An introduction to neural networks and their applications in manufacturing

B. WU

Department of Manufacturing and Engineering Systems, Brunel University, Uxbridge, Middlesex UB8 3PH, UK

It is a frequently quoted fact that today’s manufacturing functions are becoming more and more ‘inter-disciplinary’, with new approaches and techniques continuously and rapidly introduced and adopted. The recent applications of neural networks in manufacturing provide a typical example of this trend. This paper examines the structures and functions of neural networks, and provides some examples of their manufacturing applications.

Keywords: artificial intelligence, neural network, error back-propagation, competitive learning

1. Introduction

The structure and function of neural networks (NN) are based on our current understanding of the biological nervous system. NNs are built on a large number of simple, and adaptable processing units (PU) which are interconnected in such a way that they can store experiential knowledge through learning from examples and, like biological systems, have the ability to take in hazy information from the outside world and process it without an explicit set of rules. This approach (parallel and distributed) is in contrast to the traditional computing approach which processes information sequentially according to a set of exact rules. Also, their structure and function provide a typical example of the applications of systems perspective concept which puts much emphasis on, in addition to the individual elements and their operations, the relationships among the elements and how they influence each other within the system (Wu, 1992).

Perhaps due to some of the difficulties that have been experienced with the traditional expert system applications, and because of the rapid development and introduction of NN system development tools, NNs have created a substantial amount of interest in the manufacturing arena, with systems and techniques being developed for organization, operational, as well as machine-level applications. This paper will examine the basic structures and functions of NNs, and provide some examples of their manufacturing applications.

2. Basic structure of NNs

2.1. Processing units

Processing units (PU) are the building bricks of NNs. They usually take the form shown in Fig. 1, and emulate (assuming) the function of a neuron in the brain. PUs are basically a logic processing device with the fundamental function of producing an output signal as a function of the sum of their weighted inputs, and a certain threshold value. Mathematically, this is expressed as:

\[ y_i = f_i \left( \sum_{j=1}^{n} w_{ij} x_j - s_i \right) \]

where \( y_i \) represents the output signal from PU\(_i\), \( w_{ij} \) represents the weight of the \( j \) to \( i \) interconnection, \( s_i \) represents the threshold value of PU\(_i\).

The input to a PU\(_i\) can be either the output from other PUs, or directly from outside the NN, i.e. input to the NN. The output from PU\(_i\) can be used either as input to the subsequent PUs, or as an output from the NN.

The value of \( w_{ij} \) determines how strongly the output of PU\(_j\) influences the activity of PU\(_i\). The magnitude of a weight can be changed over time. During a training operation, it is mainly through this mechanism that the PU is made adaptive to new information put to it, and the learning process accomplished. As will become clear
later, the total weight matrix $W$ of a NN encompasses and reflects the NN's knowledge and skills that it has learnt through previous training, and is therefore referred to as its long-term memory.

### 2.2. Activation functions

The threshold $s_i$ acts as a filter for incoming signals. The term inside the brackets in Equation 1, $a_i$, is known as the activation of $PU_i$ which provides temporary and local information around it. This is therefore referred to as the $PU$'s short-term memory.

The value of $a_i$ is transformed by the $PU$'s output function, $f_i$, to determine the magnitude of its current output signal. A number of activation functions have been used to construct NNs, of which the step function is the simplest and the most straightforward (Fig. 2a). With the step activation function, a $PU$ produces an output signal of either '1' or '0' dependent on whether or not the level of its activation is above a certain threshold value. That is:

$$
Y_i = \begin{cases} 
1 & \text{if } a_i > 0 \\
0 & \text{otherwise}
\end{cases}
$$

where

$$a_i = \sum_{j=1}^{n} w_{ij}x_j - s_i
$$

However, in order to filter out the noise and hence enhance the ability of achieving a true steady state of operation, for some NNs a sigmoid activation is usually used in practice, expressed in the form of:

$$y_i = f_i(a_i) = \frac{1}{1 + e^{-ca_i}}
$$

where

$$a_i = \sum_{j=1}^{n} w_{ij}x_j - s_i
$$

c is a constant which determines the degree of 'uncertainty' introduced into $PU_i$ activity.

The general shape of this function is as shown in Fig. 2b. Some of the advantages offered by this type of function will become clear later in the text (1/$c$ is also known as the 'temperature'). In some cases its value can be set at an artificially high level initially to 'shake' the NN so that it has a better chance of achieving its true stable state. This is then gradually reduced to allow it to cool down to the ideal state, i.e. step function with zero degree of temperature. This is known as simulated annealing).