Feeding Neural Network Models with GPS Observations: A Challenging Task

R.F. Leandro
Department of Geodesy and Geomatics Engineering, University of New Brunswick, P.O. Box 4400, Fredericton, New Brunswick, Canada, E3B 5A3

C.A.U. Silva
Department of Civil construction, Federal Technologic Learning Centre, Belém, Para, Brazil

M.C. Santos
Department of Geodesy and Geomatics Engineering, University of New Brunswick, P.O. Box 4400, Fredericton, New Brunswick, Canada, E3B 5A3

Abstract. Much has been done in terms of functional and stochastic modelling of observations in space geodesy, aiming at the development of adequate adjustment models. One of the techniques, which has been the focus of more attention in the last years, is the Neural Network model. Although not trivial to be used, this kind of model provides an extreme adaptation capability, which can be an issue of fundamental importance for certain applications. In this paper we discuss the use of GPS observations in Neural Networks models, providing a brief description how a neural model works and what are its restrictions, as well as how to treat the GPS observations in order to satisfy them.

A Neural Network is an information processing system formed by a big number of simple processing elements, called artificial neurons. Typically the input values must be normalized, with typical range [0,1], or alternatively [-1,1]. After processed, the signal can be transformed back to its original origin and amplitude. When dealing with GPS observations, namely ranges and range rates, the absolute numerical values are usually pretty large (e.g. order of 20 millions of meters for ranges) coupled with precisions in the order of mm for carrier-phase and meter for pseudoranges. The observations need to be modified to avoid degrading their precision during the normalization, in order to make the application of neural models suitable for GPS data.

In this work methods to make the use of GPS data possible in neural models are discussed and showed with real examples. The analysis is made for both pseudoranges and carrier-phases. It is demonstrated that with the adequate treatment the use of those observables can be made without degradation of precision.

Keywords. Neural Networks, GPS.

1 Introduction

Modelling plays a fundamental role in Geodesy. Signal processing, physical phenomena functional modelling, interpolation, forecasting, stochastic modelling are a few examples of the applications that require modelling in Geodesy. In most cases adjustments are used, and in this case, the most used technique is the least squares procedure. Filters are also widely used, such as Kalman filter, which involves some of the least squares technique concepts. In the ninety’s a new technique appeared to be useful in Geodesy, called Neural Networks. Primarily developed for computing applications, such as pattern recognition, neural networks have been adapted to be used in several fields of science, including Geodesy. Those adaptations are needed because usually the situations and problems encountered in computer science are sometimes very different than in other fields. Geodesy was not an exception in this case, and because of that, the Neural Network analyst for geodesy needs to have a good knowledge in neural data processing in order to be able to use this technique successfully. Adaptation of the geodetic data may be needed in some cases to make the data useful in a neural model. This is the case explored in this paper, where the problem of using GPS data in neural networks is shown. Compatibility between GPS and the neural model has to be made possible by means of some modification in the original GPS data. This innovating synergy has made necessary the development of novel techniques in terms of GPS data handling.
Chapter 29 • Feeding Neural Network Models with GPS Observations: a Challenging Task

2 Neural Network Models

A Neural Network is an information processing system formed by a big number of simple processing elements, called artificial neurons, or simply neurons. The first artificial neuron model was presented by Rosenblatt (1958), who called it perceptron.

It is possible to introduce a functional link into the network as an additional layer of neurons, called a hidden layer. This layer can be composed of one or more neurons. The input signal of the hidden layer neurons is generated by the output signal of the input layer. The output signal of the hidden layer is used to generate the input signal to the output layer. It is also possible to introduce not just one, but several hidden layers into the model.

A perceptron computes its input as a linear combination of its input signal by using the synaptic weights. The synaptic weights play the role of parameters, which are adjusted at the training process (this procedure will be discussed later in this section). The synaptic weights hold the knowledge of the network. After that an activation function is applied to the neuron input to generate the neuron output (in the case of a single neuron it is already the output signal). One neuron can have one or more outputs, always with the same value. In the case of an identity activation function, the neuron plays the role of a linear model. The processing of a neuron k can be represented by:

\[ y_k = \varphi \left( \sum_{i=1}^{m} (x_i \cdot w_{ki}) + b_k \right), \tag{1} \]

where \( y_k \) is the neuron output, \( \varphi \) is the activation function, \( m \) is the number of input parameters, \( x_i \) is the i-th input parameter, \( w_{ki} \) is the i-th synaptic weight and \( b_k \) is the bias.

Typically the order of normalized amplitude of a neuron output is in within the range \([0,1]\), or alternatively \([-1,1]\). This range depends on the type of activation function used. The neural model also includes a term that is applied externally, called bias and represented in Figure 1 by \( b_k \). The bias has the function of increase or decrease the neuron input.

Figure 2 shows a scheme of a neural network with one hidden layer. In this example, \( x(1) \), \( x(2) \) and \( x(3) \) are the input parameters and \( y(t) \) is the output parameter. Each element, excepting the biases, is a neuron. Each of these neurons is a processing element that works according to equation (1). The synaptic links (the lines in the draw) connect the different layers, carrying the output signal of a previous one to generate the input signal of the next one. Each synaptic link of the network has a corresponding synaptic weight that is applied to the flowing signal that is going through it.

Another issue of a neural network model is the number of neurons of each layer. This number is fixed to the input and output layers, in function of the input and output parameters. For the hidden layers this number is arbitrary. The model resulting from adding hidden layers between the input and output layers is called Multilayer Perceptron (MLP). The MLP is not the only type of neural network model, but is one of the most popular ones, due to its high adaptation capability and its applicability to a wide group of different applications.

It is necessary not just to know which model will be used, but also all its characteristics, such as the number of hidden layers, the number of neurons in each hidden layer, the activation function of each...