Presentation order and on-line learning

Tom Heskes and Wim Wiegerinck

RWCP[1] Novel Functions SNN[2] Laboratory
Department of Medical Physics and Biophysics
University of Nijmegen, Geert Grooteplein 21
6525 EZ Nijmegen, The Netherlands

Submitted to ICANN '95

Abstract

We study the effect of the presentation order of training patterns on the performance of on-line learning neural networks. In the context of time series, we discuss the difference between randomized and natural learning. With regard to learning in cycles, we quantify and compare the performance of almost cyclic and purely cyclic learning.

1 Introduction

Learning plays a crucial role in most neural-network applications. Through learning the network weights are adapted to meet the requirements of the environment. Usually, the designer has access to a finite number of examples from this environment: the training set. A popular learning strategy is on-line learning: at each learning step one of the patterns is drawn from the training set and presented to the network leading to a learning step of the form

$$\Delta w(n) \;=\; w(n+1) - w(n) \;=\; \eta f(w(n), x(n)) \,, \tag{1}$$

with $w(n)$ the weight vector at iteration step n, η the learning parameter, $x(n)$ the presented training pattern, and $f(\cdot,\cdot)$ the learning rule which may be supervised or unsupervised. The larger and the more redundant the amount of information in the training set, the more attractive on-line learning rules if compared with (more advanced) batch-mode learning rules (see e.g. [1]).

Pattern-by-pattern presentation introduces stochasticity in the learning process that can be studied using techniques from theory on stochastic processes (see e.g. [2, 3]). These studies have been mainly focussed on "memory-less" on-line learning or "randomized learning", that is, at each learning step an example is drawn *at random* from the training set, without any reference to previous presentations. In this paper we will extend the theory to "correlated learning", i.e. learning strategies for which successive examples may be correlated. An appealing example is the learning of stationary time series. We will compare "randomized learning", where examples are presented in some random order unrelated to their order of appearance in the time series, and "natural learning", where the presentation order follows the time series. A strategy yielding artificial correlations is "cyclic learning". Cyclic learning ensures that in each cycle all patterns are presented exactly once. With "purely cyclic learning" examples are presented in the same order over and over again. With "almost cyclic learning" we randomize over the presentation order of the examples in each cycle. Due to space limits we will in this paper only point out the most important results (see [7, 4, 8] for more details).

[1]Real World Computing Program
[2]Dutch Foundation for Neural Networks

2 Theoretical framework

Our analysis combines the time-averaging procedure proposed in [2] with Van Kampen's expansion [3]. After M iterations of the learning step (1) we have

$$\bar{w}(n+M) - \bar{w}(n) \;=\; \eta \sum_{m=0}^{M-1} \langle f(w(n+m), x(n+m)) \rangle \;,$$

where $\langle \cdot \rangle$ and $\bar{w}$ refer to averages over both the pattern dynamics and the weight dynamics. For small learning parameters the weight dynamics is a factor η slower than the pattern dynamics. If we choose M much smaller than the typical time scale of the weight dynamics, we can approximate

$$\bar{w}(n+M) - \bar{w}(n) \;=\; \eta \sum_{m=0}^{M-1} \langle f(w(n), x(n+m)) \rangle \;+\; \ldots \;.$$

Choosing M much larger than the time scale of the pattern dynamics, we have

$$\frac{1}{M} \sum_{m=0}^{M-1} \langle f(w(n), x(n+m)) \rangle \;\approx\; \lim_{N \to \infty} \frac{1}{N} \sum_{m=0}^{N-1} \langle f(w(n), x(n+m)) \rangle \;\equiv\; F(w(n)) \;,$$

with $F(w)$ the "batch-mode learning rule" averaged over all possible training patterns. If we now transform the resulting difference equation for $\bar{w}(n)$ into a differential equation, we obtain in lowest order the "mean-field equation"

$$\frac{d\bar{w}(t)}{dt} \;=\; F(\bar{w}(t)) \;+\; \ldots \;, \tag{2}$$

with rescaled time $t \equiv \eta n$. We will assume that the average learning rule $F(w)$ is the gradient of some error potential $E(w)$. In this case we can interpret the learning rule (1) as a stochastic gradient descent on this error potential. Equation (2) then simply states that in lowest order the average network state follows the deterministic gradient descent on the same error potential. This result is completely independent of the presentation order: all dependence on the presentation order "washes out" for small learning parameters.

In a similar way we can compute the evolution of the variance

$$\Sigma(t) \;\equiv\; \left\langle [w(t) - \bar{w}(t)]\, [w(t) - \bar{w}(t)]^T \right\rangle \;.$$

After tedious calculations we obtain [8]

$$\frac{d\Sigma(t)}{dt} \;=\; -H(\bar{w}(t))\Sigma(t) - \Sigma(t)H(\bar{w}(t)) + \eta D(\bar{w}(t)) \;, \tag{3}$$

with the Hessian and diffusion matrix, respectively,

$$\begin{aligned} H(w) \;&=\; -\nabla_w^T F(w) \;=\; \nabla_w \nabla_w^T E(w) \\ D(w) \;&=\; C_0(w) + \lim_{N \to \infty} \sum_{n=1}^{N-1} \left[1 - \frac{n}{N}\right] \left[C_n(w) + C_n^T(w)\right] \equiv C_0(w) + C_+(w) \;, \end{aligned}$$

where the auto-correlation matrices are defined by

$$C_n(w) \;\equiv\; \left\langle [f(w, x(n)) - F(w)]\; [f(w, x(0)) - F(w)]^T \right\rangle \;.$$

The Hessian matrix is the local curvature of the error potential, the diffusion matrix reflects the fluctuations in the learning rule. For randomized learning $C_+(w) = 0$ and thus $D(w) = C_0(w)$ (see e.g. [3, 2]). Correlations do indeed affect the variance of the weights. If subsequent weight changes are negatively or positively correlated, the fluctuations will be smaller or larger, respectively.

3 On-line learning of stationary time series

Let us consider a network learning a chaotic times series of the form

$$x(n+1) = y(x(n)) \, ,$$

i.e. the next input is a function of the current input. Input-output combinations $\{x(n), x(n+1)\}$ are presented to a feedforward neural network. Using the squared distance between the network output $g(w, x)$ and the desired output $y(x)$ as an error criterion $e(w, x)$, the on-line learning rule is

$$\Delta w(n) \;=\; -\nabla_w \frac{1}{2} \left[g(w(n), x(n)) - g(x(n)) \right]^2 \;\equiv\; -\nabla_w e(w(n), x(n)) \, .$$

First we will discuss two different measures for the asymptotic performance of a network learning such a time series. The "representation error"

$$E_{\rm repr} \;\equiv\; \lim_{n\to\infty} \langle \langle e(w(n), x) \rangle_x \rangle \;=\; \lim_{n\to\infty} \langle E(w(n) \rangle$$

measures how well the asymptotic network state is expected to represent the *whole* time series. The "prediction error" is the expected asymptotic network error on the *next* pattern in the time series:

$$E_{\rm pred} \;\equiv\; \lim_{n\to\infty} \langle e(w(n), x(n)) \rangle$$

$E_{\rm pred}$ measures the performance locally in time, in contrast to the more global measure $E_{\rm repr}$. In the asymptotic situation the weights are assumed to be near a minimum w^* of the error $E(w)$. A local expansion around w^* then yields [8]:

$$E_{\rm repr} \;=\; E(w^*) + \frac{\eta}{4} \, {\rm Tr} \left[D(w^*) \right] \quad \text{and} \quad E_{\rm pred} \;=\; E_{\rm repr} - \frac{\eta}{2} \, {\rm Tr} \left[C_+(w^*) \right] \, .$$

For randomized learning the prediction and representation error are equal: $E_{\rm pred} = E_{\rm repr} \equiv E_{\rm ran}$. Combining these results we conclude that in lowest order the profit in prediction cancels the loss in representation and vice versa:

$$\frac{E_{\rm pred} + E_{\rm repr}}{2} \;=\; E_{\rm ran} + \ldots \, .$$

In the context of strategies to select examples (see e.g. [6]) this result implies that a strategy yielding a smaller prediction error will most likely lead to a larger representation error.

The difference between the representation and the prediction error is a local effect. However, in practice one sometimes observes dramatic global differences between natural and randomized learning (see e.g. [5]). As an illustrative example we consider a backprop network with one hidden layer of two units,

$$g(w, x) \;=\; v_0 + \sum_{\alpha=1}^{2} v_\alpha \tanh(w_{\alpha 1} x + w_{\alpha 0}) \, ,$$

trying to approximate the tent map

$$y(x) \;=\; 1 - |2x - 1| \quad \text{for} \quad 0 \le x \le 1 \, .$$

We train this network in two different ways: naturally following the order of the time series and randomly. Starting with small random weights, the network soon gets stuck on a plateau with all weights close to zero, except for the output bias $v_0 \approx 0.5$. As can be seen in Figure 1(a), the network function $g(w_{\rm plateau}, x)$ is a constant. On this plateau the average learning rule $F(w_{\rm plateau}) = 0$, while the Hessian $H(w_{\rm plateau})$ has one positive eigenvalue and further only zero eigenvalues. In other words, on a plateau the lowest orders in the mean-field approximation of

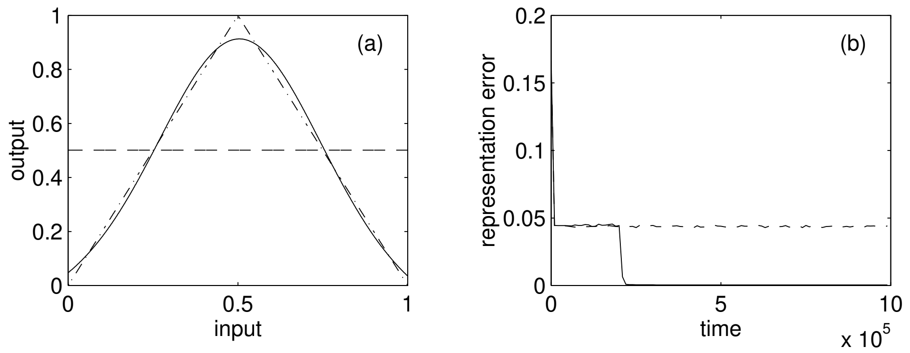

Figure 1: On-line learning of the tent map: randomized learning gets stuck on a plateau whereas natural learning reaches the optimal solution. (a) Tent map (dash-dotted), network solution on the plateau (dashed), and network solution with minimal error (solid). (b) Representation error as a function of the number of learning steps for randomized learning (dashed) and natural learning (solid). Simulations with learning parameter $\eta = 0.1$.

section 2 vanish and we have to systematically include higher order terms. Doing so, we find that for natural learning the now leading order slowly "drives" the weights from the plateau [7]. No such drive exists for randomized learning, explaining the dramatic difference between randomized and natural learning pictured in Figure 1(b). Therefore, if a particular learning problem suffers from a plateau, learning with correlated examples is strongly recommended, since it might help driving the weights off the plateau towards a better solution.

4 Cyclic learning

In principle we can apply the same machinery as before to study cyclic learning. However, it appears that the diffusion matrix $D(w)$ vanishes: the matrices $C_0(w)$ and $C_+(w)$ exactly cancel. Again we have no choice but to compute higher order terms. Let us denote a particular cycle containing each pattern exactly once by $\vec{x} \equiv \{x_0, \ldots, x_i, \ldots, x_{P-1}\}$. The tric is now to derive the learning rule for a whole cycle from the learning rules for the individual patterns:

$$w(n+P) - w(n) \;=\; \eta \sum_{i=0}^{P-1} f(w(n+i), x_i) \;.$$

For small learning parameters η the ansatz $w = \omega + (\eta P)\zeta$ yields [4]

$$\begin{aligned} \frac{d\omega(t)}{dt} &= F(\omega(t)) \\ \zeta(m+1) - \zeta(m) &= \eta P \left[b(\omega(t), \vec{x}(m)) - H(\omega(t))\zeta(m) \right] \;+\; \ldots \;, \end{aligned} \tag{4}$$

where $t \equiv \eta n$ as before, $m \equiv nP$ the time measured in cycles, $\vec{x}(m)$ the particular cycle presented at cycle step m, and $b(\cdot,\cdot)$ measuring the correlations between the learning rule $f(\cdot,\cdot)$ and its derivative $-h(\cdot,\cdot)$:

$$b(w, \vec{x}) \;\equiv\; -\frac{1}{P^2} \sum_{i=1}^{P-1} \sum_{j=1}^{i-1} h(w, x_i) f(w, x_j) + \frac{1}{2} H(w) F(w) \;.$$

Equation (4) can be viewed as the learning rule for *cycle* $\vec{x}(m)$ with all specific information about the particular cycle $\vec{x}(m)$ contained in the term $b(w, \vec{x}(m))$.

In comparing almost cyclic and purely cyclic learning for $1 \ll P \ll 1/\eta$, we will focus on the asymptotic misadjustment close to a minimum w^* defined as

$$A \equiv \left\langle (w - w^*)(w - w^*)^T \right\rangle = (\bar{w} - w^*)(\bar{w} - w^*)^T + \left\langle (w - \bar{w})(w - \bar{w})^T \right\rangle \equiv \Delta + \Sigma \,,$$

with Δ the so-called bias and Σ the variance. We will restrict ourselves to feedforward networks for function approximation where $D(w^*) \approx 2E(w^*)H(w^*)$. The asymptotic variance for on-line learning follows from the asymptotic solution of (3) and is an order of magnitude larger than the bias [3]:

$$A_{\text{online}} = \Sigma_{\text{online}} = \frac{\eta E(w^*)}{2} \,.$$

With purely cyclic learning a particular cycle $\vec{x}$ is drawn at random from the set of $P!$ possible cycles and then kept fixed. The asymptotic misadjustment is a "quenched" average of the bias over all possible cycles. We obtain [4]

$$A_{\text{purely}} = \frac{\eta^2 P E(w^*)}{6} H(w^*) \,.$$

The misadjustment is proportional to $\eta^2 P$, i.e. one order of magnitude smaller than for on-line learning. Almost cyclic learning is randomized learning with training cycles instead of training patterns. In lowest order we find [4]:

$$\Delta_{\text{almost}} = \frac{\eta^2}{2} H(w^*)B(w^*)B^T(w^*)H(w^*) \quad \text{and} \quad \Sigma_{\text{almost}} = \frac{\eta^3 P^2 E(w^*)}{12} H^2(w^*) \,,$$

with $B(w) \equiv 2P \left\langle b(w, \vec{x}) \right\rangle_{\vec{x}}$. Depending on the term ηP^2 the asymptotic misadjustment is dominated by either the bias of order η^2 or the variance of order $\eta^3 P^2$. In any case, the asymptotic misadjustment for almost cyclic learning is an order of magnitude smaller than for purely cyclic learning.

5 Conclusions

We have shown how theory on stochastic processes can be applied to study how the presentation order affects on-line learning. We summarize our main results.

- Close to a minimum, the profit in representation performance cancels the loss in prediction performance and vice versa.

- On a plateau natural learning yields a small nonzero driving force that helps the weights to find their way towards a better solution, where randomized learning seems to get stuck forever.

- Asymptotic fluctuations for cyclic learning are an order of magnitude smaller than for randomized learning; almost cyclic learning being even better than purely cyclic learning.

References

[1] E. Barnard and J. Holm. A comparative study of optimization techniques for backpropagation. *Neurocomputing*, 6:19–30, 1994.

[2] L. Hansen, R. Pathria, and P. Salamon. Stochastic dynamics of supervised learning. *Journal of Physics A*, 26:63–71, 1993.

[3] T. Heskes and B. Kappen. On-line learning processes in artificial neural networks. In J. Taylor, editor, *Mathematical Foundations of Neural Networks*, pages 199–233. Elsevier, Amsterdam, 1993.

[4] T. Heskes and W. Wiegerinck. A theoretical comparison of batch-mode, on-line, cyclic, and almost cyclic learning. *Submitted to IEEE Transactions on Neural Networks*, 1994.

[5] G. Mpitsos and M. Burton. Convergence and divergence in neural networks: processing of chaos and biological analogy. *Neural Networks*, 5:605–625, 1992.

[6] P. Munro. Repeat until bored: a pattern selection strategy. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 1001–1008, San Mateo, 1992. Morgan Kaufmann.

[7] W. Wiegerinck and T. Heskes. On-line learning with time-correlated patterns. *Europhysics Letters*, 28:451–455, 1994.

[8] W. Wiegerinck and T. Heskes. How correlations between successive examples affect on-line learning. *Submitted to Neural Computation*, 1995.