

Extrinsic Geometrical Methods for Neural Blind Deconvolution

Simone Fiori

*Dipartimento di Elettronica, Intelligenza Artificiale
e Telecomunicazioni, Università Politecnica delle Marche,
Via Breccie Bianche, I-60131 Ancona, Italy*

Abstract. The present contribution discusses a Riemannian-gradient-based algorithm and a projection-based learning algorithm over a curved parameter space for single-neuron learning. We consider the ‘blind deconvolution’ signal processing problem. The learning rule naturally arises from a criterion-function minimization over the unitary hyper-sphere setting. We consider the blind deconvolution performances of the two algorithms as well as their computational burden and numerical features.

Key Words: ‘Bussgang’-type blind deconvolution. Neural Bayesian estimation. Geodesic-based iteration. Projection-based iteration.

INTRODUCTION

Over the recent years, we have witnessed an increasing interest into application of geometrical methods to machine learning [1]. The key idea is that a network parameter space, either flat or curved, may be endowed with a specific geometric structure (in the sense of differential geometry), which is worth taking into account in the design of a network learning algorithm. A widely known example is the *natural gradient* theory. A family of multilayer perceptrons may be associated a parameter space that forms a geometrical neural manifold. Such a manifold does not possess a Euclidean geometrical structure, but a Riemannian structure. It is a common experience that the standard gradient learning algorithms, such as back-propagation, may be trapped or seriously slowed down by large plateaus located on the surface of the network error criterion during the learning processes. Recent studies have shown that learning algorithms based on natural gradient, which takes into account the geometrical structure of the neural manifold, do seem to be less affected by this difficulty [17].

The adoption of sophisticated mathematical instruments inherently brings incremental conceptual complexity into the classical learning theories and may bring additional computational/numerical burden into the related learning algorithms. With reference to the relationship between learning theories and learning algorithms, it is worth noting that the learning rules may arise from the optimization of a learning criterion via a suitable method, such as the gradient-based one, and are expressed as learning differential equations on the network parameter space. Such differential equations must then be discretized in the time domain in order to make them suitable for implementation on a computer. In the case of flat (Euclidean) parameter spaces, the discretization in the time

domain may be effected through classical numerical calculus techniques, such as the forward or backward Euler method or more sophisticated methods that aim at increasing the precision of the approximation. When dealing with curved parameter spaces, however, such methods are no longer suitable and discretization techniques developed in the mathematical field of geometrical numerical integration should be invoked, instead, [5, 11, 12]. In the present contribution, we consider a signal processing problem, namely *blind deconvolution*, which may be tackled via a single learning neuron model, whose learning strategy naturally arises as criterion-function minimization over a curved parameter space. Blind deconvolution [8, 15] is a statistical signal processing technique that aims at recovering a source signal distorted by the medium that it propagates within. Well-known engineering applications of blind deconvolution are equalization of communication channels [3], optomagnetic memory-support storage and retrieval enhancement [6], image deblurring [16] as well as geophysical measurements analysis [21]. An effective blind-deconvolution technique is known as ‘Bussgang’, which relies on the iterative Bayesian estimation of the source sequence, where the Bayesian estimator is matched to source statistics and to the model of the filter output signal [2]. Some modified Bussgang algorithms, based on neural-type approximate Bayesian estimators, have recently been proposed by the present author in [10].

The aim of the present contribution is to discuss a Riemannian-gradient-based algorithm and a projection-based algorithm for Bussgang-type blind deconvolution. The first algorithm is based on the key concept of discretizing a differential equation on manifold via suitably connected piece-wise geodesic arcs [9]. The second algorithm is based on the more familiar concept of embedding the parameter space into a larger Euclidean ambient space, which allows effecting learning steps as if the parameter space was flat, and to back-projecting the current network state into the curved parameter space through a suitable projection operator. We consider the blind deconvolution performances of the two algorithms as well as their computational requirements, in order to gain incremental knowledge on the benefits and drawbacks pertaining to both methods on a signal-processing application.

‘BUSSGANG’-TYPE LEARNING

The sampled or discrete-time system to deconvolve is described by the following input/output model:

$$x_n = \mathbf{h}^T \mathbf{s}_n + \nu_n, \quad (1)$$

where \mathbf{s}_n is the system’s input vector-stream at time $n \in 1..N \subset \mathcal{Z}$, namely $\mathbf{s}_n \stackrel{\text{def}}{=} [s_n \ s_{n-1} \ s_{n-2} \ \dots \ s_{n-L_h+1}]^T$, s_n denotes the sampled source signal and ν_n represents a zero-mean white measurement disturbance independent of the source signal. The constant L_h denotes the length of the system impulse response \mathbf{h} . The following minimal hypotheses about the system and data stream may be considered [4, 15]: The system’s impulse response satisfies $\mathbf{h}^T \mathbf{h} = 1$ and its inverse has finite energy; the system is time-invariant or slowly time-varying; the source signal s_n is a stationary, ergodic, independent identically distributed (IID) random process with mean

$\mathbb{E}_s[s_n] = 0$ and variance $\mathbb{E}_s[s_n^2] = 1$; also, the probability density function of the source signal is supposed to be symmetric around zero and non-Gaussian.

A filter described by the vector impulse response $\mathbf{w} = [w_0 \ w_2 \ w_3 \ \dots \ w_{L_w-1}]^T$ represents the approximate inverse of system (1) if filter \mathbf{w} approximately cancels the effects of channel \mathbf{h} on the source signal. Denoting with \mathbf{x}_n the vector containing the filter input samples at time $n \in 1..N \subset \mathcal{Z}$, namely $\mathbf{x}_n \stackrel{\text{def}}{=} [x_n \ x_{n-1} \ x_{n-2} \ \dots \ x_{n-L_w+1}]^T$, where the constant L_w denotes the length of the inverse filter impulse response \mathbf{w} , the output of the filter writes: $z_{m,n} = \mathbf{w}_m^T \mathbf{x}_n$, $m \in 1..M \subset \mathcal{Z}$. In this paper, we distinguish between the time-index n , which denotes sample time-ordering, and the learning-iteration index m , which denotes learning-iteration time-ordering. In on-line learning, it may hold $m = n$, while in batch-type learning, as it is the case in the present contribution, the two indices are independent.

In general, the deconvolution may only be approximate because of the possible presence of additive noise affecting the system's output measure and because a finite-impulse-response (FIR) filter cannot represent the inverse of the FIR system (1) (for more details, the Reader is addressed to [8, 10, 15]). Since \mathbf{h} and s_n are both unknown, the optimal filter \mathbf{w}_* such that $z_{*,n} \sim s_n$ has to be *blindly* identified, possibly by means of a neural algorithm. From the basic theory of blind deconvolution, it is known that the source signal may be recovered up to arbitrary amplitude scaling and time-delay [15]. In the present setting, however, we suppose, without loss of generality, that the source stream power and the system energy are known, thus the amplitude of the recovered source is controlled by the norm of the weight-vector \mathbf{w} . Also, during filter learning, the misadjustment of filter's coefficients makes the filter-output differ from the source signal.

An appropriate Bayesian estimator of the source sequence of the form $B(z_{m,n})$ can be designed according to Bayesian estimation theory. On the basis of the available memory-less Bayesian estimator, in [2] the error criterion $C(\mathbf{w}_m) \stackrel{\text{def}}{=} \frac{1}{2} \mathbb{E}_{z_{m,n}} [(z_{m,n} - B(z_{m,n}))^2]$ had been proposed by Bellini. For uniformly distributed source stream, which is of interest, e.g., in telecommunication systems, a suitable approximation of the actual Bayesian estimator $B(z)$ is the neural transfer function [15]:

$$\hat{B}(z) = \kappa \tanh(\lambda z) , \quad (2)$$

with κ and λ being properly tuned parameters. In order to select suitable values for these parameters, in [10] we proposed to adapt them through time by means of a gradient-based algorithm applied to $C(\kappa, \lambda, \mathbf{w})$. In the recent contribution [13], a batch procedure was proposed in order to optimize the values of the parameters κ and λ in the neural activation function (2) prior to filter learning. The same procedure is adopted in the present work as a pre-learning stage. (An extensive analysis of the selection criterion for these parameters is available in [14].) Also, the automatic gain control (AGC) constraint, which is typical in telecommunications, may be invoked. The AGC constraint aims at keeping constant the energy of the filter impulse response sequence, that means enforcing the constraint:

$$w_0^2 + w_2^2 + w_3^2 + \dots + w_{L_w-1}^2 = 1 . \quad (3)$$

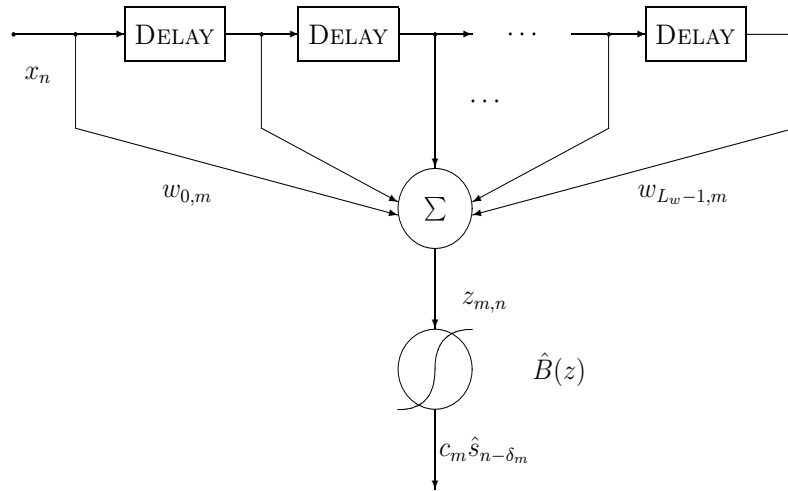


FIGURE 1. A neuron with filtering synapses. (Unitary delays endow the neural system with temporal memory, namely with samples x_{n-i} , while $w_{i,m}$ denotes the value of the i^{th} filter tap-weight at iteration m .) The quantity $c_m \in \mathcal{R}$ denotes instantaneous amplitude warping and quantity $\delta_m \in \mathcal{Z}$ denotes instantaneous group delay intrinsic to deconvolution process.

It deserves recalling here that the employed neural structure is closely related a neural multi-layer perceptron (MLP) structures endowed with filtering synapses (a review of which was given, e.g., in [20]), even if its learning paradigm is, by the ‘blind’ nature of the problem, inherently unsupervised. A sketch of the employed deconvolving neuron is shown in the Figure 1.

DISCUSSED ALGORITHMS

The aims of this section are to recall the geodesic-based deconvolution algorithm, which relies on the geometry of the parameter space induced by the AGC constraint (3), and the projection-based algorithm. Also, details on the stability of the algorithms are given.

Learning algorithm based on geodesic-arcs

The first step in the development of a gradient-based algorithm consists in the recognition of the geometry of the parameter space induced by the AGC constraint. The parameter space is the hyper-sphere $S^{p-1} \stackrel{\text{def}}{=} \{\mathbf{v} \in \mathcal{R}^p | \mathbf{v}^T \mathbf{v} = 1\}$. At every point $\mathbf{v} \in S^{p-1}$, the linear space tangent to the sphere has structure:

$$T_{\mathbf{v}} S^{p-1} \stackrel{\text{def}}{=} \{\mathbf{u} \in \mathcal{R}^p | \mathbf{u}^T \mathbf{v} = 0\}. \quad (4)$$

In fact, by definition, the tangent space at \mathbf{v} is spanned by vectors tangent to the curves belonging to the base manifold and passing by the point \mathbf{v} . Let us consider, thus, a generic smooth curve $\mathbf{v}(t)$ parameterized by $t \in \mathcal{R}$, which passes by the indicated point

at $t = 0$. The tangent vector to the curve is the velocity vector $\dot{\mathbf{v}}(t)$ at time $t = 0$. The velocity vector should satisfy $\frac{d}{dt}(\mathbf{v}^T(t)\mathbf{v}(t)) = 0$, that is, $2\dot{\mathbf{v}}^T(t)\mathbf{v}(t) = 0$. At time $t = 0$, the latter condition gives the tangency condition in (4). Also, the normal space at every point of the base-manifold, which is the orthogonal complement of the tangent space with respect to a suitable Euclidean ambient space that the manifold is embedded within, may be defined as well. Here we make use of the definition $N_{\mathbf{v}}S^{p-1} \stackrel{\text{def}}{=} \{\mathbf{r} \in \mathcal{R}^p | \langle \mathbf{r}, \mathbf{u} \rangle_{\mathcal{R}^p} = 0, \forall \mathbf{u} \in T_{\mathbf{v}}S^{p-1}\} = \{\lambda \mathbf{v} | \lambda \in \mathcal{R}\}$, where the ambient space was assumed to be \mathcal{R}^p , endowed with its canonical scalar product: $\langle \mathbf{r}_1, \mathbf{r}_2 \rangle_{\mathcal{R}^p} \stackrel{\text{def}}{=} \mathbf{r}_1^T \mathbf{r}_2$ for all $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{R}^p$. The smooth manifold S^{p-1} is turned into a Riemannian manifold by endowing it with a local scalar product $\langle \cdot, \cdot \rangle_{\mathbf{v}} : T_{\mathbf{v}}S^{p-1} \times T_{\mathbf{v}}S^{p-1} \rightarrow \mathcal{R}$.

As an optimization method that allows to look for the minimum (or local minima) of the function $f(\mathbf{v})$ over S^{p-1} , the standard Riemannian-gradient-based rule:

$$\frac{d\mathbf{v}}{dt} = -\nabla_{\mathbf{v}}^{S^{p-1}} f, \quad (5)$$

may be employed. From differential geometry, it is known that, given a regular function $f : S^{p-1} \rightarrow \mathcal{R}$, its Riemannian gradient is the vector $\nabla_{\mathbf{v}}^{S^{p-1}} f$ that satisfies the following conditions:

$$\nabla_{\mathbf{v}}^{S^{p-1}} f \in T_{\mathbf{v}}S^{p-1} \text{ and } \langle \nabla_{\mathbf{v}}^{S^{p-1}} f, \mathbf{u} \rangle_{\mathbf{v}} = \left(\frac{\partial f}{\partial \mathbf{v}} \right)^T \mathbf{u}, \forall \mathbf{u} \in T_{\mathbf{v}}S^{p-1}.$$

In order to compute the Riemannian gradient, it is necessary to select a metric. The unit-hyper-sphere S^{p-1} is a special case of a more general geometrical structure known as Stiefel manifold, for which two metrics are commonly employed: The Euclidean and the canonical metrics (see e.g. [9, 11]). In the case of S^{p-1} , these metrics coincide and are given by the uniform metric $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{\mathbf{v}} \stackrel{\text{def}}{=} \mathbf{u}_1^T \mathbf{u}_2$. By applying the above conditions, we get $\nabla_{\mathbf{v}}^{S^{p-1}} f = (\mathbf{I}_p - \mathbf{v}\mathbf{v}^T) \frac{\partial f}{\partial \mathbf{v}}$, where \mathbf{I}_p denotes the $p \times p$ identity matrix. For the blind-deconvolution problem at hand, the differential equation (5) may thus be customized as:

$$\frac{d\mathbf{w}}{dt} = -(\mathbf{I}_p - \mathbf{w}\mathbf{w}^T) \frac{\partial C(\mathbf{w})}{\partial \mathbf{w}}. \quad (6)$$

In the blind deconvolution context, the dimension of the parameter space coincides to the length of the inverse filter impulse response, namely $p = L_w$. For the partial derivative of the cost function, it holds:

$$\frac{\partial C(\mathbf{w})}{\partial \mathbf{w}} = \mathbb{E}_{\mathbf{x}}[\gamma(z)\mathbf{x}], \quad \gamma(z) \stackrel{\text{def}}{=} (B(z) - z)(B'(z) - 1). \quad (7)$$

In practice, a suitable numerical integration method should be selected in order to solve the differential equation (5). We propose here to employ the integration method based on geodesic arcs.

On a Riemannian manifold embedded in a Euclidean space, a geodesic may be defined as a curve on which a particle, departing from the point \mathbf{v}_0 with velocity \mathbf{g} , slides with constant scalar speed $\|\mathbf{g}\|$. We denote such curve as $\mathbf{v}(t) = \Gamma(t, \mathbf{v}_0, \mathbf{g})$, where the variable

$t \geq 0$ provides a parameterization. In the present context, $\mathbf{v}_0 \in S^{p-1}$ and $\mathbf{g} \in T_{\mathbf{v}_0}S^{p-1}$. The equation of the geodesic may be found by observing that the acceleration of the particle is either null or normal to the tangent space at any point, namely $\ddot{\mathbf{v}} \in N_{\mathbf{v}}S^{p-1}$. In explicit form, the equation of the geodesic on the unit hyper-sphere may be found by solving the following system:

$$\ddot{\mathbf{v}} - \lambda \mathbf{v} = \mathbf{0}, \quad \mathbf{v}(0) = \mathbf{v}_0 \in S^{p-1}, \quad \dot{\mathbf{v}}(0) = \mathbf{g} \in T_{\mathbf{v}_0}S^{p-1}. \quad (8)$$

The solution of the above differential system is [7]:

$$\mathbf{v}(t) = \Gamma(t, \mathbf{v}_0, \mathbf{g}) = \cos(\|\mathbf{g}\|t) \mathbf{v}_0 + \sin(\|\mathbf{g}\|t) \frac{\mathbf{g}}{\|\mathbf{g}\|}, \quad (9)$$

where $\|\cdot\|$ denotes the standard L_2 vector norm. It is straightforward to verify that $\mathbf{v}^T(t)\mathbf{v}(t) = 1$, $\dot{\mathbf{v}}^T(t)\mathbf{v}(t) = 0$ and that $\|\dot{\mathbf{v}}(t)\| = \|\mathbf{g}\|$, for all $t \geq 0$. The relationship (9) for the geodesic represents a ‘great circle’ on the hyper-sphere, which is a closed curve, therefore, it makes sense to restrict the value of t to an interval such that, e.g., $0 \leq \|\mathbf{g}\|t \leq \pi$.

A way to approximate the exact flow of the differential equation on a manifold (6) via geodesic arcs is to make use of the following iteration rule:

$$\mathbf{w}_m = \Gamma\left(\delta, \mathbf{w}_{m-1}, -\nabla_{\mathbf{w}_{m-1}}^{S^{p-1}} C(\mathbf{w})\right), \quad m \in 1..M, \quad (10)$$

where δ denotes an appropriate constant adaptation stepsize and $\mathbf{w}_0 \in S^{p-1}$. It is known [7], that the geodesic step (10) provides a first-order approximation to the true flow of the differential equation (6) with initial condition \mathbf{w}_m , namely $\mathbf{w}_m - \mathbf{w}(\delta) = o(\delta^2)$.

Learning algorithm based on projection

A second way individuated in this paper to perform adaptation on the unit hyper-sphere is updated-vector projection. Technically, it is first necessary to perform the embedding $S^{p-1} \hookrightarrow \mathcal{R}^p$ of the hyper-sphere into the Euclidean manifold \mathcal{R}^p , so that we get a larger ambient space to move in. Now, every updating step may be performed safely from S^{p-1} to \mathcal{R}^p , by following, e.g., Euclidean gradient direction, and then the updated vector may be projected back to the manifold S^{p-1} by the help of a suitable projector $\Pi : \mathcal{R}^p \rightarrow S^{p-1}$: The update vector before projection does not belong to the unit hyper-sphere and it is therefore necessary to project it into the sphere through the operator $P(\cdot)$. The algorithm may be formally described as:

$$\mathbf{w}_m = P\left(\mathbf{w}_{m-1} - \delta \frac{\partial C(\mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_{m-1}}\right), \quad P(\mathbf{v}) \stackrel{\text{def}}{=} \frac{\mathbf{v}}{\sqrt{\mathbf{v}^T \mathbf{v}}}. \quad (11)$$

The quantity δ in the projection-based algorithm (11) denotes again an appropriate constant adaptation stepsize.

On the stability of learning algorithms

With reference to the geodesic-based algorithm, it is easy to recognize that, if the time to within the geodesic is extended is short enough, then the algorithm essentially follows the Riemannian-gradient flow. In fact, for t small enough, the expression (9) may be approximated as:

$$\Gamma(t, \mathbf{v}_0, \mathbf{g}) \approx \left(1 - \frac{\|\mathbf{g}\|^2 t^2}{2}\right) \mathbf{v}_0 + \mathbf{g}t.$$

If this approximation is plugged in the expression (10), the result is:

$$\frac{\mathbf{w}_m - \mathbf{w}_{m-1}}{\delta} \approx -\frac{\|\nabla_{\mathbf{w}_{m-1}}^{S^{p-1}} C(\mathbf{w})\|^2 \delta}{2} \mathbf{w}_{m-1} - \nabla_{\mathbf{w}_{m-1}}^{S^{p-1}} C(\mathbf{w}).$$

The above expression shows how the approximate derivative $\frac{\mathbf{w}_m - \mathbf{w}_{m-1}}{\delta}$ has a normal component (the leftmost one on the right-hand side) and a tangent component (the rightmost one on the right-hand side). The normal component may be made arbitrarily small by properly selecting the learning stepsize δ . In any case, it is interesting to note that the normal component points towards the interior of the hyper-sphere, so it likely is not a source of instability.

About the projection-based algorithm, it falls within the class of fixed-point algorithms [13, 19]. The standard mathematical tool for proving the convergence of such kind of algorithms is the Banach theorem, which insists on the contractivity of the operator that describes how a vector-state \mathbf{w}_{m-1} is transported into \mathbf{w}_m . However, in the author's experience (and as confirmed by the numerical experiments presented in the next section), often proving/checking that such operator is not contractive is a hard task and yet the algorithm is convergent. A different approach is pursued, e.g., in [18], where a stepsize sequence is computed in such a way to ensure a fixed-point algorithm does converge, on the basis of the local curvature of the criterion function to be optimized.

RESULTS OF NUMERICAL EXPERIMENTS

In the following experiments, it is assumed that s_n is a white random signal uniformly distributed within $[-\sqrt{3}, +\sqrt{3}]$, whose length is of $N = 5,000$ samples.

The system deconvolution accuracy may be measured by means of the residual inter-symbol interference (ISI), defined as [19] $\text{ISI}_m \stackrel{\text{def}}{=} \frac{\mathbf{T}_m^T \mathbf{T}_m - T_{m,\max}^2}{T_{m,\max}^2}$, where $\mathbf{T}_m \stackrel{\text{def}}{=} \mathbf{h} \otimes \mathbf{w}_m$ denotes the convolution between the system's impulse response and the inverse filter's impulse response, and $T_{m,\max}$ denotes the component of \mathbf{T}_m having the maximal absolute value.

Whenever appropriate, thanks to the hypothesized ergodicity, the ensemble average $\mathbb{E}[\cdot]$ may be numerically estimated by $\mathbb{E}_{z_{m,n}}[\Phi(z_{m,n})] \approx \frac{1}{N} \sum_{n=1}^N \Phi(z_{m,n})$, which is a function of m , for a generic vector-valued function $\Phi: \mathcal{R} \rightarrow \mathcal{R}^p$.

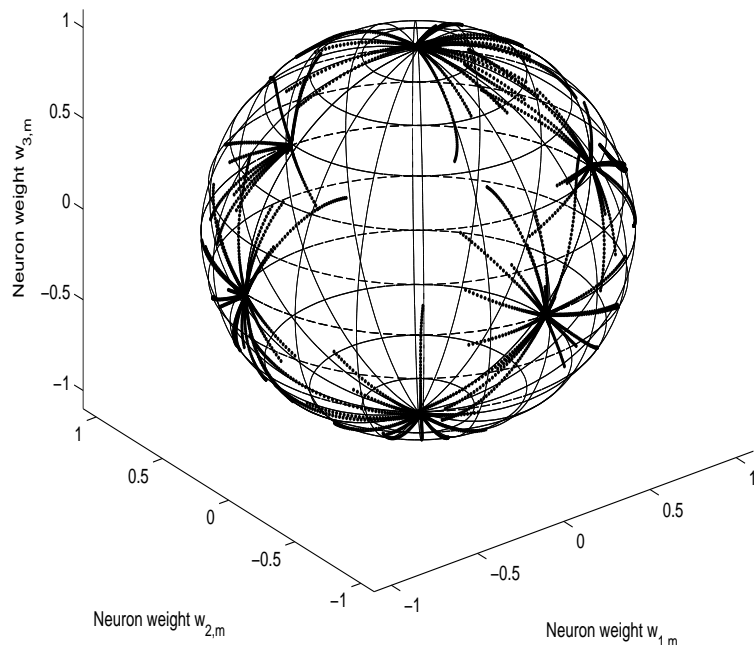


FIGURE 2. Trajectories of the randomly initialized geodesic-based blind-deconvolution algorithm on the base manifold S^2 for the experiment with the non-distorting channel.

Numerical experiments with a toy channel

In order to illustrate the behavior of the geodesic-based algorithm as well as projection-based blind-deconvolution algorithm, it deserves to consider first a low-dimensional experiment with a toy (non-distorting) channel. We considered the channel's impulse response to be $\mathbf{h} = [1]$ (namely, $L_h = 1$) and the base manifold to be S^2 (namely, $L_w = 3$). In this case, the base-manifold as well as the learning trajectory \mathbf{w}_m may be rendered in graphical way. In this experiment, the global channel-filter-cascade impulse response $\mathbf{T}_m = \mathbf{h} \otimes \mathbf{w}_m = \mathbf{w}_m$, therefore, as the channel impulse response is non-distorting, if we let the Bussgang neuron learning trajectory depart from a randomly generated weight-vector $\mathbf{w}_0 \in S^2$, it should eventually converge to one of the six attractors $[\pm 1 \ 0 \ 0]^T$, $[0 \ \pm 1 \ 0]^T$ or $[0 \ 0 \ \pm 1]^T$.

The numerical results for the geodesic-based algorithm, obtained on 100 independent trials with randomly generated initial weight-vectors on the sphere, with $M = 100$ learning iterations per trial, with learning stepsize $\delta = 0.5$, are depicted in Figure 2: All the whole trajectories are completely lying on the sphere and converge to one of the six attractors placed on it. No diverging (i.e., manifold-escaping) trajectories were observed. The numerical results for the projection-based algorithm, obtained on 100 independent trials, with $M = 100$ learning iterations per trial, with learning stepsize $\delta = 0.9$, are

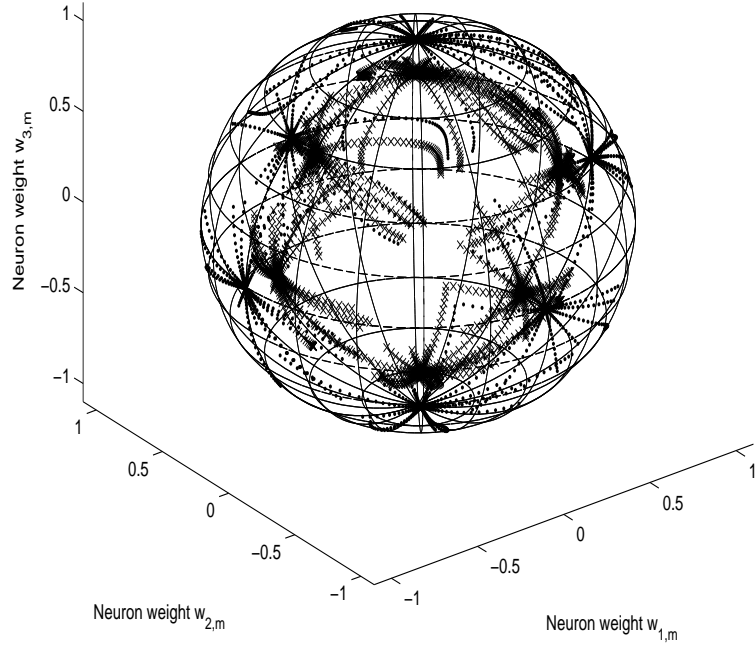


FIGURE 3. Trajectories of the randomly initialized projection-based blind-deconvolution algorithm on the base manifold S^2 for the experiment with the non-distorting channel: Dotted-line corresponds to normalized trajectory, while crossed-line corresponds to the steps before normalization.

depicted in the Figure 3: In this figure, both the filter impulse response trajectories before normalization and after normalization may be observed. All trajectories converge to one of the six attractors.

Numerical experiments with a telephonic channel

The discussed algorithms were tested to learn an inverse filter for the sampled telephonic channel described in [4] having duration $L_h = 14$: Its features are illustrated in Figure 4 (the channel impulse response has been normalized so that $\mathbf{h}^T \mathbf{h} = 1$). The length of the neural filter impulse response was assumed $L_w = 14$ as the result of validation [10, 13, 14]. In all the following experiments, the initial impulse response of the filter, namely \mathbf{w}_0 , was assumed as a null sequence, except for the 7th tap-weight that was set to 1.

Results concern the analysis of the behavior of the geodesic-based and projection-based algorithms on a noiseless channel (i.e., model (1) with $\nu_n \equiv 0$ identically). The Figure 5 illustrates the performance indices pertaining to both algorithms, in which the constant learning stepsize $\delta = 1$ was chosen for the geodesic-based algorithm and

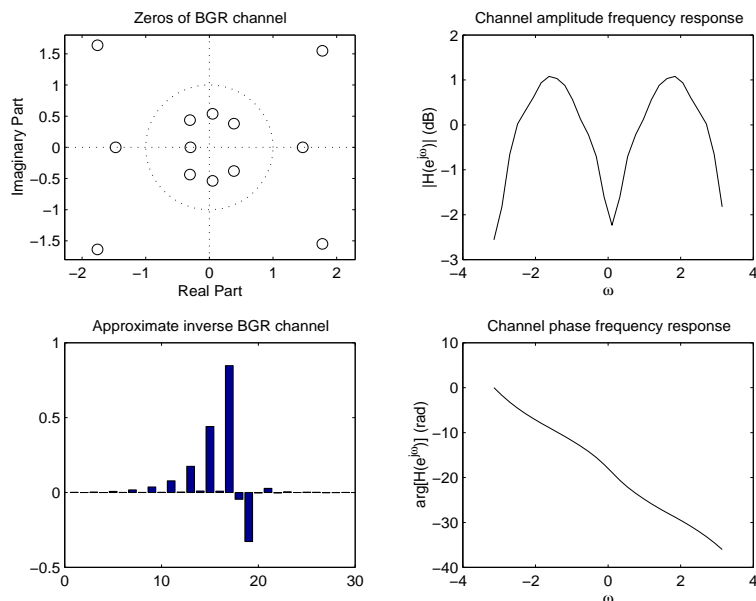


FIGURE 4. Sampled telephonic channel. Left: Zero-plot and impulse response bar. Right: Amplitude/phase of the frequency response $H(e^{j\omega})$ of the channel.

$\delta = 0.9$ was chosen for the projection-based algorithm, as a result of validation. The Figure 5 also illustrates the learnt filter after $M = 80$ learning iterations as well as the convolution \mathbf{T} after learning (this result is almost the same for both algorithms). While both algorithms perform in a satisfactory way, reaching fairly low ISI values, it is to be noted that the geodesic-based one converges more steadily, in this case.

Computational complexity comparison

The discussed algorithms were compared in terms of computational complexity, where the flops-count and the elapsed-time for every run are retained as measures of the computational burden of each algorithm provided they exhibit comparable deconvolution performances. The experiments were performed under MATLAB[®] 5.3, which provides flops count, on a 1.86GHz – 512MB machine. The results of this comparative analysis are summarized in the Table 1. Both algorithms were run on the same batch of 5,000 channel output samples, on the same noiseless BGR channel and adapted through $M = 50$ iterations. The flops count refers to the number of floating point operations required by the implemented code to run, averaged over the total number of samples (in this case $5,000 \times 50$). In this comparison, the time count refers to the total time required by each algorithm to run on the specified platform. As already noted, the deconvolution performances are comparable for the two algorithms, while the projection-based one proves to be slightly lighter from a computational point of view.

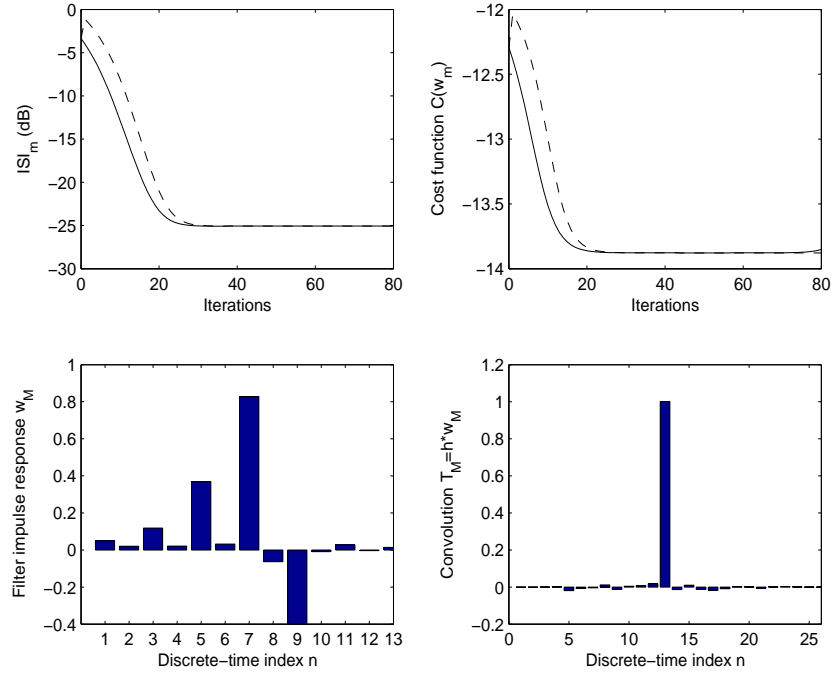


FIGURE 5. Experiments on sampled telephonic channel. Comparison of results obtained with the geodesic-based (solid line) and projection-based (dashed-line) blind deconvolution algorithms: Performance indices and learnt filter.

TABLE 1. Results of computational-complexity comparison of the geodesic-based algorithm and the projection-based algorithm.

ALGORITHM	ISI (dB)	Flops	Time (sec.s)
Geodesic-based	-25.057	80.594	0.328
Projection-based	-25.056	81.582	0.313

CONCLUSIONS

This contribution aimed at discussing a numerical comparison of a Riemannian-gradient-based and a projection-based learning algorithm over a curved parameter space for single-neuron learning with application to blind deconvolution, which may be tackled via a single learning neuron model whose learning strategy naturally arises as criterion-function minimization over the unitary hyper-sphere.

The blind deconvolution performances of the two algorithms as well as their computational burden and numerical features were considered and compared. The numerical results evidenced that both algorithms are well-behaving and that the geodesic-based algorithm exhibits steadier convergence.

REFERENCES

1. **S.-i. Amari and S. Fiori**, *Geometrical methods in neural networks and learning*, Editorial for a special issue of Neurocomputing journal (Ed.s S. Fiori and S.-i. Amari), Vol. 67C, pp. 1 - 7, August 2005
2. **S. Bellini**, *Blind equalization*, Alta Frequenza, Vol. 57, pp. 445 – 450, 1988
3. **A.J. Bell and T.J. Sejnowski**, *An information maximisation approach to blind separation and blind deconvolution*, Neural Computation, Vol. 7, No. 6, pp. 1129 – 1159, 1995
4. **A. Benveniste, M. Goursat and G. Ruget**, *Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communication*, IEEE Trans. on Automatic Control, Vol. AC-25, No. 3, pp. 385 – 399, June 1980
5. **E. Celledoni and S. Fiori**, *Neural learning by geometric integration of reduced ‘rigid-body’ equations*, Journal of Computational and Applied Mathematics (JCAM), Vol. 172, No. 2, pp. 247 – 269, December 2004
6. **S. Choi, S. Ong, J. Cho, C. You and D. Hong**, *Performances of neural equalizers on partial erasure model*, IEEE Trans. on Magnetics, Vol. 33, No. 5, pp. 2788 – 2790, Sept. 1997
7. **N. Del Buono and L. Lopez**, *Runge-Kutta type methods based on geodesics for systems of ODEs on the Stiefel manifold*, BIT – Numerical Mathematics, Vol. 41, No. 5, pp. 912 – 923, 2001
8. **Z. Ding and Y. Li**, *Blind Equalization and Identification*, Marcel Dekker, New York, 2001
9. **A. Edelman, T.A. Arias and S.T. Smith**, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis Applications, No. 20, pp. 303 – 353, 1998
10. **S. Fiori**, *A contribution to (neuromorphic) blind deconvolution by flexible approximated Bayesian estimation*, Signal Processing, Vol. 81, No. 10, pp. 2131 – 2153, Sept. 2001
11. **S. Fiori**, *A theory for learning by weight flow on Stiefel-Grassman manifold*, Neural Computation, Vol. 13, No. 7, pp. 1625 – 1647, July 2001
12. **S. Fiori**, *A theory for learning based on rigid bodies dynamics*, IEEE Trans. on Neural Networks, Vol. 13, No. 3, pp. 521 – 531, May 2002
13. **S. Fiori**, *A fast fixed-point neural blind deconvolution algorithm*, IEEE Trans. on Neural Networks, Vol. 15, No. 2, pp. 455 – 459, March 2004
14. **S. Fiori**, *Analysis of modified ‘Bussgang’ algorithms (MBA) for channel equalization*, IEEE Trans. on Circuits and Systems - Part I, Vol. 51, No. 8, pp. 1552 – 1560, August 2004
15. **S. Haykin**, *Adaptive Filtering*, (Chapter 20: Blind Deconvolution), Prentice-Hall, 1991
16. **D. Kundur and D. Hatzinakos**, *Blind image deconvolution*, IEEE Signal Processing Magazine, Vol. 13, No. 3, pp. 43 – 64, May 1996
17. **H. Park, S.-i. Amari and Y. Lee**, *An information geometrical approach on plateau problems in multilayer perceptron learning*, Journal of KISS (B): Software and Applications, Vol. 26, No. 4, pp. 546 – 556, 1999
18. **P.A. Regalia and E. Kofidis**, *Monotonic convergence of fixed-point algorithms for ICA*, IEEE Trans. on Neural Networks, Vol. 14, No. 4, pp. 943 – 949, July 2003
19. **O. Shalvi and E. Weinstein**, *Super-exponential methods for blind deconvolution*, IEEE Trans. on Information Theory, Vol. 39, No. 2, pp. 504 – 519, 1993
20. **A.C. Tsoi and A.D. Back**, *Discrete time recurrent neural network architectures: A unifying review*, Neurocomputing, Vol. 15, pp. 183 – 223, June 1997
21. **R.A. Wiggins**, *Minimum entropy deconvolution*, Geoplotting, Vol. 16, pp. 21 – 35, 1978