How Long does it Take to Evolve a Neural Net?

Marc Schoenauer and Edmund Ronald

CMAP, CNRS-URA 756, Ecole Polytechnique, 91128 Palaiseau, France

{Marc.Schoenauer,Edmund.Ronald}@polytechnique.fr

Abstract. This paper deals with technical issues relevant to artificial neural net (ANN) training by genetic algorithms. Neural nets have applications ranging from perception to control; in the context of control, achieving great precision is more critical than in pattern recognition or classification tasks. In previous work, the authors have found that when employing genetic search to train a net, both precision and training speed can be greatly enhanced by an input renormalization technique. In this paper we investigate the automatic tuning of such renormalization coefficients, as well as the tuning of the slopes of the transfer functions of the individual neurons in the net. Waiting time analysis is presented as an alternative to the classical "mean performance" interpretation of GA experiments. It is felt that it provides a more realistic evaluation of the real-world usefulness of a GA.

1 The Usefulness of Automatic Parameter Tuning

The operator of a heuristic program spends a lot of time predicting what his program will do. Now and then a test run actually validates the programmer’s prediction and life, science, and everything is wonderful – yet more often, the program goes off to do its own thing and the programmer is left to scratch his head.

In this context, manual tuning of parameters is one of the least rewarding facets of heuristic programming. For instance, the authors have spent hours in front of computer screens, when investigating neural net training. These hours were occupied by running the same program time and again with just the change of a single real parameter like the slope of the neural transfer function.

Computer programming is of necessity experimental; however every worker in the field of genetic algorithms has been brought to conjecture that the experimenter’s action could be automated. In the case of GA research, the manual variation of GA parameters – e.g. search for good mutation or crossover rates – could be replaced by the action of a meta-GA. Unfortunately, the computational expense of running a population of GA’s in parallel usually discourages the GA experimenter from pursuing such a course of automatic experimentation, although object-oriented software engineering makes a 2-level GA easily feasible. Indeed, the possibility of running a meta-GA is a design goal of the authors’ next
although object-oriented software engineering makes a 2-level GA easily feasible. Indeed, the possibility of running a meta-GA is a design goal of the authors’ next generation GA software.

In the special case of the authors’ research in neural net control, some of the parameters originally subject to tuning could be varied by the same GA employed to train the net by searching the space of weight matrices. In this document the results of this automated research is confronted with the results originally published in [Ronald & Schoenauer 93].

In summary, by confronting the results of our previous work with this automated tuning by GA, we show that the GA improves on a human operator in tuning some of the net parameters, namely the transfer functions. On the other hand, automatic renormalization, as presented in section 5 does not improve mean precision, and indeed produces barely acceptable results; but it holds the promise of obsoleting the painful data pre-processing steps which hinder real-world and industrial applications of neural nets.

We have included a waiting-time interpretation of our results; we believe that this interpretation methodology, while unusual, is more appropriate in an industrial context than mean performance: GAs are stochastic, and estimations of their performance must perforce be formulated in probabilistic terms. A waiting time estimation provides us with a confidence factor whereby a control problem can be solved with a predetermined amount of computation.

The plan of the paper follows: In section 2 below we recall the neural net formalism and its application to control, and summarize earlier work. Section 3 presents an investigation into the automatic tuning of the neural transfer function slopes. Section 4 introduces the waiting time analysis, and applies it to the results of section 3. Section 5 describes experiments in generalized tuning of both data renormalization coefficients and transfer functions. Section 6 discusses the significance of the results, and highlights some possible applications.

2 Genetic Training of Neural Net Controllers

In order to make this paper self-contained, this section summarizes the methods employed by the authors for neural net control in [Ronald & Schoenauer 93], [Schoenauer & Ronald 94]. This establishes the background for the numerical experiments on the lunar lander simulator, whose results form the body of sections 3 and 4 below. The reader desiring an overview of the field may refer to [Yao 93], a broad survey of evolutionary methods as applied to neural net training and design.