Towards Understanding Gradient Approximation in Equality Constrained Deep Declarative Networks

Stephen Gould[1] Ming Xu[1] Zhiwei Xu[1] Yanbin Liu[1]

Abstract

We explore conditions for when the gradient of a deep declarative node can be approximated by ignoring constraint terms and still result in a descent direction for the global loss function. This has important practical application when training deep learning models since the approximation is often computationally much more efficient than the true gradient calculation. We provide theoretical analysis for problems with linear equality constraints and normalization constraints, and show examples where the approximation works well in practice as well as some cautionary tales for when it fails.

1. Introduction

This paper investigates certain approximations to gradient calculations for differentiable constrained optimization problems. Our focus is on continuous optimizations problems that may be embedded within deep learning models (Gould et al., 2016; Amos & Kolter, 2017; Agrawal et al., 2019; Gould et al., 2021; Blondel et al., 2022). This is in contrast to works that compute search directions for back-propagating through discrete optimization problems where a true gradient does not exist or is uninformative (i.e., zero almost everywhere), e.g., (Blondel et al., 2020; Berthet et al., 2020; Vlastelica et al., 2020; Petersen et al., 2022).

For continuous constrained optimization problems the gradient of a solution with respect to parameters of the problem (i.e., inputs) can be determined by implicit differentiation of the problem's optimality conditions (Amos & Kolter, 2017; Agrawal et al., 2019; Gould et al., 2021). One of the main computational difficulties in the presence of constraints is evaluating quantities of the form $(AH^{-1}A^\mathsf{T})^{-1}$ where A encodes first derivatives of the constraint functions and H encodes second derivatives of the objective and constraints.

Gould et al. (2022) observed that for deep models involving optimal transport—a well-known differentiable optimization problem—ignoring the constraints in the backward pass, i.e., treating the problem as if it were unconstrained, still allows the model to learn while greatly speeding the backward pass. This prompts the question explored in this paper: why and when does this gradient approximation work?

2. Gradient Approximation

In this section we develop theoretical insights for when back-propagating through a differentiable optimization problem using an approximate gradient gives a descent direction for the global loss. Full proofs can be found in the appendix.

The following result for the derivative of the solution to parametrized equality constrained optimisation problems comes from Gould et al. (2021)[Prop. 4.5].

Proposition 2.1. (Gould et al., 2021). *Consider functions $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$. Let*

$$y(x) \in \left\{ \begin{array}{ll} \text{argmin}_{u \in \mathbb{R}^m} & f(x,u) \\ \text{subject to} & h_i(x,u) = 0, \quad i = 1, \ldots, p \end{array} \right\}.$$

Assume that $y(x)$ exists, that f and $h = [h_1, \ldots, h_p]^\mathsf{T}$ are 2^{nd}-order differentiable in the neighborhood of $(x, y(x))$, and that rank$(D_Y h(x,y)) = p$. Then for H non-singular

$$Dy(x) = H^{-1}A^\mathsf{T}\big(AH^{-1}A^\mathsf{T}\big)^{-1}\big(AH^{-1}B - C\big) - H^{-1}B$$

where $A = D_Y h(x,y) \in \mathbb{R}^{p \times m}$, $B = D^2_{XY} f(x,y) - \sum_{i=1}^p \lambda_i D^2_{XY} h_i(x,y) \in \mathbb{R}^{m \times n}$, $C = D_X h(x,y) \in \mathbb{R}^{p \times n}$, $H = D^2_{YY} f(x,y) - \sum_{i=1}^p \lambda_i D^2_{YY} h_i(x,y) \in \mathbb{R}^{m \times m}$, and $\lambda \in \mathbb{R}^p$ satisfies $\lambda^\mathsf{T} A = D_Y f(x,y)$.

Symbol D denotes the total or partial (with respect to the subscripted variable) derivative operator. We refer the reader to Gould et al. (2021) for the full derivation.

Given an incoming gradient of a loss with respect to the output (i.e., solution) $\mathrm{D}\mathcal{L}(y)$, back-propagation computes the gradient of the loss with respect to the input x via the chain rule of differentiation as $\mathrm{D}\mathcal{L}(x) = \mathrm{D}\mathcal{L}(y)\mathrm{D}y(x)$. As mentioned, however, terms involving A, namely $(AH^{-1}A^\mathsf{T})^{-1}$, may present significant computational challenges. Ignoring such terms gives a computationally much simpler expression, but how well does it approximate the true gradient?

[1] Australian National University, Canberra, Australia. Correspondence to: Stephen Gould <stephen.gould@anu.edu.au>.

Published at the Differentiable Almost Everything Workshop of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. July 2023. Copyright 2023 by the author(s).

Formally, define $\widehat{H} = \mathrm{D}^2_{YY} f(x, y)$ so that $H = \widehat{H} - \sum_{i=1}^{p} \lambda_i \mathrm{D}^2_{YY} h_i(x, y)$ for a constrained problem. Let $v^{\mathsf{T}} = \mathrm{D}\mathcal{L}(y) \in \mathbb{R}^{1\times m}$ be the incoming gradient of the loss $\mathcal{L}$ with respect to output y, let $g^{\mathsf{T}} = v^{\mathsf{T}} \mathrm{D}y(x)$ be the true gradient of the loss with respect to input x and let $\widehat{g}^{\mathsf{T}} = v^{\mathsf{T}} \widehat{\mathrm{D}y}(x) = -v^{\mathsf{T}} \widehat{H}^{-1} B$ be the approximation obtained by ignoring constraints. We wish to understand when $-\widehat{g}$ is a descent direction for $\mathcal{L}$, i.e., when is

$$g^{\mathsf{T}} \widehat{g} \geq 0\,? \tag{1}$$

To simplify analysis and make progress towards some theoretical insights we will assume a single constraint function $h(u) = 0$ that is independent of x. Furthermore, we will assume that the objective function takes the special form $f(x, u) = x^{\mathsf{T}} u + \tilde{f}(u)$. An example of this is the objective function for the optimal transport problem. Together, these assumptions imply that $C = 0$ and $B = I$ in Prop. 2.1.

Substituting for $\mathrm{D}y(x)$ and $\widehat{\mathrm{D}y}(x)$ under these assumptions we have that $-\widehat{g}$ is a descent direction if and only if,

$$v^{\mathsf{T}} \left(H^{-1} - \frac{H^{-1} a a^{\mathsf{T}} H^{-1}}{a^{\mathsf{T}} H^{-1} a} \right) \widehat{H}^{-1} v \geq 0, \tag{2}$$

where we have written $a^{\mathsf{T}} = A = \mathrm{D}_Y h(y) \in \mathbb{R}^{1\times m}$ to make it clear that we are only considering problems with a single constraint.[1] We now explore two special cases.

2.1. Special Case: Linear Constraints

Consider the case of a single linear equality constraint, $a^{\mathsf{T}} u = d$. In this case we have $\mathrm{D}^2_{YY} h(u) = 0$ and therefore $H = \widehat{H}$. The condition that our approximate gradient $\widehat{\mathrm{D}y}(x) = -H^{-1}$ always leads to a descent direction is

$$\min_{w} \; w^{\mathsf{T}} \left(I - \frac{a a^{\mathsf{T}} H^{-1}}{a^{\mathsf{T}} H^{-1} a} \right) w \geq 0 \tag{3}$$

which holds if and only if[2]

$$\max_{\|w\|=1} \; w^{\mathsf{T}} \left(\frac{a a^{\mathsf{T}} H^{-1}}{a^{\mathsf{T}} H^{-1} a} \right) w \leq 1 \tag{4}$$

where we have written $w = H^{-1} v$ from Eqn. 2.

Unfortunately this is only true when $\mathrm{cond}(H) = 1$ as the following proposition shows.

Proposition 2.2. *Let $H \in \mathbb{R}^{m\times m}$ be a non-singular symmetric matrix and let a be an arbitrary vector in $\mathbb{R}^n$. Then,*

$$1 \leq \max_{\|w\|=1} \; w^{\mathsf{T}} \left(\frac{a a^{\mathsf{T}} H^{-1}}{a^{\mathsf{T}} H^{-1} a} \right) w \leq \frac{1}{2} + \frac{\mathrm{cond}(H)}{2}.$$

The lower bound is bad news. It states that, in general, we cannot guarantee that the approximation will be a descent direction for all incoming loss gradients (unless $H \propto I$). But let us not despair. This is in the worst case. The next result concerns the expected value of $g^{\mathsf{T}} \widehat{g}$ and tells us that, if $H^{-1} v$ is isotropic Gaussian distributed, then $-v^{\mathsf{T}} \widehat{\mathrm{D}y}(x)$ is a descent direction of the loss on average.

Proposition 2.3. *Let $w \sim \mathcal{N}(0, I)$. Then*

$$\boldsymbol{E} \left[w^{\mathsf{T}} \left(I - \frac{a a^{\mathsf{T}} H^{-1}}{a^{\mathsf{T}} H^{-1} a} \right) w \right] = m - 1 \geq 0.$$

The result can be extended to multiple ($1 \leq p \leq m$) linear equality constraints $Au = d$ as follows.

Proposition 2.4. *Let $w \sim \mathcal{N}(0, I)$. Then*

$$\boldsymbol{E} \left[w^{\mathsf{T}} \left(I - A^{\mathsf{T}} \left(A H^{-1} A^{\mathsf{T}} \right)^{-1} A H^{-1} \right) w \right] = m - p \geq 0.$$

This result is encouraging: for linear equality constrained problems we can expect the approximate gradient to be a descent direction. Next we turn our attention to a non-linear constraint where the story is not as straightforward.

2.2. Special Case: Normalization Constraint

We now consider the case of a single non-linear constraint, the normalization constraint, $\|u\|^2 = 1$, which occurs in many problems such as projection onto the L_2-sphere and eigen decomposition.

Once again, let $\widehat{H} = \mathrm{D}^2_{YY} f(x, y)$ and $H = \mathrm{D}^2_{YY} f(x, y) - \lambda \mathrm{D}^2_{YY} h(y) = \widehat{H} - \lambda I$. We will assume that $\widehat{H}^{-1}$ and H^{-1} exist.[3] Here we have $a \propto y$ so the general condition for the approximate gradient $\widehat{g} = -v^{\mathsf{T}} \widehat{H}^{-1}$ to be a descent direction is

$$v^{\mathsf{T}} \left(H^{-1} - \frac{H^{-1} y y^{\mathsf{T}} H^{-1}}{y^{\mathsf{T}} H^{-1} y} \right) \widehat{H}^{-1} v \geq 0. \tag{5}$$

The left-hand side represents $g^{\mathsf{T}} \widehat{g}$. As for the linear equality constrained case, we can compute its expected value.

Proposition 2.5. *Let $H^{-1} v \sim \mathcal{N}(0, I)$ and other quantities as defined above for the normalization constrained special case. Then*

$$\boldsymbol{E} \left[g^{\mathsf{T}} \widehat{g} \right] = \sum_{i=1}^{m} \frac{\lambda_i - \lambda}{\lambda_i} - \frac{y^{\mathsf{T}} \widehat{H}^{-1} y}{y^{\mathsf{T}} H^{-1} y}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ are the eigenvalues of $\widehat{H}$.

[1] We recognize that for a single constraint the quantity $a^{\mathsf{T}} H^{-1} a$ is trivial to invert and hence the approximation here offers little computational advantage. Nevertheless, as we will see, analysis from this simplification is instructive for more general settings.

[2] See Appendix A.1 for complete derivation.

[3] This implies, in particular, that λ is not an eigenvalue of $\widehat{H}$, which is clearly not true for eigen decomposition (where we also have $B \neq I$). Still, some useful insights can be gained. A similar argument may be possible using pseudo-inverses or going back to the optimality conditions and deriving gradients directly.

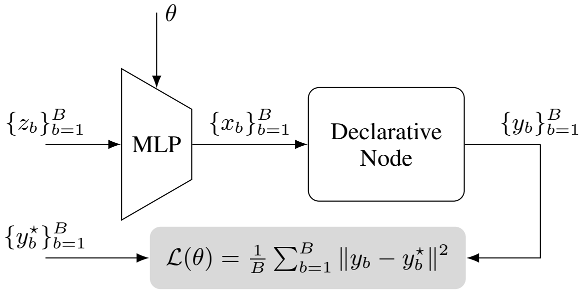


Figure 1. Common experimental setup to compare behavior of approximate and exact gradients of constrained differentiable optimisation problems in a deep declarative network. Training data is a batch of randomly sampled input-target pairs $(z_b, y_b^\star) \in \mathbb{R}^d \times \mathcal{Y}$. The input z_b passes through a multi-layer perceptron to generate the parametrization x_b for a declarative node whose output (i.e., optimal value) is y_b. Thus y_b is ultimately a function of the input z_b and network parameters θ. Training aims to adjust θ so as to minimize the square difference between output y_b and target $y_b^\star$.

The above result is for general Hessian matrix $\widehat{H}$ and arbitrary λ. Let us consider two important (non-exhaustive) cases to give concrete bounds.

Proposition 2.6. *Let $\widehat{H} \succ 0$, and let g and $\widehat{g}$ be the true and approximate gradients, respectively, as defined above.*

$$\text{(i) If } \lambda < \lambda_1, \text{ then } \mathbf{E}\left[g^\mathsf{T}\widehat{g}\right] \geq 0;$$
$$\text{(ii) If } \lambda > \lambda_m, \text{ then } \mathbf{E}\left[g^\mathsf{T}\widehat{g}\right] \leq 0,$$

where λ_1 and λ_m are the smallest and largest eigenvalues of $\widehat{H}$, respectively.

In summary, for the former case $-\widehat{g}$ is a descent direction on average, whereas for the latter case it is an ascent direction! Analogous results hold for $\widehat{H} \prec 0$.

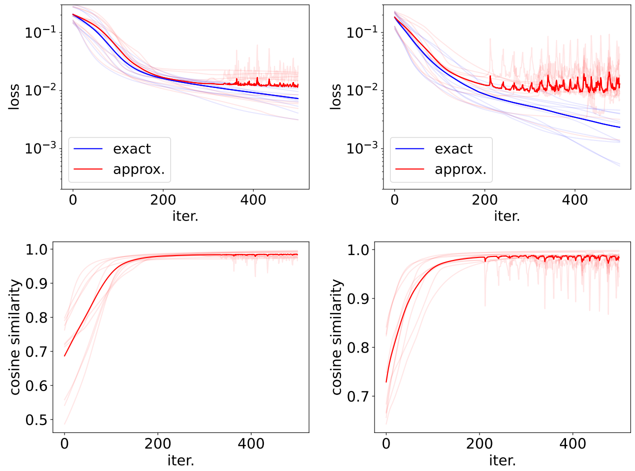
3. Examples and Experiments

In this section we experimentally validate the findings from above on three different optimization problems. Our experimental setup is depicted in Fig. 1. Briefly, a data generating network provides input for a differentiable optimization problem. We train the data generating network so that the solution of the optimization problem matches some predetermined target. Further details are provided in Appendix B.

3.1. Euclidean Projection onto L_2-sphere

Let us start with the simple problem of projecting a point $x \in \mathbb{R}^n$ onto the unit sphere,

$$y(x) \in \left\{ \begin{array}{ll} \text{argmin} & \frac{1}{2}\|u - x\|^2 \\ \text{subject to} & \|u\|_2 = 1 \end{array} \right\}. \tag{6}$$



(a) under parameterized (b) over parameterized

Figure 2. Learning curves (top) for exact and approximate gradients for projection onto the unit sphere experiments. Bottom curves show cosine similarity between approximate and exact gradients for each point on the approximate learning curve. Left versus right shows low- versus high-dimensional z_b, respectively.

Here we have closed-form solution, $y = \frac{1}{\|x\|}x$, with true and expected gradients given by

$$\mathrm{D}y(x) = \frac{1}{\|x\|}\left(I - yy^\mathsf{T}\right) \quad \text{and} \quad \widehat{\mathrm{D}y}(x) = I. \tag{7}$$

The approximate gradient always gives a descent direction (when $\mathrm{D}y(x)$ exists) since $I - yy^\mathsf{T}$ is positive semidefinite.

Experimental results in Fig. 2 confirm that the approximate gradient is always a descent direction, i.e., $g^\mathsf{T}\widehat{g} > 0$, (bottom plots), and appears to work well for learning the parameters of the MLP especially during early iterations (top plots).

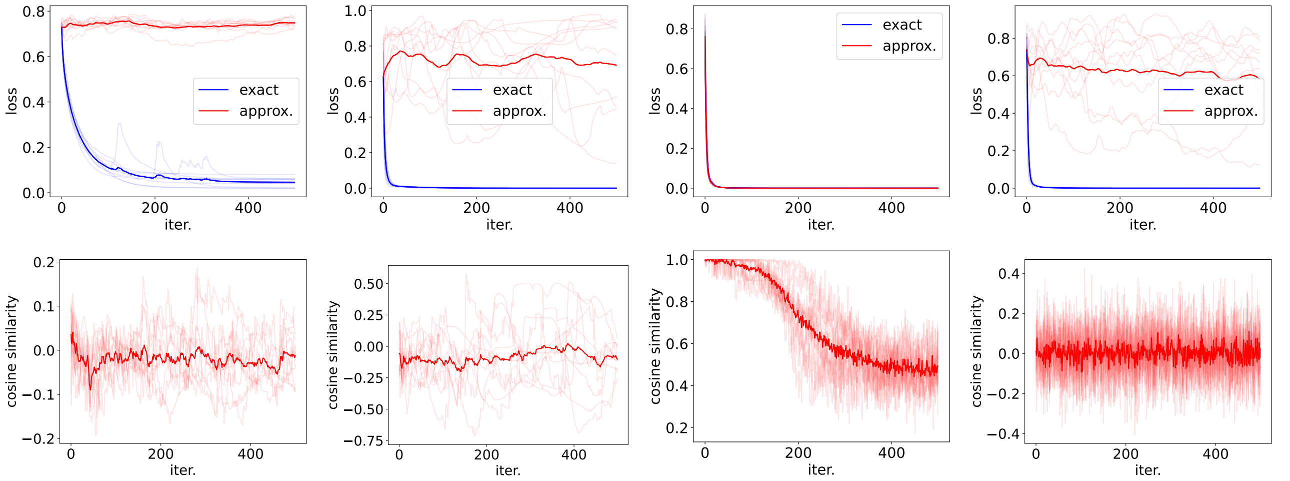
3.2. Optimal Transport

Entropy regularized optimal transport is a linear equality constrained optimization problem (Cuturi, 2013),

$$\begin{array}{ll} \text{minimize} & \langle P, M\rangle + \frac{1}{\gamma}\text{KL}(P\|rc^\mathsf{T}) \\ \text{subject to} & P1 = r \text{ and } P^\mathsf{T}1 = c, \end{array} \tag{8}$$

over variable $P \in \mathbb{R}_+^{m\times n}$, where $M \in \mathbb{R}^{m\times n}$ is an input cost matrix, r and c are positive vectors of row and column sum constraints (with $1^\mathsf{T}r = 1^\mathsf{T}c$). Hyper-parameter $\gamma > 0$ controls the strength of the regularization term.

Typical learning curves and gradient similarity per iteration is shown in Fig. 4, depicting behavior much like the previous example—the approximate gradient is always a descent direction and works especially well during the early stages of training. This is consistent with our analysis.



(a) all λ's, general X (b) largest λ, general X (c) largest λ, $X \prec 0$ (d) largest λ, rank-2 $X \succeq 0$

Figure 3. Learning curves (top) and corresponding gradient cosine similarity (bottom) for eigen decomposition experiments. For (a) the loss is applied to all eigenvectors; for (b)–(d) it is only applied to the eigenvector corresponding to the largest eigenvalue.

3.3. Eigen Decomposition

Given a real symmetric matrix $X = X^{\mathsf{T}} \in \mathbb{R}^{m \times m}$, the (unit) eigenvector associated with the largest eigenvalue of X can be found by solving the following equality constrained optimization problem (Ghojogh et al., 2019),

$$\begin{array}{ll} \text{maximize (over } u \in \mathbb{R}^m) & u^{\mathsf{T}} X u \\ \text{subject to} & u^{\mathsf{T}} u = 1. \end{array} \tag{9}$$

Here we assume that the largest eigenvalue is simple otherwise a well-defined derivative does not exist. The optimality conditions for solution $y \in \mathbb{R}^m$ are thus[4],

$$Xy - \lambda_{\max} y = 0_m \text{ and } y^{\mathsf{T}} y = 1, \tag{10}$$

which gives differentials (Magnus, 1985),

$$\mathrm{d}y = (\lambda_{\max} I - X)^{\dagger} (\mathrm{d}X) y \tag{11}$$

where † denotes pseudo-inverse. So with respect to the (i, j)-th component of X, and using symmetry, we have

$$\mathrm{D}_{X_{ij}} y(X) = -\frac{1}{2} (X - \lambda_{\max} I)^{\dagger} (y_j e_i + y_i e_j). \tag{12}$$

Ignoring the equality constraint $u^{\mathsf{T}} u = 1$ we arrive at

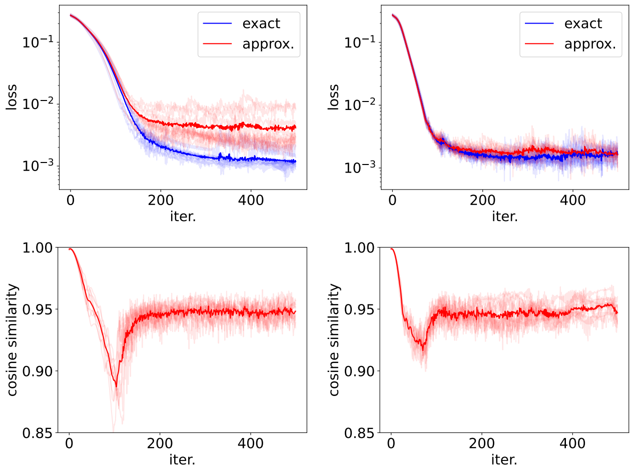
$$\widehat{\mathrm{D}_{X_{ij}} y}(X) = -\frac{1}{2} X^{\dagger} (y_j e_i + y_i e_j). \tag{13}$$

There is no computational gain here unless we need derivatives for multiple different eigenvectors and hence require multiple pseudo-inverses $(X - \lambda_k I)^{\dagger}$ for the exact gradient.

Moreover, results shown in Fig. 3 confirm our analysis that the approximation is a poor choice, and rarely a descent direction, unless y corresponds to the max. eigenvalue and all other eigenvalues are negative (equiv., the min. eigenvalue and all other eigenvalues are positive), as in Fig. 3(c).

4. Discussion

We have shown that (for certain objective functions) ignoring linear constraints gives a descent direction on average but that this does not always hold for normalization constraints. Experiments verify our analysis, and also show that even when we have a descent direction, the approximation tends to only work well in early stages of training. Whenever using approximations their behavior should be well-understood. This work is a step towards understanding of gradient approximations in differentiable optimization.



(a) under parameterized (b) over parameterized

Figure 4. Learning curves (top) and corresponding gradient cosine similarity (bottom) for optimal transport experiments.

[4] Indeed, this holds for any simple eigenvalue-eigenvector pair.

References

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, Z. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Amos, B. and Kolter, J. Z. OptNet: Differentiable optimization as a layer in neural networks. In *Proc. of the International Conference on Machine Learning (ICML)*, 2017.

Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P., and Bach, F. Learning with differentiable pertubed optimizers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. Fast differentiable sorting and ranking. In *Proc. of the International Conference on Machine Learning (ICML)*, 2020.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-Lopez, F., Pedregosa, F., and Vert, J.-P. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Ghojogh, B., Karray, F., and Crowley, M. Eigenvalue and generalized eigenvalue problems: Tutorial. Technical report, 2019.

Gould, S., Fernando, B., Cherian, A., Anderson, P., Santa Cruz, R., and Guo, E. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. Technical report, Australian National University (arXiv:1607.05447), July 2016.

Gould, S., Hartley, R., and Campbell, D. Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2021.

Gould, S., Campbell, D., Ben-Shabat, Y., Koneputugodage, C. H., and Xu, Z. Exploiting problem structure in deep declarative networks: Two case studies. In *First AAAI Workshop on Optimal Transport and Structured Data Modeling (OT-SDM)*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

Magnus, J. R. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, pp. 179–191, 1985.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017.

Petersen, F., Borgelt, C., Kuehne, H., and Deussen, O. Deep differentiable logic gate networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Seber, G. A. F. and Lee, A. J. *Linear regression analysis*. Wiley-Interscience, 2nd edition, 2003.

Vlastelica, M., Paulus, A., Musil, V., Martius, G., and Rolínek, M. Differentiation of blackbox combinatorial solvers. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2020.

A. Proofs and Derivations

A.1. Derivation of Equation 4

Consider the function $f(w) = w^\mathsf{T}\left(I - \frac{aa^\mathsf{T}H^{-1}}{a^\mathsf{T}H^{-1}a}\right)w$. If $w = 0$, then $f(w) = 0$. Otherwise,

$$f(w) \geq 0 \iff \frac{1}{\|w\|_2^2} f(w) \geq 0. \tag{14}$$

But

$$\frac{1}{\|w\|_2^2} f(w) = \frac{1}{\|w\|_2^2} w^\mathsf{T}\left(I - \frac{aa^\mathsf{T}H^{-1}}{a^\mathsf{T}H^{-1}a}\right)w = 1 - \frac{w^\mathsf{T}}{\|w\|_2}\left(\frac{aa^\mathsf{T}H^{-1}}{a^\mathsf{T}H^{-1}a}\right)\frac{w}{\|w\|_2} \tag{15}$$

Therefore,

$$\min_w w^\mathsf{T}\left(I - \frac{aa^\mathsf{T}H^{-1}}{a^\mathsf{T}H^{-1}a}\right)w \geq 0 \iff \max_{\|w\|=1} w^\mathsf{T}\left(\frac{aa^\mathsf{T}H^{-1}}{a^\mathsf{T}H^{-1}a}\right)w \leq 1. \tag{16}$$

A.2. Proof of Proposition 2.2

We begin with three useful lemmas for rank-1 quadratic forms, $f(x) = x^\mathsf{T}\left(\frac{ab^\mathsf{T}}{a^\mathsf{T}b}\right)x = x^\mathsf{T}\left(\frac{ab^\mathsf{T}+ba^\mathsf{T}}{2a^\mathsf{T}b}\right)x$ with $a^\mathsf{T}b \neq 0$.

Lemma A.1. *Let $M = ab^\mathsf{T} + ba^\mathsf{T}$. Then M has eigenvalues $\lambda_{1,2} = a^\mathsf{T}b \pm \|a\|\|b\|$ with corresponding orthonormal eigenvectors $q_{1,2} \propto \|b\|a \pm \|a\|b$.*

Proof. By direct substitution,

$$\begin{aligned}
Mq_1 &= (ab^\mathsf{T} + ba^\mathsf{T})(\|b\|a + \|a\|b) &(17)\\
&= ab^\mathsf{T}(\|b\|a + \|a\|b) + ba^\mathsf{T}(\|b\|a + \|a\|b) &(18)\\
&= (b^\mathsf{T}a + \|a\|\|b\|)\|b\|a + (\|b\|\|a\| + a^\mathsf{T}b)\|a\|b &(19)\\
&= (a^\mathsf{T}b + \|a\|\|b\|)(\|b\|a + \|a\|b) &(20)\\
&= \lambda_1 q_1 &(21)
\end{aligned}$$

and similarly for λ_2 and q_2. We can verify orthogonality of q_1 and q_2 as

$$\begin{aligned}
q_1^\mathsf{T}q_2 &= (\|b\|a + \|a\|b)^\mathsf{T}(\|b\|a - \|a\|b) &(22)\\
&= \|b\|^2\|a\|^2 - \|b\|\|a\|a^\mathsf{T}b + \|a\|\|b\|b^\mathsf{T}a - \|a\|^2\|b\|^2 &(23)\\
&= 0 &(24)
\end{aligned}$$

□

Lemma A.2. *The eigenvalue spectrum of $M = \frac{1}{2}\left(\frac{ab^\mathsf{T}+ba^\mathsf{T}}{a^\mathsf{T}b}\right)$ with $a^\mathsf{T}b \neq 0$ is*

$$\sigma\left(\frac{ab^\mathsf{T}+ba^\mathsf{T}}{2a^\mathsf{T}b}\right) = \left\{\frac{1}{2} - \frac{\|a\|\|b\|}{2|a^\mathsf{T}b|},\, 0,\, \ldots,\, 0,\, \frac{1}{2} + \frac{\|a\|\|b\|}{2|a^\mathsf{T}b|}\right\}$$

Proof. Follows from Lemma A.1 taking careful note of the sign of $a^\mathsf{T}b$. To see that all other eigenvalues are zero, note that M is a rank-2 matrix (rank-1 if a and b are linearly dependent) and so has at most two non-zero eigenvalues. □

It follows also that if a and b are linearly dependent then $\frac{1}{2}\left(\frac{ab^\mathsf{T}+ba^\mathsf{T}}{a^\mathsf{T}b}\right)$ has a single non-zero eigenvalue of 1. Moreover, for any non-orthogonal a and b, the sum of eigenvalues is equal to one.

Lemma A.3. *Let $a, b \in \mathbb{R}^n$ with $a^\mathsf{T} b \neq 0$. Then*

$$\max_{\|x\|=1} x^\mathsf{T} \left(\frac{ab^\mathsf{T}}{a^\mathsf{T} b} \right) x = \frac{1}{2} + \frac{\|a\| \|b\|}{2|a^\mathsf{T} b|}.$$

Proof. We have

$$\max_{\|x\|=1} x^\mathsf{T} \left(\frac{ab^\mathsf{T}}{a^\mathsf{T} b} \right) x = \max_{\|x\|=1} x^\mathsf{T} \left(\frac{ab^\mathsf{T} + ba^\mathsf{T}}{2a^\mathsf{T} b} \right) x = \lambda_{\text{max}} \left(\frac{ab^\mathsf{T} + ba^\mathsf{T}}{2a^\mathsf{T} b} \right) = \frac{1}{2} + \frac{\|a\| \|b\|}{2|a^\mathsf{T} b|} \tag{25}$$

by Lemma A.2. □

We are now ready to prove Prop. 2.2. From Lemma A.3 with $b = H^{-1}a$ we have

$$\max_{\|w\|=1} w^\mathsf{T} \left(\frac{aa^\mathsf{T} H^{-1}}{a^\mathsf{T} H^{-1} a} \right) w = \frac{1}{2} + \frac{\|a\| \cdot \|H^{-1}a\|}{2|a^\mathsf{T} H^{-1} a|} \tag{26}$$

By the Cauchy-Schwarz inequality $|a^\mathsf{T} H^{-1} a| \leq \|a\| \cdot \|H^{-1}a\|$ with equality if and only if a and $H^{-1}a$ are linearly dependent. This gives the lower bound,

$$\max_{\|w\|=1} w^\mathsf{T} \left(\frac{aa^\mathsf{T} H^{-1}}{a^\mathsf{T} H^{-1} a} \right) w \geq 1. \tag{27}$$

For the upper bound, observe that $|a^\mathsf{T} H^{-1} a| \geq \sigma_{\text{min}}(H^{-1}) \|a\|^2$ and $\|H^{-1}a\| \leq \sigma_{\text{max}}(H^{-1}) \|a\|$ to get

$$\frac{\|a\| \cdot \|H^{-1}a\|}{|a^\mathsf{T} H^{-1} a|} \leq \frac{\|a\| \cdot \sigma_{\text{max}}(H^{-1}) \|a\|}{\sigma_{\text{min}}(H^{-1}) \|a\|^2} = \frac{\sigma_{\text{max}}(H^{-1})}{\sigma_{\text{min}}(H^{-1})} = \text{cond}(H). \tag{28}$$

A.3. Proof of Proposition 2.3

The following general result on the expected value for a quadratic form is from Seber & Lee (2003)[Thm. 1.5, p. 9]. It can be easily proved by direct evaluation, using the cyclic property of trace, linearity of trace and expectation, and definition of variance.

Lemma A.4. (Seber & Lee, 2003). *Let $X = (x_i)$ be an $n \times 1$ vector of random variables, and let A be an $n \times n$ symmetric matrix. If $\boldsymbol{E}[X] = \mu$ and $\text{Var}(X) = \Sigma = (\sigma_{ij})$, then*

$$\boldsymbol{E}\left[x^\mathsf{T} A x\right] = \boldsymbol{tr}(A\Sigma) + \mu^\mathsf{T} A \mu.$$

The above result extends to nonsymmetric A since, $x^\mathsf{T} A x = x^\mathsf{T} \frac{1}{2}(A + A^\mathsf{T}) x$ and $\mathbf{tr}(A\Sigma) = \mathbf{tr}\big(\Sigma A^\mathsf{T}\big) = \mathbf{tr}\big(A^\mathsf{T} \Sigma\big)$ so that

$$\mathbf{tr}(A\Sigma) = \frac{1}{2} \left(\mathbf{tr}(A\Sigma) + \mathbf{tr}\big(A^\mathsf{T} \Sigma\big) \right) = \mathbf{tr}\left(\frac{1}{2}(A + A^\mathsf{T})\Sigma \right). \tag{29}$$

Now assuming the quantity w in

$$g^\mathsf{T} \widehat{g} = w^\mathsf{T} \left(I - \frac{aa^\mathsf{T} H^{-1}}{a^\mathsf{T} H^{-1} a} \right) w \tag{30}$$

is isotropic Gaussian distributed, then

$$\mathbf{E}_{w \sim \mathcal{N}(0,I)} \left[w^\mathsf{T} \left(I - \frac{aa^\mathsf{T} H^{-1}}{a^\mathsf{T} H^{-1} a} \right) w \right] = \mathbf{tr}\left(I - \frac{aa^\mathsf{T} H^{-1}}{a^\mathsf{T} H^{-1} a} \right) \tag{31}$$

$$= \mathbf{tr}(I) - \mathbf{tr}\left(\frac{aa^\mathsf{T} H^{-1}}{a^\mathsf{T} H^{-1} a} \right) \tag{32}$$

$$= m - 1 \tag{33}$$

where the first line is from Lemma A.4, the second line is by linearity of trace, and the last line is by the trace of a matrix equalling the sum of its eigenvalues, which is m for the identity and one for the second term by Lemma A.2.

A.4. Proof of Proposition 2.4

From Lemma A.4 we have

$$\begin{aligned}
\mathbf{E}_{w\sim\mathcal{N}(0,I)}\Big[w^{\mathsf{T}}\Big(I - A^{\mathsf{T}}\big(AH^{-1}A^{\mathsf{T}}\big)^{-1}AH^{-1}\Big)w\Big] &= \mathbf{tr}\Big(I - A^{\mathsf{T}}\big(AH^{-1}A^{\mathsf{T}}\big)^{-1}AH^{-1}\Big) && (34)\\
&= \mathbf{tr}(I) - \mathbf{tr}\Big(A^{\mathsf{T}}\left(AH^{-1}A^{\mathsf{T}}\right)^{-1}AH^{-1}\Big) && (35)\\
&= \mathbf{tr}(I_m) - \mathbf{tr}\Big(\big(AH^{-1}A^{\mathsf{T}}\big)^{-1}AH^{-1}A^{\mathsf{T}}\Big) && (36)\\
&= \mathbf{tr}(I_m) - \mathbf{tr}(I_p) && (37)\\
&= m - p && (38)
\end{aligned}$$

where we have used the cyclic property of trace on the third line, and that A is full rank and $H \succ 0$ on the fourth line.

A.5. Proof of Proposition 2.5

Let $\widehat{H} = Q\Lambda Q^{\mathsf{T}}$ where $\Lambda = \mathbf{diag}\,(\lambda_1, \ldots, \lambda_m)$ is a diagonal matrix containing the eigenvalues of $\widehat{H}$ arranged in ascending order. Then $H = Q(\Lambda - \lambda I)Q^{\mathsf{T}}$. Since $\widehat{H}$ and H share the same eigenvectors they are simultaneously diagonalizable and so commute. Therefore

$$\begin{aligned}
v^{\mathsf{T}}\left(H^{-1} - \frac{H^{-1}yy^{\mathsf{T}}H^{-1}}{y^{\mathsf{T}}H^{-1}y}\right)\widehat{H}^{-1}v &= v^{\mathsf{T}}\left(H^{-1}\widehat{H}^{-1} - \frac{H^{-1}yy^{\mathsf{T}}H^{-1}\widehat{H}^{-1}}{y^{\mathsf{T}}H^{-1}y}\right)v && (39)\\
&= v^{\mathsf{T}}\left(H^{-1}\widehat{H}^{-1}HH^{-1} - \frac{H^{-1}yy^{\mathsf{T}}\widehat{H}^{-1}H^{-1}}{y^{\mathsf{T}}H^{-1}y}\right)v && (40)\\
&= v^{\mathsf{T}}H^{-1}\left(\widehat{H}^{-1}H - \frac{yy^{\mathsf{T}}\widehat{H}^{-1}}{y^{\mathsf{T}}H^{-1}y}\right)H^{-1}v && (41)\\
&= w^{\mathsf{T}}\left(\widehat{H}^{-1}H - \frac{yy^{\mathsf{T}}\widehat{H}^{-1}}{y^{\mathsf{T}}H^{-1}y}\right)w && (42)
\end{aligned}$$

where in the second line we have used that $\widehat{H}$ and H commute in the second term, then factored out H^{-1} in the third line, and substituted $w = H^{-1}v$ in the last line. As for the linear equality constrained case, we can compute expectations,

$$\begin{aligned}
\mathbf{E}_{w\sim\mathcal{N}(0,I)}\left[g^{\mathsf{T}}\widehat{g}\right] &= \mathbf{E}_{w\sim\mathcal{N}(0,I)}\left[w^{\mathsf{T}}\left(\widehat{H}^{-1}H - \frac{yy^{\mathsf{T}}\widehat{H}^{-1}}{y^{\mathsf{T}}H^{-1}y}\right)w\right] && (43)\\
&= \mathbf{tr}\left(\widehat{H}^{-1}H - \frac{yy^{\mathsf{T}}\widehat{H}^{-1}}{y^{\mathsf{T}}H^{-1}y}\right) && (44)\\
&= \mathbf{tr}\Big(\widehat{H}^{-1}H\Big) - \mathbf{tr}\left(\frac{yy^{\mathsf{T}}\widehat{H}^{-1}}{y^{\mathsf{T}}H^{-1}y}\right) && (45)\\
&= \mathbf{tr}\big(Q\Lambda^{-1}Q^{\mathsf{T}}Q(\Lambda - \lambda I)Q^{\mathsf{T}}\big) - \mathbf{tr}\left(\frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}H^{-1}y}\right) && (46)\\
&= \mathbf{tr}\big(\Lambda^{-1}(\Lambda - \lambda I)\big) - \mathbf{tr}\left(\frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}H^{-1}y}\right) && (47)\\
&= \sum_{i=1}^{m}\frac{\lambda_i - \lambda}{\lambda_i} - \frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}H^{-1}y} && (48)
\end{aligned}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ are the eigenvalues of $\widehat{H}$ and $\lambda_1 - \lambda \leq \lambda_2 - \lambda \leq \cdots \leq \lambda_m - \lambda$ are the eigenvalues of H.

A.6. Proof of Proposition 2.6

We consider each case.

Case 1 ($\widehat{H} \succ 0$, $\lambda < \lambda_1$). In this case H is positive definite. Write $\widehat{H} = Q\Lambda Q^{\mathsf{T}}$ and $H = Q(\Lambda - \lambda I)Q^{\mathsf{T}}$. Then

$$
\begin{aligned}
\min_{\|y\|=1} -\frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}H^{-1}y} &= -\max_{\|y\|=1} \frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}H^{-1}y} && (49)\\
&= -\max_{\|y\|=1} y^{\mathsf{T}}H^{1/2}\widehat{H}^{-1}H^{1/2}y && (50)\\
&= -\max_{\|y\|=1} y^{\mathsf{T}}\left(Q(\Lambda - \lambda I)^{1/2}Q^{\mathsf{T}}\right)Q\Lambda^{-1}Q^{\mathsf{T}}\left(Q(\Lambda - \lambda I)^{1/2}Q^{\mathsf{T}}\right)y && (51)\\
&= -\max_{\|y\|=1} y^{\mathsf{T}}(\Lambda - \lambda I)^{1/2}\Lambda^{-1}(\Lambda - \lambda I)^{1/2}y && (52)\\
&= -\max_{i=1,\ldots,m} \left\{\frac{\lambda_i - \lambda}{\lambda_i}\right\} && (53)\\
&= \begin{cases} -\frac{\lambda_m - \lambda}{\lambda_m}, & \text{if } \lambda \geq 0 \\ -\frac{\lambda_1 - \lambda}{\lambda_1}, & \text{otherwise.} \end{cases} && (54)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\mathbf{E}_{w \sim \mathcal{N}(0,I)}\left[g^{\mathsf{T}}\widehat{g}\right] &\geq \sum_{i=1}^{m} \frac{\lambda_i - \lambda}{\lambda_i} - \max_{i=1,\ldots,m}\left\{\frac{\lambda_i - \lambda}{\lambda_i}\right\} && (55)\\
&= \begin{cases} \sum_{i=1}^{m-1} \frac{\lambda_i - \lambda}{\lambda_i}, & \text{if } \lambda \geq 0 \\ \sum_{i=2}^{m} \frac{\lambda_i - \lambda}{\lambda_i}, & \text{otherwise} \end{cases} && (56)\\
&\geq 0 && (57)
\end{aligned}
$$

since each $\frac{\lambda_i - \lambda}{\lambda_i}$ is positive.

Case 2 ($\widehat{H} \succ 0$, $\lambda > \lambda_m$). In this case H is negative definite, and we have

$$
\begin{aligned}
\max_{\|y\|=1} -\frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}H^{-1}y} &= \max_{\|y\|=1} \frac{y^{\mathsf{T}}\widehat{H}^{-1}y}{y^{\mathsf{T}}(-H^{-1})y} && (58)\\
&= \max_{\|y\|=1} y^{\mathsf{T}}(-H)^{1/2}\widehat{H}^{-1}(-H)^{1/2}y && (59)\\
&= \max_{\|y\|=1} y^{\mathsf{T}}\left(Q(\lambda I - \Lambda)^{1/2}Q^{\mathsf{T}}\right)Q\Lambda^{-1}Q^{\mathsf{T}}\left(Q(\lambda I - \Lambda)^{1/2}Q^{\mathsf{T}}\right)y && (60)\\
&= \max_{\|y\|=1} y^{\mathsf{T}}(\lambda I - \Lambda)^{1/2}\Lambda^{-1}(\lambda I - \Lambda)^{1/2}y && (61)\\
&= \max_{i=1,\ldots,m}\left\{\frac{\lambda - \lambda_i}{\lambda_i}\right\} && (62)\\
&= \frac{\lambda - \lambda_1}{\lambda_1}. && (63)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\mathbf{E}_{w \sim \mathcal{N}(0,I)}\left[g^{\mathsf{T}}\widehat{g}\right] &\leq \sum_{i=1}^{m} \frac{\lambda_i - \lambda}{\lambda_i} + \frac{\lambda - \lambda_1}{\lambda_1} && (64)\\
&= \sum_{i=2}^{m} \frac{\lambda_i - \lambda}{\lambda_i} && (65)\\
&\leq 0 && (66)
\end{aligned}
$$

since each $\frac{\lambda_i - \lambda}{\lambda_i}$ is negative.

B. Experimental Details

We follow the same experimental procedure for all three example optimization problems—Euclidean projection onto the unit sphere, optimal transport, and eigen decomposition—as depicted in Fig. 1. The network architecture consists of a three-layer multi-layer perceptron (MLP) with ReLU activation layers. The MLP maps d-dimensional raw input data z_b into the input for a deep declarative node (also known as a differentiable optimization layer) denoted x_b. This is an n-dimensional vector for Euclidean projection, an m-by-n dimensional matrix for optimal transport (for simplicity we set $n = m$), and an m-by-m real symmetric matrix for eigen decomposition. Using this input the declarative node solves the associated optimization problem, outputting the solution y_b corresponding to z_b. As such the output of the network y_b can be thought of as a function of input x_b and MLP parameters θ.

A single batch of ten input-target pairs $\{(z_b, y_b^\star)\}_{b=1}^{10}$ is randomly generated and used as training data for the parameters θ of the MLP. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 10^{-3} and run for a total of 500 iterations. The loss function of Euclidean projection and optimal transport is the mean-square-error,

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{b=1}^{B} \|y_b(z_b, \theta) - y_b^\star\|_2^2 \tag{67}$$

whereas for eigen decomposition we use the mean absolute-value of the cosine similarity,

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{b=1}^{B} |y_b(z_b, \theta)^\mathsf{T} y_b^\star|. \tag{68}$$

The latter allows us to seamlessly deal with the sign ambiguity of eigenvectors (i.e., if q is a unit eigenvector then so is $-q$).

We run five repeats of each experiment, randomly resampling the training data for each run. Learning curve plots show the loss function versus training iteration for each individual run (light) and the average over all five runs (dark).

For Euclidean projection and optimal transport we include two different input settings, $d = 5$ and $d = 100$, which we denote as under and over parameterized in the plots. This reflects the fact that it is easier to learn a mapping from high-dimensional input z_b to arbitrary target $y_b^\star$ than for low-dimensional z_b. We set $m = 10$ for both problems. For eigen decomposition we experiment with four different settings: (a) a loss on all eigenvectors y_k for a general input matrix X, (b) a loss on just the eigenvector corresponding to the maximum eigenvalue for general input matrix X, (c) the same loss but with negative definite input matrix, and (d) the same loss but with a rank-2 positive definite input matrix. In all cases we set $d = 5$ and $m = 10$.

In addition to the learning curves we plot the cosine-similarity between the true and approximate gradient of the loss with respect to the input of the declarative node x_b. This is done for each point on the learning curve for approximate gradient. A value greater than zero indicates that the corresponding approximate gradient is a descent direction with respect to x_b. We note that this does not necessarily mean that it is a descent direction with respect to the parameters θ of the MLP, which depends on the structure of $\mathrm{D}_\theta x_b$. In other words, a descent direction for x_b does not guarantee a descent direction for θ.

All experiments were run using PyTorch 1.13.0 (Paszke et al., 2017) on an Intel i7-8565U CPU @ 1.80GHz. Full source code available at `http://deepdeclarativenetworks.com`.