VIA MEDICA

Vol. 7, No. 1, pp. 58–67

ISSN 1506–9680

Review

Artificial neural networks in nuclear medicine

Dariusz Świetlik[1], Tomasz Bandurski[1], Piotr Lass[2]

[1]Laboratory of Radiological Informatics Medical University, Gdańsk, Poland
[2]Department of Nuclear Medicine, Medical University, Gdańsk, Poland

[Received 27 IV 04; Accepted 12 V 04]

Abstract

An analysis of the accessible literature on the diagnostic applicability of artificial neural networks in coronary artery disease and pulmonary embolism appears to be comparative to the diagnosis of experienced doctors dealing with nuclear medicine. Differences in the employed models of artificial neural networks indicate a constant search for the most optimal parameters, which could guarantee the ultimate accuracy in neural network activity. The diagnostic potential within systems containing artificial neural networks proves this calculation tool to be an independent or/and an additional device for supporting a doctor's diagnosis of artery disease and pulmonary embolism.

Introduction

Artificial Intelligence (AI) is a domain of science dealing with the process of modeling and solving estimation problems and it consists of the following: Artificial Neural Networks (ANN), fuzzy logic, genetic algorithms and expert systems. The idea of AI was derived from a biological inspiration i.e. ANN are derived from human brain perception and reasoning, genetic algorithms imitate the process of natural evolution, while expert systems counterfeit human expertise in a given field of knowledge.

The most frequently used method of AI in the field of medicine is the Artificial Neural Networks. ANN is used in various domains of medicine including the classification of objects and signals [1], the clustering of signals [2], optimization [3], biological systems modeling [4], process control [5] diagnostics [6] and data mining [7]. Such a broad spectrum of possibilities in the implementation of ANN makes it a valuable multipurpose arithmetical instrument.

As they are reliable in clinical decisions AI techniques have turned out to be helpful in modern nuclear medicine. The methods which are most commonly used are ANN [8–10] and expert systems [8, 11, 12]. ANN is helpful in the diagnostic process of coronary artery disease [13, 14] pulmonary embolism [15, 16], Alzheimer' disease [17, 18], parathyroid adenoma [19], diffuse parenchymal liver disease [20], renovascular hypertension [21]. Moreover, it is beneficial in the reconstruction of SPECT data as well as in the methods of gamma ray scatter correction [23, 24].

Usually the nature of biological processes which take place in living organisms is not linear. ANN offers the possibility of the reconstruction and projection modeling of these processes. In addiction to this, ANN enables mathematical modeling of not clearly understood biological phenomena such as Alzheimer's disease etiology.

The success of ANN is a result of its uncomplicated usage as a convenient instrument. The users' task is simply to gather representative data and start the learning algorithm. ANN automatically forms an adequate structure of data upon which it generates a final result.

A very interesting quality of ANN is its imitation of brain function (to some extent). This is based on the fundamental mechanism of the biological activity of nervous cells as well as their connections and interactions. On the other hand the structure and activity of ANN may be considered also as experimental model of human brain function [26].

The selected features of ANN demonstrate the discipline of AI as a state of the art method of solving problems in which analysis and transformation of data are fundamental.

Correspondence to: Dariusz Świetlik
Laboratory of Radiological Informatics
Medical University of Gdańsk
ul. Tuwima 15, 80–211 Gdańsk, Poland
e-mail: dswietlik@op.pl

Artificial neural networks

Basics

ANN is either a technical device or an algorithm which task resembles living nerve cell function. The technical apparatus accomplishing ANN is built using an electromechanical system or an electronic circuit. The ANN algorithm on the other hand is a computer program that simulates the function of such device.

ANN is usually composed of a connected group of single elements called neurons forming layers. Each of these elements has a certain number of entries and exits. The exits of one group of neurons are connected with the entries of another group. The relations between the entries and exits are modified for every neuron alone during the process of learning. An already learned ANN transforms a piece of information by its duplication among particular elements with respect to the modifications created during the learning process. The created correlations between the input and output signals may be considered as the solution of the given problem.

ANN as being a method modeled on nature itself, imitates certain biological features of the nervous system, such as the resistance to damage and the ability of learning.

History of ANN

The domain of ANN began in 1943 with McCulloch and Pitts' description of a mathematical model of a nerve cell. The publication also contained the basic rules of information processing representation in the artificial neuron [27]. In 1958, the existing technical implementations of the ANN issue were completed and officially presented [28]. The subject of biological nervous system as the inspiration of ANN researchers was presented by Taylor [29].

The first operational implementation of ANN called the Perceptron was discussed by Rosenblatt [30]. The structure of the network was formed by electromechanical and electronic elements and its task was alphanumerical pattern recognition. The publication of this project started the rapid development of an ANN resembling Perceptron based on Rosenblatt's model. An interesting achievement in the Perceptron based realization of ANN is a network consisting of a set of ADALINE elements (Adaptive Linear Element) combined into MADALINE (Many Adaline) [31, 32]. The MADALINE network was engaged in the signal processing of many electronic sources such as radar, sonar and phone lines.

Finally, in 1969, Minsky and Papert proved the limitations of one-layer Perceptron based networks [33]. The publication resulted in the slowing down of the ANN development and implementation rate, but on the other hand a few new ideas and methods emerged at that time, such as the associative networks concept and the back propagation algorithm [34].

Beginning from the mid 1980s, rapid progress in the field of ANN took place. At present, there is a constant increase in ANN implementation in different branches of science and technology.

The structure of the artificial neuron

The elemental unit of the artificial neural network is the artificial neuron (Fig. 1A). It has certain features of a living nervous cell which in turn forms the nervous system (Fig. 1A).

The nervous cell is formed by a body to which information is conducted by its ramifications called dendrites. An axon, which is a single fibre emerging from the body of a neuron, serves as an exit route for information and is connected with other neurons' dendrites through synapses.

In an artificial neuron, the analogous structure to the body of a biological cell is a summator, while an axon hillock is similar to an activation block. The numbers x_1, x_2, x_m (Fig. 1B) stand for an m number of entrance signals that represent dendrites. Every x_m value is inserted into an artificial neuron together with a numerical coefficient for an m-successive entrance called a w_m weight. The prod-

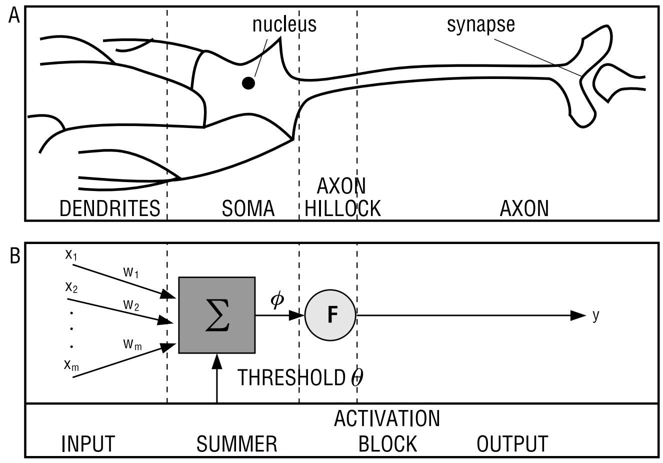

Figure 1 A. A biological neuron; **B.** An artificial neuron.

uct of an entrance signal x_m and a weight w_m represents the effectiveness of the synapses in a biological nervous cell. The analogy of an axon in an artificial neuron is a y entrance, that is represented by a numerical value.

The way an artificial neural network works

Similarly to a biological nervous cell, which transmits and transforms complicated electrochemical signals, an artificial neuron gains, transmits, transforms and utilizes entrance information.

A biological cell turns to an active state in the process of its stimulation by synapses. This condition in a cell results in an exit signal characterized by a specific shape, amplitude and duration time being sent through the axon. The act of a neuron's biological transformation from the state of equilibrium to the state of generating impulses is a rapid process. It is defined as the "ignition" of a neuron, which generates another specific signal at the exit. The "ignition" of a neuron takes place when the total signal coming through the dendrites reaches a particular threshold value. Subsequently the exit signal is passed on to other neurons by successive synapses.

The function of an artificial neuron is very similar to the role of a biological nervous cell. The neuron receives a particular number of entrance signals x_m (numbers) coming from outside. Each neuron has a threshold value θ which defines how strong the stimulation must be in order to cause an "ignition". The total sum of entrance signals input into a neuron together with adequate weights makes the value ϕ (the sum of products of neuron entrances x_m

$$\varphi = \sum_{i=1}^{m} x_i w_i$$

and adequate weights w_m,).

It is possible to calculate the stimulation of a neuron by summation of the value ϕ and the threshold value θ.

The counted value

$$\varphi + \theta = \sum_{i=1}^{m} x_i w_i + \theta$$

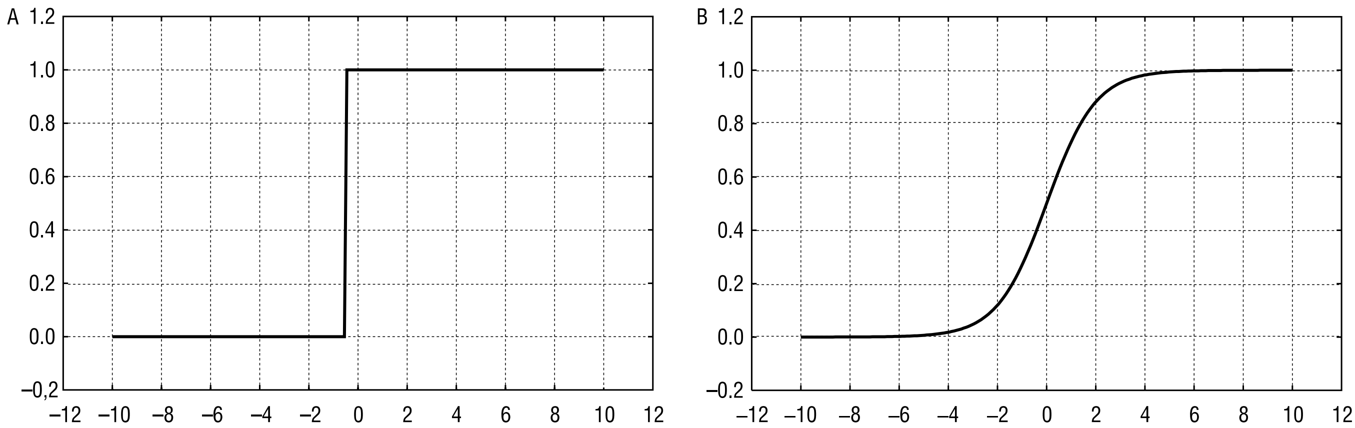

Figure 2A. Threshold; B. Sigmoid.

in approximation makes the postsynaptic potential of a biological neural cell. Then it is transformed by an agreed activation function F. The calculated total amount of the equation is the exit signal of the artificial neuron y .

$$y = F\left(\sum_{i=1}^{m} x_i w_i + \theta\right)$$

The properties and behavior of an artificial neuron depend on the used activation function F. The equation used for constructing ANN is either the threshold function (that is, such a function that will generate the values 0 or 1 at the exit) or the continuous function (for example a sigmoid). In the case of the threshold function, the final result equals 0 for entrance values less than 0 and 1 for entrance values more than or equal to 0. This kind of method is called an "all or nothing" rule and it relates to the behavior of a biological neural cell. If a continuous activation function is used, continually changing exit values will be the obtained as the result. The diagrams below show threshold and sigmoid activation function F (Fig. 2 AB).

In actual constructions of ANN, mostly linear functions are used rather than threshold activation, despite the biological rationality of the latter. This fact also has its analogy in natural cell processes. A nervous cell never settles on sending a single impulse, but sends out a whole series of them, characterized by momentary frequencies. These frequencies in a biological cell take different values in a manner similar to the linearly changing signals in an ANN during the use of continuous activation function.

Main types of ANN

The brain, which consists of nerve cells, can be referred to as a biological analogy of ANN, which is formed by linking artificial neurons (Fig. 1B) into a network (Fig. 3). Several layers compose this network, each having a particular number of neurons, previously defined by its constructor. We distinguish one entrance layer, hidden layers and an exit layer. In the simplest case, neurons belonging to one layer are not connected with each other. On the contrary, the neurons of adjacent layers are connected "one to one". The entrance signals are numerical. A hidden layer is built by neurons, whose reciprocal dependence is formed during the process of learning. The exit layer produces the calculation result. ANN entrances combine the features of sensory nerves, while exits represent the motor nerves of a brain.

Neurons forming each layer are connected in a specific way, making ANN a structure of real architecture. Depending on the way neurons are linked together, we can distinguish one-directional networks as well as ones with feedback.

Feedforward networks

ANN, in which a signal spreads in one chosen direction, are called feedforward networks. A message in such a neural network runs from neurons belonging to the entrance layer, passes through the hidden layer, to finally reach neurons forming an exit layer. Feedforward net can be divided into mono-layer (most frequently Perceptron-like) (Fig. 1B) and multilayer types (Fig. 3). A feedforward net may be built by either a singular neuron having a few entrance or several neurons with singular entrance. Both of these systems are mathematically equivalent.

The most frequently used structure of ANN is Multi Layer Perceptron (MLP). Its architecture is characterized by perceptron cells grouped in layers and forming connections with cells from each neighboring layer, while lacking connections within each layer. A significant feature of MLP is the presence of one or a few hidden layers. These consist of neurons whose reciprocal dependence forms during the process of learning. After data is brought into

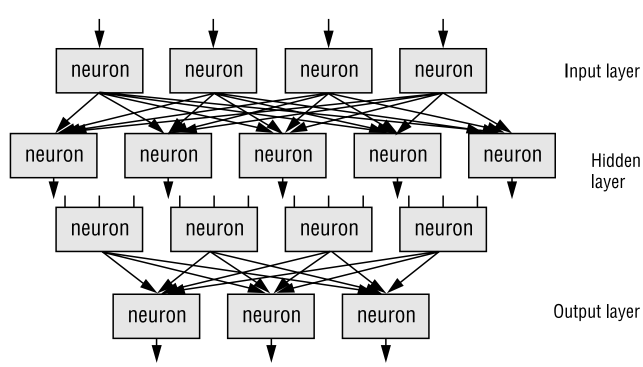

Figure 3. A feedforward multilayer neural network.

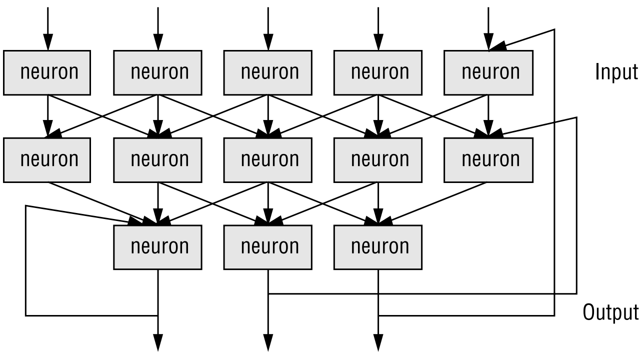

Figure 4. Recurrent neural network.

ANN, it undergoes nonlinear transformations in the neurons of the hidden layers by an undefined activation of function F. Usually the quantity of hidden layer neurons is resolved experimentally, considering the net's highest possible efficacy.

The system of a MPL-type ANN activity is formed by the following sequences:
— the insertion of particular data (signals) at the entrance of ANN;
— signals are multiplied by selected weights and summed for each neuron;
— the final sum is transformed into an argument for the activation of function F;
— the hidden layer receives signals from the exit of the entrance layer and repeats the whole above-discussed process (the entrance values are multiplied by chosen weight and then summed and transformed by the activation of function F);
— the entire process above is repeated in the exit layer, it only differs in the fact that the exit of its last AN layer is also the exit of the whole net;
— the exit layer generates the final result.

Recurrent networks

ANNs that utilize feedback are called recurrent networks (Fig. 4). The term feedback means that signals at an exit are being sent back as an entrance layer's input, which results in the dynamism of the ANN's activity. Entrance signals in a recurrent network strongly depend on the present state and on the previous cycle's exit signals.

Complex dynamics enables an ANN having feedback to have a very complex behavior and form of activity.

Teaching an ANN

In a nervous cell, the structures that plays an important part in the act of memorizing are the synapses, which contain a chemical compound called a neurotransmitter. This substance is capable of sending electrochemical signals. An artificial neuron also has a kind of inner memory, that depends on particular weights w_m and threshold θ.

From a mathematical point of view, the learning sequence consists in finding such a group of weight coefficients w_m that will ensure the possibly lowest value of error frequency in an artificial neuron, by which the best result in a particular task is obtained.

The ANN works in two stages: the first is the act of learning, during which the net uses a training sequence, followed by the second stage called exploitation, in which a testing sequence is used. While learning, a neural network gathers information on what its task is and how it ought to be accomplished, whereas in the exploitation stage the neural network has an opportunity to utilize its gained knowledge by solving problems.

Two methods are used for teaching an ANN: one which requires a teacher (being supervised) and one not involving a teacher.

Learning with a teacher

In this method of learning, the teacher, who is the neural network's constructor, includes answers to an inputted entrance signal. It might be intuitively understood that for particular entrance signal provided, exit data show what kind of answer is required. The teacher inputs pairs of values (the entrance signal and the expected answer) into an ANN. All of these pairs in a defined training group are called learning sequences. Therefore, learning involving a teacher is a process of inputting a learning sequence into the ANN that is to correctly teach a net to complete a particular task.

Learning without a teacher

There are many ways in which a neural network learns without a teacher by self-training. Generally, they can be described as methods in which only a set of example input data is given at the entrance of an ANN. The constructor does not provide info about answers expected and desired. Solely by using the entrance data, the neural network builds an optimal algorithm of activity on its own. Usually the algorithm consists of revealing and recognizing some classes of repeated entrance signal. During this already-described process, only this kind of knowledge is used that figures act as entrance data, on the basis of which neural network gathers needed information.

Algorithms for teaching a neural network

The existing algorithms for teaching ANNs are characterized by a varied level of complexity as well as effectiveness. Three basic learning algorithms are to be described: the delta rule and back-propagation, which are methods of learning with a teacher and Hebb's rule, which is an example of learning without a teacher.

Delta rule

The delta rule is a technique for learning which requires a teacher and is used in multi-layer webs. In this method, after receiving a particular signal at its entrance, every neuron defines its own exit signal and utilizes its possessed knowledge represented by previously stated weight values and entrance values. The value of the exit signal as defined by a neuron in each step of the learning process, is compared to the model answer given by the teacher in the learning sequence. If a divergence is observed, then the neuron defines the difference between its own exit signal and that value of a signal which is correct according to the teacher. The obtained difference is marked by the Greek letter δ (delta), which has given the rule its name.

A neuron uses the error signal to correct its weight parameters by following the rules below:
— the larger the detected error, the more intensely the weight parameters are changed;
— weight parameters related to these entrances, at which high values of entrance signal happened to occur are more changed

than the weights of entrances at which the entrance signal value was small.

Being aware of the value of the error made by a neuron and its entrance signal, it is possible to predict the way its weights will be modified.

A neural network using this method discontinues the process of learning when it is sufficiently trained, as small errors merely result in a minimized correction of weights. There is a rule in the delta method which depends the size of correction on the size of the entrance signal, which is transmitted to a net by a defined weight.

Additionally, in practical realizations of the described delta algorithm another element is included. The learning process may become more efficient by using a learning parameter η (learning rate). The constructor should decide on how intense the changes caused by defined entrance signal values ought to be as well as the particular value of error. The calculated correction of weights is then multiplied by this parameter. The η parameter may be chosen at random, however a particular decision has clear consequences. Selecting a low value learning rate parameter results in a slow process of ANN learning. On the other hand, choosing too high η value causes very rapid changes of the other net parameters, which, in extreme cases, lead to destabilization of the learning process. A very important aspect is selecting the η value by compromising, so as to make sure that both the benefits related to the effectiveness of the ANN work as well as the stability of the learning process are taken into consideration.

Back-propagation method

The back-propagation (BP) algorithm is at the same time, a basic and most efficient algorithm in teaching one-directional multi-layer ANN. The algorithm's name comes from the way in which error signals δ are counted, that is, in a converted order to that used for conducting signals in a neural network, which starts from the exit layer, goes through the hidden layer and reaches the entrance layer.

The BP algorithm uses the following steps in each learning sequence that consists of entrance data X and exit D:

— assigning random values near to 0 to all the weights followed by choosing a learning sequence (X, D);
— input the learning sequence at the entrance of ANN;
— delimitation the values of each element's exit for the next transformational layers, beginning from the first hidden layer in a particular layer;
— calculation of the value of errors for the exit layer. It is a value of error δ for a successive neuron in a layer;
— delimitation of the errors in following layers and the alteration of weight values;
— modifying all the weights;
— choosing the next learning sequence and going to point 2.

While teaching an ANN, it is possible to present any number of learning sequences. The structure of entrance data ought to be adequate to the structure of data used in the learning process. In a case when a neural network works with biased data, then a part of the learning sequence should also contain bias. This aspect of network exploitation is connected with its ability of generalization. For low disturbances of model signals, the acceleration of the convergence of the learning process is obtained.

The first operation of learning is initiating the primary order of weights. The most efficient and most frequently used solution is the random selection of initial weights.

Another substantial problem is the selection of a learning parameter η, which decides on the speed and stability of the learning process. For a low value of this parameter, a neural network learning proceeds at a very slow rate and the net requires a large number of operations in order to learn. If the parameter η value is high, then the algorithm of learning may become unstable. The η parameter value is usually selected from a range [0.05, 0,25]

Hebb's Rule

This is the most common method of self-training in neuronal networks. It consists of presenting several examples of entrance signals to a neuronal network, although lacking information about what is to be done with them. The neural network registers its situation and accepts a variety of signals although the significance of appearing objects and the relations between them are not defined. Based on the observations it carries out, the neural network gradually comes to understand this significance on its own and decides how these signals are related.

After inputting each successive group of entrance signals, a particular configuration of exit signals forms in the net. Some networks are very intensely stimulated, some mildly, while exit signals of others are as a matter of fact negative. Intuitively, these behaviors may be interpreted as follows: some neurons recognize presented signals as their own, other signals are treated as neutral and some signals are not accepted at all. The moment exit signals of every neuron of a net are stated, all neuron weights are changed. The domain of a change is calculated by multiplying the entrance signal coming into a particular entrance and the exit signal produced by a neuron in which weights are modified. It was concluded while analyzing the process of self-teaching using Hebb's method that if the described method is consistently used, then the initial and usually random preferences of neurons finally become systematically enhanced and precisely polarized. If a neuron has an inborn tendency in accepting signals of some sort, then, after having been shown a few displays, it learns to distinguish these signals more precisely. After a longer period of time of the self-training process, patterns of each type of signals, occurring at the entrance of a web, will form in the net. Consequently, comparable signals will be more efficiently grouped and distinguished by a certain n in the process of learning, while other types of signals will be more interesting for other neurons. As a result of the self-training process, the ANN will remember how many classes of similar signals appear at its entrance and will adjust neurons to each class, and the neuron will learn to distinguish, recognize and classify them.

However, the self-study method must also have its disadvantages. Compared to the method of learning that requires a teacher's supervision, self-study is a much slower process. It is unlikely to predict which neuron will specialize in recognizing a particular group of signals. In Hebb's method it is impossible to state whether a neural network taught in this way will learn all the presented patterns. This means that an ANN using the self-study method ought to be larger than one doing the same task but trained by a teacher. It is estimated that an ANN requires at least three times more elements in its exit layer than the number of patterns that the net is to recognize.

A significant aspect of the self-teaching process is choosing initial values of neural network weights. These values have a great effect on the final behavior of a neural network, as the teaching process merely intensifies and improves selected tendencies existing in a net from the start. The final effect, which is the result of the learning process, depends on the initial and innate qualities of a neural network.

Designing a training data set for ANN

The selection of an adequate training data set is an extremely important aspect for the correct and effective activity of ANN. Every neuronal network needs such a training data set that is required at the stage of gaining information. In order to assure an effective and stable neuronal network function, the training data set should fulfill two basic conditions:

— every possible class of data ought to be represented in the data set (usually the training data consist of several subgroups, from which every particular one is defined by a certain pattern — all these patterns should be represented during the learning sequence);
— statistical diversity ought to be represented within each class (the network activity is to be based on the "naturally" biased training sequence, not the ideally pure models — the training sequence project must ensure an adequate variety of bias effect.

On the other hand a factor which restrains the size of the example data set is the time — consuming learning process: it increases in direct proportion to the size of the training set. Consequently regarding the speed of the learning process the training data set should tend to be minimized.

By being based on the training data set, the ANN gathers information which it utilized in the process of exploitation afterwards. During the utilization of the ANN, a test data set is used. It serves for controlling the correctness of the network activity. In addition to the described separate training and test data sets, a randomized division of data is used by some authors. The most frequently used nonparametric statistical method of estimation of generalization is cross-validation [35]. This method consists of training the ANN with every possible case from the set excluding the one directly involved. This particular example case is used in the process of testing the network function. Subsequently, the same process is repeated for ever single case from the set. The obvious advantage of these methods is the effective utilization of the whole data set in the process of learning and testing of the ANN.

The size of the learning sequence depends on the size of the neuronal network for which it is designed. The greater the neuronal network, the larger learning sequence is required for its training. It is recommended that the minimum number of samples in the learning sequence should be at least 2 times higher than the network weights number. The number of the weights is calculated using the formula $M \times (N+1)$, in which M is the number of neurons in the hidden layer and N is the number of entrances of the network.

The application of artificial neural networks in nuclear medicine

In nuclear medicine, artificial neural networks have been applied as a supporting tool for diagnosis in the distinguishing of coronary artery disease [13, 14, 37–42], pulmonary embolism [15, 15, 44–50] and Alzheimer's disease [17, 18, 51].

Coronary artery disease

Artificial neural networks serve as a tool for detecting and localizing coronary artery disease mostly by using information contained in myocardial perfusion scintigrams.

Neural networks diagnosed coronary artery disease in a population of patients ranging from 74 to 410 [37, 42]. The diagnosed cases of coronary artery disease by artificial neural networks was comparable to a gold standard, which consisted of a coronary angiography [13, 14, 37–41], the interpretation of myocardial perfusion studies by two human experts [43] or all clinical patient data including ECG analysis, results of a physical exercise test as well as the patient's history [13, 14].

All the applied models of artificial neural networks had a three-layer structure and a feedforward flow of information. Each neural network consisted of three layers: an input layer, a hidden layer and an output layer. The number of neurons in an input layer varied between 11 and 256 depending on the used matrix of scintigrams and other data, whereas the number of neurons in a hidden layer ranged from 3 to 140. An output layer consisted of one, two or eight units [13, 14, 37–43].

Input signals of an artificial neural network which diagnosed coronary artery disease used all the clinical data. The myocardial perfusion images were pre-processed in order to decrease the number of variables and to extract relevant features from the images. This pre-processing was accomplished by a two-dimensional Fourier transformation that used both rest and stress images. After this transformation, 30 values were selected constituting the real and imaginary part of the Fourier coefficients were used to an input neurons [13, 14, 39–43]. Other studies have used additional features: male or female, an exercise test, resting ECG, the heart rate and workload [13, 39]. The character of input signals for artificial neural networks were designed so that each neuron would represent pixels of bull's-eye images [37, 38, 42].

When the output layer contained one neuron that encoded whether coronary artery disease was present or not the results were binary values such as 0 or 1 [13, 14, 39–43]. Another study contained eight output neurons that encoded coronary artery disease and the severity of disease was based on a classification into seven abnormal categories and one normal case [37].

In most of the studies, artificial neural networks training was based on the back propagation algorithm. It used commercial computer software that simulated artificial neural networks from among JETNET 3.0, MichiZane, VieNet2 [13, 14, 37–43].

Artificial neural networks topology for the detection coronary artery disease in present studies is given in Table 1. The list contains the number of neurons, each layer of artificial neural networks (an input layer, hidden and output), the learning method (BP, Bayes) and the number of patients.

It is based on the available studies in which artificial neural networks have been applied to predict coronary artery disease and have been shown to perform even better than experienced physicians.

Table 1. The models of artificial neural networks used to diagnose coronary artery disease

Input layer	Hidden layer	Output layer	Learning rules	Computer software	The population training test	References
256	100	8	*BP*	MichiZane	58 16	[37]
45	15	1	*BP*	VieNet2	159 *cross-validation*	[38]
45	15	2	*BP*	VieNet2	159 *cross-validation*	[38]
41	20	1	*Bayes*	Neal's Software	229 *cross-validation*	[13]
36	20	1	*Bayes*	Neal's Software	229 *cross-validation*	[13]
33	20	1	*Bayes*	Neal's Software	229 *cross-validation*	[13]
32	20	1	*Bayes*	Neal's Software	229 *cross-validation*	[13]
32	3	1	*BP*	JETNET	135 68	[39]
30	3	1	*BP*	JETNET	135 68	[39]
30	3	1	*BP*	JETNET 3.0	135 *cross-validation*	[40]
30	3	1	*BP*	JETNET 3.0	135 *cross-validation*	[41]
30	7	1	*BP*	No data	135 68 215	[14]
30	3	5	*BP*	JETNET 3.0	338 *cross-validation*	[43]
24	5	1	*BP*	NeuralDesk	410 A few training and test groups	[42]
11	20	1	*Bayes*	Program Neal	229 *cross-validation*	[13]
8	4	1	*BP*	NeuralDesk	410 A few training and test groups	[42]

Pulmonary embolism

Allowing the physician to diagnose pulmonary embolism requires of a lot of information. Apart from perfusion and ventilation lung, images a physician may also require chest radiographs, pulmonary angiography and clinical data. A lot of clinical information induced that artificial neural networks can help humans in the interpretation of ventilation-perfusion lungs scans [15, 16, 44–48].

Artificial neural networks diagnosed pulmonary embolism using a variable patients' population varying from 25 to 1087. The results of artificial neural networks were compared for accuracy in predicting the presence or absence of pulmonary embolism on pulmonary angiographies with human classifications of ventilation-perfusion lungs images [15, 16, 44–50].

In the most common studies a threelayer and feedforward perceptron architecture was used. The input layers contained variable neuron numbers from 4 to 25 and it depended on applied features. The hidden layers possessed 1 to 30 neurons. The output layers consisted of single neurons that were encoded with values between 0 and 1, the probability for pulmonary embolism [15, 16, 44–50].

The input features for artificial neural networks were derived from both ventilation and perfusion images, chest radiography and PIOPED criteria [15, 16, 44–50].

The other studies contained digital images analysis for ventilation and perfusion scans [16, 47, 48]. The method is based on using as input to the artificial neural networks, the percentage of mismatch for each segment of the lungs in ventilation-perfusion scans or the percentage of defects in the perfusion scintigrams.

The input neurons represented values that require multifractal texture analysis of perfusion lungs scans [5]. The input neurons corresponded to multifractal parameters and the fractal dimension of perfusion scintigrams.

In most of the studies, neural networks were trained using the back-propagation algorithm [15, 16, 44–49]. Neural networks also were trained using the Levenberg-Marquardt algorithm [50]. The calculations were undertaken using computer software such as: MATLAB, NeuroShell, JETNET.

The artificial neural network architecture for the detection of pulmonary embolism is given in Table 2. The list contains the number of neurons each layer (an input layer, hidden and output), the learning method, the computer software and the patient population.

The results showed that the performance of artificial neural networks was at the same level as the interpretation of lung scintigrams by experienced physicians [44–49].

Table 2. The models of artificial neural networks used to diagnose pulmonary embolism

Input layer	Hidden layer	Output layer	Learning rule	Computer software	The population training	The population test	References
25	10	1	*BP*	NeuroShell2	100	28	[44]
25	15	1	*BP*	NeuroShell2	100	28	[44]
25	20	1	*BP*	NeuroShell2	100	28	[44]
21	15	1	*BP*	No data	1064	104	[45]
21	15	1	*BP*	CodeWarrior	1064 *cross-validation*		[15]
21	1	1	*BP*	CodeWarrior	1064 *cross-validation*		[15]
20	2–30	1	*BP*	NeuroShell	150	30	[46]
18	5	1	*BP*	JETNET 3.0	509	104	[47]
18	5	1	*BP*	JETNET 3.0	1087	140	[48]
18	5	1	*BP*	JETNET 3.0	1087	102	[16]
12	4	1	*BP*	NeuroShell2	100 *cross-validation*		[49]
6	2	1	*BP*	NeuroShell2	100 *cross-validation*		[49]
6	5	1	*BP*	NeuroShell	150	30	[46]
5	16	1	*Levenber-Marquardt*	MATLAB	45 *cross-validation*		[35]
4	2	1	*Levenber-Marquardt*	MATLAB	45 *cross-validation*		[50]
4	8	1	*Levenber-Marquardt*	MATLAB	45 *cross-validation*		[50]
4	16	1	*Levenber-Marquardt*	MATLAB	45 *cross-validation*		[50]
4	24	1	*Levenber-Marquardt*	MATLAB	45 *cross-validation*		[50]

References

1. Honeyman MC, Brusic V, Stone NL, Harrison LC. Neural network-based prediction of candidate T-cell epitopes. Nat Biotechnol 1998; 16: 966–969.
2. Frank T, Kraiss KF, Kuhlen T. Comparative analysis of fuzzy ART and ART-2A network clustering performance. IEEE Trans Neural Networks 1998; 9: 544–559.
3. Zhou G, Si J. A systematic and effective supervised learning mechanism based on Jacobian rank deficiency. Neural Comput 1998; 10: 1031–1045.
4. Azen SP, Lapuerta P. Neural networks in outcomes research. Hepatology 1999; 29: 32–35.
5. Park JH, Kari S, King LE, Olsen NJ. Analysis of 31P MR spectroscopy data using artificial neural networks for longitudinal evaluation of muscle diseases: Drmatomyositis. NMR Biomed 1998; 11: 245–256.
6. Wei TT, Zhang Z, Barnhill SD, Madyastha KR, Zhang H, Oesterling JE. Understanding artificial neural networks and exploring their potential applications for the practicing urologist. Urology 1998; 52: 161–172.
7. Tardioli M. International migration destination forecast concerning the region 'Lazio'. Subst Use Misuse 1998; 33: 709–716.
8. Wallis JW. Use of artificial intelligence in cardiac imaging. J Nucl Med 2001; 42: 1192–1194.
9. Papik K, Molnar B, Schaefer R, Dombovari Z, Tulassay Z, Feher J. Application of neural networks in medicine-a review. Med Sci Monit 1998; 4: 538–546.
10. Baxt WG. Application of artificial neural networks to clinical medicine. Lancet 1995; 346: 1135–1138.
11. Garcia EV, Cooke CD, Folks RD, Santana CA, Krawczynska EG, De Braal L et al. Diagnostic performance of an expert system for the interpretation of myocardial perfusion SPECT studies. J Nucl Med 2001; 42: 1185–1191.
12. Ohlsson M. WeAidU — A decision support system for myocardial perfusion images using artificial neural networks. Artif Intell Med 2004; 30: 49–60.
13. Haraldsson H, Ohlsson M. Value of exercise data for the interpretation of myocardial perfusion SPECT. J Nucl Cardio 2002; 9: 169––173.
14. Toft JC, Lindahl D, Ohlsson M, Palmer J, Lundin A, Edenbrandt L et al. The optimal reference population for cardiac normality in myocardial SPECT in the detection of coronary artery stenoses: patients with normal coronary angiography or subjects with low likelihood of coronary artery disease? Eur J Nucl Med 2001; 28: 831–835.
15. Eng J. Predicting the presence of acute pulmonary embolism: a comparative analysis of the artificial neural network, logistic regression, and threshold models. AJR 2002; 179: 869–874.
16. Evander E, Holst H, Järund A, Ohlsson M, Wollmer P, Åström K et al. Role of acute pulmonary embolism: an evaluation using artificial neural networks. Eur J Nucl Med Mol Imaging 2003; 30: 961–965.
17. Chan KH, Johnson KA, Becker JA, Satlin A, Mendelson J, Garada B et al. A neural network classifier for cerebral perfusion imaging. J Nucl Med 1994; 35: 771–774.
18. Kippenhan JS, Barker WW, Nagel J, Grady C, Duara R. Neural-network classification of normal and Alzheimer's disease subjects using high-resolution and low-resolution PET cameras. J Nucl Med 1994; 35: 7–15.
19. Stefaniak B, Cholewinski W, Tarkowska A. Application of artificial neural network algorithm to detection of parathyroid adenoma. Nucl Med Rev 2003; 6: 111–117.

20. Ikeda H. Analysis of diffuse parenchymal liver disease by liver scintigrams: differential diagnosis using neuron and fuzzy. Osaka City Med J 1996; 42: 109–124.
21. Hamilton D, Miola UJ, Mousa D. Interpretation of captopril transplant renography using a feed forward neural network. J Nucl Med 1996; 37: 1649–1652.
22. Knoll P, Mirzaei S, Müllner A, Leitha T, Koriska K, Köhn H et al. An artificial neural net and error back propagation to reconstruct single photon emission computerized tomography data. Med Phys 1999; 26: 244–248.
23. Maksud P, Fertil B, Rica C, Fakhri GE, Aurengo A. Artificial neural network as a tool to compensate for scatter and attenuation in radionuclide imaging. J Nucl Med 1998; 39: 735–745.
24. Fakhri GE, Moore SC, Maksud P, Aurengo A, Kijewski MF. Absolute activity quantification in simultaneous $^{123}I/^{99m}Tc$ brain SPECT. J Nucl Med 2001; 42: 300–308.
25. Hasselmo ME. A computational model of the progression of Alzheimer's disease. MD Comput 1997; 14: 181–191.
26. Xing J, Gerstein GL. Networks with lateral connectivity. I. Dynamic properties mediated by the balance of intrinsic excitation and inhibition. II. Development of neuronal grouping and corresponding receptive field changes. III. Plasticity and reorganization of somatosensory cortex. J Neurophysiology 1996; 75: 184–232.
27. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943; 5: 115–133.
28. Von Neumann J. The computer and the brain. Yale Univ Press, New Haven 1958.
29. Taylor WK. Computers and the nervous system. Models and analogues in biology. Cambridge Univ Press, Cambridge 1960.
30. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 1958; 65: 386–408.
31. Widrow B, Hoff ME. Adaptive switching circuits. IRE WESCON Convention Record, New York; 1960: 96–104.
32. Widrow B. The Original Adaptive Neural Net Broom-Balancer. Proceedings of 1987 IEEE Int. Symp. on Circuits and Systems, Philadelphia, PA 1987: 351–357.
33. Minsky M, Papert S. Perceptrons. MIT Press, Cambridge 1969.
34. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representation by error propagation. In: Rumelhart DE, McCleland JL (eds.). Parallel distributed processing: explorations in the microstructure of cognition. Cambridge, Mass: MIT Press/Bradford Books 1986: 318–363.
35. Efron B, Tibshirani R. Cross-validation and the bootstrap: estimating the terror rate of a prediction rule. Technical Report, May 1995.
36. Master T. Sieci neuronowe w praktyce. Wydawnictwo Naukowo-Techniczne 1996.
37. Fujita H, Katafuchi T, Uehara T, Nishimura T. Application of artificial neural network to computer-aided diagnosis of coronary artery disease in myocardial SPECT bull's-eye images. J Nucl Med 1992; 33: 272–276.
38. Porenta G, Dorffner G, Kundrat S, Petta P, Duit-Schedlmayer J, Sochor H. Automated interpretation of planar thallium-201-dipyridamole stress-redistribution scintigrams using artificial neural networks. J Nucl Med 1994; 35: 2041–2047.
39. Lindahl D, Toft J, Hesse B, Palmer J, Ali S, Lundin A et al. Scandinavian test of artificial neural network for classification of myocardial perfusion images. Clin Physiol 2000; 4: 253–261.
40. Lindahl D, Palmer J, Ohlsson M, Peterson C, Lundin A, Edenbrandt L. Automated interpretation of myocardial SPECT perfusion images using artificial neural networks. J Nucl Med 1997; 38: 1870–1875.
41. Lindahl D, Lanke J, Lundin A, Palmer J, Edenbrandt L. Improved classifications of myocardial bull's-eye scintigrams with computer-based decision support system. J Nucl Med 1999; 40: 96–101.
42. Hamilton D, Riley PJ, Miola UJ, Amro AA. A feed forward neural network for classification of bull's-eye myocardial perfusion images. Eur J Nucl Med 1995; 22: 108–115.
43. Lindahl D, Palmer J, Edenbrandt L. Myocardial SPET: artificial neural networks describe extent and severity of perfusion defects. Clin Physiol 1999; 6: 497–503.
44. Scott JA, Palmer EL. Neural network analysis of ventilation-perfusion lung scans. Radiology 1993; 186: 661–664.
45. Tourassi GD, Floyd CE, Sostman HD, Coleman RE. Artificial neural networks for diagnosis of acute pulmonary embolism: effect of case and observer selection. Radiology 1995; 194: 889–893.
46. Fisher RE, Scott JA, Palmer EL. Neural networks in ventilation-perfusion imaging. Part I. Effects of interpretive criteria and network architecture. Radiology 1996; 198: 699–706.
47. Holst H, Åström K, Järund A, Palmer J, Heyden A, Kahl et al. Automated interpretation of ventilation-perfusion lung scintigrams for the diagnosis of pulmonary embolism using artificial neural networks. Eur J Nucl Med 2000; 27: 400–406.
48. Holst H, Måre K, Järund A, Åström K, Evander E, Tägil K et al. An independent evaluation of a new method for automated interpretation of lung scintigrams using artificial neural networks. Eur J Nucl Med 2001; 28: 33–38.
49. Scott JA, Palmer EL, Fischman AJ. How well can radiologists using neural network software diagnose pulmonary embolism? AJR 2000; 175: 399–405.
50. Tourassi GD, Frederick ED, Floyd CE, Coleman RE. Multifractal texture analysis of perfusion lung scans as a potential diagnostic tool for acute pulmonary embolism. Comput Biol Med 2001; 31: 15–25.
51. Page M, Howard RJ, O'Brien JT, Buxton-Thomas MS, Pickering AD. Use of neural networks in brain SPECT to diagnose Alzheimer's disease. J Nucl Med 1996; 37: 195–200.