A Way to Aggregate Multilayer Neural Networks

Maciej Krawczak

1 Systems Research Institute, Polish Academy of Sciences, Newelska 6, 01-447 Warsaw, Poland
2 Warsaw School of Information Technology, Newelska 6, 01-447 Warsaw, Poland
krawczak@ibspan.waw.pl

Abstract. In this paper we consider an aggregation way for multilayer neural networks. For this we will use the generalized nets methodology as well as the index matrix operators. The generalized net methodology was developed as a counterpart of Petri nets for modelling discrete event systems. First, a short introduction of these tools is given. Next, three different kinds of neurons aggregation is considered. The application of the index matrix operators allow to developed three different generalized net models. The methodology seems to be a very good tool for knowledge description.

1 Introduction

A multilayer neural network is described by layers and within each layer by a number of neurons. The neurons are connected by weighted links. The network output is strictly related to the presented input, subject to the conditions resulting from the constancy of the structure (the neuron connections), the activation functions as well as the weights.

A neural network can be considered with different degree of aggregation. A case without aggregation means that all layers and all neurons are “visible” for consideration. A second case where neurons within each layer are aggregated and only layers are “visible”. The next case, in which all layers are aggregated, and only inputs and outputs of a neural network are available.

In order to develop these three kinds of aggregation we will apply the methodology of generalized nets introduced by K. Atanassov in various works, e.g. [1], [2], [3], [4] and [5].

The generalized nets methodology is defined as an extension of the ordinary Petri nets and their modifications, but in a different way, namely the relation of places, transitions and characteristics of tokens provide for greater modelling possibilities than the individual types of Petri nets.

In the review and bibliography on generalized nets theory and applications of Radeva, Krawczak and Choy in [10] we can find a list 353 scientific works related to the generalized nets.

1.1 Multilayer Neural Networks Structure

The idea of the aggregation will be described in the following way. The network consists of \( L \) layers; each layer \( l = 0,1,2,\ldots,L \) is composed of \( N(l) \) neurons. By \( N(0) \)
we denote the number of inputs, while by $N(L)$ the number of outputs. The neurons are linked through weighted connections.

The output of the network is strictly related to the presented input, subject to the conditions resulting from the constancy of the structure, the activation functions as well as the weights. The neural network realizes the following mapping:

$$output = NN(input)$$  \(1\)

The neural network consists of neurons described by the activation function as follows

$$x_{pj(l)} = f\left(net_{pj(l)}\right)$$  \(2\)

where

$$net_{pj(l)} = \sum_{i=1}^{N(l-1)} w_{i(l-1)j(l)} \cdot x_{pi(l-1)}$$  \(3\)

while $x_{pi(l-1)}$ denotes the output of the $i$-th neuron with respect to the pattern $p$, $p = 1, 2, ..., P$, and the weight $w_{i(l-1)j(l)}$ connects the $i$-th neuron from the $(l-1)$-st layer with the $j$-th from the $l$-th layer, $j = 1, 2, ..., N(l)$, $l = 1, 2, ..., L$.

The different cases of aggregation determine different streams of information passing through the system.

### 1.2 Generalized Net Modeling

A generalized net contains tokens, which are transferred from place to place. Every token bears some information, which is described by token's characteristic, and any token enters the net with an initial characteristic. After passing a transition the tokens' characteristics are modified. The places are marked by $\bigcirc$, and the transitions by $\uparrow$.

The transition has input and output places, as shown in Fig. 1.

The basic difference between generalized nets and the ordinary Petri nets is the place – transition relation (Atanassov, 1991).

Formally, every transition is described by a seven-tuple

$$Z = \langle L', L'^*, t_1, t_2, r, M, \square \rangle$$  \(4\)

where: $L' = \{l'_1, l'_2, ..., l'_m\}$ is a finite, non-empty set of the transition's input places, $L'^* = \{l'^*_1, l'^*_2, ..., l'^*_m\}$ is a finite, non-empty set of the transition's output places, $t_1$ is the current time of the transition's firing, $t_2$ is the current duration of the transition active state, $r$ is the transition's condition determining which tokens will pass the transition, $M$ is an index matrix of the capacities of transition's arcs, $\square$ is an object of a form similar to a Boolean expression, for true value the transition becomes active.