Geom. Deep Learn. by Bronstein, Bruna, Cohen, Veličković

Cat. Deep Learn. is an — Gavranović, Lessard, Dudzik, von Glehn, Araújo, Veličković.

# Geom. Deep Learning

- One aim is to classify types of architecture.
- Have a better formal approach to creating suitable architectures based on symmetries.

"signals" $\mathcal{X}(X, \mathcal{E})$ v. space

processing layers

equivariant $f(g \cdot x) = g \cdot f(x)$

output layer invariant under group action. $f(g \cdot x) = f(x)$.

$$\mathcal{X}(X, \mathcal{E}) \xrightarrow{\rho} Y$$

have symmetries

"domain" $X$

e.g. a specific graph, a computer screen

"labels" $Y$

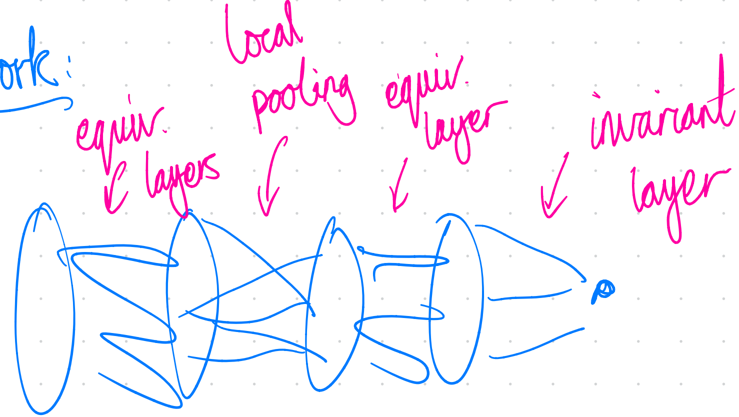
e.g. congested vs. not congested, species of bird $\sqcup$ "nothing"

e.g. screen: $w \times h$ pixels

symmetries are $\mathbb{Z}_w \times \mathbb{Z}_h$

graph: $n$ nodes, permutations $\Sigma_n$.

## GDL network:




Apparently: it's big in drug design & discovery.

## <u>CDL</u>: generalise GDL.

- GDL is based on group actions & not everything in life is a symmetry. What is a domain?

The central claim / observation: group actions on, say, sets are algebras for a certain monad (a group action monad) & equivariant maps between such sets-with-an-action are algebra homomorphisms.

What is a monad? $\mathcal{C}$ cat., it's an endofunc. $T : \mathcal{C} \to \mathcal{C}$

w/ $\eta : \mathrm{Id}_{\mathcal{C}} \Rightarrow T$, $\mu : T \circ T \Rightarrow T$ & these satisfy

$$\begin{tikzcd}
T \arrow[r, Rightarrow, "\eta_T"] \arrow[dr, Rightarrow, "\mathrm{id}"'] & T \circ T \arrow[d, Rightarrow, "\mu"] \\
& T
\end{tikzcd}
\qquad
\begin{tikzcd}
T \circ T \circ T \arrow[r, Rightarrow, "T\mu"] \arrow[d, Rightarrow, "\mu_T"'] & T \circ T \arrow[d, Rightarrow, "\mu"] \\
T \circ T \arrow[r, Rightarrow, "\mu"'] & T
\end{tikzcd}$$

<u>Algebras:</u> An obj. $A \in \mathcal{C}$ along w/ a map $\alpha : T(A) \to A$ s.t.

$$\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[dr, "\mathrm{id}_A"'] & T(A) \arrow[d, "\alpha"] \\
& A
\end{tikzcd}
\qquad
\begin{tikzcd}
TTA \arrow[r, "T\alpha"] \arrow[d, "\mu_A"'] & TA \arrow[d, "\alpha"] \\
TA \arrow[r, "\alpha"'] & A
\end{tikzcd}$$

e.g. Pick a group $G$. We have a monad $T : \mathrm{Set} \to \mathrm{Set}$ sending $S \mapsto G \times S$ (monad b.c. $G \times G \times S \xrightarrow{\mu_G \times S} G \times S$)

$\mu : T \circ T \Rightarrow T$

algebras for this are a set $S$ w/ a map $\alpha : G \times S \to S$.

↑ they are $G$-Sets.

$$\boxed{e \cdot s = s, \quad g \cdot (h \cdot s) = (gh) \cdot s.}$$

$$\begin{tikzcd} TA \arrow[r, "Th"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\circlearrowright"] & TB \arrow[d, "\beta"] \\ A \arrow[r, "h"] & B \end{tikzcd}$$

is an alg. hom.

$$\begin{tikzcd} G \times A \arrow[r, "G \times f"] \arrow[d, "\alpha"'] & G \times B \arrow[d, "\text{triv. action}"] \\ A \arrow[r, "f"] & B \end{tikzcd}$$

$$f(g \cdot x) = g \cdot f(x) = f(x).$$

e.g. $\mathbb{R}^{\mathbb{Z}_w \times \mathbb{Z}_h}$

There's a more general notion of algebras & coalgebras.

ANY endofunctor $F : \mathcal{C} \to \mathcal{C}$ has a notion:

- of algebra $\alpha : FA \to A$.
- of coalgebra $\beta : A \to FA$.

Fact: These are actually also useful.

e.g. Take $F$ to be $S \longmapsto 1 + A \times S$.

$$\begin{tikzcd} 1 + A \times S \arrow[d, "\alpha"] \\ S \end{tikzcd}$$

Special ones: initial algebra, terminal coalgebra.

$1 + A \times List(A) \xrightarrow{\alpha} List(A)$

$1 \longrightarrow List(A)$ picks out the empty list

$A \times List(A) \longrightarrow List(A)$ adjoins a new el.

e.g. automata: Mealy machine

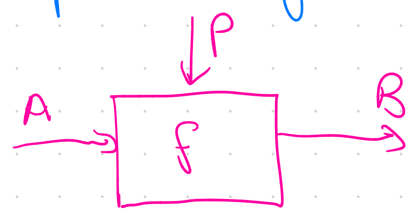
$$S \longmapsto (\underset{\text{inputs}}{I} \longrightarrow \underset{\text{outputs}}{O} \times S)$$

$$\begin{tikzcd}
\mathrm{Mealy}_{I,O} \arrow[d] \\
(I \longrightarrow O \times \mathrm{Mealy}_{I,O})
\end{tikzcd}$$

Idea: To do full arch. design, need 2-categories

They look at Para:

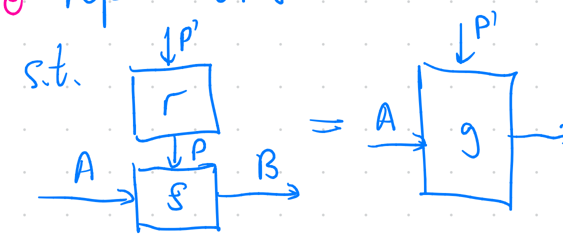
- objs are sets
- morph.s are parametric functions $A \xrightarrow{(P,f)} B$



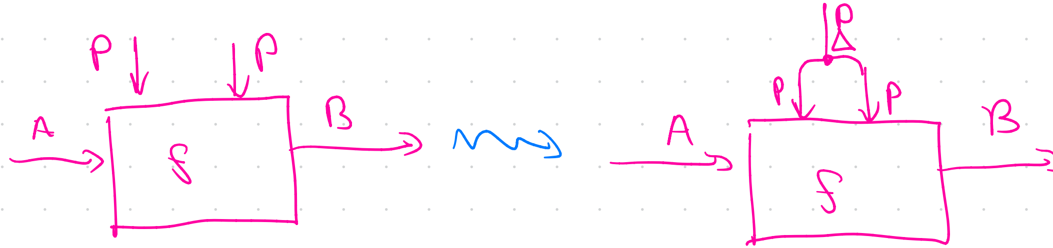

$f: P \times A \to B$.

- 2-morph.s
$$\begin{tikzcd} A \arrow[r, bend left, "{(P,f)}"] \arrow[r, bend right, "{(P',g)}"'] \arrow[r, phantom, "\Downarrow"] & B \end{tikzcd}$$
are reparametrisations $r: P' \to P$

s.t.



## Weight tying:




**Idea:** Rather than have monads on Set, work in Para & hopefully we can get alg. hom.s that actually resemble known arch.s.

2 things: 1. You need 2-monads

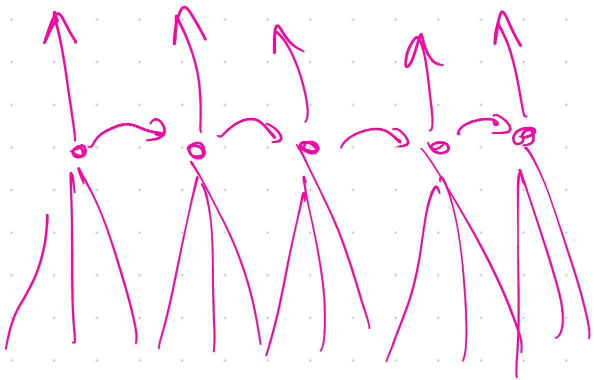
2. Commutativity is complex

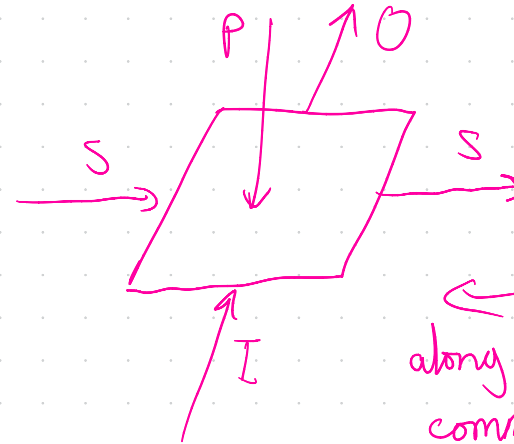
comm. pseudo-comm. lax oplax/colax

e.g. Mealy machine.

$$\begin{array}{c} S \\ \Big\downarrow (P,\ \text{cell}) \\ (I \to O \times S) \end{array}$$

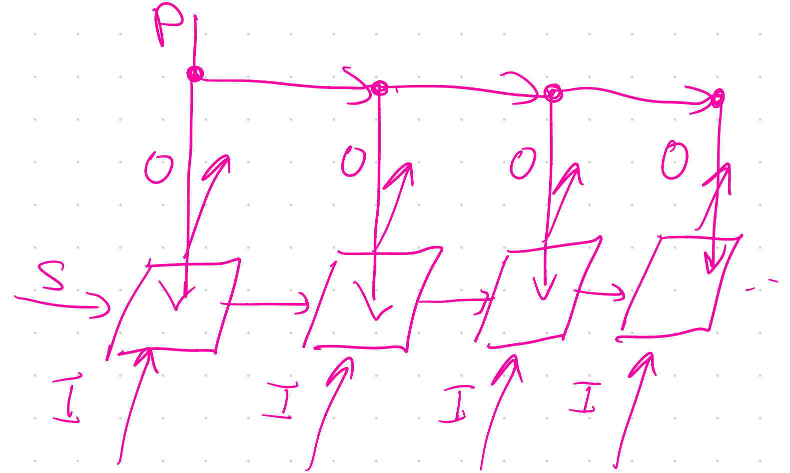
$$S \times P \times I \to O \times S$$






$\updownarrow$ between layers.

$\leftrightarrow$ along a common layer




$$\begin{tikzcd}
X \arrow[r, "{(P, fn)}"] \arrow[d, "{(P, n)}"'] & \mathrm{Mealy}_{O,I} \arrow[d, "\gamma(next)"] \arrow[dl, Rightarrow, "\Delta_X"'] \\
(I \to O \times X) \arrow[r, "{\mathrm{Para}(I \to O \times (-))(P, fn)}"'] & (I \to O \times \mathrm{Mealy}_{O,I})
\end{tikzcd}$$