Saddle-free Hessian-free optimization

Martin Arjovsky
marjovsky@dc.uba.ar

June 2, 2015

We want to calculate

$$\Delta\theta = -|\mathbf{H}|^{-1}\nabla f \tag{1}$$

If $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$ is the diagonalization of the Hessian then

$$\begin{aligned}\mathbf{H}^2 &= \mathbf{U}\mathbf{D}\mathbf{U}^{-1}\mathbf{U}\mathbf{D}\mathbf{U}^{-1} \\ &= \mathbf{U}\mathbf{D}^2\mathbf{U}^{-1} = \mathbf{U}|\mathbf{D}|^2\mathbf{U}^{-1} \\ &= \mathbf{U}|\mathbf{D}|\mathbf{U}^{-1}\mathbf{U}|\mathbf{D}|\mathbf{U}^{-1} \\ &= |\mathbf{H}|^2\end{aligned} \tag{2}$$

Therefore, multiplying equation 1 by $\mathbf{H}^2$ we get

$$\mathbf{H}^2\Delta\theta = -|\mathbf{H}|\nabla f \tag{3}$$

We solve for $\Delta\theta$ in two steps:

$$\begin{aligned}y &\leftarrow -|\mathbf{H}|\nabla f \\ \Delta\theta &\leftarrow \left(\mathbf{H}^2\right)^{-1} y\end{aligned}$$

The second step is straightforward using the same idea as in [Martens, 2010]. Using the $\mathcal{R}\{\cdot\}$ technique [Pearlmutter, 1994] we can calculate products $\mathbf{H}v$ for any vector v. Therefore, with the same cost as two gradient computations we can compute $\mathbf{H}^2 v$ for any vector v using the $\mathcal{R}\{\cdot\}$ technique again on $\mathbf{H}v$. With this, the second step is just

$$\Delta\theta \leftarrow \text{CG-Minimize}\left(\mathbf{H}^2, y\right)$$

The first step is trickier, but we can take advantage of the fact that $|\mathbf{H}| = \left(\mathbf{H}^2\right)^{1/2}$ with the last exponent in the sense of the square root of a positive definite matrix. We know how to calculate products of the form $\mathbf{H}^2 v$ for any vector v, we now want to calculate a product by the square root of that matrix. This is an instance of a more general problem: if $\mathbf{A}$ is a positive definite matrix and we know how to take products by $\mathbf{A}$, can we take products by $\mathbf{A}^{1/2}$ without much computation? The answer is yes.

To do this we use the a small variation of the method in [Allen et al., 2000] and think of the initial value problem

$$\begin{cases} dx(t)/dt = -\frac{1}{2}\left(\mathbf{A}t + (1-t)\,\mathbf{I}\right)^{-1}(\mathbf{I}-\mathbf{A})\,x(t) \\ x(0) = v \end{cases} \tag{4}$$

This always has a unique solution, which is

$$x(t) = \left(\mathbf{A}t + (1-t)\,\mathbf{I}\right)^{1/2} v$$

Therefore, if we solve this ODE we have $x(1) = \mathbf{A}^{1/2}v$ as we wanted. The original paper uses the assumption that $\mathbf{A}$ is in tridiagonal form, but only to do fast multiplication by vectors and fast linear system solutions. We already have fast multiplication by $\mathbf{H}^2$ and we can solve the systems easily and with only matrix-vector products by using conjugate gradients again. This usually takes a small constant number of iterations (about 20) to converge and is independent of the amount of parameters. To solve this ODE, I used the RKF45 method, which is very stable. Also, the amount of iterations seems to be constant with the size of the problem.

To summarize, the algorithm is

$$y \leftarrow \text{RKF45}\left(\text{Equation 4}, \mathbf{A} = \mathbf{H}^2, v = -\nabla f\right)$$

$$\Delta\theta \leftarrow \text{CG-Minimize}\left(\mathbf{H}^2, y\right)$$

I'm using the assumption that $|\mathbf{H}|$ is not degenerate, and this will be true in practice because we usually use some small damping to improve stability. Also, the proof of convergence of RKF45 in this case uses the fact that $\|\mathbf{H}^2\|_\infty < 1$. In practice, I did not found this to be necessary, but in the case we want to be sure, we can always scale the matrix-vector products and scale back the result (as in the paper).

The second step of the algorithm takes roughly the same time as HF optimization, since it is the very same idea. The first step does about 40 iterations of RKF, and a CG in each of these. In the worst case, we end up with an algorithm that is about one order of magnitude slower than HF, but a saddle free one without the need of using a low rank approximation of the Hessian. We also don't have to worry about some of the things HF does, like using the Gauss-Newton approximation of the Hessian.

References

[Allen et al., 2000] Allen, E., Baglama, J., and Boyd, S. (2000). Numerical approximation of the product of the square root of a matrix with a vector. *Linear Algebra and its Applications*, 310(1–3):167 – 181.

[Martens, 2010] Martens, J. (2010). Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 735–742.

[Pearlmutter, 1994] Pearlmutter, B. A. (1994). Fast exact multiplication by the hessian. *Neural Computation*, 6:147–160.