

# Evolving Features in Neural Networks for System Identification

Yung-Keun Kwon and Byung-Ro Moon

School of Computer Science & Engineering, Seoul National University  
Shilim-dong, Kwanak-gu, Seoul, 151-742 Korea  
{kwon, moon}@soar.snu.ac.kr

Given  $N$  data pairs  $\{X_i, y_i\}$ ,  $i = 1, 2, \dots, N$ , where each  $X_i$  is an  $n$ -dimensional vector of independent variables ( $X_i = \langle x_{i_1}, x_{i_2}, \dots, x_{i_n} \rangle$ ) and  $y_i$  is a dependent variable, the function approximation problem (FAP) is finding a function that best explains the  $N$  pairs of  $X_i$  and  $y_i$ . From the universal approximation theorem and inherent approximation capabilities proved by various researches, artificial neural networks (ANNs) are considered as powerful function approximators. There are two main issues on the feedforward neural networks' performance. One is to determine its structure. The other issue is to specify the weights of a network that minimizes its error. Genetic algorithm (GA) is a global search technique and is useful for complex optimization problems. So, it has been considered to have potential to reinforce the performance of neural networks. Many researchers tried to optimize the weights of networks using genetic algorithms alone or combined with the backpropagation algorithm. Others also tried to find a good topology that is even more difficult and called a "black art."

In this paper, instead, we use a genetic algorithm to evolve the input space. That is, a chromosome represents the meanings of input nodes. It generates a new input space using hidden neurons that play a critical role in the learning because they act as *feature detectors*. They gradually discover the salient features that characterize the training data. We try to find useful input features using a genetic algorithm. A chromosome represents the meaning of the input nodes. A chromosome consists of an array of  $I$  functions.

Roulette-wheel selection is used for parent selection. A crossover operator creates a new offspring chromosome by choosing some features from two parent chromosomes. Since each neural network has  $I$  input nodes and  $H$  hidden nodes, there are totally  $2(I+H)$  candidate features to be chosen for offspring. Crossover randomly chooses  $I$  features from the set of candidates. This process is shown in Figure 1. The neural network then takes the features in the offspring as the input nodes. The offspring first attempts to replace the inferior out of the two parents. If it fails, it attempts to replace the most inferior member of the population. It stops when there is no improvement during a given number of generations.

We attack a critical heat flux (CHF) function approximation problem which is important for the safe and economic design of many heat transfer units including nuclear reactors, fossil-fuel boilers, fusion reactors, electronic chips, etc. Each data set consists of eight independent variables and one dependent variable. We were given 1607 sets of observed data from KAERI (Korea Atomic Energy

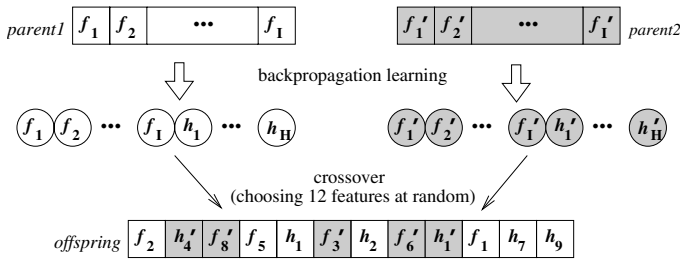


Fig. 1. Crossover process

Table 1. Performance Comparison in Terms of *LRE*

|                        | KAERI    | PLS      | WGA      | FEGA     |
|------------------------|----------|----------|----------|----------|
| $E(y/\hat{f}(X))$      | 1.002671 | 1.002562 | 1.006147 | 1.010703 |
| $\sigma(y/\hat{f}(X))$ | 0.107265 | 0.121299 | 0.098247 | 0.094624 |
| <i>LRE</i>             | 0.106979 | 0.120989 | 0.097647 | 0.093622 |

Research Institute). To evaluate the performance, we use *LRE* (Least Ratio Error) following the convention of the CHF studies:  $LRE(\hat{f}) = \sigma(\frac{y}{\hat{f}(X)})/E(\frac{y}{\hat{f}(X)})$ , where  $y$  and  $\hat{f}$  are the observed output and the estimated output, respectively.

The 10 fold cross-validation estimate of the average and the standard deviations of the observed CHF value over the predicted value are shown in Table 1. Our approach, feature-evolving genetic algorithm (FEGA), is compared with four other results. In the table, KAERI means the function which has been used at KAERI for years; PLS means a partial least squares that is a representative statistical method for soft modeling. WGA is a hybrid genetic algorithm combined with feedforward neural networks to optimize the weights. The results of WGA and FEGA are the average from 20 trials and they took almost the same running time. WGA and FEGA have the same network architecture: 12 input nodes, 6 hidden nodes, and one output node. WGA showed better performance than the existing approaches. FEGA outperformed WGA by about 4.3%. FEGA found a CHF function better than that of KAERI by about 14.3%.

**Acknowledgment.** This work was supported by grant No. (R01-2003-000-10879-0) from the Basic Research Program of the Korea Science and Engineering Foundation. This work was also supported by Brain Korea 21 Project. The ICT at Seoul National University provided research facilities for this study.