Small Steps and Giant Leaps: Minimal Newton Solvers for Deep Learning (supplementary material)

João F. Henriques Sebastien Ehrhardt Samuel Albanie Andrea Vedaldi
Visual Geometry Group, University of Oxford
{joao,hyenal,albanie,vedaldi}@robots.ox.ac.uk

A. Additional proofs

A.1. Derivation of automatic hyper-parameter tuning in closed-form

We rewrite the problem in eq. 14 as a minimization over ρ and β where $z' = \rho z - \beta \Delta z$:

$$
\begin{aligned}
z &= \arg\min_{\rho,\beta} \hat{f}(z') &\text{(A.1)}\\
&= \arg\min_{\rho,\beta} \begin{bmatrix} \rho \\ -\beta \end{bmatrix}^T \begin{bmatrix} z & \Delta_z \end{bmatrix}^T J + \frac{1}{2} \begin{bmatrix} \rho \\ -\beta \end{bmatrix}^T \begin{bmatrix} z & \Delta_z \end{bmatrix}^T H \begin{bmatrix} z & \Delta_z \end{bmatrix} \begin{bmatrix} \rho \\ -\beta \end{bmatrix} &\text{(A.2)}\\
&= \arg\min_{\rho,\beta} \begin{bmatrix} \rho \\ -\beta \end{bmatrix}^T \begin{bmatrix} z^T J \\ \Delta_z^T J \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \rho \\ -\beta \end{bmatrix}^T \begin{bmatrix} z^T \hat{H} z & z^T \hat{H} \Delta_z \\ z^T \hat{H} \Delta_z & \Delta_z^T \hat{H} \Delta_z \end{bmatrix} \begin{bmatrix} \rho \\ -\beta \end{bmatrix}. &\text{(A.3)}
\end{aligned}
$$

Since $\hat{f}$ is a quadratic function of ρ and β with PSD Hessian it is therefore convex and we can find its extrema by cancelling the gradient:

$$\nabla_{\rho,\beta} \hat{f}(z') = 0. \tag{A.4}$$

Therefore, we have:

$$
\begin{aligned}
\begin{bmatrix} z^T J \\ -\Delta_z^T J \end{bmatrix} + \begin{bmatrix} z^T \hat{H} z & -z^T \hat{H} \Delta_z \\ -z^T \hat{H} \Delta_z & \Delta_z^T \hat{H} \Delta_z \end{bmatrix} \begin{bmatrix} \rho \\ \beta \end{bmatrix} = 0 &\qquad\text{(A.5)}\\
\begin{bmatrix} -\rho \\ \beta \end{bmatrix} = \begin{bmatrix} z^T \hat{H} z & z^T \hat{H} \Delta_z \\ z^T \hat{H} \Delta_z & \Delta_z^T \hat{H} \Delta_z \end{bmatrix}^{-1} \begin{bmatrix} z^T J \\ \Delta_z^T J \end{bmatrix}, &\qquad\text{(A.6)}
\end{aligned}
$$

where the last equality can be computed by inverting the 2×2 matrix explicitly.

A.2. Proof of convergence in the quadratic case

Theorem A.1. *Let f be a convex quadratic function, and its hyper-parameters $\beta > 0$, $\rho > 0$ satisfy*

$$\tfrac{3}{2}\beta h_{\max} - 1 < \rho < 1 + \beta h_{\min}, \tag{A.7}$$

where $h_{\min}$ and $h_{\max}$ are the smallest and largest eigenvalues of the Hessian H, respectively. Then Algorithm 1 converges linearly to the minimum of f.

Corollary A.1.1. *Algorithm 1 converges for any momentum parameter $0 < \rho < 1$ with a sufficiently small learning rate $\beta > 0$, regardless of the (PSD) Hessian spectrum.*

Proof of Theorem A.1. We follow similar derivations on quadratic models by previous work on the heavy-ball method [2, 3, 1], but including our curvature term in the update. We assume the quadratic model:

$$f(w) = \frac{1}{2} w^T H w - b^T w, \tag{A.8}$$

which has Hessian matrix H, and gradient $J(w) = Hw - b$.

Without loss of generality, we will consider the pure Newton method, where H is not regularized ($\lambda = 0$):[1]

$$z_{t+1} = \rho z_t - \beta(H z_t + J(w_t)) \tag{A.9}$$
$$w_{t+1} = w_t + z_{t+1} \tag{A.10}$$

Eq. A.9 can be rearranged to

$$z_{t+1} = (\rho I - \beta H) z_t - \beta J(w_t). \tag{A.11}$$

We now perform a change of variables to diagonalize the Hessian, $H = Q \mathrm{diag}(h) Q^T$, with Q orthogonal and h the vector of eigenvalues. Let $w^* = \arg\min_w f(w) = H^{-1} b$ be the optimal solution of the minimization. Then, replacing $w_t = Q x_t + w^*$ in eq. A.11:

$$Q y_{t+1} = (\rho I - \beta H) Q y_t - \beta H Q x_t \tag{A.12}$$

with $J = H(Q x_t + w^*) - b = H(Q x_t + H^{-1} b) - b = H Q x_t$.

Then, expanding H with its eigendecomposition,

$$Q y_{t+1} = \rho Q y_t - \beta Q \mathrm{diag}(h) Q^T Q y_t - \beta Q \mathrm{diag}(h) Q^T Q x_t. \tag{A.13}$$

Left-multiplying by Q^T,and canceling out Q due to orthogonality,

$$y_{t+1} = \rho y_t - \beta \mathrm{diag}(h) y_t - \beta \mathrm{diag}(h) x_t. \tag{A.14}$$

Similarly for eq. A.10, replacing $z_t = Q y_t$ yields

$$x_{t+1} = x_t + y_{t+1}. \tag{A.15}$$

Note that each pair formed by the corresponding element of y_t and x_t is an independent system with only 2 variables, since the pairs do not interact (eq. A.14 and A.15 only contain element-wise operations). From now on, we will be working on the ith element of each vector.

We can thus write eq. A.14 and A.15 (for a single element i of each) as a vector equation:

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} y_{t+1,i} \\ x_{t+1,i} \end{bmatrix} = \begin{bmatrix} \rho - \beta h_i & -\beta h_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{t,i} \\ x_{t,i} \end{bmatrix}. \tag{A.16}$$

The matrix on the left is necessary to express the fact that the y_{t+1} factor in eq. A.15 must be moved to the left-hand side, which corresponds to iteration $t+1$ ($x_{t+1} - y_{t+1} = x_t$). Left-multiplying eq. A.16 by the inverse,[2]

$$\begin{bmatrix} y_{t+1,i} \\ x_{t+1,i} \end{bmatrix} = \begin{bmatrix} \rho - \beta h_i & -\beta h_i \\ \rho - \beta h_i & 1 - \beta h_i \end{bmatrix} \begin{bmatrix} y_{t,i} \\ x_{t,i} \end{bmatrix}. \tag{A.17}$$

This is the transition matrix R_i that characterizes the iteration, and taking its power models multiple iterations in closed-form:

$$\begin{bmatrix} y_{t,i} \\ x_{t,i} \end{bmatrix} = R_i^t \begin{bmatrix} y_{0,i} \\ x_{0,i} \end{bmatrix}. \tag{A.18}$$

The two eigenvalues of R_i are given in closed-form by:

$$\mathrm{eig}(R_i) = \frac{1}{2} \left(\rho - 2\beta h_i + 1 \pm \sqrt{(\rho - 2\beta h_i)^2 - 2\rho + 1} \right) \tag{A.19}$$

[1] For the general case, the momentum parameter ρ is simply replaced by the slightly perturbed value $\rho - \beta\lambda$ (since $\rho \gg \beta\lambda$), and similar derivations follow.

[2] We have: $\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

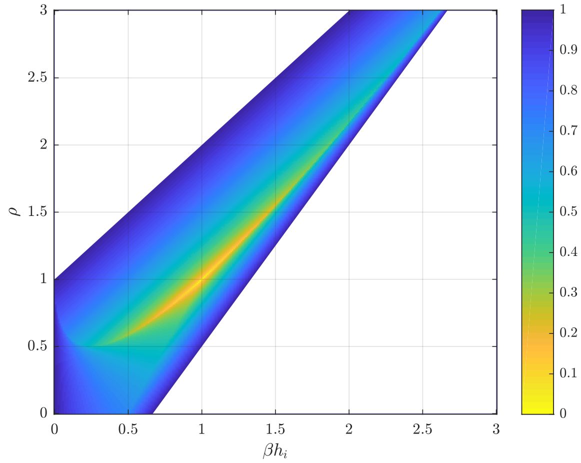

Figure A.1. Convergence rate as a function of hyper-parameters ρ, β, and Hessian eigenvalue h_i. Lower values (brighter) are better. The white areas show regions of non-convergence.

The series in eq. A.18 converges when $|\mathrm{eig}\,(R_i)| < 1$ simultaneously for both eigenvalues, which is equivalent to:

$$\tfrac{3}{2}\beta h_i - 1 < \rho < 1 + \beta h_i, \tag{A.20}$$

with $\rho > 0$ and $\beta h_i > 0$. Note that when using the Gauss-Newton approximation of the Hessian, $h_i > 0$ and thus the last condition simplifies to $\beta > 0$.

Since eq. A.20 has to be satisfied for every eigenvalue, we have

$$\tfrac{3}{2}\beta h_{\max} - 1 < \rho < 1 + \beta h_{\min}, \tag{A.21}$$

with $h_{\min}$ and $h_{\max}$ the smallest and largest eigenvalues of the Hessian H, respectively, proving the result.

The rate of convergence is the largest of the two values $|\mathrm{eig}\,(R_i)|$. When the argument of the square root in eq. A.19 is non-negative, it does not admit an easy interpretation; however, when it is negative, eq. A.19 simplifies to:

$$|\mathrm{eig}\,(R_i)| = \sqrt{\rho - \beta h_i}. \tag{A.22}$$

□

A.2.1 Graphical interpretation

The convergence rate for a single eigenvalue is illustrated in Figure A.1. Graphically, the regions of convergence for different eigenvalues will differ only by a scale factor along the βh_i axis (horizontal stretching of Figure A.1). Moreover, the largest possible range of βh_i values is obtained when $\rho = 1$, and that range is $0 < \beta h_i < \frac{4}{3}$. We can infer that the intersection of the regions of convergence for several eigenvalues will be maximized with $\rho = 1$, for any fixed β.

A.3. Proof of guaranteed descent on general non-convex functions

Theorem A.2. *Let the Hessian $\hat{H}_{t+1}$ be positive definite (which holds when the objective is convex or when Gauss-Newton approximation and trust region are used). Then the update z_{t+1} in Algorithm 1 is a descent direction when β and ρ are chosen according to eq. 18, and $z_{t+1} \neq 0$.*

Proof. To show that the update represents a descent direction, it suffices to show that $J^T z_{t+1} < 0$ (where we have written $J = J(w_t)$ to simplify notation). Since the surrogate Hessian $\hat{H}_{t+1}$ is positive definite (PD) by construction, the update $z_{t+1} = \rho z_t - \beta \Delta_{z_{t+1}}$ satisfies $z_{t+1}^T \hat{H}_{t+1} z_{t+1} > 0$. It is therefore sufficient to prove that $J^T z_{t+1} + z_{t+1}^T \hat{H}_{t+1} z_{t+1} \leq 0$.

It follows from their definition in eq. (18) that ρ and β minimise the RHS of

$$J^T z_{t+1} + \frac{1}{2} z_{t+1}^T \hat{H}_{t+1} z_{t+1} =$$

$$\begin{bmatrix} J^T \Delta_{z_{t+1}} \\ J^T z_t \end{bmatrix}^T \begin{bmatrix} -\beta \\ \rho \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -\beta \\ \rho \end{bmatrix}^T \begin{bmatrix} \Delta_{z_{t+1}}^T \hat{H}_{t+1} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} \Delta_{z_{t+1}} \\ z_t^T \hat{H}_{t+1} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} z_t \end{bmatrix} \begin{bmatrix} -\beta \\ \rho \end{bmatrix} \tag{A.23}$$

In particular, they minimise a quadratic form in $(-\beta, \rho)$ with the following symmetric Hessian

$$K = \begin{bmatrix} \Delta_{z_{t+1}}^T \hat{H}_{t+1} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} \Delta_{z_{t+1}} \\ z_t^T \hat{H} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} z_t \end{bmatrix}. \tag{A.24}$$

Moreover, for any $x = (x_1, x_2) \in \mathbb{R}^2$,

$$\begin{aligned} x^T K x &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} \Delta_{z_{t+1}}^T \hat{H}_{t+1} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} \Delta_{z_{t+1}} \\ z_t^T \hat{H} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} z_t \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= (x_1 \Delta_{z_{t+1}} + x_2 z_t)^T \hat{H} (x_1 \Delta_{z_{t+1}} + x_2 z_t). \end{aligned} \tag{A.25}$$

Consequently, K is guaranteed to be Positive Semidefinite (PSD) and the form is convex with zero gradient at the minimum. Since $z_{t+1} \neq 0$, it follows that at least one of the following holds: (1) K is invertible and hence PD (rather than simply PSD); (2) one of factors $z_t = 0$ or $\Delta_{z_{t+1}} = 0$ is zero; (3) the factors $z_t = 0$ and $\Delta_{z_{t+1}} = 0$ are colinear. In the first case we have,

$$\begin{aligned} &J^T z_{t+1} + \frac{1}{2} z_{t+1}^T \hat{H}_{t+1} z_{t+1} = \\ &\quad -\frac{1}{2} \begin{bmatrix} J^T \Delta_{z_{t+1}} \\ J^T z_t \end{bmatrix}^T \begin{bmatrix} \Delta_{z_{t+1}}^T \hat{H}_{t+1} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} \Delta_{z_{t+1}} \\ z_t^T \hat{H}_{t+1} \Delta_{z_{t+1}} & z_t^T \hat{H}_{t+1} z_t \end{bmatrix}^{-1} \begin{bmatrix} J^T \Delta_{z_{t+1}} \\ J^T z_t \end{bmatrix} \end{aligned} \tag{A.26}$$

Since the inverse of a PD matrix is PD, the RHS of eq. (A.23) is negative. Further, as $\hat{H}_{t+1}$ is PD, it follows that final term in eq. (A.23) is positive, thus K is PD, showing that $J^T z_{t+1} < 0$.

For the second case in which $z_t = 0$ or $\Delta_{z_{t+1}} = 0$, the system reduces to a trivial convex second order equation in ρ or β with a negative solution.

Finally, consider the case when z_t and $\Delta_{z_{t+1}}$ are colinear but both non-negative. Writing $\Delta_{z_{t+1}} = \alpha z_t$ for $\alpha \in \mathbb{R}$, we note that at the minimum we have

$$J^T z_{t+1} + \frac{1}{2} z_{t+1}^T \hat{H}_{t+1} z_{t+1} = -\frac{1}{2} \begin{bmatrix} J^T \Delta_{z_{t+1}} \\ J^T z_t \end{bmatrix}^T \begin{bmatrix} -\beta \\ \rho \end{bmatrix} = -\frac{1}{2} (\rho - \alpha\beta) J^T z_t. \tag{A.27}$$

Thus at the minimum (A.27) is negative, closing the proof. □

Remark. *It follows from the definition of ρ and β that if $J(w_t) = 0$, then $z_{t+1} = 0$.*

Remark. *If $z_{t+1} = 0$, then $z_{t+2} = -\beta J(w_{t+1})$, i.e. we reset the momentum variable z. This guarantees that the algorithm takes a strictly descending direction at least every two steps.*

B. Additional results and implementation details

B.1. Configurations for small-scale dataset experiments

Here we provide additional details of the small-scale datasets described in sec. 4. As noted in the main paper, to give every method the best chance of working effectively we first perform a grid-search over its hyperparameters. This search is performed for each of the small-scale dataset experiments. For each first order solver, we select the configuration which achieves the lowest average error across the final ten iterations of a trajectory. The values included in the search were:

- SGD with momentum: learning rates: Γ, momentum values: $0.9, 0.95, 0.99$
- Adam: learning rates Γ, $\beta_1 : 0.9, 0.99$, $\beta_2 : 0.99, 0.999$

where $\Gamma = 0.1, 0.05, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005$.

B.2. Hyper-parameter and gradient evolution

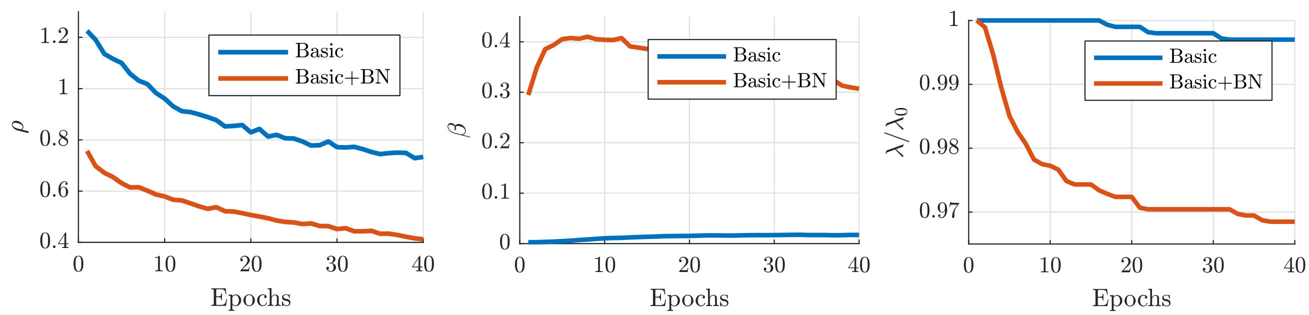

Figure B.2. **Hyper-parameter evolution during training**. Average momentum ρ (left), learning rate β (middle), and trust region λ (right), for each epoch for the basic CNN on CIFAR10, with and without batch normalisation (BN). To make their scales comparable, we plot λ divided by its initial value (which is $\lambda_0 = 1$ with batch normalisation and $\lambda_0 = 10$ without).

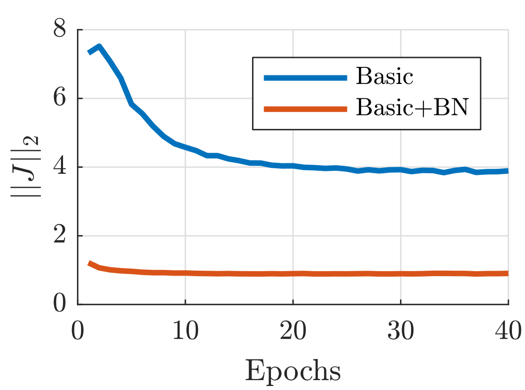

Figure B.3. **Gradient evolution during training**. Average gradient norm during each epoch for the basic CNN on CIFAR-10, with and without batch normalisation (BN).

B.3. Random architecture experiment setup

Each optimiser is tested on 50 random networks, that are held fixed across all methods. The number of convolutional layers is uniformly sampled between 3 and 10, and the number of channels in each layer is drawn uniformly, in powers of two, between 32 and 256. The kernel size is 3×3. Following each convolution (except the last one) there is a ReLU activation and batch-normalisation, and 3×3 max-pooling (stride 2) is placed with 50% chance. Training and evaluation is performed on CIFAR10, with a batch size of 256.

B.4. Wall-Clock time results with Conjugate Gradient

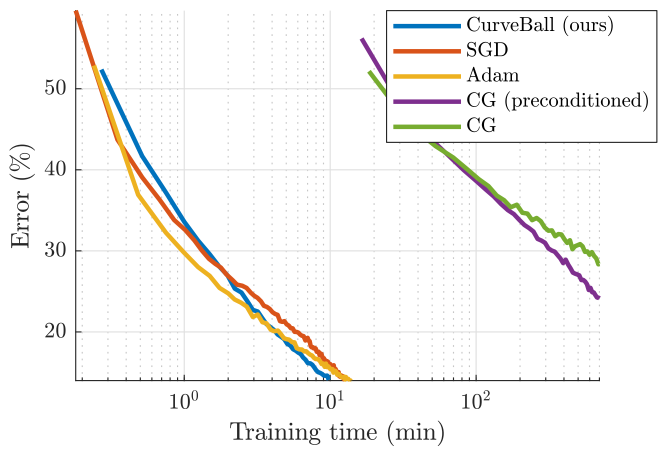

Figure B.4. **Training error vs. wall clock time** (basic CIFAR-10 model). The time axis is logarithmic to show a comparison with conjugate-gradient-based Hessian-free optimisation. Due to the CG iterations, it takes an order of magnitude more time to converge than first-order solvers and our proposed second-order solver, despite the efficient GPU implementation.

B.5. Experiments without a momentum hyper-parameter (fixed $\rho = 1$)

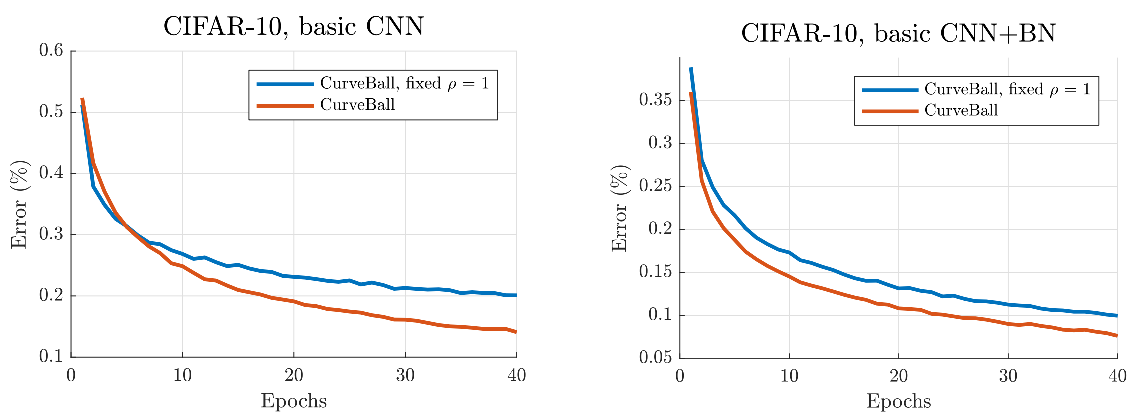

Figure B.5. **Training with fixed** $\rho = 1$. Basic CNN architecture on CIFAR-10 without and with batch normalisation, respectively. Both settings use automatic tuning of the remaining hyper-parameters (by adapting eq. 18).

References

[1] Nicolas Flammarion and Francis Bach. From averaging to acceleration, there is only a step-size. In *Conference on Learning Theory*, pages 658–695, 2015.

[2] Gabriel Goh. Why momentum really works. *Distill*, 2017.

[3] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.