Computing Derivatives of Matrix and Tensor Expressions

Sören Laue

Friedrich-Schiller-University Jena, Germany

April 9th, 2021

seit 1558

Matrix calculus

Example:

- $f(x) = x^\top A x$

.

Matrix calculus

Example:

- $f(x) = x^\top A x$
- $\frac{df}{dx} = ?$

.

seit 1558

Matrix calculus

Example:

- $f(x) = x^\top A x$
- $\frac{df}{dx} = ?$
- $\frac{d^2f}{dx^2} = ?$

.

Matrix calculus

Example:

- $f(x) = x^\top A x$
- $\frac{df}{dx} = Ax + A^\top x$
- $\frac{d^2 f}{dx^2} = ?$

.

Matrix calculus

Example:

- $f(x) = x^\top A x$
- $\frac{df}{dx} = Ax + A^\top x$
- $\frac{d^2 f}{dx^2} = A + A^\top$

.

Matrix calculus

Example:

- $f(x) = x^\top Ax$
- $\frac{df}{dx} = Ax + A^\top x$
- $\frac{d^2f}{dx^2} = A + A^\top$

No general and coherent theory known!

Matrix calculus?

- wikipedia

Matrix calculus?

- wikipedia
- matrix cookbook

Matrix calculus?

- wikipedia
- matrix cookbook
- Matrix Differential Calculus with Applications in Statistics (Magnus and Neudecker)

seit 1558

Matrix calculus?

- wikipedia
- matrix cookbook
- Matrix Differential Calculus with Applications in Statistics (Magnus and Neudecker)

Contain only collection of recipes / lookup tables.

Matrix calculus – software?

- Mathematica

Matrix calculus – software?

- Mathematica
- Maple

Matrix calculus – software?

- Mathematica
- Maple
- TensorFlow, PyTorch (non-scalar output)

Matrix calculus – software?

- Mathematica
- Maple
- TensorFlow, PyTorch (non-scalar output)
- ...

Matrix calculus – software?

- Mathematica
- Maple
- TensorFlow, PyTorch (non-scalar output)
- ...

Cannot perform matrix calculus.

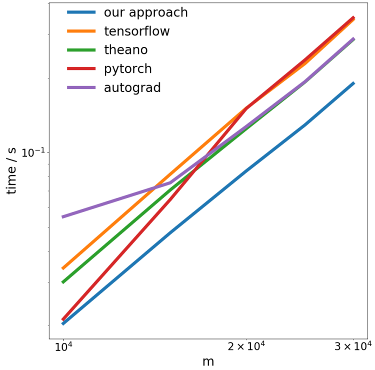
Matrix calculus

MatrixCalculus.org

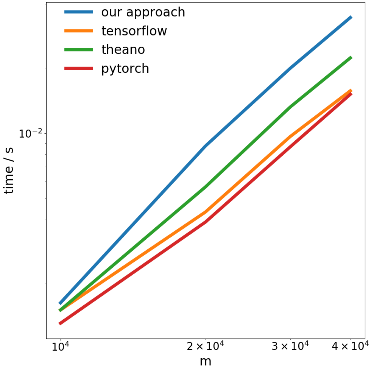
seit 1558

Running times

$$\text{gradient of } f(x) = x^\top A x$$



CPU

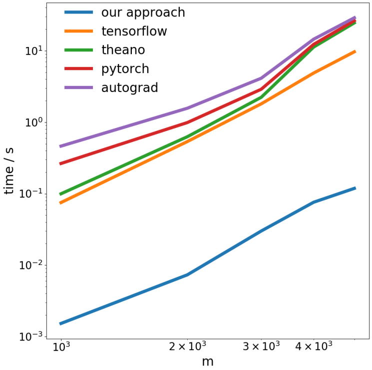


GPU

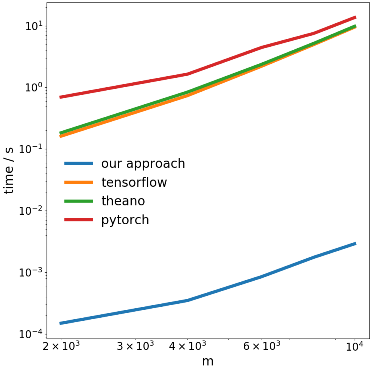
seit 1558

Running times

Hessian of $f(x) = x^\top A x$



CPU $\sim$ 100x

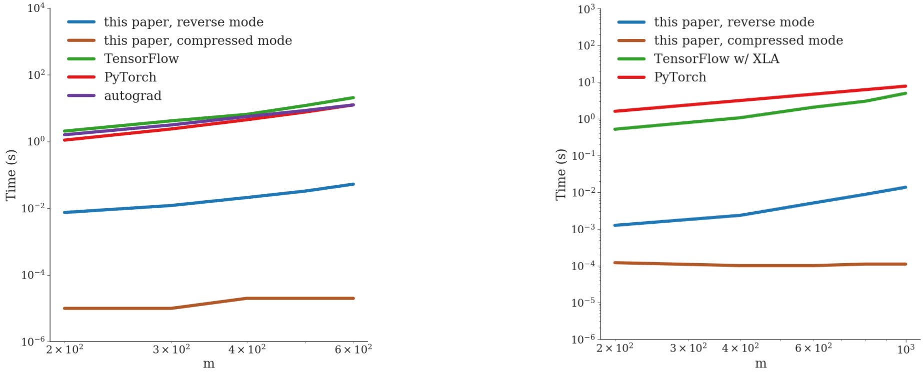


GPU $\sim$ 1000x

seit 1558

Running times

$$\text{Hessian of } f(U) = \|T - UV^\top\|_2^2$$

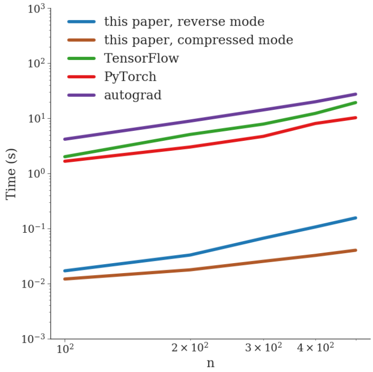


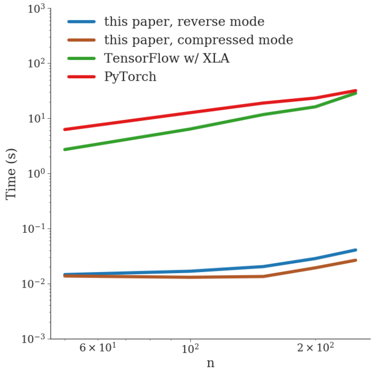
CPU $\sim$ 100x

GPU $\sim$ 1000x

Running times

Hessian of neural net (10 dense layers w/ ReLU, softmax cross-entropy)





CPU $\sim$ 100x

GPU $\sim$ 1000x

seit 1558

Algorithmic Details

Derivatives

Symbolic Differentiation vs. Automatic Differentiation

Derivatives

$$f(a) = \log(\sin(a^2))$$

Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} =$$

seit 1558

Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)}$$

Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2)$$

Derivatives

$$f(a) = \log(\sin(a^2))$$

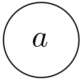
$$\frac{df}{da} = \quad \frac{1}{\sin(a^2)} \quad \cdot \; \cos(a^2) \quad \cdot \quad 2a$$

Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$

evaluation

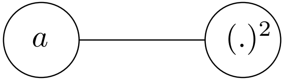


Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$

evaluation

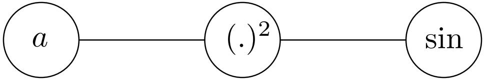


Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$

evaluation

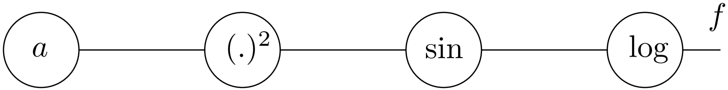


Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$

evaluation

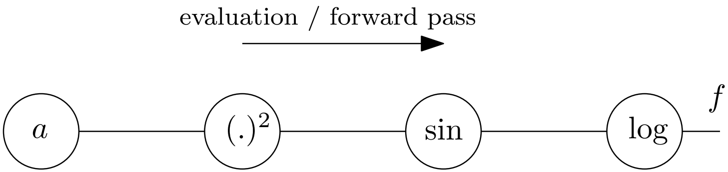


seit 1558

Derivatives

$$f(a) = \log(\sin(a^2))$$

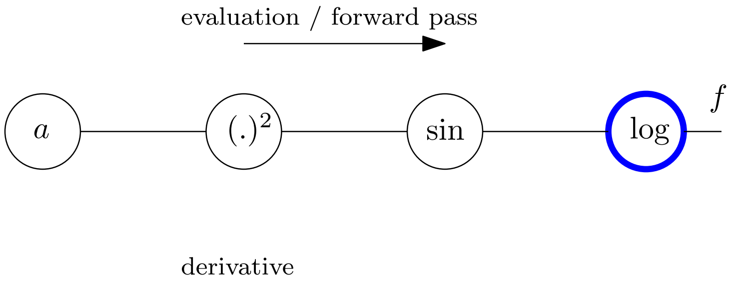
$$\frac{df}{da} = \quad \frac{1}{\sin(a^2)} \quad \cdot \ \cos(a^2) \quad \cdot \quad 2a$$



Derivatives

$$f(a) = \log(\sin(a^2))$$

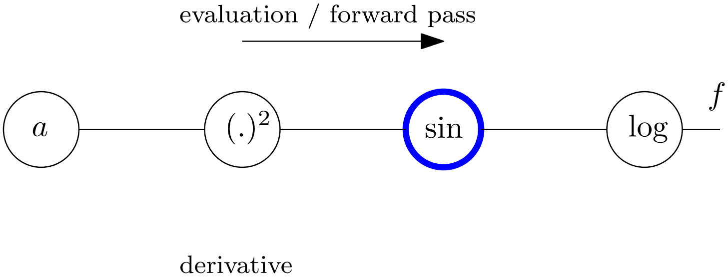
$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$



Derivatives

$$f(a) = \log(\sin(a^2))$$

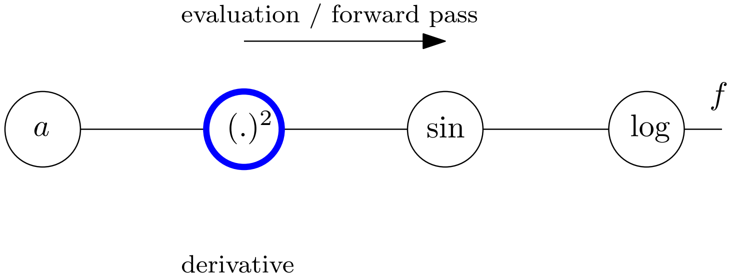
$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$



Derivatives

$$f(a) = \log(\sin(a^2))$$

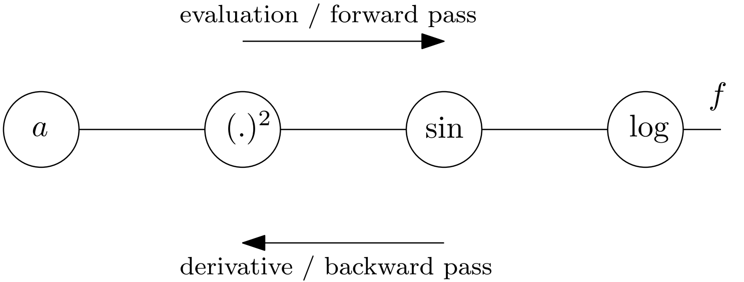
$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$



Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \quad \frac{1}{\sin(a^2)} \quad \cdot \; \cos(a^2) \quad \cdot \quad 2a$$

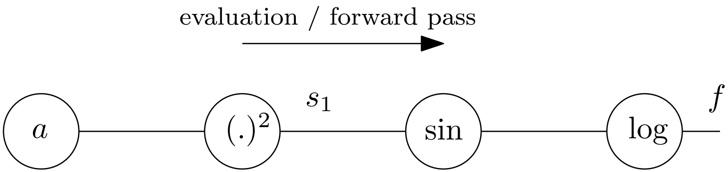


Derivatives

$$f(a) = \log(\sin(a^2))$$

$$s_1 = a^2$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$



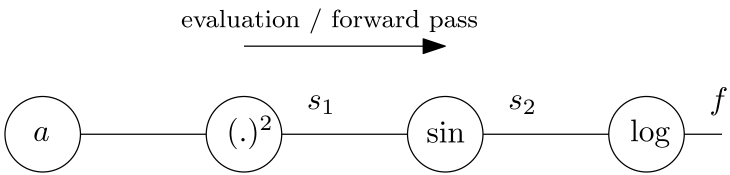
Derivatives

$$f(a) = \log(\sin(a^2))$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a$$

$$s_1 = a^2$$

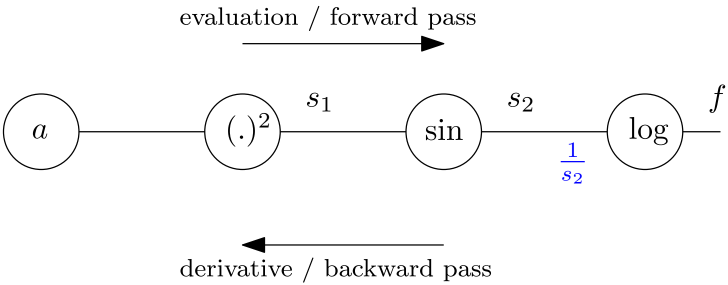
$$s_2 = \sin(a^2)$$



Derivatives

$$f(a) = \log(\sin(a^2)) \qquad s_1 = a^2$$

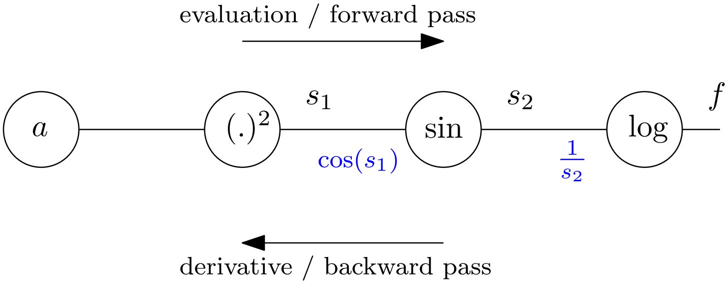
$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a \qquad s_2 = \sin(a^2)$$



Derivatives

$$f(a) = \log(\sin(a^2)) \qquad s_1 = a^2$$

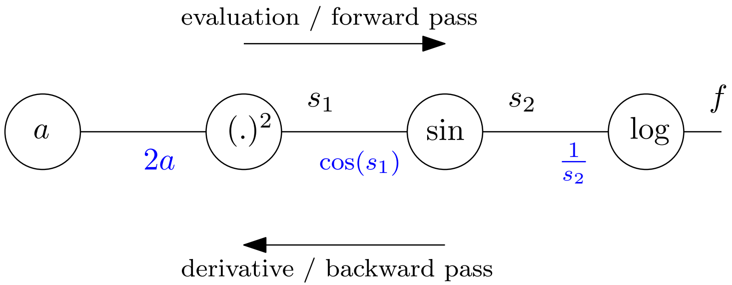
$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a \qquad s_2 = \sin(a^2)$$



Derivatives

$$f(a) = \log(\sin(a^2)) \qquad s_1 = a^2$$

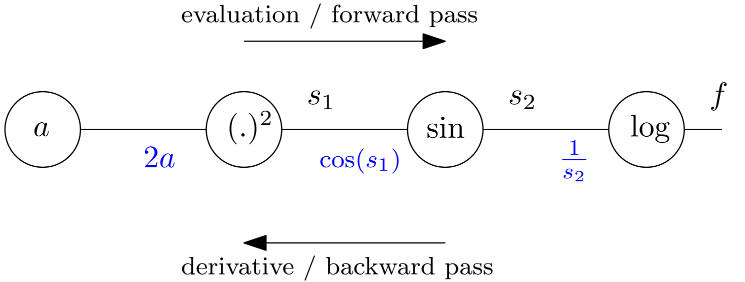
$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a \qquad s_2 = \sin(a^2)$$



Derivatives

$$f(a) = \log(\sin(a^2)) \qquad s_1 = a^2$$

$$\frac{df}{da} = \frac{1}{\sin(a^2)} \cdot \cos(a^2) \cdot 2a \qquad s_2 = \sin(a^2)$$



backpropagation / reverse mode autodiff

Derivatives

$$f(a, b) = \log(\sin(a \cdot b))$$

Derivatives

$$f(a, b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} =$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \quad \frac{1}{\sin(a \cdot b)}$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \quad \frac{1}{\sin(a \cdot b)} \quad \cdot \cos(a \cdot b)$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \quad \frac{1}{\sin(a \cdot b)} \quad \cdot \cos(a \cdot b) \quad \cdot \quad b$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \quad \frac{1}{\sin(a \cdot b)} \quad \cdot \cos(a \cdot b) \quad \cdot \quad b$$

$$\frac{df}{db} =$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)}$$

Derivatives

$$f(a, b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b)$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

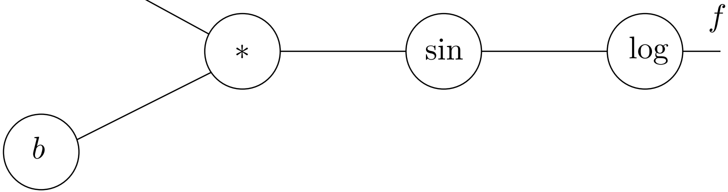
$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

Derivatives

$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \quad \frac{1}{\sin(a \cdot b)} \quad \cdot \cos(a \cdot b) \quad \cdot \quad b$$

$$\frac{df}{db} = \quad \frac{1}{\sin(a \cdot b)} \quad \cdot \cos(a \cdot b) \quad \cdot \quad a$$



Derivatives

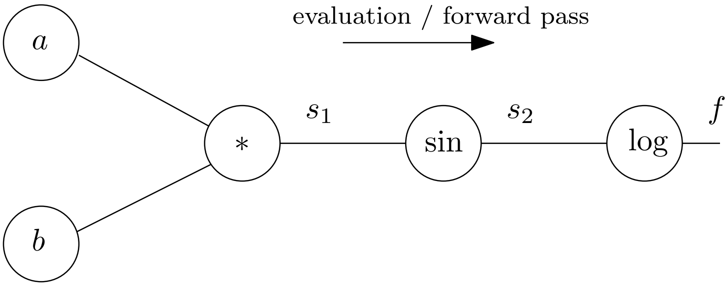
$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

$$s_1 = a \cdot b$$

$$s_2 = \sin(a \cdot b)$$



Derivatives

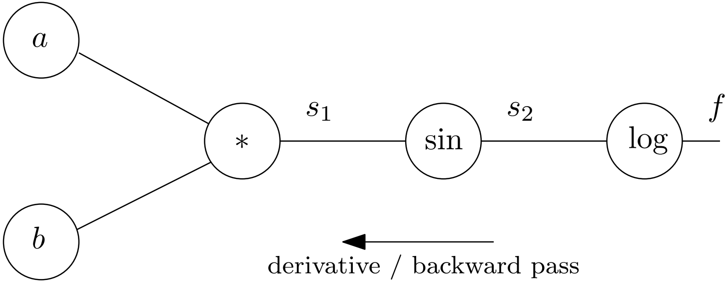
$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

$$s_1 = a \cdot b$$

$$s_2 = \sin(a \cdot b)$$



seit 1558

Derivatives

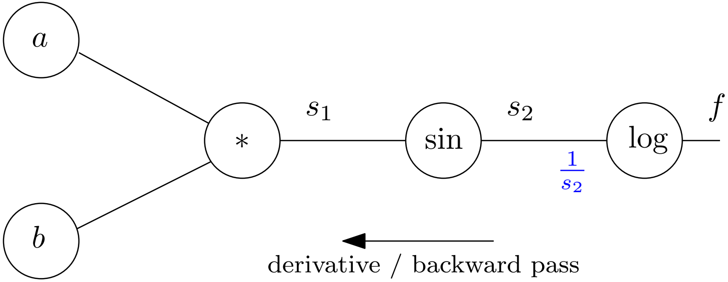
$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

$$s_1 = a \cdot b$$

$$s_2 = \sin(a \cdot b)$$



seit 1558

Derivatives

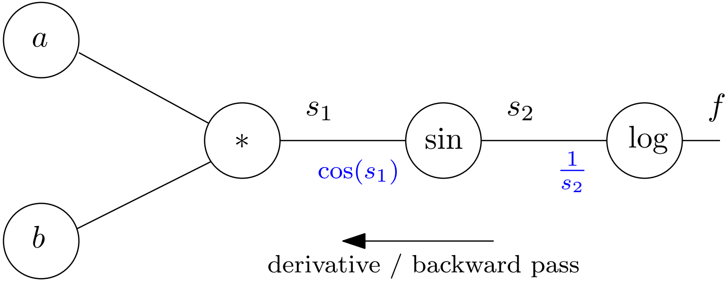
$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

$$s_1 = a \cdot b$$

$$s_2 = \sin(a \cdot b)$$



seit 1558

Derivatives

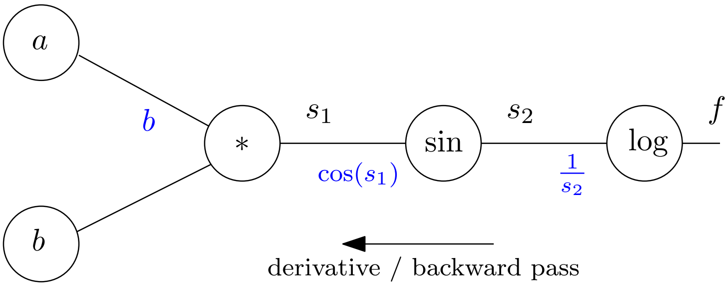
$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

$$s_1 = a \cdot b$$

$$s_2 = \sin(a \cdot b)$$



Derivatives

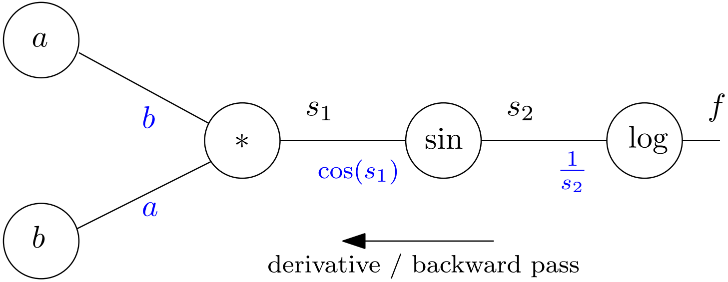
$$f(a,b) = \log(\sin(a \cdot b))$$

$$\frac{df}{da} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot b$$

$$\frac{df}{db} = \frac{1}{\sin(a \cdot b)} \cdot \cos(a \cdot b) \cdot a$$

$$s_1 = a \cdot b$$

$$s_2 = \sin(a \cdot b)$$



Derivatives

Symbolic Differentiation and Automatic Differentiation are basically the same

Derivatives

Symbolic Differentiation and Automatic Differentiation are basically the same

when allowing common subexpressions in symbolic differentiation.

Derivatives

Symbolic Differentiation and Automatic Differentiation are basically the same

when allowing common subexpressions in symbolic differentiation.

Warning: my personal view!

Derivatives

Symbolic Differentiation and Automatic Differentiation are basically the same

when allowing common subexpressions in symbolic differentiation.

Warning: my personal view!

common claim: symbolic differentiation suffers from expression swell

Matrix Calculus

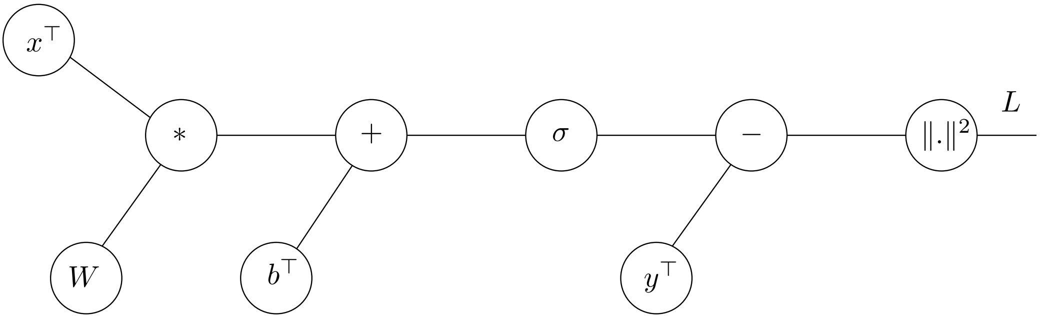
Matrix Case

Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0, 1\}^d$$

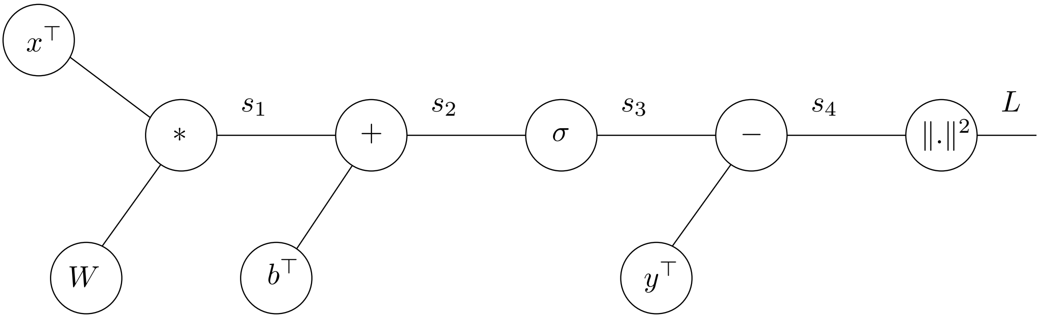
Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$



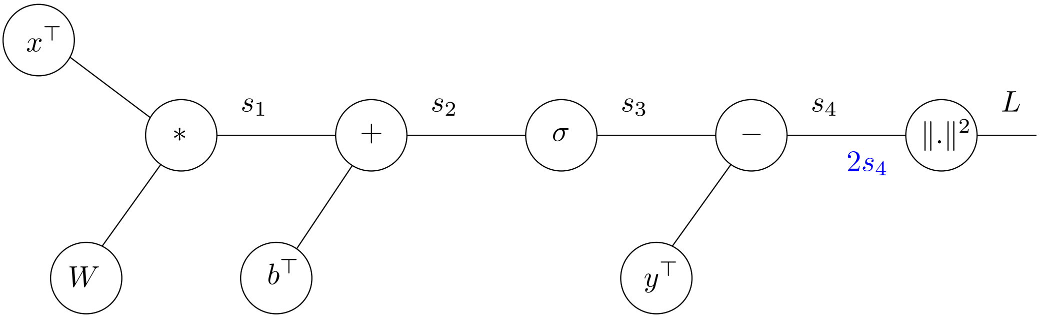
Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0, 1\}^d$$



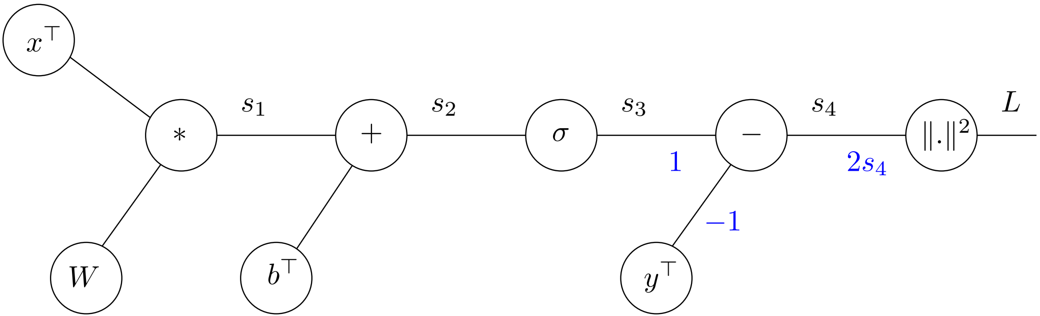
Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$



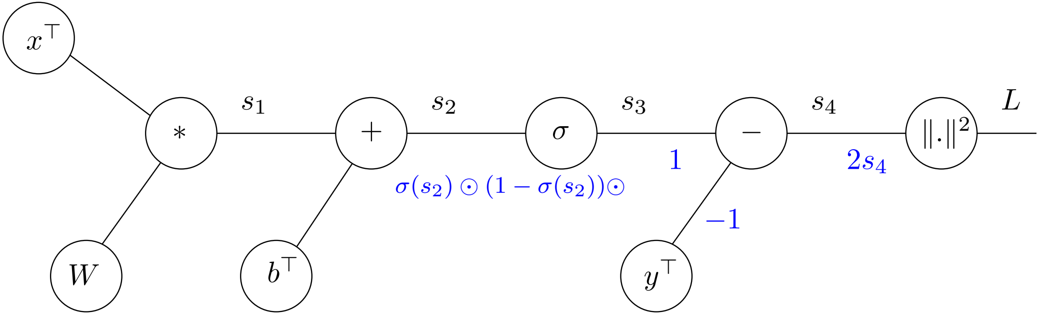
Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$



Neural Net, 1 Layer

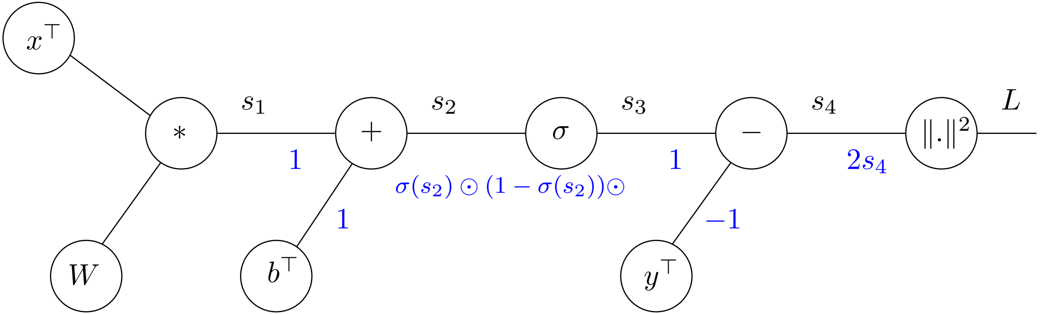
$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$



seit 1558

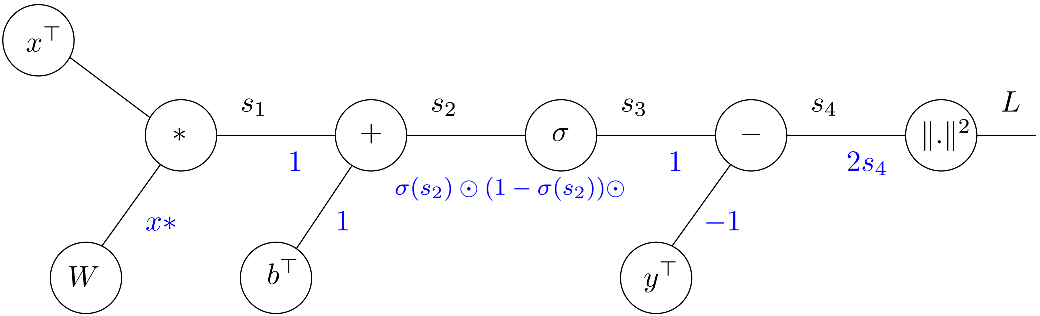
Neural Net, 1 Layer

$$L(x, W, b, y) = \left\| \sigma(x^\top W + b^\top) - y^\top \right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$



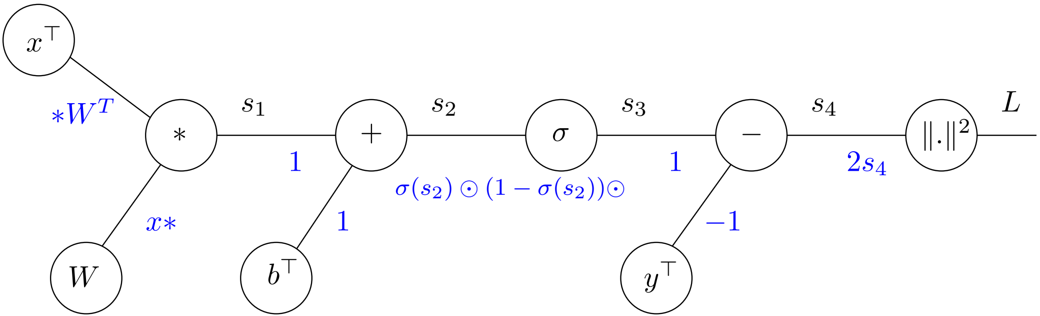
Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$



Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$

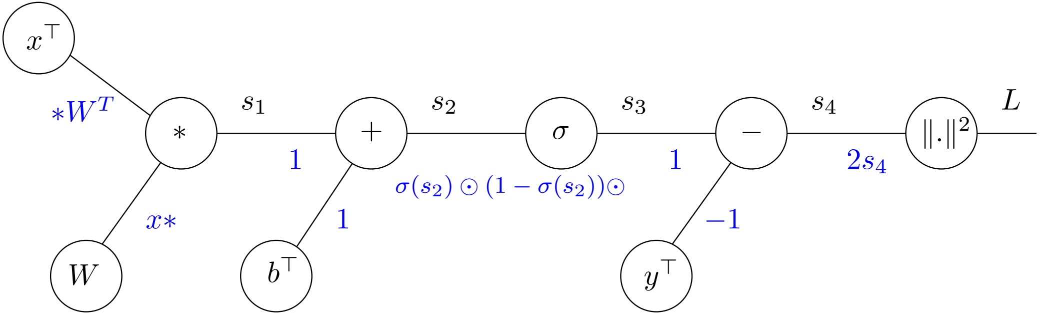


Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$

$$\frac{dL}{dW} =$$

$$\frac{dL}{db} =$$

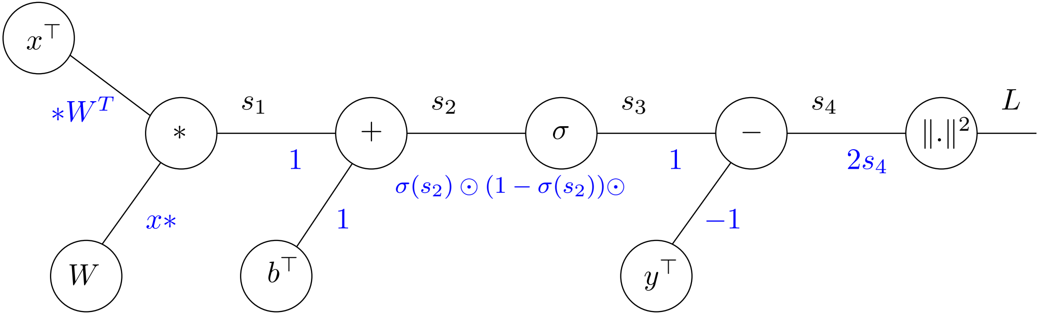


Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$

$$\frac{dL}{dW} = \; x \, * \, (\sigma(s_2) \odot (1 - \sigma(s_2)) \; \odot \; 2s_4)$$

$$\frac{dL}{db} =$$

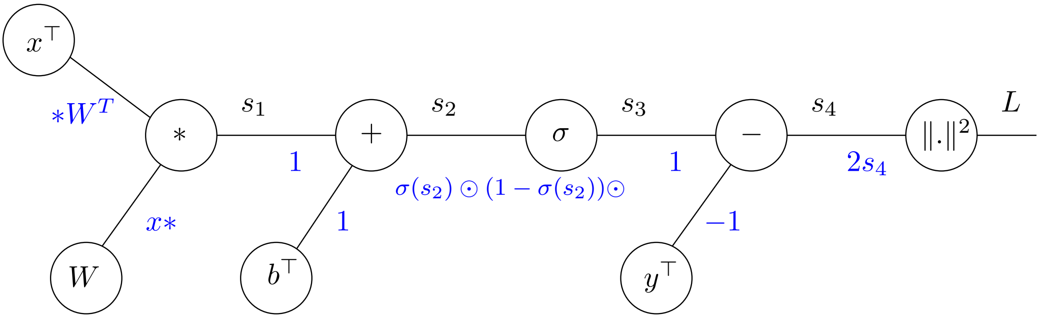


Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$

$$\frac{dL}{dW} = x * (\sigma(s_2) \odot (1 - \sigma(s_2)) \odot 2s_4)$$

$$\frac{dL}{db} = \sigma(s_2) \odot (1 - \sigma(s_2)) \odot 2s_4$$

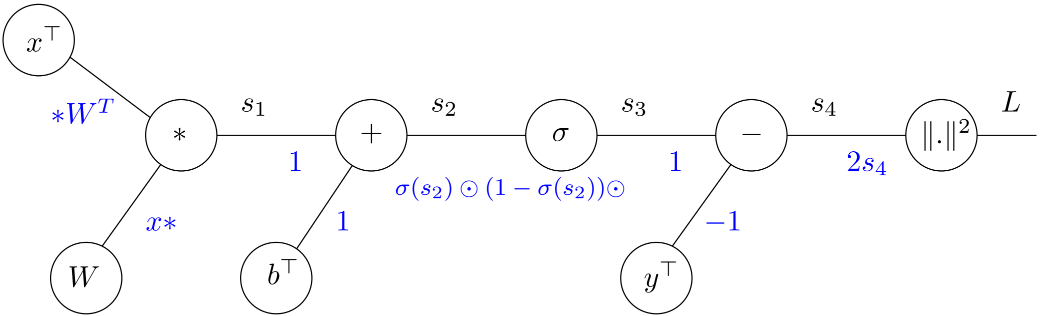


Neural Net, 1 Layer

$$L(x, W, b, y) = \left\|\sigma(x^\top W + b^\top) - y^\top\right\|^2 \qquad x \in \mathbb{R}^n, y \in \{0,1\}^d$$

$$\frac{dL}{dW} = x * (\sigma(s_2) \odot (1 - \sigma(s_2)) \odot 2s_4) \qquad s_2 = x^\top W + b^\top$$

$$\frac{dL}{db} = \sigma(s_2) \odot (1 - \sigma(s_2)) \odot 2s_4 \qquad s_4 = \sigma(x^\top W + b^\top) - y^\top$$



Matrix Calculus

Matrix case is identical to scalar case, except for the type of multiplications in the derivative.

seit 1558

Matrix Calculus

Matrix case is identical to scalar case, except for the type of multiplications in the derivative.

This is the root of all the trouble with matrix calculus.

Matrix Calculus – Matrix Notation

First attempt: Use matrix notation for matrix calculus.

Matrix Calculus – Matrix Notation

First attempt: Use matrix notation for matrix calculus.

- 24 types of matrix multiplication needed, only for the linear matrix case

Matrix Calculus – Matrix Notation

First attempt: Use matrix notation for matrix calculus.

- 24 types of matrix multiplication needed, only for the linear matrix case
- ended up in a mess

Matrix Calculus – Matrix Notation

First attempt: Use matrix notation for matrix calculus.

- 24 types of matrix multiplication needed, only for the linear matrix case
- ended up in a mess
- led to buggy implementation in SymPy

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- need for higher order tensors

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- need for higher order tensors
- Ricci notation precise, e.g., T^i_{jk}

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- need for higher order tensors
- Ricci notation precise, e.g., T^i_{jk}
- $f(x) = x^\top, \quad \nabla f(x) = ?$

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- need for higher order tensors
- Ricci notation precise, e.g., T^i_{jk}
- $f(x) = x^\top, \quad \nabla f(x) = ?$
- $\nabla f(x) = \delta_{ij}$, not the identity matrix

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- need for higher order tensors
- Ricci notation precise, e.g., T^i_{jk}
- $f(x) = x^\top, \quad \nabla f(x) = ?$
- $\nabla f(x) = \delta_{ij}, \quad$ not the identity matrix
- first usable algorithm for matrix calculus

Details – Elements of Ricci calculus

matrix notation	Ricci notation
a	a
x	x^i
$x^\top$	x_i
A	A^i_j
$\mathbb{I}$	δ^i_j

seit 1558

Details – Elements of Ricci calculus

matrix notation	Ricci notation
a	a
x	x^i
$x^\top$	x_i
A	A^i_j
$\mathbb{I}$	δ^i_j

$$A_{ij} \quad B^i_{jk} \quad C^{il}_{jk}$$

seit 1558

Details – Elements of Ricci calculus

matrix notation	Ricci notation
Ax	$A^i_j \, x^j$
$y^\top x$	$y_j \, x^j$
AB	$A^i_j \, B^j_k$
$yx^\top$	$y^i \, x_j$
$y \odot x$	$y^i \, x^i$
$A \odot B$	$A^i_j \, B^i_j$
$A \cdot \mathrm{diag}(x)$	$A^i_j \, x_j$

Details – Elements of Ricci calculus

matrix notation	Ricci notation
Ax	$A^i_j\, x^j$
$y^\top x$	$y_j\, x^j$
AB	$A^i_j\, B^j_k$
$yx^\top$	$y^i\, x_j$
$y \odot x$	$y^i\, x^i$
$A \odot B$	$A^i_j\, B^i_j$
$A \cdot \mathrm{diag}(x)$	$A^i_j\, x_j$

Ricci notation is commutative.

Details

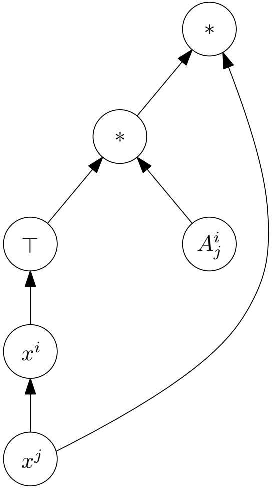
gradient of $f(x) = x^\top A x$

Details

gradient of $f(x) = x^\top A x = x_i \, A^i_j \, x^j$

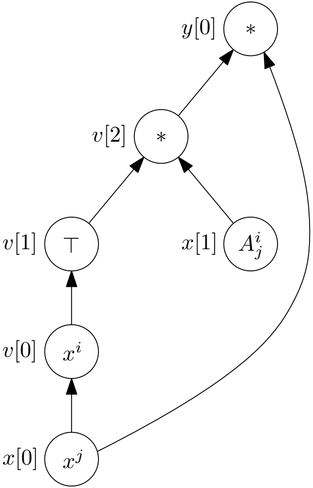
Details

gradient of $f(x) = x^\top A x = x_i\, A^i_j\, x^j$



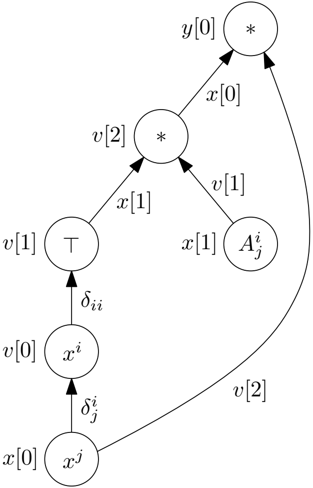
Details

gradient of $f(x) = x^\top A x = x_i\, A^i_j\, x^j$



Details

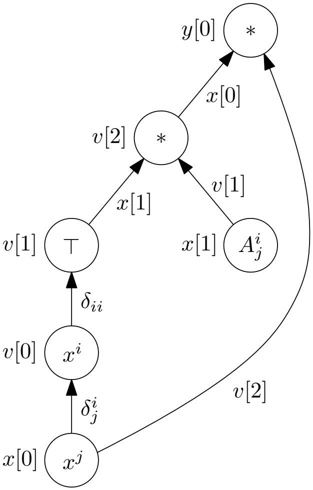
gradient of $f(x) = x^\top A x = x_i\, A^i_j\, x^j$



seit 1558

Details

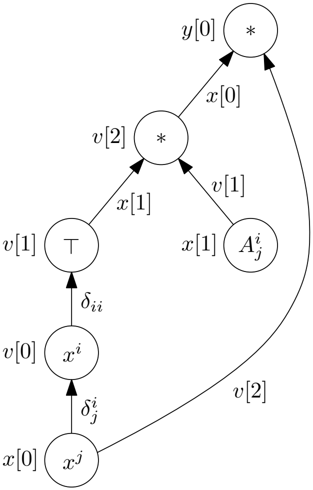
gradient of $f(x) = x^\top A x = x_i \, A^i_j \, x^j$



reverse mode:

Details

gradient of $f(x) = x^\top A x = x_i \, A^i_j \, x^j$

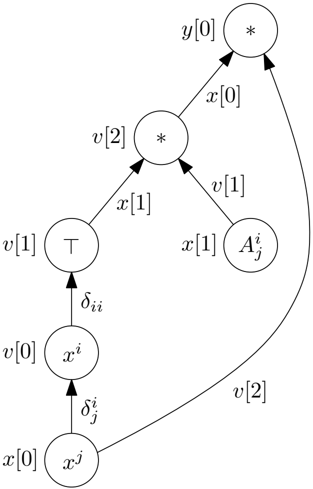


reverse mode:

$$\nabla f = ((x[0] \cdot x[1]) \cdot \delta_{ii}) \, \cdot \delta^i_j + v[2]$$

Details

gradient of $f(x) = x^\top A x = x_i\, A^i_j\, x^j$

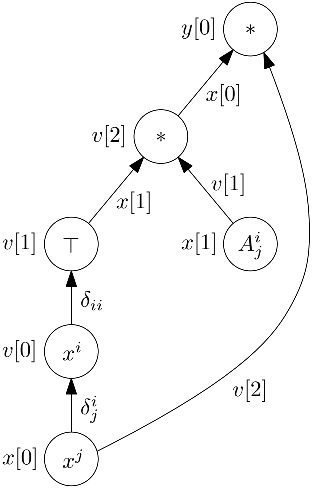


reverse mode:

$$
\begin{aligned}
\nabla f &= ((x[0] \cdot x[1]) \cdot \delta_{ii}) \cdot \delta^i_j + v[2] \\
&= \left(\left(x^j \cdot A^i_j\right) \cdot \delta_{ii}\right) \cdot \delta^i_j + x_i\, A^i_j
\end{aligned}
$$

Details

gradient of $f(x) = x^\top A x = x_i\, A^i_j\, x^j$

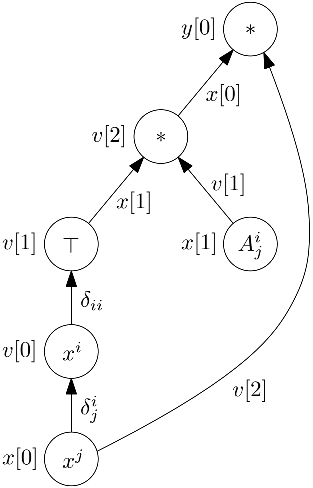


reverse mode:

$$
\begin{aligned}
\nabla f &= ((x[0]\cdot x[1])\cdot \delta_{ii})\ \cdot \delta^i_j + v[2] \\
&= \left(\left(x^j \quad \cdot A^i_j\right) \quad \cdot \delta_{ii}\right)\cdot \delta^i_j + x_i\, A^i_j \\
&= \left(\left(A^i_j \quad \cdot x^j\right) \quad \cdot \delta_{ii}\right)\cdot \delta^i_j + x_i\, A^i_j
\end{aligned}
$$

Details

gradient of $f(x) = x^\top A x = x_i\, A^i_j\, x^j$



reverse mode:

$$
\begin{aligned}
\nabla f &= ((x[0] \cdot x[1]) \cdot \delta_{ii}) \cdot \delta^i_j + v[2] \\
&= \Big(\big(x^j \cdot A^i_j\big) \cdot \delta_{ii}\Big) \cdot \delta^i_j + x_i\, A^i_j \\
&= \Big(\big(A^i_j \cdot x^j\big) \cdot \delta_{ii}\Big) \cdot \delta^i_j + x_i\, A^i_j \\
&= (Ax)^\top + x^\top A
\end{aligned}
$$

seit 1558

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- Ricci notation precise

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- Ricci notation precise
- differentiates between upper and lower indices, i.e., between covariance and contravariance of a vector

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- Ricci notation precise
- differentiates between upper and lower indices, i.e., between covariance and contravariance of a vector
- $x = x^i \quad x^\top = x_i$

Matrix Calculus – Ricci Notation

Use Ricci notation for matrix calculus.

- Ricci notation precise
- differentiates between upper and lower indices, i.e., between covariance and contravariance of a vector
- $x = x^i \quad x^\top = x_i$
- often, this precision / distinction is not needed

Matrix Calculus – Einstein Notation

Use generalized Einstein notation for matrix calculus.

Matrix Calculus – Einstein Notation

Use generalized Einstein notation for matrix calculus.

- does not distinguish between upper and lower indices

Matrix Calculus – Einstein Notation

Use generalized Einstein notation for matrix calculus.

- does not distinguish between upper and lower indices
- $T = T[i, j, \ldots]$

Matrix Calculus – Einstein Notation

Use generalized Einstein notation for matrix calculus.

- does not distinguish between upper and lower indices
- $T = T[i, j, \ldots]$
- allows for compression of derivatives

Matrix calculus – Einstein notation

Let A, B and C be tensors. Any tensor/matrix multiplication can be written as:

$$C[s_3] = \sum_{(s_1 \cap s_2) \setminus s_3} A[s_1] \cdot B[s_2],$$

where s_1, s_2 and s_3 are index sets.

Matrix calculus – Einstein notation

Let A, B and C be tensors. Any tensor/matrix multiplication can be written as:

$$C[s_3] = \sum_{(s_1 \cap s_2) \setminus s_3} A[s_1] \cdot B[s_2],$$

where s_1, s_2 and s_3 are index sets.

multiplication symbol

$$C = A *_{(s_1, s_2, s_3)} B$$

Matrix calculus – Einstein notation

Let A, B and C be tensors. Any tensor/matrix multiplication can be written as:

$$C[s_3] = \sum_{(s_1 \cap s_2) \setminus s_3} A[s_1] \cdot B[s_2],$$

where s_1, s_2 and s_3 are index sets.

multiplication symbol

$$C = A *_{(s_1, s_2, s_3)} B$$

`einsum` in NumPy

seit 1558

Matrix calculus – Einstein Notation

$$C = A *_{(s_1,s_2,s_3)} B$$

Matrix calculus – Einstein Notation

$$C = A *_{(s_1, s_2, s_3)} B$$

forward mode autodiff:

$$\dot{C} = A *_{(s_1, s_2 s_4, s_3 s_4)} \dot{B}$$

where s_4 is the new index set of $\frac{dx}{dx}$.

Matrix calculus – Einstein Notation

$$C = A *_{(s_1, s_2, s_3)} B$$

forward mode autodiff:

$$\dot{C} = A *_{(s_1, s_2 s_4, s_3 s_4)} \dot{B}$$

where s_4 is the new index set of $\frac{dx}{dx}$.

reverse mode autodiff:

$$\bar{B} = A *_{(s_1, s_5 s_3, s_5 s_2)} \bar{C}$$

where s_5 is the new index set of $\frac{df}{df}$ (f - output function).

seit 1558

Matrix calculus – Einstein Notation

$$C[s3] = g(A[s1])$$

Matrix calculus – Einstein Notation

$$C[s3] = g(A[s1])$$

forward mode autodiff:

$$\dot{C} = g'(A) *_{(s_1, s_1 s_4, s_3 s_4)} \dot{A}$$

where s_4 is the new index set of $\frac{dx}{dx}$.

seit 1558

Matrix calculus – Einstein Notation

$$C[s3] = g(A[s1])$$

forward mode autodiff:

$$\dot{C} = g'(A) *_{(s_1, s_1 s_4, s_3 s_4)} \dot{A}$$

where s_4 is the new index set of $\frac{dx}{dx}$.

reverse mode autodiff:

$$\bar{A} = g'(A) *_{(s_1, s_5 s_3, s_5 s_1)} \bar{C}$$

where s_5 is the new index set of $\frac{df}{df}$ (f - output function).

seit 1558

Matrix Calculus – Einstein Notation

Use Einstein notation for matrix calculus.

Matrix Calculus – Einstein Notation

Use Einstein notation for matrix calculus.

Matrix Calculus – Einstein Notation

Use Einstein notation for matrix calculus.

- $T = T[i, j, \ldots]$

Matrix Calculus – Einstein Notation

Use Einstein notation for matrix calculus.

- $T = T[i, j, \ldots]$
- forward and reverse mode autodiff

Matrix Calculus – Einstein Notation

Use Einstein notation for matrix calculus.

- $T = T[i, j, \ldots]$
- forward and reverse mode autodiff
- cross-country mode for highest efficiency

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

first approach based on Ricci notation

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

first approach based on Ricci notation

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
Computing Higher Order Derivatives of Matrix and Tensor Expressions. **(NeurIPS)**, 2018.

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

first approach based on Ricci notation

second approach based on generalized Einstein notation

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
Computing Higher Order Derivatives of Matrix and Tensor Expressions. **(NeurIPS)**, 2018.

seit 1558

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

first approach based on Ricci notation

second approach based on generalized Einstein notation

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
Computing Higher Order Derivatives of Matrix and Tensor Expressions. **(NeurIPS)**, 2018.

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
A Simple and Efficient Tensor Calculus. **(AAAI)**, 2020.

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

first approach based on Ricci notation

second approach based on generalized Einstein notation

simple, general, and efficient

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
Computing Higher Order Derivatives of Matrix and Tensor Expressions. **(NeurIPS)**, 2018.

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
A Simple and Efficient Tensor Calculus. **(AAAI)**, 2020.

seit 1558

Matrix and Tensor Calculus – Summary

symbolic vs. automatic differentiation

linear algebra notation not the right language for matrix and tensor calculus

first approach based on Ricci notation

second approach based on generalized Einstein notation

simple, general, and efficient

MatrixCalculus.org

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
Computing Higher Order Derivatives of Matrix and Tensor Expressions. **(NeurIPS)**, 2018.

Sören Laue, Matthias Mitterreiter, Joachim Giesen.
A Simple and Efficient Tensor Calculus. **(AAAI)**, 2020.

seit 1558