ARTICLE IN PRESS

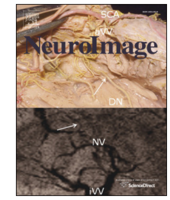
YNIMG-11296; No. of pages: 7; 4C: 5, 6

NeuroImage xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg



Technical Note

Efficient gradient computation for dynamical models

B. Sengupta, K.J. Friston, W.D. Penny

Wellcome Trust Centre for Neuroimaging, Institute of Neurology, University College London, 12 Queen Square, London WC1N 3BG, UK

ARTICLE INFO

Article history:
Accepted 12 April 2014
Available online xxxx

ABSTRACT

Data assimilation is a fundamental issue that arises across many scales in neuroscience — ranging from the study of single neurons using single electrode recordings to the interaction of thousands of neurons using fMRI. Data assimilation involves inverting a generative model that can not only explain observed data but also generate predictions. Typically, the model is inverted or fitted using conventional tools of (convex) optimization that invariably extremise some functional — norms, minimum descriptive length, variational free energy, etc. Generally, optimisation rests on evaluating the local gradients of the functional to be optimized. In this paper, we compare three different gradient estimation techniques that could be used for extremising any functional in time — (i) finite differences, (ii) forward sensitivities and a method based on (iii) the adjoint of the dynamical system. We demonstrate that the first-order gradients of a dynamical system, linear or non-linear, can be computed most efficiently using the adjoint method. This is particularly true for systems where the number of parameters is greater than the number of states. For such systems, integrating several sensitivity equations – as required with forward sensitivities – proves to be most expensive, while finite-difference approximations have an intermediate efficiency. In the context of neuroimaging, adjoint based inversion of dynamical causal models (DCMs) can, in principle, enable the study of models with large numbers of nodes and parameters.

Introduction

An important goal of systems neuroscience is to integrate empirical data from various neuroimaging modalities with biologically informed models that describe the underlying generative processes. Here, the data to be explained are for example M/EEG and fMRI recordings made while subjects perform various experimental tasks, and the underlying neurodynamic processes are framed in terms of differential equations describing activity in neural masses, mean fields, or neural fields (David et al., 2006; Deco et al., 2008; Friston et al., 2003).

Considerable insight can be gained from studying the emergent properties of such neurodynamic processes. These can then be qualitatively compared with empirical data, allowing consilience among multiple levels of description (Gazzaniga, 2010; Hopfield and Brody, 2001; Wilson, 1999). An alternative approach is to directly fit neurodynamical models to neuroimaging data using standard model fitting procedures from statistics and machine learning (Bishop, 2006; Press et al., 1992). Differences in the generative processes induced by experimental manipulations can then be associated with changes in underlying brain connectivity. One example of such an approach is Dynamic Causal Modelling (DCM) (Friston et al., 2003) which fits differential equation models to neuroimaging data using a variational Bayesian scheme (Friston et al., 2007).

More generally, in the statistics and machine learning literature various methods have been employed to fit differential equations to data, from maximum likelihood approaches (Ramsay et al., 2007) to Bayesian sampling algorithms (Calderhead and Girolami, 2009; Vyshemirsky and Girolami, 2008). The majority of these convex optimisation approaches involve computing the gradient; the change in the cost function produced by a change in model parameters. This gradient is then combined with information from line searches (e.g., Wolfe's conditions) or methods involving a Newton, quasi-Newton (low-rank) or Fisher information based curvature estimators to update model parameters (Bishop, 1995; Nocedal and Wright, 2006; Press et al., 1992). The main computational bottleneck in these algorithms is the computation of the gradient (or the curvature) of the parametric cost function. This motivates the search for efficient methods to evaluate gradients.

This paper compares three different methods for computing gradients, and studies the conditions under which each is preferred. The first is the Finite Difference (FD) method, which is the simplest and most general method — and is currently used in DCM. The second is the Forward Sensitivity (FS; also known as tangent linear) method, which has previously been proposed in the context of modeling fMRI time series (Deneux and Faugeras, 2006). The third is the Adjoint Method

E-mail addresses: b.sengupta@ucl.ac.uk (B. Sengupta), k.friston@ucl.ac.uk (K.J. Friston), w.penny@ucl.ac.uk (W.D. Penny).

http://dx.doi.org/10.1016/j.neuroimage.2014.04.040

Please cite this article as: Sengupta, B., et al., Efficient gradient computation for dynamical models, NeuroImage (2014), http://dx.doi.org/10.1016/j.neuroimage.2014.04.040

(AM) which has previously been used in the context of dynamical systems theory (Wang, 2013), weather forecasting (Errico, 1997), image registration (Clark, 2011) and single-neuron biophysics (Stemmler et al., 2012).

The paper is structured as follows — the methods section describes each approach including a mathematical derivation of the adjoint method. Examples of the FS and AM updates are then provided for the case of simple Euler integration. The results section reports numerical simulations that disclose the scaling characteristics of each method. Simulations are provided for linear dynamical and weakly-coupled oscillator systems. We conclude with a discussion of the relative merits of each method.

Methods

We consider dynamical systems of the form

$$\begin{aligned}\dot{x} &= f(x,p)\\ j(x,p) &= -\frac{1}{2}\|y-g(x,p)\|^2\end{aligned} \tag{1}$$

where x is a state variable, the dot notation denotes a time derivative $\frac{dx}{dt}$, t is time, $f(\cdot)$ is the flow equation (dynamics), and p are model parameters. The model produces a prediction via an observation function $g(x,p)$ and an instantaneous cost function $j(x,p)$ measures the squared difference from data points y. The total cost is then given by the integral up to time point T

$$J(p) = \int_0^T j(x,p)dt. \tag{2}$$

We consider three methods for computing the gradient $\frac{dJ}{dp}$.

Finite difference method

The (one-sided) finite difference approximation to the gradient is then

$$\frac{dJ}{dp_i} = \frac{J(p+\delta_i)-J(p)}{\delta_i} \tag{3}$$

where δ_i denotes a small change (generally, $\sqrt{\epsilon}$ where ϵ is the machine epsilon) to the ith parameter. The error in the computation of this gradient is of order δ_i. The computation of $\frac{dJ}{dp}$ requires $P+1$ runs of the integration process, one for each model parameter. It is also possible to use central differences

$$\frac{dJ}{dp_i} = \frac{J(p+\delta_i)-J(p-\delta_i)}{2\delta_i} \tag{4}$$

which has an error of order δ_i^2 but requires $2P+1$ runs of the integration process. Variations on the vanilla FD approach are discussed in (Press et al., 1992; Richtmeyer and Morton, 1967).

Forward Sensitivity method

The original dynamical model (Eq. (1)) can be implicitly differentiated w.r.t parameters to give

$$\frac{d\dot{x}}{dp} = \frac{\partial f}{\partial x}\frac{dx}{dp} + \frac{\partial f}{\partial p}. \tag{5}$$

If the state variables are of dimension D and the parameters of dimension P then the quantity $\frac{dx}{dp}$ is a $D+P$ matrix, which can be vectorized to form a new flow function. This forms a new dynamical system of dimension $D \times P$ that can then be integrated using any numerical method to produce $\frac{dx}{dp}$ as a function of time. The Forward Sensitivity approach has been known since the publication of Gronwall's theorem (Gronwall, 1919). The cost gradient is then given by accumulating the sensitivity derivative $\frac{dx}{dp}$ over time according to:

$$\begin{aligned}\frac{dJ}{dp} &= \int_0^T \frac{dj}{dp}dt\\ \frac{dj}{dp} &= \frac{\partial j}{\partial x}\frac{dx}{dp} + \frac{\partial j}{\partial p}\\ &= \frac{\partial j}{\partial g}\frac{\partial g}{\partial x}\frac{dx}{dp} + \frac{\partial j}{\partial g}\frac{\partial g}{\partial p}.\end{aligned} \tag{6}$$

Euler example

This section illustrates the FS approach first-order Euler integration of the dynamics

$$x_n = x_{n-1} + \tau f(x_{n-1}, p) \tag{7}$$

at discrete times $t(n)$. The FS method is based on differentiating this equation to give

$$\frac{dx_n}{dp} = \frac{dx_{n-1}}{dp} + \tau\left[\frac{\partial f}{\partial x_{n-1}}\frac{dx_{n-1}}{dp} + \frac{\partial f}{\partial p}\right]. \tag{8}$$

This method is illustrated in Fig. 1 where the solid path indicates a trajectory of points x_n for a dynamical system with parameters p and the dotted path indicates the trajectory $\underline{x}_n$ for the same dynamical system but with parameters $\underline{p} = p + \delta_i$. The dotted path can be obtained from the solid path via the total derivative $\frac{dx_n}{dp_i}$ in the direction of the perturbation, δ_i. The FS method provides a method for computing this derivative. Under a first order Euler approach for integrating the dynamics, this is implemented using the above recursion.

Because the perturbed path (dotted in Fig. 1) can be reached from the original trajectory via the total derivative $\frac{dx_n}{dp}$, there is no need to separately integrate the system with parameters $\underline{p}$. Geometrically, the points $\underline{x}_n$ in Fig. 1 can be reached via the solid and dashed lines (rather than the dotted lines).

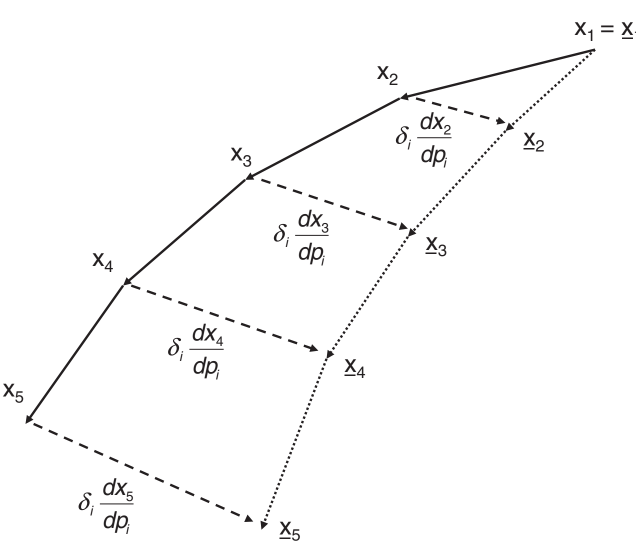


Fig. 1. Forward Sensitivity The solid path indicates a trajectory of points x_n, with $n = 1\ldots5$, for a dynamical system with parameters p. The dotted path indicates the trajectory $\underline{x}_n$ for the same dynamical system but with parameters $\underline{p} = p + \delta_i$. The dotted path can be reached from the solid path via the total derivative $\frac{dx_n}{dp}$. The Forward Sensitivity approach provides a method for computing this derivative.

Please cite this article as: Sengupta, B., et al., Efficient gradient computation for dynamical models, NeuroImage (2014), http://dx.doi.org/10.1016/j.neuroimage.2014.04.040