Learning from Noisy Data: An Exactly Solvable Model

Michael Biehl, Peter Riegler, and Martin Stechert
Institut für Theoretische Physik, Universität Würzburg, Am Hubland, D–97074 Würzburg, Germany

Exact results are derived for the learning of a linearly separable rule with a single layer perceptron. We consider two sources of noise in the training data: the random inversion of the example outputs and weight noise in the teacher network respectively. In both scenarios we investigate on–line learning schemes which utilize only the latest in a sequence of uncorrelated random examples for an update of the student weights. We study Hebbian learning as well as on-line algorithms which achieve an optimal decrease of the generalization error. The latter realize an asymptotic decay of the generalization error that coincides, apart from prefactors, with the one found for off-line schemes.

87.10,02.50,05.20

Perhaps the most interesting aspect of feed–forward neural networks [1] is their ability to learn a rule from examples by adaptation of the network parameters. Successful learning enables the *student* network to *generalize*, i.e. to assign with high probability the correct rule output to an arbitrary (novel) input.

Methods from the statistical mechanics of disordered systems [2] have been used successfully to study the performance of large networks trained with random examples of an unknown rule. For reviews of this field see e.g. [3–5].

A point of particular interest is the training with *noisy data* where some stochastic process corrupts the information contained in the examples. Usually, the training procedure is formulated as an optimization process which infers as much information as possible from a given set of examples. This is done according to more or less sophisticated iterative learning schemes which are guided by some objective, for example the students performance on the *training set*. The term *off–line learning* has been coined for such algorithms [1]. So far, most statistical mechanics treatments of off–line training with noisy examples has been restricted to the replica–symmetric analysis [2] of perceptron learning [5–7]. The importance of possible corrections due to replica symmetry breaking [2] has yet to be investigated for these models.

This work revisits the training with noisy data, but in the framework of *on–line learning* [1]. In this setting only the latest in a sequence of examples is used for an update of the student weights. Examples need not to be stored explicitly in a separate device since they are not presented repeatedly to the student. In the following we perform an exact analysis of such learning processes.

We will specifically consider the single layer perceptron [1]. Such a simple feed–forward neural network realizes a linearly separable input/output relation of the form

$$S_J(\boldsymbol{\xi}) = \mathrm{sign}\,(\mathbf{J}\cdot\boldsymbol{\xi}) \tag{1}$$

where $\boldsymbol{\xi}$ represents an N–dimensional input vector. $\mathbf{J}\in \mathbb{R}^N$ is the vector of perceptron weights, which are to be adapted in the training process.

The rule to be learnt is defined through a *teacher* vector $\mathbf{B}\in\mathbb{R}^N, \mathbf{B}^2 = 1$:

$$S_B(\boldsymbol{\xi}) = \mathrm{sign}(h_B) \quad \text{with} \quad h_B = \mathbf{B}\cdot\boldsymbol{\xi}. \tag{2}$$

Thus, the concept is in principle learnable for the perceptron student. However, we assume in the following that a sequence of examples $\{\boldsymbol{\xi}^\mu, S_T^\mu\}$ is provided for training, where the example labels $S_T^\mu = \pm 1$ can differ from S_B^μ due to some stochastic process which will be specified later on.

The generic form of on–line perceptron training considered here is

$$\mathbf{J}^{\mu+1} = \mathbf{J}^\mu + \frac{1}{N} f(Q^\mu, h_J^\mu, S_T^\mu)\boldsymbol{\xi}^\mu S_T^\mu \tag{3}$$

where the weight function f defines the actual algorithm. This function depends on quantities available to the student, such as the example label S_T^μ, the student norm $Q^\mu = \mathbf{J}^\mu\cdot\mathbf{J}^\mu$, and the normalized local potential $h_J^\mu = \mathbf{J}^\mu\cdot\boldsymbol{\xi}^\mu/\sqrt{Q^\mu}$.

The input vectors are taken to be N–dimensional vectors of i.i.d. random components ξ_j^μ with zero mean and unit variance. At each "time step" μ a new, uncorrelated example is drawn and used for an update of $\mathbf{J}^\mu$ according to Eq. (3).

On–line learning with a perceptron has recently been studied in various contexts. See e.g. [8–12] for a more detailed description of the analysis.

It is straightforward to obtain recursion equations for the quantities Q^μ and $r^\mu = \mathbf{J}^\mu\cdot\mathbf{B}/\sqrt{Q^\mu}$. In the limit $N\to\infty$ these overlaps are selfaveraging with respect to the randomness in the training data. The evolution of the order parameters in "continuous time" $\alpha = \mu/N$ is given in terms of first order differential equations:

$$\begin{aligned}\frac{dr}{d\alpha} &= \left\langle \frac{f}{\sqrt{Q}}(h_B - r h_J)S_T - \frac{r}{2}\frac{f^2}{Q}\right\rangle_{\xi,S_T}\\ \frac{dQ}{d\alpha} &= \left\langle 2\sqrt{Q} f h_J S_T + f^2\right\rangle_{\xi,S_T}\end{aligned} \tag{4}$$

where $\langle\ldots\rangle_{\xi,S_T}$ denotes the averages over the input–distribution and also over the randomness in the evaluation of S_T.

Given a specific type of noise and the weight function f, we can work out the corresponding differential equations analytically. They can be integrated at least numerically and hence one obtains the evolution of the order parameters with α, the number of examples used for training. We will consider initial conditions $r(0) = Q(0) = 0$ exclusively.

It is useful to distinguish two performance measures, the *generalization error* $\epsilon_g = \langle \Theta(-h_J S_B) \rangle_\xi$ and the *prediction error* $\epsilon_p = \langle \Theta(-h_J S_T) \rangle_{\xi, S_T}$. The quantity ϵ_g is the probability for disagreement between the student and the genuine rule output and the average is only over the input distribution. In contrast, ϵ_p compares the student output with the noisy S_T for an arbitrary input. For a student with normalized overlap r with the teacher, the generalization error is given by $\epsilon_g = (1/\pi) \arccos r$ on average over the uniform input distribution [4]. Obviously, ϵ_g is zero for perfect alignment $\mathbf{J} \propto \mathbf{B}$. The minimal ϵ_p, however, will remain non–zero in general; of course it is impossible to predict the randomized S_T without errors. The relation between generalization and prediction error will depend on the specific noise considered.

A simple choice for the weight function is $f = 1$, which corresponds to Hebbian learning [13]. We will work out and solve the differential equations for this algorithm for two different types of training noise. Various other learning schemes have been considered in the literature (e.g. [8,10–12]). Here we will focus on the on–line algorithm of *optimal generalization*, which maximizes $dr/d\alpha$ and thus the decrease of ϵ_g. Following Kinouchi and Caticha [9] one obtains the *optimal weight function*

$$f_{opt} = \frac{\sqrt{Q} S_T}{r} \left(\langle h_B \rangle_{\widetilde{P}} - r h_J \right). \tag{5}$$

The average is with respect to the distribution $\widetilde{P}(h_B | h_J, S_T)$, the conditional probability of the unknown h_B, given the student local potential and the noisy training label S_T.

To begin with, we will assume that the type and strength of the noise process are known to the student. This information can therefore be used and determines the specific form of f_{opt}. Inserting f_{opt} into (4) and solving the resulting differential equations will then yield the optimal on-line learning curve $\epsilon_g(\alpha)$ that can be achieved by an algorithm of the form (3).

a. Output noise:
In this first scenario the rule outputs are inverted independently with a probability $\lambda \leq 1/2$. Thus, only a fraction $(1 - \lambda)$ of the examples provides correct information about the rule. This type of noise was considered in [7] in the framework of off–line learning, and recently in [11,14] for on–line learning with a perceptron.

In this model, the random inputs enter only through the local potentials h_B and h_J. Their joint distribution is a twodimensional Gaussian with zero means, unit variances and correlation $\langle h_J h_B \rangle_\xi = r$. In addition one has to average explicitly over the randomness in S_T.

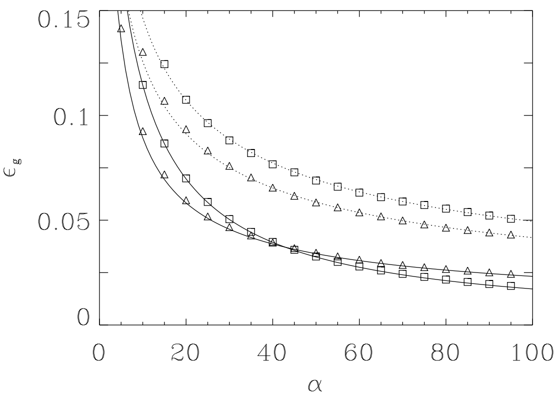

FIG. 1. Learning curves $\epsilon_g(\alpha)$ in the presence of output noise ($\square$) with $\lambda = 0.1$ and weight noise ($\triangle$) with $\omega = \cos(0.1\pi)$. The dotted lines are for simple Hebbian learning, the solid lines correspond to the respective optimal weight functions (7,9). The symbols represent the results of simulations for input dimension $N = 1000$, averaged over 100 independent runs. Error bars would be smaller than the symbol size.

The prediction error obeys the relation $\epsilon_p = \lambda + (1 - 2\lambda)\epsilon_g \geq \lambda$. As explained above it is bounded from below due to the randomness of S_T.

For Hebbian learning ($f = 1$) the differential equations are analytically solvable and we obtain

$$\epsilon_g(\alpha) = \frac{1}{\pi} \arccos \left[\left(1 + \frac{\pi}{2(1 - 2\lambda)^2 \alpha} \right)^{-1/2} \right]. \tag{6}$$

The rule is learnt perfectly as $\alpha \to \infty$ and the asymptotic generalization error decays like $\alpha^{-1/2}$ as for noise free learning [13] but with a modified prefactor. It diverges for $\lambda \to 1/2$, which corresponds to training labels completely uncorrelated with the rule. Figure 1 shows the learning curve $\epsilon_g(\alpha)$ for $\lambda = 0.1$ together with the results of numerical simulations.

For the optimized weight function (5) we obtain

$$f_{opt} = \sqrt{Q} \frac{1 - 2\lambda}{\sqrt{2\pi} r} \sqrt{1 - r^2} \frac{\exp\left(-\frac{1}{2} \frac{r^2}{1 - r^2} h_J^2 \right)}{(1 - 2\lambda)\, \Phi\left(\frac{r}{\sqrt{1 - r^2}} h_J S_T \right) + \lambda} \tag{7}$$

where $\Phi(x) = (1 + \mathrm{erf}(x/\sqrt{2}))/2$. Note that in the noiseless case ($\lambda = 0$) the optimal weight function reduces to the result previously obtained in [9].

Inserting f_{opt} into the Eqs. (4) and performing the averages over h_B, h_J and S_T one obtains decoupled differential equations for r and Q which can be integrated numerically. The resulting generalization error is shown in Fig.1 for $\lambda = 0.1$. For small values of ϵ_g we analytically obtain the asymptotic solution

$$\epsilon_g(\alpha) = \frac{2}{(1-2\lambda)^2}\left[\int_{-\infty}^{\infty}\frac{dx}{\sqrt{2\pi}}\frac{e^{-x^2}}{(1-2\lambda)\,\Phi(x)+\lambda}\right]^{-1}\alpha^{-1} \quad (8)$$

as $\alpha \to \infty$. As in the noiseless case the optimal decay of the generalization error is inversely proportional to the number of presented examples. This is in contrast to the results of [11] for the standard perceptron algorithm [1], where the decay is only proportional to $\alpha^{-1/2}$ for any nonzero λ.

The prefactor in (8) diverges for $\lambda \to 1/2$, reflecting the fact that the rule cannot be learnt in this limit. In the noiseless case we find $\epsilon_g \approx 0.88/\alpha$ for large α in accordance with the results of [9].

Note that the asymptotic decay of the generalization error (8) differs from the replica–symmetric result for off-line Bayes optimal learning [7] by exactly a factor of 2 (and thus by a factor $\sqrt{2}$ from the off–line Gibbs procedure considered in [5,7]). This interesting relation was found in [9] for noise free training where the replica–symmetric result is believed to be exact. Apparently it persists for the learning from corrupted data.

b. Weight noise:
In the second scenario, the actual weights used to evaluate the training output S_T^μ are subject to random fluctuations. We take $S_T^\mu = \mathrm{sign}(\tilde{\mathbf{B}}^\mu \cdot \boldsymbol{\xi}^\mu)$ where $\tilde{\mathbf{B}}^\mu$ is a normalized random vector with $\tilde{\mathbf{B}}^\mu \cdot \mathbf{B} = \omega \leq 1$. The noise is assumed to be isotropic, i.e. independent and identically distributed in all components B_j. So far, weight noise has been investigated only in the context of off–line learning (e.g. [5,6]).

Note that this scenario is identical to introducing an equivalent noise in the inputs used for the evaluation of S_T, but with $\mathbf{B}$ unchanged. Furthermore, the effect will be the same if this noise is imposed on the student input vectors in (3), with S_T being the true rule output for the true $\boldsymbol{\xi}$.

The average over this type of noise can be performed by introducing the additional Gaussian variable $\widetilde{h_B} = \tilde{\mathbf{B}} \cdot \boldsymbol{\xi}$ with zero mean, unit variance and correlations $\left\langle \widetilde{h_B} h_J \right\rangle_\xi = \omega r$, $\left\langle \widetilde{h_B} h_B \right\rangle_\xi = \omega$. All averages in Eq. (4) reduce to integrals over the corresponding three-dimensional density $P(h_J, h_B, \widetilde{h_B})$ with $S_T = \mathrm{sign}(\widetilde{h_B})$.

Note that the relation between generalization and prediction error is now $\epsilon_p = (1/\pi)\arccos\left(\omega\cos(\pi\epsilon_g)\right)$ implying that ϵ_p is bounded from below by $(1/\pi)\arccos\omega$.

For Hebbian Learning ($f = 1$) the result coincides with Eq. (6) replacing λ with $(1-\omega)/2$. The corresponding $\epsilon_g(\alpha)$ in Fig. 1 can therefore be interpreted as the learning curve for Hebbian learning with weight noise parameter $\omega = 1-2\lambda$. Note, however, that the respective prediction errors are not identical.

It is more instructive to compare the models for the same asymptotic value of ϵ_p, i.e. for $\lambda = (1/\pi)\arccos\omega$. Then, the expected number of corrupted outputs S_T is the same in both scenarios. Yet, weight noise will produce a label $S_T = -S_B$ with a probability which depends on the value of h_B. Mainly examples with $\boldsymbol{\xi}$ close to the decision boundary $h_B = 0$ will be affected, whereas the output noise was defined to be independent of h_B. It is straightforward to show that at the same rate of inverted outputs, the Hebbian generalization error for output noise is always larger than in the presence of weight noise (cf. Fig. 1).

Proceeding as before one obtains the optimal weight function (5)

$$f_{opt} = \sqrt{\frac{Q}{2\pi}}\frac{\omega(1-r^2)}{r\sqrt{1-\omega^2 r^2}}\frac{\exp\left(-\frac{1}{2}\frac{\omega^2 r^2}{1-\omega^2 r^2}h_J^2\right)}{\Phi\left(\frac{\omega r}{\sqrt{1-\omega^2 r^2}}h_J S_T\right)}. \quad (9)$$

Again, it is assumed that the actual noise parameter ω is known to the student and the optimal weight function reduces to the result of [9] for the noiseless case $\omega = 0$.

By solving the corresponding differential equations (4) numerically one obtains the generalization error $\epsilon_g(\alpha)$. An asymptotic solution for small ϵ_g yields for $\alpha \to \infty$ the analytic result

$$\epsilon_g(\alpha) = \frac{(1-m^2)^{1/4}}{(m\pi)^{1/2}}\left[\int_{-\infty}^{\infty}\frac{dx}{\sqrt{2\pi}}\frac{\exp\left(-\frac{1}{2}\frac{1+m^2}{m^2}x^2\right)}{\Phi(x)}\right]^{-1/2}\alpha^{-1/2}. \quad (10)$$

As for output noise, this result differs only by a prefactor from the one obtained for an off–line Gibbs algorithm with a properly defined *training energy* [5].

The generalization error decreases substantially slower in the case of noisy teacher weights than for random inversion of the output labels. Fig. 1 depicts the numerically obtained solution of the generalization error for $\omega = \cos(0.1\pi)$. This value of ω leads to an asymptotic prediction error $\epsilon_p^\infty = 0.1$ and therefore allows for a direct comparison with the results obtained for output noise with $\lambda = \epsilon_p^\infty = 0.1$ (cf. Fig. 1). For small α the generalization error decreases considerably faster in the case of weight noise than for output noise, and only for large α the decay is slower according to the different power law.

This behavior can be understood as follows: For weight noise only those input vectors are corrupted that are close to the decision boundary of the noiseless rule vector $\mathbf{B}$. Therefore the effect of the noise is not very pronounced for small α, where the student's decision boundary is still far away from the one of the teacher. In contrast, output noise inverts all inputs with the same probability, regardless of their overlap with student or teacher. Hence, learning in the case of output noise leads to a larger ϵ_g in the beginning compared to weight noise.

c. Realization of the optimal generalization

The optimal weight functions (7,9) depend both on $r = \cos(\pi\epsilon_g)$ and the noise parameters λ or ω, respectively, which will not be accessible to the student network in general. Nevertheless, by construction of f_{opt}, there is no algorithm of the form (3) that could yield a smaller value of the generalization error for a given α.

In the following we briefly show how to make the weight function independent of the unaccessible quantities without changing the asymptotic behavior of the generalization error. Here we restrict ourselves to the case of output noise, but the proposed scheme also applies to the weight noise scenario [16].

In order to circumvent the dependence of f_{opt} on r we investigate the dynamics of Q using the weight function f_{opt}. By choosing the initial condition $\sqrt{Q(0)} = r(0)$ one guarantees $\sqrt{Q(\alpha)} \equiv r(\alpha)$ as already observed in [15]. Therefore the solution for $\epsilon_g(\alpha)$ does not change if r is replaced by $\sqrt{Q}$ in (7). However, Q is the norm of the weight vector $\mathbf{J}$ and of course available to the student network.

The other peculiarity of the optimal weight function is its explicit dependence on the probability λ. One could try to replace λ with a constant estimate Λ in Eq. (7). The resulting weight function, however, does not allow for a perfect generalization for all Λ [16]. There exists a critical value $\Lambda_c(\lambda)$ such that $\epsilon_g \propto 1/\alpha$ for $\Lambda > \Lambda_c$, but ϵ_g remains finite for $\Lambda < \Lambda_c < \lambda$. Of course, only for the choice $\Lambda = \lambda$ the generalization error decreases optimally as in (8).

For an unknown noise parameter λ we therefore suggest an on–line adaptation of the parameter Λ. To this end we define a simple dynamics for Λ such that it tends to λ asymptotically as desired. As one specific example we change Λ by $(1-\Lambda)/2N$ every time the student disagrees with the noisy output of the teacher. In the case of agreement Λ is changed by $-\Lambda/2N$:

$$\Lambda^{\mu+1} = \Lambda^{\mu} + \frac{1}{2N}\left[(1-\Lambda^{\mu})\,\Theta(-h_J^{\mu}S_T^{\mu}) - \Lambda^{\mu}\Theta(h_J^{\mu}S_T^{\mu})\right]. \tag{11}$$

In the limit of large N this leads to the differential equation $d\Lambda/d\alpha = (\epsilon_p - \Lambda)/2$. Now the system is described by a set of three coupled differential equations for the dynamical variables r, Q, and Λ. By construction Λ approaches λ and the resulting generalization error asymptotically becomes identical to the optimal solution. Therefore the weight function (7) with the replacements $r \to \sqrt{Q}$ and $\lambda \to \Lambda$ from Eq. (11) provides an algorithm that realizes the optimal $1/\alpha$ decrease of the generalization error without requiring knowledge of ϵ_g or the noise parameter λ.

Finally, we illustrate the ability of this on–line algorithm to adapt to a changing noise level λ, see Fig. 2. The student network rapidly adapts to the noise and the generalization error decays proportional to $1/\alpha$. The dynamics of Λ can be further improved by replacing (11) with a schedule that approaches the asymptotic value λ even faster [16].

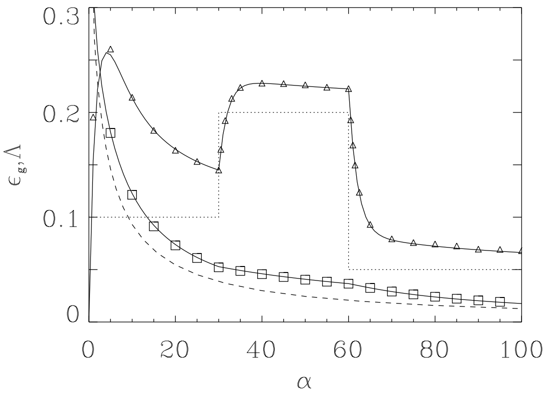

FIG. 2. Learning with variable output noise parameter λ (dotted). Shown is the learning curve ϵ_g ($\Box$) and the evolution of Λ ($\triangle$) according to (11) with $\Lambda(0) = 0$. The dashed line depicts the optimal ϵ_g as obtained for a constant noise level $\lambda = 0.05$ for comparison. Simulations as in Fig. 1.

In summary we have shown that for on–line learning from noisy training data the optimal learning curves can be calculated exactly. For output noise the asymptotic decay of the generalization error differs from the result for noiseless training only by a prefactor. Only for weight noise the generalization error is considerably worse. Since the optimal weight functions require knowledge about the noise we have proposed to introduce a dynamical quantity which adapts to the teacher's noise level.

Obviously, the two types of corrupted data require rather different learning strategies. A more complete discussion of the weight functions (7,9) will be published in [16], together with the analysis of other training schemes.

The authors acknowledge fruitful discussions with W. Kinzel, M. Opper, and G. Reents. P.R. was supported by the Deutsche Forschungsgemeinschaft. Simulations were done on the Cray–YMP EL of the Rechenzentrum der Universität Würzburg.

[1] J.A. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation* (Addison–Wesley, Redwood City, CA, 1991).

[2] M. Mézard, G. Parisi, and M.A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).

[3] S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992).

[4] T.L.H. Watkin, A. Rau, and M. Biehl, Rev. Mod. Phys. **65**, 499 (1993).

[5] M. Opper and W. Kinzel, in *Physics of Neural Networks*

III, series editors E. Domany, J.L. van Hemmen, and K. Schulten (Springer, Berlin, in press).

[6] G. Györgyi and N. Tishby, in *Neural Networks and Spin Glasses*, edited by W.K. Theumann and R. Köberle, (World Scientific, Singapore, 1990).
G. Györgyi, Phys. Rev. Lett. **64**, 2957 (1990).

[7] M. Opper, and D. Haussler, Phys. Rev. Lett. **66**, 2677 (1991), and in *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, edited by L.G. Valiant and M.K. Warmuth, (Morgan Kaufmann, San Mateo, CA, 1991).

[8] W. Kinzel and P. Rujмасán, Europhys. Lett. **13**, 473 (1990).

[9] O. Kinouchi and N. Caticha, J. Phys. A **25**, 6243 (1992).

[10] M. Biehl and H. Schwarze, J. Phys. A **26**, 2651 (1993).

[11] N. Barkai, H.S. Seung, and H. Sompolinsky, *On–line Learning of Dichotomies*, preprint 1994.

[12] M. Biehl and P. Riegler, Europhys. Lett. **28**, 525 (1994).

[13] F. Vallet, Europhys. Lett. **9**, 315 (1989).

[14] T. Heskes, in *Proceedings of the ZiF Conference on Adaptive Behavior and Learning*, (Bielefeld, 1994).

[15] M. Copelli and N. Caticha, J. Phys. A **28**, 1615 (1995).

[16] M. Biehl, P. Riegler, and M. Stechert, in preparation.