Categories of Neural Networks

Michael Moy $\in$

Robert Cardona $\in$

Alan Hylton $\in$

Goal

Find a way to look under the hood of a given neural network

Category Theory

A *category* is a collection of related objects...

Category Theory

A *category* is a collection of related objects...

- e.g. the category of all sets
- e.g. the category of all (real) vector spaces

Category Theory

A *category* is a collection of related objects...

- e.g. the category of all sets
- e.g. the category of all (real) vector spaces

...along with a collection of arrows between objects

- e.g. for sets X and Y, all functions from $X \to Y$
- e.g. for vector spaces W and V, all linear transformations from $W \to V$

Definition

A *category* $\mathcal{C}$ has

- Objects $\mathrm{Ob}(\mathcal{C})$
- For $A, B \in \mathrm{Ob}(\mathcal{C})$, all morphisms $A \to B$; denoted $\hom(A, B)$

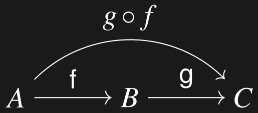
Definition

A *category* $\mathcal{C}$ has

- Objects $\mathrm{Ob}(\mathcal{C})$
- For $A, B \in \mathrm{Ob}(\mathcal{C})$, all morphisms $A \to B$; denoted $\hom(A, B)$

Composition is allowed: if there is a relation $f : A \to B$ and $g : B \to C$, then

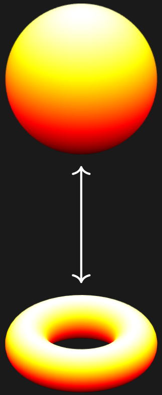
Then $g \circ f \in \hom(A, C)$



Usage I

Categories form a system out of related objects and their morphisms

Geometric

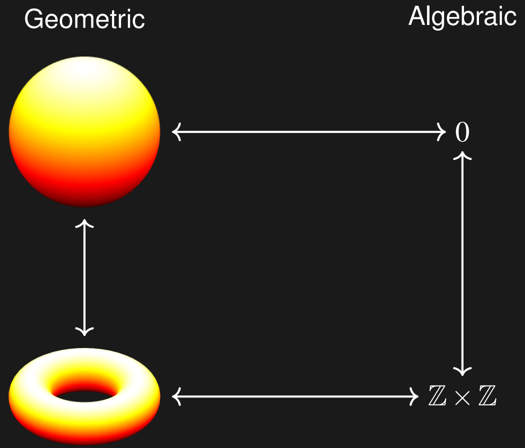


Algebraic

$$0 \updownarrow \mathbb{Z} \times \mathbb{Z}$$

Usage II

Categories can be mapped to other categories using *functors*



Goal restated

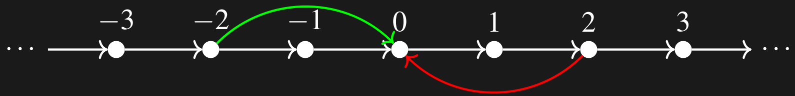
Use category theory to study neural networks

Example 1

Let $\mathcal{C}$ have integers as objects - $\{\ldots,-3,-2,-1,0,1,2,3,\ldots\}$

For any two integers x and y, define

$$\hom(x,y) = \begin{cases} \phi_{xy} & x \leq y \\ \varnothing & \text{else} \end{cases}$$



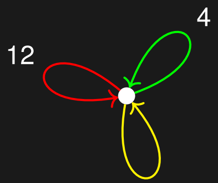
Example 2

Let $\mathcal{C}$ have one object $\bullet$

The morphisms from $\bullet$ to itself are

$$\hom(\bullet, \bullet) = \mathbb{Z}$$

Composition is given by addition



$$12 \circ 4 = 16$$

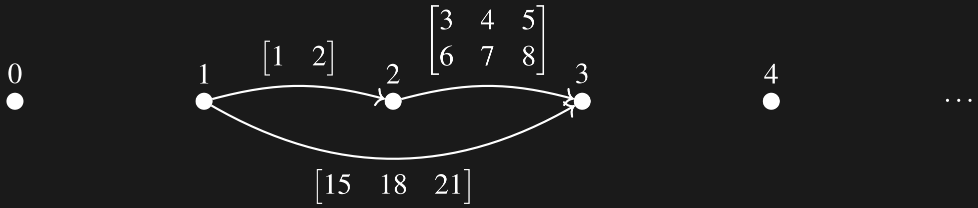
Example 3

Let $\mathcal{C}$ have natural numbers as objects - $\{0, 1, 2, 3, \dots\}$

The morphisms from m to n are

$$\hom(m, n) = \text{all } m \times n \text{ matrices}$$

Arrow composition is given by matrix multiplication

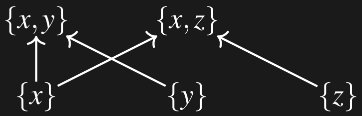


Example 4

Consider the power set of $\{x, y, z\}$: $\{\varnothing, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$

Order by set inclusion to make a category: for any elements x, y define

$$\text{hom}(x, y) = \begin{cases} \phi_{xy} & x \subseteq y \\ \varnothing & \text{else} \end{cases}$$

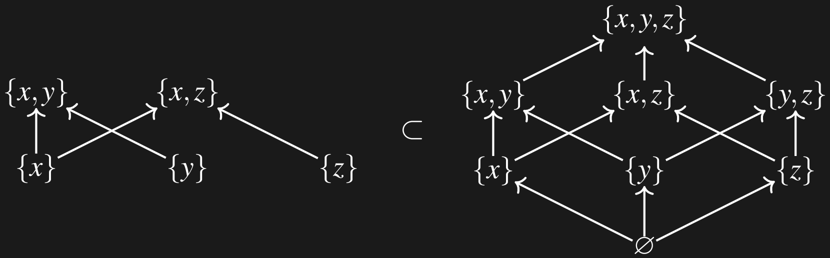


Example 4

Consider the power set of $\{x, y, z\}$: $\{\varnothing, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$

Order by set inclusion to make a category: for any elements x, y define

$$\hom(x, y) = \begin{cases} \phi_{xy} & x \subseteq y \\ \varnothing & \text{else} \end{cases}$$

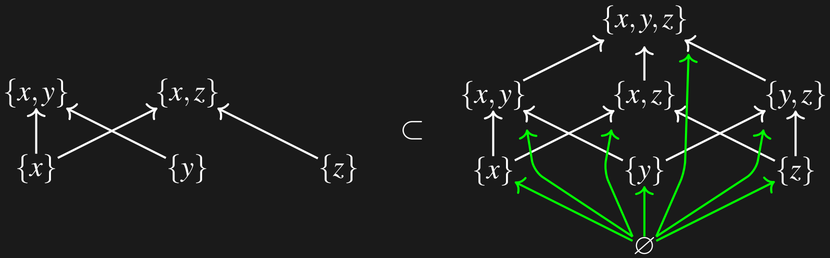


Example 4

Consider the power set of $\{x,y,z\}$: $\{\varnothing,\{x\},\{y\},\{z\},\{x,y\},\{x,z\},\{y,z\},\{x,y,z\}\}$

Order by set inclusion to make a category: for any elements x,y define

$$\hom(x,y) = \begin{cases} \phi_{xy} & x \subseteq y \\ \varnothing & \text{else} \end{cases}$$



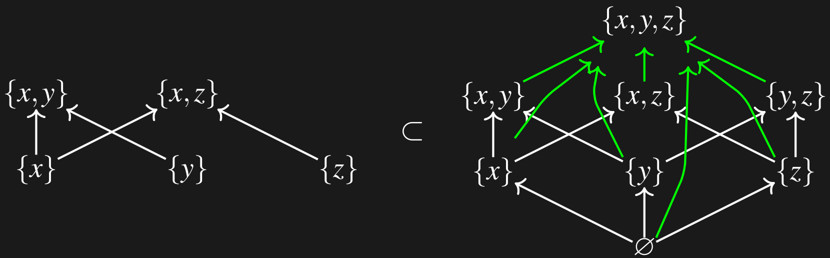
$\varnothing$ is the *initial object*

Example 4

Consider the power set of $\{x, y, z\}$: $\{\varnothing, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$

Order by set inclusion to make a category: for any elements x, y define

$$\hom(x, y) = \begin{cases} \phi_{xy} & x \subseteq y \\ \varnothing & \text{else} \end{cases}$$



$\{x, y, z\}$ is the *terminal object*

First cut

We can define a *category of neural networks* **NNet**

- Objects: natural numbers
- Morphisms: $\hom(m,n)$ is all neural networks with m inputs and n outputs
- Composition is concatenation where it makes sense

NNet has enough structure to define back propagation categorically!

(but I want more)

Our approach

A *neural network of length l* is a sequence of functions

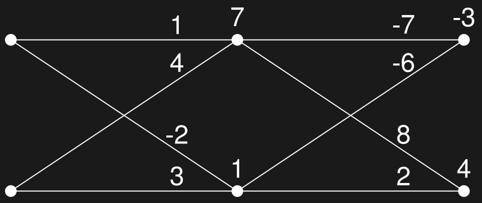
$$\left(\mathbb{R}^{n_0} \xrightarrow{N_0} \mathbb{R}^{n_1} \xrightarrow{N_1} \dots \xrightarrow{N_{l-1}} \mathbb{R}^{n_l}\right)$$

The functions N_i will be referred to as *layer functions of N*.

$$N_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i+1}} \text{ by } x \mapsto \sigma(Ax+b)$$

Notation: we use σ for activation functions

Example



$$\mathbb{R}^2 \xrightarrow{N_0} \mathbb{R}^2 \xrightarrow{N_1} \mathbb{R}^2$$

$$N_0(x) = \sigma\left(\begin{pmatrix} 1 & 4 \\ -2 & 3 \end{pmatrix} x + \begin{pmatrix} 7 \\ 1 \end{pmatrix}\right)$$

$$N_1(x) = \sigma\left(\begin{pmatrix} -7 & -6 \\ 8 & 2 \end{pmatrix} x + \begin{pmatrix} -3 \\ 4 \end{pmatrix}\right)$$

Note: in **NNet** this neural network is an arrow $2 \to 2$

Morphisms

$N = (N_0, N_1, \ldots, N_{l-1})$ and $M = (M_0, M_1, \ldots, M_{l-1})$ are neural networks of length l

A *morphism* $f : N \to M$ is a sequence of functions $(f_0, f_1, \ldots, f_l)$ such that

$$f_k \circ N_{k-1} \circ \cdots \circ N_1 \circ N_0 = M_{k-1} \circ M_{k-2} \circ \cdots \circ M_0 \circ f_0 \text{ for all } 1 \leq k \leq l$$

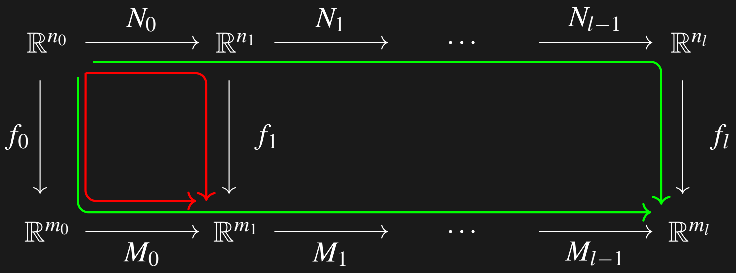
$$\begin{array}{ccccccc}
\mathbb{R}^{n_0} & \xrightarrow{N_0} & \mathbb{R}^{n_1} & \xrightarrow{N_1} & \cdots & \xrightarrow{N_{l-1}} & \mathbb{R}^{n_l} \\
\downarrow f_0 & & \downarrow f_1 & & & & \downarrow f_l \\
\mathbb{R}^{m_0} & \xrightarrow[M_0]{} & \mathbb{R}^{m_1} & \xrightarrow[M_1]{} & \cdots & \xrightarrow[M_{l-1}]{} & \mathbb{R}^{m_l}
\end{array}$$

Morphisms

$N = (N_0, N_1, \dots, N_{l-1})$ and $M = (M_0, M_1, \dots, M_{l-1})$ are neural networks of length l

A *morphism* $f : N \to M$ is a sequence of functions $(f_0, f_1, \dots, f_l)$ such that

$$f_k \circ N_{k-1} \circ \cdots \circ N_1 \circ N_0 = M_{k-1} \circ M_{k-2} \circ \cdots \circ M_0 \circ f_0 \text{ for all } 1 \leq k \leq l$$



A new hope

We can form a new category whose

- objects are neural networks of length l
- morphisms are appropriate sequences $(f_0, \ldots, f_l)$

What's left? Composition

This is done layer by layer:

$$(f_0, f_1, \ldots, f_l) \circ (g_0, g_1, \ldots, g_l) = (f_0 \circ g_0, f_1 \circ g_1, \ldots, f_l \circ g_l).$$

Categories of neural nets

Recall layer functions:

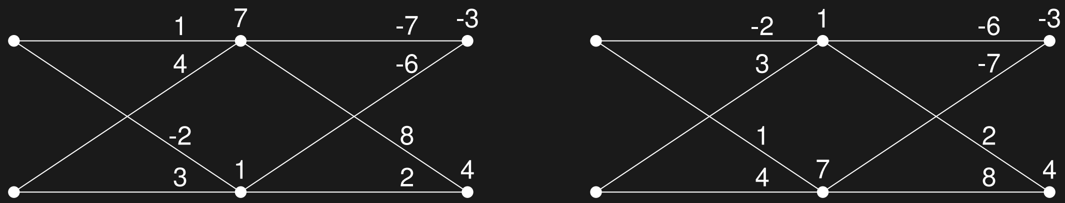
$$N_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i+1}} \text{ by } x \mapsto \sigma(Ax+b)$$

We can make several categories by picking the activation function σ:

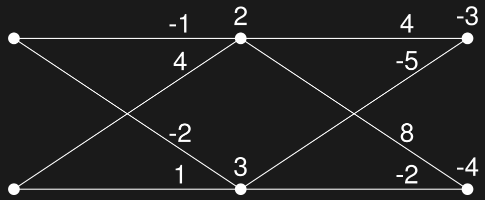
AffineNet$_l$	N_i required to be an affine function followed by any activation function
ReluAffineNet$_l$	N_i required to be an affine function followed by ReLU activation function

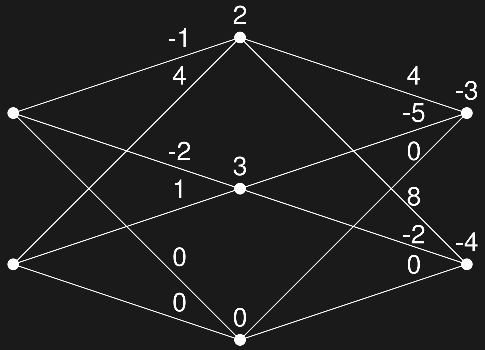
Category note: **ReluAffineNet$_l$** is a subcategory of **AffineNet$_l$**

Isomorphisms



Isomorphisms II





Terminal objects

The *terminal network* T is given by

$$\mathbb{R}^0 \xrightarrow{0} \mathbb{R}^0 \xrightarrow{0} \cdots \xrightarrow{0} \mathbb{R}^0$$

Let N be an object of $\textbf{AffineNet}_l$ or $\textbf{ReluAffineNet}_l$

There is a unique morphism $N \to T$

Initial objects

The *initial network* I is given by

$$\varnothing \to \varnothing \to \cdots \to \varnothing$$

Let N be an object of $\mathbf{AffineNet}_l$ or $\mathbf{ReluAffineNet}_l$

There is a unique morphism $I \to N$

Products

Let N and M be objects of **AffineNet**$_l$ or **ReluAffineNet**$_l$

Their *product* $N \times M$ is given by

$$\mathbb{R}^{n_0} \times \mathbb{R}^{m_0} \xrightarrow{N_0 \times M_0} \mathbb{R}^{n_1} \times \mathbb{R}^{m_1} \xrightarrow{N_1 \times M_1} \cdots \xrightarrow{N_{l-1} \times M_{l-1}} \mathbb{R}^{n_l} \times \mathbb{R}^{m_l}$$

$N \times M$ is in **AffineNet**$_l$ or **ReluAffineNet**$_l$

Ideas

1. Extend these definitions to allow networks of different lengths
2. Find interesting morphisms between realistic networks
3. Explore categorical constructions, such as equalizers
4. Use subobject language to describe subnetworks
5. Employ machinery of other categories using functors