DS 4440 Computation Graphs

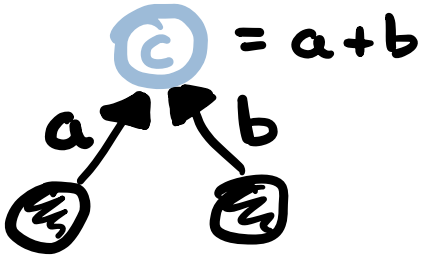
So far:

- Supervised learning
- SGD for linear models
- MLPs and non-linear models

Today: Computation Graphs as the core abstraction for Neural Networks.

The Big Idea (TM)

Models as Directed Acyclic Graphs (DAGs)



$c = a + b$

Nodes as operations (possibly identity), labeled as Variables.

Tensors (scalars above) "flow" along

edges —.

We care about forward and backward passes through such graphs.

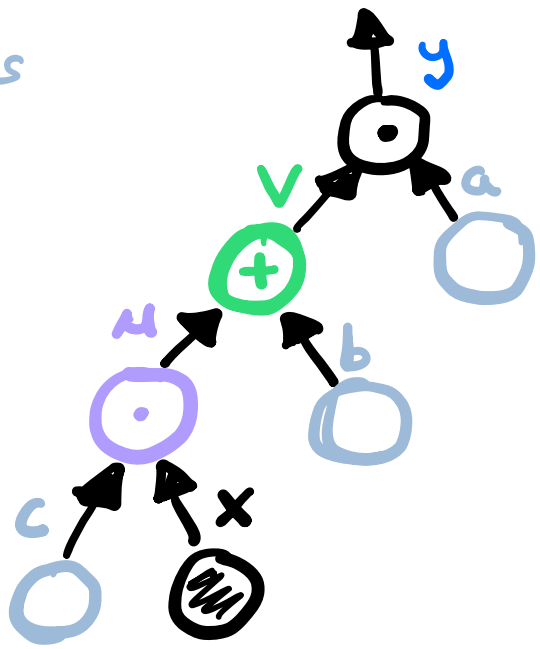
Consider

$$y = a \cdot (b + c \cdot x)$$

Parameters

$u = c \cdot x$

$v = b + u$

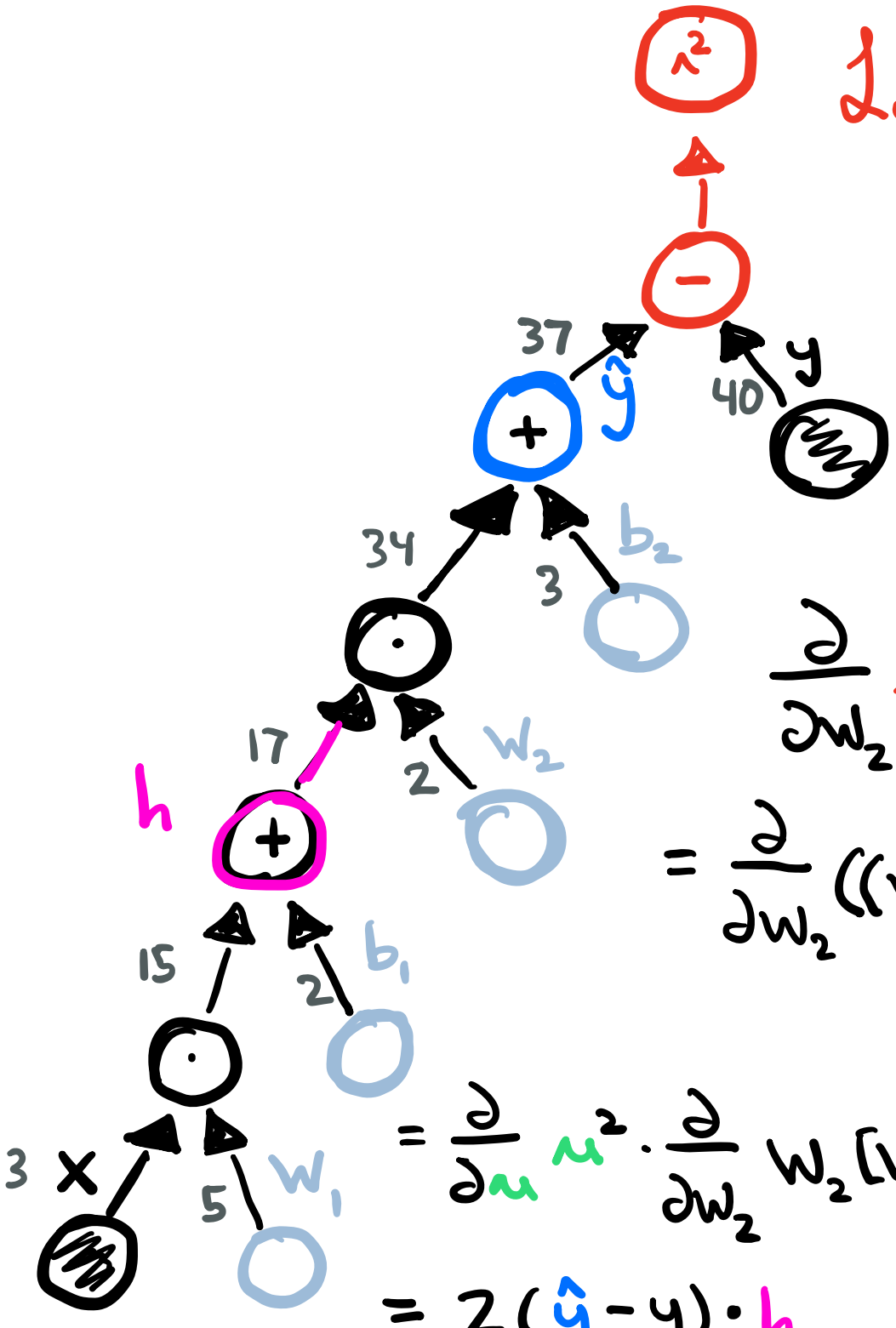


Let's build a graph for a simple regression model [Exercise! See NB]

$$\hat{y} = w_2 \cdot (w_1 \cdot x + b_1) + b_2$$

$$\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$$

Computation graph



$$\frac{\partial}{\partial W_2}\mathcal{L} = \frac{\partial}{\partial W_2}(\hat{y}-y)^2$$

$$= \frac{\partial}{\partial W_2}\underbrace{((\underbrace{W_2 \cdot [W_1 \cdot x + b_1]}_{h} + b_2) - y)^2}_{u}$$

$$= \frac{\partial}{\partial u} u^2 \cdot \frac{\partial}{\partial W_2} W_2[W_1 \cdot x + b_1] + b_2 - y$$

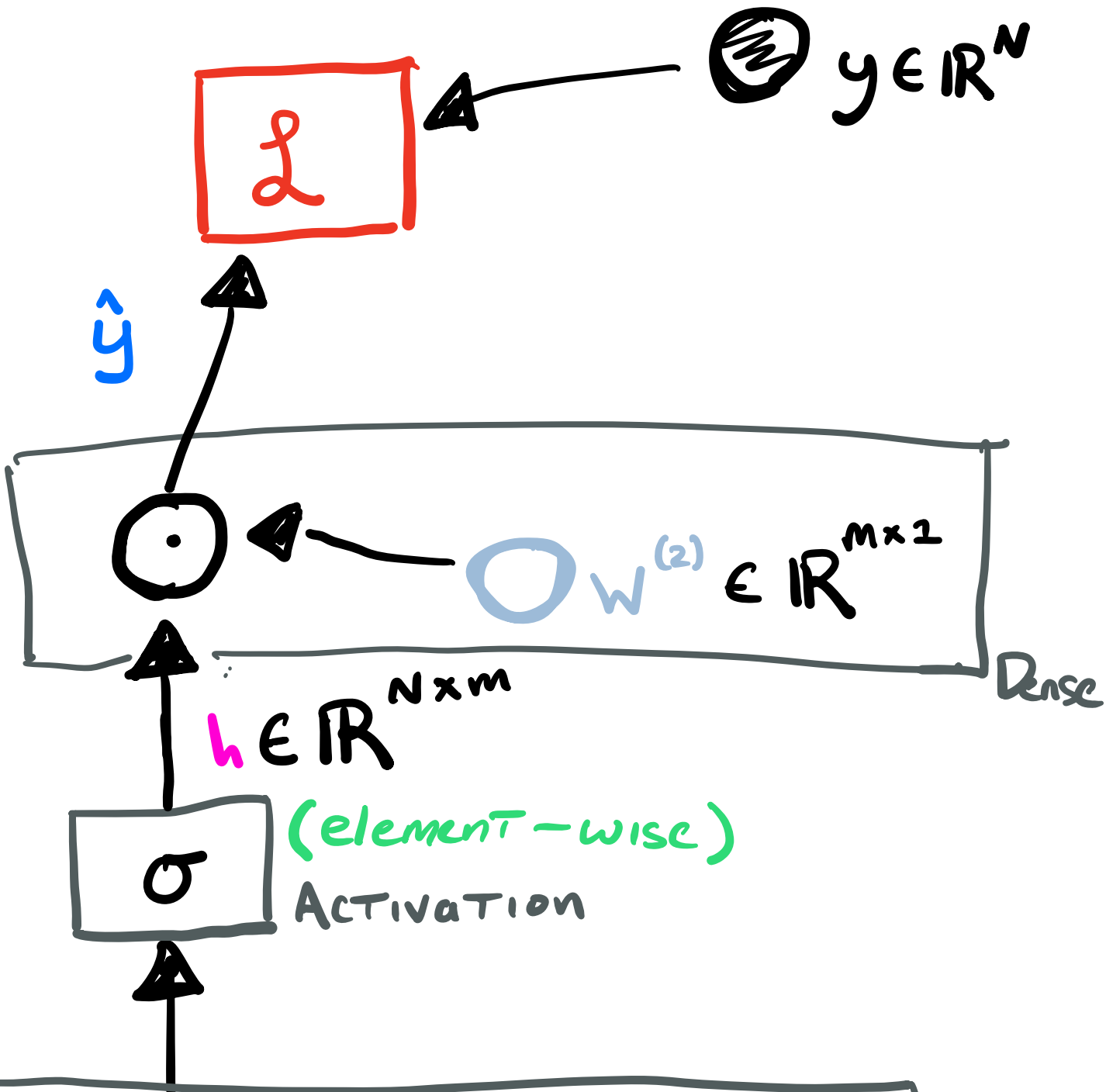
$$= 2(\hat{y}-y) \cdot h$$

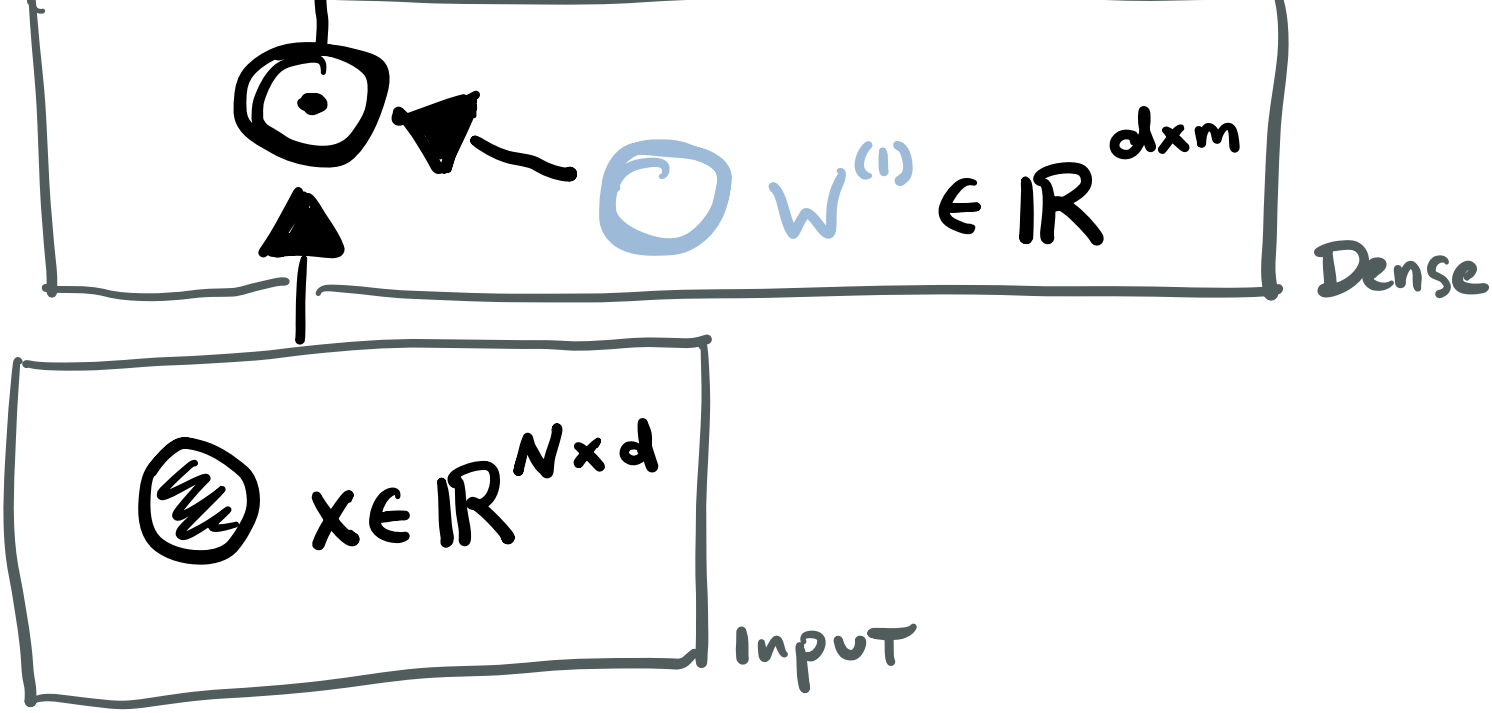
$x = 3;\ W_1 = 5;\ b_1 = 2;\ W_2 = 2;\ b_2 = 3 \quad y = 40$

(See CoLab)

We have been considering Scalars but in general Should Think of Tensors.

$$\hat{y} = W^{(2)} \cdot \sigma(W^{(1)} \cdot x)$$





Note that this can be assembled dynamically i.e. before a forward pass.

In code, each block is a Module which is a class.

We can chain these together.

(See Notebook & Textbook)