entropy

Article

Statistical Mechanics of On-Line Learning Under Concept Drift

Michiel Straat [1], Fthi Abadi [1], Christina Göpfert [2], Barbara Hammer [2] and Michael Biehl [1,*]

[1] Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands; m.j.c.straat@rug.nl (M.S.); fthialem@gmail.com (F.A.)
[2] Center of Excellence—Cognitive Interaction Technology (CITEC), Bielefeld University, Inspiration 1, 33619 Bielefeld, Germany; cgoepfert@techfak.uni-bielefeld.de (C.G.); bhammer@techfak.uni-bielefeld.de (B.H.)
* Correspondence: m.biehl@rug.nl; Tel.: +31-50-363-3997

Received: 4 September 2018; Accepted: 8 October 2018; Published: 10 October 2018

Abstract: We introduce a modeling framework for the investigation of on-line machine learning processes in non-stationary environments. We exemplify the approach in terms of two specific model situations: In the first, we consider the learning of a classification scheme from clustered data by means of prototype-based Learning Vector Quantization (LVQ). In the second, we study the training of layered neural networks with sigmoidal activations for the purpose of regression. In both cases, the target, i.e., the classification or regression scheme, is considered to change continuously while the system is trained from a stream of labeled data. We extend and apply methods borrowed from statistical physics which have been used frequently for the exact description of training dynamics in stationary environments. Extensions of the approach allow for the computation of typical learning curves in the presence of concept drift in a variety of model situations. First results are presented and discussed for stochastic drift processes in classification and regression problems. They indicate that LVQ is capable of tracking a classification scheme under drift to a non-trivial extent. Furthermore, we show that concept drift can cause the persistence of sub-optimal plateau states in gradient based training of layered neural networks for regression.

Keywords: concept drift; on-line learning; continual learning; neural networks; learning vector quantization; statistical physics of learning

1. Introduction

The many challenges of modern data science call for the design of efficient methods for automated analysis. Machine learning techniques play a key role in this context [1–3].

The development of modeling frameworks in which to obtain general insights into practically relevant phenomena is instrumental to achieve the necessary theoretical understanding. Analytical and computational approaches that come from or are related to statistical physics [4–9] have played an important role in this field and continue to do so.

1.1. Concept Drift and Continual Learning

In this contribution, we address a topic which is currently attracting increasing interest in the scientific community: the efficient training of machine learning systems in a non-stationary environment, where the target task or the statistical properties of the example data vary with time (see, for instance, [10–15] and references therein). Terms such as continual learning and lifelong learning have been coined in this context.

Frequently, the set-up of machine learning processes comprises two different stages (see, for instance, [1–3]): In the training phase, a given set of example data is analyzed, information is extracted and a corresponding hypothesis is parameterized in terms of, e.g., a classifier or regression system. In the subsequent working phase, this hypothesis is applied to novel data. Implicitly, one assumes that the training set is representative of the problem and that statistical properties of the data and the actual target task do not change after training.

For many practical applications of machine learning, the assumption of stationarity may be well justified. However, the conceptual and temporal separation of training and working phase is not very plausible in human and other biological learning processes [16,17]. As an example, in a predator and prey system, strategies can change continuously with species trying to adapt to their adversaries' behavior. In addition, in many technical applications of machine learning, the separation becomes inappropriate if the actual task of learning, e.g., the target classification, changes in time [10]. Moreover, very frequently, the training samples become available in the form of a stream of data (e.g., [11–14]). In such situations, the learning system must be able to detect and track concept drift, i.e., forget irrelevant, older information while continuously adapting to more recent inputs. Examples for this situation can be found, for instance, in robotics. Other problems, such as the filtering of spam messages in e-mail communication, resemble the predator–prey example in that the learning systems try to adapt to changing strategies of their opponents. Further applications range from fraud detection, quality control and customer segments management to drop out prediction for e-learning and gaming [10]. Overviews of earlier work and recent developments in the context of machine learning in non-stationary environments are provided, for instance, in [10–15]. While drift can occur in any learning scenario, in this contribution, we focus on supervised learning.

In the literature, two major types of non-stationary environments have been discussed [10–15]: In so-called virtual drifts, the statistical properties of the available example data change with time, while the actual target task, e.g., the classification or regression scheme, remains unaltered. The term real drift has been coined for situations in which the target itself is time-dependent. Frequently, real drift processes are accompanied by additional virtual drifts.

1.2. Models of On-Line Learning Under Concept Drift

There exists a large variety of technologies which address learning in the context of drift (see [10–14] for overviews). On a global level, one often differentiates so-called active methods, which aim for an explicit detection of drift and according action of the learning system, and passive methods, which can implicitly react to drift by their design. Popular active methods combine statistical tests for novelty detection [18] with a rearrangement or retraining of the system to account for the observed drift. The latter is particularly efficient if, for instance, ensemble methods are used [19,20]. The need for explicit drift detection often has the consequence that only specific types of drift can be dealt with (one exception being found in [20]). In particular, small gradual drifts are notoriously difficult to detect [21]. Passive methods continuously adapt the model according to the given data. Thus, they automatically react to all types of drift which is present in the training data. However, they face the classical stability-plasticity dilemma: relevant novel information has to be dealt with while preserving already learned signals. Local or hybrid schemes have been particularly successful in the past years (see, e.g., [21,22]). Other popular passive technologies rely on online learning schemes, in particular online gradient descent, which has been incorporated into drift learning strategies for the simple perceptron, neural networks, or extreme learning machines, as an example [23,24]. The behavior of such models varies extensively across different learning scenarios [11].

In this contribution, we study two basic scenarios of on-line learning in non-stationary environments, addressing binary classification and continuous regression problems. We present a mathematical model of drifting concepts in on-line training from high-dimensional data. Methods borrowed from statistical physics facilitate the study of the typical learning dynamics for different training scenarios and strategies. While the approach is suitable for virtual and real drift processes, here, we focus on the study of explicitly time-dependent target concepts.

With respect to classification, we consider Learning Vector Quantization (LVQ) as an example framework, i.e., prototype-based systems as originally suggested by Kohonen [25–29]. LVQ training is most frequently done in an on-line setting by presenting a sequence of single examples which are used to improve the system iteratively [28,29]. Therefore, LVQ should constitute a promising framework for incremental learning in the presence of concept drift.

Layered neural networks with sigmoidal, continuous activation functions serve as an example system in the context of regression. Specifically, we consider the so-called Soft Committee Machine (SCM), a shallow architecture which can be trained by means of on-line (stochastic) gradient descent [30–36]. Gradient based techniques are widely used also for multi-layered deep architectures and their suitability for the learning of non-stationary targets is a question of significant relevance [3,37].

1.3. Relation to Earlier Work

Note that several studies exist which compare different learning algorithms for streaming data experimentally (see, e.g., [11,12] and references therein). Unlike these empirical investigations, our contribution aims for a formal, mathematical framework which can abstract from the variations which occur in the course of a concrete, real world training cycle.

Methods borrowed from statistical physics have been used to analyze the typical behavior of various learning systems in model scenarios [4–7]. The particularly successful analysis of on-line learning is based on the assumption that a sequence of independently generated random N-dimensional examples is presented to the learning system [8,9,38]. Further simplifying assumptions and the consideration of the so-called thermodynamic limit $N \rightarrow \infty$ facilitate the exact mathematical description of typical learning curves in terms of ordinary differential equations (ODE). For detailed discussions of the limitations of the approach as well as extensions that allow to overcome them (see several contributions in [38] and, for instance, [39]).

Various reviews, article collections and monographs present and discuss the approach with respect to supervised learning in simple perceptrons and multilayered neural networks (see e.g., [4–9,38] and references therein). Similarly, the dynamics of unsupervised learning has been studied, including prototype-based competitive learning, Principal Component Analysis and related schemes [40–42].

Stationary model densities of clustered data, similar to the ones considered here for LVQ, have been studied with respect to several unsupervised and supervised training schemes (see [40–45] for examples and further references). Supervised LVQ training was considered more recently in the framework of simplifying model situations in [39,46–49].

The SCM in stationary environments has been studied extensively from the statistical physics perspective. Practically relevant phenomena, such as the occurrence of quasi-stationary plateau states have been investigated in great detail (see [30–36,38] for examples and further references).

The presence of concept drift has also been addressed within the statistical physics of on-line learning. State-of-the-art investigations have considered, in particular, the learning of time-dependent, linearly separable rules as a model system in [50–53]. Note that the assumption of statistically independent examples in the stream of data does not hinder the study of meaningful drift scenarios. It is, for instance, well possible to consider settings in which the characteristics of the generating density or the target itself depends, implicitly, on the previous training. As an example, adversarial drifts have been considered in [50–53] for the simple perceptron.

To the best of our knowledge, we present here the first statistical mechanics analysis of on-line learning under concept drift in prototype-based classification and layered neural networks for regression.

1.4. Outline

The main aim of this work is to present and establish a theoretical framework in which to investigate models of learning scenarios. The considered example systems, i.e., LVQ for classification and layered networks for regression, serve as examples to illustrate and demonstrate the usefulness of the methodology in obtaining principled insights into the properties of learning systems under concept drift. Typical behavior can be described in terms of learning curves, which reflect practically relevant phenomena such as the tracking of randomly varying targets or delayed learning in gradient descent due to quasi-stationary plateau states of the training process.

In the following sections, we first introduce the specific example systems, i.e., LVQ and SCM considered for classification and regression, respectively. In Section 2.3, we revisit the mathematical description of the learning dynamics in stationary environments for both systems. Next, the model is extended to include real concept drifts. We also briefly discuss the potential introduction of virtual drifts and the consideration of weight decay as an explicit mechanism of forgetting.

First results of our analysis are presented in Section 3, which exemplify and demonstrate the usefulness of the methodological approach: We obtain insights into the ability of prototype-based systems to track a time-varying classification scheme. Furthermore, we investigate the effect of concept drift on regression systems trained by gradient-based methods. In Section 4, we conclude with a general discussion and outlook on future work.

2. Models and Mathematical Analysis

We first introduce Learning Vector Quantization for classification with emphasis on the heuristic LVQ1 scheme. We further introduce a suitable, clustered density of input data, which is taken to define the target task in the model. Next, we present the Soft Committee Machine as an example regression system which can be studied in a so-called student–teacher scenario [5–7]. Here, training is based on stochastic gradient descent with respect to a suitable cost function.

In Section 2.3, we revisit the analytical treatment of on-line learning in stationary environments. We extend the mathematical framework with respect to the presence of concept drift in regression and classification in Section 2.4. In addition, we consider the incorporation of weight decay. Formally, the modifications compared to the stationary cases are identical in both scenarios.

2.1. Learning Vector Quantization

Learning Vector Quantization constitutes a family of prototype-based algorithms which are used in a wide variety of practical classification problems [26–29]. The popularity of the approach is due to several appealing properties: LVQ procedures are easy to implement and very intuitive. The classification of LVQ is based on a distance measure, frequently Euclidean, which is used to quantify the (dis-) similarity of feature vectors and class-specific prototypes. In contrast to the black-box character of many less transparent methods, LVQ allows for straightforward interpretations since the prototype vectors are embedded in the actual feature space and directly parameterize the classifier [28,29].

2.1.1. Nearest Prototype Classification and Winner-Takes-All Training

In general, several prototypes can be employed to represent each class. In this contribution, however, we restrict the analysis to simple situations with only two prototypes $\vec{w}_k \in \mathbb{R}^N$ in total, where prototype k is supposed to represent the data from Class $k \in \{1, 2\}$.

A Nearest Prototype Classification (NPC) scheme is parameterized by the prototypes with respect to the distance measure $d(\vec{w}, \vec{\xi})$: A given $\vec{\xi} \in \mathbb{R}^N$ is assigned to the class of the closest prototype. In the presence of only two prototypes, the assignment is to Class 1 if $d(\vec{w}_1, \vec{\xi}) < d(\vec{w}_2, \vec{\xi})$ and to Class 2, otherwise. In practice, ties can be broken arbitrarily.

A variety of distance measures can be used in LVQ, further enhancing the flexibility of the approach. Several popular choices, including adaptive distance measures in relevance learning, are discussed in [28,29,54]. In the following, we restrict ourselves to the most popular (squared) Euclidean measure

$$d(\vec{w},\vec{\xi}) = (\vec{w}-\vec{\xi})^2. \tag{1}$$

We assume that, in the training process, a sequence of single example data $\{\vec{\xi}^{\,\mu},\sigma^\mu\}$ is presented to the system [8,9]. At time step $\mu = 1,2,\ldots$, the vector $\vec{\xi}^{\,\mu}$ is presented, together with its class label $\sigma^\mu = 1,2$. Generic incremental or on-line LVQ updates are of the form [39,46–48]:

$$\vec{w}_k^\mu = \vec{w}_k^{\mu-1} + \Delta\vec{w}_k^\mu \text{ with } \Delta\vec{w}_k^\mu = \frac{\eta}{N} f_k\left[d_1^\mu, d_2^\mu, \sigma^\mu, \ldots\right]\left(\vec{\xi}^\mu - \vec{w}_k^{\mu-1}\right) \text{ where } d_i^\mu = d(\vec{w}_i^{\mu-1},\vec{\xi}^\mu) \tag{2}$$

and the learning rate η is scaled with the input dimension N. The precise algorithm is specified by choice of the modulation function $f_k[\ldots]$, which depends typically on the Euclidean distances of the data point from the current prototype positions and on the labels $k,\sigma^\mu = 1,2$ of the prototype and training example.

Arguably the most basic LVQ training scheme was suggested by Kohonen and is known as LVQ1 [25–27]. In analogy to the NPC concept, it updates only the currently closest prototype according to a so-called Winner-Takes-All (WTA) scheme. Formally, the LVQ1 prescription for only two competing prototypes corresponds to Equation (2) with

$$f_k[d_1^\mu, d_2^\mu, \sigma^\mu] = \Theta\left(d_{\widehat{k}}^\mu - d_k^\mu\right)\Psi(k,\sigma^\mu), \quad \text{where } \widehat{k} = \begin{cases} 2 & \text{if } k=1 \\ 1 & \text{if } k=2, \end{cases} \tag{3}$$

$$\Theta(x) = \begin{cases} 1 & \text{if } x>0 \\ 0 & \text{else,} \end{cases} \quad \text{and } \Psi(k,\sigma) = \begin{cases} +1 & \text{if } k=\sigma \\ -1 & \text{else.} \end{cases}$$

Here, the Heaviside function $\Theta(\ldots)$ singles out the winning prototype and the factor $\Psi(k,\sigma^\mu)$ determines the sign of the update: The WTA update according to Equation (3) moves the prototype towards the presented feature vector if it carries the same class label $k = \sigma^\mu$. On the contrary, if the prototype is meant to present a different class, its distance from the data point is increased even further. Note that LVQ1 cannot be interpreted as a gradient descent procedure of a suitable cost function in a straightforward way due to discontinuities at the class boundaries.

Many modifications of LVQ have been suggested and discussed in the literature, including heuristically motivated extensions of LVQ1, cost function based schemes and variants employing unconventional or adaptive distance measures [25–29,54]. Mostly, they retain the basic idea of attraction and repulsion of the winning prototypes similar to Equation (3).

2.1.2. Clustered Model Data

LVQ algorithms are most suitable for classification schemes which reflect a given cluster structure in the data. In the modeling, we therefore consider a stream of random input vectors $\vec{\xi} \in \mathbb{R}^N$ which are generated independently according to a bi-modal distribution of the form [39,46–48]

$$P(\vec{\xi}) = \sum_{m=1,2} p_m\, P(\vec{\xi} \mid m) \text{ with } P(\vec{\xi} \mid m) = \frac{1}{(2\,\pi\, v_m)^{N/2}} \exp\left[-\frac{1}{2\,v_m}\left(\vec{\xi} - \lambda\vec{B}_m\right)^2\right]. \tag{4}$$

The target classification is taken to coincide with the cluster membership here, i.e., $\sigma = m$ in Equation (3). Class-conditional densities $P(\vec{\xi} \mid m{=}1,2)$ correspond to isotropic, spherical Gaussians with variances v_m and means $\lambda\,\vec{B}_m$. Prior weights of the clusters are denoted as p_m and satisfy $p_1 + p_2 = 1$. We assume that the vectors $\vec{B}_m$ are orthonormal with $\vec{B}_1^2 = \vec{B}_2^2 = 1$ and $\vec{B}_1 \cdot \vec{B}_2 = 0$. Obviously, the classes $m = 1,2$ are not linearly separable due to the overlap of the clusters.

As an illustration, Figure 1 displays data in $N = 200$ dimensions, generated according to a density of the form in Equation (4). While the clusters are clearly visible in the subspace given by $\vec{B}_1$ and $\vec{B}_2$, projections into a randomly chosen plane completely overlap.

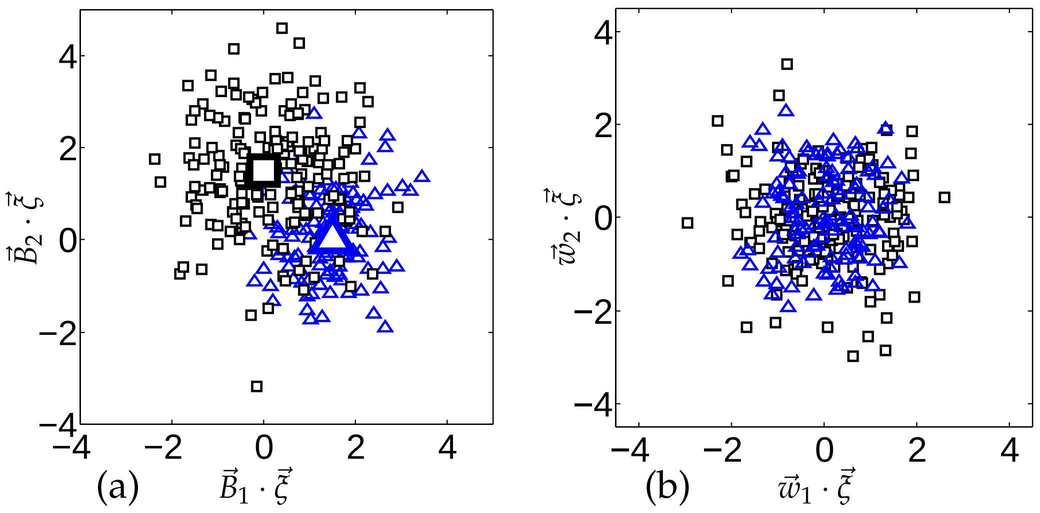

Figure 1. *Clustered Model Density*. Illustration of the clustered density, Equation (4), in $N = 200$ dimensions, here with $p_1 = 0.4, p_2 = 0.6$ and $v_1 = 0.64, v_2 = 1.44$. Triangles (squares) represent 120 (180) vectors $\vec{\xi}$ from the clusters centered at $\lambda\vec{B}_1$ ($\lambda\vec{B}_2$) with $\lambda = 1.5$, respectively. (**a**) Projections $\vec{B}_{1,2} \cdot \vec{\xi}$ of the data. The cluster centers are marked by larger symbols. (**b**) Projections $\vec{w}_{1,2} \cdot \vec{\xi}$ on two randomly chosen orthonormal vectors $\vec{w}_{1,2}$.

We denote conditional averages over $P(\vec{\xi} \mid m)$ as $\langle \cdots \rangle_m$, whereas mean values $\langle \cdots \rangle = \sum_{m=1,2} p_m \langle \cdots \rangle_m$ are defined with respect to the full density (Equation (4)). One obtains, for instance, the conditional and full averages

$$\left\langle \vec{\xi} \right\rangle_m = \lambda \vec{B}_m, \quad \langle \vec{\xi}^{\,2} \rangle_m = v_m N + \lambda^2 \text{ and } \langle \vec{\xi}^{\,2} \rangle = (p_1 v_1 + p_2 v_2) N + \lambda^2. \tag{5}$$

Note that, in the thermodynamic limit $N \to \infty$, which is considered below, λ^2 can be neglected in comparison to the terms of $\mathcal{O}(N)$ in Equation (5).

2.2. Soft Committee Machines

The term Soft Committee Machine (SCM) has been coined for feedforward neural networks with sigmoidal activations in a single hidden layer and a linear output unit (see, for instance, [30–36,55,56]). Its structure resembles that of a (crisp) committee machine with binary threshold hidden units, where the network's response is given by their majority vote (see [5–7] and references therein).

2.2.1. Network Definition

The output of an SCM with K hidden units and fixed hidden-to-output weights is of the form

$$y(\vec{\xi}) = \sum_{k=1}^{K} g(\vec{w}_k \cdot \vec{\xi}) \tag{6}$$

where $\vec{w}_k \in \mathbb{R}^N$ denotes the weight vector connecting the N-dimensional input layer with the kth hidden unit. A non-linear transfer function $g(\cdots)$ defines the hidden unit states and the final output is given as their sum. As a specific example, we consider the sigmoidal

$$g(x) = \text{erf}\left(x/\sqrt{2}\right) \quad \text{with the derivative } g'(x) = \sqrt{\frac{2}{\pi}}\, e^{-x^2/2} \tag{7}$$

The activation resembles closely other sigmoidal functions, e.g., the popular $\tanh(x)$, but offers great mathematical ease in the analytical treatment, as originally exploited in [30].

Note that the SCM, cf. Equation (6), is not quite representing a universal approximator, a property which could be achieved by introducing adaptive local thresholds $\vartheta_i \in \mathbb{R}$ in hidden unit activations of the form $g\left(\vec{w}_i \cdot \vec{\xi} - \vartheta_i\right)$ (see [57] for a general proof). Adaptive hidden-to-output weights also increase the flexibility of the SCM and have been studied in, for instance [33], from a statistical physics perspective. Here, however, the emphasis is on basic dynamical effects in the on-line training of an SCM and we restrict ourselves to the simpler model defined above.

2.2.2. Regression Scheme and On-Line Gradient Descent

In the context of continuous regression, the training of neural networks with output $y(\vec{\xi})$ based on examples $\left\{\vec{\xi}^\mu \in \mathbb{R}^N, \tau^\mu \in \mathbb{R}\right\}$ is frequently guided by the quadratic deviation of the network output from the target values [1–3]. It serves as a cost function which evaluates the network performance with respect to a single example as

$$e^\mu\left(\{\vec{w}_k\}_{k=1}^K\right) = \frac{1}{2}\left(y^\mu - \tau^\mu\right)^2 \quad \text{with the shorthand} \quad y^\mu = y(\vec{\xi}^\mu). \tag{8}$$

In stochastic or on-line gradient descent, updates of the weight vectors are based on the sequential presentation of single examples:

$$\vec{w}_k^\mu = \vec{w}_k^{\mu-1} + \Delta\vec{w}_k^\mu \quad \text{with} \quad \Delta\vec{w}_k^\mu = -\frac{\eta}{N}\frac{\partial e^\mu}{\partial \vec{w}_k} e^\mu = -\frac{\eta}{N}\left(y^\mu - \tau^\mu\right)\frac{\partial}{\partial \vec{w}_k} y^\mu \tag{9}$$

where the gradient is evaluated in $\vec{w}_k^{\mu-1}$. For the SCM architecture specified above, we have

$$\frac{\partial y^\mu}{\partial \vec{w}_k} = g'\left(h_k^\mu\right)\vec{\xi}^\mu \quad \text{and} \quad \Delta\vec{w}_k^\mu = -\frac{\eta}{N}\left(\sum_{i=1}^K \operatorname{erf}\left[\frac{1}{\sqrt{2}}h_i^\mu\right] - \tau^\mu\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(h_k^\mu\right)^2\right]\vec{\xi}^\mu \tag{10}$$

with the inner products $h_i^\mu = \vec{w}_i^{\mu-1} \cdot \vec{\xi}^\mu$ of the current weight vectors with the new example input. Note that the change of weight vectors is proportional to $\vec{\xi}^\mu$ and can be seen as a form of Hebbian Learning [1–3].

2.2.3. Student–Teacher Scenario and Model Data

To define and model meaningful learning situations, we resort to the consideration of student–teacher scenarios [5–8]. We assume that the target regression can be defined in terms of an SCM with a given number M of hidden units and a specific set of weights $\left\{\vec{B}_m \in \mathbb{R}^N\right\}_{m=1}^M$:

$$\tau(\vec{\xi}) = \sum_{m=1}^M g(\vec{B}_m \cdot \vec{\xi}). \tag{11}$$

In the model, this so-called teacher network can be equipped with $M > K$ hidden units to model regression schemes which cannot be learnt by an SCM student of the form in Equation (6). On the contrary, $K > M$ would correspond to an over-learnable target. For the discussion of these highly interesting cases in stationary environments, see, for instance, [30–34]. In a student-teacher scenario with K and M hidden units, respectively, the update of the student weight vectors by on-line gradient descent reads:

$$\vec{w}_k^\mu = \vec{w}_k^{\mu-1} - \frac{\eta}{N}\rho_k^\mu\vec{\xi}^\mu \quad \text{where} \quad \rho_k^\mu = \left(\sum_{i=1}^K \operatorname{erf}\left[\frac{h_i^\mu}{\sqrt{2}}\right] - \sum_{m=1}^M \operatorname{erf}\left[\frac{b_m^\mu}{\sqrt{2}}\right]\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(h_k^\mu\right)^2\right] \tag{12}$$

with the quantities $b_m^\mu = \vec{B}_m \cdot \vec{\xi}^\mu$ and $h_k^\mu = \vec{w}_k^{\mu-1} \cdot \vec{\xi}^\mu$.

In the following, we restrict our analysis to perfectly matching student complexity with $K = M = 2$ only, which further simplifies Equation (12). Extensions to more hidden units and settings with $K \neq M$ will be considered in forthcoming projects.

In contrast to the model for LVQ-based classification, the vectors $\vec{B}_m$ define the target output $\tau^\mu = \tau(\vec{\xi}^\mu)$ explicitly via the teacher network for any input vector. While clustered input densities of the form in Equation (4) can also be studied for feedforward networks as in [44,45], we assume here that the actual input vectors are uncorrelated with the teacher vectors $\vec{B}_m$. Consequently, we can resort to a simpler model density and consider vectors $\vec{\xi}$ of independent, zero mean, unit variance components with, e.g.,

$$P(\vec{\xi}) = \frac{1}{(2\,\pi)^{N/2}} \exp\left[-\frac{1}{2}(\vec{\xi}\,)^2\right]. \tag{13}$$

Note that Equation (13) could be recovered formally from the density (Equation (4)) as a special case with parameters $\lambda = 0$ and $v_1 = v_2 = 1$, for which the two clusters coincide around the origin and $p_{1,2}$ become irrelevant.

2.3. The Dynamics of On-Line Training in Stationary Environments

In the following, we sketch the successful theory of on-line learning [5–8,38] as, for instance, applied to the dynamics of LVQ algorithms in [39,46–48] and to on-line gradient descent in SCM in [30–36]. We refer the reader to the original publications for details. The extensions to non-stationary situations with concept drifts are discussed in Section 2.4.

The analysis follows the same key steps in both settings. We consider adaptive vectors $\vec{w}_{1,2} \in \mathbb{R}^N$ (prototypes in LVQ or student weights in the SCM) while the characteristic vectors $\vec{B}_{1,2}$ specify the target task (cluster centers in LVQ training, SCM teacher vectors for regression).

The consideration of the thermodynamic limit $N \to \infty$ is instrumental for the theoretical treatment. The limit facilitates the following key steps, which, eventually, yield an exact mathematical description of the training dynamics in terms of ordinary differential equations (ODE):

(a) *Order parameters*

The many degrees of freedom, i.e., the components of the adaptive vectors, can be characterized in terms of only very few quantities. The definition of meaningful so-called *order parameters* follows naturally from the specific mathematical structure of the model. After presentation of a number μ of examples, as indicated by corresponding superscripts, we describe the system by the projections

$$R_{im}^\mu = \vec{w}_i^\mu \cdot \vec{B}_m \quad \text{and} \quad Q_{ik}^\mu = \vec{w}_i^\mu \cdot \vec{w}_k^\mu \quad \text{with} \quad i,k,m \in \{1,2\}. \tag{14}$$

Obviously, Q_{11}^μ, Q_{22}^μ and $Q_{12}^\mu = Q_{21}^\mu$ relate to the norms and mutual overlap of the adaptive vectors, while the four quantities R_{im} specify their projections into the linear subspace defined by the characteristic vectors $\{\vec{B}_1, \vec{B}_2\}$, respectively.

(b) *Recursions*

For the order parameters, recursion relations can be derived directly from the learning algorithms in Equations (2) and (9), which are both of the generic form $\vec{w}_k^\mu = \vec{w}_k^{\mu-1} + \Delta\vec{w}_k^\mu$, by considering the corresponding inner products:

$$\begin{aligned}
\frac{R_{im}^\mu - R_{im}^{\mu-1}}{1/N} &= \eta\,\Delta\vec{w}_i^\mu \cdot \vec{B}_m \\
\frac{Q_{ik}^\mu - Q_{ik}^{\mu-1}}{1/N} &= \eta\left(\vec{w}_i^{\mu-1} \cdot \Delta\vec{w}_k^\mu + \vec{w}_k^{\mu-1} \cdot \Delta\vec{w}_i^\mu\right) + \eta^2\,\Delta\vec{w}_i^\mu \cdot \Delta\vec{w}_k^\mu.
\end{aligned} \tag{15}$$

Note that terms of order $\mathcal{O}(1/N)$ on the right hand side (r.h.s.) of Equation (15) will be neglected in the following.

(c) *Averages over the Model Data*

Applying the central limit theorem (CLT), we can perform an average over the random sequence of independent examples. Note that $\Delta\vec{w}_k^\mu \propto \vec{\xi}^\mu$ or $\Delta\vec{w}_k^\mu \propto \left(\vec{\xi}^\mu - \vec{w}_k^{\mu-1}\right)$, respectively.

Consequently, the current input $\vec{\xi}^\mu$ enters the r.h.s. of Equation (15) only through its norm $\mid \vec{\xi} \mid^2 = \mathcal{O}(N)$ and the quantities

$$h_i^\mu = \vec{w}_i^{\mu-1} \cdot \vec{\xi}^\mu \text{ and } b_m^\mu = \vec{B}_m \cdot \vec{\xi}^\mu. \tag{16}$$

Since these inner products correspond to sums of many independent random quantities in our model, the CLT implies that the projections in Equation (16) are correlated Gaussian quantities for large N and their joint density $P(h_1^\mu, h_2^\mu, b_1^\mu, b_2^\mu)$ is given completely by first and second moments.

LVQ: For the clustered density, cf. Equation (4), the conditional moments read

$$\left\langle h_i^\mu \right\rangle_m = \lambda R_{im}^{\mu-1}, \quad \left\langle b_m^\mu \right\rangle_n = \lambda \delta_{mn}, \quad \left\langle h_i^\mu h_k^\mu \right\rangle_m - \left\langle h_i^\mu \right\rangle_m \left\langle h_k^\mu \right\rangle_m = v_m\, Q_{ik}^{\mu-1},$$
$$\left\langle h_i^\mu b_n^\mu \right\rangle_m - \left\langle h_i^\mu \right\rangle_m \left\langle b_n^\mu \right\rangle_m = v_m\, R_{in}^{\mu-1}, \quad \left\langle b_l^\mu b_n^\mu \right\rangle_m - \left\langle b_l^\mu \right\rangle_m \left\langle b_n^\mu \right\rangle_m = v_m\, \delta_{ln}, \tag{17}$$

where $i,k,l,m,n \in \{1,2\}$ and $\delta_{\dots}$ is the Kronecker-Delta.

SCM: In the simpler case of the isotropic, spherical density (Equation (13)) with $\lambda = 0$ and $v_1 = v_2 = 1$ the moments reduce to

$$\left\langle h_i^\mu \right\rangle = 0, \quad \left\langle b_m^\mu \right\rangle = 0, \quad \left\langle h_i^\mu h_k^\mu \right\rangle - \left\langle h_i^\mu \right\rangle \left\langle h_k^\mu \right\rangle = Q_{ik}^{\mu-1}$$
$$\left\langle h_i^\mu b_n^\mu \right\rangle - \left\langle h_i^\mu \right\rangle \left\langle b_n^\mu \right\rangle = R_{in}^{\mu-1}, \quad \left\langle b_l^\mu b_n^\mu \right\rangle - \left\langle b_l^\mu \right\rangle \left\langle b_n^\mu \right\rangle = \delta_{ln}. \tag{18}$$

Hence, in both cases, the joint density of $h_{1,2}^\mu$ and $b_{1,2}^\mu$ is fully specified by the values of the order parameters in the previous time step and the parameters of the model density. This important result enables us to perform an average of the recursion relations (Equation (15)) over the latest training example in terms of Gaussian integrals. Moreover, the resulting r.h.s. can be expressed in closed form in $\{R_{im}^{\mu-1}, Q_{ik}^{\mu-1}\}$.

(d) *Self-Averaging Properties*

The self-averaging property of order parameters makes it possible to restrict the description to their mean values: Fluctuations of the stochastic dynamics can be neglected in the limit $N \to \infty$. This concept has been borrowed from the statistical physics of disordered materials and has been applied frequently in the study of neural network models and learning processes [4–7]. For a detailed mathematical discussion in the context of sequential on-line learning, see [58].

Consequently, we can interpret the averaged Equation (15) directly as deterministic recursions for the means of $\{R_{im}^\mu, Q_{ik}^\mu\}$ which coincide with their actual values in the thermodynamic limit.

(e) *Continuous Time Limit and ODE*

For $N \to \infty$, we can interpret the ratios on the left hand sides of Equation (15) as derivatives with respect to the continuous learning time

$$\alpha = \mu / N. \tag{19}$$

This scaling corresponds to the plausible assumption that the number of examples required for successful training is proportional to the number of degrees of freedom in the system.

Averages are performed over the joint densities $P(\{h_i, b_m\})$ corresponding to the most recent, independently drawn input vector. Here, and in the following, we have omitted the index μ.

The resulting sets of coupled ODE obtained from Equation (15) are of the generic form:

$$\left[\frac{dR_{im}}{d\alpha}\right]_{stat} = \eta F_{im} \quad \text{and} \quad \left[\frac{dQ_{ik}}{d\alpha}\right]_{stat} = \eta G^{(1)}_{ik} + \eta^2 G^{(2)}_{ik}. \tag{20}$$

Here, the subscript *stat* indicates that the ODE describe learning from a stationary density, cf. Equation (4) or (13).

LVQ: For the classification model, we have to insert the terms

$$F_{im} = \left(\langle b_m f_i\rangle - R_{im}\langle f_i\rangle\right),$$
$$G^{(1)}_{ik} = \Big(\langle h_i f_k + h_k f_i\rangle - Q_{ik}\langle f_i + f_k\rangle\Big) \text{ and } G^{(2)}_{ik} = \sum_{m=1,2} v_m p_m \langle f_i f_k\rangle_m \tag{21}$$

with the LVQ1 modulation functions f_i from Equation (3) and (conditional) averages with respect to the density (Equation (4)).

SCM: In the modeling of regression in a student–teacher scenario, we obtain

$$F_{im} = \langle \rho_i b_m\rangle, \quad G^{(1)}_{ik} = \Big\langle (\rho_i h_k + \rho_k h_i)\Big\rangle \text{ and } G^{(2)}_{ik} = \langle \rho_i \rho_k\rangle \tag{22}$$

where the quantities ρ_i are defined in Equation (12) for the latest input vector and averages are performed over the isotropic input density (Equation (13)).

In both training scenarios considered here, the r.h.s. of Equation (20), as given by Equations (21) and (22), can be expressed in terms of elementary functions. For the straightforward yet lengthy results, we refer the reader to the original literature for LVQ [39,46] and SCM [31–34].

(f) *Generalization error*

After training, the success of learning is quantified in terms of the generalization error ϵ_g, which can also be expressed as a function of order parameters.

LVQ: In the case of classification, ϵ_g is given as the probability of misclassifying a novel, randomly drawn input vector. In the LVQ model, class-specific errors corresponding to data from clusters $k = 1,2$ in Equation (4) can be considered separately:

$$\epsilon_g = p_1 \epsilon^1_g + p_2 \epsilon^2_g, \quad \text{where } \epsilon^k_g = \Big\langle \Theta\left(d_k - d_{\widehat{k}}\right)\Big\rangle_k \tag{23}$$

is the class-specific misclassification rate, i.e., the probability for an example drawn from a cluster k to be assigned to $\widehat{k} \neq k$ with $d_k > d_{\widehat{k}}$. For the derivation of the class-wise and total generalization error for systems with two prototypes as functions of the order parameters, we also refer to [39]. One obtains

$$\epsilon^k_g = \Phi\left(\frac{Q_{kk} - Q_{\widehat{k}\widehat{k}} - 2\lambda(R_{kk} - R_{\widehat{k}k})}{2\sqrt{v_k}\sqrt{Q_{11} - 2Q_{12} + Q_{22}}}\right) \text{ where } \Phi(z) = \int_{-\infty}^{z} dx\, \frac{e^{-x^2/2}}{\sqrt{2\pi}}. \tag{24}$$

SCM: For regression, the generalization error is defined as an average $\langle\cdots\rangle$ of the quadratic deviation between student and teacher output over the isotropic density, cf. Equation (13):

$$\epsilon_g = \frac{1}{2}\left\langle \left(\sum_{k=1}^{K} \text{erf}\left[\frac{h_k}{\sqrt{2}}\right] - \sum_{m=1}^{M} \text{erf}\left[\frac{b_m}{\sqrt{2}}\right]\right)^2 \right\rangle, \tag{25}$$

the full form of which can be found in [31,32] for arbitrary K and M. For $K = M = 2$ with orthonormal teacher vectors, it simplifies to

$$\epsilon_g = \frac{1}{3} + \frac{1}{\pi} \left[\sum_{i,k=1}^{2} \arcsin\left(\frac{Q_{ik}}{\sqrt{1+Q_{ii}}\sqrt{1+Q_{kk}}} \right) - 2 \sum_{i,m=1}^{2} \arcsin\left(\frac{R_{im}}{\sqrt{2}\sqrt{1+Q_{ii}}} \right) \right]. \tag{26}$$

(g) *Learning curves*

The (numerical) integration of the ODE for a given particular training algorithm, model density and specific initial conditions, $\{R_{im}(0), Q_{ik}(0)\}$ yields the temporal evolution of order parameters in the course of training.

Exploiting the self-averaging properties of order parameters once more, we can obtain the learning curves $\epsilon_g(\alpha) = \epsilon_g\left(\{R_{im}(\alpha), Q_{ik}(\alpha)\}\right)$, i.e., the generalization error after on-line training with $(\alpha\, N)$ random examples.

2.4. The Learning Dynamics Under Concept Drift

The analysis summarized in the previous section concerns learning in the presence of a stationary concept, i.e., for a density of the form of Equation (4) or (13) with characteristic vectors $\vec{B}_{1,2}$ which do not change in the course of training. Here, we discuss the effect of concept drift on the learning process within the modeling framework and consider weight decay as an explicit mechanism of forgetting.

2.4.1. Virtual Drift

Several virtual drift processes can be studied in appropriate modifications of the basic framework. Virtual drifts affect the statistical properties of observed example data, while the actual target task remains the same. As one example, time-varying label noise could be incorporated into both models in a straightforward way [5–7]. Similarly, non-stationary cluster variances in the input density, cf. Equation (4), can be considered by assuming explicitly time-dependent $v_\sigma(\alpha)$ in Equation (20). A particularly relevant case would be that of non-stationary prior probabilities $p_\sigma(\alpha)$ in classification, where a varying fraction of examples represents each of the classes in the data stream. In practical situations, varying class bias can complicate the training significantly and lead to inferior performance.

We will investigate these and similar, purely virtual drift processes in forthcoming studies.

2.4.2. Real Drift

In the presented framework, a real drift can be modeled as a process which displaces the characteristic vectors $\vec{B}_{1,2}$ (cluster centers in LVQ, teacher weight vectors in the SCM) in the N-dimensional feature space. Various scenarios could be considered; we restrict ourselves to the analysis of a random diffusion of vectors $\vec{B}_{1,2}(\mu)$. Upon presentation of example μ, we assume that random vectors $\vec{B}_{1,2}(\mu)$ are generated which satisfy the conditions

$$\vec{B}_1(\mu)\cdot\vec{B}_1(\mu-1) = \vec{B}_2(\mu)\cdot\vec{B}_2(\mu-1) = \left(1-\frac{\delta}{N}\right)$$

$$\vec{B}_1(\mu)\cdot\vec{B}_2(\mu) = 0 \text{ and } \mid \vec{B}_1(\mu)\mid^2 = \mid \vec{B}_2(\mu)\mid^2 = 1. \tag{27}$$

Here, δ quantifies the strength of the drift process. The displacement of the characteristic vectors is very small in an individual training step and we assume for simplicity that orthonormality is preserved. In terms of the above defined continuous time $\alpha = \mu/N$, the drift parameter sets the time scale $1/\delta$ on which the vectors lose memory of their previous positions according to $\vec{B}_m(\alpha_1)\cdot\vec{B}_m(\alpha_o) = \exp[-\delta(\alpha_1-\alpha_o)]$ for $\alpha_1 > \alpha_o$.

The effect of such a drift process can be accounted for in the mathematical analysis of the dynamics in a straightforward way: For a given vector $\vec{w}_i \in \mathbb{R}^N$, we obtain [50–53]

$$\left[\vec{w}_i \cdot \vec{B}_k(\mu)\right] = \left(1 - \frac{\delta}{N}\right) \left[\vec{w}_i \cdot \vec{B}_k(\mu - 1)\right] \quad \text{for } k = 1, 2 \tag{28}$$

under the above specified small displacement in discrete learning time. Hence, the drift tends to decrease the student–teacher overlaps continuously which clearly deteriorates the success of training compared with the stationary case. The resulting ODE for the training dynamics in the limit $N \to \infty$ under the drift process (Equation (27)) reads

$$\left[\frac{dR_{im}}{d\alpha}\right]_{drift} = \left[\frac{dR_{im}}{d\alpha}\right]_{stat} - \delta\, R_{im} \quad \text{and} \quad \left[\frac{dQ_{ik}}{d\alpha}\right]_{drift} = \left[\frac{dQ_{ik}}{d\alpha}\right]_{stat} \tag{29}$$

with the terms $[\cdots]_{stat}$ for stationary environments taken from Equation (20). However, as the teacher vectors are time-dependent, order parameters $R_{im}(\alpha)$ correspond to the inner products $\vec{w}_i^{\mu} \cdot \vec{B}_m(\mu)$, here.

2.4.3. Weight Decay

Possible motivations for the introduction of so-called weight decay in machine learning systems range from regularization as to reduce the risk of over-fitting in regression and classification [1–3] to the modeling of forgetful memories in attractor neural networks [59,60].

Here, we introduce weight decay as an element of explicit forgetting to potentially improve the performance of the trained systems in the presence of real concept drift. To this end, we consider the multiplication of all adaptive vectors by a factor $(1 - \gamma/N)$ before the generic learning step given by $\Delta\vec{w}_i^{\mu}$ in Equation (2) or (9), respectively:

$$\vec{w}_i^{\mu} = \left(1 - \frac{\gamma}{N}\right) \vec{w}_i^{\mu-1} + \Delta\vec{w}_i^{\mu}. \tag{30}$$

Analogous modifications of perceptron training under concept drift were discussed in [50–53], and weight decay in the SCM has been studied in [61,62]. Since the multiplications with $(1 - \gamma/N)$ accumulate in the course of training, weight decay enforces an increased influence of the most recent training data as compared to earlier examples.

In the thermodynamic limit $N \to \infty$, the modified ODE for training under real drift, cf. Equation (27), and weight decay, Equation (30), are obtained in a straightforward manner and read

$$\left[\frac{dR_{im}}{d\alpha}\right]_{decay} = \left[\frac{dR_{im}}{d\alpha}\right]_{stat} - (\delta + \gamma) R_{im} \quad \text{and} \quad \left[\frac{dQ_{ik}}{d\alpha}\right]_{decay} = \left[\frac{dQ_{ik}}{d\alpha}\right]_{stat} - 2\,\gamma\, Q_{ik} \tag{31}$$

with the terms for stationary environments in absence of weight decay, Equation (20).

3. Results and Discussion

We present and discuss first results that illustrate the usefulness of the modeling framework. First, we obtain insight into the capability of LVQ to cope with concept drift in classification. Second, we investigate the non-trivial effects of drift on the on-line gradient descent training of layered neural networks in regression tasks.

3.1. Learning Vector Quantization in the Presence of Real Concept Drift

We study the typical behavior of LVQ1 under real concept drift as defined in Section 2.4.2. Throughout the following, we consider prototypes initialized as independent, normalized random vectors with no prior knowledge of the cluster structure, which corresponds to

$$Q_{11}(0) = Q_{22}(0) = 1, \;\; Q_{12}(0) = 0 \;\text{ and }\; R_{im}(0) = 0 \;\text{ for }\; i, m \in \{1, 2\}. \tag{32}$$

Figure 2a displays example learning curves $\epsilon_g(\alpha)$ for a drift with $\delta = 1$ for different learning rates, see the caption for other model parameters. Details of the initial phase of training, depend on the interplay of initial values $Q_{ii}(0)$ and the learning rate. Note that a non-monotonic behavior of $\epsilon_g(\alpha)$ can be observed for some settings.

Monte Carlo simulations show excellent agreement with the $(N \to \infty)$ theoretical predictions already for relatively small systems. This parallels the findings presented in [39,46] for stationary environments. As just one example, Figure 2a also shows the mean and standard deviation of ϵ_g over 25 randomized runs of the training for $\eta = 1$ and $N = 1000$. A systematic comparison and discussion of the N-dependence in computer experiments of LVQ under concept drift will be presented elsewhere.

The results for large α show that the success of learning, i.e., the degree to which the drifting concept can be tracked by LVQ1, depends on the learning rate in a non-trivial way. In contrast to learning in stationary environments, the use of very small learning rates obviously fails to maintain the ability to generalize in the presence of a significant real drift. On the other hand, too large learning rates result in inferior performance as well.

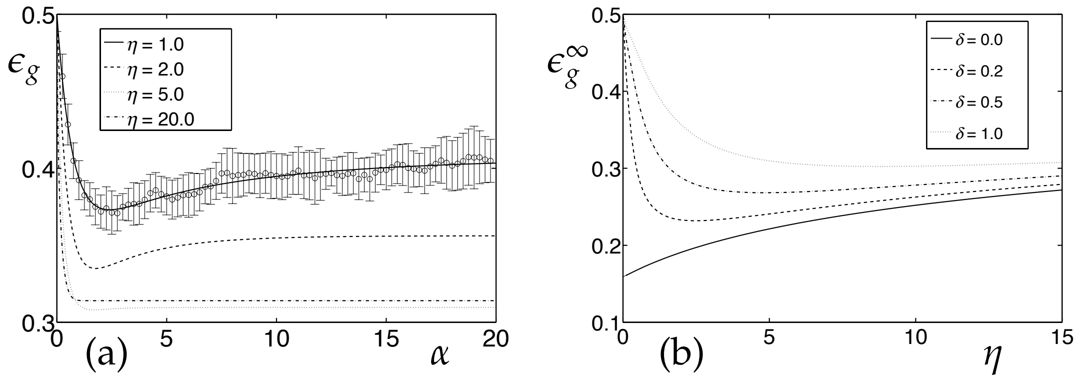

Figure 2. *LVQ under Concept Drift: Learning Curves and the Role of the Learning Rate.* LVQ1 training from data according to the model density (Equation (4)) with $\lambda = 1, p_1 = p_2 = 0.5$ and $v_1 = v_2 = 0.5$ in the presence of real concept drift. (**a**) Learning curves $\epsilon_g(\alpha)$ for $\delta = 1$ and various learning rates η. Symbols and error bars mark the mean results and standard deviations observed in 25 randomized simulations for $N = 1000$ with $\eta = 1$ as an example. (**b**) Asymptotic $(\alpha \to \infty)$ generalization error as a function of the learning rate η for different drift parameters δ and in the stationary environment with $\delta = 0$.

After presenting many examples, i.e., in the limit $\alpha \to \infty$, the system approaches a quasi-stationary state in which the LVQ prototypes track the drifting center vectors $\vec{B}_{1,2}$ with constant overlap parameters R_{im}, Q_{ik}. The configuration corresponds to the stationarity conditions

$$\left[\frac{dR_{im}}{d\alpha}\right]_{drift} = 0 \quad \text{and} \quad \left[\frac{dQ_{ik}}{d\alpha}\right]_{drift} = 0. \tag{33}$$

Figure 2b shows the $\alpha \to \infty$ asymptotic generalization error $\epsilon_g^\infty = \lim_{\alpha \to \infty} \epsilon_g(\alpha)$ as a function of η. Only in absence of drift, i.e., for $\delta = 0$, the best possible generalization ability of LVQ1 is obtained in the limit $\eta \to 0$. We refer the reader to [39,46] for a detailed discussion of ϵ_g^∞ and its dependence of the model parameters $\lambda, p_\pm$ and $v_\pm$. For $\delta > 0$, the limit $\eta \to 0$ results in trivial asymptotic behavior corresponding to random guesses, with $\epsilon_g^\infty = 1/2$ for the symmetric input density with $p_1 = p_2$ and $v_1 = v_2$, for instance.

Given the drift parameter δ, an optimal constant learning rate can be identified with respect to the generalization ability in the quasi-stationary state. The use of this learning rate yields, for $\alpha \to \infty$, the best ϵ_g^∞ achievable under drift. It is displayed in Figure 3a as a function of δ for small values of the drift parameter. The optimal quasi-stationary generalization error under concept drift scales is:

$$\left[\epsilon_g^\infty(\delta) - \epsilon_g^\infty(0)\right] \propto \delta^{1/2} \text{ for small } \delta. \tag{34}$$

As expected, the drift impedes the learning process. However, our results show that already the simplest LVQ scheme is capable of tracking randomly drifting clusters and to maintain a significant generalization ability, even in very high-dimensional spaces.

We have also studied the effect of weight decay in the presence of the above discussed real concept drift. Figure 3b displays example learning curves for LVQ1 training with various weight decay parameters γ for a given learning rate η. As these examples show, the implementation of weight decay has the potential to improve the generalization behavior significantly when tracking a drifting concept. The simultaneous optimization of learning rate and weight decay $\{\eta, \gamma\}$ with respect to the success of training in the tracking state will be addressed in forthcoming studies.

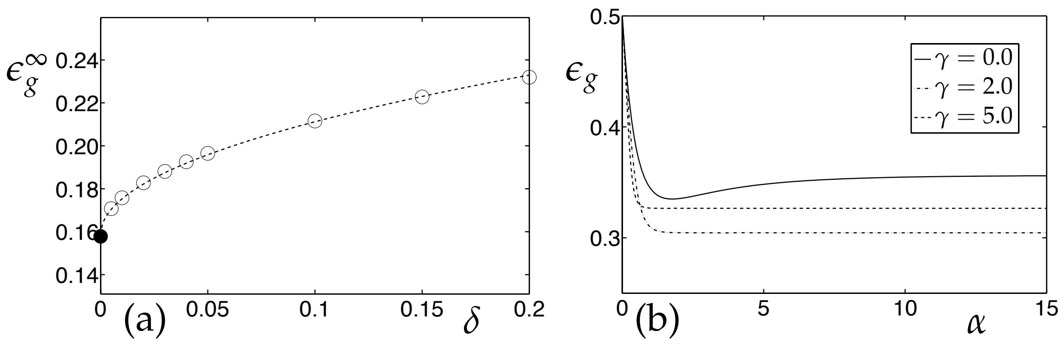

Figure 3. *LVQ under Concept Drift: Asympotic Generalization and the Influence of Weight Decay.* LVQ1 in the presence of a real drift with model parameters $\lambda = 1, v_1 = v_2 = 0.5, p_1 = p_2 = 0.5$. (**a**) The $(\alpha \to \infty)$ asymptotic generalization error of LVQ1 as obtained with an optimized constant learning rate. Empty circles correspond to numerical results for different drift parameters, the filled circle represents stationary data, for which $\epsilon_g^\infty(\delta = 0) \approx 0.158$. The dashed line corresponds to a fit of the form $\epsilon_g^\infty(\delta{=}0) + 0.166\,\delta^{1/2}$. (**b**) Learning curves in the model with learning rate $\eta = 2.0$ and drift parameter $\delta = 1.0$. The three curves correspond to learning without weight decay (upper, solid line), with $\gamma = 2$ (lower, dash-dotted line) and $\gamma = 5$ (middle, dashed line).

3.2. SCM Regression in the Presence of Real Concept Drift

Here, we present results concerning the SCM student-teacher scenario with $K = M = 2$. Already in this simplest setting and in absence of concept drift, the learning dynamics displays non-trivial phenomena which have been studied in detail in, among others, [31,32,34]. Perhaps the most interesting effect is the occurrence of quasi-stationary plateau-states which can even dominate the learning curves $\epsilon_g(\alpha)$. They reflect the existence of weakly repulsive fixed points of the ODE (Equation (20)) and correspond to sub-optimal, more or less symmetric configurations of the student network. The problem of delayed learning due to saddle points and related effects in gradient-based training is obviously also of interest in the context of Deep Learning (see [3,37,63,64] for recent investigations and further references).

In the SCM model, one can show that a plateau with $R_{ik} \approx R$ and and $Q_{ik} \approx Q$ for all $i,k \in \{1,2\}$ always exists in the case of orthonormal teacher vectors and for small learning rates [31,32,34]. In this state, all student weight vectors have acquired the same, limited knowledge of the target rule. To achieve better generalization ability, this symmetry has to be broken or, in other words, the student hidden units have to specialize and represent specific units of the teacher network.

Note that more complex fixed point configurations with different degrees of (partial) specialization can be found, in general. The number of observable plateaus depends on the learning rate and increases for larger K and M (see [34] for a detailed discussion in the absence of drift).

In practice, one expects $R_{im}(0) \approx 0$ for all i, m unless prior knowledge is available about the target. Hence, the student specialization $S_i(0) = |R_{i1}(0) - R_{i2}(0)|$ is also expected to be small, initially. A nearly unspecialized configuration with $S_i(\alpha) \approx 0$ persists in a transient phase of learning, which can extend over large values of α. The actual shape and length of the plateau depends on the precise initialization and the repulsive properties of the corresponding fixed point of the dynamics (see [34] for a detailed discussion, which also addresses the effect of finite N in Monte Carlo simulations).

Figure 4a shows an example (lowest curve) of a pronounced plateau state in on-line gradient descent for initial conditions

$$R_{im} = R_o + U(10^{-5}) \text{ with } R_o = 0.01, \;\; Q_{11} = Q_{22} = 0.5, Q_{12} = 0.49. \tag{35}$$

Here, $U(X)$ denotes a random number drawn from the interval $(0, X]$ with uniform probability, hence also $S_i(0) = \mathcal{O}(X)$. The initialization corresponds to nearly identical student vectors with little prior knowledge. It is inspired by the analyses in [32,34] which showed that the actual value of R_o is largely irrelevant for the observed plateau length, while it depends logarithmically on X [34]. Corresponding Monte Carlo simulations are shown in Figure 4a for $N = 500$ and randomly drawn initial student vectors, resulting in $R_{im}(0) = \mathcal{O}(1/\sqrt{N})$, with $Q_{ik}(0)$ fixed according to Equation (35). Simulations confirm the theoretical predictions very well, qualitatively.

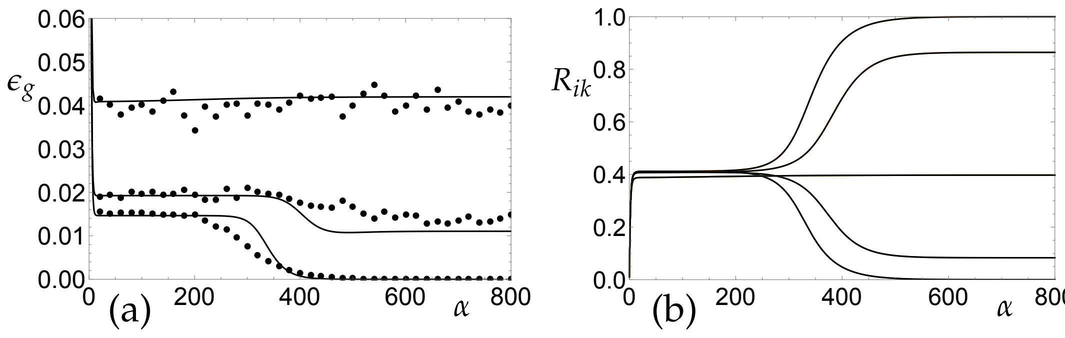

Figure 4. *Regression under Concept Drift: Learning Curves*. Gradient-based training of the Soft Committee Machine with $K = M = 2$ and orthogonal teacher vectors in the presence of real target drift, with learning rate $\eta = 0.5$ and initial conditions as specified in Equation (35). (**a**) Learning curves for the stationary case with $\delta = 0$ (lower line), for weak drift with $\delta = 0.005$ (middle) and for strong drift with $\delta = 0.03$ (upper line). Symbols represent the result of single Monte Carlo simulation runs for system size $N = 500$. (**b**) The corresponding evolution of the student–teacher overlaps $R_{11} = R_{22}$ and $R_{12} = R_{21}$ vs. α for the stationary case with $\delta = 0$ (lower and upper lines), for weak drift with $\delta = 0.005$ (intermediate) and strong drift with $\delta = 0.03$ (center, all overlaps equal).

For very slow drifts of the target concept, the behavior is still similar to the stationary case. For an example with $\delta = 0.005$, Figure 4a shows the $N \to \infty$ theoretical learning curve and Monte Carlo simulations: After a rapid, initial decrease of the generalization error, a quasi-stationary, unspecialized plateau is reached. Eventually, the symmetry is broken and the system approaches its $\alpha \to \infty$ asymptotic state, in which a smaller but non-zero $\epsilon_g^{\infty}(\delta)$ is achieved. Obviously, on-line gradient descent training enables the SCM to track the drifting target to a reasonable degree and maintains a specialized hidden unit configuration. The precise influence of finite size effects on the shape and length of plateaus in Monte Carlo simulations will be studied in greater detail in forthcoming projects.

The behavior changes significantly in the presence of stronger concept drifts: The SCM remains unspecialized even for $\alpha \to \infty$ and, consequently, the achievable generalization ability is relatively poor. Figure 4a displays the corresponding learning curve for $\delta = 0.03$ as an example, together with the result of a single Monte Carlo simulation.

Figure 4b shows the evolution of the overlap parameters $R_{im}(\alpha)$ corresponding to the learning curves displayed in Figure 4a. While for $\delta = 0.005$ the student units still specialize, the unspecialized plateau state with $R_{im} \approx R$ for all i, m persists for $\delta = 0.03$.

In Figure 5a, this is illustrated in terms of the (quasi-)stationary values of ϵ_g: The system can benefit from the specialization in terms of a low $\alpha \to \infty$ asymptotic generalization error (solid line). For $\delta \approx 0$, the achievable generalization error increases linearly with the drift parameter: $\epsilon_g^\infty(\delta) \propto \delta$. Note that $\epsilon_g^\infty(\delta = 0) = 0$ in the perfectly learnable scenario with $K = M$ considered here. On the contrary, for larger δ, the only stable fixed point of the system coincides with an unspecialized configuration (dashed line). The generalization error of the latter also displays a linear dependence on δ for slow drifts.

Weight decay can improve the performance slightly in the presence of weak concept drifts. As displayed in Figure 5a, for an example drift of $\delta = 0.015$, the parameter γ in Section 2.4.3 can be tuned to decrease the achievable generalization error in the unspecialized plateau (dashed line) and, more importantly, in the final quasi-stationary tracking state (solid line). Specialization cannot be achieved if the weight decay parameter is set too large. A more detailed analysis of the interplay of learning rate and weight decay will be presented in a forthcoming publication.

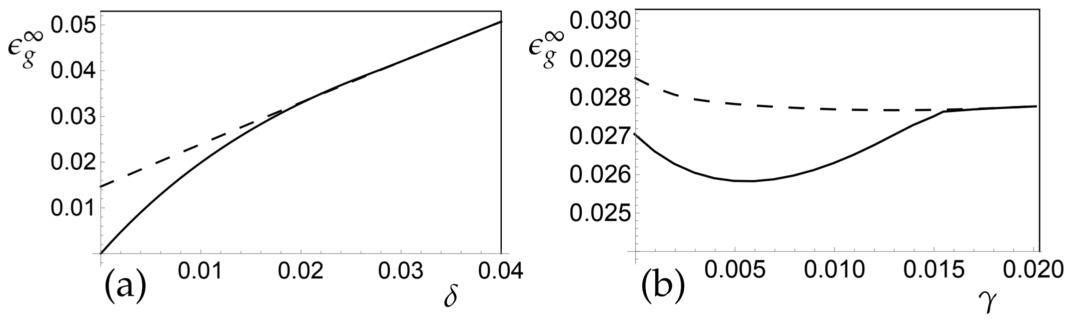

Figure 5. *Regression under Concept Drift: Plateaus and Specialized States*. Soft Committee Machine, regression in the presence of real target drift, learning rate and model parameters as in Figure 4. (**a**) The generalization error vs. the drift parameter δ for $\gamma = 0$, in the symmetric plateau state with $R_{11} = R_{22}$ and $R_{12} = R_{21}$ (dashed line) and in the $\alpha \to \infty$ stationary state (solid). (**b**) The influence of weight decay: For a given drift with $\delta = 0.015$, the $\alpha \to \infty$ asymptotic generalization error is displayed as a function of the weight decay parameter γ. In addition, the dashed line marks ϵ_g in the unspecialized plateau state.

4. Conclusions

Here, we conclude with a brief summary, provide an outlook on potential follow-up studies and discuss major challenges and open questions.

4.1. Brief Summary

In this contribution, we present a modeling framework which facilitates the systematic study and exact mathematical description of on-line learning in the presence of concept drift. The framework is illustrated by the analysis of two model scenarios: The learning of a classification scheme is exemplified in terms of prototype-based Learning Vector Quantization, trained from a stream of clustered input data. Regression problems are addressed in the context of gradient-based training of the Soft Committee Machine, a two-layered feed forward neural network with nonlinear hidden unit activation. Here, the analysis is done in the frame of a student–teacher scenario. In both setups, we study the influence of real drifts, where the target classification or regression scheme are subject to a randomized drift process.

Most importantly, we demonstrate that the presented framework is suitable for the mathematical analysis of a variety of learning and drift scenarios, including weight decay as a possible mechanism of explicit forgetting.

A discussion of the findings in detail is provided in the previous section. In brief, we show that the simple LVQ1 prescription is indeed capable of tracking time-dependent classification schemes in high-dimensional input space under randomized drift. Regression under concept drift displays non-trivial effects in terms of the success of gradient based adaptation in SCM networks. In particular, we observe the drift-induced persistence of unspecialized, sub-optimal plateaus in the learning curve. Thus, on-line learning can display quite different behavior in the presence of concept drift, depending on the underlying target and its properties. In both settings considered here, weight decay has the potential to improve the generalization behavior under drift in the quasi-stationary tracking state.

4.2. Future Work and Extensions

In the present contribution, we study only a few, simple scenarios in terms of the considered targets, drift processes and student systems. Several interesting topics can be addressed readily by straightforward modifications of the models:

- The systematic investigation of virtual drifts as in, for instance, non-stationary label noise, prior weights $p_{1,2}$ or cluster separation λ is readily possible by consideration of explicitly time-dependent ODE.
- Alternative LVQ prescriptions, as studied in [39,46–48] for stationary data, can be systematically compared in terms of their potential to deal with concept drift.
- Similarly, modifications of the basic gradient descent scheme can be considered under concept drift in the SCM student–teacher scenario (seem for instancem [35,36,38]).
- Deterministic concept drifts, similar to the processes studied in the context of perceptron training in [50–53], can be considered as well. This way, learning from an *adversary* can be modeled, where the modification of the target depends explicitly on the actual student configuration.
- The restriction to LVQ systems with one prototype per class results, effectively, in the parameterization of linear class boundaries only. This limitation can be lifted by considering distances different from the simple Euclidean measure (see, e.g., [29]). Alternatively, systems with several prototypes per class correspond to non-linear (piece-wise linear) decision boundaries which has non-trivial effects on the training dynamics, as demonstrated for stationary environments in [49].
- Similarly, the investigation of SCM student–teacher scenarios with more general settings of K and M will provide insight into the interplay of concept drift with the larger number of possible plateau states for $K, M > 2$. Over- and under-fitting effects in mismatched situations with $K \neq M$ will be in the center of interest.
- The shallow SCM architectures studied here are limited to a single hidden layer of units. The important extension to deeper networks with several hidden layers will be addressed in forthcoming studies.
- It will be interesting to explore the extent to which the theoretically studied phenomena can be observed in practical situations. To this end, we will investigate the behavior of LVQ and SCM in realistic training set-ups with real world data streams.

4.3. Perspectives and Challenges

We have demonstrated that the presented modeling framework bears the promise to provide valuable insights into the effects of concept drift in a variety of learning scenarios. Ultimately, a better understanding of relevant phenomena should facilitate the development and optimization of robust, efficient training algorithms for lifelong machine learning. Variational approaches, as discussed in, for instance [5–8,35,52,53], could play an important role in this context.

One of the most important challenges, in particular for active methods, is the reliable detection of concept drift in a stream of data. Learning systems should be able to infer not only the nature of the drift (e.g., virtual vs. real), but also estimate its strength in order to tune algorithm parameters such as learning rate or weight decay appropriately. It would be interesting to extend the framework towards such methods, which often rely on the variability of surrogates, such as changes of the observed classification error. The proposed analytical approach would enable us to obtain formal insight into the behavior of the surrogate characteristics in concrete models.

Recently suggested strategies for continual learning include so-called Dedicated Memory Models and the appropriate combination of off-line and on-line learning [21,65,66]. Suitable rejection mechanisms for the mitigation of concept drift were recently considered in [67]. Extensions of our modeling approach in these directions would be highly desirable.

Author Contributions: Conceptualization: C.G., B.H. and M.B.; Formal analysis: M.S., F.A. and M.B.; Investigation: M.S., F.A. and M.B.; Methodology: C.G., B.H. and M.B.; Software: M.S. and F.A.; Writing—original draft: M.B.; and Writing—review and editing: M.S., F.A., C.G., B.H. and M.B.

Funding: Michael Biehl acknowledges support through the Northern Netherlands Region of Smart Factories (RoSF) consortium, led by the Noordelijke Ontwikkelings en Investerings Maatschappij (NOM), The Netherlands, see http://www.rosf.nl.

Acknowledgments: We thank Anarta Ghosh, Gert-Jan de Vries and Aree Witoelar for helpful discussions regarding earlier, joint projects concerning the dynamics of LVQ [39,46–49].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CLT	Central Limit Theorem
LVQ	Learning Vector Quantization
NPC	Nearest Prototype Classification
ODE	Ordinary Differential Equations
r.h.s.	right hand side
SCM	Soft Committee Machine
WTA	Winner Takes All

References

1. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: New York, NY, USA, 2001.
2. Bishop, C. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
3. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
4. Hertz, J.A.; Krogh, A.S.; Palmer, R.G. *Introduction to the Theory of Neural Computation*; Addison-Wesley: Redwood City, CA, USA, 1991.
5. Engel, A.; van den Broeck, C. *The Statistical Mechanics of Learning*; Cambridge University Press: Cambridge, UK, 2001.
6. Seung, S.; Sompolinsky, H.; Tishby, N. Statistical mechanics of learning from examples. *Phys. Rev. A* **1992**, *45*, 6056–6091. [CrossRef] [PubMed]
7. Watkin, T.L.H.; Rau, A.; Biehl, M. The statistical mechanics of learning a rule. *Rev. Mod. Phys.* **1993**, *65*, 499–556. [CrossRef]
8. Biehl, M.; Caticha, N. The statistical mechanics of on-line learning and generalization. In *The Handbook of Brain Theory and Neural Networks*; Arbib, M.A., Ed.; MIT Press: Cambridge, MA, USA, 2003; pp. 1095–1098.
9. Biehl, M.; Caticha, N.; Riegler, P. Statistical mechanics of on-line learning. In *Similiarity Based Clustering*; Lecture Notes in Artificial Intelligence; Biehl, M., Hammer, B., Verleysen, M., Villmann, T., Eds.; Springer: Cham, Switzerland, 2009; Volume 5400, pp. 1–22.
10. Zliobaite, I.; Pechenizkiy, M.; Gama, J. An overview of concept drift applications. In *Big Data Analysis: New Algorithms for a New Society*; Big Data Analysis; Japkowicz, N., Stefanowski, J., Eds.; Springer: Cham, Switzerland, 2016; Volume 16.
11. Losing, V.; Hammer, B.; Wersing, H. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing* **2017**, *275*, 1261–1274. [CrossRef]

12. Ditzler, G.; Roveri, M.; Alippi, C.; Polikar, R. Learning in nonstationary environment: A survey. *Comput. Intell. Mag.* **2015**, *10*, 12–25. [CrossRef]
13. Joshi, J.; Kulkarni, P. Incremental learning: areas and methods—A survey. *Int. J. Data Min. Knowl. Manag. Process* **2012**, *2*, 43–51. [CrossRef]
14. Ade, R.; Desmukh, P. Methods for incremental learning: A survey. *Int. J. Data Min. Knowl. Manag. Process.* **2013**, *3*, 119–125.
15. De Francisci Morales, G.; Bifet, A. SAMOA: Scalable advanced massive online analysis. *J. Mach. Learn. Res.* **2015**, *16*, 149–153.
16. Grandinetti, L.; Lippert, T.; Petkov, N. (Eds.) *Brain-Inspired Computing. International Workshop BrainComp 2013*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8603.
17. Amunts, K.; Grandinetti, L.; Lippert, T.; Petkov, N. (Eds.) *Brain-Inspired Computing. Second International Workshop BrainComp 2015*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 10087.
18. Faria, E.R.; Gonçalves, I.J.C.R.; de Carvalho, A.C.P.L.F.; Gama, J. Novelty detection in data streams. *Artif. Intell. Rev.* **2016**, *45*, 235–269. [CrossRef]
19. Krawczyk, B.; Minku, L.L.; Gama, J.; Stefanowski, J.; Wozniak, M. Ensemble learning for data stream analysis: A survey. *Inf. Fusion* **2017**, *37*, 132–156. [CrossRef]
20. Gomes, H.M.; Bifet, A.; Read, J.; Barddal, J.P.; Enembreck, F.; Pfharinger, B.; Holmes, G.; Abdessalam, T. Adaptive random forests for evolving data stream classification. *Mach. Learn.* **2017**, *106*, 1469–1495. [CrossRef]
21. Losing, V.; Hammer, B.; Wersing, H. Tackling heterogeneous concept drift with the Self-Adjusting Memory (SAM). *Knowl. Inf. Syst.* **2018**, *54*, 171–201. [CrossRef]
22. Loeffel, P.-X.; Marsala, C.; Detyniecki, M. Classification with a reject option under Concept Drift: The Droplets algorithm. In Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA 2015), Paris, France, 19–21 October 2015; IEEE: New York, NY, USA, 2015; pp. 1–9.
23. Janakiraman, V.M.; Nguyen, X.; Assanis, D. Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines. *Neurocomput* **2016**, *177*, 304–316. [CrossRef]
24. Benczúr, A.A.; Kocsis, L.; Pálovics, R. Online machine learning in big data streams. *arXiv* **2018**, arxiv:1802.05872. Available online: http://arxiv.org/abs/1802.05872 (accessed on 13 August 2018).
25. Kohonen, T.; Barna, G.; Chrisley, R. Statistical pattern recognition with neural network: Benchmarking studies. In Proceedings of the IEEE second international conference on Neural Networks, San Diego, CA, USA, 24–27 July 1988; IEEE: New York, NY, USA, 1988; Volume 1, pp. 61–68.
26. Kohonen, T. *Self-Organizing Maps*; Springer: New York, NY, USA, 2001.
27. Kohonen, T. Improved versions of Learning Vector Quantization. In Proceedings of the 1990 IJCNN International Joint Conference on Neural Networks, San Diego, CA, USA, 17–21 June 1990; Volume 1, pp. 545–550.
28. Nova, D.; Estevez, P.A. A review of Learning Vector Quantization classifiers. *Neural Comput. Appl.* **2014**, *25*, 511–524. [CrossRef]
29. Biehl, M.; Hammer, B.; Villmann, T. Prototype-based models in machine learning. *WIREs Cogn. Sci.* **2016**, *7*, 92–111. [CrossRef] [PubMed]
30. Biehl, M.; Schwarze, H. Learning by on-line gradient descent. *J. Phys. A Math. Gen.* **1995**, *28*, 643–656. [CrossRef]
31. Saad, D.; Solla, S.A. Exact solution for on-line learning in multilayer neural. *Phys. Rev. Lett.* **1995**, *74*, 4337–4340. [CrossRef] [PubMed]
32. Saad, D.; Solla, S.A. On-line learning in soft committee machines. *Phys. Rev. E* **1995**, *52*, 4225–4243. [CrossRef]
33. Riegler, P.; Biehl, M. On-line backpropagation in two-layered neural networks. *J. Phys. A Math. Gen.* **1995**, *28*, L507–L513. [CrossRef]
34. Biehl, M.; Riegler, P.; Wöhler, C. Transient dynamics of on-line learning in two-layered neural networks. *J. Phys. A Math. Gen.* **1996**, *29*, 4769–4780. [CrossRef]
35. Vicente, R.; Caticha, N. Functional optimization of online algorithms in multilayer neural networks. *J. Phys. A Math. Gen.* **1997**, *30*, L599–L605. [CrossRef]
36. Inoue, M.; Park, H.; Okada, M. On-line learning theory of soft committee machines with correlated hidden units-steepest gradient descent and natural gradient descent. *J. Phys. Soc. Jpn.* **2003**, *72*, 805–810. [CrossRef]

37. Marcus, G. Deep learning: A critical appraisal. *arXiv* **2018**, arxiv:1801.00631. Available online: http://arxiv.org/abs/1801.00631 (accessed on 27 August 2018).
38. Saad, D. (Ed.) *On-Line Learning in Neural Networks*; Cambridge University Press: Cambridge, UK, 1999.
39. Biehl, M.; Ghosh, A.; Hammer, B. Dynamics and generalization ability of LVQ algorithms. *J. Mach. Learn. Res.* **2007**, *8*, 323–360.
40. Biehl, M.; Freking, A.; Reents, G. Dynamics of on-line competitive learning. *Europhys. Lett.* **1997**, *38*, 73–78. [CrossRef]
41. Biehl, M.; Freking, A.; Reents, G.; Schlösser, E. Specialization processes in on-line unsupervised learning. *Phil. Mag. B* **1999**, *77*, 1487–1494. [CrossRef]
42. Biehl, M.; Schlösser, E. The dynamics of on-line principal component analysis. *J. Phys. A Math. Gen.* **1998**, *31*, L97–L103. [CrossRef]
43. Barkai, N.; Seung, H.S.; Sompolinksy, H. Scaling laws in learning of classification tasks. *Phys. Rev. Lett.* **1993**, *70*, 3167–3170. [CrossRef] [PubMed]
44. Marangi, C.; Biehl, M.; Solla, S.A. Supervised learning from clustered input examples. *Europhys. Lett.* **1995**, *30*, 117–122. [CrossRef]
45. Meir, R. Empirical risk minimization versus maximum-likelihood estimation: a case study. *Neural Comput.* **1995**, *7*, 144–157. [CrossRef]
46. Ghosh, A.; Biehl, M.; Hammer, B. Performance analysis of LVQ algorithms: a statistical physics approach. *Neural Netw.* **2006**, *19*, 817–829. [CrossRef] [PubMed]
47. Biehl, M.; Ghosh, A.; Hammer, B. The dynamics of Learning Vector Quantization. In Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN 2005), Bruges, Belgium, 27–29 April 2005; Verleysen, M., Ed.; D-Side: Evere, Belgium, 2005; pp. 13–18.
48. Ghosh, A.; Biehl, M.; Hammer, B. Dynamical analysis of LVQ type learning rules. In Proceedings of the 5th Workshop on the Self-Organizing-Map (WSOM 2005), Paris, France, 5–8 September 2005; Cottrell, M., Ed.; Université de Paris: Paris, France, 2005; 6p.
49. Witoelar, A.; Ghosh, A.; de Vries, J.J.G.; Hammer, B.; Biehl, M. Window-based example selection in learning vector quantization. *Neural Comput.* **2010**, *22*, 2924–2961. [CrossRef] [PubMed]
50. Biehl, M.; Schwarze, H. On-line learning of a time-dependent rule. *Europhys. Lett.* **1992**, *20*, 733–738. [CrossRef]
51. Biehl, M.; Schwarze, H. Learning drifting concepts with neural networks. *J. Phys. A Math. Gen.* **1993**, *26*, 2651–2665. [CrossRef]
52. Kinouchi, O.; Caticha, N. Lower bounds on generalization errors for drifting rules. *J. Phys. A Math. Gen.* **1993**, *26*, 6161–6172. [CrossRef]
53. Vicente, R.; Caticha, N. Statistical mechanics of online learning of drifting concepts: A variational approach. *Mach. Learn.* **1998**, *32*, 179–201. [CrossRef]
54. Biehl, M.; Hammer, B.; Villmann, T. Distance measures for prototype based classification. In *International Workshop on Brain-Inspired Computing*; Springer: Cham, Switzerland, 2013; pp. 110–116.
55. Biehl, M.; Schlösser, E.; Ahr, M. Phase transitions in soft-committee machines. *Europhys. Lett.* **1998**, *44*, 261–266. [CrossRef]
56. Ahr, M.; Biehl, M.; Urbanczik, R. Statistical physics and practical training of soft-committee machines. *Eur. Phys. J. B* **1999**, *10*, 583–588. [CrossRef]
57. Cybenko, G. Approximations by superpositions of sigmoidal functions. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
58. Reents, G.; Urbanczik, R. Self-averaging and on-line learning. *Phys. Rev. Lett.* **1998**, *80*, 5445–5448. [CrossRef]
59. Mezard, M.; Nadal, J.P.; Toulouse, G. Solvable models of working memories. *J. Phys.* **1986**, *47*, 1457–1462. [CrossRef]
60. Van Hemmen, J.L.; Keller, G.; Kühn, R. Forgetful memories. *Europhys. Lett.* **1987**, *5*, 663–668. [CrossRef]
61. Saad, D.; Solla, S.A. Learning with noise and regularizers in multilayer neural networks. In *Advances in Neural Information Processing Systems*; Mozer, M., Jordan, M.I., Petsche, T., Eds.; MIT Press: Cambridge, MA, USA, 1997; pp. 260–266.
62. Saad, D.; Rattray, M. Learning with regularizers in multilayer neural networks. *Phys. Rev. E* **1998**, *57*, 2170–2176. [CrossRef]

63. Dauphin, Y.N.; Pascanu, R.; Gulcehre, C.; Cho, K.; Ganguli, S.; Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Proceedings of the Twenty-Eighth Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Curran Associates: Red Hook, NY, USA, 2014; pp. 2933–2941.
64. Tishby, N.; Zaslavsky, N. Deep Learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jerusalem, Israel, 26 April–1 May 2015; IEEE: New York, NY, USA, 2015; pp. 1–5.
65. Fischer, L.; Hammer, B.; Wersing, H. Combining offline and online classifiers for life-long learning (OOL). In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015), Killarney, Ireland, 12–16 July 2015; IEEE: New York, NY, USA, 2015; 8p.
66. Fischer, L.; Hammer, B.; Wersing, H. Online metric learning for an adaptation to confidence drift. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2016), Vancouver, BC, Canada, 24–29 July 2016; IEEE: New York, NY, USA, 2016; pp. 748–755.
67. Göpfert, J.P.; Hammer, B.; Wersing, H. Mitigating concept drift via rejection. In Proceedings of the 27th International Conference on Artificial Neural Networks (ICANN 2018), Rhodes, Greece, 4–7 October 2018; Kurkova, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Magogiannis, I., Eds.; Springer: New York, NY, USA, 2018; pp. 456–467.