimization.

2.1 Optimization Criteria

Both, passive and active, approaches need optimization criteria to be defined. In passive approach this criteria is necessary for finding the best partition of fixed network (see [4]). In active approach this criteria is optimization criteria, used to "grow" modules inside the network.

Optimization criteria for minimization of intermodule connections for single module m can be defined as:

$$J_e^m = \sum_{e=0}^{N_e^m} |w_e| \tag{1}$$

where N_e^m is the number of intermodule connections of module m and w_e are weights on these connections. Value J_e^m is zero if module m is disconnected from other modules.

Optimization criteria for maximization of intramodule connections for single module m can be defined as:

$$J_i^m = -\sum_{i=0}^{N_i^m} |w_i| \tag{2}$$

where N_i^m is the number of intramodule connections of module m and w_i are weights on these connections. Value J_i^m is zero if module m has no internal connections and it is less then zero if intramodule connections appear.

Rule (2) doesn't distinguish between following two situations:

a) single very high weight and a couple of small weights,
b) many moderately high weights.

Case b) represents actually what we want. To comply with this situation the criteria function (2) should be modified. To disable case a) we can ignore high weights and do not include them to rule (2). It can be done by limiting weights to fixed maximal value. New criteria function is:

$$J_i^m = -\sum_{i=0}^{N_i^m} |f_t(w_i)| \tag{3}$$

where f_t is threshold function:

$$f_t(w) = \begin{cases} w & \text{, if } |w| < w_{max} \\ w_{max} & \text{, otherwise} \end{cases} \tag{4}$$

where w_{max} is maximal desired weight, further maximization of such a high weight doesn't result in better evaluation by criteria (3).

Even if using criteria (3), it is possible that internal structures will emerge inside module not because of natural modular essence, instead only to comply with given criteria. For instance, some elimination structures, where high positive and negative weights are used to eliminate themselves and result is constant zero. Further work on analysis of ANN structures is necessary to comply with such a risks as well as for understanding of structuralization in ANNs. Different approaches for building-up strong intramodule connectivity can be used as well too, for instance pruning algorithms to eliminate weak connections inside modules or "Branch Control" approach [7] can be used to determine nodes with related function and to group them into modules.

Criteria (1), (2) and (3) are based on previous simplifications and on assumption that dependency of modules is based only on direct connections between nodes. They should be taken as basic criteria with possible further extensions.

To minimize described criteria it is necessary to use algorithm which is able to minimize arbitrary criteria or to develop a custom optimization algorithm. Although development of simple custom algorithm dealing with criteria (1) and (3) can be easy, optimization should also deal with original task criteria (usually approximation or classification) and it should allow next extensions of given criteria. Because of these reasons, using algorithm able to optimize according to arbitrary criteria may be preferred. Few well-known approaches can be used for optimization according to arbitrary criteria: random search, evolutionary computation (EC) or adaptive critic (AC) method.

2.2 Adaptive Critic Method

Adaptive critic method is crucial in the reinforcement learning algorithms[6]. "Adaptive critic" is ANN used for approximation of arbitrary optimization criteria. Once this approximation is good, optimization according to this ANN ("adaptive critic") is "equal" to optimization according to original criteria. This allows to use backpropagation algorithm for optimization according arbitrary criteria.

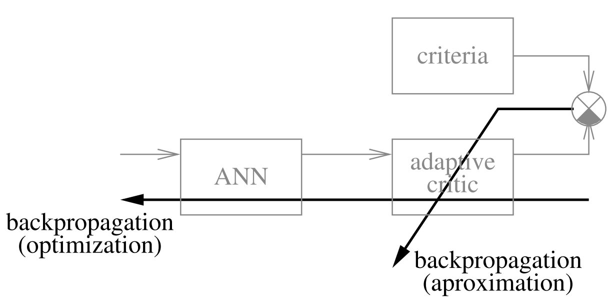

Fig.2: Optimization flows in adaptive critic method.

On figure 2 "adaptive critic" is trained (by backpropagation) to approximate "criteria" and backpropagation is also used to optimize ANN according the criteria represented by "adaptive critic". For better understanding of AC method please see [6].

2.3 Antecedent Module

NARA type networks contain antecedent module controlling the participation of other modules on the final output. This is "IF part" module in NARA and "gating" module in "Mixture of Experts" networks. Control of participation of other modules is provided by some way of weighting of outputs of these modules. In NARA, weighting is physically realized by application of (fuzzy) operators on outputs from modules. However this weighting can be also provided by simple inputs to output nodes of particular modules (figure 3).

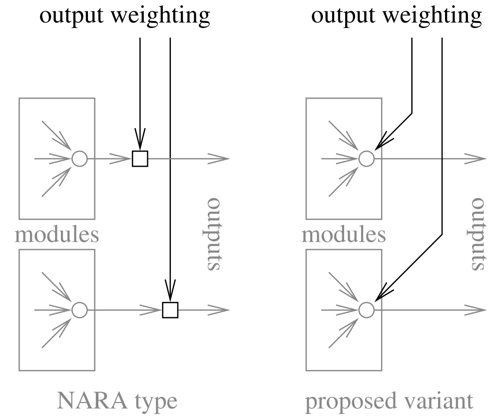

Fig.3: Weighting of modules outputs by NARA type and proposed type (circle means neuron node and square means t-norm operator).

Sufficiently intensive inhibitory signal on such a connection can effectively stop participation of given module on the final output and zero signal means full participation of module on output. This is equivalent for 0 and 1 weighting signals in NARA network.

Consider ANNs with sigmoidal activation functions $ou = 1/(1+e^{-in})$, in this case outputs ou of nodes are always positive. To obtain big inhibitory signal, it is necessary to use high negative weights on connections. This means that antecedent network should have connections to other modules with high negative weights (figure 4).

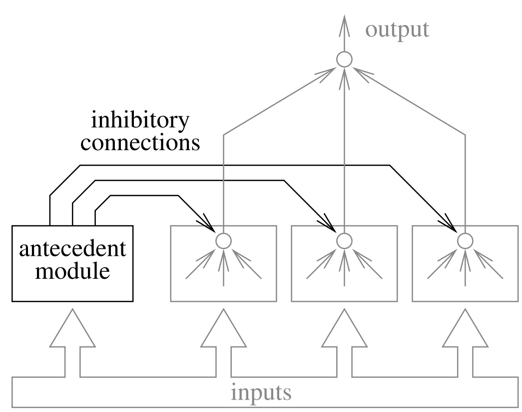

Fig. 4: Explicit antecedent module with strong inhibitory connections.

Going further with this idea, it can be useful to allow all the modules to control participation of other modules on the final output (figure 5). This approach allows emergency of hierarchical modular structures (like "Hierarchical Mixtures of Experts" presented in [3]). On the example on figure 5 the module 1 is superior in hierarchy to modules 3 and 4, and module 3 is superior to module 2.

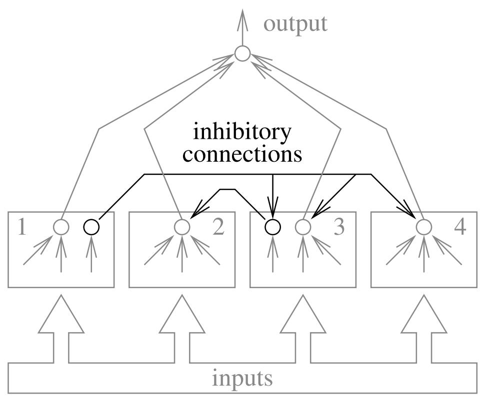

Fig.5: Hierarchical modular structure without explicit antecedent module.

This type of hierarchical modular structure can be characterized by:

1. uniform modules, no explicit antecedent module,
2. tight internal connections in every module,
3. weak intermodule connections,

3a. except few (strong) inhibitory intermodule connections.

Mentioned characteristics can be projected to the following change of criteria (1):

$$J_e^m = \sum_{e=0}^{N_e^m} |f_h(w_e)| \tag{5}$$

where function f_h is used to allow limited number of strong inhibitory intermodule connections:

$$f_h(w) = \begin{cases} 0 & \text{, if } w \in W_{inh} \\ w & \text{, otherwise} \end{cases} \tag{6}$$

where W_{inh} is set of n strongest negative intermodule connections. These connections are considered exception from "punishment" of intermodule connections and they provide antecedent part of NARA type of network.

2.4 Module Parameters

They are two important parameters of modular network: number of modules and module size (number of nodes in module). These parameters can be set fixed, as with fixed modular networks or can be adapted on-line. For on-line adaptation of parameters, the algorithm [4] from reverse-engineering of source code can be used: start with random partition of network and iteratively try similar partitions in order to find one which minimize structure optimization criteria better. This algorithm can be used for passive structure discovery in fixed network as well.

3 Experiments and Discussion

Experimental setup on figure 6 was used for proof-of-concept of proposed method. Structure with explicit antecedent module and fixed number of expected modules was defined.

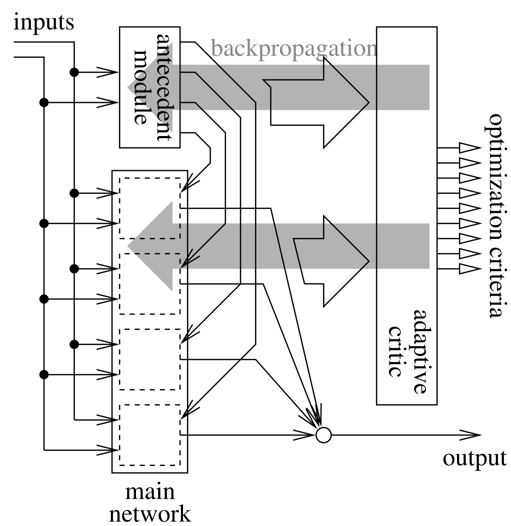

Fig.6: Experimental setup: explicit antecedent module and four expected modules in the main network.

Goal of experiment is to change plain network ("main network" on figure 6) to modular network. This can be indicated by observing of minimization of intermodule connections while "growth" of intramodule connections during the learning.

Optimization criteria for fixed number of modules were defined independently. For every expected module they were defined two criteria: one for minimization of intermodule connections (1) and second for maximization of intramodule connections (2). Next two criteria were defined for maximization of inhibitory connections from antecedent module and for actual minimization of classification error. They were 10 optimization criteria altogether. Main task was classification of two-dimensional data into two classes.

Scheme on figure 6 looks complicated, as it consists of several modules and several signal flows. However, it should be noted, that this setup was choosed in order to test "modularization" ability of proposed method. Structure on figure 5 and algorithm for adaptation of module parameters from section 2.4 should allow further refining of technique. Ideally, only two explicit modules should be used: unstructured main network and adaptive critic network.

The results of experiment are positive in sense of observing modularization process. Minimization of intermodule connections while building-up intramodule connections during learning process can be clearly observed. Experiments also showed high importance of using efficient multiobjective optimization technique for proposed type of iterative network modularization. In realistic situations proposed method involve relatively high number of optimization criteria, where concurrent optimization of all the criteria should be done. Moreover, some method for proper weighting of these criteria during learning can be useful. Discussed observations prefer evolutionary computation comparing to adaptive critic method to be used for structure optimization in this task. The multiobjective optimization is in EC area widely studied and well-known methods, like fitness sharing, are available in EC.

The time order in which criteria are satisfied can be important for overall results too. It can be assumed that development of crisp modules prior to development of strong classification (or approximation) abilities of network will lead to different modularization than reverse order. This "timing" is important as particular criteria can be weighted and optimization can be focused on different aspects of modularization during the learning. The importance of study of "timing" in modularization process is another observation from realized experiments.

Automatic modularization methods can simplify application of modular ANNs in real-world tasks. The modularization also change the "black-box" behavior of ANN to "grey-box" behavior, as the structured ANN can be better analyzed than traditional plain networks. Further, modularization can be viewed as some kind of unsupervised learning running concurrently with the main learning of network. This unsupervised learning is based on idea of expectation of modular structure of knowledge and it results in such a modular structure. Modularization of structure is an alternative method of the ANN simplification, compare to elimination of links and nodes from ANN. In this sense, modularization can be viewed as an alternative method for improvement of generalization performance of ANNs.

4 Conclusion

An method for automatic modularization of neural networks was proposed in paper. The idea of "soft module" in contrast to strictly independent modules was introduced. This idea allowed us to treat modularization as the optimization task, where optimization is optimization of weights. The optimization criteria for this task of modularization was defined and selection of proper optimization method was discussed. Realized experiments emphasized strong multiobjective character of task and we recommend to focus on this aspect of method in future work. We assume automatic modularization of ANNs an important technique for study of generalization and analysis of function of ANNs as well as for further improvement of ANNs performance.

References:[1]

[1] H. Takagi, N. Suzuki, T. Koda, Y. Kojima, Neural Networks Designed on Approximate Reasoning Architecture and Their Applications, *IEEE Transactions on Neural Networks*, Vol.3, No.5, September 1992, pp. 752-760.

[2] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive Mixtures of Local Experts, *Neural Computation*, No.3, 1991, pp. 79-87.*

[3] M. I. Jordan, R. A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, *Neural Computation*, No.6, 1994, pp. 181-214.*

[4] S. Mancoridis, B. S. Mitchell, C. Rorres, Y. Chen, E. R. Gansner, Using Automatic Clustering to Produce High-Level System Organizations of Source Code, *IEEE Proceedings of the 1998 Int. Workshop on Program Understanding (IWPC'98)*, 1998.*

[5] R. Aharonov-Barki, I. Meilijson, E. Ruppin, Who Does What? A Novel Algorithm to Determine Function Localization, *NIPS'2000*, Vol.13, 2000.*

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning (An Introduction)*, MIT Press, 1998.*

[7] Q. Xiong, K. Hirasawa, J. Hu, J. Murata, A study of Brain-like Neural Network Model, *Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies (KES'2001)*, 2001, pp. 303-307.

[1] Items marked by * are available on-line on internet.