Fuzzy Neural Networks in a Palm Environment

Samuel Moyle and Michael Watts

Department of Information Science
University of Otago School of Business
PO Box 56
Dunedin, New Zealand
smoyle@infoscience.otago.ac.nz, mike@kel.otago.ac.nz

Abstract

This paper outlines the achievements made in the area of small expert systems, in particular the use of multiple Fuzzy Neural Networks (FuNN) within a single application implemented on a PDA. Also discussed is the opportunity for using the architecture as a generic problem solving method – if a Neural Network is an appropriate solution to a problem then a PDA based implementation becomes possible.

Introduction

This paper commences with an overview of the problem addressed. The proposed solution is outlined, as is the use of Fuzzy Neural Networks in the building of a small expert system. The generic architecture implemented, and the reasons for using such a structure, is discussed. Finally, the options for future development are outlined, as proposed by expert evaluators and system developers.*

The Problem

Determining the condition of a roof can be difficult, especially for those not having a technical background. Deciding the most appropriate course of maintenance for a roof is dependent on a number of factors, dependent upon the roof type under assessment. The combination and severity of factors is used to determine what maintenance is recommended to the property owner. Often there is not a sufficiently high correlation for any one condition to be used, making the final decision complex.

In some instances input values may be contradictory to the result expected. This complexity makes training new staff difficult, especially where the staff member has little building, technical or engineering background. It is likely that more than one result is valid and it is often difficult to establish which is to be the primary decision.

Currently, maintenance experts need to climb onto the roof to be assessed. The expert must have a folder, pen, and often a tape measure. This makes the act of climbing onto the roof hazardous, an issue identified by OSH[1] (2000). Ideally, an expert need only take a palm-sized device able to accept input. This device can be stored in a pocket until needed thus freeing hands for more important purposes (like climbing ladders).

The proposed solution

To enable novice maintenance staff to quickly become trained, it is useful for the tool to make decisions given the available inputs. Before climbing onto the roof, the expert knows:

- Where the roof is (location)
- How old the roof is (age)
- The roof profile

With this information the system should be able to 'guess' what the roof maintenance expert will see.

The expert should also have opportunity to alter the parameters to reflect what is actually found, should they be different from that expected by the system. The system should be able to take the new parameters and use them to make an assessment of preferred maintenance.

Determining Suitable Results – the underlying expert system

Having determined that a palm-sized tool is appropriate, and that the user interface can be created in a compact manner, the underlying expert system needed to be developed.

Fuzzy Neural Networks (FuNN) (Kasabov et al, 1997) are excellent tools for divining rules from a data set. As there were no initial rules, accurate training of the network was imperative. Rules were extracted and assessed by an expert to confirm suitability. Another advantage is the FuNN is capable of providing clear secondary results. Priority is established by mathematical ranking. For this work the FuNN found in FuzzyCOPE 3 as developed by the

[1] OSH is the Occupational Safety and Health division of the Department of Labour. They are charged with ensuring that New Zealand businesses comply with workplace safety standards.

University of Otago (Watts, Woodford and Kasabov, 1999) was used

An expert was engaged to determine expected input and results. The test values consisted of a set of randomly generated numbers that were assigned results by the expert. This data set (inputs and expected results) was used for training and testing of the FuNN. The total set was divided into two sets – 75% used as for training, 25% for testing.

Why a generic structure?

An important factor in this development is that the use of new technologies should simplify work done. In this instance we wish to take advantage of Palm technologies in creating a small, real-time, expert system.

Palm devices are small, not just in physical size, but processor power and screen size. Memory is a constraint, as memory code chunks are 32k in size. This means that, where possible, work is broken into small pieces and allowed to reside in separate memory areas. There is a limitation in the relative distance that memory calls can be made within dynamic memory – also 32k. Note that devices with more than 2mb RAM may be implicitly instructed to use a larger memory chunk size.

This memory allocation problem was overcome through allocating the generic work to layers - allowing calls to be made between chunks. The system developed uses 4 layers.

The visible outer layer is the User Interface (UI). Any PDA User Interface should reflect the work being done and, wherever possible, be simple, intuitive and fast.

The next layer is a translation layer. The translation layer has two tasks. The first is to convert raw values from the UI layer into numeric values appropriate for the neural network. The values returned from the FuNN are translated into sentence form, which are returned to the UI layer for display.

The next layer, the neural network layer, is responsible for taking the neural network input values, bundling them with the relevant connection weight information and passing them on to the generic neural network layer.

The generic FuNN Structure layer is based on the CBIS code-base (Ward et al, 1997), as utilised by FuzzyCOPE3. This accepts the connection weight information and input values, processes them and returns the numeric results to the neural network layer.

In the PalmOS prototype constructed, these layers are in the form of code classes placed in separate code chunks.

This widens the appeal and possible applications that can be built using neural networks. The generic use of components becomes advantageous when there are a number of small, highly specialised, applications sharing resources in a single PDA.

Further Development

After evaluation of the PalmOS prototype, experts have identified areas of development that would make the tool even more useful. These include:

- Saving data for a specific property to a database for future visits.
- Combining the existing tool with a Global Positioning System (GPS) tool. Gathering of data points should enable automatic calculation of the surface area of the roof. This would also eliminate the need for the location to be entered, resulting in better location specific training of the Neural Network.
- Linking in of costing components – automating the creation of quotes.

From a development perspective, it would be beneficial to further separate the underlying expert system from the front end. PalmOS allows this through the ability to store the underlying FuNN structure in the form of a pre-compiled library. The use of libraries enables many applications to use a single library. Code optimisation can further reduce memory use and improve application speed. Only the recall components need be included in the PDA as training can be undertaken on the desktop computer.

The connection weight files can be stored in an independent PDA database. There are a number of advantages in doing this. The desktop computer can update weight files, the Palm database updated when next connected to the desktop. Connection weight files can be updated on the palm device without having to re-compile any application(s) using them. As many connection weight files as are needed can be stored on the palm – enhancing memory efficiency.

For small applications, the translation layer may be combined in the main program component as methods or classes. In order to maintain a generic structure separate classes are used in the PalmOS prototype. Separation provides the advantage that classes can be modified, added, or removed with minimal programming effort.

References

N Kasabov, J Kim, M Watts, and A Gray, 1997. *FuNN - A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition in Multi-modular Distributed Environments,* Information Sciences – Applications, 101:3-4:155-175.

Occupational Safety and Health (OSH), 2000. Guidelines for the prevention of Falls, Department of Labour, NZ Government.

R Ward, M Purvis, R Raykov, F Zhang, and M Watts, 1997. An Architecture for Distributed Connectionist Computation, In P*rogress in Connectionist-Based Information Systems, Proceedings of the ICONIP / ANZIIS / ANNES '97*, pages 721-724, Springer Verlag, Singapore.

M Watts, B Woodford, and N Kasabov, 1999. FuzzyCOPE - A Software Environment for Building Intelligent Systems - the Past, the Present and the Future. In *Emerging Knowledge Engineering and Connectionist-based Systems, Proceedings of the ICONIP/ANZIIS/ANNES Workshop "Future directions for intelligent systems and information sciences"*, 188-192, University of Otago, New Zealand.