Towards a Categorical Foundation of Deep Learning: A Survey

Una rassegna di approcci categorici al *deep learning*

Francesco Riccardo Crescenzi

11 settembre 2024

Alma mater studiorum - Università di Bologna
CdL in Matematica

We are in an AI summer, but is winter coming?

Problemi con il deep learning

Mancano fondamenta teoriche:

- approcci *ad hoc*

Problemi con il deep learning

Mancano fondamenta teoriche:

- approcci *ad hoc*
- complessità fine a se stessa

Problemi con il deep learning

Mancano fondamenta teoriche:

- approcci *ad hoc*
- complessità fine a se stessa
- assenza di garanzie di correttezza

Problemi con il deep learning

La ricerca viene rallentata da:

- *research debt*

Problemi con il deep learning

La ricerca viene rallentata da:

- *research debt*
- mancata replicabilità

Teoria delle categorie:

una lingua franca della matematica

Teoria delle categorie

La teoria delle categorie studia strutture e relazioni, e può essere vista come un'estensione del celebre Erlangen Programme.

Teoria delle categorie:

una lingua franca delle scienze

Teoria delle categorie applicata

La teoria delle categorie può essere applicata con successo anche in fisica, informatica, chimica... ovunque ci sia **composizionalità**.

Teoria delle categorie applicata

- ottiche parametriche

Teoria delle categorie applicata

- ottiche parametriche
- (co)algebre categoriche

Teoria delle categorie applicata

- ottiche parametriche
- (co)algebre categoriche
- *integral transforms*

Teoria delle categorie applicata

- ottiche parametriche
- (co)algebre categoriche
- *integral transforms*
- *functor learning*

Teoria delle categorie applicata

- ottiche parametriche
- (co)algebre categoriche
- *integral transforms*
- *functor learning*
- *string diagrams*

Lenti parametriche

per modellare il gradient-based learning

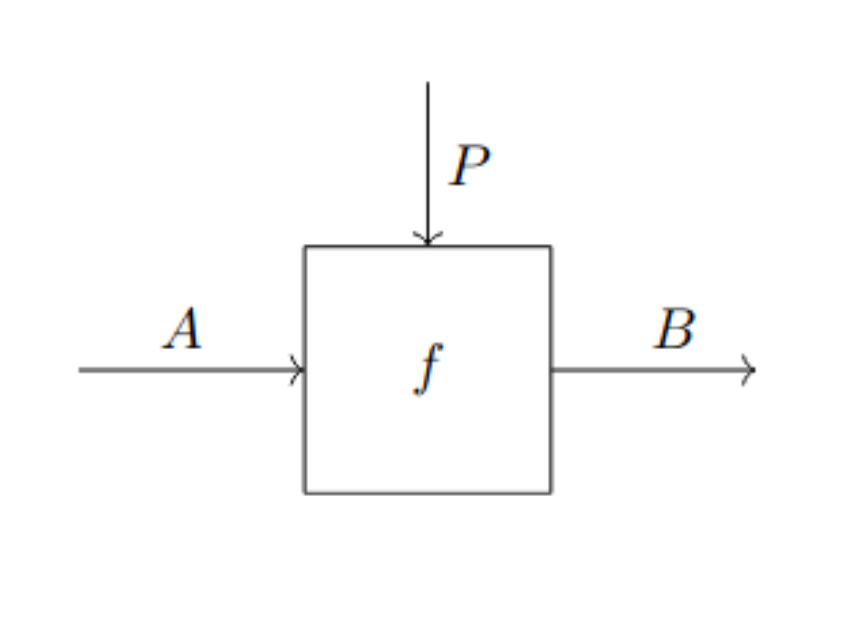
Gradient-based learning con lenti parametriche

DEFINIZIONE: Il costrutto Para

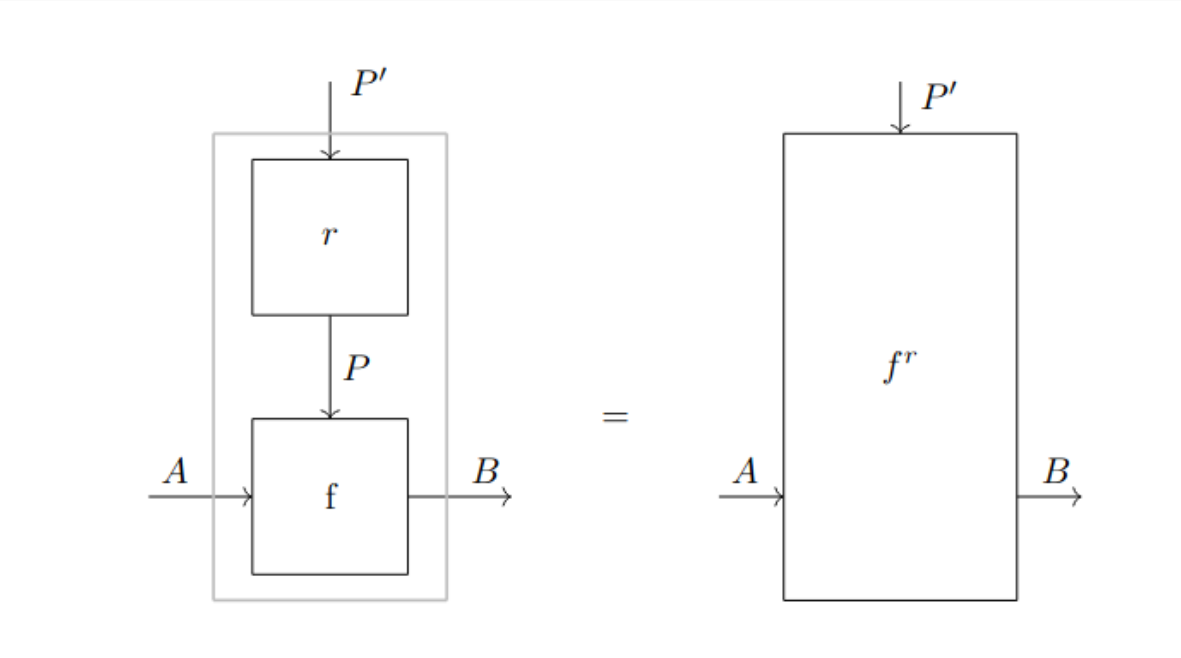
Sia $(\mathcal{C}, I, \otimes)$ una categoria monoidale strettamente simmetrica. Allora, $\mathbf{Para}_{\otimes}(\mathcal{C})$ è la 2-categoria definita come segue.

- Le 0-celle sono oggetti di $\mathcal{C}$.
- Le 1-cells sono coppie $(P, f) : A \to B$, dove $P : \mathcal{C}$ e $f : P \otimes A \to B$.
- The 2-celle sono $r : (P, f) \Rightarrow (Q, g)$, dove $r : P \to Q$ è un morfismo in $\mathcal{C}$ che rispetta certe condizioni di naturalità.

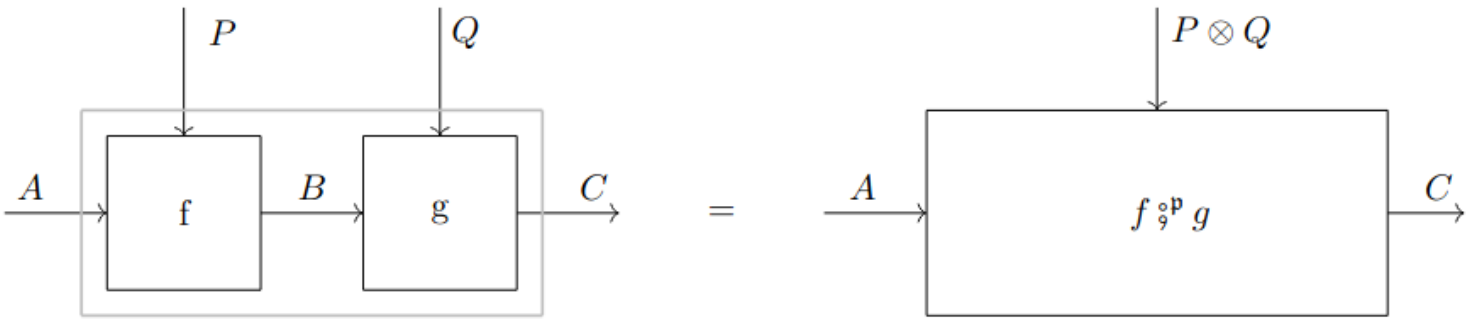
Gradient-based learning con lenti parametriche



Gradient-based learning con lenti parametriche



Gradient-based learning con lenti parametriche



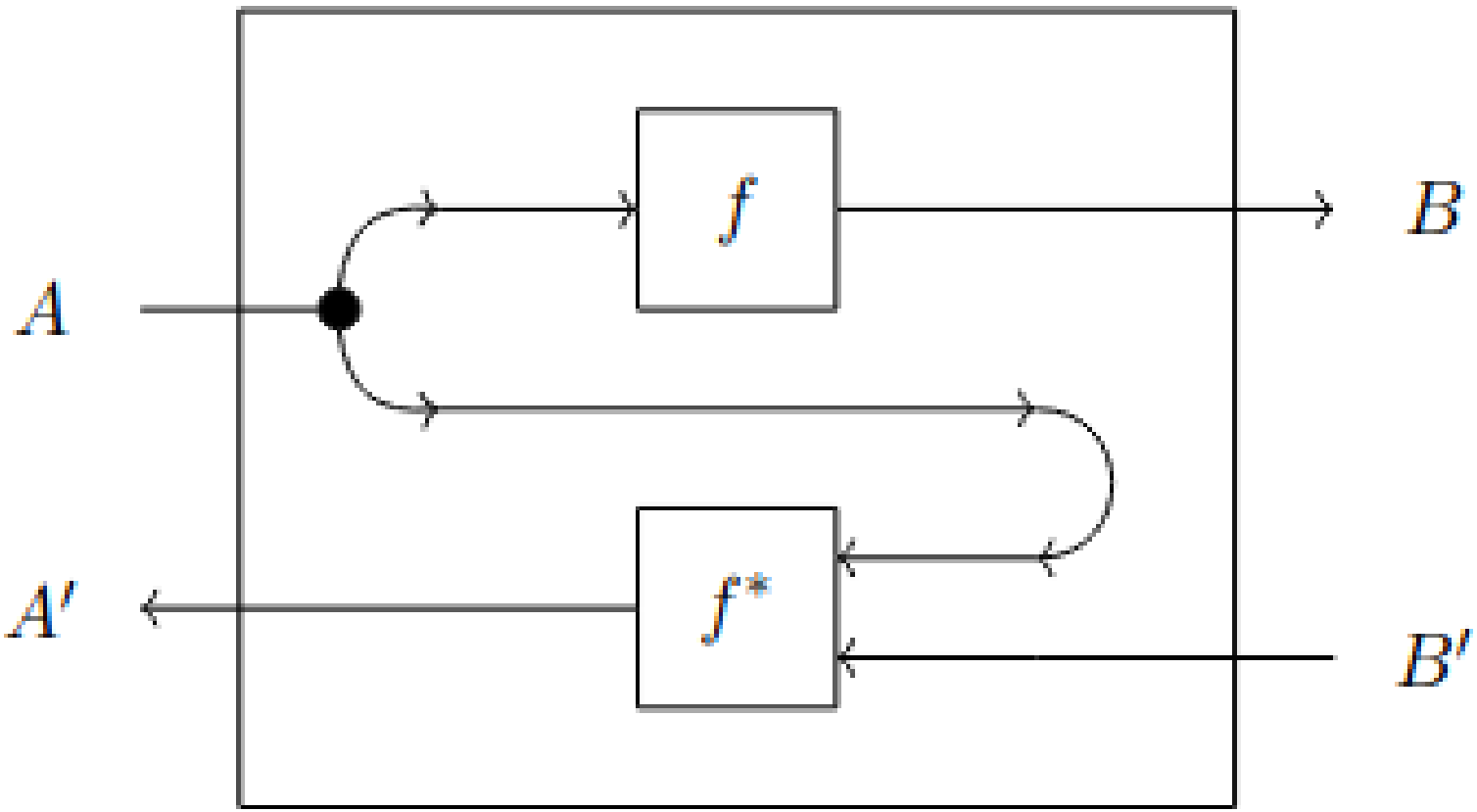
Gradient-based learning con lenti parametriche

DEFINIZIONE: Il costrutto Lens

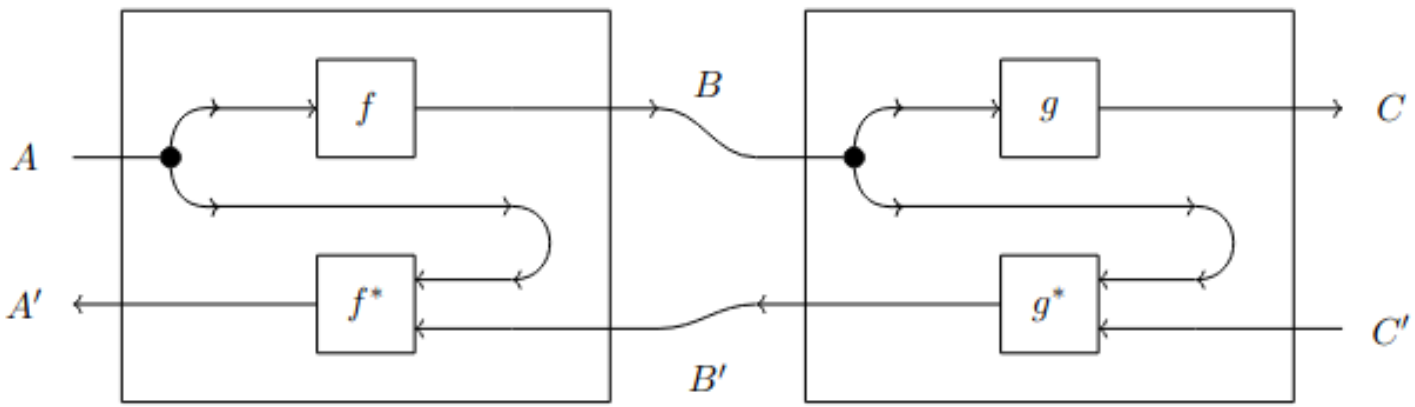
Sia $(\mathcal{C}, 1, \times)$ una categoria Cartesiana. Allora, **Lens**$(\mathcal{C})$ è la categoria definita come segue.

- Un oggetto di **Lens**$(\mathcal{C})$ è una coppia $\binom{A}{A'}$ di oggetti di $\mathcal{C}$.
- Un morfismo $\binom{A}{A'} \to \binom{B}{B'}$ (anche chiamato lente) è una coppia $\binom{f}{f'}$ di morfismi di $\mathcal{C}$ tali che $f : A \to B$ and $f' : A \times B' \to A'$. La mappa f è nota come *forward pass* della lente, mentre la mappa f' è nota come *backward pass*.

Gradient-based learning con lenti parametriche



Gradient-based learning con lenti parametriche



Gradient-based learning con lenti parametriche

DEFINIZIONE: Cartesian reverse differential category

Una *Cartesian reverse differential category* (CRDC) $\mathcal{C}$ è una categoria Cartesiana con una struttura additiva dove è definito un operatore differenziale R che ha le proprietà di una *reverse derivative*.

ESEMPIO: Smooth

Consideriamo **Smooth**, ovvero la categoria degli spazi Euclidei e delle funzioni liscie. **Smooth** è una CRDC rispetto all'operatore

$$\mathrm{R}[f] : (x, y) \mapsto \mathcal{J}_f(x)^T y.$$

Gradient-based learning con lenti parametriche

DEFINIZIONE: Lenti con backward pass additivo

Sia $\mathcal{C}$ una CRDC. Allora, definiamo la sottocategoria $\mathbf{Lens}_A(\mathcal{C})$ di $\mathbf{Lens}_A(\mathcal{C})$, i cui oggetti sono coppie $\binom{A}{A'}$ e i cui morfismi hanno la forma $\begin{pmatrix} f \\ \mathrm{R}[f] \end{pmatrix}$.

TEOREMA: Struttura cartesiana di $\mathbf{Lens}_A(\mathcal{C})$

La struttura

$$I = \binom{1}{1}, \quad \binom{A}{A} \otimes \binom{B}{B} = \binom{A \times B}{A \times B}$$

definita su $\mathbf{Lens}_A(\mathcal{C})$ è Cartesiana.

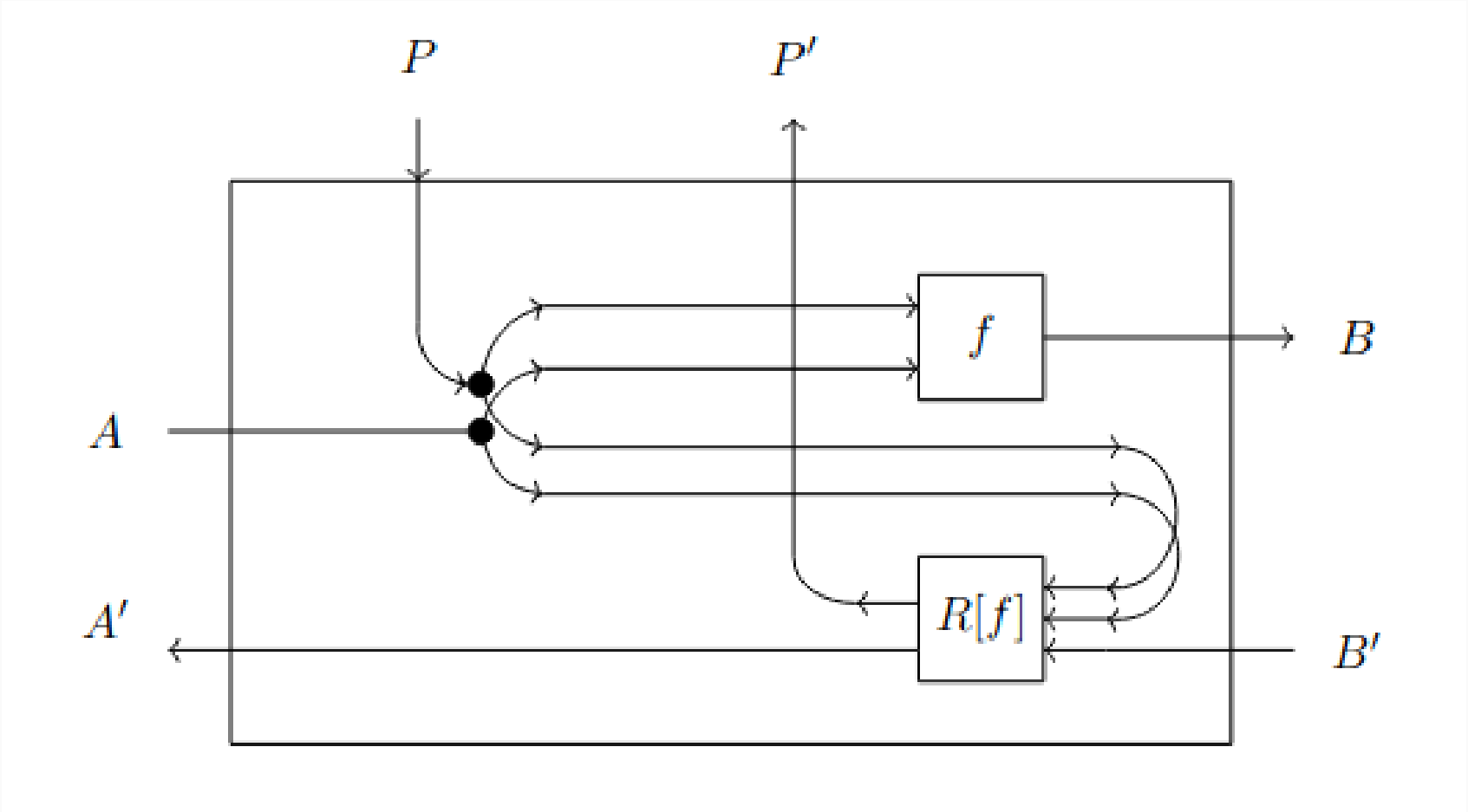
Gradient-based learning con lenti parametriche

DEFINIZIONE: Lenti parametriche

Sia $\mathcal{C}$ una CRDC. Allora, definiamo la categoria delle lenti parametriche su $\mathcal{C}$ come

$$\mathbf{Para}_{\otimes}(\mathbf{Lens}_{A}(\mathcal{C})).$$

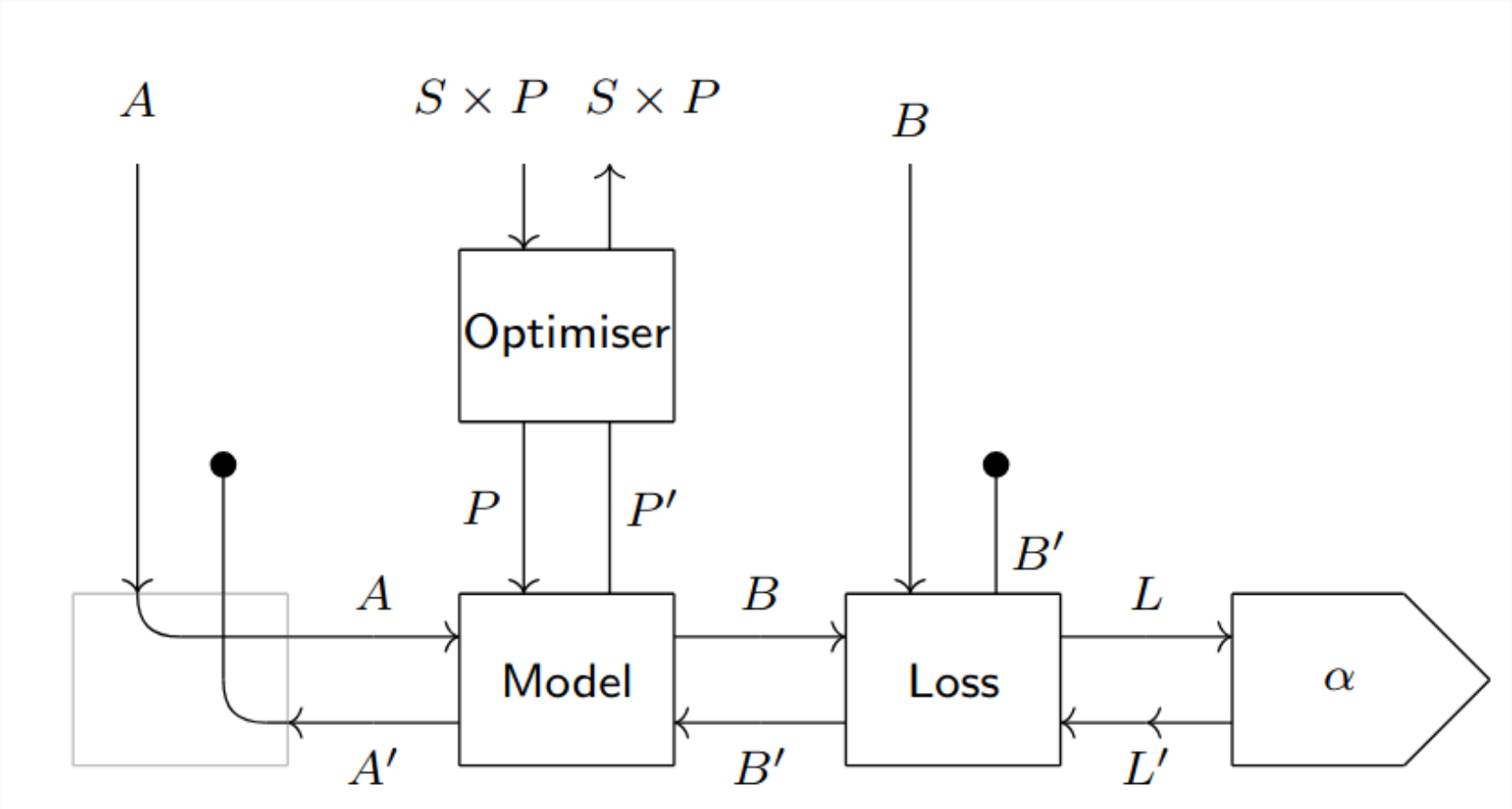
Gradient-based learning con lenti parametriche



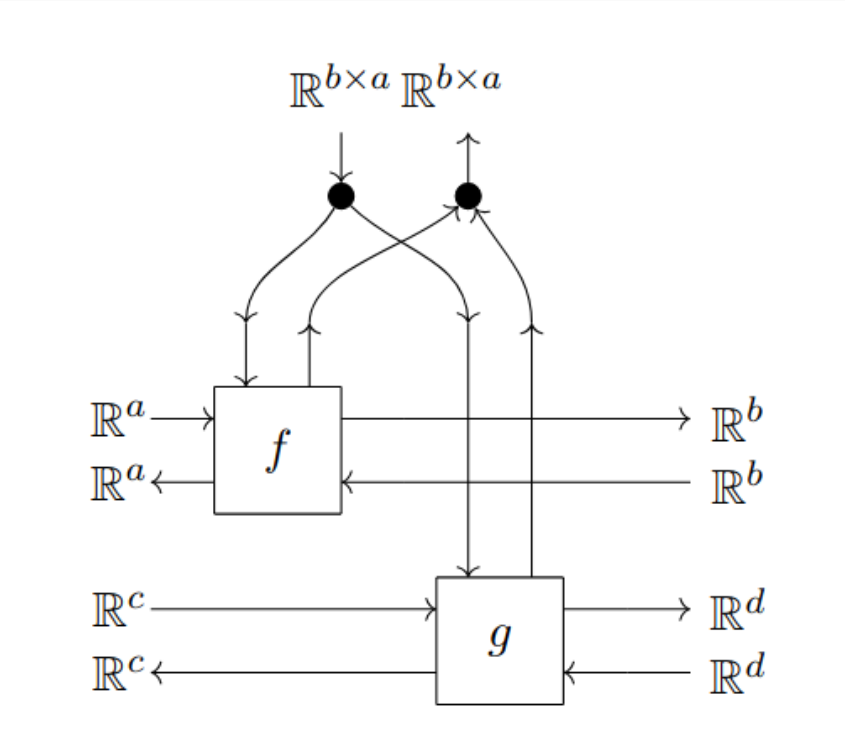
Gradient-based learning con lenti parametriche

Le lenti parametriche in $\mathbf{Para}_{\otimes}(\mathbf{Lens}_{A}(\mathcal{C}))$ supportano la *automatic differentiation* e possono essere utilizzate per implementare il *gradient-based learning*.

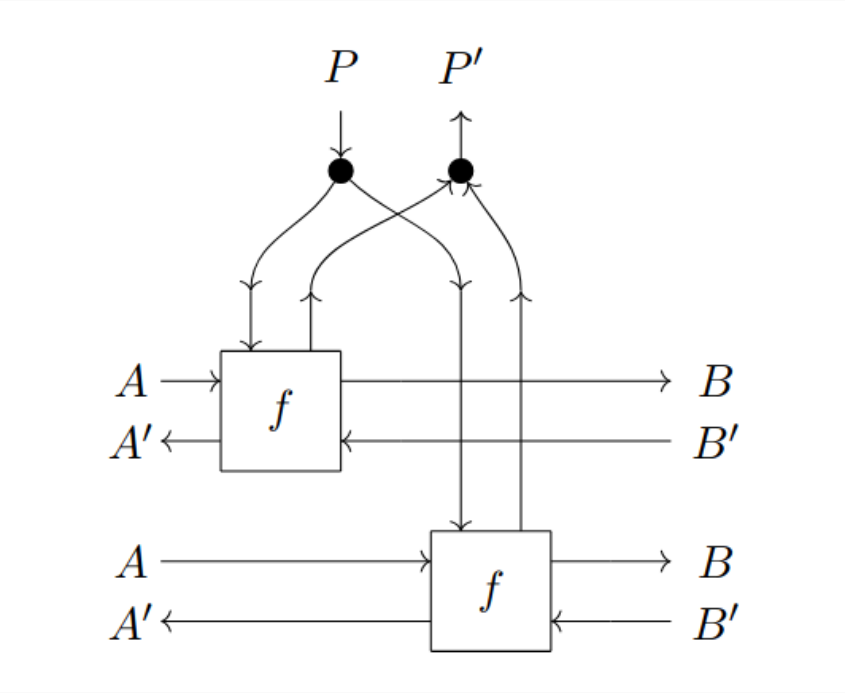
Gradient-based learning con lenti parametriche



Gradient-based learning con lenti parametriche



Gradient-based learning con lenti parametriche



Categorical deep learning:

(co)algebre categoriche
come teoria delle architetture

Dal GDL al CDL

Geometric deep learning

Il *geometric deep learning* è una teoria delle architetture di reti neurali che imita l'*Erlangen Programme*, organizzando le architetture in base al concetto di equivarianza rispetto ad azioni di gruppi.

Dal GDL al CDL

DEFINIZIONE: Funzione equivariante

Sia G be un gruppo e siano $(S, \cdot)$ e $(T, *)$ azioni di G. Una funzione $f : S \to T$ è equivariante rispetto a tali azioni se

$$f(g \cdot s) = g * f(s),$$

per ogni $s \in \mathcal{S}$ e per ogni $g \in \mathcal{G}$.

ESEMPIO

I convolutional layers delle reti neurali rappresentano mappe invarianti rispetto a traslazioni.

Dal GDL al CDL

Categorical deep learning

Il *categorical deep learning* è una teoria delle architetture di reti neurali che generalizza il GDL, organizzando le architetture in base al concetto di omomorfismo di (co)algebre categoriche.

Dal GDL al CDL

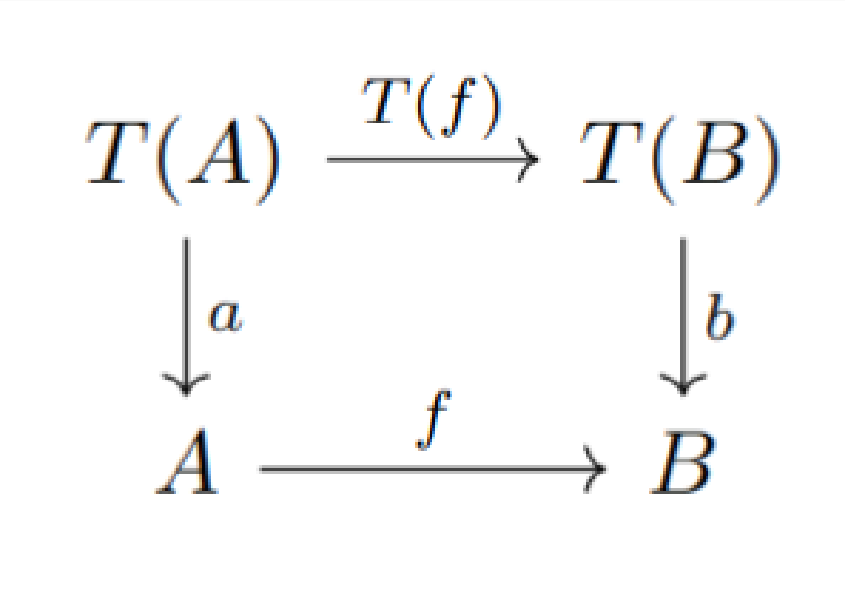
DEFINIZIONE: Algebra su un endofuntore

Sia $F : \mathcal{C} \to \mathcal{C}$ un endofuntore. Un'algebra su F è una coppia (A, a) dove A è un oggetto di $\mathcal{C}$ e $a : F(A) \to A$ è un morfismo in $\mathcal{C}$.

DEFINIZIONE: Omomorfismo di algebre

Siano (A, a) e (B, b) algebre sollo stesso endofuntore $F : \mathcal{C} \to \mathcal{C}$. Un omomorfismo di algebre $(A, a) \to (B, b)$ è un morfismo $f : A \to B$ in $\mathcal{C}$ tale che $F(f) \fatsemi b = a \fatsemi f$.

Dal GDL al CDL



Dal GDL al CDL

Il CDL generalizza il GDL poiché le azioni di un grouppo G si possono definire come algebre su una monade, e le mappe invarianti si recuperano come omomorfismi tra queste algebre.

DEFINIZIONE: Monade delle azioni di G

Consideriamo l'endofuntore $G \times - : \mathbf{Set} \to \mathbf{Set}$ che mappa $A \mapsto G \times A$ e $f \mapsto G \times f$. La monade delle azioni di G è definita dotando l'endofuntore delle transformazioni naturali di $\mu_A : (g, h, a) \mapsto (gh, a)$ e $\eta_A : a \mapsto (e, a)$.

Dal GDL al CDL

Il CDL costruisce collega algoritmi e strutture dell'informatica classica con le reti neurali.

Dal GDL al CDL

ESEMPIO: Liste

Sia A un insieme. Consideriamo l'endofuntore $1 + A \times -$ su **Set**.
Sia $\mathrm{List}(A)$ l'insieme delle liste di elementi di A. Allora, se
$\mathrm{Nil} : 1 \to \mathit{List}(A)$ mappa l'unico oggetto di 1 alla lista vuota e
$\mathrm{Cons} : A \times \mathrm{List}(A) \to \mathrm{List}(A)$ aggiunge un elemento a una lista,
$(\mathrm{List}(A), [\mathrm{Nil}, \mathrm{Cons}])$, è un algebra su $1 + A \times -$.

ESEMPIO: List folds

Consideriamo due algebre $(\mathsf{List}(A), [\mathsf{Nil}, \mathsf{Cons}])$ e $(Z, [r_0, r_1])$ su $1 + A \times -$. Un omomorfismo $f : \mathsf{List}(A) \to Z$ tra queste due algebre deve soddisfare

$$f(\mathsf{Nil}) = r_0,$$
$$f(\mathsf{Cons}(a, l)) = r_1(a, f(l)).$$

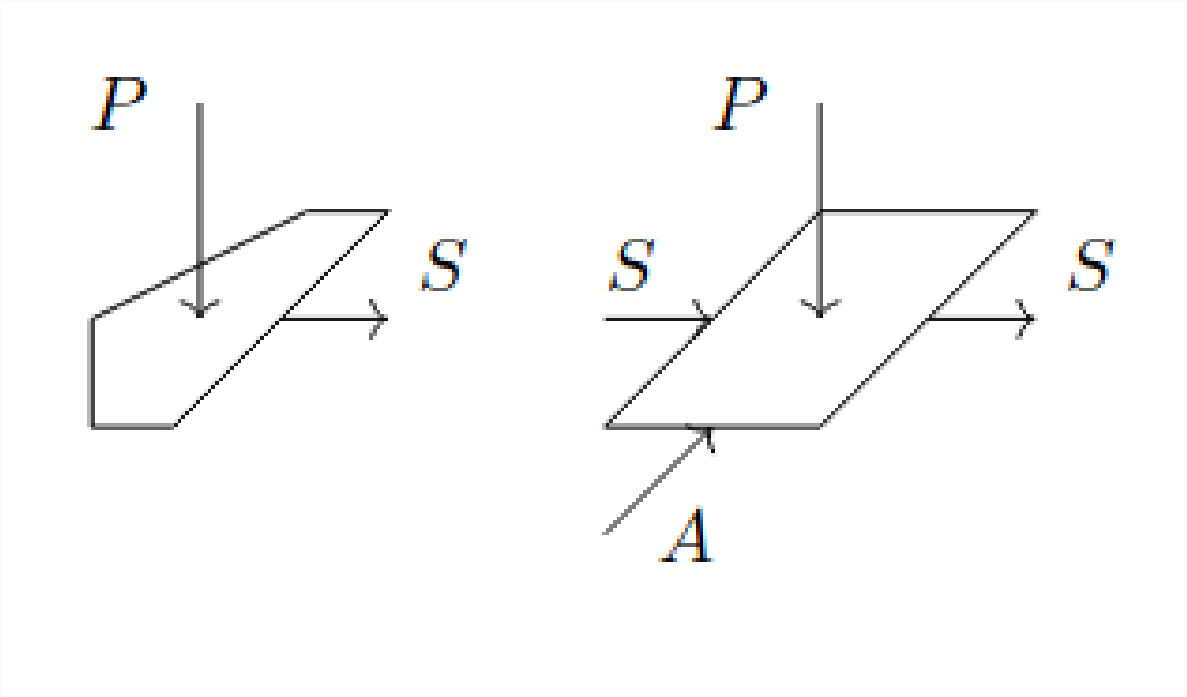
Hence, f è necessariamente un *fold* che riduce liste di elementi di A a singoli elementi di Z.

Dal GDL al CDL

ESEMPIO: Una cella di un folding RNN

Consideriamo l'endofuntore $1 + A \times X$: e la struttura cartesiana $(1, \times)$ su **Set**. Su questo funtore, può essere costruito un 2-funtore $\mathbf{Para}(1 + A \times X) : \mathbf{Para}_{\bullet}(\mathbf{Set}) \to \mathbf{Para}_{\bullet}(\mathbf{Set})$. Consideriamo un algebra $(S, (P, \mathrm{Cell}))$ su tale funtore. Tramite l'isomorfismo $P \times (1 + A \times X) \cong P + P \times A \times X$, deduciamo che $\mathrm{Cell} = [\mathrm{Cell}_0, \mathrm{Cell}_1]$, dove $\mathrm{Cell}_0 : P \to S$ e $\mathrm{Cell}_1 : P \times A \times S \to S$. Le funzioni Cell_0 e Cell_1 si possono interpretare come celle di un folding recurrent neural network.

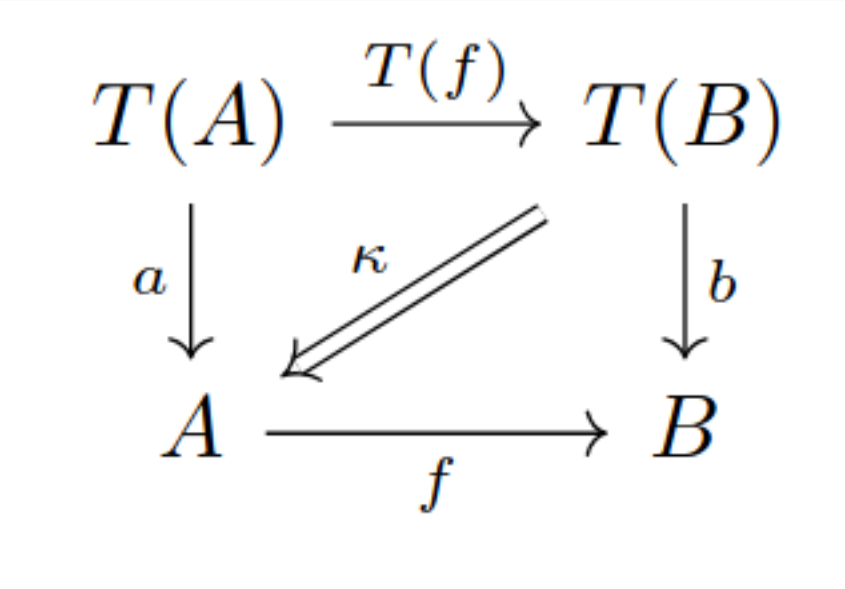
Dal GDL al CDL



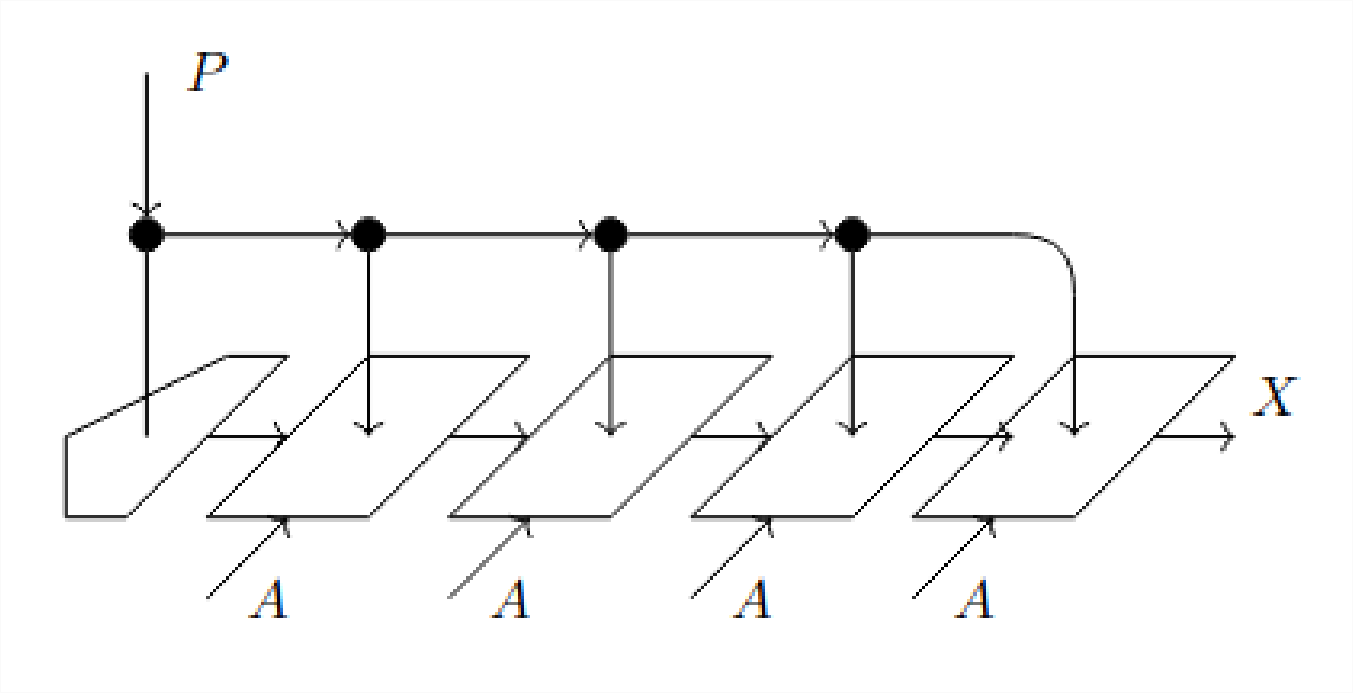
ESEMPIO: Unrolling di un folding RNN

Consideriamo le due algebre $(\mathrm{List}(A), [\mathrm{Nil}, \mathrm{Cons}])$ e $(S, (P, \mathrm{Cell}))$ sull'endofuntore $\mathbf{Para}(1 + A \times X)$. Ora consideriamo un omomorfismo di algebre
$(P, f, \Delta_P) : (\mathrm{List}(A), [\mathrm{Nil}, \mathrm{Cons}]) \to (S, (P, \mathrm{Cell}))$. Si può dimostrare che una funzione f così definita è l'unrolling di un folding recurrent neural network. L'algebra $(\mathrm{List}(A), [\mathrm{Nil}, \mathrm{Cons}])$ fornisce gli input della rete neurale, mentre l'algebra $(S, (P, \mathrm{Cell}))$ fornisce le celle.

Dal GDL al CDL



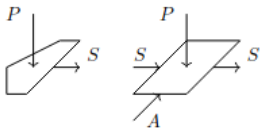
Dal GDL al CDL



Dal GDL al CDL

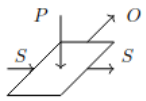
Folding recurrent neural network

$$1 + A \times S \xrightarrow{(P,\mathsf{cell}^{\mathsf{rcnt}})} S$$



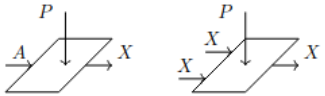
Unfolding recurrent neural network

$$S \xrightarrow{(P,\langle \mathsf{cell}_o,\mathsf{cell}_n \rangle)} O \times S$$



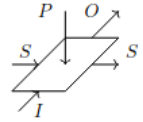
Recursive neural network

$$A + S^2 \xrightarrow{(P,\mathsf{cell}^{\mathsf{rcsv}})} S$$



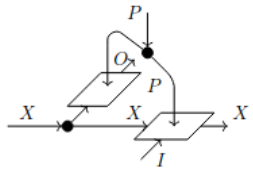
Full recurrent neural network

$$S \xrightarrow{(P,\mathsf{cell}^{\mathsf{Mealy}})} (I \to O \times S)$$



"Moore machine" neural network

$$S \xrightarrow{(P,\mathsf{cell}^{\mathsf{Moore}})} O \times (I \to S)$$



Functor learning:

modelli come funtori che sfruttano la struttura dei dati

Categorical representation learning

Categorical representation learning

Il *categorical representation learning* consiste nell'immergere funtorialmente una categoria $\mathcal{C}$ di dati in una categoria $\mathcal{R}$ di vettori latenti.

Grazie alla funtorialità dell'embedding, il *categorical representation learning* consente di preservare la struttura dei dati nello spazio latente.

Categorical representation learning

DEFINIZIONE: Struttura categorica dello spazio latente

tegoria i cui oggetti sono i vettori di $\mathbb{R}^n$ and tale che, per ogni u, v in $\mathcal{R}$, i morfismi $u \to v$ sono le matrici $M \in \mathbb{R}^{n \times n}$ tali che $v = Mu$. La composizione è l'ordinario prodotto riga per colonna e l'identità relativa a un generico $v \neq 0$ è $\mathrm{id}_v = \frac{vv^T}{|v|^2}$. L'identià relativa al vettore nullo è la matrice nulla.

Categorical representation learning

Data una categoria $\mathcal{C}$ di dati, il funtore di *embedding* $\mathcal{C} \to \mathcal{R}$ è realizzato da due **neural embedding layers**: il primo produce rappresentazioni degli elementi di $\mathcal{C}$ come vettori, mentre il secondo produce rappresentazioni dei morfismi di $\mathcal{C}$ come matrici.

Categorical representation learning

DEFINIZIONE: Negative sampling loss

La *objective function* utilizzata per addestrare i due *layers* è la *negative sampling loss*

$$\mathcal{L} = -\mathbb{E}_{(a,b)\sim p(a,b)} \left(\log P(a \to b) + \mathbb{E}_{b' \sim p(b')} \log(1 - P(a \to b')) \right),$$

dove la probabilità che sussista una relazione $a \to b$ è misurata come

$$P(a \to b) = \text{sigmoid} \left(F \left(\bigoplus_f v_a^T M_f v_b \right) \right).$$

Categorical representation learning

ESEMPIO: Traduzione non supervisionata

Se $\mathcal{C}$ e $\mathcal{D}$ sono database di formule chimiche in inglese e cinese, rispettivamente, possiamo usare il CRL per immergere le due categorie in $\mathcal{R}$ funtorialmente. Poi si può imparare un funtore $\mathcal{F} : \mathcal{R} \to \mathcal{R}$ che preservi la struttura categorica. Tale funtore opererà la traduzione.

Categorical representation learning

ESEMPIO: Traduzione non supervisionata

Se $\mathcal{C}$ e $\mathcal{D}$ sono database di formule chimiche in inglese e cinese, rispettivamente, possiamo usare il categorical representation learning per immergere le due categorie in $\mathcal{R}$ funtorialmente. Poi si può imparare un funtore $\mathcal{F} : \mathcal{R} \to \mathcal{R}$ che preservi la struttura categorica. Tale funtore effettuerà la traduzione. $\mathcal{F}$ può essere implementato come una matrice $V_{\mathcal{F}}$ che mappa

$$v \mapsto V_{\mathcal{F}} v,$$
$$M_f \mapsto V_{\mathcal{F}} M_f.$$

Categorical representation learning

ESEMPIO: Traduzione non supervisionata

La matrice può essere imparata minimizzando la *loss function*

$$\mathcal{L}_{\text{struc}} = \sum_{f} \| V_{\mathcal{F}} M_f - M_{\mathcal{F}(f)} V_{\mathcal{F}} \|^2,$$

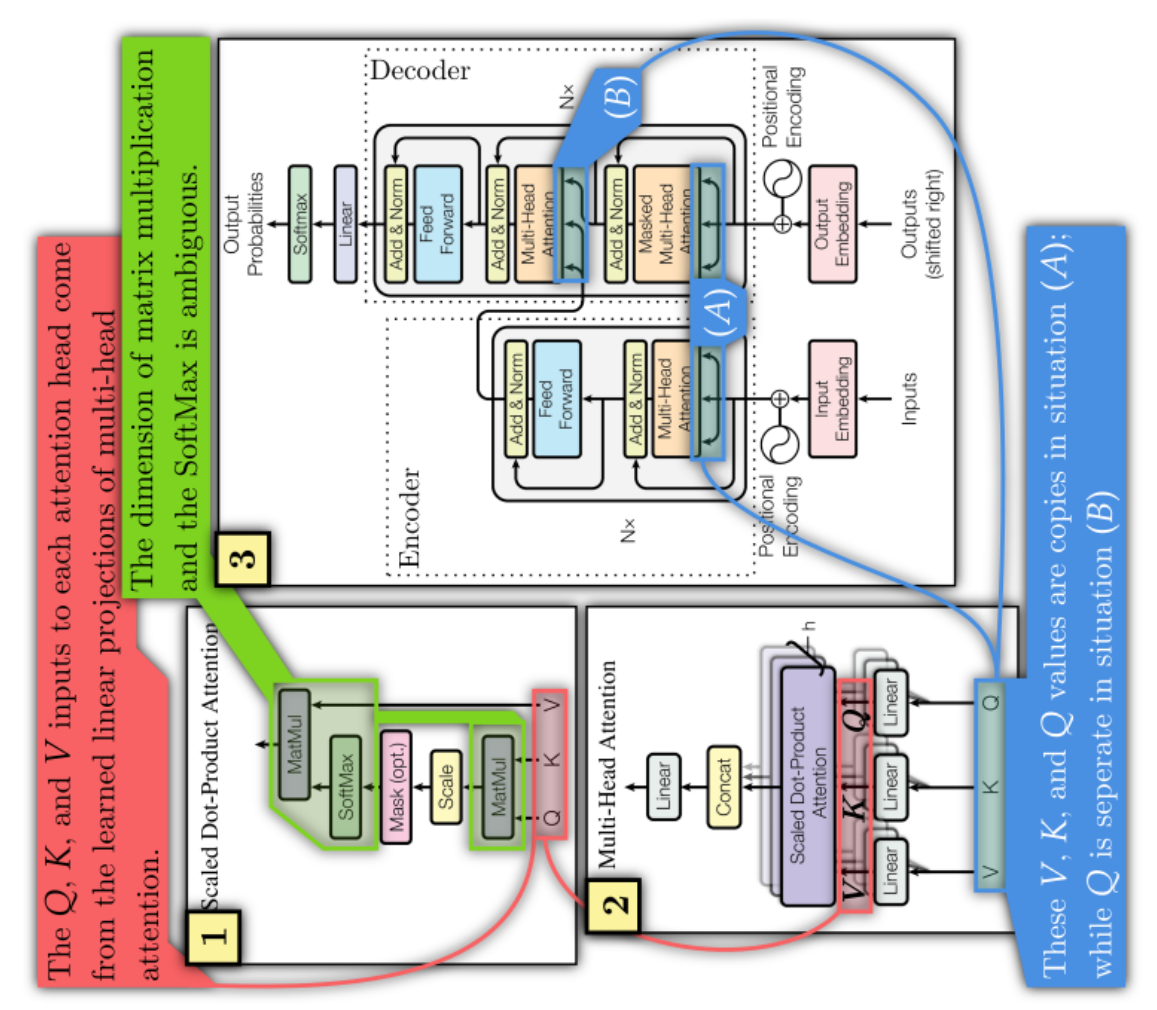
a cui si può aggiungere anche la *loss*

$$\mathcal{L}_{\text{align}} = \sum_{a \in A} \| V_{\mathcal{F}} v_a - v_{\mathcal{F}(a)} \|,$$

per dare parziale supervisione.

Neural circuit diagrams:
rappresentazioni dettagliate di architetture neurali

Neural circuit diagrams



Neural circuit diagrams

I diagrammi di *deep learning models* generalmente utilizzati nella letteratura scientifica sono inadeguati in quanto non riportano molti dettagli importanti ai fini dell'implementazione.

Neural circuit diagrams

I *monoidal string diagrams* usati in teoria delle categorie applicata sono inadeguati poiché non riescono a rappresentare funtori e trasformazioni naturali.

Neural circuit diagrams

I *functor string diagrams* prendono ispirazione dai *monoidal string diagrams* e li adattano per rappresentare funtori e trasformazioni naturali.

Neural circuit diagrams

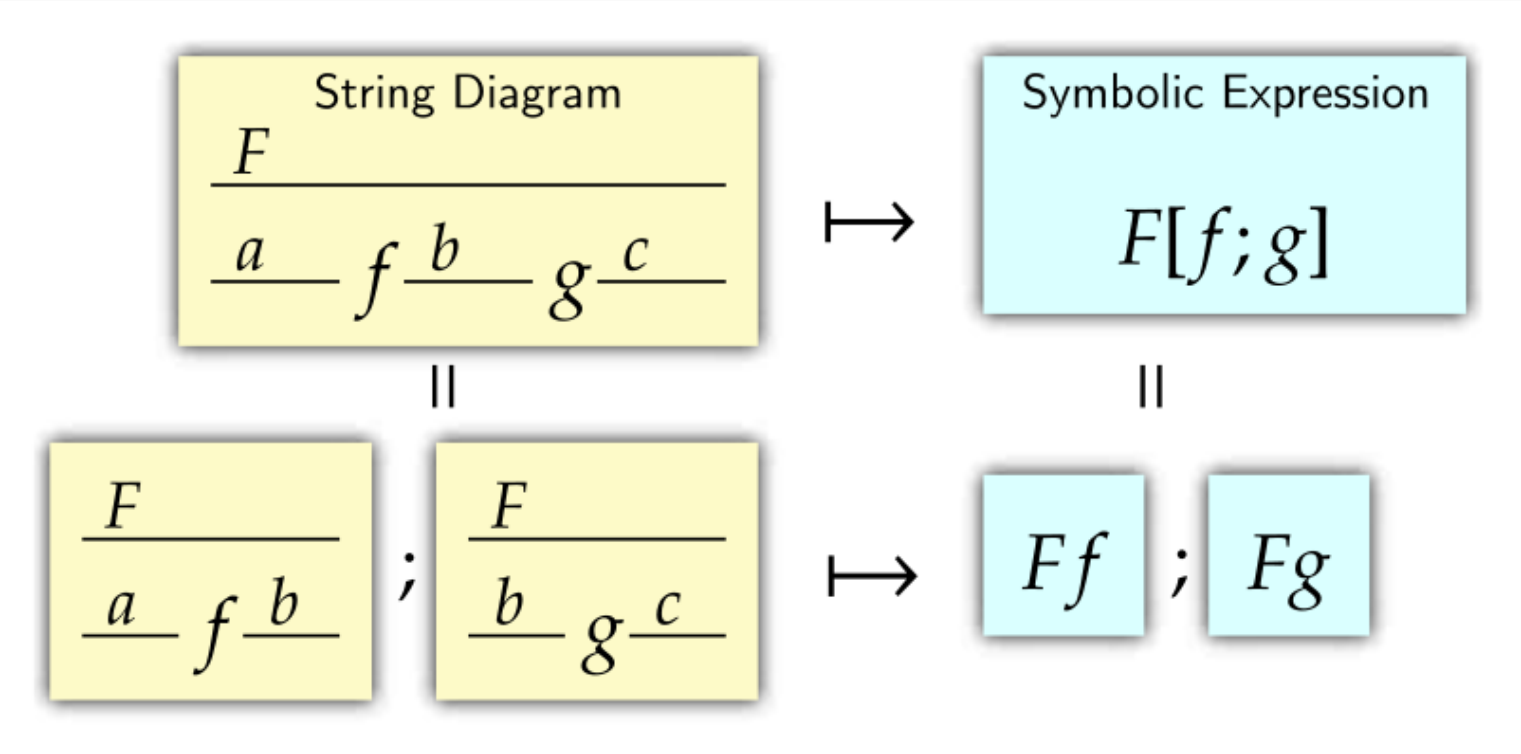
PRINCIPIO di decomposizione verticale

Uno *string diagram* può essere diviso in colonne verticali. Una singola colonna deve contenere solo oggetti o solo morfismi. Colonne con oggetti e colonne con morfismi devono alternarsi.

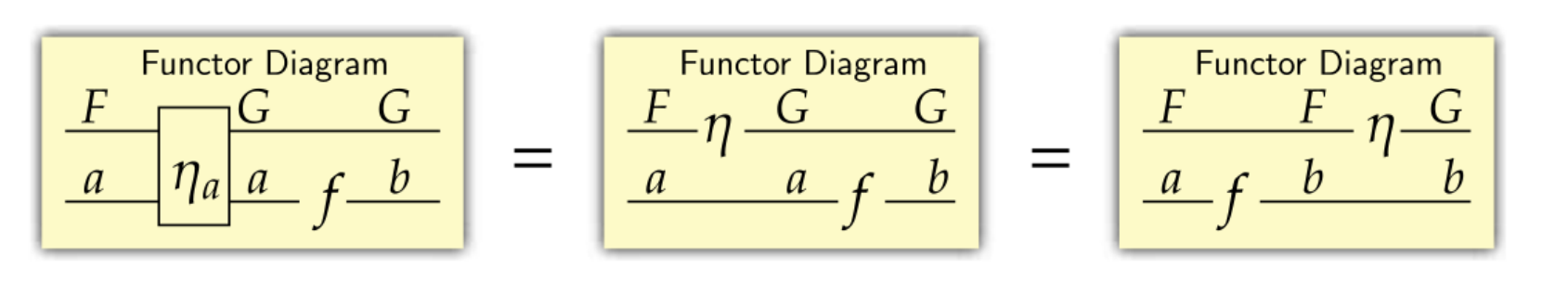
PRINCIPIO dell'espressione equivalente

Deve essere possibile sostituire ogni diagramma che segue una nuova notazione grafica con uno diagramma equivalente che segue la vecchia notazione.

Neural circuit diagrams

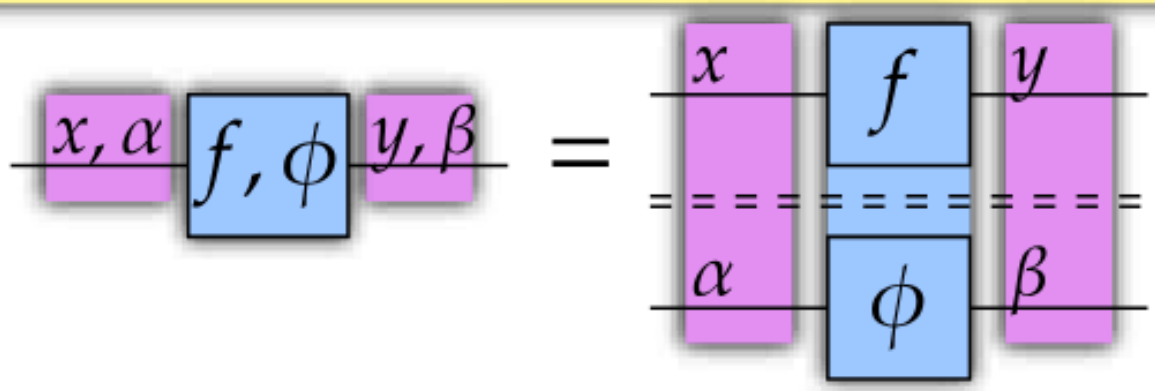


Neural circuit diagrams



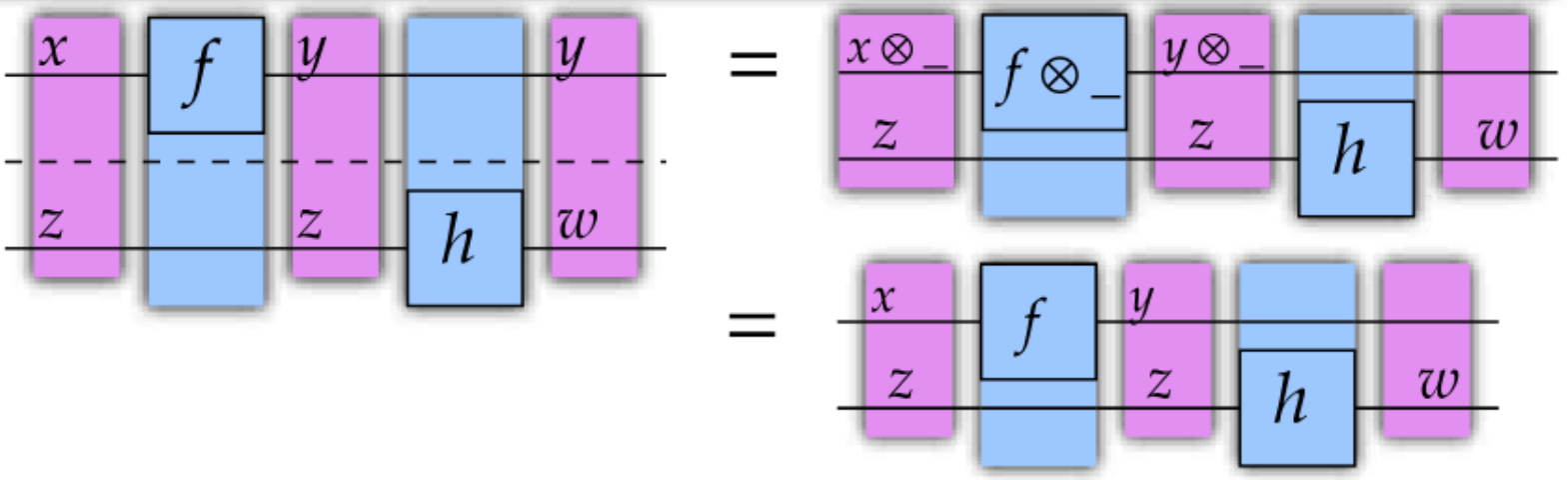
Neural circuit diagrams

We develop double-dashed lines as an equivalent expression to represent product categories $\mathbb{C} \times \mathbb{D}$, where objects and morphisms are tuples.



Neural circuit diagrams

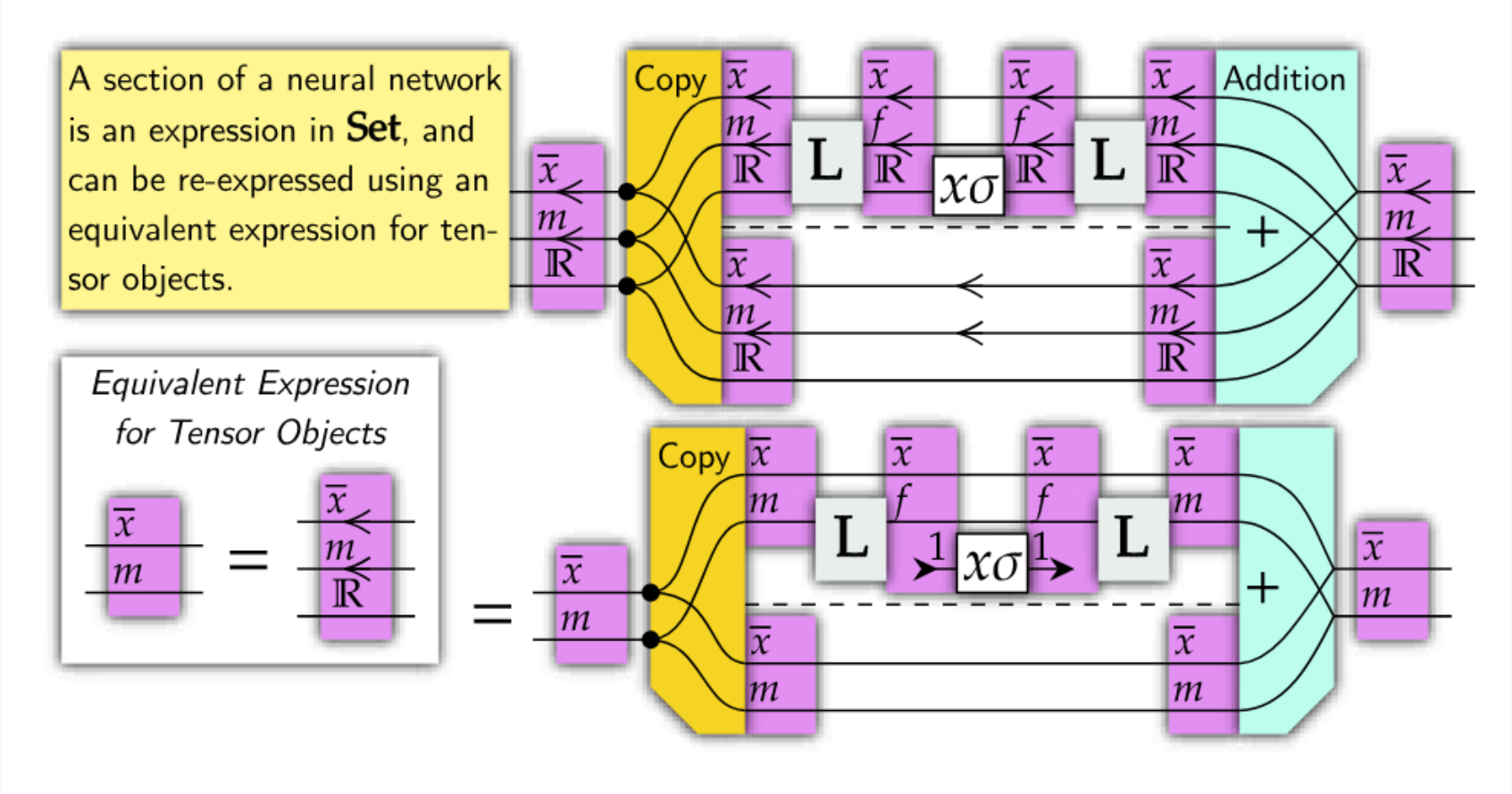
We can reexpress monoidal products using monoidal string diagrams wherein we develop equivalent expressions to view $x \otimes \_$ and $f \otimes \_$ as functors and natural transformations.



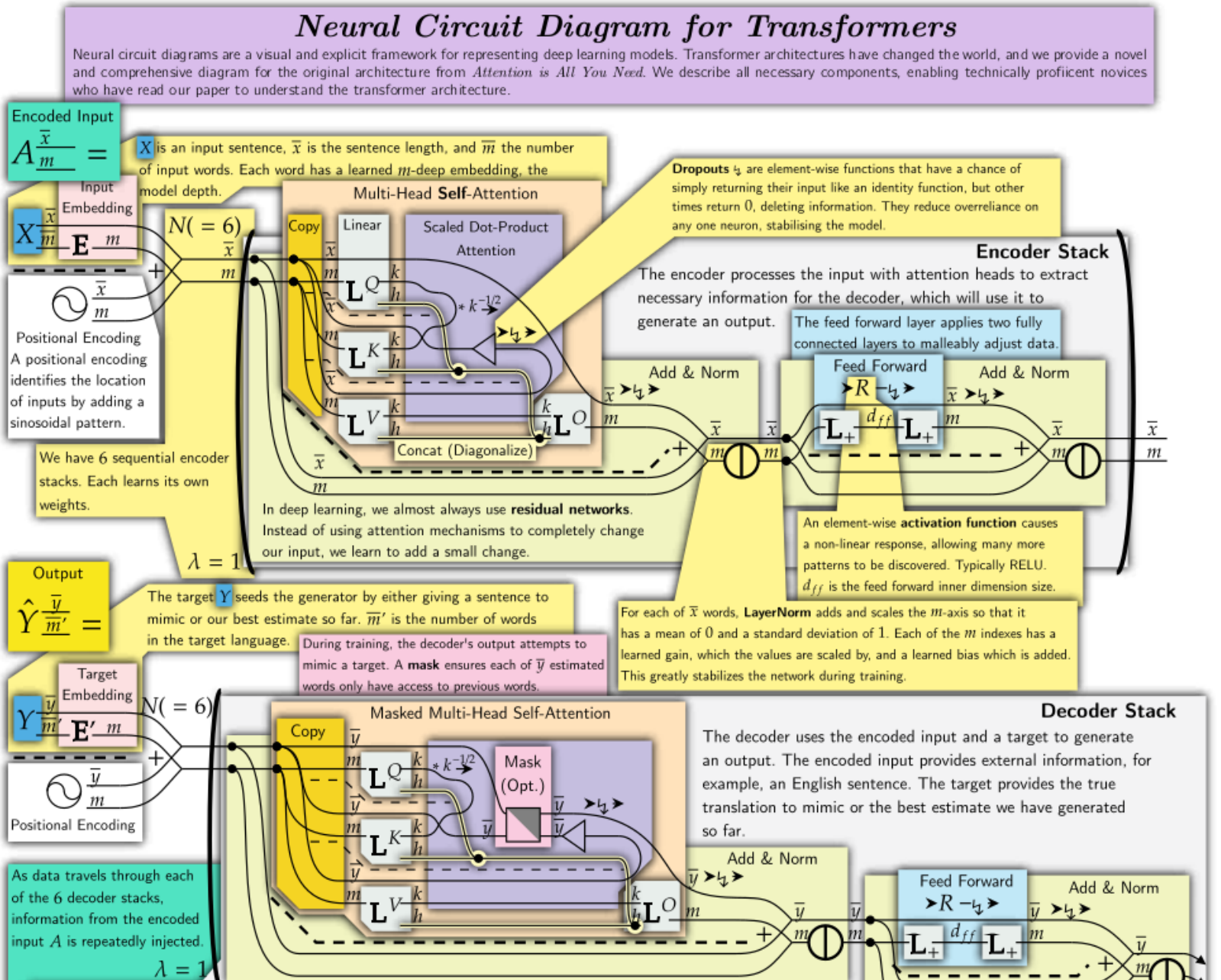
Neural circuit diagrams

I *neural circuit diagrams* sono *functor string diagrams* specializzati nel rappresentare architetture di reti neurali.

Neural circuit diagrams



Neural circuit diagrams



Prospettive future

Prospettive future

C'è una competizione in corso tra varie discipline, che puntano a spiegare le reti neurali utilizzando ciascuna i propri strumenti:

- fisica matematica,

Prospettive future

C'è una competizione in corso tra varie discipline, che puntano a spiegare le reti neurali utilizzando ciascuna i propri strumenti:

- fisica matematica,
- topologia,

Prospettive future

C'è una competizione in corso tra varie discipline, che puntano a spiegare le reti neurali utilizzando ciascuna i propri strumenti:

- fisica matematica,
- topologia,
- probabilità,

Prospettive future

C'è una competizione in corso tra varie discipline, che puntano a spiegare le reti neurali utilizzando ciascuna i propri strumenti:

- fisica matematica,
- topologia,
- probabilità,
- e così via...

Prospettive future

La teoria delle categorie, oltre a offrire strumenti propri, potrebbe creare un ponte tra queste discipline e potrebbe unificare i loro approcci in una teoria generale del deep learning.