Bundles, Lenses & Machine Learning

Jules Hedges[1]
joint work with
Brendan Fong[2] Eliana Lorch[3] David Spivak[2]

[1]Max Planck Institute for Mathematics in the Sciences

[2]MIT

[3]University of Oxford

SYCO 5, Birmingham

Bundles, Lenses & Machine Learning

Jules Hedges[1]
joint work with
Brendan Fong[2] Eliana Lorch[3] David Spivak[2]

[1]Max Planck Institute for Mathematics in the Sciences

[2]MIT

[3]University of Oxford

SYCO 5, Birmingham

Featuring zero string diagrams :(

Motivation

Machine learning is categorical in 2 different ways:

Backprop As Functor
(compositional description of ML with monoidal categories)
+
ML as differential geometry

Motivation

Machine learning is categorical in 2 different ways:

Backprop As Functor
(compositional description of ML with monoidal categories)
+
ML as differential geometry

In this talk: smoosh them together

(why? Why not)

Motivation

Machine learning is categorical in 2 different ways:

Backprop As Functor
(compositional description of ML with monoidal categories)
+
ML as differential geometry

In this talk: smoosh them together

(why? Why not)

It clarifies Backprop as Functor more than anything else

Open learners

Definition (Fong, Spivak & Tuyéras) : An open learner $X \to Y$ consists of:

- A set P of parameters

Open learners

Definition (Fong, Spivak & Tuyéras) : An open learner $X \to Y$ consists of:

- A set P of parameters
- A function $I : P \times X \to Y$ (the implementation)

Open learners

Definition (Fong, Spivak & Tuyéras) : An open learner $X \to Y$ consists of:

- A set P of parameters
- A function $I : P \times X \to Y$ (the implementation)
- A function $u : P \times X \times Y \to P$ (the update)

Open learners

Definition (Fong, Spivak & Tuyéras) : An open learner $X \to Y$ consists of:

- A set P of parameters
- A function $I : P \times X \to Y$ (the implementation)
- A function $u : P \times X \times Y \to P$ (the update)
- A function $r : P \times X \times Y \to X$ (the request)

Open learners

Definition (Fong, Spivak & Tuyéras) : An open learner $X \to Y$ consists of:

- A set P of parameters
- A function $I : P \times X \to Y$ (the implementation)
- A function $u : P \times X \times Y \to P$ (the update)
- A function $r : P \times X \times Y \to X$ (the request)

Composition of open learners is fiddly

Open learners

Definition (Fong, Spivak & Tuyéras) : An open learner $X \to Y$ consists of:

- A set P of parameters
- A function $I : P \times X \to Y$ (the implementation)
- A function $u : P \times X \times Y \to P$ (the update)
- A function $r : P \times X \times Y \to X$ (the request)

Composition of open learners is fiddly

They form a symmetric monoidal category called **Learn**

who cares about monoidal bicategories

Lenses

A lens $X \to Y$ is a function $X \to Y$ and a function $X \times Y \to X$

Lenses

A lens $X \to Y$ is a function $X \to Y$ and a function $X \times Y \to X$

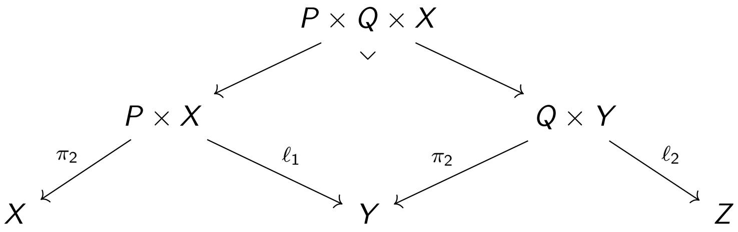
Composition of lenses is also fiddly!

Lenses

A lens $X \to Y$ is a function $X \to Y$ and a function $X \times Y \to X$

Composition of lenses is also fiddly!

Theorem (Fong & Johnson): Open learners compose by pullback of lenses:



The **Para** construction

Let $\mathcal{C}$ be a monoidal category

Define a category[1] **Para**($\mathcal{C}$) by:

[1] who cares about monoidal bicategories

The **Para** construction

Let $\mathcal{C}$ be a monoidal category

Define a category[1] $\mathbf{Para}(\mathcal{C})$ by:

- Objects: objects of $\mathcal{C}$
- Morphisms $X \to Y$: pair (A, f), A object of $\mathcal{C}$, $f : X \otimes A \to Y$

[1] who cares about monoidal bicategories

The **Para** construction

Let $\mathcal{C}$ be a monoidal category

Define a category[1] $\mathbf{Para}(\mathcal{C})$ by:

- Objects: objects of $\mathcal{C}$
- Morphisms $X \to Y$: pair (A, f), A object of $\mathcal{C}$, $f : X \otimes A \to Y$
- Identity on X: $(I, X \otimes I \xrightarrow{\cong} X)$

[1] who cares about monoidal bicategories

The **Para** construction

Let $\mathcal{C}$ be a monoidal category

Define a category[1] $\mathbf{Para}(\mathcal{C})$ by:

- Objects: objects of $\mathcal{C}$
- Morphisms $X \to Y$: pair (A, f), A object of $\mathcal{C}$, $f : X \otimes A \to Y$
- Identity on X: $(I, X \otimes I \xrightarrow{\cong} X)$
- Composition of $(B, g) \circ (A, f)$:

$$(A \otimes B, X \otimes A \otimes B \xrightarrow{f \otimes B} Y \otimes B \xrightarrow{g} Z)$$

[1] who cares about monoidal bicategories

The **Para** construction

Let $\mathcal{C}$ be a monoidal category

Define a category[1] **Para**($\mathcal{C}$) by:

- Objects: objects of $\mathcal{C}$
- Morphisms $X \to Y$: pair (A, f), A object of $\mathcal{C}$, $f : X \otimes A \to Y$
- Identity on X: $(I, X \otimes I \xrightarrow{\cong} X)$
- Composition of $(B, g) \circ (A, f)$:

$$(A \otimes B, X \otimes A \otimes B \xrightarrow{f \otimes B} Y \otimes B \xrightarrow{g} Z)$$

$\otimes$ lifts to a monoidal product on **Para**($\mathcal{C}$)

[1] who cares about monoidal bicategories

The structure of $\mathbf{Para}(-)$

A lax symmetric monoidal functor $F : \mathcal{C} \to \mathcal{D}$ lifts to

$$\mathbf{Para}(F) : \mathbf{Para}(\mathcal{D}) \to \mathbf{Para}(\mathcal{D})$$

by

$$F(A, f) : F(X) \otimes F(A) \xrightarrow{\varphi} F(X \otimes A) \xrightarrow{F(f)} F(Y)$$

The structure of **Para**$(-)$

A lax symmetric monoidal functor $F : \mathcal{C} \to \mathcal{D}$ lifts to

$$\mathbf{Para}(F) : \mathbf{Para}(\mathcal{D}) \to \mathbf{Para}(\mathcal{D})$$

by

$$F(A, f) : F(X) \otimes F(A) \xrightarrow{\varphi} F(X \otimes A) \xrightarrow{F(f)} F(Y)$$

Proposition (probably): **Para**$(-)$ defines a monad on [symmetric monoidal categories, lax symmetric monoidal functors]

Backprop as Functor

Theorem (Fong, Spivak & Tuyéras): Fix a learning rate $\varepsilon > 0$ and a differentiable cost function[2] $C : \mathbb{R}^2 \to \mathbb{R}$.

[2] such that every $\frac{\partial}{\partial y} C(x, y)$ is invertible

Backprop as Functor

Theorem (Fong, Spivak & Tuyéras): Fix a learning rate $\varepsilon > 0$ and a differentiable cost function[2] $C : \mathbb{R}^2 \to \mathbb{R}$.

Then there is a symmetric monoidal functor $F_{\varepsilon,C} : \mathbf{Para}(\mathbf{Euc}) \to \mathbf{Learn}$ defined by

- On objects $X \mapsto$ underlying set of X

[2] such that every $\frac{\partial}{\partial y} C(x, y)$ is invertible

Backprop as Functor

Theorem (Fong, Spivak & Tuyéras): Fix a learning rate $\varepsilon > 0$ and a differentiable cost function[2] $C : \mathbb{R}^2 \to \mathbb{R}$.

Then there is a symmetric monoidal functor $F_{\varepsilon,C} : \mathbf{Para}(\mathbf{Euc}) \to \mathbf{Learn}$ defined by

- On objects $X \mapsto$ underlying set of X
- On morphisms $f : P \times X \to Y$:
  - Parameters P
  - Implementation $I = f$

[2] such that every $\frac{\partial}{\partial y} C(x, y)$ is invertible

Backprop as Functor

Theorem (Fong, Spivak & Tuyéras): Fix a learning rate $\varepsilon > 0$ and a differentiable cost function[2] $C : \mathbb{R}^2 \to \mathbb{R}$.

Then there is a symmetric monoidal functor $F_{\varepsilon,C} : \mathbf{Para}(\mathbf{Euc}) \to \mathbf{Learn}$ defined by

- On objects $X \mapsto$ underlying set of X
- On morphisms $f : P \times X \to Y$:
  - Parameters P
  - Implementation $I = f$
  - Update $U(a, x, y) = a - \varepsilon \nabla_a E(a, x, y)$
  - Request $r(a, x, y) =$ (too awkward to write down)

where $E(a, x, y) = \sum_{i=1}^{\dim(Y)} C(f(p, x)_i, y_i)$ is total error

[2] such that every $\frac{\partial}{\partial y} C(x, y)$ is invertible

Backprop as Functor

Theorem (Fong, Spivak & Tuyéras): Fix a learning rate $\varepsilon > 0$ and a differentiable cost function[2] $C : \mathbb{R}^2 \to \mathbb{R}$.

Then there is a symmetric monoidal functor $F_{\varepsilon,C}$: **Para**(**Euc**) $\to$ **Learn** defined by

- On objects $X \mapsto$ underlying set of X
- On morphisms $f : P \times X \to Y$:
  - Parameters P
  - Implementation $I = f$
  - Update $U(a, x, y) = a - \varepsilon \nabla_a E(a, x, y)$
  - Request $r(a, x, y) =$ (too awkward to write down)

  where $E(a, x, y) = \sum_{i=1}^{\dim(Y)} C(f(p, x)_i, y_i)$ is total error

Update is gradient descent, and request is backpropagation

[2] such that every $\frac{\partial}{\partial y} C(x, y)$ is invertible

ML doesn't work like that

Actual backpropagation backpropagates gradients

ML doesn't work like that

Actual backpropagation backpropagates gradients

Request backpropagates a finite step in the gradient direction

ML doesn't work like that

Actual backpropagation backpropagates gradients

Request backpropagates a finite step in the gradient direction

This is a hack because objects of **Learn** doesn't have differentiable structure

ML doesn't work like that

Actual backpropagation backpropagates gradients

Request backpropagates a finite step in the gradient direction

This is a hack because objects of **Learn** doesn't have differentiable structure

(The benefit is **Learn** is more general than just ML)

Bundles

Work in a category with finite limits

A bundle over X is a morphism $\begin{array}{c} E \\ \downarrow p \\ X \end{array}$

Bundles

Work in a category with finite limits

A bundle over X is a morphism $\begin{array}{c} E \\ \downarrow p \\ X \end{array}$

Examples:

1. Trivial bundle $\begin{array}{c} X \times Y \\ \downarrow \pi_1 \\ X \end{array}$

Bundles

Work in a category with finite limits

A bundle over X is a morphism $$\begin{array}{c} E \\ \downarrow p \\ X \end{array}$$

Examples:

1. Trivial bundle $$\begin{array}{c} X \times Y \\ \downarrow \pi_1 \\ X \end{array}$$

2. Tangent bundle over a differentiable manifold $$\begin{array}{c} T\mathcal{M} \\ \downarrow \pi \\ \mathcal{M} \end{array}$$

Bundles

Work in a category with finite limits

A bundle over X is a morphism $\begin{array}{c} E \\ \downarrow p \\ X \end{array}$

Examples:

1. Trivial bundle $\begin{array}{c} X \times Y \\ \downarrow \pi_1 \\ X \end{array}$
2. Tangent bundle over a differentiable manifold $\begin{array}{c} T\mathcal{M} \\ \downarrow \pi \\ \mathcal{M} \end{array}$
3. Cotangent bundle $\begin{array}{c} T^*\mathcal{M} \\ \downarrow \pi^* \\ \mathcal{M} \end{array}$

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$
- so, the tangent bundle is trivial:

$$\begin{array}{c} T(X) \\ \downarrow{\scriptstyle \pi_2} \\ X \end{array}$$

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$
- so, the tangent bundle is trivial: $$\begin{array}{c} T(X) \\ \downarrow{\scriptstyle \pi_2} \\ X \end{array}$$

Moreover:

- every $T^*_x(X) \cong X$ unnaturally (since $X^* \cong X$)

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$
- so, the tangent bundle is trivial: $$\begin{array}{c} T(X) \\ \big\downarrow{\scriptstyle \pi_2} \\ X \end{array}$$

Moreover:

- every $T^*_x(X) \cong X$ unnaturally (since $X^* \cong X$)
- so, $T^*(X) \cong X \times X$ unnaturally

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$
- so, the tangent bundle is trivial: $$\begin{array}{c} T(X) \\ \downarrow{\scriptstyle \pi_2} \\ X \end{array}$$

Moreover:

- every $T_x^*(X) \cong X$ unnaturally (since $X^* \cong X$)
- so, $T^*(X) \cong X \times X$ unnaturally
- elements of $X \times X$ are called dual numbers

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$
- so, the tangent bundle is trivial: $$\begin{array}{c} T(X) \\ \downarrow{\scriptstyle \pi_2} \\ X \end{array}$$

Moreover:

- every $T_x^*(X) \cong X$ unnaturally (since $X^* \cong X$)
- so, $T^*(X) \cong X \times X$ unnaturally
- elements of $X \times X$ are called dual numbers
- the cotangent bundle is unnaturally equivalent to a trivial bundle

Bundles over Euclidean spaces

If $X = \mathbb{R}^n$ is a Euclidean space then

- every $T_x(X) \cong X$
- so, $T(X) \cong X \times X$
- so, the tangent bundle is trivial: $$\begin{array}{c} T(X) \\ \big\downarrow \pi_2 \\ X \end{array}$$

Moreover:

- every $T_x^*(X) \cong X$ unnaturally (since $X^* \cong X$)
- so, $T^*(X) \cong X \times X$ unnaturally
- elements of $X \times X$ are called dual numbers
- the cotangent bundle is unnaturally equivalent to a trivial bundle

Nb. **Euc** doesn't have finite limits, so we work in **Top**

Morphisms of bundles

A bundle morphism $f : \begin{array}{c} E \\ \big\downarrow p \\ X \end{array} \to \begin{array}{c} F \\ \big\downarrow q \\ Y \end{array}$ is:

Morphisms of bundles

A bundle morphism $f : \begin{array}{c} E \\ \downarrow p \\ X \end{array} \to \begin{array}{c} F \\ \downarrow q \\ Y \end{array}$ is:

- Morphisms $f : X \to Y$ and $f^{\#} : X \times_Y F \to E$

Morphisms of bundles

A bundle morphism $f : \begin{array}{c} E \\ \Big\downarrow{\scriptstyle p} \\ X \end{array} \to \begin{array}{c} F \\ \Big\downarrow{\scriptstyle q} \\ Y \end{array}$ is:

- Morphisms $f : X \to Y$ and $f^{\#} : X \times_Y F \to E$
- such that

$$\begin{array}{ccc} X \times_Y F & \longrightarrow & F \\ {\scriptstyle f^{\#}}\Big\downarrow & \lrcorner & \\ E & & \Big\downarrow{\scriptstyle q} \\ {\scriptstyle p}\Big\downarrow & & \\ X & \xrightarrow{\ f\ } & Y \end{array}$$

is a pullback

Morphisms of bundles

A bundle morphism $f : \begin{array}{c} E \\ \downarrow p \\ X \end{array} \to \begin{array}{c} F \\ \downarrow q \\ Y \end{array}$ is:

- Morphisms $f : X \to Y$ and $f^\# : X \times_Y F \to E$
- such that

$$\begin{array}{ccc} X \times_Y F & \longrightarrow & F \\ {\scriptstyle f^\#}\downarrow & \lrcorner & \\ E & & \downarrow{\scriptstyle q} \\ {\scriptstyle p}\downarrow & & \\ X & \xrightarrow{f} & Y \end{array}$$

  is a pullback
- Equivalently: f such that $X \times_Y F \to X$ factors through p

Morphisms of bundles

A bundle morphism $f : \begin{array}{c} E \\ \downarrow p \\ X \end{array} \rightarrow \begin{array}{c} F \\ \downarrow q \\ Y \end{array}$ is:

- Morphisms $f : X \to Y$ and $f^\# : X \times_Y F \to E$
- such that

$$\begin{array}{ccc} X \times_Y F & \longrightarrow & F \\ \downarrow f^\# & \lrcorner & \\ E & & \downarrow q \\ \downarrow p & & \\ X & \xrightarrow{f} & Y \end{array}$$

  is a pullback
- Equivalently: f such that $X \times_Y F \to X$ factors through p

"Every algebraic geometer knows this definition" – David Spivak

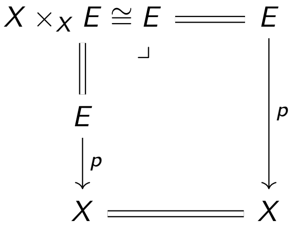
The category of bundles

Identity morphism:

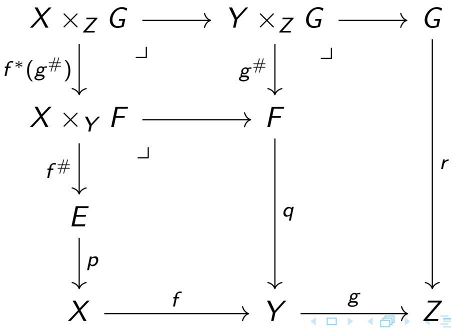
$$\begin{array}{ccc} X \times_X E \cong E & = & E \\ \| \quad\lrcorner & & \\ E & & \big\downarrow p \\ \big\downarrow p & & \\ X & = & X \end{array}$$

The category of bundles

Identity morphism:



Composition of morphisms:



Where does this come from?

From the Grothendieck construction:

$$\mathbf{Bund}(\mathcal{C}) = \int_{X \in \mathcal{C}} (\mathcal{C}/X)^{\mathrm{op}}$$

Where does this come from?

From the Grothendieck construction:

$$\mathbf{Bund}(\mathcal{C}) = \int_{X \in \mathcal{C}} (\mathcal{C}/X)^{\mathrm{op}}$$

This buys us (conjecture) a monoidal structure:

$$\begin{array}{c} E \\ \downarrow p \\ X \end{array} \otimes \begin{array}{c} F \\ \downarrow q \\ Y \end{array} = \begin{array}{c} E \times F \\ \downarrow p \times q \\ X \times Y \end{array}$$

(this might not be the right one!)

Lenses

A (bimorphic) lens $\lambda : (S, T) \to (A, B)$ consists of:

- a morphism $\lambda_v : S \to A$ called view
- a morphism $\lambda_u : S \times B \to T$ called update

Lenses

A (bimorphic) lens $\lambda : (S, T) \to (A, B)$ consists of:

- a morphism $\lambda_v : S \to A$ called view
- a morphism $\lambda_u : S \times B \to T$ called update

Composition of lenses is fiddly

Lenses

A (bimorphic) lens $\lambda : (S, T) \to (A, B)$ consists of:

- a morphism $\lambda_v : S \to A$ called view
- a morphism $\lambda_u : S \times B \to T$ called update

Composition of lenses is fiddly

Where does this come from? The Grothendieck construction:

$$\mathbf{Lens}(\mathcal{C}) = \int_{X \in \mathcal{C}} \mathrm{coKl}(X \times -)^{\mathrm{op}}$$

Lenses

A (bimorphic) lens $\lambda : (S, T) \to (A, B)$ consists of:

- a morphism $\lambda_v : S \to A$ called view
- a morphism $\lambda_u : S \times B \to T$ called update

Composition of lenses is fiddly

Where does this come from? The Grothendieck construction:

$$\mathbf{Lens}(\mathcal{C}) = \int_{X \in \mathcal{C}} \mathrm{coKl}(X \times -)^{\mathrm{op}}$$

Theorem (Lambek): $\mathrm{coKl}(X \times -) \cong \mathcal{C}[x]$, where $\mathcal{C}[x]$ is the polynomial category formed by freely adjoining $x : 1 \to X$ and closing under finite products

Lenses are bundle morphisms

Another theorem (Lambek): $\mathrm{coEM}(X \times -) \cong \mathcal{C}/X$

Lenses are bundle morphisms

Another theorem (Lambek): $\mathrm{coEM}(X \times -) \cong \mathcal{C}/X$

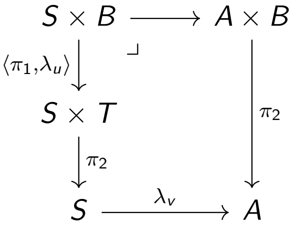
So there is a canonoical embedding $\mathcal{C}[x] \hookrightarrow \mathcal{C}/X$

Lenses are bundle morphisms

Another theorem (Lambek): $\mathrm{coEM}(X \times -) \cong \mathcal{C}/X$

So there is a canonoical embedding $\mathcal{C}[x] \hookrightarrow \mathcal{C}/X$

Grothendieck them all together: $\mathbf{Lens}(\mathcal{C}) \to \mathbf{Bund}(\mathcal{C})$
It takes a lens $\lambda : (S, T) \to (A, B)$ to the bundle morphism

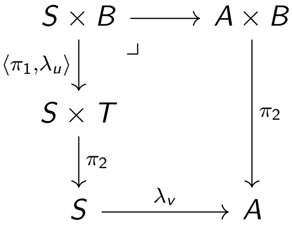


Lenses are bundle morphisms

Another theorem (Lambek): $\mathrm{coEM}(X \times -) \cong \mathcal{C}/X$

So there is a canonoical embedding $\mathcal{C}[x] \hookrightarrow \mathcal{C}/X$

Grothendieck them all together: $\mathbf{Lens}(\mathcal{C}) \to \mathbf{Bund}(\mathcal{C})$
It takes a lens $\lambda : (S, T) \to (A, B)$ to the bundle morphism



Morphisms of contangent bundles

There is a functor $\mathbf{Cot}(-) : \mathbf{DiffMfd} \to \mathbf{Bund}(\mathbf{Top})$

Morphisms of contangent bundles

There is a functor $\mathbf{Cot}(-) : \mathbf{DiffMfd} \to \mathbf{Bund}(\mathbf{Top})$

It takes $f : X \to Y$ to

$$\begin{array}{ccc} X \times_Y T^*(Y) & \longrightarrow & T^*(Y) \\ {\scriptstyle f'}\downarrow \quad \lrcorner & & \\ T^*(X) & & \downarrow{\scriptstyle \pi^*} \\ {\scriptstyle \pi^*}\downarrow & & \\ X & \xrightarrow{\quad f \quad} & Y \end{array}$$

where $f' : (x, c) \mapsto (x, c \circ J_x(f))$

Morphisms of contangent bundles

There is a functor $\mathbf{Cot}(-) : \mathbf{DiffMfd} \to \mathbf{Bund}(\mathbf{Top})$

It takes $f : X \to Y$ to

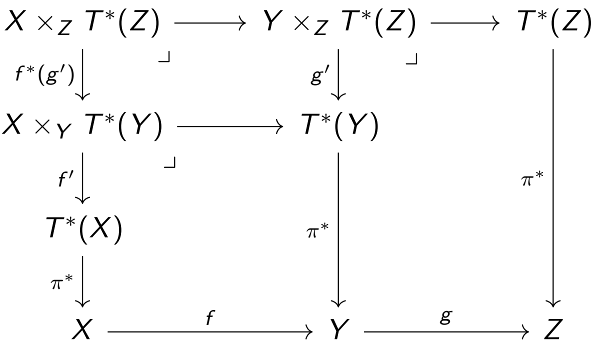
$$\begin{array}{ccc} X \times_Y T^*(Y) & \longrightarrow & T^*(Y) \\ \downarrow f' & \lrcorner & \\ T^*(X) & & \downarrow \pi^* \\ \downarrow \pi^* & & \\ X & \xrightarrow{\quad f \quad} & Y \end{array}$$

where $f' : (x, c) \mapsto (x, c \circ J_x(f))$

$J_x(f)$ is the Jacobian (matrix of partial derivatives) of f at x

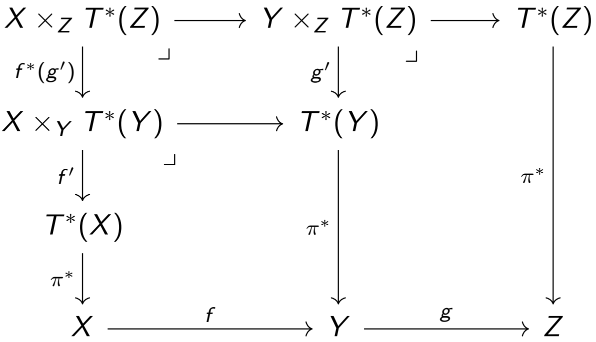
The chain rule

Functorality of **Cot**$(-)$:



The chain rule

Functorality of **Cot**$(-)$:



$(g \circ f)' = f' \circ f^*(g')$ is the chain rule in differential geometry

From **Para(Bund(Top))** to **Learn**

Consider a morphism of **Para**(**Bund**(**Top**)) in the image of

$$\mathbf{Para}(\mathbf{Cot}) : \mathbf{Para}(\mathbf{Euc}) \to \mathbf{Para}(\mathbf{Bund}(\mathbf{Top}))$$

It looks like

$$\begin{array}{ccc}
(X \times A) \times_Y T^*(Y) & \longrightarrow & Y \\
{\scriptstyle f'}\downarrow & \lrcorner & \\
T^*(X \times A) & & \downarrow{\scriptstyle \pi^*} \\
{\scriptstyle \pi^*}\downarrow & & \\
X \times A & \xrightarrow{\quad f \quad} & Y
\end{array}$$

From **Para(Bund(Top))** to **Learn**

Consider a morphism of **Para**(**Bund**(**Top**)) in the image of

$$\mathbf{Para}(\mathbf{Cot}) : \mathbf{Para}(\mathbf{Euc}) \to \mathbf{Para}(\mathbf{Bund}(\mathbf{Top}))$$

It looks like

$$\begin{array}{ccc}
(X \times A) \times_Y T^*(Y) & \longrightarrow & Y \\
\Big\downarrow{\scriptstyle f'} \quad \lrcorner & & \Big\downarrow{\scriptstyle \pi^*} \\
T^*(X \times A) & & \\
\Big\downarrow{\scriptstyle \pi^*} & & \\
X \times A & \xrightarrow{\quad f \quad} & Y
\end{array}$$

We're going to turn it into an open learner, given $\varepsilon > 0$ and differentiable $C : \mathbb{R}^2 \to \mathbb{R}$

The setup

Obviously, parameters are A and implementation is f

The setup

Obviously, parameters are A and implementation is f

We need to define $\langle U, r\rangle : A \times X \times Y \to A \times X$

so, fix $a \in A$, $x \in X$ and $y \in Y$

and fix the total error $C_y(y') = \sum_{i=1}^{\dim(Y)} C(y_i, y'_i)$

The setup

Obviously, parameters are A and implementation is f

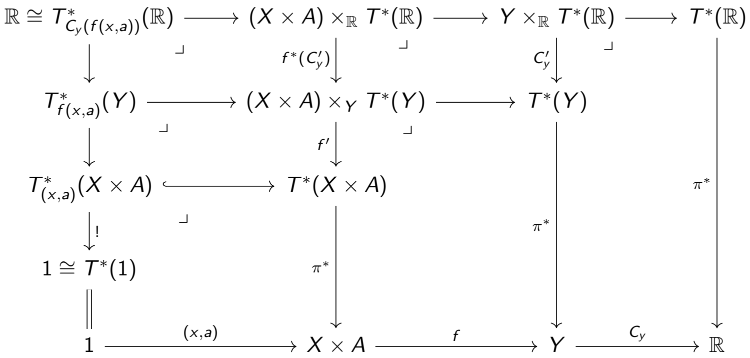
We need to define $\langle U, r \rangle : A \times X \times Y \to A \times X$

so, fix $a \in A$, $x \in X$ and $y \in Y$

and fix the total error $C_y(y') = \sum_{i=1}^{\dim(Y)} C(y_i, y'_i)$

Consider the diagram. . .

The brain exploding part



The part we don't understand

Now: Chase $1 \in \mathbb{R}$ to $T^*(X \times A)$ and then apply

$$\mu_\varepsilon : T^*(X \times A) \to X \times A$$

The result is $\langle r, U \rangle (a, x, y)$

The part we don't understand

Now: Chase $1 \in \mathbb{R}$ to $T^*(X \times A)$ and then apply

$$\mu_\varepsilon : T^*(X \times A) \to X \times A$$

The result is $\langle r, U \rangle (a, x, y)$

μ_ε takes a finite step in the gradient direction:

$$\mu_\varepsilon((x, a), (v, w)) = (x + v, a + \varepsilon w)$$

The part we don't understand

Now: Chase $1 \in \mathbb{R}$ to $T^*(X \times A)$ and then apply

$$\mu_\varepsilon : T^*(X \times A) \to X \times A$$

The result is $\langle r, U \rangle (a, x, y)$

μ_ε takes a finite step in the gradient direction:

$$\mu_\varepsilon((x, a), (v, w)) = (x + v, a + \varepsilon w)$$

What is μ_ε? We couldn't find any nice properties

The part we don't understand

Now: Chase $1 \in \mathbb{R}$ to $T^*(X \times A)$ and then apply

$$\mu_\varepsilon : T^*(X \times A) \to X \times A$$

The result is $\langle r, U \rangle (a, x, y)$

μ_ε takes a finite step in the gradient direction:

$$\mu_\varepsilon((x, a), (v, w)) = (x + v, a + \varepsilon w)$$

What is μ_ε? We couldn't find any nice properties

It looks a bit like a thing called an exponential map

The catch

Conjecture: This defines a symmetric monoidal functor

$$\mathbf{Para}(\mathbf{Bund}(\mathbf{Top})) \supseteq \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \to \mathbf{Learn}$$

The catch

Conjecture: This defines a symmetric monoidal functor

$$\mathbf{Para}(\mathbf{Bund}(\mathbf{Top})) \supseteq \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \to \mathbf{Learn}$$

Another conjecture: This commutes:

$$\begin{array}{ccc} \mathbf{Para}(\mathbf{Euc}) & \xrightarrow{\mathbf{Para}(\mathbf{Cot})} & \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \\ & \searrow & \downarrow \\ & & \mathbf{Learn} \end{array}$$

The catch

Conjecture: This defines a symmetric monoidal functor

$$\mathbf{Para}(\mathbf{Bund}(\mathbf{Top})) \supseteq \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \to \mathbf{Learn}$$

Another conjecture: This commutes:

$$\begin{array}{ccc} \mathbf{Para}(\mathbf{Euc}) & \xrightarrow{\mathbf{Para}(\mathbf{Cot})} & \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \\ & \searrow & \downarrow \\ & & \mathbf{Learn} \end{array}$$

The catch: We think **Para(Cot)** is an equivalence of categories onto its image

The catch

Conjecture: This defines a symmetric monoidal functor

$$\mathbf{Para}(\mathbf{Bund}(\mathbf{Top})) \supseteq \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \to \mathbf{Learn}$$

Another conjecture: This commutes:

$$\begin{array}{ccc} \mathbf{Para}(\mathbf{Euc}) & \xrightarrow{\mathbf{Para}(\mathbf{Cot})} & \mathrm{Im}(\mathbf{Para}(\mathbf{Cot})) \\ & \searrow & \downarrow \\ & & \mathbf{Learn} \end{array}$$

The catch: We think **Para**(**Cot**) is an equivalence of categories onto its image

So, we've just rewritten Backprop as Functor in a different way!

Even more hard questions

What happens if we extend the functor to the whole of **Para**(**Bund**(**Top**))? We have no idea!

Optimistic hope: This allows defining general "ML-like" systems, not necessarily involving gradients (eg. "discrete ML" on Bayesian networks)