Neural networks: A tool for the prediction of slope movements

Les réseaux de neurones: Un outil pour la prédiction des mouvements de pentes

F. Mayoraz, T. Cornu, D. Djukic & L. Vulliet – *Swiss Federal Institute of Technology, Lausanne, Switzerland*

ABSTRACT : This paper presents a first attempt to use artificial neural networks (ANNs) for predicting sliding velocities of large unstable slopes. The processes of the choice of the type of network and of the determinant input or output parameters for feeding these networks is emphasized. Based on data provided by the monitoring of the Sallèdes site (France), results show advantages of such techniques for the approximation of time series.

RESUME: Cet article présente un premier essai d'utilisation de réseaux de neurones artificiels (RNA) pour la prédiction des vitesses de glissement de grandes pentes instables. Le choix du type de réseau ainsi que des paramètres d'entrée et de sortie déterminants est expliqué. Les résultats basés sur les données provenant de la surveillance du site de Sallèdes (France) montrent les avantages de tels outils pour l'approximation de fonctions temporelles.

1. INTRODUCTION

1.1 Predicting landslide movements

Developing models for the kinematics of unstable slopes has always been a great challenge for engineers. This task can be achieved by two different but complementary approaches which both lead to improving prediction models as part of maintenance program, rehabilitation works planning or alarm system.

The first approach is the mechanical one, using continuum theory of varying complexity (Vulliet & Hutter 1988). However, boundary conditions are not easily known: geometry of the free surface and the sliding base — if any —, groundwater seepage conditions, time evolution of rainfalls, etc. Then the material behavior depends on complex features, on which very limited information exists : soil/rock type, heterogeneities, macro-structure, stress-strain-time material behavior, hydromechanical coupling.

The second approach is the statistical one including regression analysis, time series analysis or hydrology-derived reservoir models. These models need a lot of data. This is sometimes incompatible with financial resources that are always limited and prevent from undertaking in-depth hydro-geotechnical investigations.

Artificial Neural Networks (ANNs) belong to the second (statistical) category. Recently, civil engineers have shown an increase of interest for these techniques. Najjar et al. (1996) use ANNs to find compaction characteristics as a function of Atterberg's limits and grain size distribution values. Goh (1994) and Agrawal et al. (1995) apply ANNs to the assessment of the liquefaction susceptibility of a site submitted to an earthquake. Lee & Lee (1996) use these techniques to estimate the bearing capacity of piles. Ghaboussi et al. (1994) and Ellis et al. (1995) model constitutive laws, based upon triaxial tests. Wu et al. (1992) train ANNs to pilot damping devices aimed at reducing the acceleration of structures submitted to the effect of an earthquake.

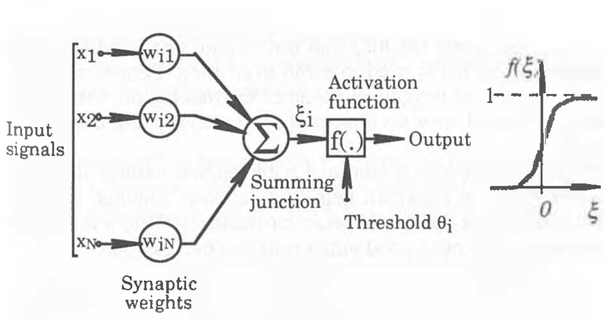

Figure 1 : Basic neuron *i* and its threshold function

This paper is a first attempt to predict slope movements with ANNs based on parameters such as rainfalls, pore pressure and previous displacements. Section 2 presents an overview of the ANN algorithm used. Section 3 explains how the ANNs are applied to the problem of slope movement prediction. Section 4 treats the selection of the proper input parameters for the ANNs. Section 5 presents the main results of the current work. Finally, Section 6 concludes the paper.

2. FUNCTION APPROXIMATION WITH ANN

2.1 Multilayered neural net structure

Multilayered perceptrons (MLPs) are the most common type of artificial neural networks (ANNs) used for both classification and function approximation (Lippmann 1987) .

The structure of MLPs is based on the widely known basic neuron shown on Figure 1. A basic neuron with N inputs contains N adjustable weights w_1, w_2, ..., w_N and performs a weighted sum of its inputs, threshold by a non-linear function. In MLP algorithms, the threshold function is usually a sigmoïd function of the form (see Figure 1):

$$f(\xi) = \frac{1}{1+e^{-\xi}} \qquad (1)$$

2.2 Transfer function of an MLP

It has been shown that a network composed of one intermediate layer of basic neurons and of one output neuron without threshold function (Figure 2) is a universal approximator. Such a network can approximate any given function with sufficient precision, provided that enough neurons are given on the intermediate layer (Leshno et al. 1993)

Given a set of points in the N-dimensional inputs space, $\{(x_1^k, x_2^k, ..., x_N^k)\}$, k=$\{1,2,...,P\}$ and the set of corresponding outputs, $\{o^k\}$, the learning process is then reduced to a multidimensional non constrained optimization process in the space defined by every weight of the system. A learning algorithm tries to fit the transfer function T to the data. This transfer function may be expressed as follows:

$$T(x_1, x_2, ..., x_N) = \sum_{i=1}^{M} v_i f(\sum_{j=1}^{N} w_{ij} x_j - \theta_i) - \mu \quad (2)$$

where x_j are the input values, N and M are the number of inputs and the number of neurons in the intermediate layer; w_{ij} are the weights of the connections between the inputs and the neurons of the intermediate layer, v_i the weights of the connections between the neurons of the intermediate layer and the output neuron; θ_j are the biases of the neurons of the intermediate layer, μ is the bias of the output neuron and f the sigmoïd function defined above.

2.3 *Supervised learning methods for MLPs*

The purpose of supervised learning algorithms for MLPs is to find the proper weight configuration so as to approximate a given function. These algorithms are usually based on iterative gradient descent methods or on Gauss-Newton methods.

The so called backpropagation principle (LeCun 1985) makes it possible to compute the gradient of the output error E of the multilayered neural net, usually defined as:

$$E = \sum_{k=1}^{P} (T(x_1^k, x_2^k, ..., x_N^k) - o^k)^2 \quad (3)$$

with respect to each of the connection weights. Here, P is the number of input vectors in the training set. The partial derivative with respect to weights of non terminal layers are computed recurrently as a function of the one of the next layer.

Many MLP training algorithms use the backpropagation principle in conjunction with gradient descent or conjugate gradient methods. The Marquardt-Levenberg algorithm, often used in non linear least squares problems, has also been adapted to learning in MLPs (Hagan & Menhaj 1994). It is an approximation to Newton's method instead of a pure gradient descent. Though its memory requirements make it impractical for large neural networks, it compares favorably against conjugate gradient techniques for learning in neural networks of reasonable size. This is the algorithm used in this study.

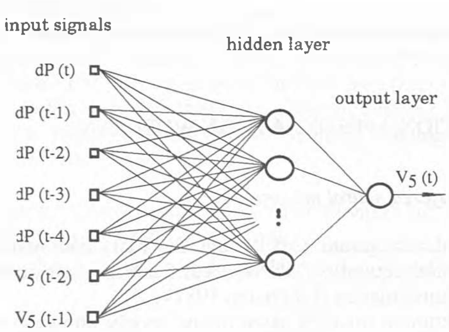

Figure 2 : Network geometry and inputs (dP = pore pressure variation, V_5 = average velocity.

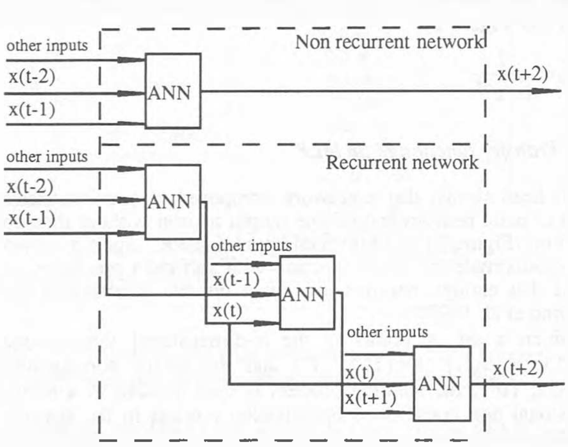

Figure 3 : Three days prediction : Non recurrent network versus recurrent network applied to the test set accounting previous values of the predicted parameter x.

3. NETWORK CONFIGURATION

3.1 *Data*

In the present analysis data are provided by the Sallèdes landslide. The site of Sallèdes was used in a research project lead by the Laboratoire des Ponts et Chaussées (LPC) on the behavior of a road fill built on a creeping slope. The sliding mass concerns an area of 250 m in length and is composed of clayey colluvium of about 6 m in thickness overlying a marl and marly limestones substratum. The sliding surface is inclined by only 7° with respect to horizontal.

The instrumentation is very complete and connected to an automatic data acquisition system (Pouget & Livet 1994). Thus continuous readings of rainfalls, pore-water pressures and displacements are available between 1988 and today. The displacements as a function of time are characterized by rapid accelerations (velocity peaks) followed by periods of rest.

In this paper, we use daily data measurements from a 365 days long period, starting on February 25th, 1992. Figure 4 shows a representation of the pore pressure evolution during the analysed period.

3.2 *Choice of input and output*

Two distinct sub-problems have been examined. The first one is the prediction of pore pressure as a function of the integral of the net rainfalls (i.e., the sum of rainfalls up to the time considered, minus the evapotranspiration). The second problem is the prediction of sliding velocity (i.e., the variation of displacement), as a function of the variation of pore pressure (Figure 2). Prediction of velocity as a function of rainfalls is then achieved by cascading the two sub-systems. Daily values of the pore pressure variation and of the displacement are smoothed, using average values over a five-days sliding window. The analysis of *data dependancies*, explained in Section 4, lead to take into account the last three values of the net rainfalls (first network) and the last four values of the pore pressure variation (second network Figure 2). For both neural networks, values of preceding time steps of the predicted parameter (pore pressure or displacement velocity) are used. The data dependancy analysis shows that the last two to three values of this parameter should be taken into account.

3.3 *Procedure*

The networks chosen for both sub-problems are characterized by one hidden layer, and an output layer with only one neuron. Every input neuron is connected to all neurons of the hidden layer (see Figure 2). The threshold function is a sigmoïd function for the hidden layer and is linear for the output layer.

The optimal number of hidden neurons is determined empirically as the minimal number of neurons for which prediction performance is satisfying, without leading to overfitting or exaggerately long learning times. Here the number of neurons of the hidden layer was found with a few trials : a good convergence as well as a fast computation speed occur with 8 neurons for both systems.

The input vectors are divided into three sets (Figure 4). Besides the two classical sets (training and test) a third one is added : the validation set, which is used to avoid overfitting. Periodically, the network applies the weights resulting of the training on this set; as soon as the global error on this set reaches a minimum, training is stopped.

The training process is launched with random values. In some case however, the algorithm gets stuck in local minima, and the global output error doesn't decrease satisfactorily. Thus a few trials are needed to find out a good initial weights configuration.

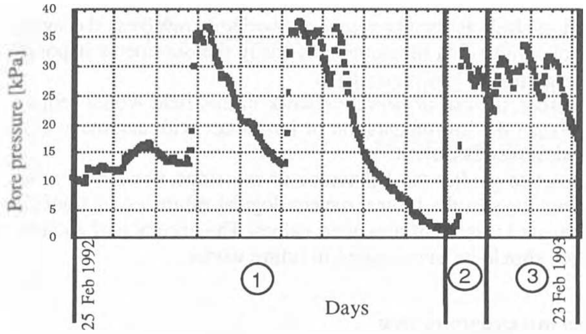

Figure 4 : Sallèdes raw data; pore pressure as a function of time.
1 = training set, 2 = validation set, 3 = test set.

3.4 *Classical MLP versus recurrent MLP*

Previous work (Mayoraz et al. 1996) used a neural network only for short term prediction (one day). Two alternate solutions have been explored to increase the delay between measurements and the prediction. The first method consists of increasing the delay between the inputs and the outputs directly in the training set of the neural network. In the second approach, the neural network is still trained to provide a one-day prediction; it is then applied recurrently to its own predicted output, in order to provide a two-days– and then a three-days prediction. Figure 3 summarizes the two alternate approaches.

4. ANALYSIS OF ESSENTIAL DATA DEPENDANCIES

It is known from statistics that, for a certain size of the training set, the confidence we can have in the estimates of the parameters of a model is, roughly speaking, inversely proportional to the number of parameters. It is, therefore, very important to eliminate the superfluous parameters in the model before the estimation takes place. In order to do so, it is necessary to check the interdependancies in the data which will serve to build the model.

4.1 *Method of analysis*

The method of the analysis of essential data dependencies which is employed here originated from the embedding dimension detection, developed for the analysis of trajectories generated by time series. The original method was firstly published by Grassberger and Procaccia (Grassberger & Procaccia 1983), who used it for detection of the embedding dimension of chaotic time series. The method was later modified for analysis of dependecies in a single time series by Pi and Peterson (Pi & Peterson 1994). The latter method is modified in such a manner as to permit analysis of the systems with inputs and outputs, by testing the interdependecies between two time series. The essence of the method lies in detecting a topological invariant of a time series. The topological invariant is a function of the distances of points of embedded data.

Suppose that the correct dimension be D_e. If we calculate this topological invariant for the data embedded in spaces of dimensions $D = 1, 2, 3, \ldots$ we shall notice that, in principle, this invariant will not change for $D > D_e$.

Let us have finite number of samples of two time series x and y. Suppose that the samples of the series y are generated from finite dimensional vectors of D_e consecutive past samples of the series x by a continuous function (map):

$$y(k) = f(x(k-1), x(k-2), \ldots, x(k-D_e)) \tag{4}$$

Dimension D_e is here unknown, and the only condition put on the function f is that it be a continuous map. For the chosen dimension D, and two real values ε and δ, we calculate the value

$$P_D(\varepsilon,\delta) = \frac{\sum_{i<j} \sigma(\varepsilon, y(i)-y(j)) \prod_{t=1}^{D} \sigma(\delta, x(i-t)-x(j-t))}{\sum_{i<j} \prod_{t=1}^{D} \sigma(\delta, x(i-t)-x(j-t))} \tag{5}$$

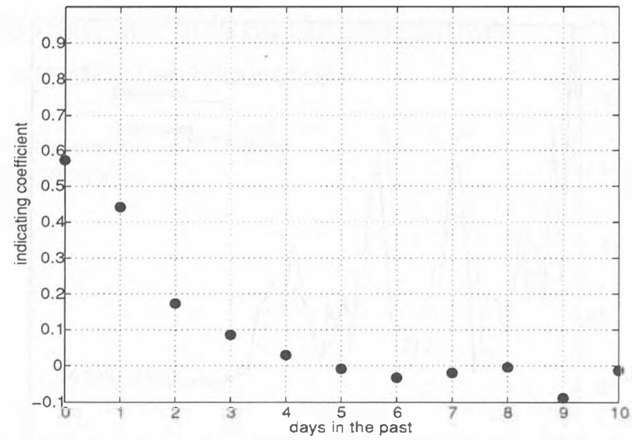

Figure 5 : Dependance of slope sliding velocity on differential pore pressure (Sallèdes landslide).

where $\sigma(a,b) = 1$ if $|b| < |a|$ and 0 otherwise. For this calculation, we use all the available data in order to achieve better accuracy. For each $D = 1, 2, 3, \ldots$, we calculate the indicating coefficient:

$$\lambda(D) = \frac{\int_0^\infty (P_D(\varepsilon) - P_{D-1}(\varepsilon))d\varepsilon}{\int_0^\infty (1 - P_0(\varepsilon))d\varepsilon} \tag{6}$$

where $P_D(\varepsilon) = \max_\delta P_D(\varepsilon,\delta)$ and $P_0(\varepsilon) = \frac{\sum_{i<j} \sigma(\varepsilon, y(i)-y(j))}{\sum_{i<j} \sigma(\infty, y(i)-y(j))}$.

In principle, we shall have that $\lambda(D) = 0$ for all $D > D_e$. Thus, we estimate the value of D_e to be the smallest D_* for which $D > D_* \Rightarrow \lambda(D) = 0$.

4.2 *Application of analysis*

The method was employed to test the time dependencies between the time series of the data relevant for the slope movement (net rainfalls, pore pressure and sliding velocity). A number of tests was performed, checking whether a data sample of one quantity depends on the immediately preceding several data samples of another quantity, and in particular, for how long in the past the dependancy extends.

Figure 5 shows a sample result of the dependancy analysis for the relationship between the sliding velocity and the differential pore pressure. The quantity represented on the graph is the indicating coefficient λ. Its meaning here is to indicate how much the today's velocity depends on a sample of the pore pressure from a certain number of days ago. The lower the value of the indicating coefficient, the less the influence there is.

From the figure, it is clearly visible that the influence of the differential pore pressure does not go very far in the past (here only 2-3 days). An important remark must be given here: since this indicating coefficient is a statistics of the training set, its accuracy is influenced by the size of the training set. We shall point out here the fact that more data is needed to calculate λ for the higher values of D. The values of indicating coefficients for the longer periods of dependence deviate, thus, more from their true values (which is in this case 0).

5. RESULTS

In this section, we present prediction results for the final prediction system, which consists of cascading the two sub-systems presented in Section 4: a first neural network predicts the pore pressure as a function of net rainfalls and pore pressure of previous days; the predicted pressure is derived and used as the input of the second neural network, which predicts the sliding velocitiy.

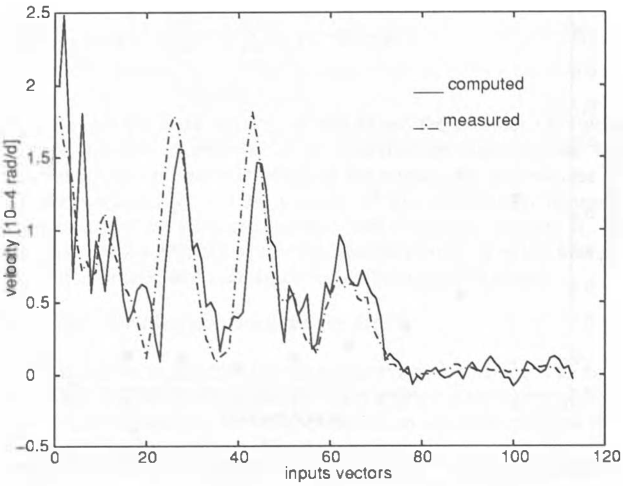

Figure 6: Three days prediction of the sliding velocity with a system based on non-recurrent neural networks (Sallèdes landslide).

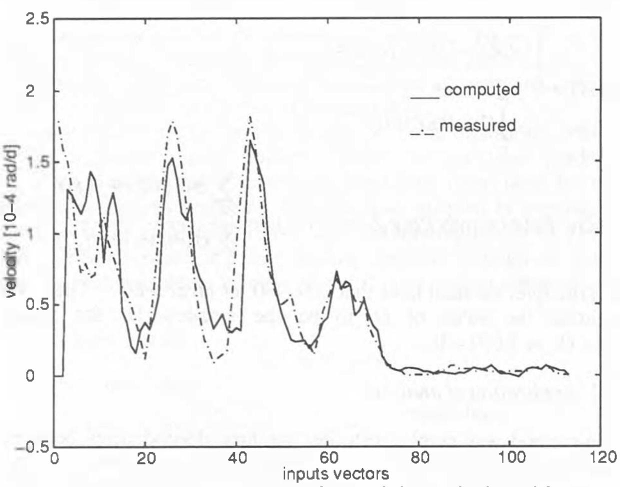

Figure 7: Three days prediction of the sliding velocity with a system based on recurrent neural networks (Sallèdes landslide).

Thanks to the cascading of the two sub-systems, sliding velocity can then be predicted without any actual measurements of the pore pressure, but only based on rainfall amounts. At a later stage of the project, this should make it possible to perform long-dated predictions, using meteorological estimates of rainfalls.

The two alternative approaches described in Figure 3 were tested on our one-year data set. Figure 6 shows a typical result obtained with non-reccurent system for a three-days prediction. Figure 7 shows the same result with reccurent neural networks.

Both networks give reliable results, but the reccurent approach proves slightly superior. Attempts to produce a prediction more than three days in the future were also performed. Prediction of pore pressure as a function of rain amounts remains reliable up to five days in the future. However, the prediction of sliding velocity as a function of pressure, and thus, the one of the complete system, deteriorates.

6. CONCLUSION

A system based on neural networks is presented, that provides a three-days prediction of the sliding velocity of a slope movement. The system is configured off-line thanks to daily measurements of rainfalls and pore pressure. When operated on-line for prediction (as part of an automatic alarm scheme), the system only requires measurements of rainfalls and memorisation of the past values of the sliding velocity.

An effective method is proposed to reduce the model in order to select only the relevant input parameters for the prediction. For this method, as well as for the neural networks themselves, the availability of regular data measurements (daily measurements if possible) is essential.

Therefore, the use of neural network in this field would require to generalize the instrumentation of landslides with automatic data acquisition systems.

In order to predict the behaviour of the slope movement more than three days in the future, meteorological estimates of rainfalls could be used instead of measured values. The accuracy of such an approach should be investigated in future works.

ACKNOWLEDGEMENTS

This work was supported in part by the Swiss Federal Highway Administration, grant no 54/95, 4217.01; this contribution is gratefully acknowledged.We would like to thank Mr. P. Pouget of the LPC in Clermont-Ferrand for giving us the data for the Sallèdes case.

REFERENCES

Agrawal, G., J.-L. Chameau and P.-L. Bourdeau 1995. *Artificial Neural Networks for Civil Engineers. Fundamentals and Applications*. Assessing the Liquefaction Susceptibility at a Site Based on Information from Penetration Testing: 10.

Ellis, G. W., C. Yao, et al. 1995. Sress-strain Modeling of Sands Using Artificial Neural Networks. *Journal of Geotechnical Engineering ASCE* 121 (5) : 429-435.

Ghaboussi, J., D. E. Sidarta and P. V. Lade 1994. Neural network based modelling in geomechanics. *Proc. Int. Conf. on Computer Methods and Advances in Geomechanics*: Rotterdam: Balkema.

Goh, A. 1994. Seismic liquefaction potential assessed by neural networks. *Journal of Geotechnical Engineering ASCE* 120 (9) : 1467-1480.

Grassberger, P. and I. Procaccia 1983. Measuring the Strangeness of Strange Attractors. *Physica D* 9 : 189-208.

Hagan, M. T. and M. B. Menhaj 1994. Training Feed-Forward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks* 5 (6) : 989-993.

LeCun, Y. 1985. A Learning Scheme for Asymetric Threshold Networks. *Proc. Cognitiva 85* : 599-604.

Lee, I. M. and J. H. Lee 1996. Prediction of pile bearing capacity using neural network. *Computers and Geotechnics* 18 (3) : 189-199.

Leshno, M., V. Y. Lin, et al. 1993. Multilayer FeedForward Networks with a Non Linear Activation Function Can Approximate any Function. *Neural Network* 6 : 861-867.

Lippmann, R. P. 1987. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine* : 4-22.

Mayoraz, F., T. Cornu and L. Vulliet 1996. Using Neural Networks to Predict Slope Movements. *Proc. Int. Symp. on Landslides* : 295-300: Balkema: 1.

Najjar, Y. M., I. A. Basheer and W. A. Naouss 1996. On the Identification of Compaction Characteristics by Neuronets. *Computers and Geotechnics* 17 : 167-187.

Pi, H. and C. Peterson 1994. Finding the Embedding Dimension and Variable Dependencies in Time Series. *Neural Computation* 6 : 509-520.

Pouget, P. and M. Livet 1994. *Relations entre la pluviométrie, la piézométrie et les déplacements d'un versant instable (site expérimental de Sallèdes)*. Etudes et recherches des laboratoires des ponts et chaussées: s. géotechnique GT 57: février 1994.

Vulliet, L. and K. Hutter 1988. A continuum model for natural slopes in slow movement. *Géotechnique* 38 (2) : 199-217.

Wu, X., J. Ghaboussi and J. H. Garrett 1992. Use of Neural Networks in Detection of Structural Damage. *Computers and Structures* 42 (4) : 649-659.