Deep learning and differentiable programming

Luis Gonzalo Sanchez Giraldo

- https://youtu.be/_rdINNHLYaQ

Programming by example.

- Given some constraints on the desired behavior of a program (a set of input output pairs examples).
- Rather than writing the rules (program) that accomplish a desired behavior (for input A we want output to be B)
- Search the program space for a program that satisfies the constraints.

How to search?

- Discrete search.
- Relax the problem and make the components differentiable so we can apply gradient based optimization.

Making problem differentiable

- Think about a program as a function parameterized with a set of real numbers.
  - $Y = f(X; W)$.
  - For each W, the function f maps input X to Y in a different way.
  - Search is for the right function becomes search for W.
- In addition, input and output spaces must be relaxed (made continuous) as well as the criteria that evaluates the fit of a particular function.

Simple example

- Learn the OR function

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Simple example ...

- Similarly, we can learn the AND function or the NAND function.

X1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

Composition of functions

Composition of functions can give us more expressive power.

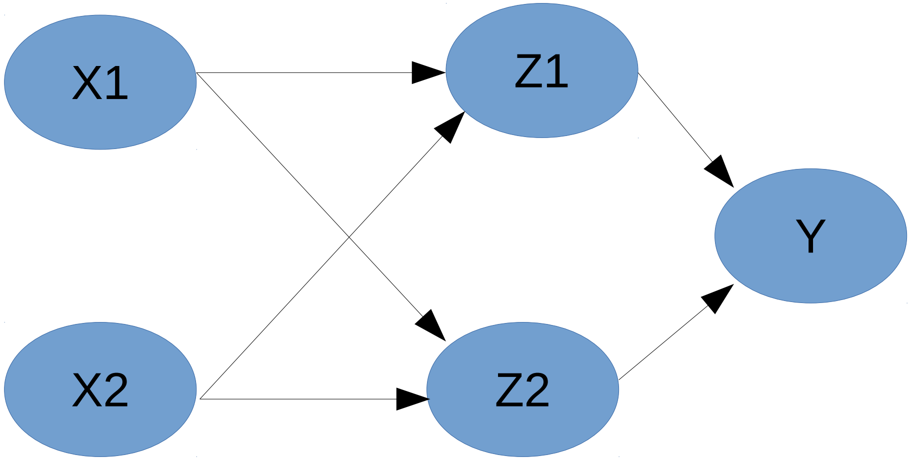
- Secondary Boolean functions such as XOR.

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Composition of functions

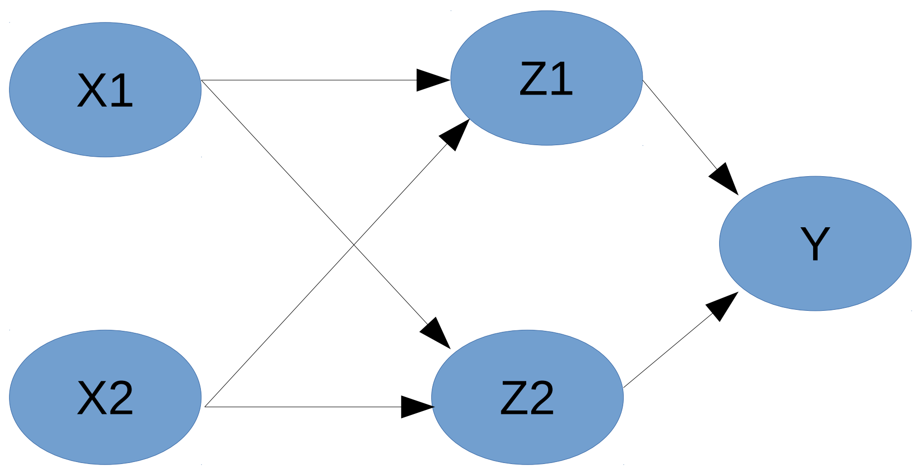
- Look at the expression for the XOR

  Y = ((X1 OR X2) OR (X1 NAND X2))
- Define intermediate variables
- Z1 = (X1 OR X2) and Z2 = (X1 NAND X2)
- Y = (Z1 OR Z2)



Composition of functions

- Look deeper into the derivatives of the function



- We want the derivatives of the fit function with respect to the parameters of the functions.

Composition of functions

- Chain rule