Deep Neural Networks Are Our Friends

Wang Ling

Google DeepMind

Outline

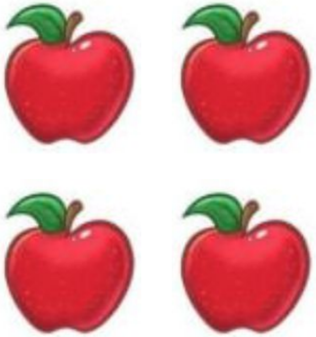
- Part I - Neural Networks are our friends
  - Numbers are our friends
  - Variables are our friends
  - Operators are our friends
  - Functions are our friends
  - Parameters are our friends
  - Cost Functions are our friends
  - Optimizers are our friends
  - Gradients are our friends

Outline

- Part I - Neural Networks are our friends
- Part 2 - Into Deep Learning
    - Nonlinear Neural Models
    - Multilayer Perceptrons
    - Using Discrete Variables
    - Example Applications

Numbers are our friends

Abby



How many apples does Abby have?

Numbers are our friends

Abby

4 

Variables are our friends

Abby

Bert

4 

5 

Variables are our friends

Abby

4 x

Bert

5 y

Operators are our friends

Bert

4 🍎

1 🍎

If Abby has 4 apples,
and gives Bert 1 apple,
how many apples will
Abby have?

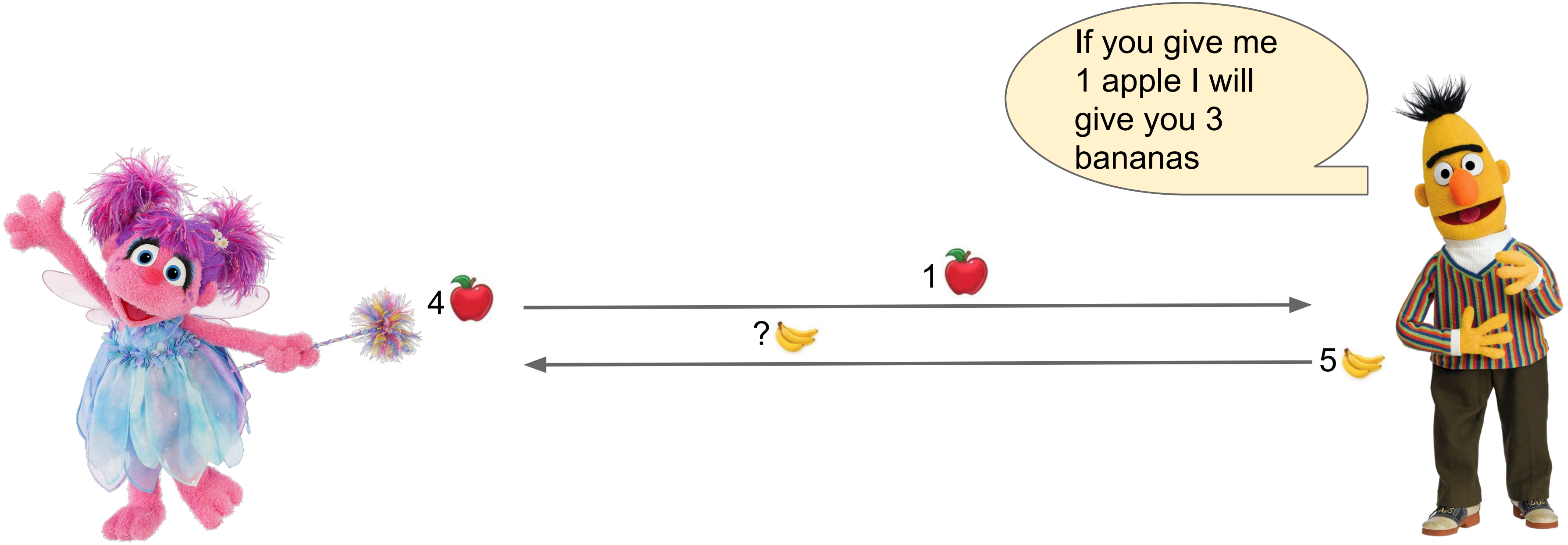
Operators are our friends

Bert

$4x - 1x = 3x$

3🍎 → 1🍎

Functions are our friends



Functions are our friends

- Input, x - Number of Apples given by Abby

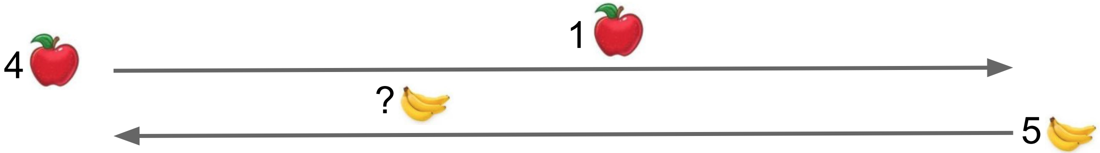
$$y = 3x$$

Functions are our friends

$y = 3x$

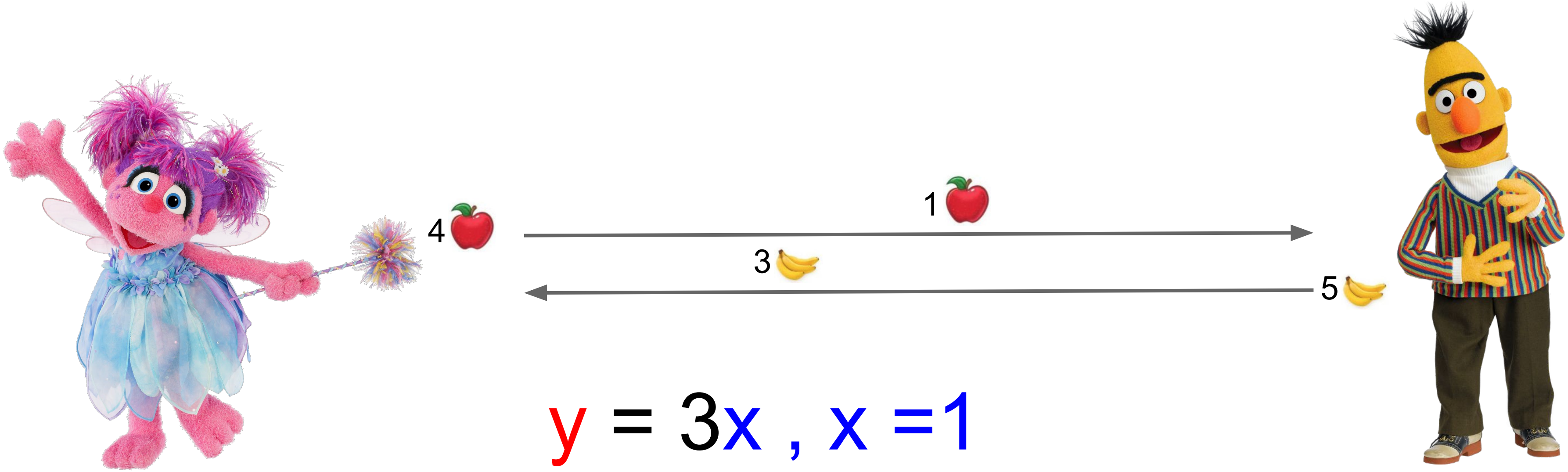
- Input, x - Number of Apples given by Abby
- Output, y - Number of Bananas received by Abby

Functions are our friends



$y = 3x\ ,\ x = 1$

Functions are our friends



$y = 3x$, $x = 1$

$y = 3$

Functions are our friends

$y = 3x$

Functions are our friends

x : English Sentence

Google
Translate
Break through language barriers.

y : Spanish Sentence

Functions are our friends

x : Board

y : Move

Functions are our friends

x : Image

y : Category

Functions are our friends

x : Board

??????????????????????????

y : Move

Functions are our friends

$y = 3x$

Cookie Monster

Functions are our friends

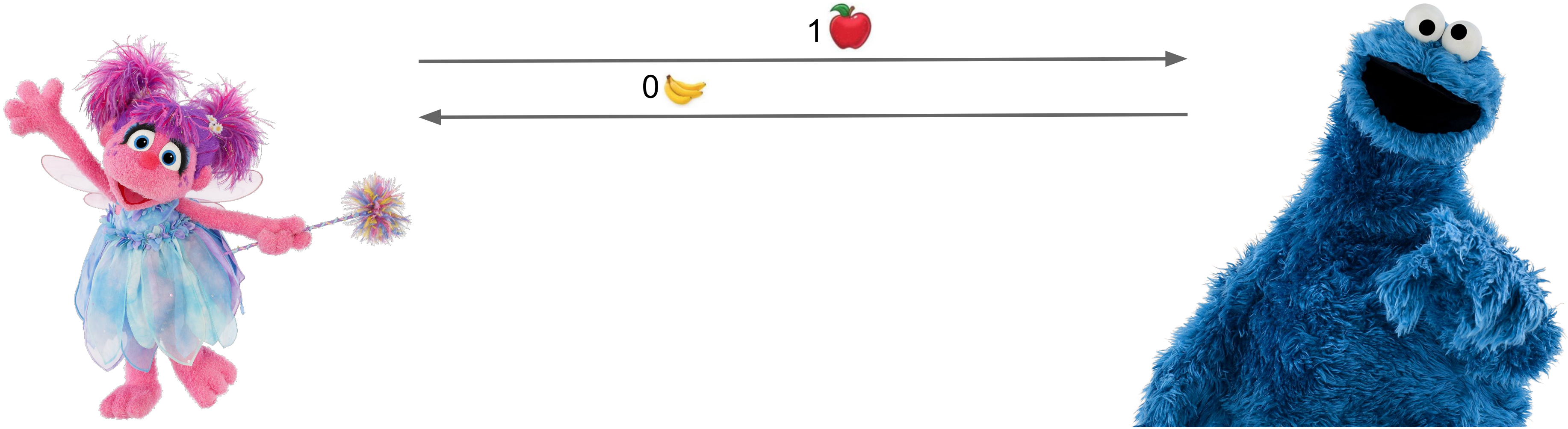
Find it out for yourself

$y = ??$

$y = 3x$

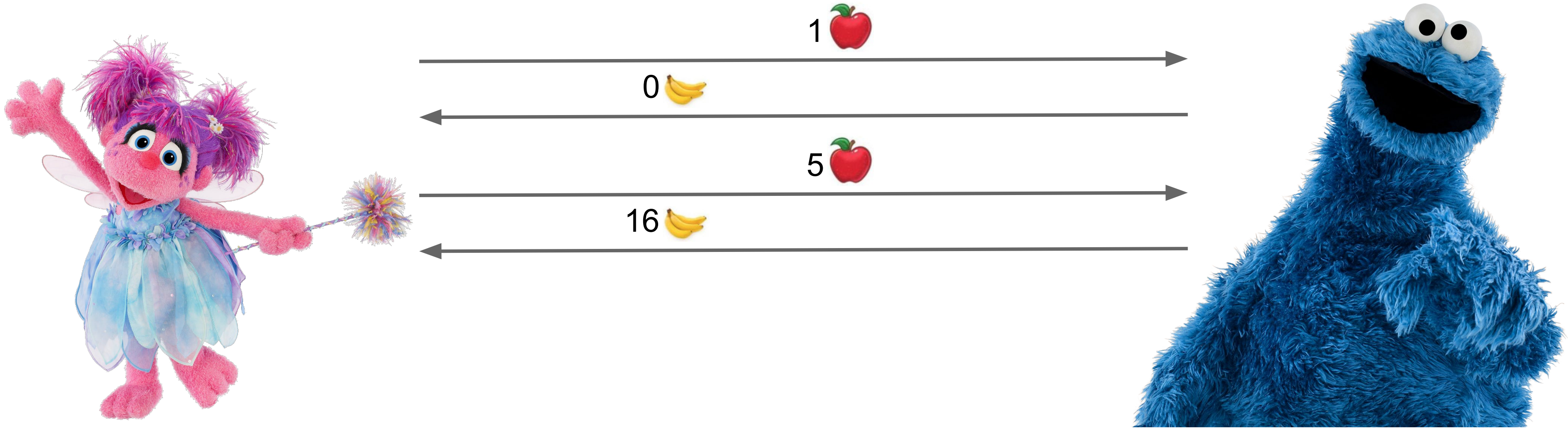
Functions are our friends

y = ??



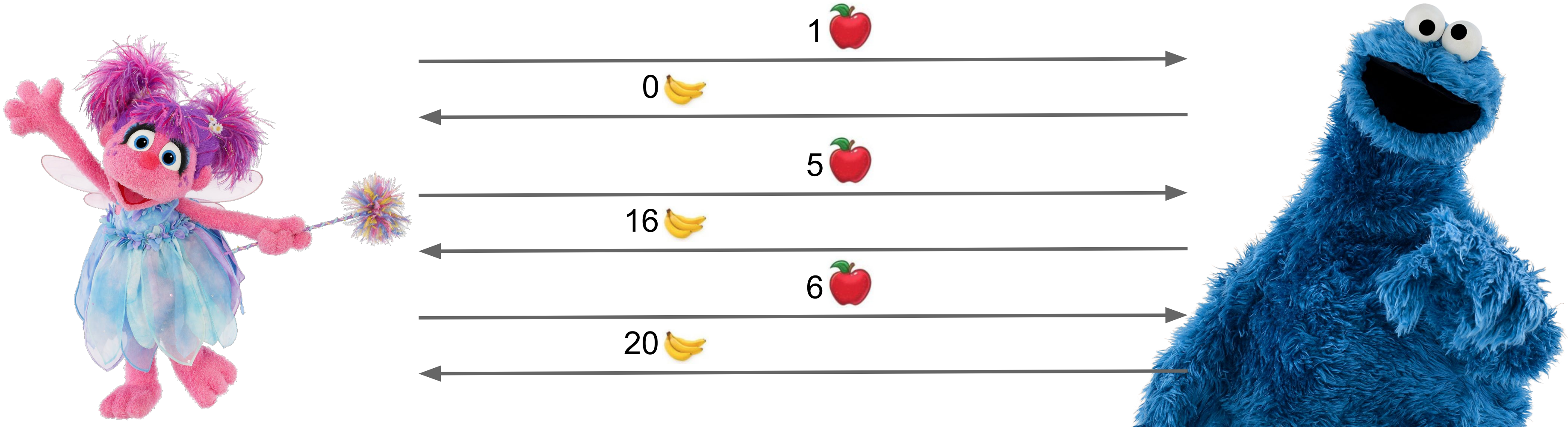
Functions are our friends

y = ??



Functions are our friends

y = ??



Functions are our friends

I want to know how many bananas I get,
but I ran out of apples....

y = ??

1 🍎 →

← 0 🍌

5 🍎 →

← 16 🍌

6 🍎 →

← 20 🍌

3 🍎 →

← ? 🍌

Parameters are our friends

- Input
- Output

$$y = 3x + 1$$

Parameters are our friends

Model

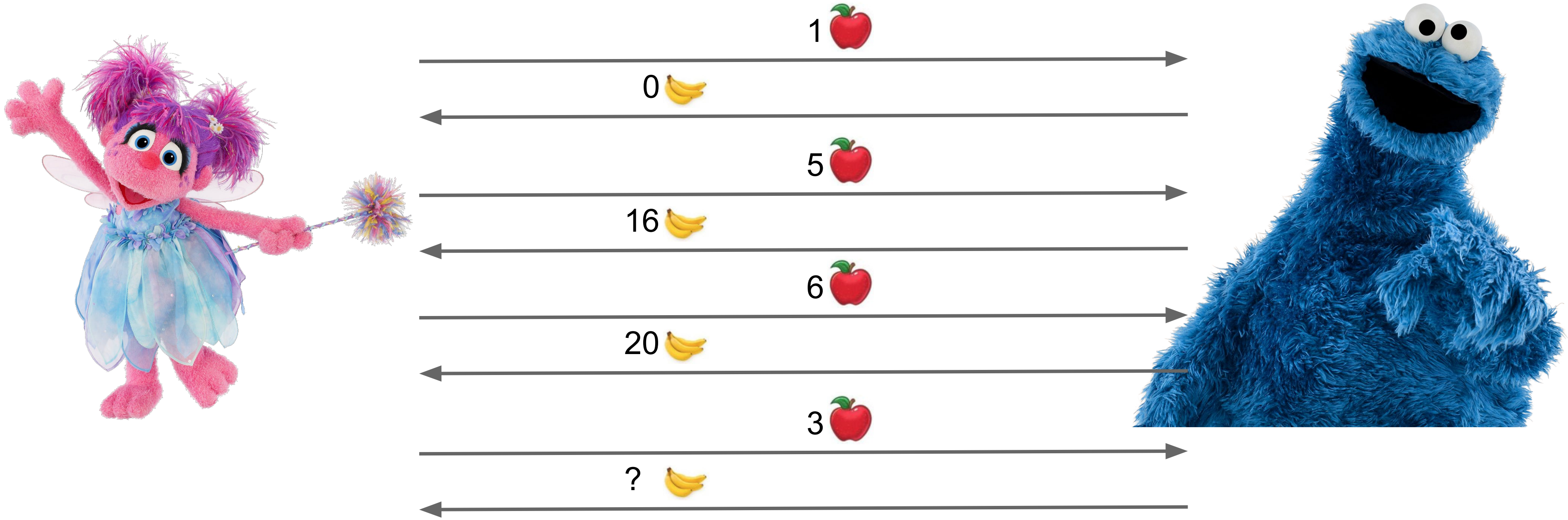
$$y = wx + b$$

- Input
- Output
- Parameters

Input - Fixed, comes from data
Parameters - Need to be estimated

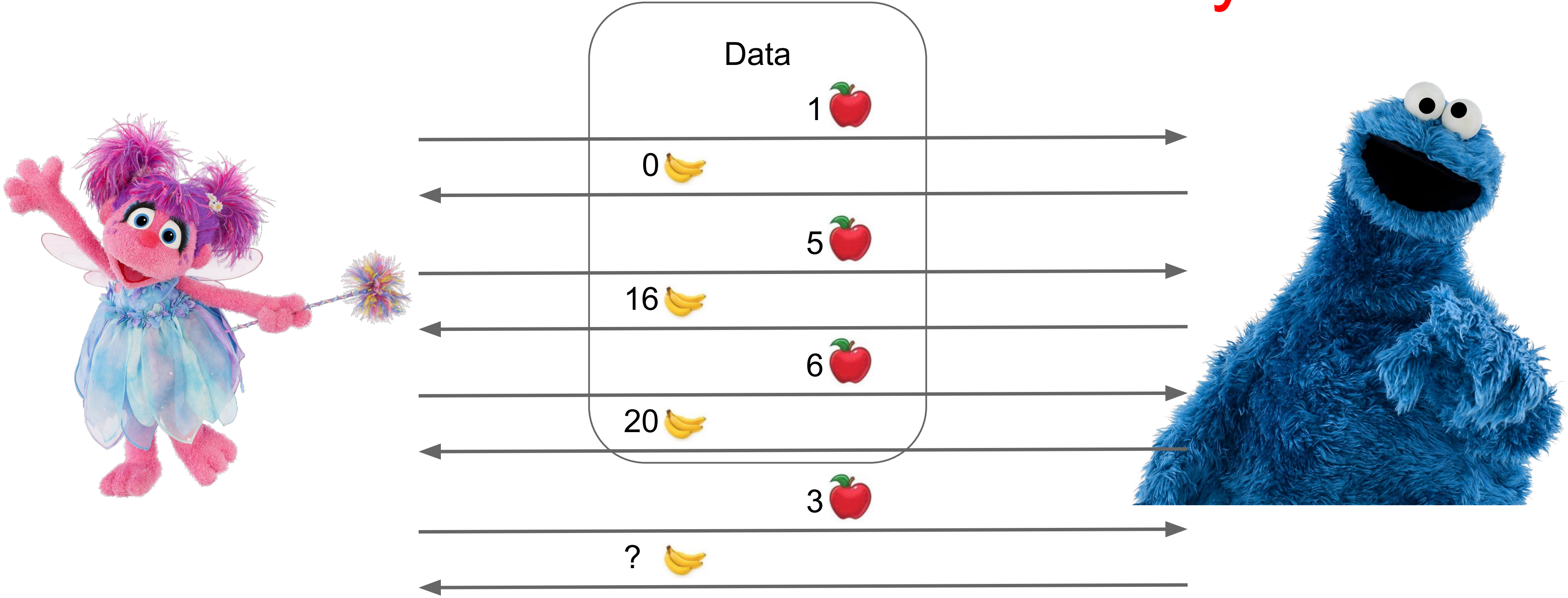
Parameters are our friends

$y = wx + b$



Parameters are our friends

$y = wx + b$



Parameters are our friends

$$y = wx + b$$

Data	
x	$\hat{y}$
1	0
5	16
6	20

3 🍎

? 🍌

Parameters are our friends

Data

x	ŷ
1	0
5	16
6	20

Model

$$y = wx + b$$

Parameters are our friends

Data

x	ŷ
1	0
5	16
6	20

Model

$$y = wx + b$$

How to find the parameters w and b?

Parameters are our friends

Data

x	$\hat{y}$
1	0
5	16
6	20

Model

$$y = wx + b$$

Model Candidate 1

$$y = 1x + 0$$

x	y
1	0
5	16
6	20

Parameters are our friends

Data

x	ŷ
1	0
5	16
6	20

Model

$$y = wx + b$$

Model Candidate 1

$$y = 1x + 0$$
$$1 = 1*1 + 0$$
$$5 = 1*5 + 0$$
$$6 = 1*6 + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Parameters are our friends

Data

x	y
1	0
5	16
6	20

Model

$$y = wx + b$$

Model Candidate 1

$$y = 1x + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Parameters are our friends

Data

x	y
1	0
5	16
6	20

Model

$$y = wx + b$$

Model Candidate 1

$$y = 1x + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Which one is better ?

Parameters are our friends

Data

x	y
1	0
5	16
6	20

Model

$$y = wx + b$$

Model Candidate 1

$$y = 1x + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Model Candidate 1

$$y = 1x + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b)$$

Model Candidate 1

$$y = 1x + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

Square Loss

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

x	ŷ	y
1	0	1
5	16	5
6	20	6

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

n	x	ŷ	y	(y-ŷ)²
0	1	0	1	
1	5	16	5	
2	6	20	6	

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

n	x	ŷ	y	(y-ŷ)²
0	1	0	1	1
1	5	16	5	
2	6	20	6	

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = w x_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	1	1
1	5	16	5	121
2	6	20	6	

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n\in\{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	1	1
1	5	16	5	121
2	6	20	6	196

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	1	1
1	5	16	5	121
2	6	20	6	196
			C(1,0)	318

Model Candidate 2

$$y = 2x + 2$$

x	ŷ	y
1	0	4
5	16	12
6	20	14

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	1	1
1	5	16	5	121
2	6	20	6	196
			C(1,0)	318

Model Candidate 2

$$y = 2x + 2$$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	4	16
1	5	16	12	16
2	6	20	14	36
			C(2,2)	68

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = w x_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Model Candidate 1

$$y = 1x + 0$$

$C(1,0)$ 318

Model Candidate 2

$$y = 2x + 2$$

$C(2,2)$ 68

Cost functions are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

How to find the parameters w and b?

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Optimizers are our friends

Data

n	x	y
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2$$

Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

Optimizers are our friends

Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

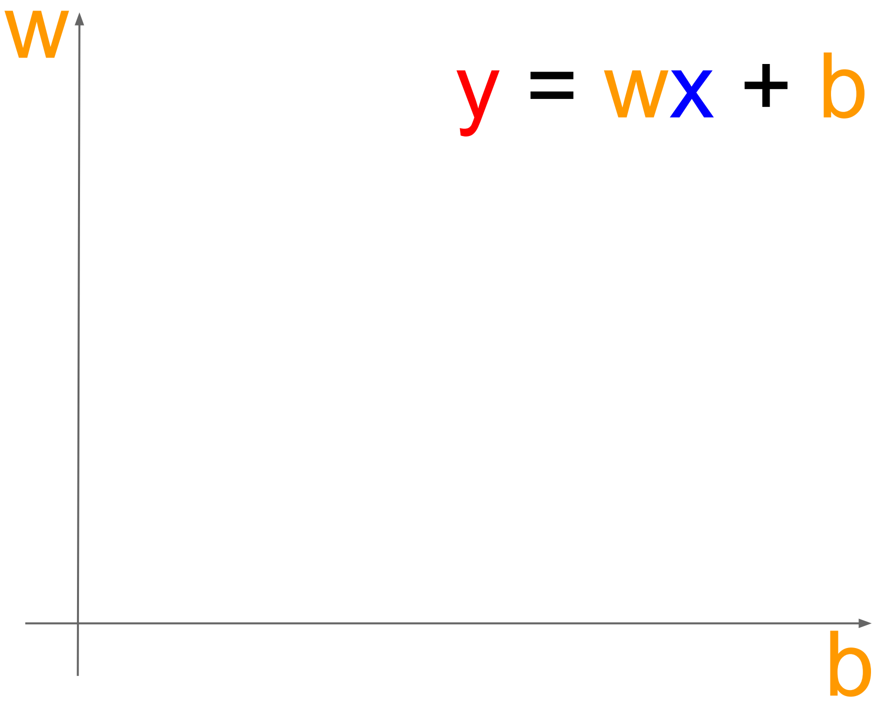
Optimizers are our friends

Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

$w_0, b_0 = 2,2 : C(w_0, b_0) = 68$

$y = wx + b$



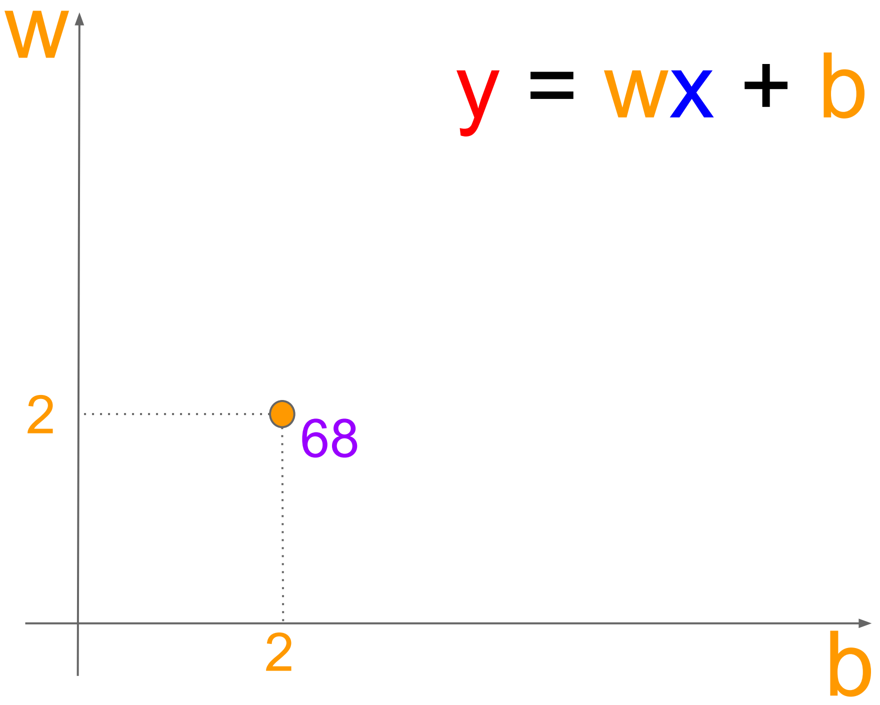
Optimizers are our friends

Optimizer

$$\arg\min C(w,b)$$
$$w,b \in [-\infty,\infty]$$

$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$

$$y = wx + b$$



Optimizers are our friends

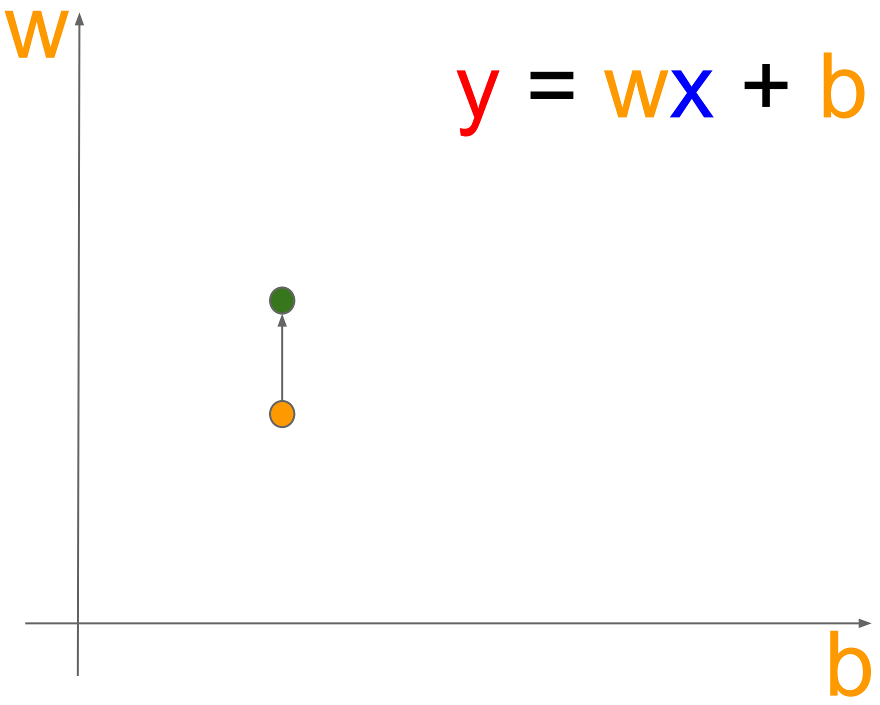
Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$

$w_1,b_1 = 3,2 : C(w_1,b_1) = ?$

$y = wx + b$



Optimizers are our friends

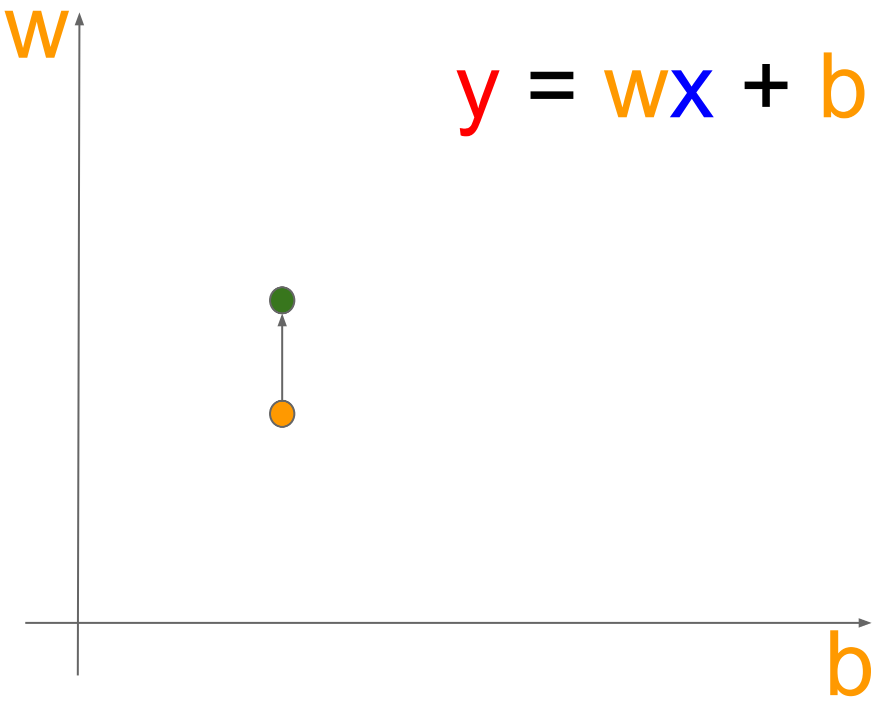
Optimizer

$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_0, b_0 = 2,2 : C(w_0, b_0) = 68$

$w_1, b_1 = 3,2 : C(w_1, b_1) = 26$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	5	25
1	5	16	17	1
2	6	20	20	0
			C(3,2)	26



Optimizers are our friends

Optimizer

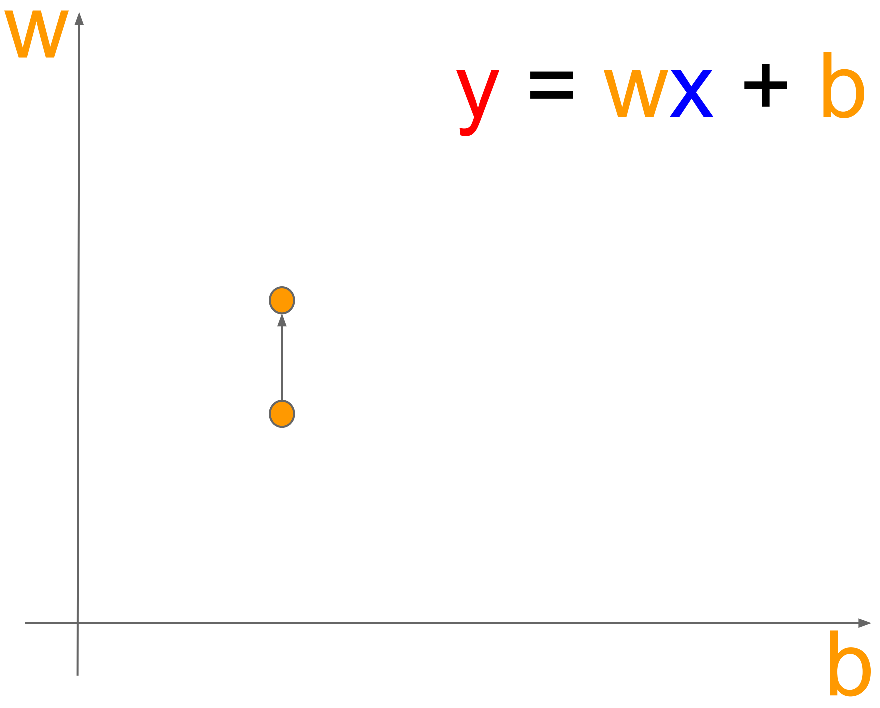
$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$

$w_1,b_1 = 3,2 : C(w_1,b_1) = 26$

n	x	ŷ	y	$(y-ŷ)^2$
0	1	0	5	25
1	5	16	17	1
2	6	20	20	0
			C(3,2)	26

$y = wx + b$



Optimizers are our friends

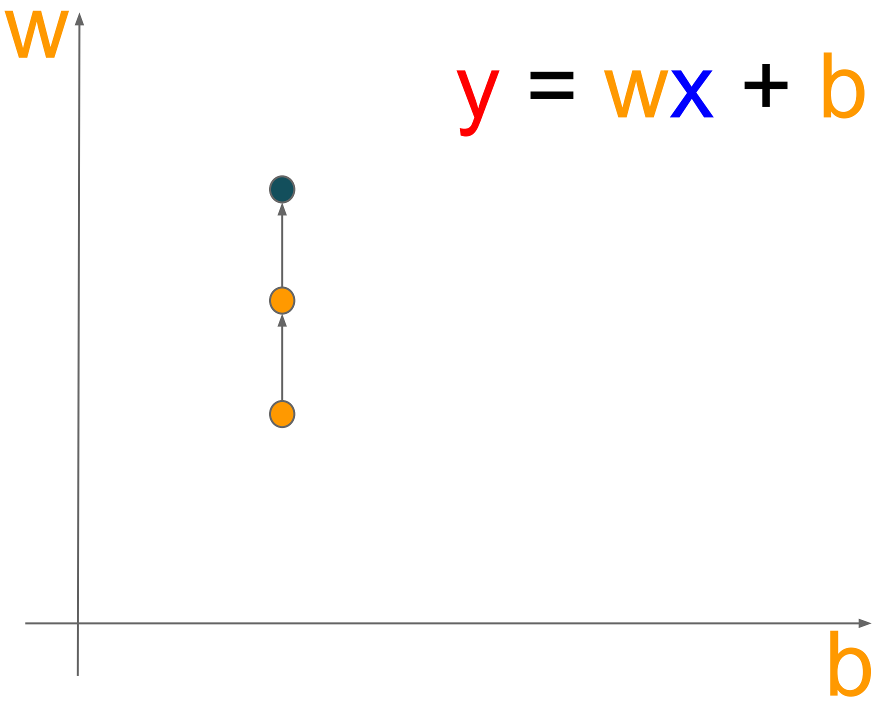
Optimizer

$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_1, b_1 = 3,2 : C(w_1, b_1) = 26$

$w_2, b_2 = 4,2 : C(w_2, b_2) = ??$

$y = wx + b$



Optimizers are our friends

Optimizer

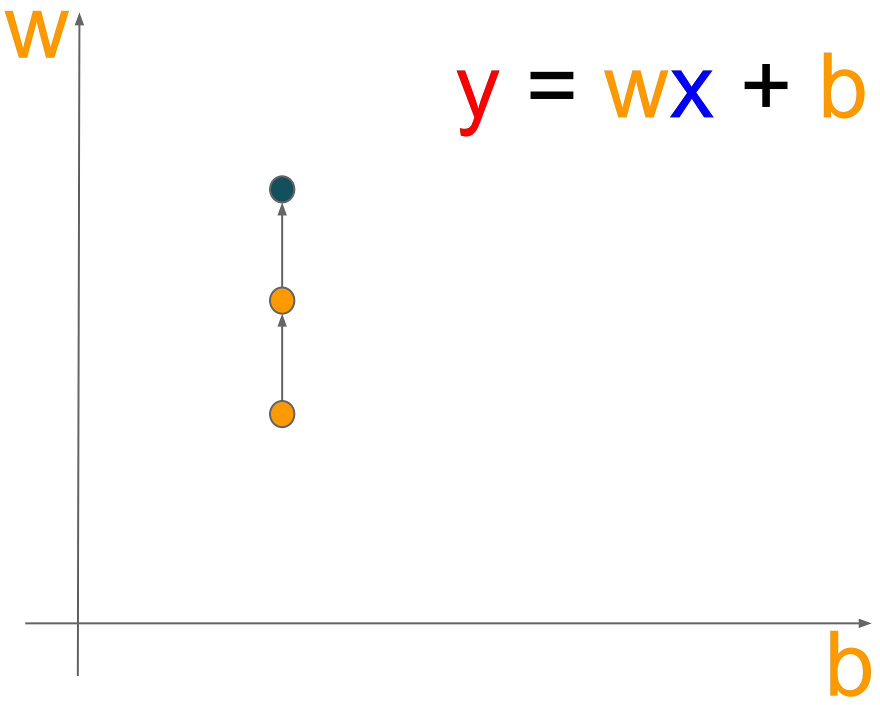
$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_1,b_1 = 3,2 : C(w_1,b_1) = 26$

$w_2,b_2 = 4,2 : C(w_2,b_2) = 136$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	6	36
1	5	16	22	64
2	6	20	26	36
			C(4,2)	136

$y = wx + b$



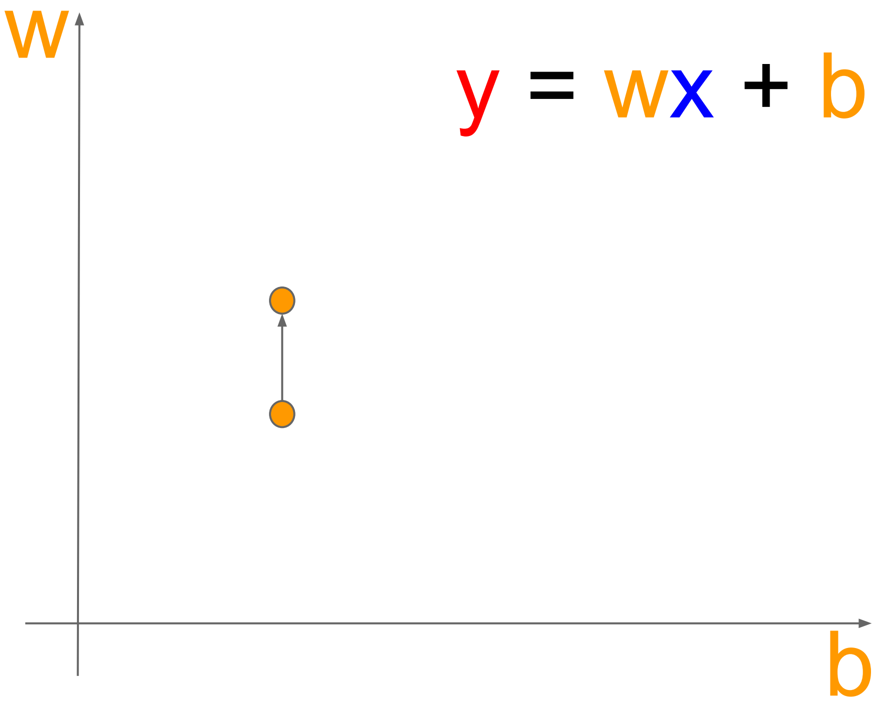
Optimizers are our friends

Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

$w_1, b_1 = 3,2 : C(w_1, b_1) = 26$

$y = wx + b$



Optimizers are our friends

Optimizer

$$\arg\min C(w,b)$$

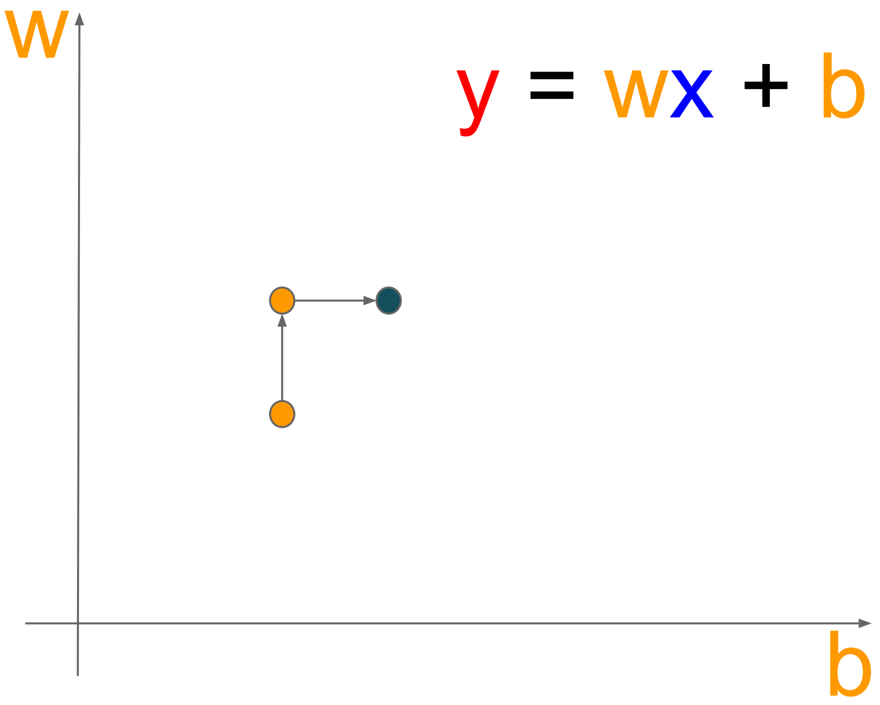
$$w,b \in [-\infty,\infty]$$

$w_1,b_1 = 3,2 : C(w_1,b_1) = 26$

$w_2,b_2 = 3,3 : C(w_2,b_2) = 41$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	6	36
1	5	16	18	4
2	6	20	21	1
			C(3,3)	41

$y = wx + b$



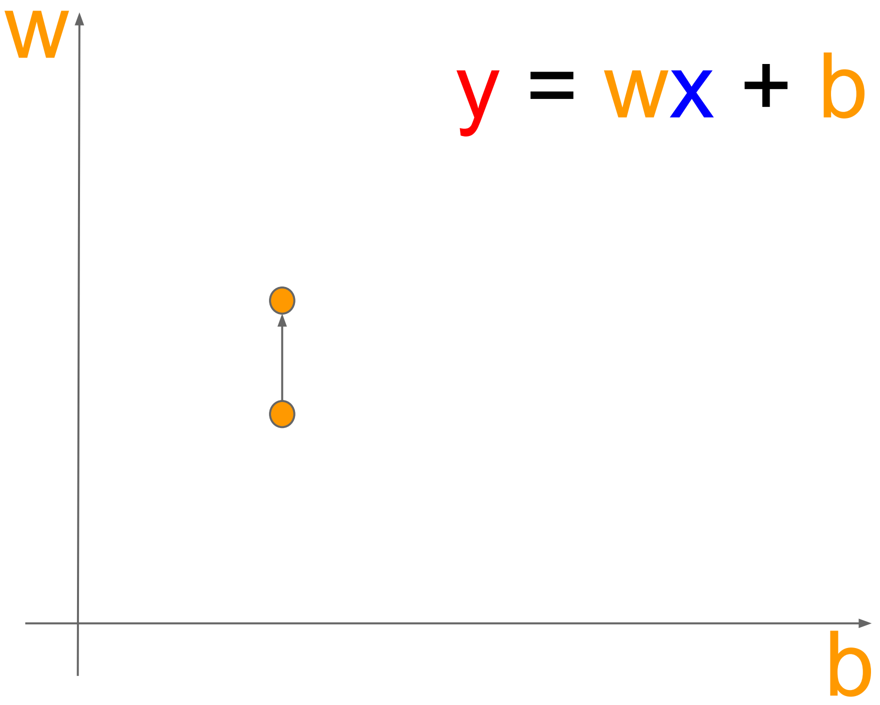
Optimizers are our friends

Optimizer

$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$$w_1, b_1 = 3,2 : C(w_1, b_1) = 26$$

$$y = wx + b$$



Optimizers are our friends

Optimizer

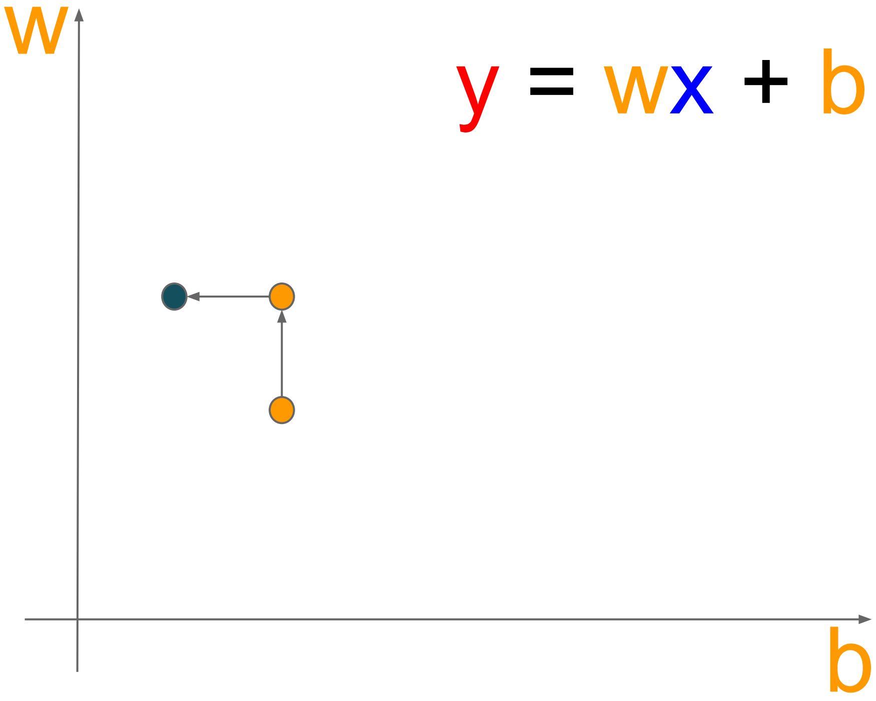
$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_1,b_1 = 3,2 : C(w_1,b_1) = 26$

$w_2,b_2 = 3,1 : C(w_2,b_2) = 17$

n	x	$\hat{y}$	y	$(y-\hat{y})^2$
0	1	0	4	16
1	5	16	16	0
2	6	20	19	1
			$C(3,1)$	17

$$y = wx + b$$



Optimizers are our friends

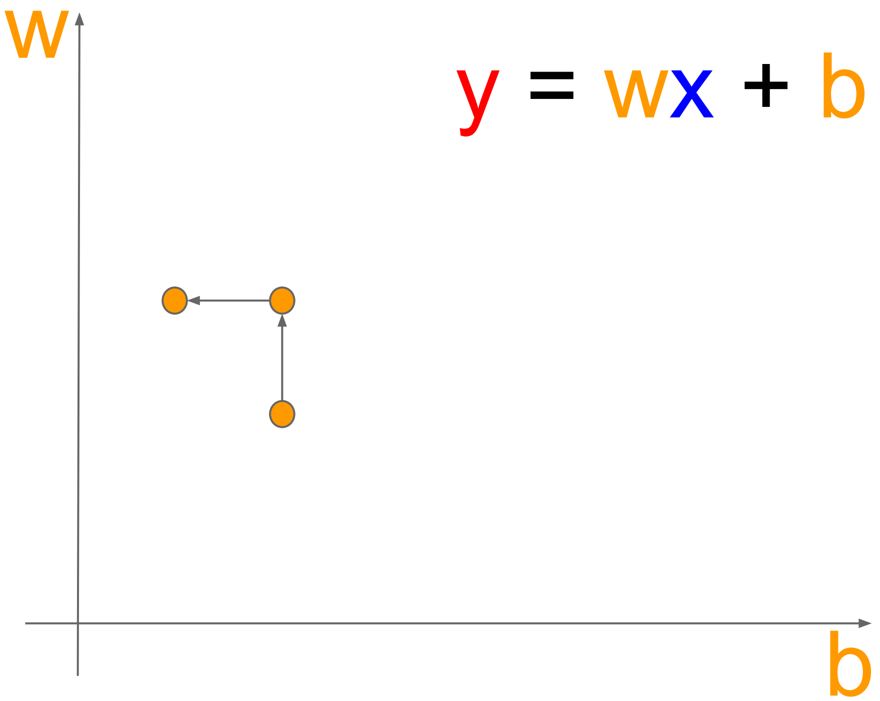
Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty,\infty]$

$w_2,b_2 = 3,1 : C(w_2,b_2) = 17$

$y = wx + b$



Optimizers are our friends

Optimizer

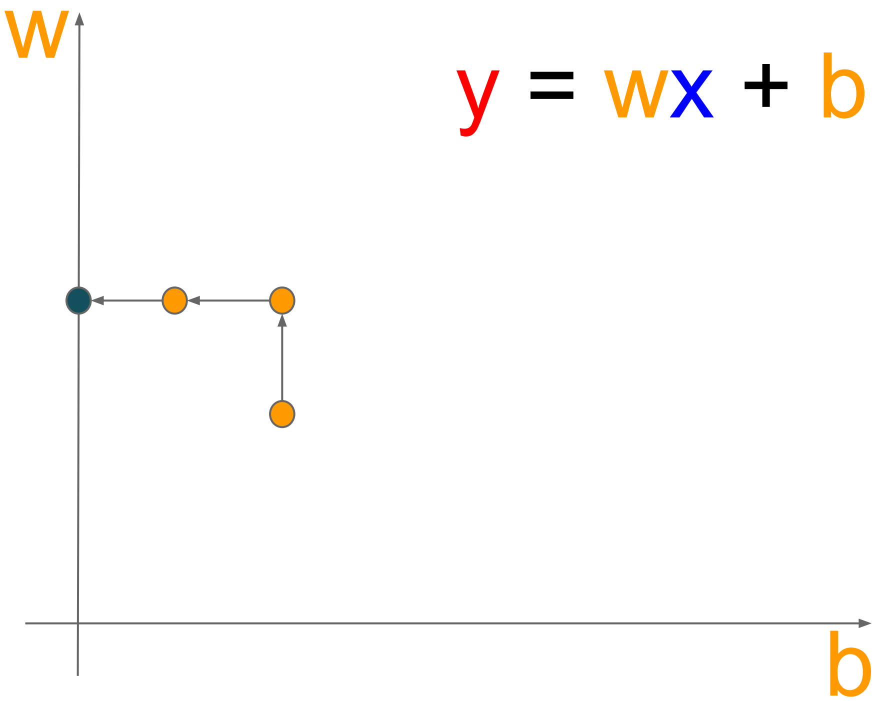
$$\arg\min C(w,b)$$

$w,b \in [-\infty,\infty]$

$w_2,b_2 = 3,1 : C(w_2,b_2) = 17$

$w_3,b_3 = 3,0 : C(w_3,b_3) = 13$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	3	9
1	5	16	15	1
2	6	20	18	4
			C(3,0)	13



$y = wx + b$

Optimizers are our friends

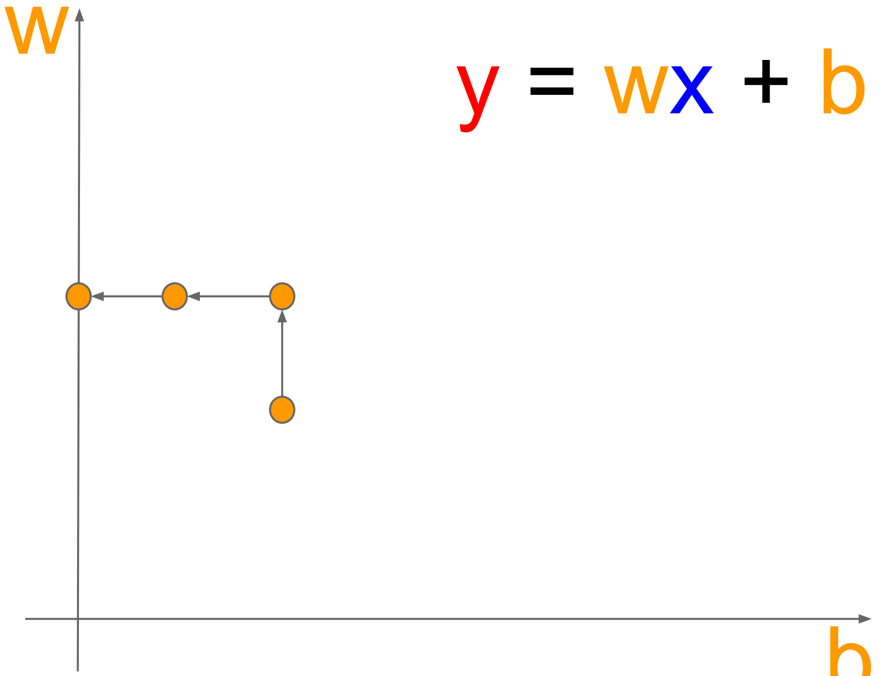
Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty,\infty]$

$w_3,b_3 = 3,0 : C(w_3,b_3) = 13$

$y = wx + b$



Optimizers are our friends

Optimizer

$$\color{steelblue}{\arg\min}\ \color{purple}{C(}\color{orange}{w,b}\color{purple}{)}$$

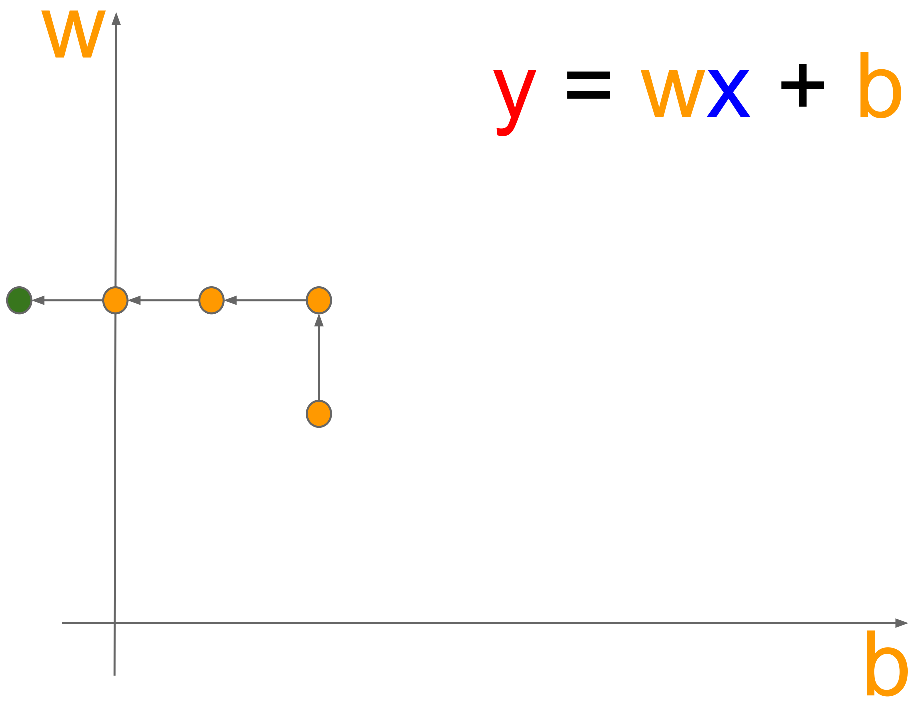
$$w,b \in [-\infty,\infty]$$

$w_3,b_3 = 3,0 : C(w_3,b_3) = 13$

$w_4,b_4 = 3,-1 : C(w_4,b_4) = 17$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	2	4
1	5	16	14	4
2	6	20	17	9
			C(3,-1)	17

$$y = wx + b$$



Optimizers are our friends

Optimizer

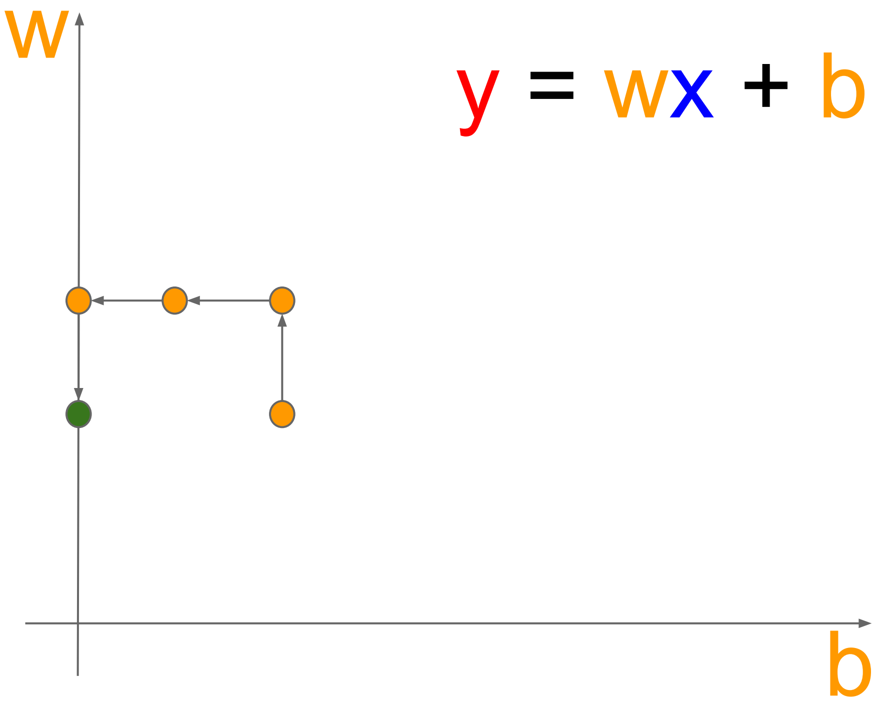
$$\arg\min C(w,b)$$

$w,b \in [-\infty,\infty]$

$w_3,b_3 = 3,0 : C(w_3,b_3) = 13$

$w_4,b_4 = 2,0 : C(w_4,b_4) = 104$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	2	4
1	5	16	10	36
2	6	20	12	64
			C(2,0)	104



$$y = wx + b$$

Optimizers are our friends

Optimizer

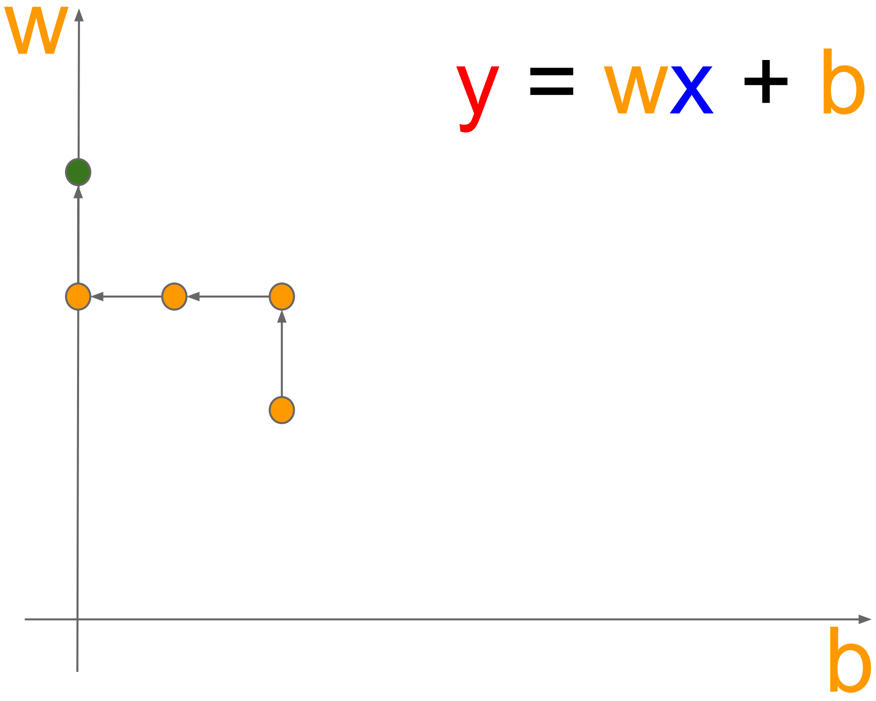
$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_3,b_3 = 3,0 : C(w_3,b_3) = 13$

$w_4,b_4 = 4,0 : C(w_4,b_4) = 104$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	4	16
1	5	16	20	16
2	6	20	24	16
			C(2,0)	54

$y = wx + b$



Optimizers are our friends

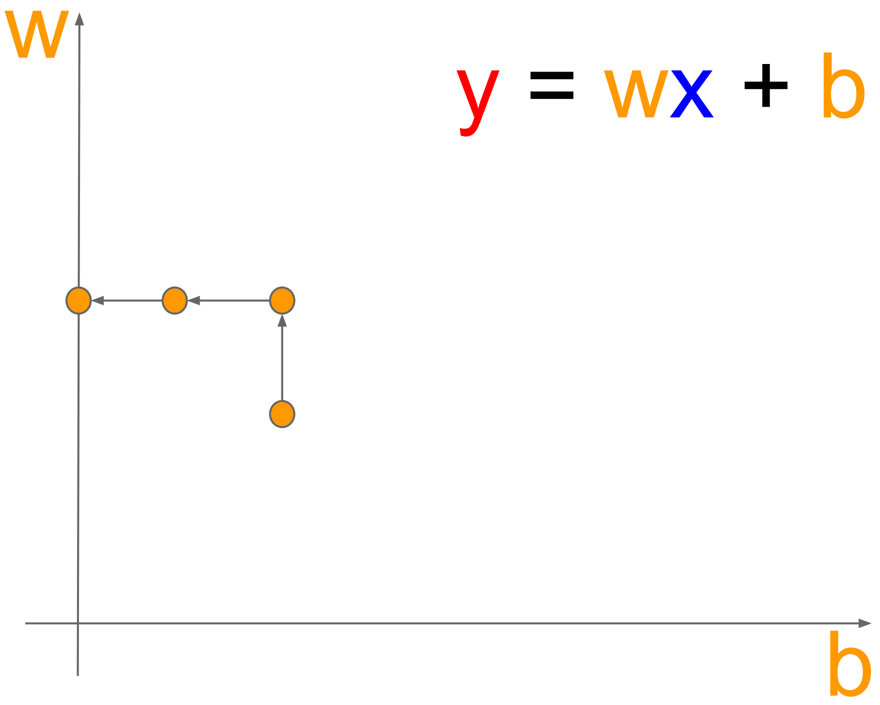
Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty, \infty]$

$w_3, b_3 = 3,0 : C(w_3, b_3) = 13$

The End?



$y = wx + b$

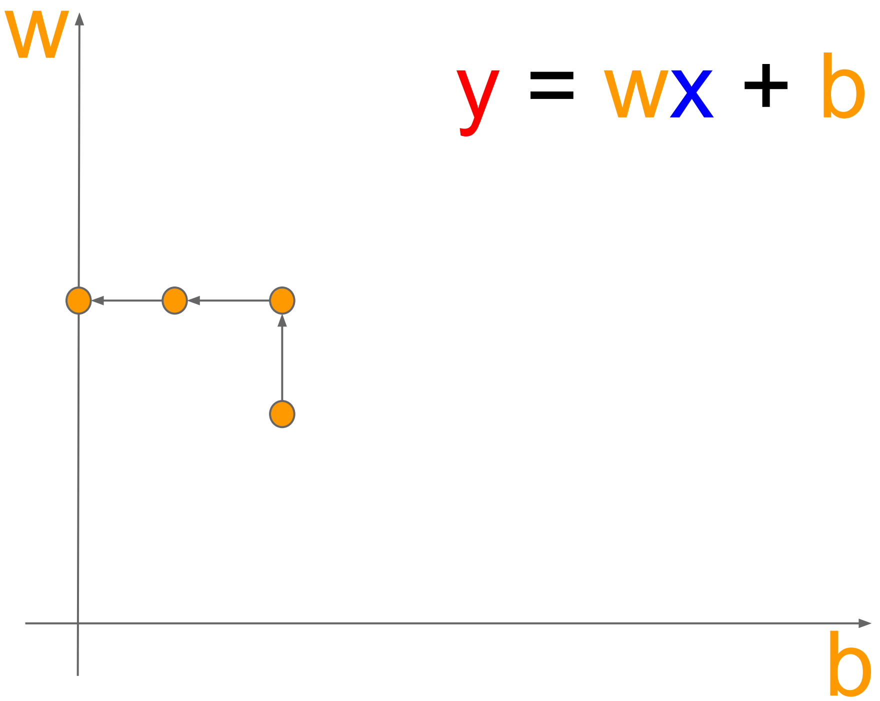
Optimizers are our friends

Optimizer

$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_?, b_? = 4, -2 : C(w_?, b_?) = ??$

$y = wx + b$



Optimizers are our friends

Optimizer

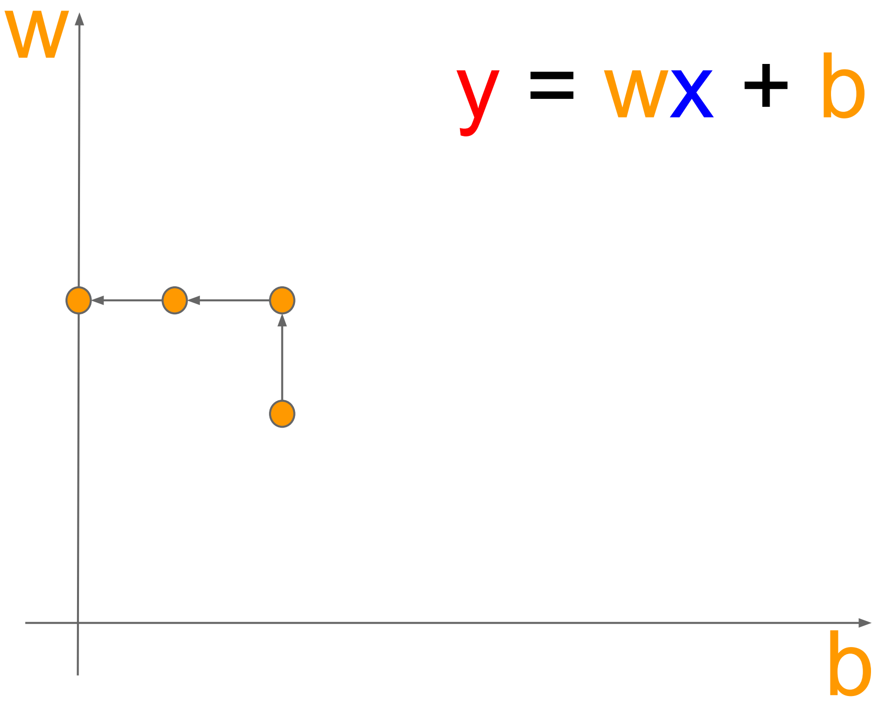
$$\arg\min C(w,b)$$

$$w,b \in [-\infty,\infty]$$

w?,b? = 4,-2 : C(w?,b?) = 12

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	2	4
1	5	16	18	4
2	6	20	22	4
			C(4,-2)	12

$$y = wx + b$$



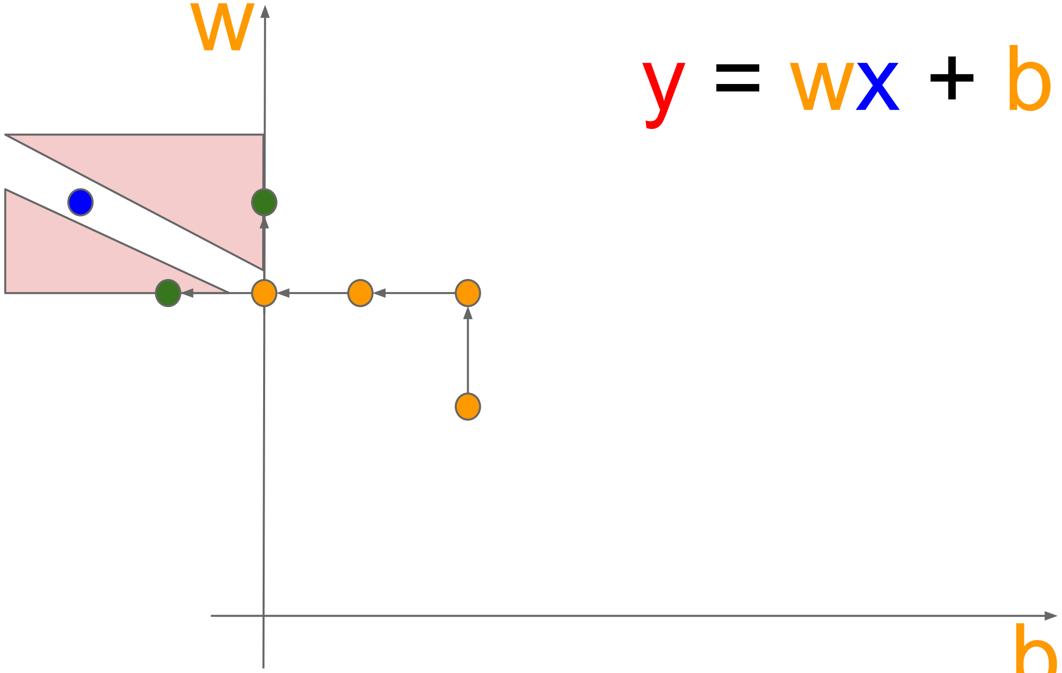
Optimizers are our friends

Optimizer

$$\underset{w,b \in [-\infty, \infty]}{\arg\min}\ C(w,b)$$

$w_3, b_3 = 3, 0 : C(w_3, b_3) = 13$

$y = wx + b$

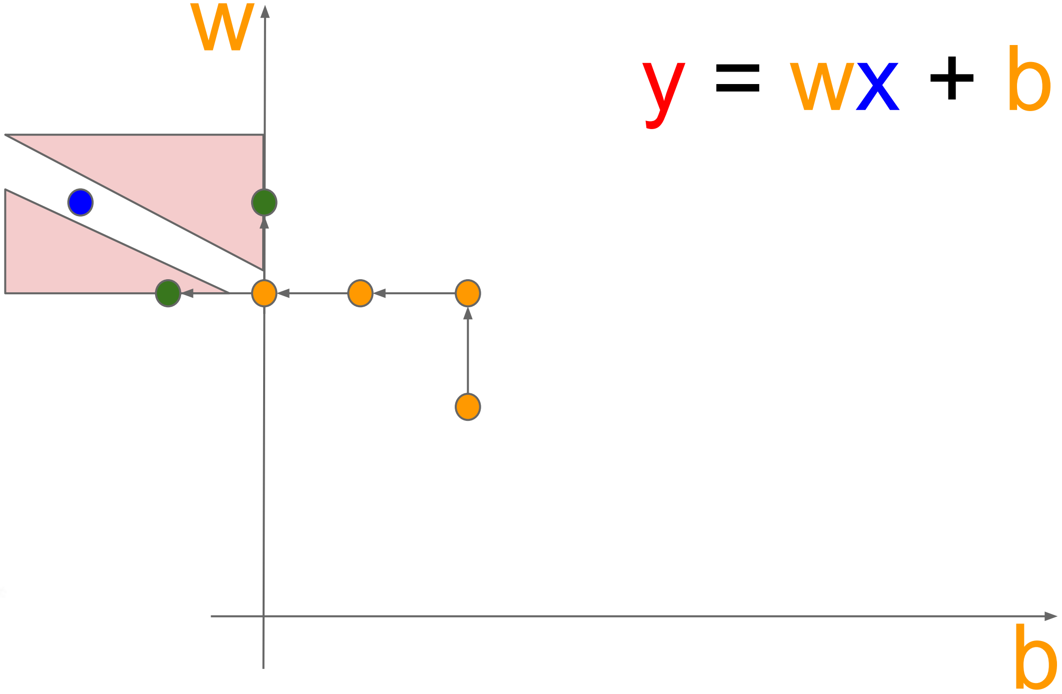


Optimizers are our friends

Optimizer

$$\arg\min_{w,b \in [-\infty, \infty]} C(w,b)$$

$w_3, b_3 = 3,0 : C(w_3, b_3) = 13$



$y = wx + b$

Optimizers are our friends

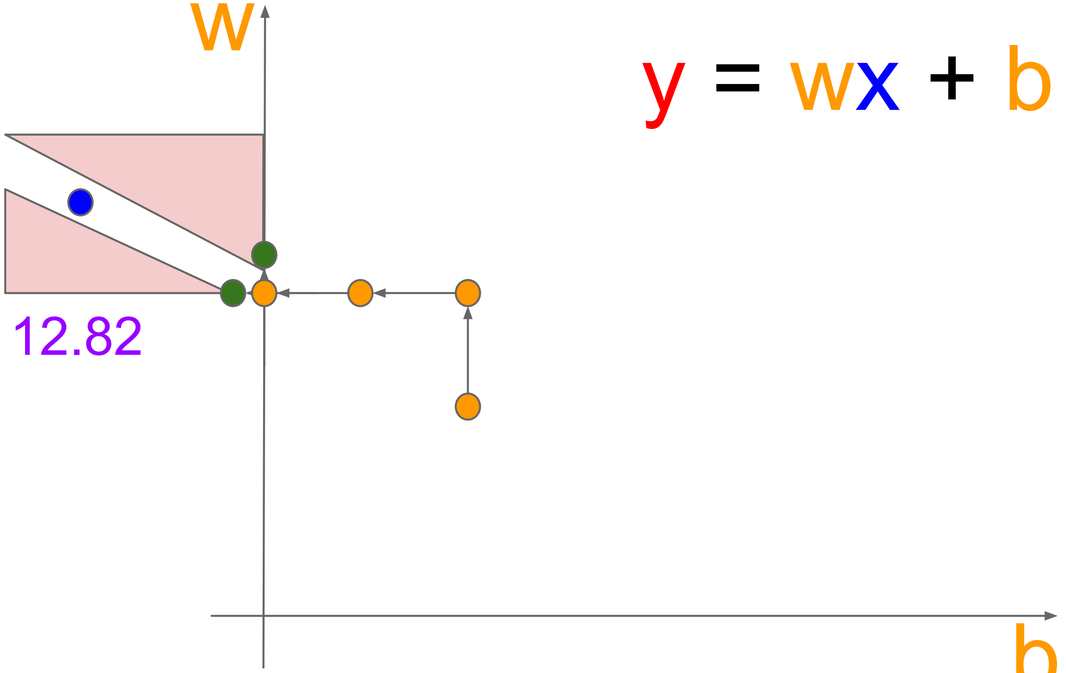
Optimizer

$$\arg\min_{w,b\in[-\infty,\infty]} C(w,b)$$

$w_3,b_3 = 3,0 : C(w_3,b_3) = 13$

$w_4,b_4 = 3.01,0 : C(w_4,b_4) = 12.82$

n	x	ŷ	y	$(y-\hat{y})^2$
0	1	0	3.01	9.06
1	5	16	15.01	0.98
2	6	20	18.01	3.96
			C(3.01,0)	12.82



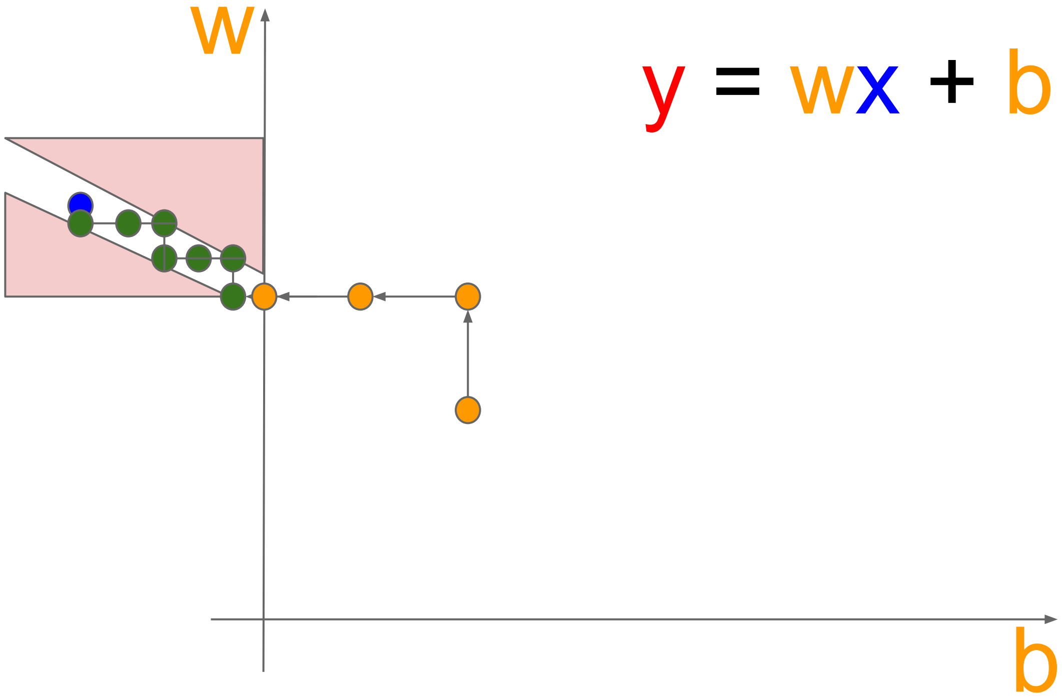
$y = wx + b$

Optimizers are our friends

Optimizer

$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_*, b_* = 4, -2 : C(w_*, b_*) = 12$



$y = wx + b$

Optimizers are our friends

Large Step Size

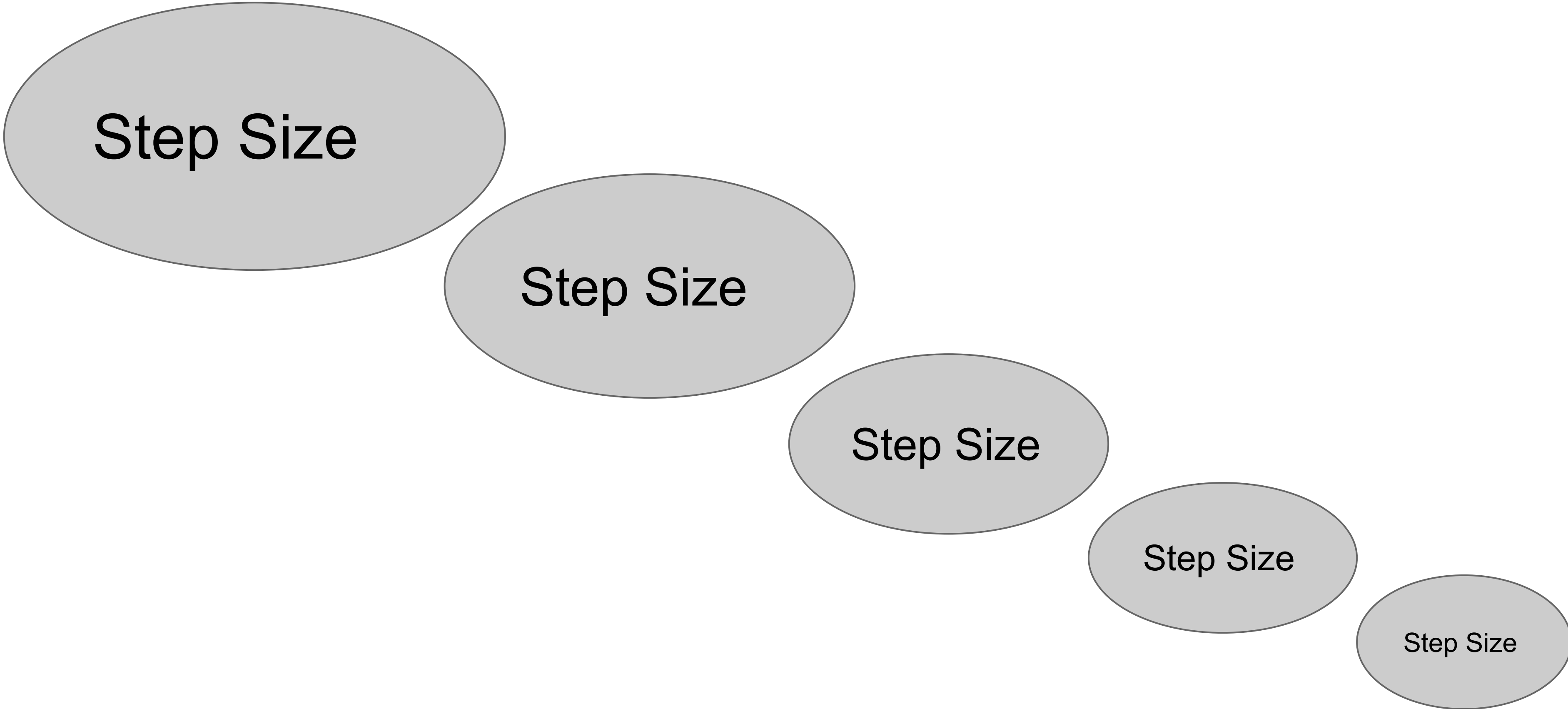
-Worse minimum
-But gets there faster

Vs

Small Step Size

-Better Minimum
-But gets there slowly

Optimizers are our friends



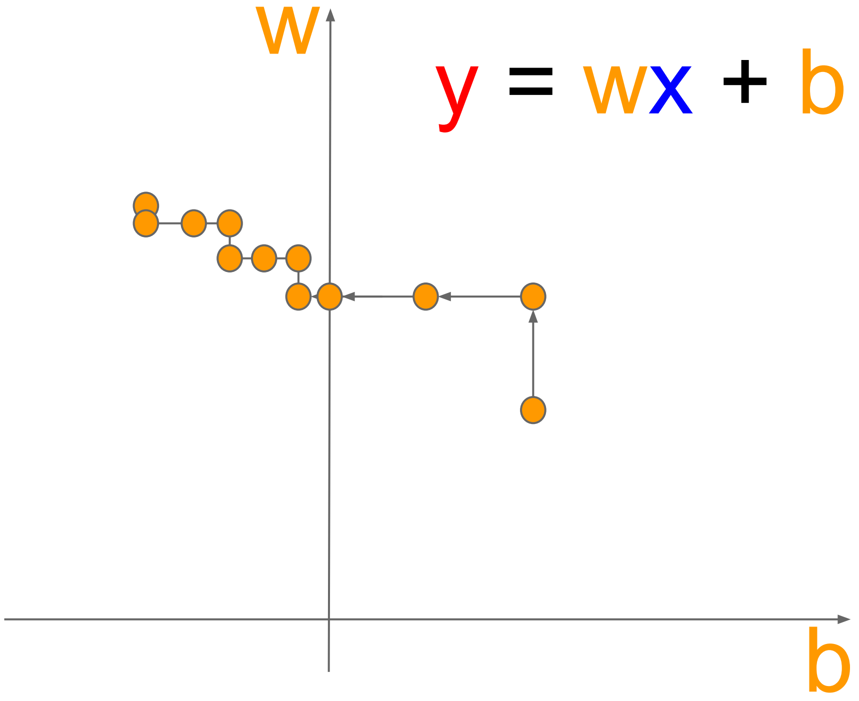
Optimizers are our friends

Optimizer

$$\arg\min_{w,b \in [-\infty, \infty]} C(w,b)$$

$w_*, b_* = 4,-2 : C(w_*, b_*) = 12$

$y = wx + b$



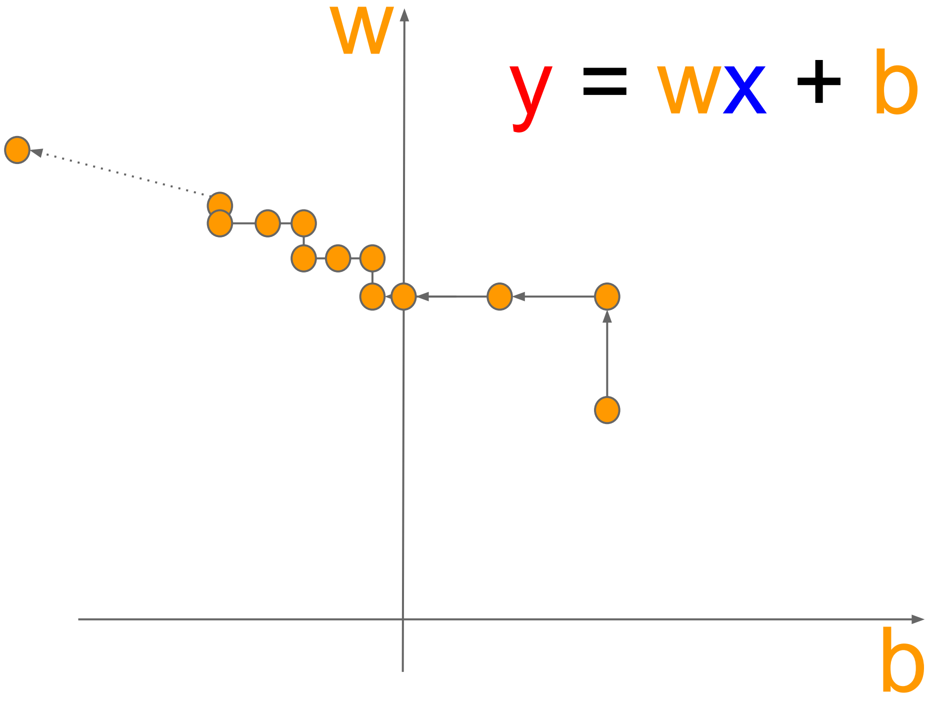
Optimizers are our friends

Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty, \infty]$

$w_*, b_* = 4,-4 : C(w_*, b_*) = 0$



Optimizers are our friends

$y = wx + b$

Data

x	$\hat{y}$
1	0
5	16
6	20

3 🍎

? 🍌

Optimizers are our friends

$$y = 4x - 4$$

Data

x	$\hat{y}$
1	0
5	16
6	20

3 🍎

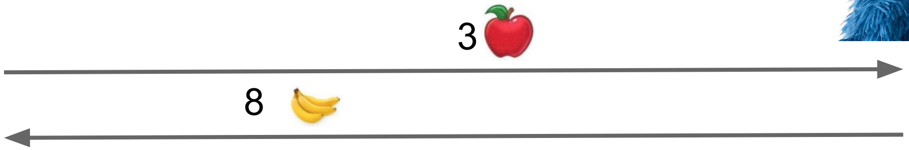
? 🍌

Optimizers are our friends

$y = 4x - 4$

Data

x	$\hat{y}$
1	0
5	16
6	20



Functions are our friends

$$y = wx + b$$

x : Image

y : Is this a cat

Functions are our friends

pixel (1,1)

pixel(1,3)

High
if cat

$$y = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$

x : Image

y : Is this a cat

Functions are our friends

pixel (1,1)

pixel(1,3)

High
if cat

$$y = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$

x : Image

y : Is this a cat

Millions of parameters

Millions of samples

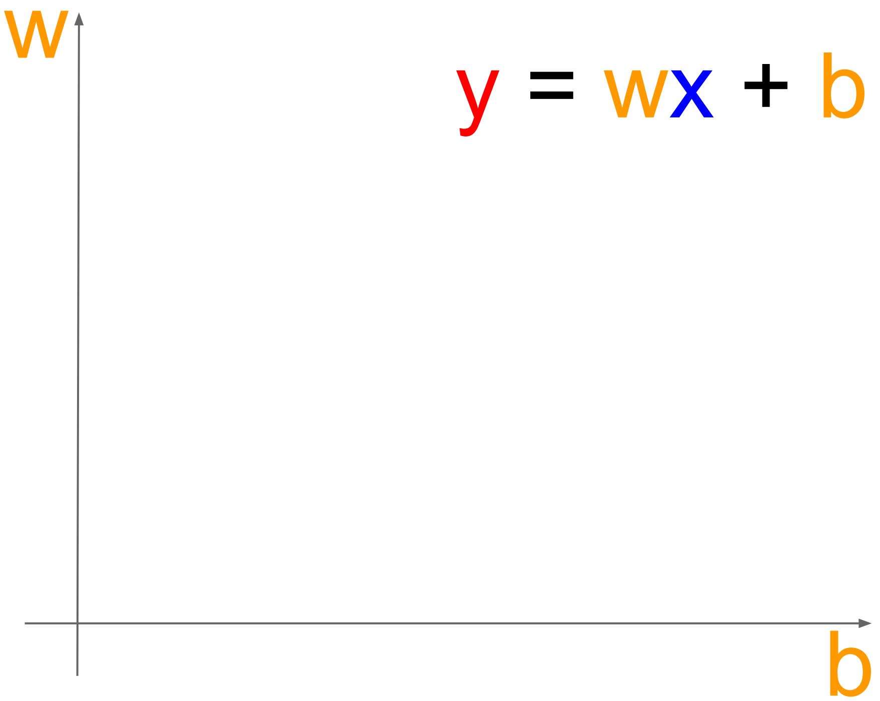
Gradients are our friends

Optimizer

$$\underset{w,b \in [-\infty, \infty]}{\arg\min}\ C(w,b)$$

Very expensive to compute (hours or days)

$$y = wx + b$$



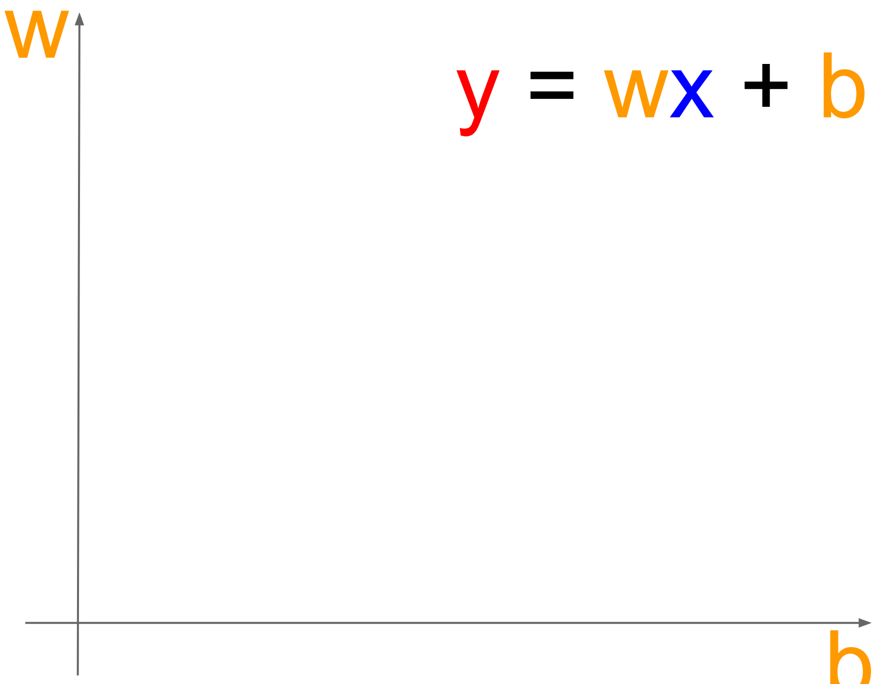
Gradients are our friends

Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

Should be used sparingly

$$y = wx + b$$



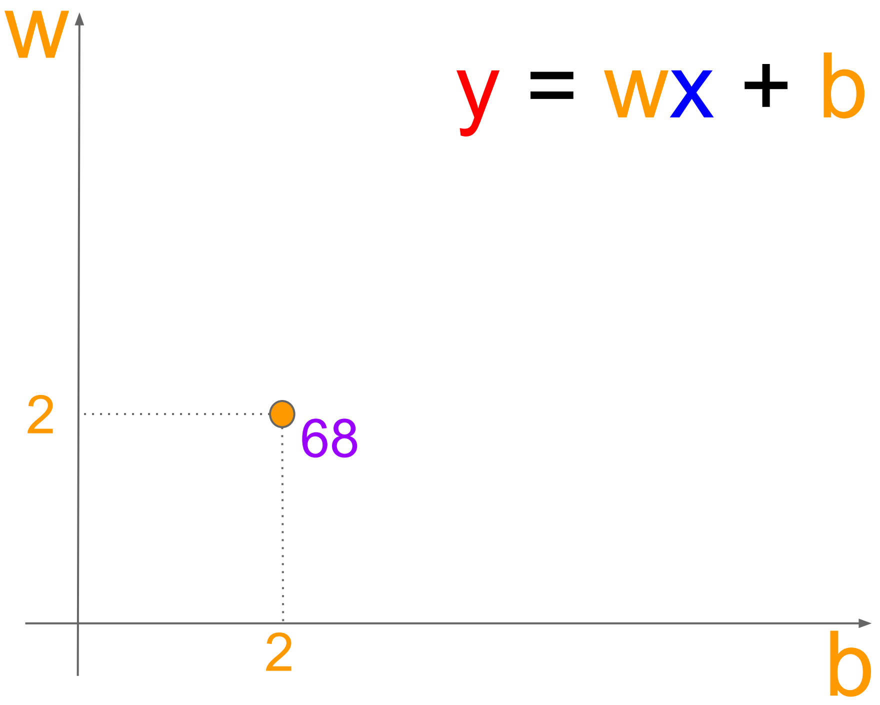
Gradients are our friends

Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty, \infty]$

$w_0, b_0 = 2,2 : C(w_0, b_0) = 68$



$y = wx + b$

Gradients are our friends

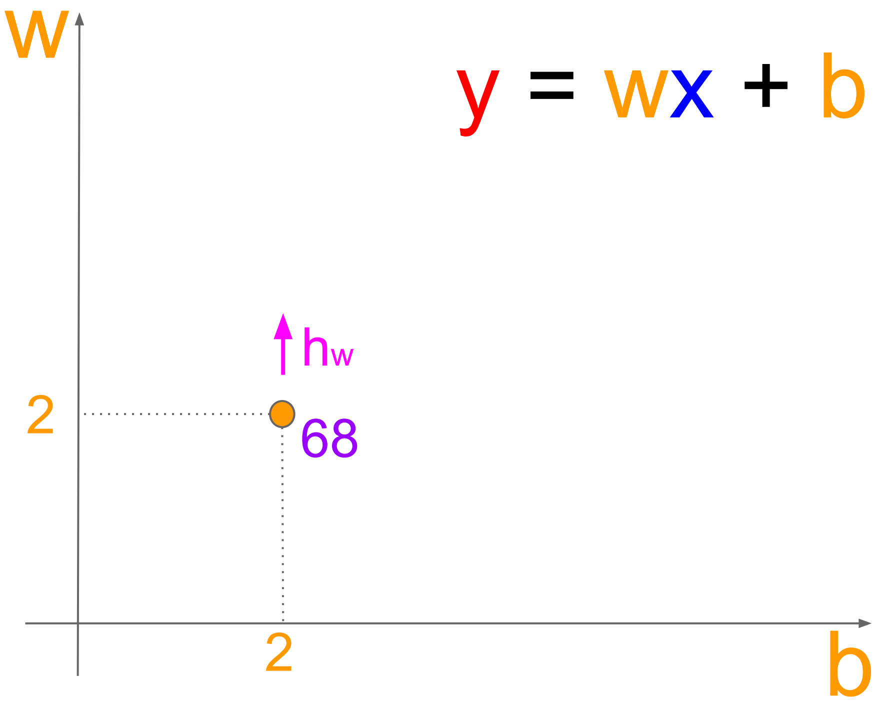
Optimizer

$$\underset{w,b \in [-\infty,\infty]}{\arg\min}\ C(w,b)$$

$w_0, b_0 = 2,2 : C(w_0,b_0) = 68$

$h_w = 1$

$y = wx + b$



Gradients are our friends

Optimizer

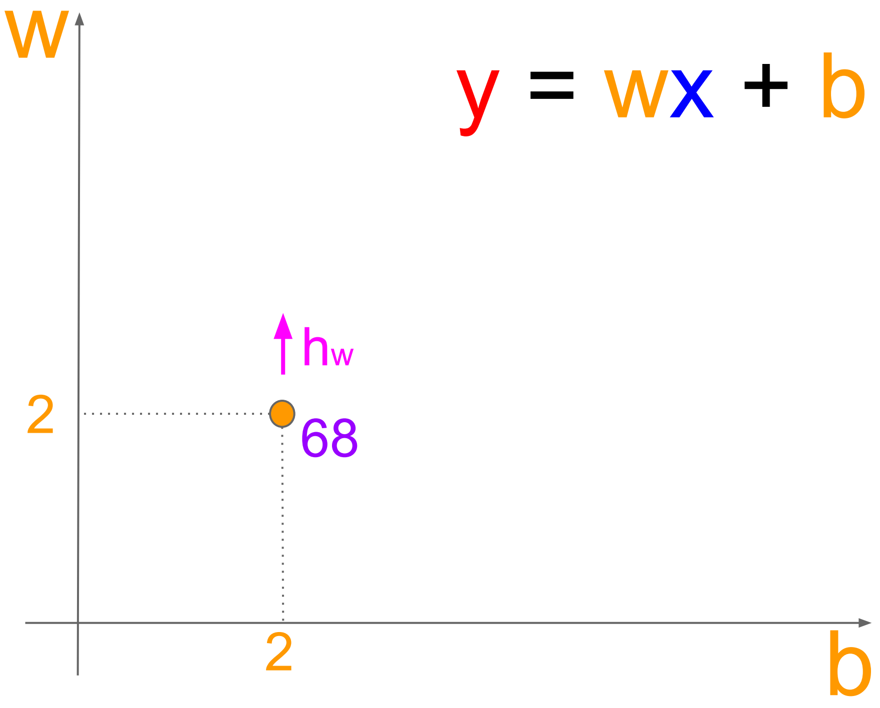
$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_0, b_0 = 2,2 : C(w_0, b_0) = 68$

$h_w = 1$

$C(w_0 + h_w, b_0) = C(3,2) = 26$

$y = wx + b$



Gradients are our friends

Optimizer

$$\arg\min C(w,b)$$

$$w,b \in [-\infty,\infty]$$

$$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$$

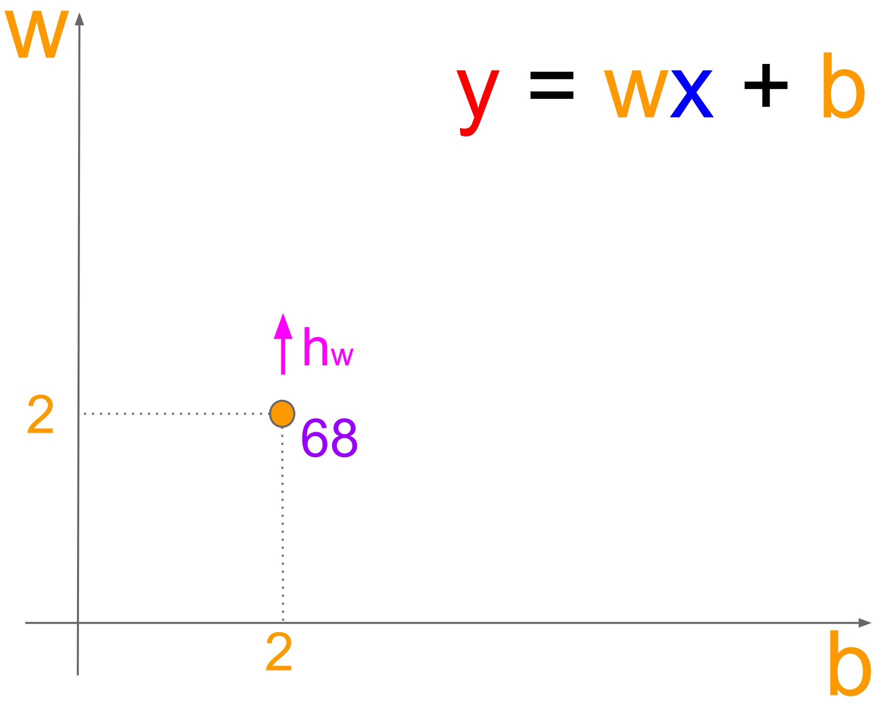
$$h_w = 1$$

$$C(w_0+h_w,b_0) = C(3,2) = 26$$

$$r = \frac{(C(w_0+1,b_0)-C(w_0,b_0))}{1}$$

$$r = \frac{(C(3,2)-C(2,2))=-42}{1}$$

$$y = wx + b$$



Gradients are our friends

Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty, \infty]$

$w_0, b_0 = 2,2 : C(w_0, b_0) = 68$

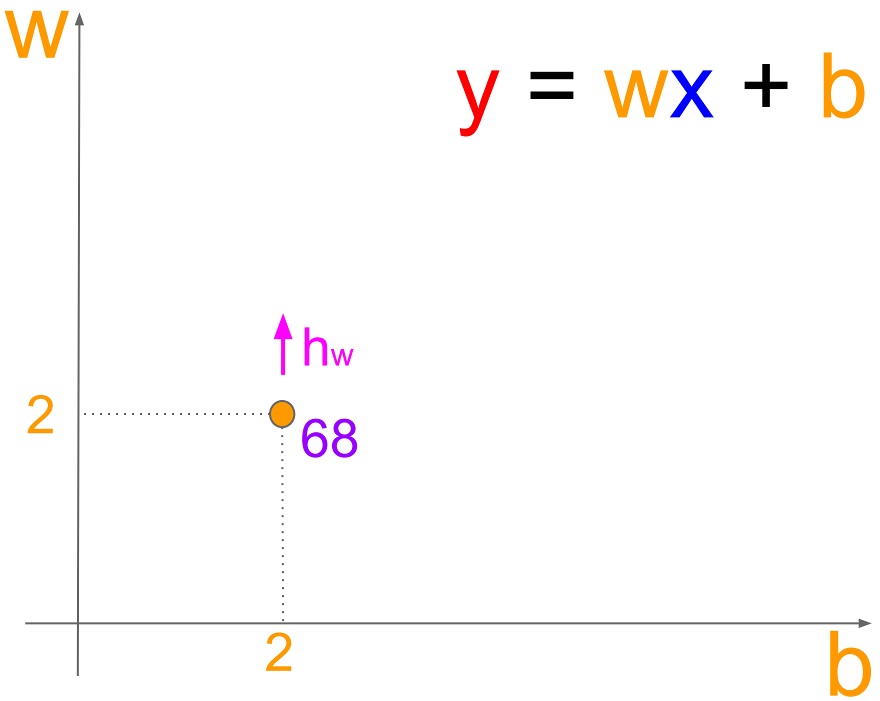
$h_w = 1$, $r = -42$

$h_w = 0.1$, $r = -98$

$h_w = 0.01$, $r = -104$

$h_w = 0.001$, $r = -104$

$y = wx + b$



Gradients are our friends

Optimizer

$$\arg\min C(w,b)$$
$$w,b \in [-\infty, \infty]$$

$w_0, b_0 = 2,2 : C(w_0, b_0) = 68$

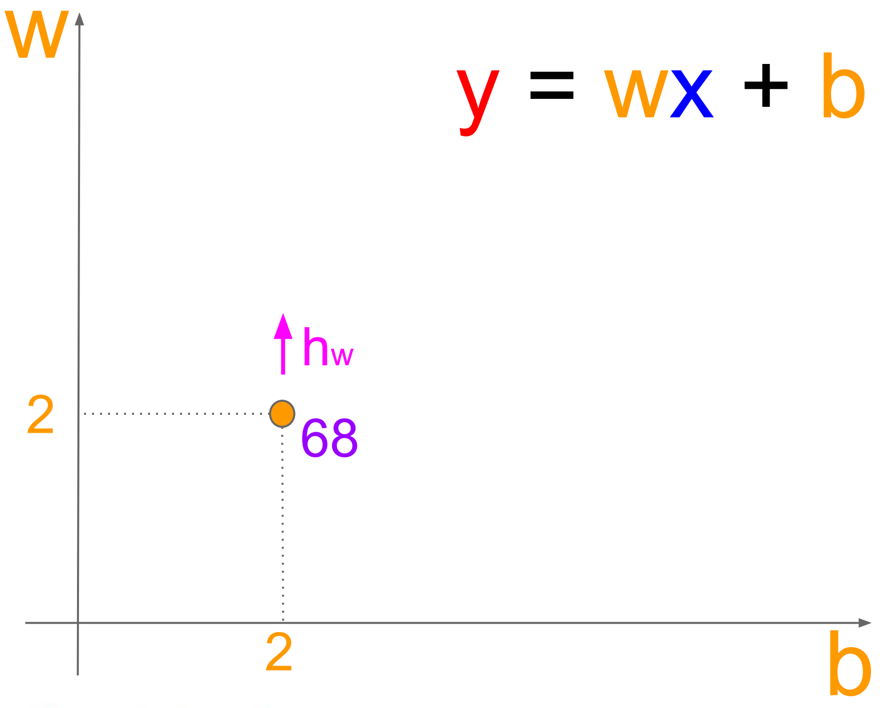
$h_w = 1$, $r = -42$

$h_w = 0.1$, $r = -98$

$h_w = 0.01$, $r = -104$

$h_w = 0.001$, $r = -104$

$h_w \to 0$, $r = \dfrac{\partial C}{\partial w}(w_0, b_0)$



$y = wx + b$

$$D_{\mathbf{u}} f(\mathbf{a}) = \lim_{h \to 0} \frac{f(\mathbf{a} + h\mathbf{u}) - \mathbf{f}(\mathbf{a})}{h}$$

Gradients are our friends

Optimizer

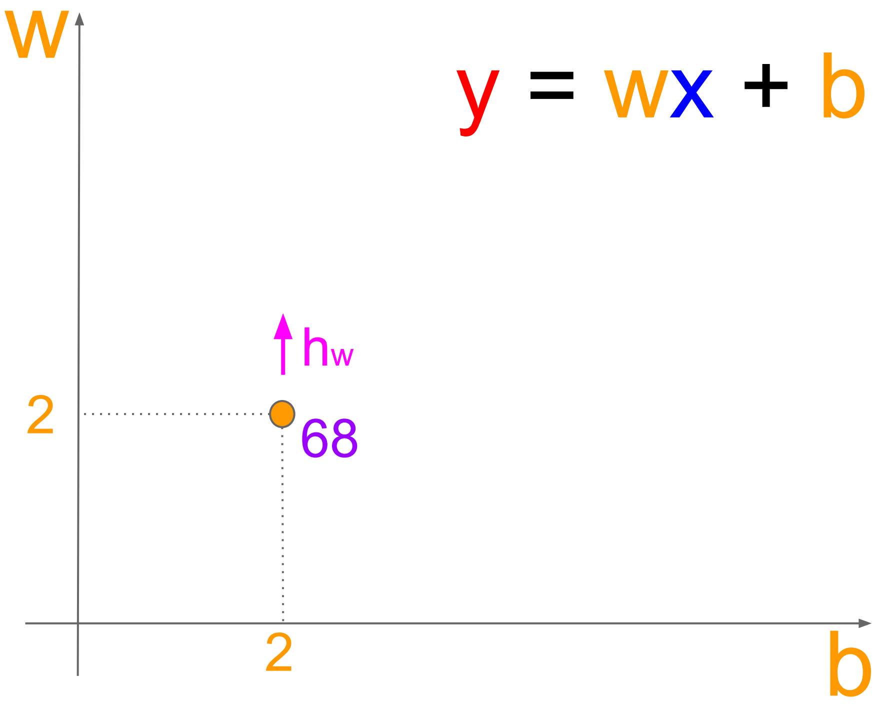
$$\arg\min C(w,b)$$

$$w,b \in [-\infty,\infty]$$

$$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w}$$

$$y = wx + b$$



Gradients are our friends

Optimizer

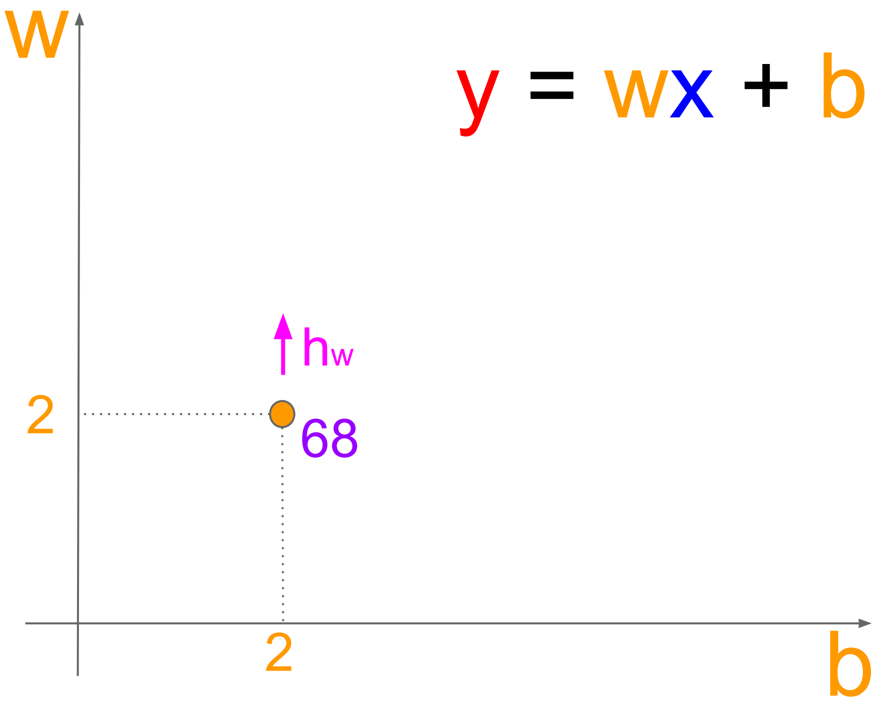
$$\arg\min C(w,b)$$

$$w,b \in [-\infty,\infty]$$

$$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n-\hat{y}_n)^2}{\partial w} = \sum_n 2(y_n-\hat{y}_n)x_n$$

$$y = wx + b$$



Gradients are our friends

Optimizer

$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_0, b_0 = 2,2 : C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w} = \sum_n 2(y_n - \hat{y}_n)x_n$$

$$h_w \rightarrow 0,\ r = \frac{\partial C}{\partial w}(w_0,b_0) = -104$$

n	x	ŷ	y	(y-ŷ)	2(y-ŷ)x
0	1	0	4	4	8
1	5	16	12	-4	-40
2	6	20	14	-6	-72

Gradients are our friends

Optimizer

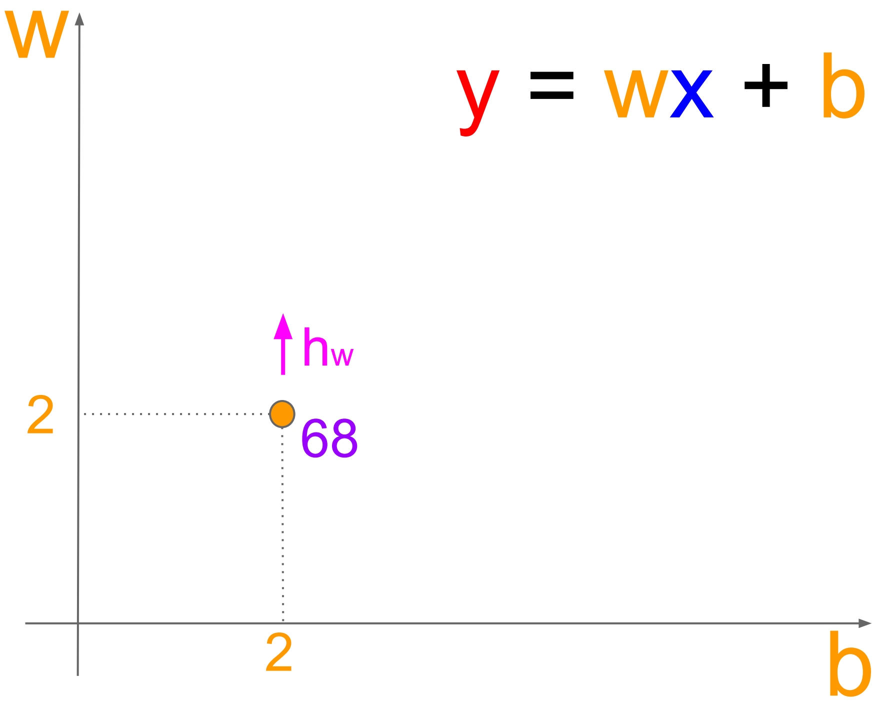
$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w_0, b_0 = 2,2 : C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w} = \sum_n 2(y_n - \hat{y}_n)x_n$$

$$\frac{\partial C}{\partial b} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial b} = \sum_n 2(y_n - \hat{y}_n)$$

$y = wx + b$



Gradients are our friends

Optimizer

$$\arg\min C(w,b)$$

$w,b \in [-\infty,\infty]$

$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$

$$h_w \rightarrow 0,\ r_w = \frac{\partial C}{\partial w}(w_0,b_0) = -104$$

$$h_b \rightarrow 0,\ r_b = \frac{\partial C}{\partial w}(w_0,b_0) = -12$$

n	x	ŷ	y	(y-ŷ)	2(y-ŷ)
0	1	0	4	4	8
1	5	16	12	-4	-8
2	6	20	14	-6	-12

Gradients are our friends

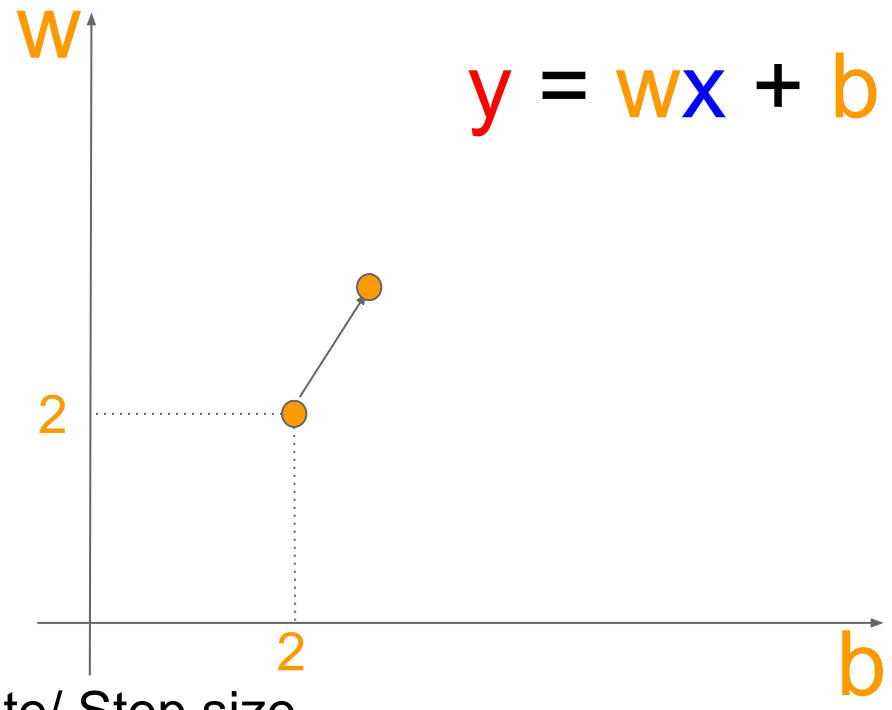
Optimizer

$$\arg\min C(w,b)$$

$$w,b \in [-\infty,\infty]$$

$$w_0,b_0 = 2,2 : C(w_0,b_0) = 68$$

$y = wx + b$



$$h_w \rightarrow 0,\ r_w = \frac{\partial C}{\partial w}(w_0,b_0) = -104$$

$$h_b \rightarrow 0,\ r_b = \frac{\partial C}{\partial w}(w_0,b_0) = -12$$

$$w_1 = w_0 - r_w \alpha$$

$$b_1 = b_0 - r_b \alpha$$

$\alpha \rightarrow$ Learning Rate/ Step size

Summary

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

Cost

$$C(w,b) = \sum_{n\in\{0,1,2\}}(y_n-\hat{y}_n)^2$$

Optimizer

$$\underset{w,b\in[-\infty,\infty]}{\arg\min}\; C(w,b)$$

Summary

Data

n	x	$\hat{y}$
0	1	0
1	5	16
2	6	20

Model

$$y_n = wx_n + b$$

System

$$y = 4x - 4$$

Cost

$$C(w,b) = \sum_{n\in\{0,1,2\}} (y_n - \hat{y}_n)^2$$

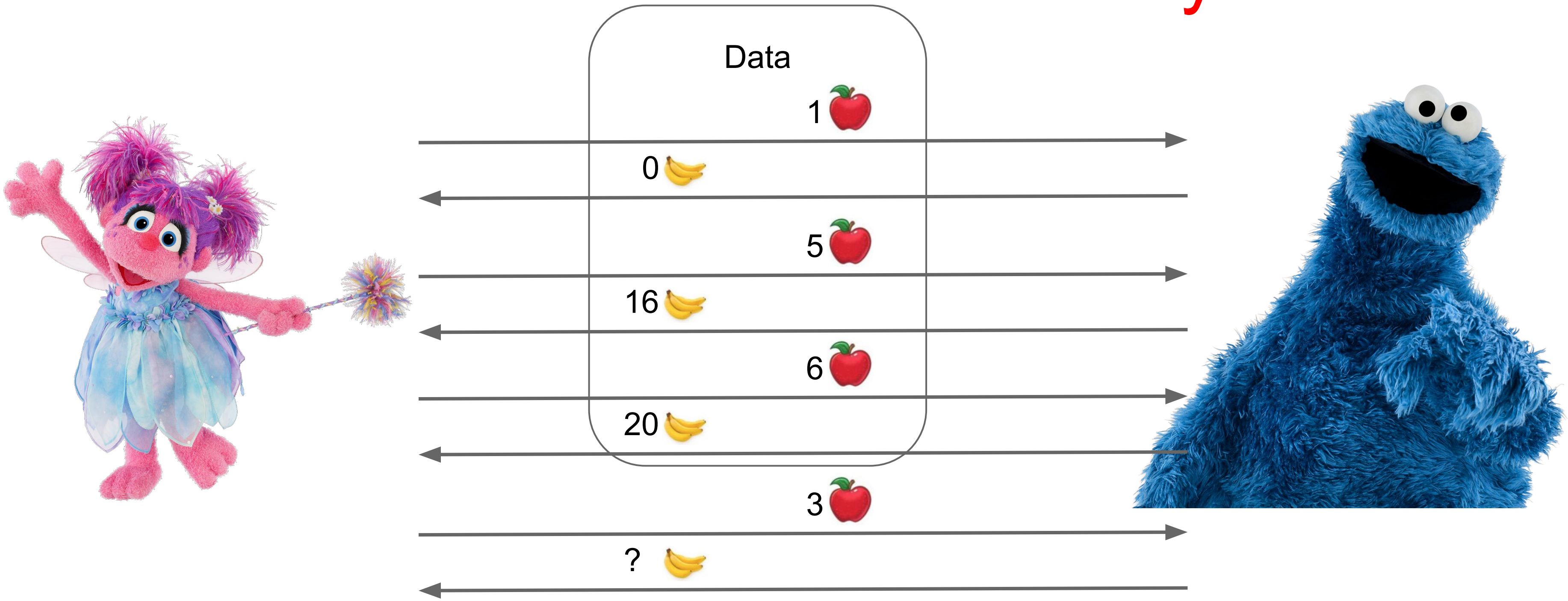
Optimizer

$$\underset{w,b\in[-\infty,\infty]}{\arg\min}\ C(w,b)$$

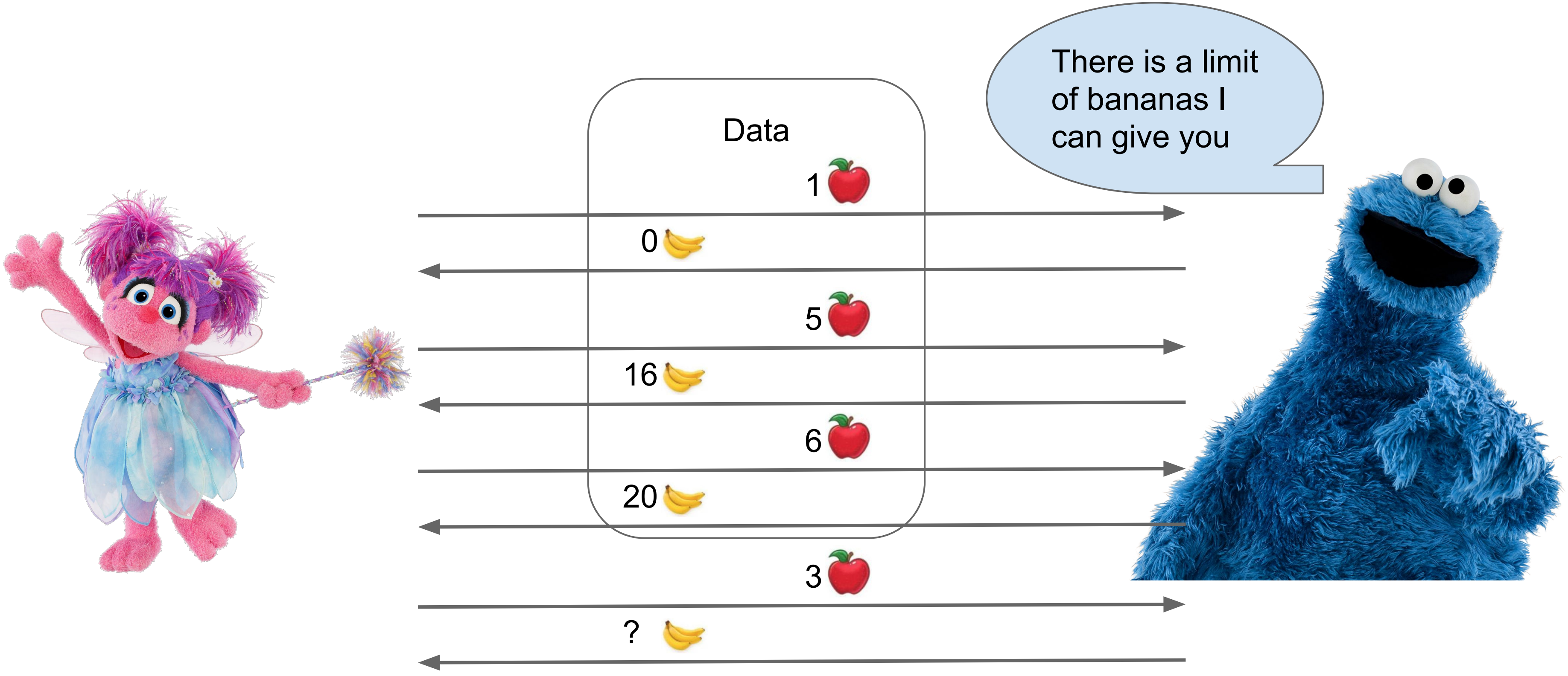
Into Deep Learning

Nonlinear Neural Models

$y = 4x - 4$



Nonlinear Neural Models

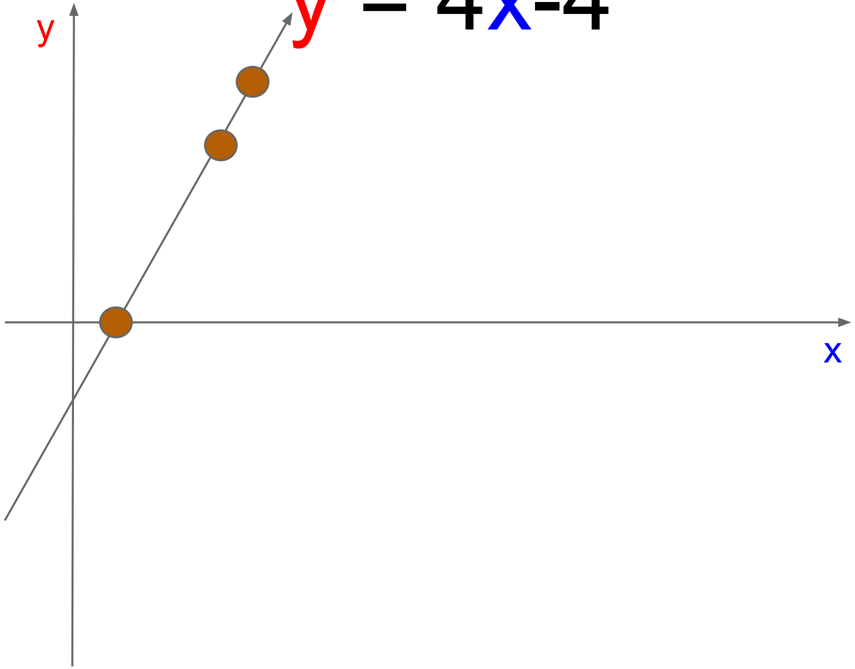


Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20

$y = 4x - 4$

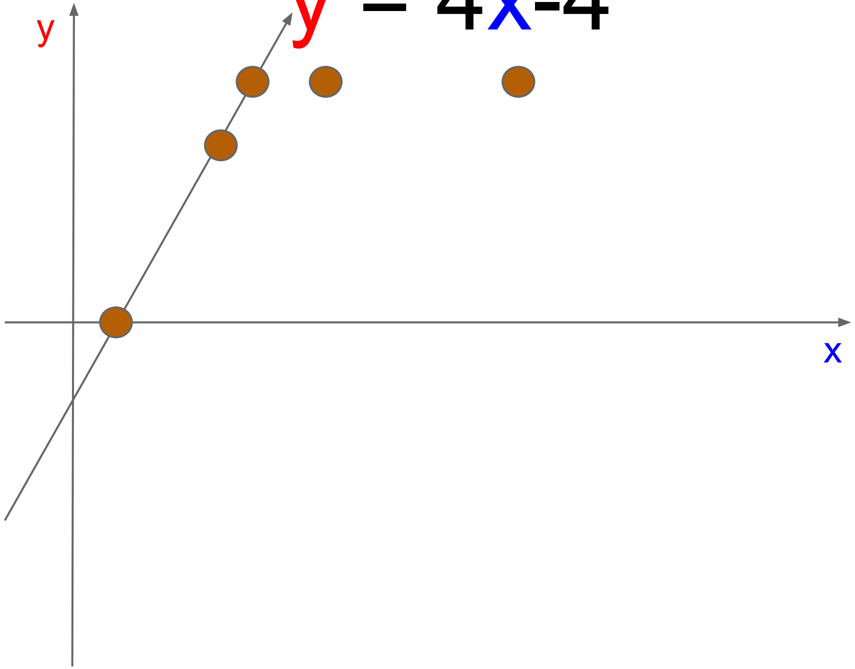


Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

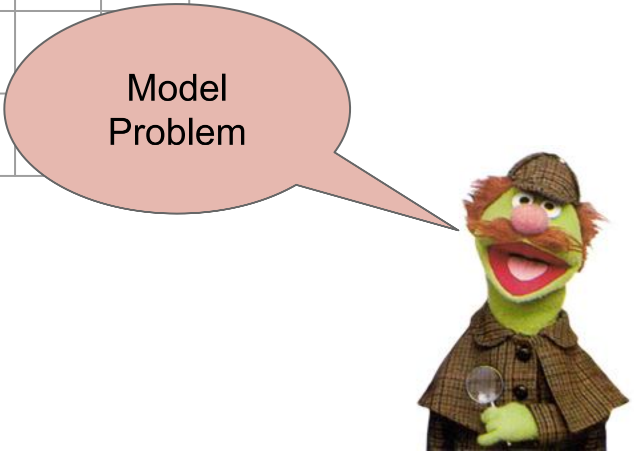
$y = 4x-4$



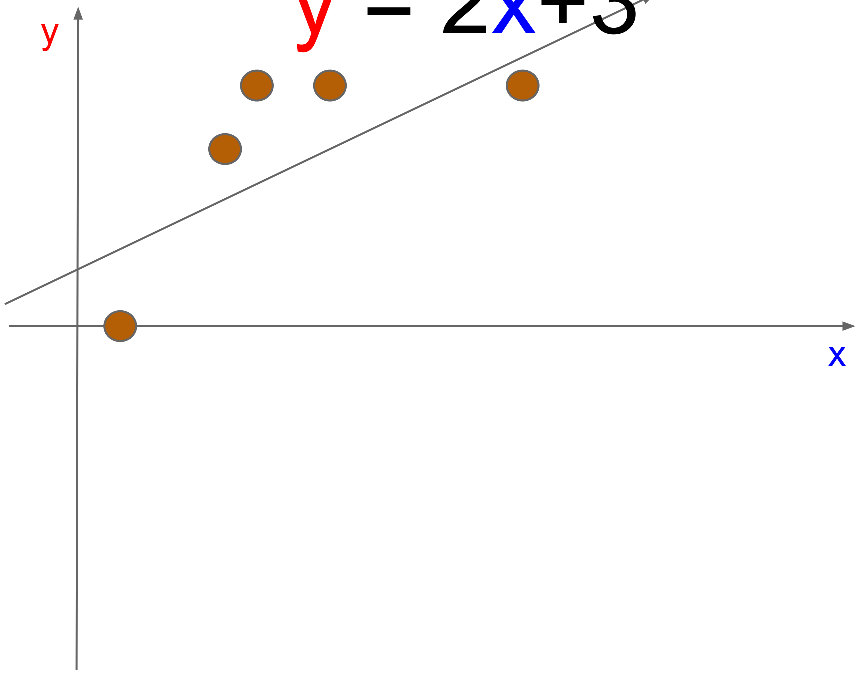
Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3		
4		



$y = 2x+3$



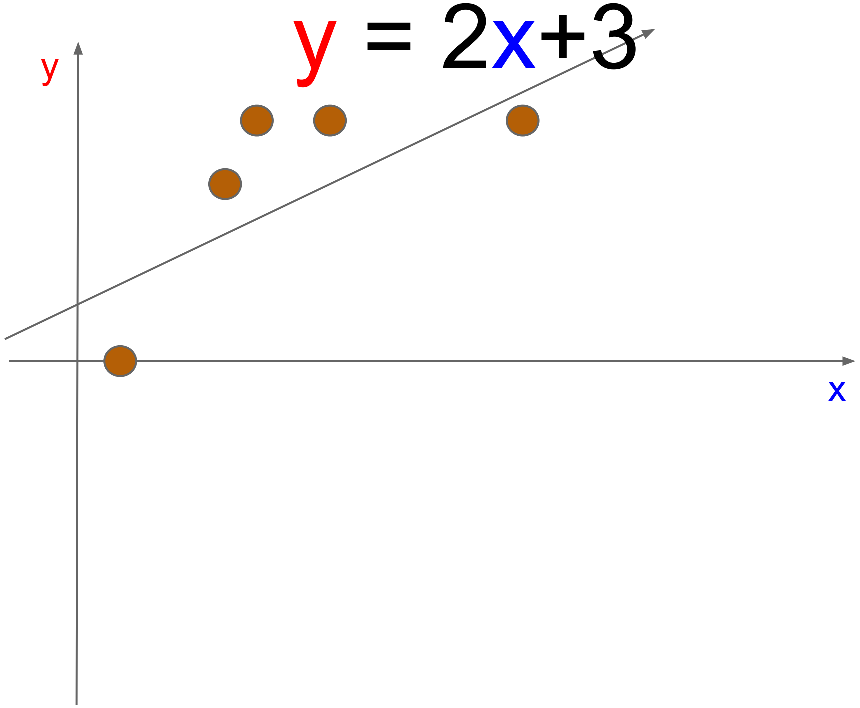
Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3		
4		

Model Problem

Underfitting

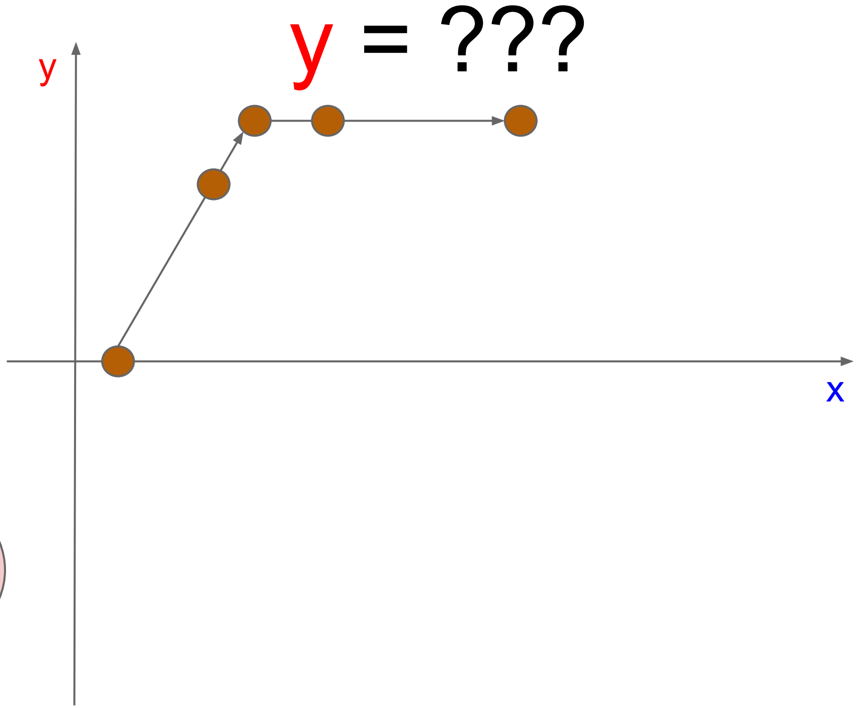


Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

Can we learn
arbitrary functions?



Nonlinear Neural Models

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

Use different linear functions depending on the value of x?

Nonlinear Neural Models

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

S_1 - 1 if x < 6 and 0 otherwise

S_2 - 1 if x >= 6 and 0 otherwise

Nonlinear Neural Models

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

S_1 - 1 if x < 6 and 0 otherwise

S_2 - 1 if x >= 6 and 0 otherwise

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (4x - 4)s_1 + (0x + 20)s_2$$

Nonlinear Neural Models

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

S_1 - 1 if x < 6 and 0 otherwise

S_2 - 1 if x >= 6 and 0 otherwise

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

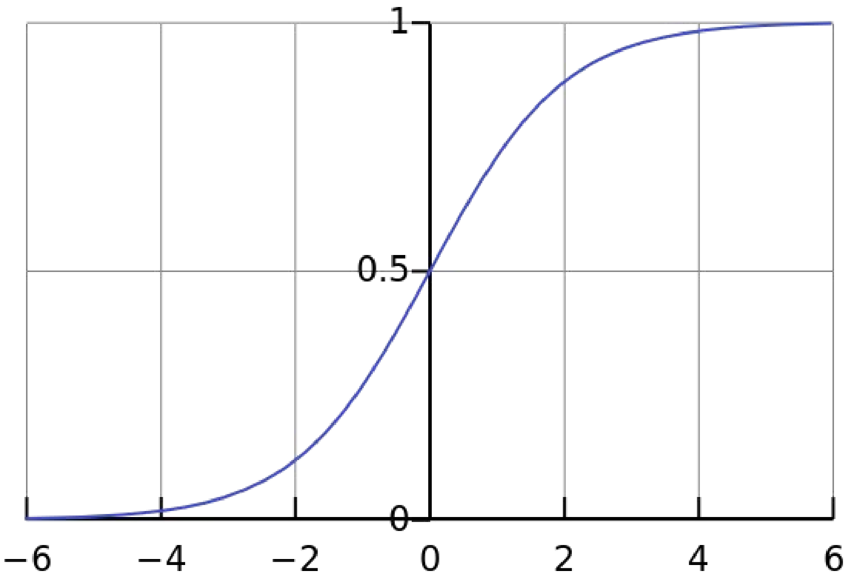
$$y = (4x - 4)s_1 + (0x + 20)s_2$$

? ?

Nonlinear Neural Models

$$s = \boldsymbol{\sigma}(wx + b)$$

$$\boldsymbol{\sigma}(t) = \frac{1}{1 + e^{-t}}$$



Nonlinear Neural Models

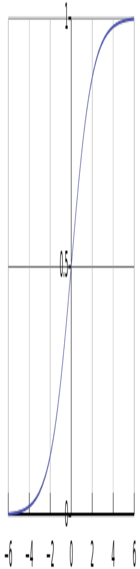
$s = \sigma(1000x)$

Nonlinear Neural Models

$s = \sigma(1000x)$

$x = 0.1$ then $\sigma(1000x) = 1$

$x = -0.1$ then $\sigma(1000x) = 0$



Nonlinear Neural Models

$$s = \boldsymbol{\sigma}(1000x - 6000)$$

$$x = 6.1 \text{ then } \boldsymbol{\sigma}(1000x - 6000) = 1$$

$$x = 5.9 \text{ then } \boldsymbol{\sigma}(1000x - 6000) = 0$$

Nonlinear Neural Models

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

$$s1 = \sigma(w_3 x + b_3)$$
$$s2 = \sigma(w_4 x + b_4)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (4x - 4)s_1 + (0x+20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$

$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (4x - 4)s_1 + (0x+20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$

$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (16)s_1 + (0x+20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$

$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (16)s_1 + (20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$

$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (16)s_1 + (20)s_2$$

$$s1 = \sigma(1000)$$
$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

	Data	
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (16)s_1 + (20)s_2$$

$$s1 = \boldsymbol{\sigma}(1000)$$

$$s2 = \boldsymbol{\sigma}(-1000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (16)1 + (20)0$$

$$s1 = \sigma(1000)$$

$$s2 = \sigma(-1000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = 16$$

$$s1 = \boldsymbol{\sigma}(1000)$$
$$s2 = \boldsymbol{\sigma}(-1000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (4x - 4)s_1 + (0x+20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$

$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

	Data	
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (32)s_1 + (0x+20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$
$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (32)s_1 + (20)s_2$$

$$s1 = \sigma(-1000x + 6000)$$

$$s2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

	Data	
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (32)s_1 + (20)s_2$$

$$s1 = \boldsymbol{\sigma}(-3000)$$

$$s2 = \boldsymbol{\sigma}(1000x - 6000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (32)s_1 + (20)s_2$$

$$s1 = \boldsymbol{\sigma}(-3000)$$

$$s2 = \boldsymbol{\sigma}(3000)$$

Nonlinear Neural Models

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (32)0 + (20)1$$

$$s1 = \boldsymbol{\sigma}(-3000)$$
$$s2 = \boldsymbol{\sigma}(3000)$$

Nonlinear Neural Models

Data

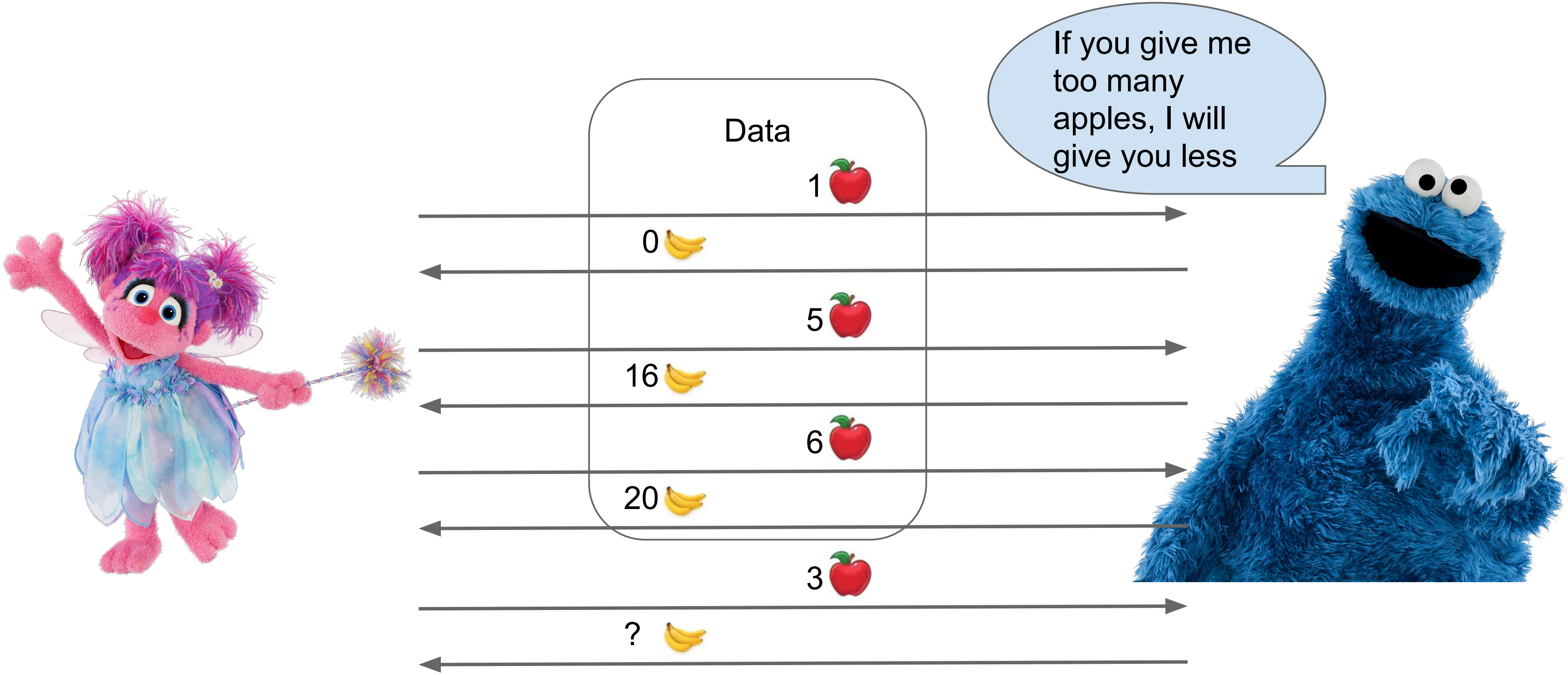
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$y = 20$

$s1 = \boldsymbol{\sigma}(-3000)$

$s2 = \boldsymbol{\sigma}(3000)$

Nonlinear Neural Models

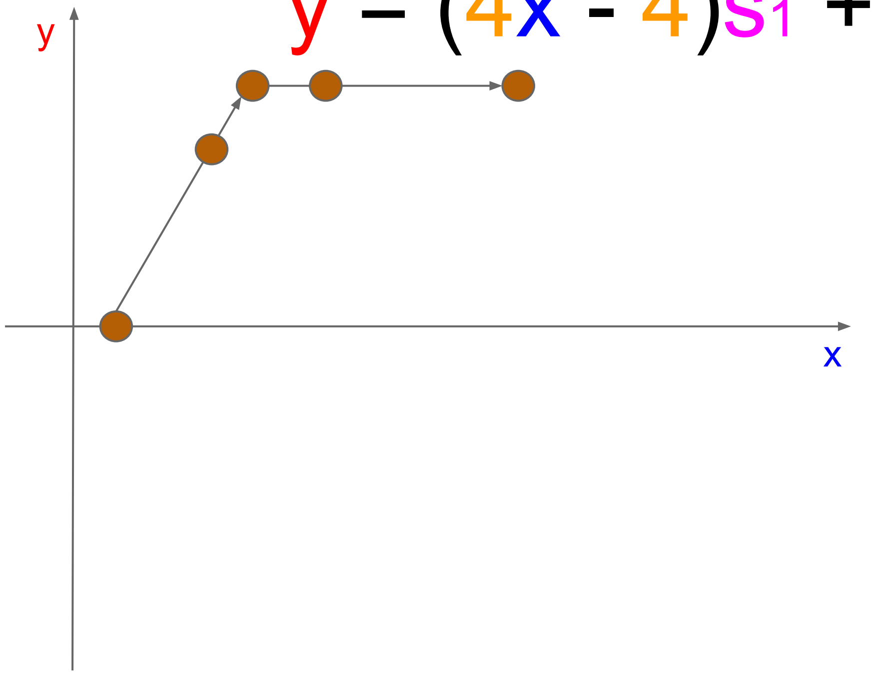


Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = (4x - 4)s_1 + (0x+20)s_2$$

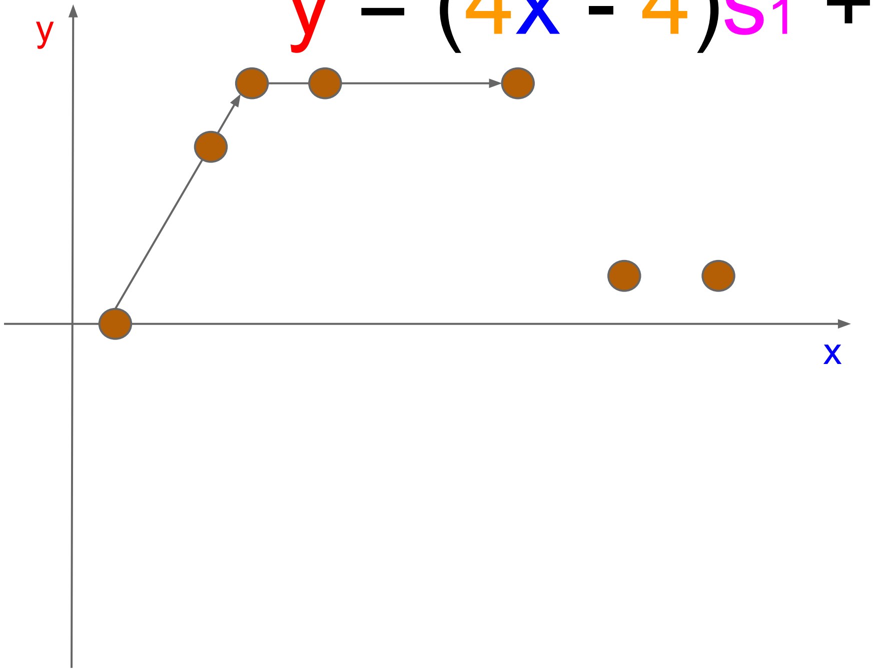


Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x+20)s_2$$



Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x+20)s_2 + (0x+1)s_3$$

$$s_1 = \sigma(-1000x + 6000)$$

$$s_2 = ????$$

$$s_3 = \sigma(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x + 20)s_2 + (0x + 1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-1000x + 6000)$$

s_2 = not s_1 and not s_3

$$s_3 = \boldsymbol{\sigma}(1000x - 15000)$$

Multilayer Perceptrons

$$y = (w_1 x + b_1) s_1 + (w_2 x + b_2) s_2 + (w_3 x + b_3) s_3$$

$$s_1 = \sigma(w_4 x + b_4)$$

$$s_2 = \sigma(w_5 s_1 + w_6 s_3 + b_5)$$

$$s_3 = \sigma(w_7 x + b_6)$$

Multilayer Perceptrons

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2 + (w_3 x + b_3)s_3$$

$$s_1 = \sigma(w_4 x + b_4)$$

Layer 1 Perceptron

$$s_2 = \sigma(w_5 s_1 + w_6 s_3 + b_5)$$

$$s_3 = \sigma(w_7 x + b_6)$$

Layer 1 Perceptron

Multilayer Perceptrons

$$y = (w_1x + b_1)s_1 + (w_2x+b_2)s_2 + (w_3x+b_3)s_3$$

$$s_1 = \sigma(w_4x + b_4)$$ Layer 1 Perceptron

$$s_2 = \sigma(w_5s_1 + w_6s_3 + b_5)$$ Layer 2 Perceptron

$$s_3 = \sigma(w_7x + b_6)$$ Layer 1 Perceptron

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x+20)s_2 + (0x+1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-1000x + 6000)$$

$$s_2 = \text{not } s_1 \text{ and not } s_3$$

$$s_3 = \boldsymbol{\sigma}(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x+20)s_2 + (0x+1)s_3$$

$$s_1 = \sigma(-1000x + 6000)$$

$$s_2 = \sigma(-1000s_1 - 1000s_3 + 500)$$

$$s_3 = \sigma(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x+20)s_2 + (0x+1)s_3$$

$$s_1 = \sigma(-1000x + 6000)$$

$$s_2 = \sigma(-1000s_1 - 1000s_3 + 500)$$

$$s_3 = \sigma(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (40)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \sigma(-1000x + 6000)$$

$$s_2 = \sigma(-1000s_1 - 1000s_3 + 500)$$

$$s_3 = \sigma(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (40)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-5000) = 0$$

$$s_2 = \boldsymbol{\sigma}(-1000s_1 - 1000s_3 + 500)$$

$$s_3 = \boldsymbol{\sigma}(-4000) = 0$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (40)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \sigma(-5000) = 0$$

$$s_2 = \sigma(-\ 0 - 0 + 500)$$

$$s_3 = \sigma(-4000) = 0$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (40)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-5000) = 0$$

$$s_2 = \boldsymbol{\sigma}(500)$$

$$s_3 = \boldsymbol{\sigma}(-4000) = 0$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (40)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-5000) = 0$$

$$s_2 = \boldsymbol{\sigma}(500) = 1$$

$$s_3 = \boldsymbol{\sigma}(-4000) = 0$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (40)0 + (20)1 + (1)0$$

$$s_1 = \boldsymbol{\sigma}(-5000) = 0$$

$$s_2 = \boldsymbol{\sigma}(500) = 1$$

$$s_3 = \boldsymbol{\sigma}(-4000) = 0$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = 20$$

$$s_1 = \boldsymbol{\sigma}(-5000) = 0$$

$$s_2 = \boldsymbol{\sigma}(500) = 1$$

$$s_3 = \boldsymbol{\sigma}(-4000) = 0$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (4x - 4)s_1 + (0x+20)s_2 + (0x+1)s_3$$

$$s_1 = \sigma(-1000x + 6000)$$

$$s_2 = \sigma(-1000s_1 - 1000s_3 + 500)$$

$$s_3 = \sigma(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (772)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \sigma(-1000x + 6000)$$

$$s_2 = \sigma(-1000s_4 - 1000s_5 + 500)$$

$$s_3 = \sigma(1000x - 15000)$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (772)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-13000) = 0$$

$$s_2 = \boldsymbol{\sigma}(-1000s_4 - 1000s_5 + 500)$$

$$s_3 = \boldsymbol{\sigma}(4000) = 1$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (772)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-13000) = 0$$

$$s_2 = \boldsymbol{\sigma}(-1000 + 0 + 500)$$

$$s_3 = \boldsymbol{\sigma}(4000) = 1$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (772)s_1 + (20)s_2 + (1)s_3$$

$$s_1 = \boldsymbol{\sigma}(-13000) = 0$$

$$s_2 = \boldsymbol{\sigma}(-500) = 0$$

$$s_3 = \boldsymbol{\sigma}(4000) = 1$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$$y = (772)0 + (20)0 + (1)1$$

$$s_1 = \boldsymbol{\sigma}(-13000) = 0$$

$$s_2 = \boldsymbol{\sigma}(-500) = 0$$

$$s_3 = \boldsymbol{\sigma}(4000) = 1$$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

$y = 1$

$s_1 = \boldsymbol{\sigma}(-13000) = 0$

$s_2 = \boldsymbol{\sigma}(-500) = 0$

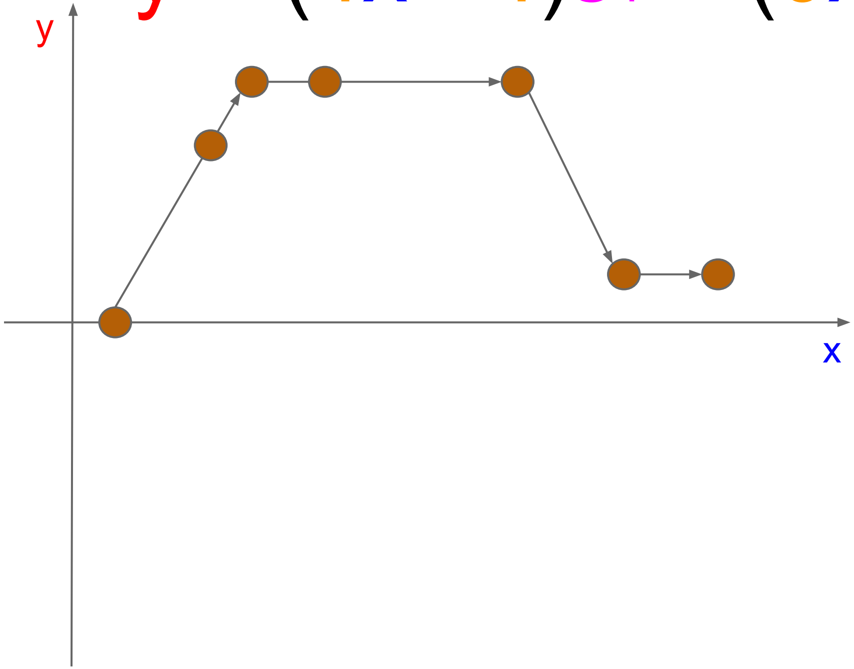
$s_3 = \boldsymbol{\sigma}(4000) = 1$

Multilayer Perceptrons

Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

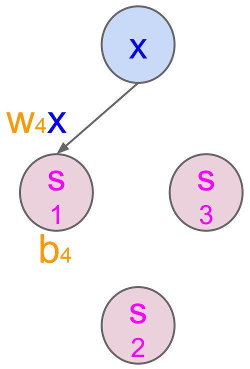
$$y = (4x - 4)s_1 + (0x+20)s_2 + (0x+1)s_3$$



Multilayer Perceptrons

$$y = (w_1x + b_1)s_1 + (w_2x+b_2)s_2 + (w_3x+b_3)s_3$$

$$s_1 = \sigma(w_4x + b_4)$$
$$s_2 = \sigma(w_5s_1 + w_6s_3 + b_5)$$
$$s_3 = \sigma(w_7x + b_6)$$



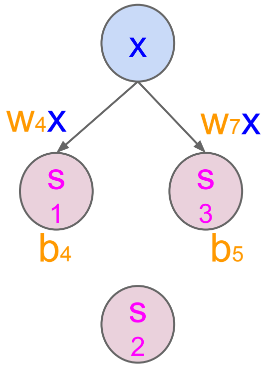
Multilayer Perceptrons

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2 + (w_3 x + b_3)s_3$$

$$s_1 = \sigma(w_4 x + b_4)$$

$$s_2 = \sigma(w_5 s_1 + w_6 s_3 + b_5)$$

$$s_3 = \sigma(w_7 x + b_6)$$



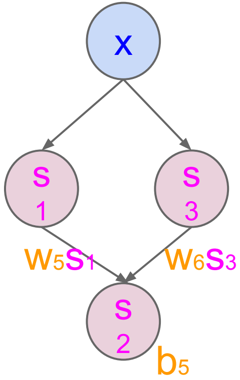
Multilayer Perceptrons

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2 + (w_3 x + b_3)s_3$$

$$s_1 = \sigma(w_4 x + b_4)$$

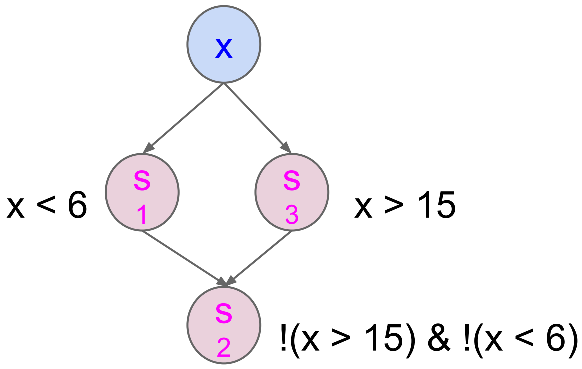
$$s_2 = \sigma(w_5 s_1 + w_6 s_3 + b_5)$$

$$s_3 = \sigma(w_7 x + b_6)$$



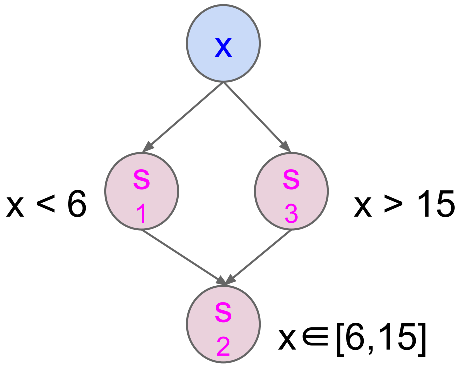
Multilayer Perceptrons

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2 + (w_3 x + b_3)s_3$$

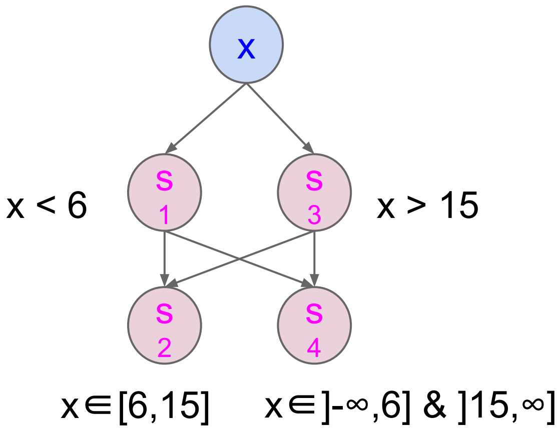


Multilayer Perceptrons

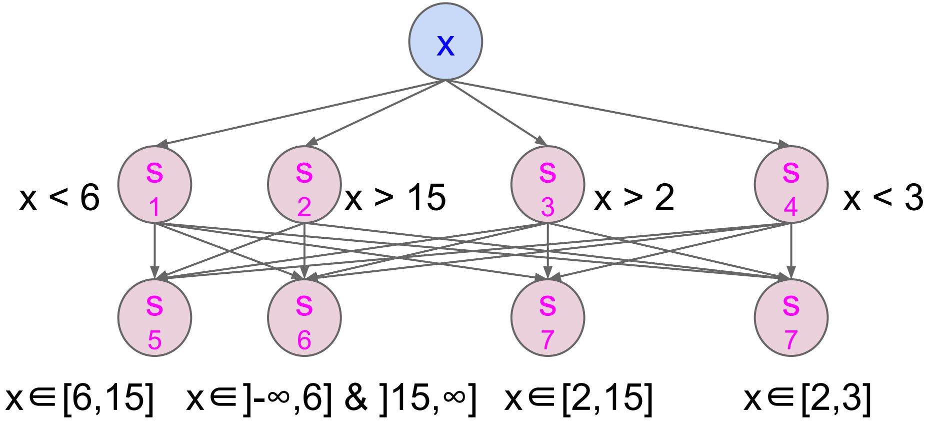
$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2 + (w_3 x + b_3)s_3$$



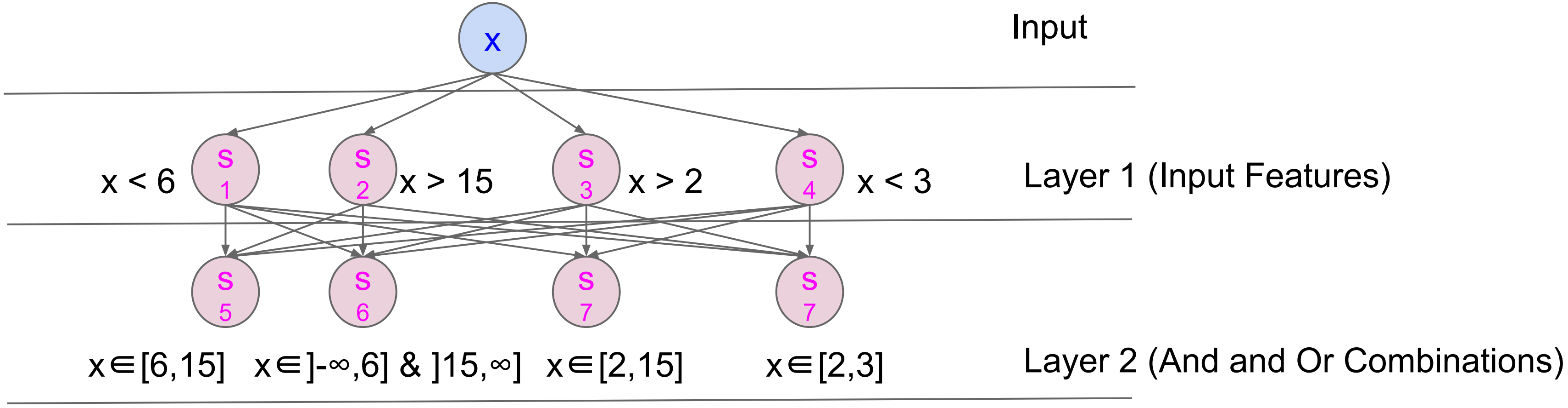
Multilayer Perceptrons



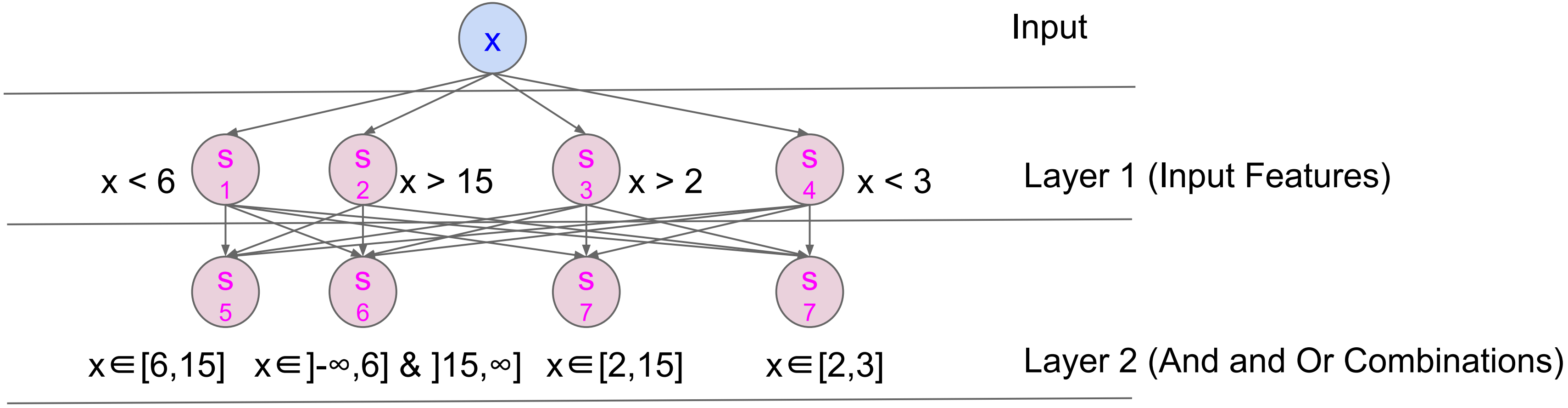
Multilayer Perceptrons



Multilayer Perceptrons



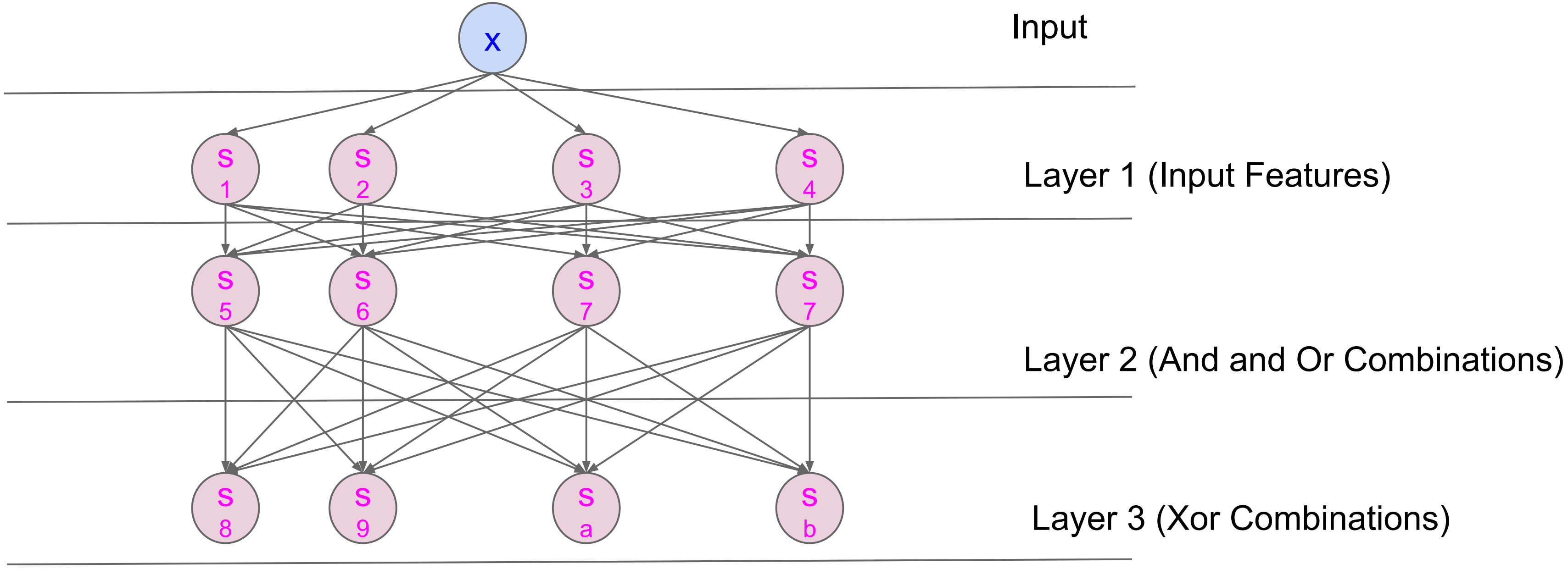
Multilayer Perceptrons



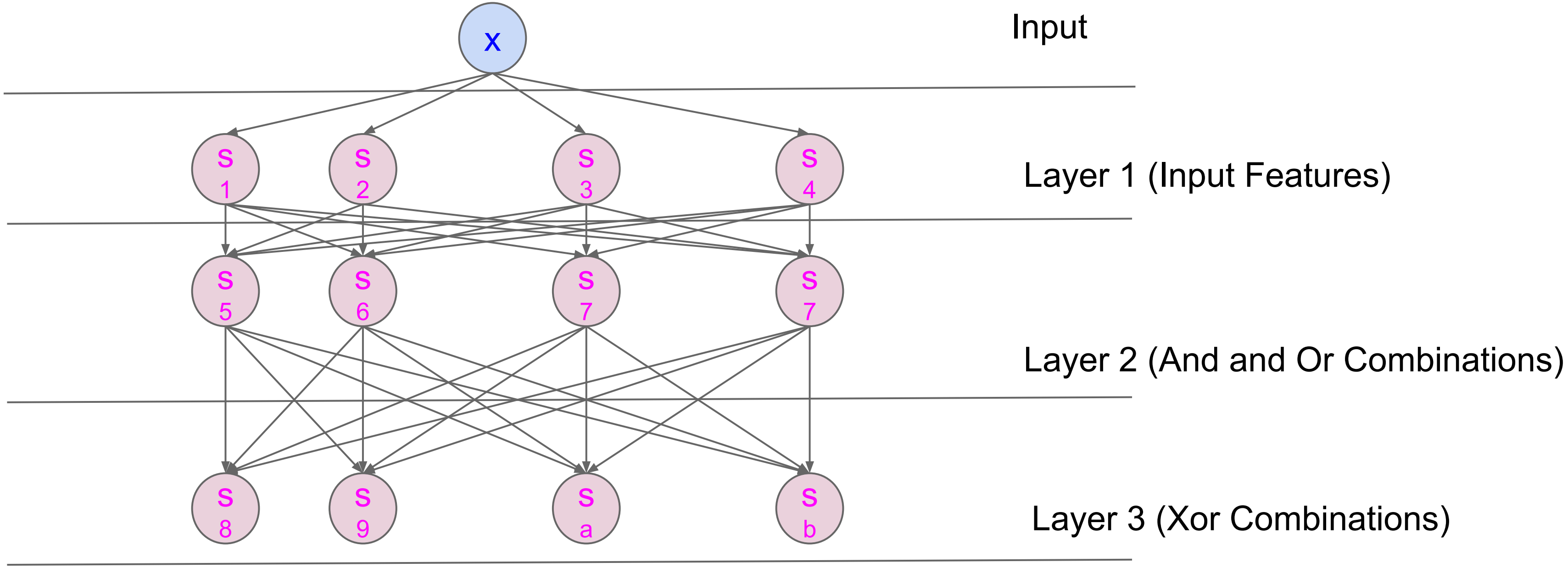
$$\mathrm{And}(s_1,s_2) = \boldsymbol{\sigma}(1000s_1 + 1000s_3 - 1500)$$

$$\mathrm{Or}(s_1,s_2) = \boldsymbol{\sigma}(1000s_1 + 1000s_3 - 500)$$

Multilayer Perceptrons



Multilayer Perceptrons

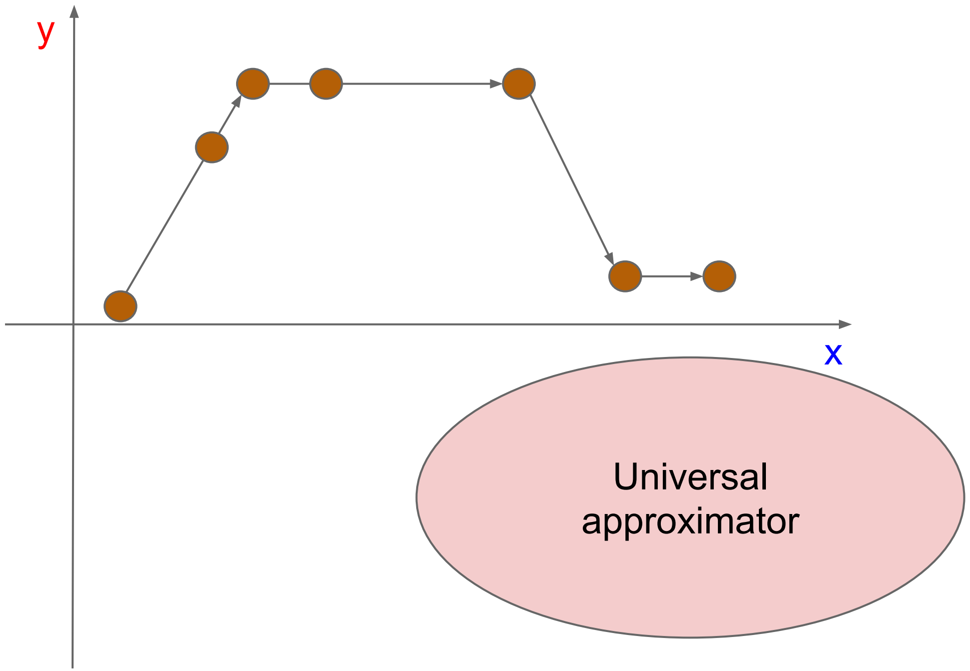


$$Xor(s_1,s_2) = Or(And(s_1,!s_2), And(!s_1,s_2))$$

Multilayer Perceptrons

Data

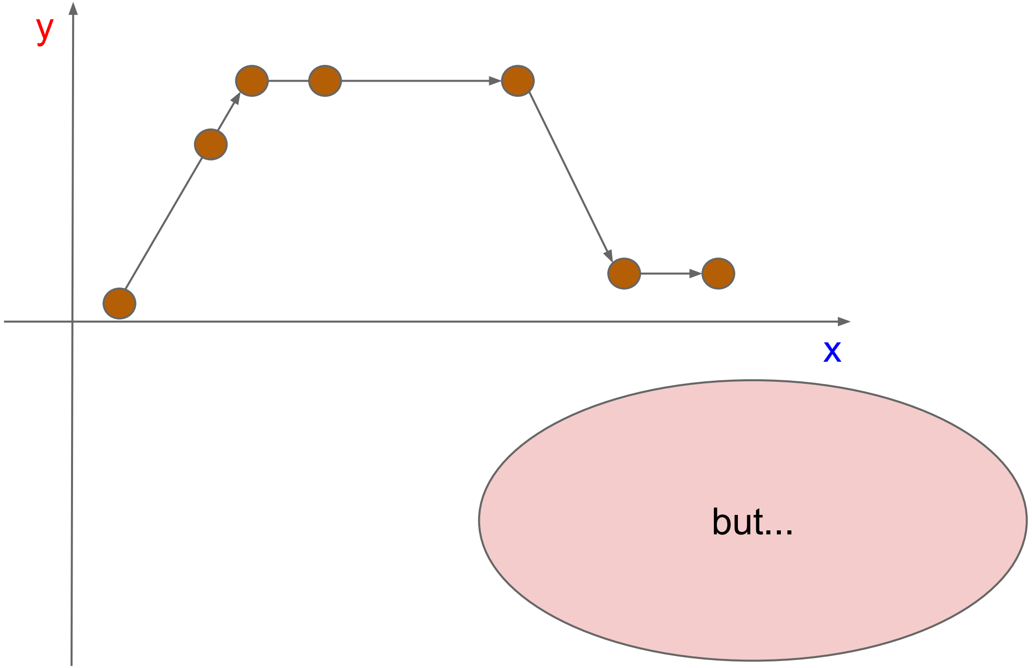
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1



Multilayer Perceptrons

Data

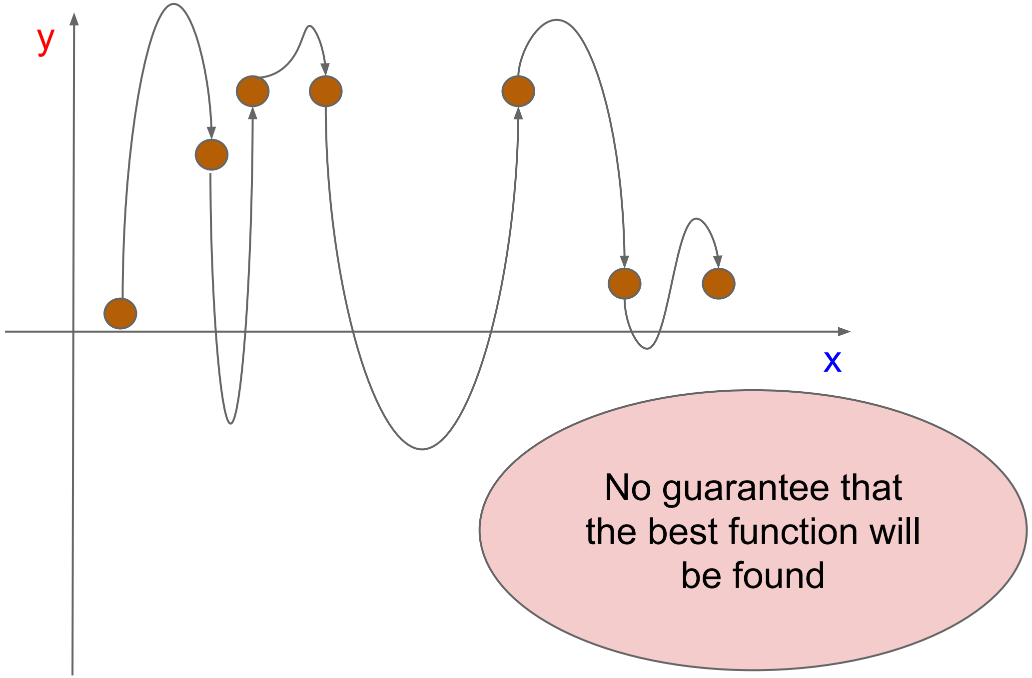
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1



Multilayer Perceptrons

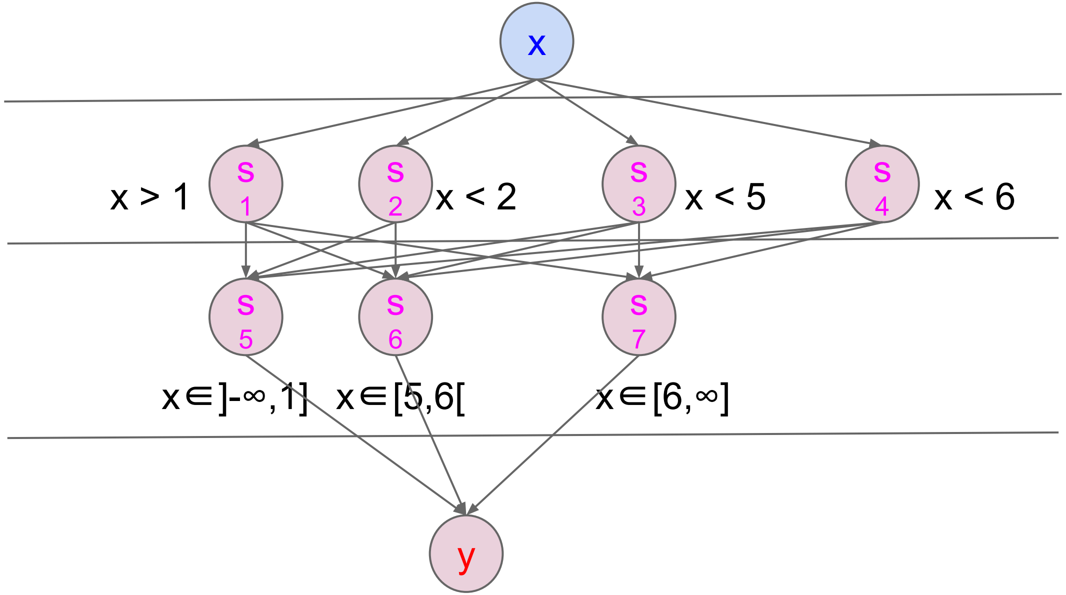
Data

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1



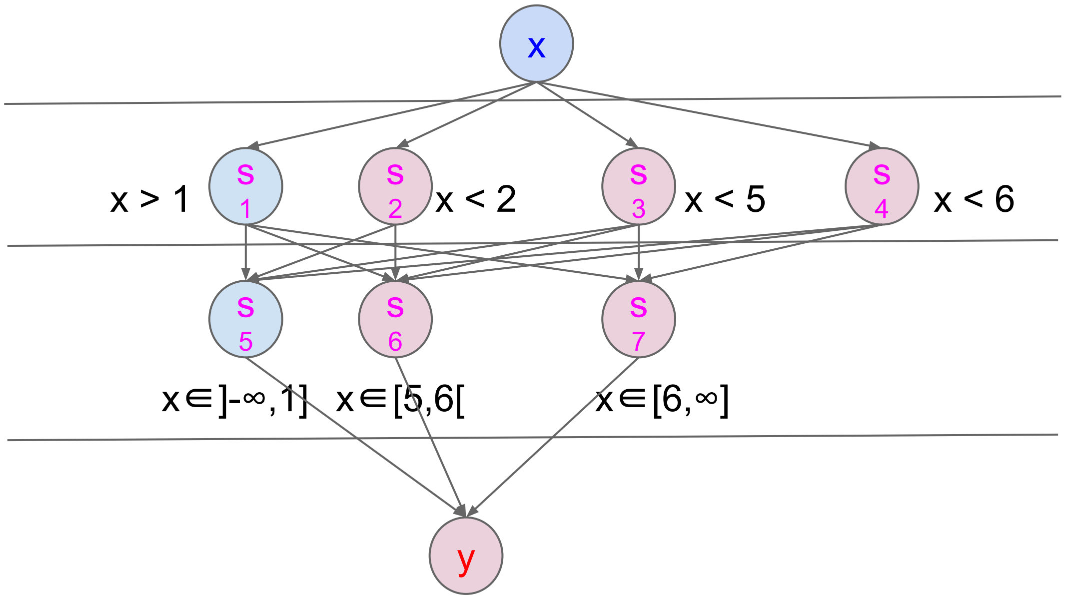
Multilayer Perceptrons

n	x	ŷ
0	1	0
1	5	16
2	6	20



Multilayer Perceptrons

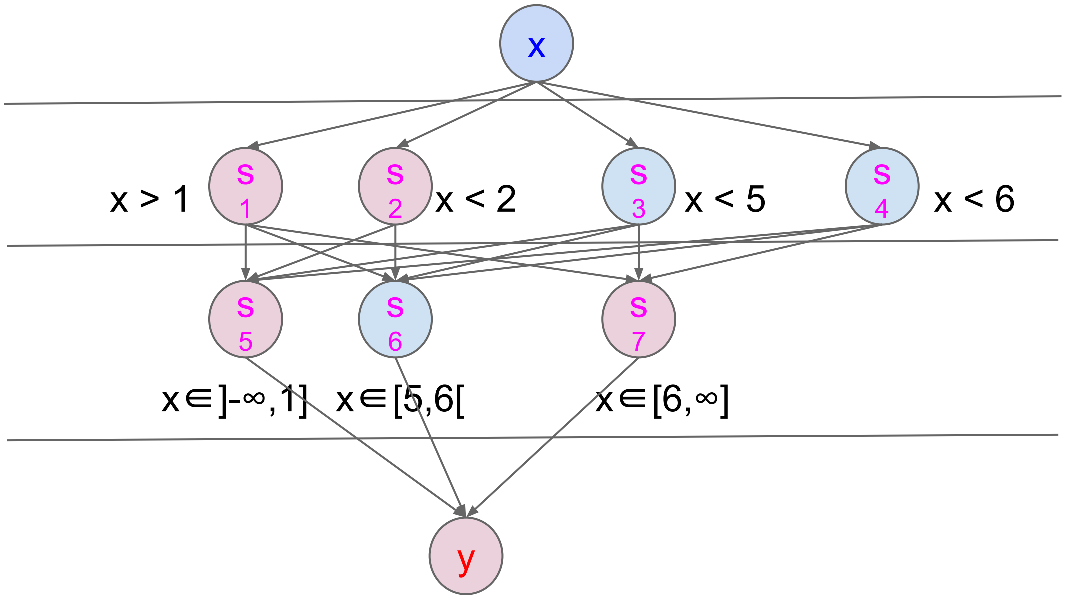
n	x	ŷ
0	1	0
1	5	16
2	6	20



$$y = 0s_5 + 16s_6 + 20s_7$$

Multilayer Perceptrons

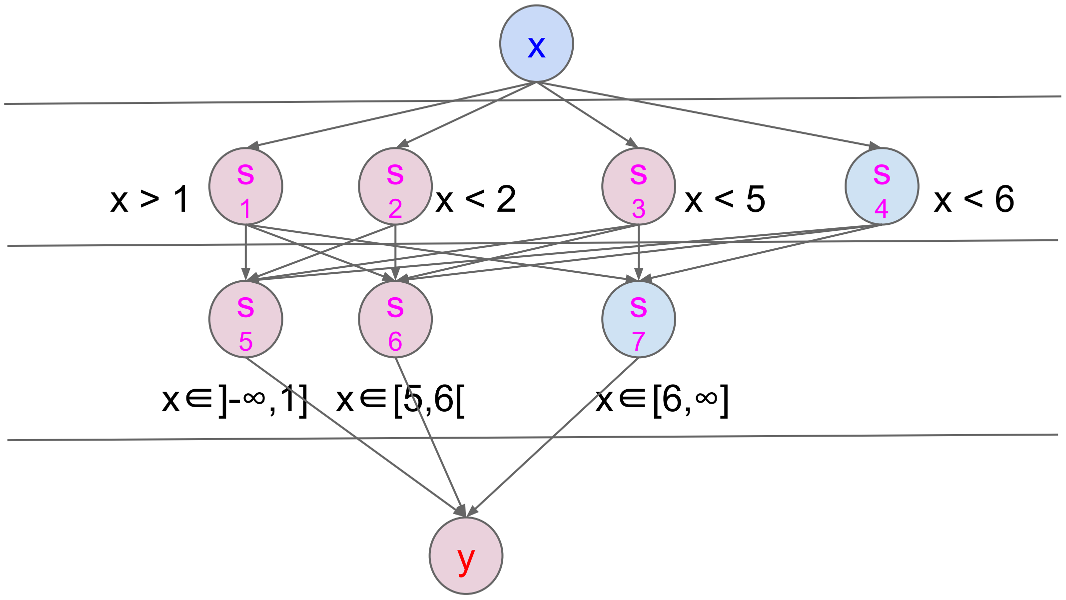
n	x	ŷ
0	1	0
1	5	16
2	6	20



$$y = 0s_5 + 16s_6 + 20s_7$$

Multilayer Perceptrons

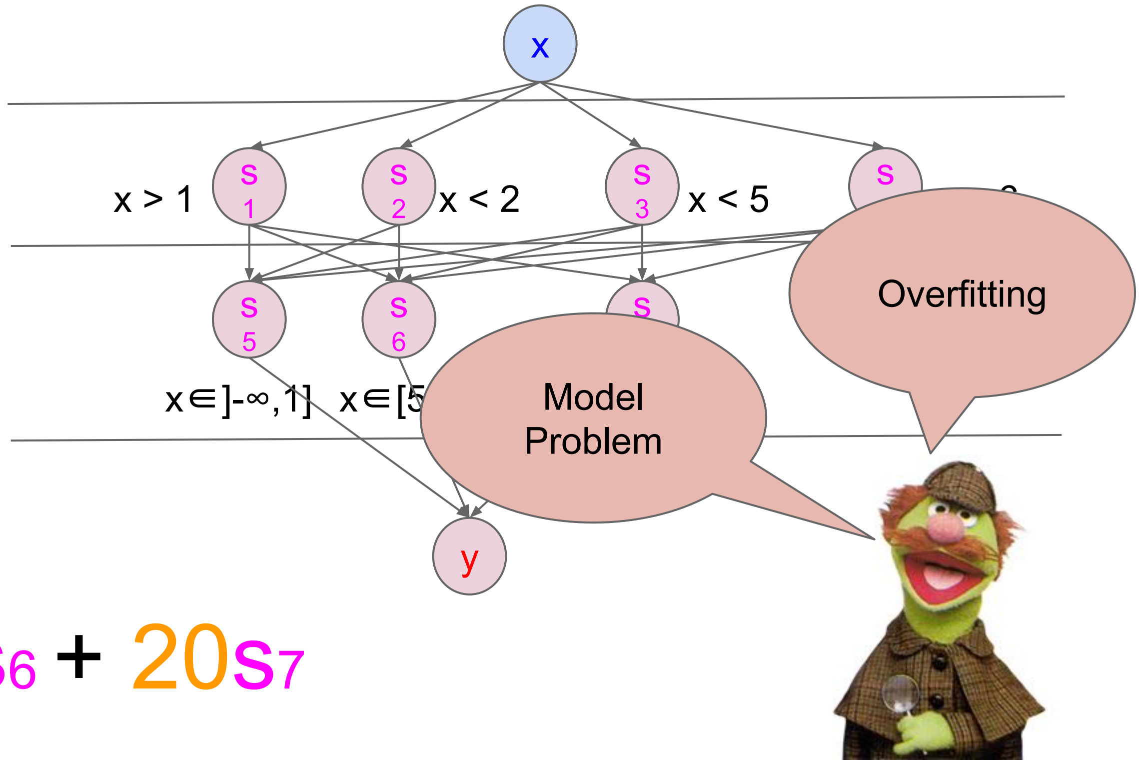
n	x	ŷ
0	1	0
1	5	16
2	6	20



$$y = 0s_5 + 16s_6 + 20s_7$$

Multilayer Perceptrons

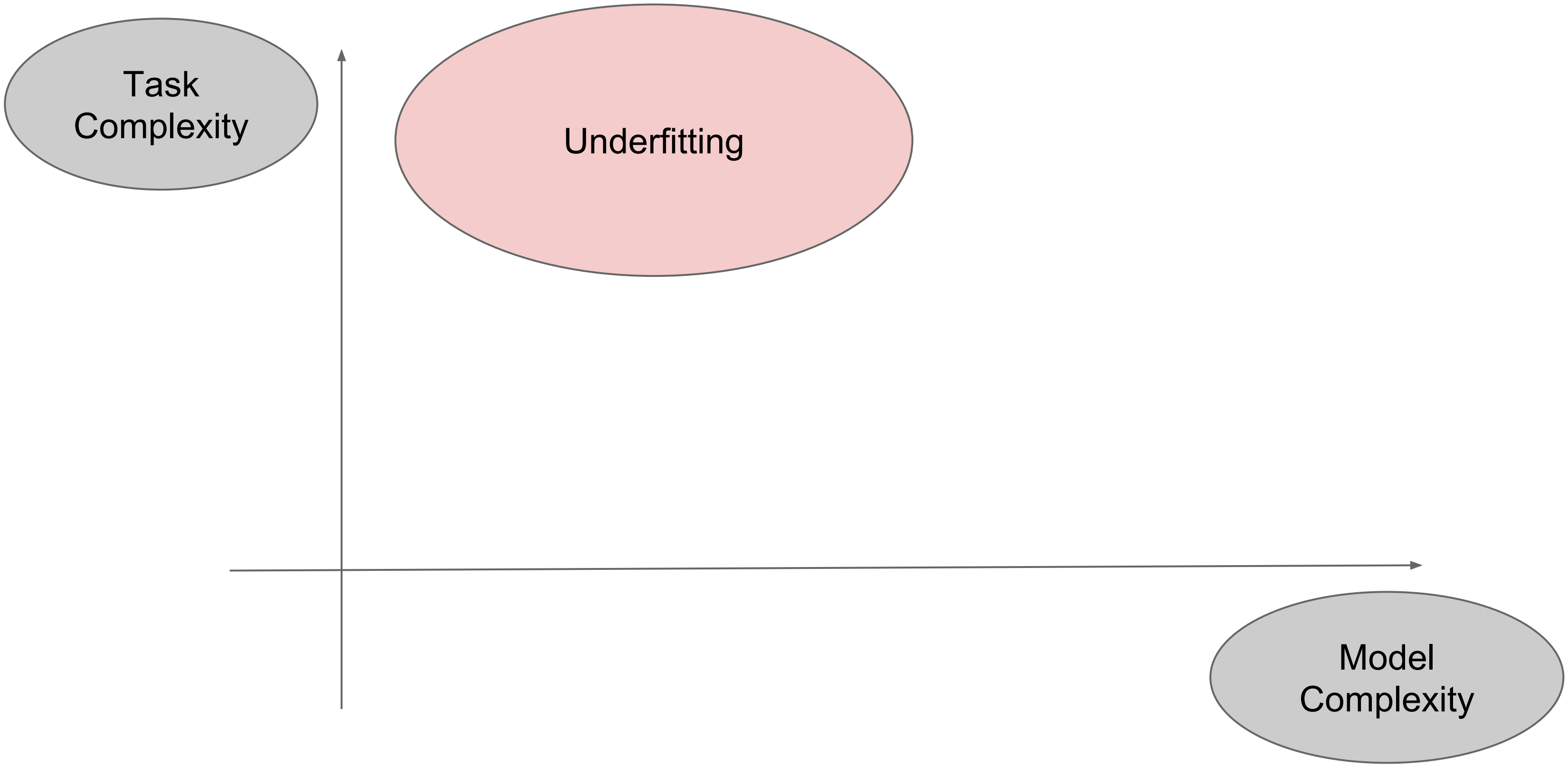
n	x	ŷ
0	1	0
1	5	16
2	6	20



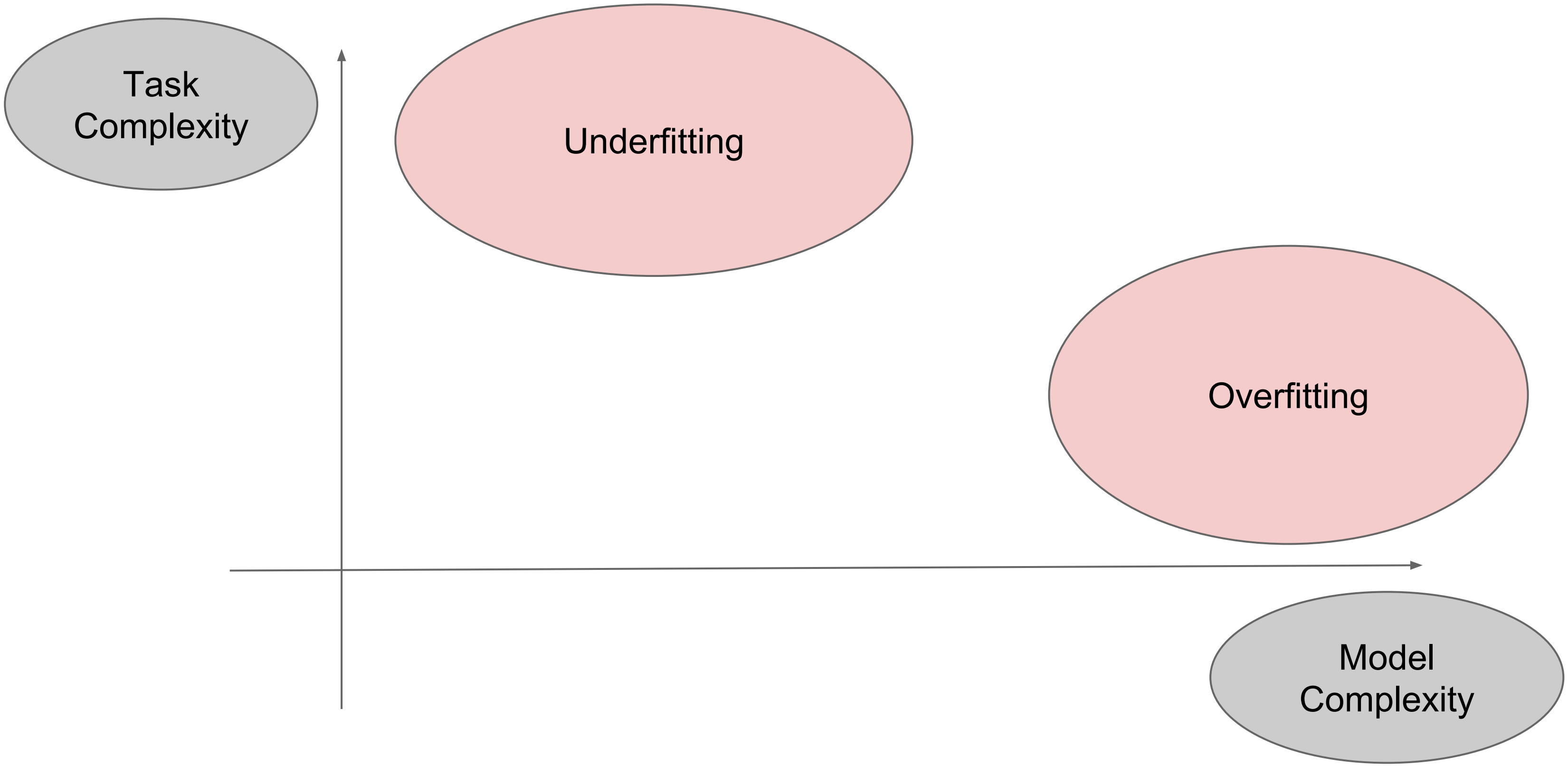
$$y = 0s_5 + 16s_6 + 20s_7$$

Multilayer Perceptrons

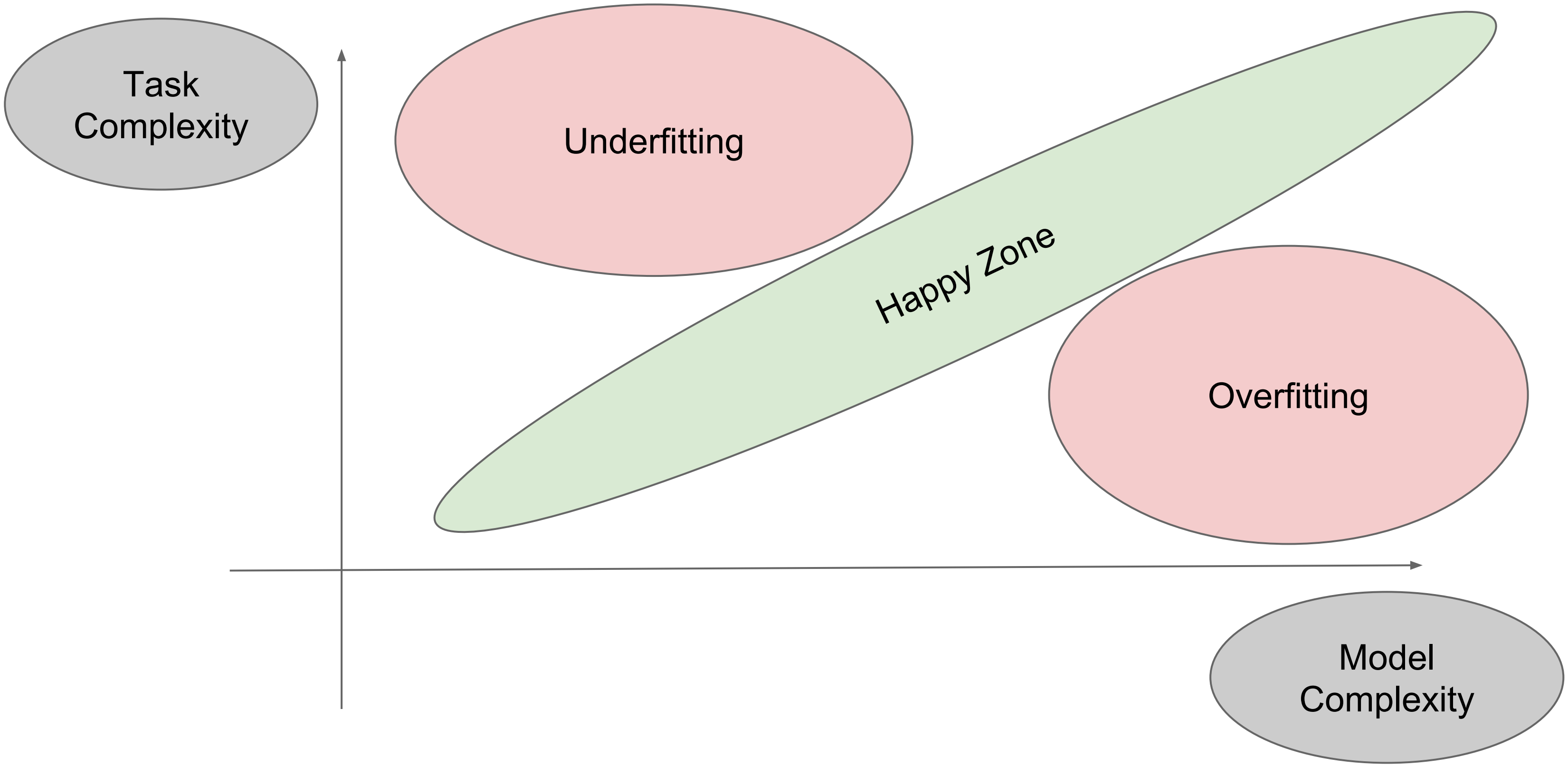
Multilayer Perceptrons



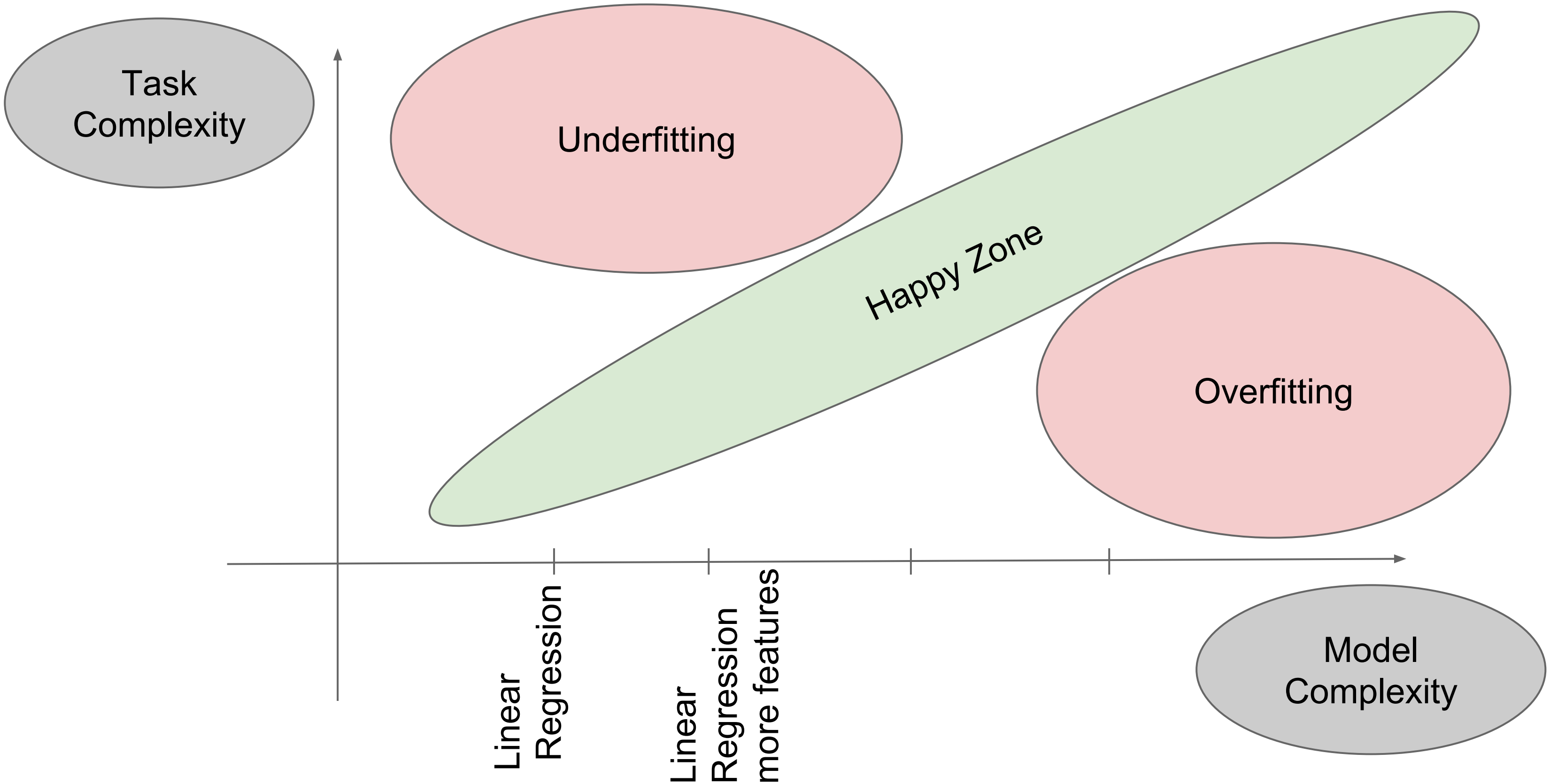
Multilayer Perceptrons



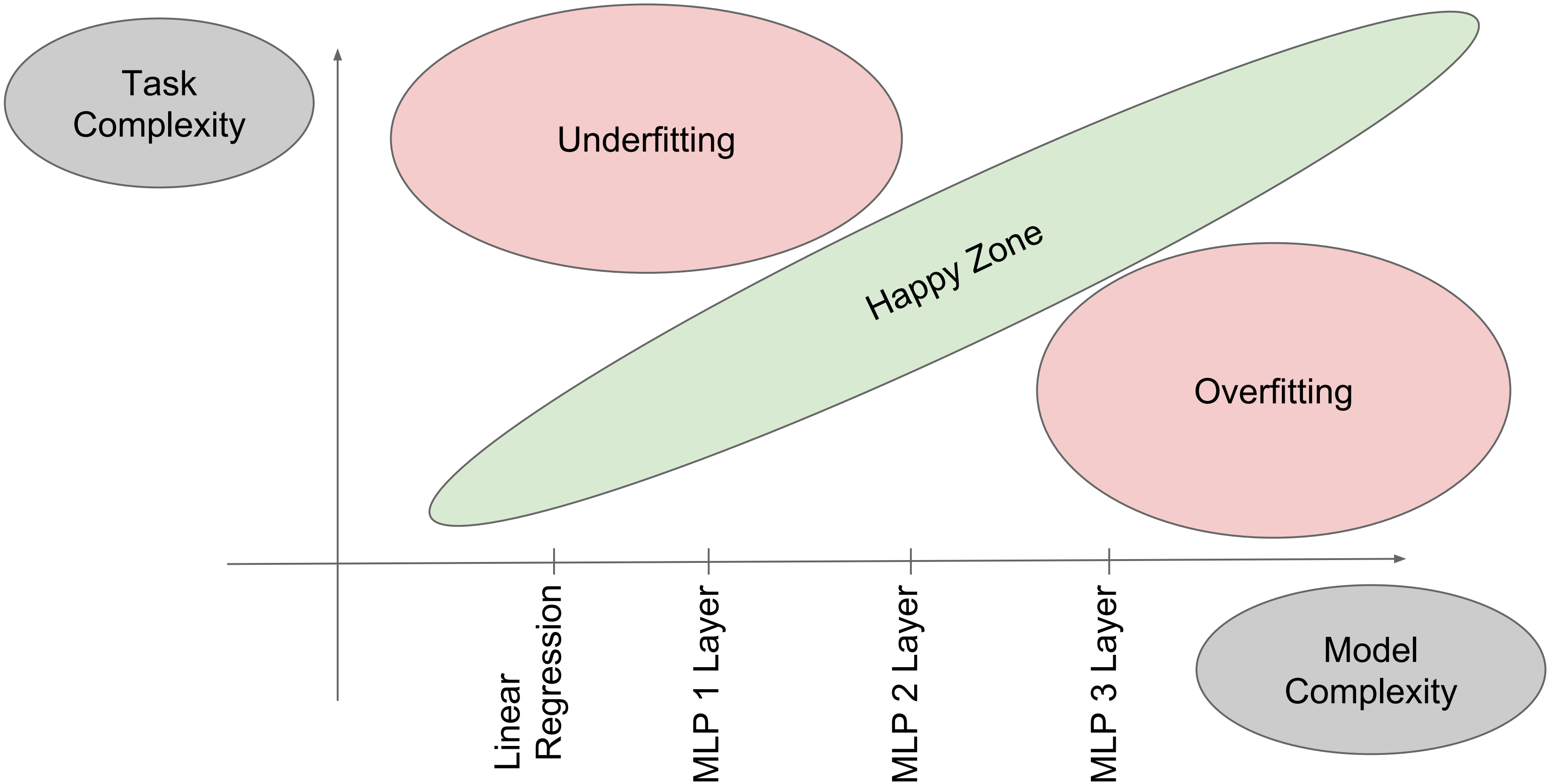
Multilayer Perceptrons



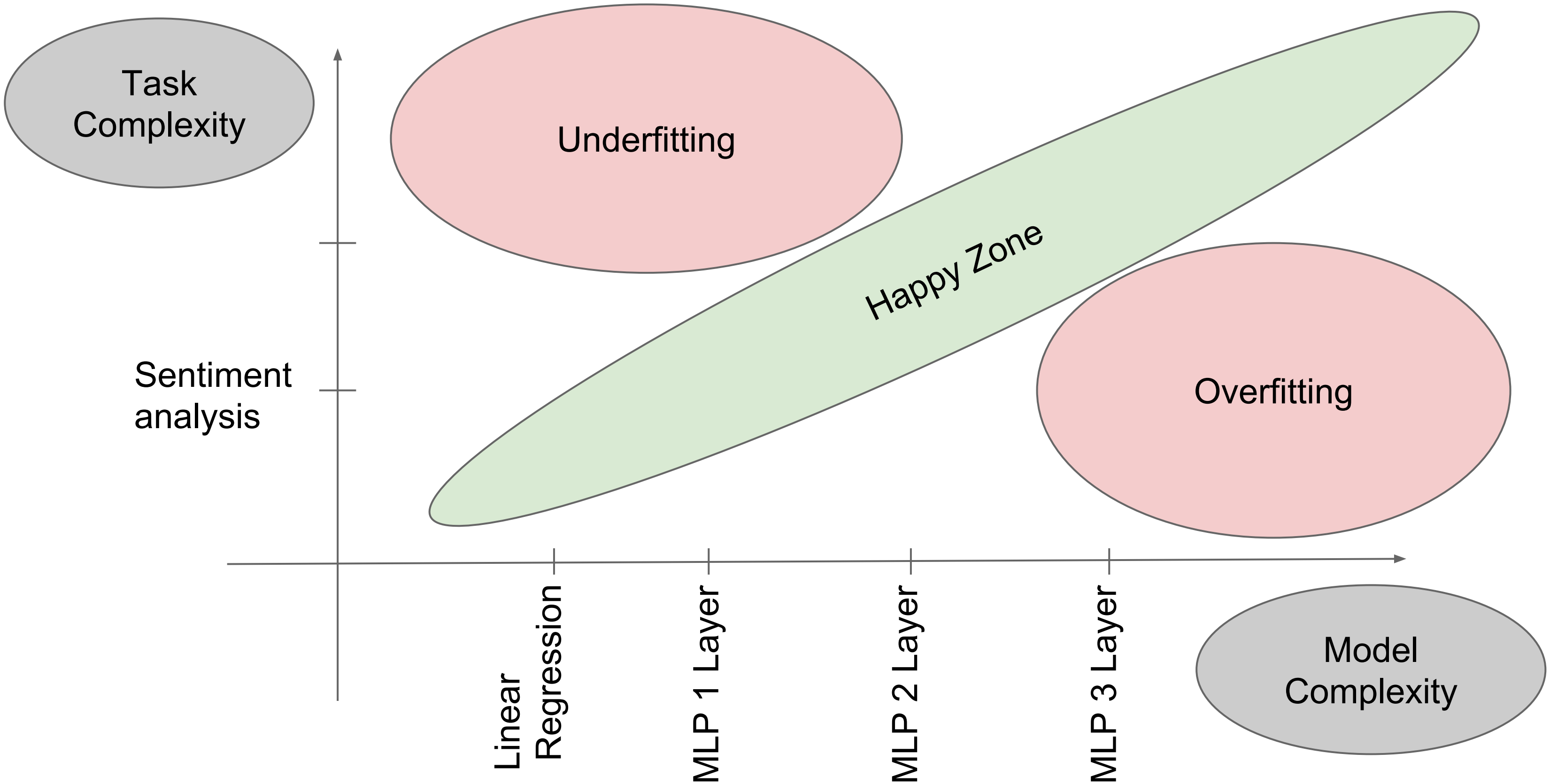
Multilayer Perceptrons



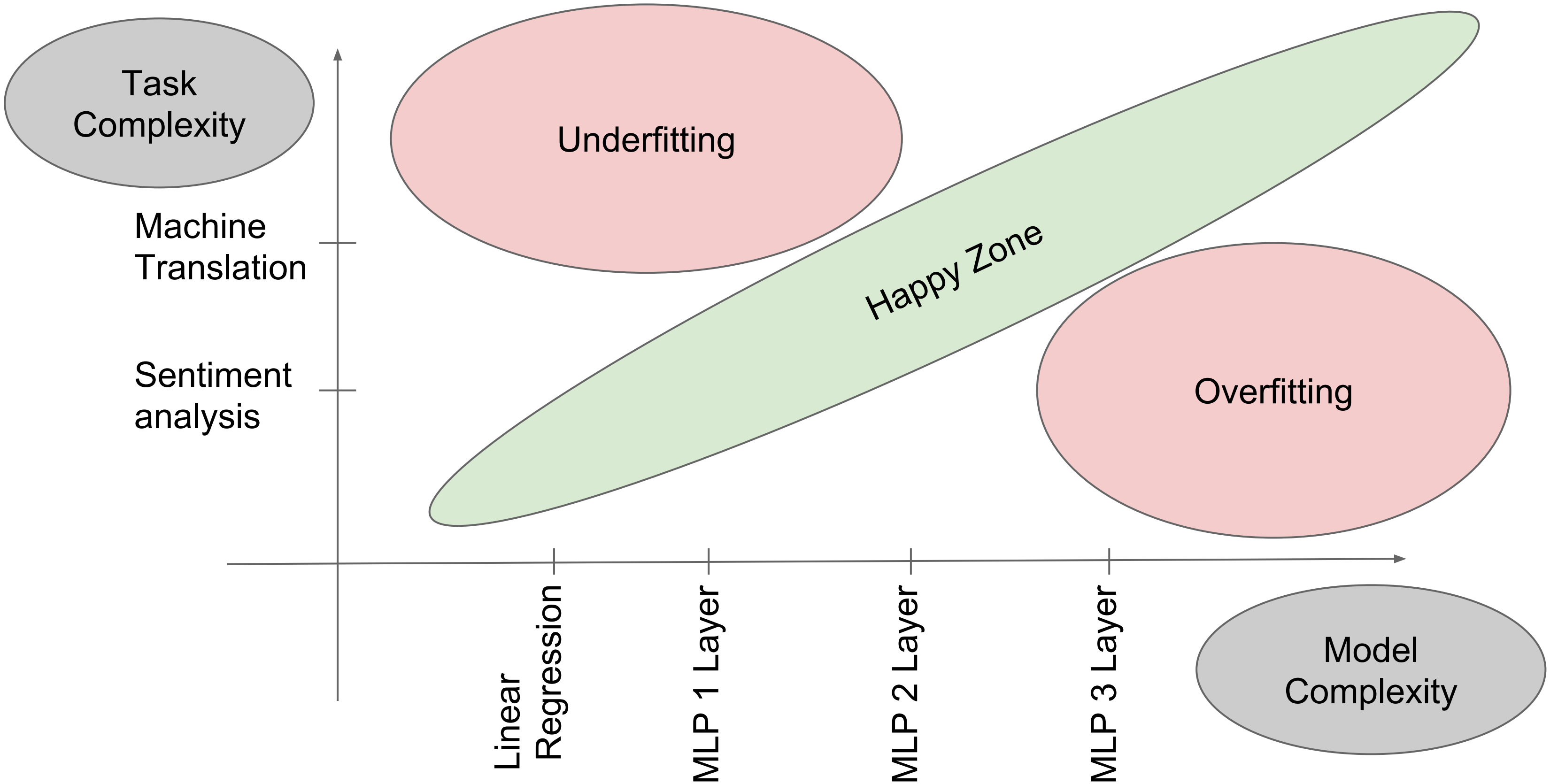
Multilayer Perceptrons



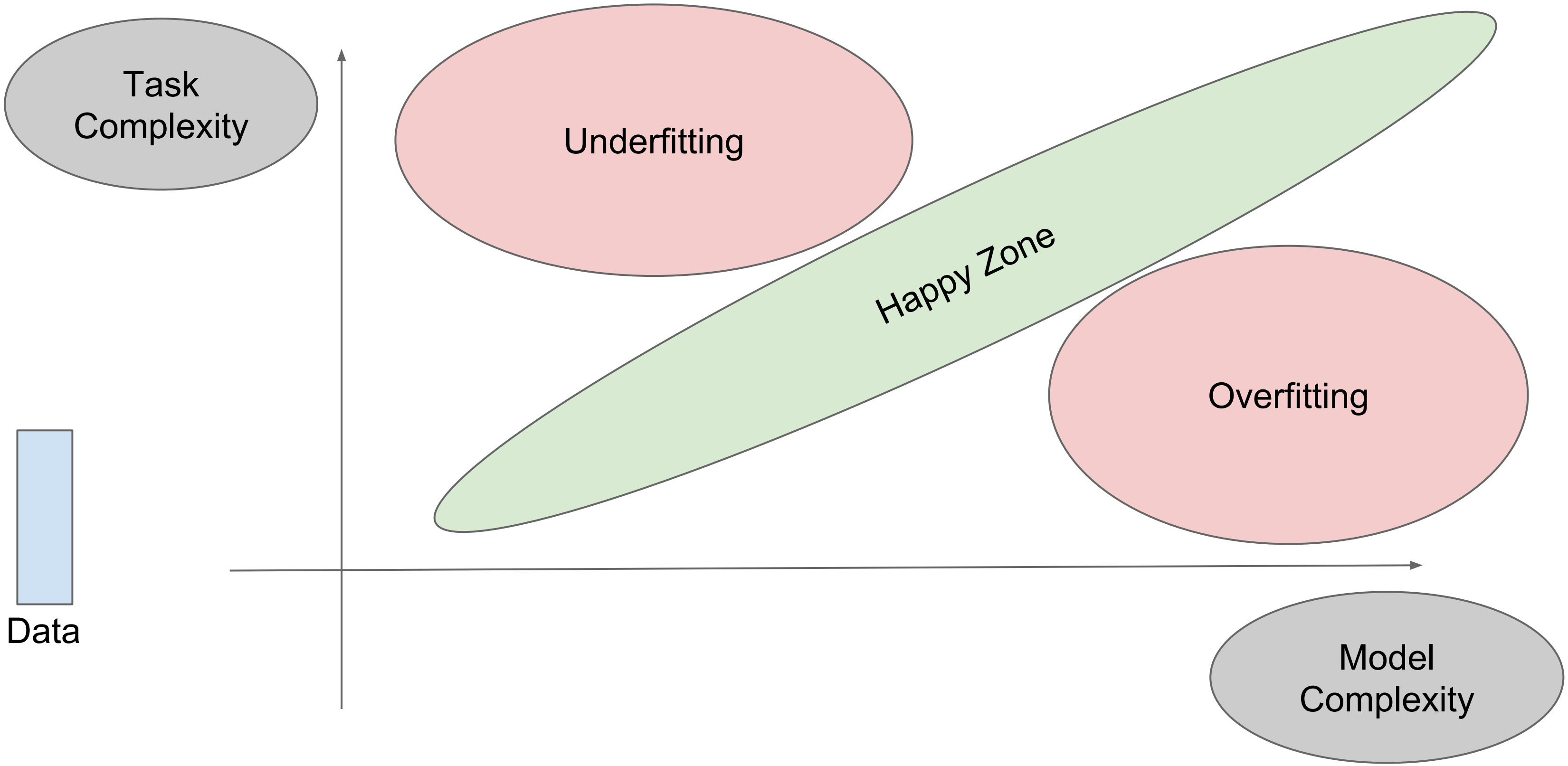
Multilayer Perceptrons



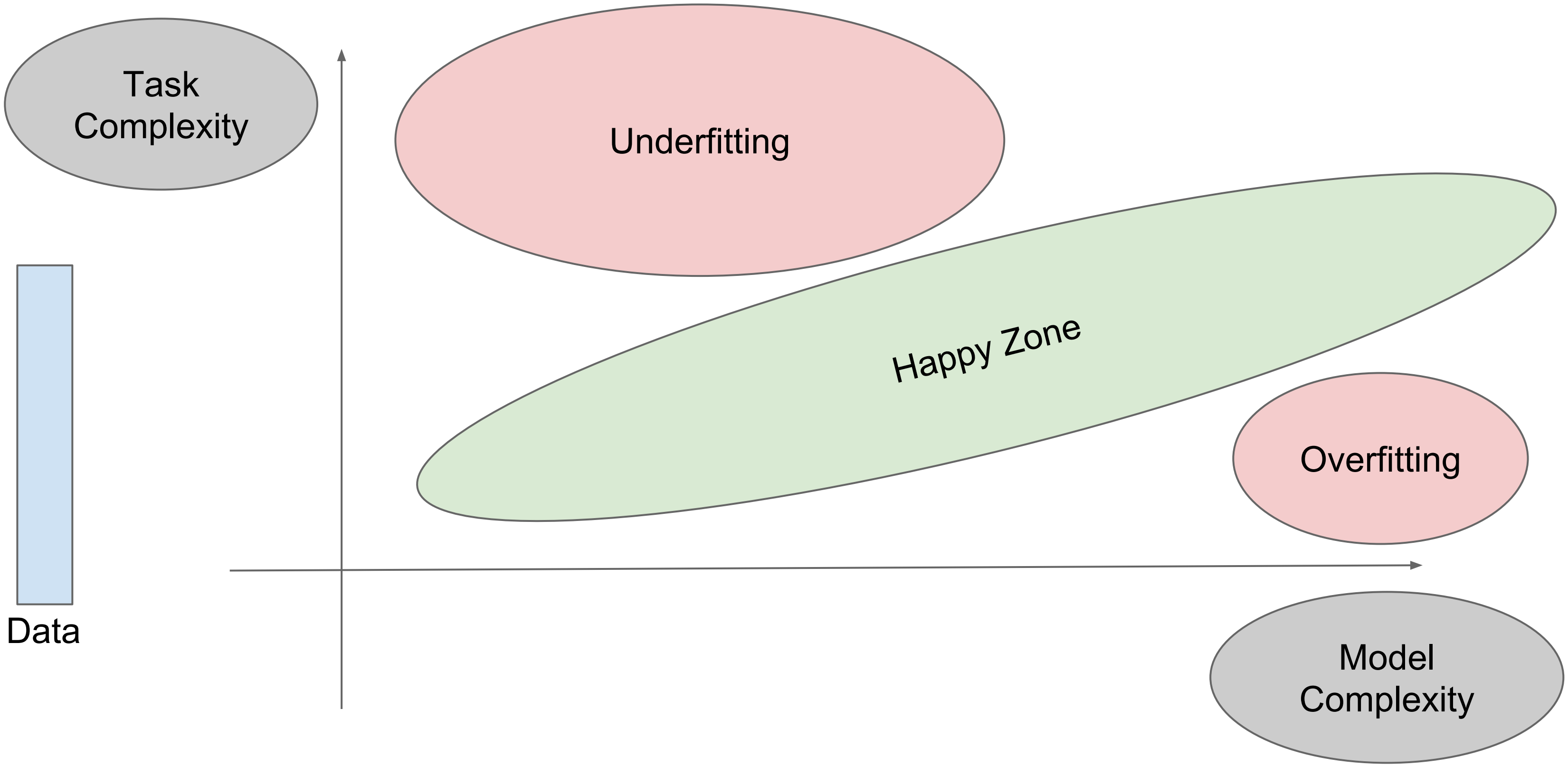
Multilayer Perceptrons



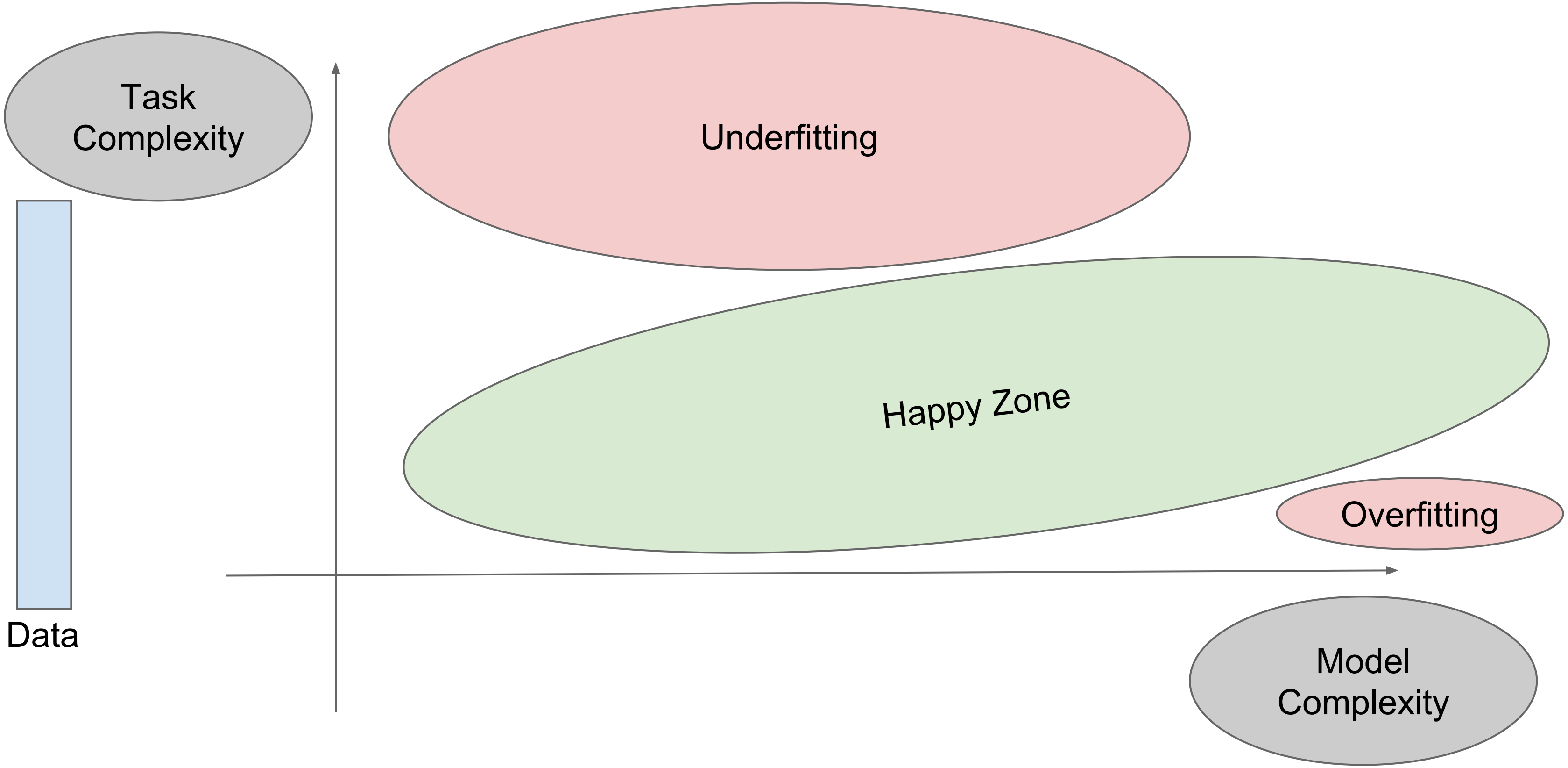
Multilayer Perceptrons



Multilayer Perceptrons

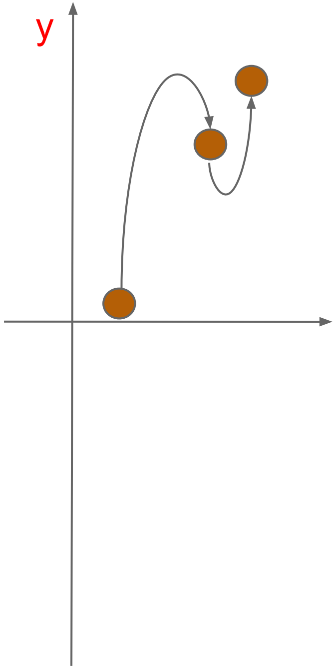


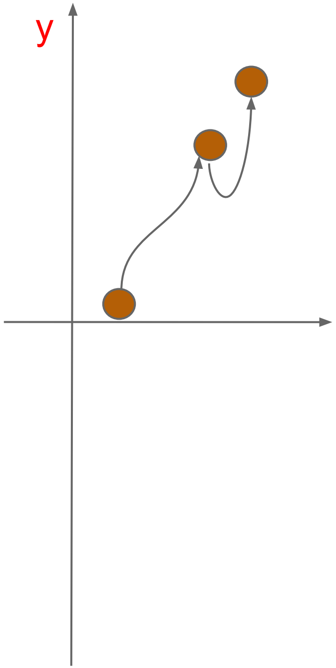
Multilayer Perceptrons

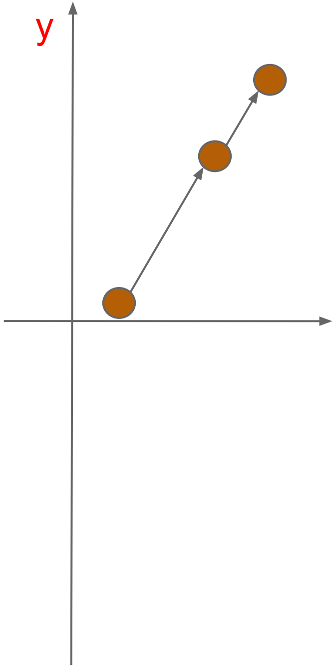


Multilayer Perceptrons

n	x	ŷ
0	1	0
1	5	16
2	6	20

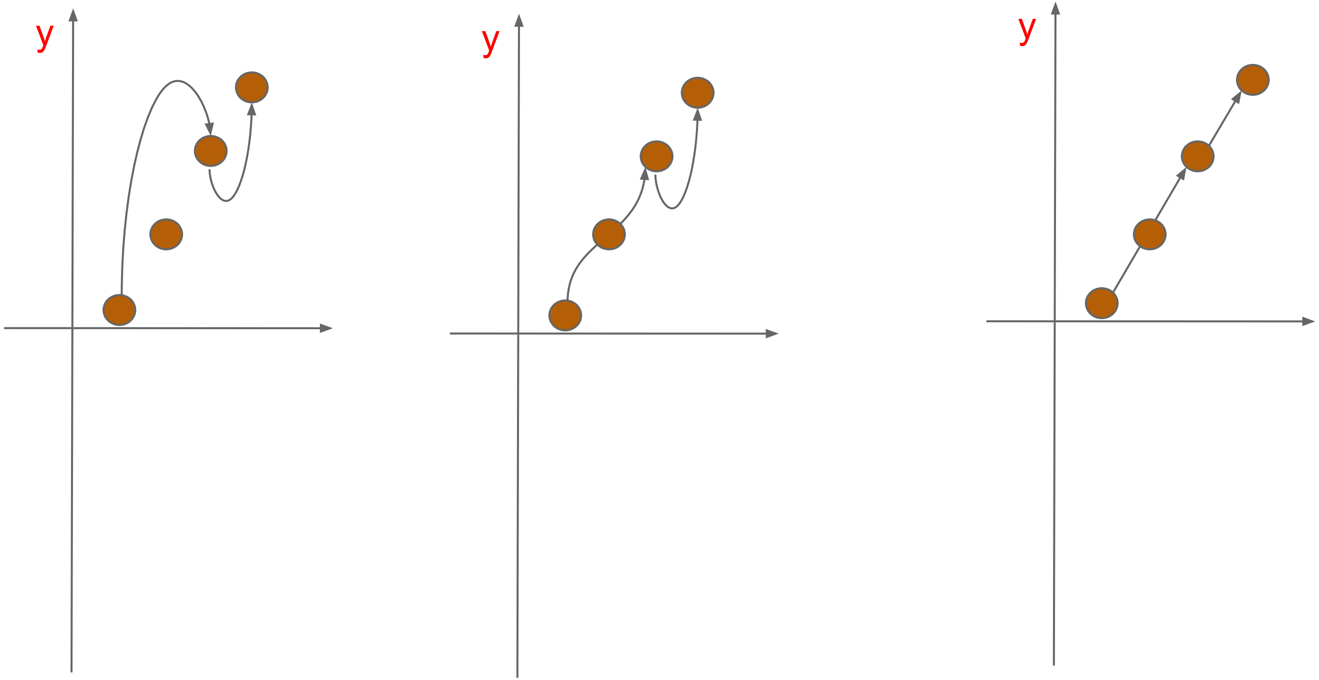






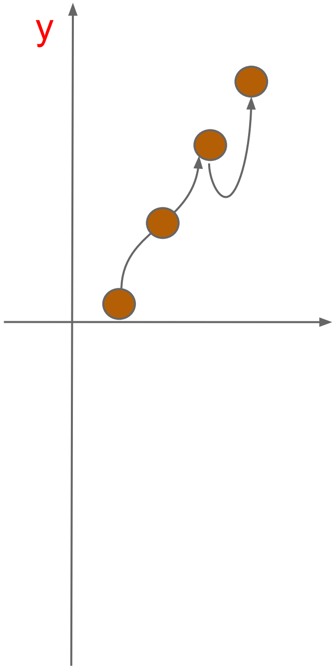
Multilayer Perceptrons

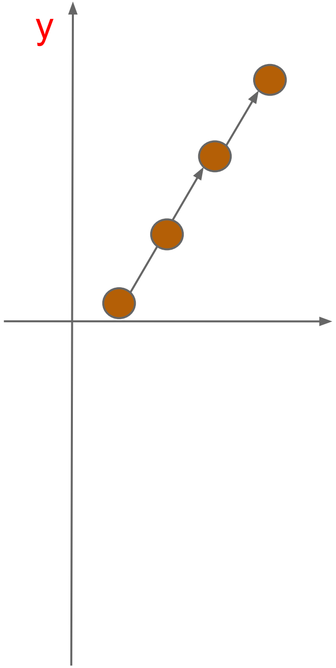
n	x	ŷ
0	1	0
1	5	16
2	6	20
3	2	4



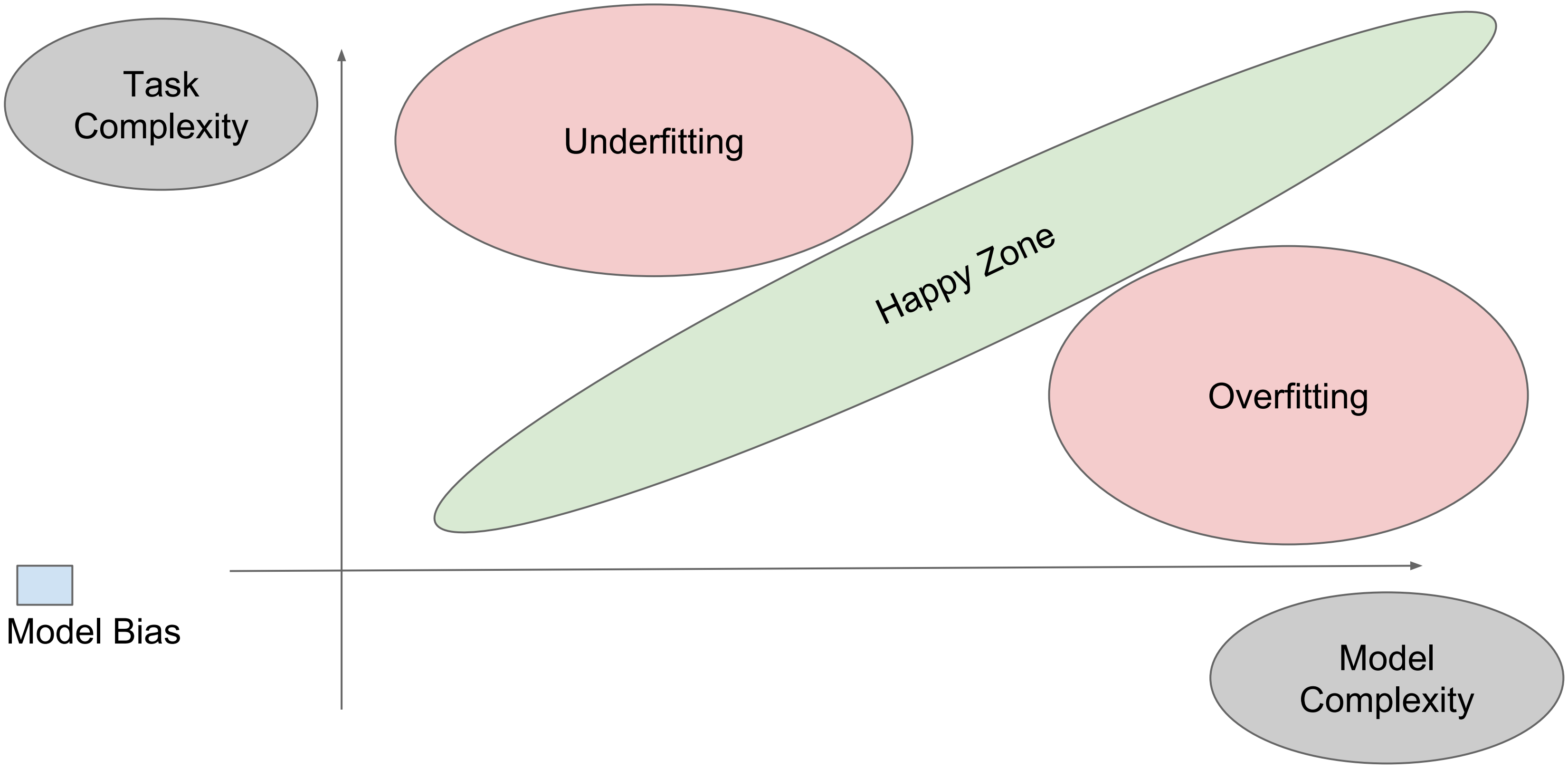
Multilayer Perceptrons

n	x	ŷ
0	1	0
1	5	16
2	6	20
3	2	4

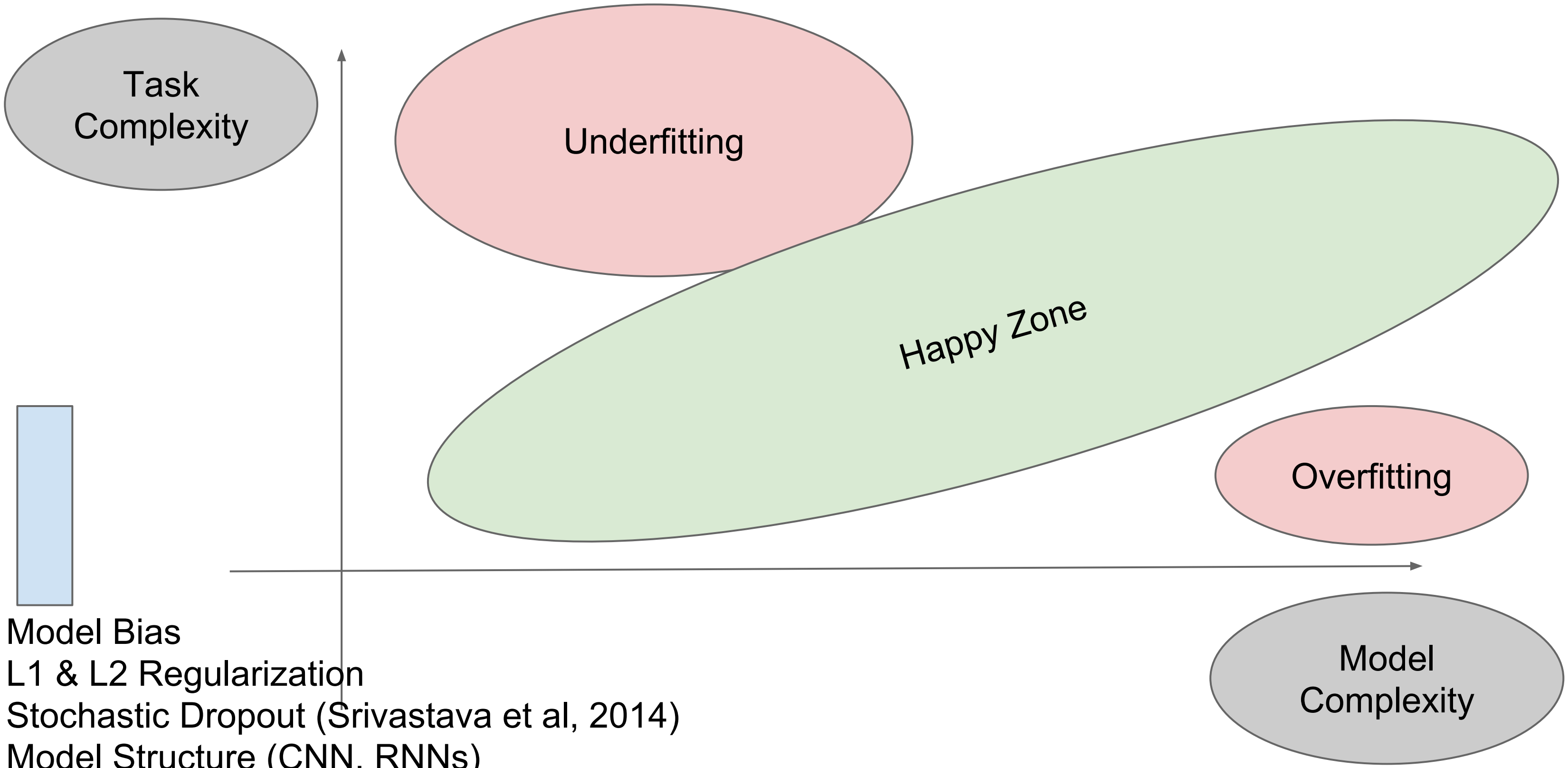




Multilayer Perceptrons



Multilayer Perceptrons



Model Bias
L1 & L2 Regularization
Stochastic Dropout (Srivastava et al, 2014)
Model Structure (CNN, RNNs)

Multilayer Perceptrons

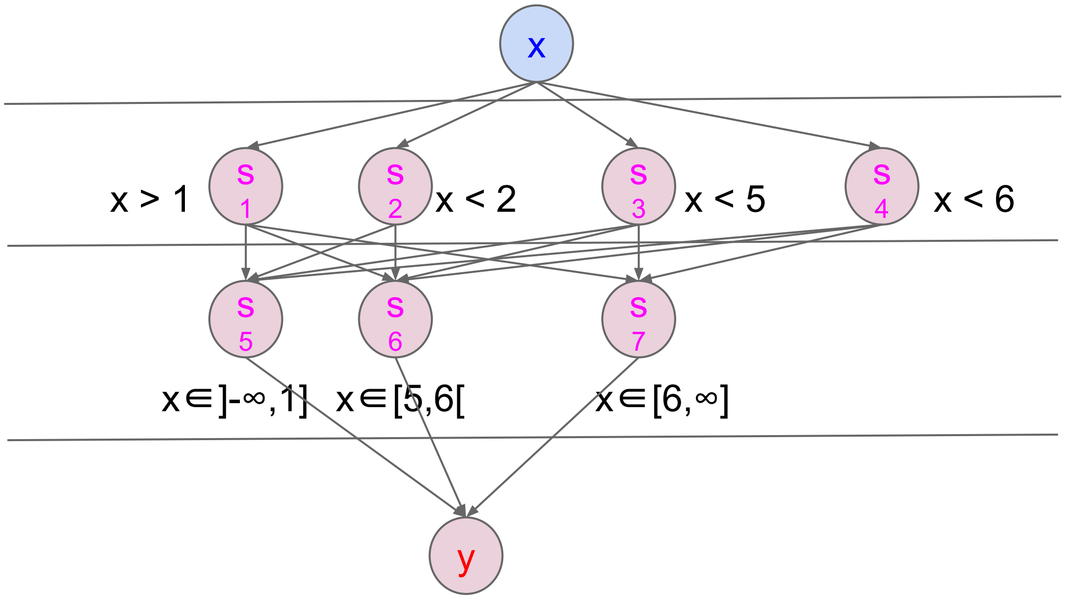
Regularization

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2 + (w+b)ß$$

ß = Regularization constant

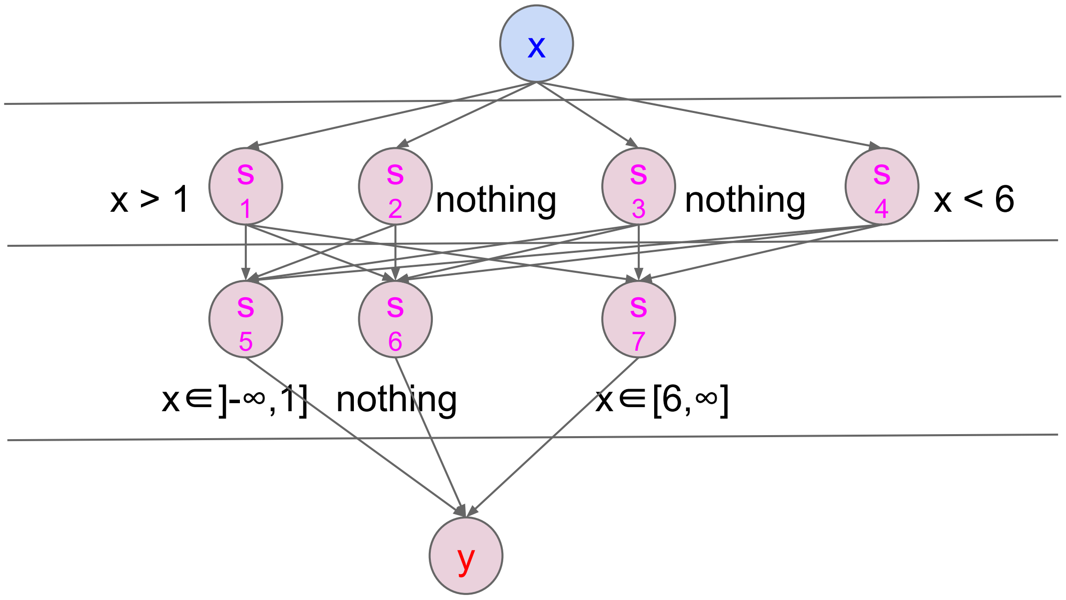
Multilayer Perceptrons

Regularization



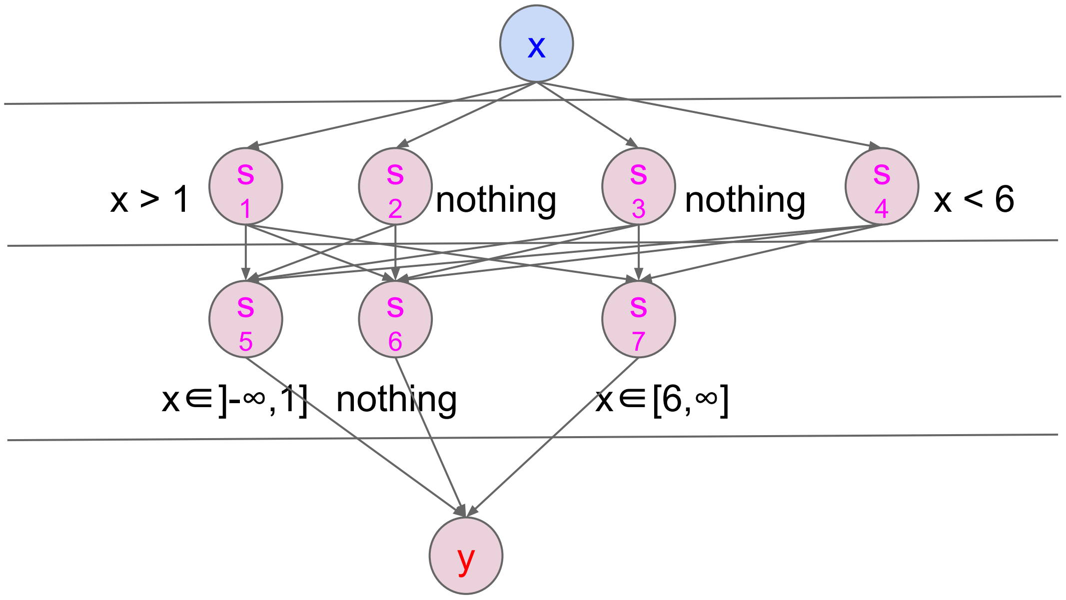
Multilayer Perceptrons

Regularization



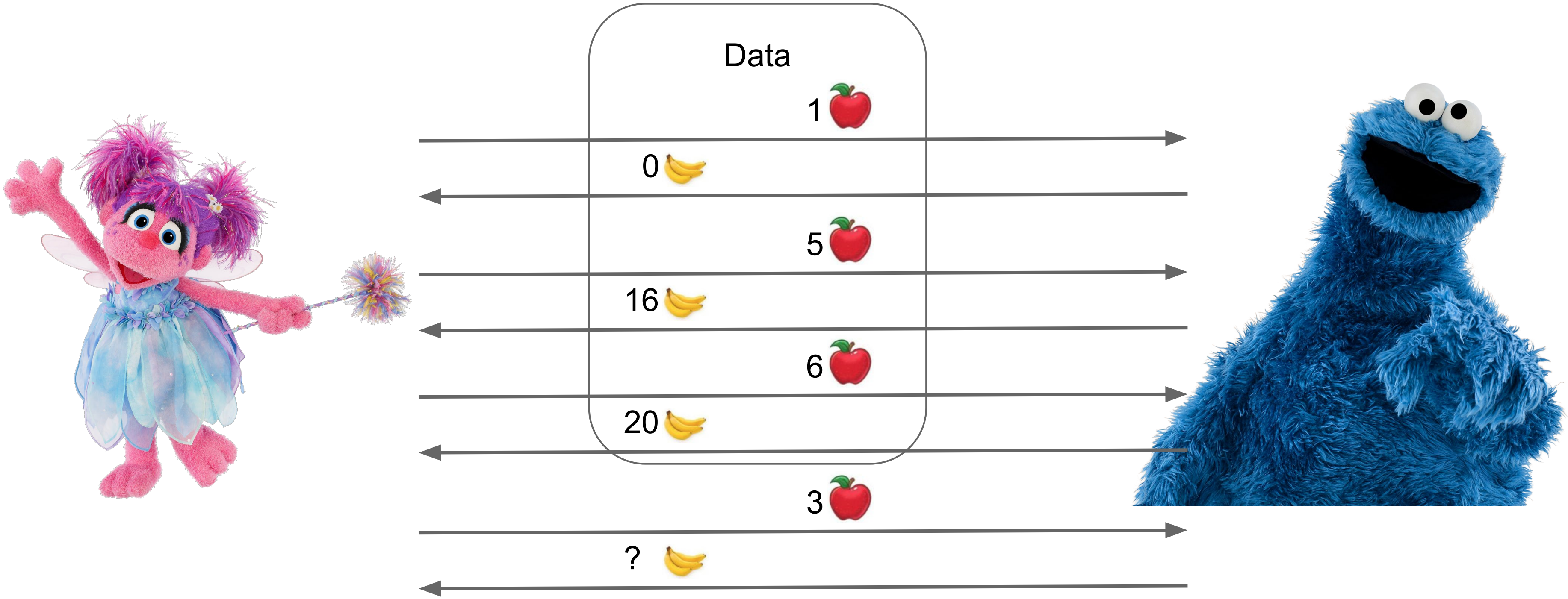
Multilayer Perceptrons

Regularization

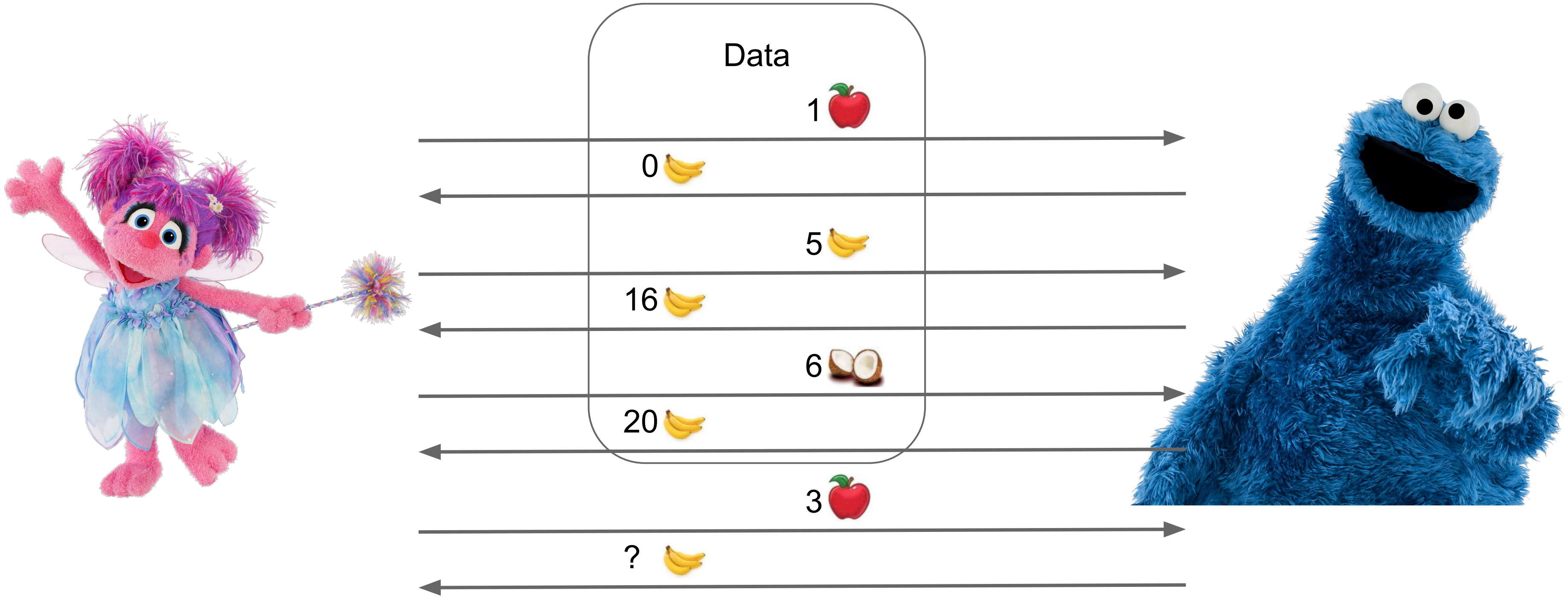


Find solutions that require less effort

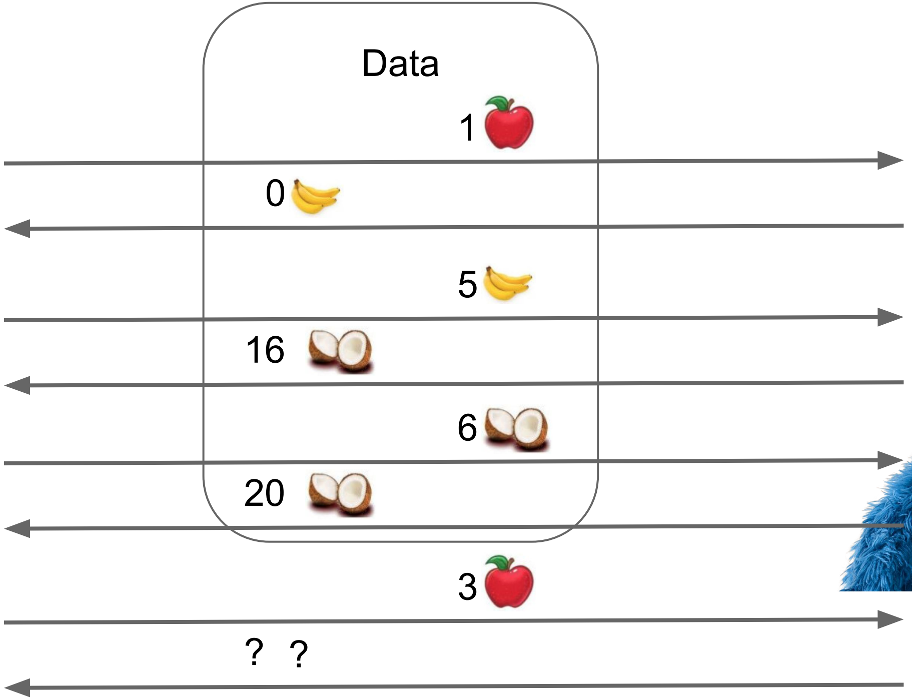
Using Discrete Variables



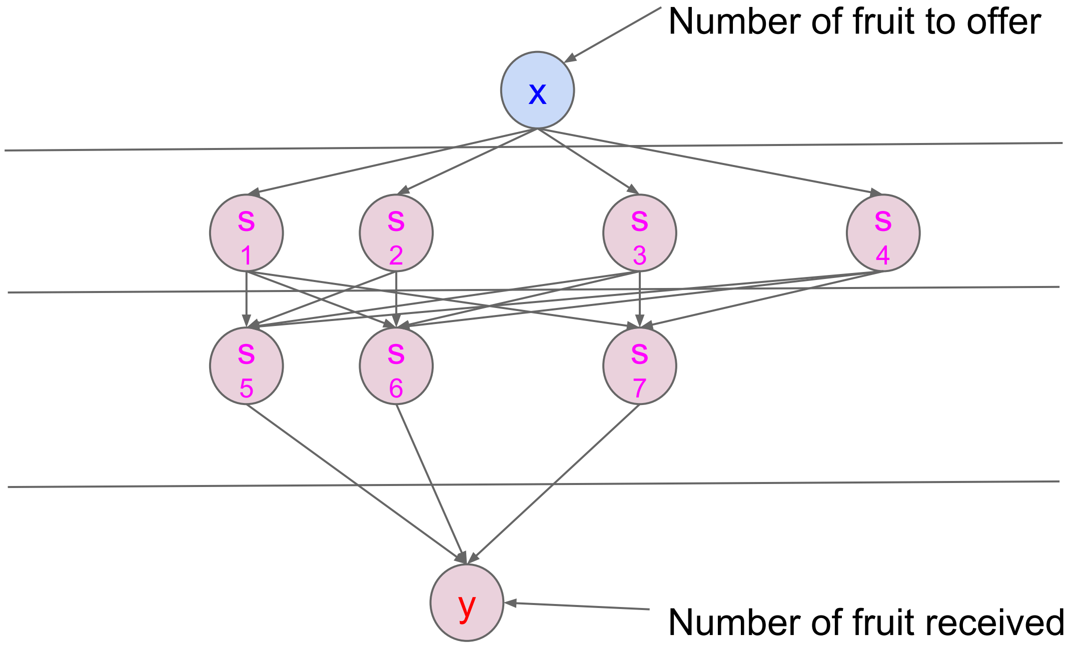
Using Discrete Variables



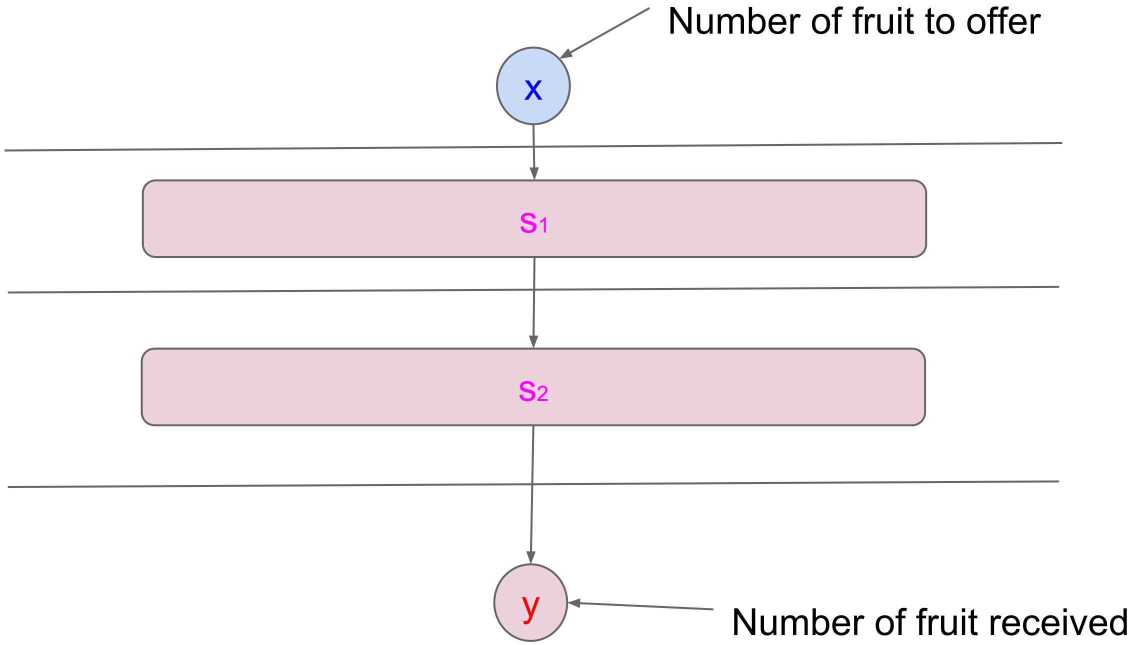
Using Discrete Variables



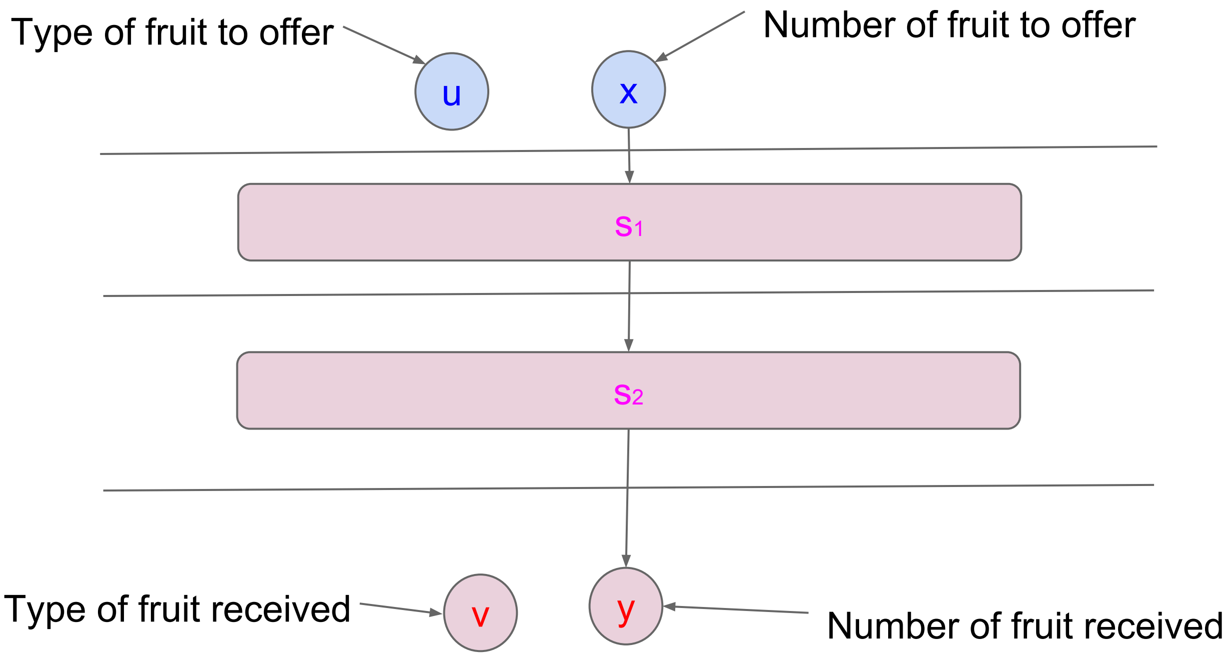
Using Discrete Variables



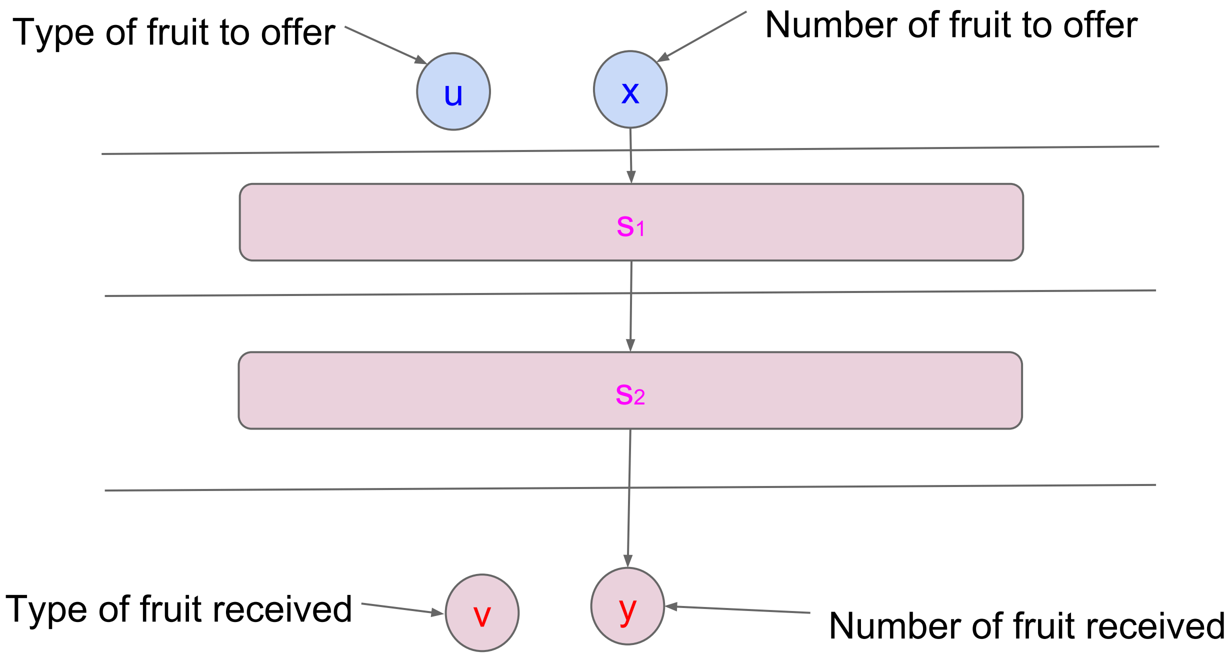
Using Discrete Variables



Using Discrete Variables



Using Discrete Variables

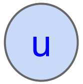


$u \in \{\text{Apple, Banana, Coconut}\}$

$v \in \{\text{Apple, Banana, Coconut}\}$

Using Discrete Variables

Lookup Tables



	e_1	e_2	e_3	e_4
Apple	0.1	-0.4	0.2	0.5
Banana	0.4	1.4	-1.0	0.1
Coconut	1.1	0.9	1.1	0.5

V = 3

Using Discrete Variables

Lookup Tables

u

	e_1	e_2	e_3	e_4
Apple	0.1	-0.4	0.2	0.5
Banana	0.4	1.4	-1.0	0.1
Coconut	1.1	0.9	1.1	0.5

V = 3

Using Discrete Variables

Lookup Tables

u

	e_1	e_2	e_3	e_4
Apple	0.1	-0.4	0.2	0.5
Banana	0.4	1.4	-1.0	0.1
Coconut	1.1	0.9	1.1	0.5

V = 3

Embedding for u

Size = 4

Using Discrete Variables

Lookup Tables

u Banana

	e_1	e_2	e_3	e_4
Apple	0.1	-0.4	0.2	0.5
Banana	0.4	1.4	-1.0	0.1
Coconut	1.1	0.9	1.1	0.5

V = 3

Embedding for u

Size = 4

Using Discrete Variables

Lookup Tables

u 1

	e_1	e_2	e_3	e_4
0	0.1	-0.4	0.2	0.5
1	0.4	1.4	-1.0	0.1
2	1.1	0.9	1.1	0.5

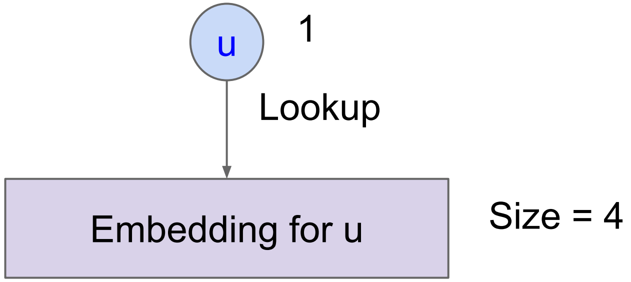
V = 3

Embedding for u

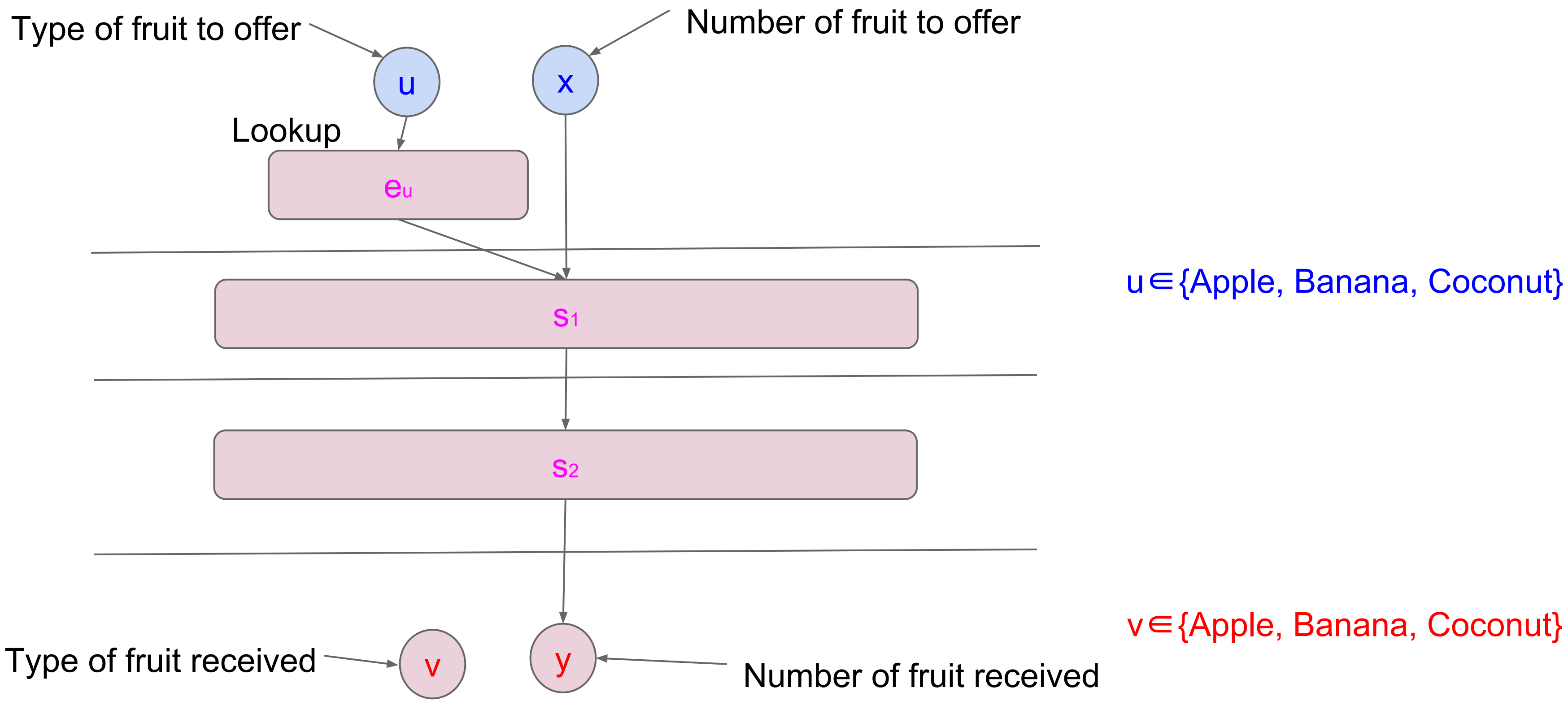
Size = 4

Using Discrete Variables

Lookup Tables



Using Discrete Variables



Using Discrete Variables

Softmax

V = 3

	Apple	Banana	Coconut
W_1	0.1	-0.4	0.2
W_2	0.4	1.4	-1.0
W_3	1.1	0.9	1.1
W_4	1.3	0.1	0.4

Using Discrete Variables

Softmax

Input vector

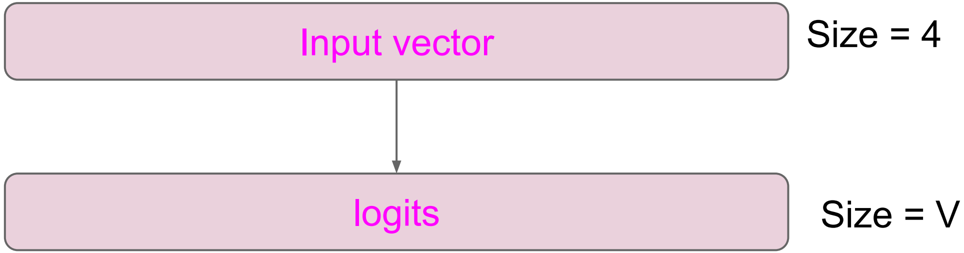
Size = 4

V = 3

	Apple	Banana	Coconut
W_1	0.1	-0.4	0.2
W_2	0.4	1.4	-1.0
W_3	1.1	0.9	1.1
W_4	1.3	0.1	0.4

Using Discrete Variables

Softmax

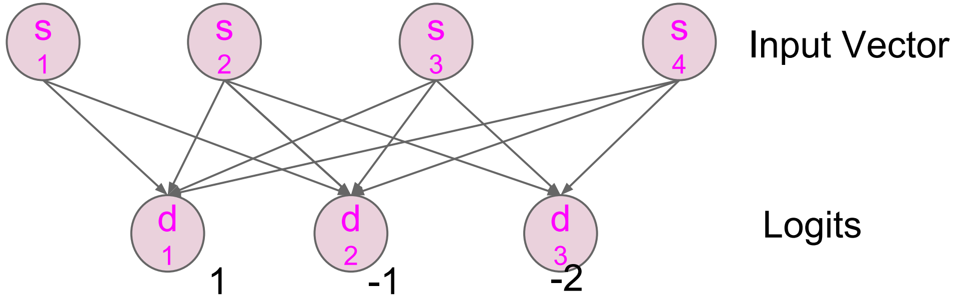


V = 3

	Apple	Banana	Coconut
W_1	0.1	-0.4	0.2
W_2	0.4	1.4	-1.0
W_3	1.1	0.9	1.1
W_4	1.3	0.1	0.4

Using Discrete Variables

Softmax

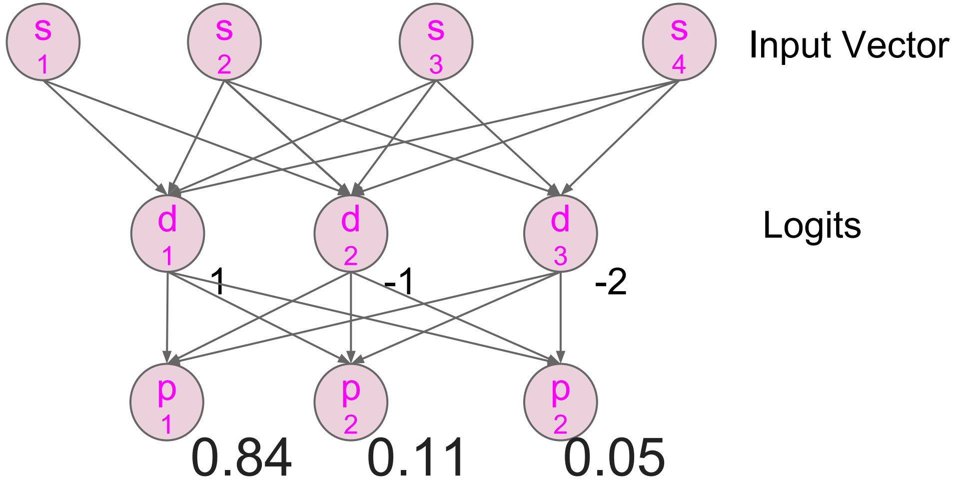


V = 3

	Apple	Banana	Coconut
W_1	0.1	-0.4	0.2
W_2	0.4	1.4	-1.0
W_3	1.1	0.9	1.1
W_4	1.3	0.1	0.4

Using Discrete Variables

Softmax

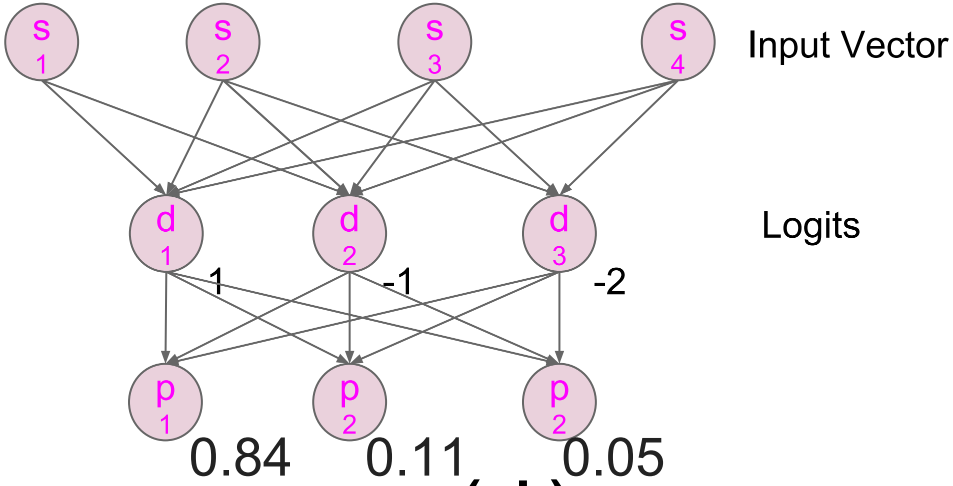


V = 3

	Apple	Banana	Coconut
W_1	0.1	-0.4	0.2
W_2	0.4	1.4	-1.0
W_3	1.1	0.9	1.1
W_4	1.3	0.1	0.4

Using Discrete Variables

Softmax

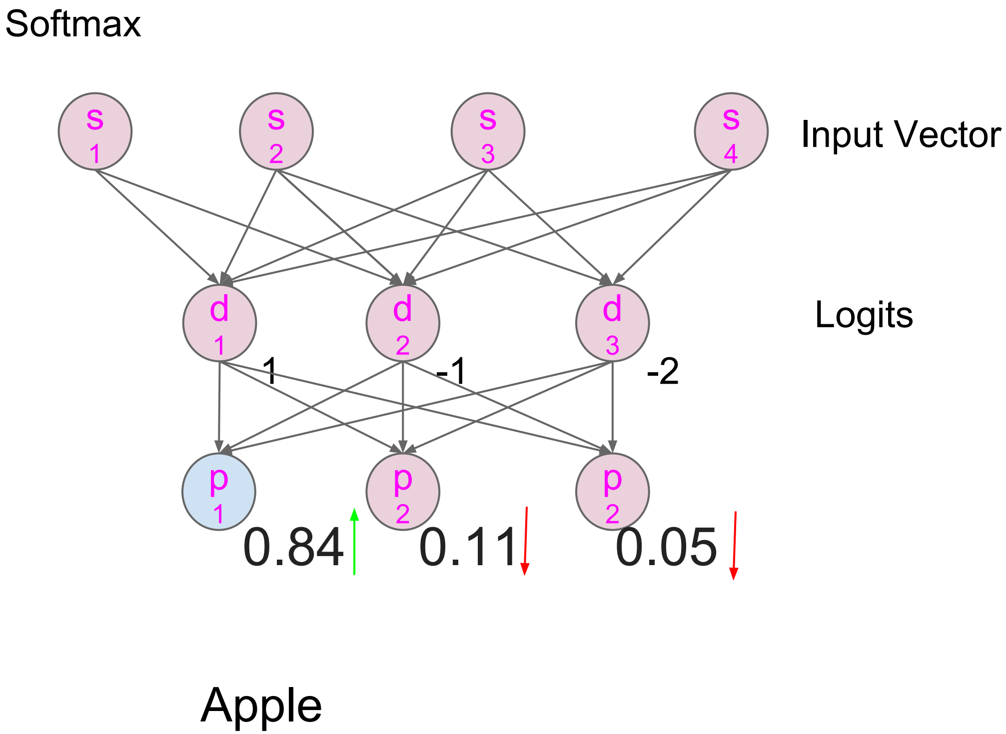


$$p_i = \frac{\exp(d_i)}{\sum \exp(d_i)}$$

V = 3

	Apple	Banana	Coconut
w_1	0.1	-0.4	0.2
w_2	0.4	1.4	-1.0
w_3	1.1	0.9	1.1
w_4	1.3	0.1	0.4

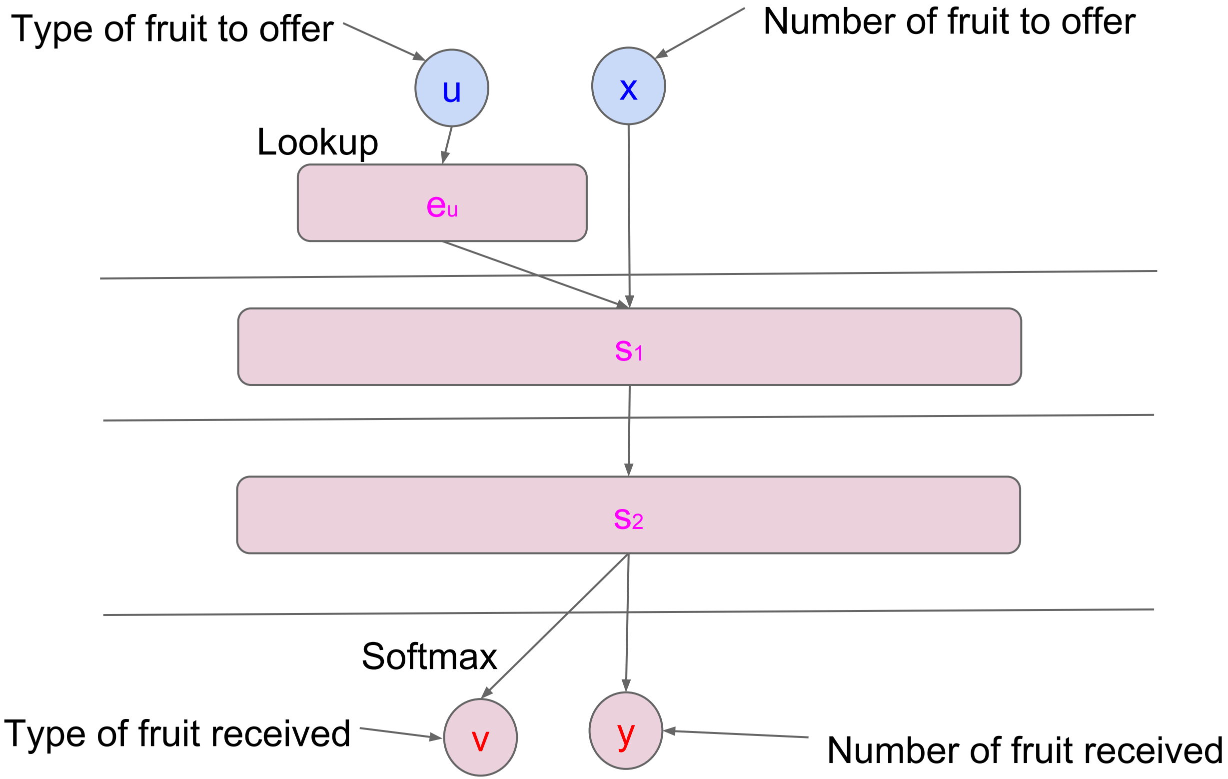
Using Discrete Variables



V = 3

	Apple	Banana	Coconut
W_1	0.1	-0.4	0.2
W_2	0.4	1.4	-1.0
W_3	1.1	0.9	1.1
W_4	1.3	0.1	0.4

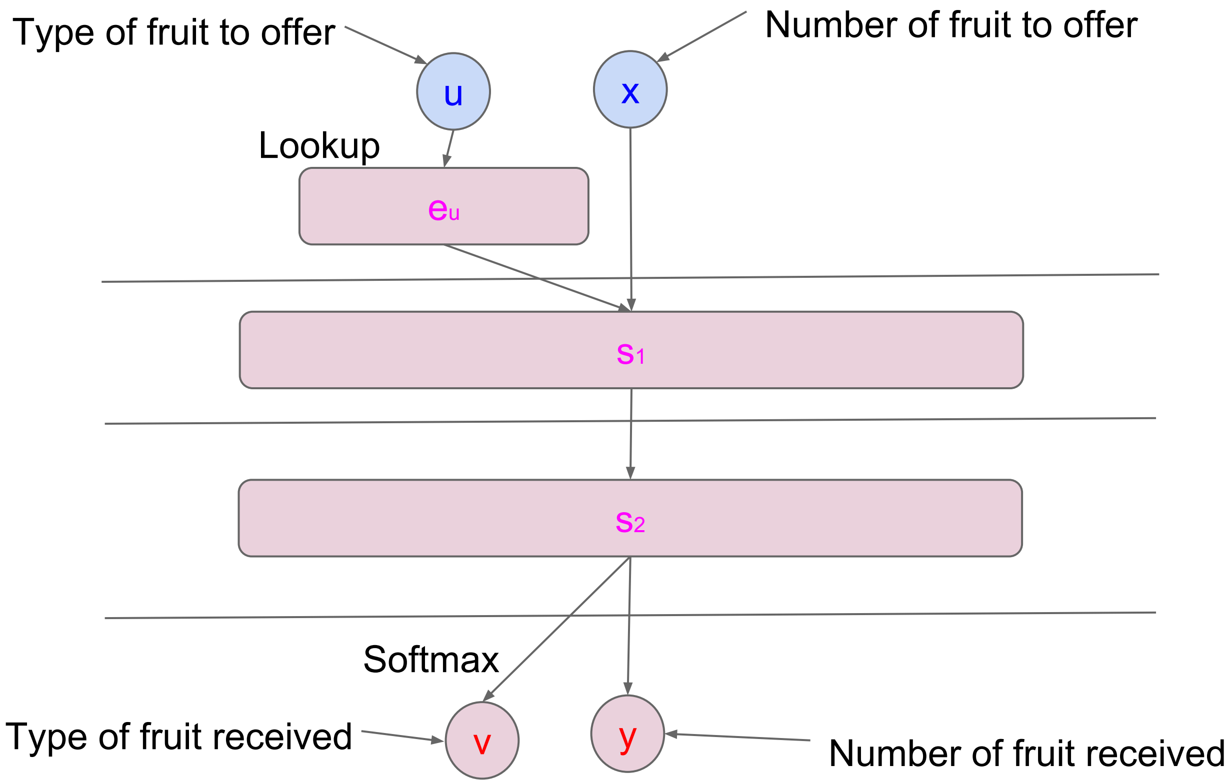
Using Discrete Variables



u∈{Apple, Banana, Coconut}

v∈{Apple, Banana, Coconut}

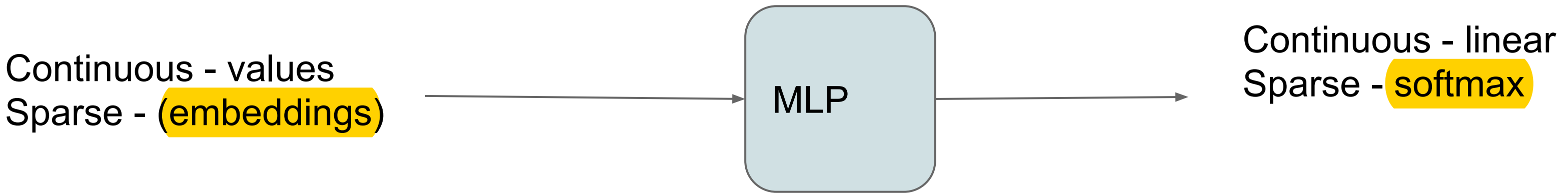
Using Discrete Variables

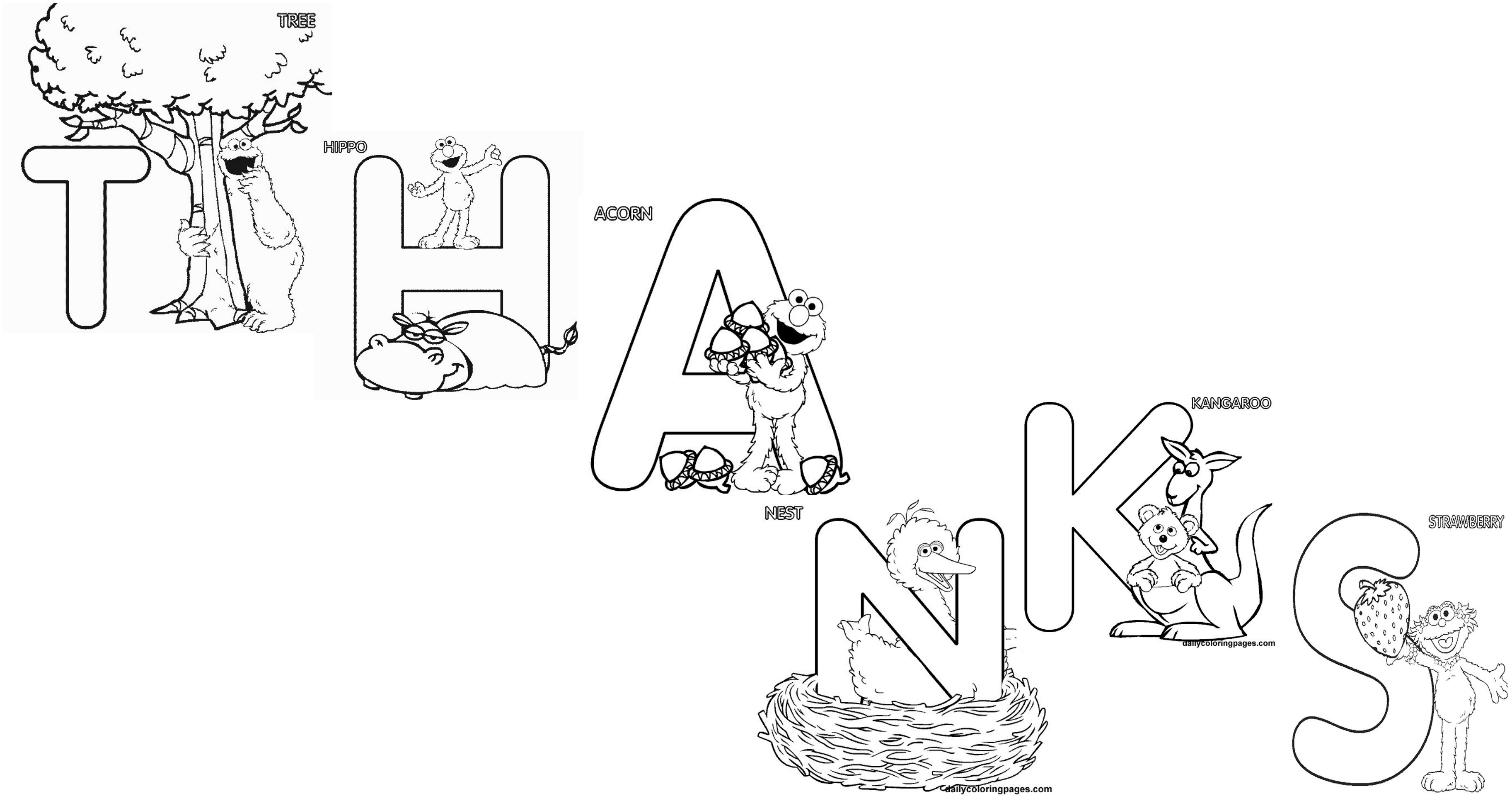


u∈{Apple, Banana, Coconut}

v∈{Apple, Banana, Coconut}

Summary


Continuous - values
Sparse - (embeddings)
MLP
Continuous - linear
Sparse - softmax


TREE
HIPPO
ACORN
NEST
KANGAROO
dailycoloringpages.com
STRAWBERRY
dailycoloringpages.com

Example Applications

Embedding Pretraining (Collobert et al, 2011)

Abby likes to eat apples and bananas

Example Applications

Embedding Pretraining (Collobert et al, 2011)

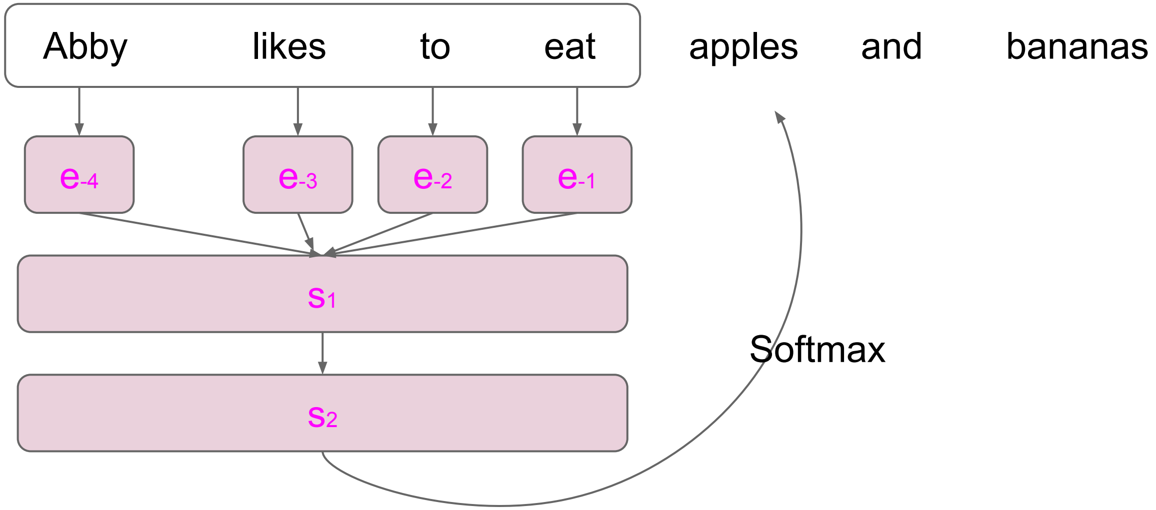
Predict

Context

Abby likes to eat apples and bananas

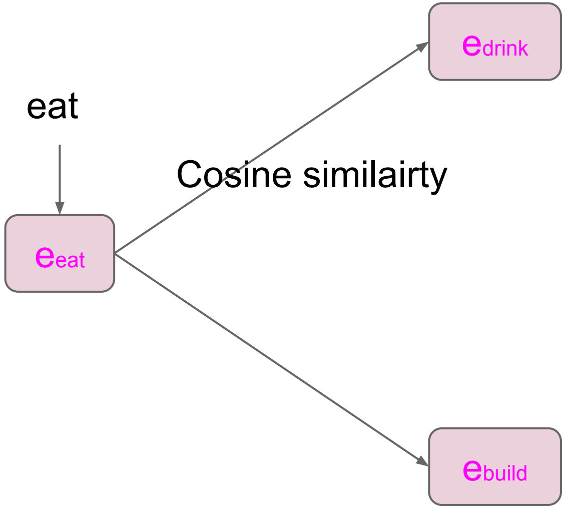
Example Applications

Embedding Pretraining (Collobert et al, 2011)



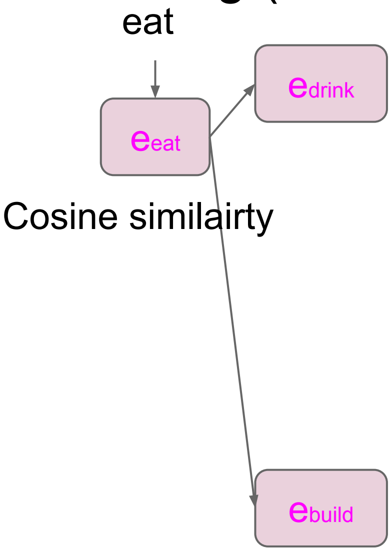
Example Applications

Embedding Pretraining (Collobert et al, 2011)



Example Applications

Embedding Pretraining (Collobert et al, 2011)



Example Applications

FRANCE 454	JESUS 1973	XBOX 6909	REDDISH 11724	SCRATCHED 29869	MEGABITS 87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

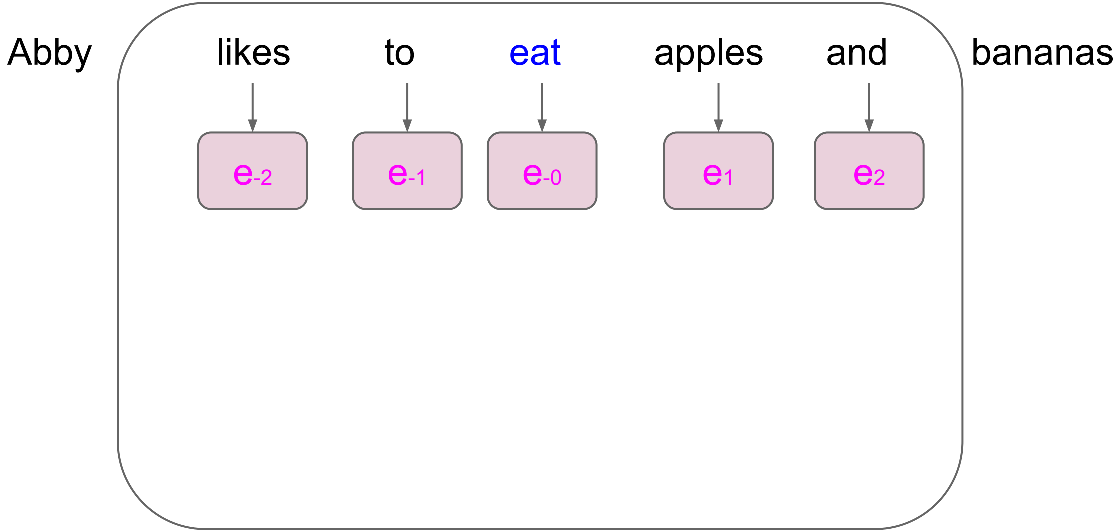
Example Applications

Window-based Tagging (Collobert et al, 2011)

Abby	likes	to	eat	apples	and	bananas
NNP	VBZ	TO	VB	NNS	CC	NNS

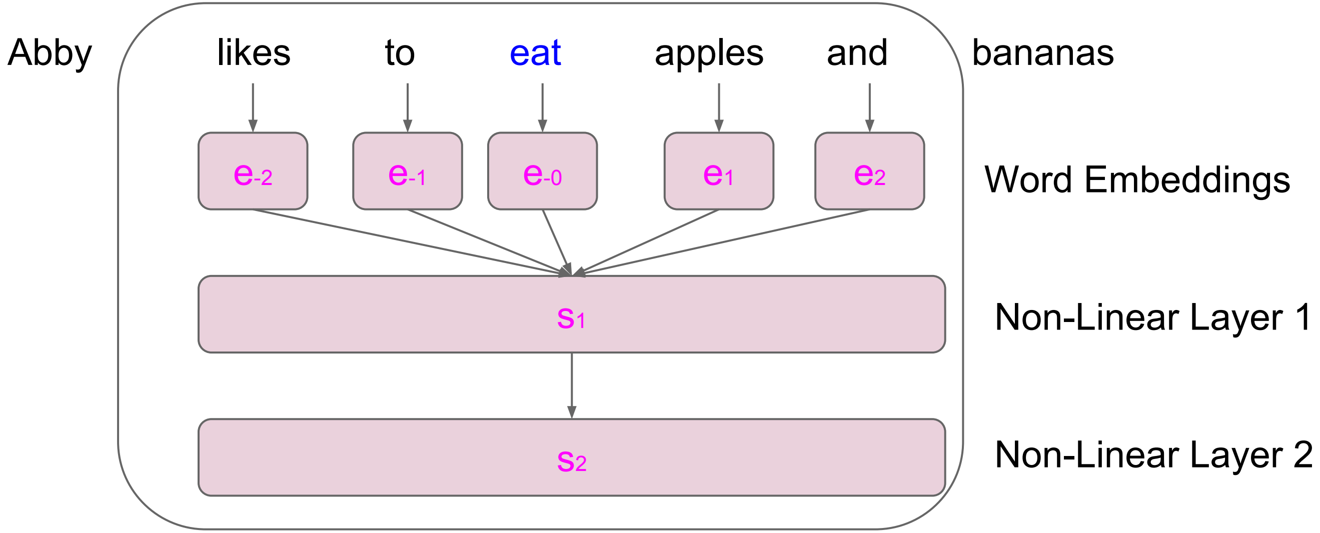
Example Applications

Window-based Tagging (Collobert et al, 2011)



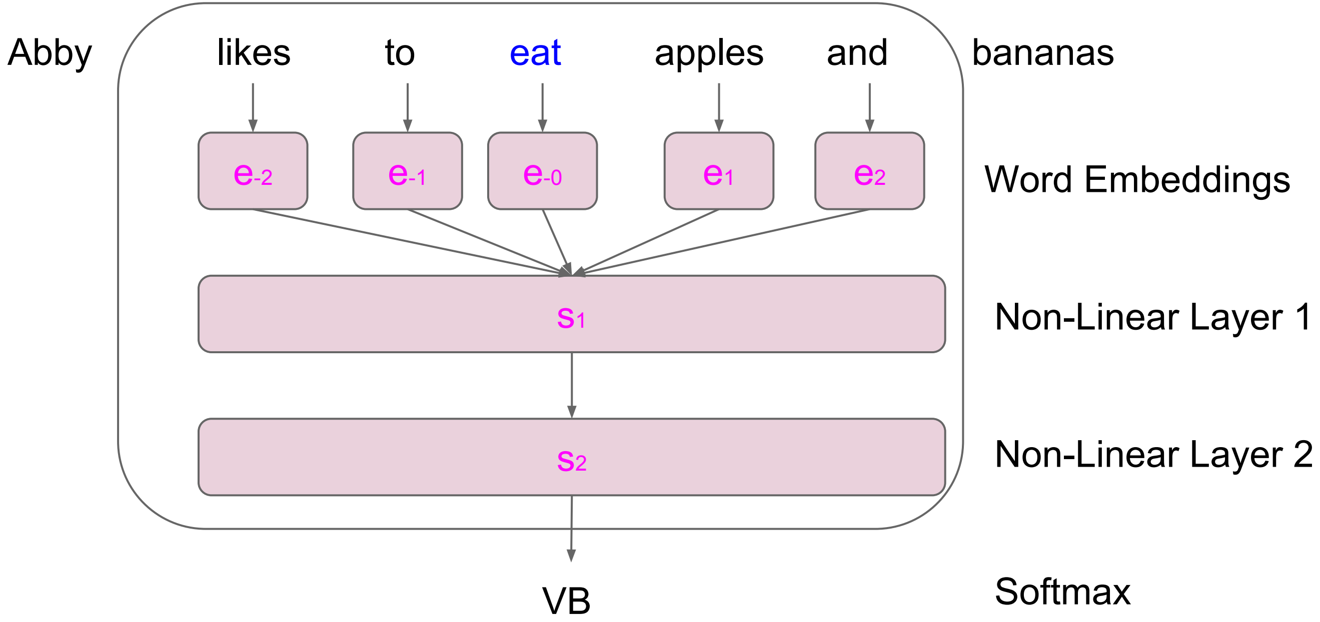
Example Applications

Window-based Tagging (Collobert et al, 2011)



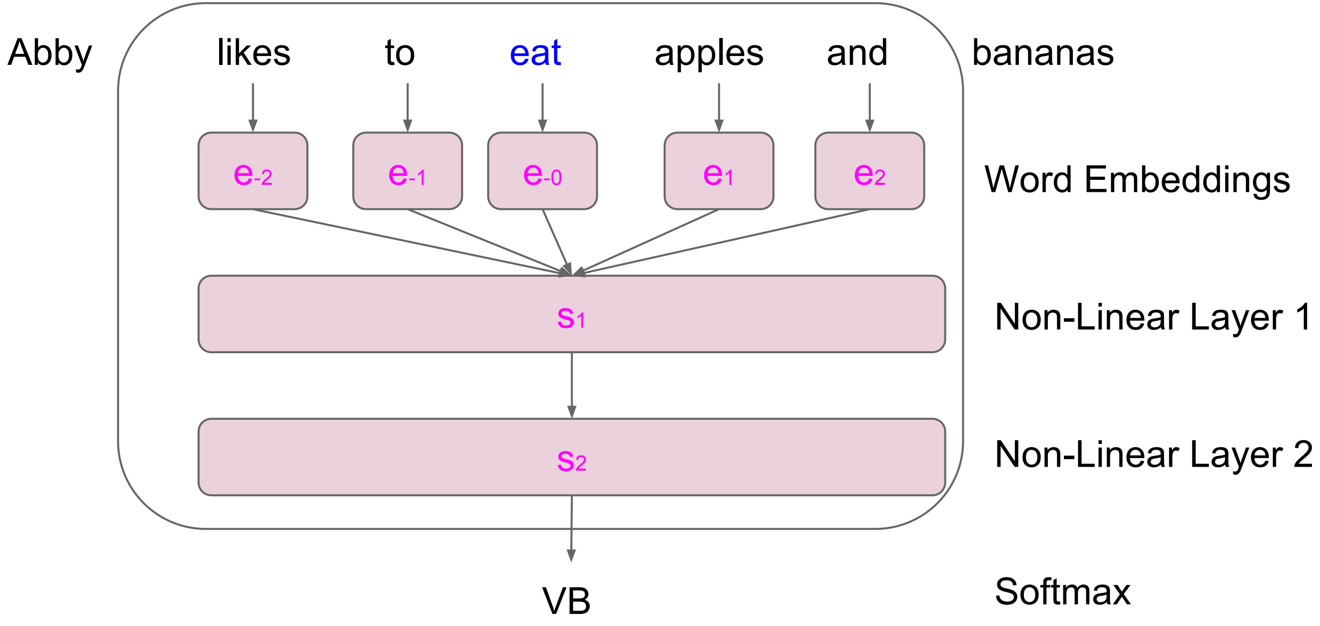
Example Applications

Window-based Tagging (Collobert et al, 2011)



Example Applications

Window-based Tagging (Collobert et al, 2011)



Example Applications

Window-based Tagging (Collobert et al, 2011)

Approach	POS (PWA)	CHUNK (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99
NN+WLL+LM1	97.05	91.91	85.68	58.18
NN+SLL+LM1	97.10	93.65	87.58	73.84
NN+WLL+LM2	97.14	92.04	86.96	58.34
NN+SLL+LM2	97.20	93.63	88.67	74.15

Example Applications

Translation Rescoring (Devlin et al, 2014)

Translation 1	John	does	to	eat	coconuts	and	bananas
Translation 2	Abby	likes	to	eat	apples	and	bananas
Translation 3	Abby	dislikes	to	drink	apples	and	bananas
Source	Abby	gosta	de	comer	macas	e	bananas

Example Applications

Translation Rescoring (Devlin et al, 2014)

<s> Abby likes to eat apples and bananas

0.2

Example Applications

Translation Rescoring (Devlin et al, 2014)

Abby likes to eat apples and bananas

0.2 0.1

Example Applications

Translation Rescoring (Devlin et al, 2014)

Abby likes to eat apples and bananas

0.2 0.1 0.3

Example Applications

Translation Rescoring (Devlin et al, 2014)

Abby	likes	to	eat	apples	and	bananas	0.000378
0.2	0.1	0.3	0.5	0.7	0.4	0.2	

Example Applications

Translation Rescoring (Devlin et al, 2014)

John does to eat coconuts and bananas	0.00003
Abby likes to eat apples and bananas	0.000378
Abby dislikes to drink apples and bananas	0.00012

Example Applications

Translation Rescoring (Devlin et al, 2014)

John	does	to	eat	coconuts	and	bananas	0.00003
Abby	likes	to	eat	apples	and	bananas	0.000378
Abby	dislikes	to	drink	apples	and	bananas	0.00012

Example Applications

Translation Rescoring (Devlin et al, 2014)

Predict

Context

Translation

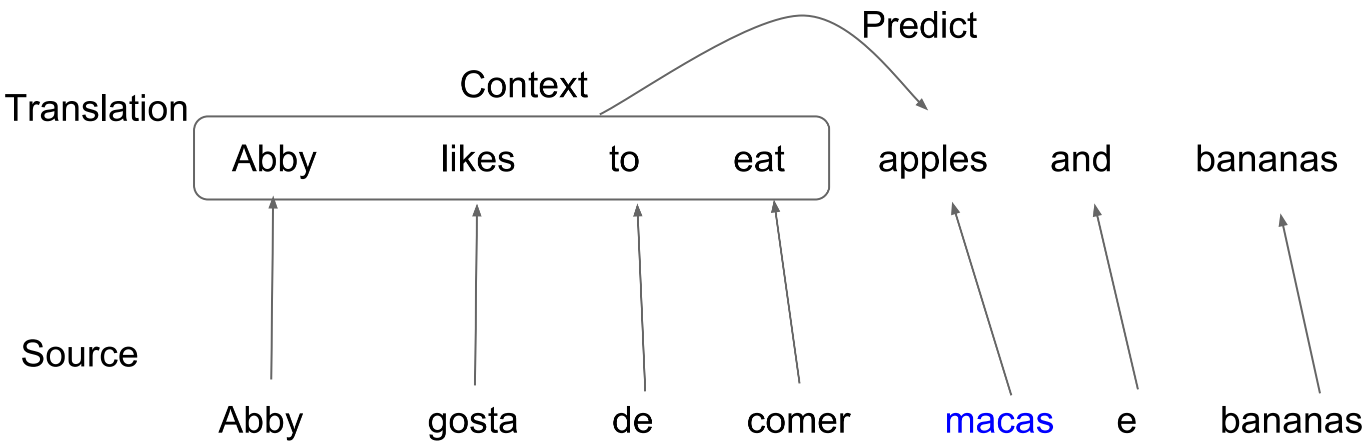
Abby likes to eat apples and bananas

Source

Abby gosta de comer macas e bananas

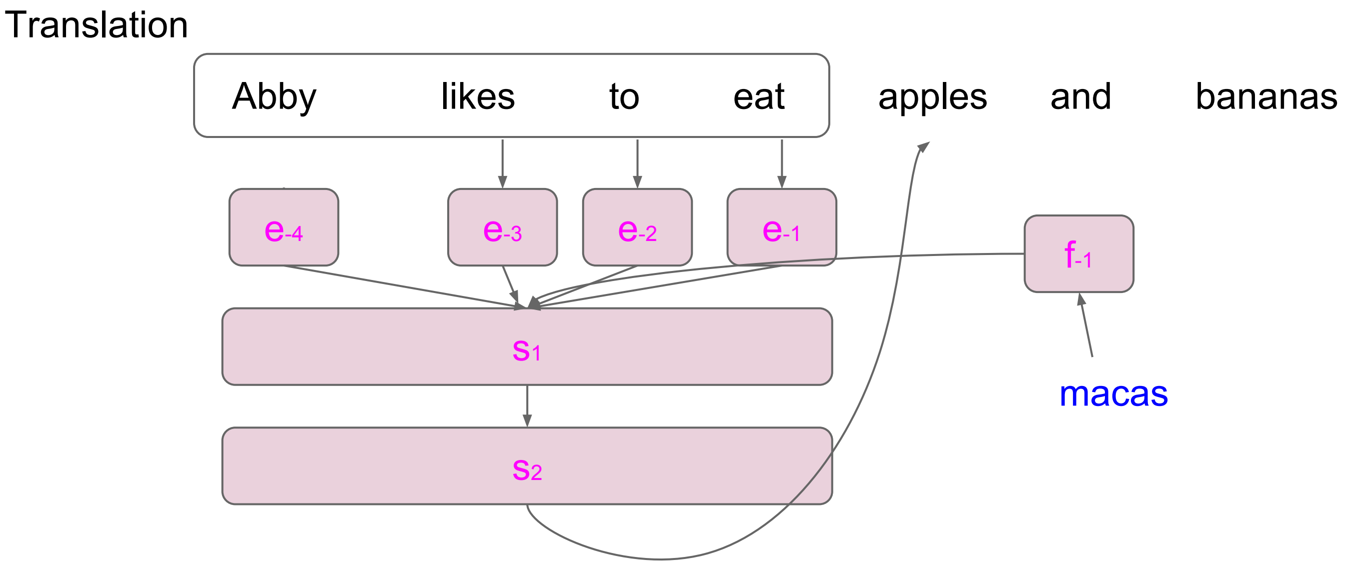
Example Applications

Translation Rescoring (Devlin et al, 2014)



Example Applications

Translation Rescoring (Devlin et al, 2014)



Example Applications

Translation Rescoring (Devlin et al, 2014)

Translation Score (BLEU)	Arabic - English	Chinese - English
Best Rescored System	52.8	34.7
1st OpenMT12	49.5	32.6
Hierarchical	43.4	30.1

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2 \qquad y = wx + b$$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (\hat{y}_n - y_n)^2}{\partial w} = \sum_n -2(\hat{y}_n - y_n)x_n$$

$$\frac{\partial C}{\partial b} = \frac{\partial \sum_n (\hat{y}_n - y_n)^2}{\partial b} = \sum_n -2(\hat{y}_n - y_n)$$

Easy!

Computation Graphs are our friends

$$y = wx + b + \tanh(yx + b)^2$$

Harder!

Computation Graphs are our friends

$$y = w_1x + b_1 + \tanh(w_2x + b_2^2)$$

Computation Graphs can compute gradients for you!

Computation Graphs are our friends

$$C(w,b) = \sum_{n\in\{0,1,2\}} (y_n - \hat{y}_n)^2 \qquad y = wx + b$$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (\hat{y}_n - y_n)^2}{\partial w} = \sum_n -2(\hat{y}_n - y_n)x_n$$

$$\frac{\partial C}{\partial b} = \frac{\partial \sum_n (\hat{y}_n - y_n)^2}{\partial b} = \sum_n -2(\hat{y}_n - y_n)$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2 \qquad y = wx + b$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial w} = \sum_n -2(\hat{y}_n - y_n)x_n$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial b} = \sum_n -2(\hat{y}_n - y_n)$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} (y_n - \hat{y}_n)^2 \qquad y = wx + b$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial b}$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n\in\{0,1,2\}} (y_n - \hat{y}_n)^2$$

$$y = o + b$$

$$o = wx$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial(\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial(\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial b}$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} (d_n)^2$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial b}$$

$$d = y - \hat{y}$$

$$y = o + b$$

$$o = wx$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial (\hat{y}_n - y_n)^2}{\partial y_n} \frac{\partial y_n}{\partial b}$$

$$c = d^2$$

$$d = y - \hat{y}$$

$$y = o + b$$

$$o = wx$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n\in\{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial o_n}\frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial(\hat{y}_n - y_n)^2}{\partial y_n}\frac{\partial y_n}{\partial b}$$

$$c = d^2$$

$$d = y - \hat{y}$$

$$y = o + b$$

$$o = wx$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

$$c = d^2$$

$$d = y - \hat{y}$$

$$y = o + b$$

$$o = wx$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n\in\{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial o_n}\frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial b}$$

$c = d^2$ Power 2

$d = y - \hat{y}$ Sub

$y = o + b$ Add

$o = wx$ Product

Sub

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

$c = d^2$ — Power 2

$d = y - \hat{y}$ — Sub

$y = o + b$ — Add

$o = wx$ — Product

Sub

forward(x,y) → z
backward(x,y,dz) → dx,dy

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

$c = d^2$ Power 2

$d = y - \hat{y}$ Sub

$y = o + b$ Add

$o = wx$ Product

Sub

forward(x,y) : return x - y
backward(x,y,dz) : return 1, -1

Computation Graphs are our friends

$$C(w,b) = \sum_{n\in\{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial o_n}\frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial b}$$

$c = d^2$ Power 2

$d = y - \hat{y}$ Sub

$y = o + b$ Add

$o = wx$ Product

Sub

forward(x,y) : return x - y
backward(x,y,dz) : return 1, -1

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

$$c = d^2$$ Power 2

$$d = y - \hat{y}$$ Sub

$$y = o + b$$ Add

$$o = wx$$ Product

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

Sub

forward(x,y) : return x - y
backward(x,y,dz) : return 1, -1

$$\frac{\partial d_n}{\partial \hat{y}_n}$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

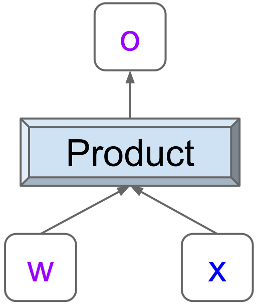
$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

$c = d^2$ Power 2

$d = y - \hat{y}$ Sub

$y = o + b$ Add



$o = wx$

Computation Graphs are our friends

$$C(w,b) = \sum_{n\in\{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial o_n}\frac{\partial o_n}{\partial w}$$

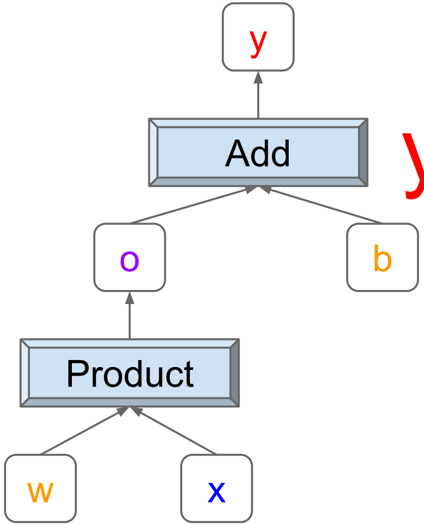
$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n}\frac{\partial d_n}{\partial y_n}\frac{\partial y_n}{\partial b}$$

$$c = d^2$$

Power 2

$$d = y - \hat{y}$$

Sub



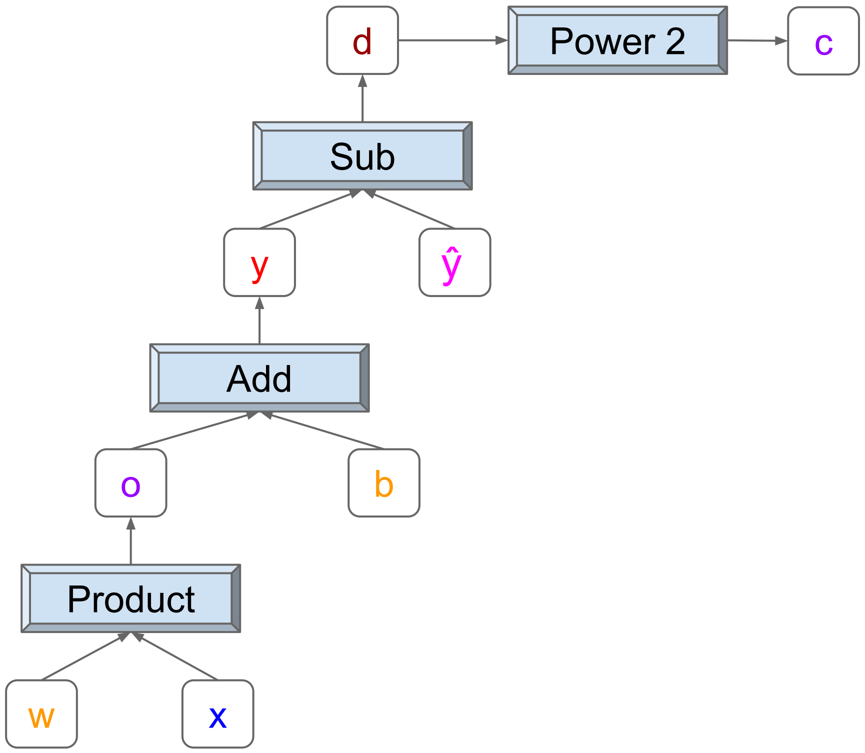
$$y = o + b$$

Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0,1,2\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

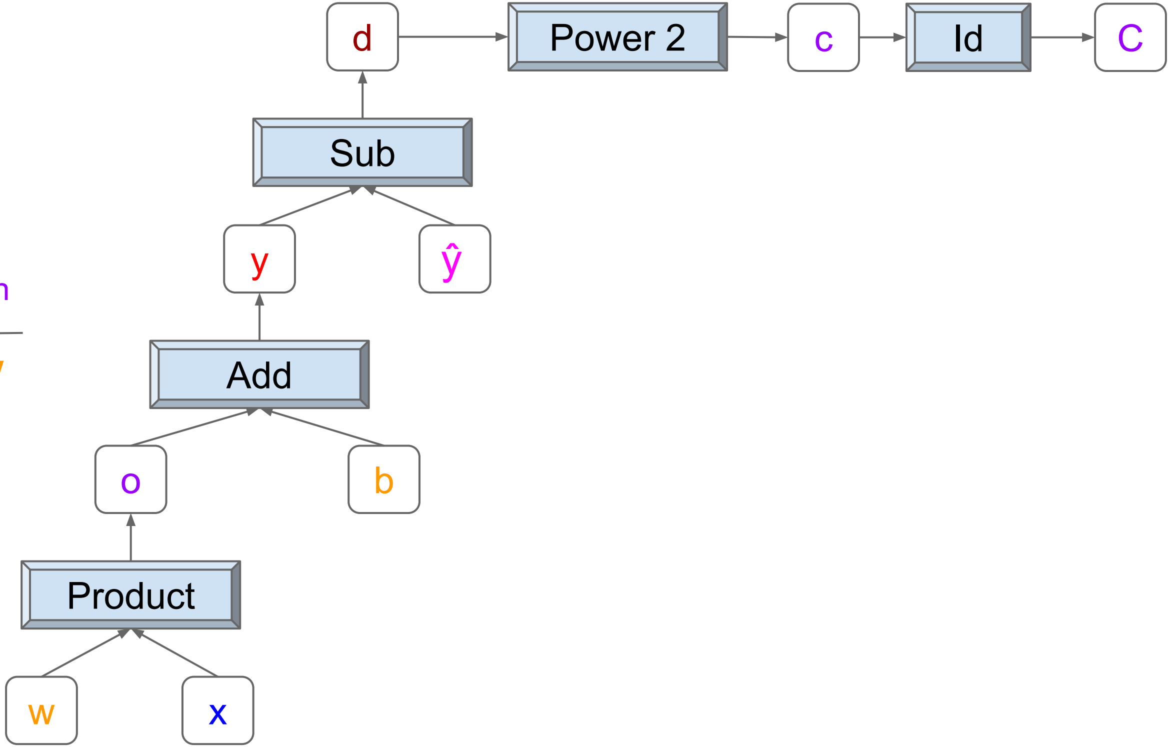


Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

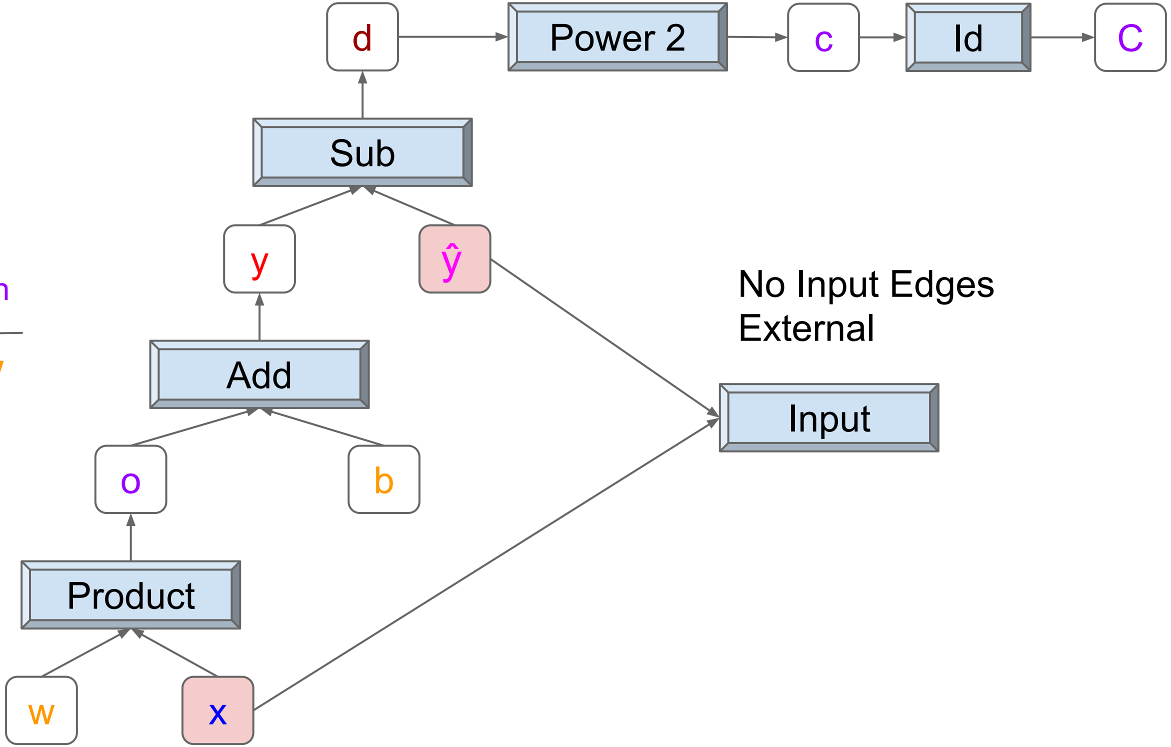


Computation Graphs are our friends

$$C(w,b) = \sum_{n \in \{0\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

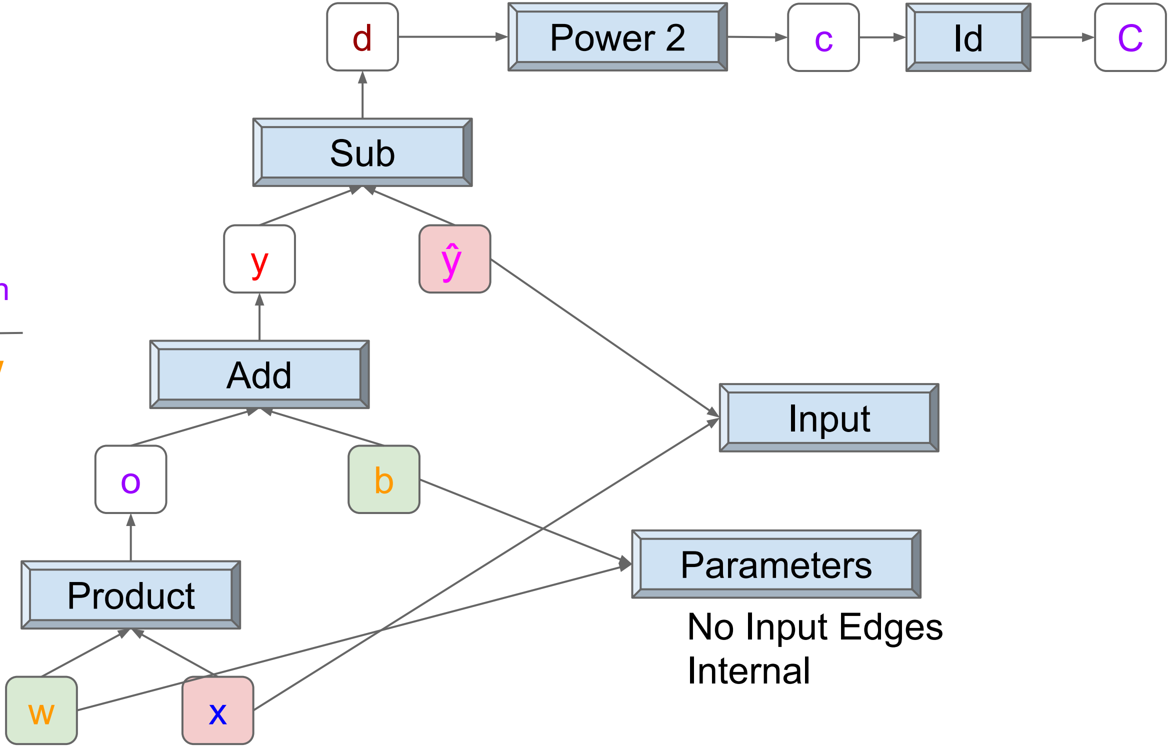


Computation Graphs are our friends

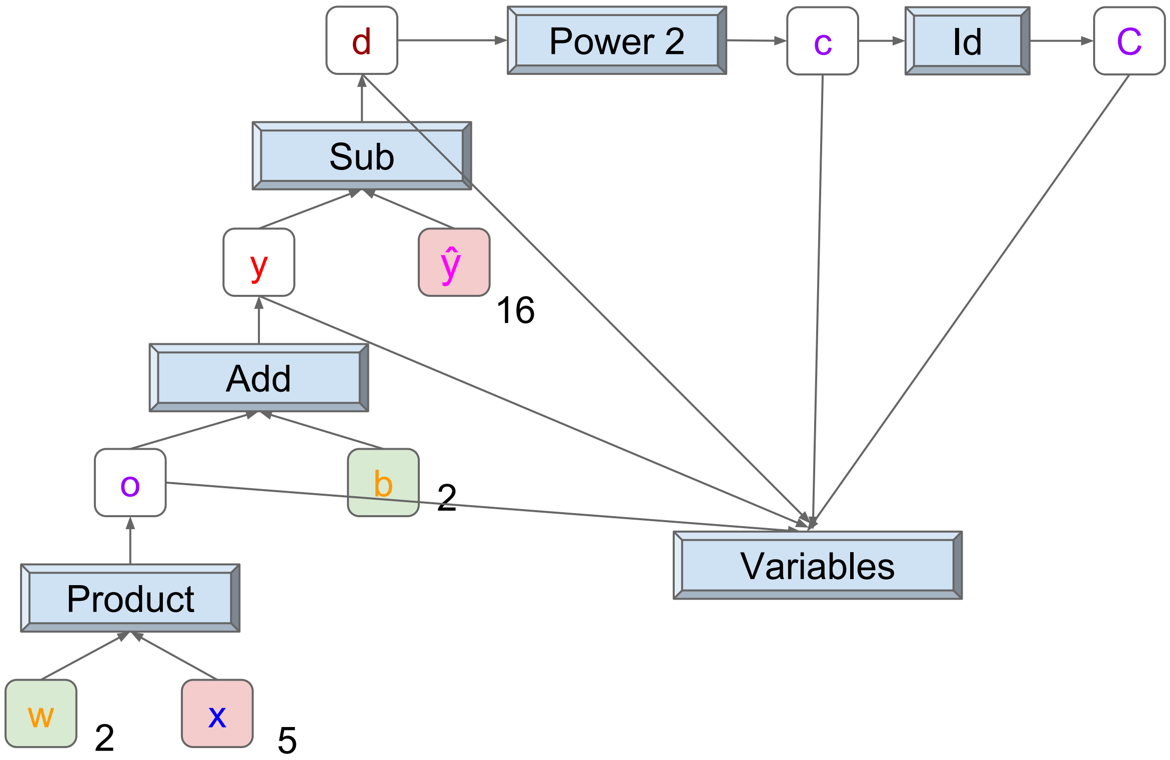
$$C(w,b) = \sum_{n \in \{0\}} c_n$$

$$\frac{\partial C}{\partial w} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial o_n} \frac{\partial o_n}{\partial w}$$

$$\frac{\partial C}{\partial b} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial y_n} \frac{\partial y_n}{\partial b}$$

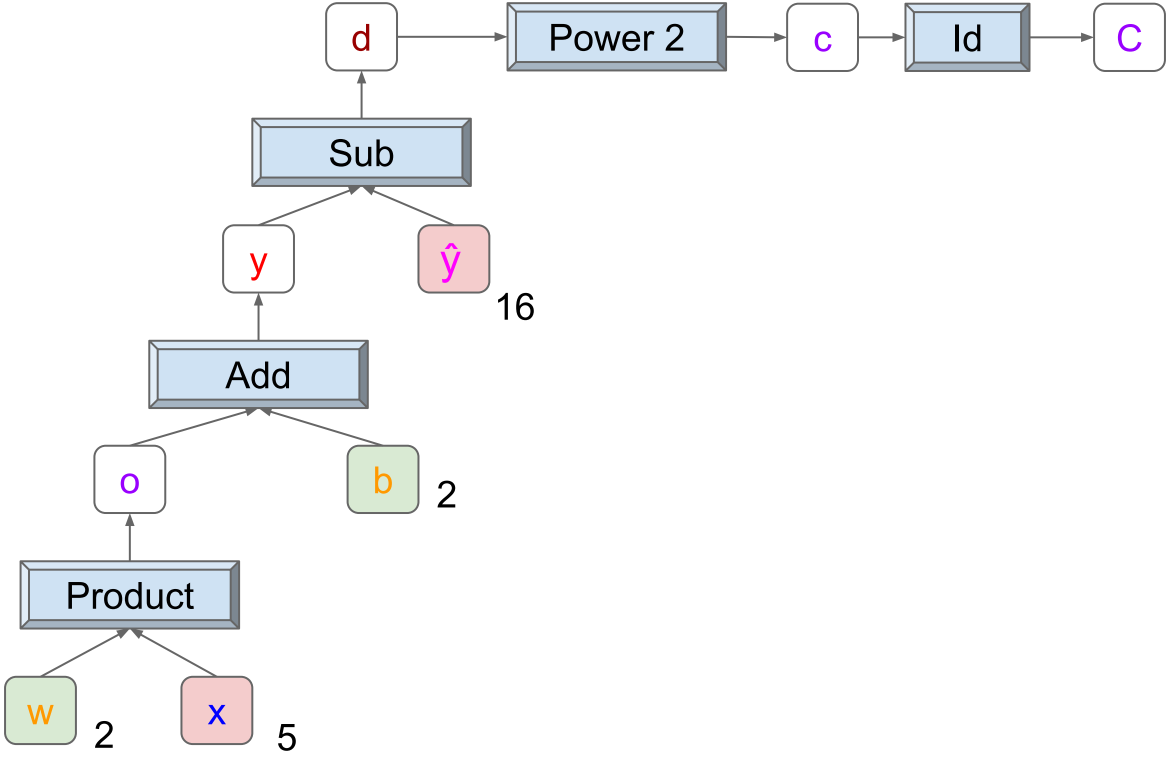


Computation Graphs are our friends



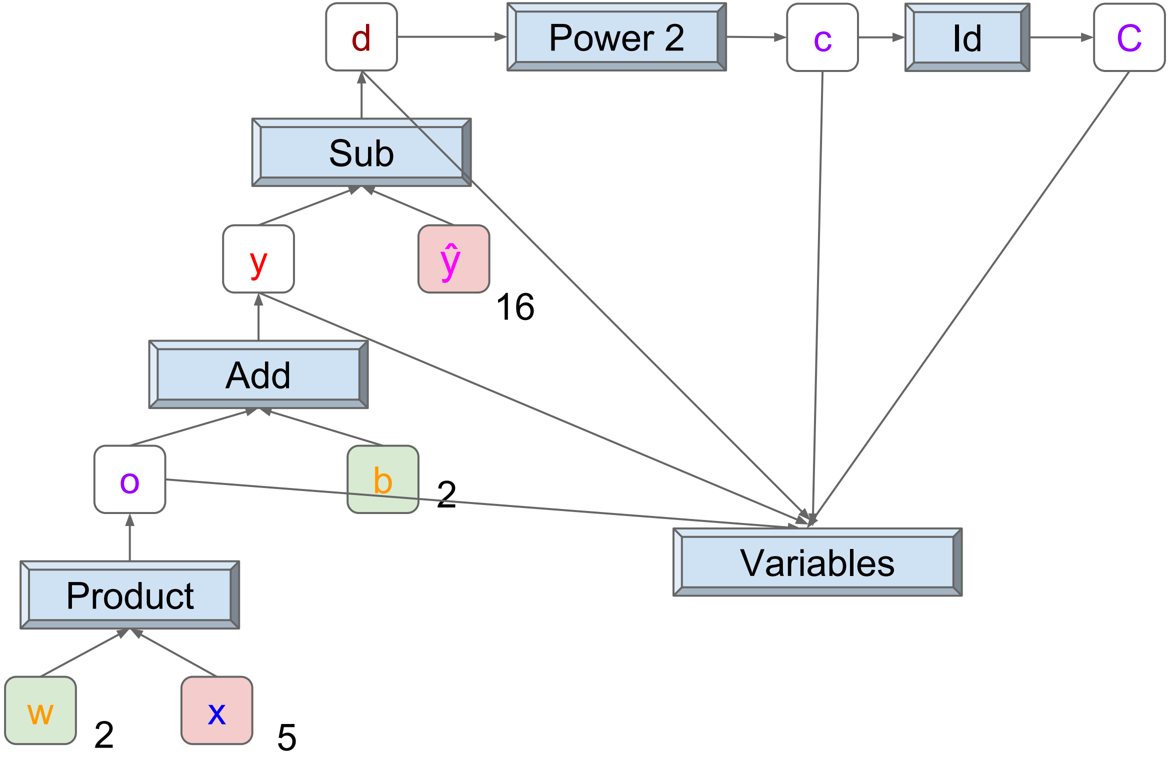
Computation Graphs are our friends

1-Initialize inputs



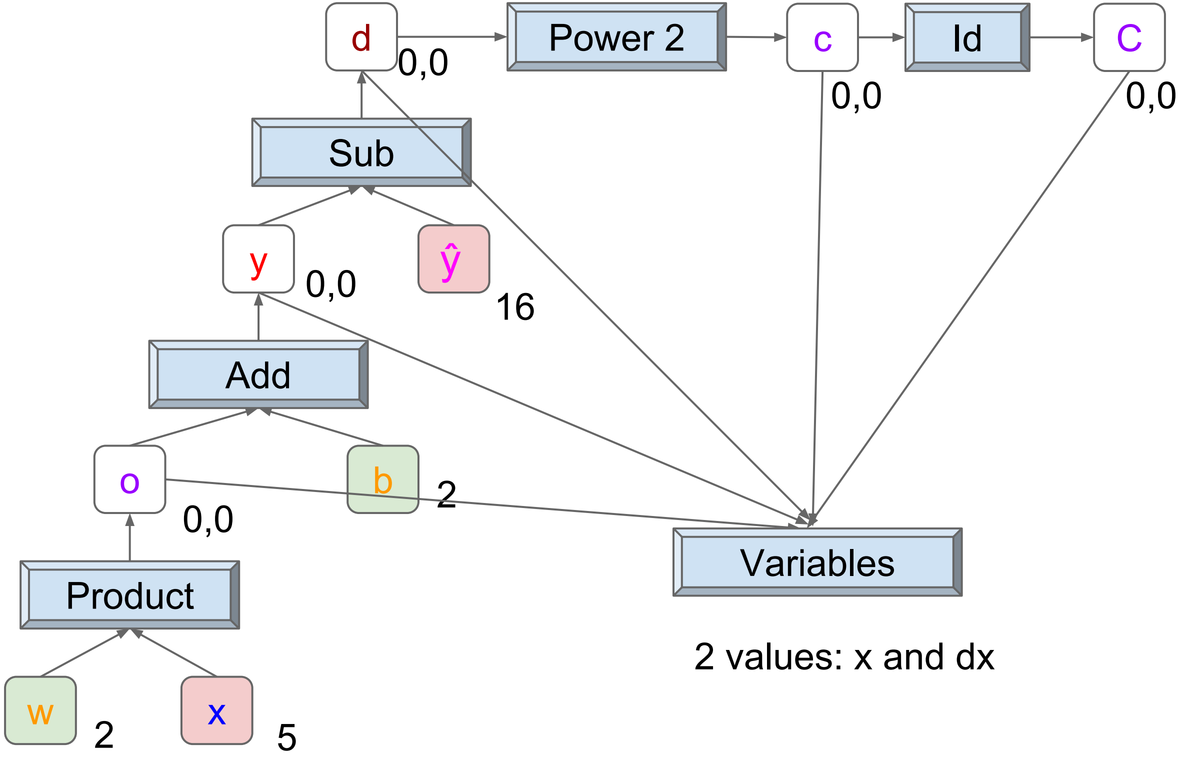
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables



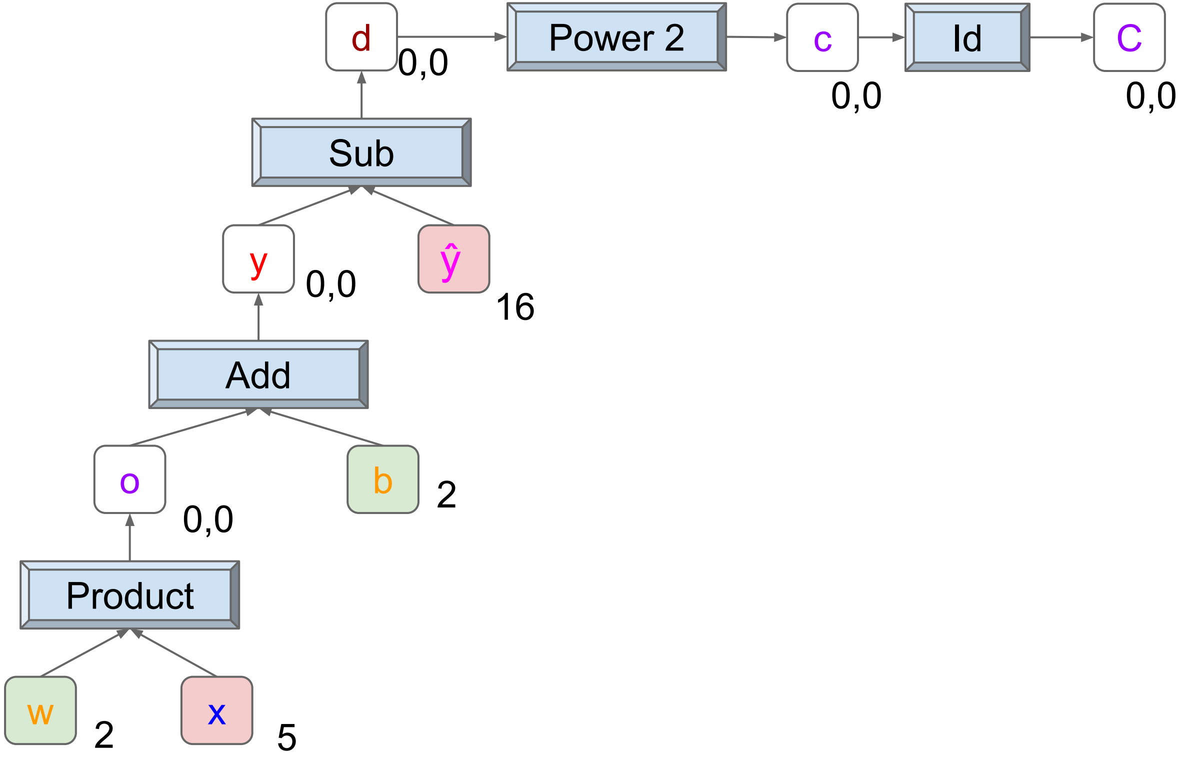
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables



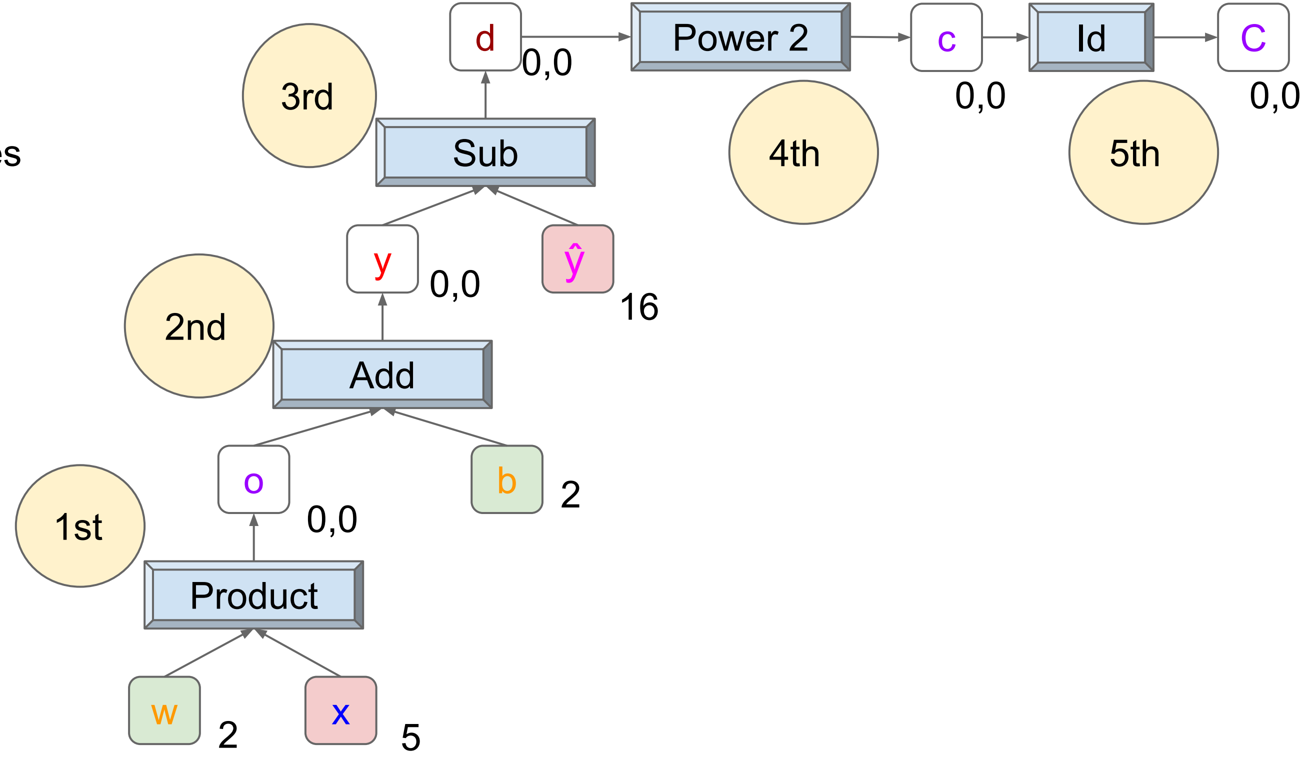
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



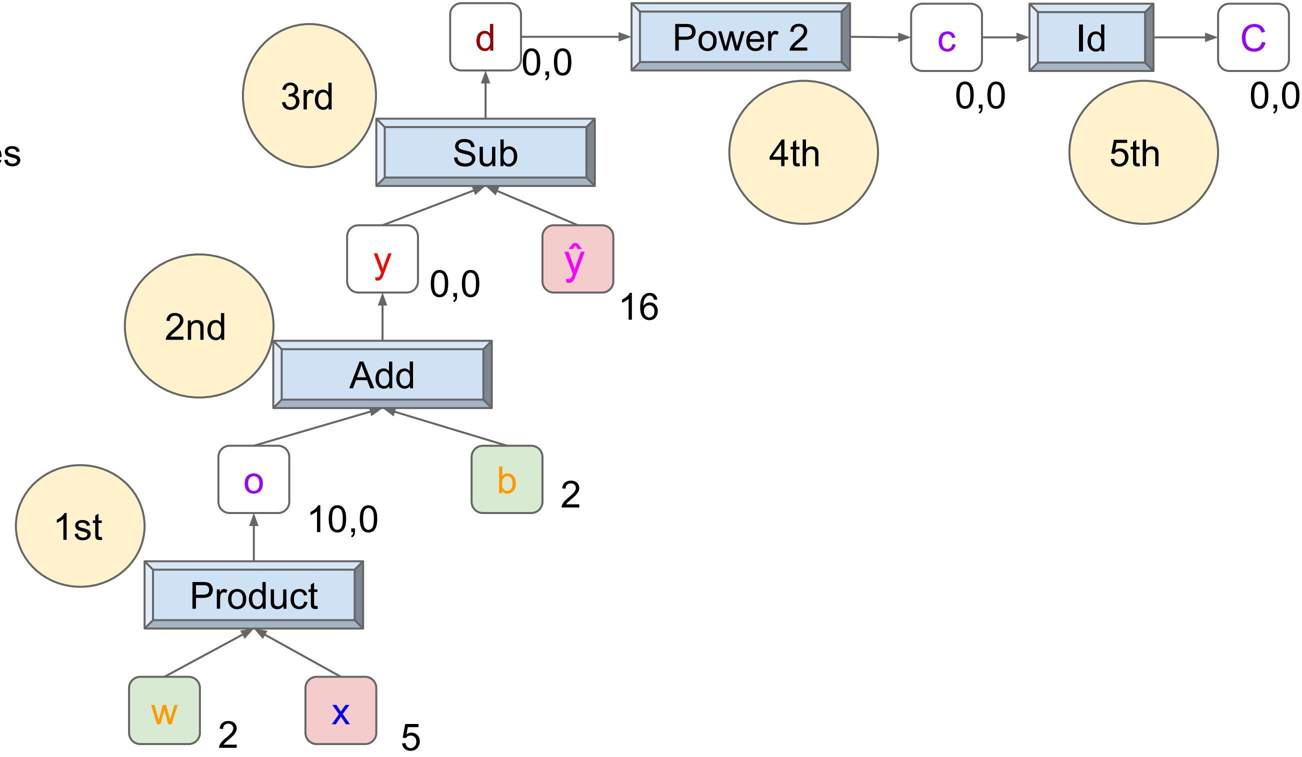
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



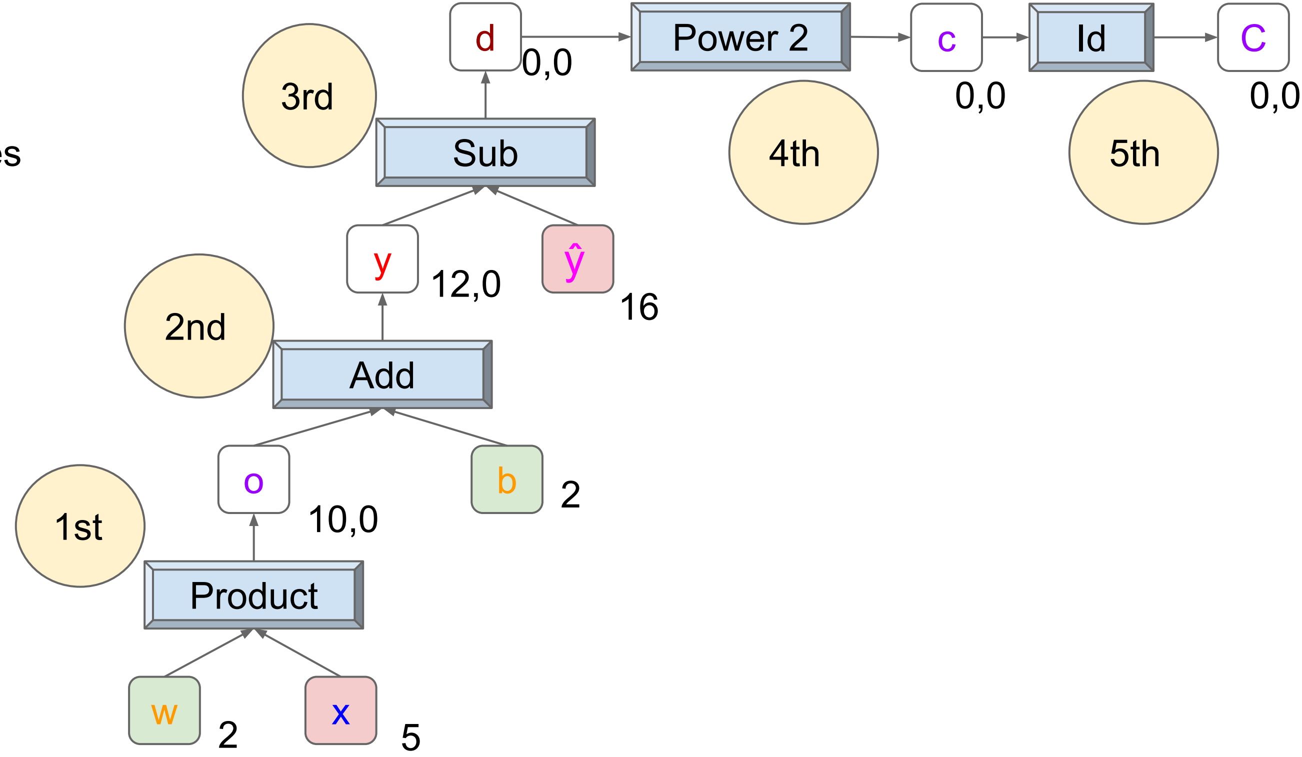
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



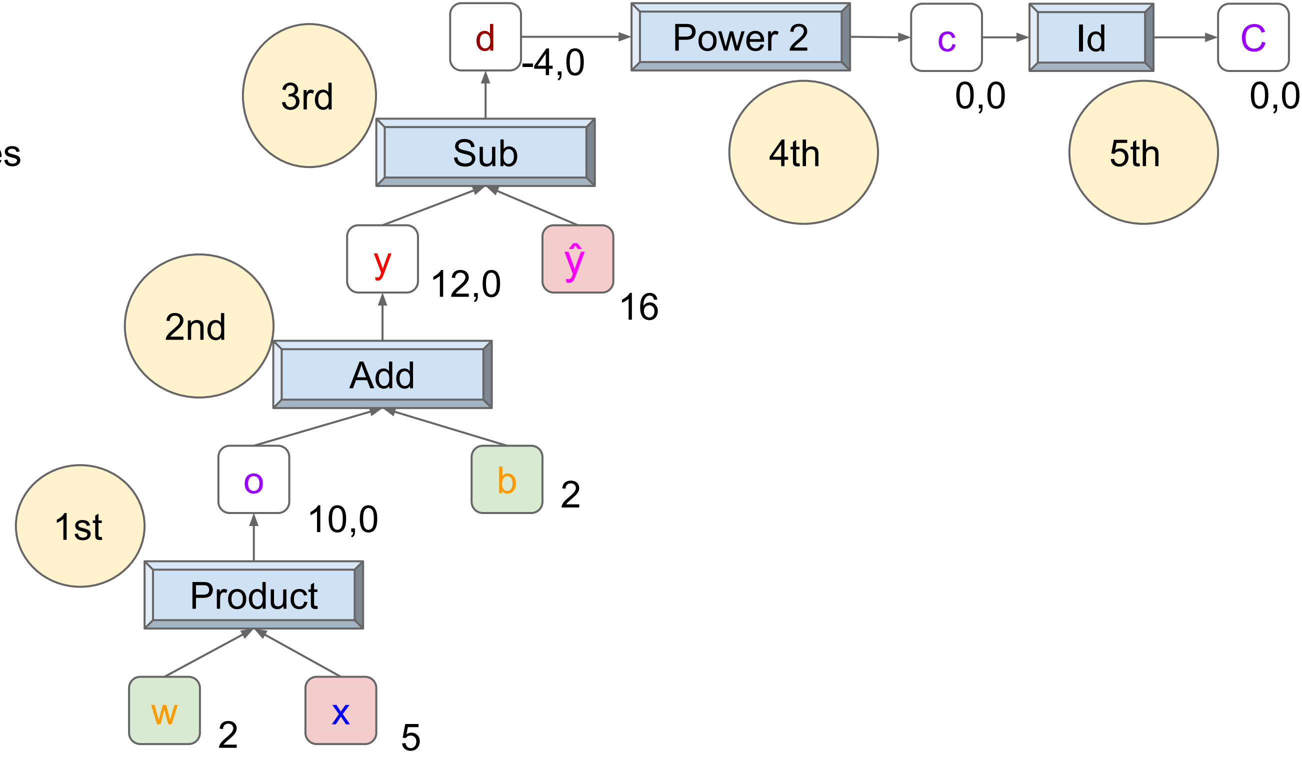
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



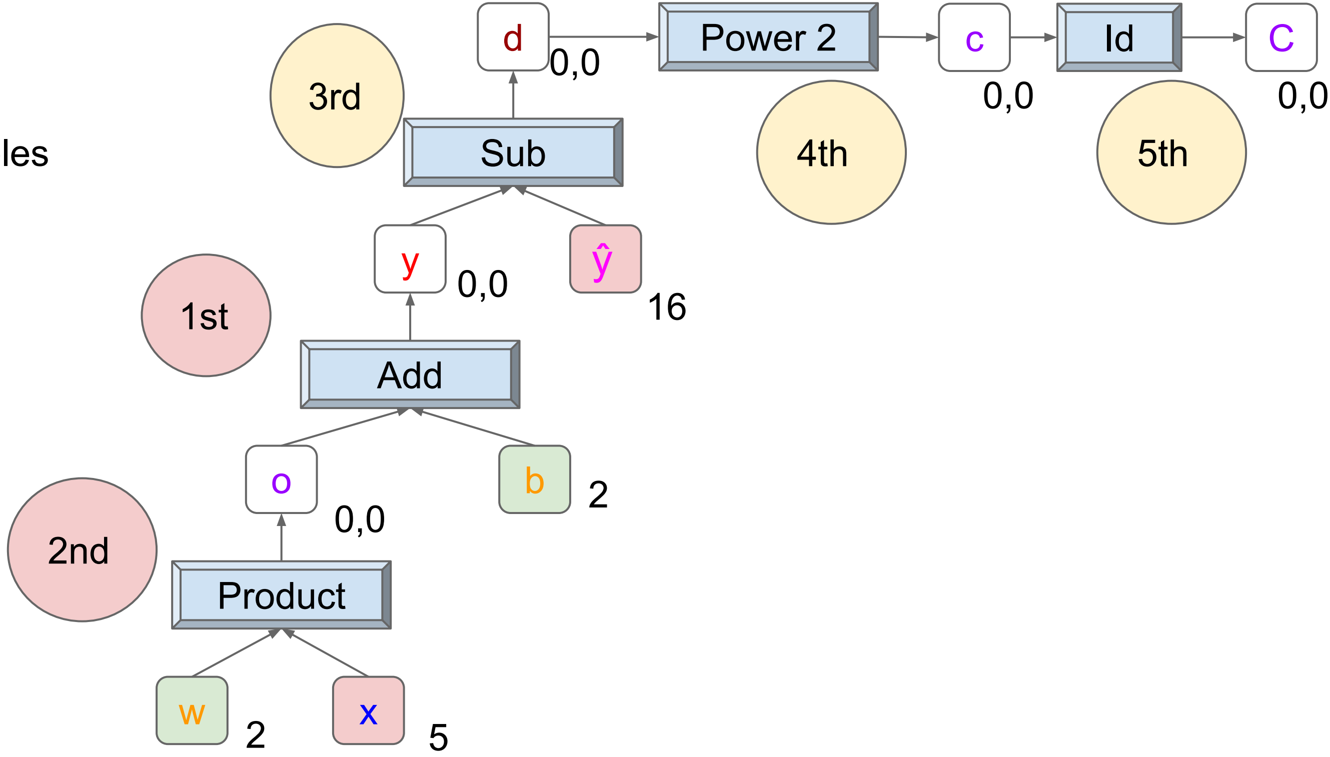
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



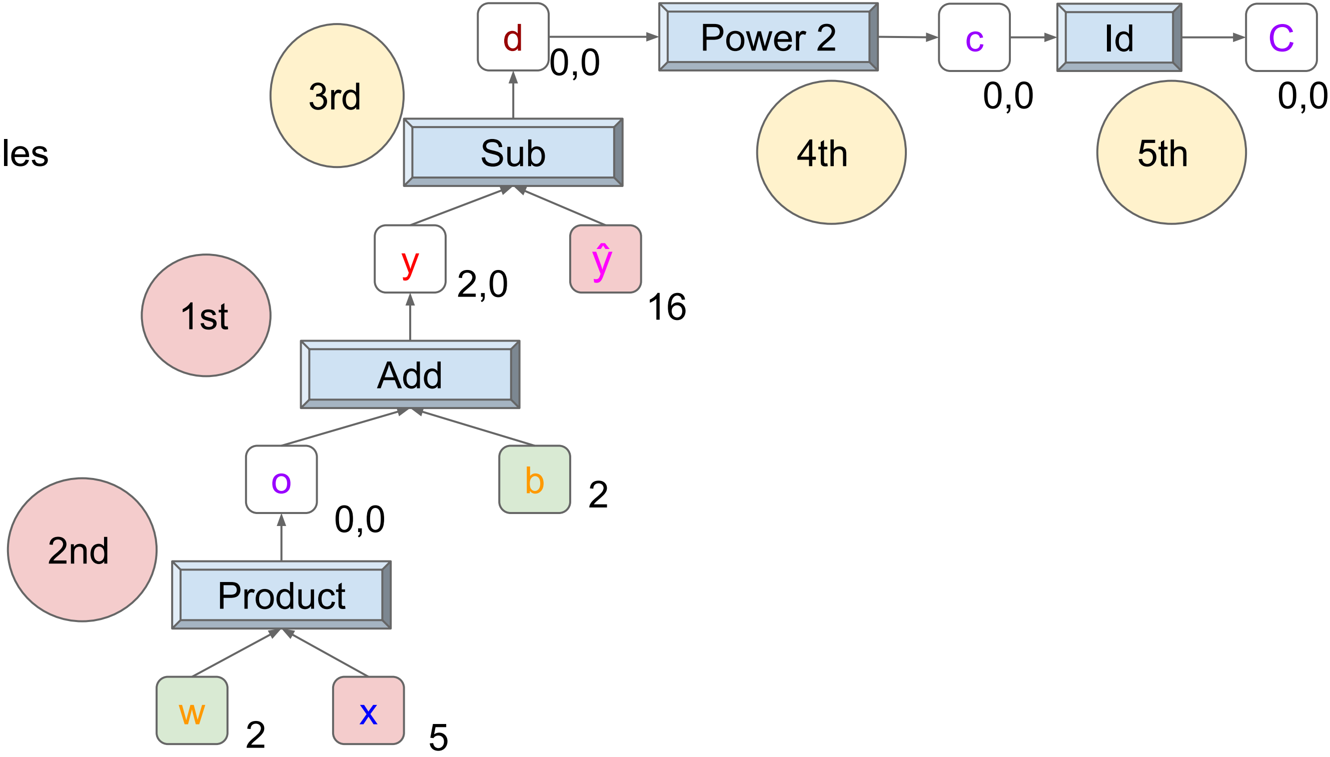
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



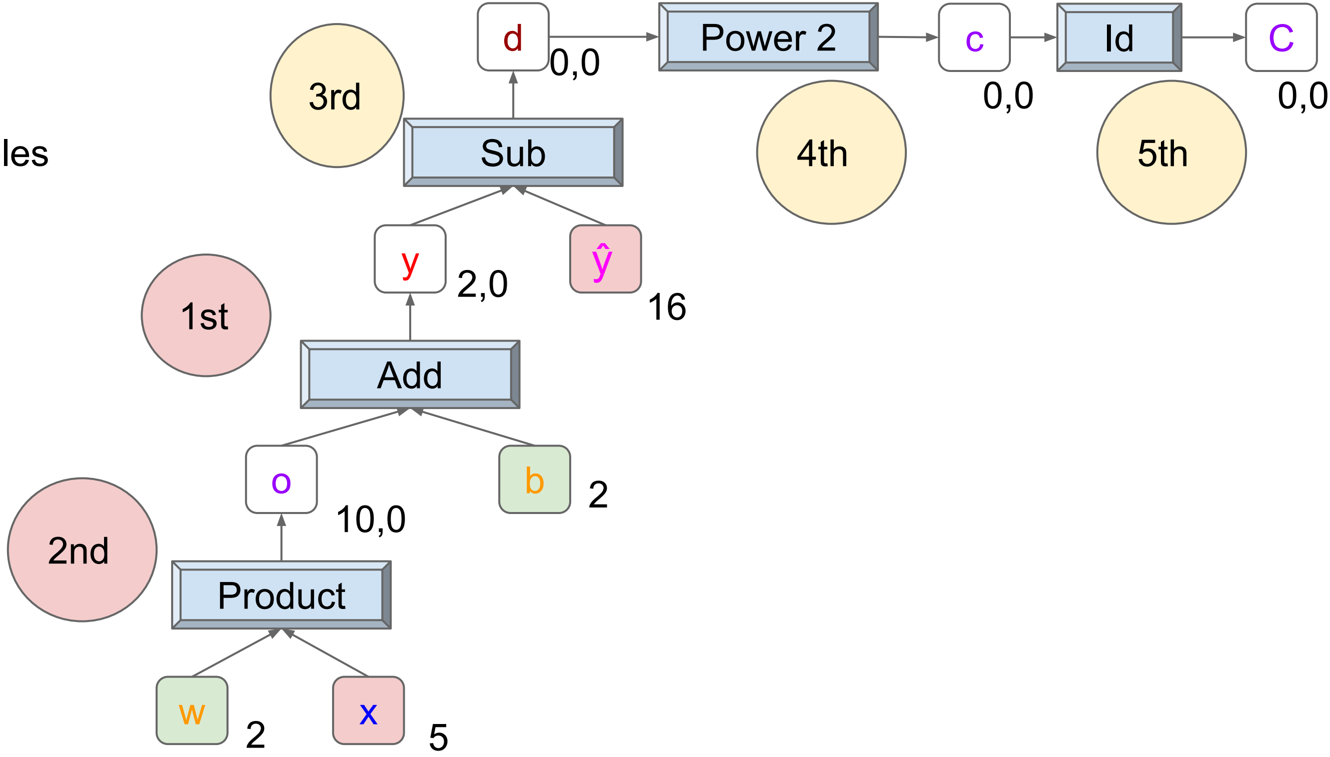
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



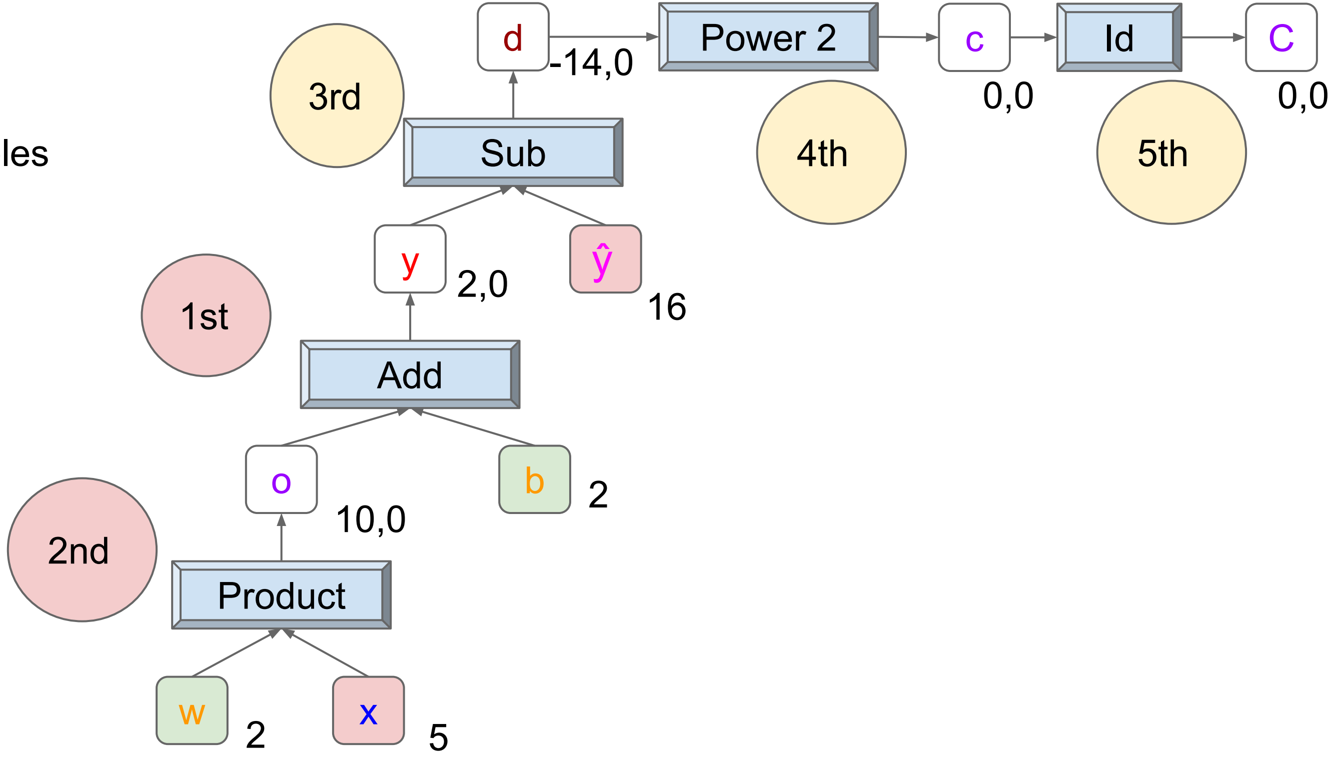
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



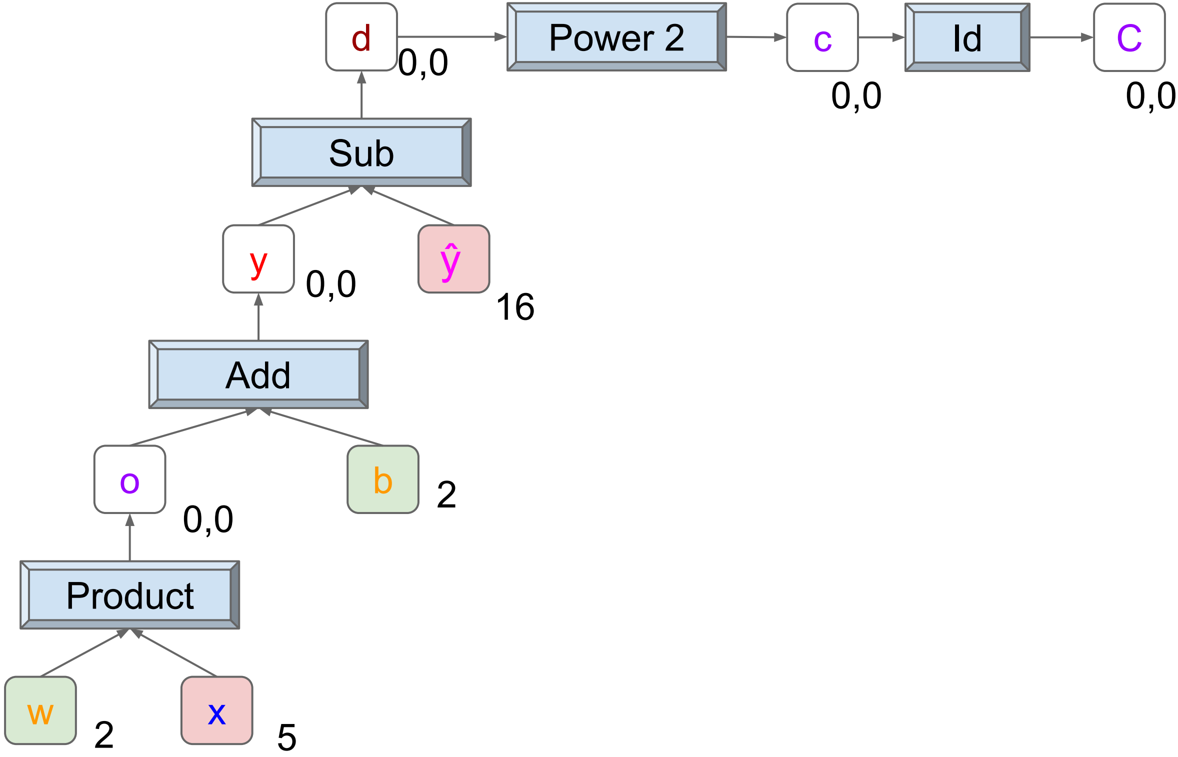
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



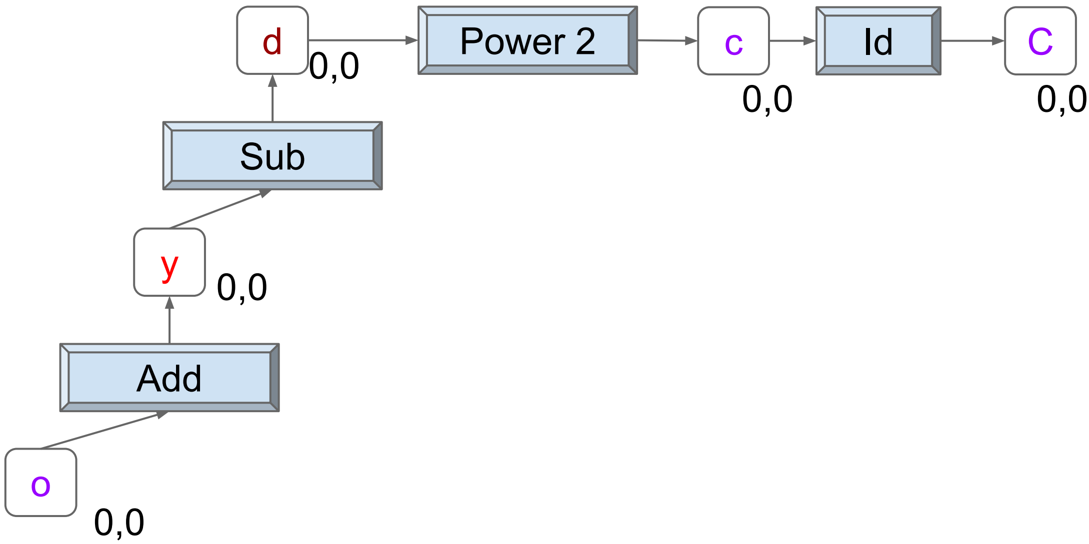
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



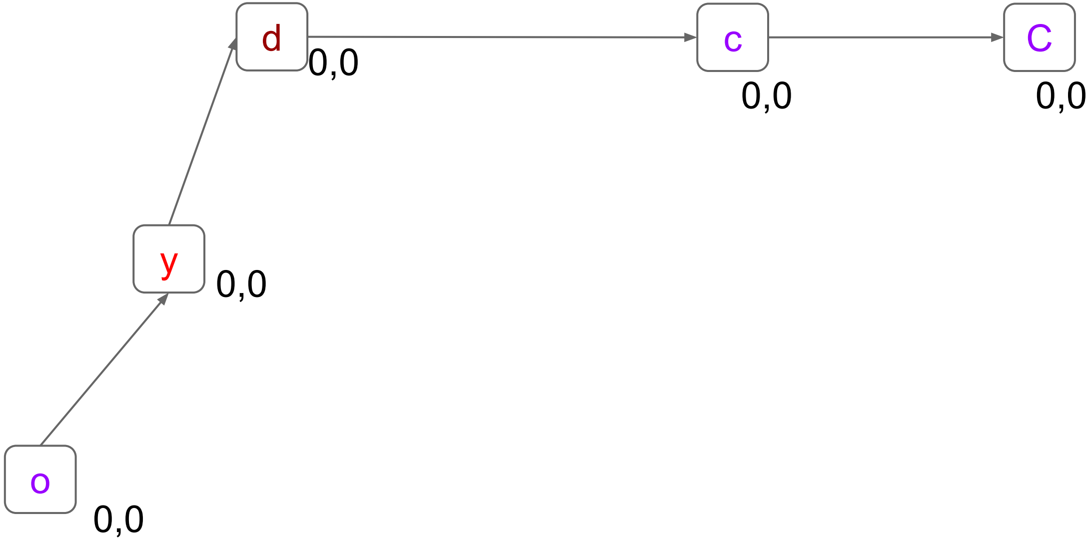
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



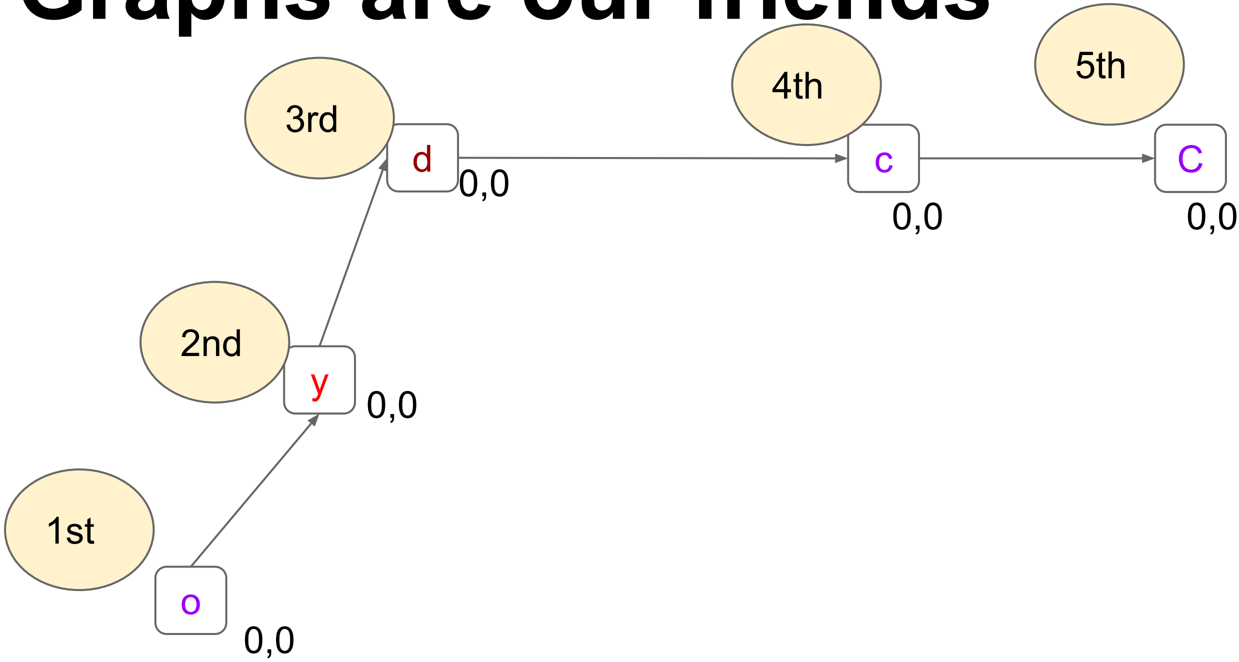
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



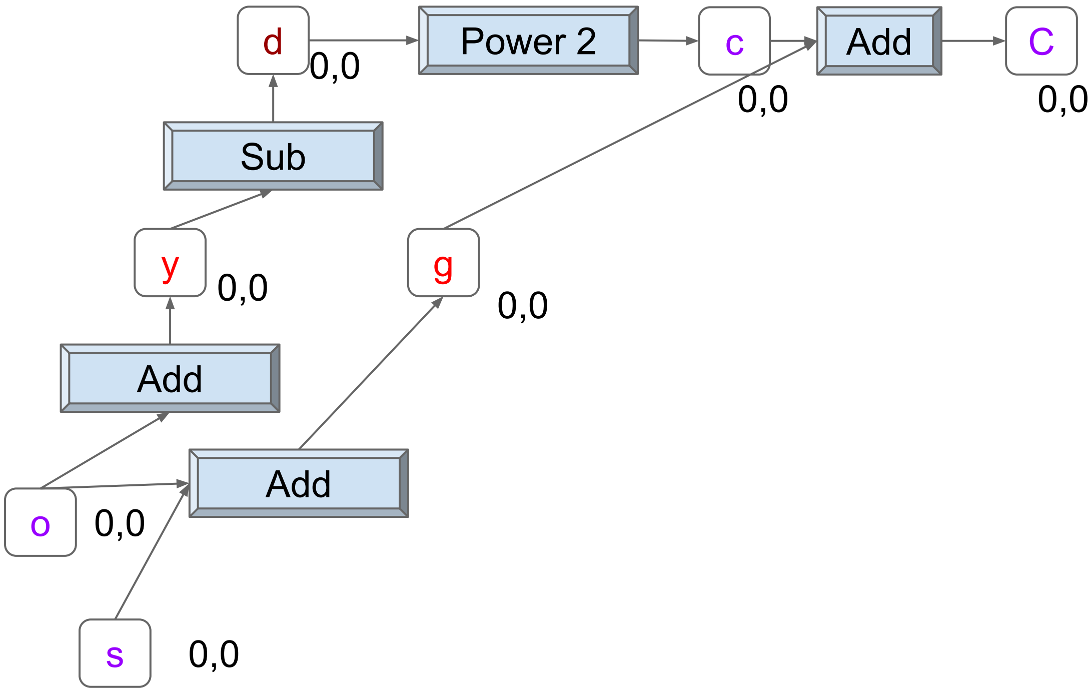
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



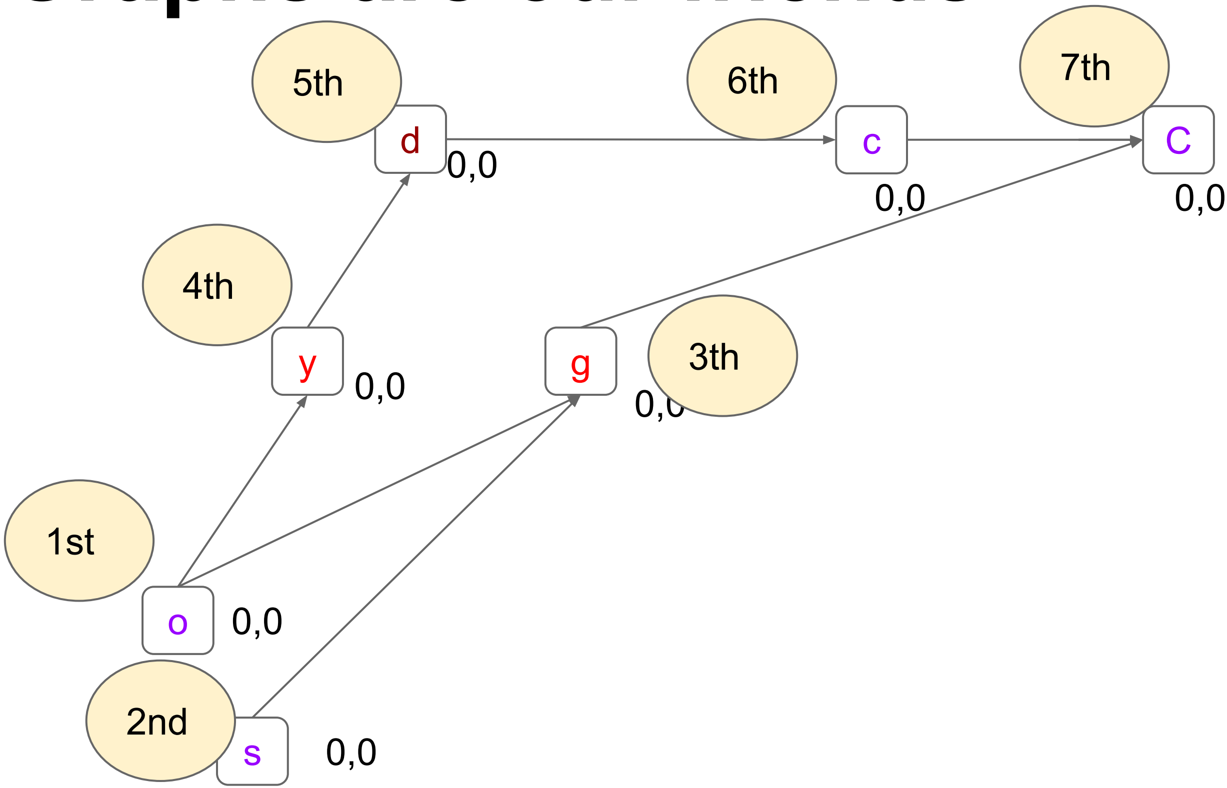
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



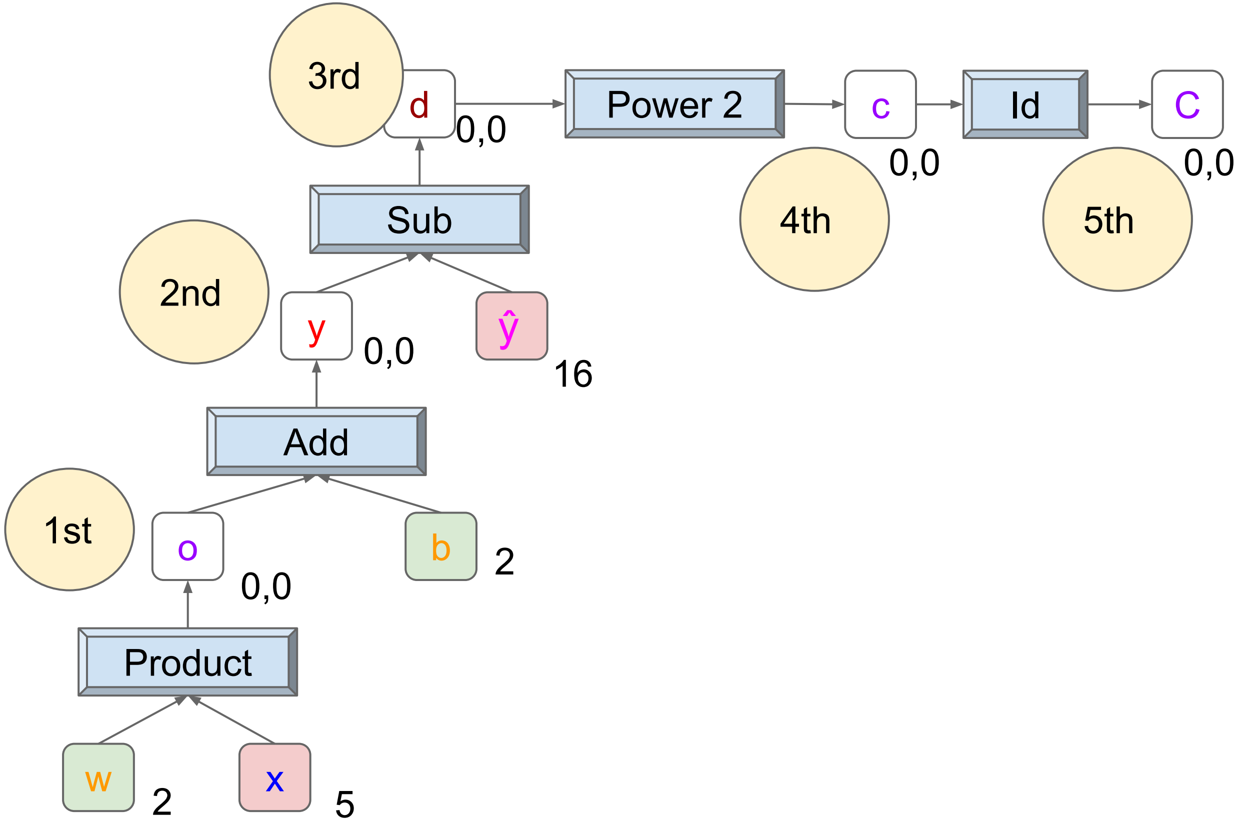
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables



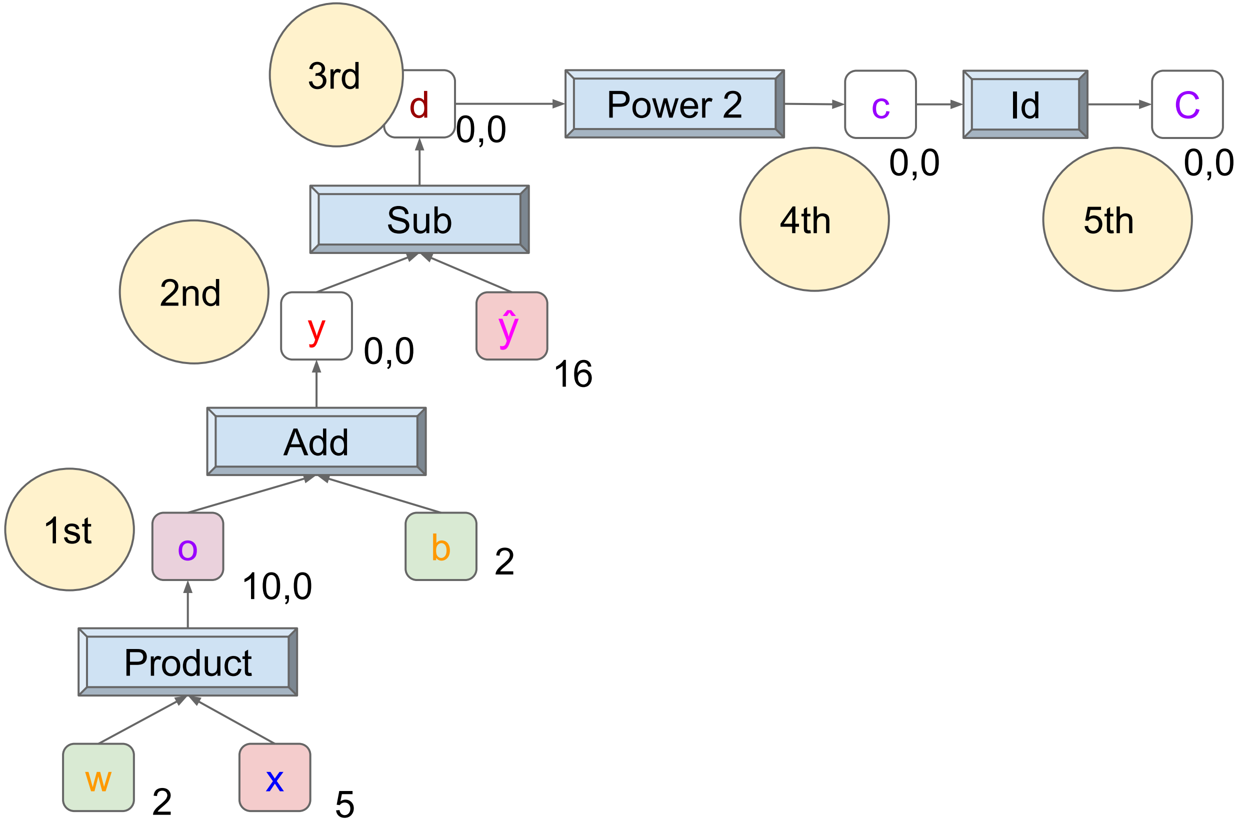
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them



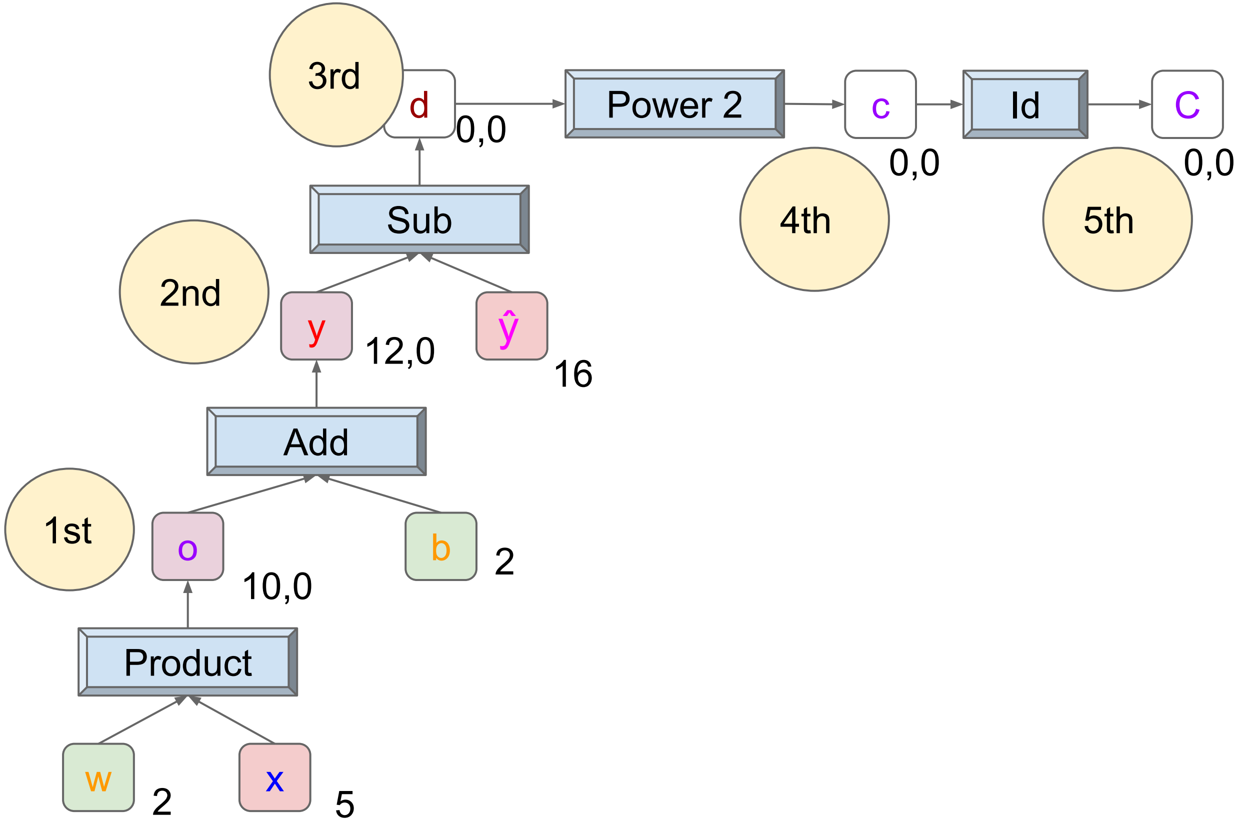
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them



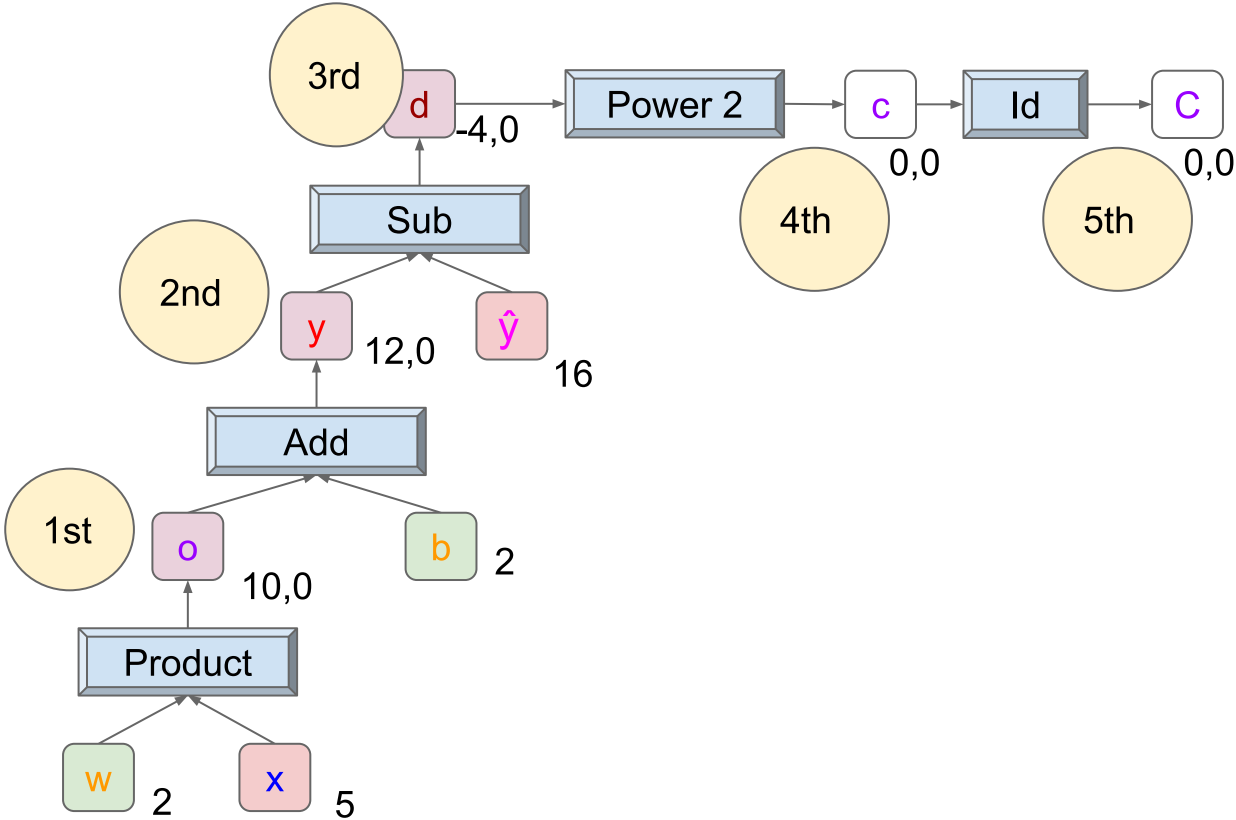
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them



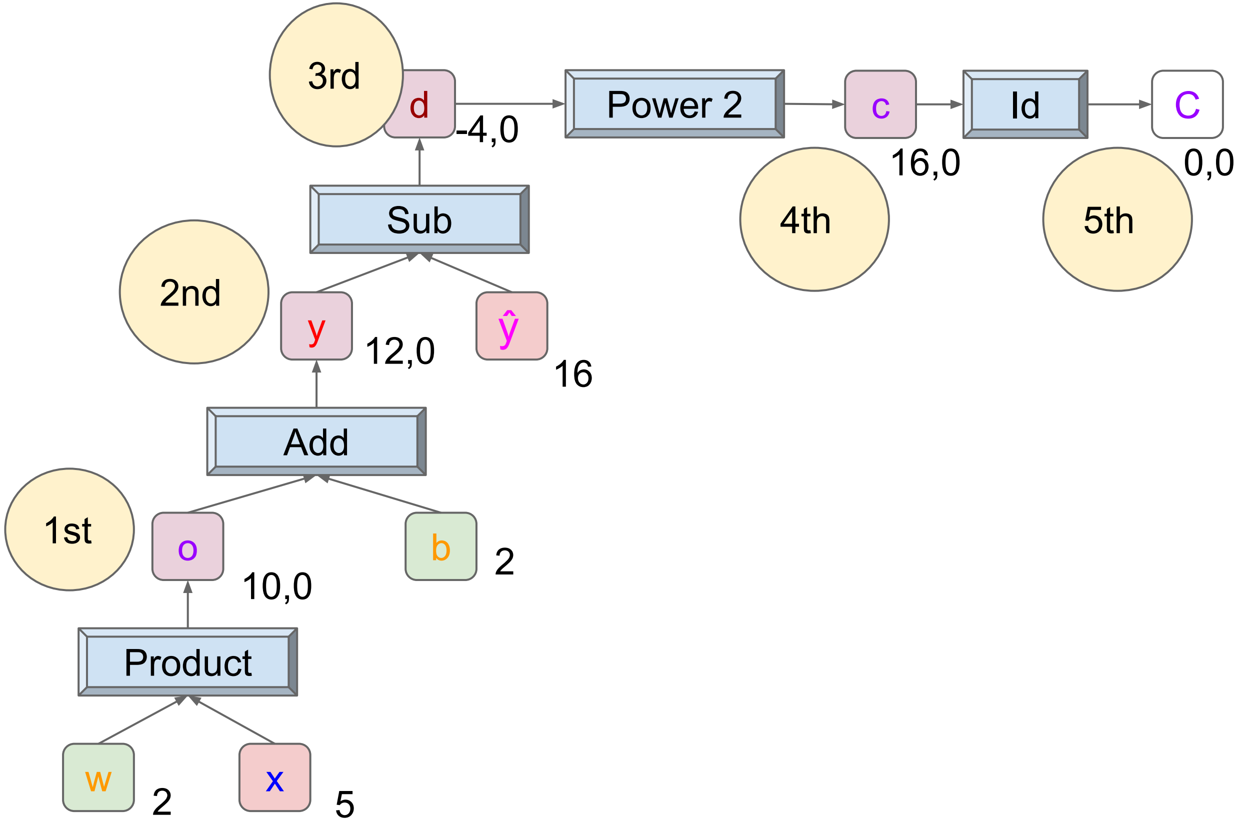
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them



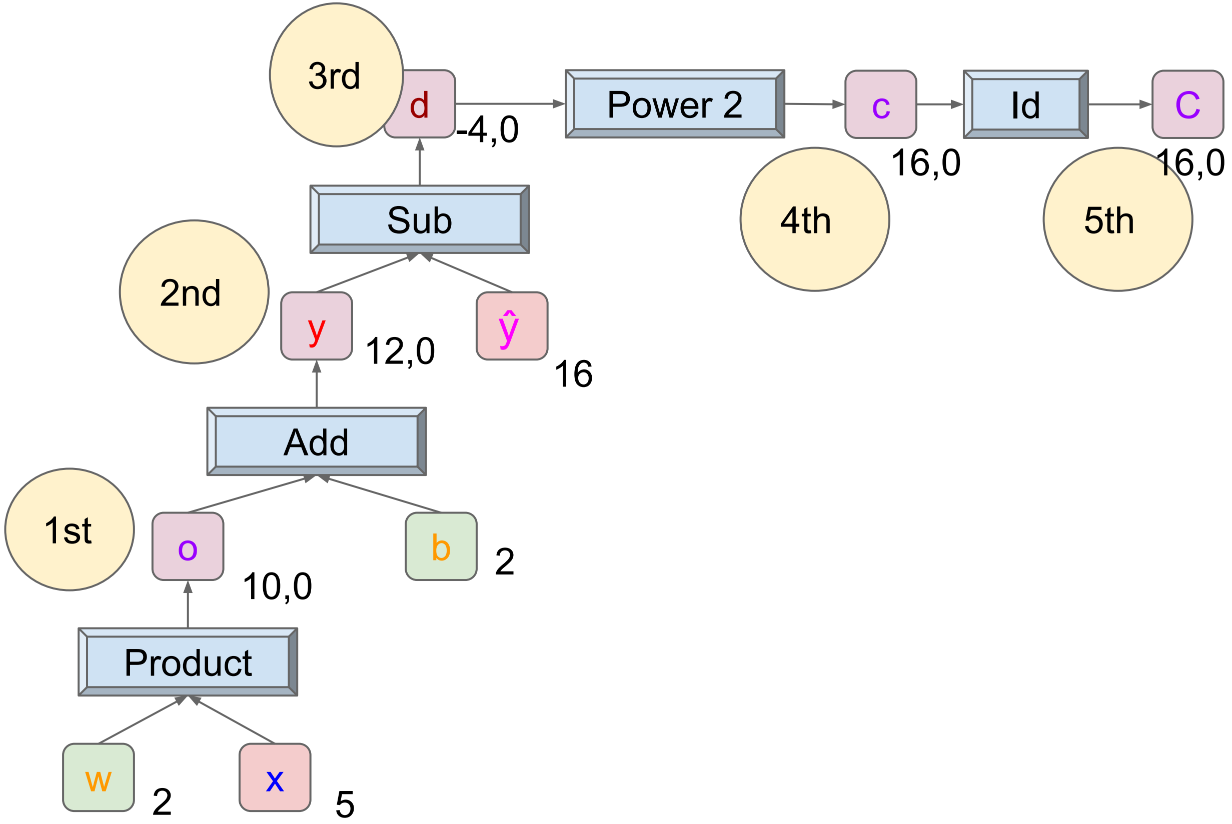
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them



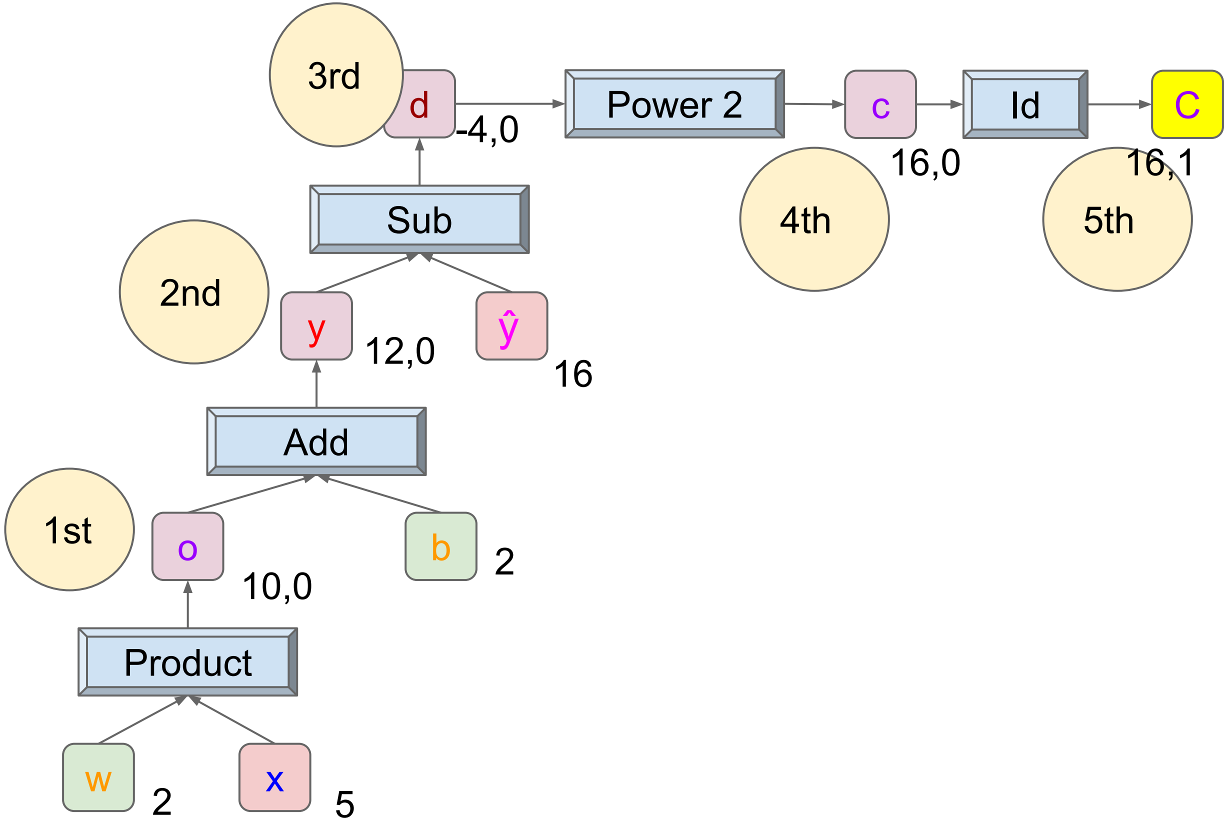
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them



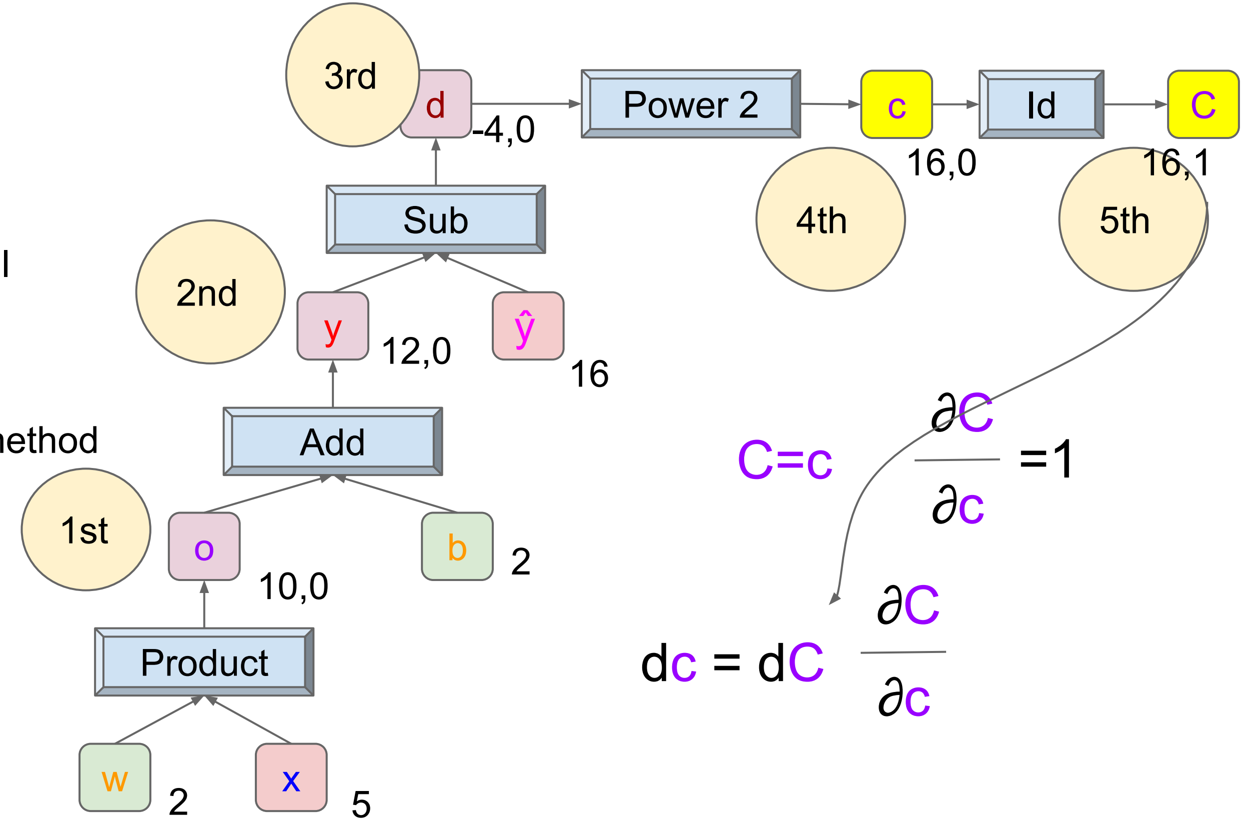
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them
5-Set gradients to final variables



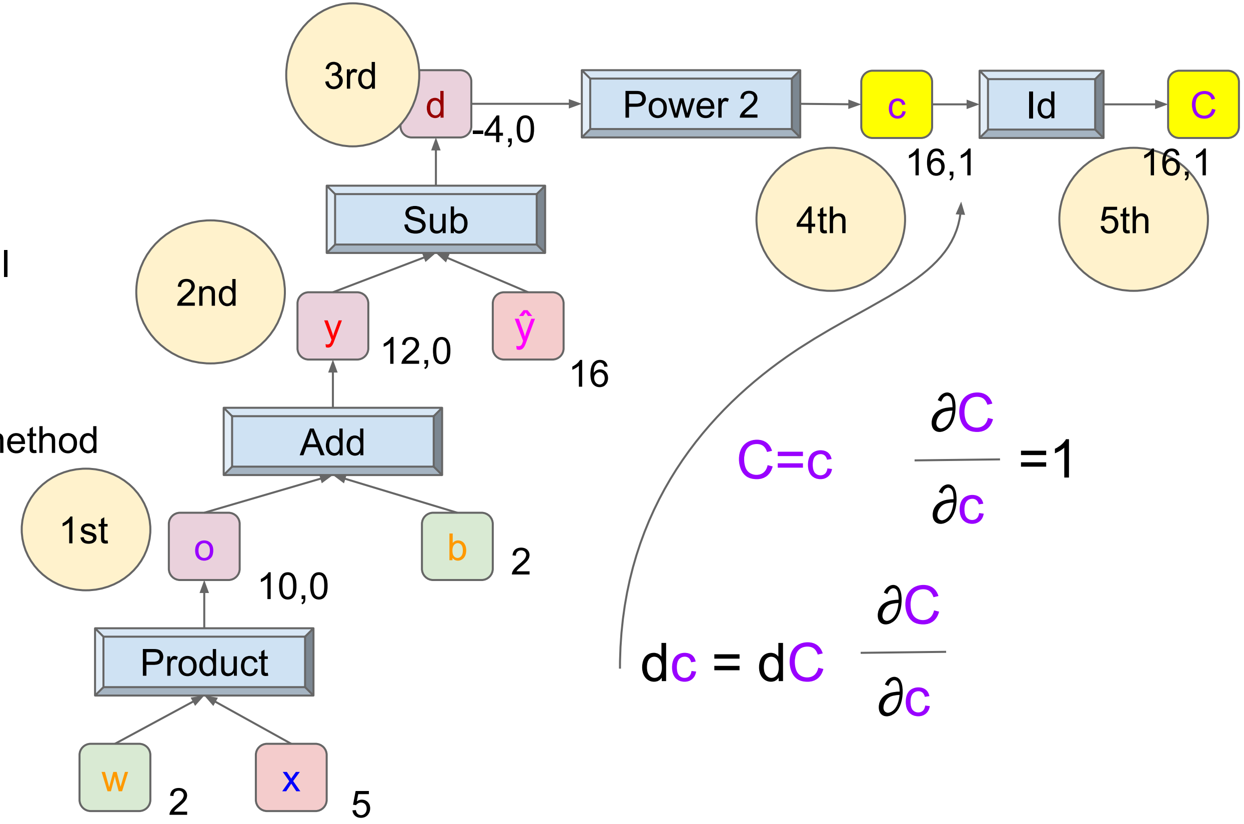
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



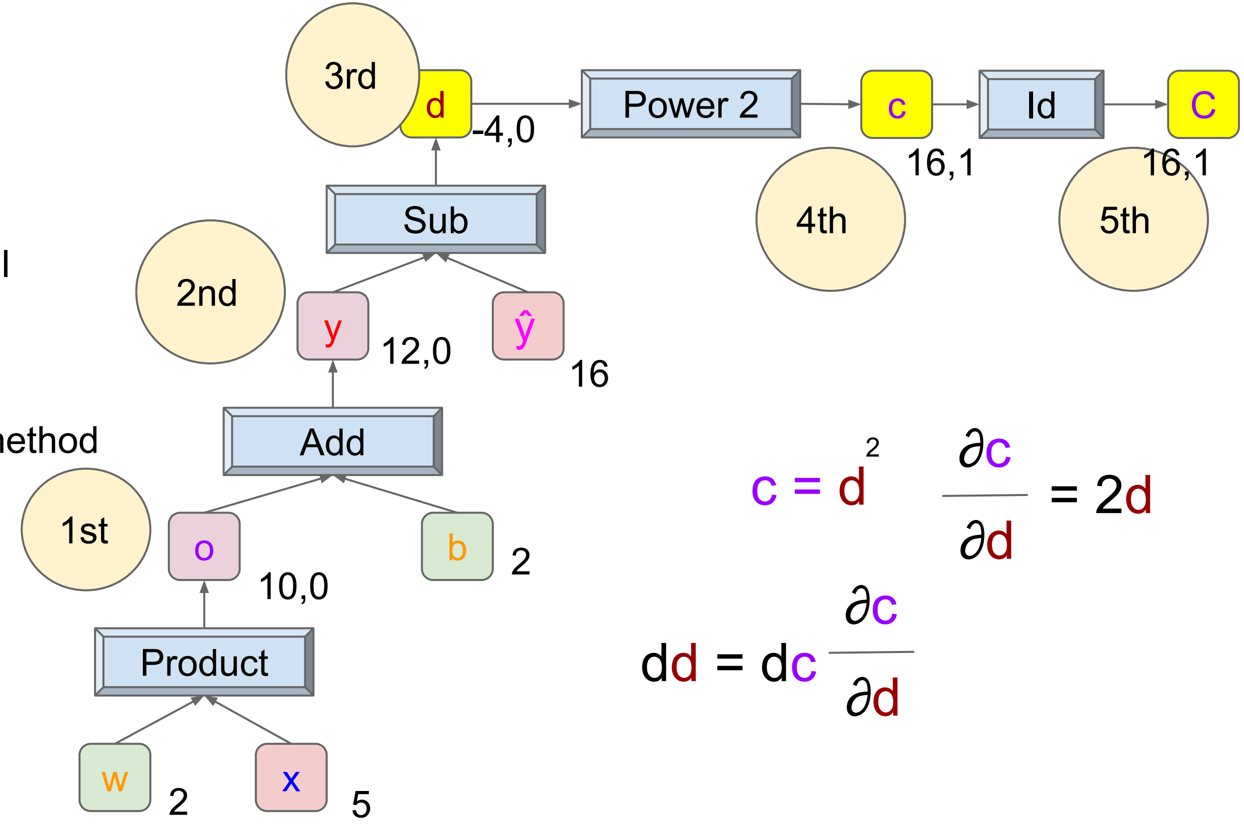
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)

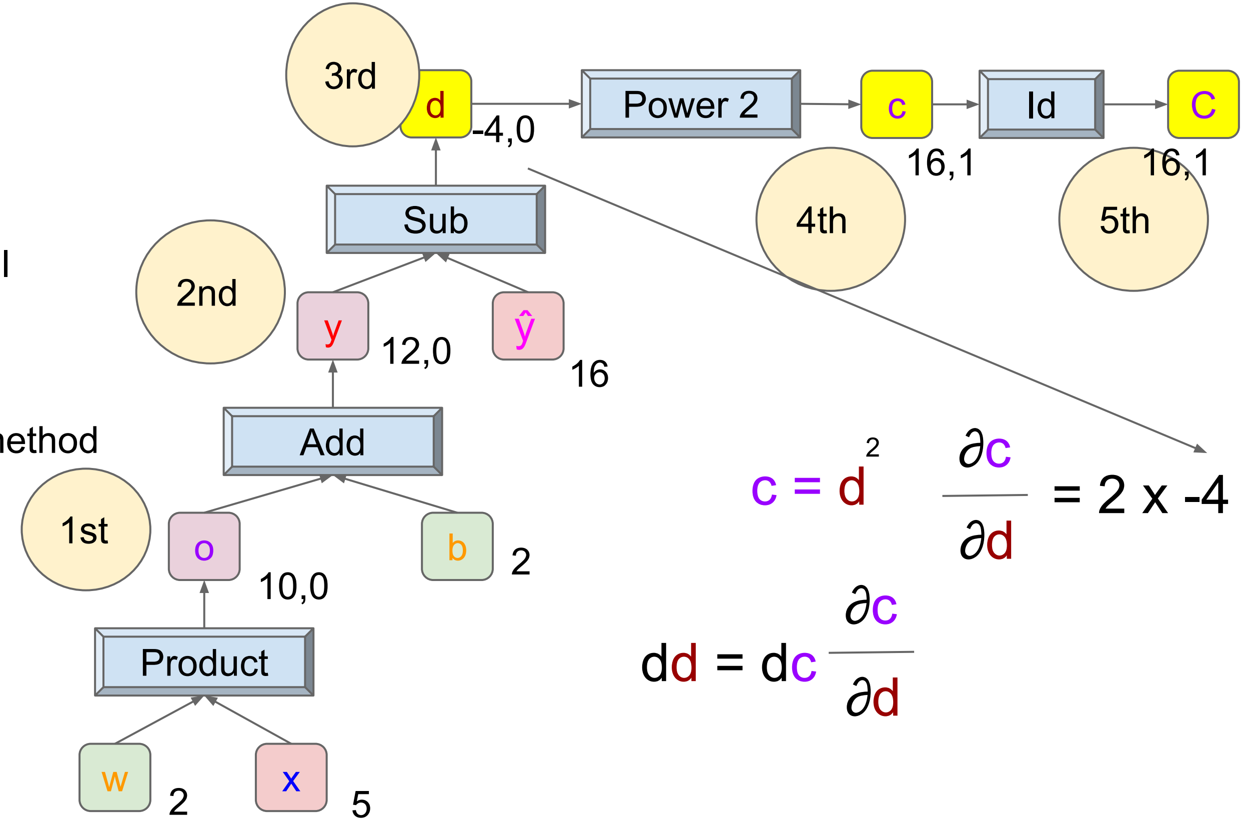


$$c = d^2 \qquad \frac{\partial c}{\partial d} = 2d$$

$$dd = dc \frac{\partial c}{\partial d}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)

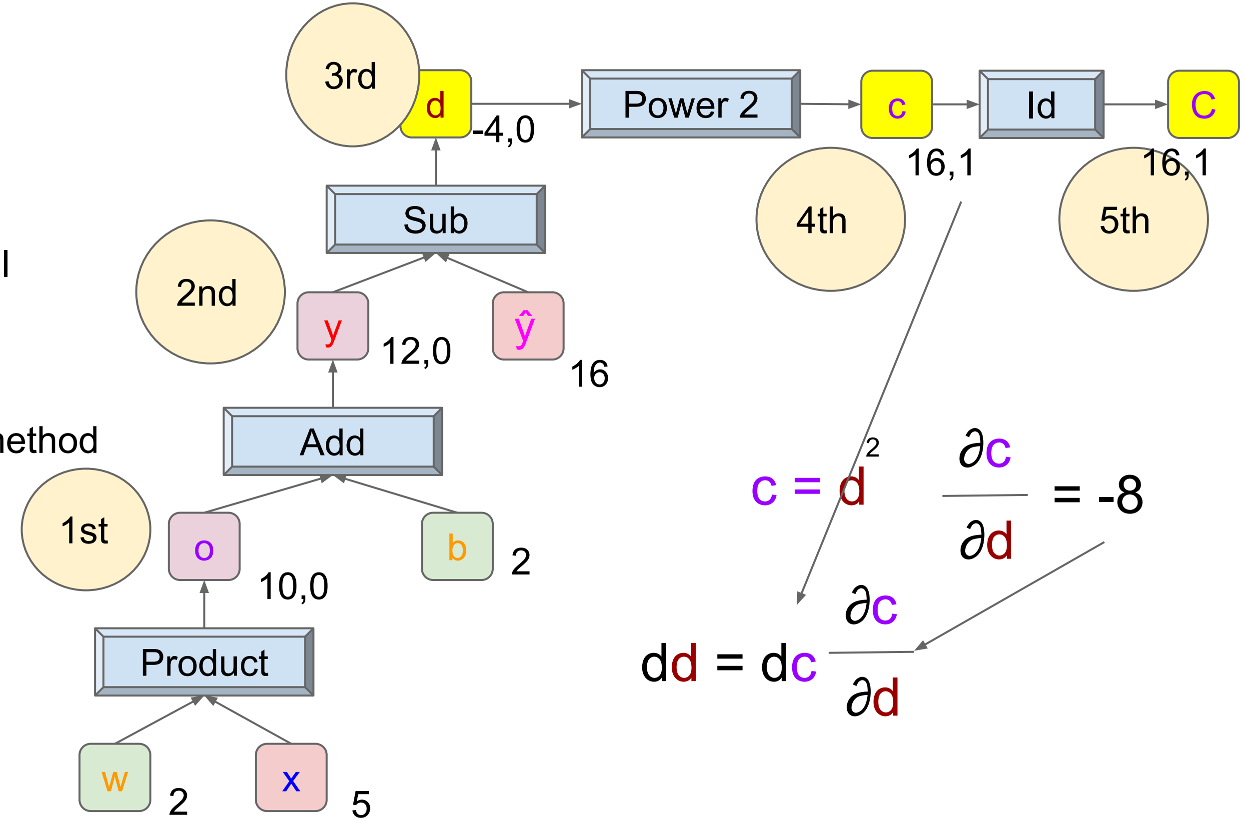


$$c = d^2 \quad \frac{\partial c}{\partial d} = 2 \times -4$$

$$dd = dc \frac{\partial c}{\partial d}$$

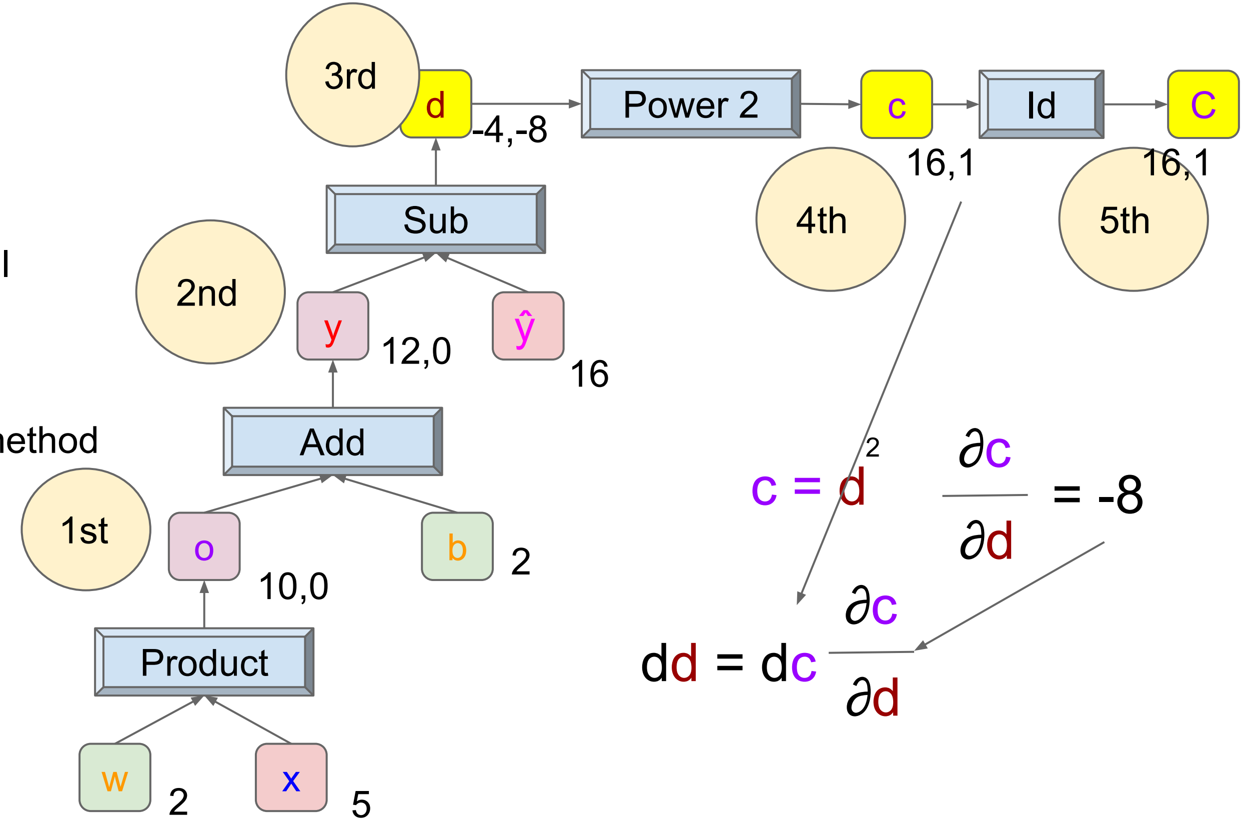
Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)

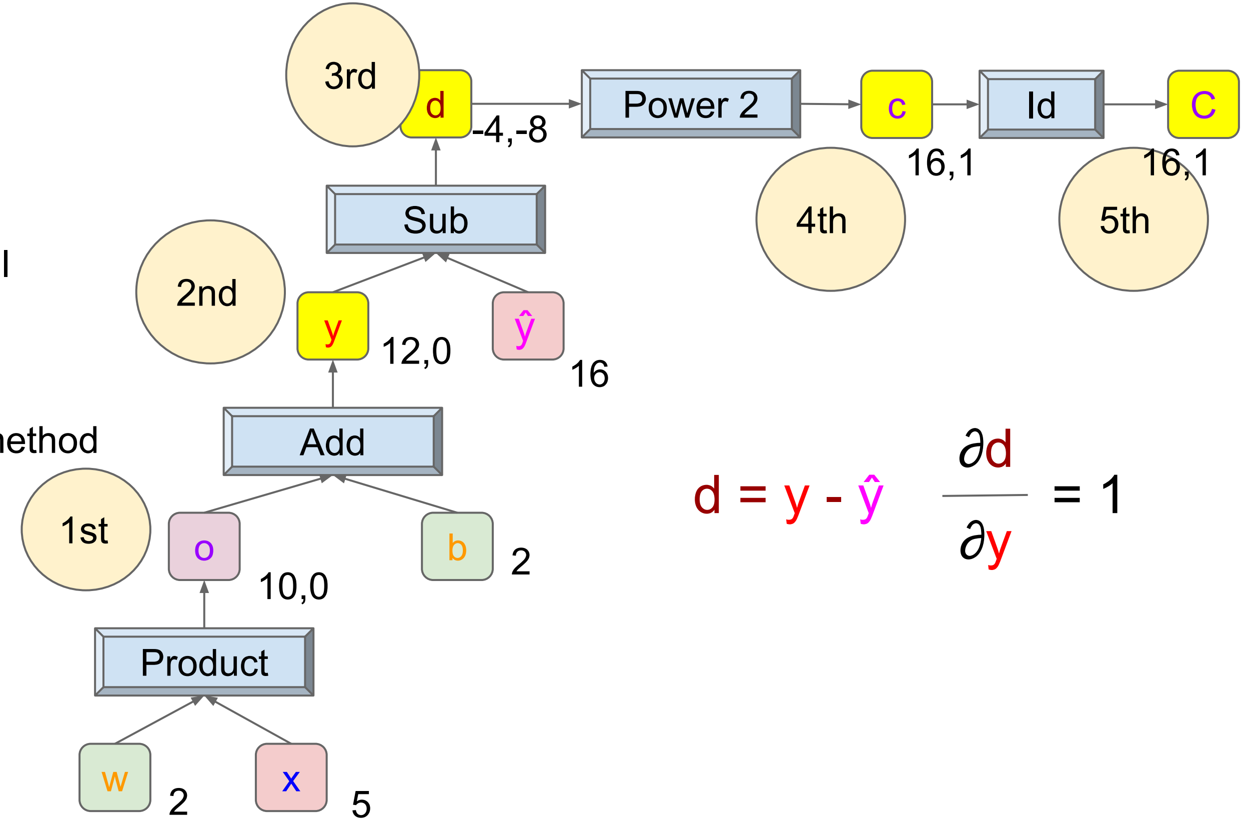


$$c = d^2 \qquad \frac{\partial c}{\partial d} = -8$$

$$dd = dc \frac{\partial c}{\partial d}$$

Computation Graphs are our friends

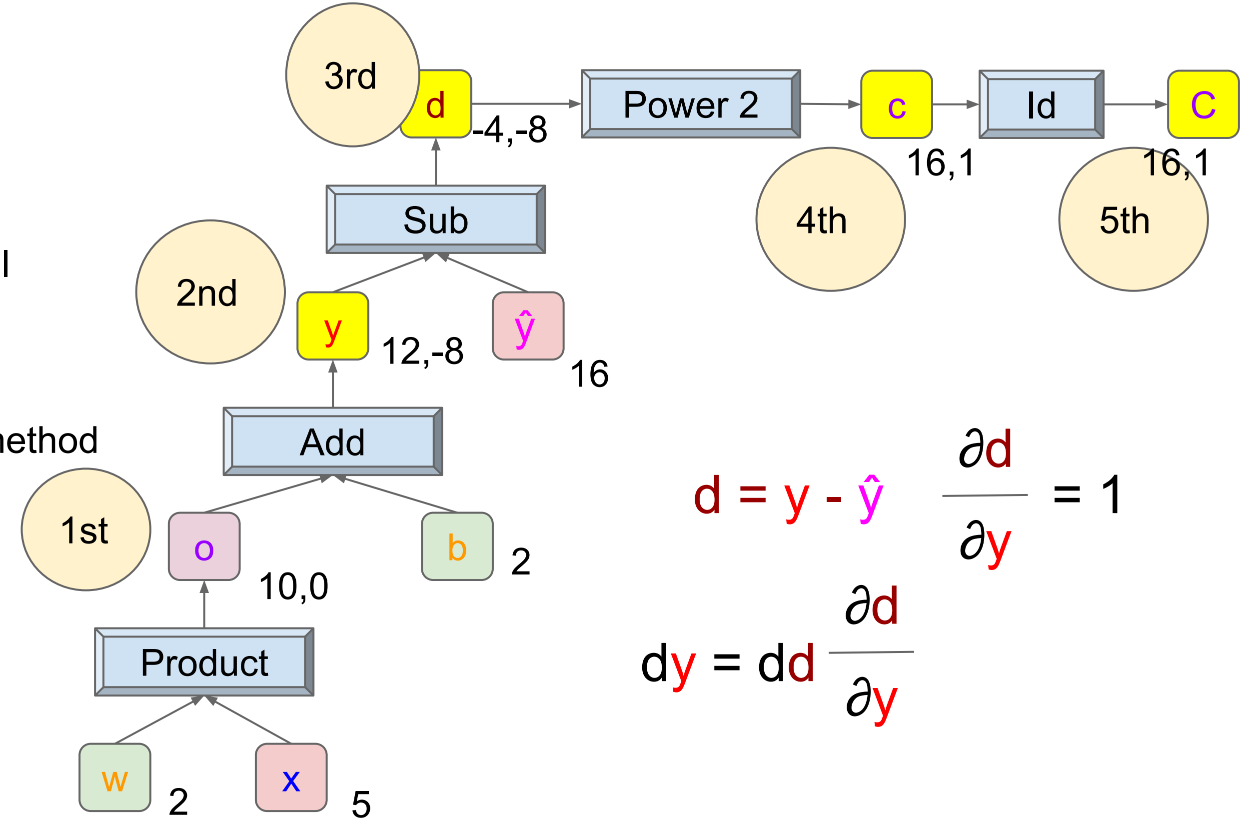
1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



$$d = y - \hat{y} \qquad \frac{\partial d}{\partial y} = 1$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)

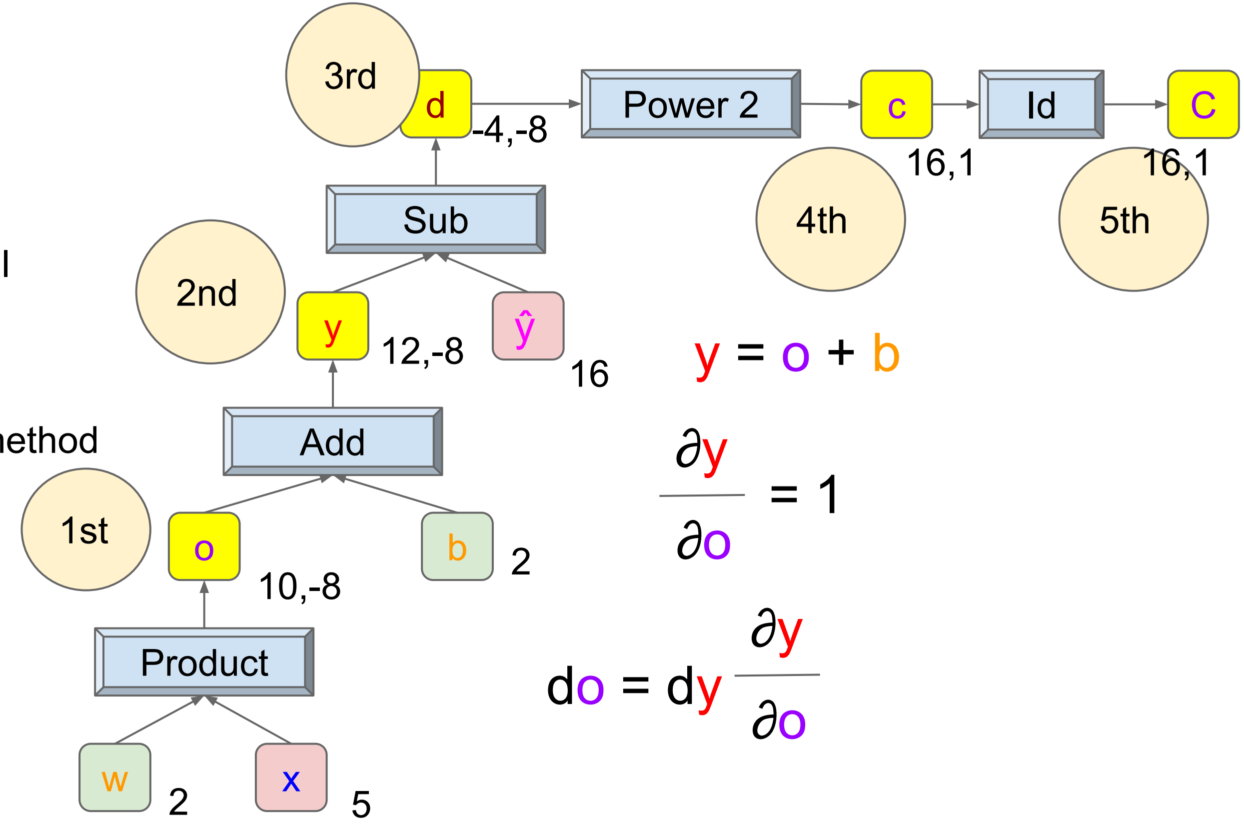


$$d = y - \hat{y} \qquad \frac{\partial d}{\partial y} = 1$$

$$dy = dd \frac{\partial d}{\partial y}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



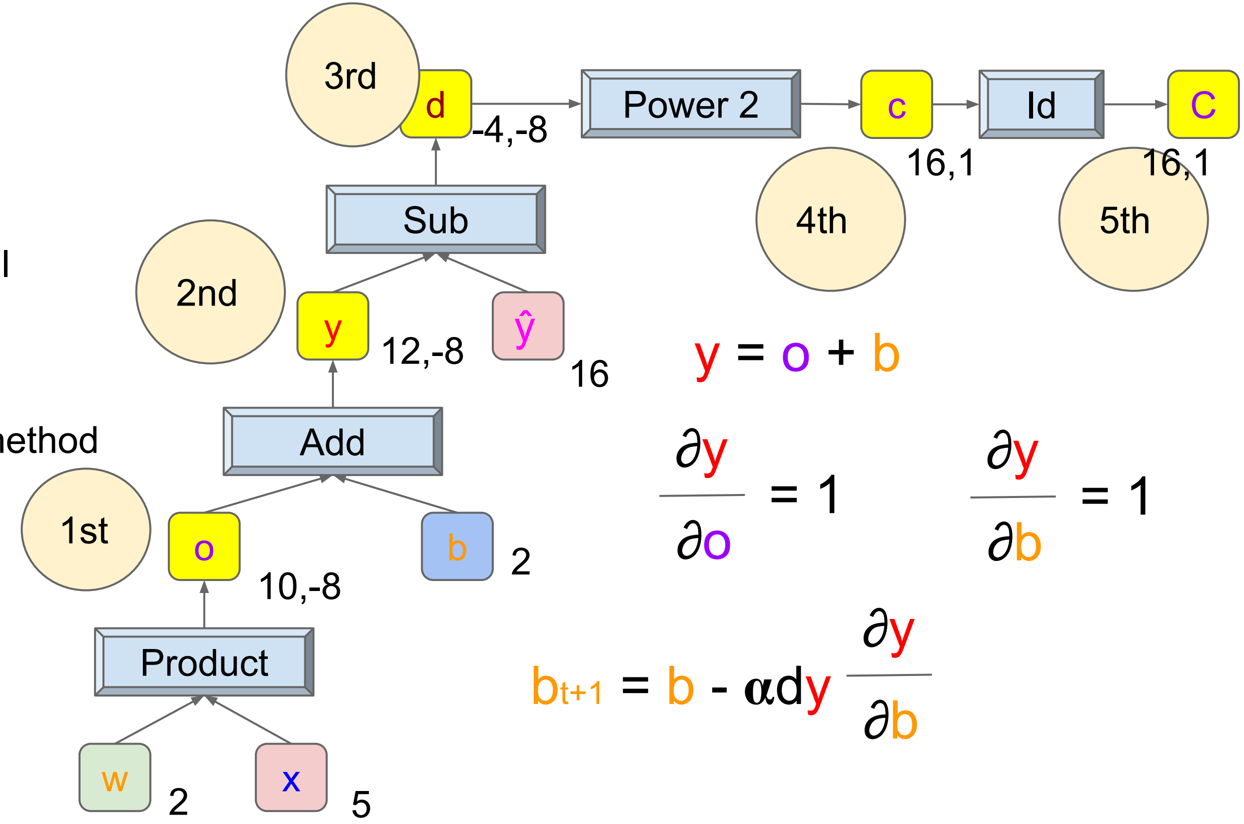
$$y = o + b$$

$$\frac{\partial y}{\partial o} = 1$$

$$do = dy \frac{\partial y}{\partial o}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



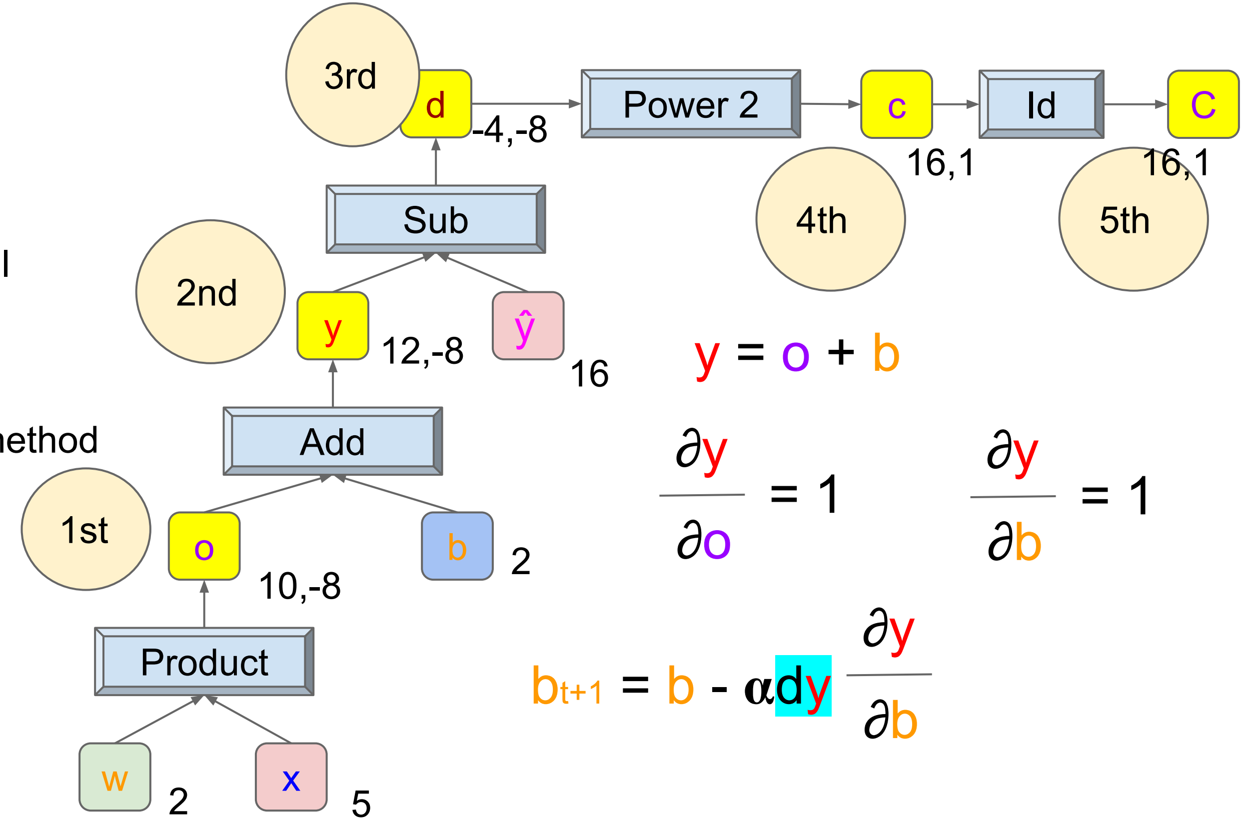
$$y = o + b$$

$$\frac{\partial y}{\partial o} = 1 \qquad \frac{\partial y}{\partial b} = 1$$

$$b_{t+1} = b - \alpha dy \frac{\partial y}{\partial b}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



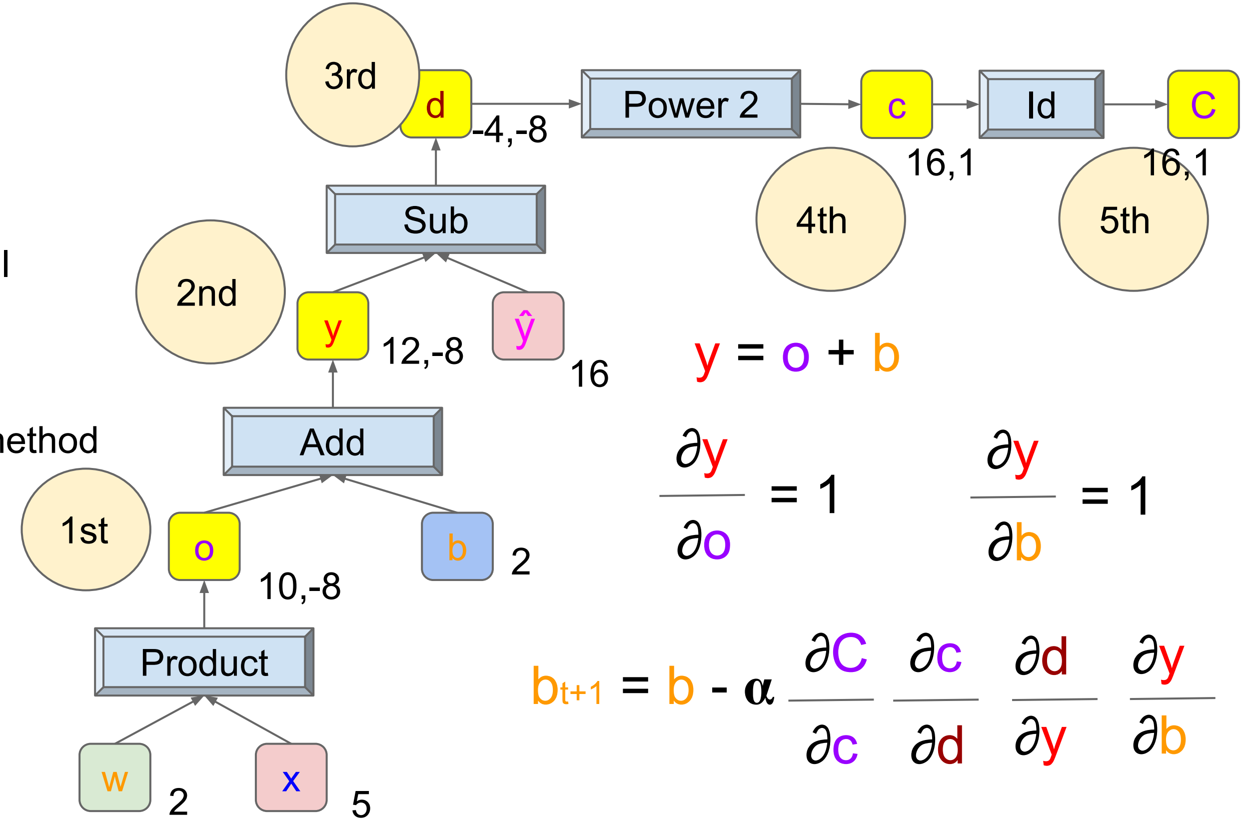
$$y = o + b$$

$$\frac{\partial y}{\partial o} = 1 \qquad \frac{\partial y}{\partial b} = 1$$

$$b_{t+1} = b - \alpha dy \frac{\partial y}{\partial b}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



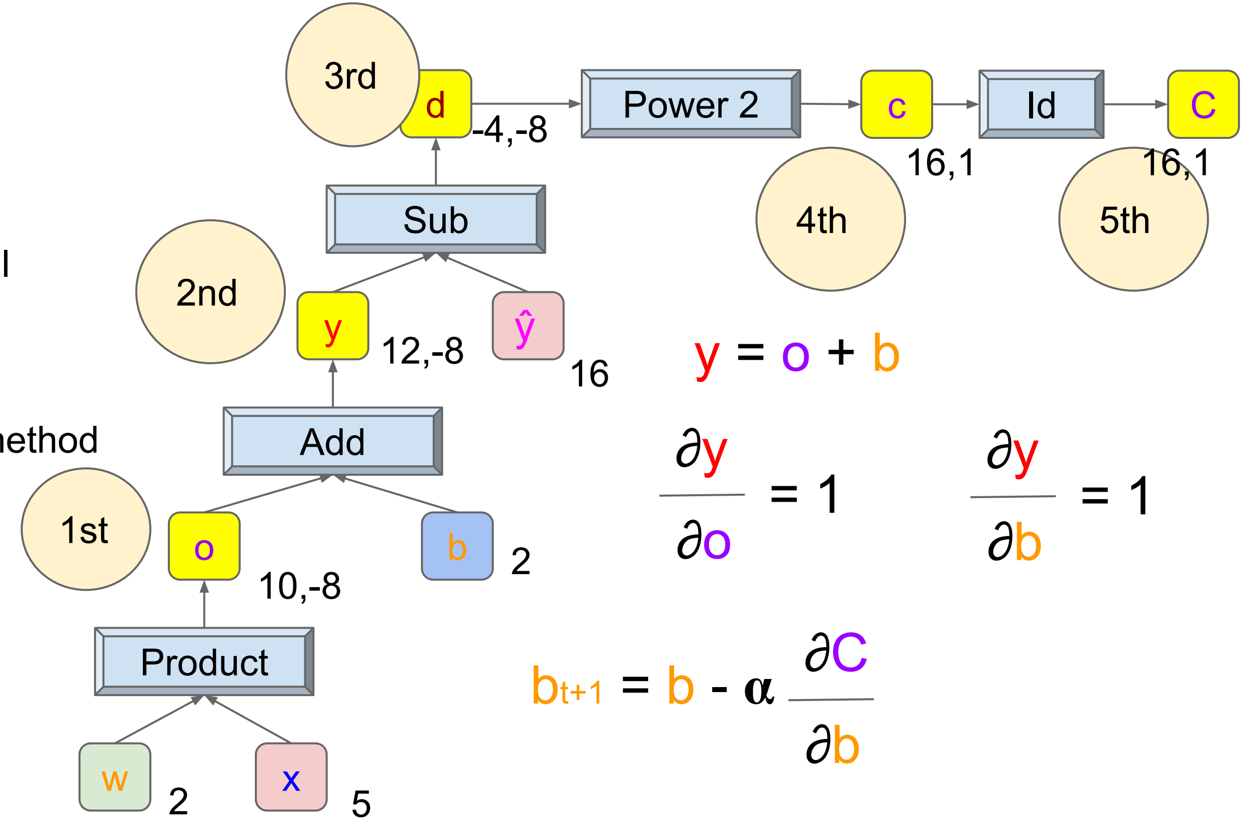
$$y = o + b$$

$$\frac{\partial y}{\partial o} = 1 \qquad \frac{\partial y}{\partial b} = 1$$

$$b_{t+1} = b - \alpha \frac{\partial C}{\partial c} \frac{\partial c}{\partial d} \frac{\partial d}{\partial y} \frac{\partial y}{\partial b}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



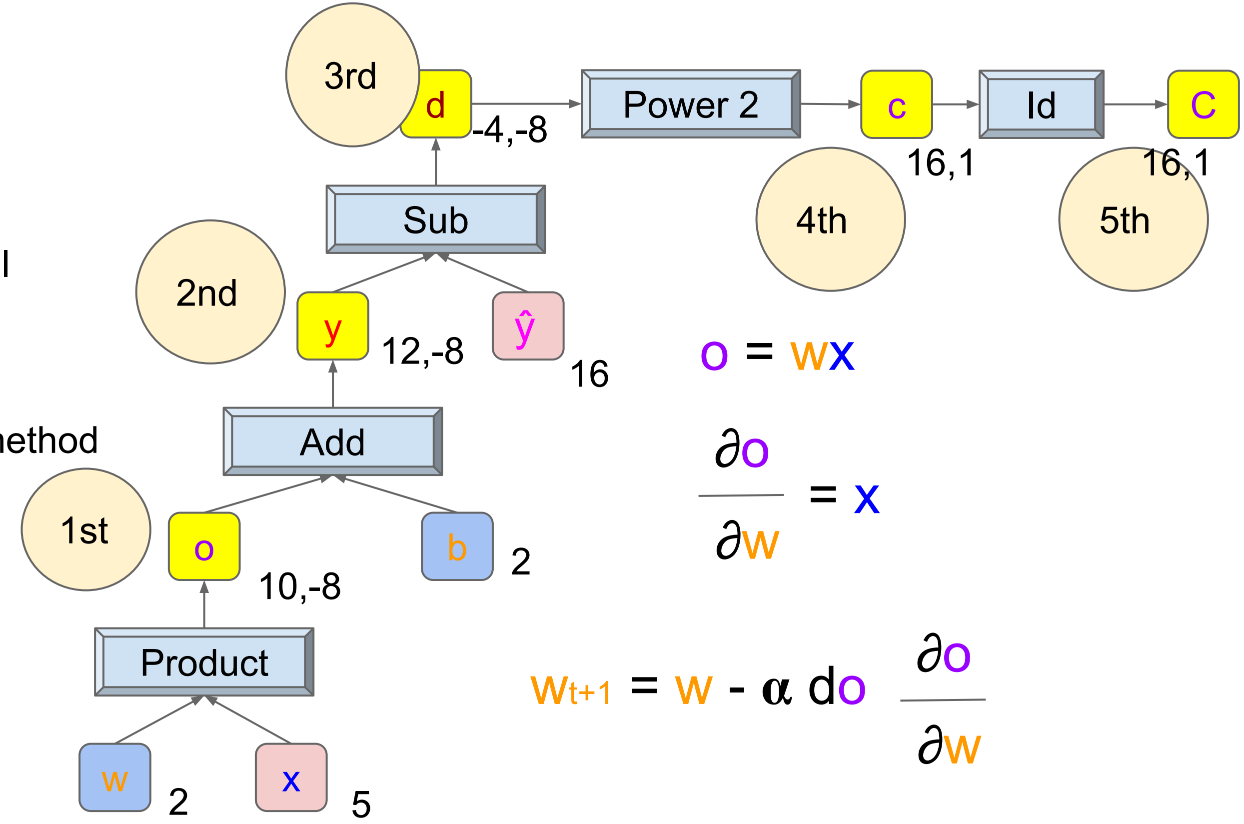
$$y = o + b$$

$$\frac{\partial y}{\partial o} = 1 \qquad \frac{\partial y}{\partial b} = 1$$

$$b_{t+1} = b - \alpha \frac{\partial C}{\partial b}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)



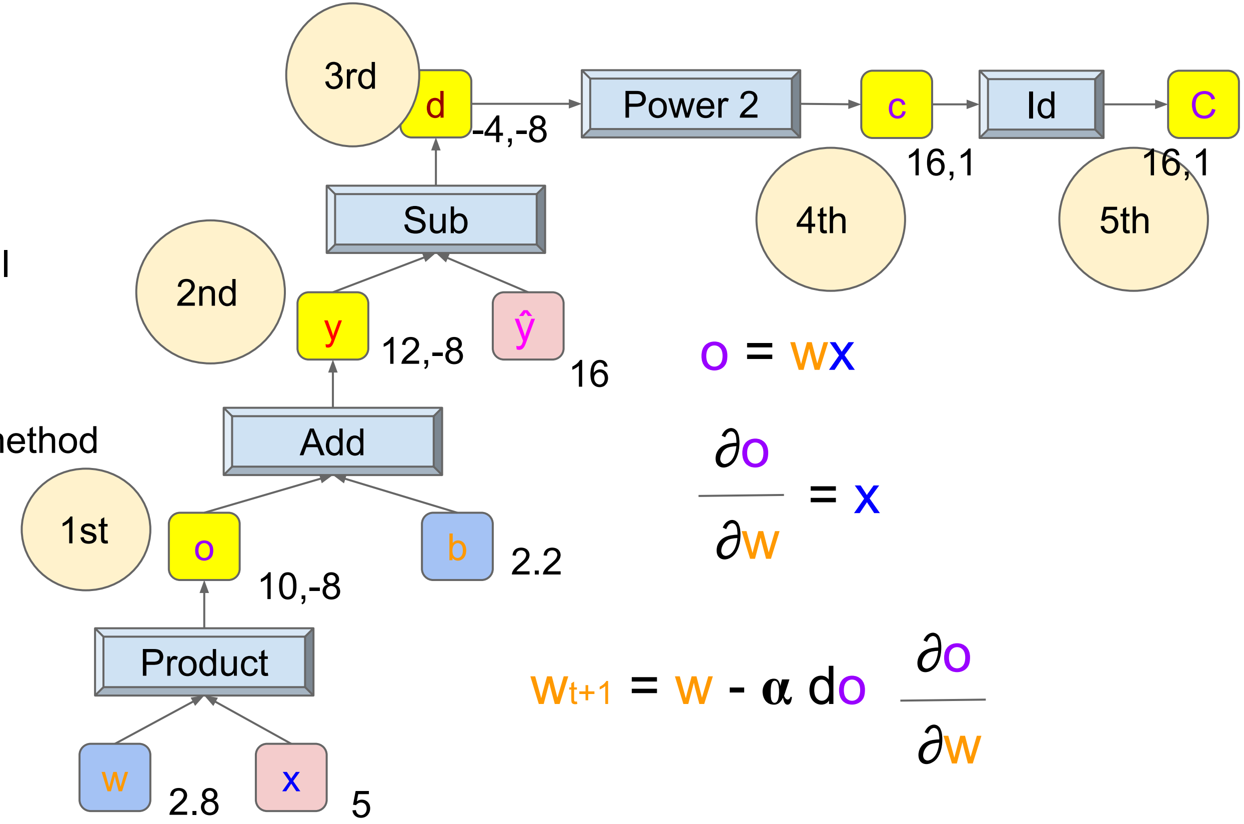
$$o = wx$$

$$\frac{\partial o}{\partial w} = x$$

$$w_{t+1} = w - \alpha \, do \, \frac{\partial o}{\partial w}$$

Computation Graphs are our friends

1-Initialize inputs
2-Initialize variables
3-Topological Sort variables
4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
5-Set gradients to final variables
6-run the operations backward method in reverse order (Backward)
7-update parameters



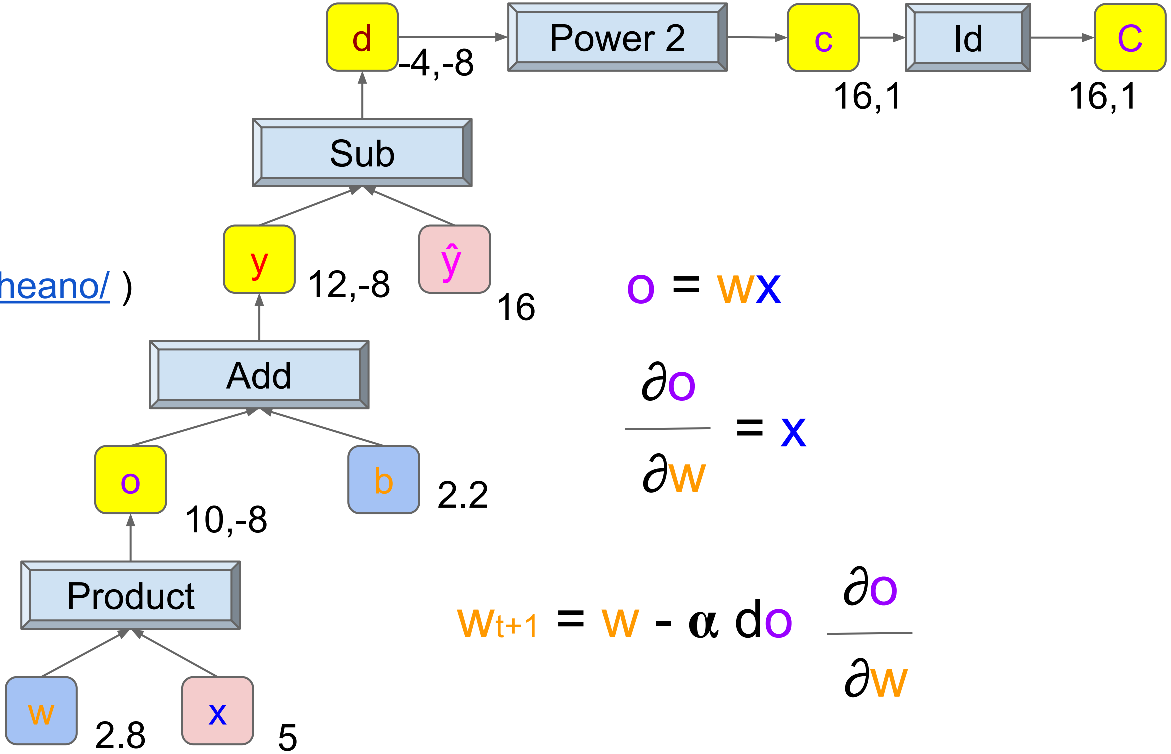
$$o = wx$$

$$\frac{\partial o}{\partial w} = x$$

$$w_{t+1} = w - \alpha\, do \frac{\partial o}{\partial w}$$

Computation Graphs are our friends

Existing Tools:
-Tensorflow (https://www.tensorflow.org)
-Torch (https://github.com/torch/nn)
-CNN (https://github.com/clab/cnn)
-JNN (https://github.com/wlin12/JNN)
-Theano (http://deeplearning.net/software/theano/)



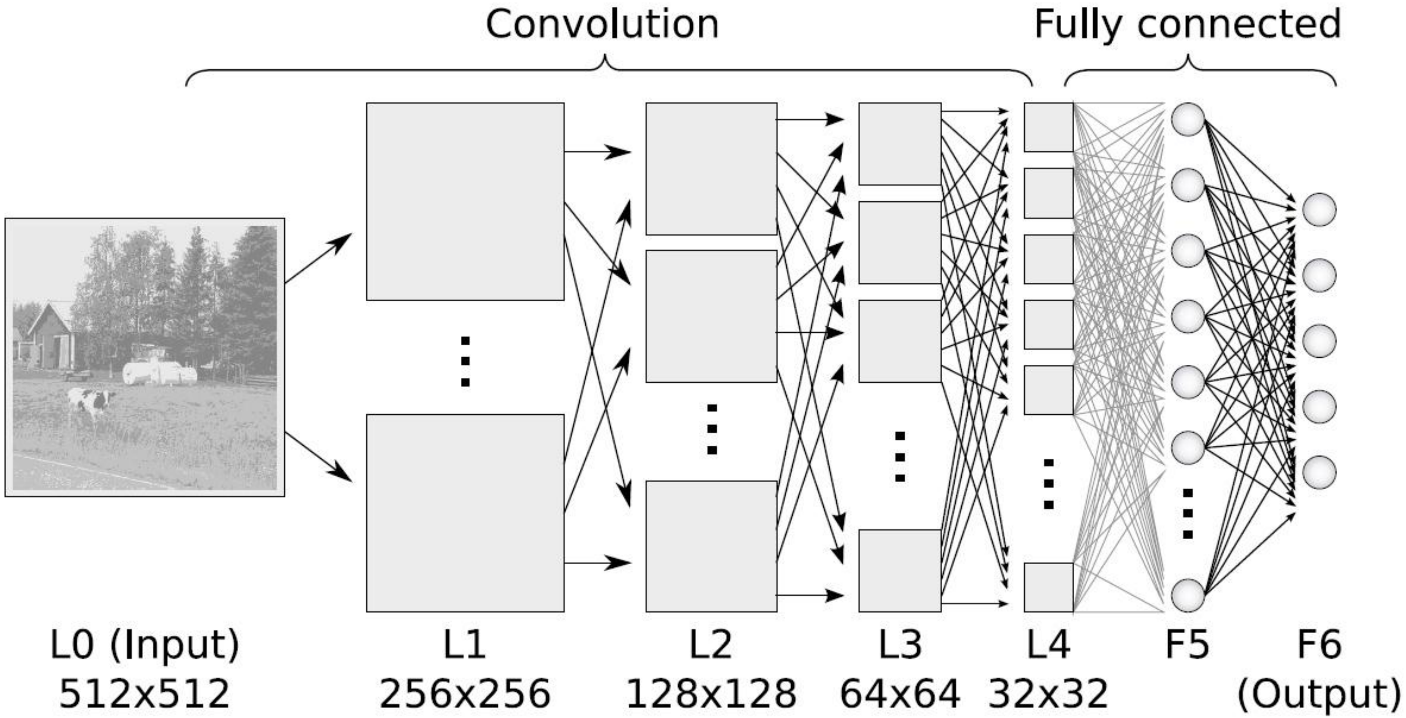
$$o = wx$$

$$\frac{\partial o}{\partial w} = x$$

$$w_{t+1} = w - \alpha \, do \, \frac{\partial o}{\partial w}$$

Deep Neural Networks are our friends?

Convolutional Neural Network



Deep Neural Networks are our friends?

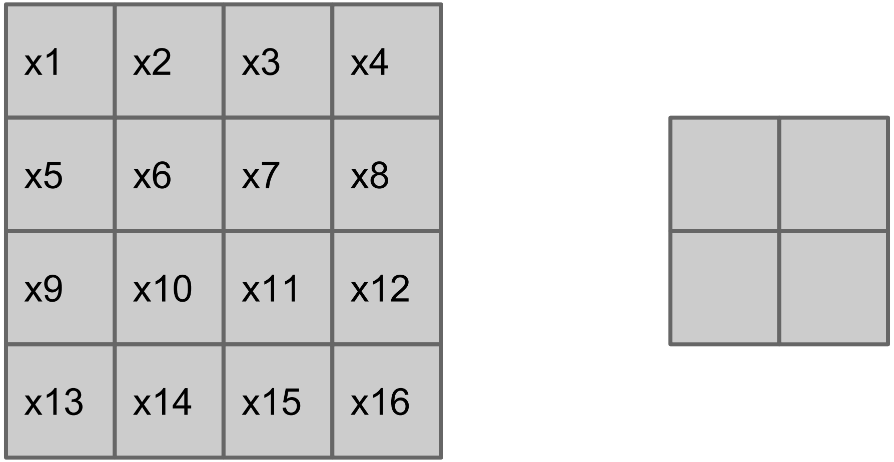
Convolutional Neural Network

x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12
x13	x14	x15	x16

4x4 image

Deep Neural Networks are our friends?

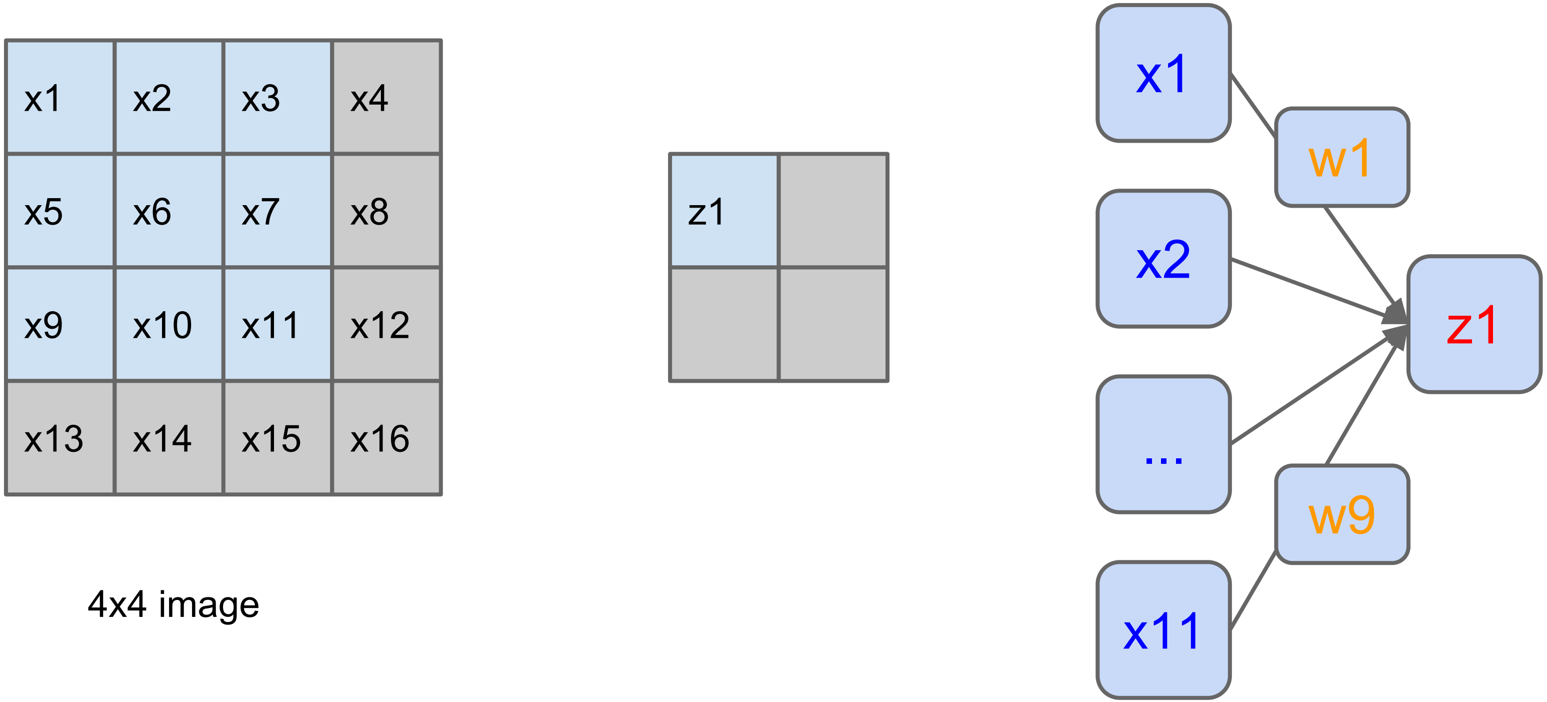
Convolutional Neural Network



4x4 image

Deep Neural Networks are our friends?

Convolutional Neural Network



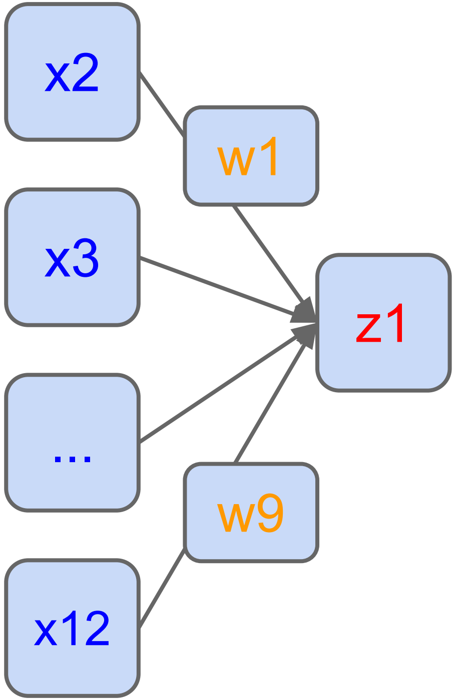
Deep Neural Networks are our friends?

Convolutional Neural Network

x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12
x13	x14	x15	x16

4x4 image

z1	z2



Deep Neural Networks are our friends?

Convolutional Neural Network

x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12
x13	x14	x15	x16

4x4 image

z1	z2
z3	z4

Deep Neural Networks are our friends?

Convolutional Neural Network