Adaptive Learning of Polynomial Networks Genetic Programming, Backpropagation and Bayesian Methods

Springer Science. Series on Genetic and Evolutionary Computation, Vol. XIV, Nikolay Nikolaev and Hitoshi Iba. 2006. Hardcover, 316 p. ISBN 0-387-31239-0.

Diego H. Milone

If you have a modeling paradigm and a problem to solve, the rationale behind the Nikolaev and Iba's book could be seen as:

1. Arrange your model structure like a tree and use all GP's power to search in the structure space.

2. Take the tree evolved by GP, rearrange it as a multi-layered artificial neural network (NN) and use all the NN machinery to fine tune it.

3. Use statistical learning approaches to get the maximum advantage from your training data and thereby enhance your model's generalization.

Nikolaev and Iba use polynomials to predict time series. Their Polynomial Neural Networks (PNN) are polynomials arranged like NN, with 1st and 2nd degree polynomials as activation functions in the neurons, and the polynomial coefficients encoded in the synaptic weights.

Following the structure above, the book is divided into three parts, plus a final chapter with some comparative examples on times-series modelling. I read the first part comfortably: Chapters 2 to 5 are a gentle review of many interesting topics in GP, very useful for model structure searching in machine learning. All the ideas are gradually and clearly presented. The examples are simple and illustrated by figures. The algorithms are summarised in tables. Each equation is well described, with a detailed explanation of its terms, the role of the variables and the importance of the result. I enjoyed reading these chapters. Moreover, I think this is a readable material for undergraduate students with minimal knowledge of GP.

Chapter 2 introduces Inductive Genetic Programming (IGP) as a specialization of the traditional GP for the identification of the model that optimally describes the provided data. IGP discovers not only the network architecture but also provides a first approximation of the NN weights. Chapter 3 presents the first key in the rationale behind the book: how to use tree representations to evolve PNN. Then, in Chapter 4, an interesting review of fitness function design for regression problems is given. Chapter 4 also summarizes many useful tools for the analysis of the fitness landscape. Closing this first part, Chapter 5 deals with the genetic operators used to guide the evolutionary search and presents useful measures for the examination of evolutionary search performance.

In the second part of the book all the machinery of NN is adapted for tuning polynomial coefficients in static and dynamic PNN models. These chapters become gradually harder, mainly because of the increasing amount of background knowledge needed by the reader. Not only is the mathematical treatment and the explanations more advanced but the examples and the interpretation of final results is reduced. The main ideas can be easily followed by the specialist but the details behind these algorithms and their adaptations to PNN require careful reading.

Chapter 6 is dedicated to the variants of the backpropagation algorithm used for training different architectures of feedforward PNN. It includes first and second order developments of the weight adaptation rules and some techniques for network pruning. Chapter 7 considers recurrent neural networks. These allow connections to previous layers as well as the next layer. Two types of training algorithms: Backpropagation Through Time and Real-Time Recurrent Learning are considered and extended for PNN. The connection with IGP is not well described and so it is not clear how the structure of the recurrences is evolved.

In a similar way to the second part, the next two chapters require the reader to have some background in statistical learning. They are not suitable for beginners. In Chapter 8, the authors revised an important number of Bayesian techniques for learning with PNN. These techniques are mainly rooted in two approaches: the evidence framework and Monte Carlo sampling of the weights. Chapter 9 provides statistical means for quantify the reliability of the learned PNN. As always it deals only with time-series modelling with PNN but gives useful mechanisms for model selection and overfitting prevention and several methods to estimate the error on PNN's predictions.

In the last experimental chapter (Chapter 10) the authors present six examples of the use of PNN. For each one, the mean squared error is used to compare PNN to other classical methods. Some results are very close and the authors should have included the results of statistical significance tests. Graphs of the original, approximated and predicted time series are given, but only for PNN and not for the other methods. The results suggest that PNN models are better than, for example, ARMA, MLP and the Elman recurrent architecture. But the main advantage of PNN is its use of the powerful machinery of IGP to select the model structure. However, it would be interesting to carry out exhaustive tests of all the proposed algorithms, both between themselves and versus other state of the art technologies. The comparison should include their computational costs.

Overall, "Adaptive Learning of Polynomial Networks" is mainly of interest to time-series modellers and advanced researchers on PNN models. Undergraduate students will probably find some chapters hard and that there is not enough on the fundamentals of GP, backpropagation or probabilistic learning. Furthermore, it does not contain exercises and I have not been able to find direct access to source codes. However, graduate students will find it both a reference book on probabilistic learning and a very gentle and useful review of interesting ideas on the application of GP to searching for machine learning architectures.