Natural Gradients Made Quick and Dirty

Jascha Sohl-Dickstein

January 23, 2010

1 Abstract

The natural gradient, as introduced by [1], allows for more efficient gradient descent by removing dependencies and biases inherent in a function’s parameterization. Several papers present the topic thoroughly and precisely [2]. It remains a very difficult idea to get your head around however. The purpose of this note is to provide simple intuition for what the natural gradient is and how it can be used. We will relate it to the more familiar ideas of covariance and signal whitening, use informal language where possible and avoid discussion of differential geometry. A companion document presents a series of cookbook techniques for practical application of the natural gradient. Readers interested in a more formal and in depth discussion should see [2]. Those interested in a significantly mathier stab at clarity should see [?].

2 Natural gradient

2.1 A simple example

We begin with a simple model which has clearly been very poorly parametrized. For this we use a two dimensional gaussian distribution, with means written in terms of the parameters \( \theta \equiv (\theta_a, \theta_b) \).

\[
q(x; \theta) = \frac{1}{2\pi} \exp \left[ -\frac{1}{2} \left( x_1 - \left[ 10\theta_a + \frac{1}{10} \theta_b \right] \right)^2 - \frac{1}{2} \left( x_2 - \left[ \frac{1}{10} \theta_a \right] \right)^2 \right] \tag{1}
\]

As an objective function \( J(\theta) \) we use the log likelihood of \( q(x; \theta) \) under an observed data distribution \( p(x) \)

\[
J(\theta) = \langle \log q(x; \theta) \rangle_{p(x)} \tag{2}
\]

Using steepest gradient descent to maximize the log likelihood involves taking steps like

\[
\Delta \theta \propto \nabla_{\theta} J(\theta) \tag{3}
\]

\[
\begin{bmatrix} \Delta \theta_a \\ \Delta \theta_b \end{bmatrix} \propto \begin{bmatrix} \langle 10 \left( x_1 - \left[ 10\theta_a + \frac{1}{10} \theta_b \right] \right) + \frac{1}{10} \left( x_2 - \left[ \frac{1}{10} \theta_a \right] \right) \rangle_{p(x)} \\ \langle \frac{1}{10} \left( x_1 - \left[ 10\theta_a + \frac{1}{10} \theta_b \right] \right) \rangle_{p(x)} \end{bmatrix} \tag{4}
\]
As can be seen in Figure 1 the steepest gradient update steps move the distribution in a less than sensible direction. The distribution is much more sensitive to changes in $\theta_a$ than $\theta_b$, so the step size in $\theta_a$ should be much smaller, but is instead much larger. In addition, $\theta_a$ and $\theta_b$ aren’t independent of each other. They move the distribution in nearly the same direction, making movement in the perpendicular direction particularly difficult. Getting the parameters here to converge via steepest descent would be a dicey proposition.

The pathological learning gradient above is indicative of a more general problem. A model’s learning gradient is effected by the parameterization of the model as well as the objective function being minimized. The effects of the parameterization can dominate learning. The natural gradient is a technique to remove the effects of model parameterization from the gradient.

2.2 A metric on the parameter space

In order to compensate for the relative scaling and dependencies in the parameterization, they are first described by putting a metric on the parameter space $\theta$ (for a simple example of a metric, see Figure 2). This metric is expressed via a symmetric matrix $G (\theta)$, which defines the length $|d\theta|$ of a small step $d\theta$ in parameter space.

$$|d\theta|^2 = \sum_i \sum_j G_{ij} (\theta) d\theta_i d\theta_j = d\theta^T G (\theta) d\theta$$  \hspace{1cm} (5)

$G (\theta)$ is chosen so that the length $|d\theta|$ provides a reasonable measure for the expected magnitude of the difference of $J (\theta + d\theta)$ from $J (\theta)$. That is, $G (\theta)$ is chosen such that $|d\theta|$ is representative of the expected magnitude of the change in the objective function resulting from a step $d\theta$. There is no uniquely correct choice for $G (\theta)$.

If the objective function $J (\theta)$ is the log likelihood of a probability distribution $q (x; \theta)$, then a measure of the information distance between $q (x; \theta + d\theta)$ and $q (x; \theta)$ is appropriate, and the Fisher information matrix is frequently used as a metric (see cookbook for equation - but don’t feel too constrained by it as a choice). For the example presented in Section 2.1 the Fisher information metric is

$$G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$  \hspace{1cm} (6)
Figure 2: A simple example of a metric. A function $f(\theta_\alpha, \theta_\beta)$ is defined on a sphere, where $\theta_\alpha \in [-\pi, \pi]$ is an azimuthal angle and $\theta_\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is an elevation angle. Note that at the poles, changes in $\theta_\beta$ correspond to much smaller movements on the surface of the sphere than changes in $\theta_\alpha$, while at the equator small steps in either $\theta_\alpha$ or $\theta_\beta$ correspond to movements on the sphere of the same length. To capture this uneven scaling, a metric is defined which provides an accurate measure of the distance traveled on the surface of a sphere for small changes $d\theta = [d\theta_\alpha, d\theta_\beta]^T$ at any $\theta = [\theta_\alpha, \theta_\beta]^T$.

Figure 3: Example of signal whitening. (a) Samples from an unwhitened distribution $x$ in 2 variables. (b) The same samples after whitening, in new variables $y = Wx = \Sigma^{-\frac{1}{2}}x$

2.3 Connection to covariance

$G(\theta)$ is an analogue of the inverse covariance matrix $\Sigma^{-1}$. Just as a signal can be whitened given $\Sigma^{-1}$ - removing all first order dependencies and scaling each dimension to unit length - the parameterization of $J(\theta)$ can also be “whitened,” removing the dependencies and differences in scaling between dimensions captured by $G(\theta)$. See Figure 3 for an example of signal whitening.

As a quick review, the covariance matrix $\Sigma$ of a signal $x$ is defined as

$$\Sigma = \langle xx^T \rangle$$

(7)

The inverse covariance matrix is frequently used as a metric on the signal $x$. This is how Mahalanobis distance is defined, in an analogue to the definition of $|d\theta|^2$ in Section 2.2.

$$|dx|^2_{\text{Mahalanobis}} = dx^T \Sigma^{-1} dx$$

(8)

In order to whiten a signal $x$, a matrix $W$ is found such that the covariance matrix for a new signal $y = Wx$ is the identity matrix. The signal $y$ is then a whitened version of $x$.

$$I = \langle yy^T \rangle = W \langle xx^T \rangle W^T = W\Sigma W^T$$

(9)

Remembering that $\Sigma^{-1}$ is symmetric, one solution\(^1\) to this system of linear equations is

$$W = \Sigma^{-\frac{1}{2}}$$

(10)

$$y = \Sigma^{-\frac{1}{2}}x$$

(11)

\(^1\)Choosing $W = \Sigma^{-\frac{1}{2}}$ leads to symmetric, or zero-phase, whitening. In some fields it is referred to as a decorrelation stretch. It is equivalent to rotating a signal to the PCA basis, rescaling each axis to have unit norm, and then performing the inverse rotation, returning the signal to its original orientation. All unitary transformations of $\Sigma^{-\frac{1}{2}}$ also whiten the signal.
Whitening is a common preprocessing step in signal processing. It prevents incidental differences in scaling between dimensions from effecting later processing stages.

2.4 “Whitening” the parameter space

If \( G \) is constant over \( \theta \), then a similar procedure can be followed to produce a “whitened” parameterization \( \phi \). We wish to find new parameters \( \phi = W\theta \) such that the metric \( G \) on \( \phi \) is the identity, as \( \Sigma^{-1} \) is the identity for a whitened signal. This will mean that a small step \( d\phi \) in any direction will tend to have the same magnitude effect - gradients in \( \phi \) will be determined by the objective function \( J(\phi) \), but will not be biased by its parameterization.

\[
|d\phi|^2 = |d\theta|^2 \tag{12}
\]

\[
d\phi^T I d\phi = d\theta^T G d\theta \tag{13}
\]

\[
d\phi^T d\phi = d\theta^T G d\theta \tag{14}
\]

\[
d\theta^T W^T W d\theta = d\theta^T G d\theta \tag{15}
\]

Noting that \( G \) is symmetric, we find that one solution to this system of linear equations is

\[
W = G^{\frac{1}{2}} \tag{16}
\]

\[
\phi = G^{\frac{1}{2}} \theta \tag{17}
\]

As can be seen in Figure 1, gradient descent steps in \( \phi \) space descend the objective function in a more direct fashion. In the \( \phi \) space, the steepest gradient is the natural gradient.

There is a catch. \( G \) is almost always a function of \( \theta \), and for most problems there is no natural parameterization \( \phi \) which will be “white” everywhere. So long as \( G(\theta) \) changes slowly though, it can be treated as constant for a single learning step. This suggests the following as an algorithm for learning in a natural parameter space.

1. Express \( J(\cdot) \) in terms of natural parameters \( \phi = G^{\frac{1}{2}} (\theta_t) \theta \)

2. Calculate an update step \( \Delta \phi \propto \nabla_{\phi} J(\phi) \)

3. Calculate the \( \theta_{t+1} = G^{-\frac{1}{2}} (\theta_t) (\phi + \Delta \phi) \) associated with the updated \( \phi \).

4. Repeat

Practically, \( G(\theta) \) can usually be treated as constant for many learning steps. As discussed in the companion cookbook, this allows the natural gradient to be combined in a plug and play fashion with other gradient descent algorithms simply by performing gradient descent over \( \phi \) rather than \( \theta \).
2.5 The natural gradient in $\theta$

When not combining the natural gradient with other learning techniques the change of variables can be dispensed with entirely, and the natural gradient $\nabla_\theta J (\theta)$ can be written directly in terms of $G (\theta)$ and the true gradient $\nabla_\theta J (\theta)$. First we write $\Delta \phi$ in terms of $\theta$, then we write $\Delta \theta$ in terms of $\Delta \phi$

$$
\Delta \phi \propto \nabla_\phi J (\phi) \quad (18)
$$

$$
= \frac{\partial \theta}{\partial \phi}^T \nabla_\theta J (\theta) \quad (19)
$$

$$
= G^{-\frac{1}{2}} \nabla_\theta J (\theta) \quad (20)
$$

(where $\frac{\partial \theta}{\partial \phi}$ is the Jacobian matrix, with entries $\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \theta_i}{\partial \phi_j}$)

$$
\Delta \theta \propto \frac{\partial \theta}{\partial \phi} \Delta \phi \quad (21)
$$

$$
= G^{-\frac{1}{2}} \Delta \phi \quad (22)
$$

$$
\propto G^{-1} \nabla_\theta J (\theta) \quad (23)
$$

Taking the natural learning step to be proportional to the natural gradient, $\Delta \theta \propto \nabla_\theta J (\theta)$, the natural gradient can be written as

$$
\nabla_\theta J (\theta) = G^{-1} (\theta) \nabla_\theta J (\theta) \quad (24)
$$

And that’s it! See Figure 1 to see this gradient applied to the example objective function from Section 2.1. If gradient descent is performed by infinitesimal steps in the direction indicated by $\nabla_\theta J (\theta)$, then the parameterization of the problem will have no effect on the path taken during learning (though choice of $G (\theta)$ will have an effect).

3 References


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