Autodiff

Zoltán Szabó

Machine Learning Journal Club, Gatsby Unit

January 11, 2016

Goal, alternatives

- Goal
  1. calculate derivative of a function/computer program,
  2. efficiently and accurately.

Goal, alternatives

- Goal
  1. calculate derivative of a function/computer program,
  2. efficiently and accurately.
- Alternatives:
  1. manual- : time-consuming, error-prone,
  2. numerical- : unstable, bad scaling,
  3. symbolic differentiation : outputs massive expressions $\Rightarrow$ slow.

Goal, alternatives

- Goal
  1. calculate derivative of a function/computer program,
  2. efficiently and accurately.
- Alternatives:
  1. manual- : time-consuming, error-prone,
  2. numerical- : unstable, bad scaling,
  3. symbolic differentiation : outputs massive expressions $\Rightarrow$ slow.
- A(utomatic) D(ifferentiation): symbolic + numerical.

One-page summary: Jacobian

1. Jacobian of $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$
   - forward AD: $c \cdot d_1 \cdot eval(f)$,
   - reverse AD: $c \cdot d_2 \cdot eval(f)$; $c < 6$.

   Shortly, computational time $= c \cdot \min(d_1, d_2) \cdot eval(f)$.

One-page summary: Jacobian/Hessian-vector products

Matrix-free computations:

2. Jacobian-vector products [1-pass, $c \cdot eval(f)$]:
   - $\mathbf{J}_f\mathbf{v}$: forward mode.
   - $\mathbf{J}_f^T\mathbf{v}$: reverse mode.
3. Hessian-vector product ($f : \mathbb{R}^d \to \mathbb{R}$):
   - $\mathbf{H}_f\mathbf{v}$: $O(d)$, although $\mathbf{H} \in \mathbb{R}^{d\times d}$!

Example

- Recursion:

$$r_1(x) = x, \ r_{n+1}(x) = 4r_n(x)[1 - r_n(x)].$$

Example

- Recursion:

$$r_1(x) = x,\ r_{n+1}(x) = 4r_n(x)[1 - r_n(x)].$$

$$\begin{aligned} r'_{n+1}(x) &= 4\left\{r'_n(x)\left[1 - r_n(x)\right] + r_n(x)[-r'_n(x)]\right\} \\ &= 4r'_n(x) - 8r_n(x)r'_n(x). \end{aligned}$$

Example

- Recursion:

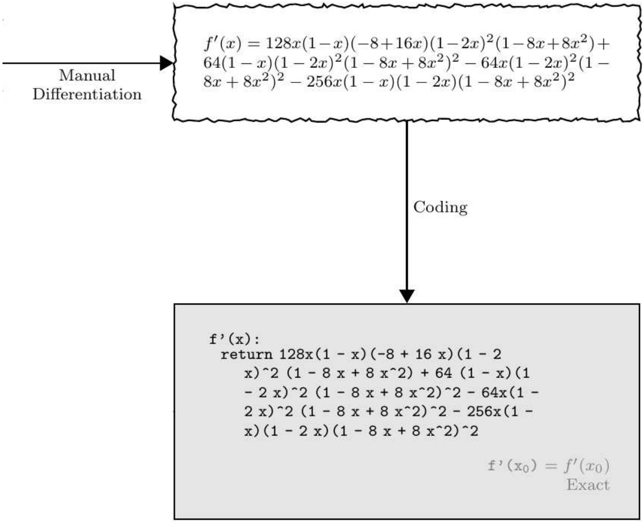
$$r_1(x) = x, \; r_{n+1}(x) = 4r_n(x)[1 - r_n(x)].$$

$$r'_{n+1}(x) = 4\left\{r'_n(x)\left[1 - r_n(x)\right] + r_n(x)[-r'_n(x)]\right\}$$

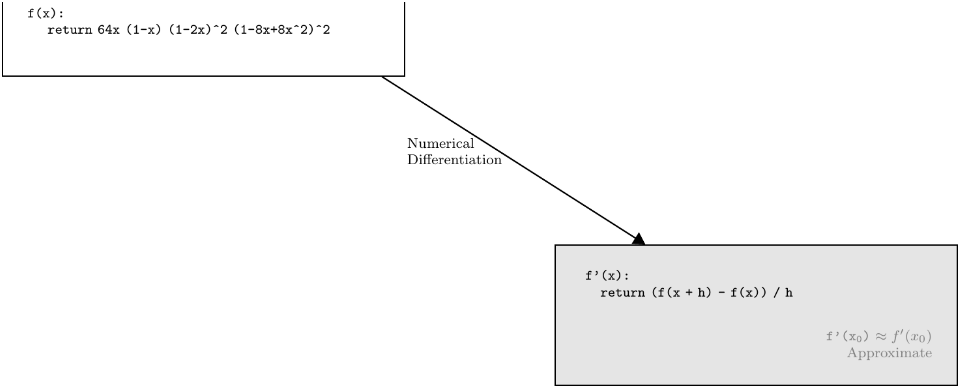
$$= 4r'_n(x) - 8r_n(x)r'_n(x).$$

- Target function (f):
  - $f(x) = r_4(x) = 64x(1-x)(1-2x)^2(1-8x+8x^2)^2$.

Option-1: manual differentiation + coding



Option-2: numerical differentation



Option-2: symbolic differentation of the closed form

Coding

```
f(x):
   v = x
   for i = 1 to 3
     v = 4v(1 - v)
   return v
```

or, in closed-form,

```
f(x):
   return 64x (1-x) (1-2x)^2 (1-8x+8x^2)^2
```

Symbolic Differentiation of the Closed-form

```
f'(x):
 return 128x(1 - x)(-8 + 16 x)(1 - 2
    x)^2 (1 - 8 x + 8 x^2) + 64 (1 - x)(1
    - 2 x)^2 (1 - 8 x + 8 x^2)^2 - 64x(1 -
    2 x)^2 (1 - 8 x + 8 x^2)^2 - 256x(1 -
    x)(1 - 2 x)(1 - 8 x + 8 x^2)^2
```

f'(x_0) $= f'(x_0)$
Exact

Option-4: automatic differentation

```
f(x):
   v = x
   for i = 1 to 3
     v = 4v(1 - v)
   return v
```

or, in closed-form,

```
f(x):
   return 64x (1-x) (1-2x)^2 (1-8x+8x^2)^2
```

Automatic Differentiation

```
f'(x):
   (v,v') = (x,1)
   for i = 1 to 3
     (v,v') = (4v(1-v), 4v'-8vv')
   return (v,v')
```

f'(x_0) $= f'(x_0)$
Exact

Assume $f : \mathbb{R}^{d_1} \to \mathbb{R}$; $h > 0$.

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i)}{h}, \qquad \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}.$$

- Complexity: $(d_1 + 1) \cdot eval(f) \to 2d_1 \cdot eval(f)$.
- Truncation error: $O(h) \to O(h^2)$.

Assume $f : \mathbb{R}^{d_1} \to \mathbb{R}$; $h > 0$.

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i)}{h}, \qquad \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}.$$

- Complexity: $(d_1 + 1) \cdot eval(f) \to 2d_1 \cdot eval(f)$.
- Truncation error: $O(h) \to O(h^2)$.

Higher order schemes:

- increased complexity, floating point errors remain.

3: Symbolic differentiation

- Computer algebra systems:
  - automatic manipulation of expressions.
  - Mathematica, Maple, Maxima.
- Manual/numerical weaknesses: ✓
- Massive & cryptic expressions.
- Requires: closed form expressions (a la manual).

3 main properties:

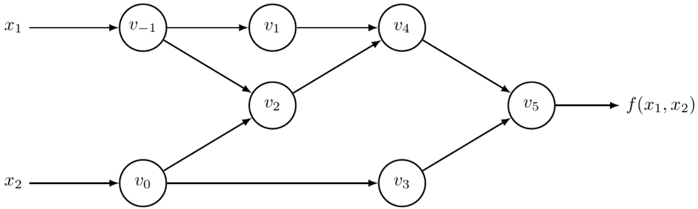
1. *Symbolic* differentiation: for the elementary operations.
2. Smart book-keeping: computational graph.
3. It gives a *numerical* value.

Notes:

- 2 modes: forward/reverse accumulation (chain rule).
- Optimal Jacobian accumulation = NP-complete.

AD-forward: $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$; $\left.\frac{\partial f}{\partial x_1}\right|_{(2,5)} = ?$

Forward Evaluation Trace		
v_{-1}	$= x_1$	$= 2$
v_0	$= x_2$	$= 5$
v_1	$= \ln v_{-1}$	$= \ln 2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$
v_3	$= \sin v_0$	$= \sin 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$
y	$= v_5$	$= 11.652$



AD-forward: $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$; $\left.\frac{\partial f}{\partial x_1}\right|_{(2,5)} = ?$

Forward Evaluation Trace		
v_{-1}	$= x_1$	$= 2$
v_0	$= x_2$	$= 5$
v_1	$= \ln v_{-1}$	$= \ln 2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$
v_3	$= \sin v_0$	$= \sin 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$
y	$= v_5$	$= 11.652$

Forward Derivative Trace		
$\dot{v}_{-1}$	$= \dot{x}_1$	$= 1$
$\dot{v}_0$	$= \dot{x}_2$	$= 0$
$\dot{v}_1$	$= \dot{v}_{-1}/v_{-1}$	$= 1/2$
$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + \dot{v}_0 \times v_{-1}$	$= 1 \times 5 + 0 \times 2$
$\dot{v}_3$	$= \dot{v}_0 \times \cos v_0$	$= 0 \times \cos 5$
$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
$\boldsymbol{\dot{y}}$	$\boldsymbol{= \dot{v}_5}$	$\boldsymbol{= 5.5}$

$$\dot{v}_i = \frac{\partial v_i}{\partial x_1} \Rightarrow \dot{v}_5 = \frac{\partial y}{\partial x_1}.$$

AD-forward

For $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$

- $\mathbf{x} = \mathbf{a}$, $\dot{\mathbf{x}} = \mathbf{e}_i$ gives the i^{th} column of

$$\mathbf{J}_f(\mathbf{a}) = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_{d_1}} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_{d_2}}{\partial x_1} & \cdots & \frac{\partial f_{d_2}}{\partial x_{d_1}} \end{bmatrix}\right|_{\mathbf{x}=\mathbf{a}}.$$

Consequence: $d_2 = 1$ would also require d_1 passes!

AD-forward

For $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$

- $\mathbf{x} = \mathbf{a}$, $\dot{\mathbf{x}} = \mathbf{e}_i$ gives the i^{th} column of

$$\mathbf{J}_f(\mathbf{a}) = \left. \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_{d_1}} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_{d_2}}{\partial x_1} & \cdots & \frac{\partial f_{d_2}}{\partial x_{d_1}} \end{bmatrix} \right|_{\mathbf{x}=\mathbf{a}} .$$

  Consequence: $d_2 = 1$ would also require d_1 passes!
- $\mathbf{x} = \mathbf{a}$, $\dot{\mathbf{x}} = \mathbf{v}$ produces $\mathbf{J}_f(\mathbf{a})\mathbf{v} \leftarrow$ 1-pass, matrix-free!

Dual numbers

- Recall: $\boxed{v_3 = \sin(v_0) \Rightarrow \dot{v}_3 = \cos(v_0)\dot{v}_0} \Rightarrow$ Idea: compute/store function values & derivatives together.
- Def.:

$$\mathbb{D} = \left\{v + \dot{v}\epsilon : \epsilon^2 = 0, (v, \dot{v}) \in \mathbb{R}^2\right\} = \mathbb{R}[\epsilon]/\left(\epsilon^2\right) = \left\{\begin{bmatrix} v & \dot{v} \\ 0 & v \end{bmatrix}\right\}.$$

Arithmetic:

$$\begin{aligned}(v + \dot{v}\epsilon) + (u + \dot{u}\epsilon) &= (u + v) + (\dot{v} + \dot{v})\epsilon,\\ (v + \dot{v}\epsilon)(u + \dot{u}\epsilon) &= (vu) + (v\dot{u} + \dot{v}u)\epsilon.\end{aligned}$$

Dual numbers: extension of functions

- Recall: $\boxed{v_3 = \sin(v_0) \Rightarrow \dot{v}_3 = \cos(v_0)\dot{v}_0}$.
- Let us extend a $g : \mathbb{R} \to \mathbb{R}$ to $\tilde{g} : \mathbb{D} \to \mathbb{D}$ as

$$\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon.$$

Dual numbers: extension of functions

- Recall: $\boxed{v_3 = \sin(v_0) \Rightarrow \dot{v}_3 = \cos(v_0)\dot{v}_0}$.
- Let us extend a $g : \mathbb{R} \to \mathbb{R}$ to $\tilde{g} : \mathbb{D} \to \mathbb{D}$ as

$$\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon.$$

- Compatibility with the chain rule:

$$\widetilde{(f \circ g)}(v + \dot{v}\epsilon) = (f \circ g)(v) + (f \circ g)'(v)\dot{v}\epsilon$$

Dual numbers: extension of functions

- Recall: $\boxed{v_3 = \sin(v_0) \Rightarrow \dot{v}_3 = \cos(v_0)\dot{v}_0}$.
- Let us extend a $g : \mathbb{R} \to \mathbb{R}$ to $\tilde{g} : \mathbb{D} \to \mathbb{D}$ as

$$\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon.$$

- Compatibility with the chain rule:

$$\begin{aligned}\widetilde{(f \circ g)}(v + \dot{v}\epsilon) &= (f \circ g)(v) + (f \circ g)'(v)\dot{v}\epsilon \\ &= f(g(v)) + f'(g(v))g'(v)\dot{v}\epsilon\end{aligned}$$

Dual numbers: extension of functions

- Recall: $\boxed{v_3 = \sin(v_0) \Rightarrow \dot{v}_3 = \cos(v_0)\dot{v}_0}$.
- Let us extend a $g : \mathbb{R} \to \mathbb{R}$ to $\tilde{g} : \mathbb{D} \to \mathbb{D}$ as

$$\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon.$$

- Compatibility with the chain rule:

$$\begin{aligned}\widetilde{(f \circ g)}(v + \dot{v}\epsilon) &= (f \circ g)(v) + (f \circ g)'(v)\dot{v}\epsilon \\ &= f(g(v)) + f'(g(v))g'(v)\dot{v}\epsilon \\ &= \tilde{f}[g(v) + g'(v)\dot{v}\epsilon]\end{aligned}$$

Dual numbers: extension of functions

- Recall: $\boxed{v_3 = \sin(v_0) \Rightarrow \dot{v}_3 = \cos(v_0)\dot{v}_0}$.
- Let us extend a $g : \mathbb{R} \to \mathbb{R}$ to $\tilde{g} : \mathbb{D} \to \mathbb{D}$ as

$$\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon.$$

- Compatibility with the chain rule:

$$\begin{aligned}
\widetilde{(f \circ g)}(v + \dot{v}\epsilon) &= (f \circ g)(v) + (f \circ g)'(v)\dot{v}\epsilon \\
&= f(g(v)) + f'(g(v))g'(v)\dot{v}\epsilon \\
&= \tilde{f}[g(v) + g'(v)\dot{v}\epsilon] \\
&= \tilde{f}[\tilde{g}(v + \dot{v}\epsilon)].
\end{aligned}$$

The extension gives the 'natural' operation on $\mathbb{D}$

- Definition for $g : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$: $\tilde{g}(\mathbf{v} + \dot{\mathbf{v}}\epsilon) := g(\mathbf{v}) + J_g(\mathbf{v})\dot{\mathbf{v}}\epsilon$.
- Example: $+$: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$$\tilde{+}(v_1 + \dot{v}_1\epsilon, v_2 + \dot{v}_2\epsilon) = +(v_1, v_2) + J_+(v_1, v_2)\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon$$

The extension gives the 'natural' operation on $\mathbb{D}$

- Definition for $g : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$: $\tilde{g}(\mathbf{v} + \dot{\mathbf{v}}\epsilon) := g(\mathbf{v}) + J_g(\mathbf{v})\dot{\mathbf{v}}\epsilon$.
- Example: $+$: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$$\begin{aligned}\tilde{+}(v_1 + \dot{v}_1\epsilon, v_2 + \dot{v}_2\epsilon) &= +(v_1, v_2) + J_+(v_1, v_2)\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon\\ &= (v_1 + v_2) + [1, 1]\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon\end{aligned}$$

- Definition for $g: \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$: $\tilde{g}(\mathbf{v} + \dot{\mathbf{v}}\epsilon) := g(\mathbf{v}) + J_g(\mathbf{v})\dot{\mathbf{v}}\epsilon$.
- Example: $+$: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$$\begin{aligned}
\tilde{+}(v_1 + \dot{v}_1\epsilon, v_2 + \dot{v}_2\epsilon) &= +(v_1, v_2) + J_+(v_1, v_2)\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon\\
&= (v_1 + v_2) + [1, 1]\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon\\
&= (v_1 + v_2) + (\dot{v}_1 + \dot{v}_2)\epsilon
\end{aligned}$$

The extension gives the 'natural' operation on $\mathbb{D}$

- Definition for $g: \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$: $\tilde{g}(\mathbf{v} + \dot{\mathbf{v}}\epsilon) := g(\mathbf{v}) + J_g(\mathbf{v})\dot{\mathbf{v}}\epsilon$.
- Example: $+$: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$$
\begin{aligned}
\tilde{+}(v_1 + \dot{v}_1\epsilon, v_2 + \dot{v}_2\epsilon) &= +(v_1, v_2) + J_+(v_1, v_2) \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} \epsilon \\
&= (v_1 + v_2) + [1, 1] \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} \epsilon \\
&= (v_1 + v_2) + (\dot{v}_1 + \dot{v}_2)\epsilon \\
&= (v_1 + \dot{v}_1\epsilon) + (v_2 + \dot{v}_2\epsilon).
\end{aligned}
$$

The extension gives the 'natural' operation on $\mathbb{D}$

- Definition for $g:\mathbb{R}^{d_1}\to\mathbb{R}^{d_2}$: $\tilde{g}(\mathbf{v}+\dot{\mathbf{v}}\epsilon) := g(\mathbf{v}) + J_g(\mathbf{v})\dot{\mathbf{v}}\epsilon$.
- Example: $+$: $\mathbb{R}\times\mathbb{R}\to\mathbb{R}$

$$\begin{aligned}\tilde{+}(v_1+\dot{v}_1\epsilon, v_2+\dot{v}_2\epsilon) &= +(v_1,v_2) + J_+(v_1,v_2)\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon\\ &= (v_1+v_2) + [1,1]\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix}\epsilon\\ &= (v_1+v_2)+(\dot{v}_1+\dot{v}_2)\epsilon\\ &= (v_1+\dot{v}_1\epsilon)+(v_2+\dot{v}_2\epsilon).\end{aligned}$$

- The same holds for: $\times$, $/$, polynomials, analytic functions.

Dual numbers: example

- Recall: $\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v + 1\epsilon)]$.

Dual numbers: example

- Recall: $\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v + 1\epsilon)]$.
- $g(v) = \frac{v^2}{v+1}$, $g'(2) =? = D[\tilde{g}(2 + 1\epsilon)]$

Dual numbers: example

- Recall: $\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v + 1\epsilon)]$.
- $g(v) = \frac{v^2}{v+1}$, $g'(2) =? = D[\tilde{g}(2 + 1\epsilon)]$
- Computation:

$$\tilde{g}(2 + \epsilon) = \frac{(2 + \epsilon)^2}{(2 + \epsilon) + 1}$$

Dual numbers: example

- Recall: $\tilde{g}(v+\dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v+1\epsilon)]$.
- $g(v) = \frac{v^2}{v+1}$, $g'(2) =? = D[\tilde{g}(2+1\epsilon)]$
- Computation:

$$\tilde{g}(2+\epsilon) = \frac{(2+\epsilon)^2}{(2+\epsilon)+1} = \frac{4+4\epsilon}{3+\epsilon}$$

Dual numbers: example

- Recall: $\tilde{g}(v+\dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v+1\epsilon)]$.
- $g(v) = \frac{v^2}{v+1}$, $g'(2) =? = D[\tilde{g}(2+1\epsilon)]$
- Computation:

$$\tilde{g}(2+\epsilon) = \frac{(2+\epsilon)^2}{(2+\epsilon)+1} = \frac{4+4\epsilon}{3+\epsilon} \stackrel{(*)}{=} \frac{4}{3} + \frac{4\times 3 - 4\times 1}{3^2}\epsilon$$

using $(*)$: $\frac{a+b\epsilon}{c+d\epsilon} = \frac{a+b\epsilon}{c+d\epsilon}\frac{c-d\epsilon}{c-d\epsilon} = \frac{ac+(bc-ad)\epsilon}{c^2\pm c\epsilon} = \frac{a}{c} + \frac{bc-ad}{c^2}\epsilon$.

Dual numbers: example

- Recall: $\tilde{g}(v+\dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v+1\epsilon)]$.
- $g(v) = \frac{v^2}{v+1}$, $g'(2) =? = D[\tilde{g}(2+1\epsilon)]$
- Computation:

$$\tilde{g}(2+\epsilon) = \frac{(2+\epsilon)^2}{(2+\epsilon)+1} = \frac{4+4\epsilon}{3+\epsilon} \overset{(*)}{=} \frac{4}{3} + \frac{4\times 3 - 4\times 1}{3^2}\epsilon$$
$$= \frac{4}{3} + \frac{8}{9}\epsilon$$

using $(*)$: $\frac{a+b\epsilon}{c+d\epsilon} = \frac{a+b\epsilon}{c+d\epsilon}\frac{c-d\epsilon}{c-d\epsilon} = \frac{ac+(bc-ad)\epsilon}{c^2 \pm c\epsilon} = \frac{a}{c} + \frac{bc-ad}{c^2}\epsilon$.

Dual numbers: example

- Recall: $\tilde{g}(v + \dot{v}\epsilon) = g(v) + g'(v)\dot{v}\epsilon \Rightarrow g'(v) = D[\tilde{g}(v + 1\epsilon)]$.
- $g(v) = \frac{v^2}{v+1}$, $g'(2) =? = D[\tilde{g}(2 + 1\epsilon)]$
- Computation:

$$\tilde{g}(2+\epsilon) = \frac{(2+\epsilon)^2}{(2+\epsilon)+1} = \frac{4+4\epsilon}{3+\epsilon} \overset{(*)}{=} \frac{4}{3} + \frac{4\times 3 - 4\times 1}{3^2}\epsilon$$
$$= \frac{4}{3} + \frac{8}{9}\epsilon \Rightarrow g'(2) = \frac{8}{9}$$

using $(*)$: $\frac{a+b\epsilon}{c+d\epsilon} = \frac{a+b\epsilon}{c+d\epsilon}\frac{c-d\epsilon}{c-d\epsilon} = \frac{ac+(bc-ad)\epsilon}{c^2 \pm c\epsilon} = \frac{a}{c} + \frac{bc-ad}{c^2}\epsilon$.

AD-reverse: $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$; $\left.\frac{\partial f}{\partial x_1}\right|_{(2,5)} = ?$

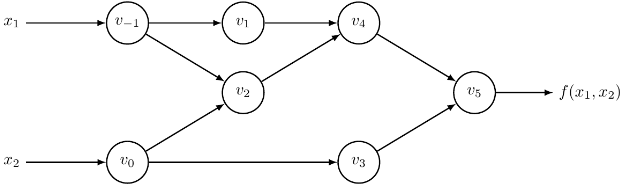
$$v_i \mapsto \bar{v}_i = \frac{\partial y}{\partial v_i} \text{ adjoint variable. } \frac{\partial y}{\partial v_0} = \frac{\partial y}{\partial v_2}\frac{\partial v_2}{\partial v_0} + \frac{\partial y}{\partial v_3}\frac{\partial v_3}{\partial v_0}, \ldots$$

Forward Evaluation Trace

$v_{-1} = x_1$		$= 2$
$v_0 = x_2$		$= 5$
$v_1 = \ln v_{-1}$		$= \ln 2$
$v_2 = v_{-1} \times v_0$		$= 2 \times 5$
$v_3 = \sin v_0$		$= \sin 5$
$v_4 = v_1 + v_2$		$= 0.693 + 10$
$v_5 = v_4 - v_3$		$= 10.693 + 0.959$
$y = v_5$		$= 11.652$

Reverse Adjoint Trace

$\bar{x}_1 = \bar{v}_{-1}$		$= \mathbf{5.5}$
$\bar{x}_2 = \bar{v}_0$		$= \mathbf{1.716}$
$\bar{v}_{-1} = \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \bar{v}_1 / v_{-1}$	$= 5.5$
$\bar{v}_0 = \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$	$= 1.716$
$\bar{v}_{-1} = \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$	$= 5$
$\bar{v}_0 = \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$	$= -0.284$
$\bar{v}_2 = \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_1 = \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_3 = \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$	$= -1$
$\bar{v}_4 = \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$	$= 1$
$\bar{v}_5 = \bar{y}$	$= 1$	



AD-reverse: matrix-free Jacobi-vector product

For $f:\mathbb{R}^{d_1}\to\mathbb{R}^{d_2}$

- $\mathbf{x}=\mathbf{a}$, $\bar{\mathbf{y}}=\mathbf{e}_i$ produces the i^{th} column of

$$\mathbf{J}_f^T(\mathbf{a}) = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_{d_2}}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_1}{\partial x_{d_1}} & \cdots & \frac{\partial f_{d_2}}{\partial x_{d_1}} \end{bmatrix}\right|_{\mathbf{x}=\mathbf{a}}.$$

  Specifically, if $d_2=1$ we get ∇f in one pass!
- $\mathbf{x}=\mathbf{a}$, $\bar{\mathbf{y}}=\mathbf{r}$ gives $\mathbf{J}_f^T(\mathbf{a})\mathbf{r}$.

Hessian-vector product: forward & reverse-AD

- $f:\mathbb{R}^d \to \mathbb{R}$.
- $\mathbf{H}_f = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$. Goal: $\mathbf{H}_f\mathbf{v}$.
- Steps:
  1. $\langle \nabla f, \mathbf{v}\rangle$: forward mode, with $\dot{\mathbf{x}} = \mathbf{v}$.
  2. $\mathbf{H}_f\mathbf{v}$: apply reverse-AD on the produced forward code.
- Complexity: $O(d)$! – $\mathbf{H}_f \in \mathbb{R}^{d\times d}$

Implementation

4 ways:

1. Elemental libraries: $\sin \rightarrow \sin_{\mathrm{AD}}$.
2. Preprocessors:
   - source code transformation,
   - auto decomposition to AD-enabled elementary operarations.
3. New languages: tightly integrated AD-capabilities (compilers).

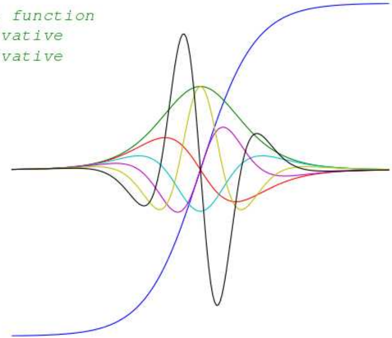
Implementation – continued

4. Operator overloading:
   - redefine elementary operations,
   - Example: autograd/Theano $\in$ Python.

Languages: AMPL, C, C++, C#, F#, Fortran, Haskell, Java, Matlab, Python, Scheme, Stan (see [1]: Table 5).

Example in Python

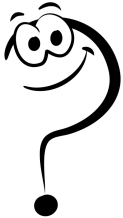
```python
import autograd.numpy as np # Thinly—wrapped numpy
from autograd import grad   # grad(f) returns f'

def f(x):                   # Define a function
    y = np.exp(-x)
    return (1.0 - y)  / ( 1.0 + y)

D_f  = grad(f)  # Obtain gradient function
D2_f = grad(D_f)        # 2nd derivative
D3_f = grad(D2_f)       # 3rd derivative
D4_f = grad(D3_f)       # etc.
D5_f = grad(D4_f)
D6_f = grad(D5_f)

import matplotlib.pyplot as plt
x = np.linspace(-7, 7, 200)
plt.plot(x, map(f, x),
         x, map(D_f , x),
         x, map(D2_f , x),
         x, map(D3_f , x),
         x, map(D4_f , x),
         x, map(D5_f , x),
         x, map(D6_f , x))
plt.show()
```



Further reading

- Refs ([1] = main source):
  1. Atılım Günes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, Jeffrey Mark Siskind. Automatic Differentiation in Machine Learning: A Survey. arXiv, 2015.
  2. Philipp Hoffmann. A Hitchiker's Guide to Automatic Differentiation. Numerical Algorithms, 2015.
- Autodiff portal: `http://www.autodiff.org/`

Thank you for the attention!



Extension of polynomials to dual numbers

- Let $g(v) = \sum_{i=0}^{n} p_i v^i \in \mathbb{R}[v]$.
- Applying g to $v + \dot{v}\epsilon$

$$\sum_{i=0}^{n} p_i (v + \dot{v}\epsilon)^i = p_0 + p_1(v + \dot{v}\epsilon) + \cdots + \underbrace{p_n(v + \dot{v}\epsilon)^n}_{v^n + nv^{n-1}\dot{v}\epsilon}$$

- Let $g(v) = \sum_{i=0}^{n} p_i v^i \in \mathbb{R}[v]$.
- Applying g to $v + \dot{v}\epsilon$

$$\sum_{i=0}^{n} p_i (v + \dot{v}\epsilon)^i = p_0 + p_1(v + \dot{v}\epsilon) + \cdots + \underbrace{p_n(v + \dot{v}\epsilon)^n}_{v^n + nv^{n-1}\dot{v}\epsilon}$$
$$= g(v) + g'(v)\dot{v}\epsilon = \tilde{g}(v + \dot{v}\epsilon).$$

Extension of analytic functions to dual numbers

- Let $g : \mathbb{R} \to \mathbb{R}$ be analytic.
- Applying g to $v + \dot{v}\epsilon$

$$g(v) + \frac{g'(v)}{1!}\dot{v}\epsilon + \underbrace{\frac{g''(v)}{2!}(\dot{v}\epsilon)^2 + \dots}_{=0}$$

$$= g(v) + g'(v)\dot{v}\epsilon = \tilde{g}(v + \dot{v}\epsilon).$$