ECE59500CV Lecture 3: Automatic Differentiation—I

Jeffrey Mark Siskind

School of Electrical and Computer Engineering

Fall 2020

A Neural Network

x_0 $\frac{\partial x_8}{\partial x_0}$

layer$_0$ θ_0 w_0

x_1 $\frac{\partial x_8}{\partial x_1}$

layer$_1$ θ_1 w_1

x_2 $\frac{\partial x_8}{\partial x_2}$

layer$_2$ θ_2 w_2

x_3 $\frac{\partial x_8}{\partial x_3}$

layer$_3$ θ_3 w_3

x_4 $\frac{\partial x_8}{\partial x_4}$

layer$_4$ θ_4 w_4

x_5 $\frac{\partial x_8}{\partial x_5}$

layer$_5$ θ_5 w_5

x_6 $\frac{\partial x_8}{\partial x_6}$

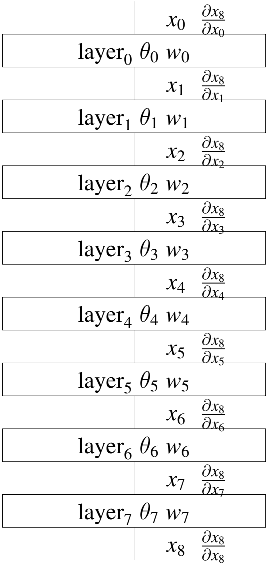
layer$_6$ θ_6 w_6

x_7 $\frac{\partial x_8}{\partial x_7}$

layer$_7$ θ_7 w_7

x_8 $\frac{\partial x_8}{\partial x_8}$

A Neural Network is a (Functional) Program



$$
\begin{array}{l}
\text{net}\ [\theta_0,\ldots,\theta_7]\ [w_0,\ldots,w_7]\ x_0 \triangleq \\
\quad \begin{array}{lrcl}
\textbf{let} & x_1 & = & \text{layer}_0\ \theta_0\ w_0\ x_0 \\
 & x_2 & = & \text{layer}_1\ \theta_1\ w_1\ x_1 \\
 & x_3 & = & \text{layer}_2\ \theta_2\ w_2\ x_2 \\
 & x_4 & = & \text{layer}_3\ \theta_3\ w_3\ x_3 \\
 & x_5 & = & \text{layer}_4\ \theta_4\ w_4\ x_4 \\
 & x_6 & = & \text{layer}_5\ \theta_5\ w_5\ x_5 \\
 & x_7 & = & \text{layer}_6\ \theta_6\ w_6\ x_6 \\
 & x_8 & = & \text{layer}_7\ \theta_7\ w_7\ x_7 \\
\end{array} \\
\quad \textbf{in}\ x_8
\end{array}
$$

A (Functional) Program

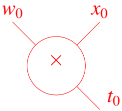
$$
\begin{array}{l}
f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
\quad \textbf{let}\ \begin{array}{rcl} t_0 & = & w_0 \times x_0 \\ t_1 & = & w_1 \times x_1 \\ y & = & t_0 + t_1 \end{array} \\
\quad \textbf{in}\ y
\end{array}
$$

A (Functional) Program is a (Neural) Network

$$
\begin{array}{l}
f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
\quad \textbf{let}\ \begin{array}{rcl} t_0 & = & w_0 \times x_0 \\ t_1 & = & w_1 \times x_1 \\ y & = & t_0 + t_1 \end{array} \\
\quad \textbf{in}\ y
\end{array}
$$

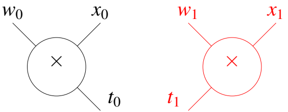
A (Functional) Program is a (Neural) Network

$$
\begin{aligned}
& f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
& \quad \textbf{let}\ \ t_0 = w_0 \times x_0 \\
& \quad\quad\quad\ \ t_1 = w_1 \times x_1 \\
& \quad\quad\quad\ \ y = t_0 + t_1 \\
& \quad \textbf{in}\ y
\end{aligned}
$$



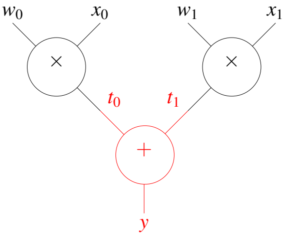
A (Functional) Program is a (Neural) Network

$$
\begin{aligned}
&f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
&\quad \textbf{let}\ \ t_0 = w_0 \times x_0 \\
&\quad\quad\quad t_1 = w_1 \times x_1 \\
&\quad\quad\quad y = t_0 + t_1 \\
&\quad \textbf{in}\ y
\end{aligned}
$$



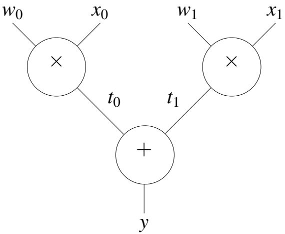
A (Functional) Program is a (Neural) Network

$$f\ [w_0, w_1]\ [x_0, x_1] \triangleq$$
$$\textbf{let}\ t_0 = w_0 \times x_0$$
$$t_1 = w_1 \times x_1$$
$$y = t_0 + t_1$$
$$\textbf{in}\ y$$

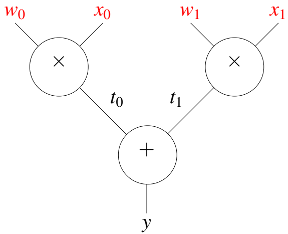


A (Functional) Program is a (Neural) Network

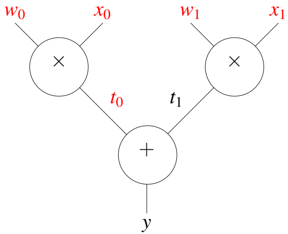
$$
\begin{aligned}
& f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
& \quad \textbf{let}\ t_0 = w_0 \times x_0 \\
& \quad\qquad t_1 = w_1 \times x_1 \\
& \quad\qquad y = t_0 + t_1 \\
& \quad \textbf{in}\ y
\end{aligned}
$$



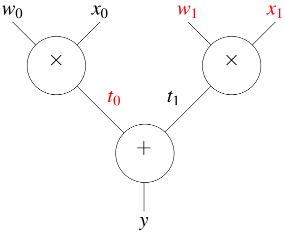
Evaluating a Network



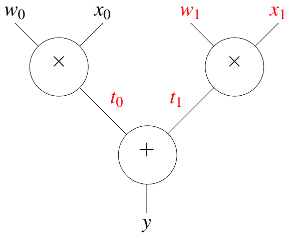
Evaluating a Network



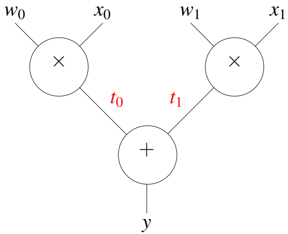
Evaluating a Network



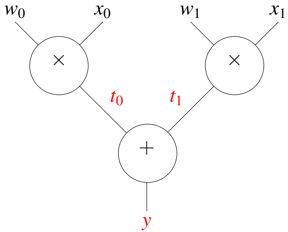
Evaluating a Network



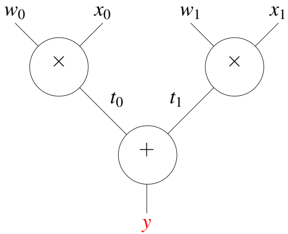
Evaluating a Network



Evaluating a Network



Evaluating a Network



The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε,

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\,f\;c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε,

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by 1!

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by 1! (noop).

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\; c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by 1! (noop).

Key idea: Only need output to be a **finite** truncated power series $a + b\varepsilon$.

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\; c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by 1! (noop).

Key idea: Only need output to be a **finite** truncated power series $a + b\varepsilon$.

The input $c+\varepsilon$ is also a truncated power series.

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\,f\;c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by 1! (noop).

Key idea: Only need output to be a **finite** truncated power series $a+b\varepsilon$.

The input $c+\varepsilon$ is also a truncated power series.

Can do a *nonstandard interpretation* of f over truncated power series.

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by 1! (noop).

Key idea: Only need output to be a **finite** truncated power series $a + b\varepsilon$.

The input $c+\varepsilon$ is also a truncated power series.

Can do a *nonstandard interpretation* of f over truncated power series.

Preserves control flow: Augments original values with derivatives.

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by $1!$ (noop).

Key idea: Only need output to be a **finite** truncated power series $a + b\varepsilon$.

The input $c+\varepsilon$ is also a truncated power series.

Can do a *nonstandard interpretation* of f over truncated power series.

Preserves control flow: Augments original values with derivatives.

$(\mathcal{D}\, f)$ is $\mathcal{O}(1)$ relative to f (both space and time).

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by $1!$ (noop).

Key idea: Only need output to be a **finite** truncated power series $a+b\varepsilon$.

The input $c+\varepsilon$ is also a truncated power series.

Can do a *nonstandard interpretation* of f over truncated power series.

Preserves control flow: Augments original values with derivatives.

$(\mathcal{D}\, f)$ is $\mathcal{O}(1)$ relative to f (both space and time).

These $a+b\varepsilon$ are called *dual numbers* and can be represented as $\langle a, b\rangle$.

The Essence of Forward-Mode AD

Taylor expansion:

$$f(c+\varepsilon) = \frac{f(c)}{0!} + \frac{f'(c)}{1!}\varepsilon + \frac{f''(c)}{2!}\varepsilon^2 + \cdots + \frac{f^{(i)}(c)}{i!}\varepsilon^i + \cdots$$

To compute $\mathcal{D}\, f\ c$:

- evaluate f at the **term** $c+\varepsilon$ to get a **power series**,
- extract the coefficient of ε, and
- multiply by $1!$ (noop).

Key idea: Only need output to be a **finite** truncated power series $a + b\varepsilon$.

The input $c+\varepsilon$ is also a truncated power series.

Can do a *nonstandard interpretation* of f over truncated power series.

Preserves control flow: Augments original values with derivatives.

$(\mathcal{D}\, f)$ is $\mathcal{O}(1)$ relative to f (both space and time).

These $a + b\varepsilon$ are called *dual numbers* and can be represented as $\langle a, b\rangle$.

(Analogous to complex numbers $a + b$i represented as $\langle a, b\rangle$.)

Complex Numbers

$\mathrm{i}^2 = -1$

$$(a + b\mathrm{i}) + (c + d\mathrm{i}) = (a + c) + (b + d)\mathrm{i}$$
$$(a + b\mathrm{i})(c + d\mathrm{i}) = ac + (ad + bc)\mathrm{i} + bd\mathrm{i}^2 = (ac - bd) + (ad + bc)\mathrm{i}$$

Dual Numbers

$\varepsilon^2 = 0$, but $\varepsilon \neq 0$

$$(a + b\varepsilon) + (c + d\varepsilon) = (a + c) + (b + d)\varepsilon$$
$$(a + b\varepsilon)(c + d\varepsilon) = ac + (ad + bc)\varepsilon + bd\varepsilon^2 = ac + (ad + bc)\varepsilon$$

Arithmetic on Truncated Power Series (i.e. Dual Numbers)

$$(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) + (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2)) = (x_0 + y_0) + (x_1 + y_1)\varepsilon + \mathcal{O}(\varepsilon^2)$$

Arithmetic on Truncated Power Series (i.e. Dual Numbers)

$$(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) + (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2)) = (x_0 + y_0) + (x_1 + y_1)\varepsilon + \mathcal{O}(\varepsilon^2)$$

$$\begin{aligned}(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) \times (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2))\\ = (x_0 \times y_0) + (x_0 \times y_1 + x_1 \times y_0)\varepsilon + \mathcal{O}(\varepsilon^2)\end{aligned}$$

Arithmetic on Truncated Power Series (i.e. Dual Numbers)

$$(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) + (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2)) = (x_0 + y_0) + (x_1 + y_1)\varepsilon + \mathcal{O}(\varepsilon^2)$$

$$\begin{aligned}(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) &\times (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2)) \\ &= (x_0 \times y_0) + (x_0 \times y_1 + x_1 \times y_0)\varepsilon + \mathcal{O}(\varepsilon^2)\end{aligned}$$

$$u\ (x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) = (u\ x_0) + (x_1 \times (u'\ x_0))\varepsilon + \mathcal{O}(\varepsilon^2)$$

Arithmetic on Truncated Power Series (i.e. Dual Numbers)

$$(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) + (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2)) = (x_0 + y_0) + (x_1 + y_1)\varepsilon + \mathcal{O}(\varepsilon^2)$$

$$(x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) \times (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2))$$
$$= (x_0 \times y_0) + (x_0 \times y_1 + x_1 \times y_0)\varepsilon + \mathcal{O}(\varepsilon^2)$$

$$u\ (x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)) = (u\ x_0) + (x_1 \times (u'\ x_0))\varepsilon + \mathcal{O}(\varepsilon^2)$$

$$b\ ((x_0 + x_1\varepsilon + \mathcal{O}(\varepsilon^2)), (y_0 + y_1\varepsilon + \mathcal{O}(\varepsilon^2)))$$
$$= (b\ (x_0, y_0)) + (x_1 \times (b^{(1,0)}\ (x_0, y_0)) + y_1 \times (b^{(0,1)}\ (x_0, y_0)))\varepsilon + \mathcal{O}(\varepsilon^2)$$