Fixing Implicit Derivatives: Trust-Region Based Learning of Continuous Energy Functions (Abridged)

Matteo Toso
CVSSP,
University of Surrey
m.toso@surrey.ac.uk

Neill D. F. Campbell
University of Bath
n.campbell@bath.ac.uk

Chris Russell
CVSSP, University of Surrey
and The Alan Turing Institute
crussell@turing.ac.uk

Abstract

We present a new technique for the learning of continuous energy functions that we refer to as **Wibergian Learning**. *One common approach to inverse problems is to cast them as an energy minimisation problem, where the minimum cost solution found is used as an estimator of hidden parameters. Our new approach formally characterises the dependency between weights that control the shape of the energy function, and the location of minima, by describing minima as fixed points of optimisation methods. This allows for the use of gradient-based end-to-end training to integrate deep-learning and the classical inverse problem methods. We show how our approach can be applied to obtain state-of-the-art results in the diverse applications of tracker fusion and multiview 3D reconstruction. The full paper can be found in the NeurIPS2019 proceedings [14].*

1. Introduction

Learning the ideal form of optimisation problems by differentiating through the location of their minima has gained much interest in recent years. One popular technique is implicit differentiation, among many successful applications: Lee *et al.* [9] and Finn *et al.* [4] use it for meta-learning. Samuel *et al.* [11] make use implicit differentiation to train continuous Markov Random Field Models, and Agrawal *et al.* [1] use it to embed differentiable optimisation layers in deep learning architectures. Amos and Kolter [2] make use of implicit differentiation to formulate quadratic programs solvers as a differentiable network layer, and Gould *et al.* [6] use it to formulate Deep Declarative Networks (DDNs), a deep learning model that embeds optimisation problems in networks trained end-to-end. Additional examples can be found in Wang *et al.* [15] and Gould *et al.* [5].

Despite its popularity, implicit differentiation can only be used in a limited range of problems: for strongly convex problems it is guaranteed to converge, but near flat minima it can be unstable and may induce exploding gradients.

We show how implicit differentiation can be derived as a fixed-point of the Newton-step algorithm. We propose an alternative method, based on the trust-region algorithm, that converges to the same fixed-points if the Newton-Step also converges, but is more stable and works on a wider range of problems. Our approach works for any continuous optimisation algorithm, and only requires that the cost function is well-behaved (*i.e.* smooth) near the minima. We provide a general overview of our approach and our experimental results. A detailed derivation of our method and a full description of the experiments can be found in [14].

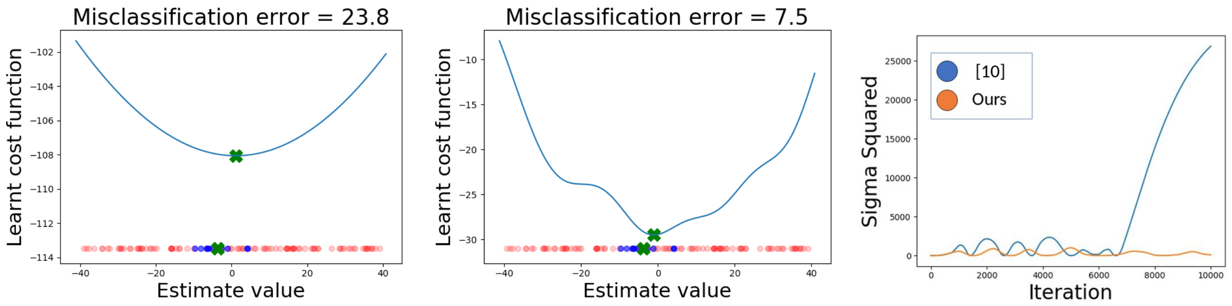
2. Formulation

Given a minimiser y^* of an arbitrary cost function $E(y; w, x)$, we seek w such that y^* is optimal with respect to some pre-existing loss function $\ell(\cdot)$ defined over the empiric distribution of training data, *i.e.*

$$\arg\min_{w} \sum_{x\in X} \ell(y^*(w); x) : y^*(w) := \arg\min_{y} E(y\,; w, x). \tag{1}$$

What makes this challenging is the decoupling of the losses on the two sides of the equation; the ideal value of y^* that minimises the loss $\ell(\cdot)$ will not, in general, be the minimiser of the energy $E(\cdot)$.

In light of this, we propose a novel local reparameterisation of y^* as a function of w, *i.e.* $y^*(w)$, and show that this allows us to compute ${}^{dy^*}\!/_{dw}$. This enables the efficient learning of w using standard methods for stochastic gradient descent and as part of an end-to-end learning framework. This reparameterisation is reminiscent of the Wiberg optimisation[16, 10], in which some variables are replaced with the analytic formula for their minimum, hence the name of our method. The key insight to our approach is that if $E(\cdot)$ is sufficiently smooth and well-behaved, the change in the solution $y^*(w) \to y^*(w')$ caused by a small perturbation of $w \to w'$ is well approximated by a single step of either Newton's method, or a more robust alternative, on y under the



(a) Sum of RBF learnt with [11]

(b) Sum of RBF learnt with our method

(c) The evolution of the parameter σ^2 for the two methods

Figure 1. Learning the kernel width of a RanSaC-like function.

new function $E(y\,; w', x)$, starting from the current solution at y^*. Here we present a general outline of our results, while a full derivation is contained in [14].

Given a local minimum y^* of the following equation

$$y^*(w) = \arg\min_{y} E(y\,; w), \tag{2}$$

we want to characterise how the location of the local minimum varies with changes in w. We drop the dependency on x for clarity of notation. Assuming the local neighbourhood about $y^*(w)$ is strongly convex, and that Newton's method given $y^*(w)$ as an input will converge in a single iteration, we can consider the second-order Taylor expansion around y. Close to a minimum of $E(\cdot)$, the expansion well models the function and the minimum of the two coincide. This leads to Newton's update rule

$$\arg\min_{y'} E(y'\,; w) \approx y - \left[\mathbf{H}E(y\,; w)\right]^{-1} \nabla E(y\,; w). \tag{3}$$

If we evaluate this at the minimum $y = y^*(w)$, we have

$$y^*(w) = y^*(w) - \left[\mathbf{H}E(y^*(w)\,; w)\right]^{-1} \nabla E(y^*(w)\,; w), \tag{4}$$

with $\nabla E(y^*(w)\,; w) = \mathbf{0}$ at optimality.

For sufficiently small updates of w, $y^*(w)$ remains in the strongly convex region about the new minimum and one iteration of Newton's method moves $y^*(w)$ directly towards the new minimum $y^*(w')$. Writing $w' = w + \Delta$, as $\Delta \to 0$ only a single iteration of Newton's method is needed to get arbitrarily close to the new minimum. That is,

$$y^*(w{+}\Delta) \approx y^*(w) - \left[\mathbf{H}E(y^*(w); w{+}\Delta)\right]^{-1} \nabla E(y^*(w); w{+}\Delta). \tag{5}$$

Writing $\mathbf{H}_{y^*}(w)$ as shorthand for the Hessian of $E(y\,; w)$ w.r.t. y at the fixed location y^*, *i.e.* $\mathbf{H}E(y^*; w)$, and $\nabla_{y^*}(w)$ as shorthand for the Jacobian of $E(y\,; w)$ with respect to y at y^*, *i.e.* $\nabla E(y^*; w)$, we rearrange and normalise the above equation.

In the limit $\Delta \to 0$, and using $\nabla_{y^*}(w) = \mathbf{0}$ by definition, this allows us to derive

$$\frac{\mathrm{d}y^*}{\mathrm{d}w} = -\mathbf{H}^{-1}_{y^*}(w) \frac{\partial \nabla_{y^*}(w)}{\partial w}. \tag{6}$$

Here we use the partial derivative to emphasise that the value of y^* on the right hand side of the equation comes from a previous iteration of Newton's method, and that it does not vary with w.

Trust-Region Based Robustness

The last equation coincides with the update rule given by implicit differentiation [11]. We can, however, see that if the function descends sharply into a flat region about a local minimum, where for all neighbourhoods, a quadratic approximation is poor, the Hessian may tend to zero around the minimum with $\mathbf{H}^{-1}_{y^*}(w)$ ill-defined. Even if $\mathbf{H}_{y^*}(w)$ is non-zero, it may become arbitrarily small leading to exploding gradients. We eliminate this instability by replacing Newton's step with the trust-region method that effectively adds a positive correction to the diagonal of the Hessian. Given $\lambda \geq 0$ and I identity matrix, we obtain the gradient as

$$\frac{\mathrm{d}y^{*\,(d)}}{\mathrm{d}w} = -\left(\mathbf{H}_{y^*}(w) + \lambda I\right)^{-1} \frac{\partial \nabla_{y^*}(w)}{\partial w}. \tag{7}$$

This can be interpreted as a damped variant of the Newton's method's gradient, as equations and (6) and (7) are solutions of the same quadratic programme, where (7) is subject to an additional constraint that $||y||_2^2 \leq k$, for some k.

Compared to the undamped formulation of (6), trust-region methods converge to a true minimum for a strictly larger class of functions making the new approach directly applicable to a wider range of problems. Moreover, as $\mathbf{H}_{y^*}(w)$ is positive semi-definite, and λI positive definite, we have $||\frac{\mathrm{d}y^{*}}{\mathrm{d}w}^{(\mathrm{d})}|| \leq \lambda^{-1} ||\frac{\partial \nabla_{y^*}(w)}{\partial w}||$, and exploding gradients can no longer be created by a single layer.

This is not just a convenience; in our experiments, we provide examples of a problem that fails to converge using [11]; but our method gives state of the art results. In practice, we fix $\lambda = 0.1$, with no additional tuning.

If the energy is quadratic, trust region analysis is unneeded, however [11] may still fail to converge for ill-posed quadratic problems. Analysis showing closed form updates

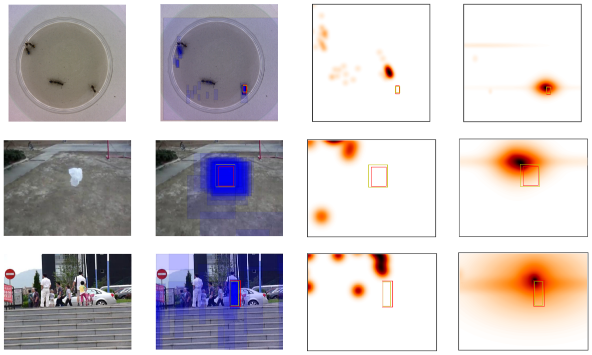


Figure 2. Learning the energy function for tracker fusion. **Left:** Input image. **Centre Left:** Overlay of 72 tracker boxes (blue), ground-truth detection (red) and our prediction (yellow). **Centre Right:** Initial energy used to predict the top left corner of the box locations before training. **Right:** Energy used to predict the same corner of box locations after training (red indicates ground-truth).

for well- and ill-posed quadratic energies guaranteed to converge are in the supplementary materials of [14].

Using the Derivative in Learning The derivatives we have specified are quite general, and, importantly, they make no assumptions about the energy minimisation technique used to obtain the optimum. In practice, approximate second order approaches such as L-BFGS [3] converge to a neighbourhood about the minimum, but the solution found does not satisfy the fixed point equation (4). In this case, a single step of (trust-region) Newton's method is required for the numeric gradients and the analytic solution to coincide. Given knowledge of how to compute the gradients, energy minimisation can be treated as a component of any end-to-end training network, which makes use of stochastic subgradient descent, and integrated directly.

3. Experiments

RanSaC as an Illustrative Example We demonstrate our approach on a simple 1-dimensional example. We consider the problem of estimating the mean of a set of 10 inliers sampled from a normal distribution $N(U[-40,40],4^2)$ in the presence of 100 outliers drawn from a broad uniform distribution $U[-40,40]$. This can be formulated as an MLESaC [13] type optimisation where the mean is estimated by minimising a one-dimensional sum of radial basis functions (RBF) centred on the samples *i.e.*: $\hat{\mu} = \arg\min_t E(t,x;\sigma) = \sum_i -\exp(-(x_i-t)^2/\sigma^2)$. We compare our approach, and that of [11], to find optimal value of σ to minimise the squared error between the estimated mean and its true value. For any choice of step-size and momentum, with probability 1 we will eventually draw a set of points that have a sufficiently small curvature about the minimum, causing an arbitrarily large step for the undamped update of [11] while our update remains bounded. This behaviour can be seen in Figure 1.

Table 1. The VOT2018 challenge; * = did not converge.

Frames Assigned at Random		Sequences Assigned at Random	
Tracker	IoU	Tracker	IoU
DLSTpp	0.530	SA_Siam_R	0.4643
FSAN	0.490	MBSiam	0.4624
SiamRPN	0.484	SiamRPN	0.4621
LSART	0.472	FSAN	0.4618
R_MCPF	0.465	LADCF	0.4601
Mean Fusion	0.238	Mean Fusion	0.2455
Median Fusion	0.428	Median Fusion	0.4458
Samuel *et al.*[11]	N/A *	Samuel *et al.*[11]	N/A *
Our Fusion	0.565	Our Fusion	0.4960

Tracker Fusion Our approach can then be used to train a model to fuse existing candidate trackers. This demonstration leverages the comprehensive evaluation work performed by the Visual Object Tracking challenge team [8]. Starting from the annotated ground-truth results and to their competition, we learn to fuse the tracking results all of 72 entries to the competition on any given frame. Our task is to predict the four corners of a bounding box given a set of candidate locations from the other trackers. This is done by modelling the upper and lower corner of the bounding boxes (see Figure 2) as a sum of 72 radial basis functions, whose standard deviation and temporal based importance weights are learned for each tracker.

In Table 1 we report the intersection over union measure both for our approach, and for the top five methods from the VOT2018 challenge on the same partitions. Our fused tracker shows state of the art results on the highly competitive VOT2018 challenge, while using implicit derivatives [11] fails to converge.

Human Pose Estimation We also consider the problem of 3D Human Pose Estimation from 2D detections. Given the multi-camera model by Tome *et al.* [12], we use our learning method to improve on their hand-tuned energy function, adapting over 6,000 hyper-parameters to a different distribution of inputs.

Full details of the model and the training process can be found in the original paper. Table 2 contains the results of evaluating our approach on the Human3.6M data-set [7]. As expected, evaluating the original model on the new data without re-tuning the parameters results in worse performances than the original ones; with our approach, we can adapt the cost function to the new data, even outperforming the original results.

Table 2. Average per joint 3D reconstruction error on Human3.6M, expressed in mm.

Multicamera	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Tome *et al.* [12] - L2	51.3	54.9	47.9	55.8	56.8	71.3	45.8	49.2	74.7	102.0	56.2	62.2	56.1	48.7	54.0	59.4
Tome *et al.* [12] - Huber	43.3	49.6	42.0	48.8	**51.1**	64.3	40.3	43.3	66.0	95.2	50.2	52.2	51.1	43.9	45.3	52.8
Ours - Baseline	85.7	90.8	79.8	87.3	107.1	94.8	78.5	87.6	102.0	100.1	95.2	85.1	92.3	85.8	87.4	91.8
Ours - L2	38.5	44.3	**39.2**	42.1	61.9	**44.4**	36.0	**38.6**	**56.7**	**65.6**	50.6	41.0	47.7	45.3	46.6	47.7
Ours - Huber	**38.2**	**42.2**	39.5	**39.1**	57.2	45.2	**34.1**	39.1	57.8	68.0	**48.7**	**39.1**	**46.7**	40.5	41.1	**46.1**

Limitations: Much like implicit derivative based learning, our approach inherits many of the advantages and disadvantages of gradient descent methods in neural nets. In particular, just as vanishing gradients, and stuck neurons, are a concern, it is possible for particular components of the energy to have too narrow a range to influence the location of minima; if this is the case, they will remain fixed. As such, it is important to use sensible initialisations and regularisers to ensure that, by default, components have a broad initial range. In general, our modified update step is most important at the start of the learning process, while the final energy functions that our algorithm converges to tends to be better behaved.

4. Conclusion

We have presented a novel approach that allows the classical energy minimisation methods of inverse problems to benefit from the end-to-end training that has been a fundamental part of the success of deep-learning. By deriving implicit differentiation as a fixed-point of the Newton-step algorithm, we were able to create a more stable alternative to implicit differentiation based upon trust-region methods. Experiments based on RanSaC and tracker fusion show explicit tasks in where our method works while implicit differentiation fails to converge.

Code is available at: `https://github.com/MatteoT90/WibergianLearning`.

References

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems 32*, pages 9562–9574. 2019.

[2] Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 2017.

[3] Richard H. Byrd, Jorge Nocedal, and Robert B. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1):129–156, Jan 1994.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 2017.

[5] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *CoRR*, 2016.

[6] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks: A new hope. Technical report, Australian National University (arXiv:1909.04866), Sep 2019.

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.

[8] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pfugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, Gustavo Fernandez, and et al. The sixth visual object tracking vot2018 challenge results, 2018.

[9] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019.

[10] Takayuki Okatani and Koichiro Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72(3):329–337, May 2007.

[11] Kegan G. G. Samuel and Marshall F. Tappen. Learning optimized MAP estimates in continuously-valued MRF models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 477–484, 2009.

[12] Denis Tome, Matteo Toso, Lourdes Agapito, and Chris Russell. Rethinking pose in 3D: Multi-stage refinement and recovery for markerless motion capture. In *2018 International Conference on 3D Vision (3DV)*, pages 474–483. IEEE, 2018.

[13] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:2000, 2000.

[14] Matteo Toso, Neill D. F. Campbell, and Chris Russell. Fixing implicit derivatives: Trust-region based learning of continuous energy functions. In *Advances in Neural Information Processing Systems 32*, pages 1476–1486. 2019.

[15] Po-Wei Wang, Priya L. Donti, Bryan Wilder, and J. Zico Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. *CoRR*, abs/1905.12149, 2019.

[16] T. Wiberg. Computation of principal components when data is missing. In *Second Symp. Computational Statistics*, pages 229–236, 1976.