Calculating Hessian matrices

Robert M. Gower
Visiting Researcher at Facebook AI NYC, Assistant Professor at Télécom Paris

Joint work with Margarida Pinheiro Mello.

August 12, 2020

Contents

Motivation

The cost of calculating derivatives dictates the design of optimization algorithms.

Motivation

The cost of calculating derivatives dictates the design of optimization algorithms.

In some problems, the premiss is: calculating the Hessian is too expensive.

Motivation

The cost of calculating derivatives dictates the design of optimization algorithms.

In some problems, the premiss is: calculating the Hessian is too expensive.

Calculating the gradient is proportional in cost to a function evaluation

Motivation

The cost of calculating derivatives dictates the design of optimization algorithms.

In some problems, the premiss is: calculating the Hessian is too expensive.

Calculating the gradient is proportional in cost to a function evaluation

So how expensive is it to calculate the Hessian?

Motivation

The cost of calculating derivatives dictates the design of optimization algorithms.

In some problems, the premiss is: calculating the Hessian is too expensive.

Calculating the gradient is proportional in cost to a function evaluation

So how expensive is it to calculate the Hessian?

Function Representation

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$

Function Representation

$$f(h(x_{-1}), g(x_{-1}, x_0))$$

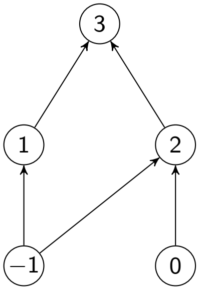
- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m\colon\ y = (y_{1-n}, \ldots, y_{m-n})^T$

Function Representation

$$f(h(x_{-1}), g(x_{-1}, x_0))$$
$$v_{-1} = x_{-1}$$
$$v_0 = x_0$$
$$v_1 = h(v_{-1})$$
$$v_2 = g(v_{-1}, v_0)$$
$$v_3 = f(v_2, v_1)$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$

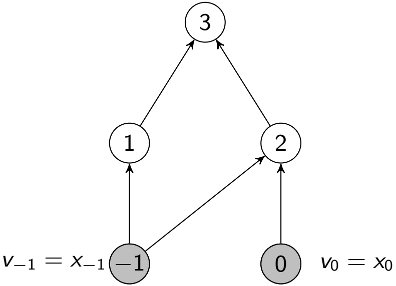
Function Representation



$$f(h(x_{-1}), g(x_{-1}, x_0))$$
$$v_{-1} = x_{-1}$$
$$v_0 = x_0$$
$$v_1 = h(v_{-1})$$
$$v_2 = g(v_{-1}, v_0)$$
$$v_3 = f(v_2, v_1)$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.

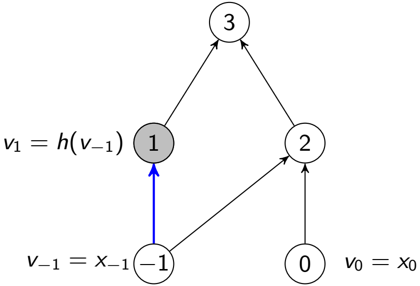
Function Representation



$$f(h(x_{-1}), g(x_{-1}, x_0))$$
$$v_{-1} = x_{-1}$$
$$v_0 = x_0$$
$$v_1 = h(v_{-1})$$
$$v_2 = g(v_{-1}, v_0)$$
$$v_3 = f(v_2, v_1)$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.

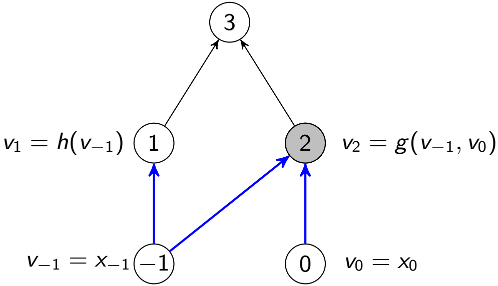
Function Representation



$$
\begin{aligned}
&f(h(x_{-1}), g(x_{-1}, x_0))\\
&v_{-1} = x_{-1}\\
&v_0 = x_0\\
&v_1 = h(v_{-1})\\
&v_2 = g(v_{-1}, v_0)\\
&v_3 = f(v_2, v_1)
\end{aligned}
$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.

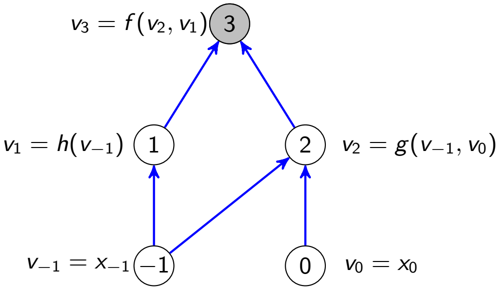
Function Representation



$$
\begin{aligned}
&f(h(x_{-1}), g(x_{-1}, x_0)) \\
&v_{-1} = x_{-1} \\
&v_0 = x_0 \\
&v_1 = h(v_{-1}) \\
&v_2 = g(v_{-1}, v_0) \\
&v_3 = f(v_2, v_1)
\end{aligned}
$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.

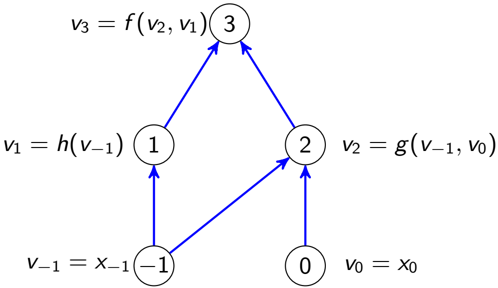
Function Representation



$$
\begin{aligned}
&f(h(x_{-1}), g(x_{-1}, x_0)) \\
&v_{-1} = x_{-1} \\
&v_0 = x_0 \\
&v_1 = h(v_{-1}) \\
&v_2 = g(v_{-1}, v_0) \\
&v_3 = f(v_2, v_1)
\end{aligned}
$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.

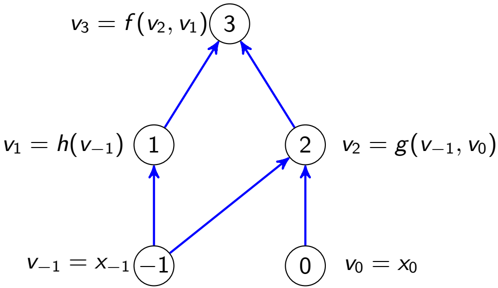
Function Representation



$$
\begin{aligned}
& f(h(x_{-1}), g(x_{-1}, x_0)) \\
& v_{-1} = x_{-1} \\
& v_0 = x_0 \\
& v_1 = h(v_{-1}) \\
& v_2 = g(v_{-1}, v_0) \\
& v_3 = f(v_2, v_1)
\end{aligned}
$$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.
- (j is a predecessor of i) $\equiv j \in P(i)$. e.g. $P(2) = \{0, -1\}$

Function Representation



$f(h(x_{-1}), g(x_{-1}, x_0))$
$v_{-1} = x_{-1}$
$v_0 = x_0$
$v_1 = h(v_{-1})$
$v_2 = g(v_{-1}, v_0)$
$v_3 = f(v_2, v_1)$

- Indices of matrices and vectors shifted by $-n$.
  $y \in \mathbb{R}^m$: $y = (y_{1-n}, \ldots, y_{m-n})^T$
- Node numbering is in order of evaluation.
- (j is a predecessor of i) $\equiv j \in P(i)$. e.g. $P(2) = \{0, -1\}$
- (i is a sucessor of j) $\equiv i \in S(j)$. e.g. $S(2) = \{3\}$

Function Evaluation $\equiv$ Computational Graph

Function Evaluation $\equiv$ Computational Graph

- Nodes for *Independent variables/node:*

$$v_{i-n} = x_{i-n}, \quad \text{for } i = 1, \ldots, n$$

Function Evaluation $\equiv$ Computational Graph

- Nodes for *Independent variables/node:*
  $$v_{i-n} = x_{i-n}, \quad \text{for } i = 1, \ldots, n$$

Function Evaluation $\equiv$ Computational Graph

- Nodes for *Independent variables/node:*

$$v_{i-n} = x_{i-n}, \quad \text{for } i = 1, \ldots, n$$

- Nodes for *Intermediate variables/nodes:*

$$v_i = \phi_i(v_{P(i)}), \quad \text{for } i = 1, \ldots, \ell.$$

Function Evaluation

Function Evaluation ≡ Computational Graph

- Nodes for *Independent variables/node:*
$$v_{i-n} = x_{i-n}, \quad \text{for } i = 1, \ldots, n$$
- Nodes for *Intermediate variables/nodes:*
$$v_i = \phi_i(v_{P(i)}), \quad \text{for } i = 1, \ldots, \ell.$$

Function Evaluation & ϕ set of *elemental* functions with derivatives coded

Function Evaluation $\equiv$ Computational Graph

- Nodes for *Independent variables/node:*
  $$v_{i-n} = x_{i-n}, \quad \text{for } i = 1, \ldots, n$$
- Nodes for *Intermediate variables/nodes:*
  $$v_i = \phi_i(v_{P(i)}), \quad \text{for } i = 1, \ldots, \ell.$$

Function Evaluation & ϕ set of *elemental* functions with derivatives coded
TIME(eval($f(x)$)) = $O(\ell + n)$.

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$v_i = v_j + v_k$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$\begin{aligned} v_i &= v_j + v_k \\ \nabla v_i &= \nabla v_j + \nabla v_k \end{aligned}$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k
\end{aligned}
$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k \\
\nabla v_i &= \nabla v_j \cdot v_k + v_j \cdot \nabla v_k
\end{aligned}
$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k \\
\nabla v_i &= \nabla v_j \cdot v_k + v_j \cdot \nabla v_k \\
v_i &= \phi_i(v_j)
\end{aligned}
$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k \\
\nabla v_i &= \nabla v_j \cdot v_k + v_j \cdot \nabla v_k \\
v_i &= \phi_i(v_j) \\
\nabla v_i &= \nabla v_j \cdot \phi_i'(v_j)
\end{aligned}
$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k \\
\nabla v_i &= \nabla v_j \cdot v_k + v_j \cdot \nabla v_k \\
v_i &= \phi_i(v_j) \\
\nabla v_i &= \nabla v_j \cdot \phi_i'(v_j) \\
\nabla v_i &= \sum_{j \in P(i)} \frac{\partial \phi_i}{\partial v_j} \nabla v_j .
\end{aligned}
$$

Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k \\
\nabla v_i &= \nabla v_j \cdot v_k + v_j \cdot \nabla v_k \\
v_i &= \phi_i(v_j) \\
\nabla v_i &= \nabla v_j \cdot \phi_i'(v_j) \\
\nabla v_i &= \sum_{j \in P(i)} \frac{\partial \phi_i}{\partial v_j} \nabla v_j.
\end{aligned}
$$

With ∇v_j, for $j \in P(i)$, one can calculate ∇v_i.

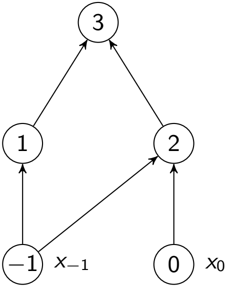
Computing the Gradient: The chain-rule

Set of elemental function = Sums, multiplication and unary functions.

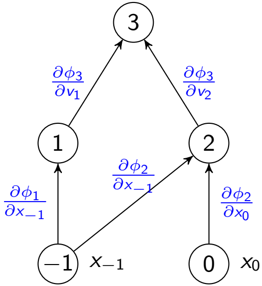
$$
\begin{aligned}
v_i &= v_j + v_k \\
\nabla v_i &= \nabla v_j + \nabla v_k \\
v_i &= v_j \cdot v_k \\
\nabla v_i &= \nabla v_j \cdot v_k + v_j \cdot \nabla v_k \\
v_i &= \phi_i(v_j) \\
\nabla v_i &= \nabla v_j \cdot \phi_i'(v_j) \\
\nabla v_i &= \sum_{j \in P(i)} \frac{\partial \phi_i}{\partial v_j} \nabla v_j.
\end{aligned}
$$

With ∇v_j, for $j \in P(i)$, one can calculate ∇v_i.
Each j passes on $\frac{\partial \phi_i}{\partial v_j} \nabla v_j$ to each sucessor i.

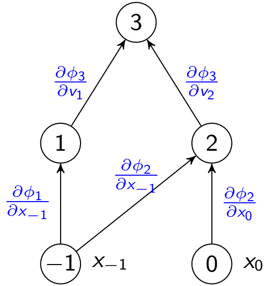
$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$

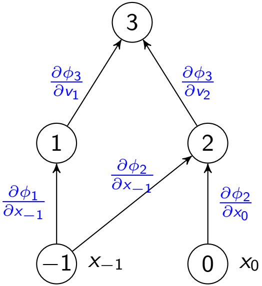


$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



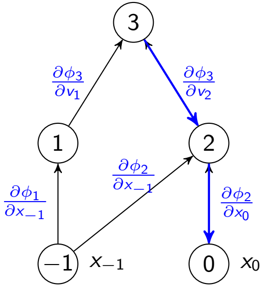
$$\frac{\partial f}{\partial x_0}$$

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



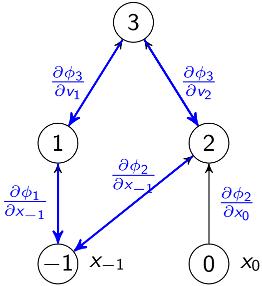
$$\frac{\partial f}{\partial x_0} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial x_0} + 0$$

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$$\frac{\partial f}{\partial x_0} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial x_0} + 0$$

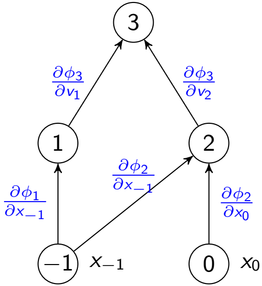
$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$$\frac{\partial f}{\partial x_0} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial x_0} + 0$$

$$\frac{\partial f}{\partial x_{-1}} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial v_{-1}} + \frac{\partial \phi_3}{\partial v_1}\frac{\partial \phi_1}{\partial v_{-1}}$$

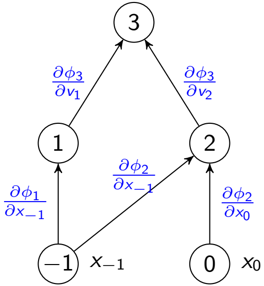
$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$$\frac{\partial f}{\partial x_0} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial x_0} + 0$$

$$\frac{\partial f}{\partial x_{-1}} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial v_{-1}} + \frac{\partial \phi_3}{\partial v_1}\frac{\partial \phi_1}{\partial v_{-1}}$$

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



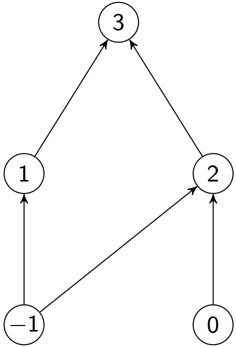
$$\frac{\partial f}{\partial x_0} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial x_0} + 0$$

$$\frac{\partial f}{\partial x_{-1}} = \frac{\partial \phi_3}{\partial v_2}\frac{\partial \phi_2}{\partial v_{-1}} + \frac{\partial \phi_3}{\partial v_1}\frac{\partial \phi_1}{\partial v_{-1}}$$

$$\frac{\partial f}{\partial x_i} = \sum_{p|\text{path from } i \text{ to } \ell} (\text{weight of path } p)$$

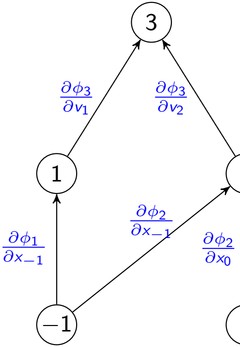
Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



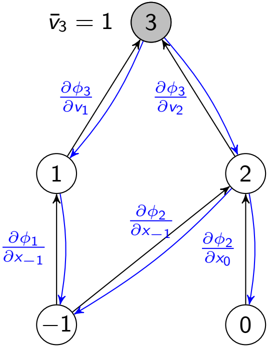
Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



Reverse Gradient - Accumulating paths

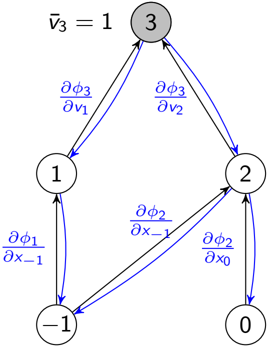
$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$$\bar{v}_i = \sum_{\text{path from } i \text{ to } \ell} (\text{path weight})$$

Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$

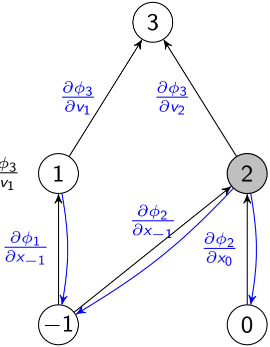


$$\bar{v}_i = \sum_{\text{path from } i \text{ to } \ell} (\text{path weight})$$

$$\bar{v}_j = \sum_{i \in S(j)} \frac{\partial \phi_i}{\partial v_j} \bar{v}_i$$

Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$\bar{v}_1 = \frac{\partial \phi_3}{\partial v_1}$

$\bar{v}_2 = \frac{\partial \phi_3}{\partial v_2}$

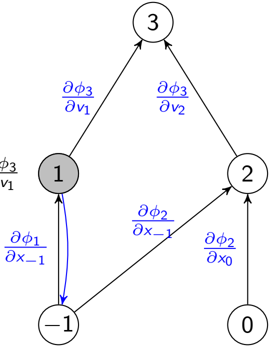
$$\bar{v}_i = \sum_{\text{path from } i \text{ to } \ell} (\text{path weight})$$

$$\bar{v}_j = \sum_{i \in S(j)} \frac{\partial \phi_i}{\partial v_j} \bar{v}_i$$

Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$

$$\bar{v}_1 = \frac{\partial \phi_3}{\partial v_1}$$



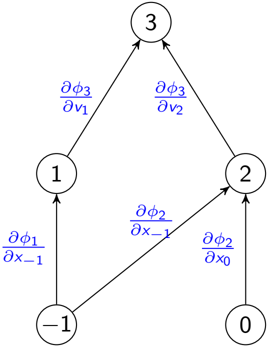
$$\bar{v}_i = \sum_{\text{path from } i \text{ to } \ell} (\text{path weight})$$

$$\bar{v}_j = \sum_{i \in S(j)} \frac{\partial \phi_i}{\partial v_j} \bar{v}_i$$

$$\bar{v}_{-1} = \frac{\partial \phi_2}{\partial x_{-1}} \bar{v}_2 \qquad \bar{v}_0 = \frac{\partial \phi_2}{\partial x_0} \bar{v}_2$$

Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



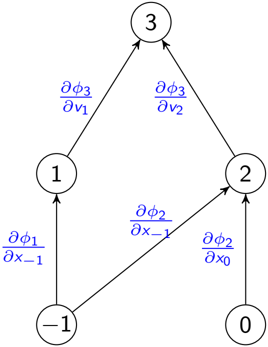
$$\bar{v}_i = \sum_{\text{path from } i \text{ to } \ell} (\text{path weight})$$

$$\bar{v}_j = \sum_{i \in S(j)} \frac{\partial \phi_i}{\partial v_j} \bar{v}_i$$

$$\bar{v}_{-1} = \frac{\partial \phi_1}{\partial x_{-1}} \bar{v}_1 + \frac{\partial \phi_2}{\partial x_{-1}} \bar{v}_2 \qquad \bar{v}_0 = \frac{\partial \phi_2}{\partial x_0} \bar{v}_2$$

Reverse Gradient - Accumulating paths

$$f(x) = \phi_3(\phi_1(x_{-1}), \phi_2(x_{-1}, x_0))$$



$$\bar{v}_{-1} = \frac{\partial \phi_1}{\partial x_{-1}} \bar{v}_1 + \frac{\partial \phi_2}{\partial x_{-1}} \bar{v}_2 \qquad \bar{v}_0 = \frac{\partial \phi_2}{\partial x_0} \bar{v}_2$$

$$\frac{\partial f}{\partial x_{-1}} = \bar{v}_{-1} \qquad \frac{\partial f}{\partial x_0} = \bar{v}_0$$

$$\bar{v}_i = \sum_{\text{path from } i \text{ to } \ell} (\text{path weight})$$

$$\bar{v}_j = \sum_{i \in S(j)} \frac{\partial \phi_i}{\partial v_j} \bar{v}_i$$

$$\text{TIME}(\nabla f(x)) = \text{TIME}(f(x))$$

Forward Hessian: McCormick and Jackson 1986

Forward Hessian: McCormick and Jackson 1986

if $$v_i = v_j + v_k$$

Forward Hessian: McCormick and Jackson 1986

$$\begin{array}{llcl} \textbf{if} & v_i & = & v_j + v_k \\ \textbf{then} & v_i'' & = & v_j'' + v_k'' \end{array}$$

Forward Hessian: McCormick and Jackson 1986

$$
\begin{array}{llcl}
\textbf{if} & v_i & = & v_j + v_k \\
\textbf{then} & v_i'' & = & v_j'' + v_k'' \\
\textbf{if} & v_i & = & v_j \cdot v_k
\end{array}
$$

Forward Hessian: McCormick and Jackson 1986

$$
\begin{array}{llcl}
\textbf{if} & v_i & = & v_j + v_k \\
\textbf{then} & v_i'' & = & v_j'' + v_k'' \\
\textbf{if} & v_i & = & v_j \cdot v_k \\
\textbf{then} & v_i'' & = & v_j \cdot v_k'' + \nabla v_j \cdot \nabla v_k^T + v_j'' \cdot v_k + \nabla v_k \cdot \nabla v_j^T
\end{array}
$$

Forward Hessian: McCormick and Jackson 1986

$$
\begin{array}{lrcl}
\textbf{if} & v_i & = & v_j + v_k \\
\textbf{then} & v_i'' & = & v_j'' + v_k'' \\
\textbf{if} & v_i & = & v_j \cdot v_k \\
\textbf{then} & v_i'' & = & v_j \cdot v_k'' + \nabla v_j \cdot \nabla v_k^T + v_j'' \cdot v_k + \nabla v_k \cdot \nabla v_j^T \\
\textbf{if} & v_i & = & \phi_i(v_j)
\end{array}
$$

Forward Hessian: McCormick and Jackson 1986

$$
\begin{array}{lrcl}
\textbf{if} & v_i & = & v_j + v_k \\
\textbf{then} & v_i'' & = & v_j'' + v_k'' \\
\textbf{if} & v_i & = & v_j \cdot v_k \\
\textbf{then} & v_i'' & = & v_j \cdot v_k'' + \nabla v_j \cdot \nabla v_k^T + v_j'' \cdot v_k + \nabla v_k \cdot \nabla v_j^T \\
\textbf{if} & v_i & = & \phi_i(v_j) \\
\textbf{then} & v_i'' & = & \nabla v_j^T \cdot \phi_i''(v_j) \cdot \nabla v_j + \phi_i'(v_j) \cdot v_j''
\end{array}
$$

Forward Hessian: McCormick and Jackson 1986

$$
\begin{array}{llcl}
\textbf{if} & v_i & = & v_j + v_k \\
\textbf{then} & v_i'' & = & v_j'' + v_k'' \\
\textbf{if} & v_i & = & v_j \cdot v_k \\
\textbf{then} & v_i'' & = & v_j \cdot v_k'' + \nabla v_j \cdot \nabla v_k^T + v_j'' \cdot v_k + \nabla v_k \cdot \nabla v_j^T \\
\textbf{if} & v_i & = & \phi_i(v_j) \\
\textbf{then} & v_i'' & = & \nabla v_j^T \cdot \phi_i''(v_j) \cdot \nabla v_j + \phi_i'(v_j) \cdot v_j''
\end{array}
$$

$$
v_i'' = \sum_{j,k \in P(i)} \nabla v_j \cdot \frac{\partial^2 \phi_i}{\partial v_j \partial v_k} \cdot \nabla v_k^T + \sum_{j \in P(i)} \frac{\partial \phi_i}{\partial v_j} \cdot v_j''
$$

Forward Hessian: McCormick and Jackson 1986

$$
\begin{array}{lcll}
\textbf{if} & v_i & = & v_j + v_k \\
\textbf{then} & v_i'' & = & v_j'' + v_k'' \\
\textbf{if} & v_i & = & v_j \cdot v_k \\
\textbf{then} & v_i'' & = & v_j \cdot v_k'' + \nabla v_j \cdot \nabla v_k^T + v_j'' \cdot v_k + \nabla v_k \cdot \nabla v_j^T \\
\textbf{if} & v_i & = & \phi_i(v_j) \\
\textbf{then} & v_i'' & = & \nabla v_j^T \cdot \phi_i''(v_j) \cdot \nabla v_j + \phi_i'(v_j) \cdot v_j''
\end{array}
$$

$$
v_i'' = \sum_{j,k \in P(i)} \nabla v_j \cdot \frac{\partial^2 \phi_i}{\partial v_j \partial v_k} \cdot \nabla v_k^T + \sum_{j \in P(i)} \frac{\partial \phi_i}{\partial v_j} \cdot v_j''
$$

Each $j, k \in P(i)$ passes on $\nabla v_j \cdot \frac{\partial^2 \phi_i}{\partial v_j \partial v_k}$ and $\sum_{j \in P(i)} \frac{\partial \phi_i}{\partial v_j} \cdot v_j''$ to i

Compute all gradients + Hessians of predecessors

Forward Hessian resume

- For each node, store and calculate a $n \times n$ matrix.

Forward Hessian resume

- For each node, store and calculate a $n \times n$ matrix.
- Is it necessary to calculate the gradient and Hessian of each node?

Forward Hessian resume

- For each node, store and calculate a $n \times n$ matrix.
- Is it necessary to calculate the gradient and Hessian of each node?
- Gain a deeper understanding on the problem using the graph of the reverse gradient

Calculating the Hessian using the computational graph

- Function's computational graph + $\bar{v}_i$ nodes and dependencies = reverse gradient computational graph.

Calculating the Hessian using the computational graph

- Function's computational graph + $\bar{v}_i$ nodes and dependencies = reverse gradient computational graph.
- Interpret partial derivative on augmented graph: Second order derivative.

Calculating the Hessian using the computational graph

- Function's computational graph + $\bar{v}_i$ nodes and dependencies = reverse gradient computational graph.
- Interpret partial derivative on augmented graph: Second order derivative.
- How best to accumulate all second order derivatives? Eliminate unnecessary symmetries on augmented graph.

The adjoint variables of the Reverse Gradient satisfy

The adjoint variables of the Reverse Gradient satisfy

$$\bar{v}_j = \sum_{i \in S(j)} \bar{v}_i \frac{\partial \phi_i}{\partial v_j} \equiv \bar{\varphi}_j.$$

- Gradient's graph has $2(\ell + n)$ nodes: $(v_{1-n}, \ldots, v_\ell)$ and $(\bar{v}_{1-n}, \ldots, \bar{v}_\ell)$.

The adjoint variables of the Reverse Gradient satisfy

$$\bar{v}_j = \sum_{i \in S(j)} \bar{v}_i \frac{\partial \phi_i}{\partial v_j} \equiv \bar{\varphi}_j.$$

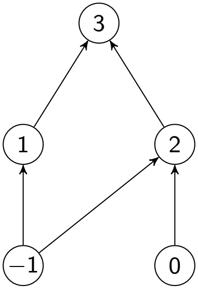
- Gradient's graph has $2(\ell + n)$ nodes: $(v_{1-n}, \ldots, v_\ell)$ and $(\bar{v}_{1-n}, \ldots, \bar{v}_\ell)$.
- node $\bar{\mathrm{j}} \longleftrightarrow \bar{v}_j$.

The adjoint variables of the Reverse Gradient satisfy

$$\bar{v}_j = \sum_{i \in S(j)} \bar{v}_i \frac{\partial \phi_i}{\partial v_j} \equiv \bar{\varphi}_j.$$

- Gradient's graph has $2(\ell + n)$ nodes: $(v_{1-n}, \ldots, v_\ell)$ and $(\bar{v}_{1-n}, \ldots, \bar{v}_\ell)$.
- node $\bar{\jmath} \longleftrightarrow \bar{v}_j$.
- $\bar{i} \in P(\bar{\jmath})$ iff $j \in P(i)$.

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

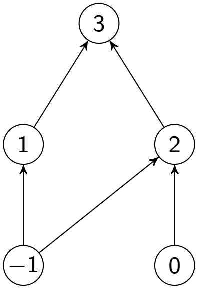


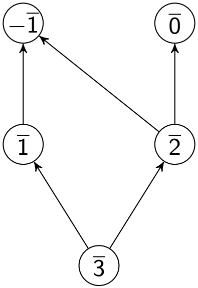
$$\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}$$

$$\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}$$

$f(x) = \sin(x_{-1})(x_{-1} + x_0)$

mirror graph: $\bar{i} \in P(\bar{j})$ iff $j \in P(i)$

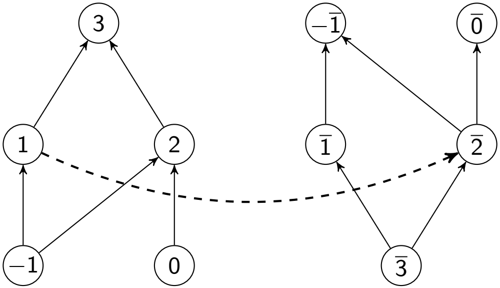




$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \\ \bar{v}_1 &= \bar{v}_3 v_2 \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \end{aligned}$$

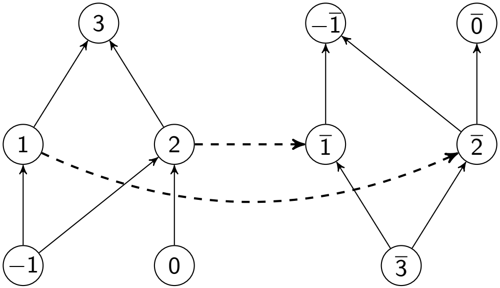
$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



$$\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}$$

$$\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \quad \longleftarrow \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}$$

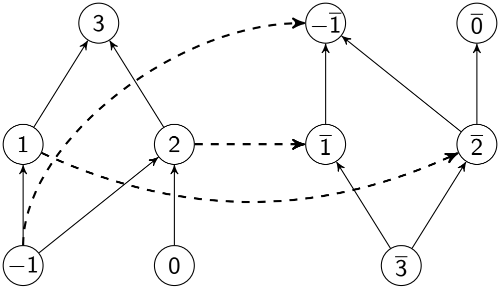
$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \\ \bar{v}_1 &= \bar{v}_3 v_2 \quad \longleftarrow \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \end{aligned}$$

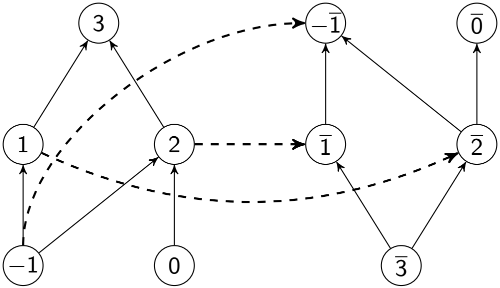
$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



$$
\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \quad \longleftarrow
\end{aligned}
$$

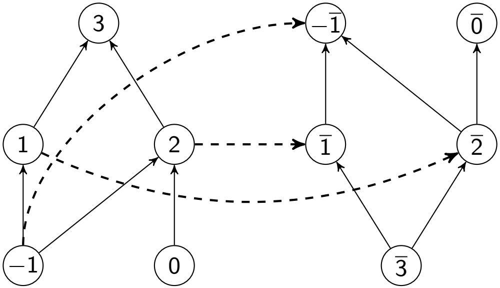
$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \\ \bar{v}_1 &= \bar{v}_3 v_2 \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \end{aligned}$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



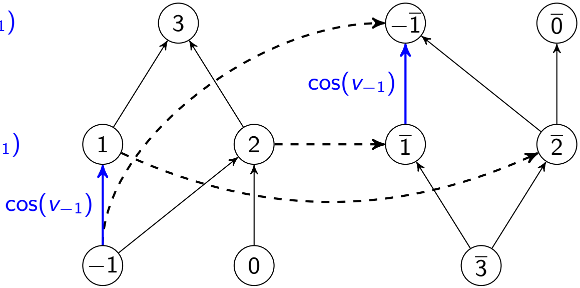
$$
\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}
$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

$$\frac{\partial \phi_1}{\partial v_{-1}} = \cos(v_{-1})$$

$$\frac{\partial \bar{\varphi}_{-1}}{\partial \bar{v}_1} = \cos(v_{-1})$$

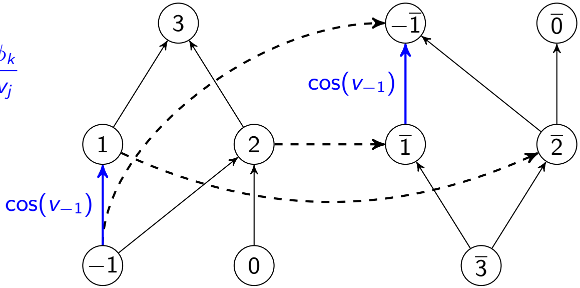


$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \quad \longleftarrow \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \\ \bar{v}_1 &= \bar{v}_3 v_2 \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \quad \longleftarrow \end{aligned}$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

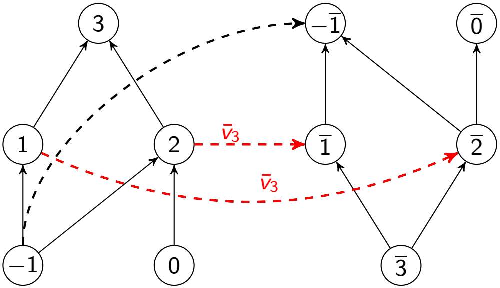
$$\frac{\partial \bar{\varphi}_j}{\partial \bar{v}_k} = c_{kj} = \frac{\partial \phi_k}{\partial v_j}$$



$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \quad \longleftarrow \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \\ \bar{v}_1 &= \bar{v}_3 v_2 \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \quad \longleftarrow \end{aligned}$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



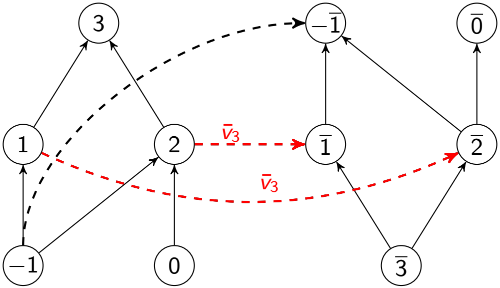
$$
\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \quad \longleftarrow \\
\bar{v}_1 &= \bar{v}_3 v_2 \quad \longleftarrow \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}
$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

$$\frac{\partial \bar{\varphi}_1}{\partial v_2} = \bar{v}_3$$

$$\frac{\partial \bar{\varphi}_2}{\partial v_1} = \bar{v}_3$$

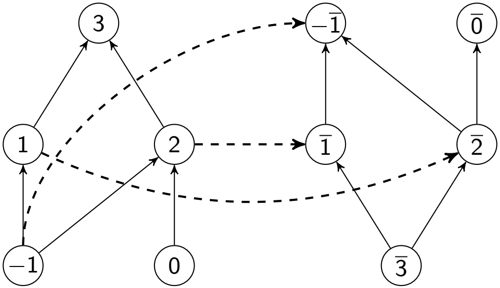


$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \quad \longleftarrow \\ \bar{v}_1 &= \bar{v}_3 v_2 \quad \longleftarrow \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \end{aligned}$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

$$\frac{\partial \bar{\varphi}_j}{\partial \bar{v}_k} = \bar{c}_{kj} = \frac{\partial \bar{\varphi}_k}{\partial \bar{v}_j}$$



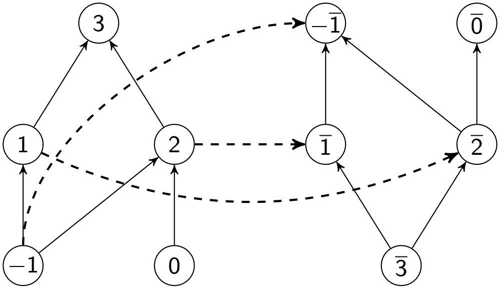
$$\begin{aligned} v_{-1} &= x_{-1} \\ v_0 &= x_0 \\ v_1 &= \sin(v_{-1}) \\ v_2 &= (v_{-1} + v_0) \\ v_3 &= v_1 v_2 \end{aligned}$$

$$\begin{aligned} \bar{v}_3 &= 1 \\ \bar{v}_2 &= \bar{v}_3 v_1 \\ \bar{v}_1 &= \bar{v}_3 v_2 \\ \bar{v}_0 &= \bar{v}_2 1 \\ \bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1}) \end{aligned}$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

$$\frac{\partial \bar{\varphi}_j}{\partial \bar{v}_k} = \bar{c}_{kj} = \frac{\partial \bar{\varphi}_k}{\partial \bar{v}_j}$$

$$\bar{c}_{kj} = \sum_{i \in S(k) \cap S(j)} \bar{v}_i \frac{\partial^2 \phi_i}{\partial v_j \partial v_k}$$



$$
\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}
$$

- Swapped orientation, same weight

$$\frac{\partial \bar{\varphi}_j}{\partial \bar{v}_k} = \frac{\partial}{\partial \bar{v}_k}\left(\sum_{i \in S(j)} \bar{v}_i \frac{\partial \phi_i}{\partial v_j}\right) = \frac{\partial \phi_k}{\partial v_j} \equiv c_{kj}$$

- Swapped orientation, same weight

$$\frac{\partial \bar{\varphi}_j}{\partial \bar{v}_k} = \frac{\partial}{\partial \bar{v}_k}\left(\sum_{i \in S(j)} \bar{v}_i \frac{\partial \phi_i}{\partial v_j}\right) = \frac{\partial \phi_k}{\partial v_j} \equiv c_{kj}$$

- *Nonlinear edges*, swapped orientation, same weight

$$\frac{\partial \bar{\varphi}_j}{\partial v_k} = \sum_{i \in S(j)} \bar{v}_i \frac{\partial^2 \phi_i}{\partial v_j \partial v_k} = \sum_{i \in S(k) \cap S(j)} \bar{v}_i \frac{\partial^2 \phi_i}{\partial v_j \partial v_k} = \frac{\partial \bar{\varphi}_k}{\partial v_j} \equiv \bar{c}_{kj}.$$

- Swapped orientation, same weight

$$\frac{\partial \bar{\varphi}_j}{\partial \bar{v}_k} = \frac{\partial}{\partial \bar{v}_k} \left(\sum_{i \in S(j)} \bar{v}_i \frac{\partial \phi_i}{\partial v_j} \right) = \frac{\partial \phi_k}{\partial v_j} \equiv c_{kj}$$

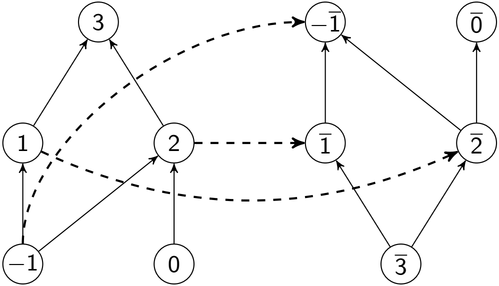
- *Nonlinear edges*, swapped orientation, same weight

$$\frac{\partial \bar{\varphi}_j}{\partial v_k} = \sum_{i \in S(j)} \bar{v}_i \frac{\partial^2 \phi_i}{\partial v_j \partial v_k} = \sum_{i \in S(k) \cap S(j)} \bar{v}_i \frac{\partial^2 \phi_i}{\partial v_j \partial v_k} = \frac{\partial \bar{\varphi}_k}{\partial v_j} \equiv \bar{c}_{kj}.$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \sum_{p | \text{path from } i \text{ to } \bar{j}} (\text{Weight of } p)$$

.

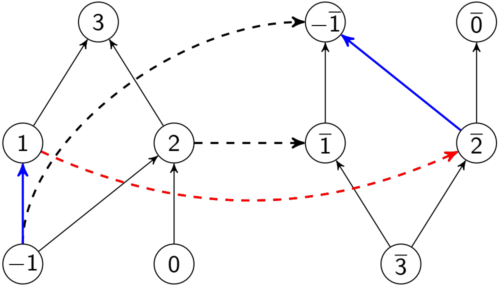
$f(x) = \sin(x_{-1})(x_{-1} + x_0)$



$$
\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}
$$

$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$

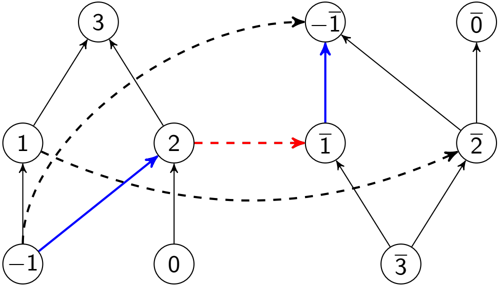


$$\frac{\partial^2 f}{\partial x_{-1}^2} = c_{1\,-1}\,\bar{c}_{21}\,c_{2\,-1}$$

$$\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}$$

$$\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}$$

$f(x) = \sin(x_{-1})(x_{-1} + x_0)$

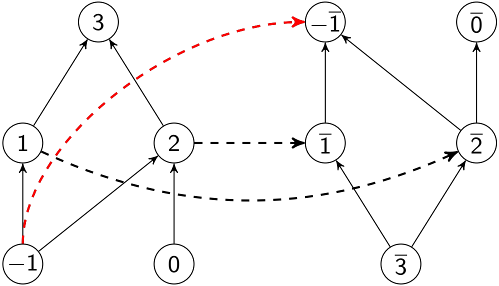


$\frac{\partial^2 f}{\partial x_{-1}^2} = c_{1-1}\bar{c}_{21}c_{2-1} + c_{2-1}\bar{c}_{21}c_{1-1}$

$$
\begin{aligned}
v_{-1} &= x_{-1} \\
v_0 &= x_0 \\
v_1 &= \sin(v_{-1}) \\
v_2 &= (v_{-1} + v_0) \\
v_3 &= v_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_3 &= 1 \\
\bar{v}_2 &= \bar{v}_3 v_1 \\
\bar{v}_1 &= \bar{v}_3 v_2 \\
\bar{v}_0 &= \bar{v}_2 1 \\
\bar{v}_{-1} &= \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})
\end{aligned}
$$

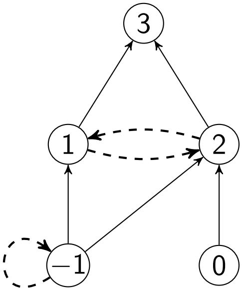
$$f(x) = \sin(x_{-1})(x_{-1} + x_0)$$



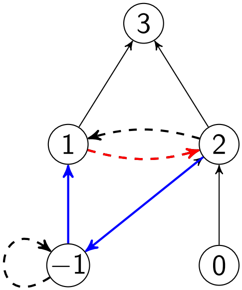
$$\frac{\partial^2 f}{\partial x_{-1}^2} = c_{1-1}\bar{c}_{21}c_{2-1} + c_{2-1}\bar{c}_{21}c_{1-1} + \bar{c}_{-1-1}$$

$$v_{-1} = x_{-1}$$
$$v_0 = x_0$$
$$v_1 = \sin(v_{-1})$$
$$v_2 = (v_{-1} + v_0)$$
$$v_3 = v_1 v_2$$

$$\bar{v}_3 = 1$$
$$\bar{v}_2 = \bar{v}_3 v_1$$
$$\bar{v}_1 = \bar{v}_3 v_2$$
$$\bar{v}_0 = \bar{v}_2 1$$
$$\bar{v}_{-1} = \bar{v}_2 1 + \bar{v}_1 \cos(v_{-1})$$

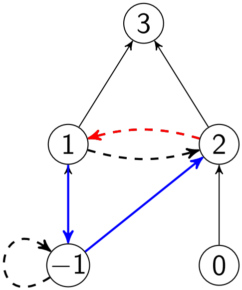


- Fold mirror graph.



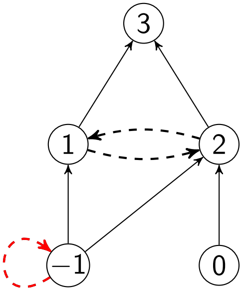
$$\frac{\partial^2 f}{\partial x_{-1}^2} = c_{1-1}\,\bar{c}_{21}\,c_{2-1}$$

- Fold mirror graph.



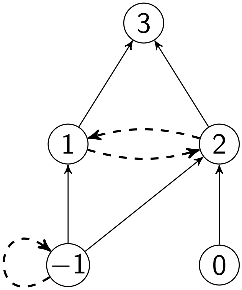
- Fold mirror graph.

$$\frac{\partial^2 f}{\partial x_{-1}^2} = c_{1-1}\bar{c}_{21}c_{2-1} + c_{2-1}\bar{c}_{21}c_{1-1}$$

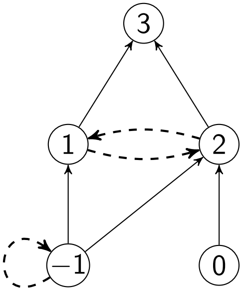


- Fold mirror graph.

$$\frac{\partial^2 f}{\partial x_{-1}^2} = c_{1-1}\bar{c}_{21}c_{2-1} + c_{2-1}\bar{c}_{21}c_{1-1} + \bar{c}_{-1-1}$$

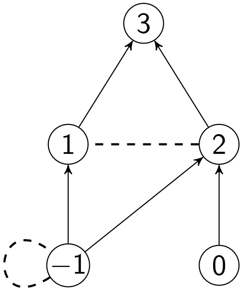


- Fold mirror graph.
- More symmetry



- Fold mirror graph.
- More symmetry
- Symmetric nonlinear edges:

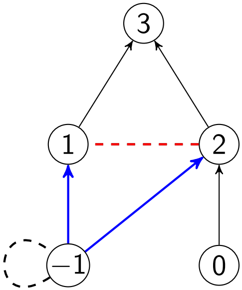
$$k \dashrightarrow j \quad \text{iff} \quad k \dashleftarrow j$$
$$\bar{c}_{kj} = \bar{c}_{jk}$$



- Fold mirror graph.
- More symmetry
- Symmetric nonlinear edges:

$$k \dashrightarrow j \quad \text{iff} \quad k \dashleftarrow j$$

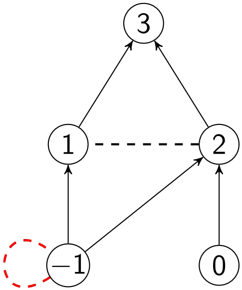
$$\bar{c}_{kj} = \bar{c}_{jk}$$



$$\frac{\partial^2 f}{\partial x_{-1}^2} = 2 c_{1-1} \bar{c}_{21} c_{2-1}$$

- Fold mirror graph.
- More symmetry
- Symmetric nonlinear edges:

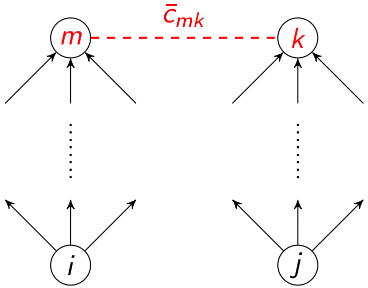
$$k \dashrightarrow j \quad \text{iff} \quad k \dashleftarrow j$$
$$\bar{c}_{kj} = \bar{c}_{jk}$$

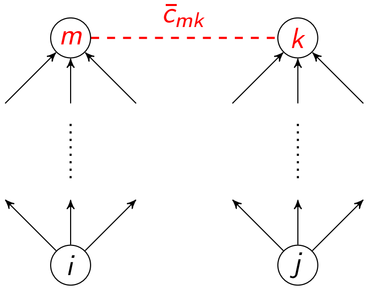


$$\frac{\partial^2 f}{\partial x_{-1}^2} = 2c_{1-1}\bar{c}_{21}c_{2-1} + \bar{c}_{-1-1}$$

- Fold mirror graph.
- More symmetry
- Symmetric nonlinear edges:

$$k \dashrightarrow j \quad \text{iff} \quad k \dashleftarrow j$$
$$\bar{c}_{kj} = \bar{c}_{jk}$$


$\bar{c}_{mk}$
m
k
i
j



$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \sum_{\substack{\text{nonlinear} \\ \text{edge } \{m,k\}}} \sum_{\{p|\text{ from } i \text{ to } m\}} (\text{ weight of } p)\, \bar{c}_{mk} \sum_{\{p|\text{ from } j \text{ to } k\}} (\text{ weight of } p)\,.$$

Building shortcuts

- $P(m) = \{i, j\}$.

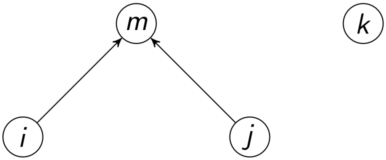


Figure: Pushing the edge $\{m, k\}$

Building shortcuts

- $P(m) = \{i, j\}$.
- $(m, k) \in$ path

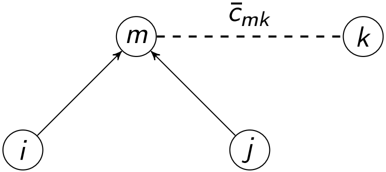


Figure: Pushing the edge $\{m, k\}$

Building shortcuts

- $P(m) = \{i, j\}$.
- $(m, k) \in$ path
- $\Rightarrow$ $(i, m, k) \in$ path and $(j, m, k) \in$ path

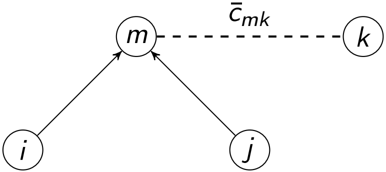


Figure: Pushing the edge $\{m, k\}$

Building shortcuts

- $P(m) = \{i, j\}$.
- $(m, k) \in$ path
- $\Rightarrow (i, m, k) \in$ path and $(j, m, k) \in$ path

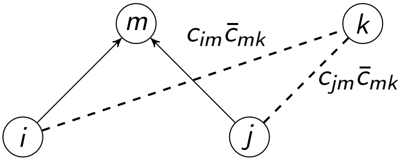
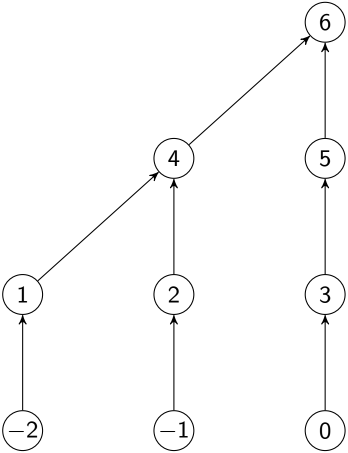


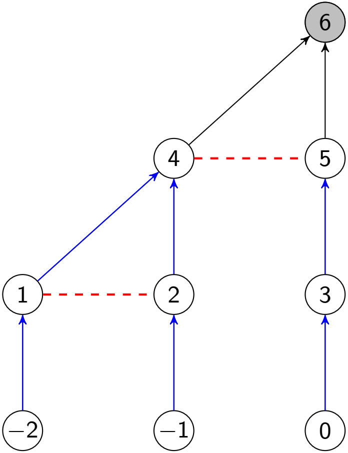
Figure: Pushing the edge $\{m, k\}$

Simple example of edge_pushing execution



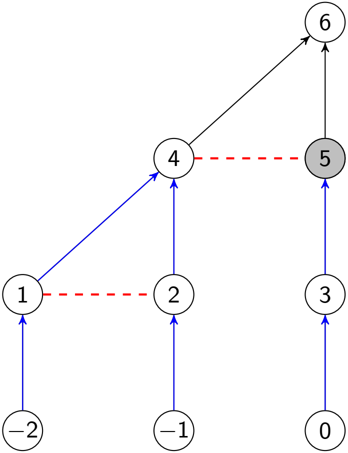
$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$
$v_1 = v_{-2} + 1$
$v_2 = v_{-1} + 1$
$v_3 = v_0 + 1$
$v_4 = v_1 v_2$
$v_5 = 3v_3$
$v_6 = v_4 v_5$
$\bar{v}_6 = 1$
$\bar{v}_5 = v_4$
$\bar{v}_4 = v_5$

Simple example of edge_pushing execution



$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$
$v_1 = v_{-2} + 1$
$v_2 = v_{-1} + 1$
$v_3 = v_0 + 1$
$v_4 = v_1 v_2$
$v_5 = 3v_3$
$v_6 = v_4 v_5$
$\bar{v}_6 = 1$
$\bar{v}_5 = v_4$
$\bar{v}_4 = v_5$

Simple example of edge_pushing execution

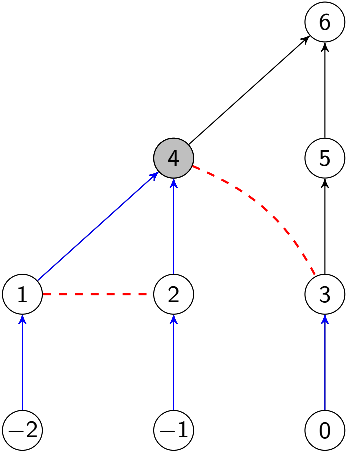


$$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$$

$$v_1 = v_{-2} + 1$$
$$v_2 = v_{-1} + 1$$
$$v_3 = v_0 + 1$$
$$v_4 = v_1 v_2$$
$$v_5 = 3 v_3$$
$$v_6 = v_4 v_5$$

$$\bar{v}_6 = 1$$
$$\bar{v}_5 = v_4$$
$$\bar{v}_4 = v_5$$

Simple example of edge_pushing execution

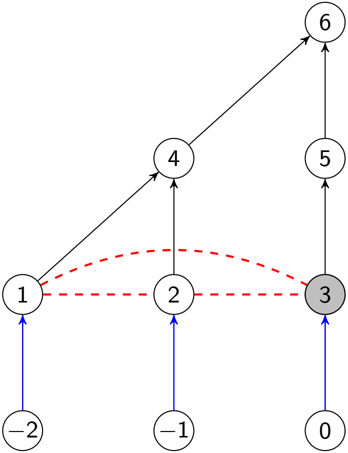


$$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$$

$$\begin{aligned}
v_1 &= v_{-2}+1\\
v_2 &= v_{-1}+1\\
v_3 &= v_0+1\\
v_4 &= v_1 v_2\\
v_5 &= 3v_3\\
v_6 &= v_4 v_5
\end{aligned}$$

$$\begin{aligned}
\bar{v}_6 &= 1\\
\bar{v}_5 &= v_4\\
\bar{v}_4 &= v_5
\end{aligned}$$

Simple example of edge_pushing execution

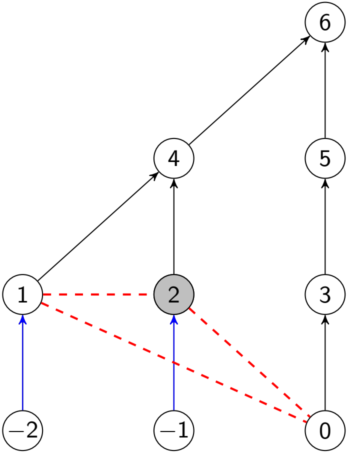


$$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$$

$$v_1 = v_{-2} + 1$$
$$v_2 = v_{-1} + 1$$
$$v_3 = v_0 + 1$$
$$v_4 = v_1 v_2$$
$$v_5 = 3v_3$$
$$v_6 = v_4 v_5$$

$$\bar{v}_6 = 1$$
$$\bar{v}_5 = v_4$$
$$\bar{v}_4 = v_5$$

Simple example of edge_pushing execution

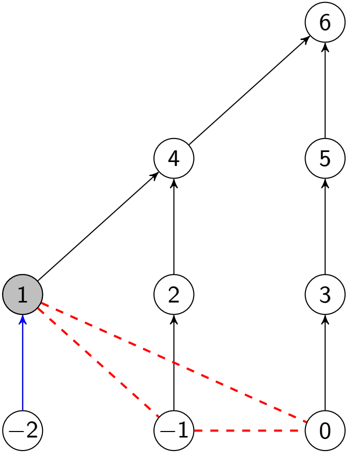


$$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$$

$$\begin{aligned}
v_1 &= v_{-2}+1\\
v_2 &= v_{-1}+1\\
v_3 &= v_0+1\\
v_4 &= v_1 v_2\\
v_5 &= 3v_3\\
v_6 &= v_4 v_5
\end{aligned}$$

$$\begin{aligned}
\bar{v}_6 &= 1\\
\bar{v}_5 &= v_4\\
\bar{v}_4 &= v_5
\end{aligned}$$

Simple example of edge_pushing execution

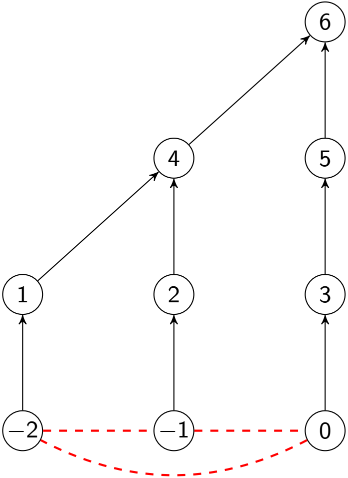


$$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$$

$$v_1 = v_{-2} + 1$$
$$v_2 = v_{-1} + 1$$
$$v_3 = v_0 + 1$$
$$v_4 = v_1 v_2$$
$$v_5 = 3v_3$$
$$v_6 = v_4 v_5$$

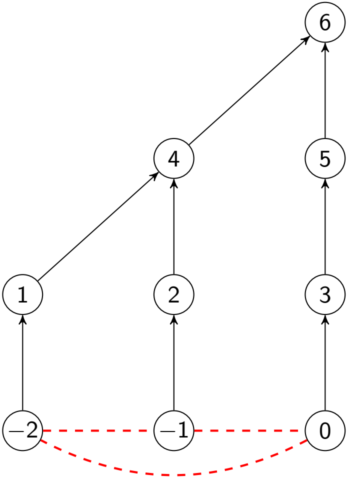
$$\bar{v}_6 = 1$$
$$\bar{v}_5 = v_4$$
$$\bar{v}_4 = v_5$$

Simple example of edge_pushing execution



$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$
$v_1 = v_{-2} + 1$
$v_2 = v_{-1} + 1$
$v_3 = v_0 + 1$
$v_4 = v_1 v_2$
$v_5 = 3v_3$
$v_6 = v_4 v_5$
$\bar{v}_6 = 1$
$\bar{v}_5 = v_4$
$\bar{v}_4 = v_5$

Simple example of edge_pushing execution

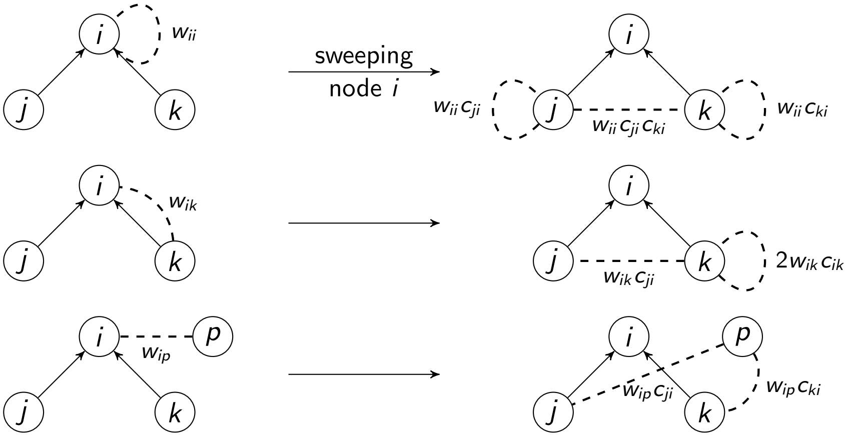


$$f(x) = (x_{-2}+1)(x_{-1}+1)3(x_0+1)$$

$$\begin{aligned} v_1 &= v_{-2}+1 \\ v_2 &= v_{-1}+1 \\ v_3 &= v_0+1 \\ v_4 &= v_1 v_2 \\ v_5 &= 3v_3 \\ v_6 &= v_4 v_5 \end{aligned}$$

$$f'' = \begin{pmatrix} 0 & \frac{\partial^2 f}{\partial x_{-1}\partial x_0} & \frac{\partial^2 f}{\partial x_{-2}\partial x_0} \\ X & 0 & \frac{\partial^2 f}{\partial x_{-1}\partial x_{-2}} \\ X & X & 0 \end{pmatrix}$$

pushing of nonlinear edges

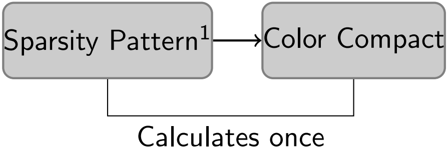


The pseudo-code of `edge_pushing`

Input: $x \in \mathbb{R}^n$,
for $i = \ell, \ldots, 1$ **do**
    Create nonlinear edges if ϕ_i is nonlinear ;
    Push nonlinear edges adjacent to i;
end

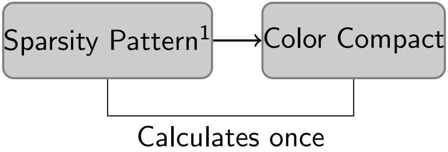
Competitor for `edge_pushing`: Graph coloring

- `edge_pushing` .
  - *A new framework for Hessian automatic differentiation*
    RMG & M. P. Mello, 2012
- Benchmark: graph coloring methods
  - *Efficient Computation of Sparse Hessians Using Coloring and Automatic Differentiation*, A. H. Gebremedhin, A. Pothen, A. Tarafdar & A. Walther, 2009
  - *What Color Is Your Jacobian? Graph Coloring for Computing Derivatives*, A. H. Gebremedhin, F. Manne, A. Pothen, 2005

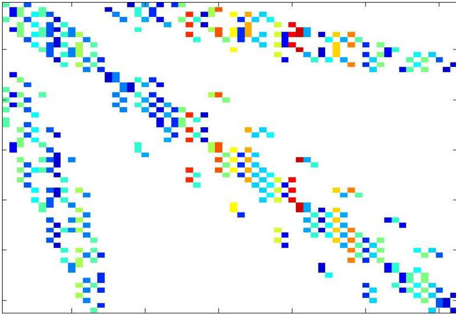


$f''(x)$ $\Rightarrow$ $f''(x)S$

[1] Uses Walther's 2008 algorithm



$f''(x)$



$\Rightarrow$

$f''(x)S$

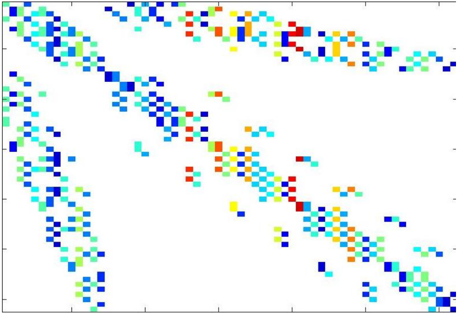
[1] Uses Walther's 2008 algorithm

Sparsity Pattern[1] → Color Compact → Derive Compact → Recover

Calculates once

Repeat for each $x \in \mathbb{R}^n$

$f''(x)$



$\Rightarrow$

$f''(x)S$

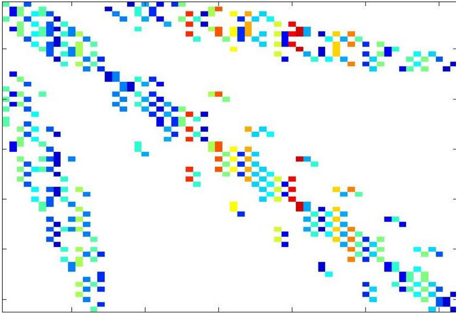
[1] Uses Walther's 2008 algorithm

Sparsity Pattern[1] → Color Compact → Derive Compact → Recover

Calculates once (Sparsity Pattern, Color Compact)

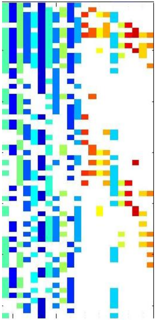
Repeat for each $x \in \mathbb{R}^n$ (Derive Compact, Recover)

$f''(x)$



$\Rightarrow$

$f''(x)S$



[1] Uses Walther's 2008 algorithm

- Invests a large initial time in 1st run $\Rightarrow$ fast subsequent runs.

- Invests a large initial time in 1st run $\Rightarrow$ fast subsequent runs.
- Two different coloring methods with different recoveries: Star and Acyclic.

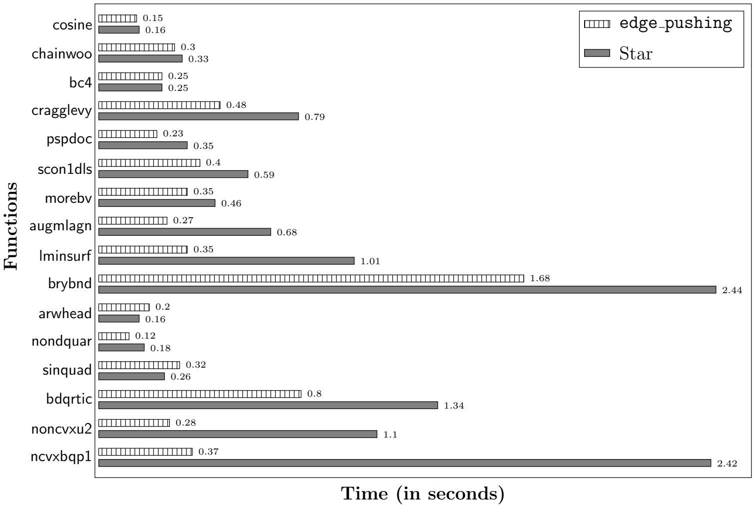
Test set chosen from CUTE

	$n = 50'000$.		
		# colors	
Name	Pattern	Star	Acyclic
cosine	B 1	3	2
chainwoo	B 2	3	3
bc4	B 1	3	2
cragglevy	B 1	3	2
pspdoc	B 2	5	3
scon1dls	B 2	5	3
morebv	B 2	5	3
augmlagn	5×5 diagonal blocks	5	5
lminsurf	B 5	11	6
brybnd	B 5	13	7
arwhead	arrow	2	2
nondquar	arrow + B 1	4	3
sinquad	frame + diagonal	3	3
bdqrtic	arrow + B 3	8	5
noncvxu2	irregular	12	7
ncvxbqp1	irregular	12	7

Numeric Results edge_pushing × Colouring methods

Name	Star 1st	Star 2nd	Acyclic 1st	Acyclic 2nd	e_p
cosine	9.93	0.16	9.68	2.52	0.15
chainwoo	35.07	0.33	33.24	5.08	0.30
bc4	10.02	0.25	10.00	2.56	0.25
cragglevy	28.17	0.79	28.15	2.60	0.48
pspdoc	10.31	0.35	10.27	4.39	0.23
scon1dls	11.00	0.59	10.97	4.96	0.40
morebv	10.36	0.46	10.33	4.49	0.35
augmlagn	15.99	0.68	8.36	16.74	0.27
lminsurf	9.30	1.01	9.24	3.89	0.35
brybnd	11.87	2.44	11.73	12.63	1.68
arwhead	176.50	0.16	45.86	0.24	0.20
nondquar	166.59	0.18	28.64	2.57	0.12
sinquad	606.72	0.26	888.57	1.51	0.32
bdqrtic	262.64	1.34	96.87	7.80	0.80
noncvxu2	29.69	1.10	29.27	7.76	0.28
ncvxbqp1	13.51	2.42	–	–	0.37
Averages	87.98	0.78	82.08	5.32	0.41
Variances	25 083.44	0.54	50 313.10	19.32	0.14

Graphical comparison: Star 2nd run versus edge_pushing.



Summing up

- Graph representation:

Summing up

- Graph representation:
    - New algorithm.

Summing up

- Graph representation:
    - New algorithm.
    - New perspective.

Summing up

- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).

Summing up

- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).
- Algebraic representation:

Summing up

- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).
- Algebraic representation:
  - New correctness.

Summing up

- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).
- Algebraic representation:
  - New correctness.
  - New algorithms.

Summing up

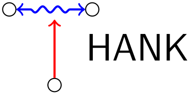
- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).
- Algebraic representation:
  - New correctness.
  - New algorithms.
- `edge_pushing`

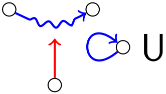
Summing up

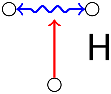
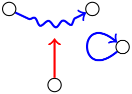
- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).
- Algebraic representation:
  - New correctness.
  - New algorithms.
- `edge_pushing`
  - Exploits the symmetry and sparsity.

Summing up

- Graph representation:
  - New algorithm.
  - New perspective.
  - Propagate from known contributions (nonlinear edges).
- Algebraic representation:
  - New correctness.
  - New algorithms.
- `edge_pushing`
  - Exploits the symmetry and sparsity.
  - Promising test results.


HANK


U

 HANK  U

QUESTIONS?

Follow up work:

- Computing the sparsity pattern of Hessians using automatic differentiation, RMG. and M. P. Mello. , ACM Transactions on Mathematical Software, 2014.
- High order reverse automatic differentiation with emphasis on the third order, RMG. and A. L. Gower, Mathematical Programming, 2014.