Measurement 46 (2013) 3192–3197

Contents lists available at SciVerse ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

# Construction of consistent neural network empirical physical formulas for detector counts in neutron exit channel selection

Serkan Akkoyun *, Nihat Yildiz

*Department of Physics, Cumhuriyet University, 58140 Sivas, Turkey*

ARTICLE INFO

*Article history:*
Received 19 November 2012
Received in revised form 12 February 2013
Accepted 14 May 2013
Available online 30 May 2013

*Keywords:*
Neural network
Empirical physical formula
Neutron exit channel

ABSTRACT

Proper selection of neutron exit channels following heavy-ion reactions is important in nuclear structure physics. A knowledge of detector counts versus number of neutron interaction points per event can be useful in this selection. In this paper, we constructed layered feedforward neural networks (LFNNs) consistent empirical physical formulas (EPFs) to estimate the detector counts versus number of neutron interaction points per event. The LFNN-EPFs are of explicit mathematical functional form. Therefore, by various suitable operations of mathematical analysis, these LFNN-EPFs can be used to derivate further physical functions which might be potentially relevant to neutron exit channel selection.


## 1. Introduction

Proper selection of neutron exit channels following heavy-ion reactions is important in nuclear physics. Different techniques have been employed to correctly select exit channels [1–6]. In [1–3], neutron detector arrays were used and in [4–6], studies following fusion-evaporation reactions were performed for the selection. Channel selection from heavy-ion reactions allows nuclear spectroscopic studies of very weakly populated nuclei. The measurement of the neutron multiplicities in these reactions is particularly important in the rare-earth region where evaporation of the neutrons is a dominant channel.

In this paper, based on the data obtained by specific Monte Carlo simulations for varying neutron multiplicities, we aimed to select heavy-ion reaction neutron exit channels. In this selection, we observed that knowledge of detector counts versus number of neutron interaction points per event could be useful. The detector we used in this paper was a segmented HPGe planar detector. Note that in our simulations, no ancillary detector was used because the detector we used was alone sufficient to obtain the number of neutron interaction points.

The importance of this paper in neutron exit channel selection is briefly as follows. The physics involved in neutron interactions in the detector is highly nonlinear. Therefore, in many cases it may be difficult to construct explicit form of empirical physical formulas (EPFs) for nonlinear detector count functions. These EPFs would then be used for specific purposes in analyzing detector data. To overcome EPF construction difficulties just mentioned, in this paper we constructed consistent layered feedforward neural network (LFNN) [7] EPFs for counts. The LFNN-EPF construction was solely based on our previous theoretical treatment [8]. The LFNN-EPFs are of explicit mathematical functional form. Therefore, by various suitable operations of mathematical analysis, these LFNN-EPFs can be used to derivate further physical functions which might be potentially relevant to neutron exit channel selection.

## 2. Theories

### 2.1. The LFNN basics and its relevance detector counts EPFs

General LFNN-EPFs is analyzed in depth in our previous work [8]. A number of specific LFNN-EPFs have been reported, including radiation measurement application [9]. Still, here we again give the minimum LFNN fundamentals.

* Corresponding author. Tel.: +90 346 2191010x1413.
*E-mail address:* sakkoyun@cumhuriyet.edu.tr (S. Akkoyun).

0263-2241/$ - see front matter 
http://dx.doi.org/10.1016/j.measurement.2013.05.024

We also mention briefly LFNN-EPFs in general and detector counts LFNN-EPF in particular.

*2.1.1. The LFNN basics*

An artificial neural network (ANN) [7] mimics the brain functionality and consists of interconnecting artificial neurons which have adaptive synaptic weights. The ANN learns new information by weight modification. LFNN, particular kind of ANN, is a one input-many intermediate (hidden)-one output layer device, all layers of which are interconnected by adaptable weights (Fig. 1).

Theoretically speaking, a single hidden layer LFNN is sufficient for excellent nonlinear function approximation [10]. Therefore, in this paper we used single hidden layer LFNNs. Also, for simplicity and without loss of generality; we only explain the single hidden layer LFNN functionality, although in Fig. 1, multi-layer LFNN is illustrated. Borrowing from [10], for a LFNN with single hidden layer, in Fig. 1, the desired output vector $\vec{y}$ is approximated by a network multi-output vector $\vec{f}$ which is defined by the following equation:

$$\vec{f}: R^p \to R^r : \vec{f}_k(\vec{x}) = \sum_{j=1}^{h_1} \beta_{jk} G(A_j(\vec{x})), \vec{x} \in R^p, \beta_{jk} \in R, A_j \in A^p, \text{ and } k = 1, \ldots, r, \tag{1}$$

where $A^p$ is the set of all functions of $R^p \to R$ defined by $A(\vec{x}) = \vec{w} \cdot \vec{x} + b$. '$\cdot$' is the scalar product, $\vec{w}$ is input to hidden layer weight vector, $\vec{x}$ is the LFNN input vector in Fig. 1, and $b$ is the bias weight. In Fig. 1, the columns of the weight matrices $w^1$ and $w^2$ correspond to weight vectors defined in $A(\vec{x})$ and $\vec{\beta}$ in Eq. (1). However, as can be seen from Fig. 1 and Eq. (1), the correspondences $w^1 \to A(\vec{x})$ and $w^2 \to \vec{\beta}$ are valid only for single hidden layer LFNN. For the multi-hidden layer LFNN, both Eq. (1) and the correspondences must be altered accordingly. Another point is that, in Eq. (1), the hidden neuron activation function

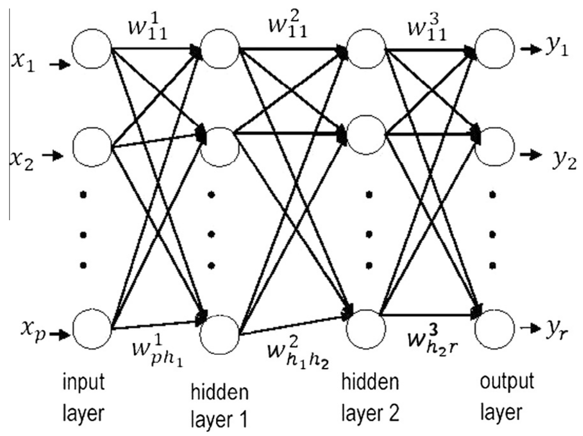


**Fig. 1.** Fully connected one input-many hidden-one output layer LFNN. Only two hidden layers are shown. $x_i(i = 1,\ldots,p)$ and $y_i(i = 1,\ldots,r)$ are, respectively, input and output vector components. Circles: artificial neurons, arrows: adaptable synaptic weights. $w^i_{jk}$: weight vector component, where $i$ is a layer index, $jk$ weight component from the $j$th neuron of $i$th layer and to $k$th neuron of $(i + 1)$th layer. Hidden layer neurons are respectively $h_1$ and $h_2$.

$G:R \to R$ can be theoretically any well-behaved nonlinear function; proving that a LFNN is a universal nonlinear function approximator. In applications, $G$ is frequently chosen as a kind of nonlinear sigmoid function defined by the following equation:

$$G: R \to [0,1] \text{ or } [-1,1], \text{ non-decreasing, } \lim_{\lambda\to\infty} G(\lambda) = 1, \text{ and } \lim_{\lambda\to-\infty} G(\lambda) = 0 \text{ or } -1. \tag{2}$$

By using the LFNN constructed in line with Eqs. (1) and (2), sample train data is simultaneously presented to both input and output layers. By a suitable modification algorithm, the LFNN modifies its weights until an acceptable error level between predicted and desired outputs is attained. Then, by using LFNN of the final weights, the test set performance of the network is tested over a previously unseen data set. If test data predictions are good enough, the LFNN is considered to have consistently learned or generalized the inherent functional relationship existing between input and output data.

*2.1.2. LFNN relevance to the detector counts EPF construction*

Since a deterministic or random EPF is usually a mathematical vector function $\vec{y}: R^p \to R^r$ between the physical variables under investigation, particularly LFNN (not any other ANN) is relevant to EPF construction. Therefore, being a general input–output function estimator, the LFNN defined by Eq. (1) is particularly relevant in this context. But, in physics, although there can be several independent variables [$p \succ 1$ in Eq. (1)], the number of the dependent variables is usually one [$r = 1$ in Eq. (1)]. Train sample data for independent and dependent physical variables are presented to the input and output layers respectively. Then, after a suitable weight adaptation process, the LFNN finally estimates the unknown generally nonlinear EPF. Note that EPF is a general abstract term, and in this paper it is concretely used for the detector counts (see, Section 3.4). It must also be firmly stated that, depending on the number of hidden layer, hidden units and the kind of activation functions etc., we can construct infinitely many LFNN-EPFs, all of which are compatible with Eqs. (1) and (2). But, as shown in [8], in practice between these infinitely many numbers of final approximation functions, any of them can be safely chosen as the desired EPF. Before closing this section, it is useful to point out that the ANN modeling has an obvious superiority over some other well-known statistical curve-fitting techniques, (see, for more [11]).

## 3. LFNN application details

*3.1. The detector simulation data for LFNN-EPF counts*

The LFNN-EPFs were constructed based on the data obtained by the Geant4 [12] Monte Carlo simulation counts resulted from neutron interactions in the detector. Here, we think that it is necessary to explain briefly the reason to construct a neural network based on a procedure like Monte Carlo with an uncertainty in given outputs, can be successful to predict an issue. The Monte Carlo techniques are based on specific statistical distributions. Therefore,