Statistical Mechanics of On-line learning

Michael Biehl[1], Nestor Caticha[2], and Peter Riegler[3]

[1] University of Groningen, Inst. of Mathematics and Computing Science,
P.O. Box 407, 9700 AK Groningen, The Netherlands,
[2] Instituto de Fisica, Universidade de São Paulo,
CP66318, CEP 05315-970, São Paulo, SP Brazil,
[3] Fachhochschule Braunschweig/Wolfenbüttel, Fachbereich Informatik,
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Germany

Abstract. We introduce and discuss the application of statistical physics concepts in the context of on-line machine learning processes. The consideration of typical properties of very large systems allows to perfom averages over the randomness contained in the sequence of training data. It yields an exact mathematical description of the training dynamics in model scenarios. We present the basic concepts and results of the approach in terms of several examples, including the learning of linear separable rules, the training of multilayer neural networks, and Learning Vector Quantization.

1 Introduction

The perception that Statistical Mechanics is an inference theory has opened the possibility of applying its methods to several areas outside the traditional realms of Physics. This explains the interest of part of the Statistical Mechanics community during the last decades in machine learning and optimization and the application of several techniques of Statistical Mechanics of disordered systems in areas of Computer Science.

As an inference theory Statistical Mechanics is a Bayesian theory. Bayesian methods typically demand extended computational resources which probably explains why methods that were used by Laplace, were almost forgotten in the following century. The current availability of ubiquitous and now seemingly powerful computational resources has promoted the diffusion of these methods in statistics. For example, posterior averages can now be rapidly estimated by using efficient Markov Chain Monte Carlo methods, which started to be developed to attack problems in Statistical Mechanics in the middle of last century. Of course the drive to develop such techniques (starting with [1]) is due to the total impossibility of introducing classical statistics methods to study thermodynamic problems. Several other techniques Statistical Mechanics have also found their way into statistics. In this paper we review some applications of Statistical Mechanics to artificial machine learning.

Learning is the change induced by the arrival of information. We are interested in learning from examples. Different scenarios arise if examples are considered in batches or just one at a time. This last case is the on-line learning

scenario and the aim of this contribution is to present the characterization of on-line learning using methods of Statistical Mechanics. We will consider in section 2 simple linearly separable classification problems , just to introduce the idea. The dynamics of learning is described by a set of stochastic difference equations which show the evolution, as information arrives, of quantities that characterize the problem and in Statistical Mechanics are called order parameters. Looking at the limit of large dimensions and scaling the number of examples in a correct manner, the difference equations simplify into deterministic ordinary differential equations. Numerical solutions of the ODE give then, for the specific model of the available information such as the distribution of examples, the learning curves.

While this large dimension limit may seem artificial, it must be stressed that it is most sensible in the context of thermostatistics, where Statistical Mechanics is applied to obtain thermodynamic properties of physical systems. There the dimension, which is the number of degrees of freedom, is of the order of Avogadro's number ($\approx 10^{23}$). Simulations however have to be carried for much smaller systems, and this prompted the study of finite size corrections of expectations of experimentally relevant quantities, which depend on several factors, but typically go to zero algebraically with the dimension. If the interest lies in intensive quantities such as temperature, pressure or chemical potential then corrections are negligible. If one is interested on extensive quantities, such as energy, entropy or magnetization, one has to deal with their densities, such as energy per degree of freedom. Again the errors due to the infinite size limit are negligible. In this limit, central limit theorems apply , resulting in deterministic predictions and the theory is in the realm of thermodynamics. Thus this limit is known as the thermodynamic limit (TL). For inference problems we can use the type of theory to control finite size effects in the reverse order. We can calculate for the easier deterministic infinite limit and control the error made by taking the limit. We mention this and give references, but will not deal with this problem except by noticing that it is theoretically understood and that simulations of stylized models, necessarily done in finite dimensions, agree well with the theoretical results in the TL. The reader might consider that the thermodynamic limit is equivalent to the limit of infinite sequences in Shannon's channel theorem.

Statistical Mechanics (see e.g. [2,3]) had its origins in the second half of the XIX century, in an attempt, mainly due to Maxwell, Boltzmann and Gibbs to deduce from the microscopic dynamical properties of atoms and molecules, the macroscopic thermodynamic laws. Its success in building a unified theoretical framework which applied to large quantities of experimental setups was one of the great successes of Physics in the XIX century. A measure of its success can be seen from its role in the discovery of Quantum Mechanics. Late in the XIX century, when its application to a problem where the microscopic laws involved electromagnetism, i.e the problem of Black Body radiation, showed irreconcilable results with experiment, Max Planck showed that the problem laid with the classical laws of electromagnetism and not with Statistical Mechanics. This started the revolution that led to Quantum Mechanics.

This work is organized as follows. We introduce the method in section 2 the main ideas in a simple model. Section 3 will look into the extension of on-line methods to the richer case of two-layered networks which include universal approximators. In section 4 we present the latest advances in the area which deal with characterizing theoretically clustering methods such as Learning Vector Quantization (LVQ).

This paper is far from giving a complete overview of this successful approach to machine learning. Our intention is to illustrate the basic concepts in terms of selected examples, mainly from our own work. Also references are by no means complete and serve merely as a starting point for the interested reader.

2 On-line learning in Classifiers: Linearly separable case

We consider a supervised classification problem, where vectors $\boldsymbol{\xi} \in \mathbf{R}^N$ have to be classified into one of two classes which we label by $+1$ or -1 respectively. These vectors are drawn independently from a probability distribution $P_o(\boldsymbol{\xi})$. The available information is in the form of example pairs of vector-label: $(\boldsymbol{\xi}_\nu, \sigma_\nu)$, $\nu = 1, 2...\mu$. The scenario of on-line learning is defined by the fact that we take into account one pair at a time, which permits to identify ν and μ as time indexes. We also restrict our attention to the simple case where examples are used once to induce some change in our machine and then are discarded. While this seems quite inefficient since recycling examples to extract more information can indeed be useful, it permits to develop a simple theory due to the assumption of independence of the examples. The recycling of examples can also be treated ([4]) but it needs a repertoire of techniques that is beyond the scope of this review. For many simple cases this will be seen to be quite efficient.

As a measure of the efficiency of the learning algorithm we will concentrate on the generalization error, which is the probability of making a classification error on a new, statistically independent example $\boldsymbol{\xi}_{\mu+1}$. If any generalization is at all possible, of course there must be an underlying rule to generate the example labels, which is either deterministic

$$\sigma^B = f_B(\boldsymbol{\xi}) \tag{1}$$

or described by the conditional probability $P(\sigma^B | f_B(\boldsymbol{\xi}))$ depending on a transfer function f_B parameterized by a set of K unknown parameters B. At this point we take B to be fixed in time, a constraint that can be relaxed and still be studied within the theory, see [5,6,7,8].

Learning is the compression of information from the set of μ example pairs into a set of M weights $J_\mu \in \mathbf{R}^M$ and our machine classifies according to

$$\sigma^J = g_J(\boldsymbol{\xi}) \tag{2}$$

The generalization error is

$$e_G(\mu) = \int dP_o(\boldsymbol{\xi}) \int \prod_{\nu=1}^{\mu} dP_o(\boldsymbol{\xi}_\nu) \sum_{\sigma_\nu=\pm 1} P(\sigma_\nu^B | f_B(\boldsymbol{\xi}_\nu)) \Theta(-\sigma^{J_\mu} \sigma^B(\boldsymbol{\xi}))$$
$$= \langle \Theta(-\sigma^{J_\mu}(\boldsymbol{\xi}) \sigma^B(\boldsymbol{\xi})) \rangle_{\{\sigma_\nu, \boldsymbol{\xi}_\nu\}_{\nu=1,\mu}, \sigma, \boldsymbol{\xi}}, \tag{3}$$

where the step function $\Theta(x) = 1$ for $x > 0$ and zero otherwise. As this stands it is impossible to obtain results other than of a general nature. To obtain sharp results we have to specify a model.

The transfer functions f_B and g_J specify the architectures of the rule and of the classifier, while $P(\sigma_\mu | f_B(\boldsymbol{\xi}_\mu))$ models possible noise in the specification of the supervision label. The simplest case that can be studied is where both f_B and g_J are linearly separable classifiers of the same dimension: $K = M = N$,

$$\sigma^J = \text{sign}(\boldsymbol{J}.\boldsymbol{\xi}), \qquad \sigma^B = \text{sign}(\boldsymbol{B}.\boldsymbol{\xi}) \tag{4}$$

As simple and artificial as it may be, the study of this special case serves several purposes and is a stepping stone into more realistic scenarios.

An interesting feature of Statistical Mechanics lies in that it points out what are the relevant order parameters in a problem. In physics, this turns out to be information about what are the objects of experimental interest.

Without any loss we can take all vectors $\boldsymbol{\xi}$ and $\boldsymbol{B}$ to be normalized as $\boldsymbol{\xi} \cdot \boldsymbol{\xi} = N$ and $\boldsymbol{B} \cdot \boldsymbol{B} = 1$. For $\boldsymbol{J}$ however, which is a dynamical quantity that evolves under the learning algorithm still to be specified we let it free and call $\boldsymbol{J} \cdot \boldsymbol{J} = Q$. Define the fields

$$h = \boldsymbol{J} \cdot \boldsymbol{\xi}, \qquad b = \boldsymbol{B} \cdot \boldsymbol{\xi} \tag{5}$$

To advance further we choose a model for the distribution of examples $P_o(\boldsymbol{\xi})$ and the natural starting point is to choose a uniform distribution over the N-dimensional sphere. Different choices to model specific situation are of course possible. Under these assumptions, since the scalar products of (4) are sums of random variables, for large N, h and b are correlated Gaussian variables, completely characterized by

$$\begin{aligned} &\langle h \rangle = \langle b \rangle = 0, \\ &\langle h^2 \rangle = Q, \quad \langle b^2 \rangle = 1, \\ &\langle hb \rangle = \boldsymbol{J} \cdot \boldsymbol{B} = R. \end{aligned} \tag{6}$$

It is useful to introduce the overlap $\rho = R/\sqrt{Q}$ between the rule and machine parameter vectors. The joint distribution is given by

$$P(h, b) = \frac{1}{2\pi\sqrt{(1-\rho^2)}} e^{-\frac{1}{2(1-\rho^2)}(h^2 - 2\rho hb + b^2)}. \tag{7}$$

The correlation is the overlap ρ, which is related to the angle ϕ between $\boldsymbol{J}$ and $\boldsymbol{B}$: $\phi = \cos^{-1}\rho$, it follows that $|\rho| \leq 1$. It is geometrically intuitive and also easy

to prove that the probability of making an error on an independent example, the generalization error, is $\phi/2\pi$:

$$e_G = \frac{1}{2\pi}\cos^{-1}\rho \tag{8}$$

The strategy now is to introduce a learning algorithm, i.e to define the change that the inclusion of a new example causes in J, calculate the change in the overlap ρ and then obtain the learning curve for the generalization error. We will consider learning algorithms of the form

$$J_{\mu+1} = J_\mu + \frac{F}{N}\boldsymbol{\xi}_{\mu+1}, \tag{9}$$

where F, called the modulation function of vector $\boldsymbol{\xi}_{\mu+1}$, should depend on the supervised information, the label $\sigma^B_{\mu+1}$. It may very well depend on some additional information carried by hyperparameters or on $\boldsymbol{\xi}$ itself. It is F that defines the learning algorithm. We consider the case where F is a scalar function, but it could differ for different components of $\boldsymbol{\xi}$. Projecting (9) into $\boldsymbol{B}$ and into $\boldsymbol{J}$ we obtain respectively

$$R_{\mu+1} = R_\mu + \frac{F}{N}b_\mu \tag{10}$$

$$Q_{\mu+1} = Q_\mu + 2\frac{F}{N}h_\mu + \frac{F^2}{N}. \tag{11}$$

which describe the learning dynamics. We can also write an equivalent equation for the overlap ρ which is valid for large N and $\boldsymbol{\xi}$ on the hypersphere and

$$\begin{aligned}\rho_{\mu+1} &= \frac{J_{\mu+1}.B}{\sqrt{Q_{\mu+1}}} = \\ &= \rho_\mu\left(1 - \frac{1}{N}\frac{F}{\sqrt{Q_\mu}}h_{\mu+1} - \frac{1}{2N}(\frac{F}{\sqrt{Q_\mu}})^2\right) + \frac{1}{N}\frac{F}{\sqrt{Q_\mu}}b_{\mu+1}\end{aligned} \tag{12}$$

$$\Delta\rho_{\mu+1} = \frac{1}{N}\frac{F}{\sqrt{Q_\mu}}(b_{\mu+1} - \rho_\mu h_{\mu+1}) - \frac{1}{2N}\frac{\rho_\mu F^2}{Q_\mu} \tag{13}$$

Since at each time step μ a random vector is drawn from the distribution P_o equations (12) and (13) are stochastic difference equations. We now take the thermodynamic limit $N \to \infty$ and average over the test example $\boldsymbol{\xi}$. Note that each example induces a change of the order parameters of order $1/N$. Hence, one expects the need of order N many examples to create a change of order 1. This prompts the introduction of $\alpha = \lim_{N\to\infty}\mu/N$ which by measuring the number of examples measures time. The behavior of ρ and $\frac{\Delta\rho}{\Delta\alpha}$ are very different in the limit. It can be shown (see [9]) that order parameters such as ρ, R, Q self-average. Their fluctuations tend to zero in this limit, see Fig. 3 for an example. On the other hand $\frac{\Delta\rho}{\Delta\alpha}$ has fluctuations of order one and we look at its average over the test vector:

$$\frac{d\rho}{d\alpha} = \langle\frac{\Delta\rho}{\Delta\alpha}\rangle_{h,b,\xi}. \tag{14}$$

the pairs (Q, $\Delta Q/\Delta\alpha$) and (R, $\Delta R/\Delta\alpha$) behave in a similar way. We average over the fields h, b and over the labels σ, using $P(\sigma|b)$ .This leads to the coupled system of ordinary differential equations, which for a particular form of F were introduced in [10].

$$\frac{dR}{d\alpha} = \sum_{\sigma} \int dhdbP(h,b)P(\sigma|b)\,[Fb] = \langle Fb\rangle \tag{15}$$

$$\frac{dQ}{d\alpha} = \sum_{\sigma} \int dhdbP(h,b)P(\sigma|b)\left[2Fh + F^2\right] = \langle 2Fh + F^2\rangle, \tag{16}$$

where the angular brackets stand for the average over the fields and label. Since the generalization error is directly related to ρ it will be useful to look at the equivalent set of equations for ρ and the length of the weight vector $\sqrt{Q}$:

$$\frac{d\rho}{d\alpha} = \sum_{\sigma} \int dhdbP(h,b)P(\sigma|b)\left[\frac{F}{\sqrt{Q}}(b-\rho h) - \frac{\rho F^2}{2Q}\right] \tag{17}$$

$$\frac{d\sqrt{Q}}{d\alpha} = \sum_{\sigma} \int dhdbP(h,b)P(\sigma|b)\left[Fh + \frac{1}{2}\frac{F^2}{\sqrt{Q}}\right] \tag{18}$$

We took the average with respect to the two fields h and b as if they stood on symmetrical grounds, but they don't. It is reasonable to assume knowledge of h and σ but not of b. Making this explicit

$$\frac{d\rho}{d\alpha} = \sum_{\sigma} \int dhP(h)P(\sigma)\left[\frac{F}{\sqrt{Q}}\langle b-\rho h\rangle_{b|\sigma h} - \frac{\rho F^2}{2Q}\right] \tag{19}$$

$$\frac{d\sqrt{Q}}{d\alpha} = \sum_{\sigma} \int dhP(h)P(\sigma)\left[Fh + \frac{1}{2}\frac{F^2}{\sqrt{Q}}\right] \tag{20}$$

call

$$F^* = \frac{\sqrt{Q}}{\rho}\langle b-\rho h\rangle_{b|\sigma h} \tag{21}$$

where the average is over unavailable quantities. Equation 19 can then be written as

$$\frac{d\rho}{d\alpha} = \frac{\rho}{Q}\sum_{\sigma} \int dhP(h)P(\sigma)\left[FF^* - \frac{1}{2}F^2\right] \tag{22}$$

This notation makes it natural to ask for an interpretation of the meaning of F^*. The differential equations above describe the dynamics for *any* modulation function. We can ask ([11]) if there is a modulation function optimal in the sense of maximizing the information gain per example, which can be stated as a variational problem

$$\frac{\delta}{\delta F}\left(\frac{d\rho}{d\alpha}\right) = 0 \tag{23}$$

This problem can be solved for a few architectures, including some networks with hidden layers [12,16], although the optimization becomes much more difficult. Within the class of algorithms we have considered, the performance bound is given by the modulation function F^* given by Eq. (21).

The optimal resulting algorithm has several features that are useful in considering practical algorithms, such as the optimal annealing schedule of the leaning rate and, in the presence of noise, adaptive cutoffs for surprising examples.

As an example we discuss learning of a linearly separable rule (4) in some detail. There, an example will be misclassified if $\theta(-\sigma^J\sigma^B) > 0$ or equivalently $h\sigma^B < 0$. We will refer to the latter quantity as aligned field. It basically measures the correctness ($h\sigma^B > 0$) or wrongness ($h\sigma^B < 0$) of the current classification.

Fig. 1 depicts the optimal modulation function for inferring a linearly separable rule from linearly separable data. Most interesting is the dependence on $\rho = \cos(\pi\epsilon_g)$: For $\rho = 0$ ($\epsilon_g = 1/2$) the modulation function does not take into account the correctness of the current examples. The modulation function is constant, which corresponds to Hebbian learning. As ρ increases, however, for already correctly classified examples the magnitude of the modulations function decreases with increasing aligned field. For misclassified examples, however, the update becomes the larger the smaller the aligned field is. In the limit $\rho \to 1$ ($\epsilon_g \to \infty$) the optimal modulation function approaches the Adatron algorithm ([13]) where only misclassified examples trigger an update which is proportional to the aligned field. In addition to that, for the optimal modulation function $\rho(\alpha) = \sqrt{Q(\alpha)}$, *i.e.* the order parameter Q can be used to estimate ρ.

Now imagine, that the label of linearly separable data is noisy, *i.e.* it is changed with a certain probability. In this case it would be very dangerous to follow an Adatron-like algorithm and perform an large update if $h\sigma^B < 0$, since the seeming misclassification might be do to a corrupted label. The optimal modulation function for that case perceives this danger and introduces a sort of cutoff w.r.t. the aligned field. Hence, no considerable update is performed if the aligned field becomes too large in magnitude. Fig. 1 shows the general behavior. [14] gives further details and an extension to other scenarios of noisy but otherwise linearly separable data sets.

These results for optimal modulations functions are, in general, better understood from a Bayesian point of view ([17,18,19]). Suppose the knowledge about the weight vector is coded in a Gaussian probability density. As a new example arrives, this probability density is used as a prior distribution. The likelihood is built out of the knowledge of the network architecture, of the noise process that may be corrupting the label and of the example vector and its label. The new posterior is not, in general Gaussian and a new Gaussian is chosen, to be the prior for the next learning step, in such a way as to minimize the information loss. This is done by the maximum entropy method or equivalently minimizing the Kullback-Leibler divergence. It follows that the learning of one example induces a mapping from a Gaussian to another Gaussian, which can be described by update equations of the Gaussian's mean and covariance. These equations define a learning algorithm together with a tensor like adaptive schedule for the

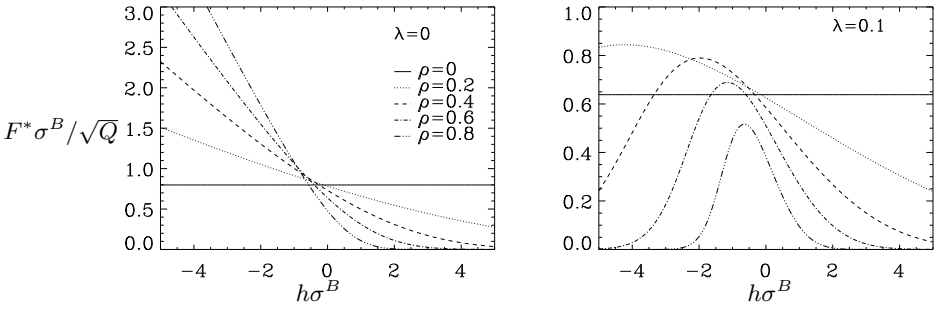

Fig. 1. Left: Optimal modulation function F^* for learning a linearly separable classification of type (4) for various values of the order parameter ρ. Right: In addition to (4) the labels σ^B are subject to noise, i.e. are flipped with probability λ. The aligned field $h\sigma^B$ is a measure of the agreement of the classification of the current example with the given label. In both cases, all examples are weighted equally for $\rho = 0$ irrespective of the value of the aligned field $h\sigma^B$. This corresponds to Hebbian learning. In the noiseless case ($\lambda = 0$) examples with a negative aligned field receive more weight as ρ increases while those with positive aligned field gain less weight. For $\lambda > 0$ also examples with large negative aligned fields are not taken into account for updating. These examples are most likely affected by the noise and therefore are deceptively misclassified. The optimal weight function possesses a cutoff value at negative values of $h\sigma^B$. This cutoff decreases in absolute value with increasing λ and ρ.

learning rate. This algorithms are closely related to the variational algorithm defined above. The Bayesian method has the advantage of being more general. It can also be readily applied to other cases where the Gaussian prior is not adequate [19].

In the following sections we follow the developments of these techniques in other directions. We first look into networks with hidden layers, since the experience gained in studying on-line learning in these more complex architectures will be important to better understand the application of on-line learning to the problem of clustering.

3 On-line Learning in Two-Layered Networks

Classifiers such as (4) are the most fundamental learning networks. In terms of network architecture the next generalization are networks with one layer of hidden units and a fixed hidden-to-output relation. An important example is the so-called committee machine, where the overall output is determined by a majority vote of several classifiers of type (4). For such networks the variational approach of the previous section can be readily applied [12,16].

General two-layered networks, however, have variable hidden-to-output weights and are "soft classifiers", *i.e.* have continuous transfer functions. They consist

of a layer of N inputs, a layer of K hidden units, and a single output. The particular input-output relation is given by

$$\sigma(\boldsymbol{\xi}) = \sum_{i=1}^{K} w_i g(\boldsymbol{J}_i \cdot \boldsymbol{\xi}). \tag{24}$$

Here, $\boldsymbol{J_i}$ denotes the N-dimensional weight vector of the i-th input branch and w_i the weight connecting the i-th hidden unit with the output. For a (soft) committee machine, $w_i \equiv 0$ for all branches i. Often, the number K of hidden weight vectors $\boldsymbol{J_i}$ is chosen as $K \ll N$. In fact, most analyses specialize on $K = \mathcal{O}(1)$. This restriction will also be pursued here. Note that the overall output is linear in w_i, in contrast to the outputs of the hidden layer which in general depend nonlinearly on the weights $\boldsymbol{J_i}$ via the transfer function g.

The class of networks (24) is of importance for several reasons: Firstly, they can realize more complex classification schemes. Secondly, they are commonly used in practical applications. Finally, two-layered networks with a linear output are known to be universal approximators [20].

Analogously to section 2 the network (24) is trained by a sequence of uncorrelated examples $\{(\boldsymbol{\xi}^\nu, \tau^\nu)\}$ which are provided by an unknown rule $\tau(\boldsymbol{\xi})$ by the environment. As above, the example input vectors are denoted by $\boldsymbol{\xi}^\mu$, while here τ^μ is the corresponding correct rule output.

In a commonly studied scenario the rule is provided by a network of the same architecture with hidden layer weights $\boldsymbol{B}_i$, hidden-to-output weights v_i, and an in general different number of hidden units M:

$$\tau(\boldsymbol{\xi}) = \sum_{k=1}^{M} v_k g(\boldsymbol{B}_k \cdot \boldsymbol{\xi}). \tag{25}$$

In principle, the network (24) can implement such a function if $K \geq M$.

As in (9) the change of a weight is usually taken proportional to the input of the corresponding layer in the network, *i.e.*

$$\boldsymbol{J_i}^{\mu+1} = \boldsymbol{J_i}^{\mu} + \frac{1}{N} F_i \boldsymbol{\xi}^{\mu+1} \tag{26}$$

$$w_i^{\mu+1} = w_i^{\mu} + \frac{1}{N} F_w g(\boldsymbol{J_i} \cdot \boldsymbol{\xi}^{\mu+1}). \tag{27}$$

Again, the modulation functions F_i, F_w will in general depend on the recently provided information $(\boldsymbol{\xi}^\mu, \tau^\mu)$ and the current weights.

Note, however, that there is an asymmetry between the updates of $\boldsymbol{J_i}$ and w_i. The change of the former is $\mathcal{O}(1/N)$ due to $|\boldsymbol{\xi}^2| = \mathcal{O}(N)$. As $\sum_{i=1}^{K} g^2(\boldsymbol{J_i} \cdot \boldsymbol{\xi}) = \mathcal{O}(K)$ a change of the latter according to

$$w_i^{\mu+1} = w_i^{\mu} + \frac{1}{K} F_w g(\boldsymbol{J_i} \cdot \boldsymbol{\xi}^{\mu+1}). \tag{28}$$

seems to be more reasonable. For reasons that will become clear below we will prefer a scaling with $1/N$ as in (27) over a scaling with $1/K$, at least for the time being.

Also note from (24, 25) that the stochastic dynamics of $\boldsymbol{J_i}$ and w_i only depends on the fields $h_i = \boldsymbol{J_i}\cdot\boldsymbol{\xi}$, $b_k = \boldsymbol{B_k}\cdot\boldsymbol{\xi}$ which can be viewed as a generalization of (ref to eq 5). As in section 2, for large N these become Gaussian variables. Here, they have zero means and correlations

$$\langle h_i h_j\rangle = \boldsymbol{J_i}\cdot\boldsymbol{J_j} =: Q_{ij}\ ,\ \ \langle b_k b_l\rangle = \boldsymbol{B_k}\cdot\boldsymbol{B_l} =: T_{kl}\ ,\ \ \langle h_i b_k\rangle = \boldsymbol{J_i}\cdot\boldsymbol{B_k} =: R_{ik}, \quad (29)$$

where $i,j = 1\ldots K$ and $k,l = 1\ldots M$.

Introducing $\alpha = \mu/N$ as above, the discrete dynamics (26, 27) can be replaced by a set of coupled differential equations for R_{ik}, Q_{ij}, and w_i in the limit of large N: Projecting (26) into $\boldsymbol{B_k}$ and $\boldsymbol{J_j}$, respectively, and averaging over the randomness of $\boldsymbol{\xi}$ leads to

$$\frac{dR_{ik}}{d\alpha} = \langle F_i b_k\rangle \quad (30)$$

$$\frac{dQ_{ij}}{d\alpha} = \langle F_i h_j + F_j h_i + F_i F_j\rangle, \quad (31)$$

where the average is now with respect to the fields $\{h_i\}$ and $\{b_k\}$. Hence, the microscopic stochastic dynamics of $\mathcal{O}(K\cdot N)$ many weights $\boldsymbol{J_i}$ is replaced by the macroscopic dynamics of $\mathcal{O}(K^2)$ many order parameters R_{ik} and Q_{ij}, respectively. Again, these order parameters are self-averaging, *i.e.* their fluctuations vanish as $N\to\infty$. Fig. 3 exemplifies this for a specific dynamics.

The situation is somewhat different for the hidden-to-output weights w_i. In the transition from microscopic, stochastic dynamics to macroscopic, averaged dynamics the hidden-layer weights $\boldsymbol{J_i}$ are compressed to order parameters which are scalar products, cf. (29). The hidden-to-output weights, however, are not compressed into new parameters of the form of scalar products. (Scalar products of the type $\sum_i w_i v_i$ do not even exist for $K\neq M$.) Scaling the update of w_i by $1/N$ as in (27) allows to replace $1/N$ by the differential $d\alpha$ as $N\to\infty$. Hence, the averaged dynamics of the hidden-to-output weight reads

$$\frac{dw_i}{d\alpha} = \langle F_w g(h_i)\rangle. \quad (32)$$

Note that the r.h.s. of these differential equations depend on R_{ik}, Q_{ij} via (29) and, hence, are coupled to the differential equations (30, 31) of these order parameters as well. So in total, the macroscopic description of learning dynamics consists of the coupled set (30, 31, 32).

It might be surprising that the hidden-to-output weights w_i by themselves are appropriate for a macroscopic description while the hidden weights $\boldsymbol{J_i}$ are not. The reason for this is twofold. First, the number K of w_i had been taken to be $\mathcal{O}(1)$, *i.e.* it does not scale with the dimension of inputs N. Therefore, there is no need to compress a large number of microscopic variables into a small number of macroscopic order parameters as for the $\boldsymbol{J_i}$. Second, the change in w_i had been chosen to scale with $1/N$. For this choice one can show that like R_{ik} and Q_{ij} the weights w_i are self-averaging.

For a given rule $\tau(\boldsymbol{\xi})$ to be learned, the generalization error is

$$\epsilon_g(\{\boldsymbol{J_i}, w_i\}) = \langle \epsilon(\{\boldsymbol{J_i}, w_i\}) \rangle_{\boldsymbol{\xi}}, \tag{33}$$

where $\epsilon(\{\boldsymbol{J_i}, w_i\}) = \frac{1}{2}(\sigma - \tau)^2$ is the instantaneous error. As the outputs σ and τ depend on $\boldsymbol{\xi}$ only via the fields $\{h_i\}$ and $\{b_k\}$, respectively, the average over $\boldsymbol{\xi}$ can be replaced by an average over these fields. Hence, the generalization error only depends on order parameters R_{ik}, Q_{ij}, w_i as well as on T_{kl} and v_k.

In contrast to section 2 it is a difficult task to derive optimal algorithms by a variational approach, since the generalization error (33) is a function of several order parameters. Therefore on-line dynamics in two-layer networks has mostly been studied in the setting of heuristic algorithms, in particular for backpropagation. There, the update of the weights is taken proportional to the gradient of the instantaneous error $\epsilon = \epsilon(\{\boldsymbol{J_i}, w_i\}) = \frac{1}{2}(\sigma - \tau)^2$ with respect to the weights:

$$\boldsymbol{J_i}^{\mu+1} = \boldsymbol{J_i}^{\mu} - \frac{\eta_J}{N} \boldsymbol{\nabla}_{\boldsymbol{J_i}} \epsilon \tag{34}$$

$$w_i^{\mu+1} = w_i^{\mu} - \frac{\eta_w}{N} \frac{\partial \epsilon}{\partial w_i} \tag{35}$$

The parameters η_J and η_w denote learning rates which scale the size of the update along the gradient of the instantaneous error. In terms of (26, 27) backpropagation corresponds to the special choices $F_J = -\eta_J(\sigma - \tau) w_i g'(h_i)$ and $F_w = -\eta_w(\sigma - \tau)$. A common choice for the transfer function is $g(x) = \mathrm{erf}(x/\sqrt{2})$. With this specific choice, the averaging in the equations of motion (30, 31, 32) can be performed analytically for general K and M [22,23,24]

Independent of the particular choice of learning algorithms a general problem in two-layered networks is caused by the inherent permutation symmetry: The i-th input branch of the adaptive network (24) does not necessarily specialize on the i-th branch in the network (25). Without loss of generality, however, one can relabel the dynamical variables such as if this were indeed the case. Nevertheless, this permutation symmetry will turn out to be a dominant feature because it leads to a deceleration of learning due to plateau states in the dynamics of the generalization error. Fig. 2 gives an example.

These plateau states correspond to configurations which are very close to certain fixed points of the set of differential equations (30, 31, 32) for the order parameters. In the simplest case the vectors $\boldsymbol{J_i}$ are almost identical during the plateau phase. They have – apart from small deviations – the same scalar product with each vector $\boldsymbol{B_k}$. These fixed points are repulsive, so small fluctuations will cause a specialization of each $\boldsymbol{J_i}$ towards a distinct $\boldsymbol{B_k}$ which then leads to minimum generalization error. If there are several such repulsive fixed points there can even be cascades of plateaus. The lengths of the plateaus can be shown to depend crucially on the choice of initial conditions as well as on the dimension N of the inputs [25].

For backpropagation, the differential equations (30, 31, 32) can easily be used to determine those learning rates η_J and η_w which lead to the fastest decrease of the generalization error. This is most interesting for the learning rate η_w of

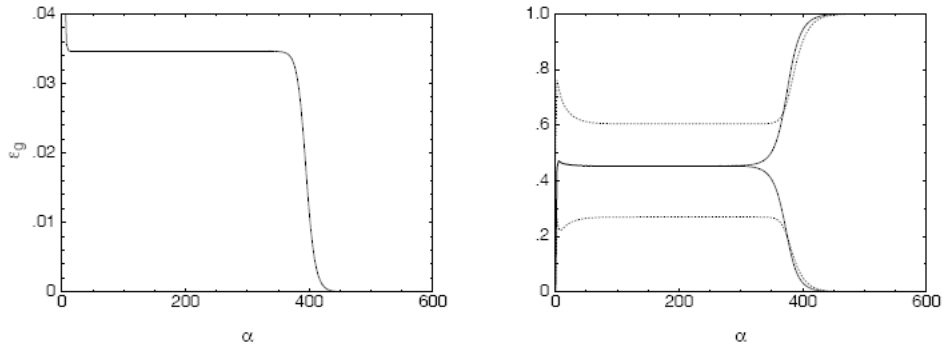

Fig. 2. Time evolution for on-line backpropagation in the $K = M = 2$ learning scenario (24,25,34) with $T_{nm} = \delta_{nm}$ and fixed $w_i = v_i$. Left: generalization error $\epsilon_g(\alpha)$. Right: order parameters R_{in} (full curves) and Q_{ik} (dotted curves). The plateaus in both graphs are due to the internal permutation symmetry of(25) w.r.t. the summation index.

the hidden-to-output weights as it turns out that the decrease of $\epsilon_g(\alpha)$ is largest as $\eta_w \to \infty$.

Obviously, this divergence of η_w indicates that one should have chosen a different scaling for the change of the weights w_i, namely a scaling with $1/K$ as in (28) as opposed to (27). For such a scaling, the weights w_i will not be self-averaging anymore, however, see Fig. 3. Hence, equations (30, 31, 32) fail to provide a macroscopic description of the learning dynamics in this case. This does by no means signify that they are inapplicable, however. The optimal choice $\eta_w \to \infty$ simply indicates that the dynamics of the hidden-to-output weights w_i is on a much faster time scale compared to the time scale α on which the self-averaging quantities R_{ik} and Q_{ij} change.

An appropriate method to do deal with such situations is known as adiabatic elimination. It relies on the assumption that the mean value of the fast variable has a stable equilibrium value at each macroscopic time α. One obtains this equilibrium value from the zero of the r.h.s. of (32) with respect to w_i, *i.e.* by investigating the case $dw_i/d\alpha = 0$. The equilibrium values for w_i are thus obtained as functions of R_{ik} and Q_{ij} and can be further used to eliminate any appearance of w_i in (30, 31). See [21] for details.

The variational approach discussed in Sec. 2 has also been applied to the analysis of multilayered networks, in particular the soft committee machine [16,26]. Due to the larger number of order parameters the treatment is much more involved than for the simple perceptron network. The investigations show that, in principle, it is possible to reduce the length of the plateau states as discussed above drastically by using appropriate training prescriptions.

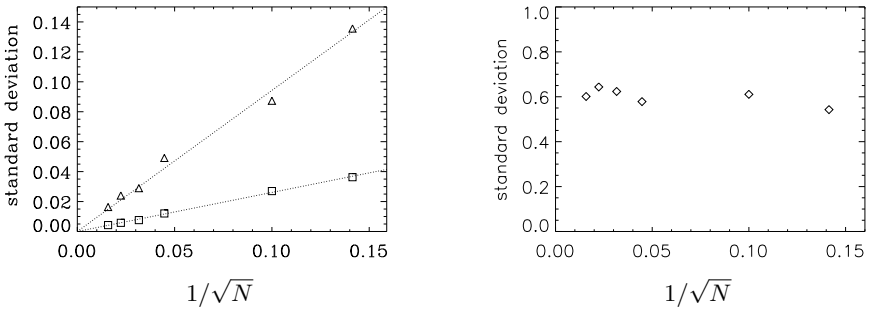

Fig. 3. Finite size analysis of the order parameters R ($\triangle$), Q ($\square$) and w ($\circ$, right panel) in the dynamics (34,35) for the special case $K{=}M{=}1$. Shown are the observed standard deviations as a function of the system size for a fixed value of α. Each point depicts an average taken over 100 simulation runs. As can be seen, R and Q become selfaveraging in the thermodynamic limit $N \to \infty$, *i.e.* their fluctuations vanish in this limit. In contrast to that, the fluctuations of w remain finite if one optimizes the learning rate η_w, which leads to the divergence $\eta_w \to \infty$.

4 Dynamics of prototype based learning

In all training scenarios discussed above, the consideration of isotropic, i.i.d. input data yields non-trivial insights, already. The key information is contained in the training labels and, for modeling purposes, we have assumed that they are provided by a teacher network.

In practical situations one would clearly expect the presence of structures in input space, e.g. in the form of clusters which are more or less correlated with the target function. Here we briefly discuss how the theoretical framework has been extended in this direction. We will address, mainly, supervised learning schemes which detect or make use of structures in input space. Unsupervised learning from unlabeled, structured data has been treated along the very same lines but will not be discussed in detail, here. We refer to, for instance, [27,28] for overviews and [34,35,38,43,45] for example results in this context.

We will focus on prototype based supervised learning schemes which take into account label information. The popular Learning Vector Quantization algorithm [32] and many of its variants follow the lines of competitive learning. However the aim is to obtain prototypes as typical representatives of their classes which parameterize a distance based classification scheme.

LVQ algorithms can be treated within the same framework as above. The analysis requires only slight modifications due to the assumption of a non-trivial input densities.

Several possibilities to model anisotropy in input space have been considered in the literature, a prominent example being unimodal Gaussians with distinct principal axes [30,29,34,35]. Here, we focus on another simple but non-trivial

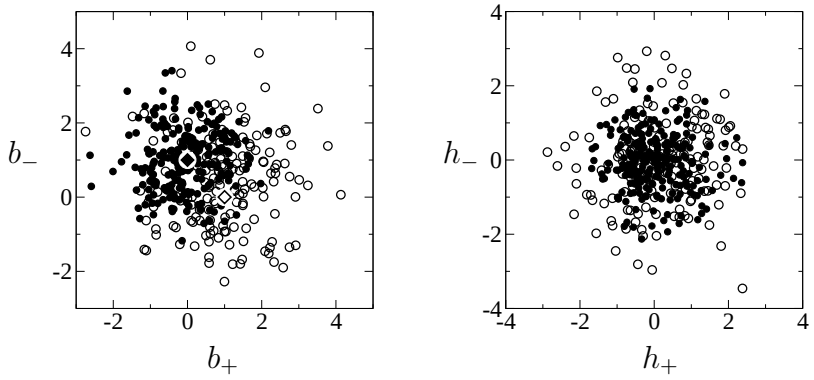

Fig. 4. Data as generated according to the density (36) in $N = 200$ dimensions with example parameters $p_- = 0.6, p_+ = 0.4$, $v_- = 0.64$, and $v_+ = 1.44$. The open (filled) circles represent 160 (240) vectors $\boldsymbol{\xi}$ from clusters centered about orthonormal vectors $\ell \boldsymbol{B}_+$ ($\ell \boldsymbol{B}_-$) with $\ell = 1$, respectively. The left panel displays to the projections $b_\pm = \boldsymbol{B}_\pm \cdot \boldsymbol{\xi}$ and diamonds mark the position of the cluster centers. The right panel shows projections $h_\pm = \boldsymbol{w}_\pm \cdot \boldsymbol{\xi}$ of the same data on a randomly chosen pair of orthogonal unit vectors $\boldsymbol{w}_\pm$.

model density: we assume feature vectors are generated independently according to a mixture of isotropic Gaussians:

$$P(\boldsymbol{\xi}) = \sum_{\sigma=\pm 1} p_\sigma P(\boldsymbol{\xi} \mid \sigma) \text{ with } P(\boldsymbol{\xi} \mid \sigma) = \frac{1}{\sqrt{2\pi v_\sigma}^N} \exp\left[-\frac{1}{2\, v_\sigma} \left(\boldsymbol{\xi} - \ell \boldsymbol{B}_\sigma\right)^2\right]. \tag{36}$$

The conditional densities $P(\boldsymbol{\xi} \mid \sigma = \pm 1)$ correspond to clusters with variances v_σ centered at $\ell\, \boldsymbol{B}_\sigma$. For convenience, we assume that the vectors $\boldsymbol{B}_\sigma$ are orthonormal: $\boldsymbol{B}_\sigma^2 = 1$ and $\boldsymbol{B}_+ \cdot \boldsymbol{B}_- = 0$. The first condition sets only the scale on which the parameter ℓ controls the cluster offset. The orthogonality condition fixes the position of cluster centers with respect to the origin in feature space which could be chosen arbitrarily. Similar densities have been considered in, for instance, [31,37,38,39,40].

In the context of supervised Learning Vector Quantization, see next section, we will assume that the target classification coincides with the cluster membership label σ for each vector $\boldsymbol{\xi}$. Due to the significant overlap of clusters, this task is obviously not linear separable.

Note that linearly separable rules defined for bimodal input data similar to (36) have been studied in [36]. While transient learning curves of the perceptron can differ significantly from the technically simpler case of isotropic inputs, the main results concerning the ($\alpha \to \infty$)–asymptotic behavior persist.

Learning Vector Quantization (LVQ) a particularly intuitive and powerful family of algorithms which has been applied in a variety of practical problems

[41]. LVQ identifies prototypes, i.e. typical representatives of the classes in feature space which then parameterize a distance based classification scheme.

Competitive learning schemes have been suggested in which a set of prototype vectors is updated from a sequence of example data. We will restrict ourselves to the simplest non-trivial case of two prototypes $\boldsymbol{w}_+, \boldsymbol{w}_- \in I\!R^N$ and data generated according to a bi-modal distribution of type (36).

Most frequently a *nearest prototype* scheme is implemented: For classification of a given input $\boldsymbol{\xi}$, the distances

$$d_s(\boldsymbol{\xi}) = (\boldsymbol{\xi} - \boldsymbol{w}_s)^2 \,, \;\; s = \pm 1 \tag{37}$$

are determined and $\boldsymbol{\xi}$ is assigned to class $+1$ if $d_+(\boldsymbol{\xi}) \leq d_-(\boldsymbol{\xi})$ and to class -1 else. The (squared) Euclidean distance (37) appears to be a natural choice. In practical situations, however, it can lead to inferior performance and the identification of an appropriate distance or similarity measure is one of the key issues in applications of LVQ.

A simple two prototype system as described above parameterizes a linearly separable classifier, only. However, we will consider learning of a non-separable rule where non-trivial effects of the prototype dynamics can be studied in this simple setting already. Extensions to more complex models with several prototypes, i.e. piecewise linear decision boundaries and multi-modal input densities are possible but non-trivial, see [45] for a recent example.

Generically, LVQ algorithms perform updates of the form

$$\boldsymbol{w}_s^{\mu+1} = \boldsymbol{w}_s^{\mu} + \frac{\eta}{N} f(d_+^{\mu}, d_-^{\mu}, s, \sigma^{\mu}) \, (\boldsymbol{\xi}^{\mu} - \boldsymbol{w}_s^{\mu}) \,. \tag{38}$$

Hence, prototypes are either moved towards or away from the current input. Here, the modulation function f controls the sign and, together with an explicit learning rate η, the magnitude of the update.

So–called Winner-Takes-All (WTA) schemes update only the prototype which is currently closest to the presented input vector. A prominent example of supervised WTA learning is Kohonen's original formulation, termed LVQ1 [32,33]. In our model scenario it corresponds to Eq. (38) with

$$f(d_+^{\mu}, d_-^{\mu}, s, \sigma^{\mu}) \; = \; \Theta(d_{-s}^{\mu} - d_{+s}^{\mu}) \, s \, \sigma^{\mu} \tag{39}$$

The Heaviside function singles out the winning prototype, and the product $s\,\sigma^{\mu} = +1(-1)$ if the labels of prototype and example coincide (disagree).

For the formal analysis of the training dynamics, we can proceed in complete analogy to the previously studied cases of perceptron and layered neural networks. A natural choice of order parameters are the self- and cross-overlaps of the involved N-dimensional vectors:

$$R_{s\sigma} = \boldsymbol{w}_s \cdot \boldsymbol{B}_{\sigma} \quad \text{and} \quad Q_{st} = \boldsymbol{w}_s \cdot \boldsymbol{w}_t \quad \text{with} \quad \sigma, s, t \in \{-1, +1\} \tag{40}$$

While these definitions are formally identical with Eq. (29), the role of the reference vectors $\boldsymbol{B}_{\sigma}$ is not that of *teacher vectors*, here.

Following the by now familiar lines we obtain a set of coupled ODE of the form

$$\begin{aligned}\frac{dR_{S\tau}}{d\alpha} &= \eta\,(\langle b_\tau\, f_S\rangle \;-\; R_{S\tau}\;\langle f_S\rangle)\\ \frac{dQ_{ST}}{d\alpha} &= \eta\,(\langle h_S\, f_T + h_T\, f_S\rangle \;-\; Q_{ST}\;\langle f_S \;+ f_T\rangle)\\ &\quad +\eta^2 \sum_{\sigma=\pm 1} v_\sigma p_\sigma \left\langle f_S\, f_T\right\rangle_\sigma .\end{aligned} \tag{41}$$

Here, averages $\langle\ldots\rangle$ over the full density $P(\boldsymbol{\xi})$, Eq. (36) have to be evaluated as appropriate sums over conditional averages $\langle\ldots\rangle_\sigma$ corresponding to $\boldsymbol{\xi}$ drawn from cluster σ:

$$\langle\ldots\rangle \;=\; p_+\;\langle\ldots\rangle_+ + p_-\;\langle\ldots\rangle_- .$$

For a large class of LVQ modulation functions, the actual input $\boldsymbol{\xi}^\mu$ appears on the right hand side of Eq. (41) only through its length and the projections

$$h_s \;=\boldsymbol{w}_s\cdot\boldsymbol{\xi} \;\text{ and }\; b_\sigma \;=\boldsymbol{B}_\sigma\cdot\boldsymbol{\xi} \tag{42}$$

where we omitted indices μ but implicitly assume that the input $\boldsymbol{\xi}$ is uncorrelated with the current prototypes $\boldsymbol{w}_s$. Note that also Heaviside terms as in Eq. (39) do not depend on $\boldsymbol{\xi}$ explicitly, for example:

$$\Theta\,(d_- - d_+) = \Theta\,[+2(h_+ - h_-) - Q_{++} + Q_{--}]\,.$$

When performing the average over the actual example $\boldsymbol{\xi}$ we first exploit the fact that

$$\lim_{N\to\infty}\left\langle\boldsymbol{\xi}^2\right\rangle/N \;=\; (v_+p_+ + v_-p_-)$$

for all input vectors in the thermodynamic limit. Furthermore, the joint Gaussian density $P(h_+^\mu, h_-^\mu, b_+^\mu, b_-^\mu)$ can be expressed as a sum over contributions from the clusters. The respective conditional densities are fully specified by first and second moments

$$\begin{aligned}&\langle h_s\rangle_\sigma = \ell R_{s\sigma}, \quad \langle b_\tau\rangle_\sigma = \ell\delta_{\tau\sigma}, \quad \langle h_s h_t\rangle_\sigma - \langle h_s\rangle_\sigma\,\langle h_t\rangle_\sigma = v_\sigma\, Q_{st}\\ &\langle h_s b_\tau\rangle_\sigma - \langle h_s\rangle_\sigma\,\langle b_\tau\rangle_\sigma = v_\sigma\, R_{s\tau}, \quad \langle b_\rho b_\tau\rangle_\sigma - \langle b_\rho\rangle_\sigma\,\langle b_\tau\rangle_\sigma = v_\sigma\,\delta_{\rho\tau}\end{aligned} \tag{43}$$

where $s,t,\sigma,\rho,\tau\in\{+1,-1\}$ and $\delta_{\ldots}$ is the Kronecker-Delta. Hence, the density of $h_\pm$ and $b_\pm$ is given in terms of the model parameters $\ell, p_\pm, v_\pm$, and the above defined set of order parameters in the previous time step.

After working out the system of ODE for a specific modulation function, it can be integrated, at least numerically. Here we consider prototypes that are initialized as independent random vectors of squared length $\hat{Q}$ with no prior knowledge about the cluster positions. In terms of order parameters this implies in our simple model

$$Q_{++}(0) = Q_{--}(0) = \hat{Q},\ \text{and}\ Q_{+-}(0) = R_{S\sigma}(0) = 0 \quad \text{for all}\ \ S,\sigma. \tag{44}$$

As in any supervised scenario, the success of learning is to be quantified in terms of the generalization error. Here we have to consider two contributions for misclassifying data from cluster $\sigma = 1$ or $\sigma = -1$ separately:

$$\epsilon \;=\; p_+\,\epsilon_+ \;+\; p_-\,\epsilon_- \quad \text{with} \quad \epsilon_\sigma \;=\; \langle \Theta\left(d_{+\sigma} - d_{-\sigma}\right)\rangle_\sigma \,. \tag{45}$$

Exploiting the central limit theorem in the same fashion as above, one obtains for the above contributions $\epsilon_\pm$:

$$\epsilon_\sigma \;=\; \Phi\left(\frac{Q_{\sigma\sigma} - Q_{-\sigma-\sigma} - 2\ell(R_{\sigma\sigma} - R_{-\sigma\sigma})}{2\sqrt{v_\sigma}\sqrt{Q_{++} - 2Q_{+-} + Q_{--}}}\right) \tag{46}$$

where $\Phi(z) = \int_{-\infty}^{z} dx\, e^{-x^2/2}/\sqrt{2\pi}$.

By inserting $\{R_{S\sigma}(\alpha), Q_{ST}(\alpha)\}$ we obtain the learning curve $\epsilon_g(\alpha)$, i.e. the typical generalization error after on-line training with $\alpha\, N$ random examples. Here, we once more exploit the fact that the order parameters and, thus, also ϵ_g are self-averaging non-fluctuating quantities in the thermodynamic limit $N \to \infty$.

As an example, we consider the dynamics of LVQ1, cf. Eq. (39). Fig. 5 (left panel) displays the learning curves as obtained for a particular setting of the model parameters and different choices of the learning rate η. Initially, a large learning rate is favorable, whereas smaller values of η facilitate better generalization behavior at large α. One can argue that, as in stochastic gradient descent procedures, the best asymptotic ϵ_g will be achieved for small learning rates $\eta \to 0$. In this limit, we can omit terms quadratic in η from the differential equations and integrate them directly in rescaled time $(\eta\alpha)$. The asymptotic, stationary result for $(\eta\alpha) \to \infty$ then corresponds to the best achievable performance of the algorithm in the given model settings. Figure 5 (right panel) displays, among others, an example result for LVQ1.

With the formalism outlined above it is readily possible to compare different algorithms within the same model situation. This concerns, for instance, the detailed prototype dynamics and sensitivity to initial conditions. Here, we restrict ourselves to three example algorithms and very briefly discuss essential properties and the achievable generalization ability:

- **LVQ1:** This basic prescription was already defined above as the original WTA algorithm with modulation function

$$f_s = \Theta(d_{-s} - d_{+s})\, s\, \sigma.$$

- **LVQ+/-:** Several modifications have been suggested in the literature, aiming at better generalization behavior. The term LVQ+/- is used here to describe one basic variant which updates two vectors at a time: the closest among all prototypes which belong to the same class (a) and the closest one among those that represent a different class (b). The so-called *correct winner* (a) is moved towards the data while the *wrong winner* (b) is pushed even

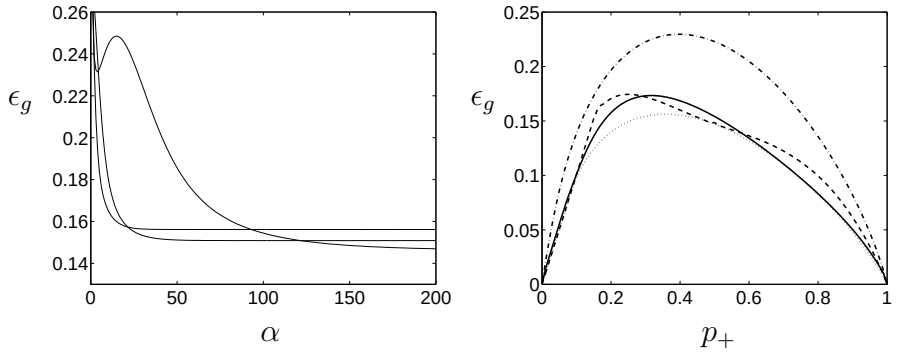

Fig. 5. LVQ learning curves and comparison of algorithms.
Left panel: $\epsilon_g(\alpha)$ for $\ell = 1.2, p_+ = 0.8$, and $v_+ = v_- = 1$. Prototypes were initialized as in Eq. (44) with $\hat{Q} = 10^{-4}$. From bottom to top at $\alpha = 200$, the graphs correspond to learning rates $\eta = 0.2, 1.0$, and 2.0, respectively.
Right panel: Achievable generalization error as a function of $p_+ = 1 - p_-$ in the model with $\ell = 1, v_+ = 0.25$, and $v_- = 0.81$. Initialization of all training schemes as specified in the left panel. The solid line marks the result for LVQ1 in the limits $\eta \to 0$ and $\eta\alpha \to \infty$, the dashed line corresponds to an idealized early stopping procedure applied to LVQ+/-, and the chain line represents the $\alpha \to \infty$ asymptotic outcome of LFM training. In addition, the dotted curve represents the best possible linear decision boundary constructed from the input density.

farther away. In our simple model scenario this amounts to the modulation function

$$f_s = s\,\sigma = \pm 1.$$

– **LFM:** The so-called *learning from mistakes* (LFM) scheme performs an update of the LVQ+/- type, but only for inputs which are misclassified before the learning step:

$$f_s = \Theta(d_\sigma - d_{-\sigma})\, s\,\sigma.$$

The prescription can be obtained as a limiting case of various algorithms suggested in the literature, see [42] for a detailed discussion. Here we emphasize the similarity with the familiar perceptron training discussed in the first sections.

Learning curves and asymptotic behavior of LVQ1 are exemplified in Fig. 5. As an interesting feature one notes a non-monotonicity of $\epsilon_g(\alpha)$ for small learning rates. It corresponds to an over-shooting effect observed in the dynamics of prototypes when approaching their stationary positions [42]. It is important to note that this very basic, heuristic prescription achieves excellent performances: Typically, the asymptotic result is quite close to the optimal ϵ_g, given by the best linear decision boundary as constructed from the input density (36).

The naive application of LVQ+/- training results in a strong instability: In all settings with $p_+ \neq p_-$, the prototype representing the *weaker* class will be pushed away in the majority of training steps. Consequently, a divergent behavior is observed and, for $\alpha \to \infty$, the classifier assigns all data to the the stronger cluster with the trivial result $\epsilon_g = \min\{p_+, p_-\}$. Several measures have been suggested to control the instability. In Kohonen's LVQ2.1 and other modifications, example data are only accepted for update when $\boldsymbol{\xi}$ falls into a *window* close to the current decision boundary. Another intuitive approach is based on the observation that $\epsilon_g(\alpha)$ generically displays a pronounced minimum before the generalization behavior deteriorates. In our formal model, it is possible to work out the location of the minimum analytically and thus determine the performance of an idealized *early stopping* method. The corresponding result is displayed in Fig. 5 (right panel, dashed line) and appears to compete well with LVQ1. However, it is important to note that the quality of the minimum in $\epsilon_g(\alpha)$ strongly depends on the initial conditions. Furthermore, in a practical situation, successful early stopping would require the application of costly validation schemes.

Finally, we briefly discuss the LFM prescription. A peculiar feature of LFM is that the stationary position of prototypes does depend strongly on the initial configuration [42]. On the contrary, the $\alpha \to \infty$ asymptotic decision boundary is well-defined. In the LFM prescription, emphasis is on the classification; the aspect of Vector Quantization (representation of clusters) is essentially disregarded. While at first sight clear and attractive, LFM yields a far from optimal performance in the limit $\alpha \to \infty$. Note that already in perceptron training as discussed in the first sections, a naive learning from mistakes strategy is bound to fail miserably in general cases of noisy data or unlearnable rules.

The above considerations concern only the simplest LVQ training scenarios and algorithms. Several directions in which to extend the formalism are obviously interesting.

We only mention that unsupervised prototype based learning has been treated in complete analogy to the above [43,44]. Technically, it reduces to the consideration of modulation functions which do not depend on the cluster or class label. The basic competitive WTA Vector Quantization training would be represented, for instance, by the modulation function $f_s = \Theta(d_{-s} - d_{+s})$ which always moves the winning prototype closer to the data. The training prescription can be interpreted as a stochastic gradient descent of a cost function, the so-called quantization error [44]. The exchange and permutation symmetry of prototypes in unsupervised training results in interesting effects which resemble the plateaus discussed in multilayered neural networks, cf. section 3.

The consideration of a larger number of Gaussians contributing to the input density is relatively simple. Thus, it is possible to model more complex data structures and study their effect on the training dynamics. The treatment of more than two prototypes is also conceptually simple but constitutes a technical challenge. Obviously, the number of order parameters increases. In addition, the r.h.s. of the ODE cannot be evaluated analytically, in general, but involve numerical integrals. Note that the dynamics of several prototypes representing

the same class of data resembles strongly the behavior of unsupervised Vector Quantization. First results along these lines have been obtained recently, see for instance [45].

The variational optimization, as discussed for the perceptron in detail, should give insights into the essential features of robust and successful LVQ schemes. Due to the more complex input density, however, the analysis proves quite involved and has not yet been completed.

A highly relevant extension of LVQ is that of relevance learning. Here, the idea is to replace the simple minded Euclidean metrics by an adaptive measure. An important example is a weighted distance of the form

$$d(\boldsymbol{w},\boldsymbol{\xi}) = \sum_{j=1}^{N} \lambda_j^2 \ (w_j - \xi_j)^2 \quad \text{with} \quad \sum_{j=1}^{N} \lambda_j^2 = 1$$

where the normalized factors λ_j are called relevances as they measure the importance of dimension j in the classification. Relevance LVQ (RLVQ) and related schemes update these factors according to a heuristic or gradient based scheme in the course of training [46,47]. More complex schemes employ a full matrix of relevances in a generalized quadratic measure or consider local measures attached to the prototypes [48].

The analysis of the corresponding training dynamics constitutes another challenge in the theory of on-line learning. The description has to go beyond the techniques discussed in this paper, as the relevances define a time-dependent linear transformation of feature space.

5 Summary and Outlook

The statistical physics approach to learning has allowed for the analytical treatment of the learning dynamics in a large variety of adaptive systems. The consideration of typical properties of large systems in specific model situations complements other approaches and contributes to the theoretical understanding of adaptive systems.

Here, we have highlighted only selected topics as an introduction to this line of research. The approach is presented, first, in terms of the perceptron network. Despite its simplicity, the framework led to highly non-trivial insights and facilitated the putting forward of the method. For instance, the variational approach to optimal training was developed in this context. Gradient based training in multilayered networks constitutes an important example for the analysis of more complex architectures. Here, non-trivial effects such as quasistationary plateau states can be observed and investigated systematically. Finally, a recent application of theoretical framework concerns prototype based training in so-called Learning Vector Quantization.

Several interesting questions and results have not been discussed at all or could be mentioned only very briefly: the study of finite system sizes, on-line learning in networks with discrete weights, unsupervised learning and clustering,

training from correlated data or from a fixed pool of examples, query strategies, to name only a few. We can only refer to the list of references, in particular [27] and [28] may serve as a starting point for the interested reader.

Due to the conceptual simplicity of the approach and its applicability in a wide range of contexts it will certainly continue to facilitate better theoretical understanding of learning systems in general. Current challenges include the treatment of non-trivial input distributions, the dynamics of learning in more complex networks architectures, the optimization of algorithms and their practical implementation in such systems, or the investigation of component-wise updates as, for instance, in relevance LVQ.

References

1. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of State calculations by fast computing machines. J. Chem. Phys. 21, 1087 (1953)
2. Huang, K.: Statistical Mechanics. Wiley and Sons, New York (1987)
3. Jaynes, E.T.: Probability Theory: The Logic of Science. Bretthorst, G.L. (ed.), Cambridge University Press, Cambridge, UK (2003)
4. Mace, C.W.H., Coolen, T.: Dynamics of Supervised Learning with Restricted Training Sets. Statistics and Computing 8, 55-88 (1998)
5. Biehl, M., Schwarze, M.: On-line learning of a time-dependent rule Europhys. Lett. 20, 733-738 (1992)
6. Biehl, M., Schwarze, H.: Learning drifting concepts with neural networks. Journal of Physics A: Math. Gen. 26, 2651-2665 (1993)
7. Kinouchi, O., Caticha, N.: Lower bounds on generalization errors for drifting rules. J. Phys. A: Math. Gen.26, 6161-6171 (1993)
8. Vicente, R, Kinouchi, O., Caticha, N.: Statistical Mechanics of Online Learning of Drifting Concepts: A Variational Approach. Machine Learning 32, 179-201 (1998)
9. Reents, G., Urbanczik, R.: Self-averaging and on-line learning. Phys. Rev. Lett. 80, 5445-5448 (1998)
10. Kinzel, W., Rujan, P.: Improving a network generalization ability by selecting examples. Europhys. Lett. 13, 2878 (1990)
11. Kinouchi, O., Caticha, N.: Optimal generalization in perceptrons. J. Phys. A: Math. Gen. 25, 6243-6250 (1992)
12. Copelli, M., Caticha, N.: On-line learning in the committee machine. J. Phys. A: Math. Gen. 28, 1615-1625 (1995)
13. Biehl, M., Riegler, P.: On-line Learning with a Perceptron. Europhys. Lett. 78: 525-530 (1994)
14. Biehl, M., Riegler, P., Stechert, M.: Learning from Noisy Data: An Exactly Solvable Model. Phys. Rev. E 76, R4624-R4627 (1995)
15. Copelli, M., Eichhorn, R., Kinouchi, O., Biehl, M., Simonetti, R., Riegler, P., Caticha, N.: Noise robustness in multilayer neural networks. Europhys. Lett. 37, 427-432 (1995)
16. Vicente, R., Caticha, N.: Functional optimization of online algorithms in multilayer neural networks. J. Phys. A: Math. Gen. 30, L599-L605 (1997)
17. Opper, M.: A Bayesian approach to on-line learning. In: [27], pp. 363-378 (1998)
18. Opper, M., Winther, O.: A mean field approach to Bayes learning in feed-forward neural networks. Phys. Rev. Lett. 76, 1964-1967 (1996)

19. Solla, S.A., Winther, O.: Optimal perceptron learning: an online Bayesian approach. In: [27], pp. 379-398 (1998)
20. Cybenko, G.V.: Approximation by superposition of a sigmoidal function. Math. of Control, Signals and Systems 2, 303-314 (1989)
21. Endres, D., Riegler, P.: Adaptive systems on different time scales. J. Phys. A: Math. Gen. 32: 8655-9663 (1999)
22. Biehl, M., Schwarze, H.: Learning by on-line gradient descent. J. Phys A: Math. Gen. 28, 643 (1995)
23. Saad, D., Solla, S.A.: Exact solution for on-line learning in multilayer neural networks. Phys. Rev. Lett. 74, 4337-4340 (1995)
24. Saad, D., Solla, S.A.: Online learning in soft committee machines. Phys. Rev. E 52, 4225-4243 (1995)
25. Biehl, M., Riegler, P., Wöhler, C.: Transient Dynamics of Online-learning in two-layered neural networks. J. Phys. A: Math. Gen. 29: 4769 (1996)
26. Saad, D, Rattray, M.: Globally optimal parameters for on-line learning in multilayer neural networks. Phys. Rev. Lett. 79, 2578 (1997)
27. Saad, D. (ed.): On-line learning in neural networks. Cambridge University Press, Cambridge, UK (1998)
28. Engel, A., Van den Broeck, C.: The Statistical Mechanics of Learning. Cambridge University Press, Cambridge, UK (2001)
29. Schlösser, E., Saad, D., Biehl, M.: Optimisation of on-line Principal Component Analysis. J. Physics A: Math. Gen. 32, 4061 (1999)
30. Biehl, M., Schlösser, E.: The dynamics of on-line Principal Component Analysis. J. Physics A: Math. Gen. 31: L97 (1998)
31. Biehl, M., Mietzner, A.: Statistical mechanics of unsupervised learning. Europhys. Lett. 27, 421-426 (1993)
32. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1997)
33. Kohonen, T.: Learning vector quantization. In: Arbib, M.A. (ed.) *The Handbook of Brain Theory and Neural Networks.*, pp. 537-540. MIT Press, Cambridge, MA (1995)
34. Van den Broeck, C., Reimann, P.: Unsupervised Learning by Examples: On-line Versus Off-line. Phys. Rev. Lett. 76, 2188-2191, (1996)
35. Reimann, P, Van den Broeck, C, Bex, G.J.: A Gaussian Scenario for Unsupervised Learning. J. Phys. A: Math. Gen. 29, 3521-3533 (1996)
36. Riegler, P., Biehl, M., Solla, S.A., Marangi, C.: On-line learning from clustered input examples. In: Marinaro, M., Tagliaferri, R. (eds.) *Neural Nets WIRN Vietri-95, Proc. of the 7th Italian Workshop on Neural Nets*, pp. 87-92. World Scientific, Singapore (1996)
37. Marangi, C., Biehl, M., Solla, S.A.: Supervised learning from clustered input examples. Europhys. Lett. 30, 117-122 (1995)
38. Biehl, M.: An exactly solvable model of unsupervised learning. Europhysics Lett. 25, 391-396 (1994)
39. Meir, R.: Empirical risk minimization versus maximum-likelihood estimation: a case study. Neural Computation 7, 144-157 (1995)
40. Barkai, N, Seung, H.S., Sompolinksy, H.: Scaling laws in learning of classification tasks. Phys. Rev. Lett. 70, 3167-3170 (1993)
41. Neural Networks Research Centre. Bibliography on the self-organizing maps (SOM) and learning vector quantization (LVQ). Helsinki University of Technology, available on-line: http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html (2002)

42. Biehl, M, Ghosh, A., Hammer, B.: Dynamics and generalization ability of LVQ algorithms. J. Machine Learning Research 8, 323-360 (2007)
43. Biehl, M., Freking, A., Reents, G.: Dynamics of on-line competitive learning. Europhysics Letters 38, 73-78 (1997)
44. Biehl, M., Ghosh, A., Hammer, B.: Learning Vector Quantization: The Dynamics of Winner-Takes-All algorithms. Neurocomputing 69, 660-670 (2006)
45. Witeolar, A., Biehl, M., Ghosh, A., Hammer, B.: Learning Dynamics of Neural Gas and Vector Quantization. Neurocomputing 71, 1210-1219 (2008)
46. Bojer, T., Hammer, B., Schunk, D., Tluk von Toschanowitz, K.: Relevance determination in learning vector quantization. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks ESANN 2001, pp. 271-276. D-facto publications, Belgium (2001)
47. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15, 1059-1068 (2002)
48. Schneider, P, Biehl, M., Hammer, B.: Relevance Matrices in Learning Vector Quantization In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks ESANN 2007, pp. 37-43, d-side publishing, Belgium (2007)