University of Salzburg
Institute of Computer Science

SatMAS - Input pattern selection in an MAS

Thomas Aschauer
Thorsten Engl

Date: March 28, 2001
Version: 1.0

Contents

List of Figures

Chapter 1

Introduction

This chapter gives an overview about motivation for input pattern selection in an MAS. Additionally we give an overview of already existing methods of resolution.

1.1 Introduction to Artificial Neural Networks

1.1.1 Introduction to Artificial Intelligence

Science arises from the very human desire to understand and control the world. The invention of electronic computers greatly enhanced the human ability to model the scientific and technical problems into programs, and to solve them in an astonishingly fast way. Traditionally, problems were encoded into programs in the following steps:

1. Scientists research and understand a scientific model
2. Reduce this model into a few formula or rules
3. Code these formulas and rules into computer readable languages

Then after the program was developed, it could be used with different input data to solve a particular kind of problem. In analyzing this problem-solving model, we could figure out two presumptions: first, people should have full knowledge of the steps to solve the problem; second, these steps could be encoded into computer understandable languages. And in this traditional model, programs and computers had no intelligence of their own; they only conducted what programmers told them to do. But as people's ability to understand the world increased, they found the two presumptions were hardly true for many complicated problems, which were too complex to be reduced into a serial of predefined steps, and too difficult to program by hand.

As biological science was highly developed, people found out that each biological unit was just a physical unit that followed a set of very simple rules, but it could solve a large amount of complex problems, even problems they had never met before. So why not let electronic computers mimic those biological units, and while programmed to follow a set of simple pre-programmed rules, to solve complex problem in an innovative way. In this sense, let computers became "intelligent". So inspired by the biological evolution, there emerged the new technology of artificial intelligence.

1.1.2 What is an ANN?

An ANN[1] is an information-processing system that is based on generalizations of human cognition or neural biology.

The key features of NN consists of (taken from [1]):

- Information processing occurs at many simple elements called neurons.
- Signals are passed between neurons over connection links.
- Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
- Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A neural network (NN) is characterized by its particular:

- Architecture; its pattern of connections between the neurons.
- Learning Algorithm; its method of determining the weights on the connections.
- Activation function; which determines its output.

1.2 Introduction to Multi Agent Systems

MAS[2] are computational systems in which two or more agents interact or work together to perform some set of tasks or to satisfy some set of goals. These systems may be comprised of homogeneous or heterogeneous agents. An agent in the system is considered a locus of problem-solving activity, it operates asynchronously with respect to other agents, and it has a certain level of autonomy. Agent autonomy relates to an agents ability to make its own decisions about what activities to do, when to do them, what type of information should be communicated and to whom, and how to assimilate the information received.

Characteristics of Agents:

Autonomy An agent works independent from his host.

Single minded The agent performs a clear predefined job, his behavior is corresponded to this task.

Reactive The agents program run is event driven.

Environment dependent Interfaces to his environment and available resources determine his activities.

Permanent State information stay for the whole program run.

[1] Acronym for Artificial Neural Network. Short form: NN

[2] Acronym for Multi Agent System

Additionally, Agents may have some optional characteristics:

Interactive Communication between different agents is possible.

Mobile The agent moves between different hosts.

Adaptive He automatically adjusts himself to changed boundary conditions.

1.2.1 Mixture of Experts

A mixture of experts is a probabilistic model that can be interpreted as a mixture model for estimating conditional probability distributions. The model consists of a gating network that divides the problem into smaller problems and makes expert networks specialize on each of these sub problems. In terms of a mixture model the expert networks correspond to conditional component densities and a gating network to input dependent mixture coefficients.
Note that, the gating network splits the data in a "soft" way, allowing several experts to be selected at a time.

Since the gating networks deals with the decomposition in smaller tasks the choice of the type of gating network is an important one.

There a three well known types for gating networks:

- Single Layer Perceptron with a soft-max activation function (standard mixture of experts model)
- Multi Layer Perceptron with a soft-max output activation function (also known as gated experts)
- Using Gaussian kernels to divi de the input space with soft hyper-ellipsoids

For complex compositions sometimes a hierarchical mixture of experts is used. This mixture has a tree structure, where the leaves contain the expert networks and the non-terminal nodes contain the gating networks.

A short overview of these gating network types is given in [2]

1.2.2 Cooperation in MAS

One of the key problems in cooperative MAS is how to get agents to cooperate effectively [3]. The need to interact in such systems occurs because agents solve sub-problems that are interdependent, either through contention for resources or through relationships among the sub-problems. These relationships arise from two basic situations related to the natural decomposition of domain problem solving into sub problems.

- The first situation is where the subproblems are the same or overlapping, but different agents have either alternative methods or data that can be used to generate a solution.
  For example, in a distributed situation assessment application, overlapping subproblems occur when different agents are interpreting data from different sensors (independent information sources) that have overlapping sensor regions (cover similar information).

- Another form of interdependence occurs when two sub-problems are part of a larger problem in which a solution to the larger problem requires that certain constraints exist among the solutions to its subproblems.
  For example, in a distributed expert system application involving the design of an artifact where each agent is responsible for the design of a different component (subproblem), there are constraints among these subproblems that must be adhered to if the individual component designs will mesh together into an acceptable overall design.
  Depending upon the character of subproblem interdependencies, the interactions among agents in a MAS can be complex, often requiring a multistep dialogue similar to an asynchronous co-routine type of inter-action.

Knowledge Query and Manipulation Language

KQML [3] is a standardised language for inter agent communication. In KQML messages are called Performatives. The syntax of these Performatives follows the Common Lisp Polish Prefix Notation.

Blackboard Architecture

The blackboard architecture is a design pattern that supports system where nondeterministic solving strategies are used [4].

In many cases there are no known strategies how agents bring their results together. This is where the blackboard architecture takes place.

There will be a "blackboard", data storage, where all elements of the solution space and corresponding control information are hold.

Each agent sends its output to the blackboard and a central control component decides if the agents solution is plausible. Agents have the ability to use already existing solutions from other agents to establish a new hypothesis. The control component may reject an existing hypothesis or declare it as the final solution.

1.3 Motivation for Input Pattern Selection

When training an ANN we often face the problem of huge amounts of possible training data. This would result in extreme time consuming training cycles if all available data is used for teaching the network. So it's important to select only the essential part of the training pattern and keep the training data set as small as possible.

On the other hand, minimization of generalization error has also to be guaranteed, in order to ensure that the trained network can properly yield an optimal result. The selection of training data presented to the neural network influences whether or not the network learns a particular task. Like a child, how well a network will learn depends on the examples presented. A good set of examples, which illustrate the tasks to be learned well, is necessary for the desired learning to take place. The set of training examples must also reflect the variability in the patterns that the network will encounter after training. At first glance this appears to be a contradictory pair of objectives. However,

[3] Acronym for Knowledge Query and Manipulation Language

just as the generalization error must be as low as possible, data sampling which involves both collection and measurement of data is expensive and therefore needs to be reduced to a minimum. There are several methods of resolution for selecting a well fitting subset of the original (in most cases very huge) data set. However, the problem of selecting the optimal training set has not yet been solved.

1.4 Existing Methods of Input Pattern Selection

- Active Learning
  - Active Selection
  - Active Sampling
- Dynamic Pattern Selection
- Training Data Selection with Genetic Algorithms
- Training Data Selection with Multi Layer Neural Networks

1.4.1 Active Selection

The starting point for active learning is the observation that the traditional approach of randomly selecting training samples leads to large, highly redundant training sets. Such training sets can be obtained if the learner is enabled to select those training data that he/she expects to be most informative. In this case, the learner is no longer a passive recipient of information but takes an active role in the selection of the training data.

A deeper introduction to active learning can be achieved in [5].

1.4.2 Active Sampling

Recent research has shown active learning methods to be effective in increasing the modeling reliability of a neural network system. An active learning agent has the ability to query its environment in order to make a selection of its training data. One approach to the implementation of active leaning is to use querying-by-committee. This results in considerably reduced data collection and at the same time does not compromise the accuracy of identification. A nonlinear plant with both clean and noisy data is successfully modeled by such a technique and a feed forward neural network controller based upon such a model is demonstrated to perform effectively.

Minimized Data Collection - Active Querying Example

This is a data gathering method based on active querying. In this method data is reduced to a minimum, yet modeling accuracy is not compromised. The active querying criterion is determined by whether or not several neural network models agree when they are fitted to random sub samples of a small amount of collected data.

For details see [6].

1.4.3 Dynamic Pattern Selection

In contrast to active pattern selection, the dynamic pattern selection algorithm achieves concise training sets by continually validating the generalization properties of the net.

Details of this method can be found in [7].

1.4.4 Training Data Selection with Genetic Algorithms

In this method a genetic algorithm is employed for the parallel selection of appropriate input pattern for the training data set.

For an example see [8].

1.4.5 Training Data Selection with Multi Layer Neural Networks

This method selects a small number of training data, which guarantee both generalization and fast training of the MLNNs applied to pattern classification. The generalization will be satisfied using the data locate close to the boundary if the pattern classes. However, if these data are only used in the training, convergence is slow. Therefore the MLNN is first trained using some number of the data, which are randomly selected (Step 1). The data, for which the output error is relatively large, are selected. Furthermore, they are paired with the nearest data belong to the different class. The newly selected data are further paired with the nearest data. Finally, pairs of data, which locate close to the boundary, can be found. Using these pairs of the data, the MLNNs are further trained(Step 2). Since, there are some variations to combine Steps 1 and 2, the proposed method can be applied to both off-line and on-line training. The proposed method can reduce the number of the training data, at the same time, can hasten the training.

A detailed description can be found at [9].

Appendix A

Reports

This chapter contains our reports and results of the week.

Bibliography

[1] Doron Shalvi. An introduction to artificial neural networks, 1997. http://www.enee.umd.edu/medlab/papers/nnsumm/nnsumm.html.

[2] Perry Moerland. Classification using localized mixtures of experts. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'99)*, volume 2, pages 838–843. London: IEE, 1999. (IDIAP-RR 98-14).

[3] Victor R. Lesser. Cooperative multiagent systems: A personal view of the state of the art. In *IEEE Transaction on Knowledge and Data Engineering*, volume 11, No. 1 Jan/Feb 1999, pages 133–142, 1999.

[4] Frank Buschmann; Regine Meunier; Hans Rohnert; Peter Sommerlad; Michael Stal. *Pattern-orientierte Software-Architektur. Ein Pattern-System.* Addison-Wesley, 1st edition, 1998. ISBN 3-8273-1282-5.

[5] H. Ritter M. Hasenjäger. Active learning in neural networks. Technical report, Technische Fakultät, Universität Bielefeld, Germany, 1998.

[6] Leonard G.C. Hamey Tirthankar RayChaudhuri. Accurate modelling with minimised data collection – an active learning algorithm. Technical report, School of MPCE, Macquarie University, New South Wales 2109, Australia, 1995.

[7] Axel Röbel. The dynamic pattern selection algorithm: Effective training and controlled generalization of backpropagation neural networks. Technical report, TU Berlin, Germany, 1998.

[8] Helmut Mayer Roland Schwaiger. Is evolution a good teacher? genetic algorithms create training data sets for artificial neural networks. Technical report, Department of Computer Science, University of Salzburg, A-5020 Salzburg, Austria, 1998.

[9] Kenji NAKAYAMA Kazuyuki HARA. Training data selection method for generalization by multilayer neural networks. Technical report, 1998.