Adjoint mode and rollout response

Darian Nwankwo

May 24, 2024

1 Thoughts

Of the computations required to solve for $\bar{v}$, we have a handle on everything except (I believe) for differentiating with respect to function values and locations independently. As it currently stands, when we compute perturbations of the policy solve, it is doing so with respect to (x_k, y_k, f_k^+) simultaneously and returning the sum, i.e.

$$H_\alpha \delta x^{k+1} + \sum_{j=1}^{k} \left(\frac{\partial \nabla \alpha}{\partial x^j} \delta x^j + \frac{\partial \nabla \alpha}{\partial y_j} \delta y_j \right) + \frac{\partial \nabla \alpha}{\partial f^{+k}} \delta f^{+k} = 0.$$

Therefore, the problem becomes disambiguating our existing computations such that we have a handle on each term in this sum instead of returning the resulting sum. I'm convinced of the logic thereafter, now it is a matter of ensuring the code computes each respective component as we see fit.

1.1 Request

Let's pair program to disambiguate our sum such that we have a handle on each result operand.

2 Original Discourse

Consider a rollout trajectory to time step t. We are interested in computing the derivative of the function value y_t at time t respect to the initial condition $\mathbf{x}_0$.

The basic setup for the forward computation is

$$\begin{aligned} \mathbf{r}_j(\mathbf{x}_j; \mathbf{x}_0, y_0, \ldots, \mathbf{x}_{j-1}, y_{j-1}) &= 0 \\ f(\mathbf{x}_j) - y_j &= 0. \end{aligned}$$

Here $\mathbf{r}_j$ should be thought of as the gradient of the acquisition function at step j (i.e. we seek to maximize the acquisition function at each step). I will not write out all the details of this calculation here, as we have it elsewhere.

We can write the usual “forward mode” computation of the derivatives by differentiating each equation in term and then doing forward substitution.

$$\begin{bmatrix}
f'(\mathbf{x}_0) & -1 & & & & & & & \\
\frac{\partial \mathbf{r}_1}{\partial \mathbf{x}_0} & \frac{\partial \mathbf{r}_1}{\partial y_0} & \frac{\partial \mathbf{r}_1}{\partial \mathbf{x}_1} & & & & & & \\
 & & f'(\mathbf{x}_1) & -1 & & & & & \\
\frac{\partial \mathbf{r}_2}{\partial \mathbf{x}_0} & \frac{\partial \mathbf{r}_2}{\partial y_0} & \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{r}_2}{\partial y_1} & \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}_2} & & & & \\
 & & & & f'(\mathbf{x}_2) & -1 & & & \\
\vdots & & & & & & \ddots & & \\
\frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_0} & \frac{\partial \mathbf{r}_t}{\partial y_0} & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{r}_t}{\partial y_1} & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_{t-1}} & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_t} & \\
 & & & & & & & f'(\mathbf{x}_t) & -1
\end{bmatrix}
\begin{bmatrix}
\delta \mathbf{x}_0 \\ \delta y_0 \\ \delta \mathbf{x}_1 \\ \delta y_1 \\ \delta \mathbf{x}_2 \\ \delta y_2 \\ \vdots \\ \delta \mathbf{x}_t \\ \delta y_t
\end{bmatrix} = 0.$$

We are interested in sensitivity with respect to $\delta \mathbf{x}_0$, so we rewrite this as

$$\begin{bmatrix}
-1 & & & & & & & \\
\frac{\partial \mathbf{r}_1}{\partial y_0} & \frac{\partial \mathbf{r}_1}{\partial \mathbf{x}_1} & & & & & & \\
 & f'(\mathbf{x}_1) & -1 & & & & & \\
\frac{\partial \mathbf{r}_2}{\partial y_0} & \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{r}_2}{\partial y_1} & \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}_2} & & & & \\
 & & & f'(\mathbf{x}_2) & -1 & & & \\
\vdots & & & & & \ddots & & \\
\frac{\partial \mathbf{r}_t}{\partial y_0} & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{r}_t}{\partial y_1} & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_{t-1}} & \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_t} & \\
 & & & & & & f'(\mathbf{x}_t) & -1
\end{bmatrix}
\begin{bmatrix}
\delta y_0 \\ \delta \mathbf{x}_1 \\ \delta y_1 \\ \delta \mathbf{x}_2 \\ \delta y_2 \\ \vdots \\ \delta \mathbf{x}_t \\ \delta y_t
\end{bmatrix} = -
\begin{bmatrix}
f'(\mathbf{x}_0) \\ \frac{\partial \mathbf{r}_1}{\partial \mathbf{x}_0} \\ 0 \\ \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}_0} \\ 0 \\ \vdots \\ \frac{\partial \mathbf{r}_t}{\partial \mathbf{x}_0} \\ 0
\end{bmatrix} \delta \mathbf{x}_0.$$

We can write this more concisely as

$$Lv = -g\,\delta \mathbf{x}_0,$$

where L is a (block) lower triangular matrix, v is the vector of variations, and $-g\,\delta \mathbf{x}_0$ is the right hand side. We are interested in $\delta y_t = e_m^T v$, where e_m is the last column of the identity matrix. The usual forward mode differentiation corresponds to solving

$$\delta y_t = -e_m^T (L^{-1} g) \delta \mathbf{x}_0;$$

backward mode corresponds to solving

$$\delta y = (-e_m^T L^{-1}) g \delta \mathbf{x}_0$$

We can also write this as $\delta y = \bar{v}^T g \delta \mathbf{x}_0$, where

$$L^T \bar{v} = -e_m$$

Backsubstitution for this system starts with $\bar{y}_t = 1$ and then computes

$$\left(\frac{\partial \mathbf{r}_j}{\partial \mathbf{x}_j}\right)^T \bar{\mathbf{x}}_j = -\nabla f(\mathbf{x}_t) \bar{y}_t - \sum_{i=j+1}^{t} \left(\frac{\partial \mathbf{r}_i}{\partial \mathbf{x}_j}\right)^T \bar{\mathbf{x}}_i$$

$$\bar{y}_j = \sum_{i=j+1}^{t} \left(\frac{\partial \mathbf{r}_i}{\partial y_j}\right)^T \bar{\mathbf{x}}_i$$

for j running backward down to zero. Once we have computed this, we get

$$g^T \bar{v} = \nabla f(\mathbf{x}_0)\bar{y}_0 + \sum_{j=1}^{t} \left(\frac{\partial \mathbf{r}_j}{\partial \mathbf{x}_0} \right)^T \bar{\mathbf{x}}_j$$

This is the gradient of y_t with respect to changes in the $\mathbf{x}_0$, which is what we wanted to compute.

Note that the final gradient depends *linearly* on the function gradients $\nabla f(\mathbf{x}_j)$. Therefore, we can trivially compute the expectation – just replace the random variables $\nabla f(\mathbf{x}_j)$ in the expressions above with their expectations!