LEARNING IN NEURAL NETWORKS

J. Bernasconi

Asea Brown Boveri Corporate Research

CH-5405 Baden, Switzerland

Abstract
Learning is one of the most important aspects of neural networks, and there exist many different learning paradigms. In this article, we concentrate on supervised learning from examples and provide a brief introduction to two of the most widely used learning procedures, "Error Backpropagation" and "Boltzmann Machine Learning". Both procedures can be viewed as strategies to minimize a suitably chosen error measure, and their performance depends on a number of parameters and implementation details. A simple model problem is used to illustrate how these dependences can affect the learning behavior.

1. Introduction
An artificial neural network consists of a set of units (formal neurons), each connected to some number of other units in the system. The state of the i-th unit is described by a scalar variable $S_i$, and each connection $j \rightarrow i$ carries a weight $W_{ij}$ which can be positive, zero, or negative. Depending on the type of network considered, the weights are chosen symmetric ($W_{ij} = W_{ji}$) or asymmetric ($W_{ij} \neq W_{ji}$), and the $S_i$ either assume only a discrete set of values (e.g., $\{0,1\}$ or $\{-1,+1\}$) or vary continuously (e.g., between 0 and 1, or between $-1$ and $+1$).

Artificial neural networks can be considered as grossly simplified models of the human brain. The units represent the neurons whose state of activity is measured by the variables $S_i$ (e.g., $S_i = 1$ if neuron $i$ is firing, and $S_i = 0$ if neuron $i$ is quiescent), and the $W_{ij}$ denote the strengths of the synapses. These can be excitatory ($W_{ij} > 0$) or inhibitory ($W_{ij} < 0$). Neural networks are also closely related to spin systems in statistical physics ($S_i =$ spin variable, $W_{ij} =$ interaction strength), and this analogy has recently led to considerable advances in the analysis of neural network properties [1-3].
If we consider neural networks as computing architectures, the units represent simple processing elements which update their states in a synchronous or asynchronous manner. The update rule is local and uniform, and usually taken to be of the form

\[ S_i = f(\sum_j W_{ij} S_j - \theta_i) \]  

where \( f \) is a nonlinear activation function, e.g., a threshold function or a sigmoid-type function such as \( f(x) = 1/(1 + \exp(-x)) \). The updated value of \( S_i \) thus only depends on the total weighted input to unit \( i \) and on a threshold \( \theta_i \) which can be regarded as an extra weight (associated with the connection to a unit whose value is always equal to \(-1\)). Certain types of neural networks (e.g., the Boltzmann machine [4,5]) employ stochastic units. In these cases, Eq.(1) is replaced by a probabilistic rule, i.e., \( f(\sum_j W_{ij} S_j - \theta_i) \) represents the probability that \( S_i \) takes one of two possible values.

In neural networks, input and output are represented by the \( S_i \)-configurations of certain groups of units, and Eq.(1) defines a dynamical process which associates each input configuration with an output configuration. The resulting output configurations, of course, depend on the chosen weights \( W_{ij} \), i.e., information or knowledge is stored in the pattern of weights and not in the processing units. In a learning phase, these weights therefore have to be adjusted in such a way that the network performs a given task as well as possible.

Quite generally, a neural network is characterized by its topology, by the type of units used, by the form of the update rule, and by the learning procedure. In this paper, we are primarily concerned with the learning behavior of neural networks, and we restrict ourselves to supervised learning from examples. In section 2, we introduce two of the most widely used learning procedures, "Error Backpropagation" and "Boltzmann Machine Learning", and section 3 is devoted to a discussion of some implementation issues. In section 4, we briefly review some recent results concerning the performance of these learning algorithms. The efficiency of a given algorithm depends on a number of parameters and implementation details, and in section 5 we use a simple model problem to illustrate how these dependences can affect the learning behavior.

2. Supervised Learning from Examples

We shall be concerned with neural networks in which the units are divided into input units, output units, and so-called hidden units. If an explicit distinction is required, the state variables \( S_i \) of the input units will be denoted by \( I_i \), and those of the output units by \( O_i \). The networks are supposed to perform a given pattern association task (classification, diagnosis, etc.) which can be expressed in terms of a specific input/output