4.1 Introduction

The process of learning from examples (or supervised learning) of multilayer neural networks can be considered when a set \( \{ input_p, output_p \} \), \( p = 1, 2, ..., P \), is available. The aim is to configure a neural network in such a way as to generate the definite output if the respective input feeds the network. The generation should be done with highest possible accuracy, the requirement that can be written as a difference between the desired \( output_p \) and the output generated by the network \( NN_p \), summed over all the available examples \( P \), as

\[
E = \sum_{p=1}^{P} \phi (output_p - NN_p (input_p))^2 \leq \varepsilon
\]  

(4.1)

where \( E \) denotes the performance index of learning (or the error of learning), \( \phi ( ) \) is a measure of error, and \( \varepsilon \) is a prescribed positive small value.

In this book we will consider the Euclidean measure of the learning error

\[
E = \frac{1}{2} \sum_{p=1}^{P} \left( output_p - NN_p (input_p) \right)^2 = \sum_{p=1}^{P} E_p.
\]  

(4.2)

This form of the error is the most widely used one. The rationale came from the theory of signal processing, where energy of the error between two signals has a similar form. The term *energy function* for neural networks was introduced by Hopfield (1982) and was borrowed from statistical mechanics of magnetic systems. Additionally, the form (4.2) together with introducing the sigmoidal activation functions by Hinton and Sejnowski (1983) made the learning of the neural networks as a nonlinear differentiable optimisation problem.

The learning process of the neural networks is considered as an iterative procedure. There are two main ways of updating the weights. The first is based on the *incremental learning*, i.e. the weights are updated after presentation of every pair \( \{ input_p, output_p \} \), \( p = 1, 2, ..., P \), namely
Learning from Examples

\[ w_{p}^{\text{new}} = w_{p}^{\text{old}} + \Delta w_{p} \]  (4.3)

in such a way that

\[ E_{p}\left(w_{p}^{\text{old}}\right) < E_{p}\left(w_{p}^{\text{new}}\right). \]  (4.4)

The second way is called \textit{batch learning}, and the weights are updated after presentation the whole set of the training pairs, namely

\[ w_{\text{new}} = w_{\text{old}} + \Delta w \]  (4.5)

in order to obtain

\[ E\left(w_{\text{old}}\right) < E_{p}\left(w_{\text{new}}\right). \]  (4.6)

In the subsequent material, the \textit{old} values of the weights will be called as \textit{nominal} values and denoted by \( \bar{w} \), while the updated values will be denoted by \( w' \).

Additionally, let us remind of the following aspects related to the considered neural networks:

- the network consists of \( L \) layers, each layer is labelled by \( l = 0,1,2,...,L \), the layer \( l = 0 \) denotes the external inputs to the network,
- each layer is composed of \( N(l) \), \( l = 0,1,2,...,L \), neurons, where \( N(0) \) denotes the number of inputs,
- the neurons belonging to the \((l-1)\)-st layer are connected with the neurons of the \( l \)-th layer;
- the activation function of each neuron is defined by one of the following sigmoidal functions

\[ f\left(\text{net}_{j(l)}\right) = \frac{1}{\exp(-\text{net}_{j(l)})} \]  (4.7)

\[ f\left(\text{net}_{j(l)}\right) = \frac{2}{1+\exp(-\text{net}_{j(l)})} - 1 \]  (4.8)

where

\[ \text{net}_{j(l)} = \text{net}_{j(l)} = \sum_{i(l-1)=0}^{N(l-1)} w_{i(l-1)j(l)} x_{i(l-1)} \]  (4.9)

- the learning of the network is understood as an adjustment of the following weights

\[ w_{i(l-1)j(l)}, \ i(l-1) = 1,2,...,N(l-1), \ j(l) = 1,2,...,N(l), \ l = 1,2,...,L \]

meant to minimize the performance index describing the error of learning