Tuning Function Point Analysis Model by Using Fuzzy Neural Network

HO-LEUNG TSOI

Griffith University
Nathan, Queensland, Australia

Abstract:- Estimation of the size and time required for software development is probably the most difficult task of software projects development. Functional Point Analysis (FPA) model are gaining a wide popularity for assessing software size. By the definition of Function Point Analysis model, all 14 General System Characteristics (GSC) are not totally independent. A fuzzy neural network model for tuning the GSCs is presented in this paper. This model has a distinguishing feature in that it can express complex nonlinear GSC linguistically. Using the fuzzy rules and tuning the connection weights of this model, we can find the optimalized connection weights through learning process. With these optimal connection weights, the estimator can estimating software size more accurated.

Keywords: Software Estimation, Functio Point Analysis, Neural Network

1 Introduction

Estimation of effort and the developing duration of software has become one of most important topic in software engineering. Lederer [3] claimed that software cost estimating is an important concern for software managers and other software professionals. Software development is a complex environment and the size of the application is one of the most important elements that affect software project development costs [1,8]. The project managers need the size of an application as a component of the measurement of productivity in system development and maintenance activities, and as a component of estimating the effort needed for such activities. Most current cost estimation models include project size as an important parameter or else for deriving an estimate of effort/schedule from a previous size estimate. The estimation result will be used as a technical productivity and performance indicators that facilitate the tracking and control the software development period. If the software is over estimation, the organisation will waste a lot of money. On the other hand, under-estimation will lead insufficient money to development. This implies that either the quality of final products will be affected or the software is abandoned before completion.

Over the past twenty years, dozens of software cost-estimation methods and tools have been proposed to tackle this problem but the outcome is not very significant. The objective of the research discussed in this paper has been to make a reliable prediction of software size estimation.

2 The Neural Network Tuning Model

The fuzzy B-spline FPA model has been proposed by Yau and Tsoi in 1998. They claimed that by using the B-spline curve fitting method, different membership functions can be derived to represent the possibility distribution of the overall estimate. It can be used to describe the

probabilistic behaviour of the software size estimate on which risk control can be based [13]. The FFPA model based on the assumption that all complexity factors are independent. Actually, many researchers believe that some factors are not totally independent. It is still an open question for further investigation, but further discussion is out of the scope of this paper. This paper proposed a fuzzy neural network to tackle the relationship problem among the 14 complexity factors.

The Neural Network (NN) consists of two major sections, namely premise part and consequence part. The premise part comprises a number of cells, each of which receives a set of input values with the fuzzy linguistic process [2,5,6,9]. The B-spline membership functions (BMF's) have been adopted as fuzzy membership functions.
The input fuzzy values have been translated as the control points of B-spline curves to construct fuzzy membership functions. The consequence part comprises of a two-layer Neural Network (NN). When compared to traditional neural network methods, the NN is suitable for describing the imprecision and uncertainty while the neural network (NN) layer are capable of learning from input-output data.

Figure 1 shows the basic structure of the fuzzy neural network which represents B-spline fuzzy loic reasoning by the neural network. The NN consists of five layers. The Layer I and layer II are the premise section of the NN. There are N linguistic inputs in Layer I and form into B-spline membership functions. The consequence section consists of three layers - III and IV. Defuzzification is adopted in Layer V to find out the output of the fuzzy system.

2.1 Fuzzy Neural Network for the GSCs

Fuzzy modelling is a method based on fuzzy set theory for describing the non-linear relation between an input vector x=[x1, x2, …xn] and an output vector y=y1, y2, …, yn]. If we have some training input data Xq, q= 1,2, .. ,n, and the desired output values are Yp, p=1,2, ...,m, we can establish the inference rules of simplified reasoning by experts [10,11,12]. The ith fuzzy rule base can be formulated as follows:

R_i : IF X_1 is B_1^i and ... X_n is B_n^i
THEN Y_1 is W_1^i and ... Y_p is W_p^i

Where n = {1,2, … 14} and m = {1,2, …, 14}

X_j	is the jth input variable
Y_k	is the kth output variable
R_i	is the ith inference rule of the fuzzy system in the premise section
B_q^i	is the BMF for corresponding term, i^{th} rule in the neural network, and q is number of the previous part (number of input).
W_p^i	is the weight factors of output Y_p and i is number of the rule.

The above rules are defined in simple language terms and the fuzzy system requires N "if .. then" rules to cover all possible scenarios. The B-spline membership functions (BMFs) [4,10,11,13] is adopted as membership functions of the above fuzzy system. To construct the BMF, we need a set of empirical data which is given by experts with a primary fuzzy term. The empirical data is $D(x_j)$, (j=0,1, … ,m), and the grade of the fuzzy membership function corresponding to x_j is $\mu(x_j)$ [13]. The truth value m_i for the ith rule can be calculated as

$$B_q^i(x_q) = \sum_{j=0}^{m} p_j N_{j,2}(x_q) \qquad 1 \le k \le m, \qquad \text{(EQ 1)}$$

and

$$m_i = B_1^i(x_1) * B_2^i(x_2) * \cdots * B_n^i(x_n) = \prod_{j=1}^{n} B_j^i(x_j) \qquad \text{(EQ 2)}$$

The output of the fuzzy reasoning can be derived from the following equations:

$$Y_p = \frac{A}{B} \qquad (p = 1,2,\ldots,14) \qquad \text{(EQ 3)}$$

where

$$A = \sum_{i=0}^{x} m_i W_p^i \quad , \quad B = \sum_{i=0}^{x} m_i \qquad (p = 1,2,\ldots,14)$$

x is the max. number of rule in the fuzzy neural network

and the output is $Y = [Y_1, Y_2, \ldots, Y_{14}]$ (EQ 4)

Eqs. (1) to (3) describe the simplified fuzzy reasoning method, where we choose the B-spline membership functions as membership functions of the antecedent part. Then the fuzzy controller based on neural networks [2,4,10,11,12,13] can be established. Figure 1 shows the configuration of fuzzy neural networks. There are five different layers in this network. The layer I are nodes for each system characteristics. N different number nodes are used in layer I. Nodes at layer II are input nodes which represent the five fuzzified statements -- Almost Certain, Very Likely, Probably, Unlikely, and Extreme Unlikely for system characteristics. Nodes at layer III are the BMF's to represent the terms of the respective fuzzified statements. In this layer, the BMF can be established by one or two fuzzified statements and performs a single membership function. Nodes at Layer IV are fuzzy rules and form the fuzzy rule base layer. The links between layer III and IV are full connected by weight values, denoted as W_p^i. The output Y_z of the n input and m output fuzzy model can be defined as:

$$Y(t+1) = Y_p(t) - k \frac{\partial e_p}{\partial Y_p} \qquad (p = 1,2,\ldots,m)$$

Layer V is the output layer and define the consequences of the rule nodes from layer IV. To find out a crisp output value for the conventional FPA, the centre of gravity defuzzification method [7] is adopted.

$$\mathrm{DI}_{\text{output}} = \frac{\int m(x)dx}{\int x dx}$$

3 Function Point Analysis Tuning (FPAT) Model

J. Albrecht devised the Function Point Analysis (FPA) model in 1977 to help measure the size of a computerized business information system. FPA model consists of a two- step process to calculate the function point count. The basic function points are categorized into five groups: outputs, inquiries, inputs, files, and Interfaces which relate more closely to the functions performed by the software as compared to other measures, such as lines of code. In addition, fourteen general systems characteristics (GSCs) are used to construct a value adjustment factor (VAF) which is used to adjust the basic function point count so as to get a better prediction. FPA provides a measure of project size which is independent of language and has been proven as a reliable method for measuring the size of computer software.

Despite FPA is widely used by many practitioners, it is not free of criticisms. Most criticisms are that GSCs aren't really worth using and only some of the GSCs were relevant in the adjustment

process and modification should be applied. Several researchers have investigated the GSCs and supported that GSCs are practical benefits for estimation size but need to be reviewed and revised [14]. Yau and Tsoi [13] propose to apply the fuzzy set theory to overcome the ambiguity of experts' judgement for counting the GSC. The results may be more useful for understanding project cost drivers and for comparing similar projects. In order to study the inter-relationships among the fourteen GSCs, a Function Point Analysis Tuning Model (FPAT) has been proposed in this paper.

Using Algorithm:

1. Request two experts to evaluate the 14 system characteristics of a sample system and give some training data Xq. For this sample system, we have the desired values Y^*p and the threshold value.
2. Using the evaluation points to construct the shape of membership functions (B-Spline Membership Function -- $B^i_j(t)$) for each system characteristics.
3. Set the initial values of $W^i_p(t)$ and the learning rates η_B and η_w of the system.
4. Using the training input-output data (Xq, Y^*p) to adjust the value of $B^i_j(t)$ and $W^i_p(t+1)$ of the system.
5. For each adjustment process, a new objective function e_p is calculated. Repeat the adjustment process until $|e_p(t+1) - e_p(t)|$ is less than the desired threshold value.
6. This adjusted system (14 system characteristics) can be used to estimate another software.

Figure 2 shows the basic structure of the fuzzy neural network which represents B-spline fuzzy logic reasoning by the neural network. The FPAT consists of five layers. The Layer I and layer II are the premise section of the FPAT. There are fourteen linguistic inputs in Layer I and form into B-spline membership functions. The consequence section consists of three layers - III and IV. Defuzzification is adopted in Layer V to find out the output of the fuzzy system.

4. Conclusion

Unexpected event is a major reason that causes inaccurate result of software estimation. To minimise the errors among different complexity factors, a neural system based on the B-spline membership function have been proposed. The inter-connected module is firstly trained by the information of the input data and will be trained again by a back-propagation learning rule in second and third layers. Finally, a crisp value will be generated in layer IV.

Reference:

[1] B. Boehm, and P.N. Papacio, "Understanding the Controlling Software Costs", IEEE Trans. on Software Engineering, Vol. 14, No. 10, 1988.

[2] Horikawa, T. Furuhashi , and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neual Networks with the Back-Progation Algorithm," IEEE Trans. on Neual Networks, Vol. 3, No. 5 September 1992.

[3] Albert L. Lederer, Jayesh Prasad, "Software management and cost

estimating error", The Journal of Systems and Software, 50, 2000.

[4] L. Lin, and C. S. George Lee, "Neual-Network-Based Fuzzy Logic Control and Decision System", IEEE Trans. on Computer, Vol. 40, No. 12 December 1991.

[5] H. Nomura, I. Hayashi and N. Walami, "Self-Tuning Method of Fuzzy Control by Descent Method," Proceedings of 4th IFSA Congress, Engineering, pp. 49-52, 1991.

[6] H. Nomura, I. Hayashi and N. Walami, " A Learning Method of Fuzzy Inference Rules by Descent Method," IEEE International Conference on Fuzzy System, pp. 203-210, 1992

[7] K.M. Passion and S. Yurkovich, "Fuzzy Control" Addison-Wesley, 1998.

[8] B. Samson, D. Ellison, and P. Dugard. Software cost estimating using an Albus Perceptron (CMAC). *Information and Software Technology*, 39:55-60, 1997.

[9] T. Sugeno and M. Nishida, "Fuzzy Control of Model Car," Fuzzy Sets and Systems 16 (1985) 103-113.

[10] C.H. Wang, W.Y. Wang, T.T. Lee, and P.S. Tseng, "Fuzzy B-spline membership function (BMF) and its applications in fuzzy-neural Control", IEEE Trans. on System, Man, and Cybernetics, Vol. 25, pp. 841-851, May 1995.

[11] W.Y. Wang, T.T. Lee, C.L. Liu, and C.H. Wang, "Function Approximation Using Fuzzy Neural Networks with Robust Learning Algorithm", IEEE Trans. on System, Man, and Cybernetics, Vol. 27, No. 4, pp. 740-747, August 1997.

[12] C.H. Wang, H.L. Liu, and C.T. Lin, "Dynamic Optimal Learning Rates of a Certain Class of Fuzzy Neural Networks and its Applications with Genetic Algorithm", IEEE Trans. on System, Man, and Cybernetics, Vol. 31, No. 3, pp. 467-475, August 2001.

[13] C. Yau and H.L. Tsoi, "Modelling the probabilistic behaviour of Function Point Analysis" Information and Software Technology, 2:59-68, 1998.

[14] C. Kemerer. An Empirical Validation of Software Cost Estimation Models. Communication of the ACM, 30(5):416-429, 1987.

Figure 1. The basic structure of the Fuzzy Neural Network

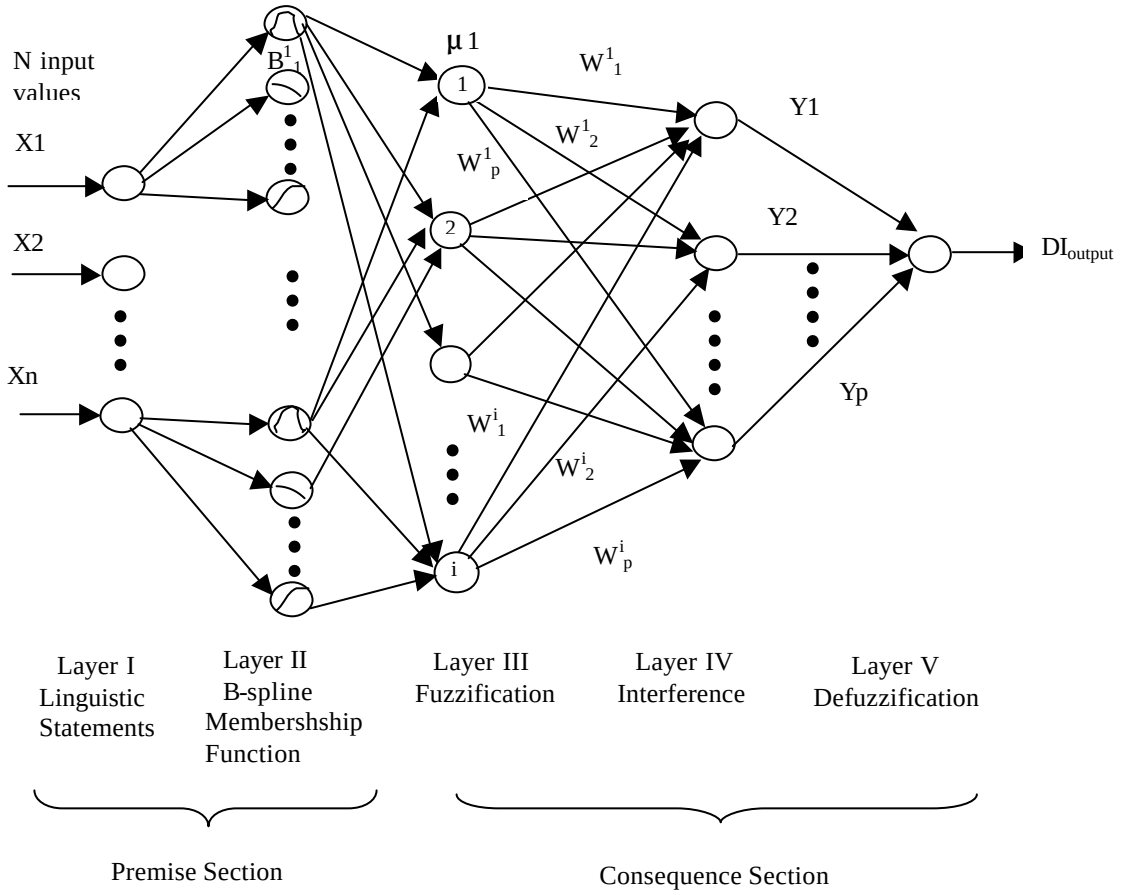

Figure 2. The FPAT Model for 14 GSCs