J Radioanal Nucl Chem (2010) 283:403–407
DOI 10.1007/s10967-009-0413-z

Combined application of Monte Carlo method and neural networks to simulate qualitative prompt gamma neutron activation analysis

V. Doostmohammadi · D. Sardari · A. M. Nasrabadi

Received: 16 July 2009 / Published online: 3 December 2009

Abstract Prompt gamma spectrum produced by thermal neutron absorption and fast neutron inelastic scattering is simulated using Monte Carlo code MCNP4C. The simulated spectrum is analyzed with artificial neural network techniques. The neural network in our study is trained based on back-propagation algorithm with 138 gamma ray spectra. Elements existing in the 20 different substances are specified. The ANN could identify elements correctly in 96% of input cases.

Keywords Gamma-ray spectroscopy · Neutron activation analysis · Monte Carlo · Neural network

Introduction

Prompt gamma neutron activation analysis (PGNAA) is a kind of non-destructive test being used to study biological, environmental and industrial samples [1–4]. Inspection of sealed or buried objects for explosive materials, analysis of ore samples for their content of elements, such as precious metals, are among applications of this method.

In this method, each element is specified by gamma ray energy line or lines emitted promptly after a thermal neutron is absorbed by the nucleus of a specific nuclide. The same procedure takes place after a fast neutron undergoes inelastic scattering inside the sample. PGNAA is a high precision and quick method, but it suffers from spectral complexity when there are too many different elements in the same sample. In addition, gamma ray emitted from material existing in laboratory environment (shields, detector material, etc.) produces extra peaks causing error in analysis results [5]. Thus software techniques are employed to overcome this difficulty [6, 7].

In the present work, artificial neural network (ANN) techniques are used to extract qualitative analytical data from the spectrum. The neural network can find a specific pattern in the complicated input data. The ANN is trained by a gamma ray spectrum library which is developed in the present work. The spectrum produced by a set of elements is given to the input of the ANN. With use of Monte Carlo code MCNP4C a large number of gamma ray spectra were produced. Experimental and the library spectrum should be produced for the same geometry. Production of library spectra in laboratory is usually an expensive and time consuming exercise, thus Monte Carlo method is utilized.

Materials and methods

Monte Carlo simulation

Gamma ray spectra produced by inelastic scattering of impinging fast neutrons and capture of thermal neutrons in the sample were generated by use of the Monte Carlo computer code MCNP4C. Gamma rays from most common elements such as Oxygen, Nitrogen, Carbon, Hydrogen, Sodium, Aluminum, Calcium, Potassium, Iron, Lead, Iodine, Copper, etc. are considered. A library of spectra including 138 members was produced. This includes a gamma ray spectrum of each individual element and spectra generated from various combinations of elements.

V. Doostmohammadi · D. Sardari (✉)
Faculty of Engineering, Science and Research Branch, Islamic Azad University Tehran, P.O. Box, 14515-775 Tehran, Iran
e-mail: Sardari@srbiau.ac.ir

A. M. Nasrabadi
Department of Biomedical Engineering, Faculty of Engineering, Shahed University, Tehran, Iran

A simple geometry consisting of a spherical shell with inner radius of 3 cm and outer radius of 4 cm (Fig. 1) is introduced to the computer code. By use of mode card in MCNP4C code, only neutrons and photons are considered. Importance factor of 1 is assigned for all particles. A fission neutron source emitting thermal and fast neutrons with energy up to 14 MeV was placed at the center of sphere. The spectrum of prompt gamma on the outer surface of sphere was computed by F1 tally in MCNP code up to 11 MeV. The spectra were 1,024 channels, which is sufficient when NaI or BGO detectors are used.

The simulated gamma ray spectrum from TNT is depicted in Fig. 2. In this figure four major peaks showing the existence of elements O, N, C and H are clearly visible and are marked with their energies.

Neural network implementation

A multilayer neural network including input layer, output layer and hidden layer was chosen. The input layer consisted of 1,024 neurons, one input corresponding to each energy channel separately. During this study, it was shown that a hidden layer consisting of 70 neurons produces the minimum error in the output. In the course of training, error was computed with Eqs. 2 and 3 below. Output layer was made up of 82 neurons, chosen in proportion to the multitude of elements appearing in the analysis result. Elements with atomic number up to $Z = 82$ (lead) might appear in the output. In case of existence of an element in the sample, signal is generated in the relevant output. The neural network was trained by use of back-propagation algorithm. In this algorithm, the performance of neural network is fixed over two steps. At the first step, a pre-specified input drawn from the library is fed to the first layer of the network. At the second step, the output is compared with the result that is expected from that input. The neural network parameters are adjusted in a way to reduce the difference between the observed output and the expected one to a minimum. The error in jth output neuron after n iteration is defined as [8]:

$$e_{\mathrm{j}}(n) = d_{\mathrm{j}}(n) - y_{\mathrm{j}}(n) \tag{1}$$

where $d_{\mathrm{j}}(n)$ is the desired output and $y_{\mathrm{j}}(n)$ is the observed output. The error energy for the jth neuron is defined as:

$$E_j(n) = \frac{1}{2}e_j^2(n) \tag{2}$$

where $E_j(n)$ is the instantaneous value of the error energy for neuron j in the nth training example; $e_j(n)$ is the error in jth output neuron after n iteration.

The error energy is a parameter that might be used in training the neural network. This is accomplished by varying the weight of each neuron by use of Eq. 3 as follows [8]:

$$\Delta w_{ji}(n) = -\eta\frac{\partial E(n)}{\partial w_{ji}(n)} \tag{3}$$

where w_{ji} is the weighting value; η is the learning factor; $\partial E/\partial w$ is the partial derivative of error energy as function of weight value.

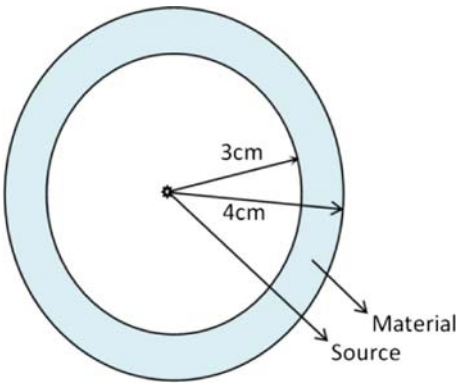

Fig. 1 The simulated geometry of source and sample to generate the gamma ray spectrum

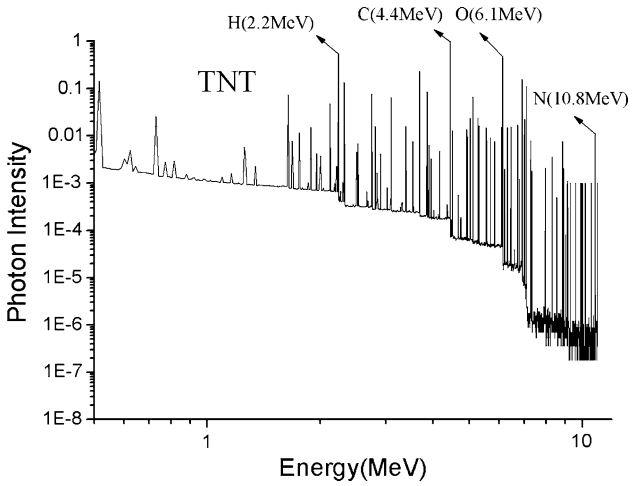

Fig. 2 The spectrum of prompt gamma ray emitted from TNT simulated by MCNP4C code

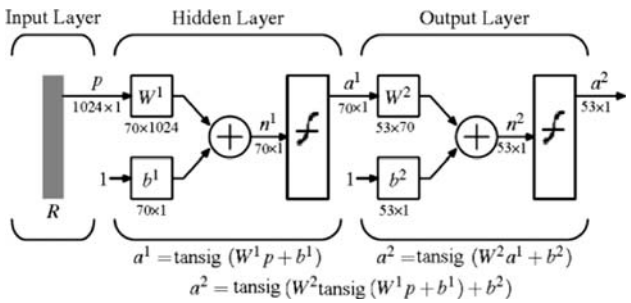

Fig. 3 General structure of a neural network

Tansig function was chosen as the activation function of neurons. This function varies in the range [−1,1]. The schematic structure of our neural network is shown in Fig. 3.

ANN input and training

The training procedure began with feeding the spectrum of each element separately to the ANN. Since one purpose of the present research was to help detecting the explosive material, most important elements were Carbon and Hydrogen. On the other hand, C and H emit only one gamma ray line each, making the task of their detection with use of neural network difficult. Thus the ANN was trained with the spectra of material containing rather Carbon and Hydrogen. This fact along with the required low error, led us to generate more and more gamma ray spectra with MCNP and at last a library of 138 spectra was produced for training purpose.

The simulated gamma ray spectrum was distributed in 1,024 channels. The content of each channel was fed to a separate neuron. The neural network was trained with 138 various spectra as inputs. In Fig. 4, it is shown that the output precision of the neural network improves with increase in the repetition of trainings. With 138 training spectra, the precision in ANN performance amounts to more than 96%.

Existence of an element in the sample under investigation is depicted as a digit "1" in the relevant output of the neural network and lack of the element would be demonstrated with "0". The acceptance threshold at the output was set to 0.4, it means output higher than this value would be considered as "1".

Among back-propagation algorithms that are available in MATLAB 7.6 toolbox, SCG (scaled conjugate gradient) algorithm was chosen due to its higher speed and lower error [9]. Each training period was stopped when the relative error in the same training sequence became less than 10^{-7}.

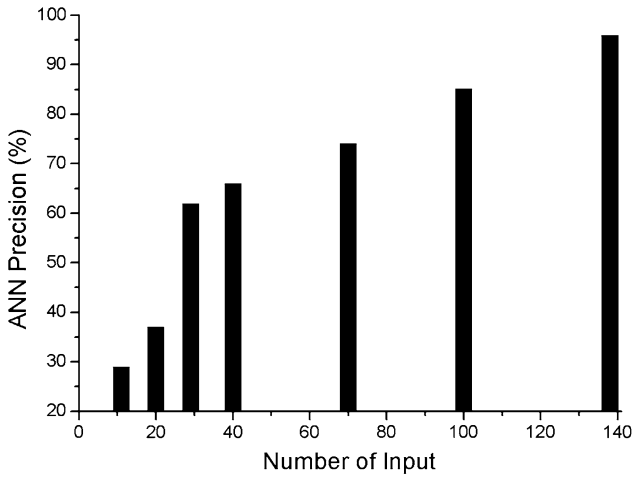

Fig. 4 Precision of ANN response improves with increase in number of cases used for training

Table 1 The results of ANN response to identify elements in the sample

Substance and chemical composition	Desired and network response			
Na_2CO_3	C	O	Na	
Sodium carbonate anhydrous	0.65	0.9994	1	
$C_4H_8N_8O_8$	H	C	N	O
HMX	1	0.247	1	0.999
$Ca(OH)_2$	H	O	Ca	
Calcium hydroxide	0.9999	0.9999	0.9957	
$C_{11}H_{26}O_4N_2$	H	C	N	O
Nylon	1	0.5398	0.598	0.82
CH_4N_2O	H	C	N	O
Carbamide	1	0.1342	0.9851	0.5642
C_5H_8	H	C		
Rubber	1	0.927		
KIO_3	O	K	I	
Potassium iodate	0.9897	0.62	0.9987	
NH_4NO_3 Ammonium nitrate	H	N	O	
	1	0.9579	0.9986	
H_2O	H	O		
Water	1	0.9995		
CO_2	C	O		
Carbon dioxide	0.9976	1		
HNO_3	H	N	O	
Nitric acid	0.9492	0.9901	1	
Al_2O_3	O	Al		
Aluminium oxide	1	0.9763		
$NaO_3H_5C_4$	H	C	O	Na
Oxobutyric acid sodium	1	0.9974	0.9681	0.997
$Pb(HCO)_3$	H	C	O	Pb
Lead hydroxide carbonate	0.437	0.549	1	0.9954
Fe_2O_3	O	Fe		
Iron oxide	0.9377	0.9992		
$CaCO_3$	C	O	Ca	
Calcium carbonate	0.995	0.993	0.958	
Al_2Cu	Al	Cu		
	0.9986	0.9769		
NaPb	Na	Pb		
	1	0.9694		
K_2I	K	I		
	0.9709	1		
CuI	Cu	I		
	0.4261	1		

Results

Twenty different kinds of substances were considered for validation of ANN outputs and their PGNAA were produced by MCNP4C computer code. These substances are listed in Table 1. The result of neural network computation is shown in Figs. 5, 6, 7, 8 for $Ca(OH)_2$, KIO_3, Na_2CO_3 and $Pb(HCO)_3$ respectively. In this figure, the neural network response is drawn as function of atomic number of the element existing in the sample being analyzed. As an example, in Fig. 5 the presence of lines on numbers 1, 8, and 20 indicates the existence of Hydrogen, Oxygen and Calcium in the sample. For this case, the gamma ray spectrum of $Ca(OH)_2$ has been fed to the input of neural network. As another example, the ANN response to gamma ray spectrum of $Pb(HCO)_3$ are 0.437, 0.543, 1, 0.9954 for H, C, O and Pb respectively. Since all the mentioned values are greater than 0.4, the ANN response shows the presence of these elements in the sample. The expected response along with the real response of the network for every specific substance is included in Table 1.

In Table 1, it is observed that for HMX and for Carbamide the ANN was unable to detect Carbon. This is due to the fact that the single peak of this element at 4.4 MeV is hard to detect in the resulting complex gamma ray spectrum. It is expected that with more training, this deficiency will be resolved.

Discussion and conclusion

An artificial neural network has been proposed for analyzing complicated gamma ray spectra from PGNAA

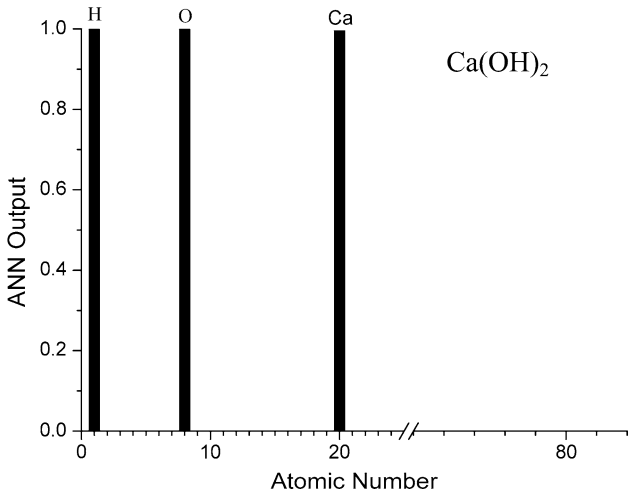

Fig. 5 ANN response to identify elements in $Ca(OH)_2$

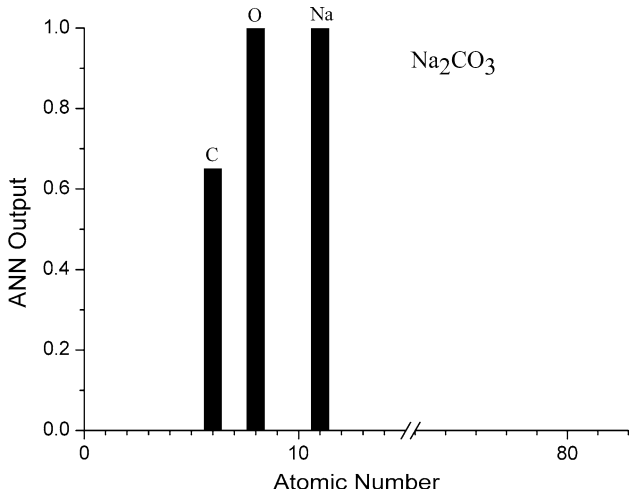

Fig. 7 ANN response to identify elements in Na_2CO_3

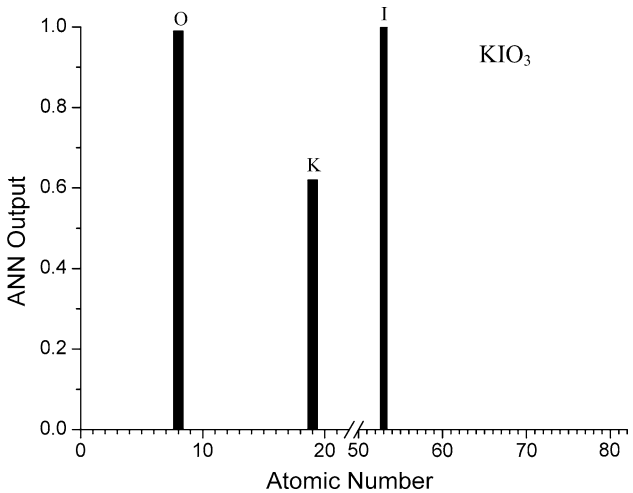

Fig. 6 ANN response to identify elements in KIO_3

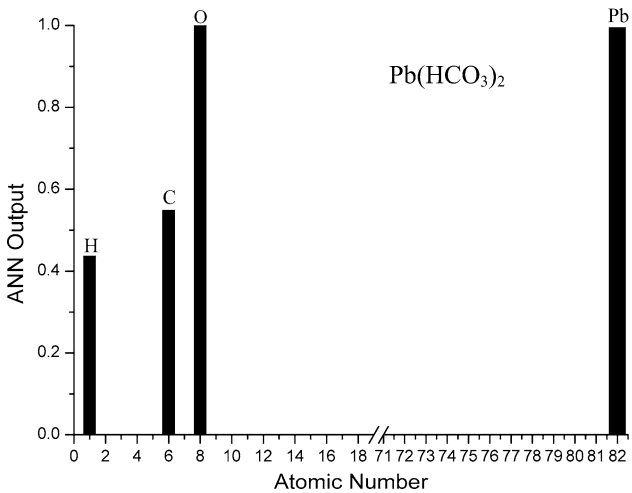

Fig. 8 ANN response to identify elements in $Pb(HCO)_3$

experiments. In the present work, feasibility of training and application of a neural network with 1,024 inputs and 82 outputs was shown in elemental analysis of chemical samples. Gamma ray spectra representing various samples were simulated using the MCNP code. The neural network was able to sense correctly the existence of elements in the sample in 96% of cases. To enhance the reliability of this neural network, it is recommended to train it with more inputs and more elements. Improved resolution in the input spectra due to the utilization of high-resolution detectors might be considered and a more realistic simulation of the experimental geometry would be helpful to obtain better results.

References

1. Dorsey DJ, Hebner R, Charlton WS (2005) J Radioanal Nucl Chem 265:315
2. Miyoshi M, Shimono M, Hasenaka T, Sano T, Fukuoka T (2008) J Radioanal Nucl Chem 278:343
3. Giles IS, Peisach M (1976) J Radioanal Nucl Chem 32:105
4. Wilde HR, Herzog W (1982) J Radioanal Nucl Chem 71:253
5. Alfassi ZB, Chung C (1995) Prompt gamma neutron activation analysis. CRC Press, Boca Raton, FL
6. Nunes WV (2002) Appl Radiat Isot 56:937–943
7. Yoshida E et al (2002) Nucl Instrum Methods A 484:557
8. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, Englewood Cliffs, NJ
9. Matlab (2008) Version 7.6, Mathwork Inc., Help files