Lens Semantics of Machine Learning

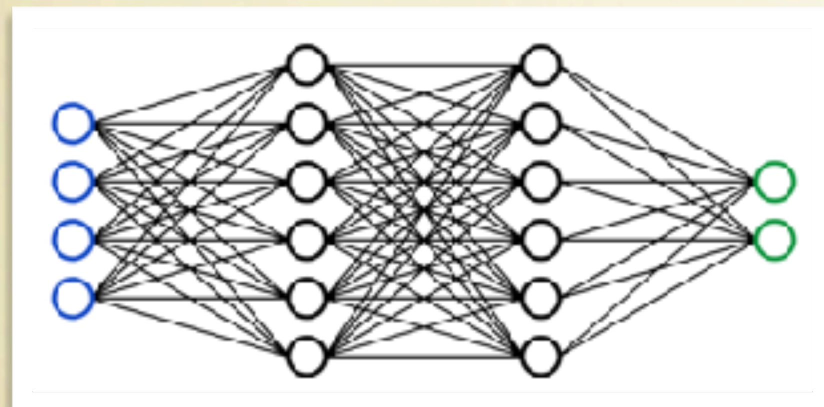
Fabio Zanasi
University College London

Based on joint work with
Geoff Cruttwell, Bruno Gavranovic, Neil Ghani, Paul Wilson

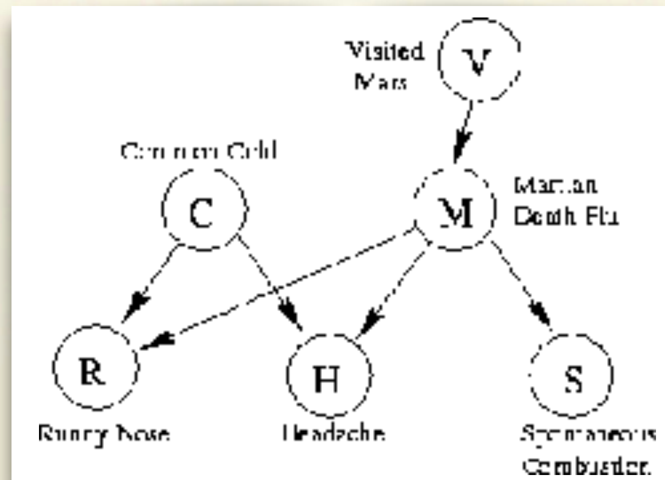
IFIP WG1.3 Meeting
January 2022

String Diagrams in Computer Science

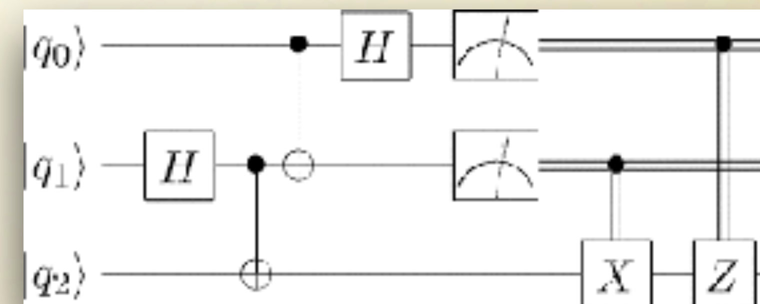
Neural networks



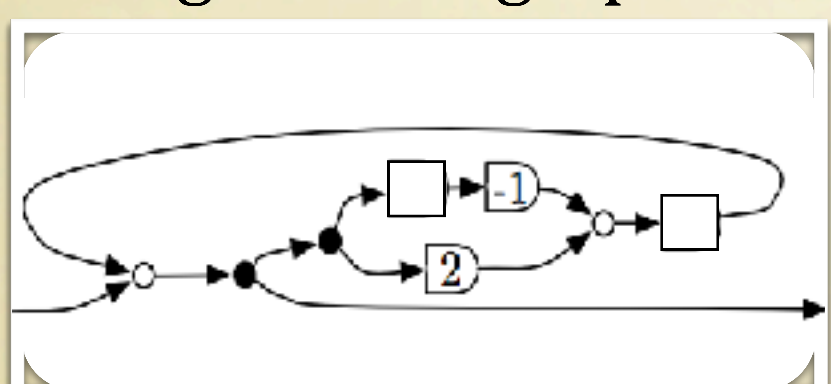
Bayesian networks



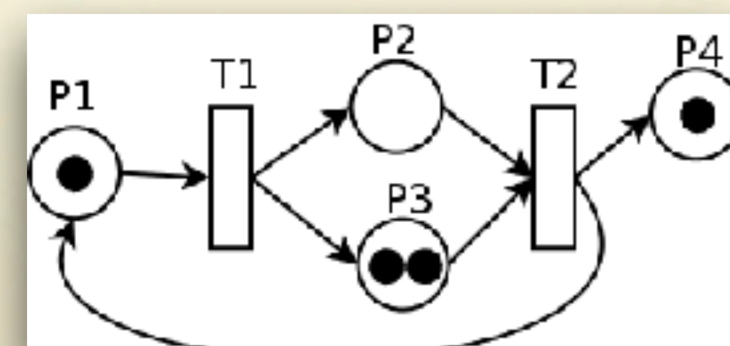
Quantum circuits



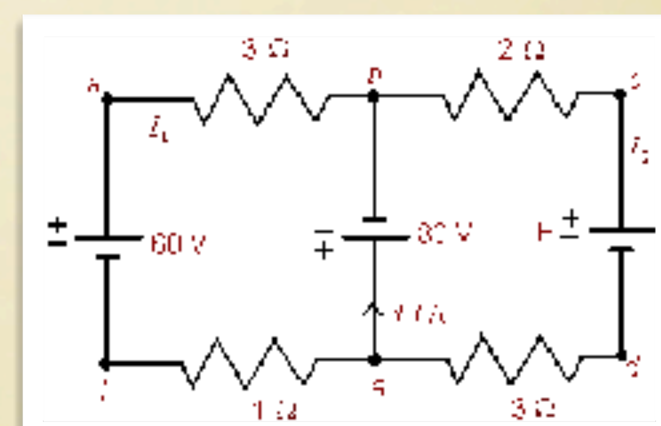
Signal flow graphs



Petri nets



Electrical Circuits

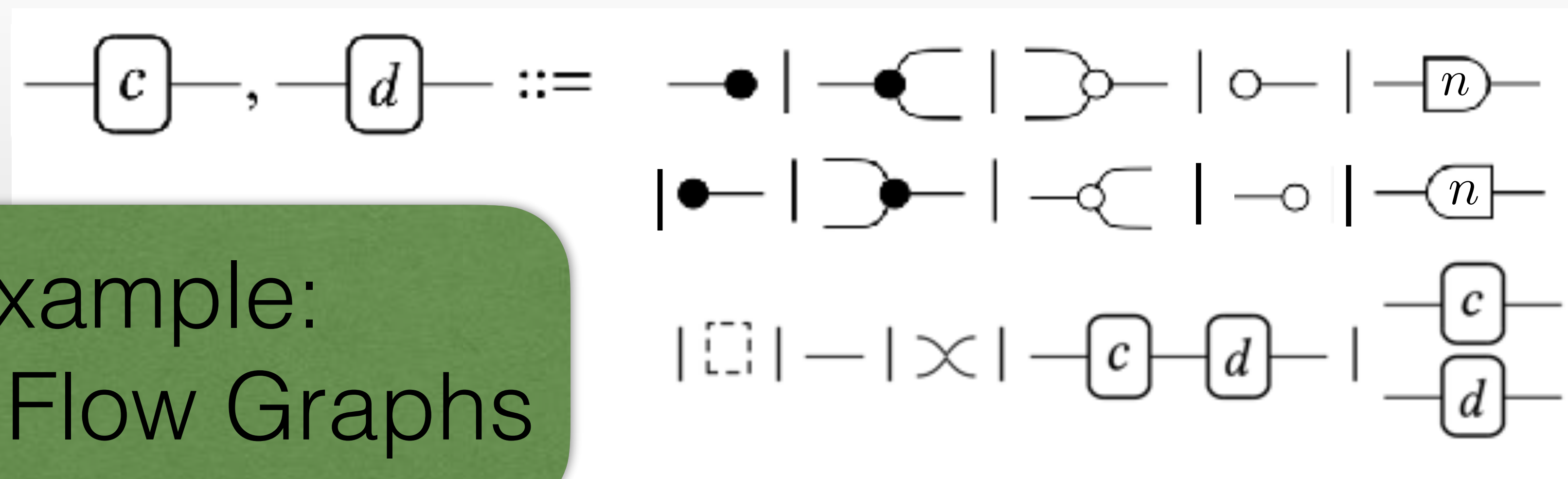


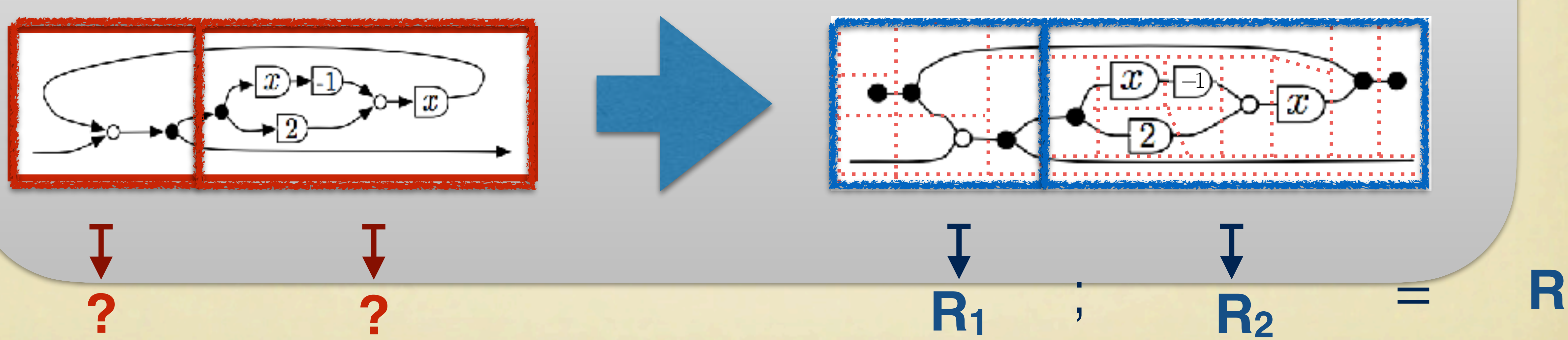
Key Ideas

Graphical notation becomes **formal syntax**

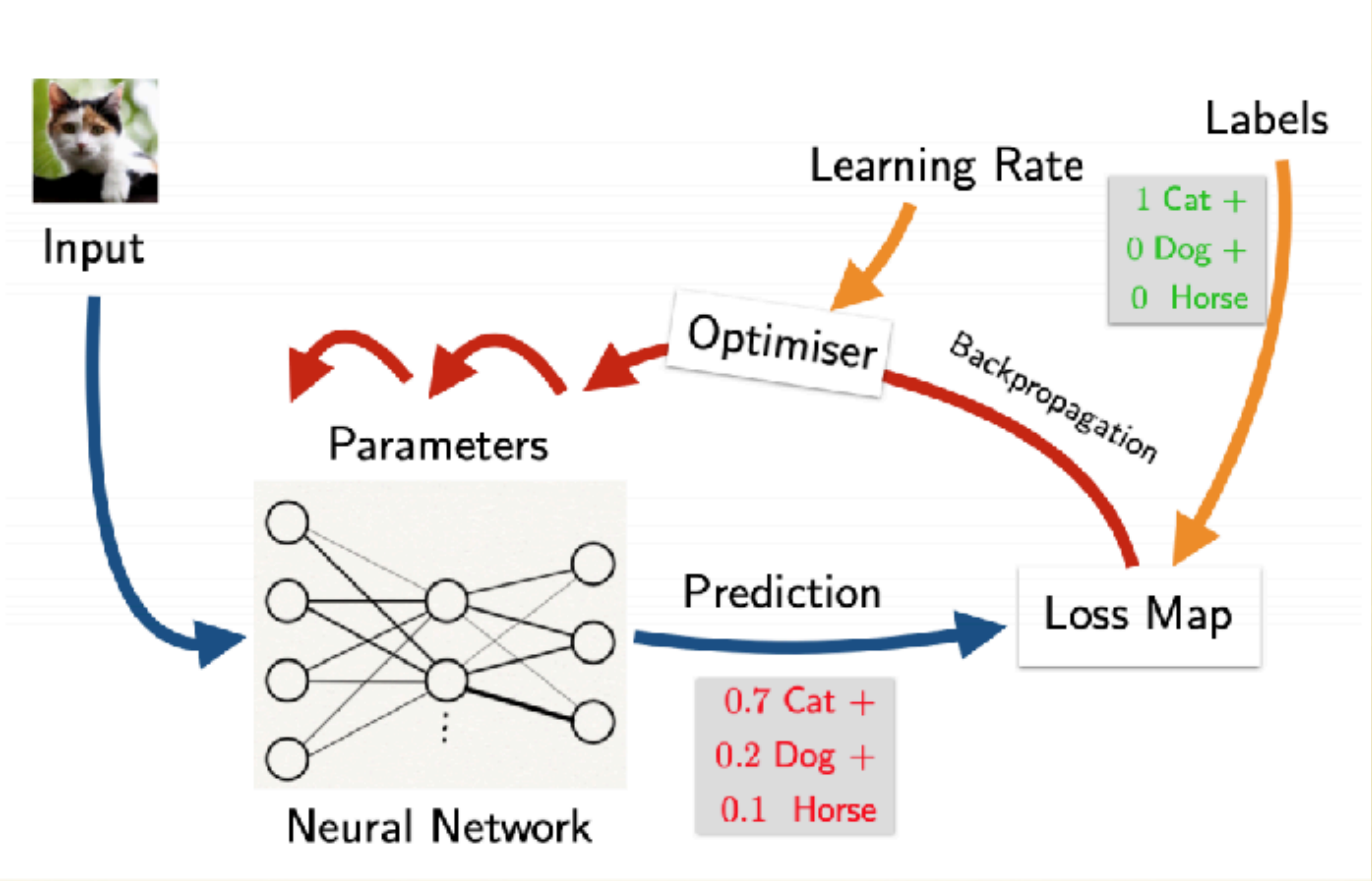
and it is given **compositional** semantics

Example:
Signal Flow Graphs





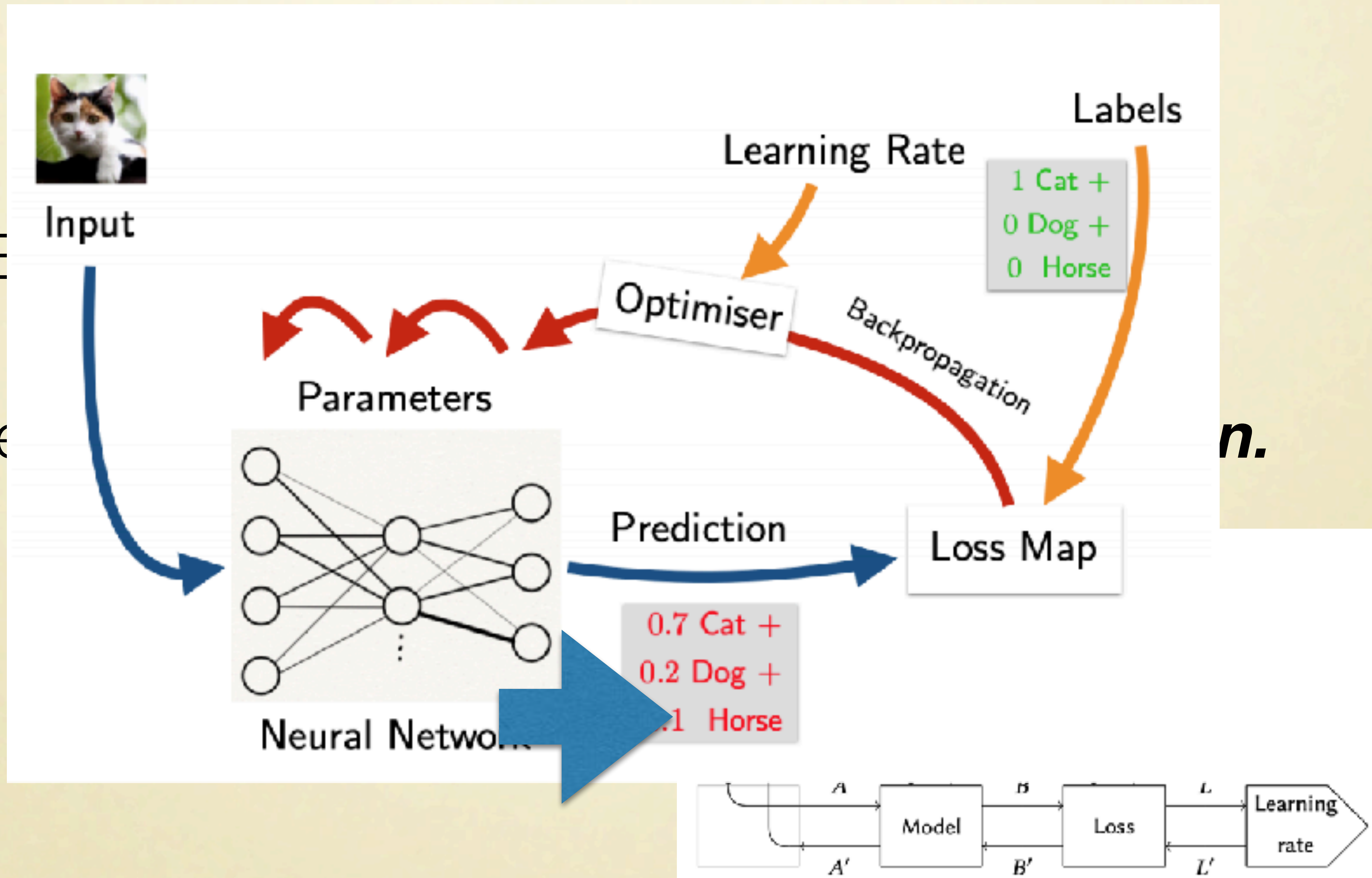
Case study: Gradient-Based Learning



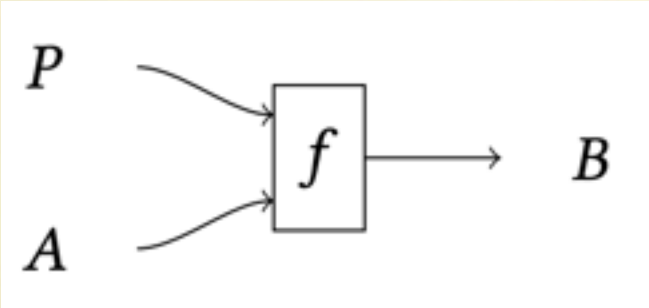
What are the fundamental semantic structures underpinning (gradient-based) learning?

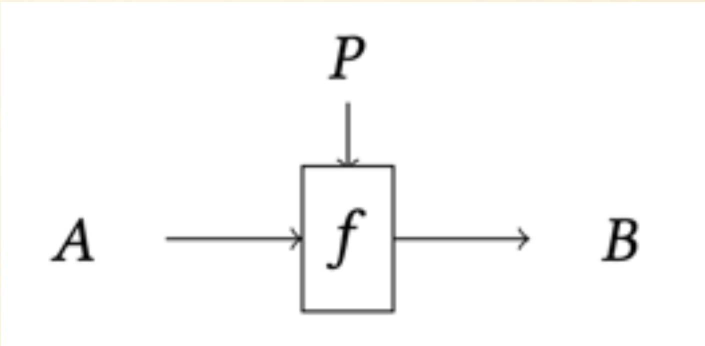
Roadmap

Plan:



Parametric Maps



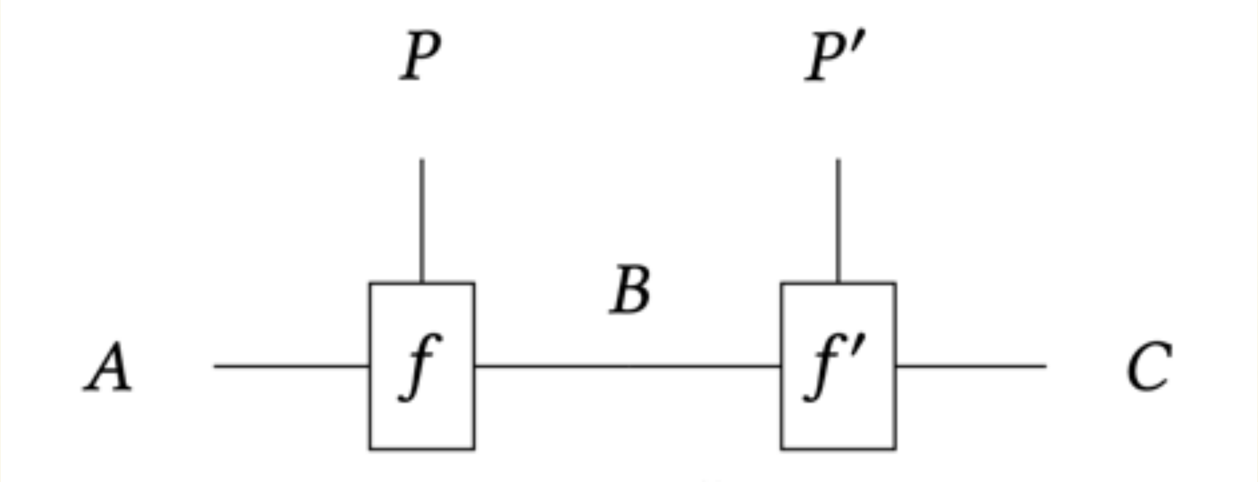


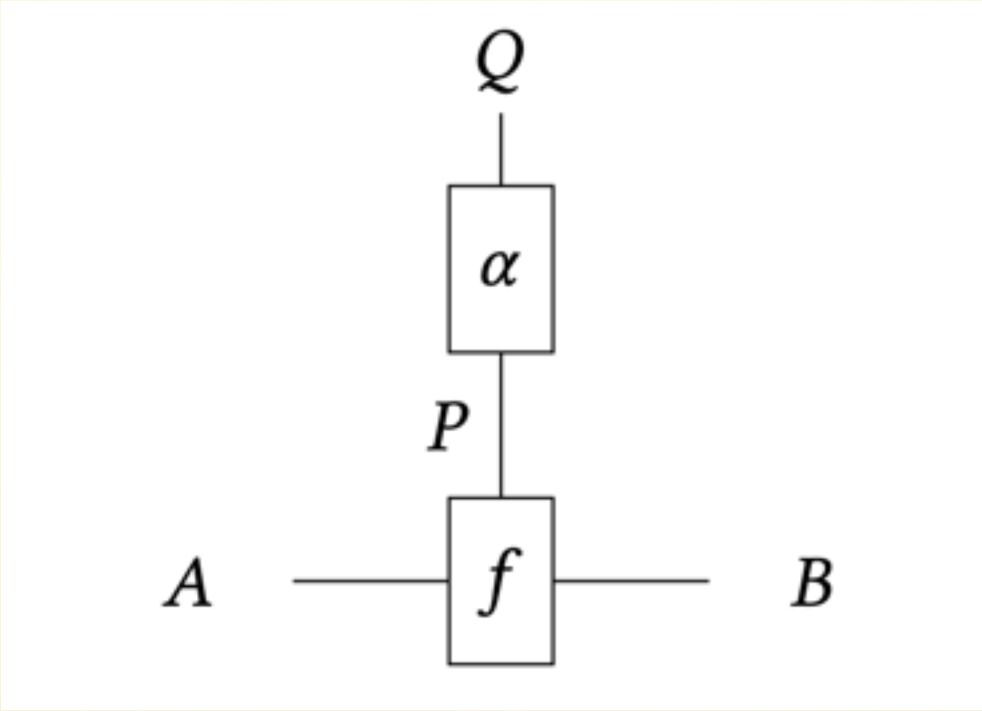
Para(C) $f\colon A \to B$

C $f\colon P \times A \to B$

$$\frac{f\colon A \to B}{f\colon P \times A \to B}$$

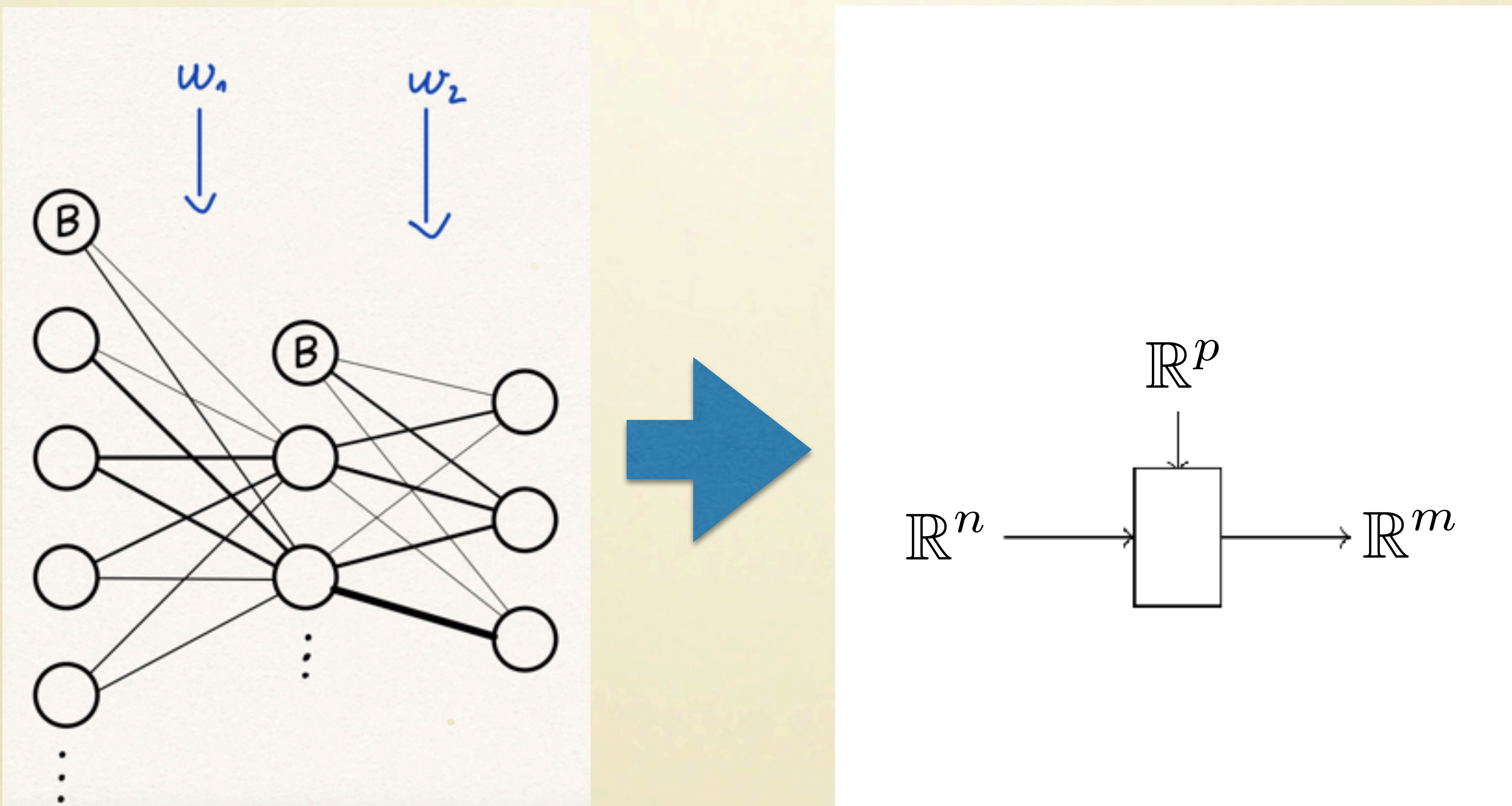
Composing Parametric Maps





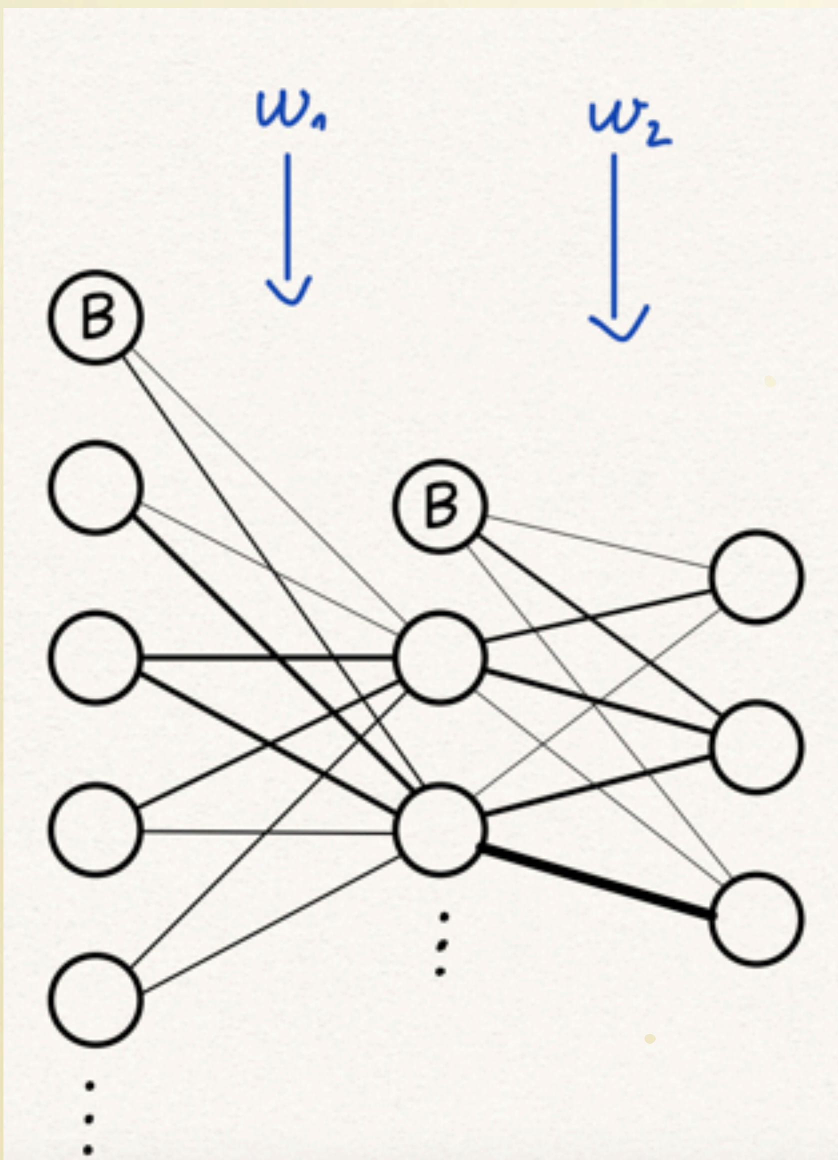
Parametric Maps

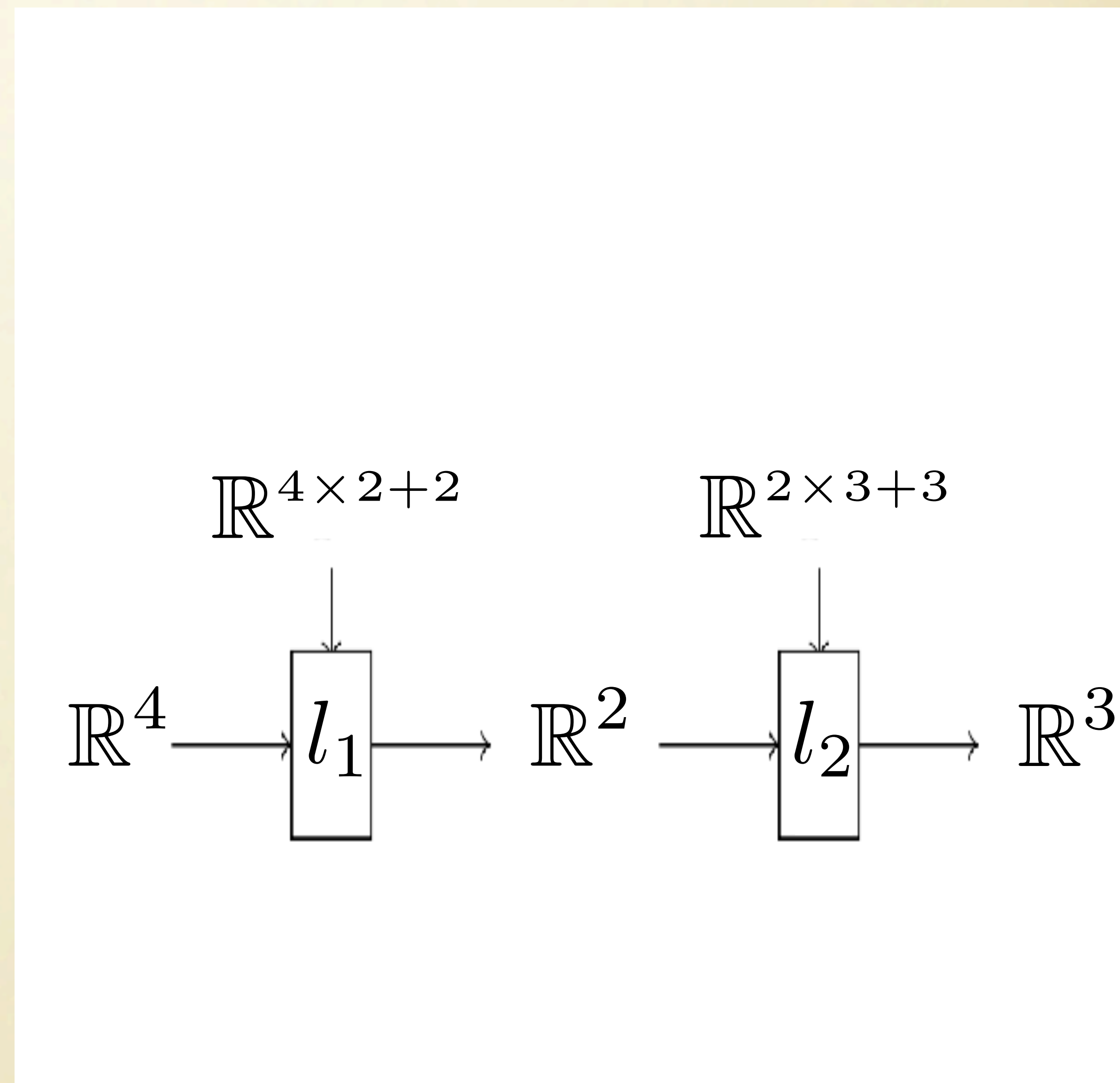
Example: a neural network is a parametric map in **Smooth**



Parametric Maps

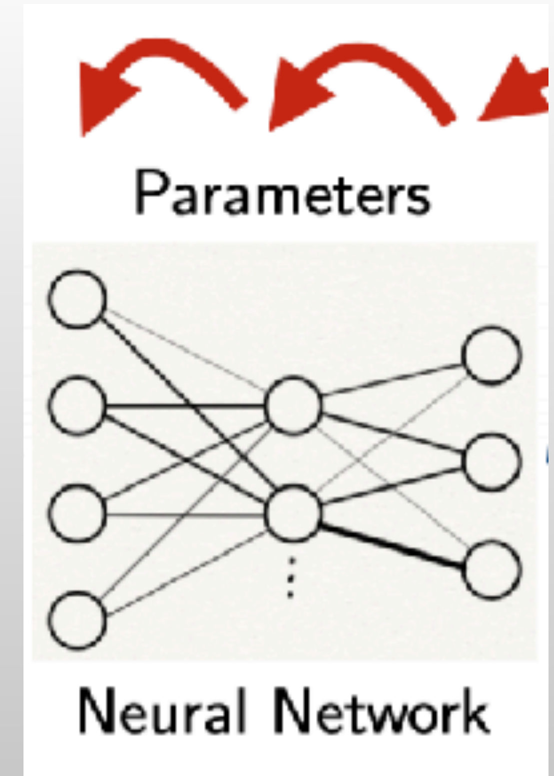
Example: a neural network is a parametric map in **Smooth**





Lenses

$\mathsf{Lens}(\mathcal{C})$ $\quad \dfrac{l \colon (A, A') \to (B, B')}{\mathsf{get}_l \colon A \to B \qquad \mathsf{put}_l \colon A \times B' \to A'}$



Reverse derivative categories *

Robin Cockett
University of Calgary, Department of Computer Science, Canada

Geoffrey Cruttwell
Mount Allison University, Department of Mathematics and Computer Science, Ca

Jonathan Gallagher
Dalhousie University, Department of Mathematics and Statistics, Canada

Jean-Simon Pacaud Lemay
University of Oxford, Department of Computer Science, UK
jean-simon.lemay@kellogg.ox.ac.uk

Benjamin MacAdam
University of Calgary, Department of Computer Science, Canada

Gordon Plotkin
Google Research

Dorette Pronk
Dalhousie University, Department of Mathematics and Statistics, Canada

a neural network is a lens in **Smooth**

$$f \colon A \to B$$

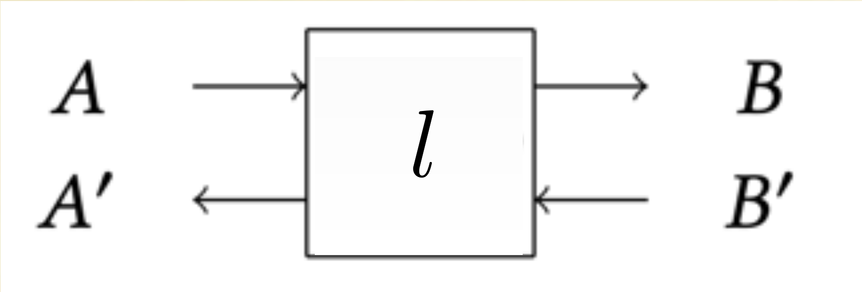
$$R[f] \colon A \times B' \to A'$$

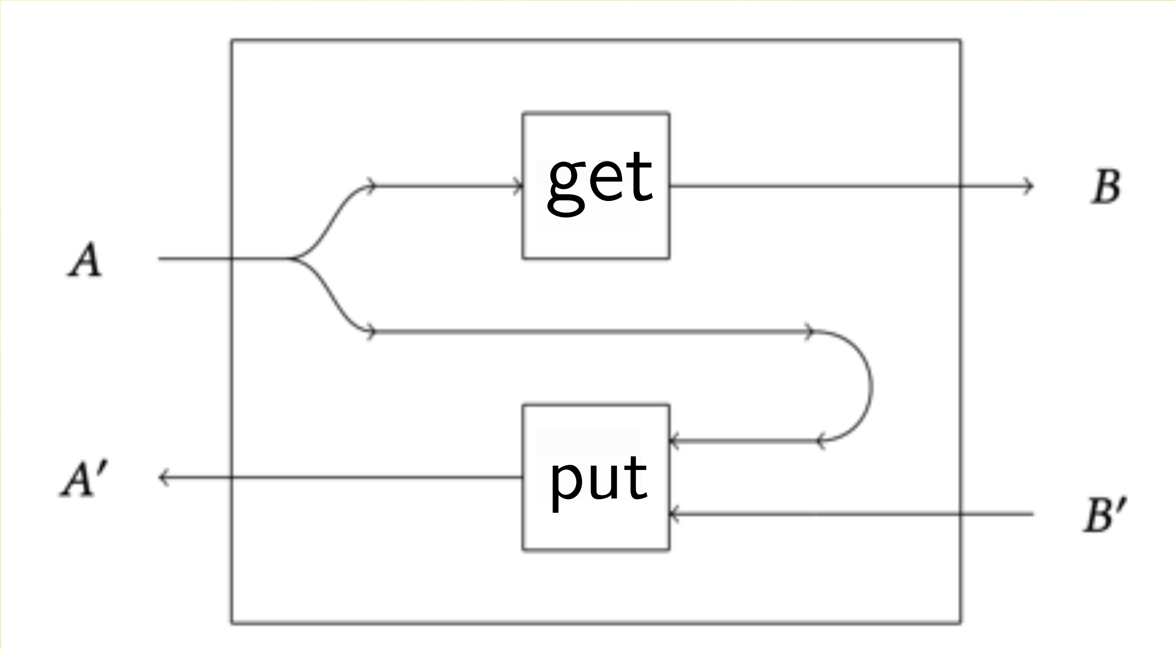
$$(\vec{x}, \vec{y}) \mapsto J_f(\vec{x})^T \cdot \vec{y}$$

String Diagrams for Lenses

We use string diagrams in $\mathsf{Tamb}(\mathcal{C})$

Faithful embedding of $\mathsf{Lens}(\mathcal{C})$ in $[\mathsf{Lens}(\mathcal{C})^{op}, \mathsf{Set}] \cong \mathsf{Tamb}(\mathcal{C})$

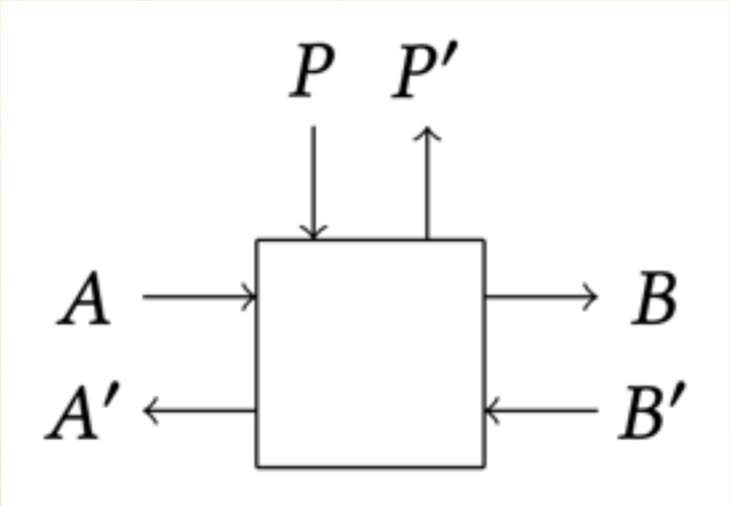




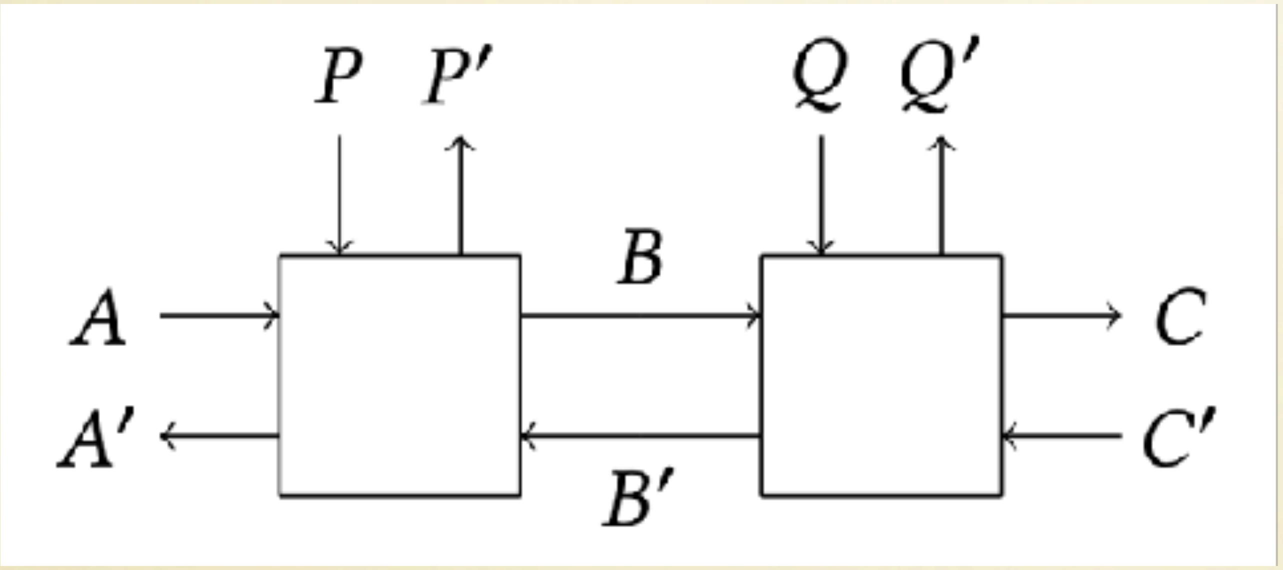
Parametric Lenses

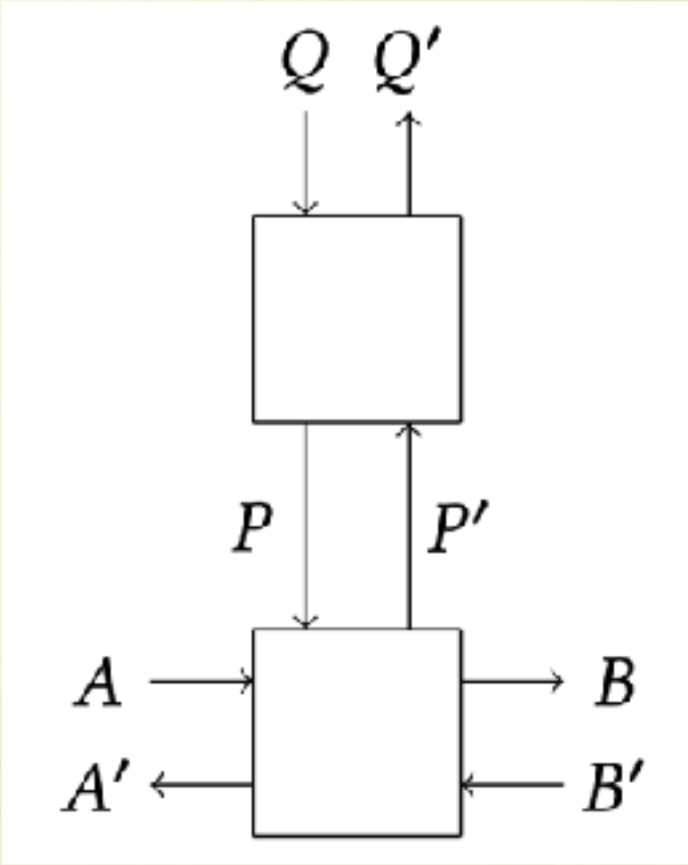
Parametric lenses = parametric maps in ***Lens(C)***

= maps in ***Para(Lens(C))***



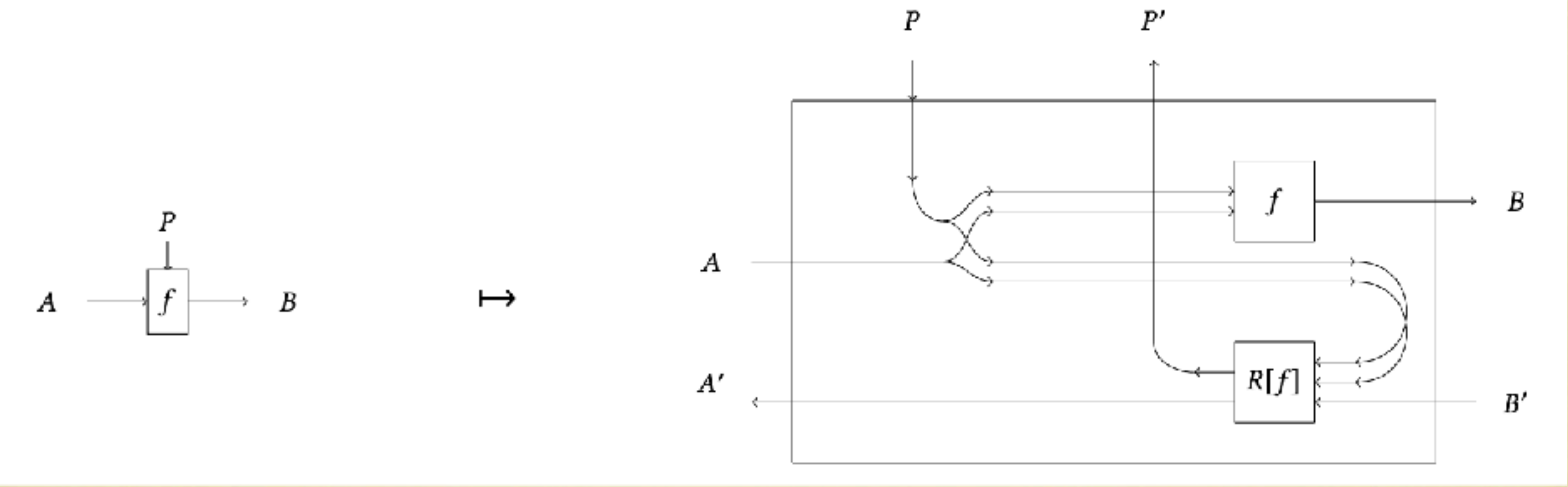
Composing Parametric Lenses



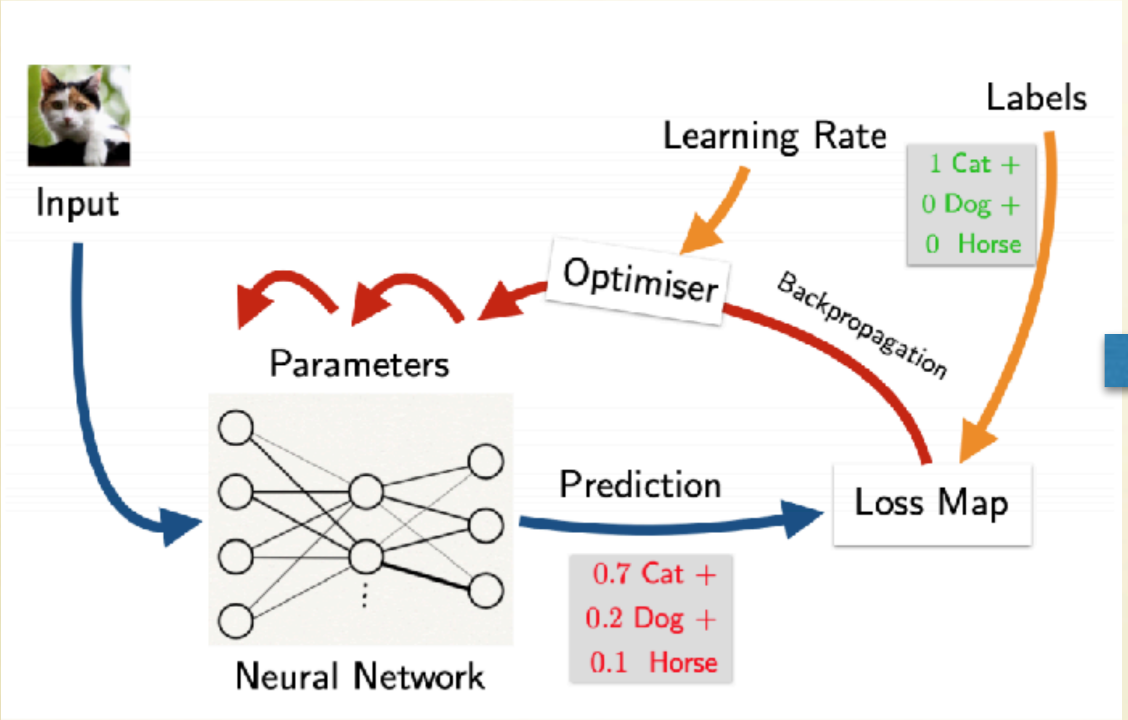


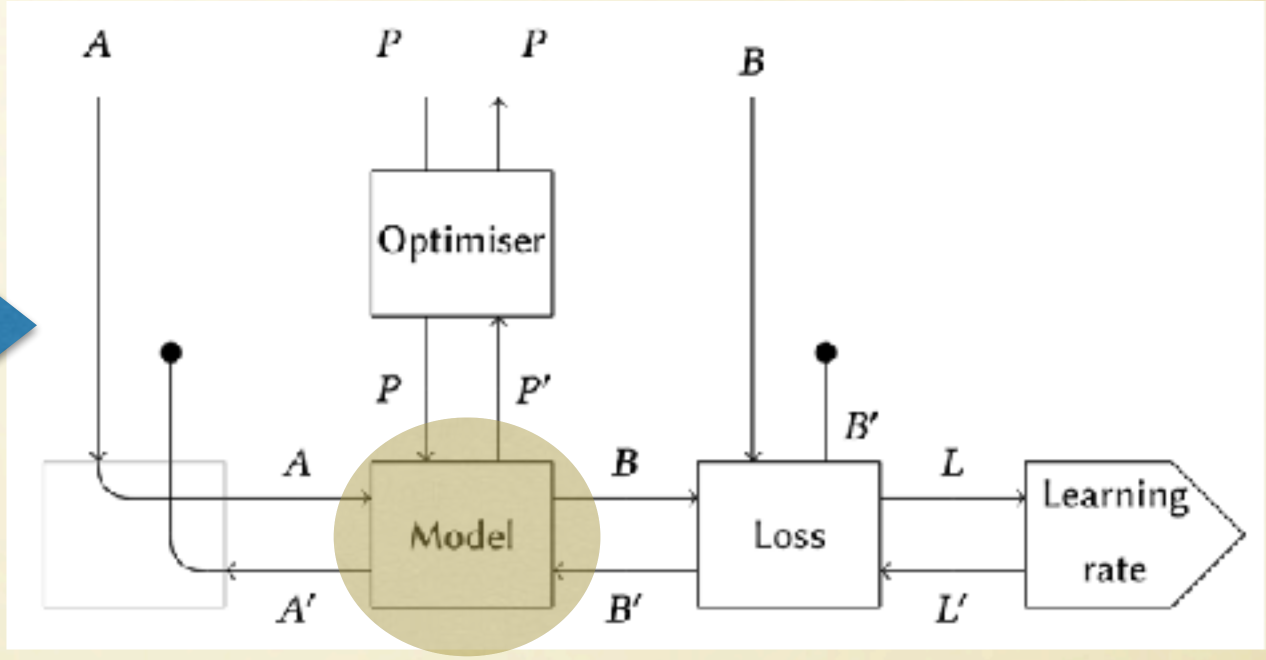
Parametric Lenses

Example: a neural network is a parametric lens in **Smooth**

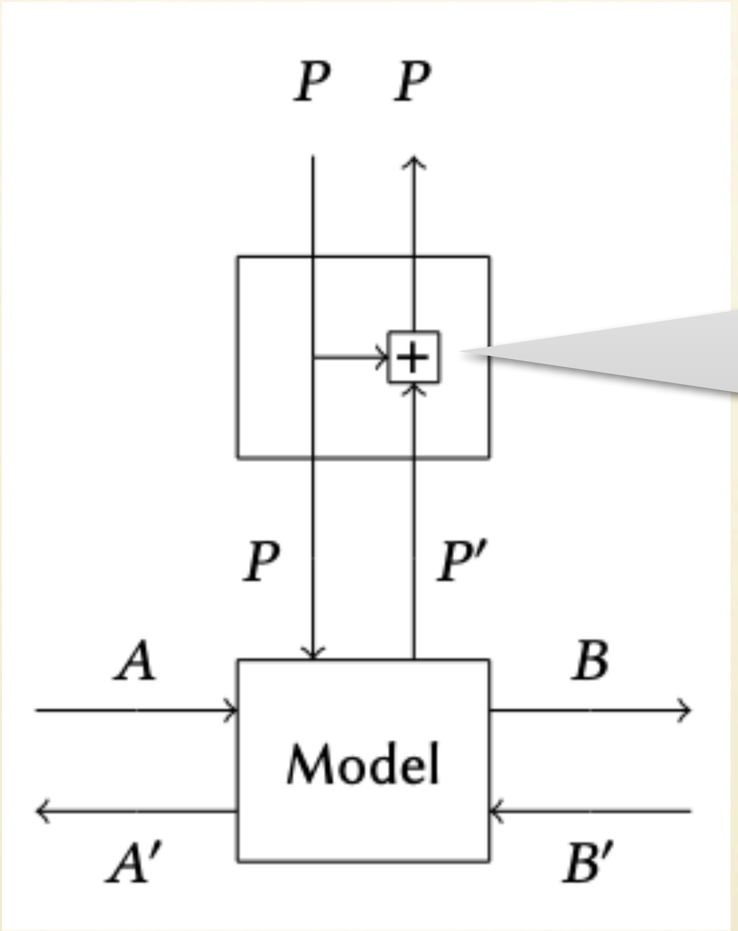


Where we are, so far





Optimiser as reparametrisation

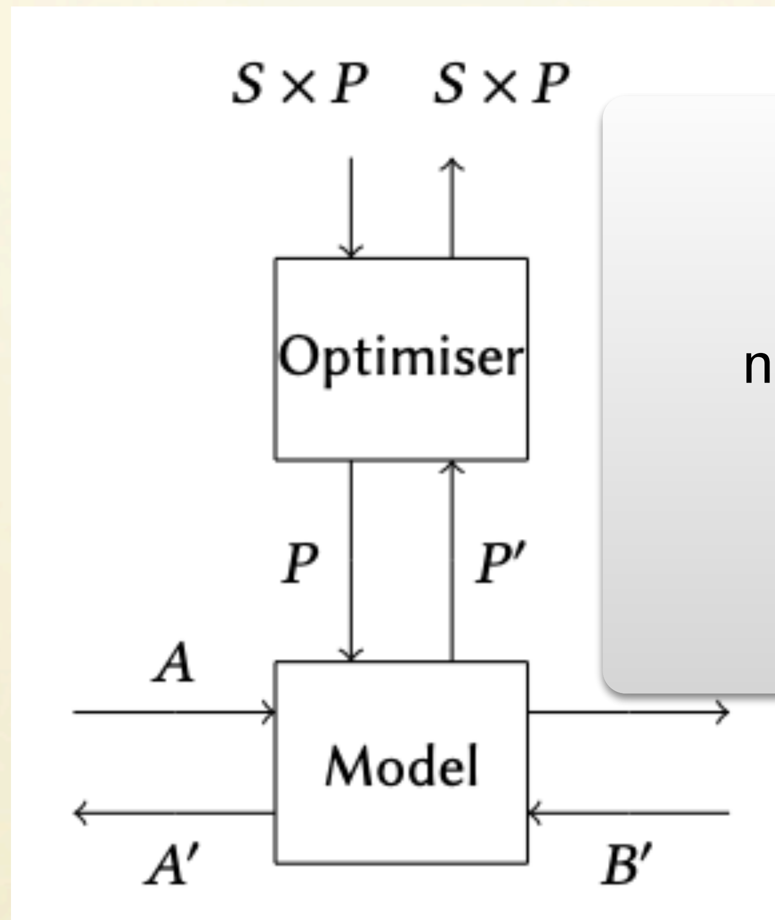


Lens(Smooth)

$$\frac{\mathsf{gd} : (P, P) \to (P, P')}{\begin{array}{c}\mathsf{get}_{gd} : p \mapsto p \\ \mathsf{put}_{\mathsf{gd}} : (p, p') \mapsto p + p'\end{array}}$$

Basic gradient descent

Optimiser as reparametrisation

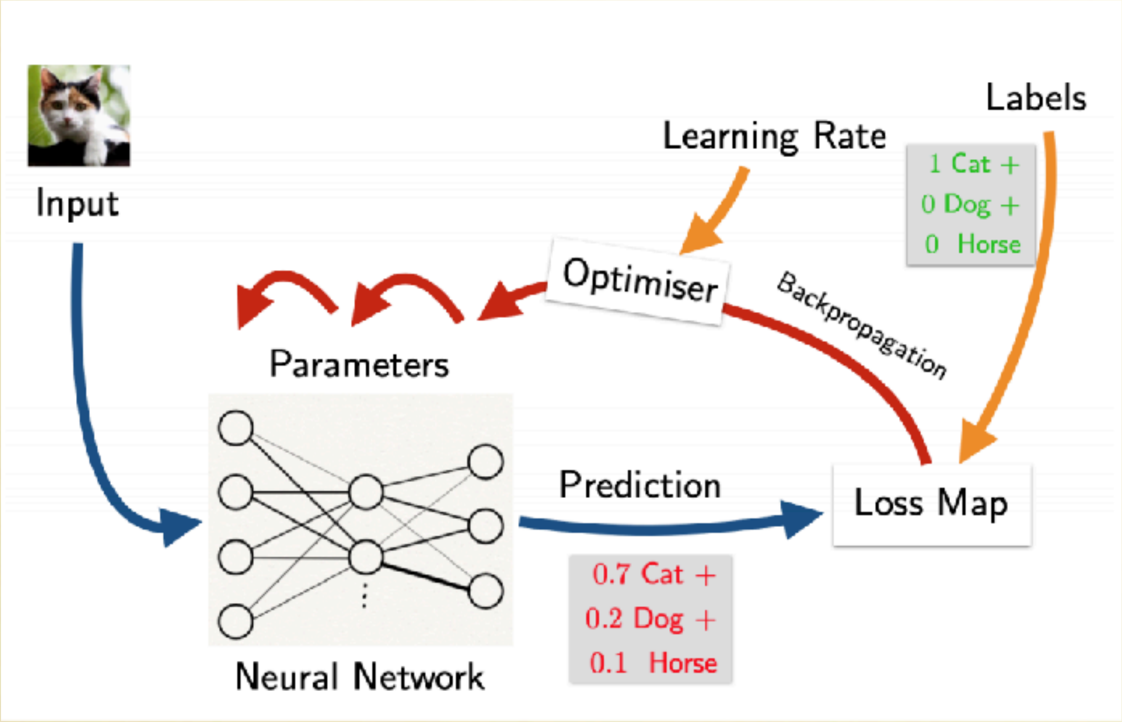


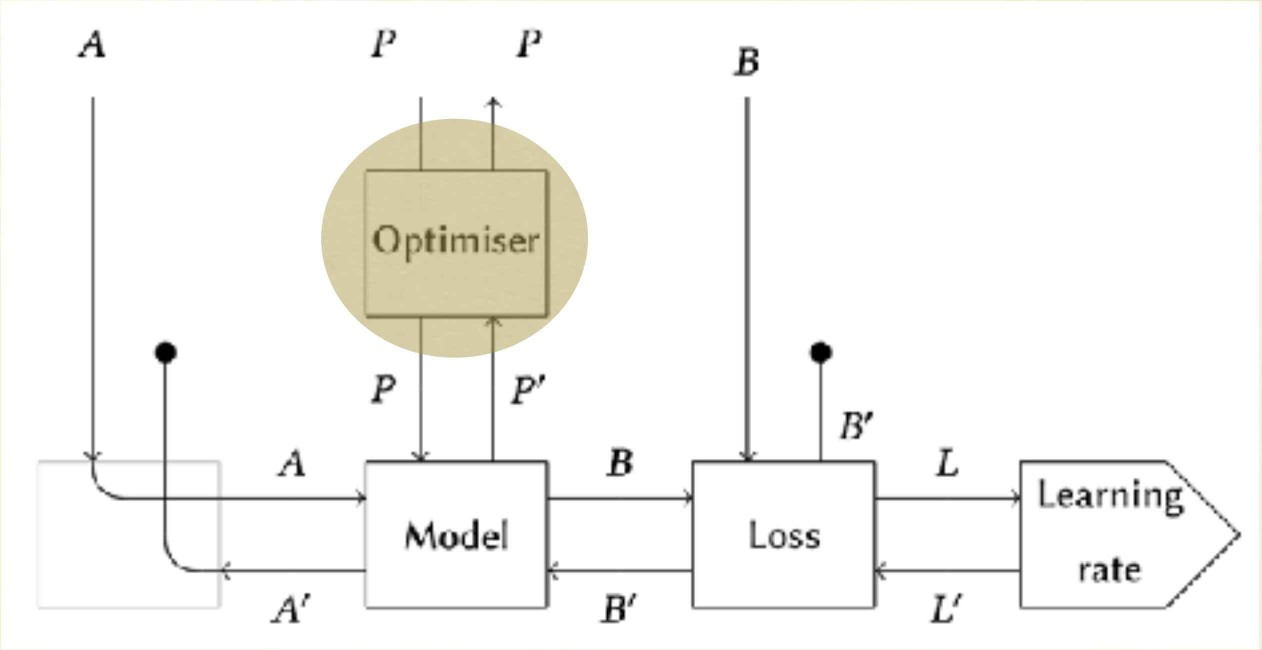
Lens(Smooth)

$$\frac{\mathsf{nm}\colon (S \times P, S \times P) \to (P, P')}{\begin{array}{l}\mathsf{get}_{\mathsf{nm}} : (s, p) \to p - \gamma s \\ \mathsf{put}_{nm} : (s, p, p') \mapsto (\gamma s + p', p - s)\end{array}}$$

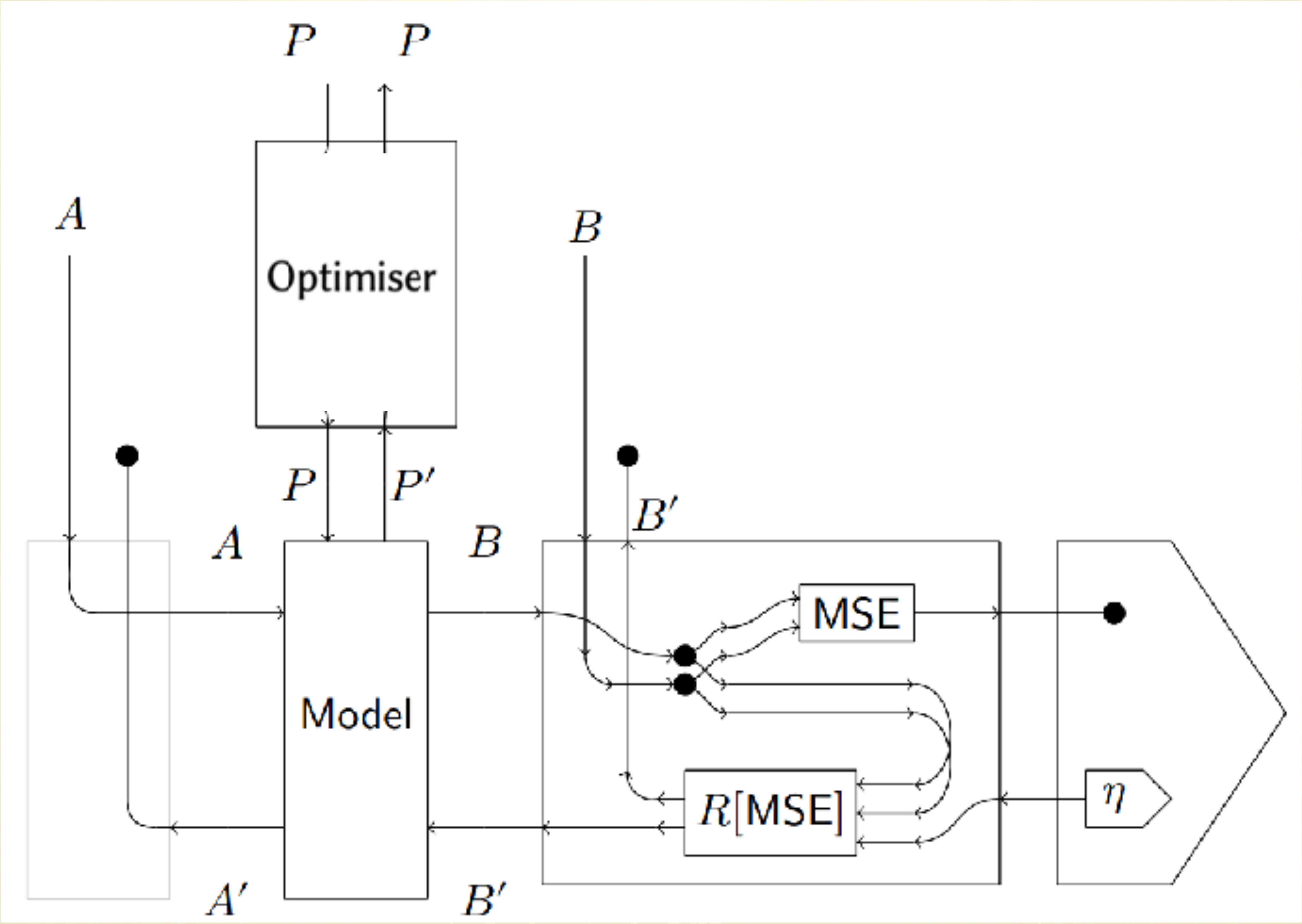
`Stateful' optimiser such as **Adagrad**, **Adam**, **Nesterov**, **Momentum**, etc. can be all formalised uniformly as reparametrisations.

Where we are, so far

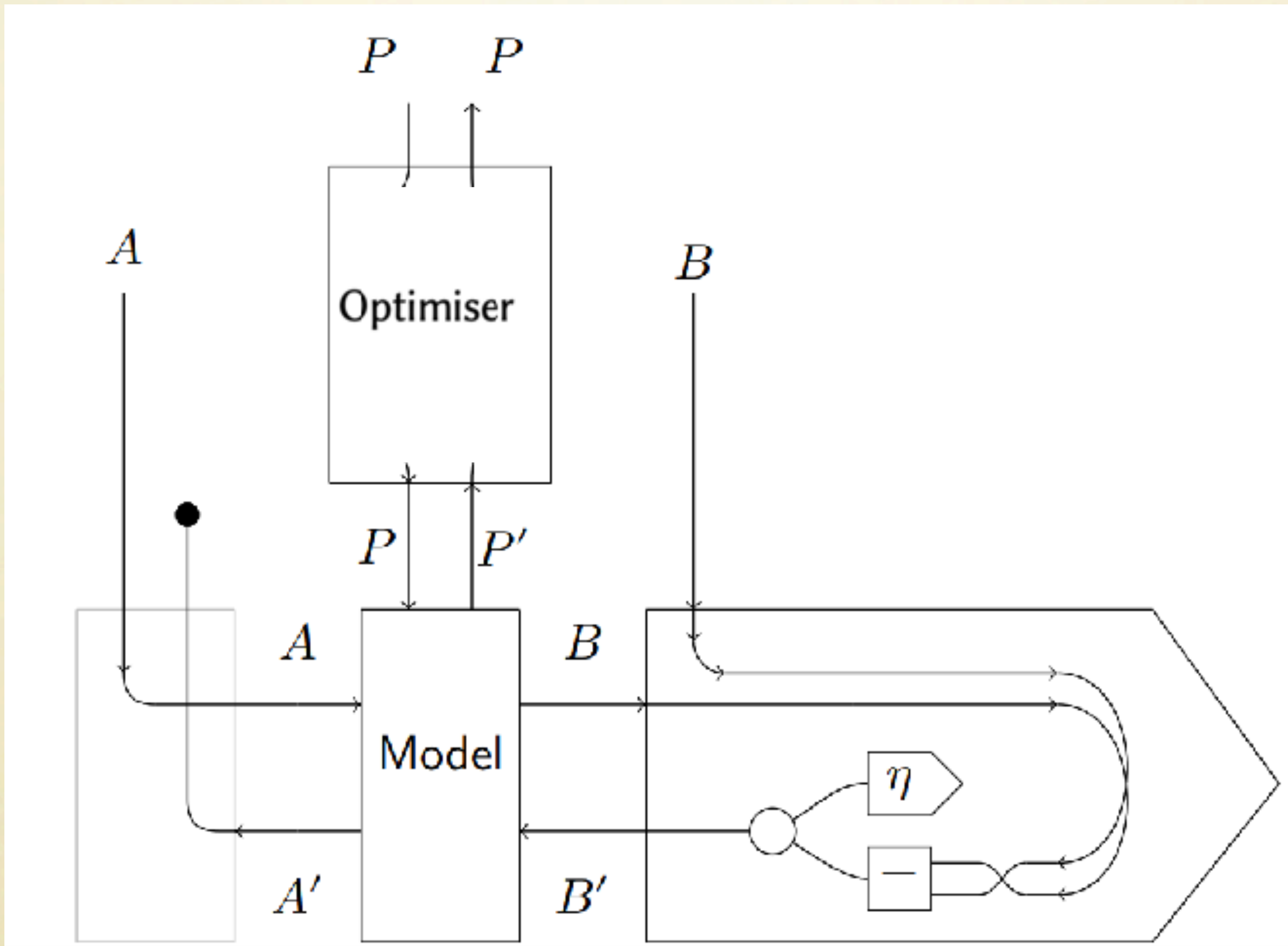




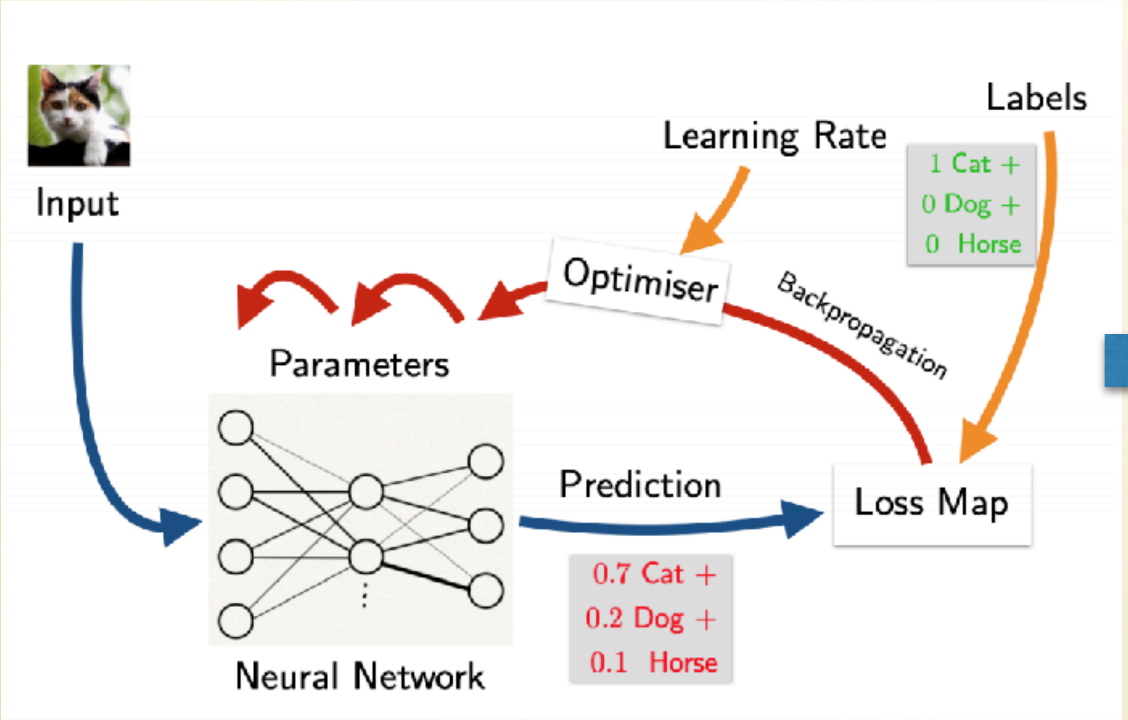
Mean Squared Error

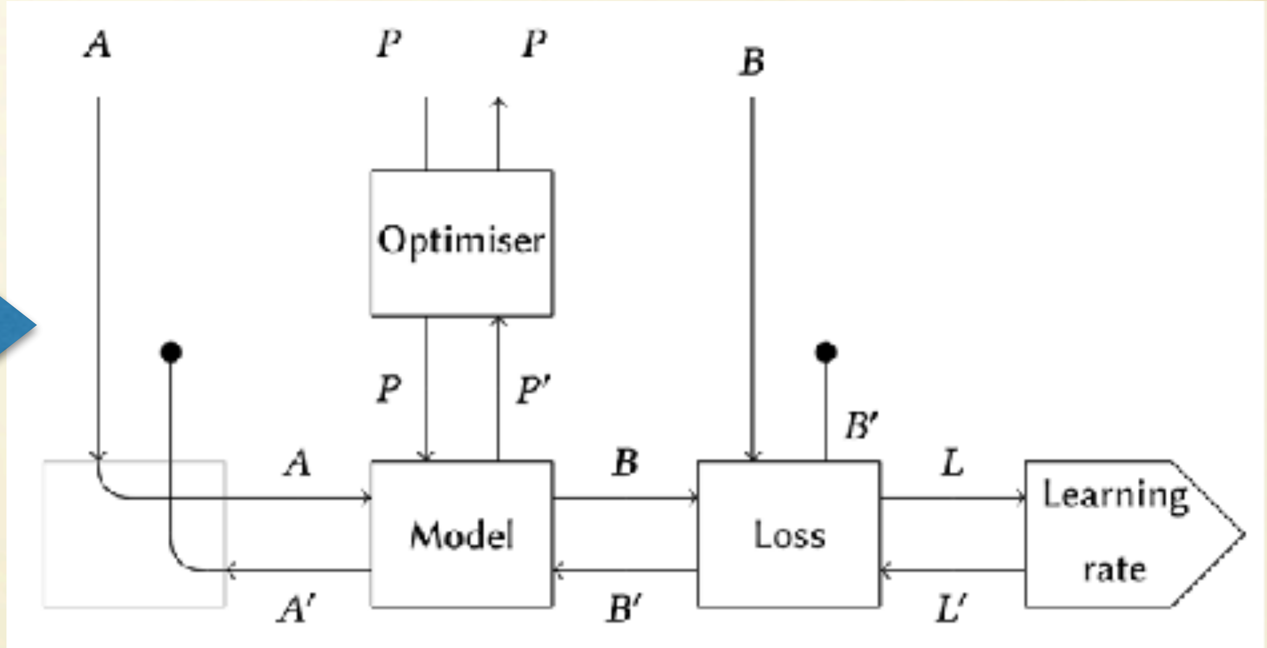


Mean Squared Error



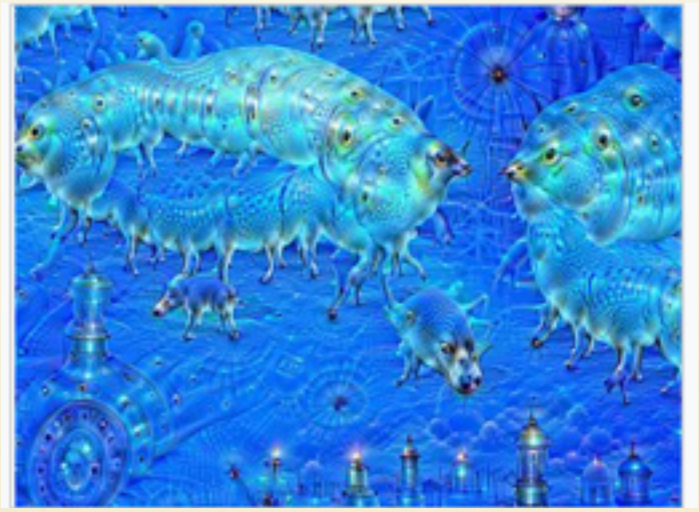
Where we are, so far





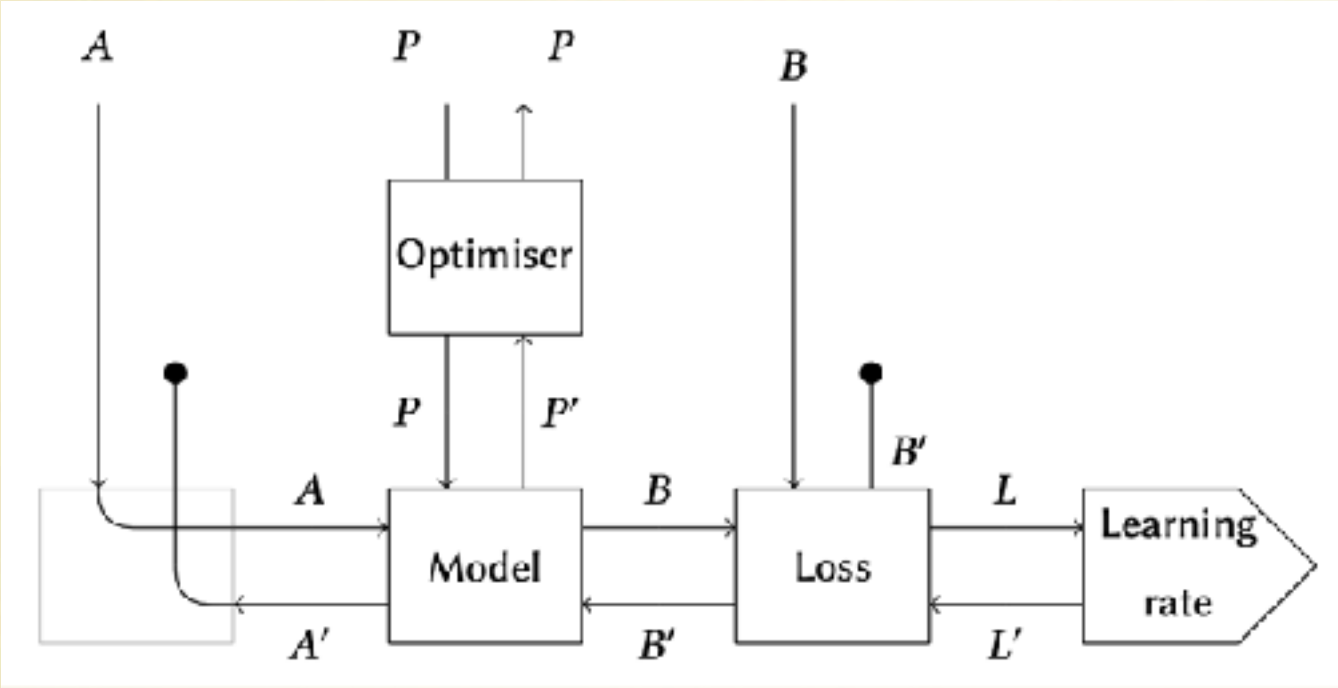
Variations I: Deep Dreaming

Input Image

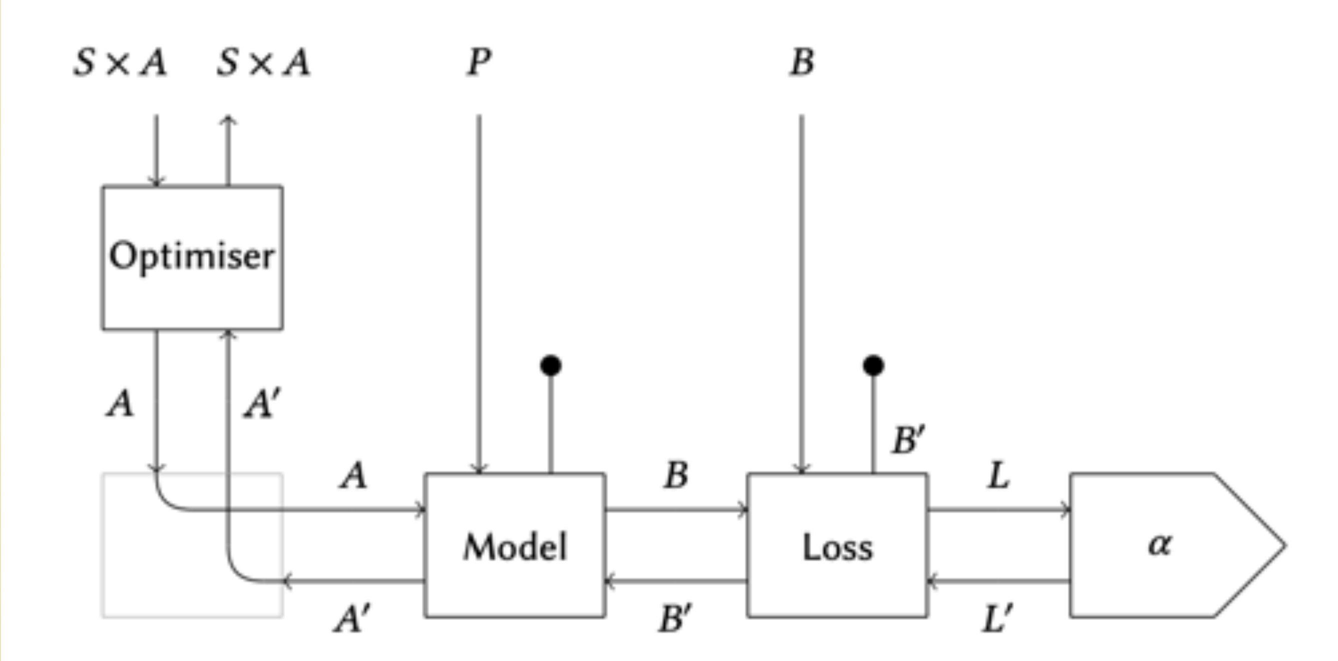


``Over-processed'' Image

Variations I: Deep Dreaming

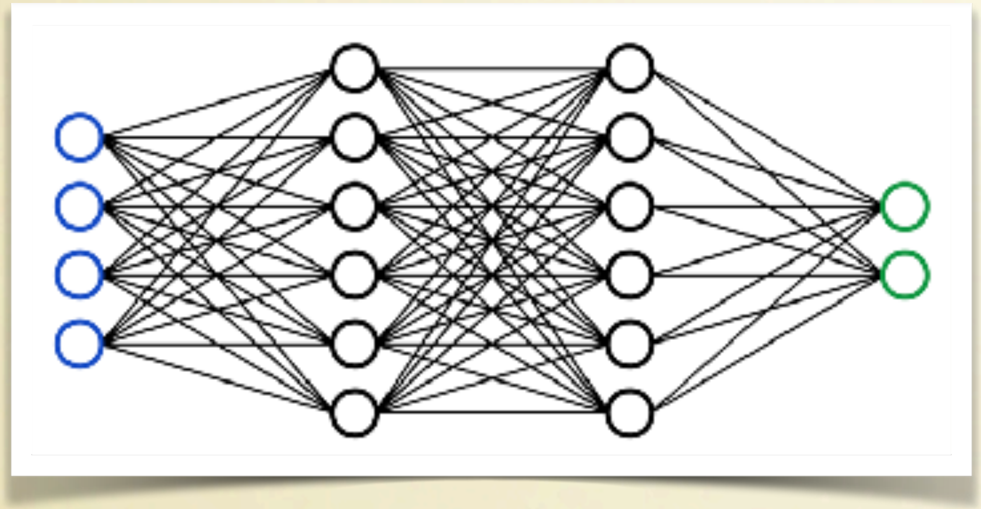


Parameter Learning

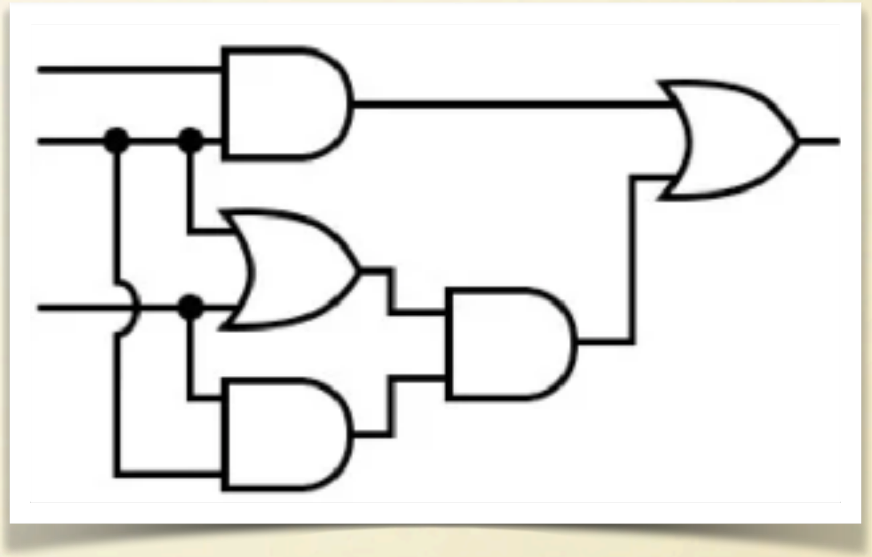


Deep dreaming

Variations II: Learning Boolean circuits



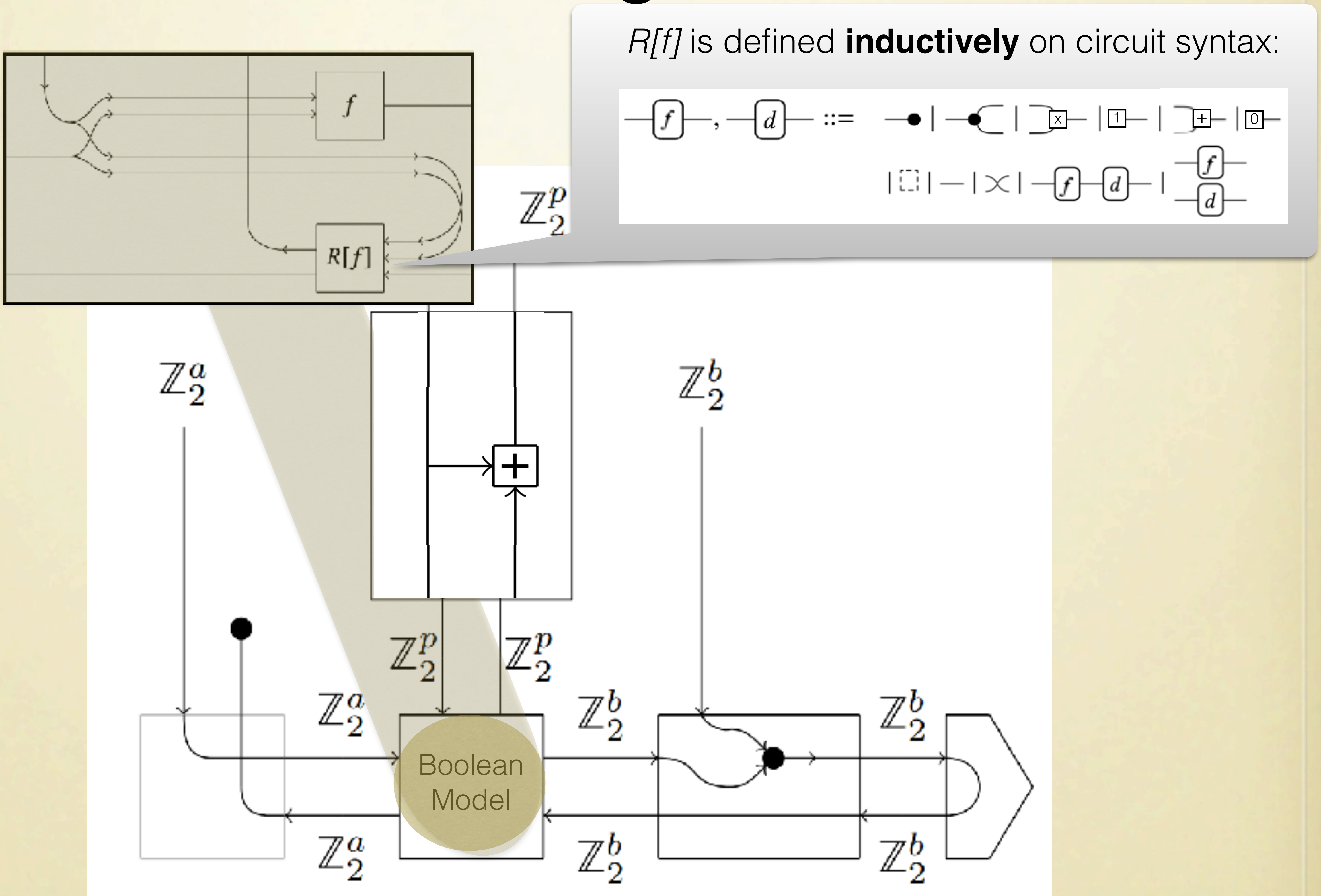
Neural networks



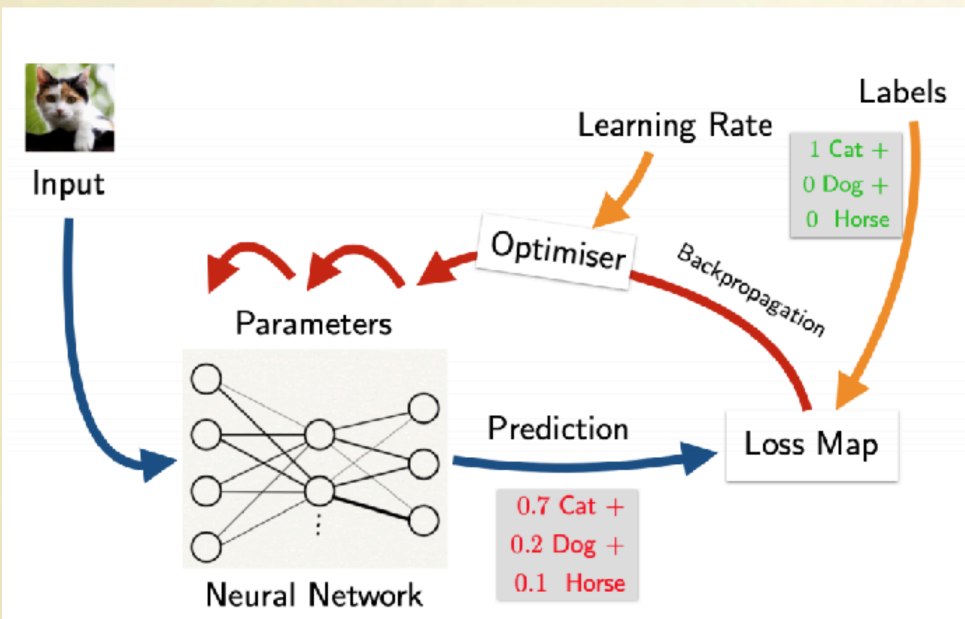
Boolean circuits

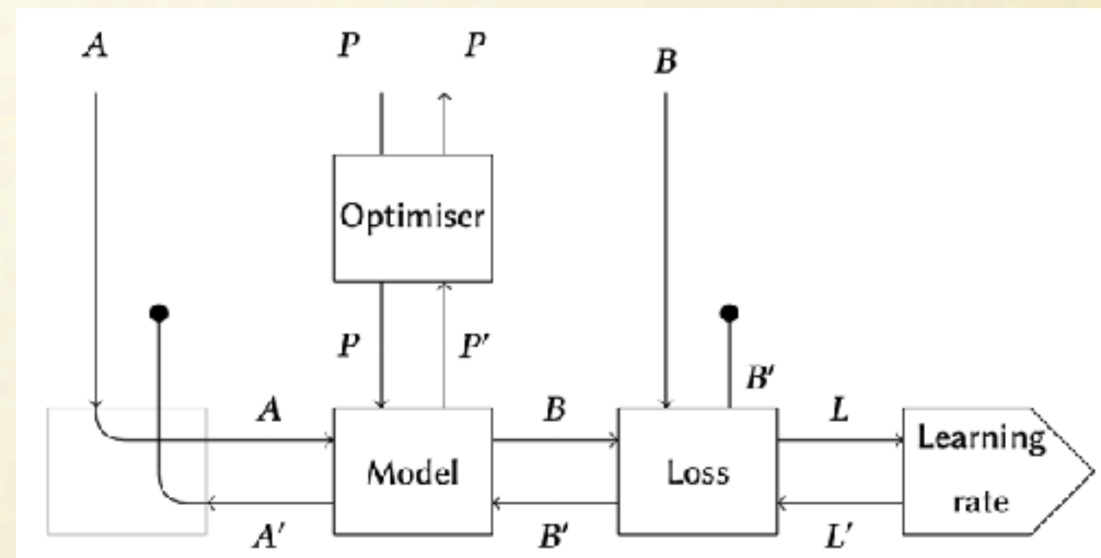
Instead of ***Smooth*** we work in ***Bool***

Variations II: Learning Boolean circuits



Discussion





An algebraic framework for gradient-based learning.

Discussion

Uniformity

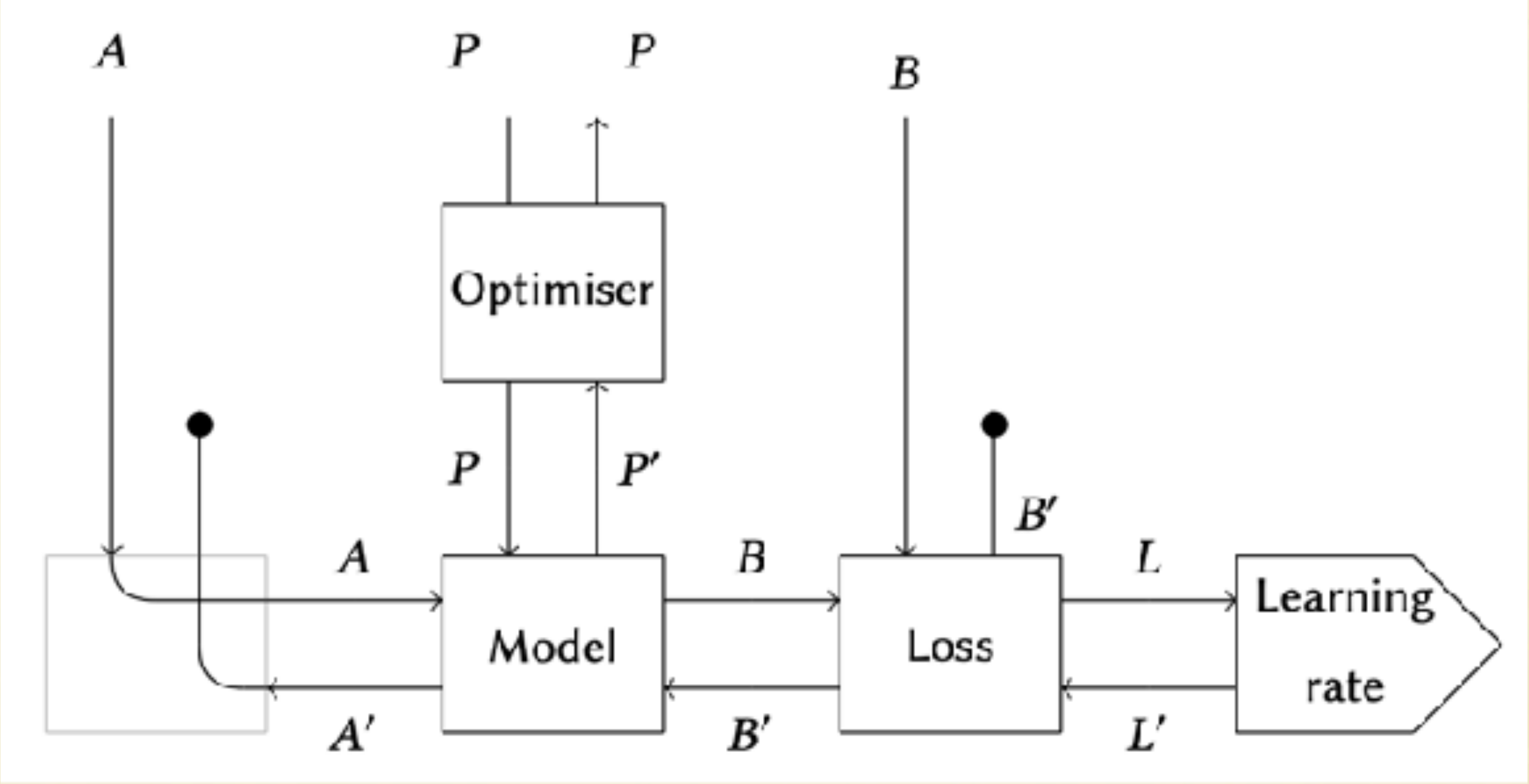
All components at work in the learning process are reduced to a single concept: parametric lenses.

Abstraction

A wide range of optimisers, models, loss maps, etc. are instances of the framework.

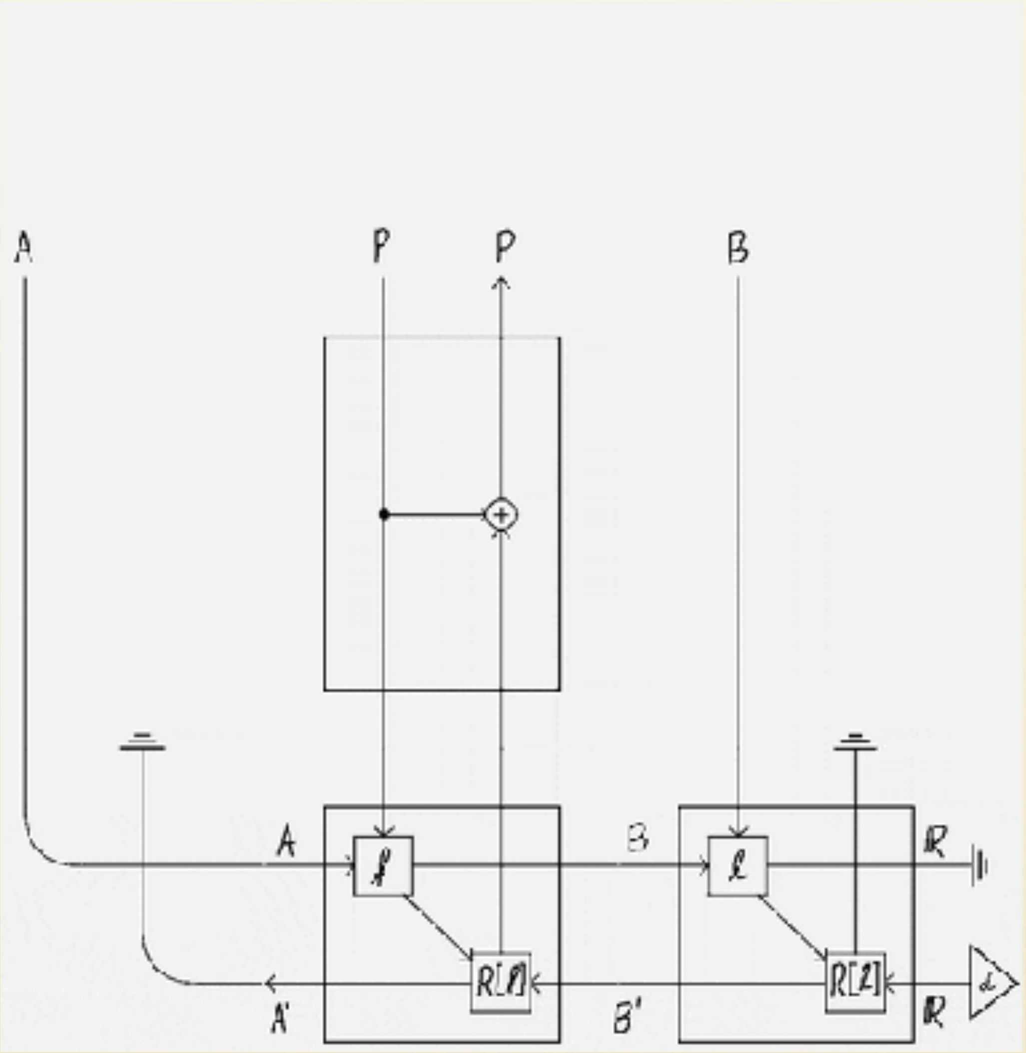
Discussion

Compositional reasoning



Discussion

Resource-sensitive Formalism



Discussion

Going forward

Lens-Theoretic Learning beyond gradient-descent

- Learning Neural Networks
- Learning Boolean Circuits
- Bayesian Update
- Automata Learning
- Learning Dynamical Systems

Discussion

Implementation

https://github.com/statusfailed/numeric-optics-python/

http://catgrad.com/p/reverse-derivative-ascent

Bibliography

G. Cruttwell, B. Gavranovic, N. Ghani, P. Wilson, F. Zanasi - *Categorical Foundations of Gradient-Based Learning*, ESOP 2022

P. Wilson, F. Zanasi - *Reverse Derivative Ascent: a Categorical Approach to Learning Boolean Circuits*, ACT 2020