A Gradient Flow Modification to Improve Learning from Differentiable Quantum Simulators

Patrick Schnell [1] **Nils Thuerey** [1]

Abstract

Propagating gradients through differentiable simulators allows to improve the training of deep learning architectures. We study an example from quantum physics that, at first glance, seems not to benefit from such gradients. Our analysis shows the problem is rooted in a mismatch between the specific form of loss functions used in quantum physics and its gradients; the gradient can vanish for non-equal states. We propose to add a scaling term to fix this problematic gradient flow and regain the benefits of gradient-based optimization. We chose two experiments on the Schroedinger equation, a prediction and a control task, to demonstrate the potential of our method.

1. Introduction

Differentiable simulators have found their way into Deep Learning (Um et al., 2020; Wang et al., 2020; Holl et al., 2020), opening the doors to numerous new training methods. They enable to propagate feedback more directly, based on linear approximations instead of on, for instance, expensive trial-and-error-like search in various reinforcement learning algorithms (Sutton & Barto, 2018). This can lead to more efficient learning demonstrated in various areas, such as mechanical systems (Toussaint et al., 2018; de Avila Belbute-Peres et al., 2018) or fluids dynamics (Schenck & Fox, 2018).

Nevertheless using gradients from differentiable simulators is not free from problems. Non-convexity or explodingly large gradients can hinder successful learning and can make it necessary to rethink how gradients flow in a given training setup. Identifying and removing these weaknesses is key to harness the potential that gradient-based optimization promises.

In this work, we investigate such a case of an ill-behaved gradient flow found in quantum mechanics where, for physical reasons, a loss other than the omnipresent mean squared error loss has to be used. This quantum loss comes with a suboptimal gradient flow, giving vanishing gradients for non-matching inputs, and therefore we propose a modification of its gradient intended to fix this shortcoming. We demonstrate the effectiveness of our method on two tasks, a prediction and control setup, on the Schroedinger equation.

2. Modifying the Gradient Flow of a Quantum Loss Function

We start with a brief look into quantum physics that will allow us to illustrate the difficulties that arise from the loss functions of quantum physics and to explain the solution we propose.

2.1. Quantum Formalism and Quantum Loss Function

Physicists model quantum states as vectors of a complex Hilbert space $\mathcal{H}$ with norm 1. Two such vectors ψ and $\psi' \in \mathcal{H}$ are physically interpretable only up to a global phase or, put differently, two vectors ψ and ψ' correspond to the same quantum state if there is a phase factor $e^{i\theta}$ such that $\psi = e^{i\theta}\psi'$. To ensure consistency all operations on quantum states are required to respect this principle. As an example, time evolution of a quantum state ψ is described by the Schroedinger equation, for a system with one spatial dimension and a potential V given as follows:

$$i\partial_t\psi = \left(-\partial_x^2 + V\right)\psi \tag{1}$$

As a further example and central to our discussion, similarity between quantum states ψ_1 and ψ_2 is measured by using the inner product $\langle \cdot, \cdot \rangle$ of $\mathcal{H}$. This allows us to define a loss function L that is invariant under global phase shifts and therefore consistent with the mathematical framework of

---

[1]School of Computation, Information and Technology, Boltzmannstrasse 3, 85748 Garching, Germany. Correspondence to: Patrick Schnell <patrick.schnell@tum.de>, Nils Thuerey <nils.thuerey@tum.de>.

Published at the Differentiable Almost Everything Workshop of the 40^{th} *International Conference on Machine Learning*, Honolulu, Hawaii, USA. July 2023.

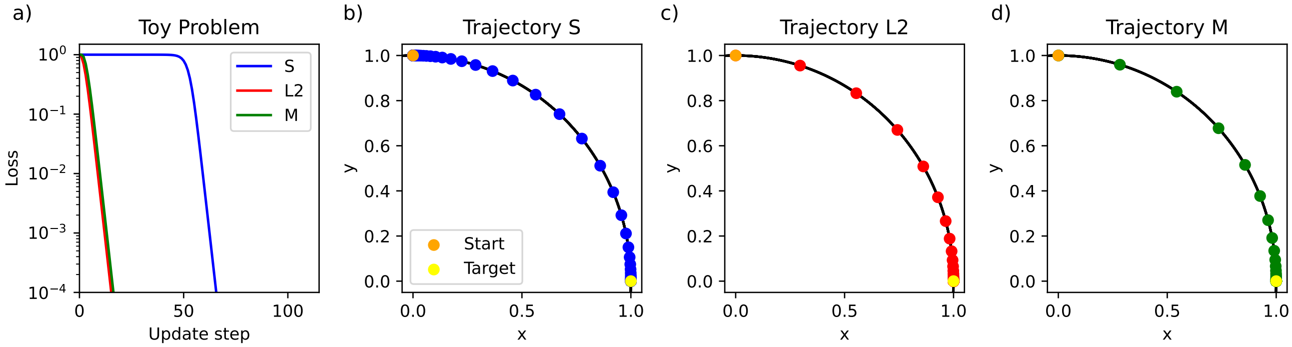

Figure 1. Toy example: a) Gradient descent loss curves for the standard gradient (S), mean squared error gradient (L2) and our modified gradient (M). b) - d) Corresponding trajectories in the x-y-plane

quantum mechanics:

$$L(\psi_1, \psi_2) = 1 - |\langle \psi_1, \psi_2 \rangle|^2 \quad (2)$$

While this loss function is physically sound and established, it reveals a weakness when viewed through the lens of back-propagation: any two quantum states ψ_1 and ψ_2 orthogonal to each other give a vanishing gradient yet their loss value is not zero. When solving a learning task such a mismatch between quantum loss and its gradient can negatively affect the learning dynamic; model updates will be dominated by data points with an already close to optimal prediction because gradients of data points with poor prediction vanish.

2.2. A Simplified Example and Standard Gradient

To find a better loss function we illustrate the behavior of the quantum loss with a simpler, visualizable example that still captures the geometric essence of the problem. Let us consider normalized vectors on $\mathbb{R}^2$. We define $v(\theta) = (\cos(\theta), \sin(\theta))^T$, a target state $w = (1, 0)^T$ and the analogon to the quantum loss (2) together with its gradient G_S, the standard gradient:

$$L(\theta) = 1 - \langle v(\theta), w \rangle^2 \quad (3)$$

$$G_S = \nabla L \quad (4)$$

For $\theta = \pi/2$ the vectors v and w are orthogonal and the gradient of L vanishes, the same problem as for the quantum loss. We initialize θ to be slightly smaller than $\pi/2$ and visualize the optimization dynamics with gradient descent in Figure 1a and b. What we see is that the gradient descent steps can only slowly escape from the local maximum around $\pi/2$ and therefore the loss curve is extremely flat at the beginning.

2.3. Mean Squared Error Gradient

As a first step towards improvement, it appears natural to consider the gradient G_{L2} that belongs to the mean squared error or L2 loss.

$$G_{L2}(\theta) = (v(\theta) - w)\frac{\partial v(\theta)}{\partial \theta} \quad (5)$$

It is the simplest loss for optimization; gradients are a measure of how much v and w differ and, contrary to the quantum loss, do not vanish if v and w are orthogonal. However, using it in a quantum physics task is questionable as it is not invariant under global phase shifts required by quantum states. Nevertheless we will consider this loss in the toy example since we deal with a real vector space here. Figure 1a and c show that the absence of the problematic gradients enables fast progress also in the beginning.

2.4. Modified Gradient

As a final step, we modify the gradient of the inner product losses (2) and (3) directly to counteract the negative effects of the inner product. For this we add an additional factor to the standard gradient G_S, giving us the modified gradient G_M:

$$G_M = \frac{\nabla L}{\sqrt{1 - L}} \quad (6)$$

The scaling of this expression is chosen to cancel exactly the term responsible for the vanishing gradients and it is completely in tune with the quantum formalism since it

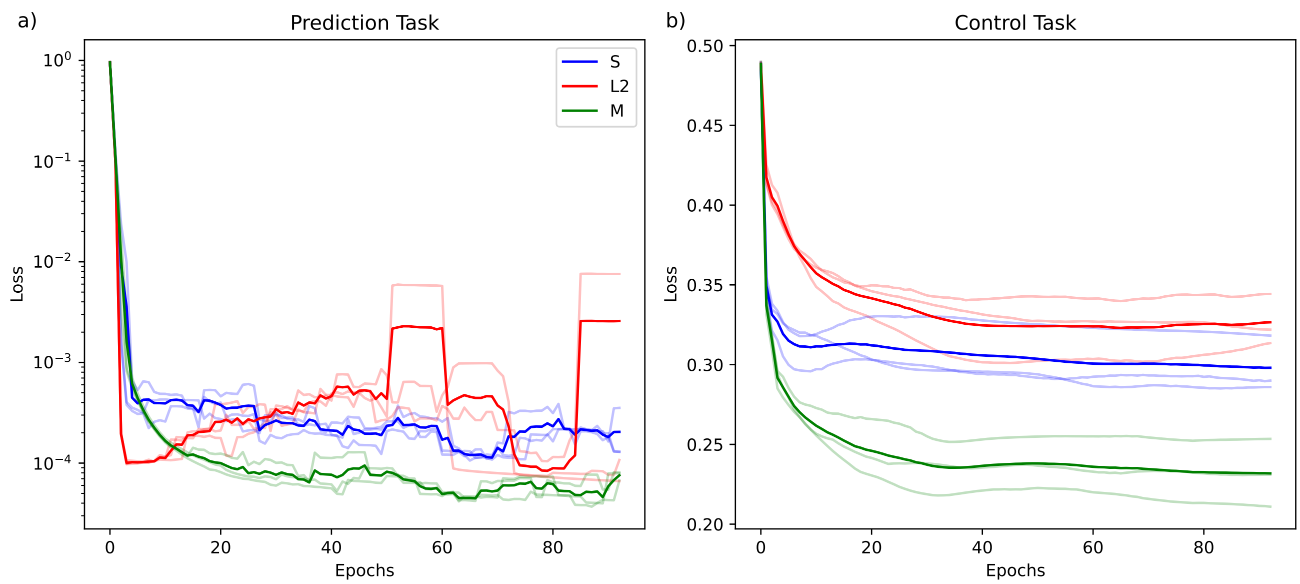

Figure 2. Quantum simulations: Adam loss curves for the standard gradient (S), mean squared error gradient (L2) and our modified gradient (M) for a) prediction task and b) control task. The plots show three training curves for each method (transparent colors) and their mean curve (opaque colors). To increase visual quality we supressed the spikey behavior by smoothing loss curves over ten subsequent epochs in both plots.

depends only on L, a quantity invariant under global phase shifts. In the toy example, Figure 1a and d show that this gradient also successfully overcomes the difficulties at the beginning. With these three types of gradients, we move to the actual quantum examples to see if we receive a similar picture.

3. Experiments

For our experiments, we implement a numerical simulation of equation (1) (Borzì et al., 2017), using a uniform one-dimensional grid of 20 points with Dirichlet boundary conditions and a modified Crank-Nicolson scheme (Winckel et al., 2009) with a time step of 0.2. We consider two types of tasks, prediction and control, which we evaluate by using the quantum loss (2) but train differently using the three types of gradients obtained from (4), (5) and (6). For the network architecture, we use in both tasks a fully-connected network with 2 hidden layers and a total of about 10000 parameters.

3.1. Prediction Task

In this type of task, we give a quantum state as input to a neural network and train it to return the corresponding quantum state at later time, i.e. the network learns what the quantum simulator does. For this we generate a set of 100-step time trajectories of 100 randomly initialized quantum states and use every state-next-state pair to build a data set for training. For optimization we use Adam with learning rate 0.005 and a mini batch size of 200. For each method we start three runs with different initializations and show these training curves together with their mean curve in Figure 2a.

We see that the standard gradient (blue) achieves only a suboptimal result, which we consider to be a consequence of the suboptimal gradients. Using the mean squared error gradient (red), physically questionable but mathematically practical, minimizes the loss to a similar level with notably fast progress especially in the beginning. Our modified gradient (green) behaves best: the usage of better and physically consistent gradients reflects itself in overall less spikey curves and loss values of about a factor of 2 better than those of the other methods.

3.2. Control Task

As a second example, we consider the task of manipulating a quantum system by external action to steer it into a given configuration. In contrast to the prediction task, we now require differentiability of the simulator to effectively compute solutions of this inverse problem. For this we solve for a time-dependent function $u(t)$, introduced via the potential V into the Schroedinger equation (1). In our numerical approximation we use 50 simulator time steps through which gradients are propagated in order to find a control signal. We set up the learning problem by generating 100 initial states and 1 final state, resulting in 100 different versions of such a control task. The network learns to predict for each initial state the corresponding signal $u(t)$ that brings the initial state into the target state. For optimization we again use Adam with learning rate 0.005 and a mini batch size of 10.

As before we conduct three runs for each method and show the results in Figure 2b. In this experiment the mean squared error gradient (red) yields the worst results, next is the standard gradient (blue), and our modified gradient (green) finds the best solution. Overall the learning curves look more stable than for the other task. These results again support our picture that incorporating optimization principles in a physically plausible way leads to better learning performance.

4. Related Work

Optimization

Optimization is a wide field with an abundance of algorithms (Ye et al., 2019). Minimizing loss functions of training setups with differentiable simulators can be challenging. Our work presents a simple adjustment that improves the training behavior. Besides standard deep learning methods (Kingma & Ba, 2015), there exist also specialized methods for optimizing learning setups with differentiable simulators (Holl et al., 2022; Schnell et al., 2022).

Incorporating Differentiable Models

Many works in deep learning involve differentiable formulations of, for instance, discrete operations (Petersen et al., 2022), rendering (Kato et al., 2020), and especially physics simulators. Examples are found in robotics (Toussaint et al., 2018), rigid bodies (de Avila Belbute-Peres et al., 2018), molecular dynamics (Wang et al., 2020), cloth models (Liang et al., 2019), fluid dynamics (Schenck & Fox, 2018). The considered tasks involve reconstruction (Holl et al., 2022), numerical error correction (Um et al., 2020) or control (Holl et al., 2020). Including a differentiable simulator means also incorporating physical principles, but is not the only way to do this. In a different spirit, this can also be done by using non-standard loss functions (Raissi et al., 2019; Tompson et al., 2017). On the technical side, various software frameworks are available for the efficient implementation of differentiable models (Hu et al., 2020; Holl et al., 2020; Schoenholz & Cubuk, 2019).

5. Discussion

Our work serves as an example of how rethinking gradient flows can improve training in deep learning tasks that involve differentiable simulators. We designed our method to counteract weaknesses of the quantum physical loss function but to still be in tune with physical principles. In two experiments we demonstrated that upholding both of these ideas achieves better results compared to the other two methods that integrate only one of them.

Our work can be extended in several ways. On the physics side, investigating more complex quantum systems such as spin systems or multiparticle systems would present a case with more complex interactions. On the learning side, more sophisticated learning formulations such as an actor-critic setup would offer an interesting opportunity for further studies.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG - German Research Foundation), Project-ID: 195170736-TRR109.

References

Borzì, A., Ciaramella, G., and Sprengel, M. *Formulation and Numerical Solution of Quantum Control Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974843.

de Avila Belbute-Peres, F., Smith, K., Allen, K., Tenenbaum, J., and Kolter, J. Z. End-to-end differentiable physics for learning and control. In *Advances in Neural Information Processing Systems*, 2018.

Holl, P., Koltun, V., and Thuerey, N. Learning to control pdes with differentiable physics. In *International Conference on Learning Representations (ICLR)*, 2020.

Holl, P., Koltun, V., and Thuerey, N. Scale-invariant learning by physics inversion. In *Conference on Neural Information Processing Systems*, 2022.

Hu, Y., Anderson, L., Li, T.-M., Sun, Q., Carr, N., Ragan-Kelley, J., and Durand, F. Difftaichi: Differentiable programming for physical simulation. *International Conference on Learning Representations (ICLR)*, 2020.

Kato, H., Beker, D., Morariu, M., Ando, T., Matsuoka, T., Kehl, W., and Gaidon, A. Differentiable rendering: A survey. *ArXiv*, abs/2006.12057, 2020.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Liang, J., Lin, M., and Koltun, V. Differentiable cloth simulation for inverse problems. In *Advances in Neural Information Processing Systems*, pp. 771–780, 2019.

Petersen, F., Borgelt, C., Kuehne, H., and Deussen, O. Deep differentiable logic gate networks. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=vF3WefcoePW.

Raissi, M., Perdikaris, P., and Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Schenck, C. and Fox, D. Spnets: Differentiable fluid dynamics for deep neural networks. In *Conference on Robot Learning*, pp. 317–335, 2018.

Schnell, P., Holl, P., and Thuerey, N. Half-inverse gradients for physical deep learning. *International Conference on Learning Representations (ICLR)*, 2022.

Schoenholz, S. S. and Cubuk, E. D. Jax, md: End-to-end differentiable, hardware accelerated, molecular dynamics in pure python. *arXiv:1912.04232*, 2019.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of Machine Learning Research*, pp. 3424–3433, 2017.

Toussaint, M., Allen, K., Smith, K., and Tenenbaum, J. B. Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems*, 2018.

Um, K., Brand, R., Fei, Y. R., Holl, P., and Thuerey, N. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in Neural Information Processing Systems*, 2020.

Wang, W., Axelrod, S., and Gómez-Bombarelli, R. Differentiable molecular simulations for control and learning. *arXiv:2003.00868*, 2020.

Winckel, G. v., Borzì, A., and Volkwein, S. A globalized newton method for the accurate solution of a dipole quantum control problem. *SIAM Journal on Scientific Computing*, 31(6):4176–4203, 2009. doi: 10.1137/09074961X.

Ye, N., Roosta-Khorasani, F., and Cui, T. Optimization methods for inverse problems. In *2017 MATRIX Annals*, pp. 121–140. Springer, 2019.