Automatic Differentiation

Gabriel Peyré

https://mathematical-tours.github.io

cnrs dépasser les frontières

1794 ENS ÉCOLE NORMALE SUPÉRIEURE

Automatic Differentiation

Setup: $f : \mathbb{R}^n \to \mathbb{R}$ computable in K operations.

Hypothesis: elementary operations $(a \times b, \log(a), \sqrt{a} \dots)$ and their derivatives cost $O(1)$.

Question: What is the complexity of computing $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$?

Automatic Differentiation

Setup: $f : \mathbb{R}^n \to \mathbb{R}$ computable in K operations.

Hypothesis: elementary operations $(a \times b, \log(a), \sqrt{a} \ldots)$ and their derivatives cost $O(1)$.

Question: What is the complexity of computing $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$?

Finite differences: $$\nabla f(x) \approx \frac{1}{\varepsilon}(f(x + \varepsilon\delta_1) - f(x), \ldots f(x + \varepsilon\delta_n) - f(x))$$

$K(n+1)$ operations, intractable for large n.

Automatic Differentiation

Setup: $f : \mathbb{R}^n \to \mathbb{R}$ computable in K operations.

Hypothesis: elementary operations $(a \times b, \log(a), \sqrt{a} \dots)$ and their derivatives cost $O(1)$.

Question: What is the complexity of computing $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$?

Finite differences: $$\nabla f(x) \approx \frac{1}{\varepsilon}(f(x+\varepsilon\delta_1) - f(x), \dots f(x+\varepsilon\delta_n) - f(x))$$

$K(n+1)$ operations, intractable for large n.

Theorem: there is an algorithm to compute ∇f in $O(K)$ operations. [Seppo Linnainmaa, 1970]

This algorithm is reverse mode automatic differentiation

→ it is not numerical calculus (exact computations).

→ it is not formal calculus (algorithms matter).

Python Libraries

PyTorch

TensorFlow

Forward Mode and Dual Numbers

Dual number associated to $(x, x') \in \mathbb{R}^2$: $x + \varepsilon x'$ with $\varepsilon^2 = 0$.

In particular: $(x + \varepsilon x')(y + \varepsilon y') = xy + \varepsilon(xy' + yx')$.

$$\frac{1}{x + \varepsilon x'} = \frac{1}{x} - \varepsilon \frac{x'}{x^2}$$

Forward Mode and Dual Numbers

Dual number associated to $(x, x') \in \mathbb{R}^2$: $x + \varepsilon x'$ with $\varepsilon^2 = 0$.

In particular: $(x + \varepsilon x')(y + \varepsilon y') = xy + \varepsilon(xy' + yx')$.

$$\frac{1}{x + \varepsilon x'} = \frac{1}{x} - \varepsilon \frac{x'}{x^2}$$

Proposition: if P is a polynomial, $P(x + \varepsilon) = P(x) + \varepsilon P'(x)$.

Forward Mode and Dual Numbers

Dual number associated to $(x, x') \in \mathbb{R}^2$: $x + \varepsilon x'$ with $\varepsilon^2 = 0$.

In particular: $(x + \varepsilon x')(y + \varepsilon y') = xy + \varepsilon(xy' + yx')$.

$$\frac{1}{x + \varepsilon x'} = \frac{1}{x} - \varepsilon \frac{x'}{x^2}$$

Proposition: if P is a polynomial, $P(x + \varepsilon) = P(x) + \varepsilon P'(x)$.

Function overloading: $f : \mathbb{R} \to \mathbb{R}$, $f(x + \varepsilon x') \stackrel{\text{def.}}{=} f(x) + \varepsilon f'(x) x'$.

Example: $\cos(x + \varepsilon x') = \cos(x) - \varepsilon x' \sin(x)$.

Proposition: $(f \circ g)(x + \varepsilon) = f(g(x)) + \varepsilon f'(g(x)) g'(x)$

Forward Mode and Dual Numbers

Dual number associated to $(x, x') \in \mathbb{R}^2$: $x + \varepsilon x'$ with $\varepsilon^2 = 0$.

In particular: $(x + \varepsilon x')(y + \varepsilon y') = xy + \varepsilon(xy' + yx')$.

$$\frac{1}{x + \varepsilon x'} = \frac{1}{x} - \varepsilon \frac{x'}{x^2}$$

Proposition: if P is a polynomial, $P(x + \varepsilon) = P(x) + \varepsilon P'(x)$.

Function overloading: $f : \mathbb{R} \to \mathbb{R}$, $f(x + \varepsilon x') \overset{\text{def.}}{=} f(x) + \varepsilon f'(x)x'$.

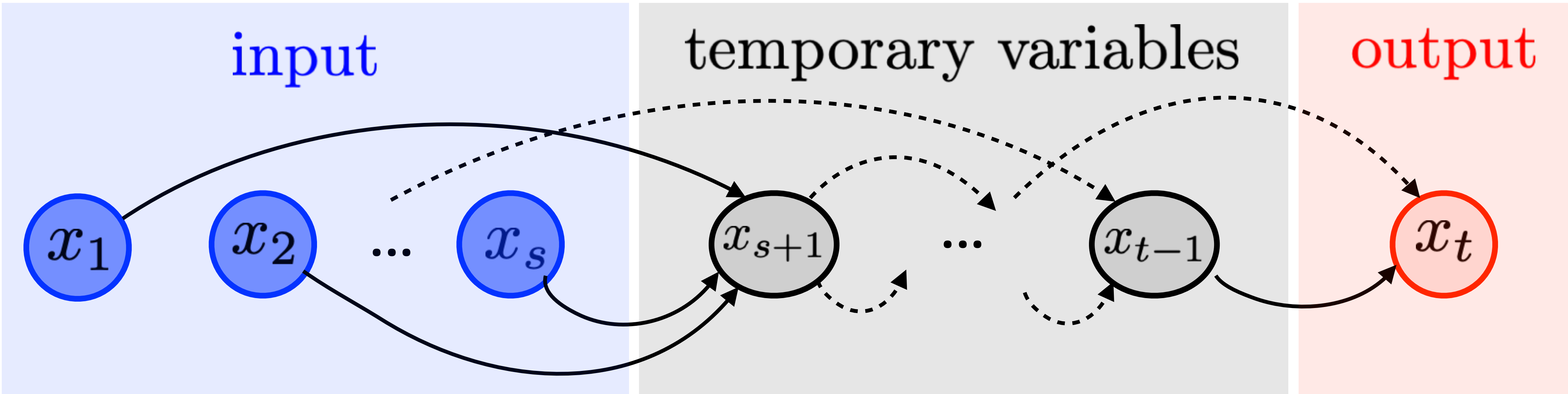
Example: $\cos(x + \varepsilon x') = \cos(x) - \varepsilon x' \sin(x)$.

Proposition: $(f \circ g)(x + \varepsilon) = f(g(x)) + \varepsilon f'(g(x))g'(x)$

Higher dimension: $f(x_1 + \varepsilon, x_1, \ldots, x_n) = f(x) + \varepsilon \frac{\partial f}{\partial x_1}(x)$

$\to$ complexity scales like $O(Kn) \sim$ finite differences.

Computational Graph



Computer program ⇔ directed acyclic graph ⇔ linear ordering of nodes $(x_k)_k$

forward

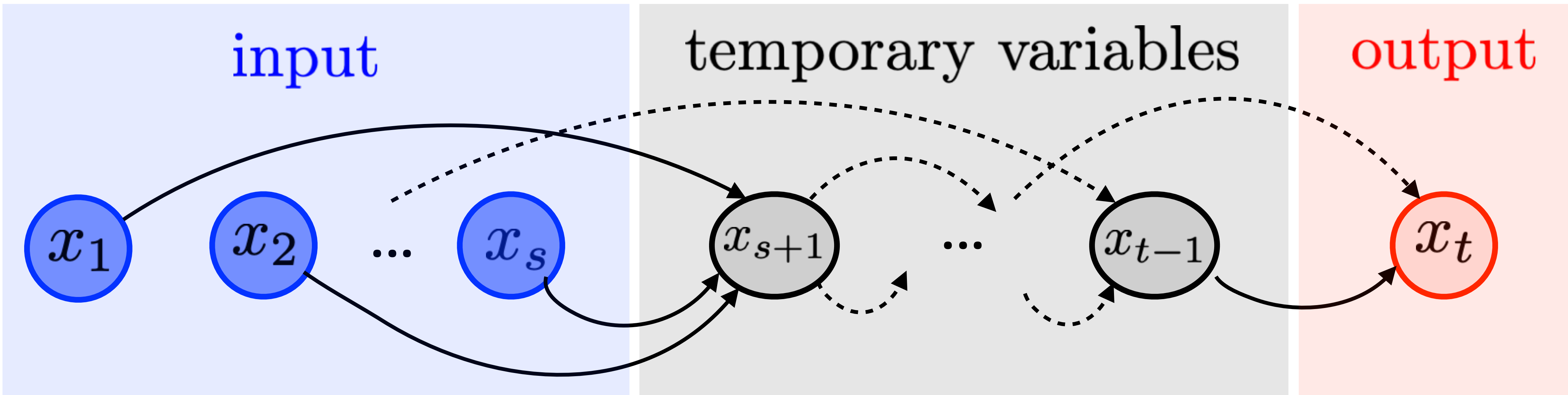
function $x_t = f(x_1, \ldots, x_s)$

  for $k = s+1, \ldots, t$

    $x_k = f_k(x_1, \ldots, x_{k-1})$

  return x_t

Computational Graph



Computer program $\Leftrightarrow$ directed acyclic graph $\Leftrightarrow$ linear ordering of nodes $(x_k)_k$

forward

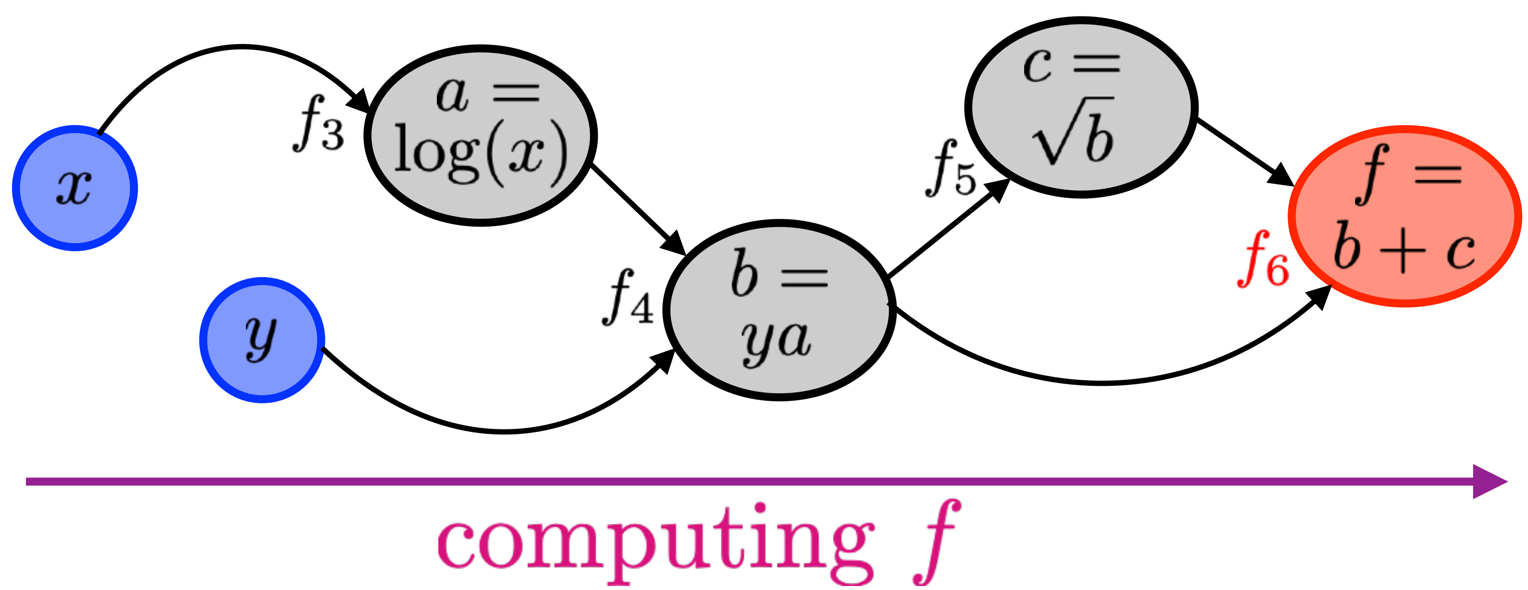
function $x_t = f(x_1, \ldots, x_s)$

  for $k = s+1, \ldots, t$

    $x_k = f_k(x_1, \ldots, x_{k-1})$

  return x_t

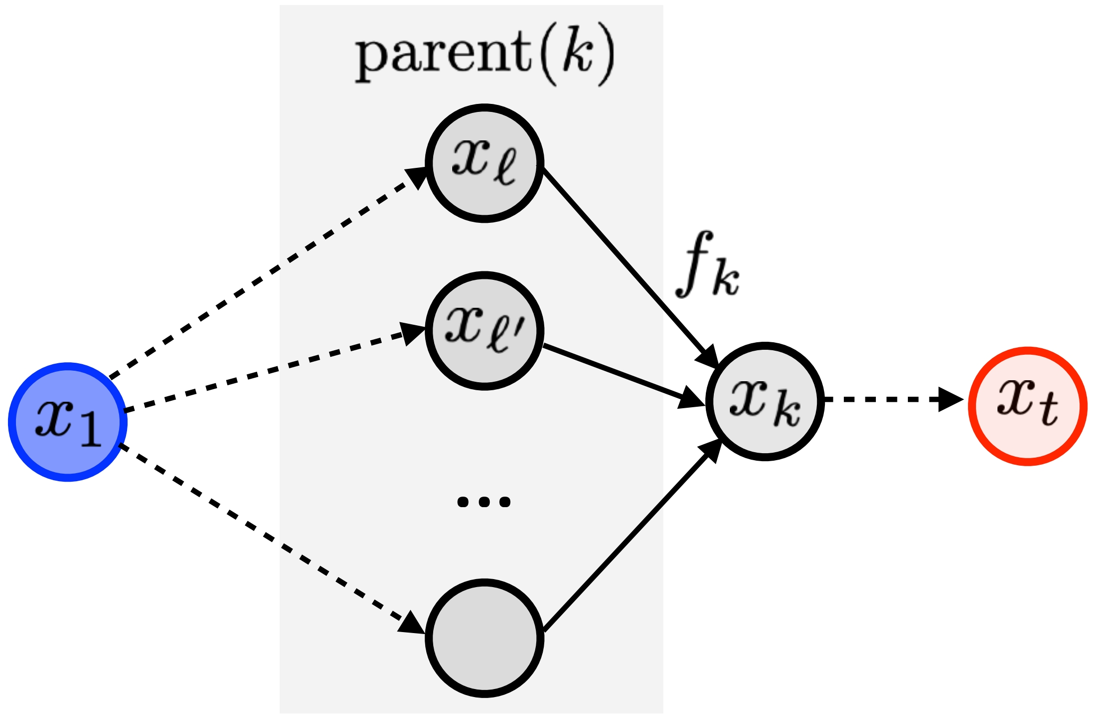
$$f(x, y) = y \log(x) + \sqrt{y \log(x)}$$



Forward Chain Rule

$$\frac{\partial x_k}{\partial x_1} = \text{“}\sum_{\ell \in \text{parent}(k)} \left[\frac{\partial x_k}{\partial x_\ell}\right] \times \frac{\partial x_\ell}{\partial x_1}\text{”}$$

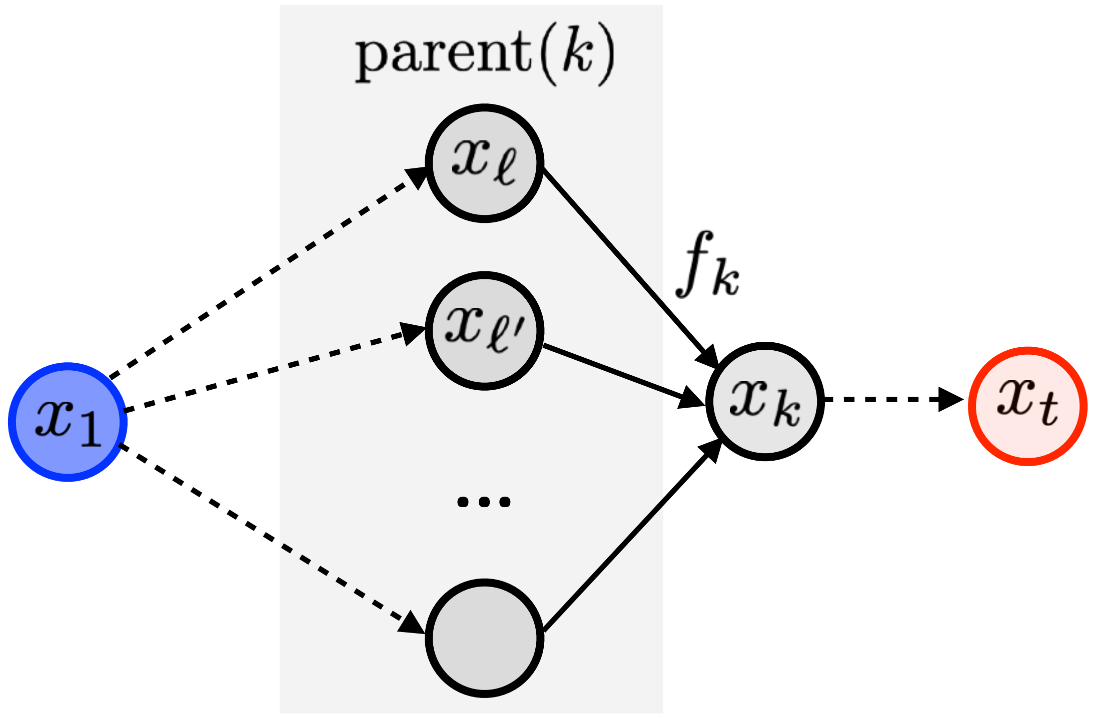
$$= \sum_{\ell \in \text{parent}(k)} \frac{\partial f_k}{\partial x_\ell}(x_1, \ldots, x_{k-1}) \times \frac{\partial x_\ell}{\partial x_1}$$



Forward Chain Rule

$$\frac{\partial x_k}{\partial x_1} = \text{“}\sum_{\ell \in \text{parent}(k)} \left[\frac{\partial x_k}{\partial x_\ell}\right] \times \frac{\partial x_\ell}{\partial x_1}\text{”}$$

$$= \sum_{\ell \in \text{parent}(k)} \frac{\partial f_k}{\partial x_\ell}(x_1, \ldots, x_{k-1}) \times \frac{\partial x_\ell}{\partial x_1}$$



forward

```
function $x_t = f(x_1, \ldots, x_s)$
    for $k = s+1, \ldots, t$
    |  $x_k = f_k(x_1, \ldots, x_{k-1})$
    return $x_t$
```

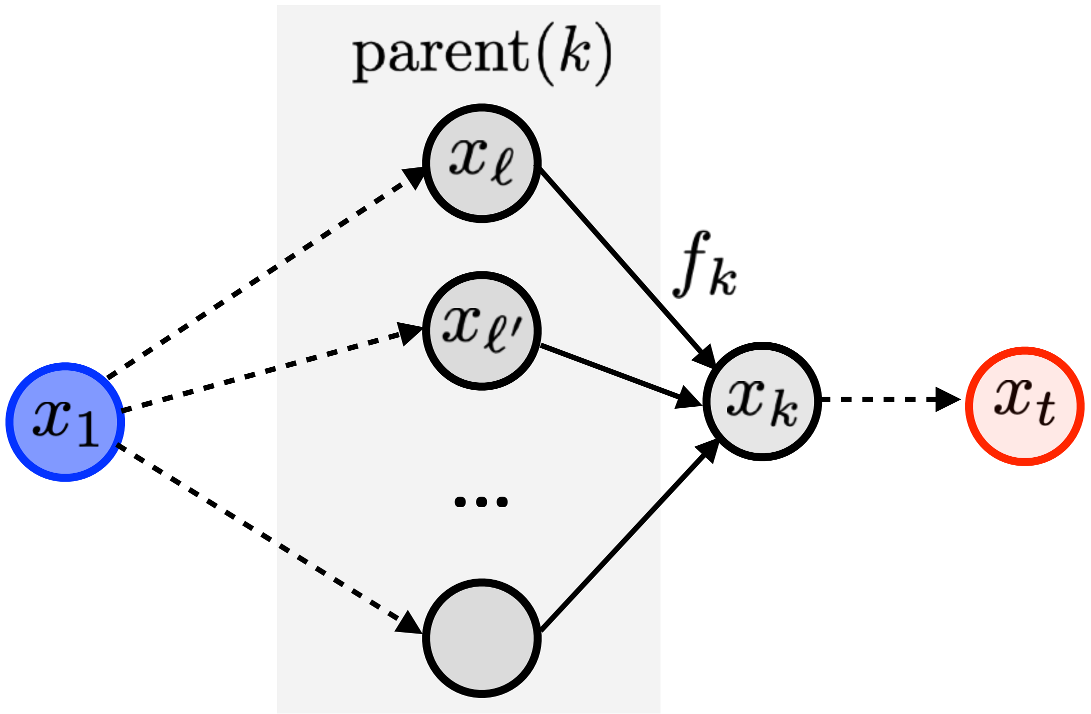
forward

```
function $\frac{\partial x_t}{\partial x_1} = \frac{\partial f}{\partial x_1}(x_1, \ldots, x_s)$
    $\frac{\partial x_1}{\partial x_1} = \mathrm{Id}_{n_1 \times n_1}$
    for $k = 2, \ldots, s$    $\frac{\partial x_k}{\partial x_1} = 0_{n_k \times n_1}$
    for $k = s+1, \ldots, t$
    |  $\frac{\partial x_k}{\partial x_1} = \sum_{\ell \in \text{parent}(k)} \frac{\partial f_k}{\partial x_\ell}(x_1, \ldots, x_{k-1}) \times \frac{\partial x_\ell}{\partial x_1}$
    return $\frac{\partial x_t}{\partial x_1}$
```

Forward Chain Rule

$$\frac{\partial x_k}{\partial x_1} = \text{“}\sum_{\ell \in \text{parent}(k)} \left[\frac{\partial x_k}{\partial x_\ell}\right] \times \frac{\partial x_\ell}{\partial x_1}\text{”}$$

$$= \sum_{\ell \in \text{parent}(k)} \frac{\partial f_k}{\partial x_\ell}(x_1, \ldots, x_{k-1}) \times \frac{\partial x_\ell}{\partial x_1}$$



forward

```
function x_t = f(x_1, ..., x_s)
    for k = s + 1, ..., t
        | x_k = f_k(x_1, ..., x_{k-1})
    return x_t
```

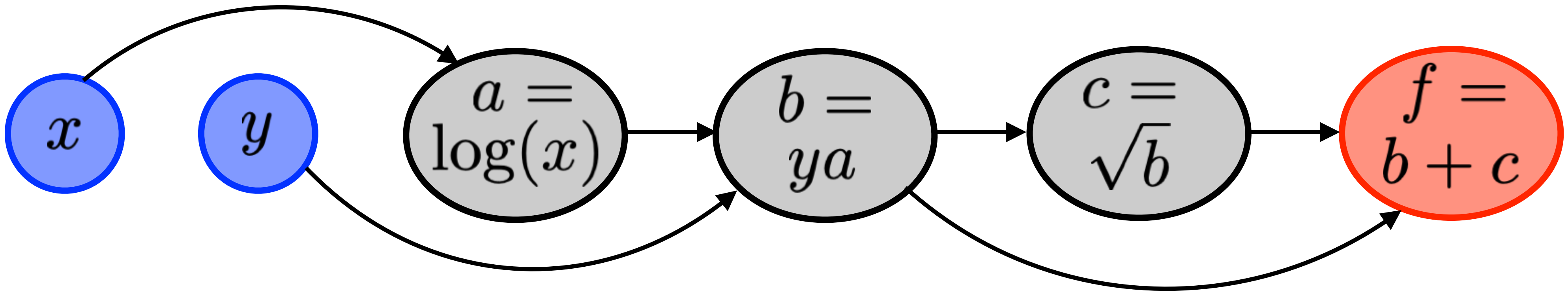
forward

```
function ∂x_t/∂x_1 = ∂f/∂x_1 (x_1, ..., x_s)
    ∂x_1/∂x_1 = Id_{n_1 × n_1}
    for k = 2, ..., s      ∂x_k/∂x_1 = 0_{n_k × n_1}
    for k = s + 1, ..., t
        | ∂x_k/∂x_1 = Σ_{ℓ ∈ parent(k)} ∂f_k/∂x_ℓ (x_1, ..., x_{k-1}) × ∂x_ℓ/∂x_1
    return ∂x_t/∂x_1
```

Assuming $\begin{cases} |\text{parent}(k)| = O(1), \\ n_k = O(1) \end{cases}$ $\rightarrow$ Complexity: $O(K \sum_{k=1}^{s} n_k) \sim$ finite differences.

Example

$$f(x, y) = y \log(x) + \sqrt{y \log(x)}$$

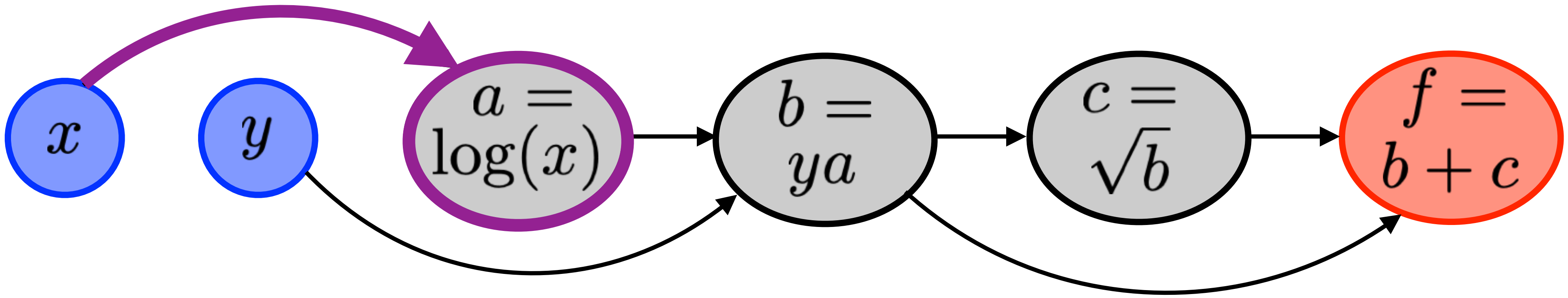


$$\frac{\partial f}{\partial x}$$

$$\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0$$

Example

$$f(x, y) = y\log(x) + \sqrt{y\log(x)}$$



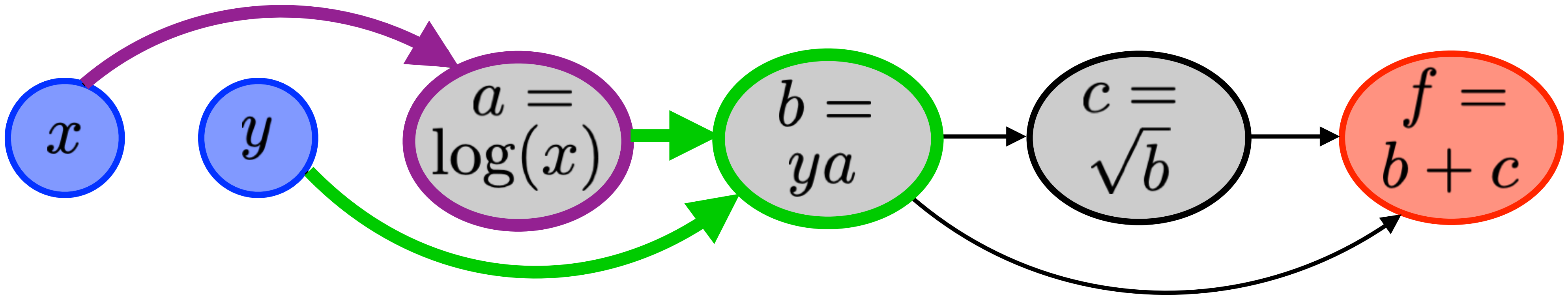
$$\frac{\partial f}{\partial x}$$

$$\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0$$

$$\frac{\partial a}{\partial x} = \left[\frac{\partial a}{\partial x}\right]\frac{\partial x}{\partial x} = \frac{1}{x}\frac{\partial x}{\partial x} \qquad \{x \mapsto a = \log(x)\}$$

Example

$$f(x, y) = y \log(x) + \sqrt{y \log(x)}$$



$$\frac{\partial f}{\partial x}$$

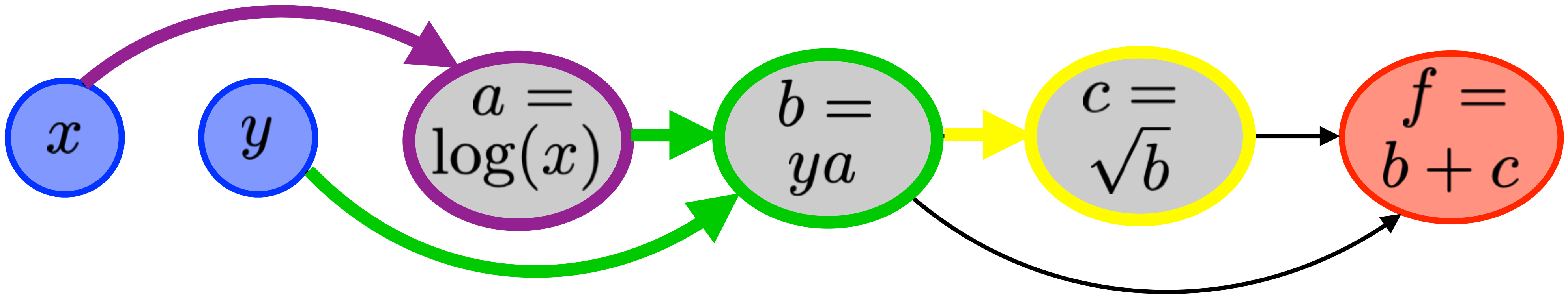
$$\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0$$

$$\frac{\partial a}{\partial x} = \left[\frac{\partial a}{\partial x}\right] \frac{\partial x}{\partial x} = \frac{1}{x} \frac{\partial x}{\partial x} \qquad \{x \mapsto a = \log(x)\}$$

$$\frac{\partial b}{\partial x} = \left[\frac{\partial b}{\partial a}\right] \frac{\partial a}{\partial x} + \left[\frac{\partial b}{\partial y}\right] \frac{\partial y}{\partial x} = y \frac{\partial a}{\partial x} + a \frac{\partial y}{\partial x} \qquad \{(y, a) \mapsto b = ya\}$$

Example

$$f(x,y) = y\log(x) + \sqrt{y\log(x)}$$



$$\frac{\partial f}{\partial x}$$

$$\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0$$

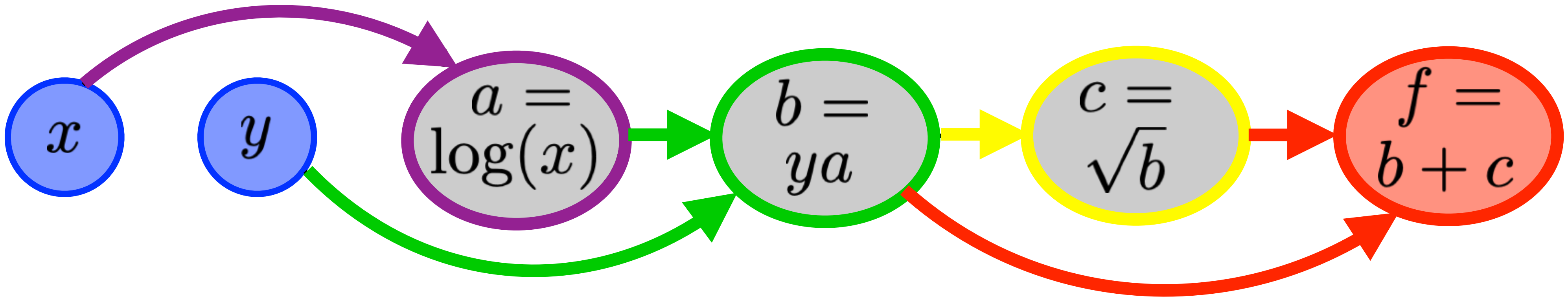
$$\frac{\partial a}{\partial x} = \left[\frac{\partial a}{\partial x}\right]\frac{\partial x}{\partial x} = \frac{1}{x}\frac{\partial x}{\partial x} \qquad \{x \mapsto a = \log(x)\}$$

$$\frac{\partial b}{\partial x} = \left[\frac{\partial b}{\partial a}\right]\frac{\partial a}{\partial x} + \left[\frac{\partial b}{\partial y}\right]\frac{\partial y}{\partial x} = y\frac{\partial a}{\partial x} + a\frac{\partial y}{\partial x} \qquad \{(y,a) \mapsto b = ya\}$$

$$\frac{\partial c}{\partial x} = \left[\frac{\partial c}{\partial b}\right]\frac{\partial b}{\partial x} = \frac{1}{2\sqrt{b}}\frac{\partial b}{\partial x} \qquad \{b \mapsto c = \sqrt{b}\}$$

Example

$$f(x, y) = y \log(x) + \sqrt{y \log(x)}$$



$$\frac{\partial f}{\partial x}$$

$$\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0$$

$$\frac{\partial a}{\partial x} = \left[\frac{\partial a}{\partial x}\right] \frac{\partial x}{\partial x} = \frac{1}{x} \frac{\partial x}{\partial x} \qquad \{x \mapsto a = \log(x)\}$$

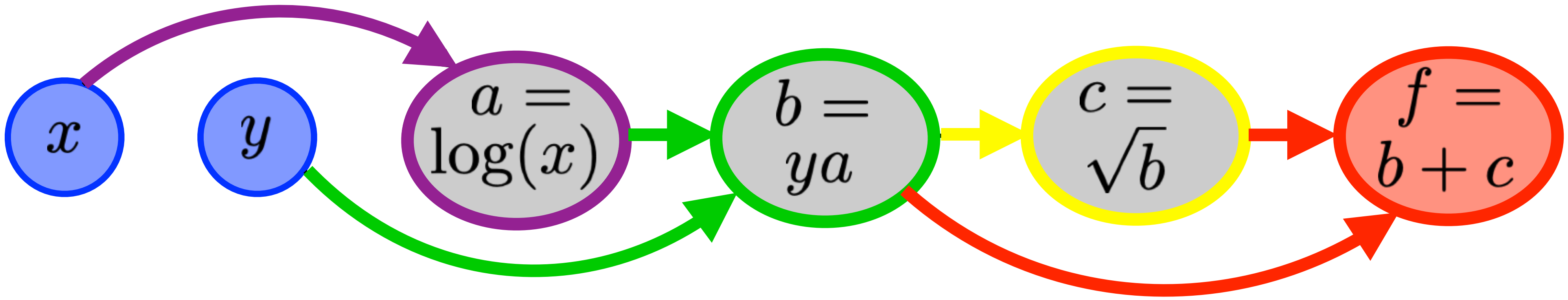
$$\frac{\partial b}{\partial x} = \left[\frac{\partial b}{\partial a}\right] \frac{\partial a}{\partial x} + \left[\frac{\partial b}{\partial y}\right] \frac{\partial y}{\partial x} = y \frac{\partial a}{\partial x} + a \frac{\partial y}{\partial x} \qquad \{(y, a) \mapsto b = ya\}$$

$$\frac{\partial c}{\partial x} = \left[\frac{\partial c}{\partial b}\right] \frac{\partial b}{\partial x} = \frac{1}{2\sqrt{b}} \frac{\partial b}{\partial x} \qquad \{b \mapsto c = \sqrt{b}\}$$

$$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial b}\right] \frac{\partial b}{\partial x} + \left[\frac{\partial f}{\partial c}\right] \frac{\partial c}{\partial x} = 1 \frac{\partial b}{\partial x} + 1 \frac{\partial c}{\partial x} \qquad \{(b, c) \mapsto f = b + c\}$$

Example

$$f(x,y) = y\log(x) + \sqrt{y\log(x)}$$



$$\frac{\partial f}{\partial x}$$

$$\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0$$

$$\frac{\partial a}{\partial x} = \left[\frac{\partial a}{\partial x}\right]\frac{\partial x}{\partial x} = \frac{1}{x}\frac{\partial x}{\partial x} \qquad \{x \mapsto a = \log(x)\}$$

$$\frac{\partial b}{\partial x} = \left[\frac{\partial b}{\partial a}\right]\frac{\partial a}{\partial x} + \left[\frac{\partial b}{\partial y}\right]\frac{\partial y}{\partial x} = y\frac{\partial a}{\partial x} + a\frac{\partial y}{\partial x} \qquad \{(y,a) \mapsto b = ya\}$$

$$\frac{\partial c}{\partial x} = \left[\frac{\partial c}{\partial b}\right]\frac{\partial b}{\partial x} = \frac{1}{2\sqrt{b}}\frac{\partial b}{\partial x} \qquad \{b \mapsto c = \sqrt{b}\}$$

$$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial b}\right]\frac{\partial b}{\partial x} + \left[\frac{\partial f}{\partial c}\right]\frac{\partial c}{\partial x} = 1\frac{\partial b}{\partial x} + 1\frac{\partial c}{\partial x} \qquad \{(b,c) \mapsto f = b + c\}$$

$$\frac{\partial f}{\partial y}$$

$$\frac{\partial x}{\partial y} = 0, \quad \frac{\partial y}{\partial y} = 1$$

$$\frac{\partial a}{\partial y} = \left[\frac{\partial a}{\partial x}\right]\frac{\partial x}{\partial y} = 0 \qquad \{x \mapsto a = \log(x)\}$$

$$\frac{\partial b}{\partial y} = \left[\frac{\partial b}{\partial a}\right]\frac{\partial a}{\partial y} + \left[\frac{\partial b}{\partial y}\right]\frac{\partial y}{\partial y} \qquad \{(y,a) \mapsto b = ya\}$$

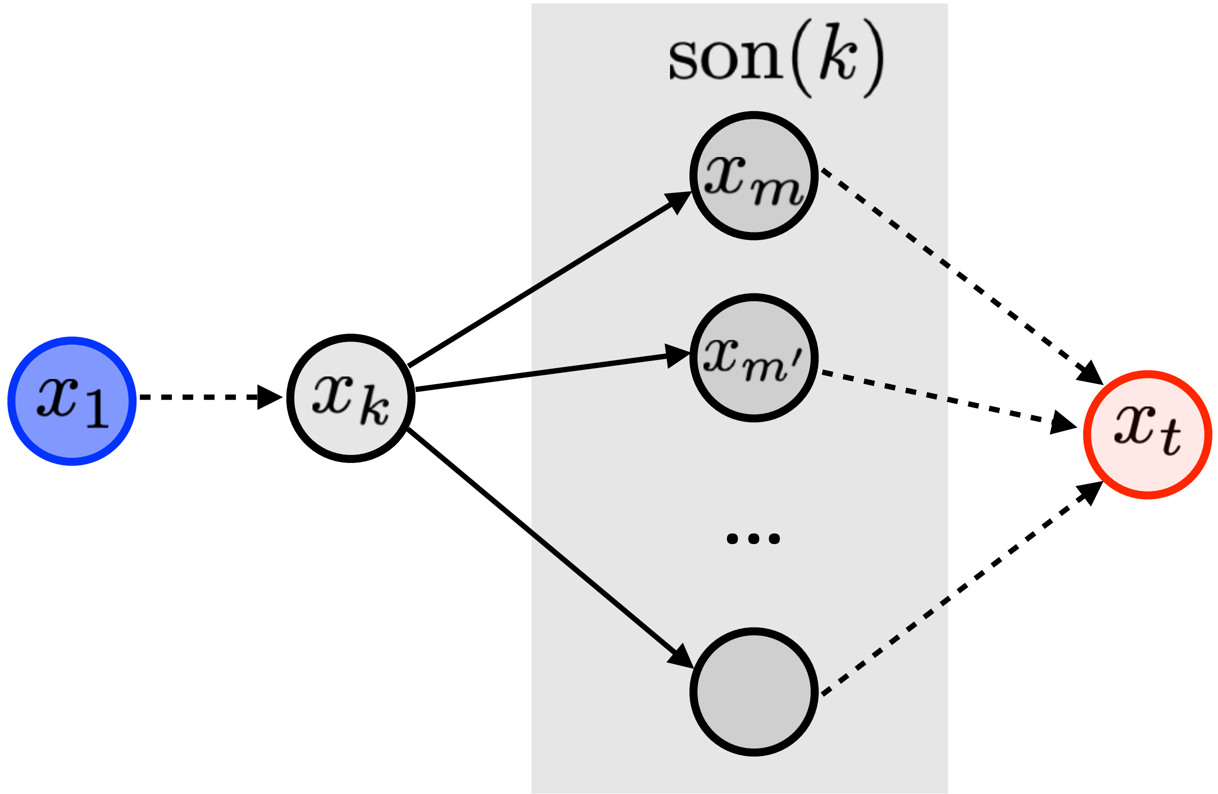
$$\frac{\partial c}{\partial y} = \left[\frac{\partial c}{\partial b}\right]\frac{\partial b}{\partial y} = \frac{1}{2\sqrt{b}}\frac{\partial b}{\partial y} \qquad \{b \mapsto c = \sqrt{b}\}$$

$$\frac{\partial f}{\partial y} = \left[\frac{\partial f}{\partial b}\right]\frac{\partial b}{\partial y} + \left[\frac{\partial f}{\partial c}\right]\frac{\partial c}{\partial y} \qquad \{(b,c) \mapsto f = b + c\}$$

Backward Chain Rule

$$\frac{\partial x_t}{\partial x_k} = \text{``}\sum_{m \in \mathrm{son}(k)} \frac{\partial x_t}{\partial x_m} \times \left[\frac{\partial x_m}{\partial x_k}\right]\text{''}$$

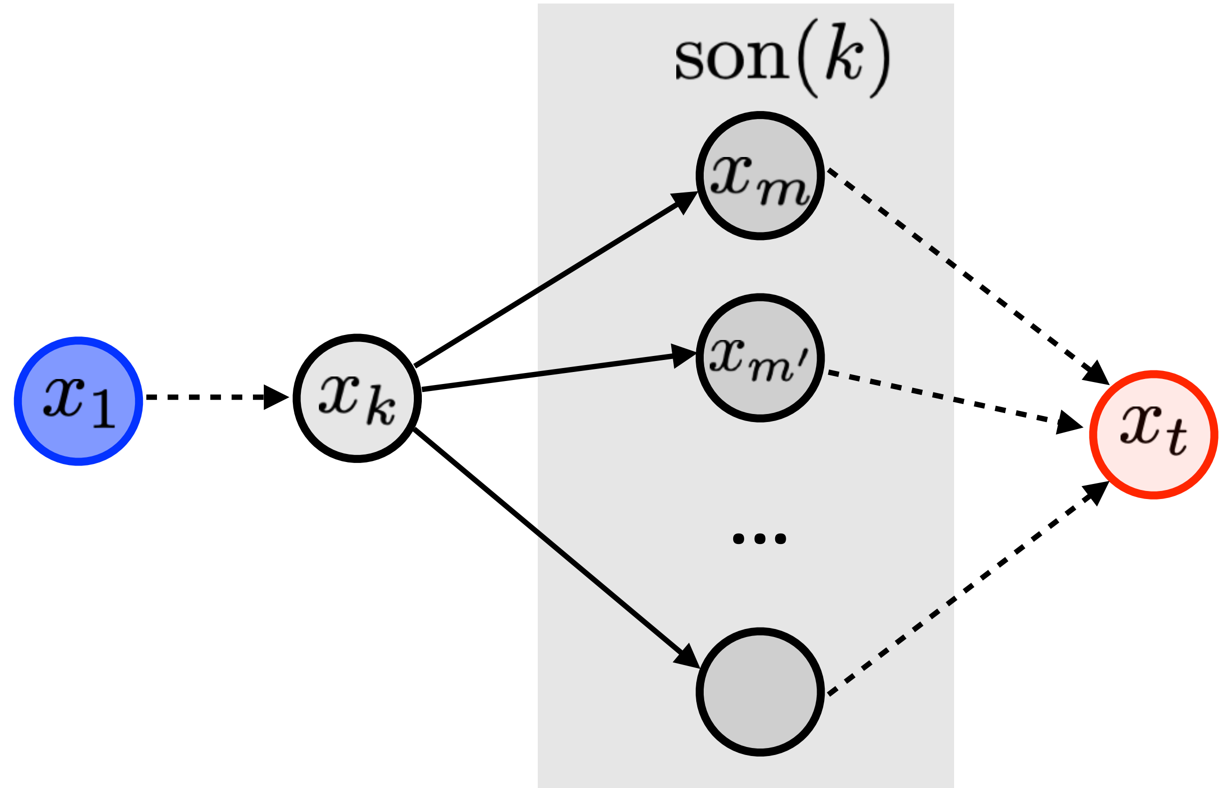
$$= \sum_{m \in \mathrm{son}(k)} \frac{\partial x_t}{\partial x_m} \times \frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}$$



Backward Chain Rule

$$\frac{\partial x_t}{\partial x_k} = \text{“}\sum_{m\in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \left[\frac{\partial x_m}{\partial x_k}\right]\text{”}$$

$$= \sum_{m\in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \frac{\partial f_m(x_1,\dots,x_m)}{\partial x_k}$$



forward

```
function x_t = f(x_1, ..., x_s)
  for k = s+1, ..., t
  | x_k = f_k(x_1, ..., x_{k-1})
  return x_t
```

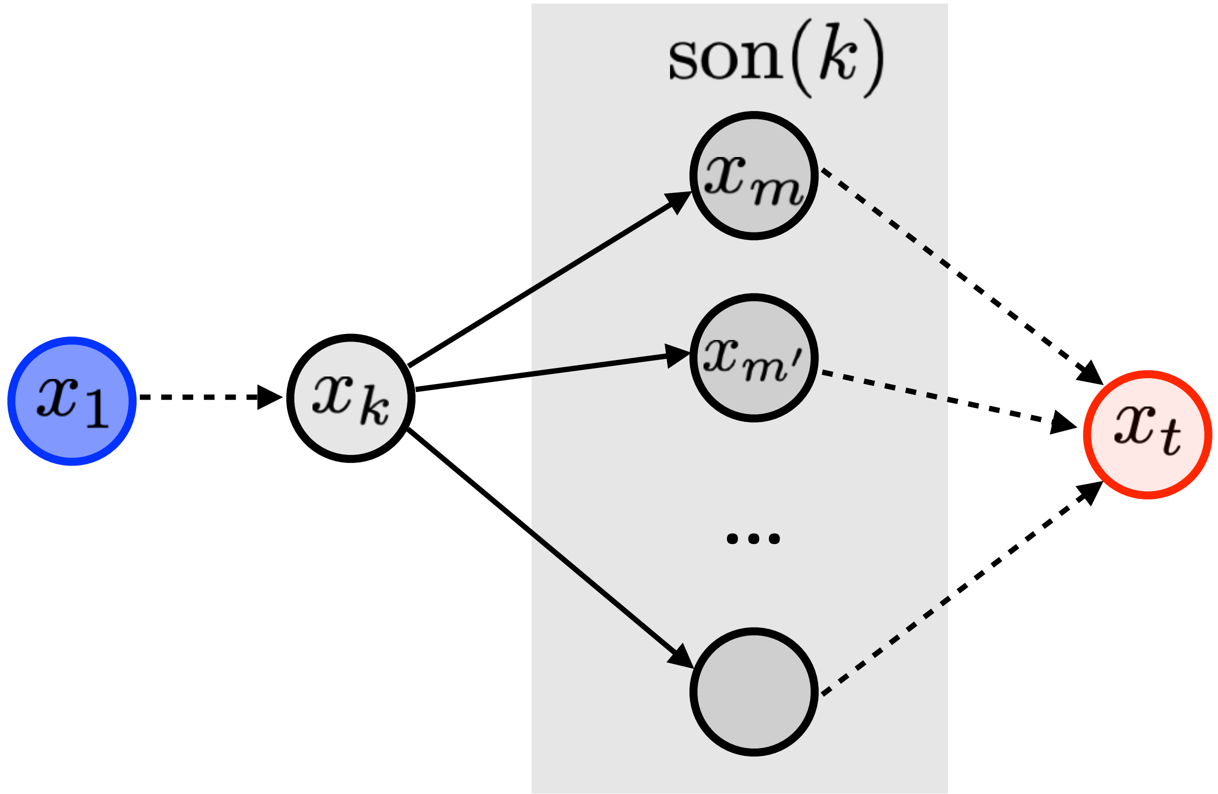
backward

```
function ∂f/∂(x_1,...,x_s) (x_1, ..., x_s)
  ∂x_t/∂x_t = Id_{n_t × n_t}
  for k = t-1, t-2, ..., 1
  | ∂x_t/∂x_k = Σ_{m∈son(k)} ∂x_t/∂x_m × ∂f_m(x_1,...,x_m)/∂x_k
  return (∂x_t/∂x_1, ..., ∂x_t/∂x_s)
```

Backward Chain Rule

$$\frac{\partial x_t}{\partial x_k} = \text{“} \sum_{m \in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \left[\frac{\partial x_m}{\partial x_k}\right] \text{”}$$

$$= \sum_{m \in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}$$



forward

function $x_t = f(x_1, \ldots, x_s)$
for $k = s+1, \ldots, t$
| $x_k = f_k(x_1, \ldots, x_{k-1})$
return x_t

backward

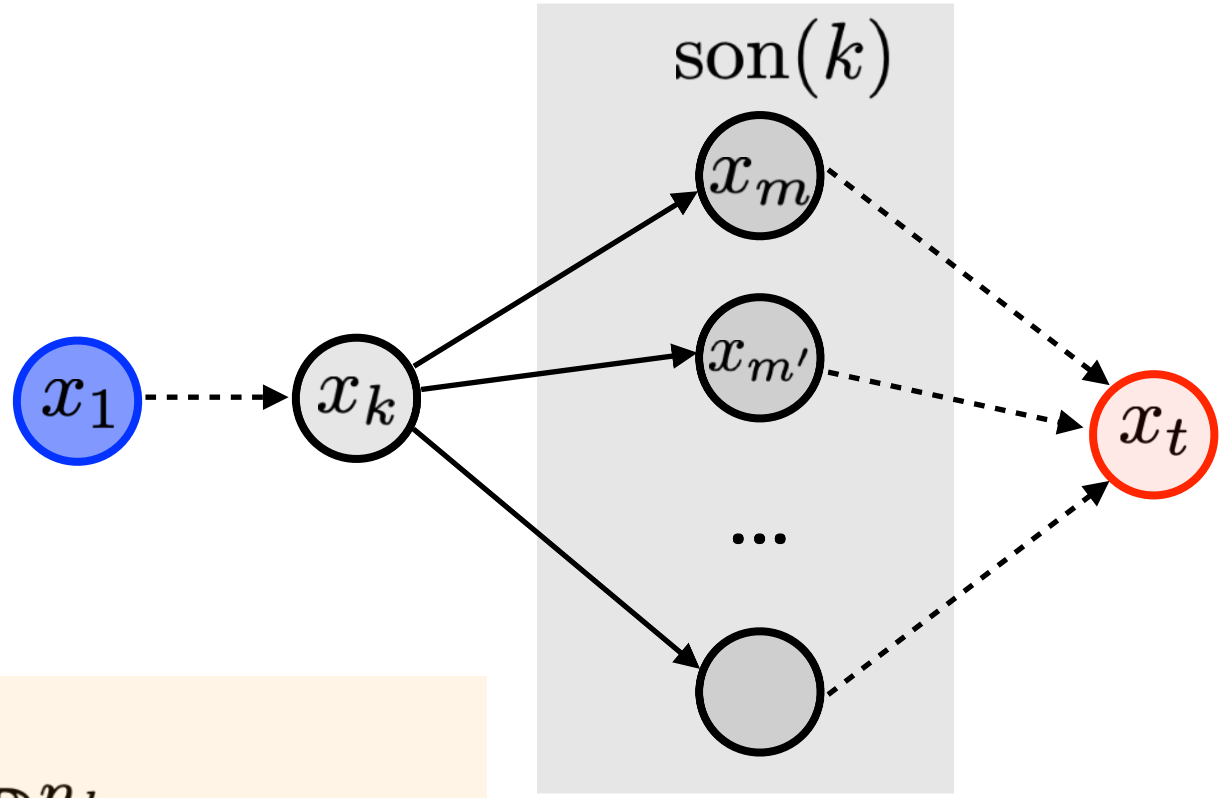
function $\frac{\partial f}{\partial(x_1, \ldots, x_s)}(x_1, \ldots, x_s)$
$\frac{\partial x_t}{\partial x_t} = \text{Id}_{n_t \times n_t}$
for $k = t-1, t-2, \ldots, 1$
| $\frac{\partial x_t}{\partial x_k} = \sum_{m \in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}$
return $\left(\frac{\partial x_t}{\partial x_1}, \ldots, \frac{\partial x_t}{\partial x_s}\right)$

$\rightarrow$ needs to store all intermediate $(x_k)_k$ in memory.

Gradient Backpropagation

$$\frac{\partial x_t}{\partial x_k} = \text{``}\sum_{m \in \mathrm{son}(k)} \frac{\partial x_t}{\partial x_m} \times \left[\frac{\partial x_m}{\partial x_k}\right]\text{''}$$

$$= \sum_{m \in \mathrm{son}(k)} \frac{\partial x_t}{\partial x_m} \times \frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}$$



If $n_t = 1$: $\nabla_{x_k} f = \left(\frac{\partial x_t}{\partial x_k}\right)^\top \in \mathbb{R}^{n_k}$

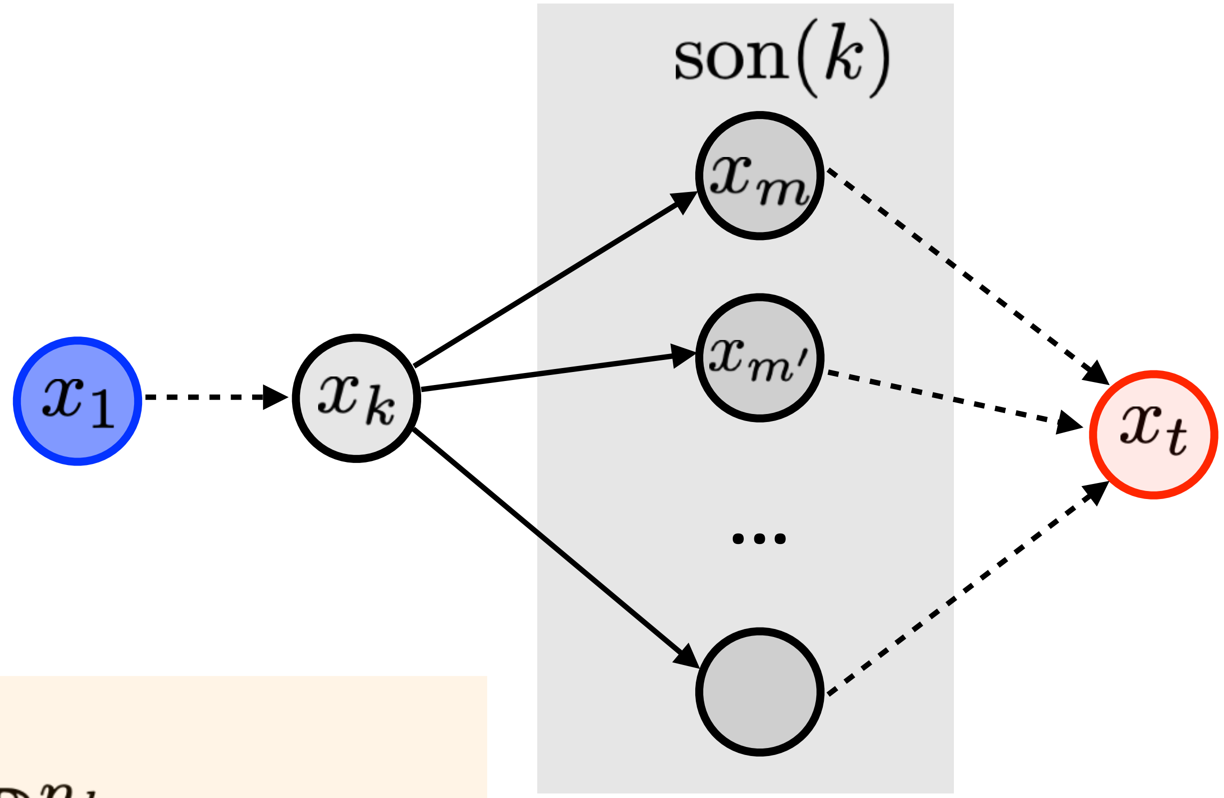
Back-propagation of gradients:

$$\nabla_{x_k} f = \sum_{m \in \mathrm{son}(k)} \left(\frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}\right)^\top \nabla_{x_m} f$$

Gradient Backpropagation

$$\frac{\partial x_t}{\partial x_k} = \text{“} \sum_{m \in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \left[\frac{\partial x_m}{\partial x_k}\right] \text{”}$$

$$= \sum_{m \in \text{son}(k)} \frac{\partial x_t}{\partial x_m} \times \frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}$$



If $n_t = 1$: $\nabla_{x_k} f = \left(\frac{\partial x_t}{\partial x_k}\right)^\top \in \mathbb{R}^{n_k}$

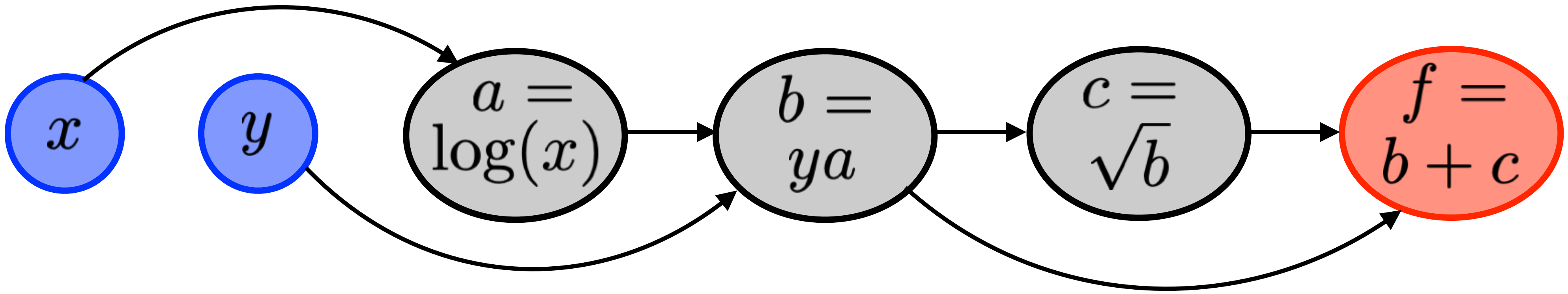
Back-propagation of gradients:

$$\nabla_{x_k} f = \sum_{m \in \text{son}(k)} \left(\frac{\partial f_m(x_1, \ldots, x_m)}{\partial x_k}\right)^\top \nabla_{x_m} f$$

Assuming $\begin{cases} |\text{parent}(k)| = O(1), \\ n_k = O(1) \end{cases} \longrightarrow$ Complexity: $O(K) \ll$ finite differences.

Example

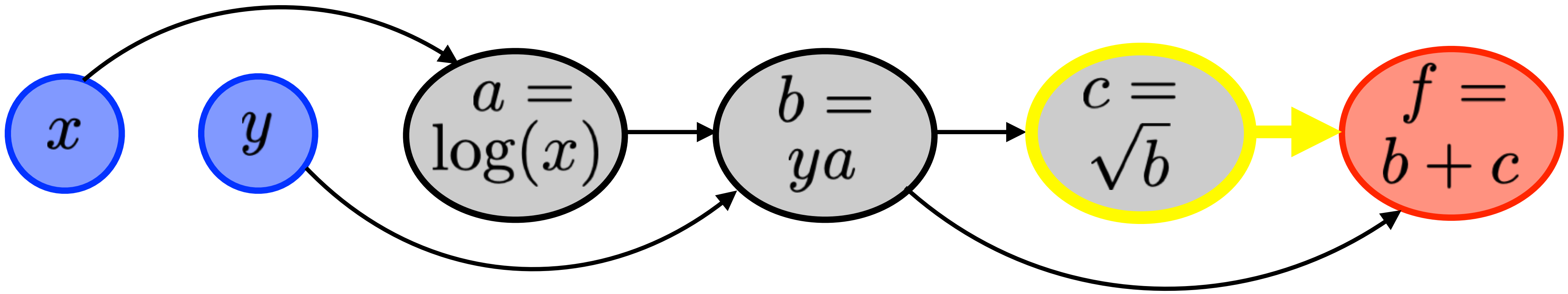
$$f(x, y) = y \log(x) + \sqrt{y \log(x)}$$



$$\frac{\partial f}{\partial f} = 1$$

Example

$$f(x, y) = y\log(x) + \sqrt{y\log(x)}$$



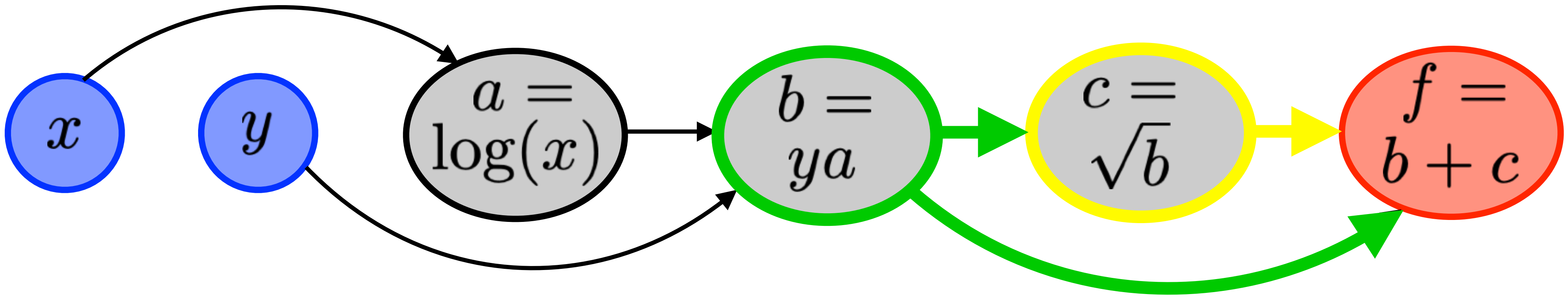
$$\frac{\partial f}{\partial f} = 1$$

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial c}\right] = \frac{\partial f}{\partial f}1$$

$$\{c \mapsto f = b + c\}$$

Example

$$f(x, y) = y\log(x) + \sqrt{y\log(x)}$$



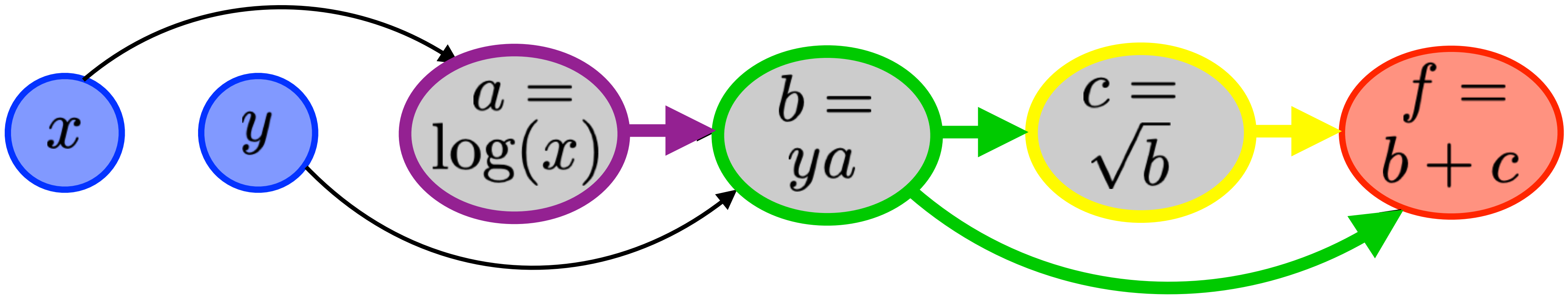
$$\frac{\partial f}{\partial f} = 1$$

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial c}\right] = \frac{\partial f}{\partial f}1 \qquad \{c \mapsto f = b + c\}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\left[\frac{\partial c}{\partial b}\right] + \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial b}\right] = \frac{\partial f}{\partial c}\frac{1}{2\sqrt{b}} + \frac{\partial f}{\partial f}1 \qquad \{b \mapsto c = \sqrt{b}, b \mapsto f = b + c\}$$

Example

$$f(x,y) = y\log(x) + \sqrt{y\log(x)}$$



$$\frac{\partial f}{\partial f} = 1$$

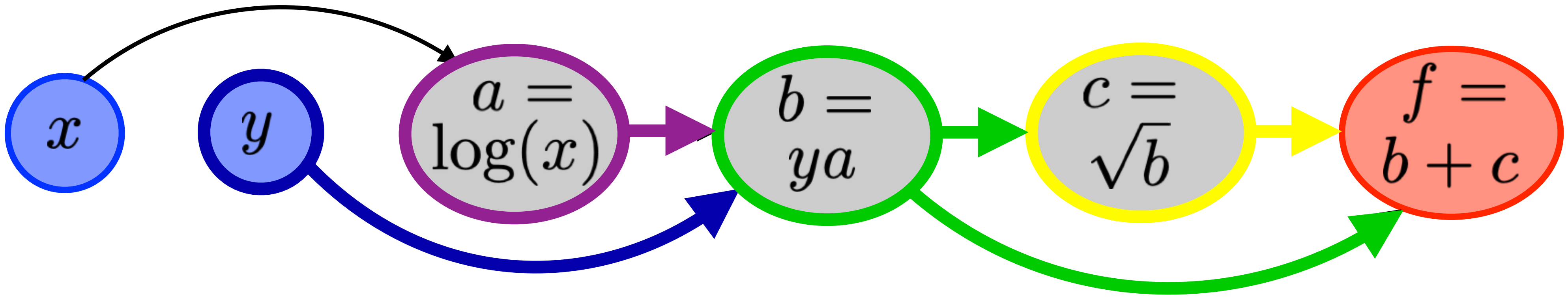
$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial c}\right] = \frac{\partial f}{\partial f}1 \qquad \{c \mapsto f = b + c\}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\left[\frac{\partial c}{\partial b}\right] + \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial b}\right] = \frac{\partial f}{\partial c}\frac{1}{2\sqrt{b}} + \frac{\partial f}{\partial f}1 \qquad \{b \mapsto c = \sqrt{b}, b \mapsto f = b + c\}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\left[\frac{\partial b}{\partial a}\right] = \frac{\partial f}{\partial b}y \qquad \{a \mapsto b = ya\}$$

Example

$$f(x, y) = y\log(x) + \sqrt{y\log(x)}$$



$$\frac{\partial f}{\partial f} = 1$$

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial c}\right] = \frac{\partial f}{\partial f}1 \qquad \{c \mapsto f = b + c\}$$

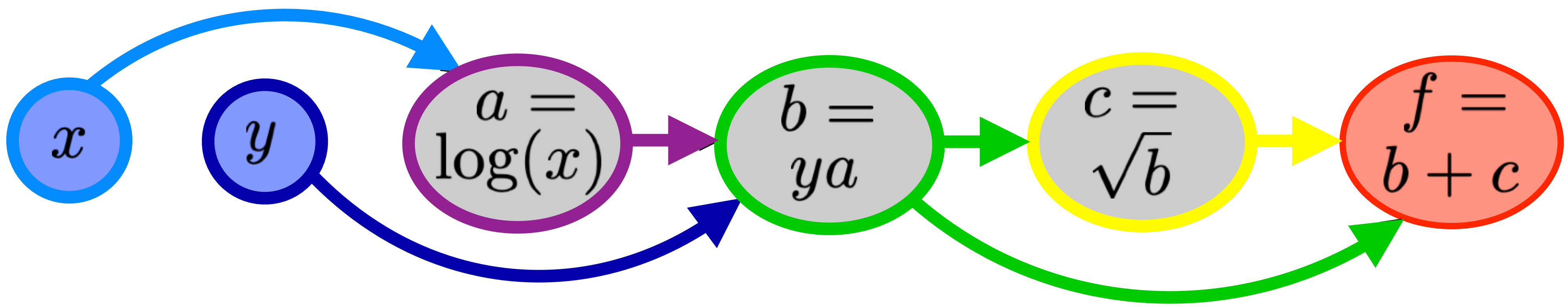
$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\left[\frac{\partial c}{\partial b}\right] + \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial b}\right] = \frac{\partial f}{\partial c}\frac{1}{2\sqrt{b}} + \frac{\partial f}{\partial f}1 \qquad \{b \mapsto c = \sqrt{b}, b \mapsto f = b + c\}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\left[\frac{\partial b}{\partial a}\right] = \frac{\partial f}{\partial b}y \qquad \{a \mapsto b = ya\}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial b}\left[\frac{\partial b}{\partial y}\right] = \frac{\partial f}{\partial b}a \qquad \{y \mapsto b = ya\}$$

Example

$$f(x, y) = y\log(x) + \sqrt{y\log(x)}$$



$$\frac{\partial f}{\partial f} = 1$$

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial c}\right] = \frac{\partial f}{\partial f}1 \qquad \{c \mapsto f = b + c\}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\left[\frac{\partial c}{\partial b}\right] + \frac{\partial f}{\partial f}\left[\frac{\partial f}{\partial b}\right] = \frac{\partial f}{\partial c}\frac{1}{2\sqrt{b}} + \frac{\partial f}{\partial f}1 \qquad \{b \mapsto c = \sqrt{b}, b \mapsto f = b + c\}$$

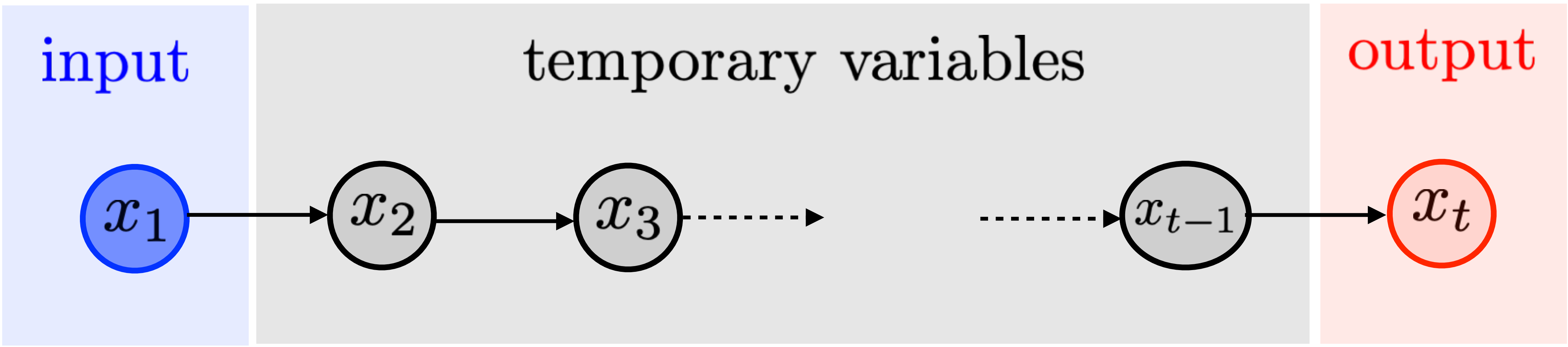
$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\left[\frac{\partial b}{\partial a}\right] = \frac{\partial f}{\partial b}y \qquad \{a \mapsto b = ya\}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial b}\left[\frac{\partial b}{\partial y}\right] = \frac{\partial f}{\partial b}a \qquad \{y \mapsto b = ya\}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\left[\frac{\partial a}{\partial x}\right] = \frac{\partial f}{\partial a}\frac{1}{x} \qquad \{x \mapsto a = \log(x)\}$$

Differentiating Composition of Functions

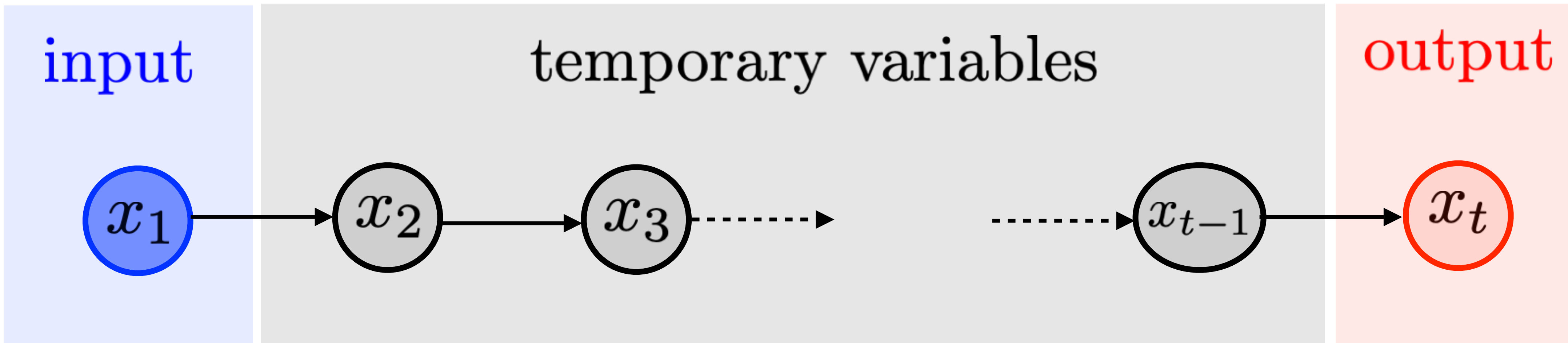
$$f = f_t \circ f_{t-1} \circ \ldots \circ f_2 \circ f_1 \qquad x_k = f_k(x_{k-1})$$



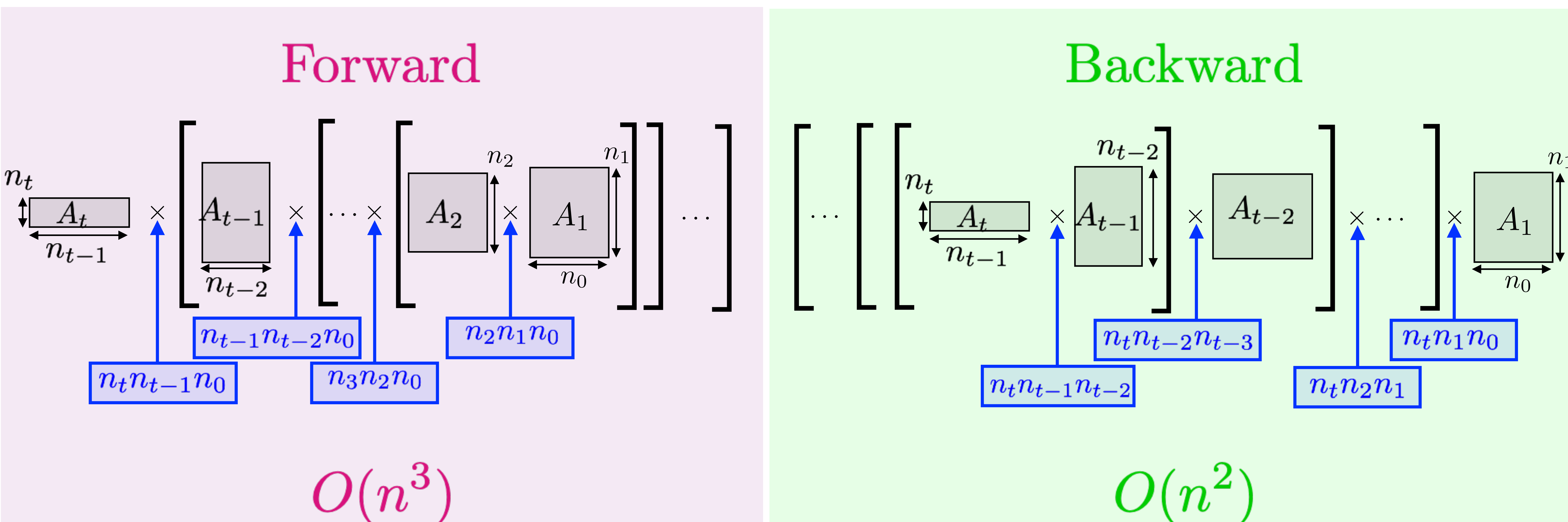
$$\partial f(x) = A_t \times A_{t-1} \times \ldots A_2 \times A_1 \qquad A_k \overset{\text{def.}}{=} \partial f_k(x_{k-1}) \in \mathbb{R}^{n_k \times n_{k-1}}$$

Differentiating Composition of Functions

$$f = f_t \circ f_{t-1} \circ \ldots \circ f_2 \circ f_1 \qquad x_k = f_k(x_{k-1})$$

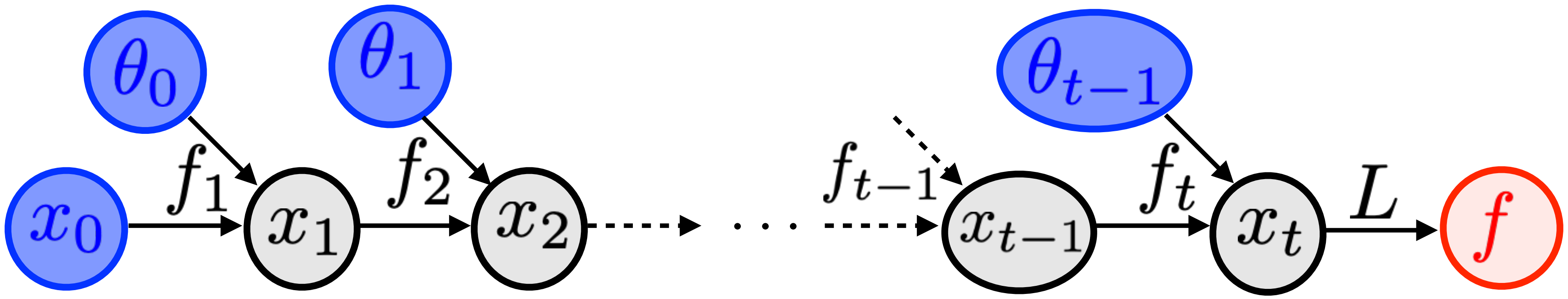


$$\partial f(x) = A_t \times A_{t-1} \times \ldots A_2 \times A_1 \qquad A_k \stackrel{\text{def.}}{=} \partial f_k(x_{k-1}) \in \mathbb{R}^{n_k \times n_{k-1}}$$



(if $n_t = 1$, $n_k = n$)

Feedforward Architecture



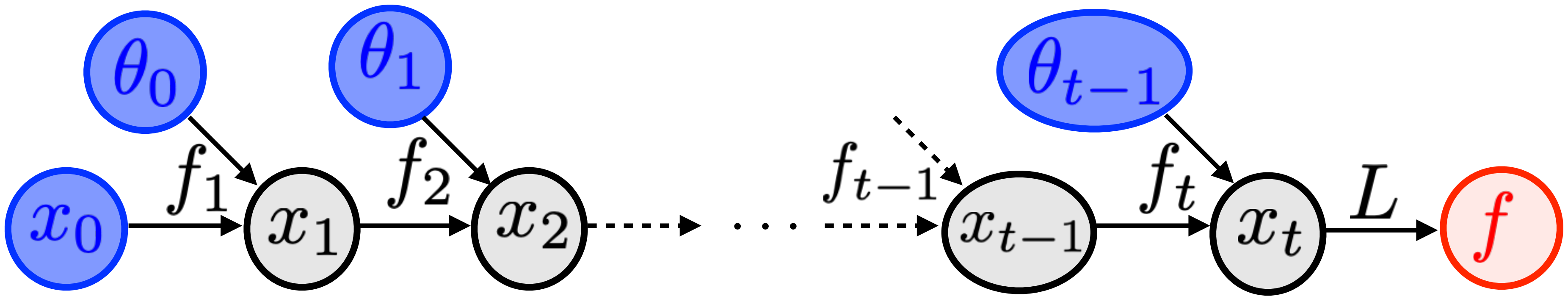
forward

$$\texttt{for } k = 1, \ldots, t-1, t$$

$$x_k = f_k(x_{k-1}, \theta_{k-1})$$

$$f(\theta) \stackrel{\text{def.}}{=} L(x_t)$$

Feedforward Architecture



forward

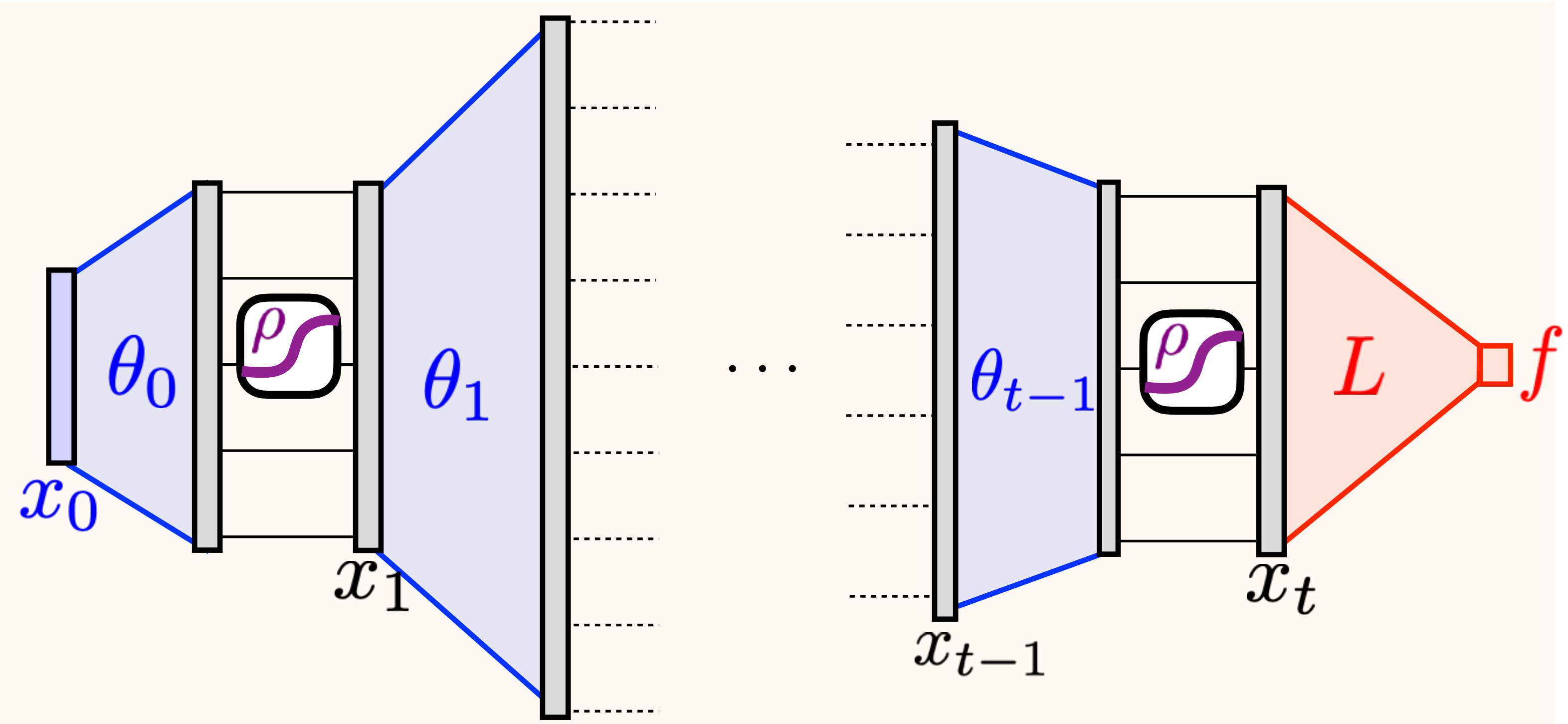
$\texttt{for } k = 1, \ldots, t-1, t$

$$x_k = f_k(x_{k-1}, \theta_{k-1})$$

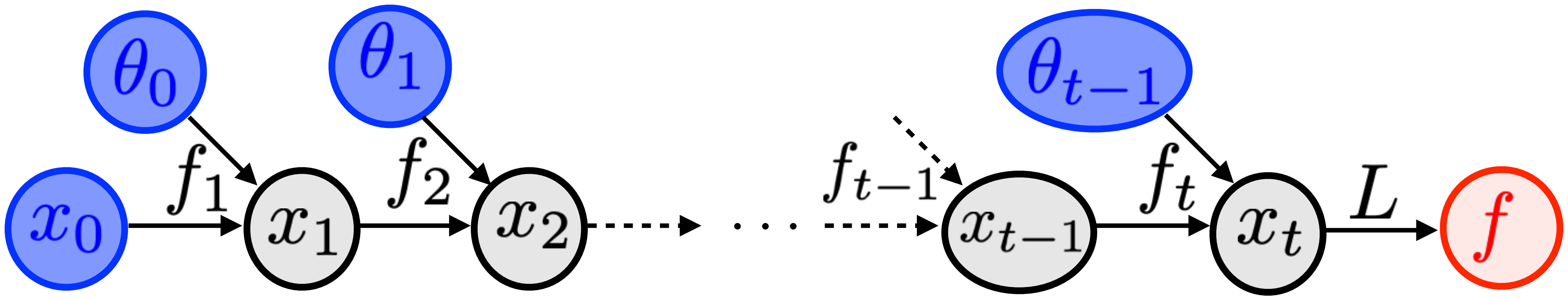
$$f(\theta) \overset{\text{def.}}{=} L(x_t)$$

Example: perceptrons

$$f_k(x_{k-1}, \theta_{k-1}) = \rho(\theta_{k-1} x_{k-1})$$



Feedforward Architecture



forward

$$\texttt{for } k = 1, \dots, t-1, t$$

$$x_k = f_k(x_{k-1}, \theta_{k-1})$$

$$f(\theta) \stackrel{\text{def.}}{=} L(x_t)$$

backward

$$\nabla_{x_t} f = \nabla L(x_t)$$

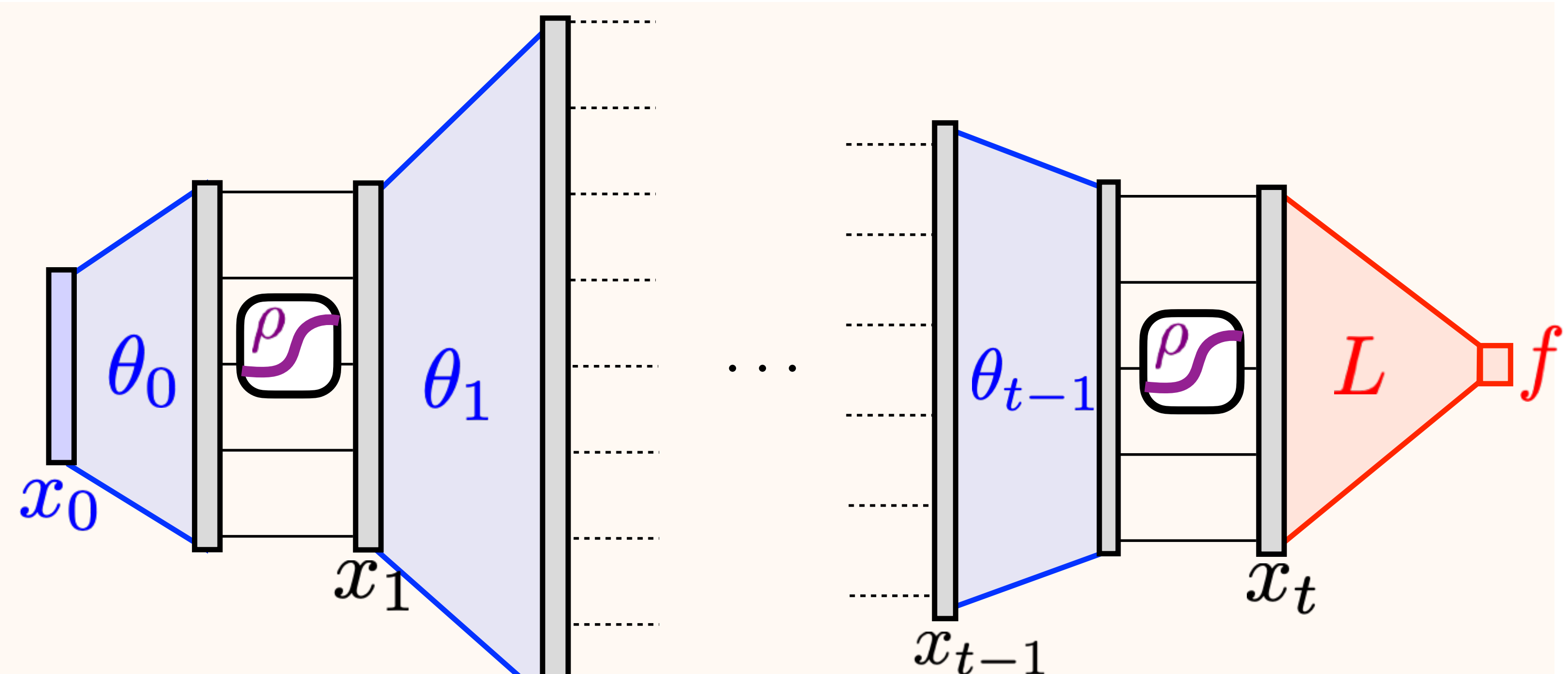
$$\texttt{for } k = t, t-1, \dots, 1$$

$$\nabla_{x_{k-1}} f = [\partial_x f_k(x_{k-1}, \theta_{k-1})]^\top \nabla_{x_k} f$$

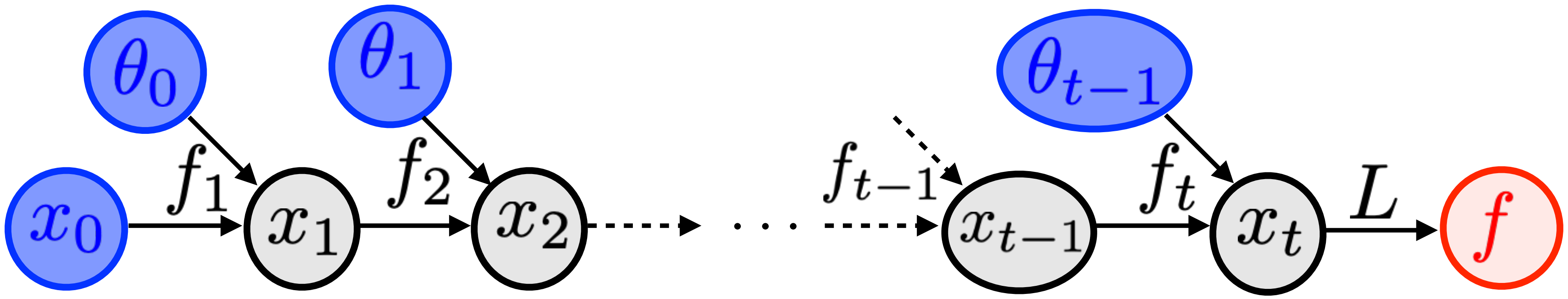
$$\nabla_{\theta_{k-1}} f = [\partial_\theta f_k(x_{k-1}, \theta_{k-1})]^\top (\nabla_{x_k} f)$$

Example: perceptrons

$$f_k(x_{k-1}, \theta_{k-1}) = \rho(\theta_{k-1} x_{k-1})$$



Feedforward Architecture



forward

$$\texttt{for } k = 1, \ldots, t-1, t$$
$$x_k = f_k(x_{k-1}, \theta_{k-1})$$
$$f(\theta) \overset{\text{def.}}{=} L(x_t)$$

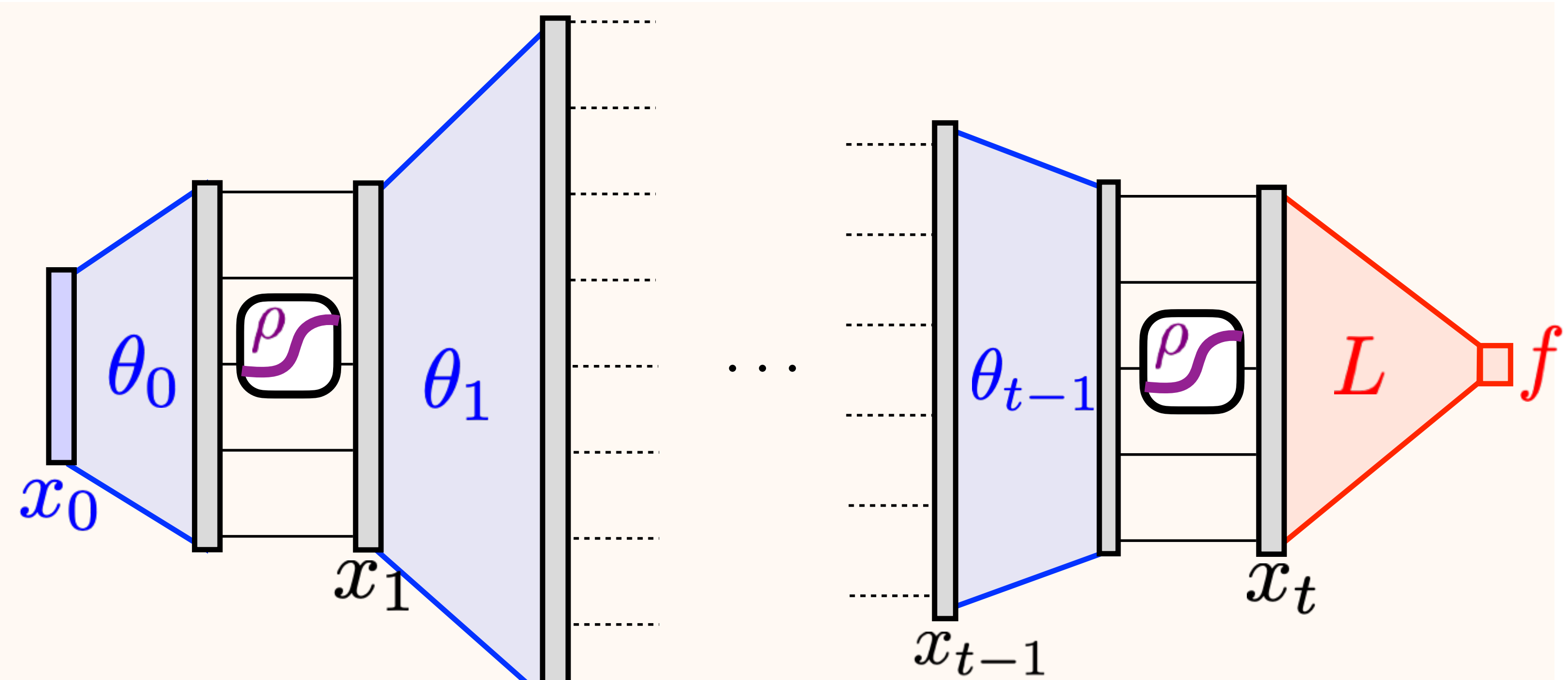
backward

$$\nabla_{x_t} f = \nabla L(x_t)$$
$$\texttt{for } k = t, t-1, \ldots, 1$$
$$\nabla_{x_{k-1}} f = [\partial_x f_k(x_{k-1}, \theta_{k-1})]^\top \nabla_{x_k} f$$
$$\nabla_{\theta_{k-1}} f = [\partial_\theta f_k(x_{k-1}, \theta_{k-1})]^\top (\nabla_{x_k} f)$$

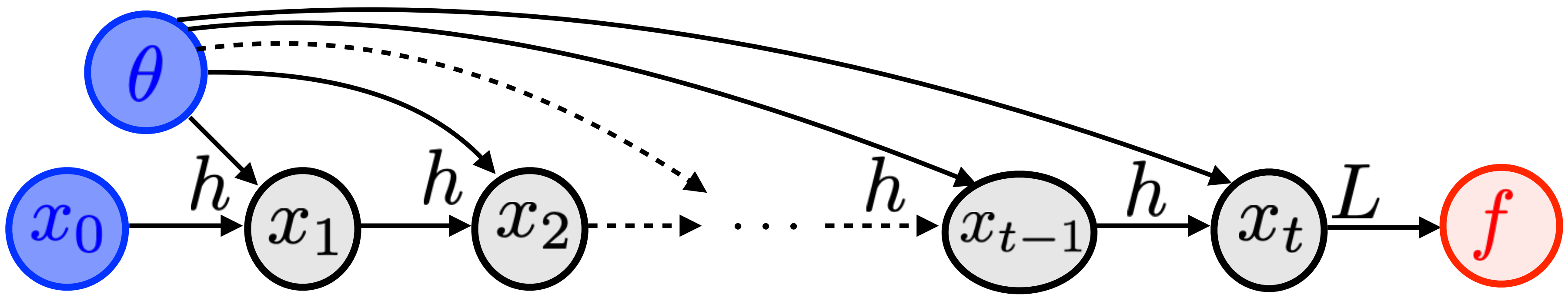
Example: perceptrons

$$f_k(x_{k-1}, \theta_{k-1}) = \rho(\theta_{k-1} x_{k-1})$$

$$[\partial_\theta f_k(x, \theta)]^\top z = w x^\top$$
$$[\partial_x f_k(x, \theta)]^\top z = \theta^\top (w \odot z)$$
$$z \overset{\text{def.}}{=} \rho'(\theta x)$$



Recurrent Architecture



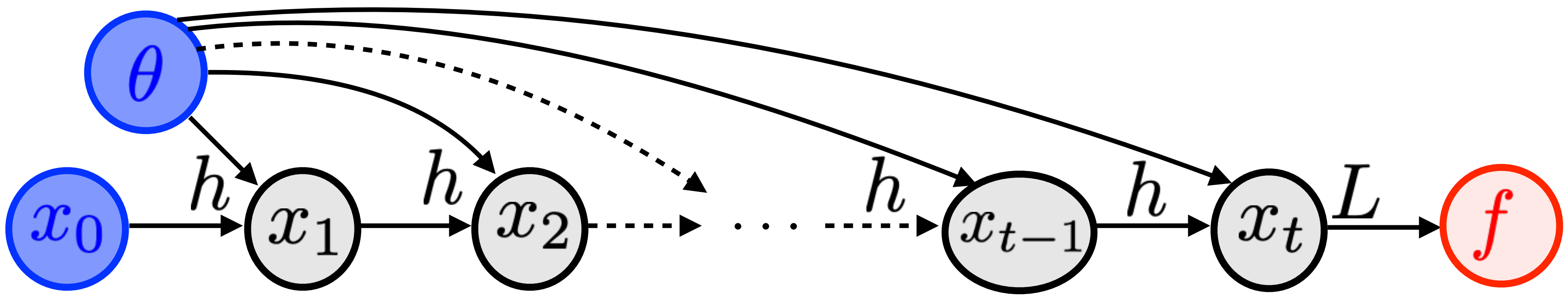
forward

$$\texttt{for } k = 1, \ldots, t-1, t$$

$$x_k = h(x_{k-1}, \theta)$$

$$f(\theta) = L(x_t)$$

Recurrent Architecture

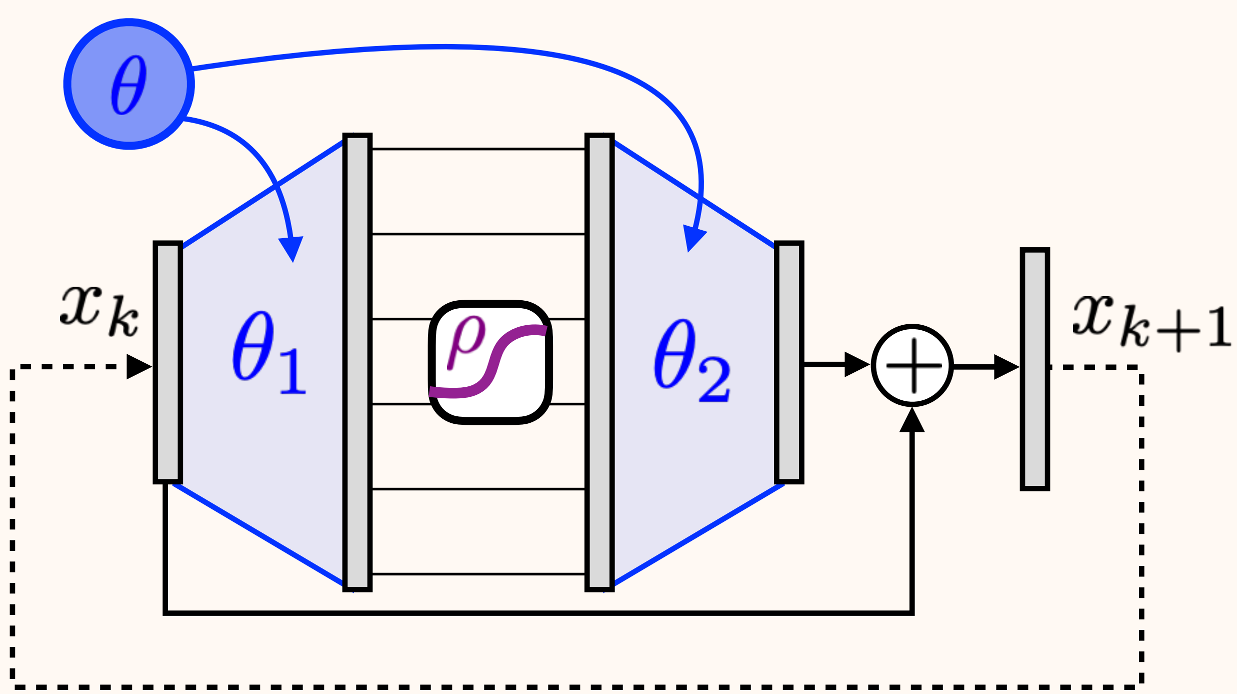


forward

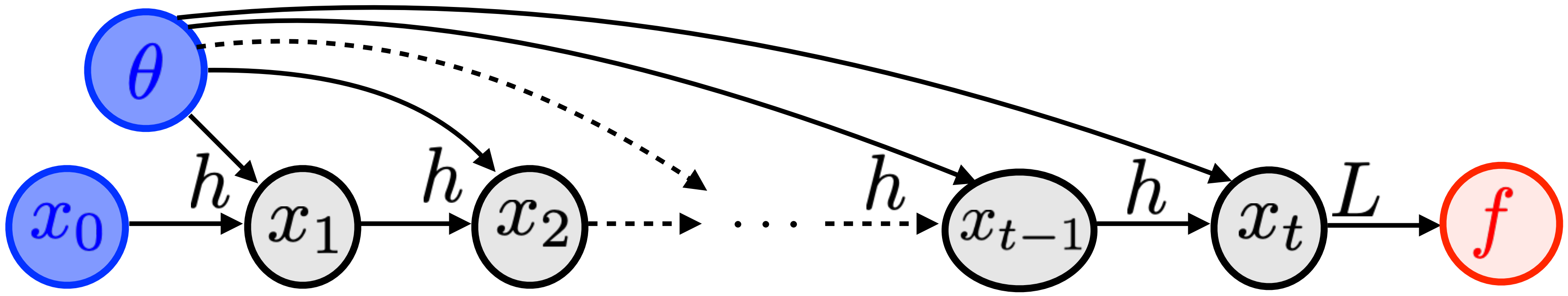
$$\texttt{for } k = 1, \ldots, t-1, t$$
$$x_k = h(x_{k-1}, \theta)$$
$$f(\theta) = L(x_t)$$

Example: residual networks

$$h(x, \theta) = x + \theta_2 \rho(\theta_1 x)$$



Recurrent Architecture



forward

for $k = 1, \dots, t-1, t$

$$x_k = h(x_{k-1}, \theta)$$

$$f(\theta) = L(x_t)$$

backward

$$\nabla_{x_t} f = \nabla L(x_t)$$

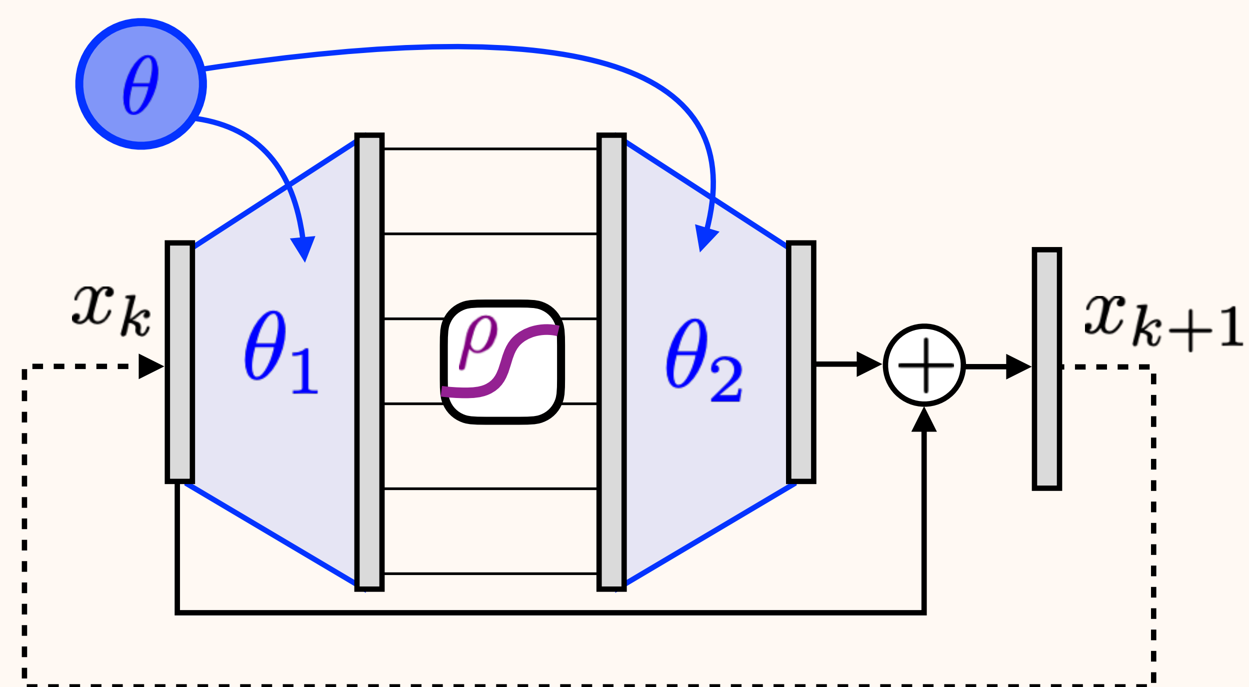
for $k = t, t-1, \dots, 1$

$$\nabla_{x_{k-1}} f = [\partial_x h(x_{k-1}, \theta)]^\top \nabla_{x_k} f$$

$$\nabla_\theta f = \sum_k [\partial_\theta (x_{k-1}, \theta)]^\top \nabla_{x_k} f$$

Example: residual networks

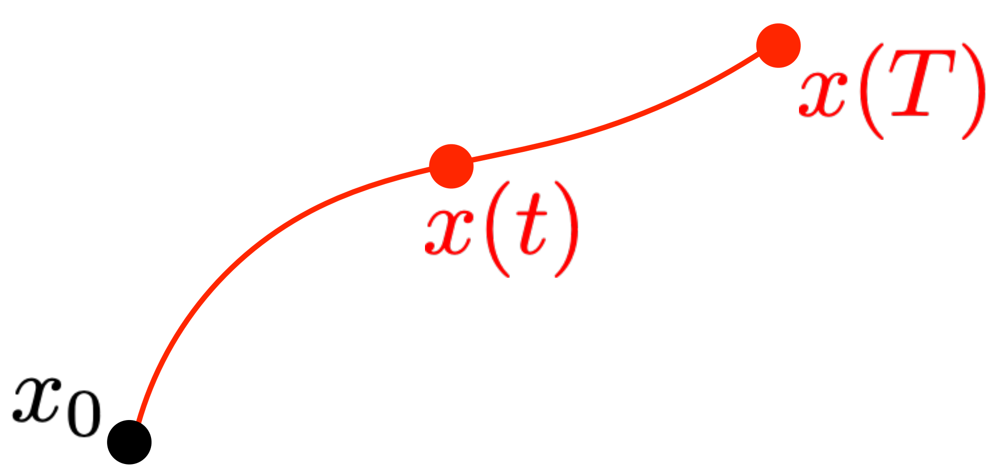
$$h(x, \theta) = x + \theta_2 \rho(\theta_1 x)$$



Adjoint State Method

Optimal control:

$$\dot{x}(t) = u(x(t), \theta)$$
$$f(\theta) = L(x(T))$$



Adjoint State Method

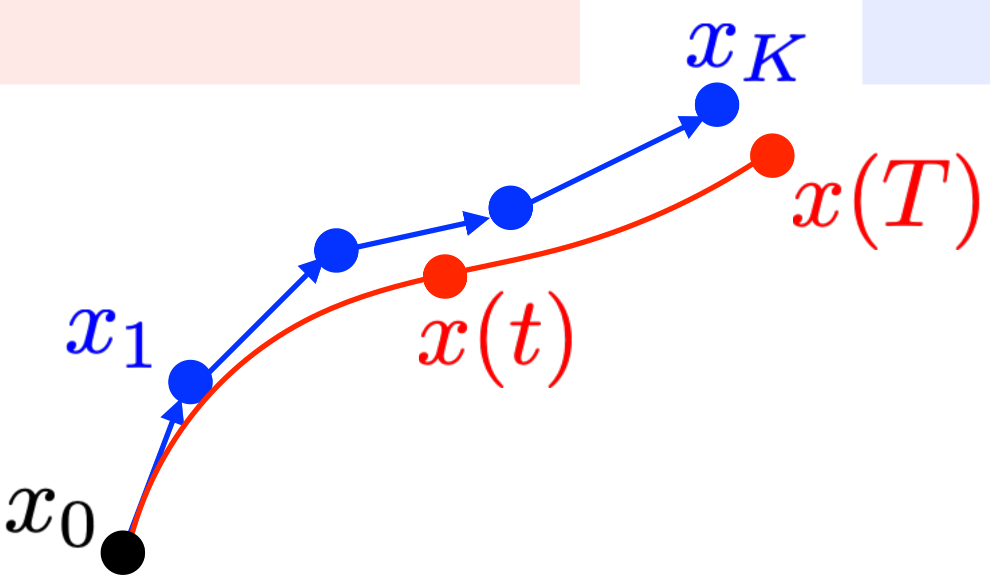
Optimal control:

$$\dot{x}(t) = u(x(t), \theta)$$
$$f(\theta) = L(x(T))$$

$t = \tau k$ →

Discretization:

$$x_{k+1} = x_k + \tau u(x_k, \theta)$$
$$f(\theta) = L(x_K)$$



Adjoint State Method

Optimal control:

$$\dot{x}(t) = u(x(t), \theta)$$
$$f(\theta) = L(x(T))$$

$t = \tau k$

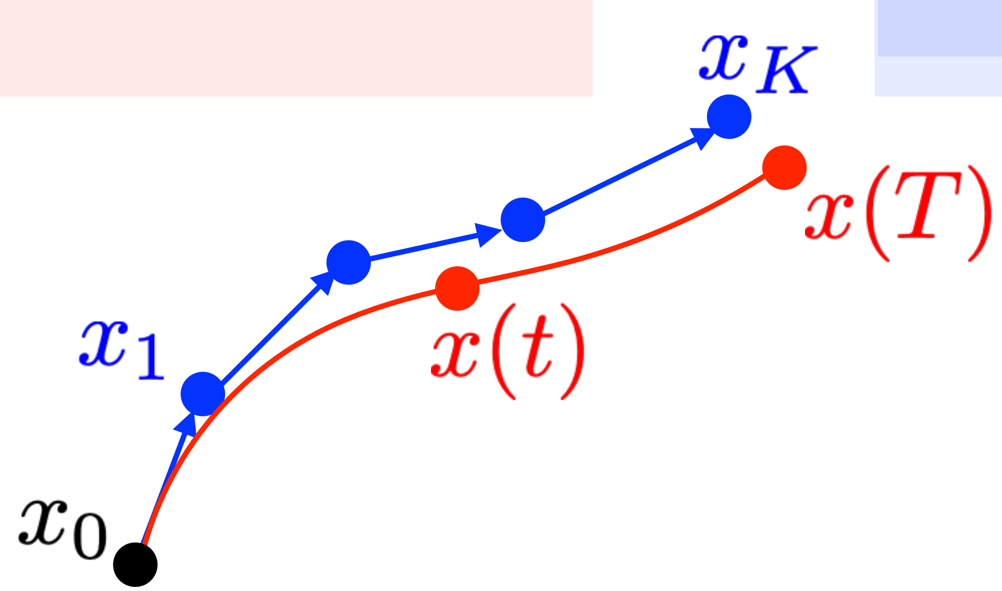
Discretization:

$$x_{k+1} = x_k + \tau u(x_k, \theta)$$
$$f(\theta) = L(x_K)$$

$$z_k \stackrel{\text{def.}}{=} \nabla_{x_k} f(\theta)$$

$$z_{k-1} = z_k + \tau [\partial u(x_k, \theta)]^\top z_k$$

$$\nabla_\theta f(\theta) = \sum_k [\partial_\theta h(x_{k-1}, \theta)]^\top z_k$$



Adjoint State Method

Optimal control:

$$\dot{x}(t) = u(x(t), \theta)$$
$$f(\theta) = L(x(T))$$

$$z(t) \overset{\text{def.}}{=} \nabla_{x(t)} f(\theta)$$

$$\dot{z}(t) = -[\partial_x u(x(t), \theta)]^\top z(t)$$

$$\nabla_\theta f(\theta) = \int_0^T [\partial_\theta f(x(t), \theta)]^\top z(t) \mathrm{d}t$$

$t = \tau k$ →

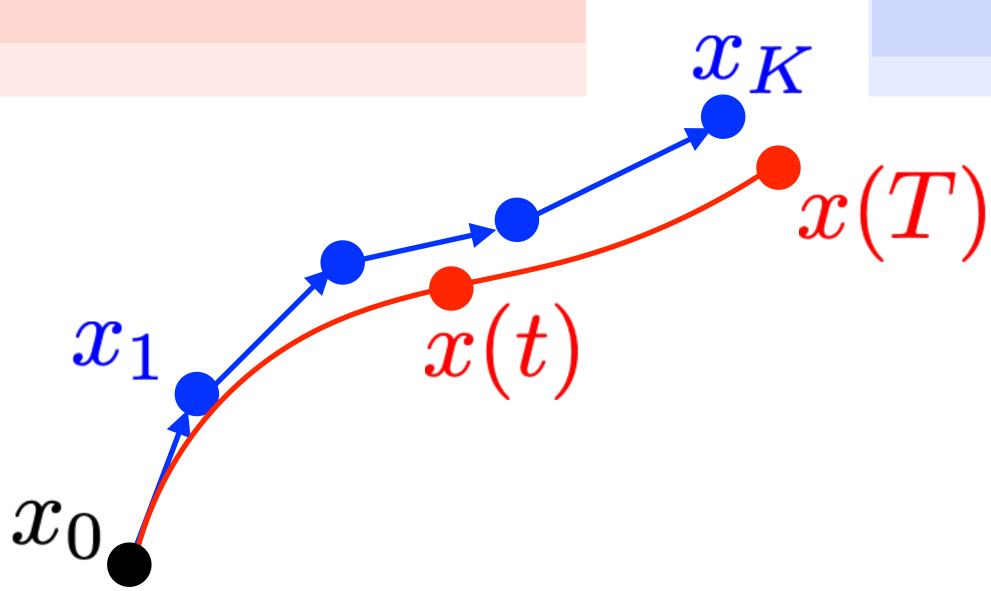
Discretization:

$$x_{k+1} = x_k + \tau u(x_k, \theta)$$
$$f(\theta) = L(x_K)$$

$$z_k \overset{\text{def.}}{=} \nabla_{x_k} f(\theta)$$

$$z_{k-1} = z_k + \tau[\partial u(x_k, \theta)]^\top z_k$$

$$\nabla_\theta f(\theta) = \sum_k [\partial_\theta h(x_{k-1}, \theta)]^\top z_k$$



Conservative Systems: Invertible Architectures

Curse of auto-diff: memory grows with #iterations K.

Conservative Systems: Invertible Architectures

Curse of auto-diff: memory grows with #iterations K.

Generic method: checkpointing.

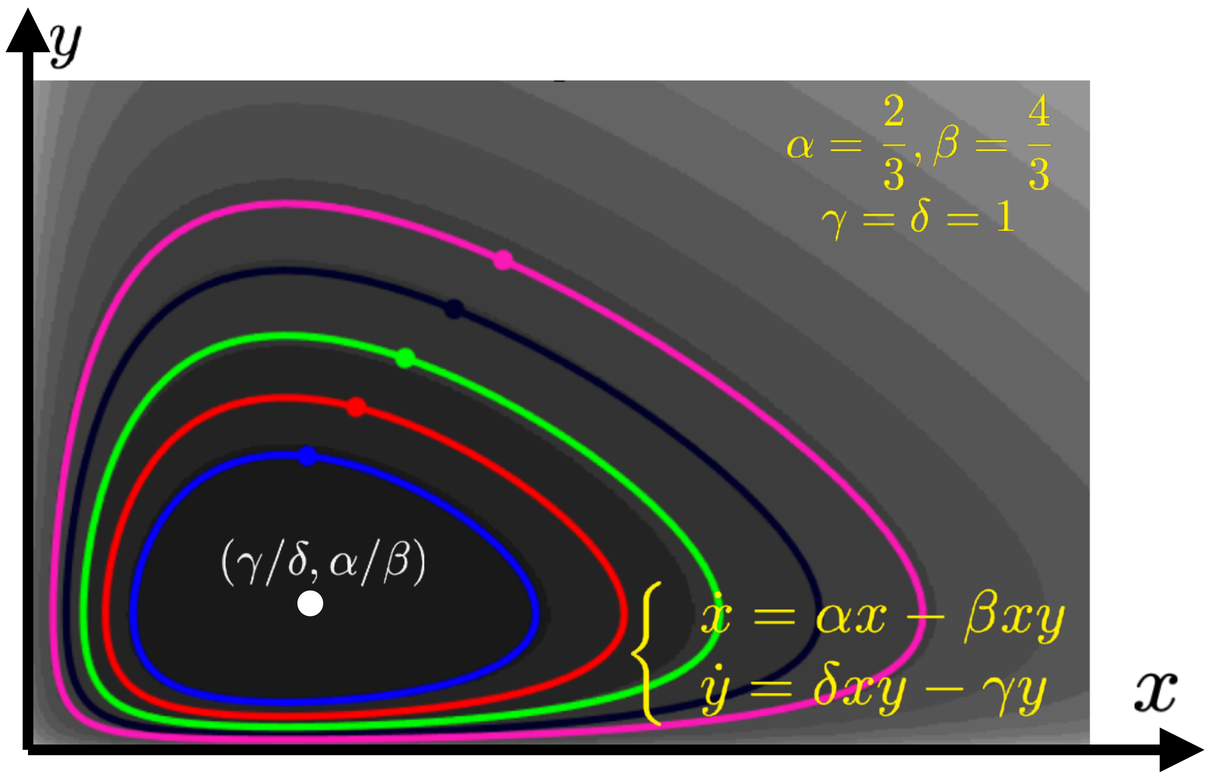
Conservative Systems: Invertible Architectures

Curse of auto-diff: memory grows with #iterations K.

Generic method: checkpointing.

Alternative: build "invertible" architectures.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} u(y) \\ v(x) \end{pmatrix}$$



Conservative Systems: Invertible Architectures

Curse of auto-diff: memory grows with #iterations K.

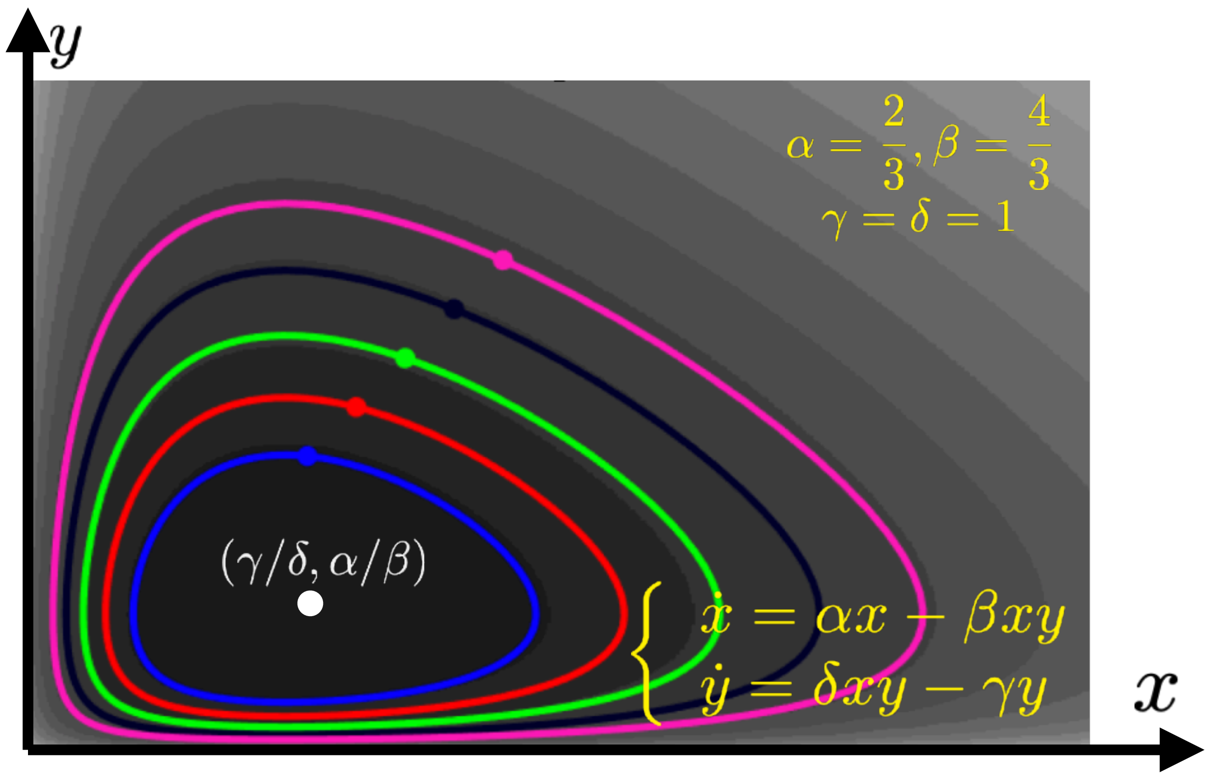
Generic method: checkpointing.

Alternative: build "invertible" architectures.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} u(y) \\ v(x) \end{pmatrix}$$

↓ inverse

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = - \begin{pmatrix} u(y) \\ v(x) \end{pmatrix}$$

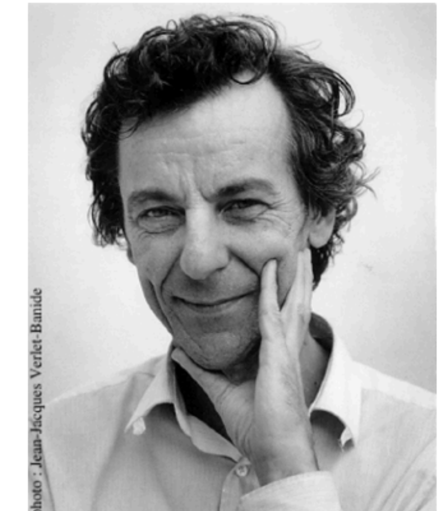


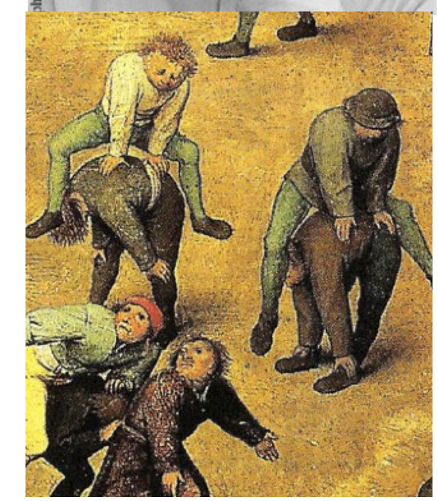
Conservative Systems: Invertible Architectures

Curse of auto-diff: memory grows with #iterations K.

Generic method: checkpointing.

Alternative: build "invertible" architectures.





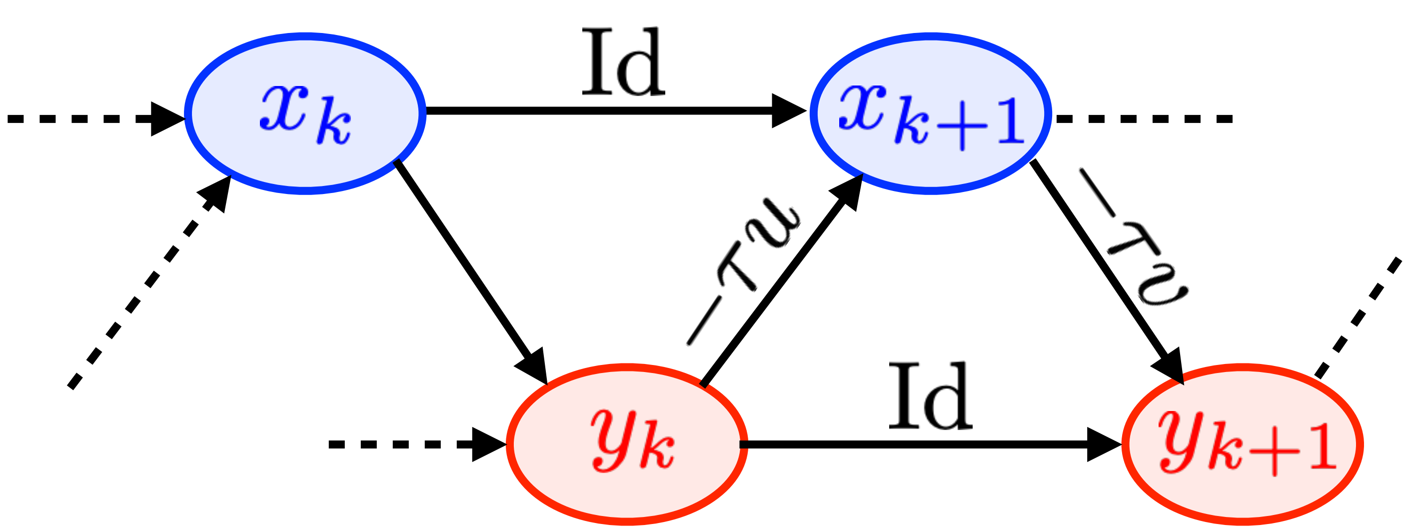
$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{pmatrix}u(y)\\ v(x)\end{pmatrix}$$

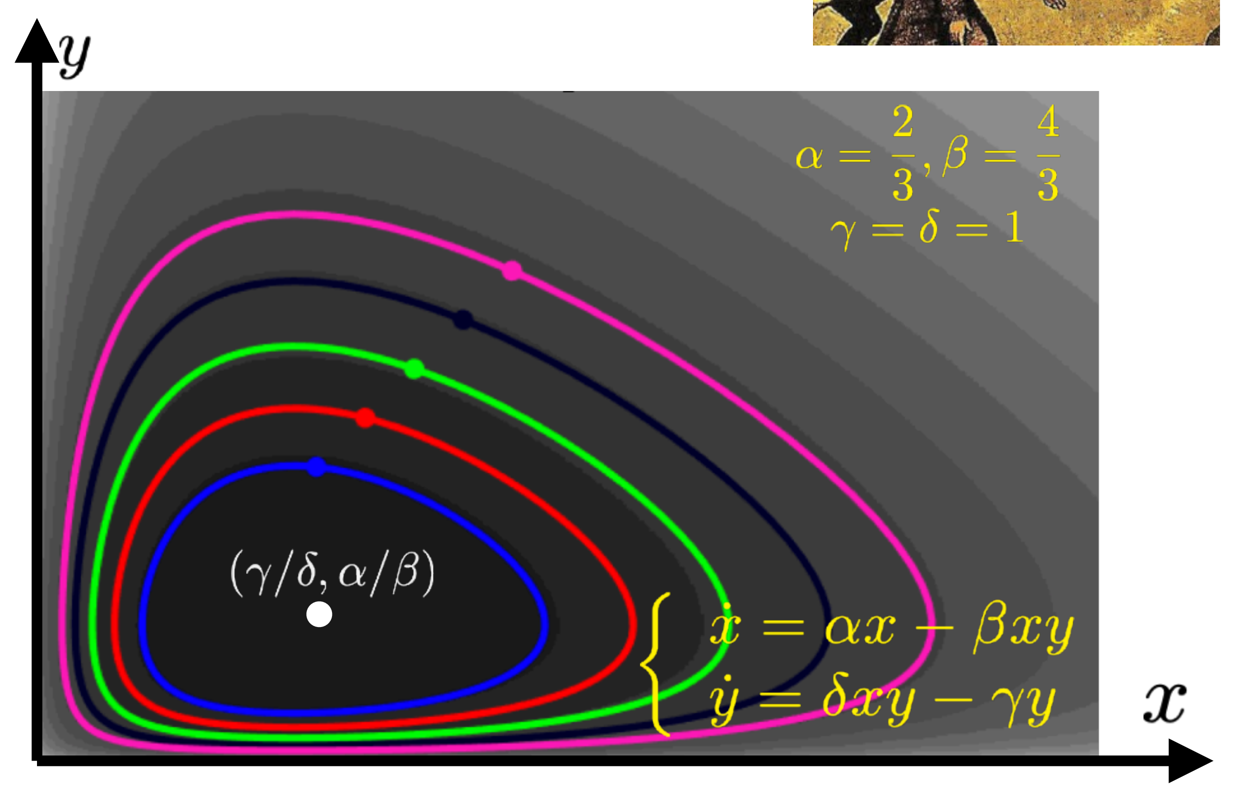
leapfrog → Verlet

$$\begin{cases} x_{k+1} = x_k + \tau u(y_k) \\ y_{k+1} = y_k + \tau v(x_{k+1}) \end{cases}$$

↓ inverse

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = -\begin{pmatrix}u(y)\\ v(x)\end{pmatrix}$$



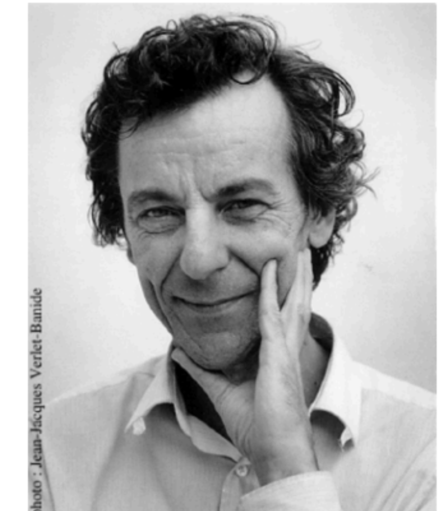


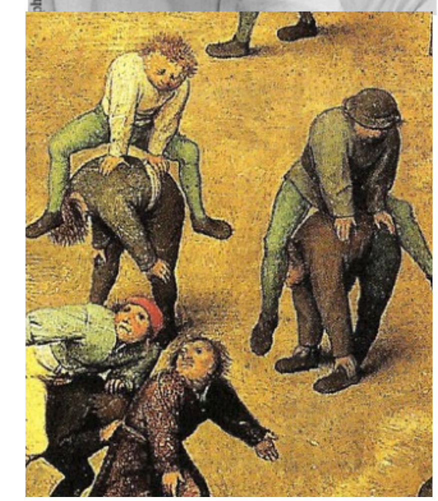
Conservative Systems: Invertible Architectures

Curse of auto-diff: memory grows with #iterations K.

Generic method: checkpointing.

Alternative: build "invertible" architectures.





$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{pmatrix}u(y)\\ v(x)\end{pmatrix}$$

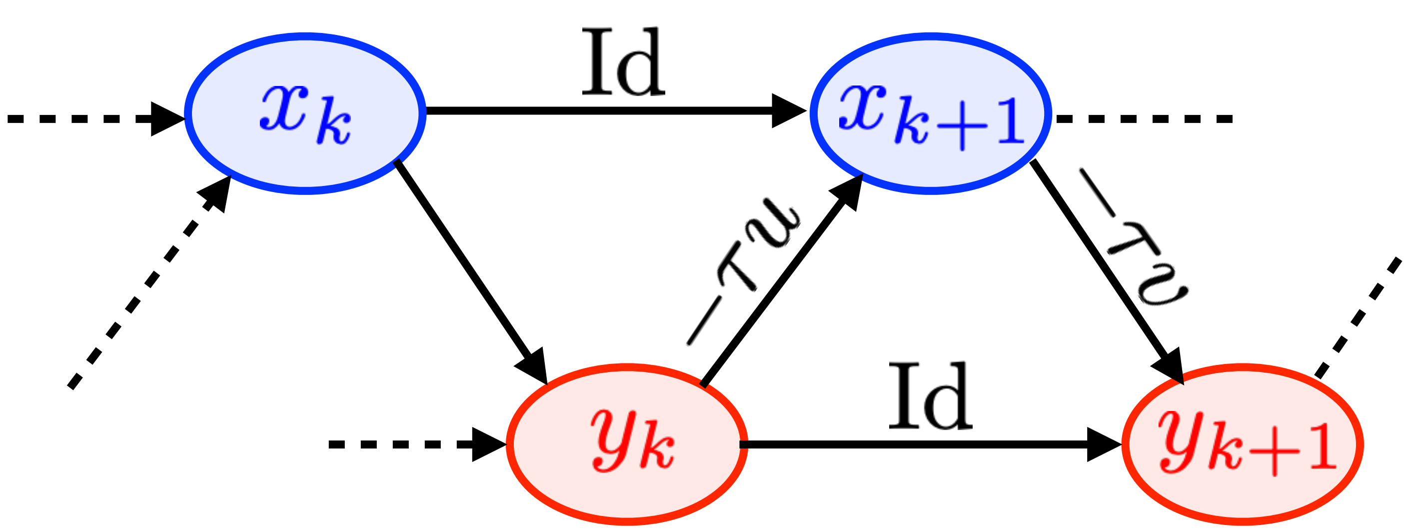
$\xrightarrow{\text{leapfrog Verlet}}$

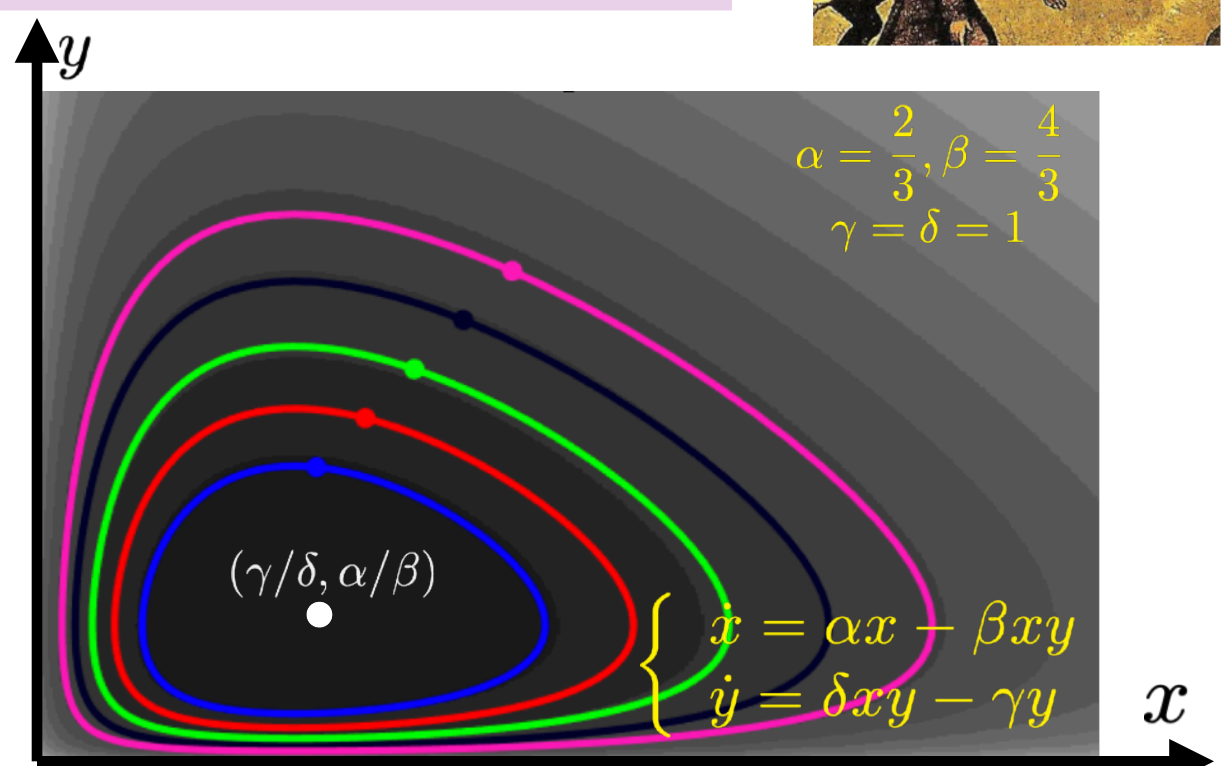
$$\begin{cases} x_{k+1} = x_k + \tau u(y_k) \\ y_{k+1} = y_k + \tau v(x_{k+1}) \end{cases}$$

inverse $\downarrow$ (left); inverse $\downarrow$ (right)

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = -\begin{pmatrix}u(y)\\ v(x)\end{pmatrix}$$

$$\begin{cases} y_k = y_{k+1} - \tau v(x_{k+1}) \\ x_k = x_{k+1} - \tau u(y_k) \end{cases}$$



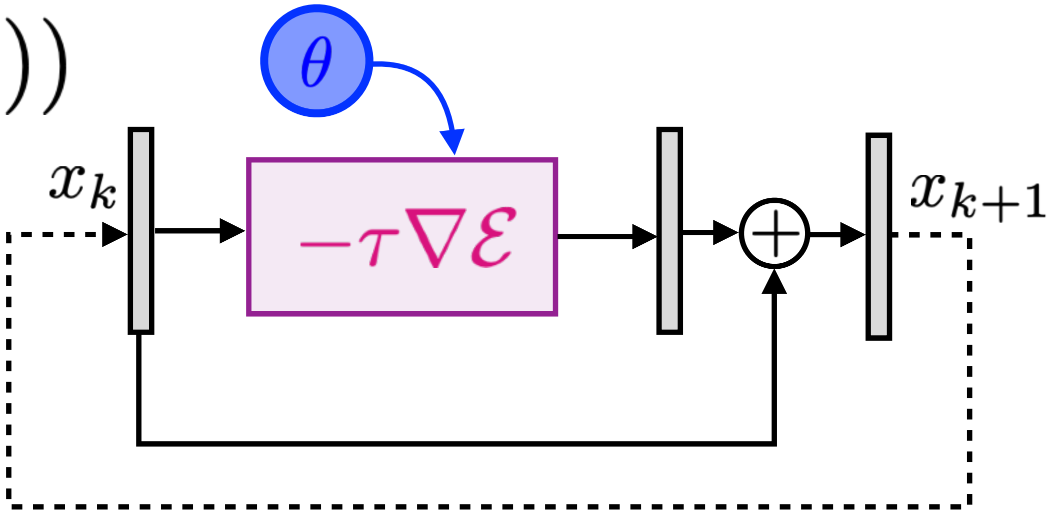


Dissipative Systems: Argmin Layers

$$x(\theta) \stackrel{\text{def.}}{=} \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \, \mathcal{E}(x, \theta) \qquad f(\theta) \stackrel{\text{def.}}{=} L(x(\theta))$$

Dissipative Systems: Argmin Layers

$$x(\theta) \overset{\text{def.}}{=} \underset{x \in \mathbb{R}^n}{\text{argmin}}\ \mathcal{E}(x, \theta) \qquad f(\theta) \overset{\text{def.}}{=} L(x(\theta))$$

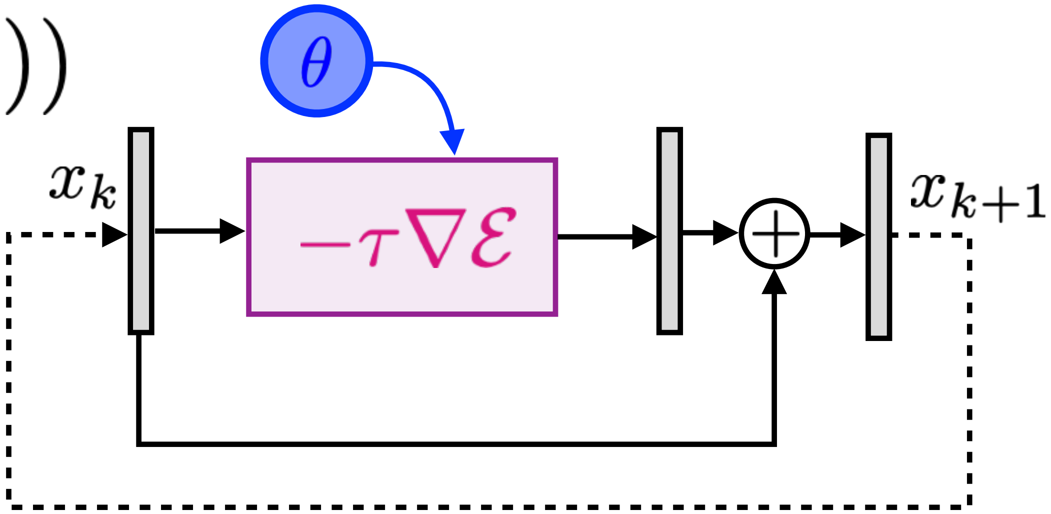


$$x_{k+1} = x_k - \tau\nabla\mathcal{E}(x_k, \theta) \iff \text{ResNet}$$

$\rightarrow$ Memory exploses with #iterations.

Dissipative Systems: Argmin Layers

$$x(\theta) \overset{\text{def.}}{=} \underset{x \in \mathbb{R}^n}{\text{argmin}}\, \mathcal{E}(x, \theta) \qquad f(\theta) \overset{\text{def.}}{=} L(x(\theta))$$

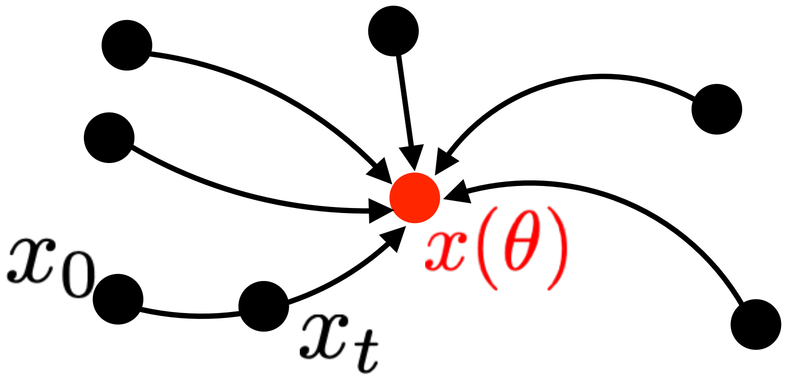


$$x_{k+1} = x_k - \tau \nabla \mathcal{E}(x_k, \theta) \;\Leftrightarrow\; \text{ResNet}$$

$\rightarrow$ Memory exploses with #iterations.

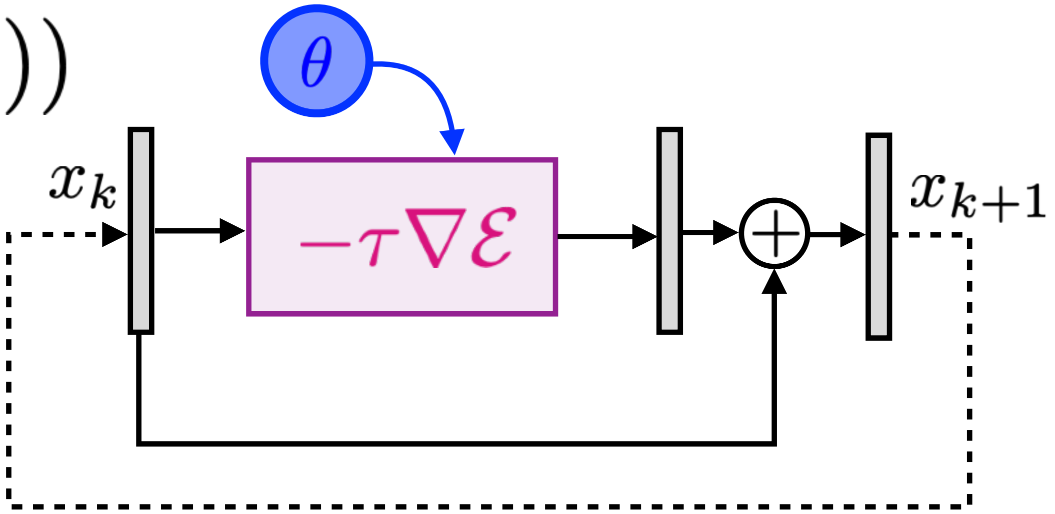
$$\dot{x}_t = -\nabla \mathcal{E}(x_t, \theta)$$

$\rightarrow$ Flow is non-conservative, $t \mapsto x_t$ ill-posed.



Dissipative Systems: Argmin Layers

$$x(\theta) \overset{\text{def.}}{=} \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\, \mathcal{E}(x, \theta) \qquad f(\theta) \overset{\text{def.}}{=} L(x(\theta))$$

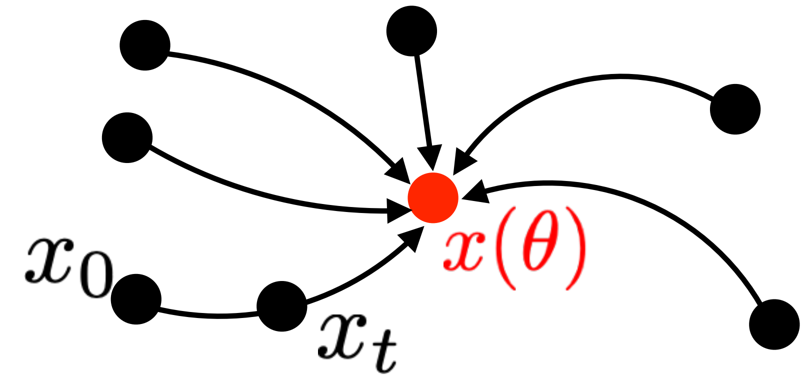


$$x_{k+1} = x_k - \tau \nabla \mathcal{E}(x_k, \theta) \quad \Leftrightarrow \quad \text{ResNet}$$

$\rightarrow$ Memory exploses with #iterations.

$$\dot{x}_t = -\nabla \mathcal{E}(x_t, \theta)$$

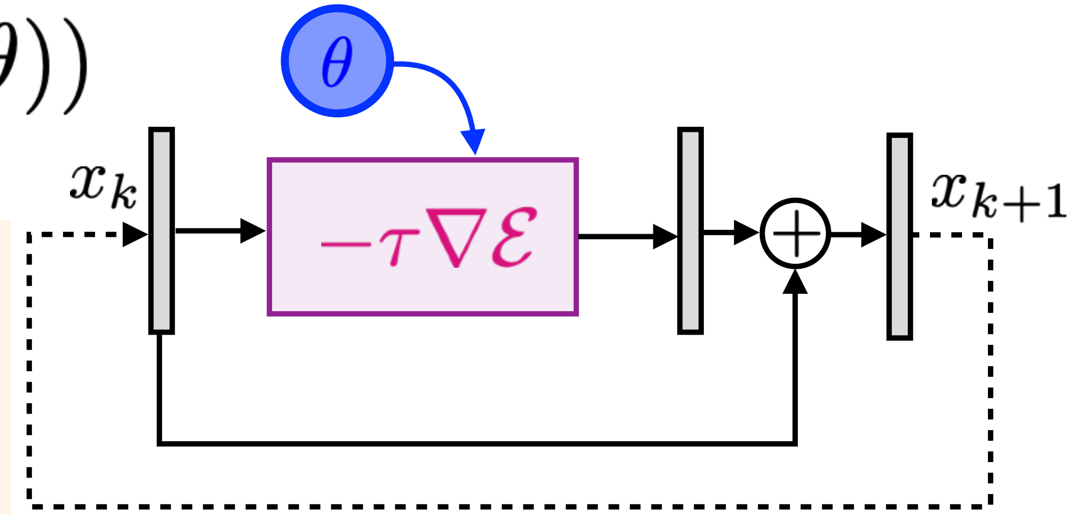
$\rightarrow$ Flow is non-conservative, $t \mapsto x_t$ ill-posed.



Fixed point equation: $\quad \nabla_x \mathcal{E}(x(\theta), \theta) = 0$

Dissipative Systems: Argmin Layers

$$x(\theta) \overset{\text{def.}}{=} \underset{x \in \mathbb{R}^n}{\text{argmin}}\, \mathcal{E}(x,\theta) \qquad f(\theta) \overset{\text{def.}}{=} L(x(\theta))$$

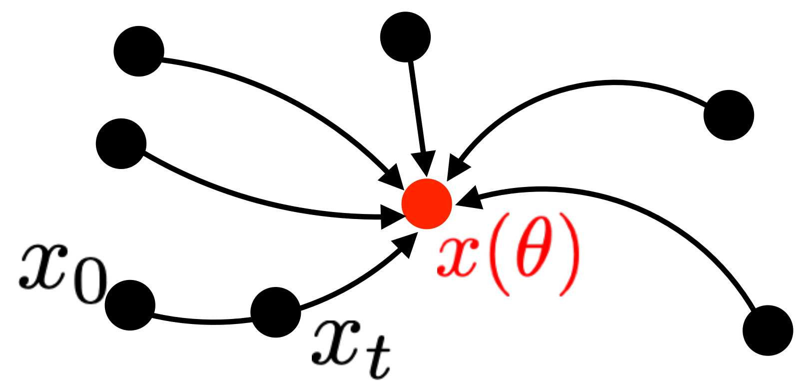


$$x_{k+1} = x_k - \tau \nabla \mathcal{E}(x_k, \theta) \iff \text{ResNet}$$

$\to$ Memory exploses with #iterations.

$$\dot{x}_t = -\nabla \mathcal{E}(x_t, \theta)$$

$\to$ Flow is non-conservative, $t \mapsto x_t$ ill-posed.



Fixed point equation: $\nabla_x \mathcal{E}(x(\theta), \theta) = 0$

Implicit function theorem:

$$\nabla f(\theta) = -\left(\frac{\partial^2 \mathcal{E}}{\partial x \partial \theta}(x(\theta),\theta)\right)^\top \left(\frac{\partial^2 \mathcal{E}}{\partial^2 x}(x(\theta),\theta)\right)^{-1} \nabla L(x(\theta))$$

$n \times n$
linear system

Example: Sinkhorn

Entropic optimal transport: between (θ_1, θ_2), $K \overset{\text{def.}}{=} e^{-\frac{c}{\varepsilon}}$

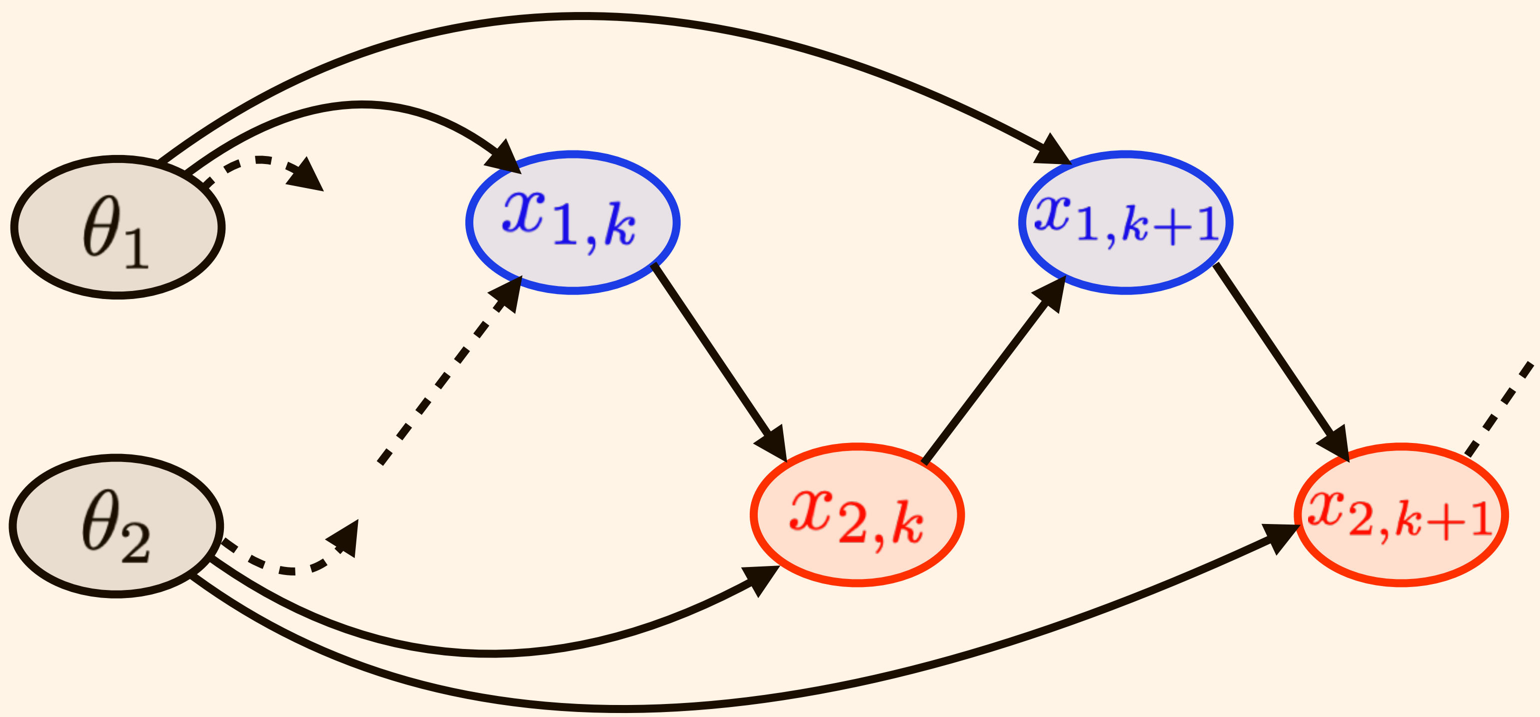
$$x(\theta) \overset{\text{def.}}{=} \underset{x}{\operatorname{argmin}}\, \mathcal{E}(x, \theta) = -\langle \theta_1, \log(x_1) \rangle - \langle \theta_2, \log(x_2) \rangle + \langle K x_1, x_2 \rangle$$

Example: Sinkhorn

Entropic optimal transport: between (θ_1, θ_2), $K \overset{\text{def.}}{=} e^{-\frac{c}{\varepsilon}}$

$$x(\theta) \overset{\text{def.}}{=} \underset{x}{\text{argmin}}\ \mathcal{E}(x,\theta) = -\langle \theta_1, \log(x_1) \rangle - \langle \theta_2, \log(x_2) \rangle + \langle K x_1, x_2 \rangle$$

Sinkhorn: $x_{1,k+1} = \dfrac{\theta_1}{K x_{2,k}}$ $\quad x_{2,k+1} = \dfrac{\theta_2}{K^\top x_{1,k+1}}$

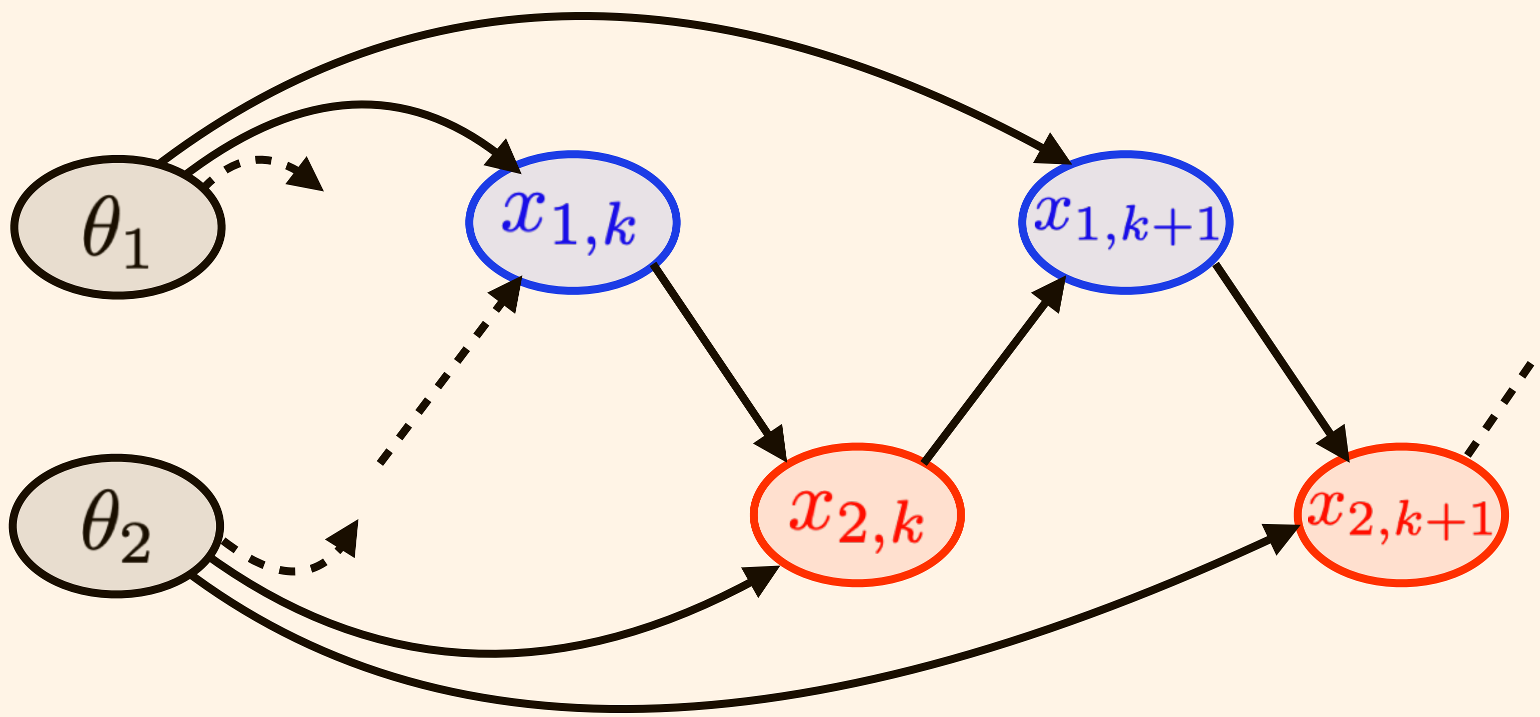


Example: Sinkhorn

Entropic optimal transport: between (θ_1, θ_2), $K \overset{\text{def.}}{=} e^{-\frac{c}{\varepsilon}}$

$$x(\theta) \overset{\text{def.}}{=} \underset{x}{\operatorname{argmin}} \, \mathcal{E}(x, \theta) = -\langle \theta_1, \log(x_1) \rangle - \langle \theta_2, \log(x_2) \rangle + \langle K x_1, x_2 \rangle$$

Sinkhorn: $\quad x_{1,k+1} = \dfrac{\theta_1}{K x_{2,k}} \qquad x_{2,k+1} = \dfrac{\theta_2}{K^\top x_{1,k+1}}$



Computing $[\partial x(\theta)]^\top$:
- back-propagation through Sinkhorn.
- Hessian inversion (implicit function)

Take Home Messages

- is not just formal or numerical calculus ;
- is not just the chain rule ;
- is not just the adjoint state method ;
- is not just backpropagation ;

Take Home Messages

- is not just formal or numerical calculus ;
- is not just the chain rule ;
- is not just the adjoint state method ;
- is not just backpropagation ;
- is time efficient ;
- is memory inefficient … but this can be mitigated:
  - Checkpointing,
  - Implicit function theorem,
  - Reversing the flow.