Natural Gradients Made Quick and Dirty: Companion Cookbook

Jascha Sohl-Dickstein

January 23, 2010

1 Recipes and tricks

1.1 Natural gradient

The natural gradient is

$$\tilde{\nabla}_\theta J(\theta) = G^{-1}(\theta)\,\nabla_\theta J(\theta) \tag{1}$$

where $J(\theta)$ is an objective function to be minimized with parameters θ, and $G(\theta)$ is a metric on the parameter space. Learning should be performed with an update rule

$$\theta_{t+1} = \theta_t + \tilde{\Delta}\theta_t \tag{2}$$

$$\tilde{\Delta}\theta \propto -\tilde{\nabla}_\theta J(\theta) \tag{3}$$

with steps taken in the direction given by the natural gradient.

1.2 Metric $G(\theta)$

If the objective function $J(\theta)$ is the negative log likelihood of a probabilistic model $q(x;\theta)$ under an observed data distribution $p(x)$

$$J(\theta) = -\left\langle \log q(x;\theta) \right\rangle_{p(x)} \tag{4}$$

then the Fisher information matrix

$$G_{ij}(\theta) = \left\langle \frac{\partial \log q(x;\theta)}{\partial \theta_i} \frac{\partial \log q(x;\theta)}{\partial \theta_j} \right\rangle_{q(x;\theta)} \tag{5}$$

is a good metric to use.

If the objective function is *not* of of the form given in Equation 4, and cannot be transformed into that form, then greater creativity is required. See Section 1.8 for some basic hints.

Remember, as discussed in Section 1.10, even if the metric you choose is approximate, it is still likely to speed learning!

1.3 Fisher information over data distribution

The Fisher information matrix (Equation 5) requires averaging over the model distribution $q(x;\theta)$. For some models this is very difficult to do. If that is the case, instead taking the average over the empirical data distribution $p(x)$

$$G_{ij}(\theta) = \left\langle \frac{\partial \log q(x;\theta)}{\partial \theta_i} \frac{\partial \log q(x;\theta)}{\partial \theta_j} \right\rangle_{p(x)} \tag{6}$$

provides an effective alternative natural gradient.

1.4 Energy approximation

Learning in a probabilistic model of the form

$$q(\mathbf{x}) = \frac{e^{-E(\mathbf{x};\theta)}}{Z(\theta)} \tag{7}$$

is in general very difficult, since it requires working with the frequently intractable partition function integral $Z(\theta) = \int e^{-E(\mathbf{x};\theta)} d\mathbf{x}$. There are a number of techniques which can provide approximate learning gradients (eg contrastive divergence, score matching, mean field theory, variational bayes, minimum probability flow). Turning those gradients into natural gradients is difficult though, as the Fisher information depends on the gradient of $\log Z(\theta)$. Practically, simply ignoring the $\log Z(\theta)$ terms entirely and using a metric

$$G_{ij}(\theta) = \left\langle \frac{\partial E(x;\theta)}{\partial \theta_i} \frac{\partial E(x;\theta)}{\partial \theta_j} \right\rangle_{p(x)} \tag{8}$$

averaged over the data distribution works surprisingly well, and frequently greatly accelerates learning.

1.5 Diagonal approximation

$G(\theta)$ is a square matrix of size NxN, where N is the number of parameters in the vector θ. For problems with large N, $G^{-1}(\theta)$ can be impractically expensive to compute, or even apply. For almost all problems however, the natural gradient still improves convergence even when off diagonal elements of $G(\theta)$ are neglected

$$G_{ij}(\theta) = \delta_{ij} \left\langle \left(\frac{\partial \log q(x;\theta)}{\partial \theta_i} \right)^2 \right\rangle_{q(x;\theta)} \tag{9}$$

making inversion and application cost $O(N)$ to perform.

If the parameters can be divided up into several distinct classes (for instance the covariance matrix and means of a gaussian distribution), various block diagonal forms may also be worth considering.

1.6 Regularization

Even if evaluating the full G is easy for your problem, you may still find that G^{-1} explodes[1]. Dealing with this - solving a set of linear equations subject to some regularization, rather than using the exact matrix inverse - is an entire field of study in computer science. Here we give one simple plug and play technique, called stochastic robust approximation [section 6.4.1 in http://www.stanford.edu/ boyd/cvxbook/], for regularizing the matrix inverse. If G^{-1} is replaced with

$$G_{reg}^{-1} = \left(G^T G + \epsilon \mathbf{I}\right)^{-1} G^T \tag{10}$$

where ϵ is some small constant (say 0.01), the matrix inverse will be much better behaved.

Alternatively, techniques such as ridge regression can be used to solve the linear equation

$$G\left(\theta\right) \tilde{\nabla}_\theta J\left(\theta\right) = \nabla_\theta J\left(\theta\right) \tag{11}$$

for $\tilde{\nabla}_\theta J\left(\theta\right)$.

1.7 Combining the natural gradient with other techniques using the natural parameter space ϕ

It can useful to combine the natural gradient with other gradient descent techniques. Blindly replacing all gradients with natural gradients frequently causes problems (line search implementations, for instance, depend on the gradients they are passed being the true gradients of the function they are descending). For a fixed value of G though there is a natural parameter space.

$$\phi = G^{\frac{1}{2}}\left(\theta_{fixed}\right)\theta \tag{12}$$

in which the steepest gradient is the same as the natural gradient.

In order to easily combine the natural gradient with other gradient descent techniques, fix θ_{fixed} to the initial value of θ and perform gradient descent over ϕ using any preferred algorithm. After a significant number of update steps convert back to θ, update θ_{fixed} to the new value of θ, and continue gradient descent in the new ϕ space.

1.8 Natural gradient of non-probabilistic models

The techniques presented here are not unique to probabilistic models. The natural gradient can be used in any context where a suitable metric can be written

[1] This is a general problem when taking matrix inverses. A matrix A with random elements - or with noisy elements - will tend to have a few very very small eigenvalues. The eigenvalues of A^{-1} are the inverses of the eigenvalues of A. A^{-1} will thus tend to have a few very very large eigenvalues, which will tend to make the elements of A^{-1} very very large. Even worse, the eigenvalues and eigenvectors which most dominate A^{-1} are those which were smallest, noisiest and least trustworthy in A.

for the parameters. There are several approaches to writing an appropriate metric.

1. If the objective function is of a form
$$J\left(\theta\right) = \left\langle l\left(x;\theta\right)\right\rangle_{p(x)} \tag{13}$$
where $\left\langle\cdot\right\rangle_{p(x)}$ indicates averaging over some data distribution $p(x)$, then it is sensible to choose a metric based on
$$G_{ij}\left(\theta\right) = \left\langle \frac{\partial l\left(x;\theta\right)}{\partial\theta_i}\frac{\partial l\left(x;\theta\right)}{\partial\theta_j}\right\rangle_{p(x)} \tag{14}$$

2. Similarly, imagine that the given penalty function is the log likelihood of a probabilistic model, and rewrite the problem as if it were probabilistic. Then use the Fisher information metric on its probabilistic interpretation.

   For example, the task of trying to minimize an L2 penalty function $\left|\left|y - f\left(x;\theta\right)\right|\right|^2$ over observed pairs of data $p\left(x,y\right)$ can be made probabilistic. Imagine that the L2 penalty instead represents a conditional gaussian $q\left(y|x;\theta\right) \propto \exp\left(-\left|\left|y - f\left(x;\theta\right)\right|\right|^2\right)$ over y, and use the observed marginal $p\left(x\right)$ over x to build a joint distribution $q\left(x,y;\theta\right) = q\left(y|x;\theta\right)p\left(x\right)$.[2] This generates the metric:
$$\begin{aligned} G_{ij}\left(\theta\right) &= \left\langle \frac{\partial \log\left[q\left(y|x;\theta\right)p\left(x\right)\right]}{\partial\theta_i}\frac{\partial \log\left[q\left(y|x;\theta\right)p\left(x\right)\right]}{\partial\theta_j}\right\rangle_{q(y|x;\theta)p(x)} && (15)\\ &= \left\langle \frac{\partial \log q\left(y|x;\theta\right)}{\partial\theta_i}\frac{\partial \log q\left(y|x;\theta\right)}{\partial\theta_j}\right\rangle_{q(y|x;\theta)p(x)} && (16)\end{aligned}$$

3. Find a set of transformations $T\left(\theta\right)$ to apply to the parameters which you believe the distance measure $\left|d\theta\right|$ should be invariant to, and then find a metric $G\left(\theta\right)$ such that it is. That is find $G\left(\theta\right)$ such that the following relationship holds for any invariant transformation $T\left(\theta\right)$.
$$d\theta^T G\left(\theta\right)d\theta = T\left(d\theta\right)^T G\left(T\left(\theta\right)\right)T\left(d\theta\right) \tag{17}$$
where $T\left(d\theta\right) \equiv T\left(\theta + d\theta\right) - T\left(\theta\right)$.

   A special case of this approach involves functions parametrized by a matrix, as illustrated in the next section.

[2] Amari suggests using some uninformative model distribution $q\left(x\right)$ over the inputs, such as a gaussian distribution, rather than taking $p\left(x\right)$ from the data []. Either works fine. Using the data gets you closer to the desired distribution, but at the expense of extra computation if the uninformative marginal allows a closed form solution for $G\left(\theta\right)$.

1.9 $W^T W$

If a function depends on a (square, non-singular) matrix W, it frequently aids learning a great deal to take

$$\Delta W_{nat} \propto \frac{\partial J\left(W\right)}{\partial W} W^T W \tag{18}$$

The algebra leading to this rule is complex, but as discussed in the previous section it falls out of a demand that the distance measure $|dW|$ be invariant to a set of transformations. In this case, those transformations are right multiplication by any (non singular) matrix Y.

$$d\theta^T G\left(\theta\right) d\theta = \left(d\theta Y\right)^T G\left(\theta Y\right)\left(d\theta Y\right) \tag{19}$$

1.10 What if my approximation of $\Delta\theta_{nat}$ is wrong?

For any positive definite H, movement in a direction

$$\tilde{\Delta}\theta = H\Delta\theta \tag{20}$$

will descend the objective function. If the wrong H is used, gradient descent is performed in a suboptimal way . . . which is the problem when steepest gradient descent is used as well. Making an educated guess as to H rarely makes things worse, and frequently helps a great deal. Don't be scared to experiment!

2 References

Natural Gradient Works Efficiently in Learning, Neural computation [0899-7667] Amari (1998) volume: 10 issue: 2 page: 251

Differential geometry in statistical inference / S.-I. Amari ... [et al.] Publisher Hayward, Calif. : Institute of Mathmatical Statistics, 1987.

http://www.stanford.edu/ boyd/cvxbook/

Shun'ichi Amari, Hiroshi Nagaoka - Methods of information geometry, Transactions of mathematical monographs; v. 191, American Mathematical Society, 2000