Received: 17 July 2018 | Revised: 10 October 2018 | Accepted: 4 December 2018

DOI: 10.1002/xrs.2996

REVIEW

WILEY

Application of artificial neural networks to X-ray fluorescence spectrum analysis

Fei Li[1,2] | Zhixing Gu[1,2] | Liangquan Ge[2] | Di Sun[2] | Xutao Deng[2] | Shun Wang[2] | Bo Hu[2] | Jingru Xu[2]

[1]Applied Nuclear Technology in Geosciences Key Laboratory of Sichuan Province, Chengdu, China

[2]College of Nuclear Technology and Automation Engineering, Chengdu University of Technology, Chengdu, China

Correspondence
Zhixing Gu, Applied Nuclear Technology in Geosciences Key Laboratory of Sichuan Province, Chengdu, China.
Email: 820500585@qq.com

Funding information
Provincial Key Lab of Applied Nuclear Techniques in Geosciences, Grant/Award Numbers: gnzds2018005 and gnzds201800; National Key R&D Project, Grant/Award Number: 2017YFC0602100

X-ray fluorescence (XRF) is widely applied as a mature nondestructive testing method, and appropriate improvement of quantitative analysis methods can improve the accuracy of XRF. Artificial neural network is an intelligent information processing system, its developments and application in XRF are reviewed, and representative models (back propagation, radial basis function, genetic algorithm artificial neural network, and others) are discussed in more details in overfitting, generalization, and algorithm efficiency. Potential directions of developing artificial neural network applied in XRF are proposed in this review as a further study.

1 | INTRODUCTION

Since 1950, X-ray researchers have taken advantage of synchrotrons' copious production of continuous X-ray spectra, and the combination of X-rays with electronics has led to a cornucopia of highly efficient analytical tools.[1] In 1983, Kikkert demonstrated that X-ray fluorescence (XRF) is a mature method for element component analysis.[2] In XRF, the electrons in the inner shell are bombarded by photons, electrons, protons, alpha particles, or heavy ions with certain energy to form electron vacancy; electrons in high-energy shell migrate to fill the corresponding vacancy; and X-ray can be emitted. Element in the sample can be determined by measuring the wavelength or energy of the characteristic X-ray, whereas the concentration of element can be calculated through the determination of characteristic X-ray intensity, ionization cross section, fluorescence yield, etc.[3–5] Due to the matrix effect, scattering background, and electronics noise, it is a complex problem to calculate concentration of element; therefore, accurately quantitative analysis becomes one of the core research directions in XRF. The main XRF quantitative analysis methods can be considered as two categories, semiquantitative analysis methods based on XRF physical mechanism (fundamental parameters approach, fundamental approach, theory influence coefficient method, etc.)[6–8] and quantitative analysis methods based on principle of statistics (partial least square [PLS], artificial neural network (ANN), wavelet analysis, and other chemometric methods).[9–11] ANN is to use mathematical or computer models to simulate the brain's processing of information based on a large number of interconnected neurons[12]; it has strong nonlinear processing ability, anti-interference ability, high parallelism, autonomous learning ability, and generalization ability. The vital ability of ANN is nonlinear processing; ANN calculates the weighted sum of hidden layer neurons through the data of input layer and uses the nonlinear activation function to connect neurons of hidden layer and output layer to

achieve the nonlinear mapping.[13,14] ANN establishes a reasonable topology, and the predicted result is more accurately.[15–17] At least 30 different types of ANN models have been developed,[18] and the most familiar models in XRF are back propagation (BP), radial basis function (RBF), self-generating neural network (SGNN), and Hopfield network. Although these models have favorable nonlinear data processing capability, there are focus of improvement in overfitting problem, generalization ability, and iteration efficiency.

In this review, we mainly demonstrated several kinds of typical ANN models (variants) to overcome the challenges in quantitative analysis applied in XRF, such as Self-improving Segmented Particle Swarm Optimization (SPSO) Adaptive Neural Network (SANN), Deep Sparse Auto-encoder Neural Network (DSAENN), Single Component Prediction Based On Backward Error Propagation (SCP-BEP), Multiple Component Prediction Based On Backward Error Propagation (MCP-BEP), Backward Propagation Network Model Based On Principal Component Analysis (PCA-BP), Genetic Algorithm Backward Propagation (GA-BP), Mind Evolutionary Algorithm Backward Propagation (MEA-BP), and Probabilistic Neural Network (PNN).[19–24] We did not discuss all available models in this field; instead, the focus of this review is on overfitting problem, generalization ability, and iteration efficiency. Representative samples are discussed in more details because of their reliability and applicability.

2 | OVERFITTING

For machine learning, overfitting refers to a model that violates the Occam's razor principle, that is, the use of models (procedures) which include more terms than necessary or use more complicated approaches than necessary[25]; in addition, they are not irreplaceable conditions. In mathematical, a hypothesis space H is given, a hypothesis h belongs to H, and if there are other hypotheses h' belonging to H, the error rate of h in the training example is smaller than h', but in the distribution of the whole instance the error rate of h' is smaller than h, then it is assumed that h makes the training data overfitting.[26] An important improvement to neural network algorithms is to avoid overfitting. In XRF, overfitting has always been the main reason for the accuracy of prediction. The most obvious consequence of overfitting is the poor performance of the validation dataset. In addition, overfitting functions may require collecting redundant data, lead to higher error prone and increase workload, especially if each individual piece of information must be collected through manual observation and manual data entry. For more complex programs, overfitting can make its modeling more difficult to achieve.[25] Overfitting of neural networks is caused by excessive hidden nodes, errors in measurement data, large detector noise, weak component correlation, and excessive connection weights. In some systems, the fitted states may not be synchronized.[26,27]

In response to this problem, Poggio proposed a weight decay method, Holmstrom proposed a noise injection method, and Haykin proposed a cross-validation method.[28–30] The other methods are optimized approximation algorithm, error regularization, etc.[31,32]

2.1 | Early stopping

Among these improved methods, the most popular one should be the variant of the cross-validation method, the early stopping method. The early stop method divides the data set into a training set and a smaller verification set to obtain a continuous estimate of generalization performance.[33] But judging when to stop the training process is not simple. In 1998, Prechlet proposed three types of stopping criteria to solve this problem to avoid the inadequacy and subjectivity of stopping training.[34] Rowinski, P. M.; Piotrowski, A. P.; and Napiorkowski, J. J. suggested using the Generalization Loss class method proposed by Prechlet to terminate training when the test error exceeds 20% of the minimum value proposed in advance.[35,36]

M. I. Kaniu used the method proposed by H. Demuth in the application of energy-dispersive X-ray fluorescence (EDXRF) and scattering assessment for soil quality in 2012; when the neural network test error increases, the training process is automatically stopped to minimize overfitting.[37–39] The experimental results show that the method can better overcome the degree of overfitting and give accurate results. In this experiment, glycerol (a simulate of *organic* soil solution) and kaolin (a model clay soil) doped with soil micronutrients (Fe, Cu, and Zn) and macronutrients (NO_3^-, SO_4^{2-}, and $H_2PO_4^-$) nutrients were used to train the calibration model used. They use a BP neural network consisting of an input layer, an implicit layer, and an output layer. It is obvious that the scatter-analyte (low Z) concentration relationship is nonlinear. The results in Table 1 show the comparisons of PLS to ANN. Apart from Zn, all the other components are analyzed more accuracy by ANN.

The results in Table 2 show that only the accuracy of Mg with standard error of prediction (SEP) of 0.08%, Fe with SEP of 4.02 μg/g, and Cu with SEP of 0.88 μg/g is better than PLS. However, it also performs well in predicting these components, and the result also indicates that the combination of ANN and PLS is more accuracy.

TABLE 1 Summary of PLS and ANN nutrient (SQIs) prediction performances for simulate soil validation samples

SQI	Atomic number (*Z*)	Chemometric technique used	SEP	R^2
NO_3^- (%)	8a	PLS	0.15	0.939
		ANN	0.09	0.998
$H_2PO_4^-$ (%)	11a	PLS	0.70	0.729
		ANN	0.05	0.997
SO_4^{2-} (%)	12a	PLS	0.71	0.956
		ANN	0.16	0.969
Fe (μg/g)	26	PLS	43.5	0.961
		ANN	18.2	0.995
Cu (μg/g)	29	PLS	58.2	0.912
		ANN	4.5	0.999
Zn (μg/g)	30	PLS	19.0	0.985
		ANN	26.6	0.930

Note. ANN, artificial neural network; PLS, partial least square; SEP, standard error of prediction.

TABLE 2 Summary of PLS and ANN nutrient (SQI) prediction performances for field soil test samples

SQI	Atomic number (*Z*)	Chemometric technique used	SEP	R^2
Ca (%)	6	PLS	0.05	0.928
		ANN	0.83	0.847
N (%)	7	PLS	0.01	0.969
		ANN	0.02	0.800
Na (%)	11	PLS	0.01	0.977
		ANN	0.02	0.796
Mg (%)	12	PLS	0.21	0.913
		ANN	0.08	0.991
P (μg/g)	15	PLS	1.98	0.982
		ANN	6.70	0.920
Fe (μg/g)	26	PLS	5.85	0.937
		ANN	4.02	0.956
Cu (μg/g)	29	PLS	0.93	0.847
		ANN	0.88	0.874
Zn (μg/g)	30	PLS	0.70	0.955
		ANN	0.66	0.534

Note. ANN, artificial neural network; PCR, principle component regression; PLS, partial least square; RRMSD, root-mean-square difference.

2.2 | Noise injection

Noise injection method is to add noise to the input data while training the network. In 1991, Sietsma and Dow confirmed that adding noise can improve the generalization ability of neural networks.[40] In practical applications, the variants proposed by Holmstrom and Koistinen[29] that add Gaussian noise to the input data are the most acceptable. Jitter is a concrete way to add noise, and L. Holmström and Y. Grandvalet theoretically explored the effectiveness of the method.[41,42] In 2004 R. M. Zur in their research mentioned two Jitter methods: the first method is to add random noise vector to each training data during iterations and the other is to use a number of random vectors to extend the training data set before training. Their research results showed that to a certain extent, training with a single extended training data set can actually improve the performance of neural networks.[43] Reed theoretically studied the close similarity between noise injection and other methods (including error regularization) for improving the generalization characteristics of ANNs in 1995.[44] In 2008 and 2010, Vincent has applied this method by adding noise into input units of an autoencoder, and under this operation the neural network reconstructs the noise-free input.[45,46] Van der Maaten also explored deterministic normalizers corresponding to different index family noise distributions in 2013, but there are no hidden layers in their models.[47]

In 2014, Nitish Srivastava at the University of Toronto proposed the dropout neural network structure when studying the problem of overfitting of deep neural networks.[48] The essence of this method can be explained as adding noise to the hidden units of the neural network, which can be considered as a form of model averaging. The key to this approach is to randomly drop cells including their connections during training to prevent overcompatibility of the cells. During the training process, the samples are rejected from the exponential number of thin networks. During the training process, dropout samples from an exponential number of *thinned* networks. Although in the test, by simply averaging the predictions of all thin networks by using another thinned network with smaller weights, the approximation of the effect is easily obtained. Research shows that this method effectively reduces overfitting. In general, it is optimal to reject 20% of input units and 50% of hidden units.

It is worth that a clear average of the predictions of many refinement models at the exponential level is not feasible. Here an approximate averaging method is used for the dropout neural network. The method takes a unit that is retained by the probability p during training. At the time of the test, the output weights of that unit are multiplied by p. This method ensures that for any hidden unit, the expected output (under the distribution used to drop units at training time) is the same as the actual output at the time of the test. When training a drop network, this approximate averaging method is used at test time, and the generalized error can be significantly reduced on various classification problems compared with other regularization methods.