Neural Networks 81 (2016) 72–80

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Neural network training as a dissipative process

Marco Gori, Marco Maggini, Alessandro Rossi *

Department of Information Engineering and Mathematics, University of Siena, Italy

HIGHLIGHTS

- A learning theory based on the variational principle of least cognitive action.
- Supervised On-line Learning evolving as a dissipative dynamic system.
- Stochastic or Batch Gradient Descent are obtained by varying the dissipation level.
- Experimental evaluation on standard and custom benchmarks.

ARTICLE INFO

Article history:
Received 19 February 2016
Received in revised form 11 May 2016
Accepted 28 May 2016
Available online 21 June 2016

Keywords:
Temporal manifolds
Regularization networks
Dissipative systems
On-line back-propagation

ABSTRACT

This paper analyzes the practical issues and reports some results on a theory in which learning is modeled as a continuous temporal process driven by laws describing the interactions of intelligent agents with their own environment. The classic regularization framework is paired with the idea of temporal manifolds by introducing the principle of *least cognitive action*, which is inspired by the related principle of mechanics. The introduction of the counterparts of the kinetic and potential energy leads to an interpretation of learning as a dissipative process. As an example, we apply the theory to supervised learning in neural networks and show that the corresponding Euler–Lagrange differential equations can be connected to the classic gradient descent algorithm on the supervised pairs. We give preliminary experiments to confirm the soundness of the theory.

1. Introduction

In a seminal paper on regularization networks, Poggio and Girosi (1989) provided a nice formulation of supervised learning using differential operators. In the simplest case, the differential operator P, used to define the regularization term, is the gradient, but they considered a more general class of operators based also on high-order derivatives. The proposed variational formulation of learning was based on performing regularization in terms of smoothness of the task f, so as they ended-up into the Euler–Lagrange equation

$$\hat{P}Pf(x) = \frac{1}{\lambda}\sum_{i=1}^{N}(y_i - f(x))\delta(x - x_i), \tag{1}$$

where $\hat{P}$ is the adjoint operator of P, (x_i, y_i), $i = 1, \ldots, N$ are the supervised examples, and λ is a settable parameter to balance the regularization with respect to the approximation accuracy on the training set.[1] Interestingly, this partial differential equation provides a very good framework to introduce kernel machines. In fact, the Green function $G(\cdot, \cdot)$ of the linear operator $L = \hat{P}P$, defined as its response to the Dirac delta impulse (i.e. $LG = \delta$), can be considered as a kernel (Gnecco, Gori, & Sanguineti, 2013; Schoelkopf & Smola, 1998). As a consequence, the solution of the Euler–Lagrange equation (1) can be written as the superposition of the responses to Dirac delta impulses centered in the points of the training set (i.e. $G(x, x_i)$), which leads to the classic representer theorem of kernel machines (Gnecco et al., 2013; Schoelkopf & Smola, 1998). Needless to say, the related mathematical and algorithmic framework has played a crucial role in the development of the field of machine learning in the last twenty years.

Recently, Eq. (1) has been reformulated in a more general framework in which, instead of dealing with intelligent agents that interact with the environment only by supervised examples, the agent optimizes its behavior also to satisfy a given set of constraints

* Correspondence to: Department of Information Engineering and Mathematics, University of Siena, Via Roma 56, 53100, Siena, Italy.
E-mail addresses: marco.gori@unisi.it (M. Gori), marco.maggini@unisi.it (M. Maggini), rossi111@unisi.it (A. Rossi).

http://dx.doi.org/10.1016/j.neunet.2016.05.005
0893-6080/

[1] The equation is derived in the case of a quadratic loss function on the supervised examples.

(Gnecco, Gori, Melacci, & Sanguineti, 2015). New representer theorems are given to express the optimal solution for a large class of constraints. Some guidelines were given to parallel the plain kernel-based solution of (1) with cases where special kernels arise from the marriage of the regularization operators with the specific constraints (see also Melacci & Gori, 2013). A classic example of constraint studied in Gnecco et al. (2015) is the one which imposes the brightness invariance in computer vision, which makes it possible to estimate the optical flow (Horn & Schunck, 1981). Interestingly, the kernel-based approaches used so far, along with their mathematical apparatus, are clearly the only viable solution. As a matter of fact, the differential equation (1), as well as the related Euler–Lagrange equations derived in Gnecco et al. (2015), cannot be tackled with an efficient numerical solution. They are in fact formulated in the feature space, whose dimension makes it unfeasible to grid the space for the classic discretization of the differential operator[2] $L = \hat{P}P$. This fundamental computational issue suggests that any biological system involved in learning processes should obey to models which somehow circumvent the curse of dimensionality related to Eq. (1).

This paper arises from the wish of facing this fundamental issue. When shifting to natural learning processes, one can promptly realize that the described formulations are essentially operating according to batch-mode and, most importantly, they miss the truly meaning of time. What if we shift towards a formulation in which an intelligent agent lives in an environment that provides constraints on a temporal basis? Unlike what is assumed in the above formulation, the incoming points in the feature space belong to a certain temporal manifold (i.e. a trajectory in the feature space), which is expected to drive the learning process. We fully rely on the principle that there is no need to deal with the reaction of the constraints and with the corresponding regularization in high-dimensional feature spaces, since we can simply focus the attention on the temporal manifold by approaching learning as a continuous temporal process. Bearing in mind this principle, we reformulate learning by introducing the *principle of least cognitive action*. These concepts have been first proposed in Frandina, Gori, Lippi, Maggini, and Melacci (2013), where it is shown how this variational formulation of learning can be easily associated with the classic technique of gradient descent in on-line back-propagation. A solid theoretical formulation of the principle and the incorporation of the energy dissipation in the process has been given in Betti and Gori (2015). The work is inspired to the related principle of mechanics and to the Hamiltonian framework for modeling the motion of particles. Unlike mechanics, however, the cognitive action that we define is in fact the objective to be minimized, more than a functional for which to discover a stationary point. This duality is based on a proper introduction of the "kinetic energy" and of the "potential energy", that leads to a surprisingly natural interpretation of learning as a dissipative process. The kinetic energy reflects the temporal variation of the synaptic connections, while the potential energy is a penalty that describes the degree of satisfaction of the environmental constraints. The proposed approach to learning naturally incorporates time in its truly continuous structure, so as the evolution of the weights of the neural synapses follows equations that resemble laws of physics (see Table 1). The most important conclusion is that the corresponding Euler–Lagrange equations involve temporal functions, allowing us to circumvent the mentioned curse of dimensionality issue. Hence, the proposed theory makes it possible to formulate a sound learning process by the direct solution of differential equations on a temporal manifold, avoiding numerical approximations.

We show the application of the theory to the classic case of supervised learning in neural networks and give the first example of a new learning algorithm derived from the theory. We show how to face practical issues and how the parameters of the proposed model affect the learning process. Finally, we provide experiments to support the soundness of the theory.

2. The principle of least cognitive action

We assume that the agent processes an input evolving in a feature space in $\mathbb{R}^d$, as expressed by a function of time $u : [t_0, t_1] \to \mathbb{R}^d$, that is mapped to the output $z \in \mathbb{R}^m$ by means of $z(t) = f(w(t), u(t))$. For now we do not impose any constraint on the structure of f, that, for example, could be implemented by a feedforward neural network, whose weights are stacked into the vector $w(t)$ at time t. The task is to learn the weights $w(t) \in \mathbb{R}^n$ of the function f. We formulate the online learning process of the agent by providing the constraints that define the interactions with the environment, according to the general framework defined in Gnecco et al. (2015). A quite general case is the one in which the tasks should satisfy the equation $\phi(f(w(t), u(t))) = 0$, where $\phi(\cdot) \geq 0$ is the function modeling the constraints of the agent's environment. The fulfillment of the constraints during the learning process can be enforced by the minimizing penalty

$$V(w) = \int_{t_0}^{t_1} \phi(f(w(t), u(t)))dt. \tag{2}$$

We can think of V as a potential energy connected with the constraint ϕ, such that the aim of learning is to develop configurations with small potential. The classic case of supervised learning can be incorporated into this framework when considering the Dirac distribution

$$\phi(f(w(t), u(t))) = \sum_{t_k \leq t} \overline{V}(y_k, f(w(t), u(t))) \cdot \delta(t - t_k) \tag{3}$$

where $\overline{V}(y, s)$ is a loss function.[3] Here supervisions are provided by an external teacher at discrete time instants t_k, $k \in \mathbb{N}$, by specifying a target value y_k. The supervised pairs $(u(t_k), y_k)$ are collected into the set $\mathcal{L} = \{(u(t_k), y_k)\}_{k \in \mathbb{N}}$ and are provided while the agent's input $u(t)$ evolves in time.

By following a parallel with the classical mechanics, the weights w can be thought of as the *Lagrangian coordinates* in a virtual mechanical system, whose kinetic energy is defined as $K = \sum_{i=1}^{n} \mu_i \dot{w}_i^2$. The value μ_i, $i = 1, \ldots, n$, represents the mass of the particle associated with w_i. Taking inspiration from Eq. (1), we may generalize the concept of velocity by means of differential operator

$$P = \sum_{j=0}^{\ell} \alpha_j \frac{d^j}{dt^j}. \tag{4}$$

The coefficients α_j of P are the regularization weights related to the correspondent order of derivative applied to $w_i(t)$. The generalized velocity turns out to be Pw and the correspondent kinetic energy is

$$K(Pw) = \sum_{i=1}^{n} \mu_i (Pw_i)^2. \tag{5}$$

Now we can provide a formulation which resembles the principle of least action in physics by defining the Lagrangian

$$F(t, w, Pw) = K(Pw) + \gamma V(w), \tag{6}$$

[2] Methods like Runge–Kutta have a unmanageable complexity for the dimensions of interest in machine learning.

[3] For instance, $\overline{V}(y, s)$ can be the quadratic loss $\overline{V}(y, s) = \frac{1}{2}(y - s)^2$.