The application of artificial neural networks to the inversion of the positron lifetime spectrum*

An Ran(安 然), Zhang Jie(张 杰), Kong Wei(孔 伟), and Ye Bang-Jiao(叶邦角)†

Department of modern physics, University of Science and Technology of China, Hefei 230026, *China*

(Received 29 November 2011; revised manuscript received 22 June 2012)

A new method of processing positron annihilation lifetime spectra is proposed. It is based on an artificial neural network (ANN)-back propagation network (BPN). By using data from simulated positron lifetime spectra which are generated by a simulation program and tested by other analysis programs, the BPN can be trained to extract lifetime and intensity from a positron annihilation lifetime spectrum as an input. In principle, the method has the potential to unfold an unknown number of lifetimes and their intensities from a measured spectrum. So far, only a proof-of-principle type preliminary investigation was made by unfolding three or four discrete lifetimes. The present study aims to design the network. Besides, the performance of this method requires both the accurate design of the BPN structure and a long training time. In addition, the performance of the method in practical applications is dependent on the quality of the simulation model. However, the chances of satisfying the above criteria appear to be high. When appropriately developed, a trained network could be a very efficient alternative to the existing methods, with a very short identification time. We have used the artificial neural network codes to analyze data such as the positron lifetime spectra for single crystal materials and monocrystalline silicon. Some meaningful results are obtained.

Keywords: positron lifetime spectrum, neural network

PACS: 78,70.Bj, 29.85.Fj **DOI:** 10.1088/1674-1056/21/11/117803

1. Introduction

Positron annihilation technology (PAT) is a sensitive and non-destructive detection means in studying the micro-defects of materials.[1−5] Inversion of discrete positron lifetimes and their distributions, as well as the corresponding amplitudes from positron lifetime spectra are traditionally solved by algorithmic procedures such as POSITRONFIT,[6−8] CONTIN,[9,10] MELT,[11−15] and PAScual.[16,17] These programs use an arithmetic based on the statistical minimization principle or the maximum entropy principle.

An alternative methodology that has been widely used to solve several types of physical chemistry calculation problems is called an artificial neural network (ANN). By using data from simulated positron spectra which are created by a simulation program (PAScual), an ANN can be trained with various parameters, such as lifetime and intensity. One advantage of this method is, as usual for ANN, that once the network is trained, the identification procedure is very fast when many spectra are processed. In its most advanced form, the number of lifetimes and distributions do not need to be known in advance, and it can be determined by the program itself. However, existing studies show that to achieve the same flexibility with MELT, i.e., to identify an unknown number of distributions, the network and the training must be very carefully designed.

2. General principles

Determining parameters of a distribution from measured spectra is a so-called inverse task. It is very seldom that an inverse task can be solved analytically in an exact manner by inverting the relationship. Even so, the inversion is seldom unique, and thus some extra criteria need to be used to select the true solution. In addition, a measured distribution often contains fluctuations and measurement errors. Thus, the inversion also needs to be the most likely one and needs to be optimized in some sense.

An artificial neural network represents a powerful method for dealing with such problems.[17] In particular, they are very effective in extracting parameters from a distribution which is a non-linear function of

*Project supported by the National Natural Science Foundation of China (Grant Nos. 10835006 and 10975133).

†Corresponding author. E-mail: bjye@ustc.edu.cn

© 2012 Chinese Physical Society and IOP Publishing Ltd

the searched parameters, and also contains noise. In order to achieve this, the network needs to be trained with a large number of training samples, i.e., realizations of the distribution where the searched parameters are also known. We do not need to know an actual analytical relationship connecting the parameters with the distributions at this stage, except numerical values of the training samples (input and output).

In general, in the event of a positron annihilation lifetime spectroscopy (PALS), a measurable spectrum consists of three (or more) exponential decay components, each of which has a different intensity and lifetime(I_i, τ_i). As discussed in several literatures,[12,17] one component of a measured spectrum $y(t)$ can be written as

$$y(t) = N_0 R(t) \int_0^\infty \frac{f(\tau)}{\tau} \exp\left(-\frac{t}{\tau}\right) \mathrm{d}\tau + B, \quad (1)$$

where $R(t)$ is the detector resolution function which is usually assumed to be a quasi-Gaussian function convoluted with the measured signal, B is the background, and $f(\tau)$ is the positron lifetime profile which can be expressed as

$$f(\tau) = \sum_{i=1}^{3} I_i \delta(\tau - \tau_i), \quad (2)$$

where I_i is the intensity, and τ_i is the lifetime of the $i-$th component. In more general cases, any of the three lifetimes can display a variation around the mean value, which enables us to use a narrow Gaussian distribution instead of the distribution given by Eq. (2), i.e.,

$$f(\tau) = \sum_{i=1}^{3} \frac{I_i}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(\tau - \tau_i)^2}{2\sigma_i^2}\right]. \quad (3)$$

As mentioned above, the fourth lifetime constant τ_4 can be present sometimes.

To determine the unknown intensities and lifetime parameters (which form a PALS spectrum) by means of a neural network is through the use of a large number of spectra with known intensities and lifetimes to train samples. The training samples for each pair of intensity and lifetime parameters must be different. That is to say, the training samples are required to be chosen to cover the whole region of possible values in the application of the trained network. Obviously, it is impractical to use measurements as training samples. However, the samples can be very easily generated by a simulation program such as PAScual. The program has also been used to test various algorithms.[17]

3. The PALS spectrum generator

The simulation lifetime spectra were generated with an algorithm based on Eqs. (1) and (3). We use the PAScual program to produce a batch of positron annihilation spectra, and test them with another program: Lifetime 9. It seems that all simulation spectra are credible.

In this paper, the extent of the variation of the intensities and lifetimes were set as: $I_1 = 0.25$, $\tau_1 = 0.130$ ns, $I_2 = 0.45$, $\tau_2 = 0.35$ ns, $I_3 = 0.32$, and $\tau_3 = 2.5$ ns. The range of the parameters is described in detail in Table 1. The total number of counts was taken to be 10 million. In theory, these parameters could also be changed during the training, so as to make the trained network capable of interpreting data from various measurements with different number of counts. However, we only check the algorithm with simulated data in the recall phase, and can choose the number of counts to be 10 million also for the test patterns. Then, data for training can be collected to test the network. Finally, a background and random noise are added to the simulated counts in each channel. A channel width of 24.5 ps was similarly chosen as other reported simulation studies and measurements. An example of a simulated spectrum is shown in Fig. 1.

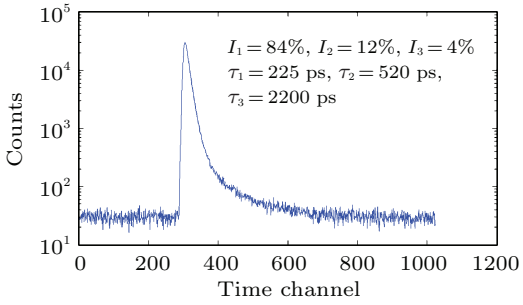


Fig. 1. (colour online) Simulated positron lifetime spectrum.

4. Network structure and training

We have used a simple three-layered feed-forward network with a backward error propagation-Back propagation network. The principles of such networks are described in standard textbooks and review articles.[18] The structure of the network is shown in Fig. 2. The number of the input and output nodes is defined by the number of input and output data; the number of the nodes in the hidden layers is a random

parameter. The best value in each case can be found by trial and error. In the present situation, the input networks contain the PALS spectrum data which has the same channel length. In this work, 1024 channels were used. The number of the output nodes is equal to six, i.e., three intensities and three lifetime constants. The number of nodes in the hidden layers is chosen to be 20, 30, 40, 50, 60, and 70.

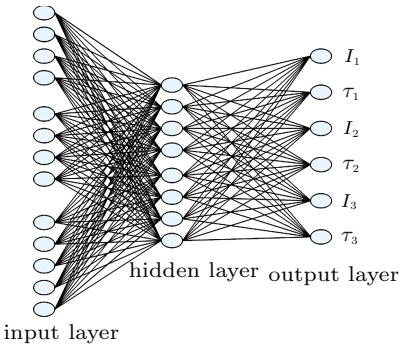


Fig. 2. Structure of the neural network for three lifetimes and three intensities.

We trained 920 samples, and different spectra with different intensities and lifetimes were used in a random arrangement until the root-mean-square network error is below the desired limit of 0.04%. The root-mean-square error is defined as the squared sum of deviations of each actual node output from the correct ("target") value, which serves as a measure of the training network. The precision of the network will be comparable to the root-mean-square value of the training.

It is seen that in these exploration tests, we have used a root-mean-square value which leads to a better precision than the usual traditional algorithms. The reason for this is that to achieve better precision, the networks need more smaller root-mean-square values and more training samples, more training epochs, smaller training error goals, and a much longer training time.[18] At this stage it was clear that, due to the large size of the networks, i.e., 1024 input nodes, the training time is quite long.

We construct six neural networks which are different from the number of hidden layer neurons. The number of hidden layer neurons is 20, 30, 40, 50, 60, and 70. After the training is over, i.e., the root-mean-square value decreases to lower than 0.04%, another 100 samples were produced by PAScual which was then used to test the performance of the network.

5. Results

The results are summarized in Table 1 and Fig. 3. In both cases, the relative error is used to describe network performance. In Table 1, the relative standard deviation is given, which is calculated based on the results of all test samples using the traditional programs. In Fig. 3, the relative error is shown for each test sample as a function of the parameters to be determined. Since the lifetime spectra contain physical information, the errors regarding to the lifetimes are shown.

It is seen that the relative standard deviation of the number of hidden layer neurons (20, 30, 40, and 50) is larger than that obtained from traditional algorithms. However, when the number of hidden layer neurons is 60 and 70, the relative standard deviation is smaller than the traditional algorithms. In addition, when the number of the hidden layer neurons is 60, the networks can have a better performance. If the number of the hidden layer neurons is too large, the networks will overfit.[19] This results in the capability that a network with a suitable number of hidden layer neurons can provide a smaller error in this situation. If we set

Table 1. Parameters of all simulated positron lifetime spectra. The number of output-input training patterns is 920, and the number of test patterns is 100. The desired target limit for the training root-mean-square error was 0.04%. The number of the hidden layer neurons for average errors 1–6 is 20, 30, 40, 50, 60, and 70, respectively.

	I_1/%	τ_1/ns	I_2/%	τ_2/ns	I_3/%	τ_3/ns
Average error 1	1.62	10.71	1.07	3.37	10.34	1.18
Average error 2	11.47	10.24	2.04	4.48	8.25	0.56
Average error 3	1.39	11.03	1.96	5.64	1.51	0.55
Average error 4	17.69	10.44	3.52	0.28	7.72	1.6
Average error 5	4.84	0.07	1.60	0.68	6.73	1.21
Average error 6	7.51	0.28	1.95	0.13	4.03	0.83

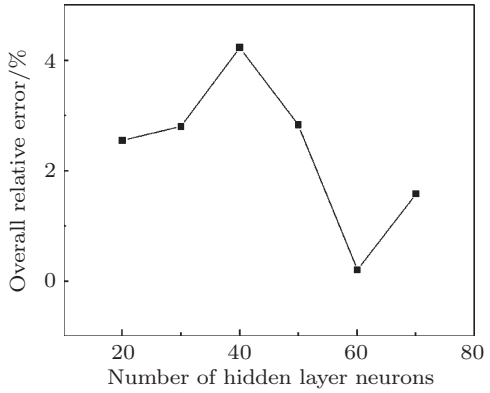


Fig. 3. Overall relative error versus the number of hidden layer neurons.

a smaller training error and more training epochs, the networks can reach any desired level that is comparable with current methods. In this direction, applications of both faster training methods and a parallel algorithm are planned.

Besides the simulated positron spectra, we also use the networks to inverse the real positron annihilation lifetime spectra such as a lifetime spectrum of silicon single crystal structure material. We have 30 spectra of silicon to analyze. We use the Lifetime9 program to analyze the silicon spectra recorded by our apparatus. The distribution from the intensities and lifetimes is shown in Table 2. Then, we use the trained networks to inverse these spectra, and the results are also shown in Table 2. We can see that the network result is a little larger than that from Lifetime9. The results demonstrate that the network can analyze single-crystal-structure material, but we should use more samples to train the networks to obtain more accurate results.

Table 2. Distributions of the lifetime and intensity obtained from the positron spectrum measured for silicon.

	I_1	τ_1/ps	I_2	τ_2/ps	I_3	τ_3/ps
Lifetime9	83.9%	213.2	15.5%	383.2	0.660%	1.795
network	84.1%	216.8	15.1%	408.9	0.860%	2.048

6. Conclusions

An extrapolation of the results shown here indicates that several factors determine the intensities and lifetimes, and the ANN training procedure can be very stable. It can also be used to extract three lifetimes and three intensities without prior knowledge of the number of lifetimes present in the measurements. Determination of the lifetime distributions has not been as successful as the maximum entropy program MELT.

The largest advantage of the ANN is expected from its extreme speed in identification. It takes a very long time to train a network, and the running time is only one epoch of millisecond. Thus, if a large number of unfolding procedures need to be made, the ANN-based identification procedure may show significant advantages. We have already used the ANN code to analyze single-crystal materials such as silicon and iron. The results show that this method is effective.

References

[1] Schultz P J and Lynn K G 1988 *Rev. Mod. Phys.* **60** 701
[2] Puska M J and Nieminen R M 1994 *Rev. Mod. Phys.* **66** 841
[3] Chen X L, Weng H M, Xi C Y and Ye B J 2007 *Acta Phys. Sin.* **56** 6695 (in Chinese)
[4] Xu H X, Hao Y P, Han R D, Wen H M, Du H J and Ye B J 2011 *Acta Phys. Sin.* **60** 067803
[5] Beling C D and Hu Y F 2005 *Chin. Phys. Lett.* **14** 2293
[6] Eldrup M and Kirkegaa P 1974 *Comput. Phys. Commun.* **7** 401
[7] Eldrup M, Lightbody D and Sherwood J N 1981 *Chem. Phys.* **63** 51
[8] Kansy J 1996 *Nucl. Instrum. Methods Phys. Res. Sect.* A **374** 235
[9] Kirkegaa P and Eldrup M 1972 *Comput. Phys. Commun.* **3** 240
[10] Provencher S W 1982 *Comput. Phys. Commun.* **27** 213
[11] Gregory R B and Zhu Y K 1992 *Nucl. Instrum. Methods Phys. Res. Sect.* A **290** 172
[12] Wang C L, Hirade T and Maurer F H J 1998 *J. Chem. Phys.* **108** 4654
[13] Shukla A, Peter M and Hoffmann L 1993 *Nucl. Instrum. Methods Phys. Res. Sect.* A **335** 310
[14] Wang C L and Maurer F H J 1996 *Macromolecules* **29** 8249
[15] Hoffmann L, Shukla A, Peter M, Barbiellini B and Manuel A A 1993 *Nucl. Instrum. Methods Phys. Res. Sect.* A **335** 276
[16] Pascual-Izarra C, Dong A W and Pas S J 2009 *Nucl. Instrum. Methods Phys. Res. Sect.* A **603** 456
[17] Pham B, Guagliardo P and Williams J 2011 *J. Phys. Conf. Ser.* **262** 012048
[18] Hagan M T and Menhaj M B 1994 *IEEE T. Neural. Networ.* **5** 989
[19] Hyvarinen A and Oja E 2000 *Neural Networks* **13** 411