ACTA MONTANA IRSM AS CR (1996)
Series A, No. 10 (102), 133–144

PREDICTION METHODS BASED ON MULTICHANNEL STATISTICAL EXTRAPOLATION AND THEIR APPLICATION ON INDUCED SEISMIC EVENTS

RADIM ČÍŽ and VLADIMÍR RUDAJEV

Institute of Rock Structure and Mechanics, Academy of Sciences of the Czech Republic
V Holešovičkách 41, 182 09 Prague, Czech Republic

ABSTRACT. Seismic data obtained by means of local network in Kladno mine region are the basis for establishing the time series describing the seismic activity of rock bursts area. In such a way number of individual tremors occurrence and their maximum amplitude per day was derived. These seismic time series were complemented by the information of excavation activities, namely as for the length of entries per day. The questions of extrapolation of these input series were based on the use of neural networks.

KEYWORDS: induced seismicity, neural networks, time series

1. INTRODUCTION

The occurrence of earthquakes and induced seismic phenomena can be understood as a stochastic time–space processes. Similarity of shallow tectonic earthquakes and mine tremors – physically equal as concerns a sudden release of seismic energy – indicates the possibility of applying very similar methods of events monitoring as well as the procedures of data processing and results interpretation. In many cases mine tremors can be studied as the models of shallow tectonic earthquakes. The major difference, however, lies in the nature of both these phenomena: As concerns mine induced seismic events, besides different nature conditions, anthropogenic activities play the decisive role. There are several parameters such as a primary 3–axial stress fields, namely size and geometry of excavation volume, rate of excavation process, mining technology applied (especially the parameters of blasting), etc. All these factors result in the redistribution of the next local stress field in massif. Given combination of the stress/strain changes and a primary stress field may – under particular rock parameters – trigger mine tremors.

As for the mine tremors, we have for our disposal relatively detailed information on local geological and texture structure, on tectonical field, on physical rock properties and on geometry of focus region. In numerous mines local seismic networks have been installed; they allow the objective source parameters to be determined, such as focus location, focus time, seismic energy release (and magnitude, respectively), seismic moment, stress drop, source mechanism, etc. Regardless all this detailed knowledge, however, until now neither a reliable deterministic prognosis, nor a statistical extrapolation of mine tremor occurrence have been found yet. It follows that a question of mine tremors prediction still remains much important. Therefore, this contribution deals with extrapolation of mine tremor time series (as for origin time and events number) utilizing a method of neural networks. The advantage of this approach, evidently, lies in its adaptability as for the momentary state of extrapolated tremor series and simultaneous comparison with the past period data. The algorithm of neural network is suitable for examining non–stationary processes of seismic energy release.

2. Neural Networks – Theoretical Background

Artificial neural networks (ANNs) or simply neural networks is a term used to describe models which simulate certain basic structures of the human brain to imitate some of its functions. It is important to understand what neural networks can do [Duch, Diercksen 1994] and when their application may lead to new results, hard obtaining with standard methods. Neural networks are algorithms capable of solving such tasks as optimization, pattern recognition, filtering, prediction, association or interpolation. ANNs are composed of simple processing units (called "neurons") operating on their local data and communicating with other units. Thanks to this global communication the ANN has stable states consistent with the current input and output values. The weights of connections between the neurons are adjustable parameters. Their modification allows the network to realize a variety of functions.

Neural networks have a lot of good properties [Müller, Reinhardt 1990; Masters 1993], which are successfully used. First of all it is possibility of network to learn, it means to change their own parameters and structure according to specific information to better correspond to limited requirements. It has to have certain feedback, which controls output results and changes network parameters. In the case of supervised learning a set of input and output patterns is shown to the network and the parameters are adjusted (this is called "learning" or "adaptation") until the outputs given by the network are identical to the desired ones. In the case of unsupervised learning a set of input and output patterns is not at disposal, the network has to learn of its own errors and to fulfil acceptable criterion. In both cases the network try to built its own inner model, which would simulate real system. This ability is next very good property using by prediction [Weigend, Gershenfeld 1994; Aso et. al. 1994].

The three phases in a neural network processing are: Learning, Generalization and Operation. Learning is the process by which the network "learns" to "recognize" input patterns in the form of a training set of data. Each neural network is accompanied by a set of rules which define how the neural network learns. Generalization is the ability of a network to give an acceptable response for inputs which are not members of the training sets. Operation is the application of the network to perform the tasks for which it was designed.

The fundamental processing unit of neural network is one neuron, which operates according to relation [Murat, Rudman 1992; Šnorek, Jiřina 1996]

$$y_i = S\Big(\sum_{r=1}^{m} w_{ri} x_r + \Theta_i\Big), \qquad (1)$$

where y_i is neuron output, x_r are neuron inputs, w_{ri} are weights, S is usually non–linear transfer function and Θ_i is a bias value. It receives any number (n) of inputs and delivers a single output. The neuron consists of an input function, or summation function, the result of which is feed to a transfer, or activation, function which in turn determines the output. This function is usually non–linear and maps the total input of a neuron to an output value. The basic neuron structure is shown in Fig. 1. One of the most important features of a transfer function is stability. The derivative of the transfer function determines the proportion of error distributed to each connection when the weights are adjusted. On the basis of biological motivations the sigmoid shaped function (Fig. 2) is often used. Its derivative has a Gaussian shape that helps stabilize the network.

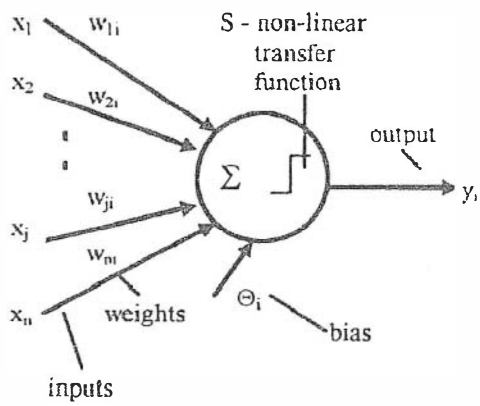

FIG. 1. Basic neuron

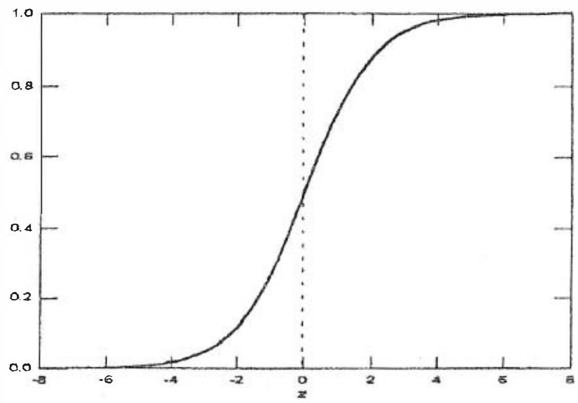

FIG. 2. Sigmoid transfer function $S(z)$ used in our example $S(z) = 1/(1 + e^{-z})$

The network architecture is the manner in which the neurons are organized and connected, i.e., the topology of the network. Neurons can be combined to form layers. Layers can be connected in any number of ways but two general types of connection are distinguished: partial and full. In a fully connected network the output from every neuron in one layer is connected to every neuron in the next layer. All other networks are partially connected. The strength of a connection between two neurons is given by the connection weight between them. Three types of layers are distinguished: the input layer which accepts the input pattern to the network, the output layer which delivers the network output. The third type of layers are intermediate layers or hidden layers, which process the accepted data. An example is shown in Fig. 3.

Memory of neural networks lies in the weights interconnecting neurons. The weights w_{ji} are calculated during a supervised training routine in which a set of inputs is presented to the network together with a target output. The network

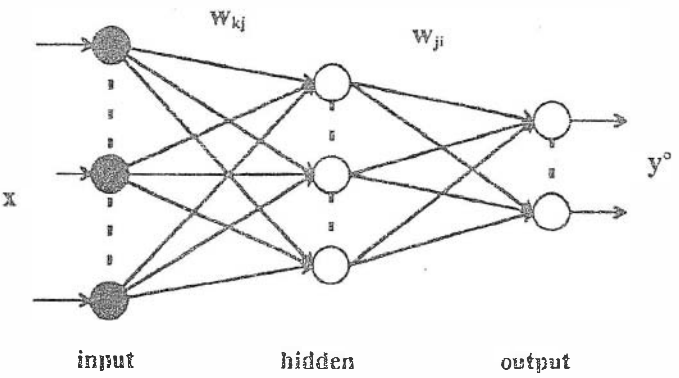

Fig. 3. The typical architecture of a multilayered feed-forward neural network with one hidden layer

undergoes a process of learning which is any change $\triangle w_{ji}$ in the memory. One of the most popular ways to train a neural network (this is the determination of weights w_{ji}) is the back–propagation algorithm (BP) [Rumelhalt et. al. 1986; Poulton 1992; Dowd, Saraç 1994], which is based on a type of least–squares error minimization by a gradient descent method. They derived an algorithm based on the following generalized delta rules. The neural networks, which have parameters adjusted by BP algorithm are called back propagation networks.

The goal of back–propagation algorithm is to minimize the error between the computed output and the desired target output for given training set. Thus we choose to minimize the error function, so called energetic function of neural network

$$E = \frac{1}{2} \sum_{r=1}^{n} (y_r - d_r)^2 \,, \tag{2}$$

where r runs over the number of output neurons and d_r is target output. First step of BP algorithm is weights initialization by small random number. Input training patterns are forward propagated through network from layer to layer applying weights, calculate output using relation (1). Now, the gradient of the energy function is computed and the weight changes are given according to following delta rules [Rumelhalt et al. 1986]

$$\triangle w_{ji}^{l}(t) = -\eta \frac{\partial E}{\partial w_{ji}^{l}} = -\eta \delta_i^l y_j^{l-1}(t) \,, \tag{3}$$

where upper index l denotes the layer, which are weights connected with, δ_i^l is error of ith neuron in lth layer and η is length of learning step. The error of output neurons are calculated by relation

$$\delta_i^o = (y_i^o - d_i)\, y_i^o\, (1 - y_i^o) \tag{4}$$

and passed back to each hidden neuron. The upper index o marks neurons in output layer and y_i^o is the output of the network. Appropriate weight changes are made at each layer simultaneously by relation

$$w_{ji}^{l}(t+1) = w_{ji}^{l}(t) + \triangle w_{ji}^{l}(t) \,. \tag{5}$$

Back propagated error of hidden neurons is given by the equation

$$\delta_j^{h-1} = y_j^{h-1}\left(1 - y_j^{h-1}\right)\sum_{r=1}^{n} w_{jr}^h \cdot \delta_r^h\,, \tag{6}$$

where h denotes the hidden layer. All training patterns (x) are passed through the network. Total error is computed as a summary of individual patterns error. Then the procedure is repeated until the total error reaches a predetermined minimum value.

Multilayered networks trained by back propagation algorithm are relatively easy to understand and implement. However, back propagation also suffers from several disadvantages. There is no guarantee that the network will converge in a finite number of steps. Paralysis, in which weights take on extreme values and learning ceases, can be a severe problem unless very small step sizes are used. Small step sizes can greatly increase training times. Back propagation network can also get trapped in local minima on highly convoluted energetic surfaces.

Back propagation method is a steepest descent technique with a locality constraint. As such, it can be slow to converge when the energetic surface is fairly flat and small adjustments are made to the connection weights. When the energetic surface is highly convoluted, large weight changes may overstep the global minimum. There is no guarantee that the direction of the negative gradient is in the same direction as the global minimum. Therefore, the extended delta rule was used to attempt to overcome the above mentioned limitations. The extended delta rule is given by relation [Šnorek, Jiřina 1996]

$$\Delta w_{ji}^l(t) = -\eta\delta_i^l y_j^{l-1}(t) + \alpha\Delta w_{ji}^l(t-1)\,, \tag{7}$$

where α is momentum. The principal improvement to extended delta rule is a time-varying momentum term for each connection weight. The extended delta rule also places more restrictions on the rate of decrement and increment of the step size and momentum terms.

3. Application of Neural Networks Method

The above described method allows to process arbitrary number of input series simultaneously. We analysed the following series: number of individual tremors occurrence and their maximum amplitude per day. These data were obtained by means of local seismic network built up in Kladno coal mine basin. This network consists of nine seismic stations; four of them are surface stations, three other ones are located in the excavation level (450 m under surface). The last two stations are situated along the vertical profile in the depths 200 and 300 meters, respectively, see [Růžek 1995]. The tremors were found to be located in the overburden sandstone strata in the interval 20 – 150 meters above the coal seam, see Fig. 4. Horizontal range of the focus was limited within the radius 200 meters. The input set of data – from the viewpoint of their space distribution – can be understood as homogeneous.

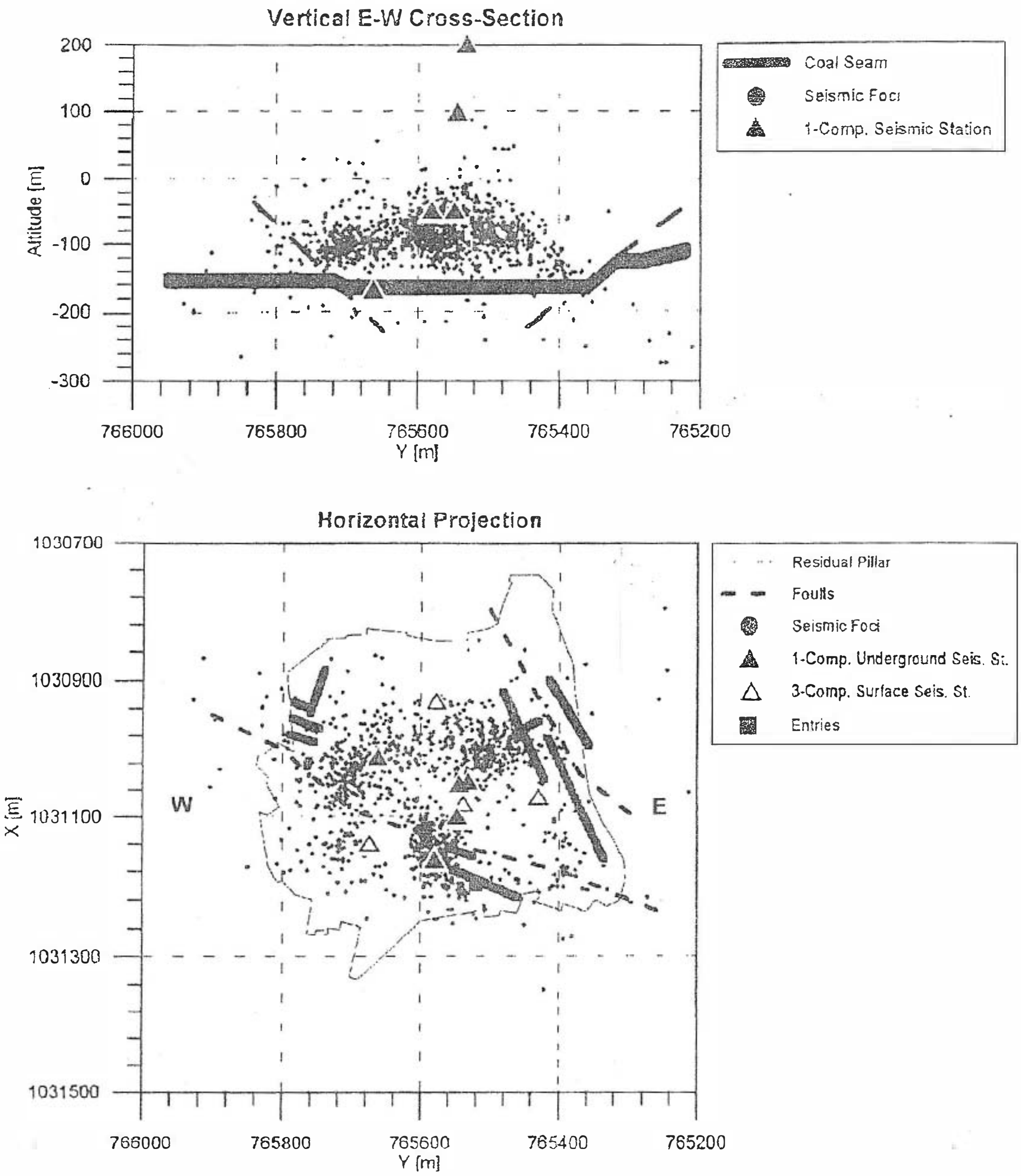

Fig. 4. Schematic maps of localized events (January – April 1994); Kladno–Mayrau coal region

Concerning the local magnitudes determined of maximum ground velocity amplitude $A_{\max}$ and hypocentre distance r by relation

$$M = \log\left(\frac{A_{\max}}{2\pi}\right) + 0.96 \cdot \log r\,. \tag{8}$$

The maximum M values did not exceed 2.5, see [Číž, Růžek 1996]. Utilizing the local seismic data the parameters of the region of question were determined; in

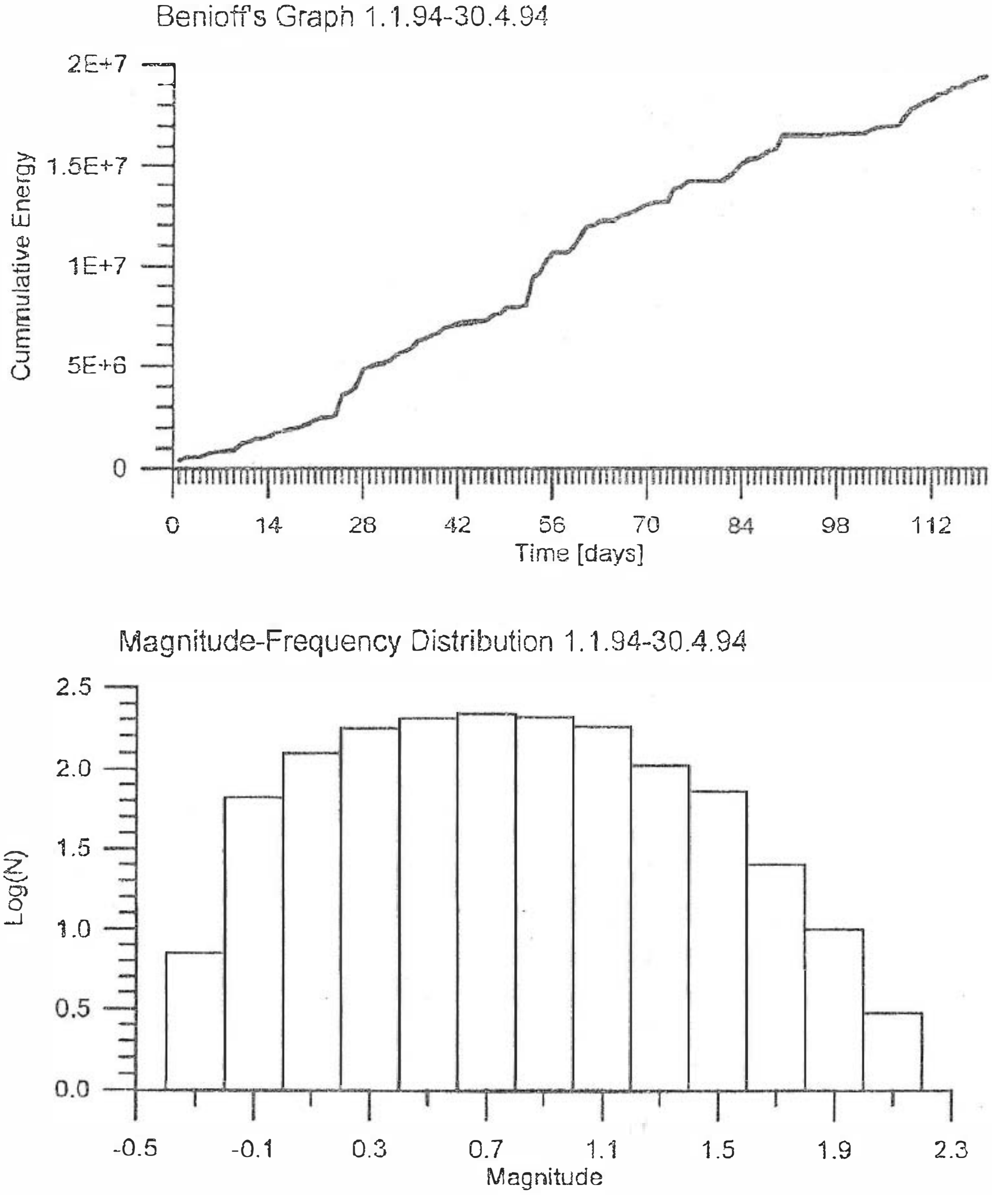

FIG. 5. Characteristic of seismicity

Fig. 5 both pattern of cumulative energy in time and magnitude–frequency distribution are shown. The course of cumulative energy indicates division of whole the time interval into several stationary parts, which are suitable for predictive purposes. The magnitude–frequency distribution, which is often used for seismic hazard assessment for analyzed seismic tremors, does not follow supposed exponential distribution: it can better be described by a bilogarithmic distribution. For more details, see, e.g. [Rudajev et al. 1995; Rudajev et al. 1996].

In Fig. 6 the time series of mine tremors number and their maximum amplitude per day are presented; this seismic sequences were complemented by the information

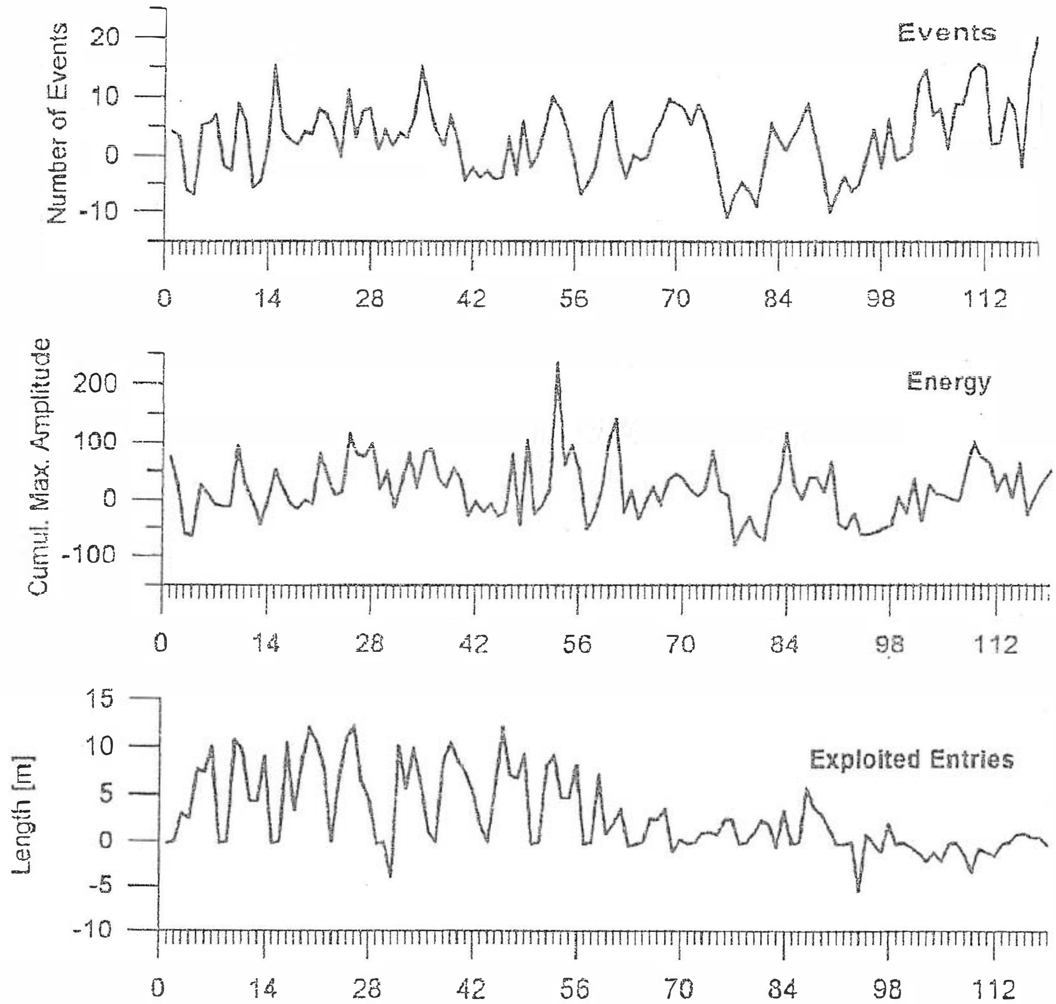

Fig. 6. Input seismic and mining time series

on excavation activities, namely as for the extension of entries in the region under study. This picture shows a centered series (relieved of 7–day period) in which training and predicted intervals are involved.

Back propagation neural network with one hidden layer was applied for the above described seismic and mining time series. The network has 15 input neurons recapting of 5 inputs elements of each time series. The output layer contains only one output neuron, which predicts future number of seismic events. The best results were obtained with 6 hidden neurons. Owing to sigmoid function with limits 0 and 1, the input series were transformed to interval $\langle 0, 1\rangle$. Training process started with 50 patterns showing on Fig. 5 and every one new pattern was added to the training set. The patterns were sequentially given to the network, a momentum parameter α was 0.7 and learning rate η was set to be 0.3. After about 2000 cycles the network was adapted.

In the right part of the Fig. 7 also the diagram of predicted series in time is shown. The quantitative evaluation prediction Q of reliability was reached by means of least

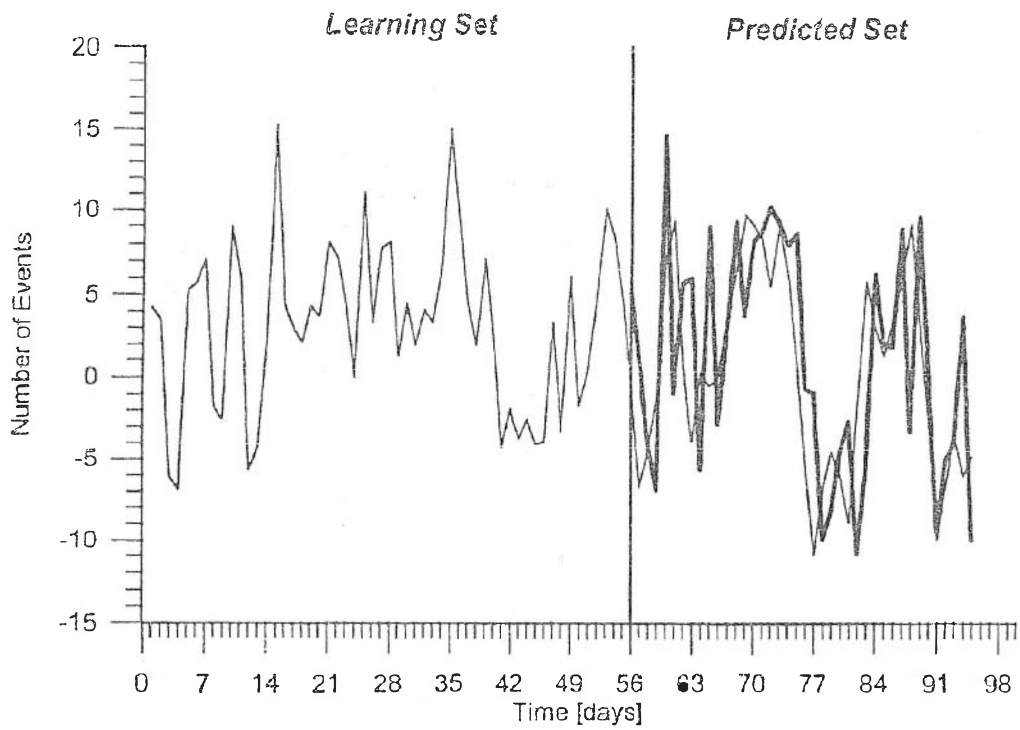

FIG. 7. Actual (thin line) and predicted (thick line) time series of number of events per day

square method according to following equation

$$Q = \frac{\sum_{i=1}^{n} (\hat{x}_i - x_i)^2}{\sum_{i=1}^{n} x_i^2}. \tag{9}$$

Here $\hat{x}_i$ denotes predicted number of mining tremors per day, x_i is the observed number of the same quality and $\sum x_i^2$ denotes dispersion of the observed time series. The application of the above equation indicates that the Q–value approaches to 1. It follows that the use of method on the input three series in question does not seen to be suitable.

4. CONCLUSIONS

It was found, that seismic data complemented by fundamental excavation parameters, such as e.g. length of entries, volume and rate of exploitation, blasting works etc., are not yet sufficient enough for reaching unambiguous mine tremors prediction. This result reflects the fact, that mining tremors occur due to final deformational process in a given part of rock massif. Reasonable utilizing of the methods of prediction is conditioned by complementation input seismic series by the values of massif properties variable in time and space during a massif load, in levels near to critical straights values. It should be pointed out, when studying mining induced seismic events,that majority of the above mentioned parameters can be tested and measured in the vicinity of mine tremors foci. Changes of rock massif parameters can be determined, e.g. by the measurement of seismic wave velocities and absorption values of artificially generated pulses and by monitoring the nature acoustic emission, which occurs due to increased level of rock loading.

Appendix: Derivation of Gradient Delta Rule (3)

We suppose error function after relation (2)

$$E = \frac{1}{2}\sum_{r=1}^{n}(y_r - d_r)^2 \tag{A1}$$

and individual neurons process inputs after relation (1)

$$y_i = S(z_i) = S\Big(\sum_{r=1}^{n} w_{ri}x_r + \Theta_i\Big)\,, \tag{A2}$$

where the total inputs x_r to neuron i is a linear function of the outputs z_i. The bias value Θ_i is equivalent to a threshold of the opposite sign. Outputs non–linear transfer function of neuron is sigmoid function given by Fig. 2

$$y_i = \frac{1}{1 + \mathrm{e}^{-z_i}}\,. \tag{A3}$$

The aim is to find a set of weights that minimizes the error function (A1). To minimize E by gradient descent it is necessary to compute the partial derivative of E with respect to each weight in the network. The backward pass which propagates derivative from the top layer back to the bottom one is more complicated. The backward pass starts by computing $\partial E/\partial w_{ji}^o$ for each of the output neurons and this partial is written as

$$\frac{\partial E}{\partial w_{ji}^o} = \frac{\partial E}{\partial y_i^o}\,\frac{\partial y_i^o}{\partial z_i}\,\frac{\partial z_i}{\partial w_{ji}^o}\,, \tag{A4}$$

where the first partial derivative is derivative of error function (A1)

$$\frac{\partial E}{\partial y_i^o} = (y_i^o - d_i)\,. \tag{A5}$$

The second partial derivative in expression (A4) is derivative of sigmoidal function. This derivative can be better expressed as the function of output value y_i

$$\frac{\partial y_i^o}{\partial z_i^o} = y_i^o\,(1 - y_i^o)\,. \tag{A6}$$

The least partial derivative in (A4) is computed following

$$\frac{\partial z_i^o}{\partial w_{ji}^o} = y_j^h \,, \tag{A7}$$

where upper index h signs the hidden layer. Now we can compose the partial results (A5), (A6) and (A7) we get

$$\frac{\partial E}{\partial w_{ji}^o} = \underbrace{(y_i^o - d_i)\, y_i^o\, (1 - y_i^o)}_{\delta_i^o} y_j^h \,. \tag{A8}$$

The simplest version of gradient descent is to change each weight by following delta rule

$$\Delta w_{ji}^o(t) = -\eta \nabla E = -\eta\, \frac{\partial E}{\partial w_{ji}^o} = -\eta \delta_i^o y_j^h(t)\,. \tag{A9}$$

Computation of weight changes in hidden layer is similar to weight computation in output layer. For each weight of hidden layer is computed $\partial E / \partial w_{kj}^h$ by relation

$$\frac{\partial E}{\partial w_{kj}^h} = \underbrace{\overbrace{\frac{\partial E}{\partial y_i^o} \frac{\partial y_i^o}{\partial z_i^o}}^{\delta_i^o} \overbrace{\frac{\partial z_i^o}{\partial y_j^h}}^{w_{ji}^o} \cdot \frac{\partial y_{ji}^h}{\partial z_j^h}}_{\delta_j^h} \frac{\partial z_j^h}{\partial w_{kj}^h} \,. \tag{A10}$$

From this relation it is seen that the error of hidden neurons δ_j^h is computed by multiplying error of output neuron δ_i^o and appropriate weight w_{ji}^o. The error of output layer is thereby transferred to the hidden layer. Therefore the output layer involves many output neurons than the error of hidden neuron is given by the following relation

$$\delta_j^h = y_j^h\, (1 - y_j^h) \sum_{r=1}^{n} w_{jr}^o \delta_r^o \,. \tag{A11}$$

The weight changes in hidden layer is given by the same delta rule as in output layer

$$\Delta w_{kj}^h(t) = -\eta\, \frac{\partial E}{\partial w_{kj}^h} = -\eta \delta_i^h y_k^{h-1}(t)\,. \tag{A12}$$

According to above described technique is error back propagated from the top layer to the bottom layer and the weights are adjusted after relation (5). Form here is the name back–propagation method and back–propagation neural networks.

Acknowledgement: This paper was particularly funded by the Grant Agency of the Czech Republic, registration number of the project 105/96/1065.

References

Aso T., Ogawa T., Abe M. (1994), *Application of Back–Propagation Neural Computing for the Short–Term Prediction of Solar Flares*, J. Geomag. Geoelectr. **46**, 663–668.

Číž R., Růžek B. (1996), *Periodicity of Mining and Induced Seismicity in the Mayrau Mine, Czech Republic*, Studia Geoph. et Geodaet., in press.

Duch W., Diercksen G.H.F. (1994), *Neural networks as tools to solve problems in physics and chemistry*, Computer Physics Communications **82**, 91–103.

Dowd P.A., Saraç C. (1994), *A Neural Network Approach to Geostatistical Simulation*, Mathematical Geology **26**, no. 4, 491–503.

Masters T. (1993), *Practical Neural Network Recipes in C++*, Academic Press, San Diego CA.

Murat M.E., Rudman A.J. (1992), *Automated First Arrival Picking: A Neural Network Approach*, Geophysical Prospecting **40**, 587–604.

Müller B., Reinhardt J. (1990), *Neural Networks, An Introduction*, Springer–Verlag, Berlin Heidelberg.

Poulton M.M., Sternberg B.K., Glass Ch.E. (1992), *Location of Subsurface Target in Geophysical Data Using Neural Networks*, Geophysics **57**, no. 12, 1534–1544.

Rudajev V., Lokajíček T., Číž R. (1995), *Structural Elements of Rocks and Energy–Frequency Distribution*, Acta Montana **A8** (96).

Rudajev V., Vilhelm J., Kozák J., Lokajíček T. (1996), *Statistical Precursors of Instability of Loaded Rock Samples Based on Acoustic Emission*, Int. J. Rock Mech. Min. Sci. & Geomech. Abstr., in press.

Růžek B. (1995), *Seismology Monitoring*, Acta Montana **A7** (96), 11–19.

Rumelhalt D.E., Hilton G.E., Wiliams R.J. (1986), *Learning Representations by Back–Propagating Errors*, Nature **323**, 533–536.

Šnorek M., Jiřina M. (1996), *Neuronové sítě a neuropočítače*, Neural Networks and Neurocomputers, Vydavatelství ČVUT, Praha.