

Chapter 2

Artificial Neural Network Architectures and Training Processes

2.1 Introduction

The architecture of an artificial neural network defines how its several neurons are arranged, or placed, in relation to each other. These arrangements are structured essentially by directing the synaptic connections of the neurons.

The topology of a given neural network, within a particular architecture, can be defined as the different structural compositions it can assume. In other words, it is possible to have two topologies belonging to the same architecture, where the first topology is composed of 10 neurons, and the second is composed of 20 neurons. Moreover, one can consist of neurons with logistic activation function, while the other one can consist of neurons with the hyperbolic tangent as the activation function.

On the other hand, training a particular architecture involves applying a set of ordinated steps to adjust the weights and thresholds of its neurons. Hence, such adjustment process, also known as learning algorithm, aims to tune the network so that its outputs are close to the desired values.

2.2 Main Architectures of Artificial Neural Networks

In general, an artificial neural network can be divided into three parts, named layers, which are known as:

(a) *Input layer*

This layer is responsible for receiving information (data), signals, features, or measurements from the external environment. These inputs (samples or patterns) are usually normalized within the limit values produced by activation functions. This normalization results in better numerical precision for the mathematical operations performed by the network.

(b) *Hidden, intermediate, or invisible layers*

These layers are composed of neurons which are responsible for extracting patterns associated with the process or system being analyzed. These layers perform most of the internal processing from a network.

(c) *Output layer*

This layer is also composed of neurons, and thus is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers.

The main architectures of artificial neural networks, considering the neuron disposition, as well as how they are interconnected and how its layers are composed, can be divided as follows: (i) single-layer feedforward network, (ii) multilayer feedforward networks, (iii) recurrent networks and (iv) mesh networks.

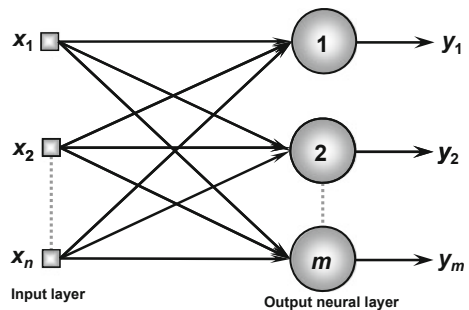
2.2.1 Single-Layer Feedforward Architecture

This artificial neural network has just one input layer and a single neural layer, which is also the output layer. Figure 2.1 illustrates a simple-layer feedforward network composed of n inputs and m outputs.

The information always flows in a single direction (thus, unidirectional), which is from the input layer to the output layer. From Fig. 2.1, it is possible to see that in networks belonging to this architecture, the number of network outputs will always coincide with its amount of neurons. These networks are usually employed in pattern classification and linear filtering problems.

Among the main network types belonging to feedforward architecture are the Perceptron and the ADALINE, whose learning algorithms used in their training processes are based respectively on Hebb's rule and Delta rule, as it will be discussed in the next chapters.

Fig. 2.1 Example of a single-layer feedforward network



2.2.2 Multiple-Layer Feedforward Architectures

Differently from networks belonging to the previous architecture, feedforward networks with multiple layers are composed of one or more hidden neural layers (Fig. 2.2). They are employed in the solution of diverse problems, like those related to function approximation, pattern classification, system identification, process control, optimization, robotics, and so on.

Figure 2.2 shows a feedforward network with multiple layers composed of one input layer with n sample signals, two hidden neural layers consisting of n_1 and n_2 neurons respectively, and, finally, one output neural layer composed of m neurons representing the respective output values of the problem being analyzed.

Among the main networks using multiple-layer feedforward architectures are the Multilayer Perceptron (MLP) and the Radial Basis Function (RBF), whose learning algorithms used in their training processes are respectively based on the generalized delta rule and the competitive/delta rule. These concepts will be addressed in the next chapters.

From Fig. 2.2, it is possible to understand that the amount of neurons composing the first hidden layer is usually different from the number of signals composing the input layer of the network. In fact, the number of hidden layers and their respective amount of neurons depend on the nature and complexity of the problem being mapped by the network, as well as the quantity and quality of the available data about the problem. Nonetheless, likewise for simple-layer feedforward networks, the amount of output signals will always coincide with the number of neurons from that respective layer.

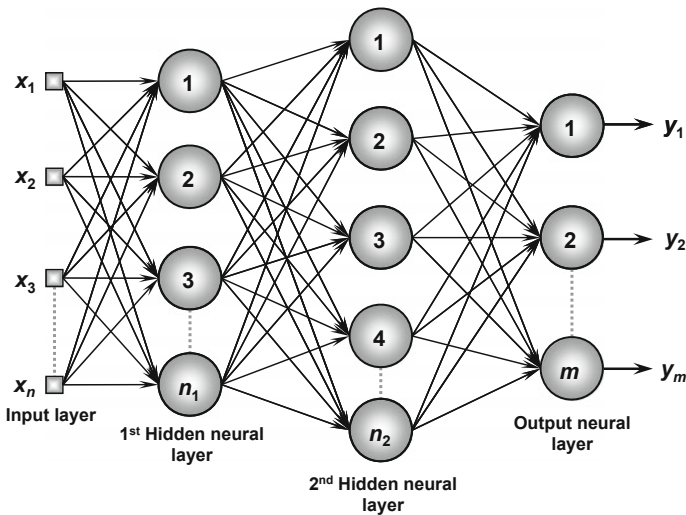
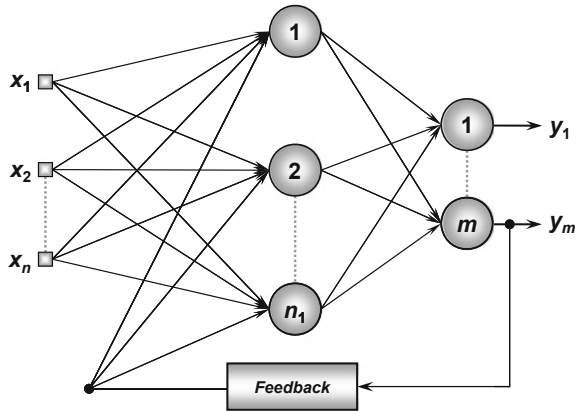


Fig. 2.2 Example of a feedforward network with multiple layers

Fig. 2.3 Example of a recurrent network



2.2.3 Recurrent or Feedback Architecture

In these networks, the outputs of the neurons are used as feedback inputs for other neurons. The feedback feature qualifies these networks for dynamic information processing, meaning that they can be employed on time-variant systems, such as time series prediction, system identification and optimization, process control, and so forth.

Among the main feedback networks are the Hopfield and the Perceptron with feedback between neurons from distinct layers, whose learning algorithms used in their training processes are respectively based on energy function minimization and generalized delta rule, as will be investigated in the next chapters.

Figure 2.3 illustrates an example of a Perceptron network with feedback, where one of its output signals is fed back to the middle layer.

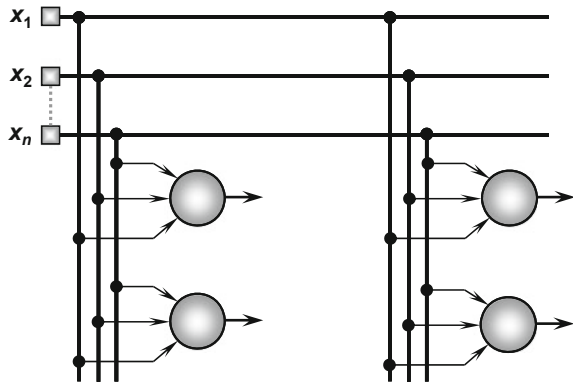
Thus, using the feedback process, the networks with this architecture produce current outputs also taking into consideration the previous output values.

2.2.4 Mesh Architectures

The main features of networks with mesh structures reside in considering the spatial arrangement of neurons for pattern extraction purposes, that is, the spatial localization of the neurons is directly related to the process of adjusting their synaptic weights and thresholds. These networks serve a wide range of applications and are used in problems involving data clustering, pattern recognition, system optimization, graphs, and so forth.

The Kohonen network is the main representative of mesh architectures, and its training is performed through a competitive process, as will be described in the following chapters. Figure 2.4 illustrates an example of the Kohonen network where its neurons are arranged within a two-dimensional space.

Fig. 2.4 Structure of a mesh network



From Fig. 2.4, it is possible to verify that in this network category, the several input signals are read by all neurons within the network.

2.3 Training Processes and Properties of Learning

One of the most relevant features of artificial neural networks is their capability of learning from the presentation of samples (patterns), which expresses the system behavior. Hence, after the network has learned the relationship between inputs and outputs, it can generalize solutions, meaning that the network can produce an output which is close to the expected (or desired) output of any given input values.

Therefore, the training process of a neural network consists of applying the required ordinated steps for tuning the synaptic weights and thresholds of its neurons, in order to generalize the solutions produced by its outputs.

The set of ordinated steps used for training the network is called learning algorithm. During its execution, the network will thus be able to extract discriminant features about the system being mapped from samples acquired from the system.

Usually, the complete set containing all available samples of the system behavior is divided into two subsets, which are called training subset and test subset. The training subset, composed of 60–90 % of random samples from the complete set, will be used essentially in the learning process. On the other hand, the test subset, which is composed of 10–40 % from the complete sample set, will be used to verify if the network capabilities of generalizing solutions are within acceptable levels, thus allowing the validation of a given topology. Nonetheless, when dimensioning these subsets, statistical features of the data must also be considered.

During the training process of artificial neural networks, each complete presentation of all the samples belonging to the training set, in order to adjust the synaptic weights and thresholds, will be called training epoch.

2.3.1 Supervised Learning

The supervised learning strategy consists of having available the desired outputs for a given set of input signals; in other words, each training sample is composed of the input signals and their corresponding outputs. Henceforth, it requires a table with input/output data, also called attribute/value table, which represents the process and its behavior. It is from this information that the neural structures will formulate “hypothesis” about the system being learned.

In this case, the application of supervised learning only depends on the availability of that attribute/value table, and it behaves as if a “coach” is teaching the network what is the correct response for each sample presented for its input.

The synaptic weights and thresholds of the network are continually adjusted through the application of comparative actions, executed by the learning algorithm itself, that supervise the discrepancy between the produced outputs with respect to the desired outputs, using this difference on the adjustment procedure. The network is considered “trained” when this discrepancy is within an acceptable value range, taking into account the purposes of generalizing solutions.

In fact, the supervised learning is a typical case of pure inductive inference, where the free variables of the network are adjusted by knowing a priori the desired outputs for the investigated system.

Donald Hebb proposed the first supervised learning strategy in 1949, inspired by neurophysiological observations (Hebb 1949).

2.3.2 Unsupervised Learning

Different from supervised learning, the application of an algorithm based on unsupervised learning does not require any knowledge of the respective desired outputs.

Thus, the network needs to organize itself when there are existing particularities between the elements that compose the entire sample set, identifying subsets (or clusters) presenting similarities. The learning algorithm adjusts the synaptic weights and thresholds of the network in order to reflect these clusters within the network itself.

Alternatively, the network designer can specify (a priori) the maximum quantity of these possible clusters, using his/her knowledge about the problem.

2.3.3 Reinforcement Learning

Methods based on reinforcement learning are considered a variation of supervised learning techniques, since they continuously analyze the difference between the

response produced by the network and the corresponding desired output (Sutton and Barto 1998). The learning algorithms used on reinforcement learning adjust the internal neural parameters relying on any qualitative or quantitative information acquired through the interaction with the system (environment) being mapped, using this information to evaluate the learning performance.

The network learning process is usually done by trial and error because the only available response for a given input is whether it was satisfactory or unsatisfactory. If satisfactory, the synaptic weights and thresholds are gradually incremented to reinforce (reward) this behavioral condition involved with the system.

Several learning algorithms used by reinforcement learning are based on stochastic methods that probabilistically select the adjustment actions, considering a finite set of possible solutions that can be rewarded if they have chances of generating satisfactory results. During the training process, the probabilities associated with action adjustment are modified to enhance the network performance (Tsoukalas and Uhrig 1997).

This adjustment strategy has some similarities to some dynamic programming techniques (Bertsekas and Tsitsiklis 1996; Watkins 1989).

2.3.4 Offline Learning

In offline learning, also named batch learning, the adjustments on the weight vectors and thresholds of the network are performed after all the training set is presented, since each adjustment step takes into account the number of errors observed within the training samples with respect to the desired values for their outputs.

Therefore, networks using offline learning requires, at least, one training epoch for executing one adjustment step on their weights and thresholds. Hence, all training samples must be available during the whole learning process.

2.3.5 Online Learning

Opposite to offline learning, in online learning, the adjustments on the weights and thresholds of the network are performed after presenting each training sample. Thus, after executing the adjustment step, the respective sample can be discarded.

Online learning with this configuration is usually used when the behavior of the system being mapped changes rapidly, thus the adoption of offline learning is almost impractical because the samples used at a given moment may no more represent the system behavior in posterior moments.

However, since patterns are presented one at a time, weight and threshold adjustment actions are well located and punctual, and they reflect a given behavioral circumstance of the system. Therefore, the network will begin to provide accurate responses after presenting a significant number of samples (Reed and Marks II, 1999).

2.4 Exercises

1. Write about the advantages and disadvantages involved with online and offline learning.
2. Consider an application with four inputs and two outputs. The designers of this application state that the feedforward network to be developed must present exactly four neurons in the first hidden layer. Discuss about the pertinence of this information.
3. Relating to the previous exercise, cite some factors that influence the determination of the hidden layers number of a multiple layer feedforward network.
4. What are the eventual structural differences observed between recurrent networks and feedforward networks.
5. In what application categories the employment of recurrent neural networks is essential?
6. Draw a block diagram illustrating how the supervised training works.
7. Write about the concepts of training methods and learning algorithms, further explaining the concept of training epoch.
8. What are the main differences between supervised and unsupervised training methods?
9. What are the main differences between supervised and reinforcement learning methods?
10. Considering a specific application, explain what performance criterion could be used for adjusting the weights and thresholds of a network using reinforcement learning method.



<http://www.springer.com/978-3-319-43161-1>

Artificial Neural Networks

A Practical Course

Nunes Silva, I.; Hernane Spatti, D.; Andrade Flauzino, R.;

Liboni, L.H.B.; dos Reis Alves, S.F.

2017, XX, 307 p. 203 illus., 13 illus. in color., Hardcover

ISBN: 978-3-319-43161-1