Available online at www.sciencedirect.com
SCIENCE DIRECT
Neurocomputing 69 (2006) 660–670
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Learning vector quantization: The dynamics of winner-takes-all algorithms

Michael Biehl[a,*], Anarta Ghosh[a], Barbara Hammer[b]

[a]Rijksuniversiteit Groningen, Mathematics and Computing Science, P.O. Box 800, NL-9700 AV Groningen, The Netherlands
[b]Clausthal University of Technology, Institute of Computer Science, D-98678 Clausthal-Zellerfeld, Germany

Available online 26 January 2006

Abstract

Winner–Takes–All (WTA) prescriptions for learning vector quantization (LVQ) are studied in the framework of a model situation: two competing prototype vectors are updated according to a sequence of example data drawn from a mixture of Gaussians. The theory of on-line learning allows for an exact mathematical description of the training dynamics, even if an underlying cost function cannot be identified. We compare the typical behavior of several WTA schemes including basic LVQ and unsupervised vector quantization. The focus is on the learning curves, i.e. the achievable generalization ability as a function of the number of training examples.

Keywords: Learning vector quantization; Prototype-based classification; Winner-takes-all algorithms; On-line learning; Competitive learning

1. Introduction

Learning vector quantization (LVQ) as originally proposed by Kohonen [10] is a widely used approach to classification. It is applied in a variety of practical problems, including medical image and data analysis, e.g. proteomics, classification of satellite spectral data, fault detection in technical processes, and language recognition, to name only a few. An overview and further references can be obtained from [1].

LVQ procedures are easy to implement and intuitively clear. The classification of data is based on a comparison with a number of so-called prototype vectors. The similarity is frequently measured in terms of Euclidean distance in feature space. Prototypes are determined in a training phase from labeled examples and can be interpreted in a straightforward way as they directly represent typical data in the same space. This is in contrast with, say, adaptive weights in feedforward neural networks or support vector machines which do not allow for immediate interpretation as easily. Among the most attractive features of LVQ is the natural way in which it can be applied to multi-class problems.

In the simplest so-called *hard* or *crisp* schemes any feature vector is assigned to the closest of all prototypes and the corresponding class. In general, several prototypes will be used to represent each class. Extensions of the deterministic assignment to a probabilistic *soft* classification are conceptually straightforward but will not be considered here.

Plausible training prescriptions exist which employ the concept of on-line competitive learning: prototypes are updated according to their distance from a given example in a sequence of training data. Schemes in which only the winner, i.e. the currently closest prototype is updated have been termed winner-takes-all algorithms and we will concentrate on this class of prescriptions here.

The ultimate goal of the training process is, of course, to find a classifier which labels novel data correctly with high probability, after training. This so-called generalization ability will be in the focus of our analysis in the following.

Several modifications and potential improvements of Kohonen's original LVQ procedure have been suggested. They aim at achieving better approximations of Bayes optimal decision boundaries, faster or more robust

*Corresponding author. Fax: +31 50 363 3800.
E-mail address: m.biehl@rug.nl (M. Biehl).

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.12.007

convergence, or the incorporation of more flexible metrics, to name a few examples [5,8–10,16].

Many of the suggested algorithms are based on plausible but purely heuristic arguments and they often lack a thorough theoretical understanding. Other procedures can be derived from an underlying cost function, such as *Generalized Relevance* LVQ [8,9] or LVQ2.1, the latter being a limit case of a statistical model [14,15]. However, the connection of the cost functions with the ultimate goal of training, i.e. the generalization ability, is often unclear. Furthermore, several learning rules display instabilities and divergent behavior and require modifications such as the *window rule* for LVQ2.1 [10].

Clearly, a better theoretical understanding of the training algorithms should be helpful in improving their performance and in designing novel, more efficient schemes.

In this work we employ a theoretical framework which makes possible a systematic investigation and comparison of LVQ training procedures. We consider on-line training from a sequence of uncorrelated, random training data which is generated according to a model distribution. Its purpose is to define a non-trivial structure of data and facilitate our analytical approach. We would like to point out, however, that the training algorithms do not make use of the form of this distribution as, for instance, density estimation schemes.

The dynamics of training is studied by applying the successful theory of online learning [2,6,12] which relates to ideas and concepts known from statistical physics. The essential ingredients of the approach are (1) the consideration of high-dimensional data and large systems in the so-called thermodynamic limit and (2) the evaluation of averages over the randomness or *disorder* contained in the sequence of examples. The typical properties of large systems are fully described by only a few characteristic quantities. Under simplifying assumptions, the evolution of these so-called *order parameters* is given by deterministic coupled ordinary differential equations (ODE) which describe the dynamics of on-line learning exactly in the thermodynamic limit. For reviews of this very successful approach to the investigation of machine learning processes consult, for instance, [2,6,17].

The formalism enables us to compare the dynamics and generalization ability of different WTA schemes including basic LVQ1 and unsupervised vector quantization (VQ). The analysis can readily be extended to more general schemes, approaching the ultimate goal of designing novel and efficient LVQ training algorithms with precise mathematical foundations.

The paper is organized as follows: in the next section we introduce the model, i.e. the specific learning scenarios and the assumed statistical properties of the training data. The mathematical description of the training dynamics is briefly summarized in Section 3, technical details are given in an Appendix. Results concerning the WTA schemes are presented and discussed in Section 4 and we conclude with a summary and an outlook on forthcoming projects.

2. The model

2.1. Winner-takes-all algorithms

We study situations in which input vectors $\boldsymbol{\xi} \in \mathbb{R}^N$ belong to one of two possible classes denoted as $\sigma = \pm 1$. Here, we restrict ourselves to the case of two prototype vectors w_S where the label $S = \pm 1$ (or $\pm$ for short) corresponds to the represented class.

In all WTA-schemes, the squared Euclidean distances $d_S(\boldsymbol{\xi}) = (\boldsymbol{\xi} - w_S)^2$ are evaluated for $S = \pm 1$ and the vector $\boldsymbol{\xi}$ is assigned to class σ if $d_{+\sigma} < d_{-\sigma}$.

We investigate incremental learning schemes in which a sequence of single, uncorrelated examples $\{\boldsymbol{\xi}^\mu, \sigma^\mu\}$ is presented to the system. The analysis can be applied to a larger variety of algorithms but here we treat only updates of the form

$$\boldsymbol{w}_S^\mu = \boldsymbol{w}_S^{\mu-1} + \Delta \boldsymbol{w}_S^\mu \quad \text{with}$$
$$\Delta \boldsymbol{w}_S^\mu = \frac{\eta}{N}\, \Theta_S^\mu g(S, \sigma^\mu)\, (\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1}). \tag{1}$$

Here, the vector $\boldsymbol{w}_S^\mu$ denotes the prototype after presentation of μ examples and the learning rate η is rescaled with the vector dimension N. The Heaviside term

$$\Theta_S^\mu = \Theta(d_{-S}^\mu - d_{+S}^\mu) \quad \text{with } \Theta(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{else} \end{cases}$$

singles out the current prototype $\boldsymbol{w}_S^{\mu-1}$ which is closest to the new input $\boldsymbol{\xi}^\mu$ in the sense of the measure $d_S^\mu = (\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1})^2$. In this formulation, only the *winner*, say, $\boldsymbol{w}_S$ can be updated whereas the *looser* $\boldsymbol{w}_{-S}$ remains unchanged. The change of the winner is always along the direction $\pm(\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1})$. The function $g(S, \sigma^\mu)$ further specifies the update rule. Here, we focus on three special cases of WTA learning:

(I) LVQ1: $g(S, \sigma) = S\sigma = +1$ (resp. -1) for $S = \sigma$ (resp. $S \neq \sigma$). This extension of competitive learning to labeled data corresponds to Kohonen's original LVQ1. The update is towards $\boldsymbol{\xi}^\mu$ if the example belongs to the class represented by the winning prototype, the *correct winner*. On the contrary, a *wrong winner* is moved away from the current input.

(II) LVQ+: $g(S, \sigma) = \Theta(S\sigma) = +1$ (resp. 0) for $S = \sigma$ (resp. $S \neq \sigma$). In this scheme the update is non-zero only for a correct winner and, then, always positive. Hence, a prototype $\boldsymbol{w}_S$ can only accumulate updates from its own class $\sigma = S$. We will use the abbreviation LVQ+ for this prescription.

(III) VQ: $g(S, \sigma) = 1$. This update rule disregards the actual data label and always moves the winner towards the example input. It corresponds to unsupervised vector quantization and aims at finding