THE UNIVERSITY
OF BIRMINGHAM

Late Breaking Papers at EuroGP'98: the First European Workshop on Genetic Programming (Paris, 14-15 April 1998)

edited by Riccardo Poli, W B Langdon, Marc Schoenauer, Terry Fogarty and Wolfgang Banzhaf

CSRP-98-10
April 1998

School of Computer Science
Research Reports

Late Breaking Papers at EuroGP'98:
the First European Workshop on
Genetic Programming (Paris, 14-15 April 1998)

edited by Riccardo Poli, W B Langdon, Marc Schoenauer,
Terry Fogarty and Wolfgang Banzhaf

CSRP-98-10
April 1998

School of Computer Science
and
Cognitive Science Research Centre
University of Birmingham
Birmingham B15 2TT
United Kingdom

PREFACE

This booklet contains the late-breaking papers of the First European Workshop on Genetic Programming (EuroGP'98) held in Paris on April 14–15 1998. The purpose of the late-breaking papers was to provide attendees with information about research that was initiated, enhanced, improved, or completed after the original paper submission deadline in December 1997.

To ensure coverage of the most up-to-date research, the deadline for submission was set only a month before the workshop. Late-breaking papers were examined for relevance and quality by the organisers of the EuroGP'98 and one of the other members of the programme committee (Bill Langdon), but no formal review process took place.

The 7 late-breaking papers in this booklet (which was distributed at the workshop) were presented during a poster session held on the evening of Wednesday 15 April 1998 during EuroGP'98.

Authors individually retain copyright (and all other rights) to their late-breaking papers.

This booklet is available as technical report CSRP-98-10 from the School of Computer Science, The University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK.

April 1998

Riccardo Poli
W. B. Langdon
Marc Schoenauer
Terry Fogarty
Wolfgang Banzhaf

TABLE OF CONTENTS

Constructive Learning with Genetic Programming

K. Govinda Char
Department of Electronics and Electrical Engineering
E-mail: kchar@elec.gla.ac.uk

Abstract

This paper investigates the role of genetic programming (GP) as a meta-learning paradigm for connectionist networks. Through a novel approach it is shown how the flexibility in network architecture and learning can be achieved by providing a very general definition for learning and by imposing a single potential constraint within the representation that GP employs.

Keywords: GP, connectionist networks, micro-macro dynamics, learning, emergence.

Introduction

Connectionist networks, despite being powerful paradigms for learning, suffer from a number of limitations that stem from rigidity in terms of fixing the type of network architecture, the type of node activation function and the type of learning for a given problem, a priori, by the designer. The way in which a learning algorithm is implemented for an assumed architecture adds to the above limitations. In recent years, evolutionary algorithms have been successfully employed in evolving flexible architectures and learning mechanisms for known network types (Chalmers, 1990; Dasdan and Oflazar, 1994; Radi and Poly, 1998). If the aim is to achieve real flexibility the architecture and the learning should be allowed to evolve *during* the process of problem solving, that is while interacting with the task environment. The proposed approach, by naturally combining genetic programming (Koza 1994) and connectionist networks, demonstrates how potential learning rules can be evolved dynamically by providing a general definition for learning and by imposing a potential constraint in the representational structure, that is the genotype that GP employs. It seems that connectionism itself has to be approached from a different perspective if one has to realize its true potentialities.

The Framework

The simulations employ a feature map (Kohonen 1989) as a framework to illustrate the evolution of learning. The map essentially consists of a number of cells (in a competitive layer) competing for a particular signal component from a given input signal space. The *winner* cell in the network is determined according to the minimum value of the Euclidean distance $\| x - w_n \|$, where x and w_n are the input and the reference vectors respectively. The learning rule, typically employs an external supervisor to find the winner and adapts its weights for maximal response. The result of the training is described as a process of *self-organization* that is capable of enforcing a topological ordering. The network learning rule is expressed as:

$$w_n(t+1) = w_n(t) + \in(t)\, g(n-n_0, t)\, (x - w_n(t)) \;\forall\, n \qquad (1)$$

where n and n_0 represent a cell (in general) and the winner cell respectively in the network. The parameter $\varepsilon(t)$ is the learning rate. The term $g(n- n_0)$ usually represents a Gaussian function and is expressed as:

$$g(n - n_0) = exp(-p) \text{ where } p = \| n - n_0 \|^2 / 2\Delta^2 \qquad (2)$$

This function g has the effect of inducing a lateral-inhibition among the cells and is essential for the success of the algorithm. This function has its maximum size (normalized to unity) when n coincides with n_0 and it decays to zero at larger distances. The steepness of the decay is characterized by the width parameter $\Delta(t)$. The variance $\Delta^2/2$ controls the radius of the group of cells that are adapted. Thus the winner cell in the network is maximally adapted and the surrounding cells are adapted to a lesser extent depending on the distance $\| n- n_0 \|$. The learning rate $\varepsilon(t)$ and $\Delta(t)$ are initially large but reduce monotonically as the learning progresses. Kohonen made certain approximations in generalizing the above rule. Firstly, the total synaptic strength per cell is constant and is the same for every cell. Secondly, all the input signal vectors have the same intensity. Thirdly, the sigmoid function is approximates to a step function. With these approximations every cell can have only one of two states, a zero or a one.

The Problem

The sample vectors are drawn from a two-dimensional signal space with real-valued components, taking on a value in a subspace $V \in \Re^n$ with an unknown probability distribution. For the simulations, the sensory input stimuli are provided by a vector (x,y) with components distributed in a chosen subset of a square $[-1,+1]^2$. The learning rule mainly consists of the following steps: apply exemplars from the given input signal space for a number of epochs; find the *winning* cell and adapt the network weights according to the equations (1) and (2).

The learning rule, in essence, moves the two-dimensional reference vector associating each of the cells towards the two-dimensional input signal so as to minimize the *quantization error*. The quantization error for a given input signal is the distance between the signal and the reference vector of the *winning* cell over a number of epochs and is defined in terms of error as:

$$Error = \sum ABS(x- wix_{win}) + ABS(y- wiy_{win}) \qquad (3)$$

where x, y and wix_{win}, wiy_{win} are the components of the signal and the reference vectors respectively. The quantization error is finally defined as:

The *quantization error* = *Error*/(number of cells) (4)

The training enforces a topological ordering where adjacent vectors in $\Re^n$ are mapped on adjacent (or identical) cells in the competitive layer.

Whether GP is able to evolve the variety of concepts and sequence them appropriately to yield a Kohonen type of learning is to be investigated.

The GP approach

The *key* aspects of the simulation include

- providing a general definition for a connectionist learning rule as *a sequence of interacting concepts.*
- imposing a *single potential constraint* that the network weight adaptation should be an *integral* part of the representational structure, that is the genotype that the GP employs. In this context, the weight adaptation is seen as a symbolic concept, the adaptation process itself being subsymbolic.
- employing a *potential strategy* such as *micro-macro dynamics* that enables GP to realize the notion of *emergence* through its primitives. GP's primitives are the micro concepts that should enable it to form macro concepts.

Under these assumptions, GP is required to evolve the concept of a *winning* cell, induce the appropriate direction of weight adaptation for the given signal components and evolve a neighbourhood strategy such as a Gaussian to adapt this cell maximally compared with the rest of the cells in the network. Further, the concepts need to be appropriately sequenced. The simulations initially use a network that has a fixed number of cells to investigate whether GP is capable of evolving any valid learning mechanism.Two possible approaches, the general and the modular approach (Govinda Char, 1998) have been attempted. The general approach is not efficient as it leads to an extremely large search space and also yields learning rules that are difficult to interpret. The modular approach is quite effective and will be considered.

Advantages of modularity

As a meta-learning system, GP seems to be more powerful with the modular approach using ADFs due to the following reasons.

1. Given the general definition for a learning rule as a sequence of interacting concepts, it is possible to modularise each of the macro concepts and make them interact through ADFs. For instance, one of the ADFs can be employed to evolve the concept of the *winner* while another can adapt the network weights. GP's primitives as micro-concepts form macro-concepts that, in turn, can be represented in terms of ADFS.
2. The approach enables the tractability and interpretability of the rules that evolve. The formation of concepts and their sequencing can be interpreted easily with the GP hierarchy.
3. As an expression with ADFs can have a number of value-returning branches, each of the ADFs can be assigned an *explicit* fitness function if needed.
4. The weight adaptation is an integral part of the representational structure and can be achieved through ADFs.
5. The search space becomes more focused (towards the regions of potential concepts) through the use of automatically defined functions enabling the evolution of valid learning mechanisms. This is vital if one expects to achieve a good performance in a reasonable amount of time with the evolutionary paradigm.
6. Co-evolution of neural network structures along with the learning is essential for a variety of problem environments and is possible with a grammar such as the cellular encoding (CE) (Gruau, 1994) that is compatible with genetic programming. The primitives for the grammar can be implemeted through ADFs. The advantage of such an approach is that GP can induce the type of network for particular type of inputs/signals including the temporal signals. Also, applications that incorporate different types of architectures and learning at various hierarchical levels are feasible with the proposed method.