GENERALIZED OPEN LEARNERS (AGENTS)

WHAT: UNDERSTAND THE INFORMATION FLOW IN ARTIFICIAL NEURAL NETWORKS

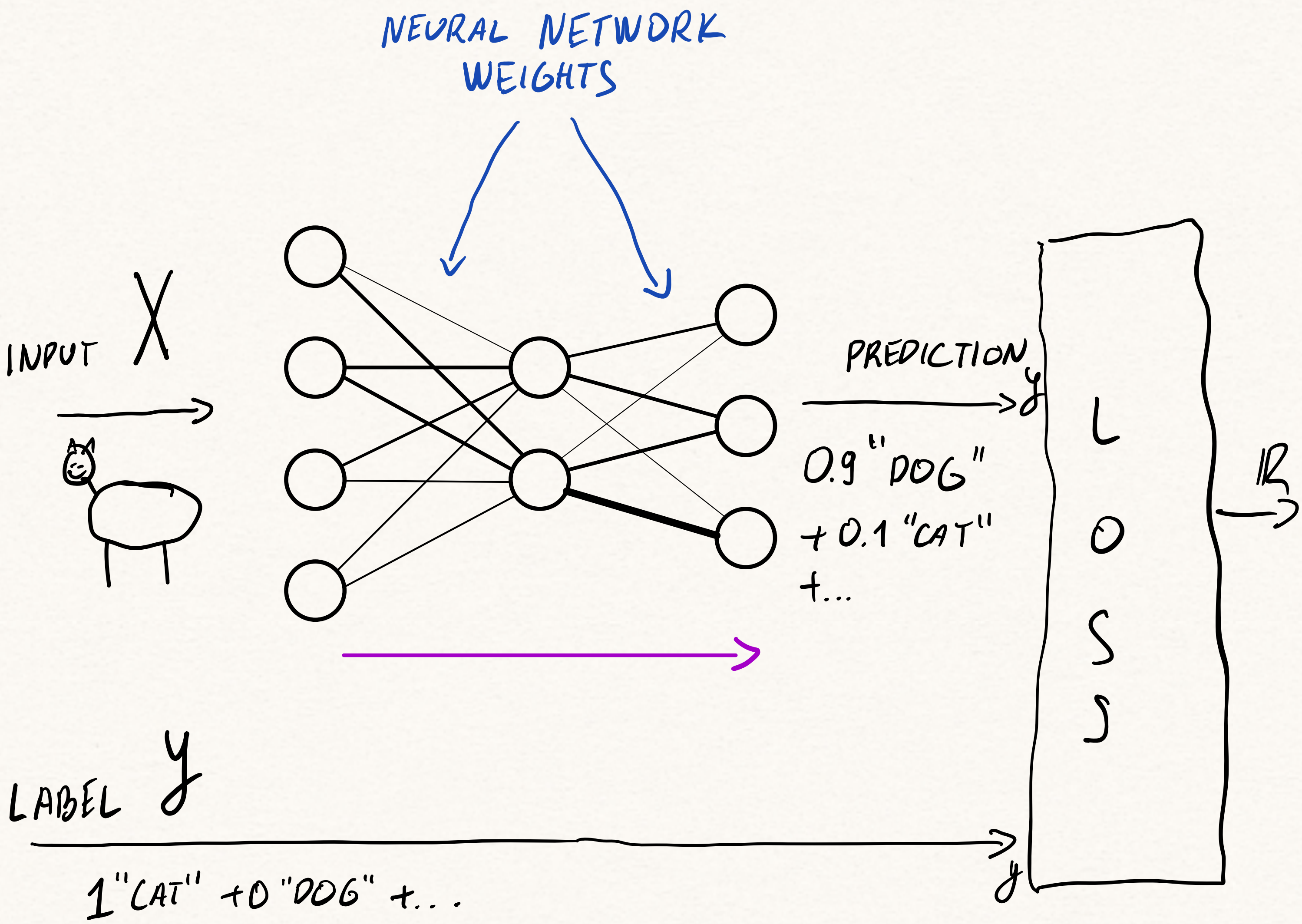
WHY: CONCRETE PROXY FOR UNDERSTANDING GENERAL CYBERNETICS SYSTEMS

HOW: TAKE PARAMETERIZATION AND BIDIRECTIONALITY SERIOUSLY

SUPERVISED LEARNING WITH NEURAL NETWORKS IN ONE SLIDE

$$X \longrightarrow \boxed{\;?\;} \longrightarrow y$$

DATASET: List $X \times Y$



- WEIGHTS CONTROL THE NN PERFORMANCE
- PROPAGATING CHANGES - „BACKPROPAGATION"

ONE STEP, THIS IS ITERATED

RESULTS:

„PARAMETERIZATION" and „BIDIRECTIONALITY"

can be defined separately and then composed

- GENERALIZED OPEN LEARNERS (Agents)

- Abstract treatment of gradient-based learning
  - Backpropagation – RDCs
  - Loss function
  - Update rule

SIMPLER, PEDAGOGICAL

- Works on Euclidean spaces and Boolean circuits

- Defined optimizers (GRADIENT DESCENT, MOMENTUM, ADAM...) as lenses

- Optimizers are 2-cells in GEN. OPEN LEARNERS

GAME PLAN:

- PARAMETERIZATION
- BIDIRECTIONALITY
- PARAMETERIZATION + BIDIRECTIONALITY
- DIFFERENTIATION (HOW DO WE CONSTRUCT LEARNERS?)
- HOW DOES LEARNING WORK?
- EXAMPLES

PARAMETERIZATION

Fix a SMC $(\mathcal{C}, \otimes, I)$.

DEF. $\mathbf{Para}(\mathcal{C})$

Objects - objects of $\mathcal{C}$

CATEGORY OF ELEMENTS

$$\mathbf{Para}(\mathcal{C})(A,B) = \int^{P:\mathcal{C}^{op}} \mathcal{C}(P \otimes A, B)$$

$$A \xrightarrow{(P:\mathcal{C},\, f: P\otimes A \to B)} B$$

2-cells are reparameterizations: a 2-cell

$$\begin{tikzcd} A \arrow[r, bend left, "{(P,f)}"] \arrow[r, bend right, "{(Q,g)}"'] \arrow[r, phantom, "\Downarrow r"] & B \end{tikzcd}$$

is a map $Q \xrightarrow{r} P$ such that

$$\begin{tikzcd} Q \otimes A \arrow[rr, "r \otimes A"] \arrow[dr, "g"'] & & P \otimes A \arrow[dl, "f"] \\ & B & \end{tikzcd}$$

EXAMPLE.

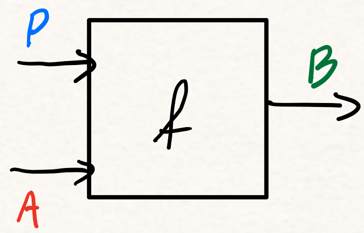
$\mathbf{Para}(\mathbf{Set})$, $\mathbf{Para}(\mathbf{Smooth})$, $\mathbf{Para}(\mathbf{Optic}(\mathcal{C}))$, ...

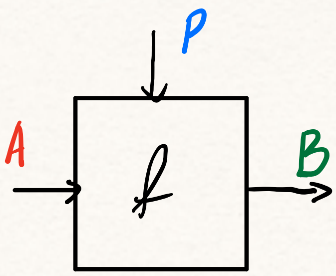
$\cong$

$\mathbf{Para}$

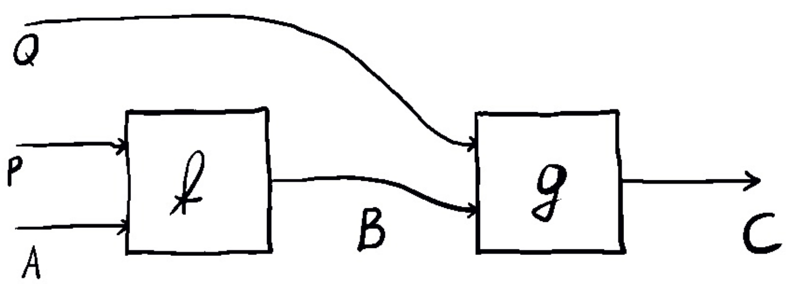
GRAPHICAL LANGUAGE

$$f : P \otimes A \longrightarrow B$$

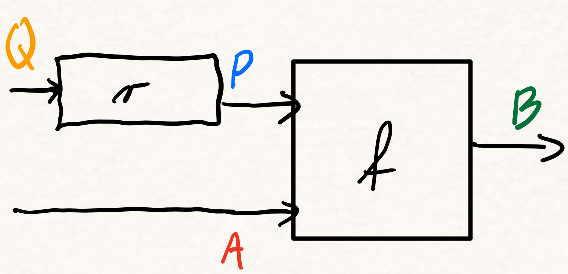


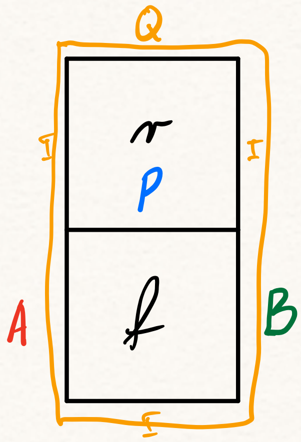


I DIDN'T TELL YOU HOW COMPOSITION WORKS'

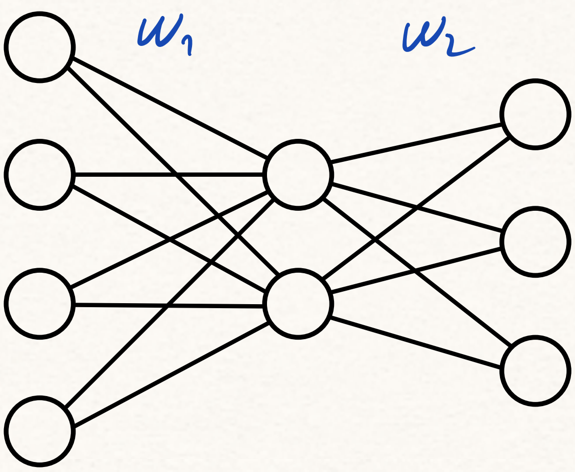


$$\begin{tikzcd} A \arrow[r, bend left, "{(P,f)}"] \arrow[r, bend right, "{(Q,g)}"'] \arrow[r, phantom, "\Downarrow r"] & B \end{tikzcd}$$

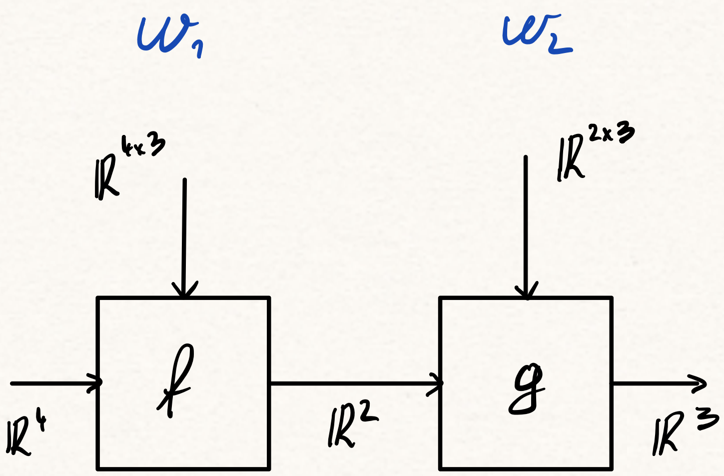




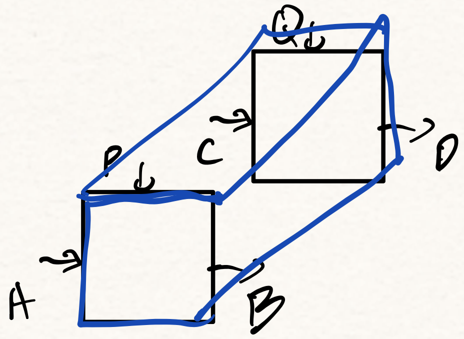
- TILING LANGUAGE OF DOUBLE CATEGORIES (MYERS)

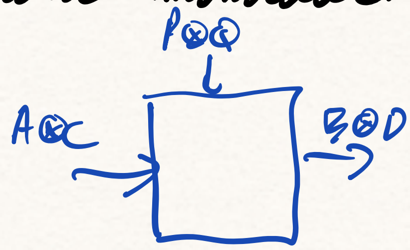


$\Longleftrightarrow$



PROP. Para($\mathcal{C}$) is symmetric monoidal.





Para is natural w.r.t. base change.

DEF.

Let $G : \mathcal{C} \longrightarrow \mathcal{D}$ be a sym. monoidal functor. We

define $\quad \mathrm{Para}(G) : \mathrm{Para}(\mathcal{C}) \longrightarrow \mathrm{Para}(\mathcal{D})$ ⟵ RELEVANT TO BACKPROPAGATION

$$\begin{tikzcd} A \arrow[r, mapsto] \arrow[d, "{(P,f)}"'] & G(A) \arrow[d, "{(G(P),f')}"] \\ B \arrow[r, mapsto] & G(B) \end{tikzcd}$$

where f' is the composite

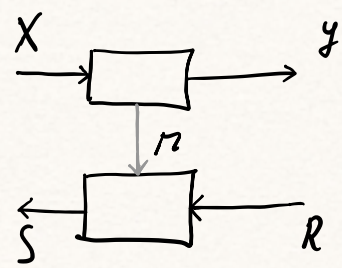
$$G(P) \otimes G(A) \xrightarrow[\mu_{P,A}]{} G(P \otimes A) \xrightarrow{G(f)} G(B)$$

ALSO: Para is an endofunctor on SMC (also a monad) + a lot more structure

BIDIRECTIONALITY
(lenses, optics, dependent lenses, Cont, Poly, ...)

- CURRENTLY FOCUSING ON OPTICS AS THE CANONICAL "BIDIRECTIONAL" STRUCTURE, OPEN TO AMENDMENTS

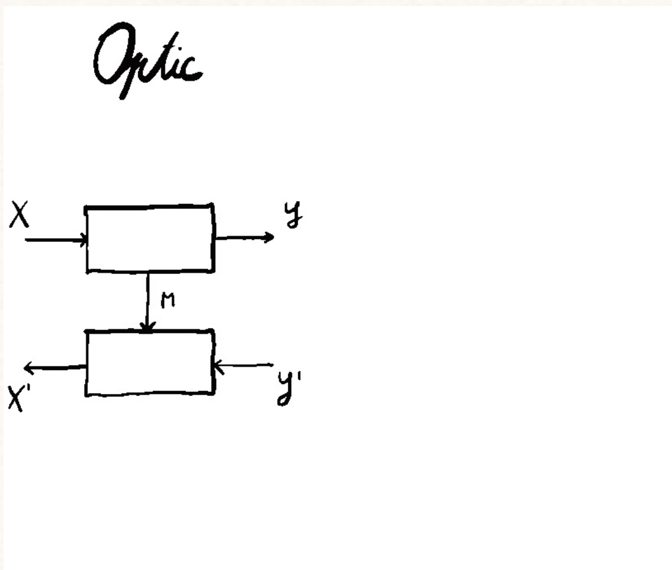
DEF. Category $\mathbf{Optic}(\mathcal{C})$



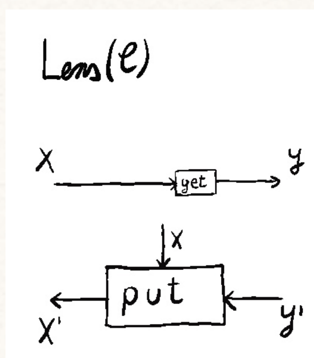
"FROM THE OUTSIDE"

- Objects – pairs of objects in $\mathcal{C}$ $\binom{X}{S}$
- $$\mathbf{Optic}(\mathcal{C})\left(\binom{X}{S},\binom{Y}{R}\right) = \int^{M:\mathcal{C}} \mathcal{C}(X, M\otimes Y)\times\mathcal{C}(M\otimes R, S)$$

  ← COEND

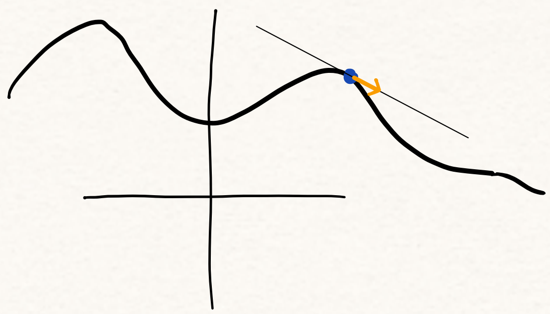


PROP. If $\mathcal{C}$ is Cartesian, then $\mathbf{Lens}(\mathcal{C}) \cong \mathbf{Optic}(\mathcal{C})$



PROP. $\mathrm{Optic}(\mathcal{C})$ is symmetric monoidal.

EXAMPLE. GRADIENT DESCENT



$$\begin{aligned} P \times P &\xrightarrow{u} P \\ (p, \nabla p) &\longmapsto p - \alpha \nabla p \end{aligned}$$

is a lens, for $\mathcal{C} := \mathrm{Smooth}$

$$\begin{pmatrix} P \\ P \end{pmatrix} \xrightarrow{(id_P, u)} \begin{pmatrix} P \\ P \end{pmatrix}$$

EXAMPLE. OTHER OPTIMIZERS:

$$\begin{pmatrix} S \times P \\ S \times P \end{pmatrix} \longrightarrow \begin{pmatrix} P \\ P \end{pmatrix}$$

- MOMENTUM,

  $$\begin{aligned} get : P \times P &\longrightarrow P \\ (v, p) &\longmapsto p \end{aligned}$$

  $$\begin{aligned} put : P \times P \times P &\longrightarrow P \times P \\ (v, p, \nabla p) &\longmapsto (v', p - v') \end{aligned}$$

  where $v' = \gamma v + \varepsilon p'$

- NESTEROV MOMENTUM

  $$\begin{aligned} get : P \times P &\longrightarrow P \\ (v, p) &\longmapsto p - \gamma v \end{aligned}$$

  put – same as above

- ADAGRAD

- ADAM

  ...

PARAMETERIZATION + BIDIRECTIONALITY

$$\mathcal{C} \longmapsto \mathrm{Optic}(\mathcal{C}) \longmapsto \mathrm{Para}(\mathrm{Optic}(\mathcal{C}))$$

- Objects - objects of $\mathrm{Optic}(\mathcal{C})$ - pairs $\begin{pmatrix} X \\ S \end{pmatrix}$ in $\mathcal{C}$

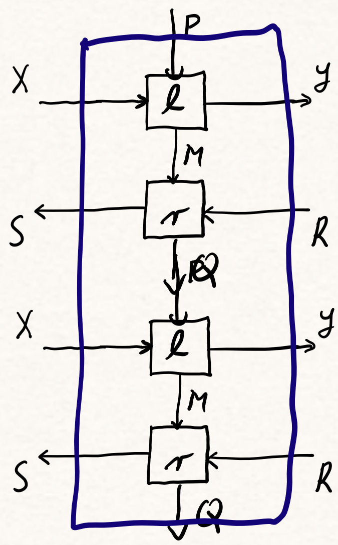
- Morphisms $\begin{pmatrix} X \\ S \end{pmatrix} \xrightarrow{\left(\begin{pmatrix} P \\ Q \end{pmatrix}, f\right)} \begin{pmatrix} Y \\ R \end{pmatrix}$ where $f : \begin{pmatrix} P \otimes X \\ Q \otimes S \end{pmatrix} \longrightarrow \begin{pmatrix} Y \\ R \end{pmatrix}$

(M, l, r)

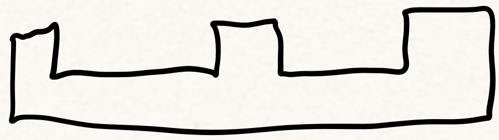
$M : \mathcal{C}$

$l : P \otimes X \longrightarrow M \otimes Y$

$r : M \otimes R \longrightarrow Q \otimes S$

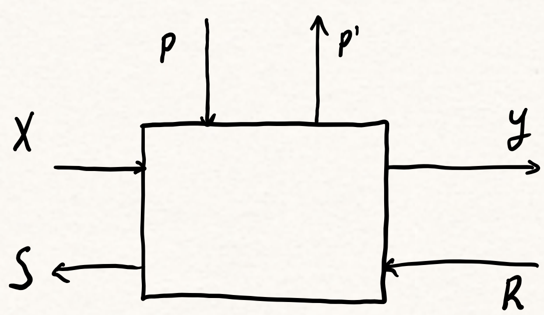


$$\begin{pmatrix} X \otimes X \\ S \otimes S \end{pmatrix} \longrightarrow \begin{pmatrix} Y \otimes Y \\ R \otimes R \end{pmatrix}$$



$\begin{pmatrix} x : X \\ S_x \end{pmatrix}$

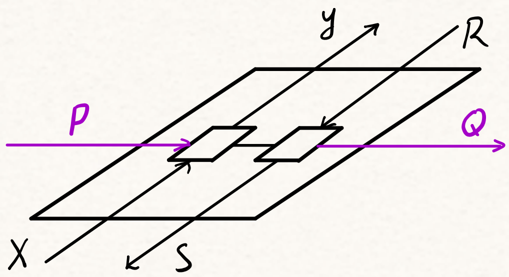
$S : X \to \mathrm{Set}$

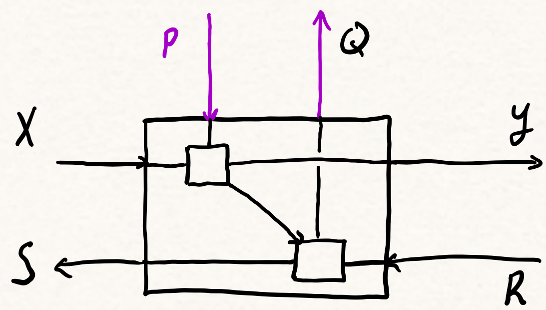


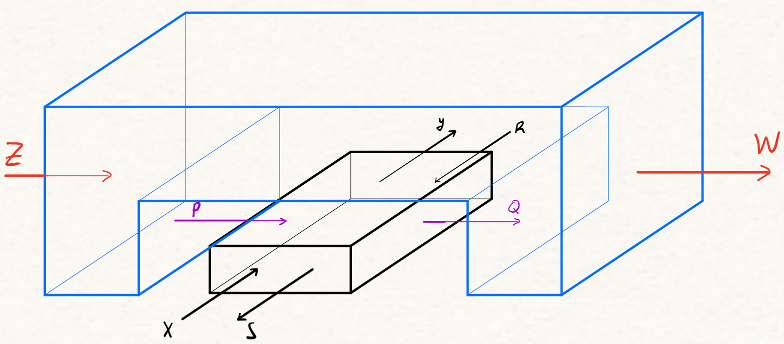
- We automatically get two parameter ports

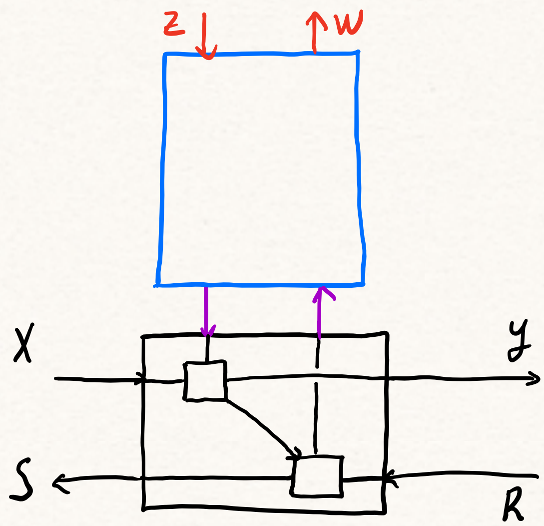
• A 2-cell $\begin{tikzcd} \begin{pmatrix} X \\ S \end{pmatrix} \arrow[r, bend left=50, "{\left(\begin{smallmatrix} P \\ Q \end{smallmatrix}, f\right)}"] \arrow[r, bend right=50, "{\left(\begin{smallmatrix} Z \\ W \end{smallmatrix}, g\right)}"'] \arrow[r, phantom, "\Downarrow r"] & \begin{pmatrix} Y \\ R \end{pmatrix} \end{tikzcd}$ is an optic

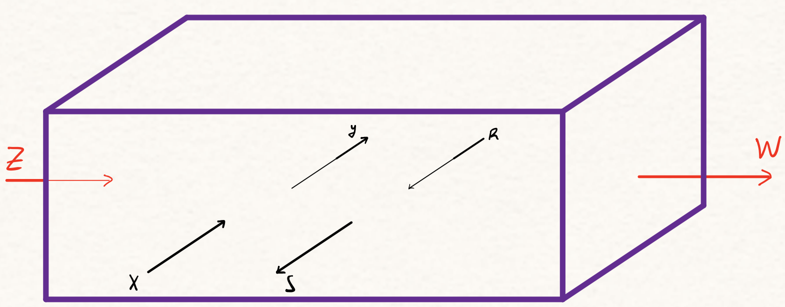
$$\begin{pmatrix} Z \\ W \end{pmatrix} \xrightarrow{r} \begin{pmatrix} P \\ Q \end{pmatrix}$$

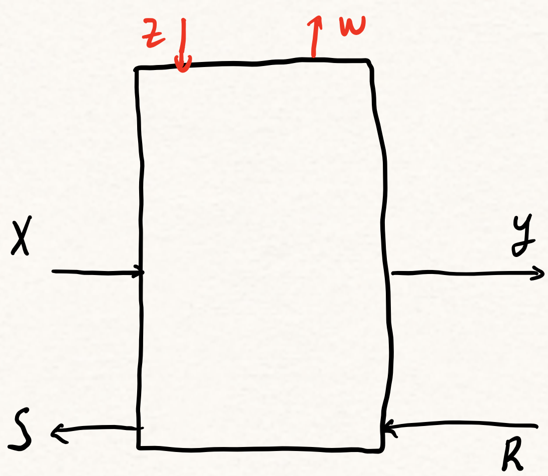












ASIDE: category „Learn" from Backprop as functor

Definition II.1. *Let A and B be sets. A* supervised learning algorithm, *or simply* learner, *$A \to B$ is a tuple (P, I, U, r) where P is a set, and I, U, and r are functions of types:*

$$I : P \times A \to B,$$
$$U : P \times A \times B \to P,$$
$$r : P \times A \times B \to A.$$

THEOREM. $\mathbf{Learn} \cong \mathbf{Para}(\mathbf{Optic}(\mathbf{Set}))$

HOW DO WE ACTUALLY CONSTRUCT SOME USEFUL LEARNERS (NEURAL NETWORKS)?

$\downarrow$

(THIS IS WHAT TENSORFLOW/PYTORCH/JAX ARE DOING)

DIFFERENTIATION

- Reverse derivative categories (RDC).

DEF. A RDC $\mathcal{C}$ is a category which for every

$$A \xrightarrow{f} B$$ the „get" map

has a map

the „put" map of a lens

$$R[f] : A \times B \longrightarrow A$$

subject to some conditions.

EXAMPLE. Smooth IS A RDC. BoolCirc IS A RDC

EXAMPLE. Let $\mathcal{C} := \mathrm{Smooth}$

Let $\begin{array}{c} \mathbb{R} \xrightarrow{f} \mathbb{R} \\ x \longmapsto x^2 \end{array}$. Then $\begin{array}{c} \mathbb{R} \times \mathbb{R} \xrightarrow{R[f]} \mathbb{R} \\ (x, w) \longmapsto 2x w \end{array}$ $\quad f'(x) \cdot w$

PROP. For each RDC $\mathcal{C}$ there is a symm. mon. functor

$$\begin{tikzcd} \mathcal{C} \arrow[d, "F"'] \\ \mathrm{Lens}(\mathcal{C}) \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, mapsto] & B \arrow[d, mapsto] \\ \begin{pmatrix} A \\ A \end{pmatrix} \arrow[r, "{(f, R[f])}"'] & \begin{pmatrix} B \\ B \end{pmatrix} \end{tikzcd}$$

$\mathrm{Optic}(\mathcal{C})$

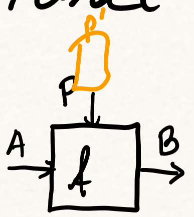
FUNCTORIALITY OF F IS THE CHAIN RULE.

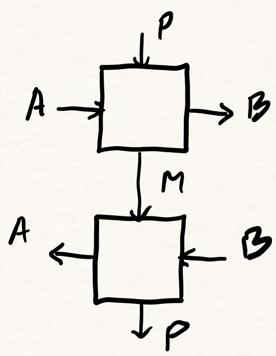
THEOREM. THE ACTION OF Para ON THE FUNCTOR

$$\mathcal{C} \xrightarrow{F} \mathrm{Optic}(\mathcal{C})$$

RESULTS IN A FUNCTOR

$$\mathrm{Para}(\mathcal{C}) \xrightarrow{\mathrm{Para}(F)} \mathrm{Para}(\mathrm{Optic}(\mathcal{C}))$$





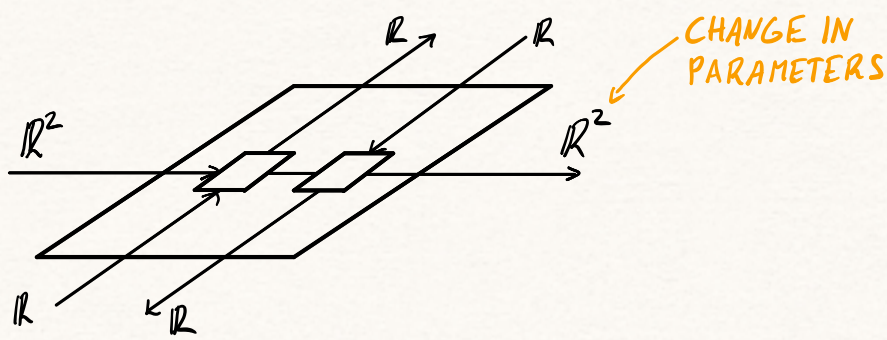
EXAMPLE. Consider a linear layer $(\mathbb{R}^2, f) : \mathrm{Para}(\mathrm{Smooth})(\mathbb{R}, \mathbb{R})$

where

$$\begin{aligned} \mathbb{R}^2 \times \mathbb{R} &\xrightarrow{f} \mathbb{R} \\ ((w_0, w_1), x) &\longmapsto w_0 + w_1 x \end{aligned}$$

Then

$$\begin{aligned} \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} &\xrightarrow{R[f]} \mathbb{R}^2 \times \mathbb{R} \\ ((w_0, w_1), x, dy) &\longmapsto ((dy, x\,dy), w_1 dy) \end{aligned}$$



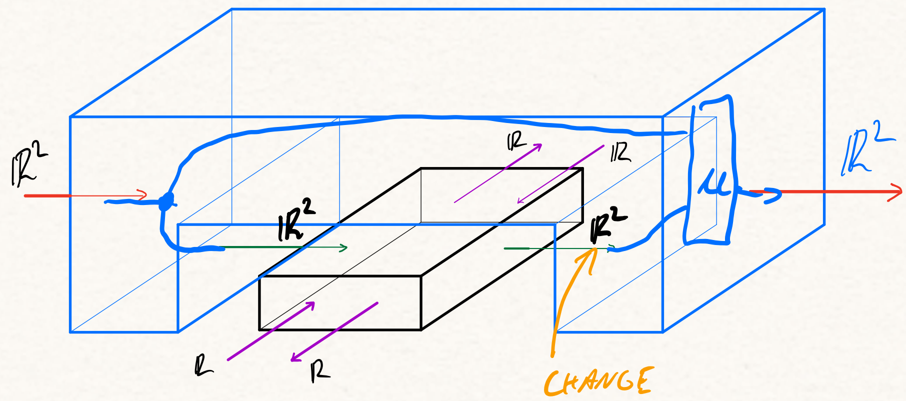
HOW DOES (DERIVATIVE BASED) (SUPERVISED) LEARNING HAPPEN?

1) UPDATE MAP

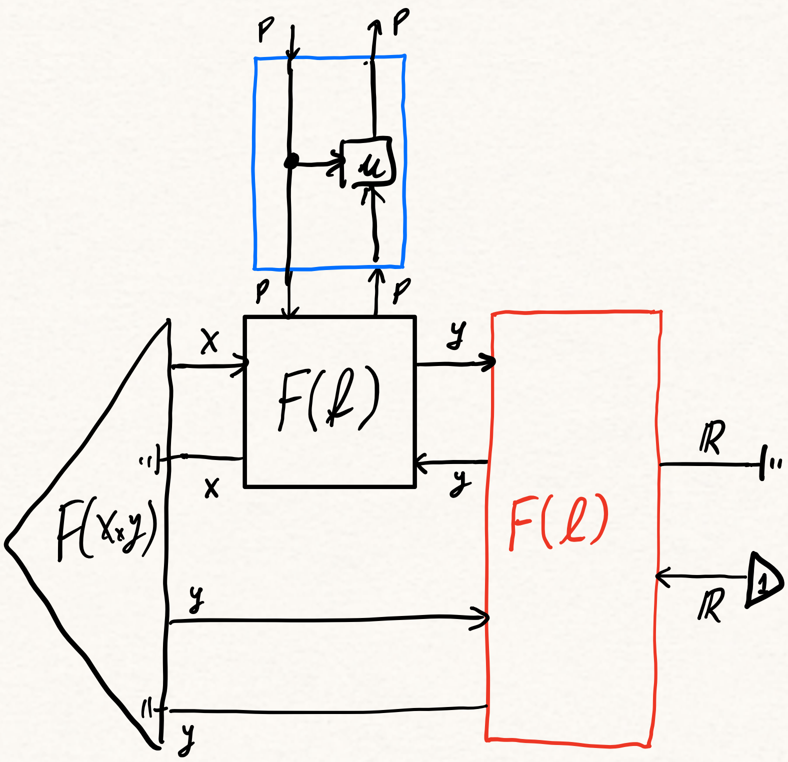
THM. *Gradient descent is a reparameterization.*

$$\begin{aligned} P \times P &\xrightarrow{\ u\ } P \\ (p, \nabla p) &\longmapsto p - \alpha \nabla p \end{aligned}$$

$$\begin{pmatrix} P \\ P \end{pmatrix} \xrightarrow{(id_P,\, u)} \begin{pmatrix} P \\ P \end{pmatrix}$$



2) LOSS FUNCTION



$f : \mathcal{C}(P \times X, y)$

$\ell : \mathcal{C}(y \times y, \mathbb{R})$

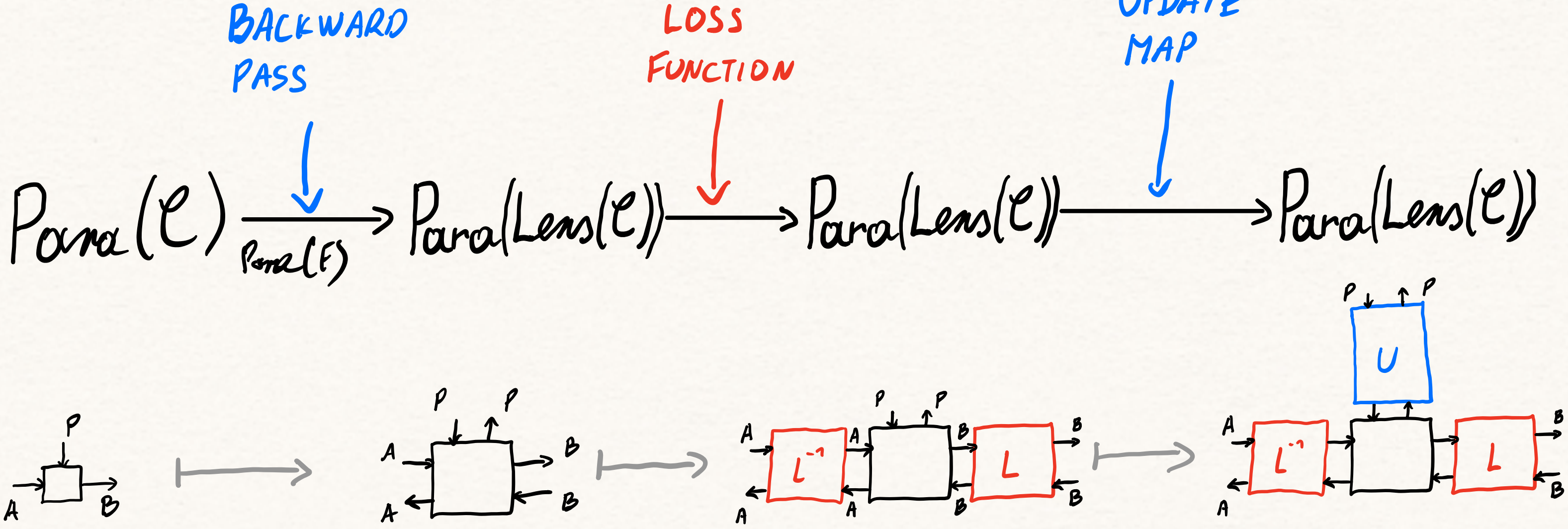
$u : \mathrm{Lens}(\mathcal{C})\left(\begin{smallmatrix} P \\ P \end{smallmatrix}, \begin{smallmatrix} P \\ P \end{smallmatrix}\right)$

THEOREM:

- FIX A RDC $\mathcal{C}$
- FIX LOSS FUNCTION DATA (MSE, SOFTMAX CROSS ENTROPY...)
- FIX PARAMETER UPDATE DATA (G.D., MOMENTUM, ADAGRAD, ADAM, ...)

Then we can define a functor $\mathrm{Para}(\mathcal{C}) \longrightarrow \mathrm{Para}(\mathrm{Lens}(\mathcal{C}))$ which creates a learner.

UPDATE



Theorem III.2. *Fix a real number $\varepsilon > 0$ and $e(x, y)\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ differentiable such that $\frac{\partial e}{\partial x}(x_0, -)\colon \mathbb{R} \to \mathbb{R}$ is invertible for each $x_0 \in \mathbb{R}$. Then we can define a faithful, injective-on-objects, strong symmetric monoidal functor*

$$L_{\varepsilon,e}\colon \underset{\substack{\cong \\ \mathrm{Para(Smooth)}}}{\mathsf{Para}} \longrightarrow \mathsf{Learn} \cong \mathrm{Para(Optic(Set))}$$

that sends each parametrised function $I\colon P \times A \to B$ to the learner (P, I, U_I, r_I) defined by

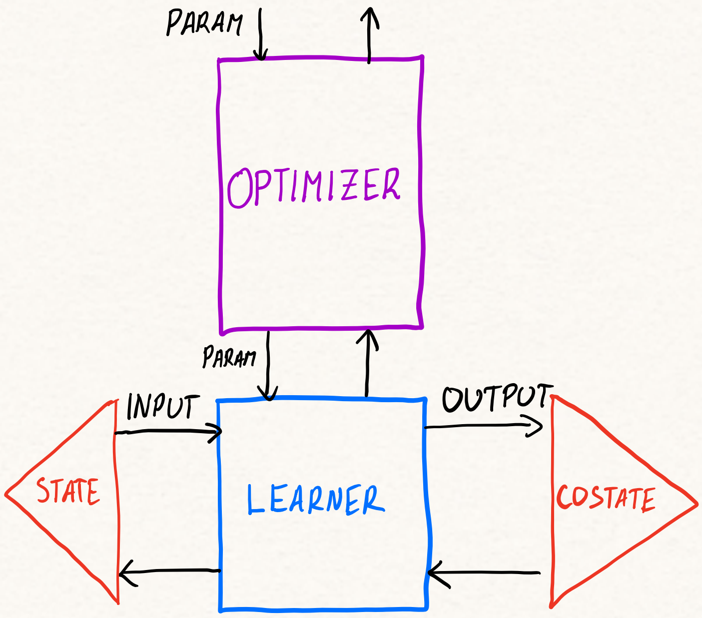
$$U_I(p, a, b) := p - \varepsilon \nabla_p E_I(p, a, b)$$

and

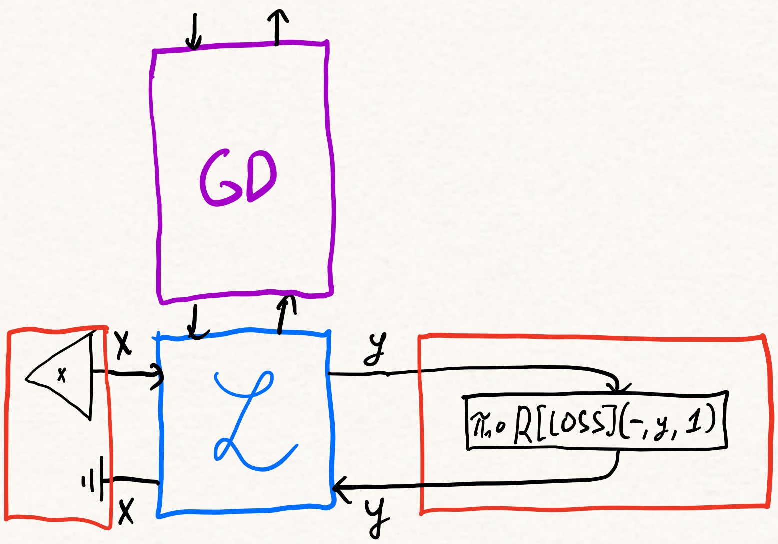
$$r_I(p, a, b) := f_a\Big(\nabla_a E_I(p, a, b)\Big),$$

where $E_I(p, a, b) := \sum_j e(I_j(p, a), b_j)$, and f_a is component-wise application of the inverse to $\frac{\partial e}{\partial x}(a_i, -)$ for each i.

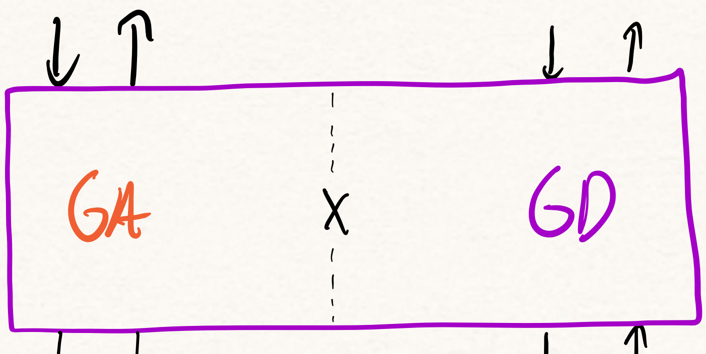
EXAMPLES

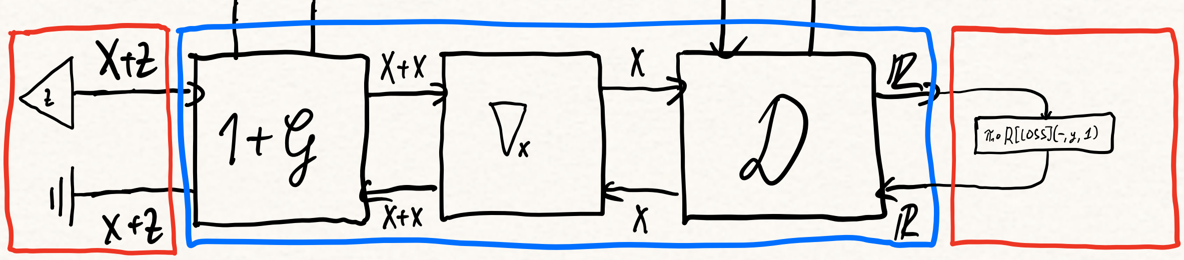


Learning on Smooth

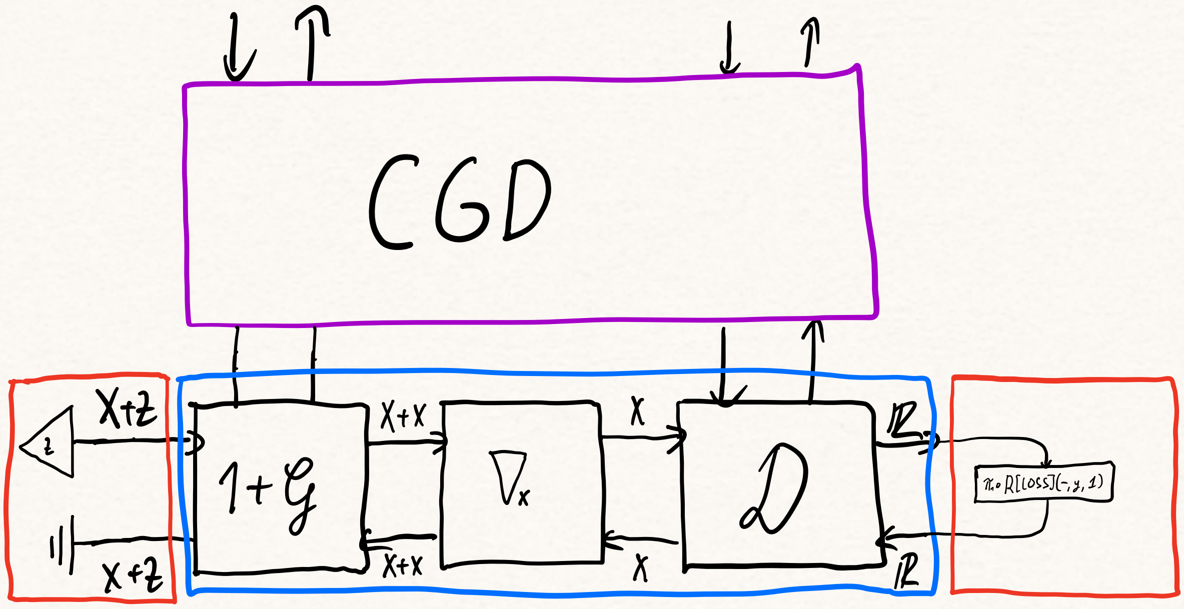


GENERATIVE ADVERSARIAL NETWORKS

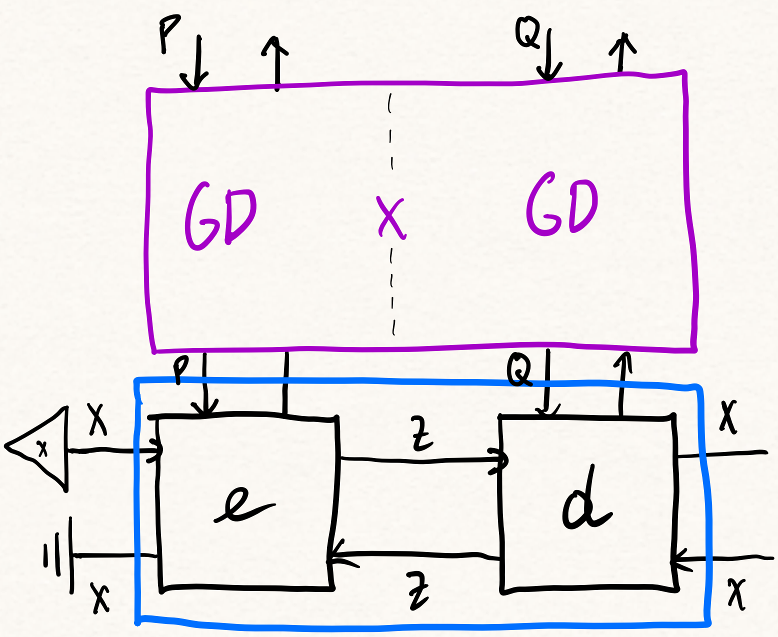




CGD



AUTOENCODERS



$\dim(Z) \ll \dim(X)$

Learning on BoolCirc

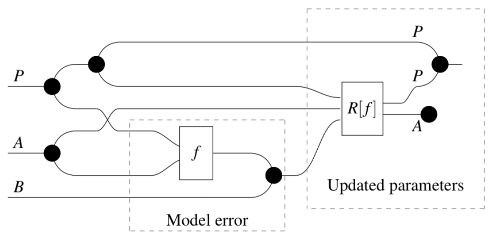
4 Reverse Derivative Ascent

4.1 Reverse Derivative Ascent Algorithm

We now introduce our machine learning algorithm, *reverse derivative ascent*. The definition refers to the category **BoolCirc**, as boolean circuits are our motivating example. However, our formulation makes sense in any reverse differential category.

We proceed in two parts: the inner 'step' of the algorithm, which we call `rdaStep`, and the outer 'iteration' of `rdaStep`, which is `rda`.

Definition 20. Let $f : p+a \to b$ be a boolean circuit in **BoolCirc**, thus computing a parametrised boolean function with p parameters. We define $\mathtt{rdaStep}_f$ as



(7)

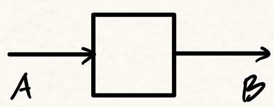
TODO/FUTURE WORK

- DEPENDENT TYPES
- META-LEARNING
- AUTOMATA LEARNING
- OPEN GAMES
- LEARNING ITERATION/REPEATED GAMES
- OPEN DYNAMICAL SYSTEMS?
- PROBABILITY DISTRIBUTIONS?
- Para AND Optic ARE PRETTY GENERAL, WHAT ELSE CAN WE MODEL?

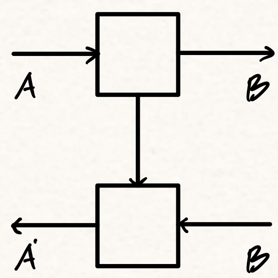
References:

- BACKPROP AS FUNCTOR (FONG, SPIVAK, TUYERAS)
- OPEN GAMES (GHANI, HEDGES, ...)
- REVERSE DERIVATIVE CATEGORIES (COCKETT, CRUTTWELL, GALLAGHER, ...)
- REVERSE DERIVATIVE ASCENT (ZANASI, WILSON)
- DIOPTICS (DARLYMPLE)

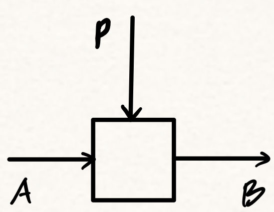
$\mathcal{C}$



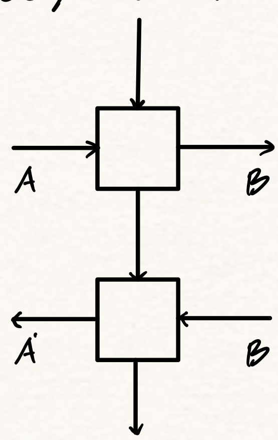
$\mathrm{Optic}(\mathcal{C})$



$\mathrm{Para}(\mathcal{C})$



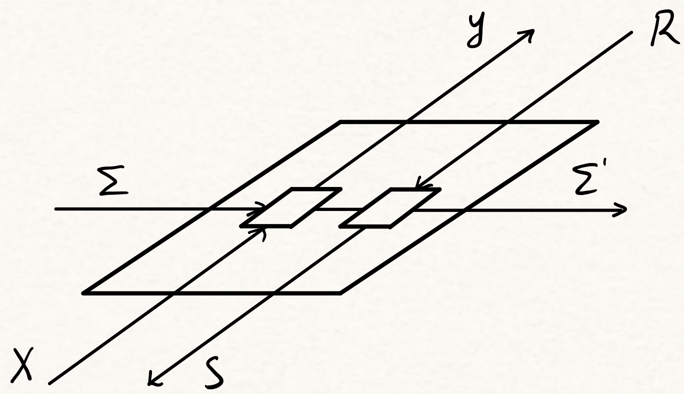
$\mathrm{Para}(\mathrm{Optic}(\mathcal{C}))$



EXTRA: OPEN GAMES, COSTRATEGIES

DEF. (CONTEXT)

$$\bar{\mathcal{C}}(A,B) = \int^{M} \mathcal{C}(I, A\otimes M) \times \mathcal{C}(B\otimes M, I)$$



$$s : (\Sigma \to \Sigma') \to P(\Sigma)$$