Optimal online learning: a Bayesian approach

Sara A. Solla [a,b,1] and Ole Winther [c,1]

[a] *Department of Physics and Astronomy, Northwestern University, Evanston, IL 60208, USA*

[b] *Department of Physiology, Northwestern University Medical School, Chicago, IL 60611, USA*

[c] *Theoretical Physics II, Lund University, S-223 62 Lund, Sweden*

Abstract

A recently proposed Bayesian approach to online learning is applied to learning a rule defined as a noisy single layer perceptron. In the Bayesian online approach, the exact posterior distribution is approximated by a simple parametric posterior that is updated online as new examples are incorporated to the dataset. In the case of binary weights, the approximate posterior is chosen to be a biased binary distribution. The resulting online algorithm is shown to outperform several other online approaches to this problem.

Neural networks are adaptive systems characterized by a set of parameters $\mathbf{w}$, the weights and biases that specify the connectivity among the *neuronal* computational elements. Consider the case of multilayer architectures, which implement deterministic maps from an input space $\{\mathbf{x}\}$ onto an output space $\{\mathbf{y}\}$. For a fixed architecture, the properties of the specific input-output map $\mathbf{y} = f_{\mathbf{w}}(\mathbf{x})$ implemented by the network depend only on the values selected for the network parameters $\mathbf{w}$. Of particular interest is the ability of these systems to learn from examples. In the *supervised learning* scenario investigated here, each example is in the form of an input-output pair $(\mathbf{x}, \mathbf{y})$. A training set of size t is denoted by $D_t = \{(\mathbf{x}^\mu, \mathbf{y}^\mu), 1 \leq \mu \leq t\}$; every example is an input-output pair labeled by the integer counting index μ.

Traditional formulations of the learning problem are based on a dynamical prescription for the adaptation of the parameters $\mathbf{w}$. The learning process

[1] We acknowledge the support of CONNECT, the Computational Neural Network Center at the Niels Bohr Institute, Copenhagen, Denmark; CONNECT provided the hospitable environment where most of the discussions that led to this work took place.

generates a trajectory in $\{\mathbf{w}\}$ space that starts from a random initial assignment $\mathbf{w}^0$ and leads to a specific $\mathbf{w}^\star$ that is in some sense *optimal*. Two learning modalities need to be distinguished: *offline* and *online*. In offline or batch learning, all examples in the training set are used at every time step to update the current values of the network parameters $\mathbf{w}$. In online learning, the parameters are updated after the presentation of each example: $\mathbf{w}^{\mu+1} = \mathbf{w}^{\mu} + \Delta\mathbf{w}^{\mu}$, where the update $\Delta\mathbf{w}^{\mu}$ is only a function of the current position $\mathbf{w}^{\mu}$ and the new example: $\mathbf{w}^{\mu+1} = F(\mathbf{w}^{\mu}, (\mathbf{x}^{\mu+1}, \mathbf{y}^{\mu+1}))$.

Algorithmic approaches that result in specific values $\mathbf{w}^\star$ for the network parameters are to be contrasted to a probabilistic Bayesian formulation based on the information provided by a probability distribution over the parameter space $\{\mathbf{w}\}$. In the Bayesian approach, the learning process is described by the evolution of a prior distribution $p(\mathbf{w})$ into a posterior that incorporates the information provided by the data. The posterior, constructed as the appropriately normalized product of the prior and the likelihood of the data, quantifies the a posteriori belief in each possible setting of the parameters $\{\mathbf{w}\}$, and it plays a crucial role in the prediction of new data.

In the Bayesian framework, training examples $(\mathbf{x}^{\mu}, \mathbf{y}^{\mu})$ are assumed to be independently drawn from a distribution $p((\mathbf{x}, \mathbf{y})|\mathbf{w})$, where $\mathbf{w}$ is the unknown parameter vector to be estimated from the data D_t. The statistical independence of the individual examples results in a multiplicative form for the likelihood of the training set,

$$p(D_t|\mathbf{w}) = \prod_{\mu=1}^{t} p((\mathbf{x}^{\mu}, \mathbf{y}^{\mu})|\mathbf{w}) \ . \tag{1}$$

We write $p((\mathbf{x}, \mathbf{y})|\mathbf{w}) = p(\mathbf{y}|\mathbf{x}, \mathbf{w})\, p(\mathbf{x})$, where $p(\mathbf{y}|\mathbf{x}, \mathbf{w})$ models the input-output relation, while the input distribution $p(\mathbf{x})$ is independent of $\mathbf{w}$. The prior probability $p(\mathbf{w})$ must assign nonzero probability to the *true value* of the parameters, as actually used in the generation of the data. It is in this sense that the analysis is restricted to the case of *realizable* learning scenarios.

Bayes rule provides a prescription for writing the posterior distribution $p(\mathbf{w}|D_t)$ in terms of the prior and the likelihood of the training set:

$$p(\mathbf{w}|D_t) = \frac{p(D_t|\mathbf{w})\ p(\mathbf{w})}{\int d\mathbf{w}\, p(D_t|\mathbf{w})\ p(\mathbf{w})} \ . \tag{2}$$

The denominator

$$p(D_t) = \int d\mathbf{w}\, p(D_t|\mathbf{w})\ p(\mathbf{w}) \tag{3}$$

provides a normalization constant for the posterior $p(\mathbf{w}|D_t)$ and measures the likelihood of the data given the model; $p(D_t)$ is the Bayesian *evidence* whose use has been proposed for model selection.

The posterior probability (2) plays a crucial role in the prediction of new data. Given a new input $\mathbf{x}$, the Bayesian *predictive probability* $p(\mathbf{y}|\mathbf{x}, D_t)$ is computed through a weighted average over the parameter space $\mathbf{w}$, and it is the posterior that assigns a weight to every possible parameter setting:

$$p(\mathbf{y}|\mathbf{x}, D_t) = \int d\mathbf{w}\ p(\mathbf{y}|\mathbf{x}, \mathbf{w})\ p(\mathbf{w}|D_t)\ . \tag{4}$$

Predictions based on the Bayes algorithm are guaranteed to minimize the average prediction error through the choice of output $\mathbf{y}$ which maximizes the predictive probability (4) for a given input $\mathbf{x}$. Predictions based on this approach are optimal in the sense of yielding the minimal average prediction error; the average is to be taken over all possible data sources within the family defined by the prior. The optimality of Bayesian predictors thus holds under the assumption that the prior beliefs are correct.

The procedure described above is intrinsically *offline*, as the computation of the posterior (2) requires knowledge of the entire training set. A prescription is needed for the online update of the posterior (2) when a new example is added to the training set: $D_t \to D_{t+1} = \{(\mathbf{x}^\mu, \mathbf{y}^\mu), 1 \leq \mu \leq t+1\}$. A controlled approximation is now introduced that leads to an online implementation of Bayesian learning [1,2]: the true posterior (2) is approximated by a simple parametric distribution $p(\mathbf{w}|\mathbf{A}_t)$, where $\mathbf{A}_t$ refers to the current values of a set of parameters $\{\mathbf{A}\}$ which characterize the distribution (e.g. the first two moments for a Gaussian $p(\mathbf{w}|\mathbf{A})$). The online Bayesian procedure refers to the update of the distribution parameters $\{\mathbf{A}\}$ as opposed to the network parameters $\{\mathbf{w}\}$: $\mathbf{A}_{t+1} = F(\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))$.

The resulting algorithm consists of two steps:

* Add an example – the current posterior is updated exactly according to Bayes rule:

$$p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1})) = \frac{p(\mathbf{y}^{t+1}|\mathbf{x}^{t+1}, \mathbf{w})\ p(\mathbf{w}|\mathbf{A}_t)}{\int d\mathbf{w}\, p(\mathbf{y}^{t+1}|\mathbf{x}^{t+1}, \mathbf{w})\ p(\mathbf{w}|\mathbf{A}_t)}\ . \tag{5}$$

* Approximate – the updated posterior is parametrized:

$$p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1})) \to p(\mathbf{w}|\mathbf{A}_{t+1})\ . \tag{6}$$

The procedure is illustrated in Fig. 1.

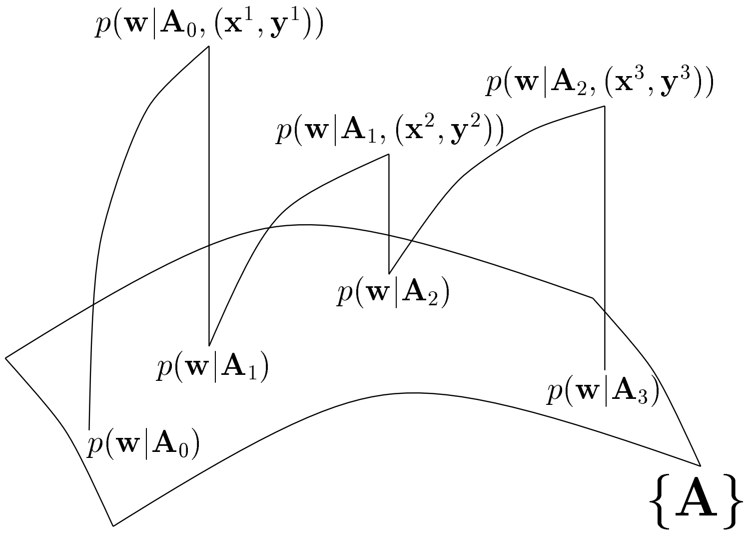

Fig. 1. The update of the approximate posterior.

Part of the information provided by the new example $(\mathbf{x}^{t+1}, \mathbf{y}^{t+1})$ is discarded in the parametrization step. In order to minimize this information loss, the new parameter values $\mathbf{A}_{t+1}$ are chosen so as to minimize the *Kullback-Leibler distance* between the two distributions,

$$D_{KL}\left[p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))||p(\mathbf{w}|\mathbf{A}_{t+1})\right]$$

$$= \int d\mathbf{w}\ p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))\ \ln \frac{p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))}{p(\mathbf{w}|\mathbf{A}_{t+1})}\ . \quad (7)$$

If the original network parameters are continuous, the posterior is approximated by a Gaussian $p(\mathbf{w}|\mathbf{A})$. The parameters $\{\mathbf{A}\}$ then refer to the mean $\langle \mathbf{w} \rangle$ and the covariance matrix $\mathbf{C} = \langle \mathbf{w}\,\mathbf{w}^T \rangle - \langle \mathbf{w} \rangle \langle \mathbf{w}^T \rangle$. The minimization of the distance (7) between the updated distribution and its parametrized form is achieved by choosing $\mathbf{A}_{t+1}$ such that $p(\mathbf{w}|\mathbf{A}_{t+1})$ and $p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))$ have the same first and second moments [1,3].

Of particular interest is the case of binary network parameters [2,4]. A prior distribution of the form

$$p(\mathbf{w}) = \prod_{i=1}^{N} \left[\frac{1}{2}\ \delta(w_i - 1) + \frac{1}{2}\ \delta(w_i + 1)\right] \quad (8)$$

evolves into a posterior that is parametrized as a biased binary distribution of the form

$$p(\mathbf{w}|\mathbf{A}) = \prod_{i=1}^{N} \left[\frac{1 + \langle w_i \rangle}{2}\ \delta(w_i - 1) + \frac{1 - \langle w_i \rangle}{2}\ \delta(w_i + 1)\right]\ . \quad (9)$$

The parameters $\{\mathbf{A}\}$ then refer to the components of the mean weight vector $\langle \mathbf{w} \rangle$. The minimization of the Kullback-Leibler distance (7) between the updated distribution and its parametrized form is achieved by matching the mean of $p(\mathbf{w}|\mathbf{A}_{t+1})$ to that of $p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))$:

$$\langle \mathbf{w} \rangle_{t+1} = \sum_{\{w_i = \pm 1\}} \mathbf{w} \; p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1})) \; . \tag{10}$$

We now apply this approach to the case of a simple perceptron, for which scalar classification labels $y = \pm 1$ are generated through $y = f(\mathbf{x}, \mathbf{w}) = \text{sign}(\mathbf{w} \cdot \mathbf{x})$. Both $\mathbf{w}$ and $\mathbf{x}$ are N-dimensional vectors with components $\{w_i \in \{-1, +1\}, x_i \in \Re, 1 \leq i \leq N\}$. Noise is introduced through a label flip with probability κ. The likelihood of output y is thus given by

$$\begin{aligned} p(y|\mathbf{x}, \mathbf{w}) &= \kappa \; \Theta(-y\, f(\mathbf{x}, \mathbf{w})) + (1-\kappa) \; \Theta(y\, f(\mathbf{x}, \mathbf{w})) \\ &= \kappa + (1 - 2\kappa) \; \Theta(y \; \mathbf{w} \cdot \mathbf{x}) \; , \end{aligned} \tag{11}$$

where $\Theta()$ is the step-function, $\Theta(u) = 1$ for $u > 0$ and $\Theta(u) = 0$ otherwise.

The update rule (10) cannot be expressed analytically in closed form for arbitrary N, but in the $N \to \infty$ limit the average over $p(\mathbf{w}|\mathbf{A}_t, (\mathbf{x}^{t+1}, \mathbf{y}^{t+1}))$ can be performed analytically through an unexpectedly successful if standard application of the cavity method [4]. The performance of this algorithm is shown in Fig. 2, where the generalization error ϵ is shown as a function of the normalized time variable $\alpha = t/N$ for $\kappa = 0$.

The Bayesian algorithm described here is compared to results obtained by three types of algorithms for the binary perceptron: clipping by taking the sign of the average weights obtained by this algorithm [4], clipping by taking the sign of the average weights obtained by a continuous weight algorithm [4], and the optimal binarization of continuous weights [5]. The small α behavior shows two groups: the two algorithms based on clipping average weights exhibit poorer performance, while the other two algorithms do quite well; the behavior of the latter group is likely to be described in this regime by pure Hebbian learning of the form $\mathbf{w} \propto \sum_\mu y^\mu \mathbf{x}^\mu$. As α increases, the performance of the algorithm based on an optimal binarization of the continuous weights solution deteriorates and approaches that of the algorithm based on a simple clipping of the continuous weights solution. The clipped version of the binary weights algorithm, which performs poorly in the small α regime, approaches the performance of the binary weights algorithm with increasing α; a result that indicates that in the large α regime the components of the average weight vector tend to ± 1 for the binary weights algorithm. As shown in Fig. 2, it is the binary weights algorithm based on (10) which is optimal among the various online algorithms considered here.

In summary, we have reviewed the Bayesian approach to online learning originally proposed by (Opper 1996) and generalized by (Winther & Solla 1997), and we have applied this approach to the analysis of learning a noisy binary perceptron. The current outstanding challenge is that of extending this approach to learning scenarios involving multilayer networks.

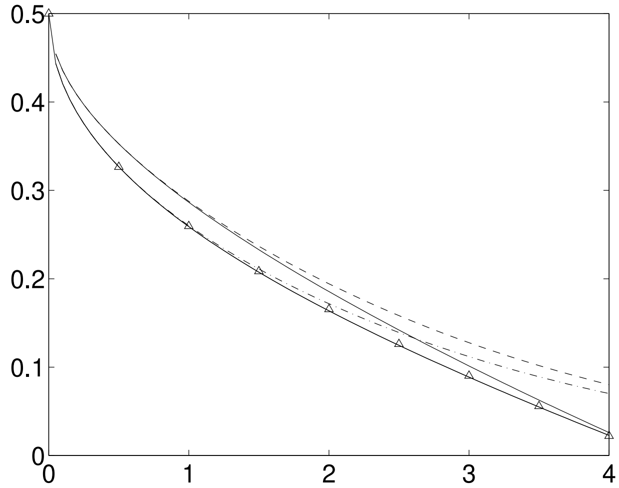

Fig. 2. Learning curves (ϵ versus α) for the binary simple perceptron at $\kappa = 0$. The full line with triangles shows the result of the Bayesian online binary weights algorithm. Simulation results (shown as triangles) were obtained for $N = 1000$ and averaged over 100 runs. The upper full line shows the result of taking the sign of the solution to the binary weights algorithm. The dashed line shows the result of taking the sign of the solution to the continuous weights algorithm. The dashed-dotted line is for the optimal binarization of the continuous weights solution.

References

[1] M. Opper, Online versus Offline Learning from Random Examples: General Results, *Phys. Rev. Lett.* **77** (1996) 4671.

[2] O. Winther and S. A. Solla, Optimal Bayesian Online Learning, in: K-Y. M. Wong, I. King and D-Y. Yeung, eds., *Theoretical Aspects of Neural Computation (TANC-97)* (Springer Verlag, Singapore, 1998).

[3] M. Opper, A Bayesian Approach to Online Learning, in: D. Saad ed., *On-Line Learning in Neural Networks* (Cambridge University Press, Cambridge, 1998).

[4] S. A. Solla and O. Winther, Optimal Perceptron Learning: an Online Bayesian Approach, in: D. Saad ed., *On-Line Learning in Neural Networks* (Cambridge University Press, Cambridge, 1998).

[5] J. Schietse, M. Bouten and C. Van den Broeck, Training Binary Perceptrons by Clipping, *Europhys. Lett.* **32** (1995) 279.