Toward Novel Optimizers: A Moreau-Yosida View of Gradient-Based Learning

Alessandro Betti[1](✉), Gabriele Ciravegna[2], Marco Gori[1,3], Stefano Melacci[3], Kevin Mottin[1], and Frédéric Precioso[1]

[1] Université Côte d'Azur, Inria, CNRS, Laboratoire I3S, Maasai Team, Nice, France
{alessandro.betti,kevin.mottin,frederic.precioso}@inria.fr
[2] Politecnico di Torino, Torino, Italy
gabriele.ciravegna@polito.it
[3] Department of Information Engineering and Mathematics, University of Siena, Siena, Italy
marco.gori@unisi.it, mela@diism.unisi.it

Abstract. Machine Learning (ML) strongly relies on optimization procedures that are based on gradient descent. Several gradient-based update schemes have been proposed in the scientific literature, especially in the context of neural networks, that have become common optimizers in software libraries for ML. In this paper, we re-frame gradient-based update strategies under the unifying lens of a Moreau-Yosida (MY) approximation of the loss function. By means of a first-order Taylor expansion, we make the MY approximation concretely exploitable to generalize the model update. In turn, this makes it easy to evaluate and compare the regularization properties that underlie the most common optimizers, such as gradient descent with momentum, ADAGRAD, RMSprop, and ADAM. The MY-based unifying view opens to the possibility of designing novel update schemes with customizable regularization properties. As case-study we propose to use the network outputs to deform the notion of closeness in the parameter space.

1 Introduction

Gradient based optimization procedures are arguably one of the main ingredients of Machine Learning (ML). As a matter of fact, the success of deep learning strongly relies on the efficiency of Stochastic Gradient Descent (SGD) to solve large scale optimization problems [3]. It is pretty common to introduce the gradient descent method in Euclidean spaces, leveraging on the geometrical interpretation of the *direction of steepest descent* [5], that yields the definition of gradient descent by means of an iterative procedure. In particular, we are given a function f which we aim at minimizing, and which depends on some parameters $w \in \mathbb{R}^N$. If w^k are the values of the parameters at the k-th step of the gradient descent, their update scheme is given by

$$w^{k+1} = w^k - \tau \nabla f(w^k), \tag{1}$$

R. Basili et al. (Eds.): AIxIA 2023, LNAI 14318, pp. 218–230, 2023.
https://doi.org/10.1007/978-3-031-47546-7_15

for some initial $w^0 = w_0 \in \mathbb{R}^N$ and $\tau > 0$ (learning rate, step size), being ∇f the gradient of f with respect to w. The ML literature includes several works aimed at providing adaptive values for τ, eventually considering a specific learning rate for each component of w, or introducing further terms in the update rule [7,10,19,22]. Despite their large ubiquity in implementations of ML-based solutions, such works are inspired by different principles and they are presented starting from different problem formulations, that make it hard to quickly compare them. Moreover, simply looking at the final update rules they devise is not enough to fully grasp what are the expected effects they bring to the optimization. Delving into their details to trace connections among them is definitely possible, but it is not straightforward.

Motivated by these considerations, in this paper we propose to reconsider the aforementioned update rules, casting them all into a *unified view* in which w^{k+1} is presented as the solution of a newly introduced optimization problem with well-defined and clearly interpretable regularity conditions. Such conditions constrain the variation between w^k and w^{k+1}, thus making it easier to understand the expected properties of the solution (w^{k+1}) and to compare different approaches. Our idea is rooted on the so-called Moreau-Yosida (MY) approximation of a function [1], that we apply to the loss f, and it is motivated by evident analogies between the stationary points defining the MY approximation and update rules in the form of Eq. (1). It turns out that directly dealing with such approximation is not enough to recover the update scheme of Eq. (1), thus we investigate the first-order Taylor expansion of f in the MY approximation to devise a related optimization problem whose solution is indeed equivalent to Eq. (1) under certain regularity conditions. Beyond the benefits introduced by the interpretability of the proposed unified view, it is important to remark that our main goal is to provide researches with a formulation that more easily opens to the development of novel, more informed, optimizers. A related approach is well-known and exploited in the specific context of the online learning community [4], still not widely known to a wider ML audience.

In order to emphasize the usefulness of the proposed uniform view, we consider a case-study in which data is continuously streamed and learning proceeds in a continual manner [6,12,18]. Since our contribution is theoretical, our goal is to showcase the flexibility of the MY view in injecting problem-related prior knowledge in the update scheme, while proposing powerful experimentally-validated optimizers goes beyond the scope of this paper. In particular, we exploit the MY approximation to design a novel update scheme for the weights of a neural network , modulating the strength of the updates in function of the variations of the predictions over time. In the considered setting it is widely known that it is hard to find a good trade-off between plasticity and stability [18], and we follow the assumption for which strong output variations might be associated to significant changes in the data, thus they require the network to be more plastic in order to adapt to the novel information. Differently, small variations triggers less significant updates, preserving the already learned information, thus favouring stability. We notice that while the use of a regularization based on information from the output space of a model is a well-known principle behind Manifold Regularization [2,15], here we show how it can be exploited in continual learning by direct injection in the update rule of the model parameters. Moreover, recent approaches to continual learning exploited related heuristics to improve the quality of the learning process [13], even if without a clear theoretical formalization behind them.

The paper is organized as follows (see Fig. 1): Sect. 2 is devoted to the description of the MY approximation, its properties, the uniform view of gradient-based learning, and its relationships with proximal algorithms. Section 3 revisits commonly used optimization methods in Machine Learning in the context of the proposed uniform view, describing all of them in a unique, common, framework. In Sect. 4, we present a case-study based on an output-modulated update scheme. Section 5 concludes the paper with our final discussions.

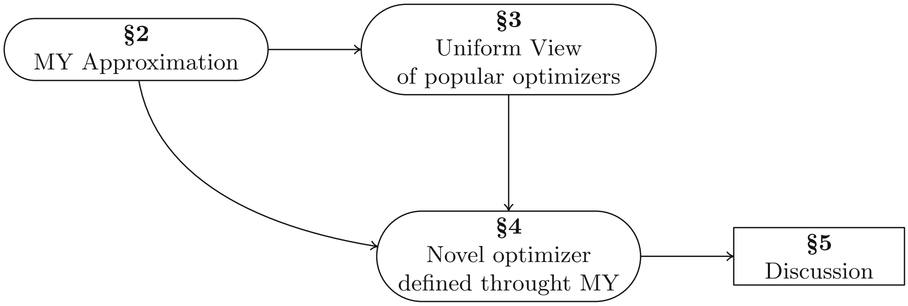

Fig. 1. Conceptual scheme of the organization of the article, that highlights the main theoretical contributions.

2 From Moreau-Yosida Approximations to Gradient-Based Learning

In this section we formally introduce the Moreau-Yosida (MY) approximation that we consider in the context of this paper, Definition 1. Then, we will exploit such notion to setup an update scheme, Eq. (9), based on a related optimization problem whose minima are described by the same recursive rule of gradient based methods. This fully connects the MY approximation and gradient-based learning, yielding a uniform view that we will exploit in the rest of the paper. For completeness, we also discuss its relations with proximal algorithms (Sect. 2.1).

Definition 1 (Moreau-Yosida Approximation). *Given a non-negative, lower semicontinuous function* $f\colon \mathbb{R}^N \to \mathbb{R}$*, such as for many implementations* *of the empirical risk in Machine Learning, the Moreau-Yosida approximation [1] of f evaluated in point w, referred to as $f_{\mathrm{MY}}(w)$, is*[1]

$$f_{\mathrm{MY}}(w) := \min_{w' \in \mathbb{R}^N} f(w') + \frac{1}{2\tau}\|w' - w\|^2, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm and $\tau > 0$.

The objective function in the optimization problem of Eq. (2) is composed of two terms: the left-most one is the value of f in w' (for example, a loss function aggregated over the available data), being w' the variable of the minimization problem, while the right-most term is a regularizer that penalizes strong variations between such w' and the target point of the approximation, i.e., w. As a result, given $w^\star$ that minimizes the objective of Eq. (2),

$$w^\star \in \arg\min_{w' \in \mathbb{R}^N} f(w') + \frac{1}{2\tau}\|w' - w\|^2, \tag{3}$$

we have that

$$f_{\mathrm{MY}}(w) = f(w^\star) + \frac{1}{2\tau}\|w^\star - w\|^2,$$

where smaller values of τ forces $w^\star$ and w to be closer (in the sense of standard Euclidean topology). See Fig. 2. In the case of hard-constrained optimization, the problem of Eq. (2) is conceptually equivalent to the following constrained optimization problem,

$$\min_{w' \in \mathbb{R}^N} f(w');$$

$$\text{s.t. } \|w' - w\| \leq \epsilon,$$

for some $\epsilon > 0$ (that increases when τ decreases), that makes evident the notion of spatial locality of the MY approximation.[2] When $f \in C^1(\mathbb{R}^N; \mathbb{R})$, the stationary points of the objective function in Eq. (2) are those $w^\star$ for which the gradient ∇f of f is null,[3] thus

$$\nabla f(w^\star) + \frac{1}{\tau}(w^\star - w) = 0, \tag{4}$$

from which we get

$$w^\star = w - \tau \nabla f(w^\star), \tag{5}$$

that is an alternative representation of Eq. (3).

[1] Notice that the minimum here exists since $f + \|\cdot - w\|^2/2$ is lower semicontinuous (because f is lower semicontinuous) and coercive since $f \geq 0$ and $\|\cdot - w\|^2/2$ is coercive (then the sublevels of $f + \|\cdot - w\|^2/2$ are contained in the sublevels of $\|\cdot - w\|^2/2$, which are compact). The existence of the minimum then follows from Weierstrass. .

[2] Of course, this formulation calls for Lagrange multiplier theory to be solved.

[3] Notice that in general in Eq. (2) such strong assumption on differentiability is not required, since the MY approximation is general and can be applied even in contexts, like functional analysis, where the notion of gradient could not be clear.

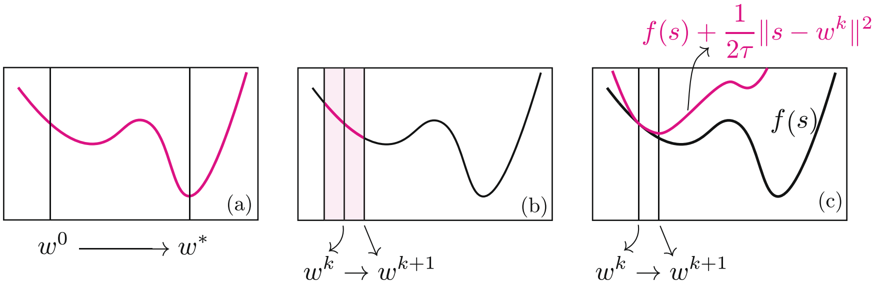
$f(s) + \frac{1}{2\tau}\|s - w^k\|^2$
$f(s)$
(a)
(b)
(c)
$w^0 \longrightarrow w^*$
$w^k \to w^{k+1}$
$w^k \to w^{k+1}$

Fig. 2. Three different optimization steps. In (a) it is represented the global optimization step from w^0 to $w^{k+1} \in \arg\min_{w'} f(w')$. In (b) the global step is made local by requiring that the minimization should be done around (magenta strip) the previous step w^k as described in Eq. (2). In (c) we graphically show the equivalence between the local step and the one induced by the MY regularization as it is described by Eq. (7). Picture (b) and (c), as it is noted in Eq. (8), are equivalent to an implicit gradient descent step.

Bridging Gradient-Based Learning. Let us consider the sequence of points $(w^k)_{k \geq 0}$ defined by the classic recursive relation of gradient-based methods with learning rate τ, as the one of Eq. (1),

$$\begin{cases} w^0 = w_0 \in \mathbb{R}^N; \\ w^{k+1} = w^k - \tau \nabla f(w^k), \quad k > 0. \end{cases} \tag{6}$$

where w_0 is a fixed initial point. We can exploit the MY approximation of f in Eq. (2) and, in particular, the expression of the argument that minimizes its optimization problem, Eq. (3), to devise another related recursive relation, replacing $w^\star$ with w^{k+1} and w with w^k,

$$\begin{cases} w^0 = w_0 \in \mathbb{R}^N; \\ w^{k+1} \in \text{arg min}_{w' \in \mathbb{R}^N} f(w') + \frac{1}{2\tau}\|w' - w^k\|^2, \quad k > 0, \end{cases} \tag{7}$$

that, accordingly to Eq. (5), is equivalent to[4]

$$\begin{cases} w^0 = w_0 \in \mathbb{R}^N; \\ w^{k+1} = w^k - \tau \nabla f(w^{k+1}), \quad k > 0. \end{cases} \tag{8}$$

It is evident that Eq. (8) describes an *implicit* scheme, from which w^{k+1} cannot be immediately computed, since it belongs to both the sides of the equation. This is different from Eq. (6), that is indeed an *explicit* scheme. For ML application it is critical to rely on explicit schemes, since solving the implicit Eq. (7) is usually unfeasible. A further step can be done to recover the *explicit* gradient descent

[4] In the reminder of the paper we will omit to write explicitly the initialization of the method $w^0 = w_0$ and we will just describe the recursion relation for $k > 0$.

method from MY approximation. The basic observation is that whenever f is smooth, at least in the neighbourhoods of w^k, we can approximate the value of f using a Taylor expansion truncated to the first order,

$$f(w') \approx f(w^k) + \nabla f(w^k) \cdot (w' - w^k),$$

where $\cdot$ is the standard scalar product in $\mathbb{R}^N$. Using this approximation in Eq. (7), and considering that $f(w^k)$ is a constant for the optimization problem reported in such an equation (thus it can be dropped), we get

$$w^{k+1} \in \arg\min_{w' \in \mathbb{R}^N} \nabla f(w^k) \cdot (w' - w^k) + \frac{1}{2\tau}\|w' - w^k\|^2, \quad k > 0. \tag{9}$$

Following the same procedure we already applied in Eq. (4), we can write down the equation of the stationary points of the newly introduced Eq. (9), that holds for $w' = w^{k+1}$, $\nabla f(w^k) + (w^{k+1} - w^k)/\tau = 0$, which ultimately yields $w^{k+1} = w^k - \tau \nabla f(w^k)$, that is exactly the *explicit* gradient descent method of Eq. (6). Equation (9) represents the MY view on gradient-based methods of Eq. (6). Figure 2 qualitatively depicts the connections between classic gradient descent and the MY view.

Uniform View. The MY scheme in Eq. (9) allows us to clearly spot the role of the variation between $w' = w^{k+1}$ and w^k, that is involved both in the squared regularizer (weighed by learning rate τ), and in the modulating term of the gradient of loss function f. Equation (9) is a powerful tool that is general enough to describe several variants of gradient-based updates, also referred to as optimizers. This is achieved by tweaking the regularizer $\|w' - w^k\|^2$ or adapting ∇f, thus clearly showing what are the expected properties of the resulting update rule, that is the topic covered in the following Sect. 3. Differently, in Sect. 4 we will use this framework to describe a case-study in which a new gradient-based method is introduced, as a proof-of-concept of the versatility of this view toward the design of novel optimizers.

2.1 Relations with Proximal Algorithms

In is interesting to formally analyze the relations of the MY view and proximal algorithms. A proximal algorithm is an algorithm that solves a *convex* optimization problem making use of the *proximal operator* of the objective function (see [17]). The simplest example of this class of methods, which is called "proximal minimization algorithm", defines a minimizing sequence that is obtained by the repeating application of the proximal operator $\mathrm{prox}_{\tau f}$ (that we will formally define shortly) to an initial point w_0:

$$\begin{cases} w^0 = w_0 \in \mathbb{R}^N; \\ w^{k+1} = \mathrm{prox}_{\tau f}(w^k), \quad k > 0. \end{cases} \tag{10}$$

Such method is closely related to the Moreau-Yosida approximation of the objective function f, once we provided and discuss the definition of the proximal operator. Let us suppose that f, in addition of being lower semicontinuous and non-negative (hence proper), is also convex (we recall that what we presented so far did not have such a requirement, introduced here only with the goal of discussing proximal algorithms). Then, the function $f(w') + 1/(2\tau)\|w' - w\|^2$ of which we are taking the minimum in Eq. (2) is strongly convex. This implies that the minimizer of Eq. (2), that we will denote as w'_w, is unique. The operator implementing the mapping $w \mapsto w'_w$ is called *proximal operator* or *proximation* (for more details, see [17,21]). In order to be able to write an explicit representation of this operator, let us recall from subdifferential calculus [21] that if f is convex, and if g is C^1 near w, being w a point that minimizes $f + g$, then

$$-\nabla g(w) \in \partial f(w),$$

where ∂f is the subdifferential of f. If we apply this result to $f(w')+1/(2\tau)\|w' - w\|^2$ with $g(w') = 1/(2\tau)\|w' - w\|^2$ we immediately have

$$\partial f(w'_w) + \frac{1}{\tau}(w'_w - w) \ni 0,$$

which is the relation that uniquely characterizes w'_w and that, in particular, guarantees that the operator $\mathrm{prox}_{\tau f}$, implemented as

$$\mathrm{prox}_{\tau f}(w) := (I + \tau\partial f)^{-1}(w) \equiv w'_w,$$

is uniquely defined. It should now be apparent that the proximal operator and the MY approximation of Definition 1 are very closely related. For instance, when f is convex we can write

$$f_{\mathrm{MY}}(w) = f(\mathrm{prox}_{\tau f}(w)) + \frac{1}{2\tau}\|\,\mathrm{prox}_{\tau f}(w) - w\|^2,$$

and we can relate the gradient[5] of f_{MY} with the proximal operator as follows,

$$\mathrm{prox}_{\tau f}(w) = w - \tau\nabla f_{\mathrm{MY}}(w).$$

Finally we notice that in the convex case the arg min in Eq. (7) becomes a singleton and hence the algorithm in Eq. (7) coincides with the one in Eq. (10). For completeness, we mention that there exists scientific literature that studies proximal algorithms also in the non-convex case [9,11].

3 Moreau-Yosida View of Popular Optimizers

In this section we will discuss how the MY view of Eq. (9) is general enough to can describe several existing optimizers widely employed in the ML community. In particular, we will consider the case of Stochastic Gradient Descent (SGD)

[5] It is indeed a standard result that under the convexity assumption f_{MY} is differentiable.

[20], Heavy Ball method (gradient descent with momentum) [19], AdaGrad [7], RMSprop [22], and ADAM [10].

SGD. The largely popular Stochastic Gradient Descent method [20] is based on a stochastic objective that involves a fraction of the available data at each iteration. We use it as starting example to introduce further notation that will be helpful in the following, being it a pretty straightforward case to cast in the MY view. We consider a training set with n samples, and we assume that $f = (1/n)\sum_{i=1}^{n} f_i$, being f_i the loss evaluated on the i-th example. At each iteration of gradient descent, a mini-batch of independently (randomly) sampled data is considered. If we indicate with I_k the set of m indices of the examples in the k-th mini-batch, and with $(1/m)\sum_{i\in I_k} f_i(w^k)$ the loss function evaluted on it, then the standard update scheme in the SGD algorithm is

$$w^{k+1} = w^k - \tau \frac{1}{m} \sum_{i\in I_k} \nabla f_i(w^k), \quad k > 0.$$

The MY view is obtained without applying any changes to the squared regularized, as expected, and by simply replacing the term $\nabla f(w^k)$ in Eq. (6) with $(1/m)\sum_{i\in I_k} \nabla f_i(w^k)$,

$$w^{k+1} \in \mathop{\arg\min}_{w'\in\mathbb{R}^N} g_k \cdot (w' - w^k) + \frac{1}{2\tau}\|w' - w^k\|^2, \quad k > 0, \tag{11}$$

where we introduced the notation $g_k := (1/m)\sum_{i\in I_k} \nabla f_i(w^k)$, that we will keep using in the following, since it allows us to be general enough to go back to classic full-batch gradient descent by setting g_k to $\nabla f(w^k)$.

Heavy Ball Method. Also known as gradient descent with momentum [19], is based on an explicit formula that involves the latest variation $w^k - w^{k-1}$ between the learnable parameters,

$$w^{k+1} = w^k - \alpha g_k + \beta(w^k - w^{k-1}), \quad k > 0, \tag{12}$$

with $\alpha, \beta > 0$. The MY view be obtained introducing a higher order regularization term and by writing the constants α and β as $\alpha = \mu\tau/(\mu + \tau)$, $\beta = \tau/(\mu + \tau)$,

$$w^{k+1} \in \mathop{\arg\min}_{w'\in\mathbb{R}^N} g_k \cdot (w' - w^k) + \frac{\|w' - w^k\|^2}{2\tau} + \frac{\|w' - 2w^k + w^{k-1}\|^2}{2\mu}, \quad k > 0.$$

Interestingly, the MY view clearly shows that, compared to vanilla gradient descent, here we also have an addition regularity term (weighed by $1/(2\mu)$), that enforces second-order information on the update step. An alternative definition of the method is sometimes given using exponential averages, that can still be cast in the MY view. Let $(a_k)_{k\geq 0}$ be a sequence, then the exponential moving average with discount factor δ of $(a_k)_{k\geq 0}$ is the sequence $(\langle a\rangle_k^\delta)_{k\geq 0}$ where the k-th term is computed as

$$\langle a\rangle_k^\delta := (1 - \delta)\sum_{j=0}^{k} \delta^{k-j} a_j. \tag{13}$$

Using such a notion, a popular alternative form of the update rule of Eq. (12) is $w^{k+1} = w^k - \tau\langle g\rangle_k^\beta$. In this case, using Eq. (13) we can rewrite the MY view as

$$w^{k+1} \in \operatorname*{arg\,min}_{y\in\mathbb{R}^N} \langle g\rangle_k^\beta \cdot (y - w^k) + \frac{1}{2\tau}\|y - w^k\|^2, \quad k > 0,$$

i.e., by replacing the gradient term g_k with its exponential moving average with decay factor β. The higher-order regularity is replaced by a smoothing operation on the gradient term g_k. We will use this alternative form also when describing the ADAM algorithm.

AdaGrad and RMSprop. Both methods (see [7] and [22]) can be cast in the MY view in a related manner. Their update schemes are defined as

$$w^{k+1} = w^k - \tau(H_k)^{-1}g_k, \quad k > 0.$$

where H_k is a diagonal matrix with the following elements in the diagonal,

$$(H_k)_{ii} := \begin{cases} \varepsilon + \sqrt{\sum_{n=0}^k (g_i^n)^2}; & \text{for AdaGrad} \\ \varepsilon + \sqrt{\sum_{n=0}^k \langle g_i^2\rangle_n^\beta}, & \text{for RMSprop,} \end{cases}$$

where $\varepsilon > 0$, $i \in \{1, 2, \dots, N\}$, g_i^k is the i-th component of the gradient and, for i fixed in $\{1, \dots, N\}$, $((g_i^2)_n)_{n\geq 0}$ is the sequence of the square of the i-th component of the gradient, where $(g_i^2)_n := (g_i^n)^2$. The MY view can be described by changing the metric with which we assess the closeness of the next point with respect to the current value of w^k. In particular, both the methods are obtained from Eq. (11) with the substitution $\|w' - w^k\|^2/(2\tau) \longrightarrow (w' - w^k) \cdot H_k(w' - w^k)/(2\tau)$, that clearly emphasizes the change of metric induced by matrix H_k when evaluating the regularization term in the variation of the learnable parameters.

ADAM. As it is well known, ADAM [10] consists in a specific way to put together gradient descent with momentum and RMSprop. In order to see exactly how it can be expressed in the regularization approach à la Moreau-Yosida, first we need to introduce a further definition of normalized exponential moving average. Let $(\langle a\rangle_k^\gamma)_{k\geq 0}$ be the exponential moving average of the sequence $(a_k)_{k\geq 0}$ with discount factor δ, then we define

$$\widehat{\langle a\rangle}_k^\delta := \frac{\langle a\rangle_k^\delta}{1 - \delta^{k+1}}, \qquad k \geq 0. \tag{14}$$

The update scheme of ADAM is then,

$$w^{k+1} = w^k - \tau(\hat{H}_k)^{-1}\widehat{\langle g\rangle}_k^{\beta_1}, \quad k > 0,$$

where $\hat{H}_k$ is a diagonal matrix with

$$(\hat{H}_k)_{ii} = \varepsilon + \sqrt{\sum_{n=0}^k \widehat{\langle g_i^2\rangle}_n^{\beta_2}}, \quad i \in \{1, \dots, N\},$$

and β_1 and β_2 are the customizable parameters of the ADAM algorithm. Following the exact same procedure we described in the case of the Heavy Ball method, using Eq. (14) we can describe ADAM in the context of the MY view as

$$w^{k+1} \in \operatorname*{arg\,min}_{y \in \mathbb{R}^N} \widehat{\langle g \rangle}_k^{\beta_1} \cdot (y - w^k) + \frac{1}{2\tau}(y - w^k) \cdot \hat{H}_k (y - w^k), \quad k > 0, \tag{15}$$

where it can be easily noticed both (*i.*) the change of metric in the regularity term due to the role of matrix $\hat{H}_k$, and (*ii.*) the smoothing operation on g_k, that, as discussed in the case of the Heavy Ball method, indirectly introduces a higher-order regularity condition on the variation of the weights.

4 Case Study: Output-Modulated Update Scheme

When introducing the MY view of Sect. 2, we discussed essentially how a gradient step is equivalent to solving a local minimization problem on a risk f around the current step w^k. As a proof-of-concept of the versatility of this view toward the design of novel optimizers, we consider a specific use-case based on neural networks.

Scenario. Let $\nu(\cdot, w)\colon \mathbb{R}^d \to \mathbb{R}^p$ be a neural network that, given a set of weights $w \in \mathbb{R}^N$, maps some input data $x \in \mathbb{R}^d$ into an output $\nu(x, w) \in \mathbb{R}^p$. The risk f is a measure of the performance of the network ν. We focus on a continual learning scenario, in which the data samples are not i.i.d., for example due to the fact that tasks and task-related data are presented in a sequential manner, where the data distribution changes over time [8,14,18]. In this case, the learning process might be plagued by several similar/related samples streamed in neighboring steps, and by abrupt changes in the data when switching from one task to another. The plasticity of the model should then adapt over time, being the network more plastic when never-seen-before data is provided, while it should not be too prone to overfitting data that was already shown several times in the last time frame. We make the hypotetis that the variations in the network outputs could indirectly tell if we are in front of data that is similar to what was just observed, or if we are switching to different data. Of course, there are no attempts to solve catastrophic forgetting issues of to propose a novel continual learning algorithm, since what we are presentin is just a case-study to support the simplicity of injecting novel knowledge in the update procedure by working on the MY view.

Output-Based Modulation. In Eq. (7) the notion of closeness in the space of parameters is given by the term $\|w' - w^k\|^2/(2\tau)$. Here the general idea is to modify this natural Euclidean topology by taking into account the way in which the changes in the parameters, induced by learning, is affected by the variations of the outputs of the model. The simplest way in which we can achieve this is by introducing a *modulation function* $\psi(\nu)$ that takes into account the outputs of the network ν in order to appropriately weigh the regularization term in Eq. (7);

what we are proposing hence is to study variants of a gradient descent method that are obtained thought the following transformation of the "distance" term:

$$\frac{1}{2\tau}\|w' - w^k\|^2 \longrightarrow \frac{\psi(\nu)}{2\tau}\|w' - w^k\|^2. \tag{16}$$

The MY-view, by explicitly showing the regularizer of the variation of the weights, allows us to immediately connect the modulated term with its effects in the update rules, as we are going to investigate by providing the precise form and structure of $\psi(\nu)$. Since the output of the network depends on the input data other than the value of the parameters, we also need to define a protocol with which data is introduced to the learner. At each time step, we are given a mini-batch of data of size m, with sample indices collected in the already introduced I_k, and using the same definition of g_k we presented in the case of SDG, i.e., $g_k = \frac{1}{m}\sum_{i\in I_k} \nabla f_i(w^k)$. We are now in position to discuss our proposal for the modulation function ψ, that we select to be

$$\psi(\nu) := \frac{1}{\gamma_k}\left\| \frac{1}{m}\sum_{i\in I_k} \nu(x_i, w^k) - \frac{1}{m}\sum_{j\in I_{k-1}} \nu(x_j, w^{k-1})\right\|^{-2}, \tag{17}$$

being $\gamma_k > 0$ a normalization factor that can eventually vary over time (that is the reason for its subscript). When the average outputs of the network produced at step $k-1$ are similar to the ones at time k, then $\psi(\nu)$ is small, and vice-versa. The MY view is then

$$\begin{aligned} w^{k+1} \in \, & \operatorname*{arg\,min}_{w'\in\mathbb{R}^N} g_k \cdot (w' - w^k) \\ & + \frac{1}{2\tau}\frac{\|w' - w^k\|^2}{\gamma_k \left\| \frac{1}{m}\sum_{i\in I_k} \nu(x_i, w^k) - \frac{1}{m}\sum_{j\in I_{k-1}} \nu(x_j, w^{k-1})\right\|^2}, \quad k > 0, \end{aligned} \tag{18}$$

from which it is easy to compute the stationarity condition of the objective function to get

$$w^{k+1} = w^k - \tau\gamma_k \left\| \frac{1}{m}\sum_{i\in I_k} \nu(x_i, w^k) - \frac{1}{m}\sum_{j\in I_{k-1}} \nu(x_j, w^{k-1})\right\|^2 g_k,$$

for $k > 0$. The last equation shows the update rule of our newly designed optimizer, which embeds our knowledge/intuitions on the learning problem at hand. The whole process we followed in devising it, shows how the such knowledge was initially injected into the MY view having a clear understanding of the regularization properties we aimed at enforcing, and only afterwards we obtained the update rule, in which the gradient of the loss is shown jointly with what comes from the proposed regularization.

5 Conclusions and Future Work

In this work we discussed how Moreau-Yosida approximation can be a powerful tool to efficiently devise optimization methods for Machine Learning. We presented a framework that offers a unified view on many existing gradient-based methods, but that also gives new insights and suggests different interpretations of them. We used the Moreau-Yosida view in the context of a continual learning use-case, devising a customized output-modulated gradient-based step, with a mechanism that inhibits updates of the parameters in presence of stationary outputs of a neural network. Future works will focus on the usage of the Moreau-Yosida view of this paper to design novel optimizers for Machine Learning, injecting well-defined and understandable properties in the optimization problem that yields the update scheme of the model parameters. While this paper is focused on the theoretical aspects of the proposed approach, in future work we plan to experimentally investigate the impact of novel optimizers designed within our uniform view, also considering modern benchmarks and learning scenarios [16].

Acknowledgment. This work has been supported by the French government, through the 3IA Coôte d'Azur, Investment in the Future, project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

References

1. Ambrosio, L., Gigli, N., Savaré, G.: Gradient Flows. In Metric Spaces and in the Space of Probability Measures. Springer, Basel (2005). https://doi.org/10.1007/b137080
2. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. **7**(85), 2399–2434 (2006). http://jmlr.org/papers/v7/belkin06a.html
3. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) Proceedings of COMPSTAT 2010, pp. 177–186. Springer, Cham (2010). https://doi.org/10.1007/978-3-7908-2604-3_16
4. Cesa-Bianchi, N., Orabona, F.: Online learning algorithms. Annual review of statistics and its application (2021)
5. Curry, H.B.: The method of steepest descent for non-linear minimization problems. Q. Appl. Math. **2**(3), 258–261 (1944)
6. Delange, M., et al.: A continual learning survey: defying forgetting in classification tasks. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 3366–3385 (2021)
7. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**(7), 2121–2159 (2011)
8. French, R.M.: Catastrophic forgetting in connectionist networks. Trends Cogn. Sci. **3**(4), 128–135 (1999)
9. Gu, B., Wang, D., Huo, Z., Huang, H.: Inexact proximal gradient methods for non-convex and non-smooth optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Le, H., Gillis, N., Patrinos, P.: Inertial block proximal methods for non-convex non-smooth optimization. In: International Conference on Machine Learning, pp. 5671–5681. PMLR (2020)
12. Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., Sanner, S.: Online continual learning in image classification: an empirical survey. Neurocomputing **469**, 28–51 (2022)

13. Marullo, S., Tiezzi, M., Betti, A., Faggi, L., Meloni, E., Melacci, S.: Continual unsupervised learning for optical flow estimation with deep networks. In: Conference on Lifelong Learning Agents, pp. 183–200. PMLR (2022)
14. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: the sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
15. Melacci, S., Belkin, M.: Laplacian support vector machines trained in the primal. J. Mach. Learn. Res. **12**(3), 1149–1184 (2011)
16. Meloni, E., Betti, A., Faggi, L., Marullo, S., Tiezzi, M., Melacci, S.: Evaluating continual learning algorithms by generating 3D virtual environments. In: Cuzzolin, F., Cannons, K., Lomonaco, V. (eds.) Continual Semi-Supervised Learning. LNCS, pp. 62–74. Springer, Cham (2022)
17. Parikh, N., Boyd, S., et al.: Proximal algorithms. Found. Trends® Optim. **1**(3), 127–239 (2014)
18. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: a review. Neural Netw. **113**, 54–71 (2019)
19. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)
20. Robbins, H., Monro, S.: A stochastic approximation method. Annals Math. Stat. **22**, 400–407 (1951)
21. Rockafellar, R.T.: Convex analysis, vol. 11. Princeton University Press, Princeton (1997)
22. Swersky, K., Hinton, G., Srivastava, N.: RMSProp: divide the gradient by a running average of its recent magnitude. Neural Networks for Machine Learning, COURSERA (2012)