

Deep Neural Networks Are Our Friends



Wang Ling



Outline

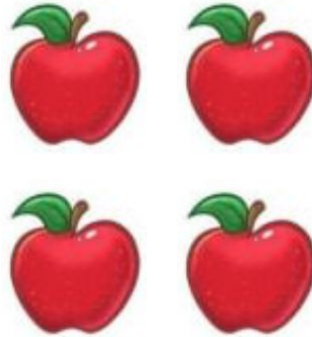
- Part I - Neural Networks are our friends
 - Numbers are our friends
 - Variables are our friends
 - Operators are our friends
 - Functions are our friends
 - Parameters are our friends
 - Cost Functions are our friends
 - Optimizers are our friends
 - Gradients are our friends

Outline

- Part 1 - 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

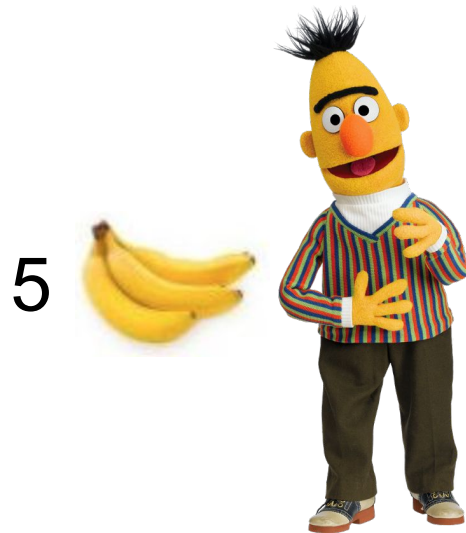


Variables are our friends

Abby



Bert



Variables are our friends

Abby



Bert

$5y$



Operators are our friends



1 🍏

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

Bert



Operators are our friends



$$4x - 1x = 3x$$

Bert



Functions are our friends



4 🍏

1 🍏

? 🍌

5 🍌

If you give me
1 apple I will
give you 3
bananas



Functions are our friends

$$y = 3x$$

- Input, x - Number of Apples given by Abby

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



4 🍏

1 🍏

? 🍌

5 🍌



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

Functions are our friends



4 🍏

1 🍏

3 🍌

5 🍌



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

$$y = 3$$

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

$x : \text{Image}$



$y : \text{Category}$



Functions are our friends

x : Board



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

y : Move



Functions are our friends

$$y = 3x$$

Cookie Monster



Functions are our friends

$$y = ??$$

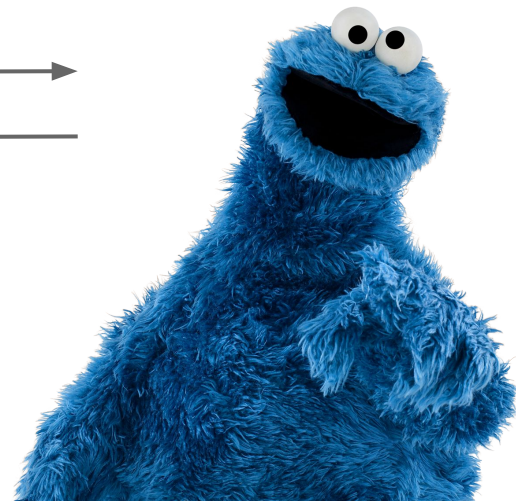
$$y = 3x$$

Find it out for
yourself



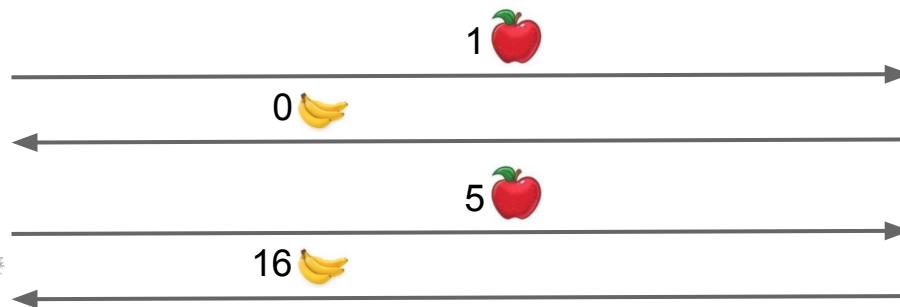
Functions are our friends

$$y = ??$$



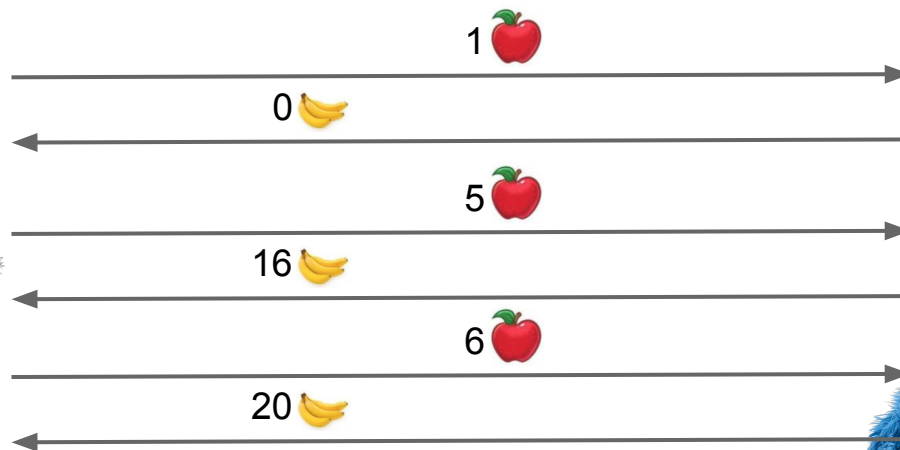
Functions are our friends

$y = ??$



Functions are our friends

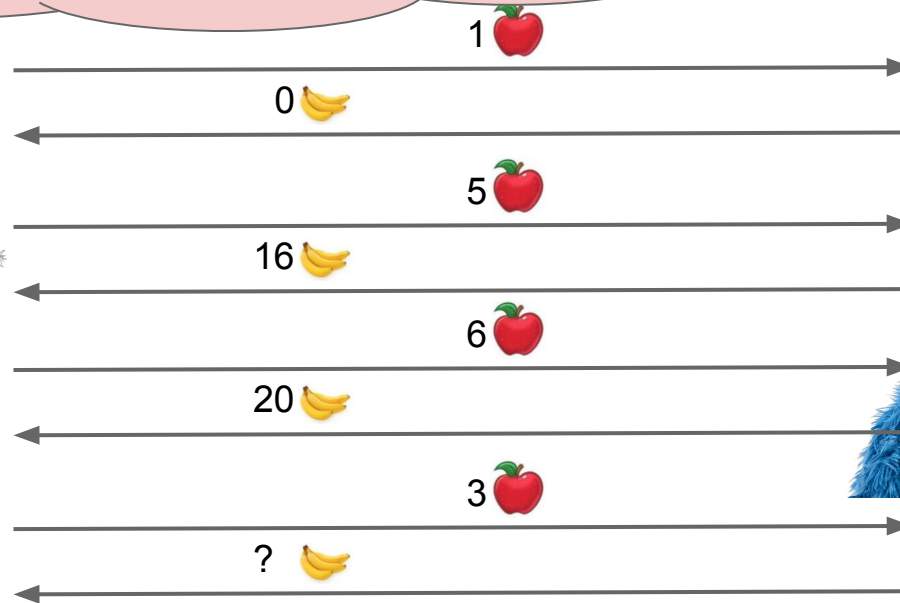
$y = ??$



Functions are our friends

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

$y = ??$



Parameters are our friends

$$y = 3x + 1$$

- Input
- Output

Parameters are our friends

Model

$$y = wx + b$$

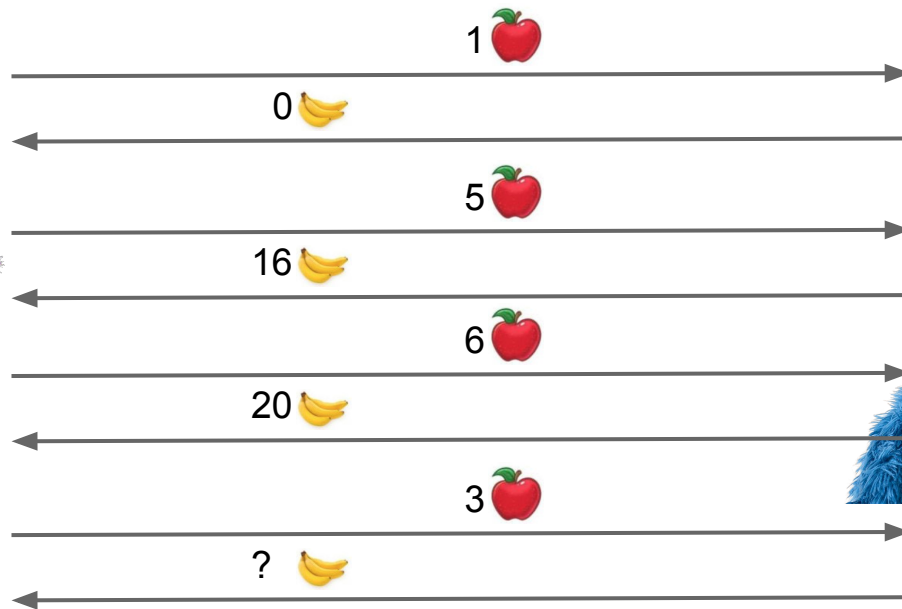
- Input
- Output
- Parameters

Input - Fixed, comes from data

Parameters - Need to be estimated

Parameters are our friends

$$y = wx + b$$



Parameters are our friends

$$y = wx + b$$



Data	
	1 🍏
0 🍌	
	5 🍏
16 🍌	
	6 🍏
20 🍌	
	3 🍏
? 🍌	



Parameters are our friends

$$y = wx + b$$



Data	
x	y
1	0
5	16
6	20



Parameters are our friends

Data	
x	y
1	0
5	16
6	20

Model

$$y = wx + b$$

Parameters are our friends

Data	
x	y
1	0
5	16
6	20

Model

$$y = wx + b$$

How to find the parameters w and b ?

Parameters are our friends

Data	
x	y
1	0
5	16
6	20

$$y = wx + b$$

Model

Model Candidate 1

$$y = 1x + 0$$

x	y
1	0
5	16
6	20

Parameters are our friends

Data	
x	y
1	0
5	16
6	20

Model

$$y = wx + b$$

Model
Candidate 1

$$\begin{aligned}y &= 1x + 0 \\ 1 &= 1 * 1 + 0 \\ 5 &= 1 * 5 + 0 \\ 6 &= 1 * 6 + 0\end{aligned}$$

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

Model
Candidate 2

$$y = 2x + 2$$

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

Model
Candidate 2

$$y = 2x + 2$$

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

Model
Candidate 2

$$y = 2x + 2$$

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

Model
Candidate 2

$$y = 2x + 2$$

x	y	\hat{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

Cost

$$C(w, b)$$

Model

$$y_n = wx_n + b$$

Model Candidate 1

$$y = 1x + 0$$

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

Model Candidate 2

$$y = 2x + 2$$

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

Square Loss

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

Cost

Model Candidate 2

$$y = 2x + 2$$

x	y	\hat{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

$$y = 1x + 0$$

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

Cost

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

Model
Candidate 2

$$y = 2x + 2$$

x	y	\hat{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

$$y = 1x + 0$$

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

Cost

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

Model Candidate 2

$$y = 2x + 2$$

x	y	\hat{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

$$y = 1x + 0$$

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

Cost

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

Model
Candidate 2

$$y = 2x + 2$$

x	y	\hat{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

$$y = 1x + 0$$

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

Cost

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

Model
Candidate 2

$$y = 2x + 2$$

x	y	\hat{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

$$y = 1x + 0$$

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

Cost

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

Model Candidate 2

$$y = 2x + 2$$

x	y	\hat{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	$= 1x + 0$

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

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

Model Candidate 2	
y	$= 2x + 2$

n	x	y	\hat{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

$$y_n = wx_n + b$$

Model
Candidate 1

$$y = 1x + 0$$

$$C(1,0) \quad 318$$

Cost

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

Model
Candidate 2



$$y = 2x + 2$$

$$C(2,2) \quad 68$$

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

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

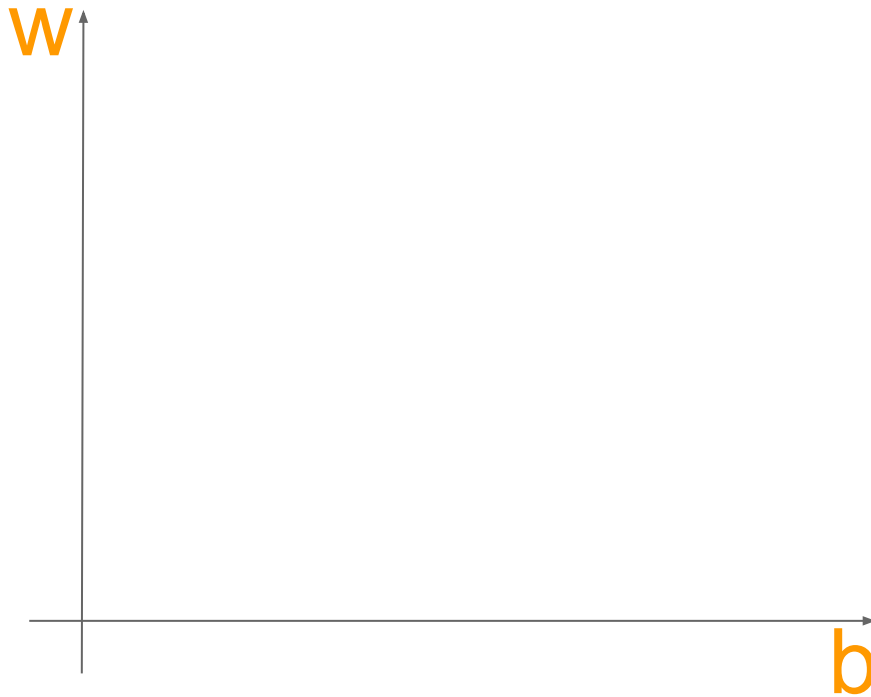
Optimizer

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

Optimizers are our friends

Optimizer

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



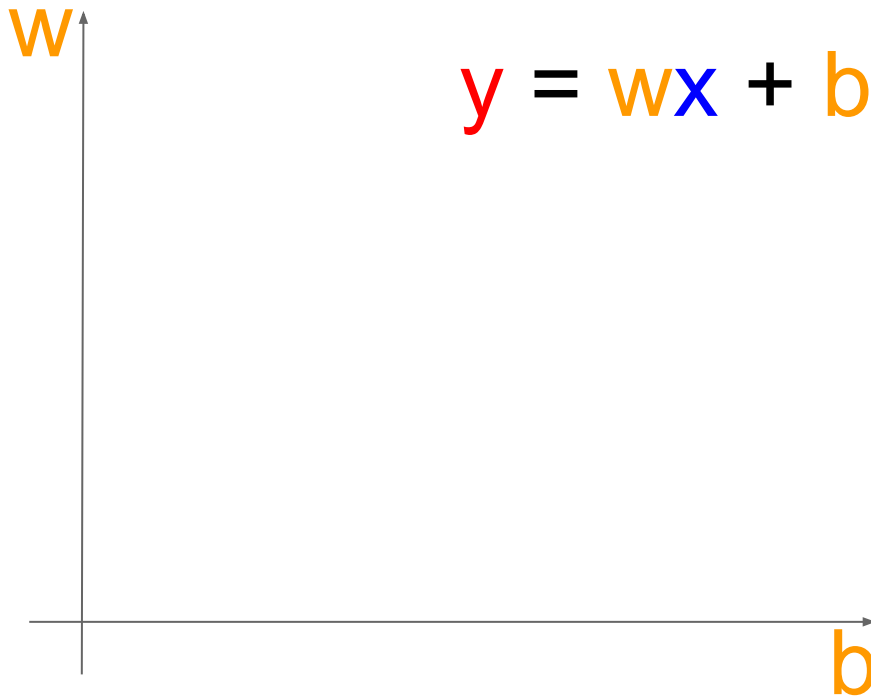
Optimizers are our friends

Optimizer

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

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

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



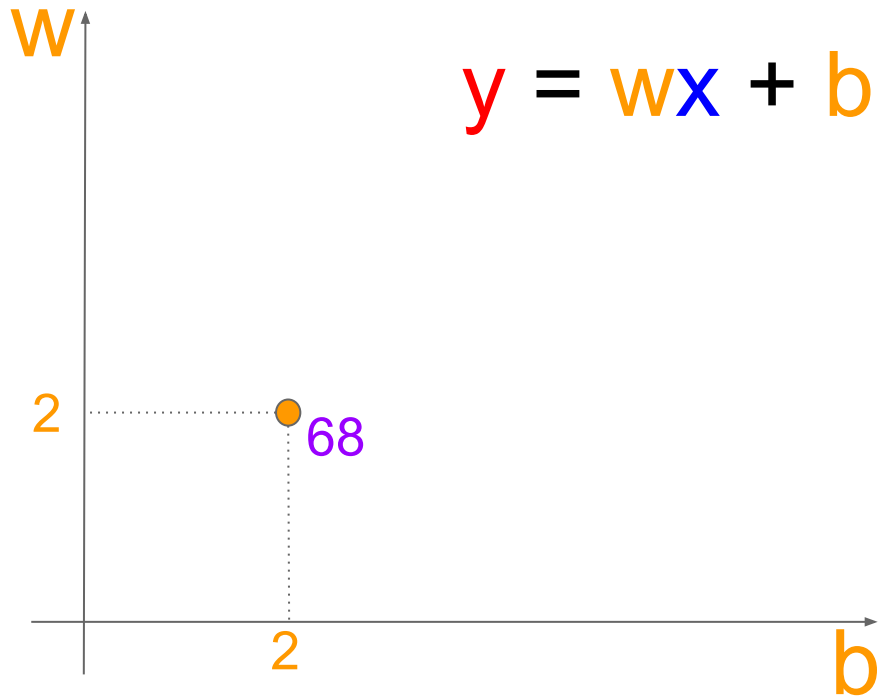
Optimizers are our friends

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

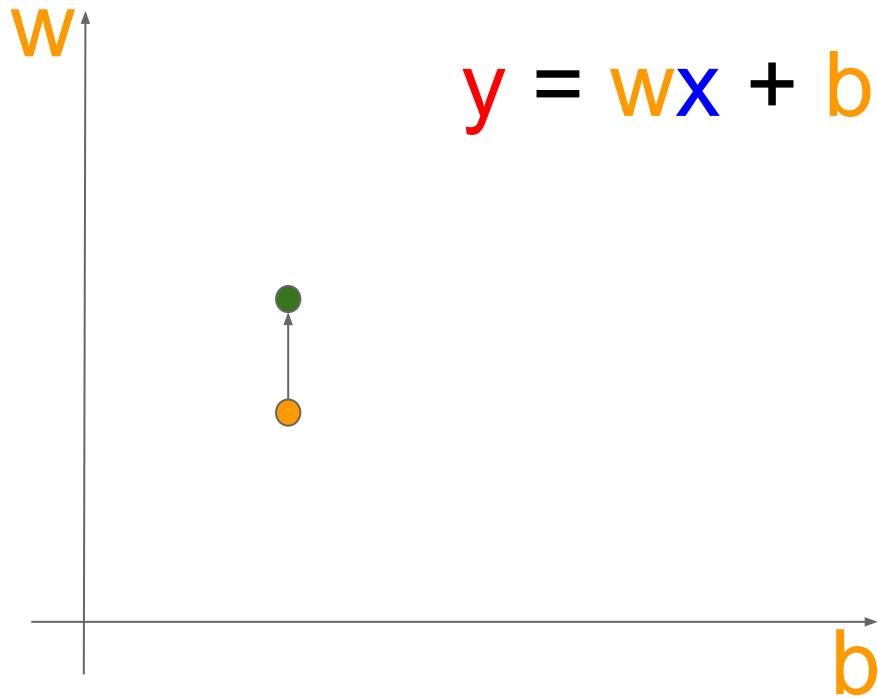
Optimizer

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

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

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

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



Optimizers are our friends

Optimizer

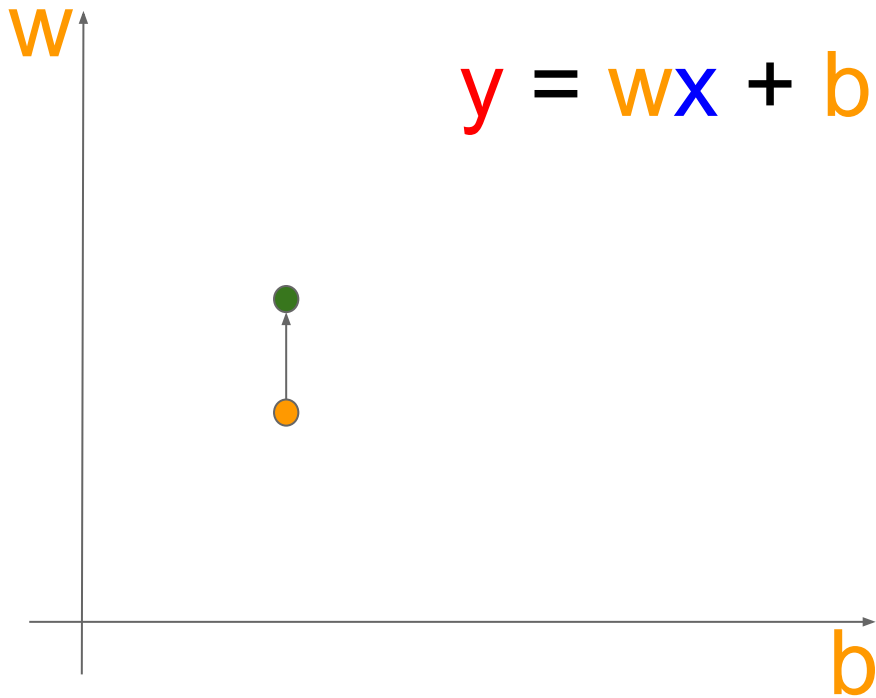
$$\arg \min C(w, b)$$

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

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

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

n	x	y	\hat{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

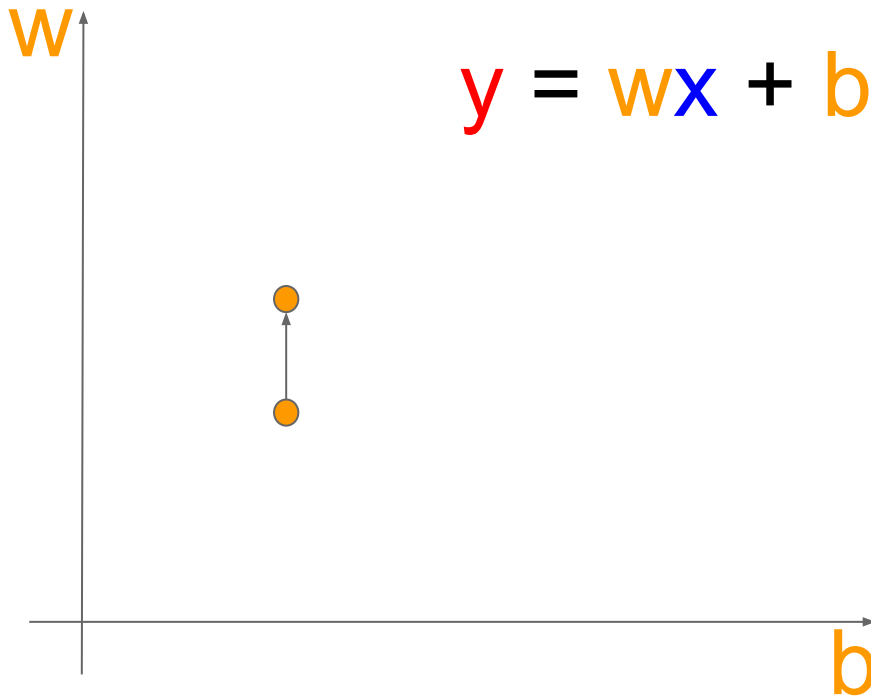
$$\arg \min C(w, b)$$

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

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

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

n	x	y	\hat{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

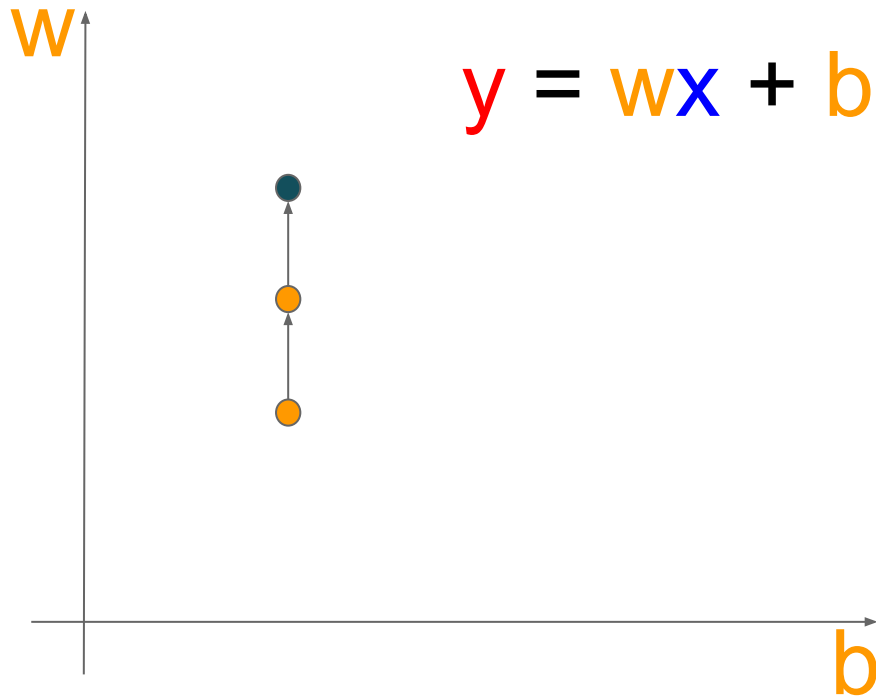
Optimizer

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

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

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

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



Optimizers are our friends

Optimizer

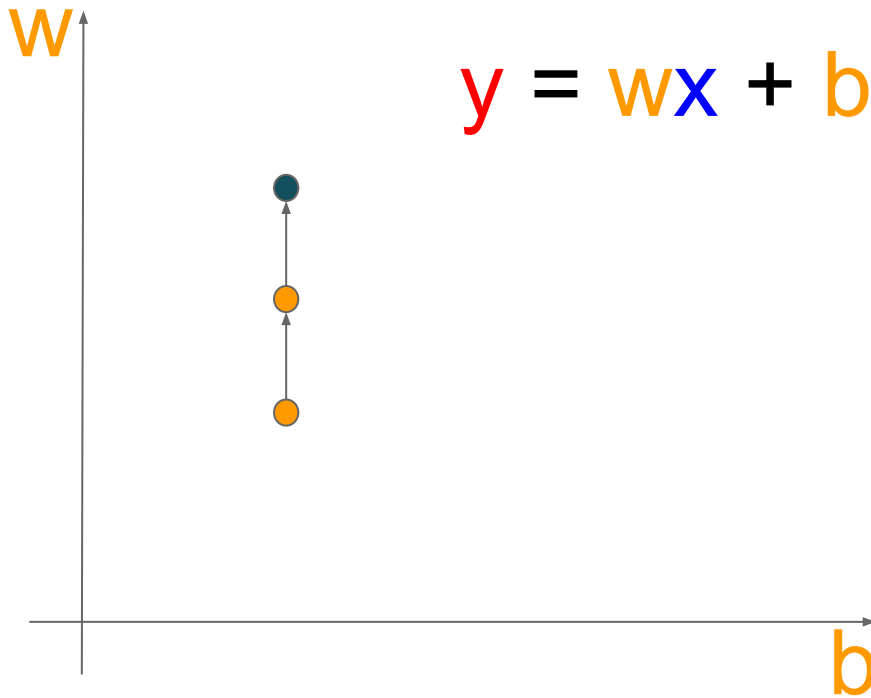
$$\arg \min C(w, b)$$

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

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

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

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



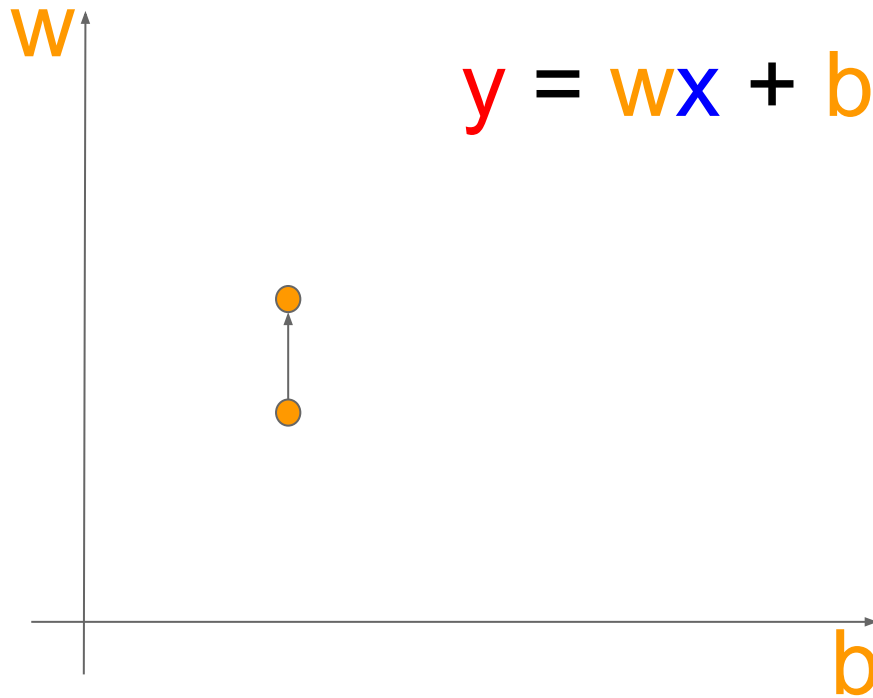
Optimizers are our friends

Optimizer

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

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

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



Optimizers are our friends

Optimizer

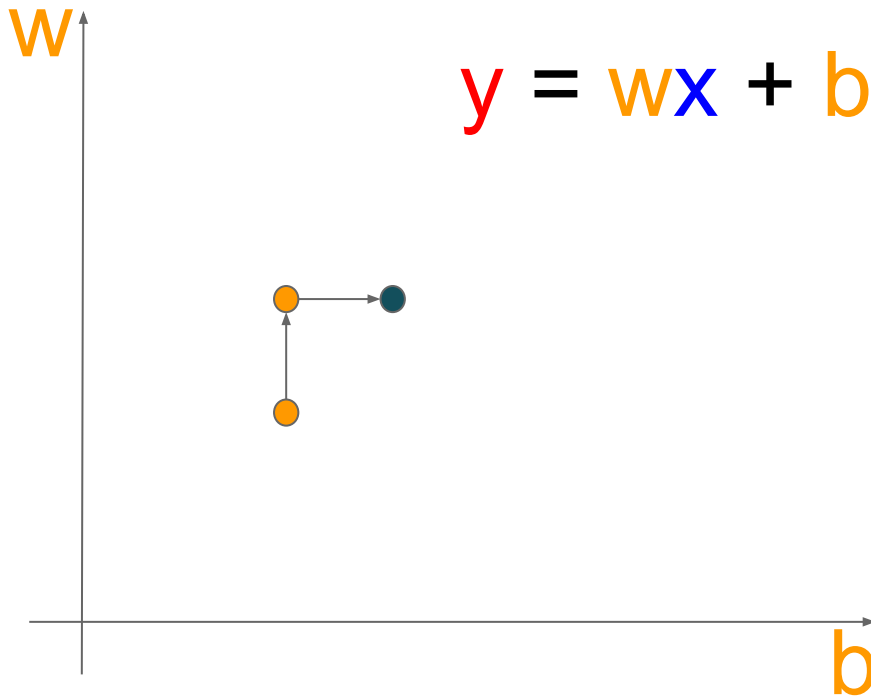
$$\arg \min C(w, b)$$

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

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

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

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



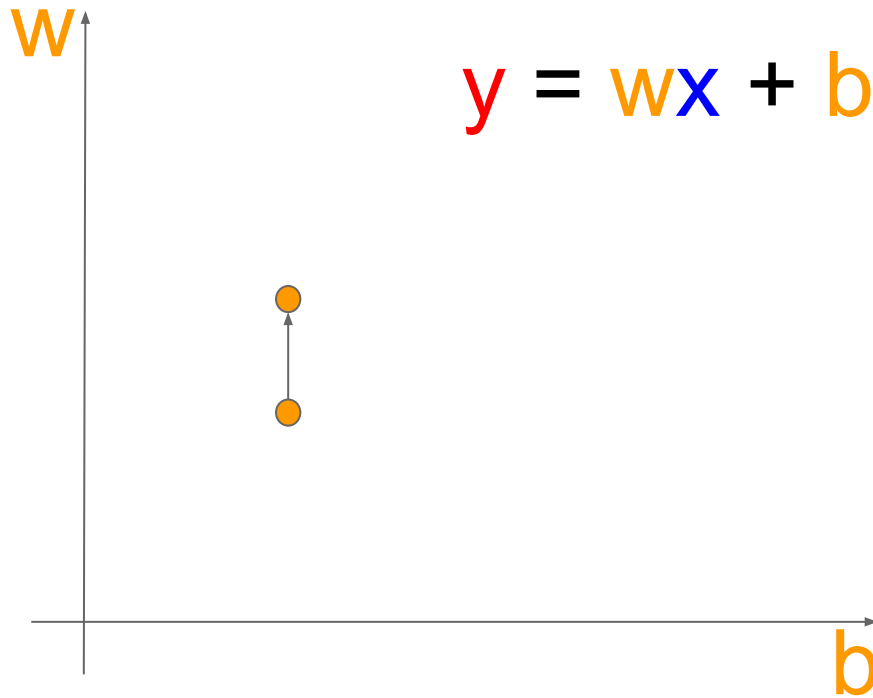
Optimizers are our friends

Optimizer

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

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

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



Optimizers are our friends

Optimizer

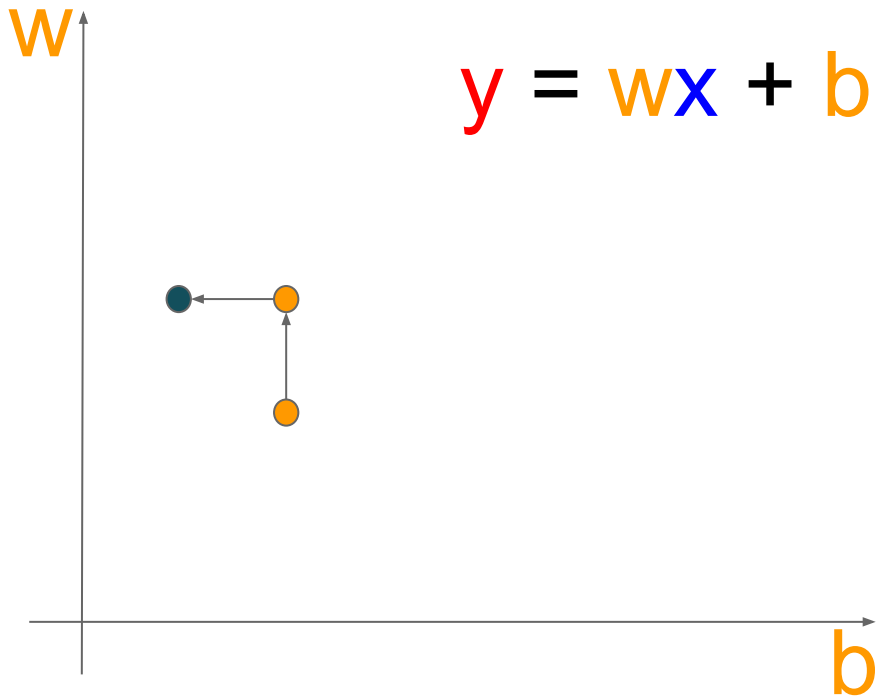
$$\arg \min C(w, b)$$

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

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

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

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



$$y = wx + b$$

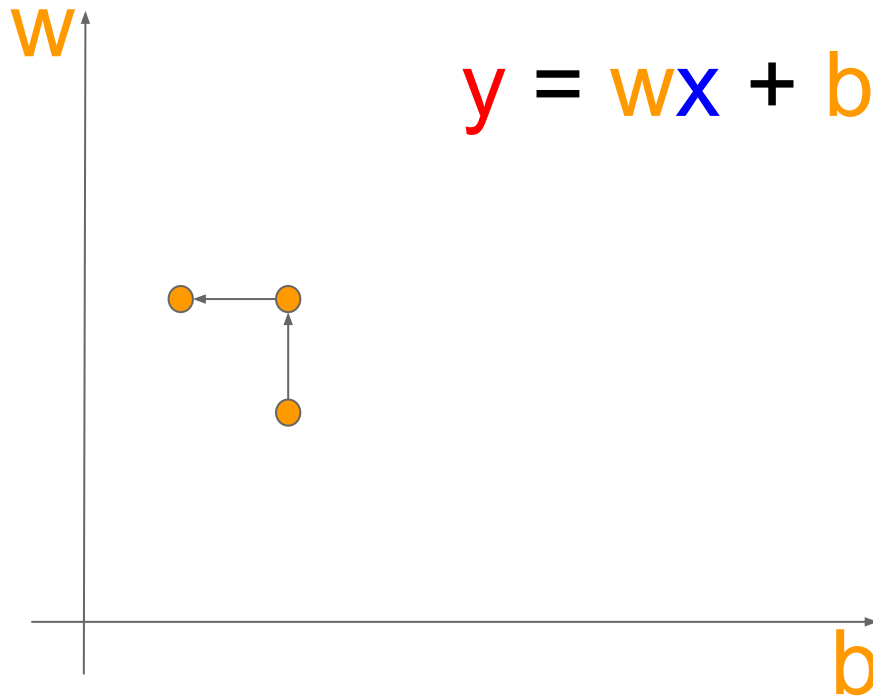
Optimizers are our friends

Optimizer

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

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

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



Optimizers are our friends

Optimizer

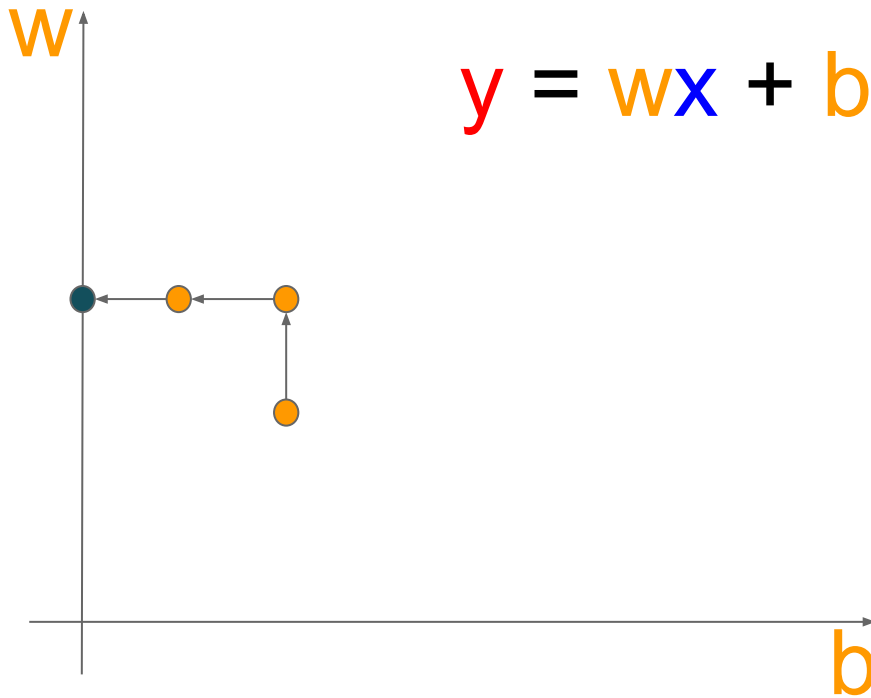
$$\arg \min C(w, b)$$

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

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

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

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



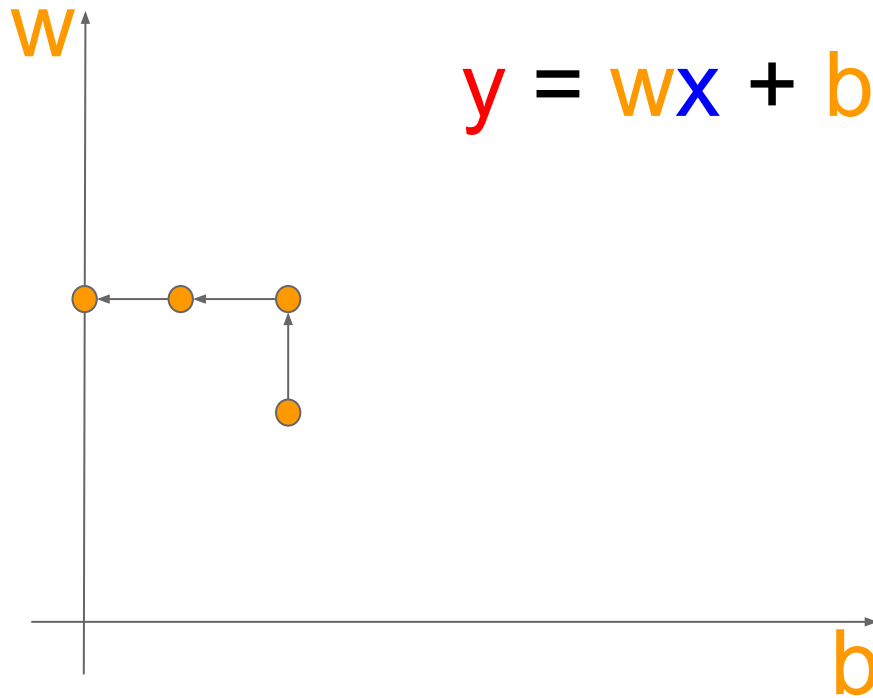
Optimizers are our friends

Optimizer

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

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

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



Optimizers are our friends

Optimizer

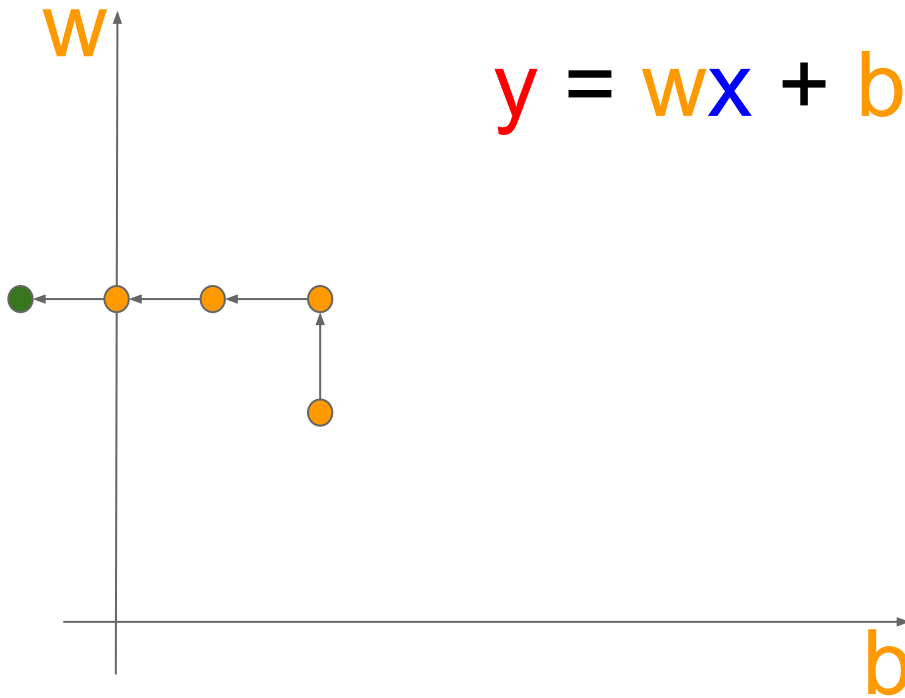
$$\arg \min C(w, b)$$

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

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

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

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



Optimizers are our friends

Optimizer

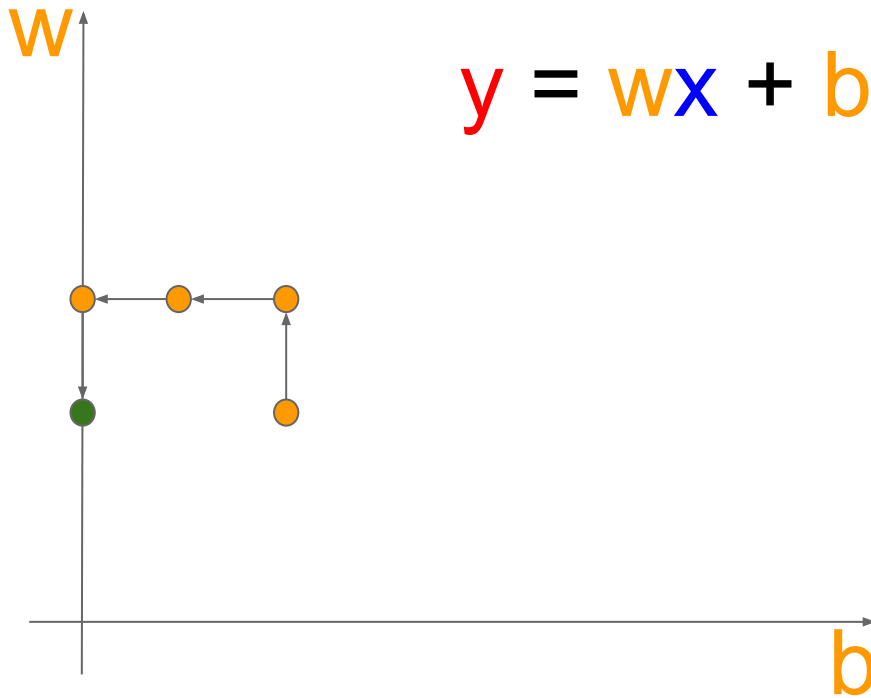
$$\arg \min C(w, b)$$

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

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

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

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



Optimizers are our friends

Optimizer

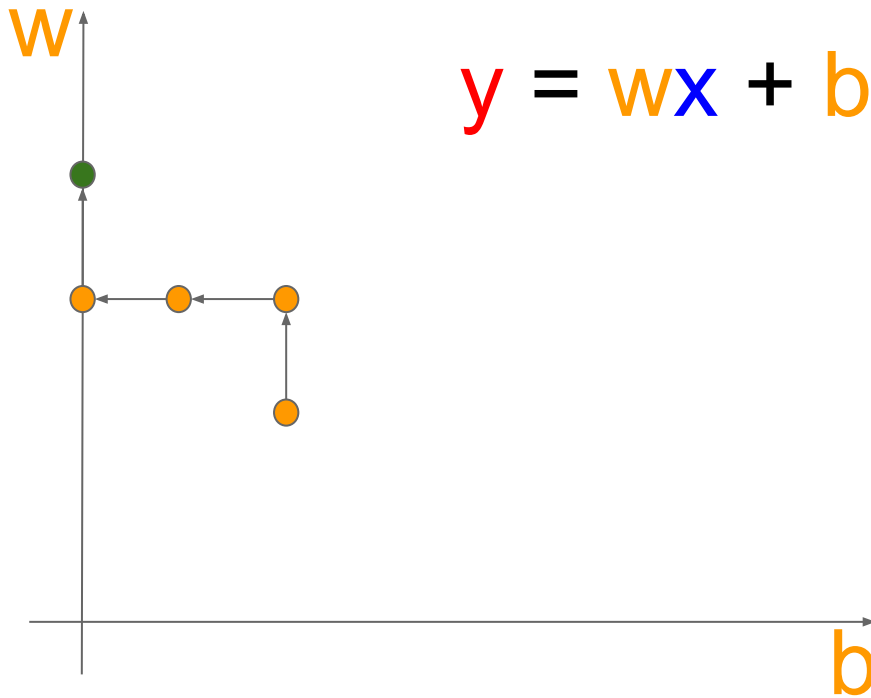
$$\arg \min C(w, b)$$

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

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

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

n	x	y	\hat{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

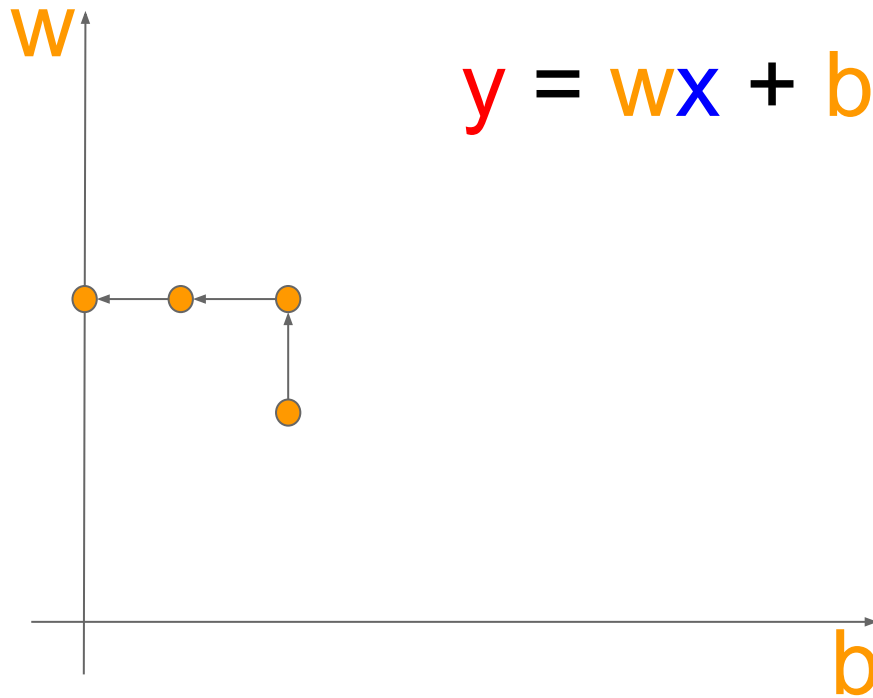
Optimizer

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

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

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

The End?



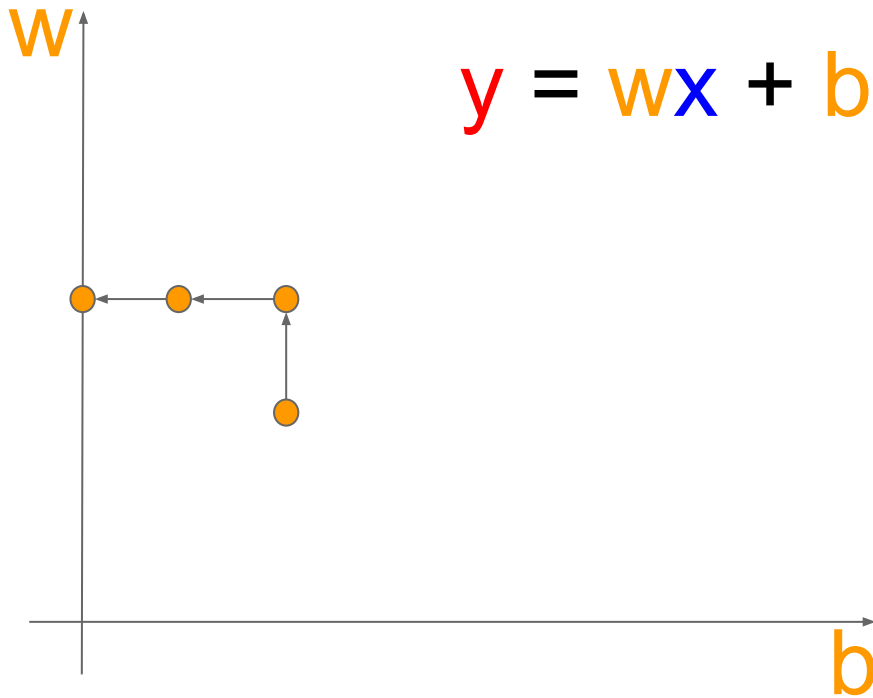
Optimizers are our friends

Optimizer

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

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

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



Optimizers are our friends

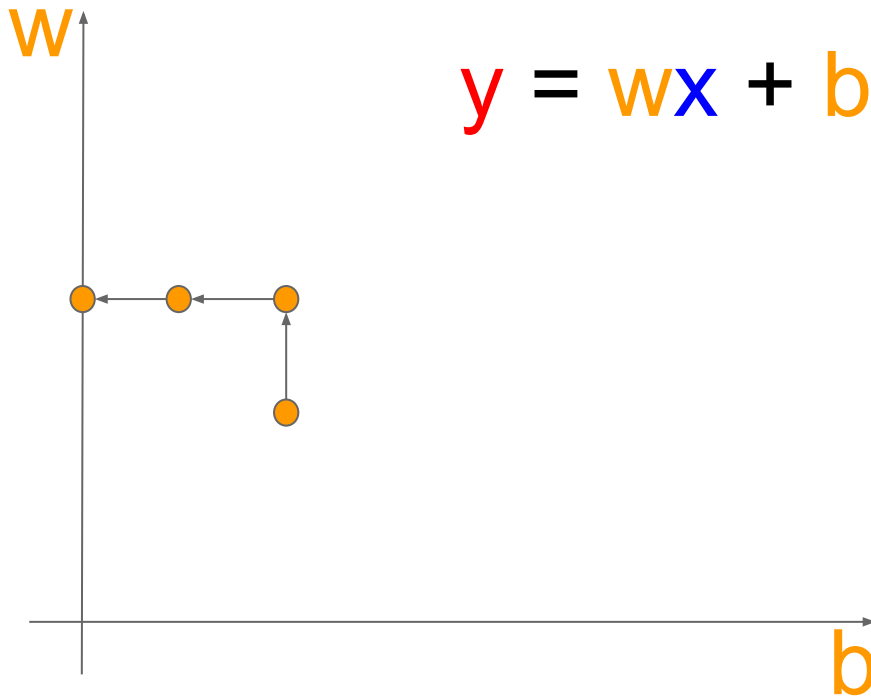
Optimizer

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

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

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

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



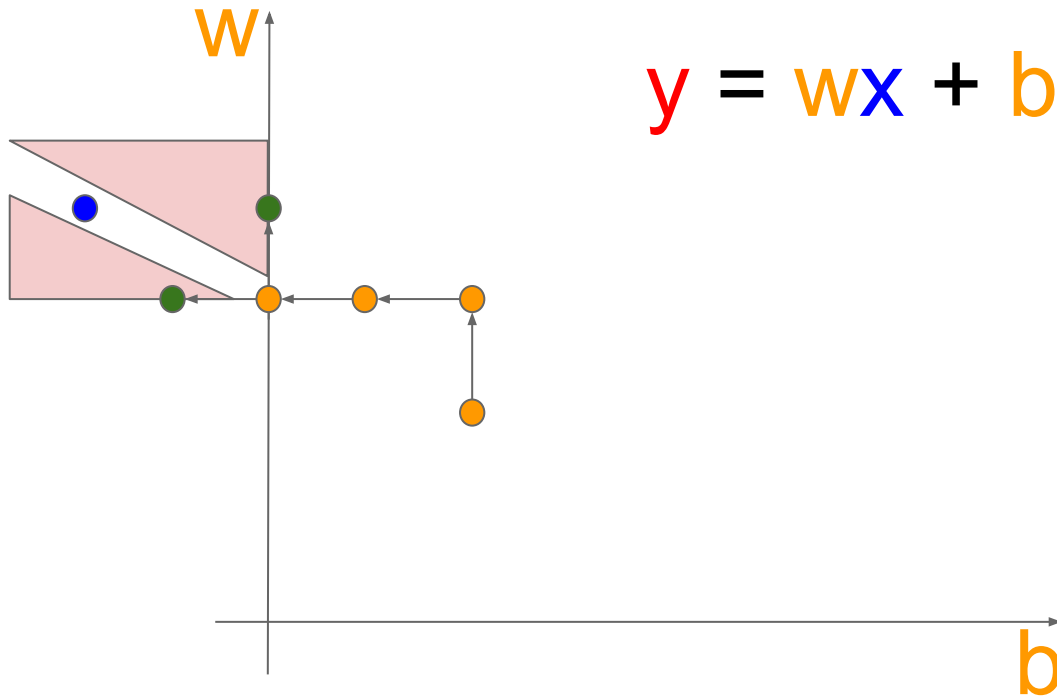
Optimizers are our friends

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

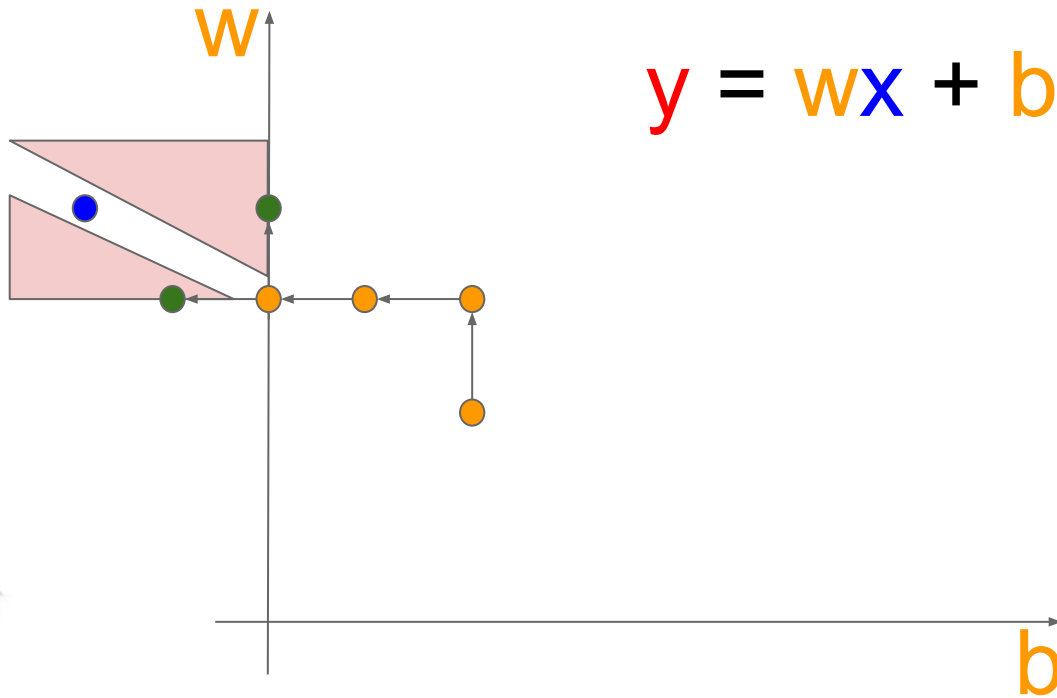
Optimizer

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

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

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

Search
Problem



$$y = wx + b$$

Optimizers are our friends

Optimizer

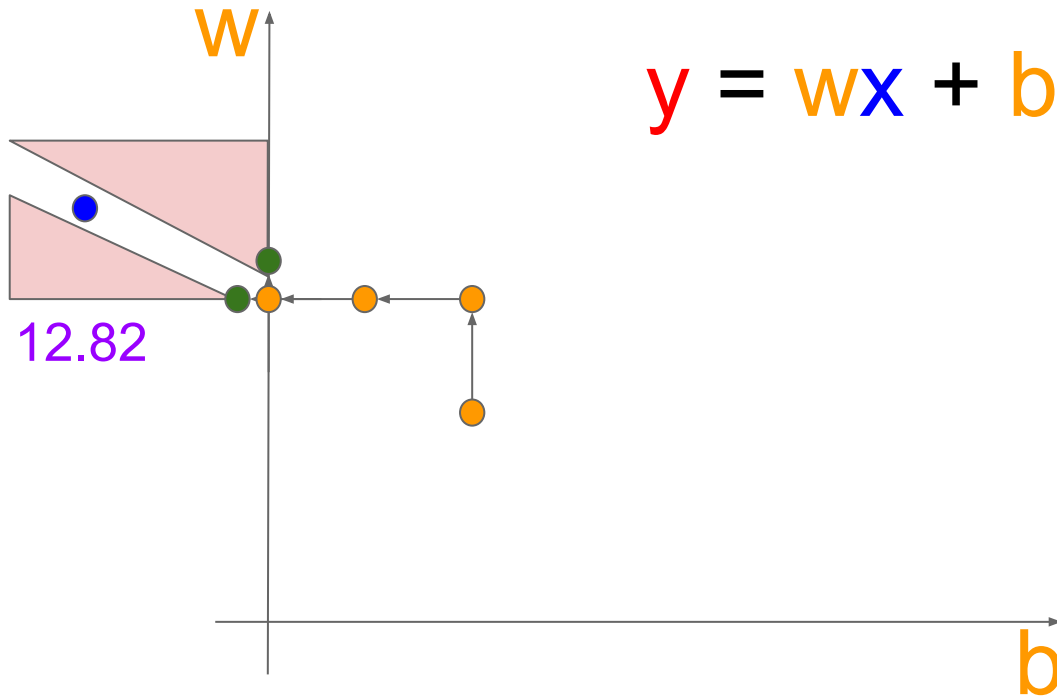
$$\arg \min C(w, b)$$

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

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

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

n	x	y	\hat{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



$$y = wx + b$$

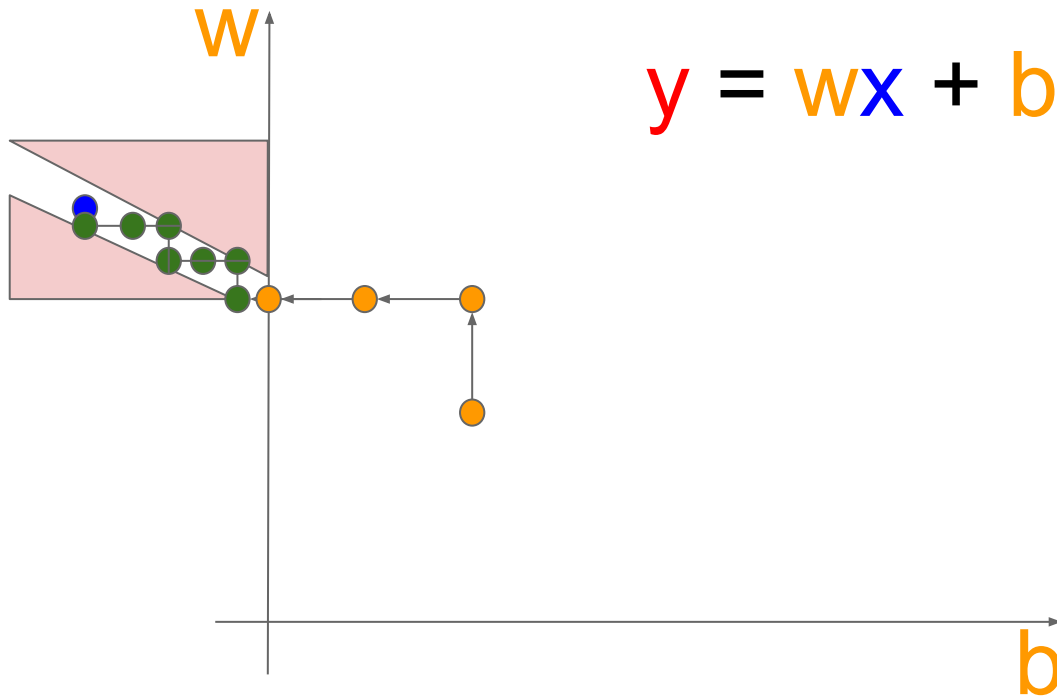
Optimizers are our friends

Optimizer

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

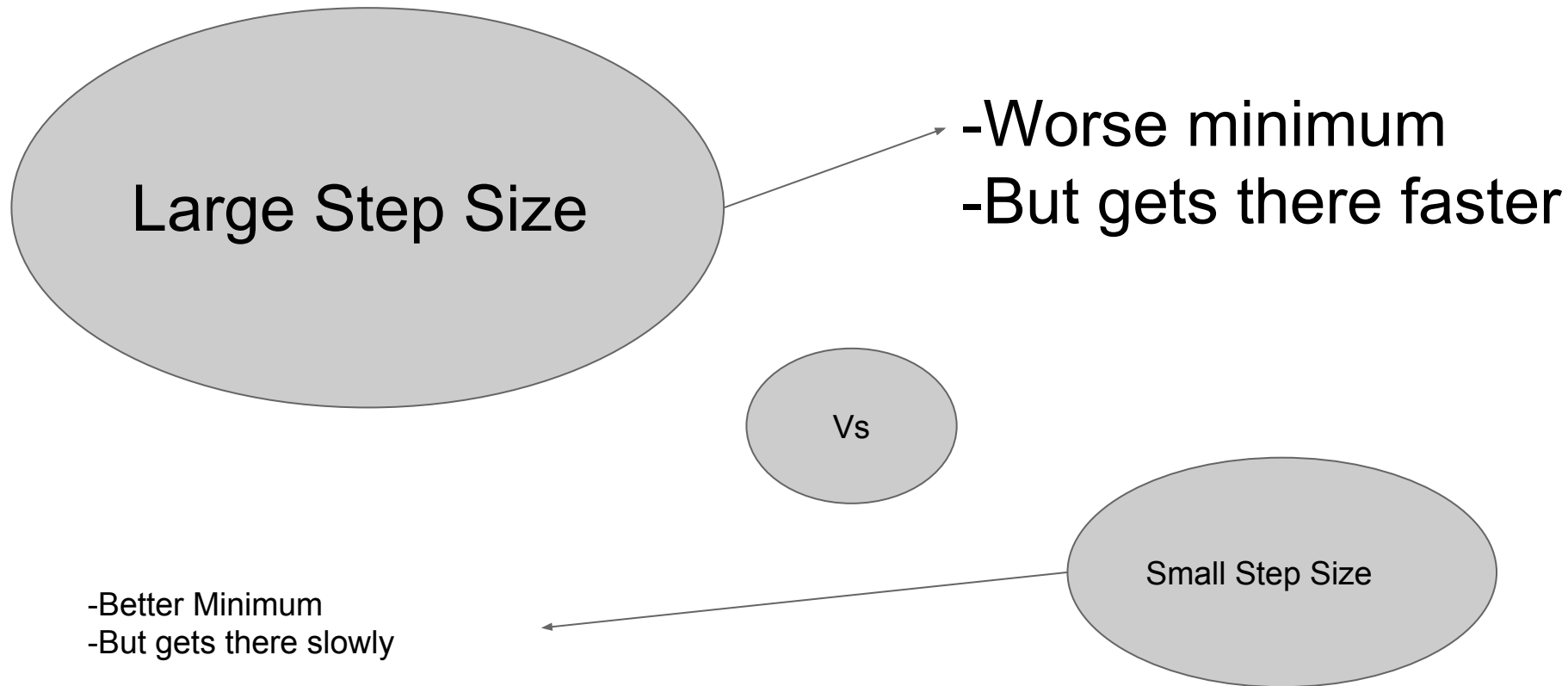
$$w, b \in [-\infty, \infty]$$

$$w^*, b^* = 4, -2 : C(w^*, b^*) = 12$$



$$y = wx + b$$

Optimizers are our friends



Optimizers are our friends



Step Size

Step Size

Step Size

Step Size

Step Size

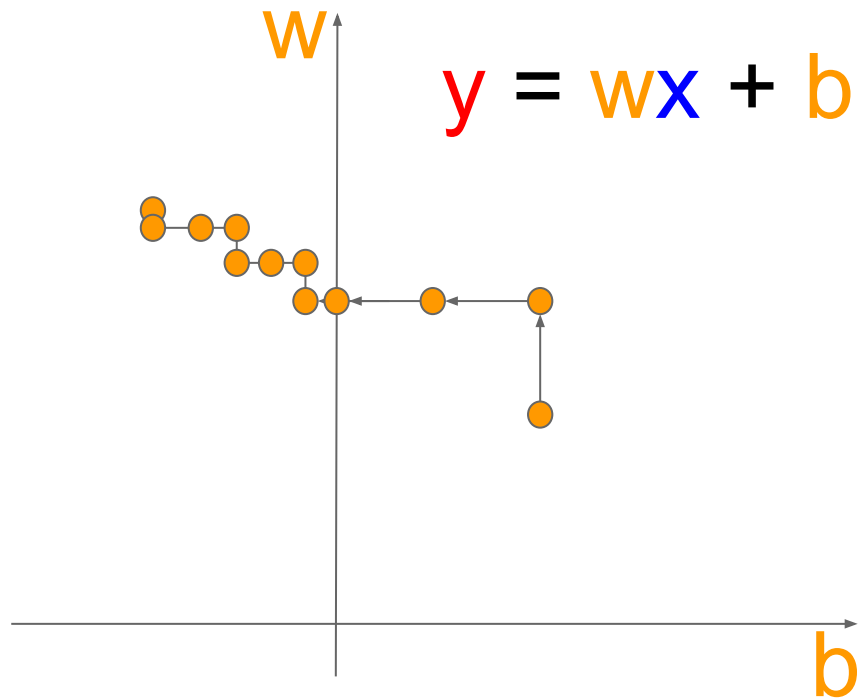
Optimizers are our friends

Optimizer

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

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

$$w^*, b^* = 4, -2 : C(w^*, b^*) = 12$$



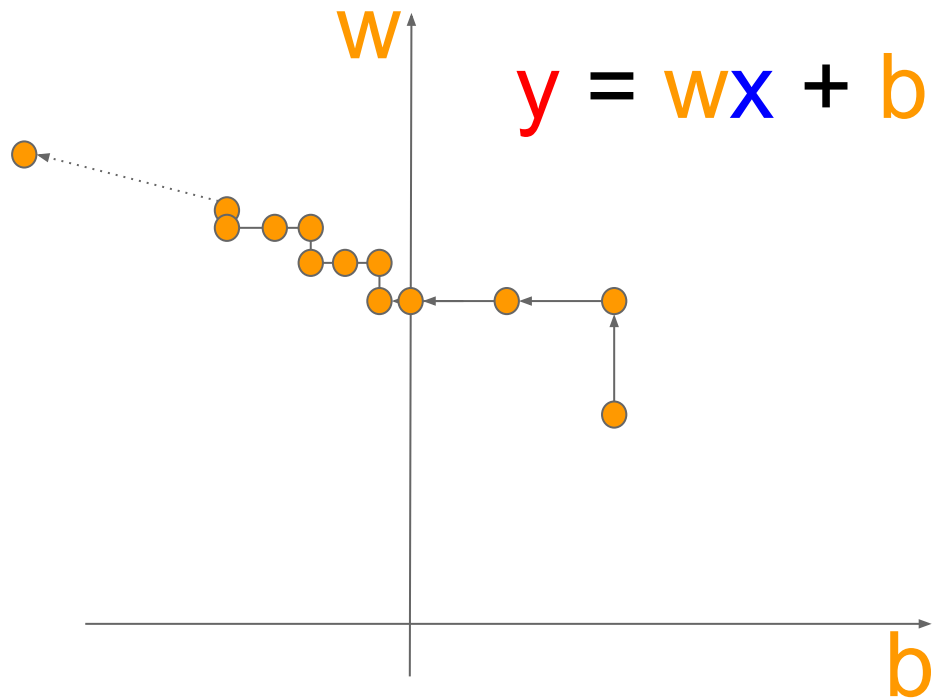
Optimizers are our friends

Optimizer

$$\arg \min C(\mathbf{w}, \mathbf{b})$$

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

$$w^*, b^* = 4, -4 : C(w^*, b^*) = 0$$



Optimizers are our friends

$$y = wx + b$$



Data	
x	y
1	0
5	16
6	20



Optimizers are our friends

$$y = 4x - 4$$



Data	
x	y
1	0
5	16
6	20



Optimizers are our friends

$$y = 4x - 4$$



Data	
x	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

High
if cat

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

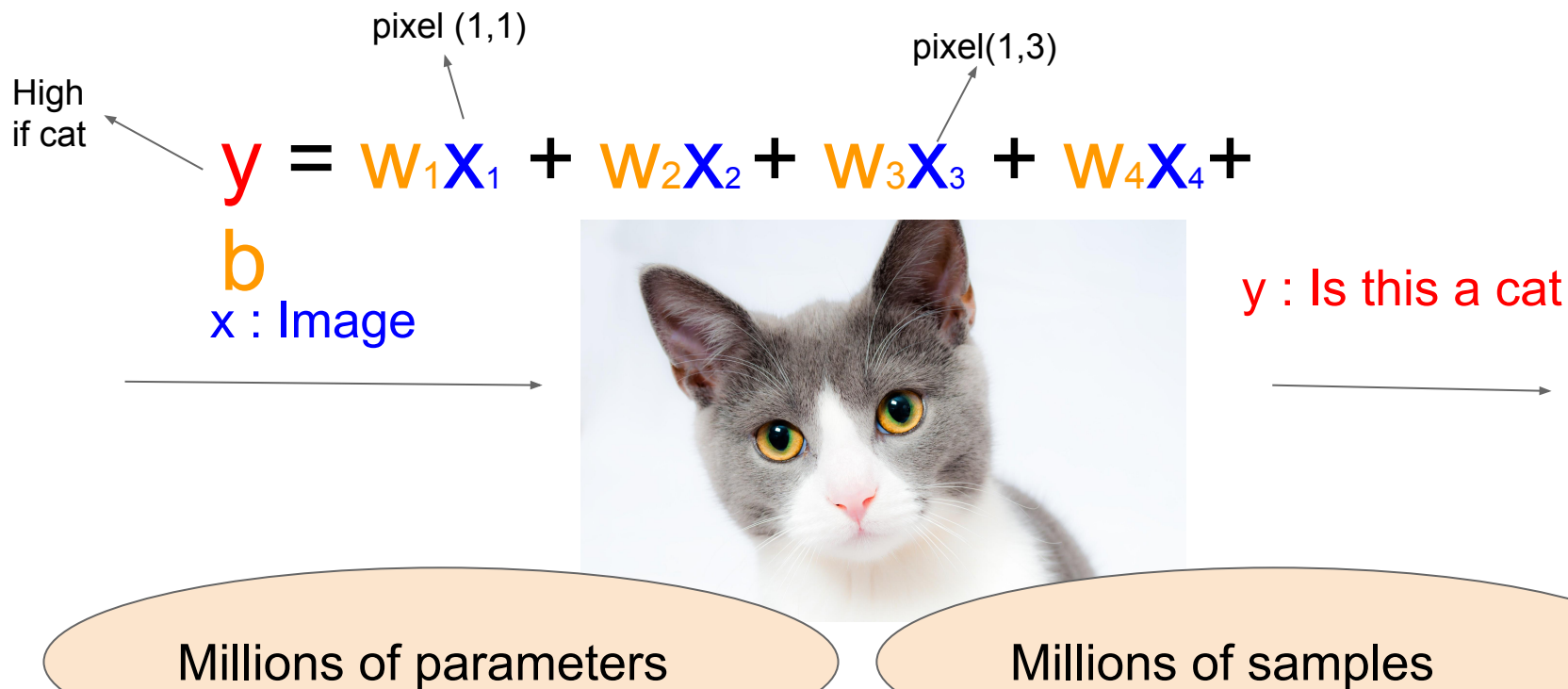
pixel (1,1) pixel(1,3)

b
 x : Image



y : Is this a cat

Functions are our friends



Gradients are our friends

Optimizer

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

Very expensive
to compute
(hours or days)

w

$$y = wx + b$$

b

Gradients are our friends

Optimizer

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

Should be used sparingly

w

$$y = wx + b$$

b

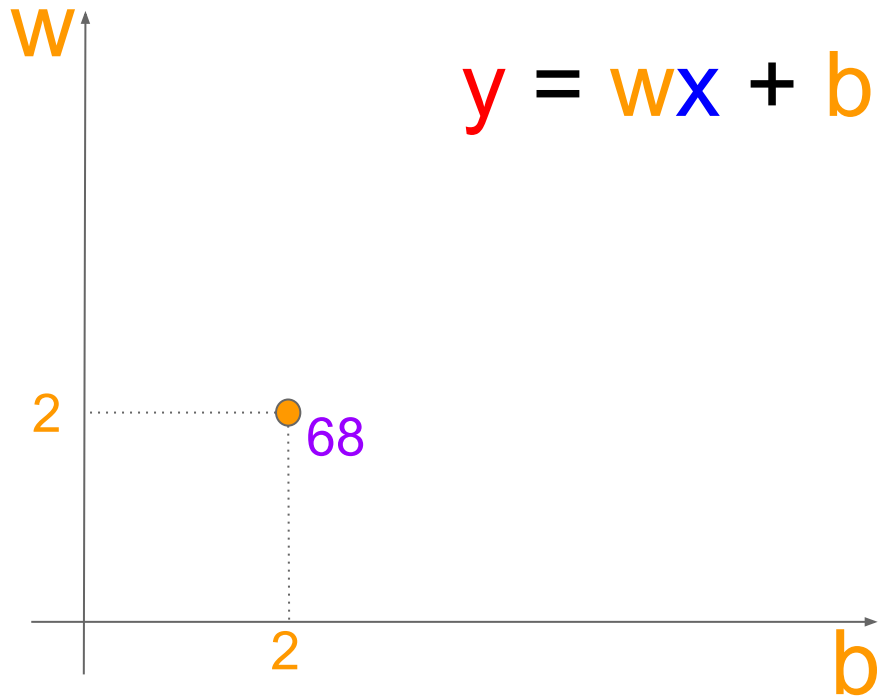
Gradients are our friends

Optimizer

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

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

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



Gradients are our friends

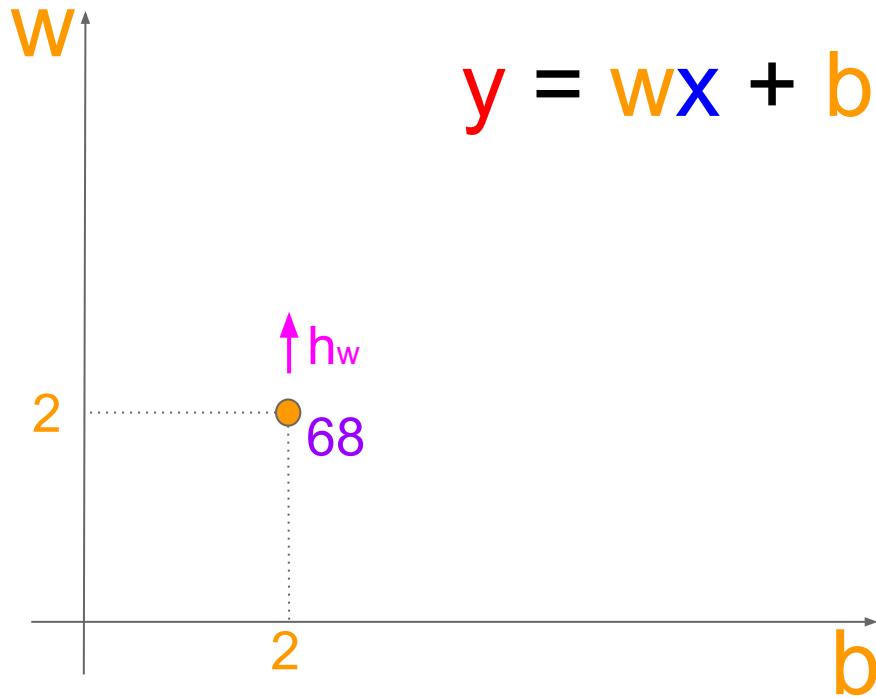
Optimizer

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

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

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

$$h_w = 1$$



Gradients are our friends

Optimizer

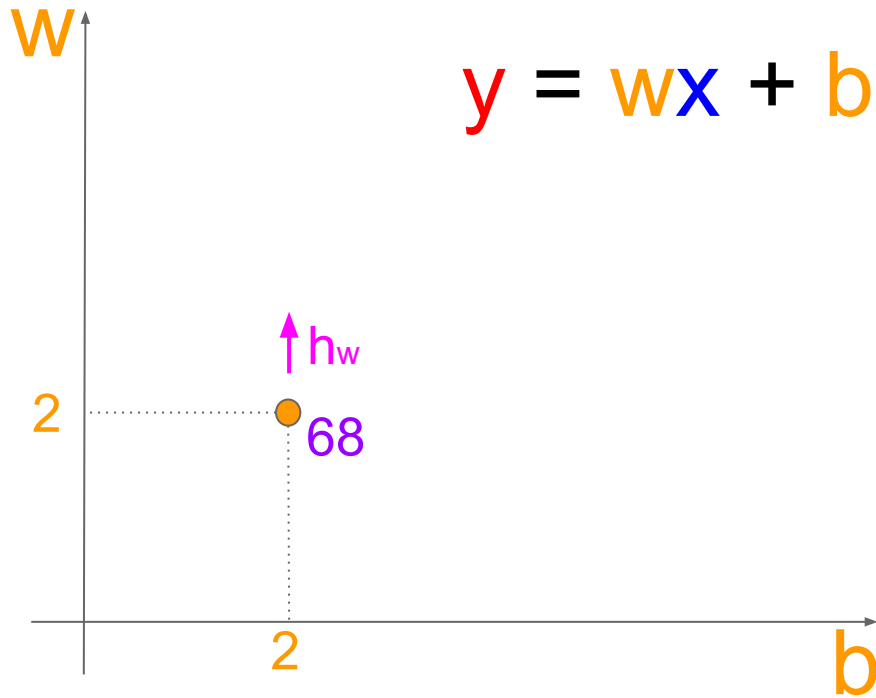
$$\arg \min C(w, b)$$

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

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

$$h_w = 1$$

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



$$y = wx + b$$

Gradients are our friends

Optimizer

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

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

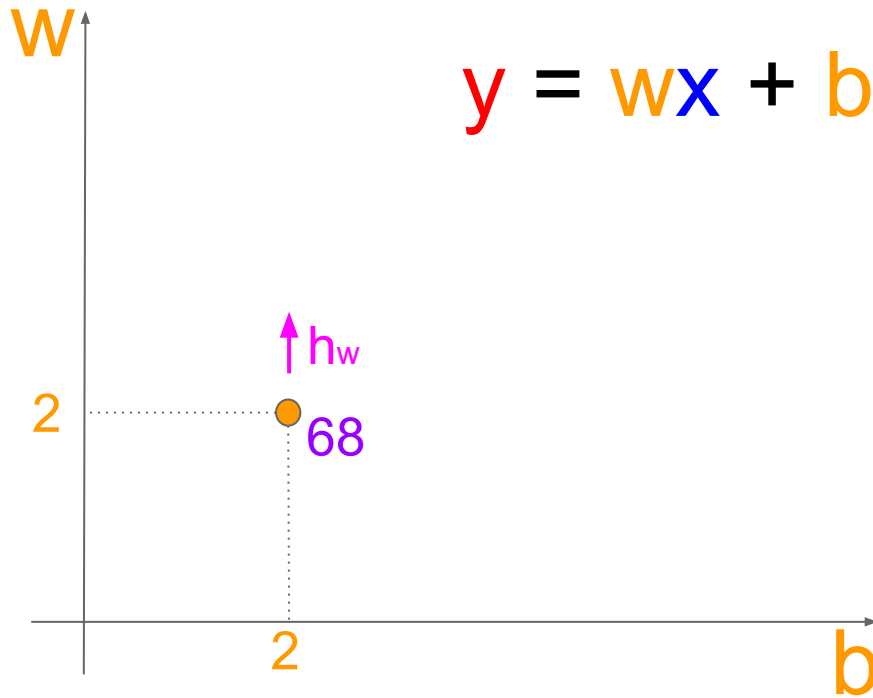
$$w_0, b_0 = 2, 2 : C(w_0, b_0) = 68$$

$$h_w = 1$$

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

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

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



$$y = wx + b$$

Gradients are our friends

Optimizer

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

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

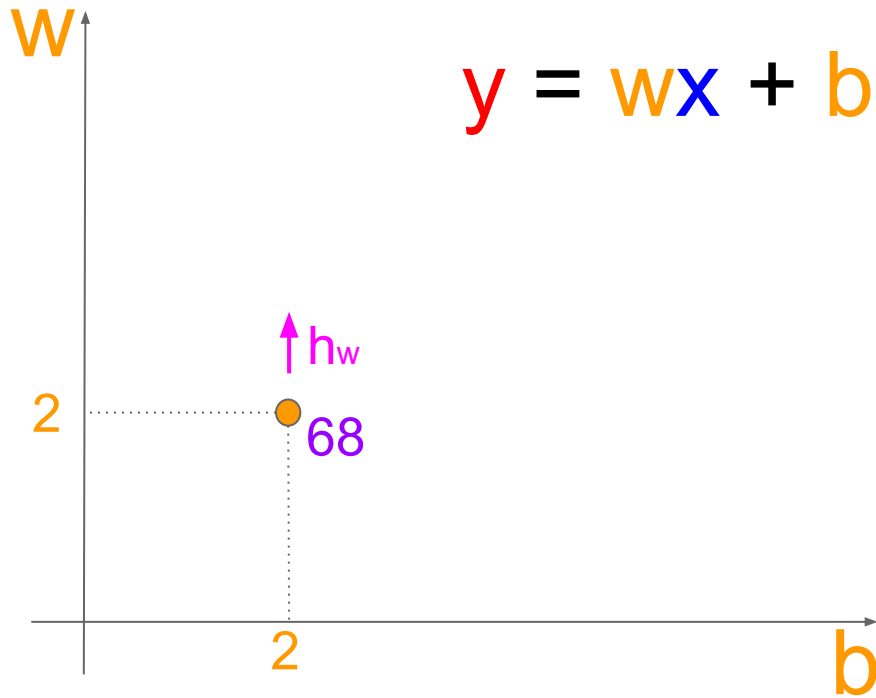
$$w_0, b_0 = 2, 2 : C(w_0, b_0) = 68$$

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

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

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

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



Gradients are our friends

Optimizer

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

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

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

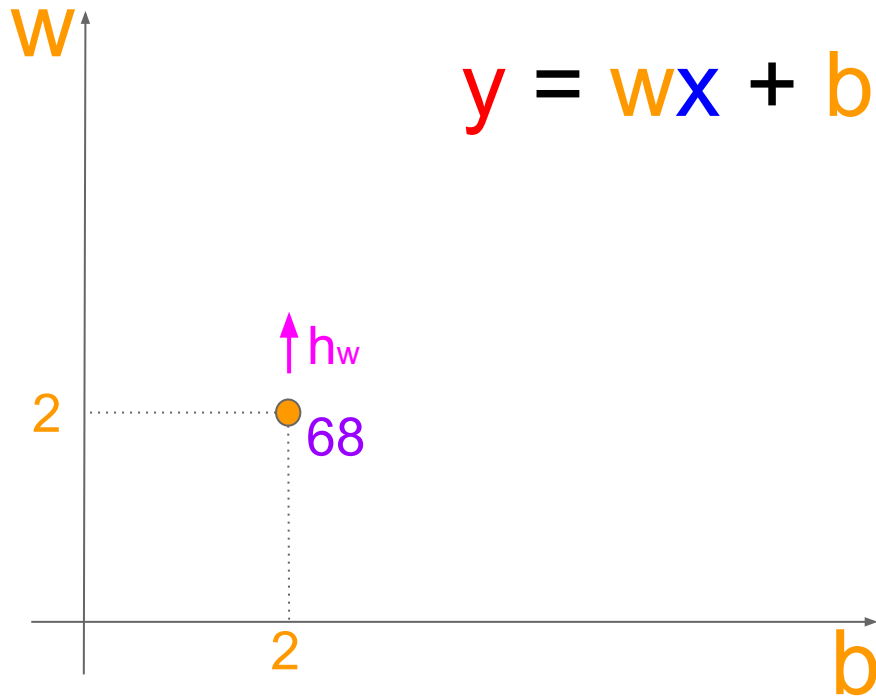
$$h_w = 1, r = -42$$

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

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

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

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



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

Gradients are our friends

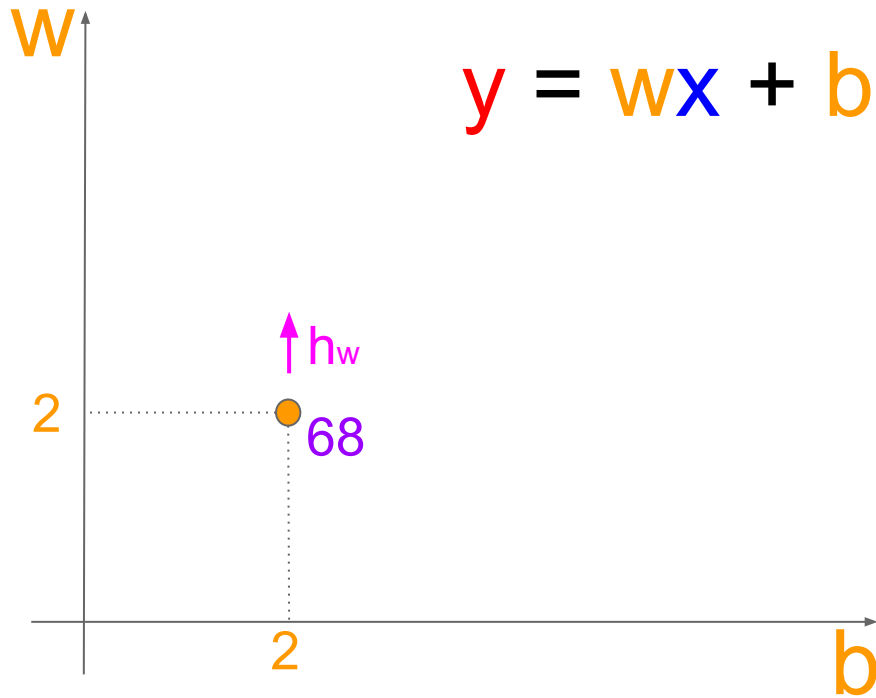
Optimizer

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

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

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

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



Gradients are our friends

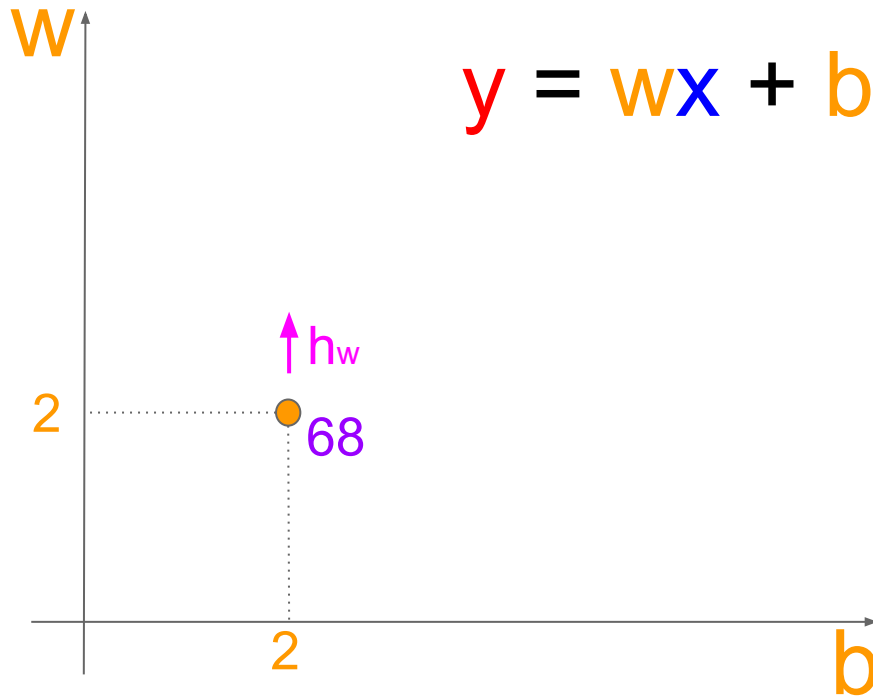
Optimizer

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

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

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

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



Gradients are our friends

Optimizer

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

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

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

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

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

n	x	y	\hat{y}	$(\hat{y} - y)$	$2(\hat{y} - y)x$
0	1	0	4	4	8
1	5	16	12	-4	-40
2	6	20	14	-6	-72

Gradients are our friends

Optimizer

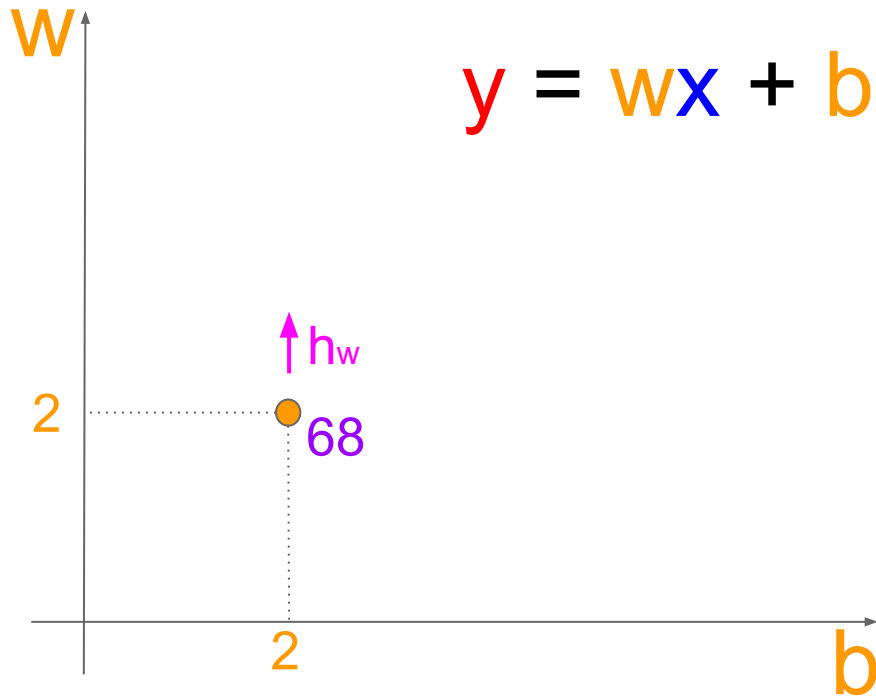
$$\arg \min C(w, b)$$

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

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

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

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



Gradients are our friends

Optimizer

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

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

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

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

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

n	x	y	\hat{y}	$(\hat{y}-y)$	$2(\hat{y}-y)$
0	1	0	4	4	8
1	5	16	12	-4	-8
2	6	20	14	-6	-12

Gradients are our friends

Optimizer

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

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

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

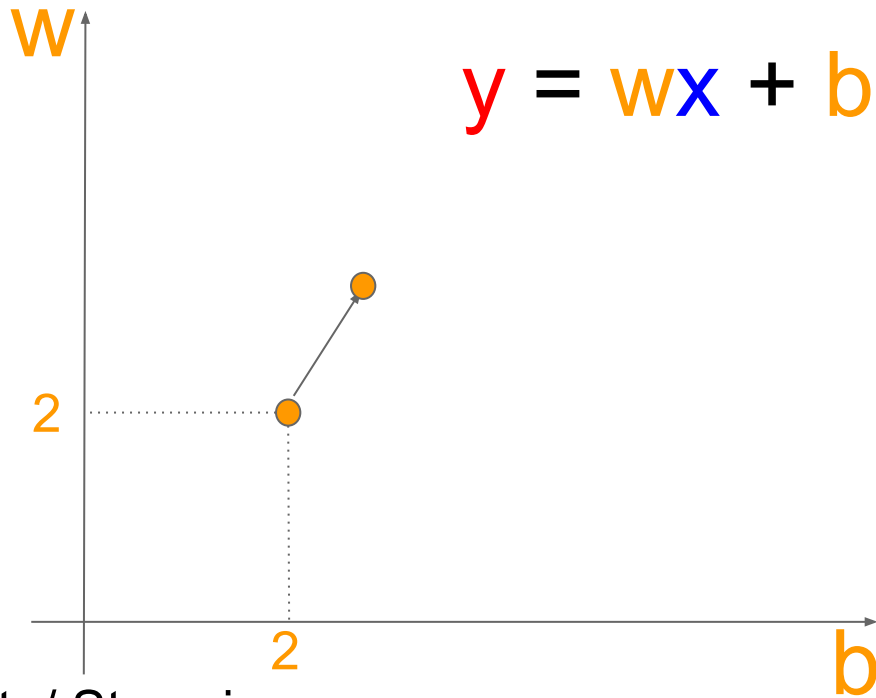
$$h_w \rightarrow 0, r_w = \frac{\partial C}{\partial w}(w_0, b_0) = -104$$

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

$$w_1 = w_0 - r_w a$$

$$b_1 = b_0 - r_b a$$

$a \rightarrow$ Learning Rate/ Step size



Summary

Data

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

Model

$$y_n = wx_n + b$$

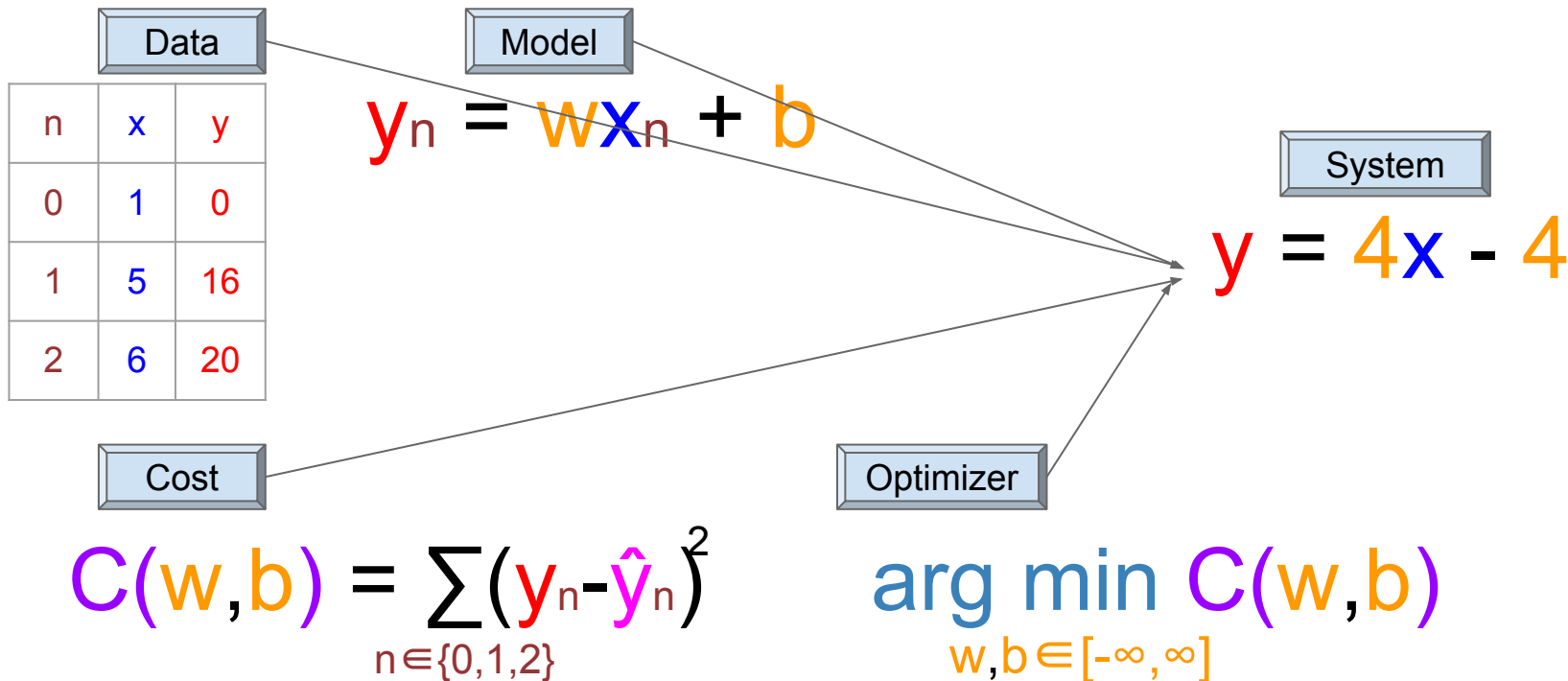
Cost

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

Optimizer

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

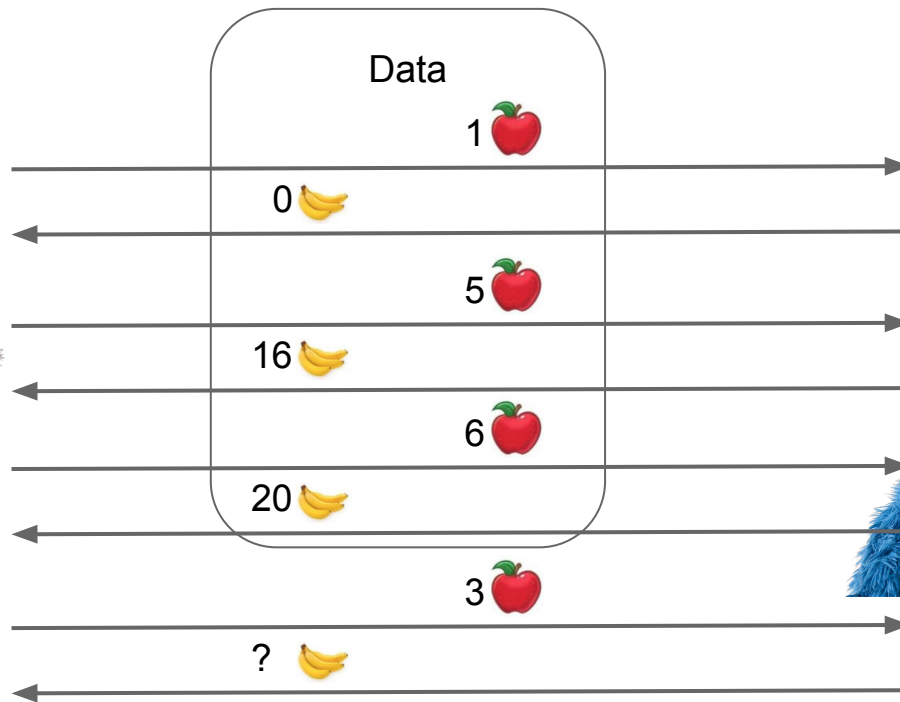
Summary



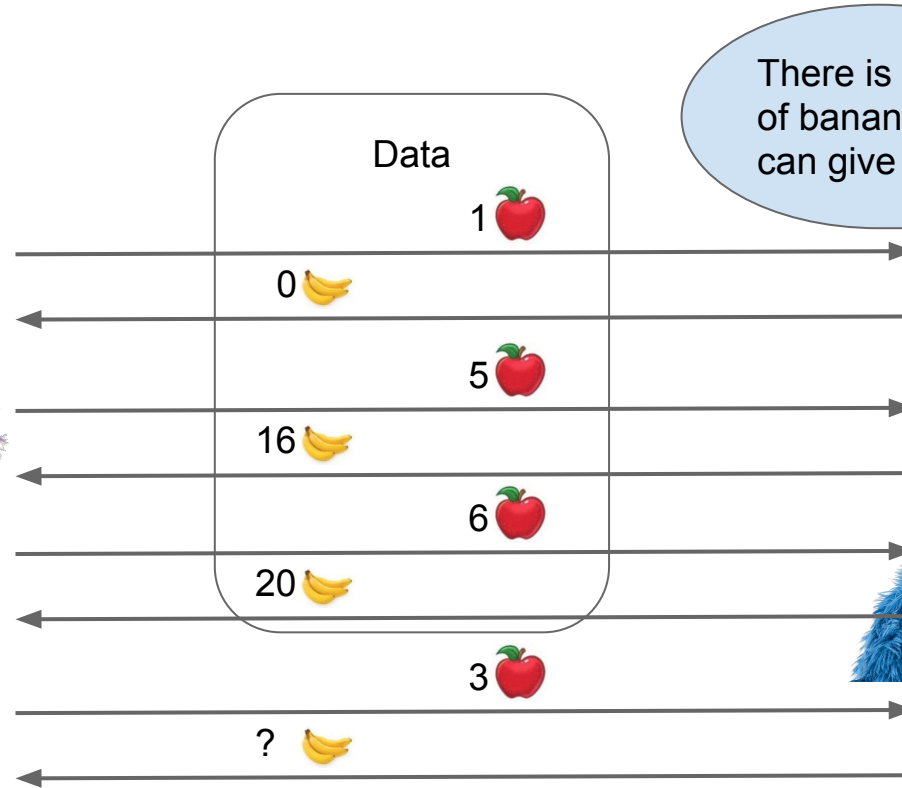
Into Deep Learning

Nonlinear Neural Models

$$y = 4x - 4$$



Nonlinear Neural Models

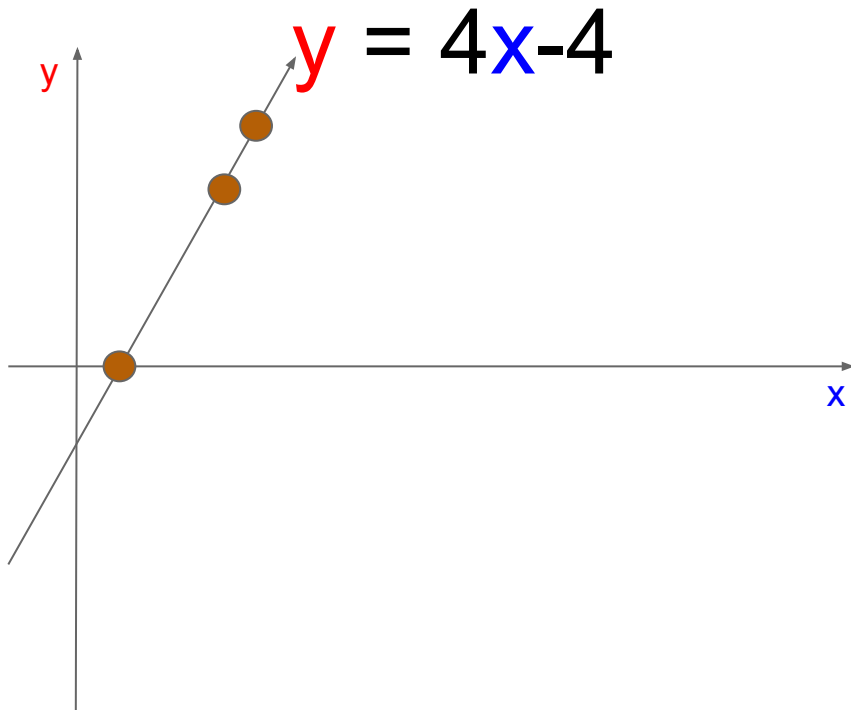


There is a limit of bananas I can give you



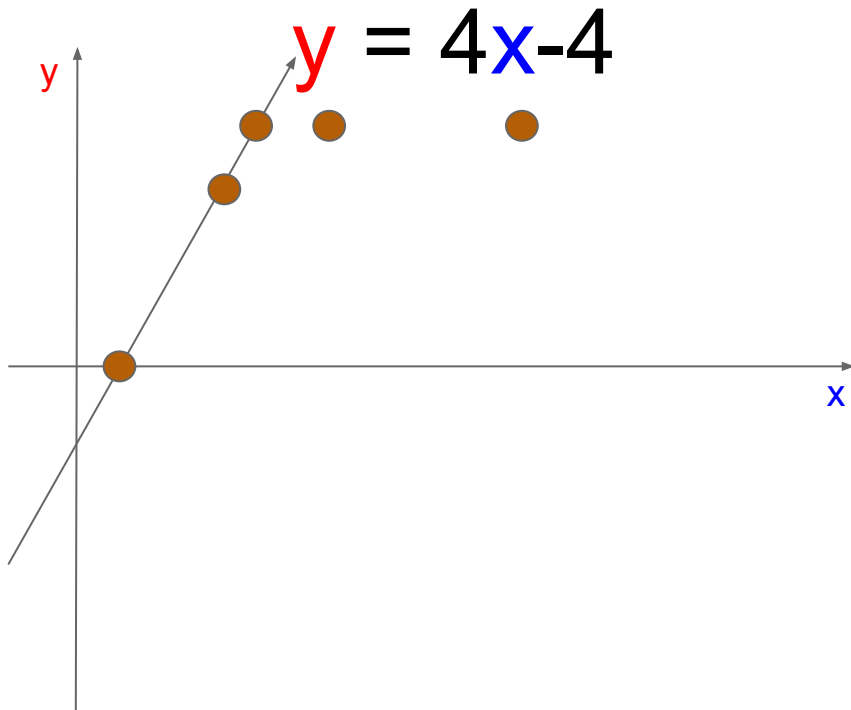
Nonlinear Neural Models

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



Nonlinear Neural Models

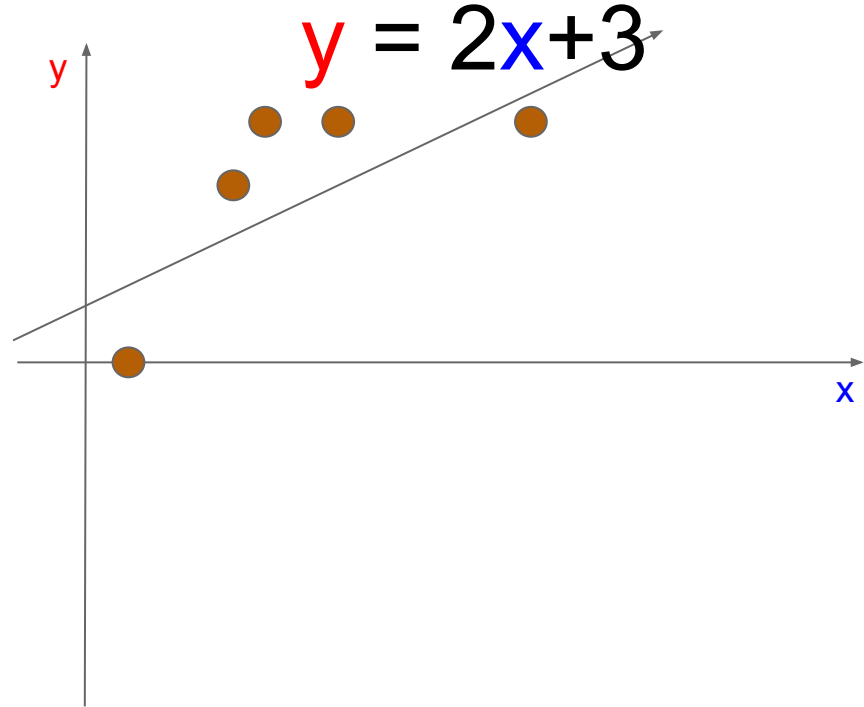
Data		
n	x	y
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20



Nonlinear Neural Models

Data		
n	x	y
0	1	0
1	5	16
2	6	20
3		
4		

Model
Problem

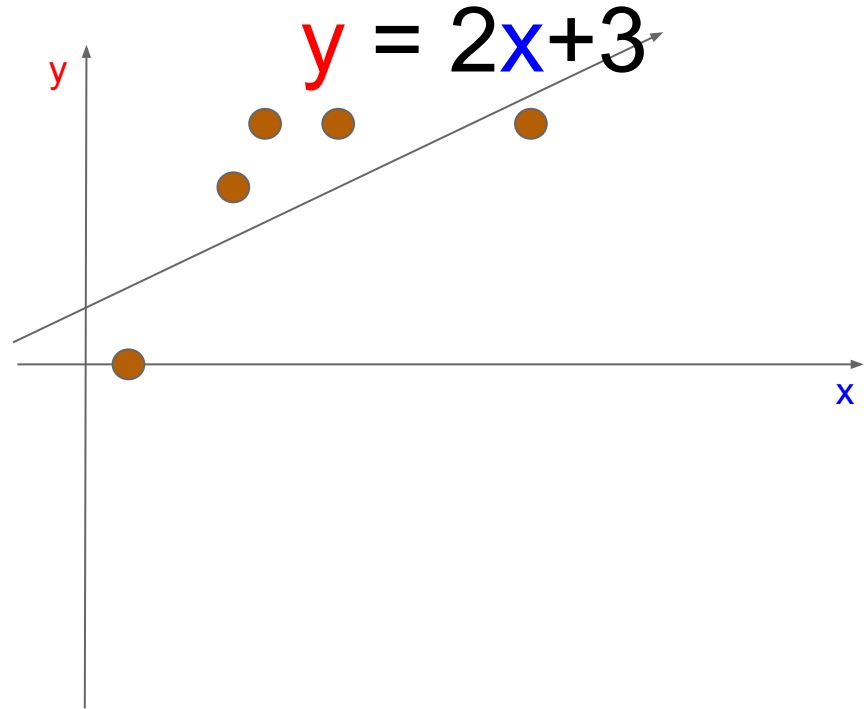


Nonlinear Neural Models

Data		
n	x	y
0	1	0
1	5	16
2	6	20
3		
4		

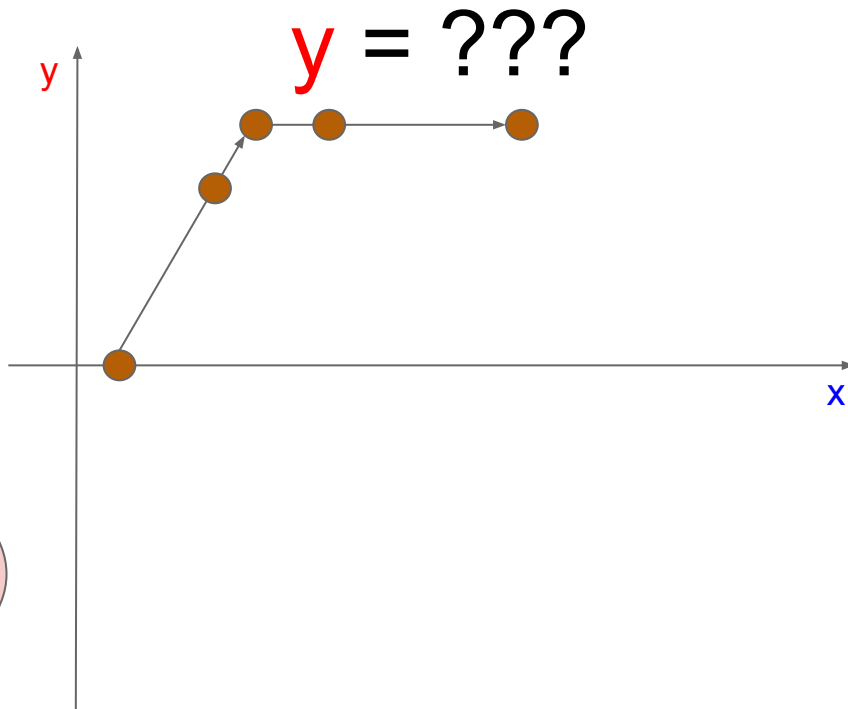
Model
Problem

Underfitting



Nonlinear Neural Models

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



Can we learn
arbitrary functions?

Nonlinear Neural Models

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

Use different linear functions
depending on the value of x ?

Nonlinear Neural Models

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

s_1 - 1 if $x < 6$ and 0 otherwise

s_2 - 1 if $x \geq 6$ and 0 otherwise

Nonlinear Neural Models

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

s_1 - 1 if $x < 6$ and 0 otherwise

s_2 - 1 if $x \geq 6$ and 0 otherwise

Data

n	x	y
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

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

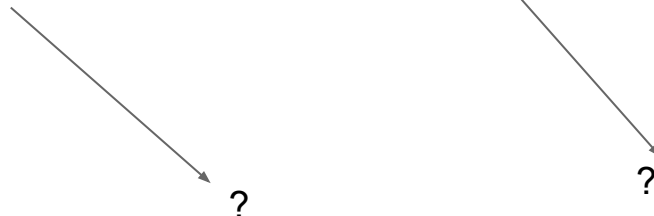
s_1 - 1 if $x < 6$ and 0 otherwise

s_2 - 1 if $x \geq 6$ and 0 otherwise

Data

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

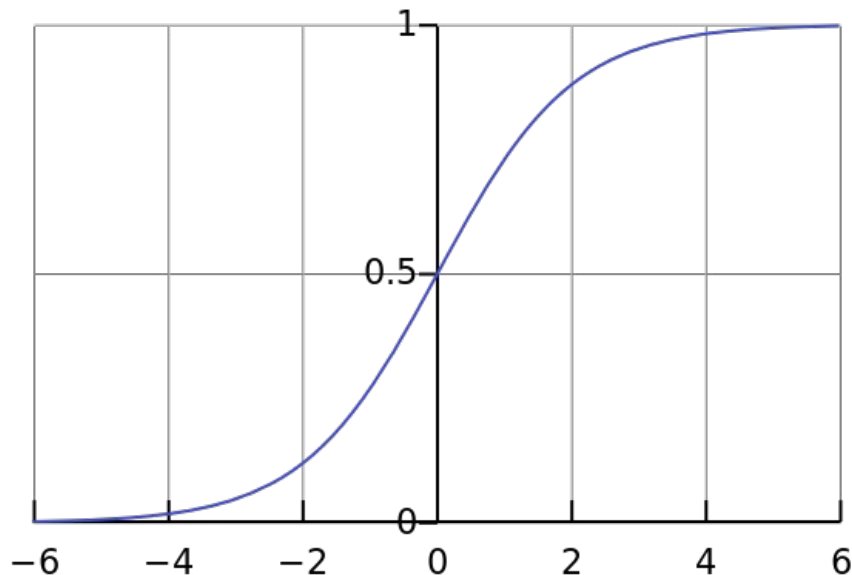
$$y = (4x - 4)s_1 + (0x + 20)s_2$$



Nonlinear Neural Models

$$s = \sigma(wx + b)$$

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



Nonlinear Neural Models

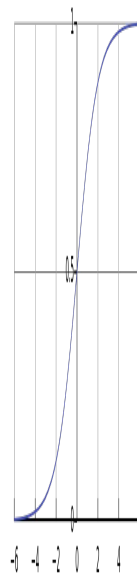
$$s = \sigma(1000x)$$

Nonlinear Neural Models

$$s = \sigma(1000x)$$

$$x = 0.1 \text{ then } \sigma(1000x) = 1$$

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



Nonlinear Neural Models

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

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

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

Nonlinear Neural Models

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

$$s_1 = \sigma(w_3x + b_3)$$

$$s_2 = \sigma(w_4x + b_4)$$

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

$$s_1 = \sigma(1000)$$

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

$$s_1 = \sigma(1000)$$

$$s_2 = \sigma(-1000)$$

Nonlinear Neural Models

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

$$y = 16$$

$$s1 = \sigma(1000)$$

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

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

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

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

$$s_1 = \sigma(-3000)$$

$$s_2 = \sigma(1000x - 6000)$$

Nonlinear Neural Models

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

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

$$s_1 = \sigma(-3000)$$

$$s_2 = \sigma(3000)$$

Nonlinear Neural Models

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

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

$$s1 = \sigma(-3000)$$

$$s2 = \sigma(3000)$$

Nonlinear Neural Models

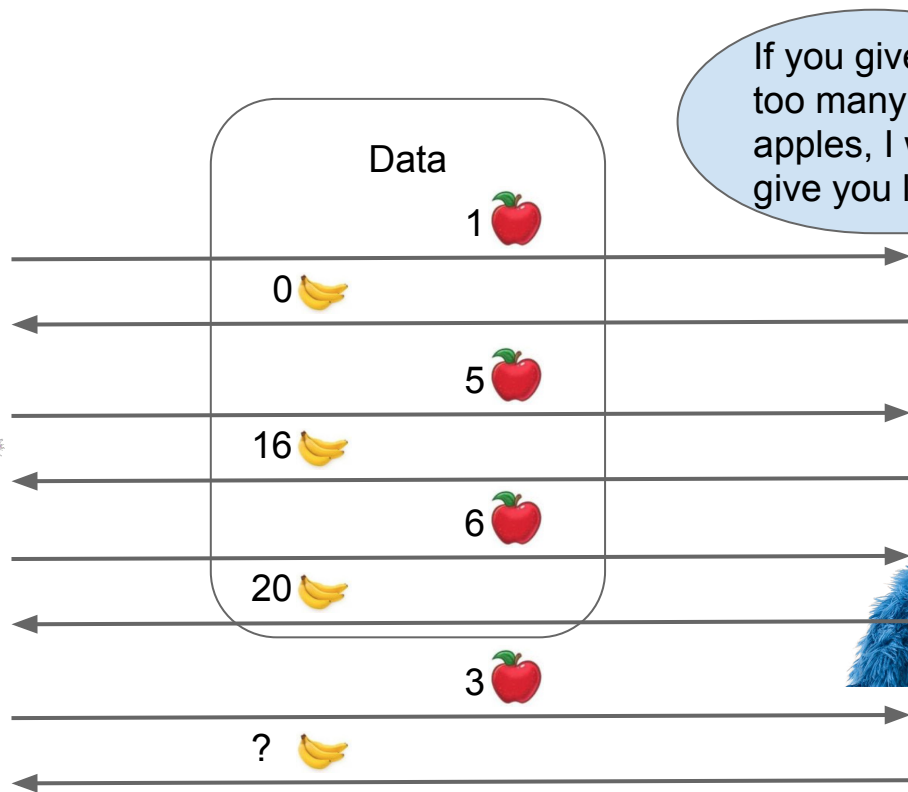
Data		
n	x	y
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20

$$y = 20$$

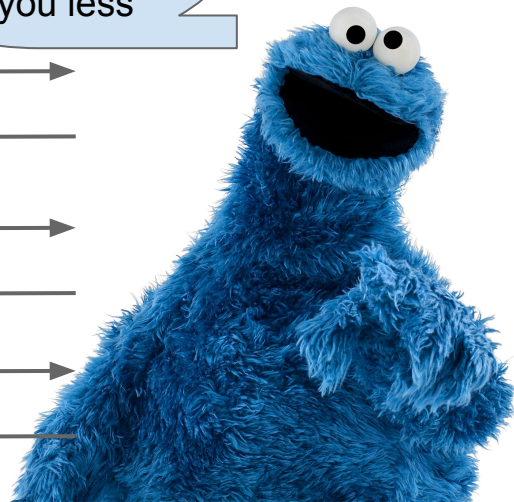
$$s1 = \sigma(-3000)$$

$$s2 = \sigma(3000)$$

Nonlinear Neural Models

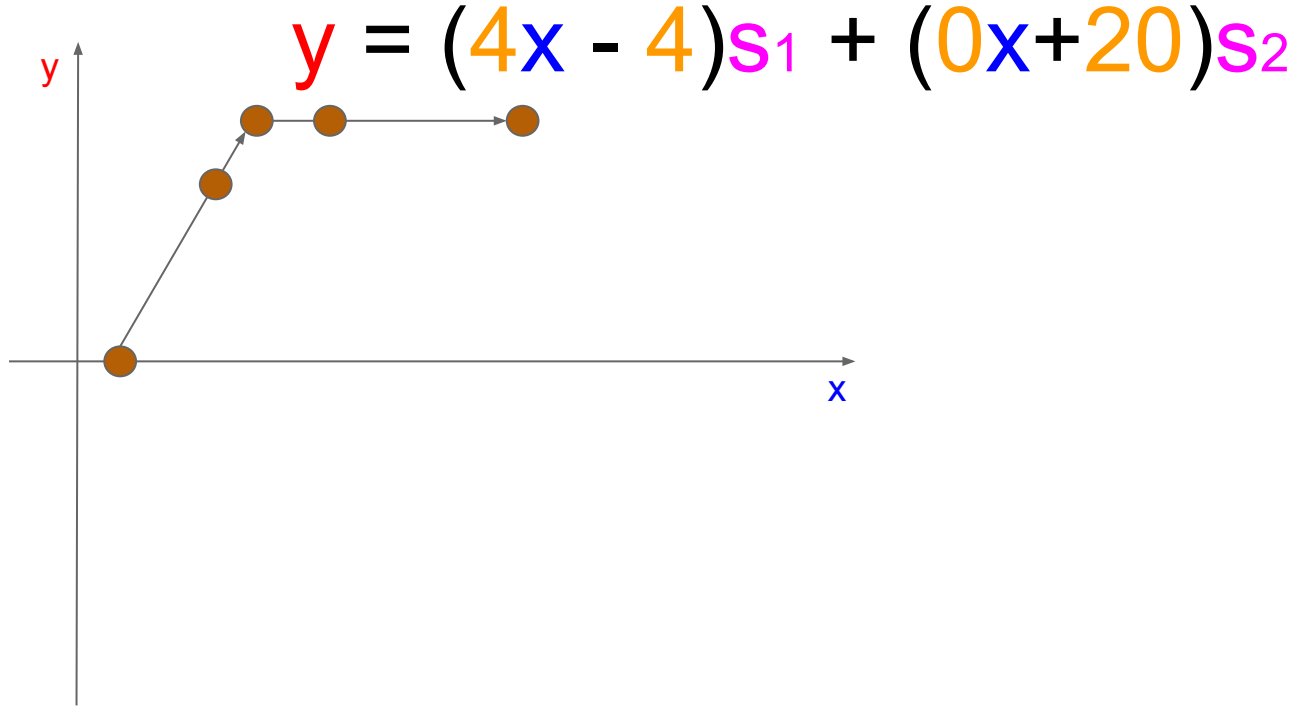


If you give me too many apples, I will give you less



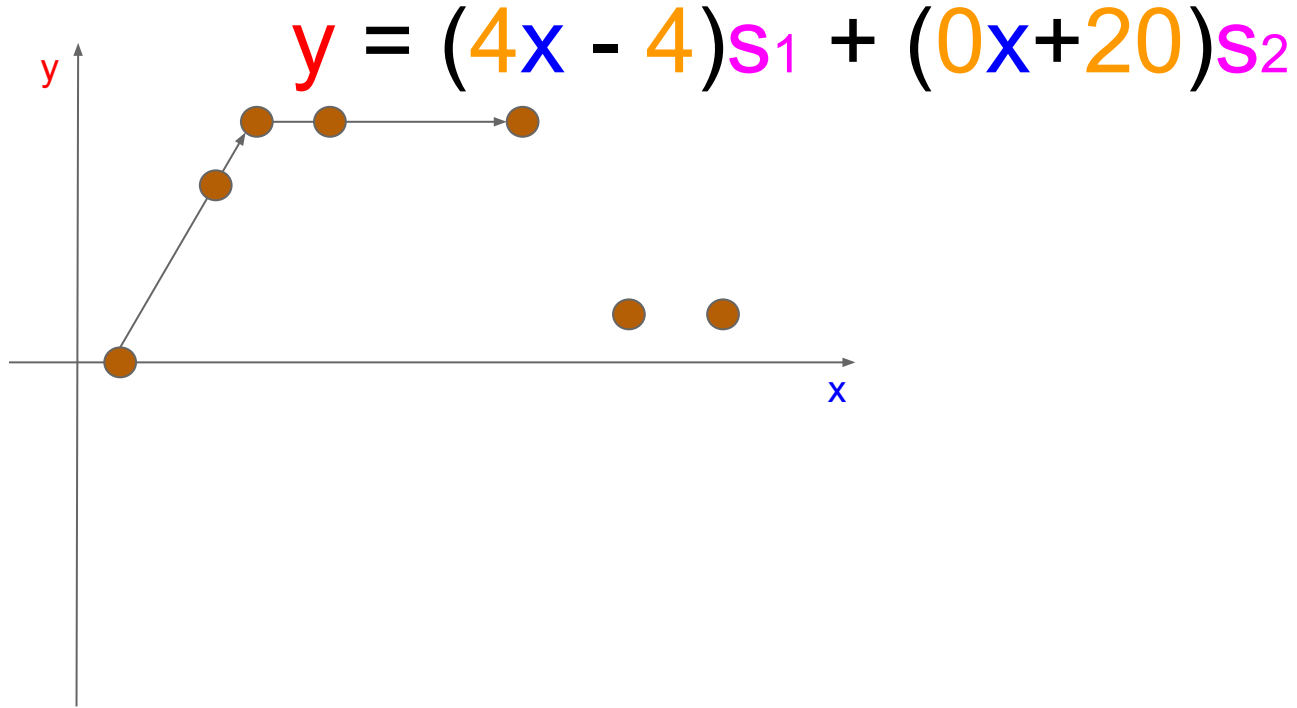
Multilayer Perceptrons

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



Multilayer Perceptrons

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

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

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

$$s_2 = \text{????}$$

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

Multilayer Perceptrons

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

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

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

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

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

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

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

$$s_1 = \sigma(w_4x + b_4)$$

Layer 1 Perceptron

$$s_2 = \sigma(w_5s_1 + w_6s_3 + b_5)$$

$$s_3 = \sigma(w_7x + b_6)$$

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	y
0	1	0
1	5	16
2	6	20
3	9	20
4	11	20
5	15	1
6	19	1

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

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

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

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

Multilayer Perceptrons

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

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

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

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

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

Multilayer Perceptrons

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

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

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

$$s_2 = \sigma(500)$$

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

Multilayer Perceptrons

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

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

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

$$s_2 = \sigma(500) = 1$$

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

Multilayer Perceptrons

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

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

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

$$s_2 = \sigma(500) = 1$$

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

Multilayer Perceptrons

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

$$y = 20$$

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

$$s_2 = \sigma(500) = 1$$

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

Multilayer Perceptrons

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

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

$$s_1 = \sigma(-13000) = 0$$

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

$$s_3 = \sigma(4000) = 1$$

Multilayer Perceptrons

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

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

$$s_1 = \sigma(-13000) = 0$$

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

$$s_3 = \sigma(4000) = 1$$

Multilayer Perceptrons

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

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

$$s_1 = \sigma(-13000) = 0$$

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

$$s_3 = \sigma(4000) = 1$$

Multilayer Perceptrons

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

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

$$s_1 = \sigma(-13000) = 0$$

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

$$s_3 = \sigma(4000) = 1$$

Multilayer Perceptrons

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

$$y = 1$$

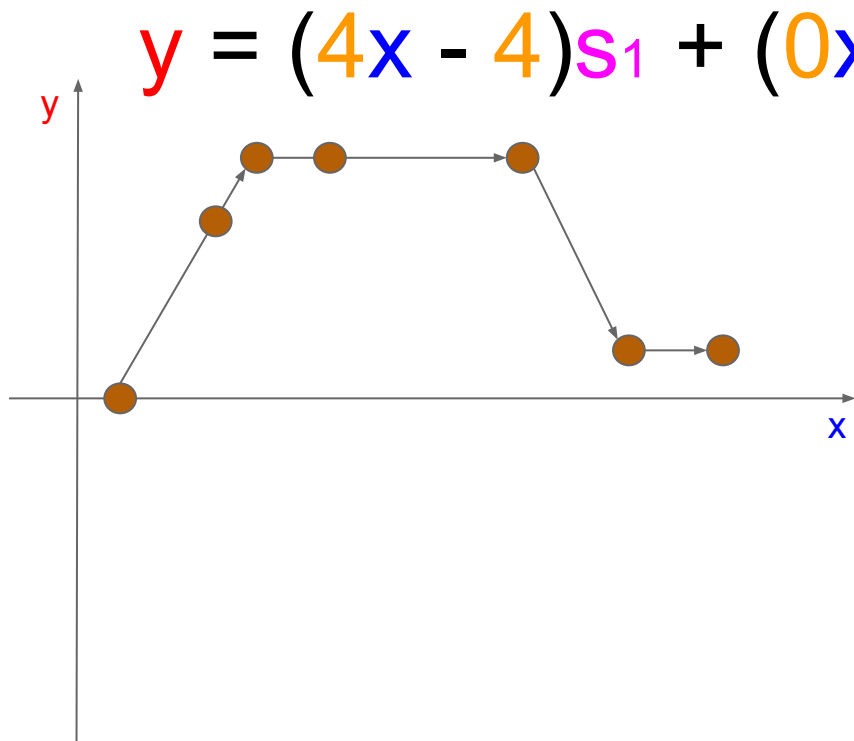
$$s_1 = \sigma(-13000) = 0$$

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

$$s_3 = \sigma(4000) = 1$$

Multilayer Perceptrons

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



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

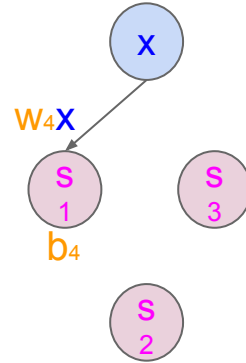
Multilayer Perceptrons

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

$$s_1 = \sigma(w_4x + b_4)$$

$$s_2 = \sigma(w_5s_1 + w_6s_3 + b_5)$$

$$s_3 = \sigma(w_7x + b_6)$$



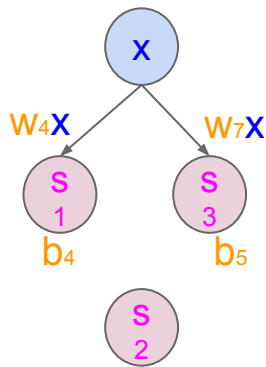
Multilayer Perceptrons

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

$$s_1 = \sigma(w_4x + b_4)$$

$$s_2 = \sigma(w_5s_1 + w_6s_3 + b_5)$$

$$s_3 = \sigma(w_7x + b_6)$$



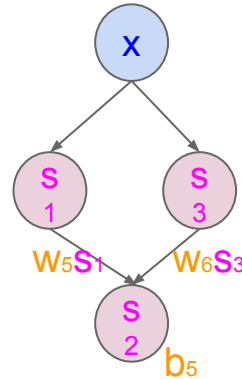
Multilayer Perceptrons

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

$$s_1 = \sigma(w_4x + b_4)$$

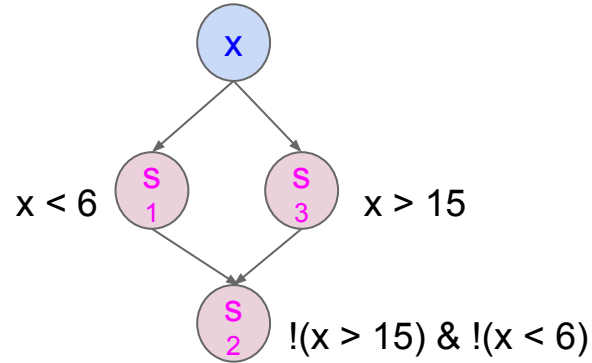
$$s_2 = \sigma(w_5s_1 + w_6s_3 + b_5)$$

$$s_3 = \sigma(w_7x + b_6)$$



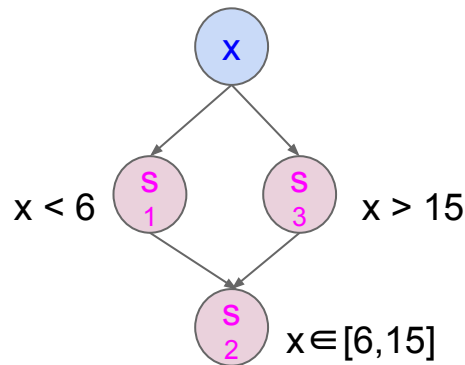
Multilayer Perceptrons

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

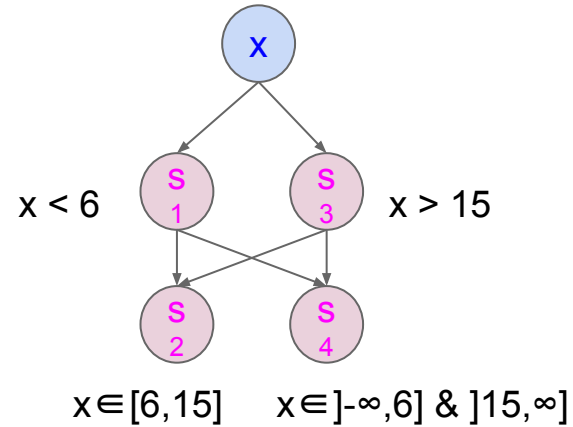


Multilayer Perceptrons

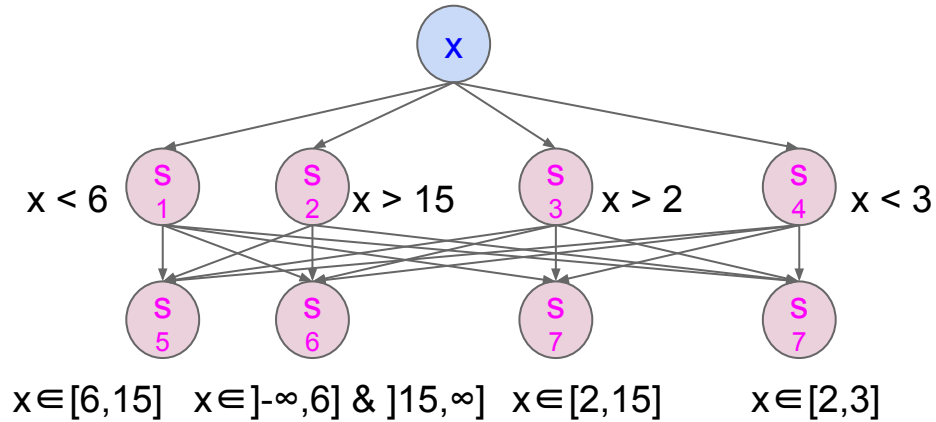
$$y = (w_1x + b_1)s_1 + (w_2x + b_2)s_2 + (w_3x + b_3)s_3$$



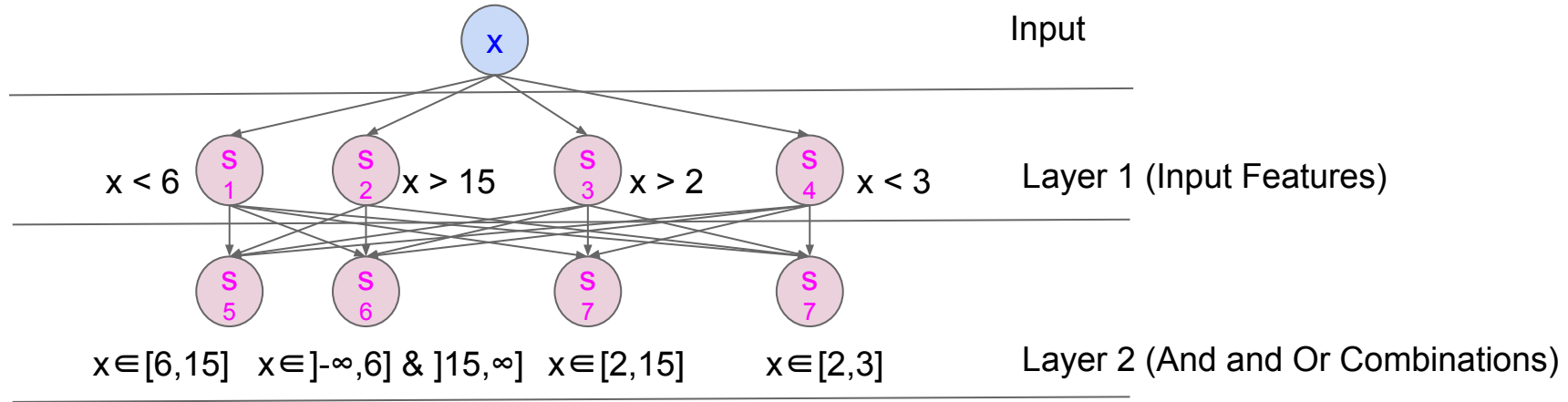
Multilayer Perceptrons



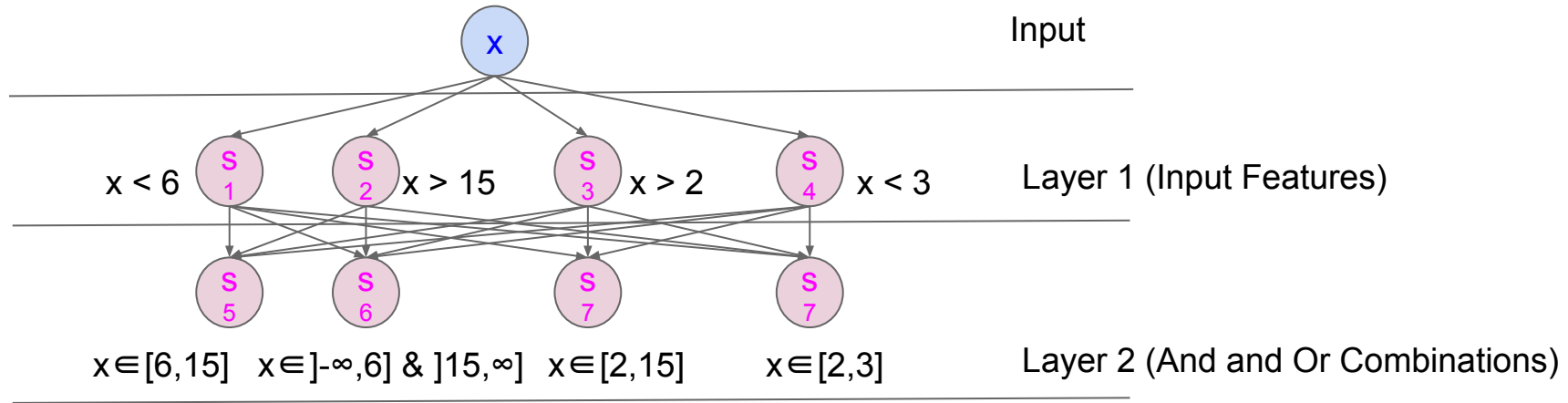
Multilayer Perceptrons



Multilayer Perceptrons



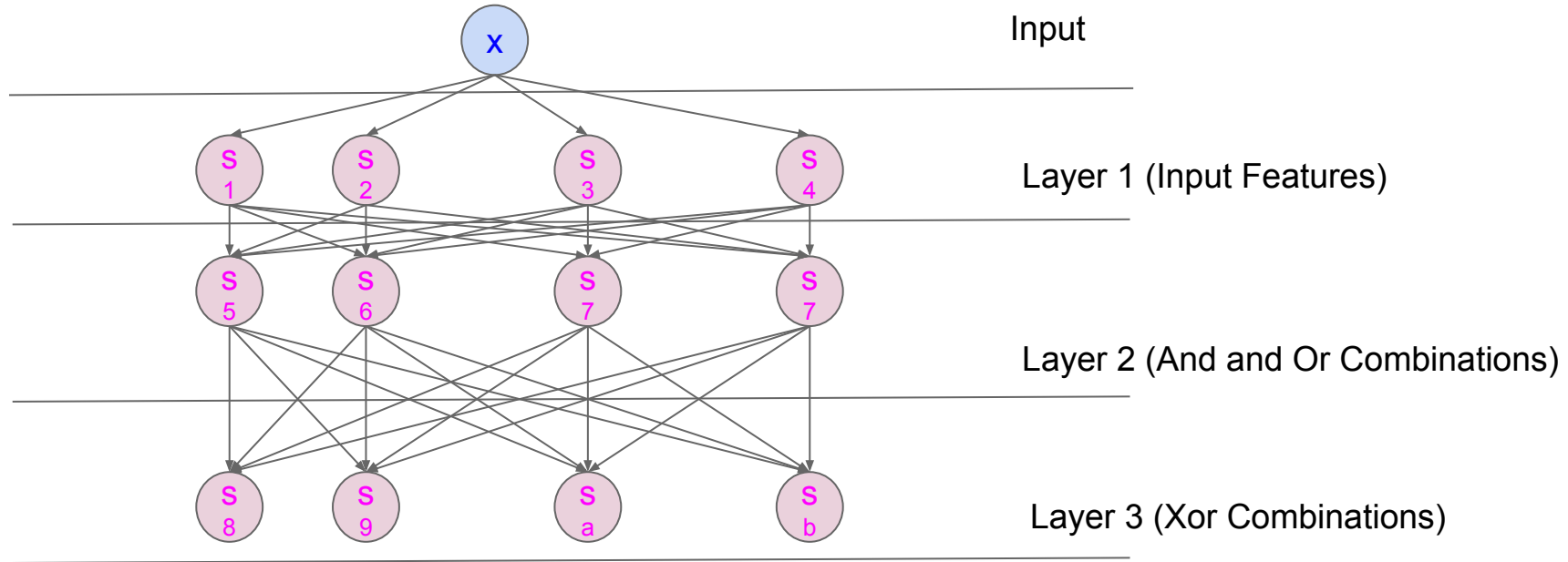
Multilayer Perceptrons



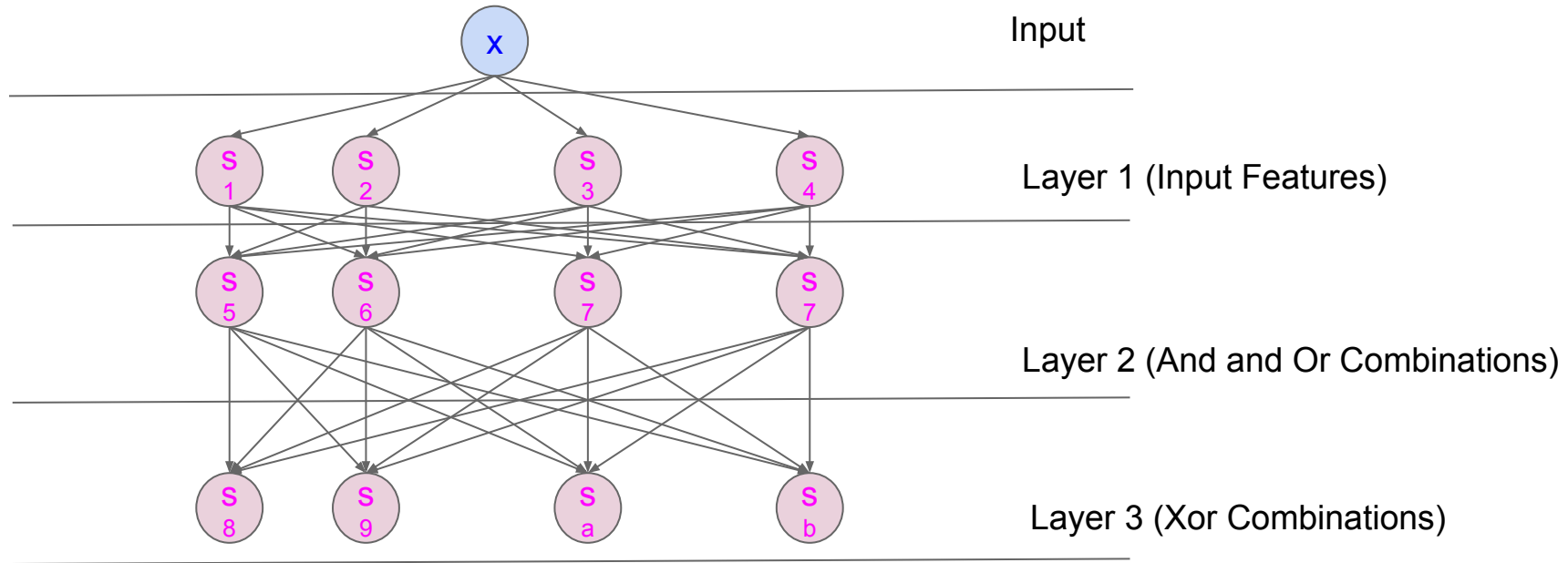
$$\text{And}(s_1, s_2) = \sigma(1000s_1 + 1000s_3 - 1500)$$

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

Multilayer Perceptrons



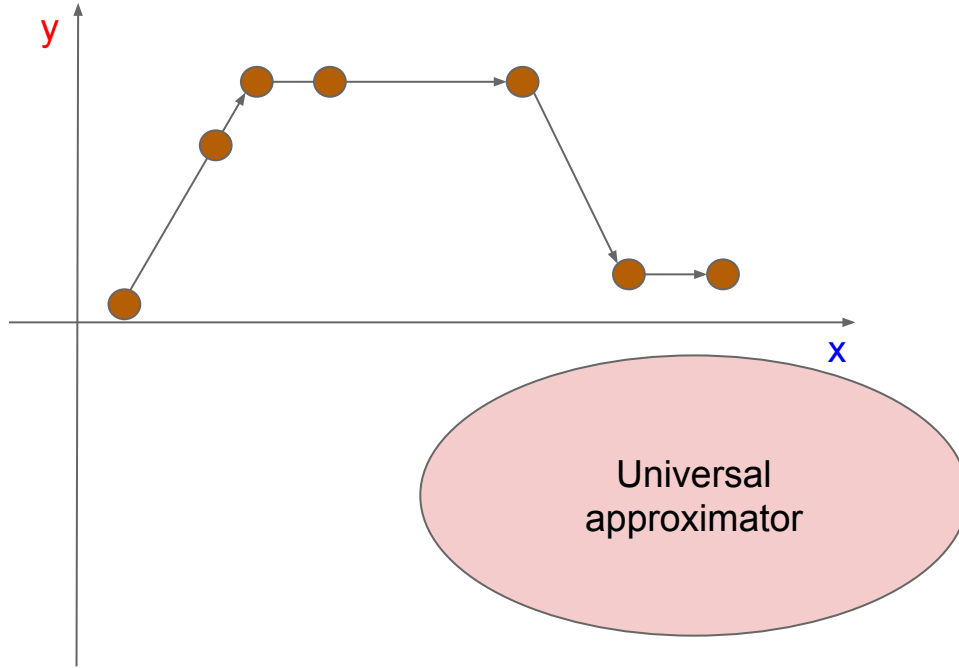
Multilayer Perceptrons



$$\text{Xor}(s_1, s_2) = \text{Or}(\text{And}(s_1, !s_2), \text{And}(!s_1, s_2))$$

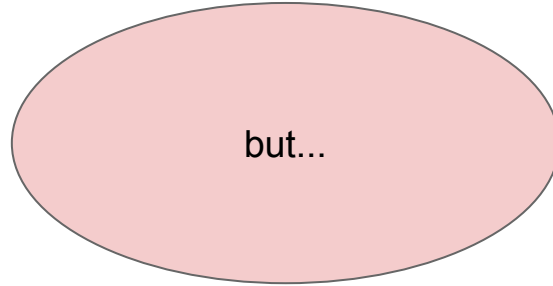
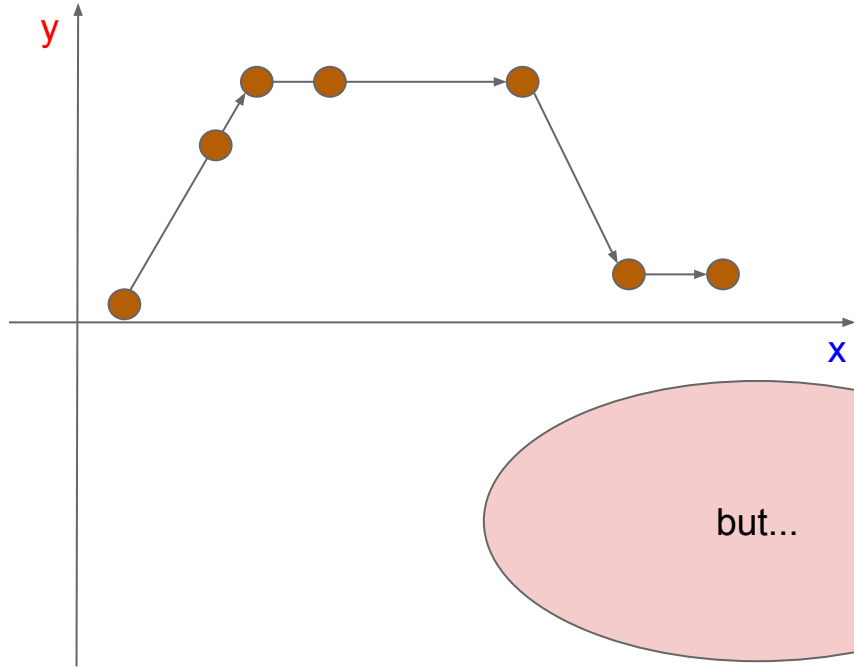
Multilayer Perceptrons

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



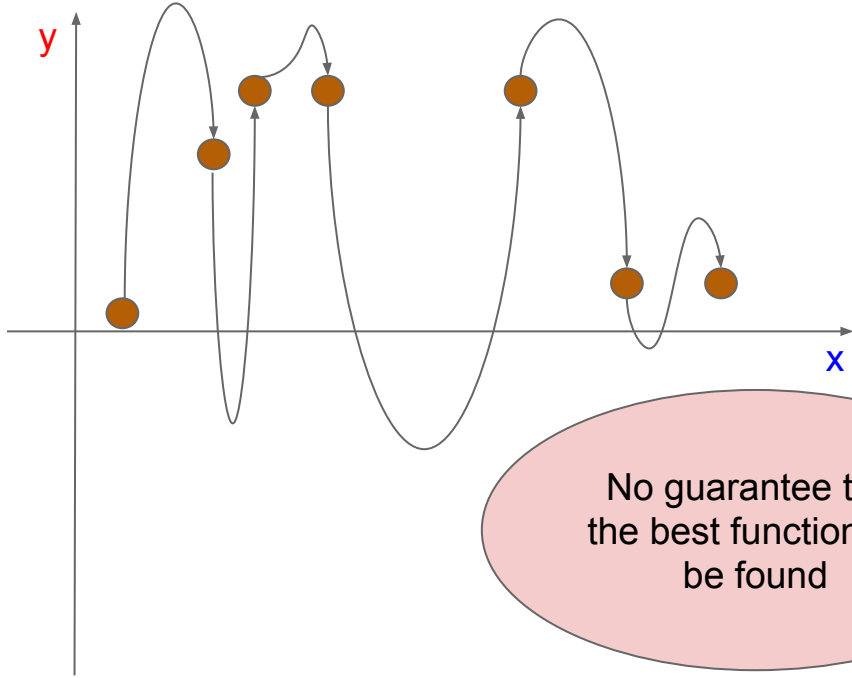
Multilayer Perceptrons

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



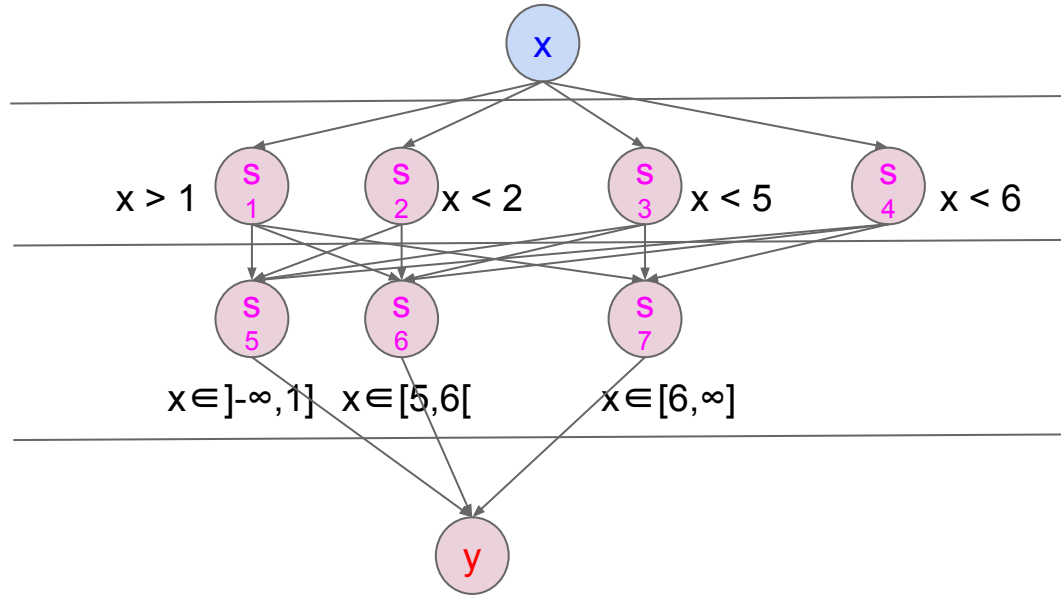
Multilayer Perceptrons

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



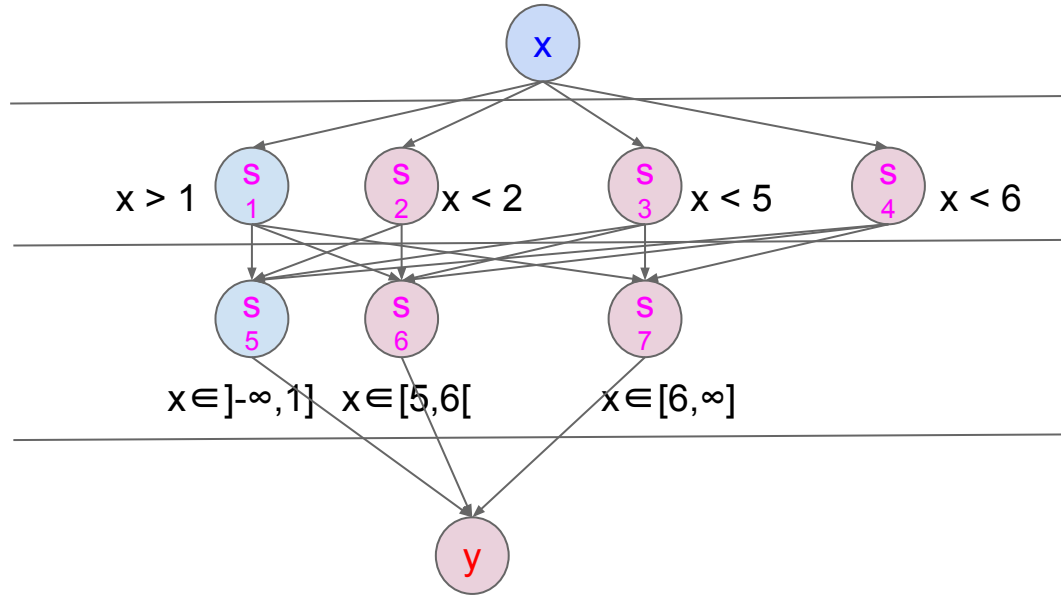
Multilayer Perceptrons

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



Multilayer Perceptrons

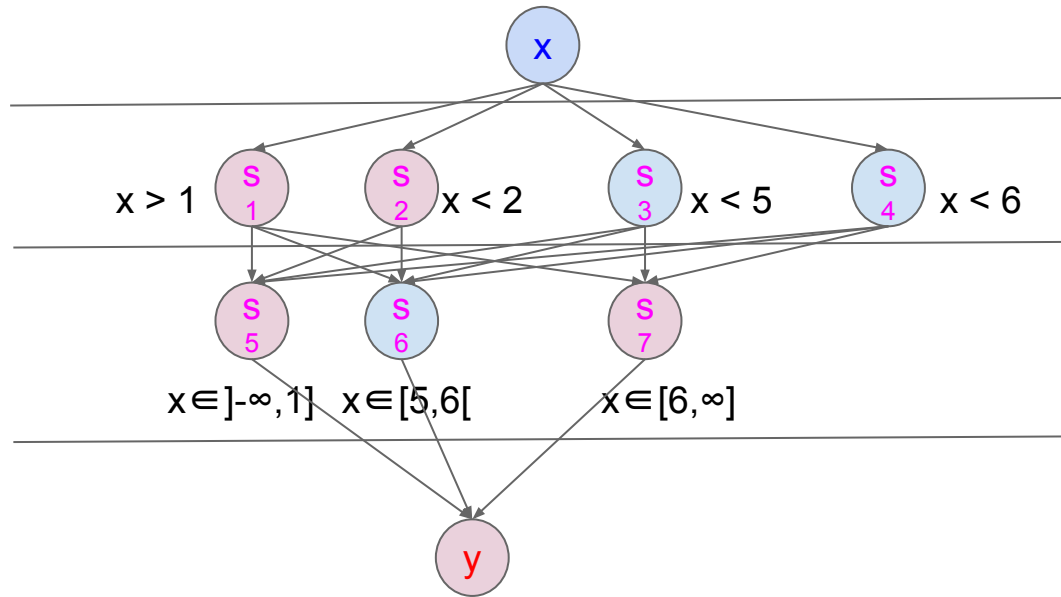
n	x	y
0	1	0
1	5	16
2	6	20



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

Multilayer Perceptrons

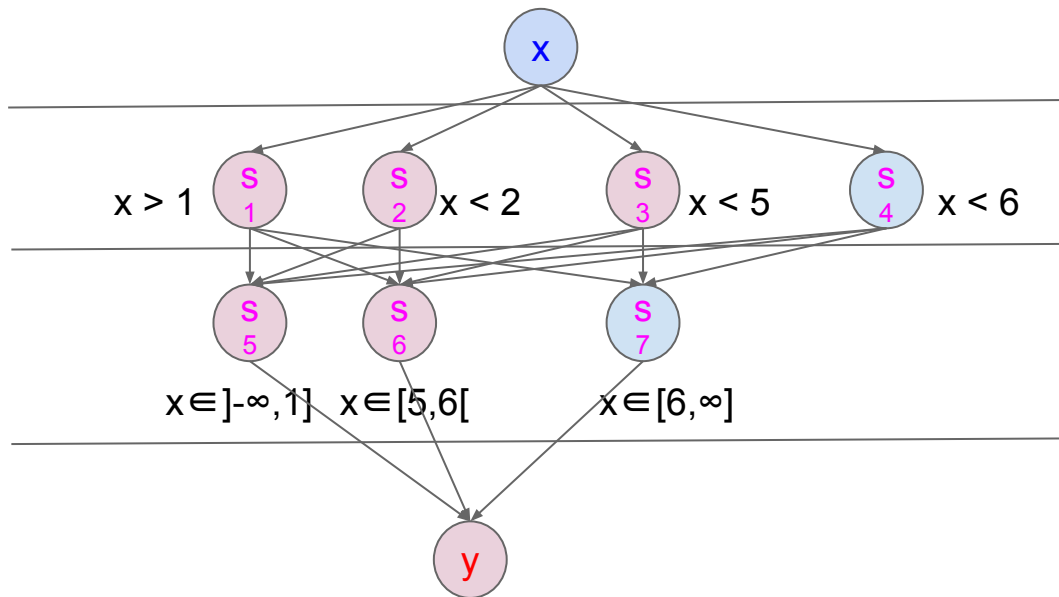
n	x	y
0	1	0
1	5	16
2	6	20



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

Multilayer Perceptrons

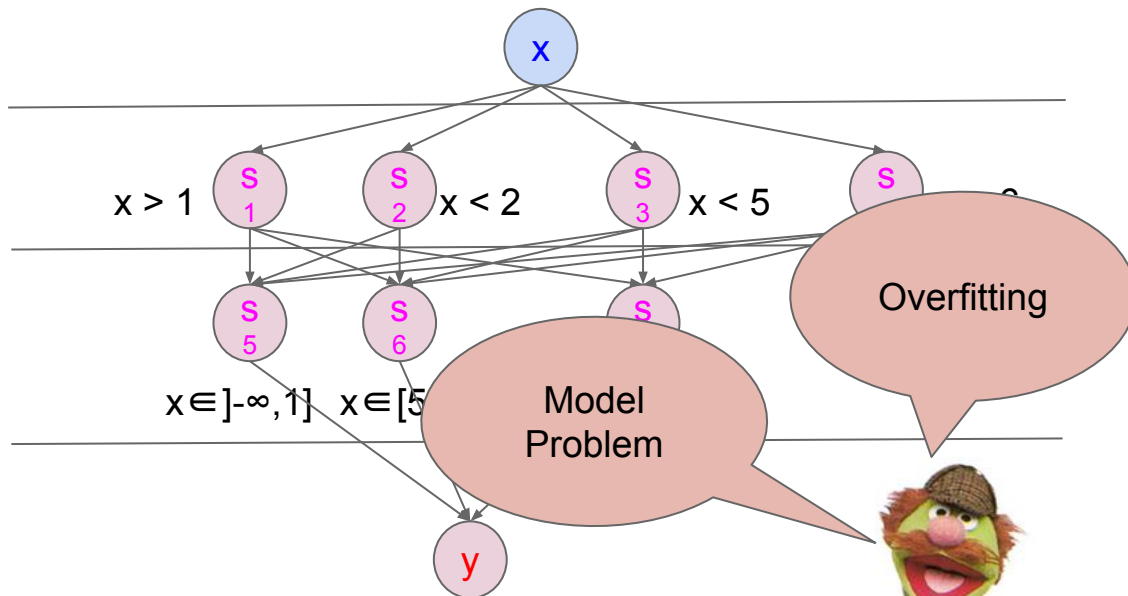
n	x	y
0	1	0
1	5	16
2	6	20



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

Multilayer Perceptrons

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



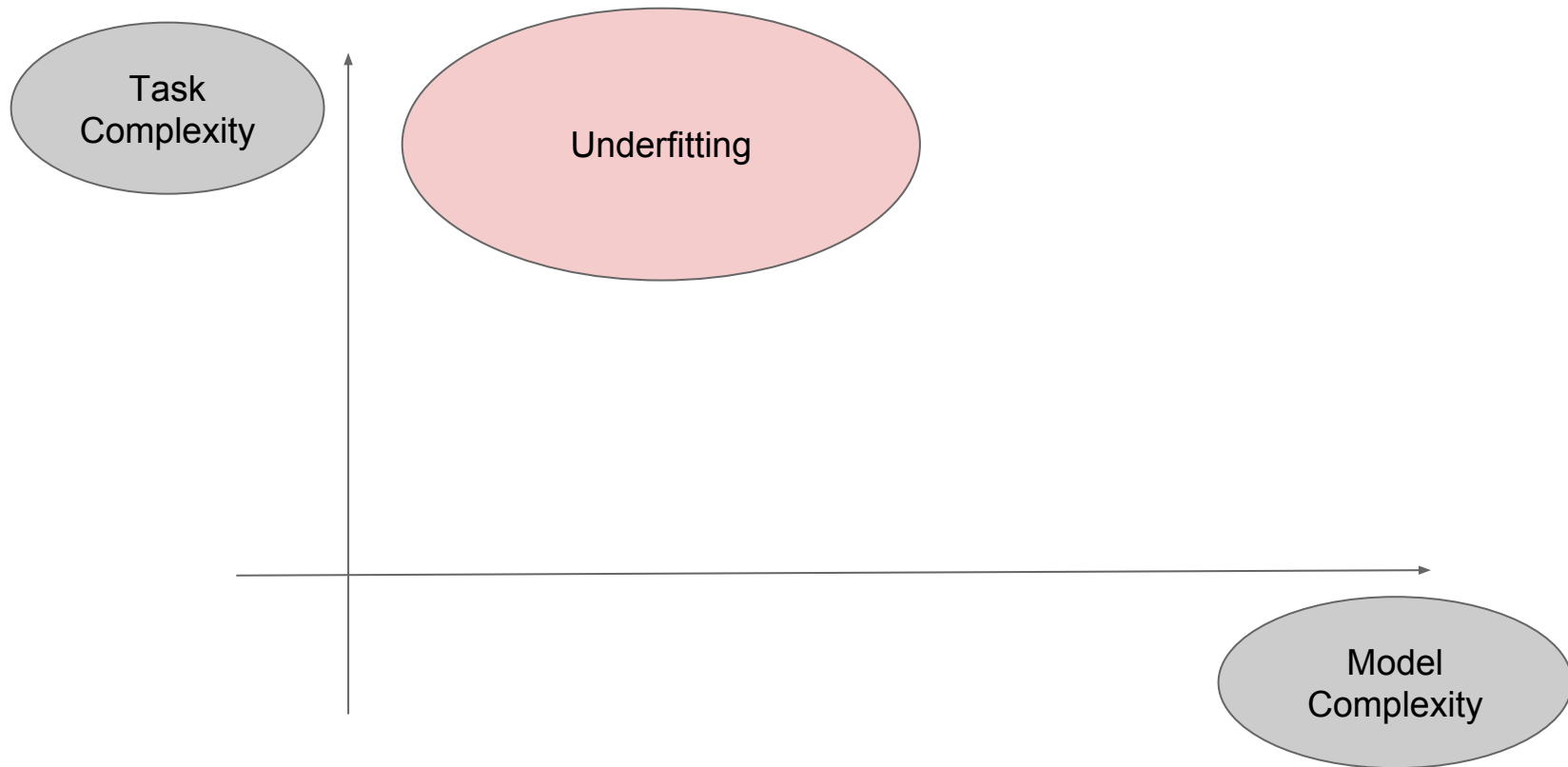
$$y = 0s_5 + 16s_6 + 20s_7$$



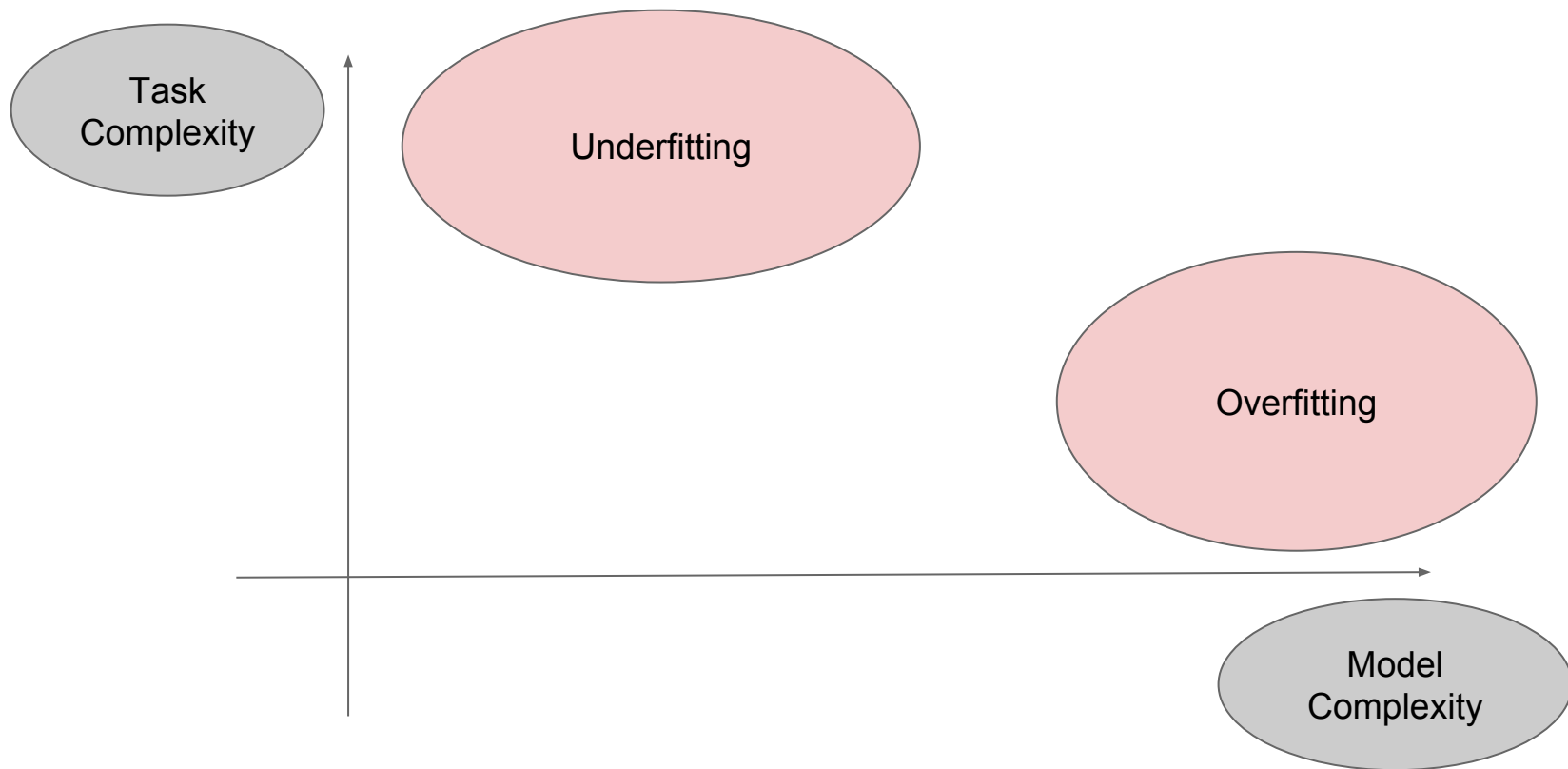
Multilayer Perceptrons



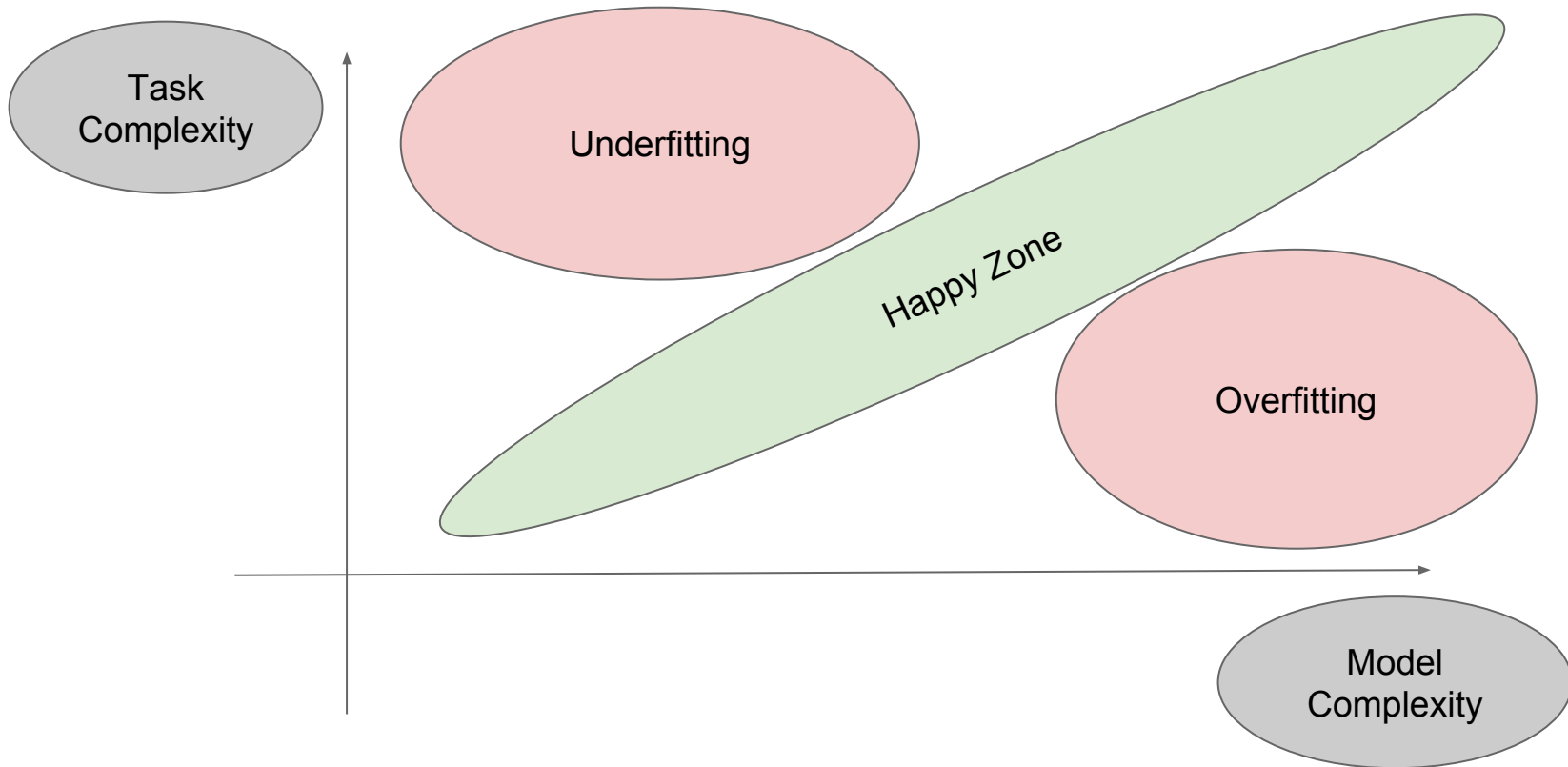
Multilayer Perceptrons



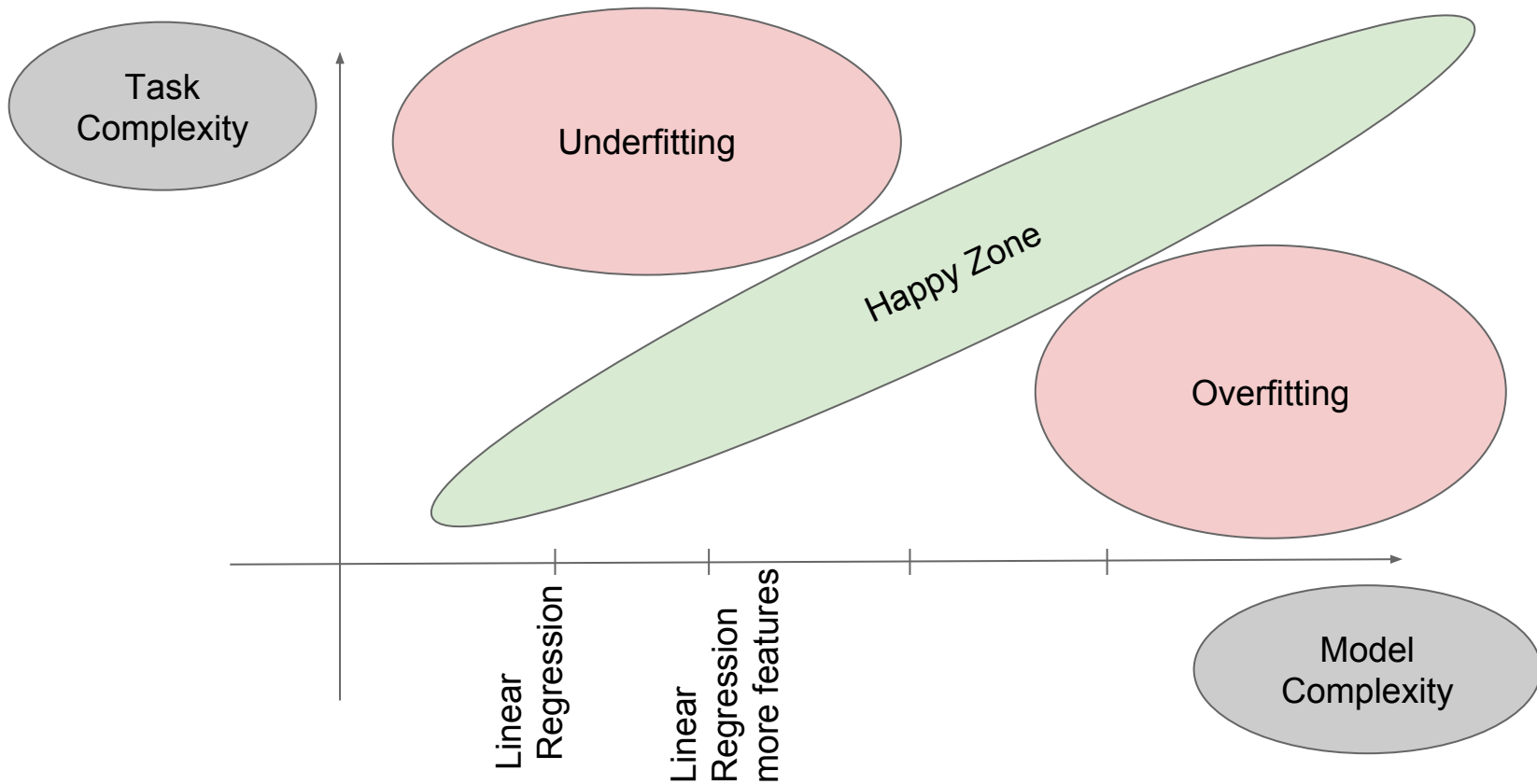
Multilayer Perceptrons



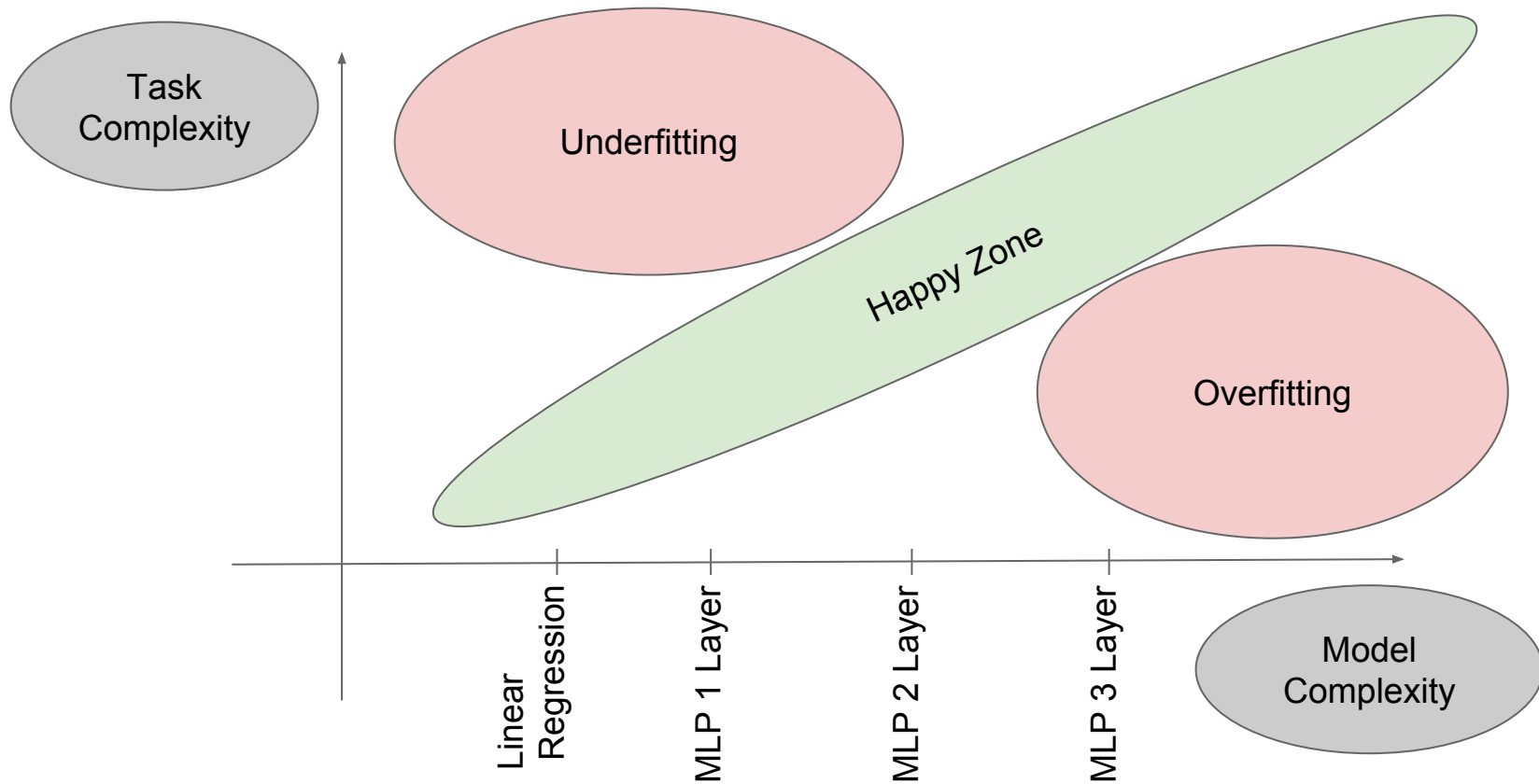
Multilayer Perceptrons



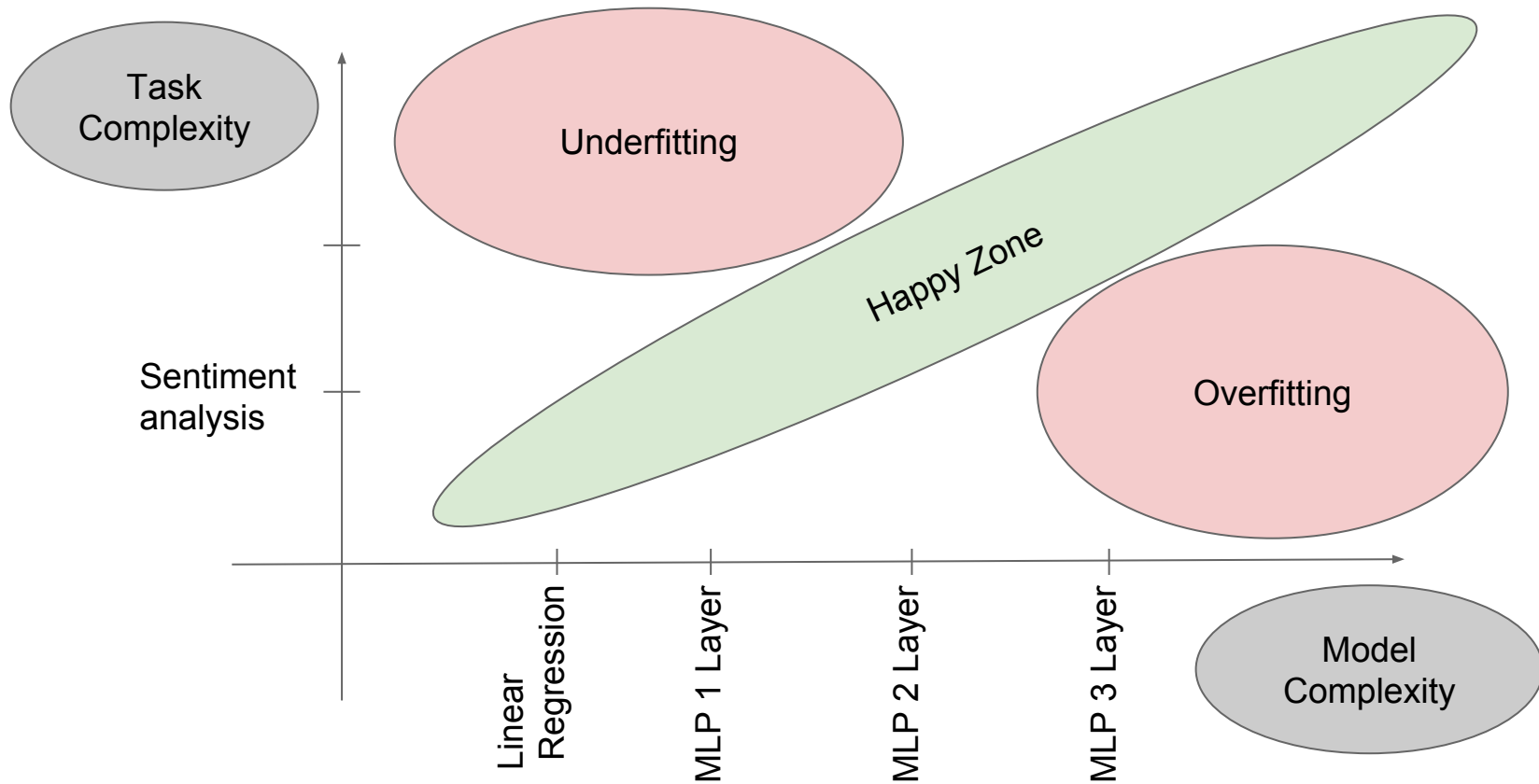
Multilayer Perceptrons



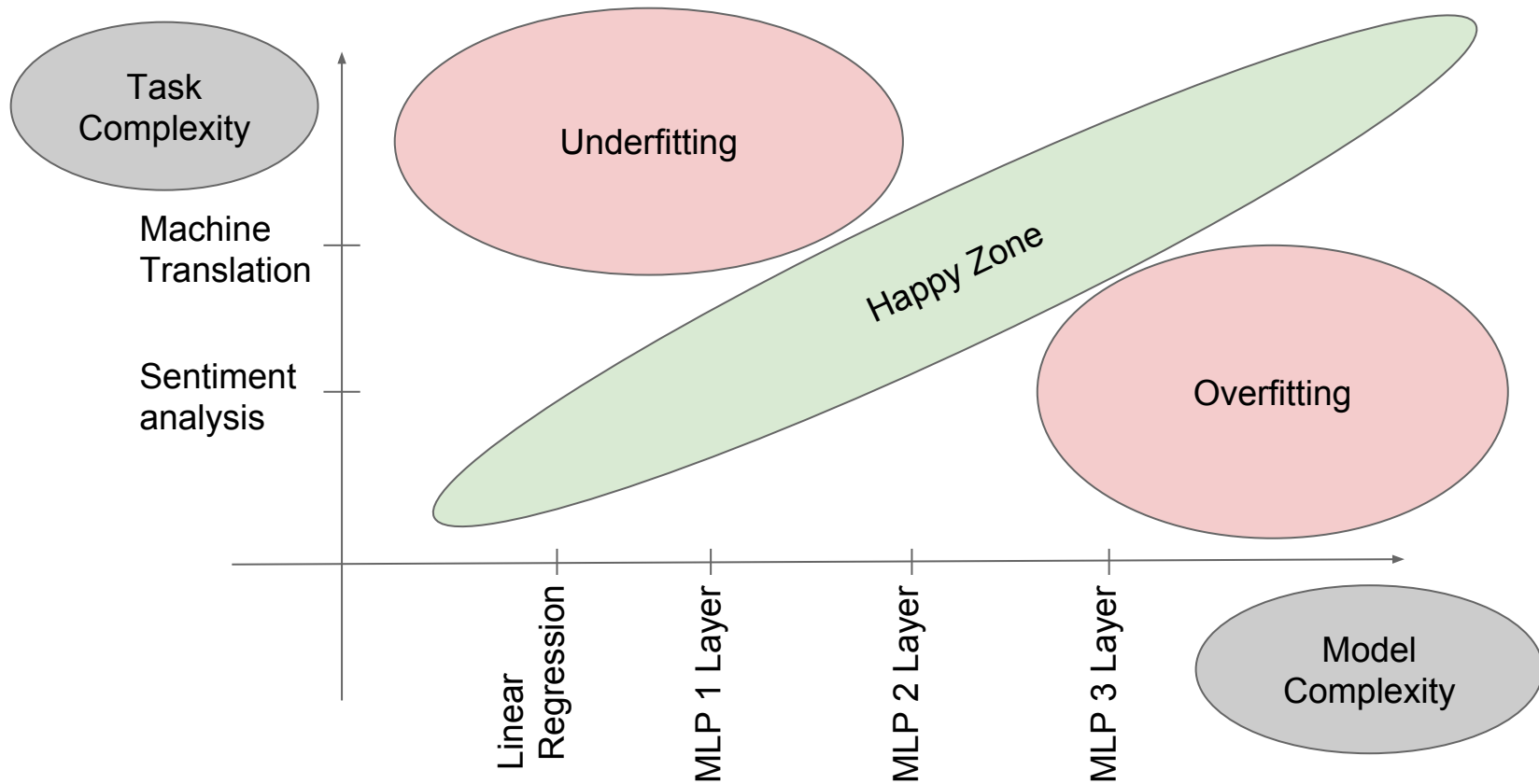
Multilayer Perceptrons



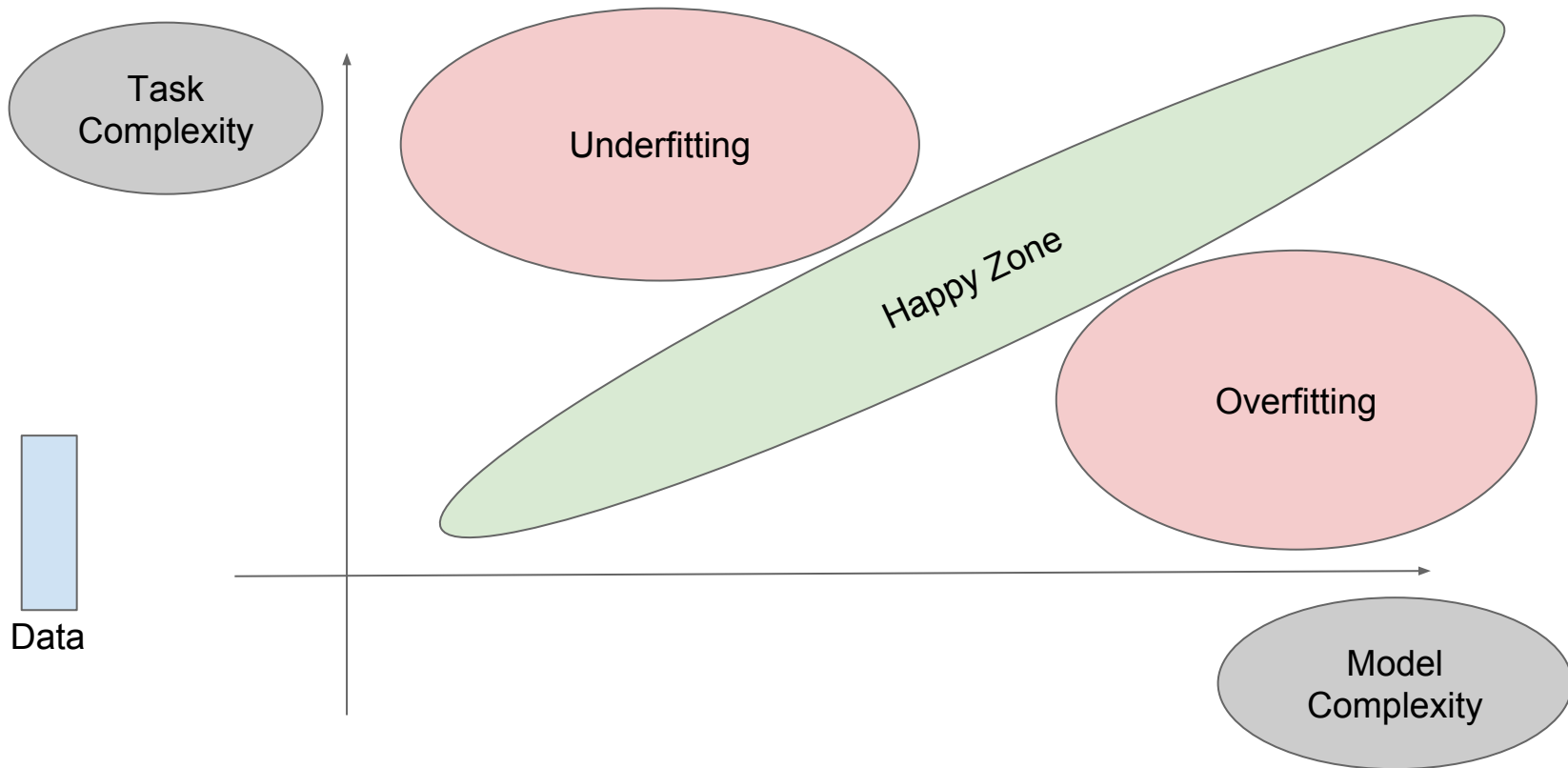
Multilayer Perceptrons



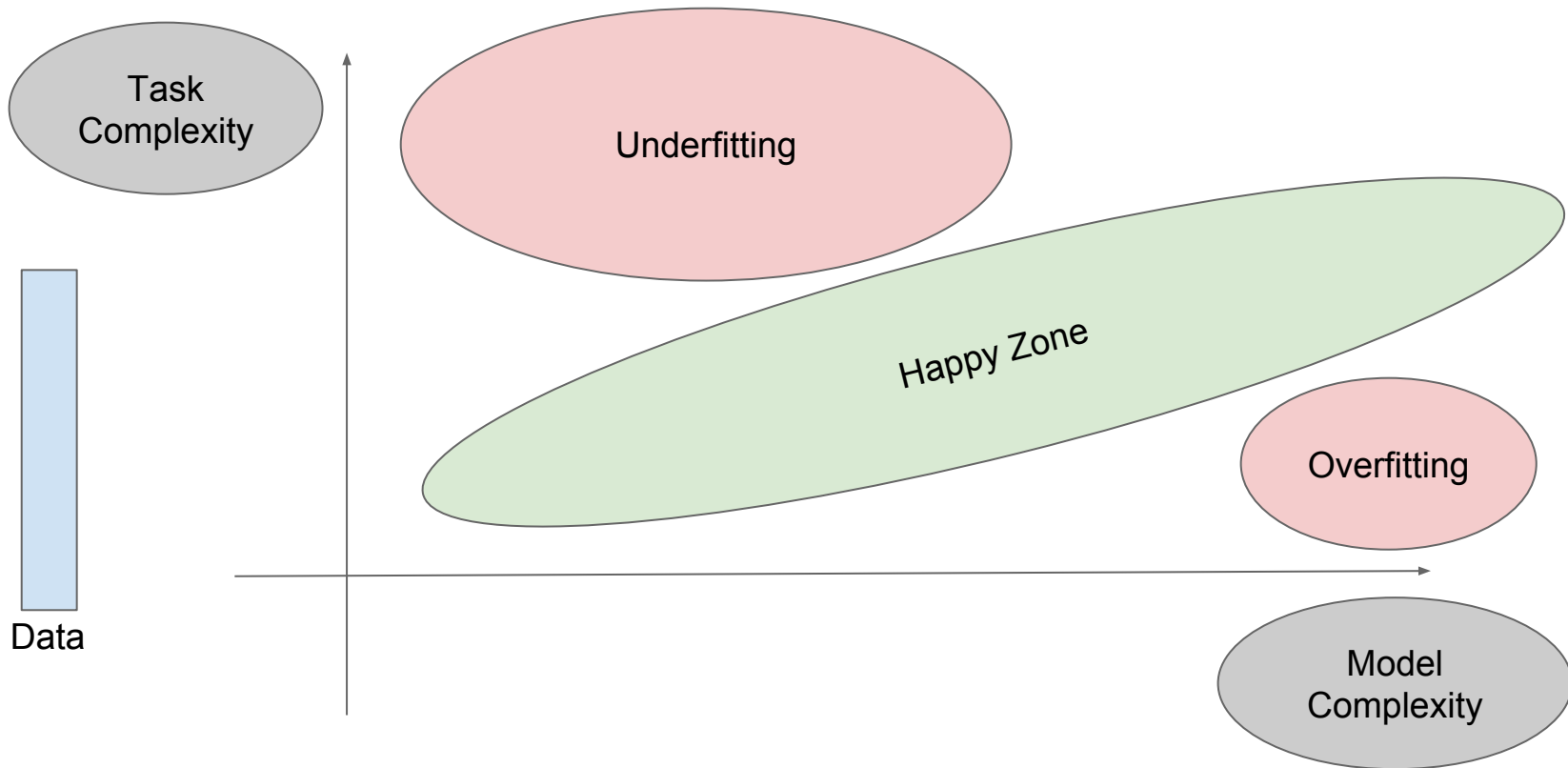
Multilayer Perceptrons



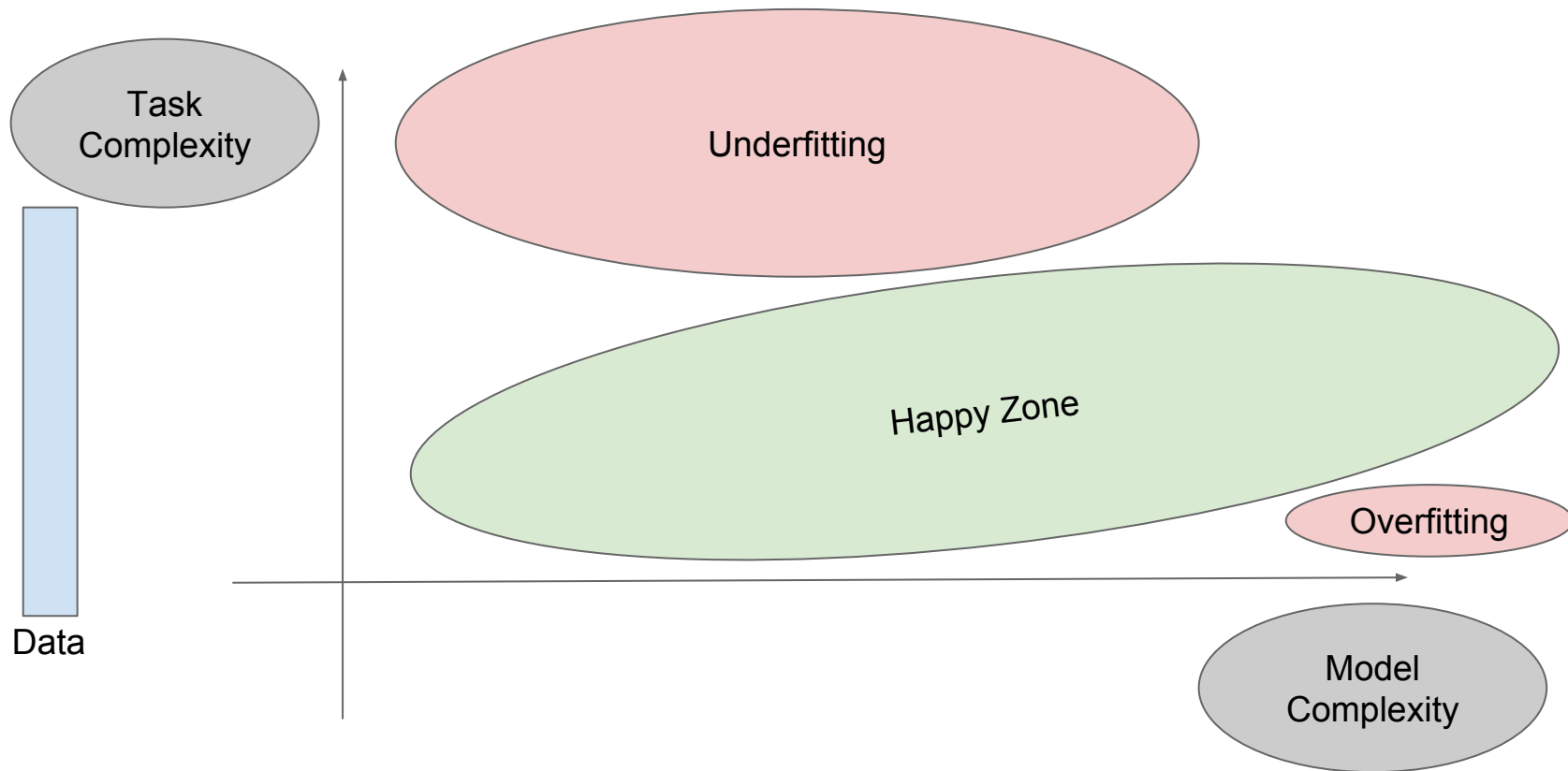
Multilayer Perceptrons



Multilayer Perceptrons

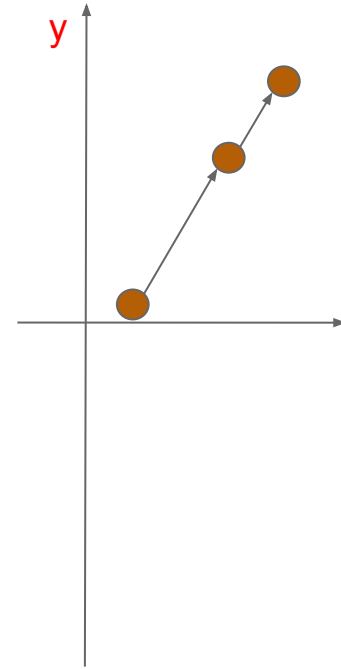
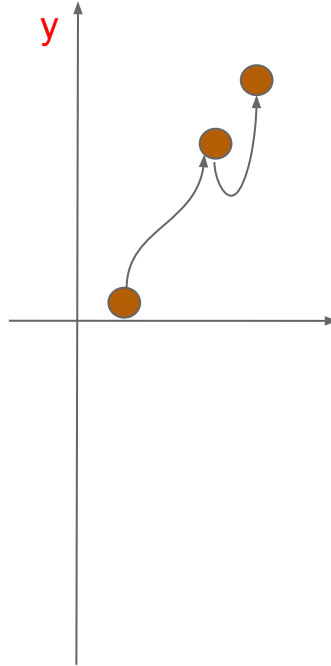
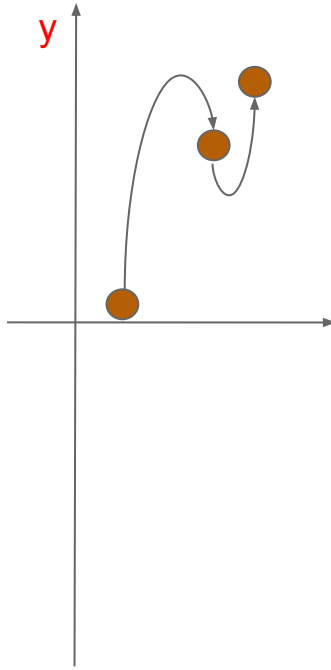


Multilayer Perceptrons



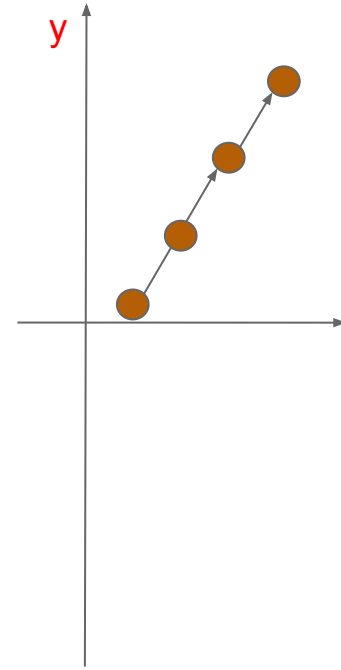
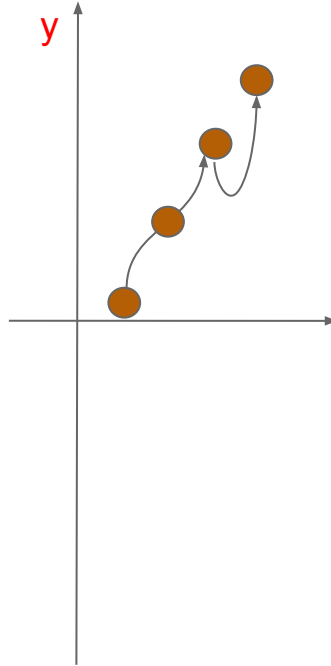
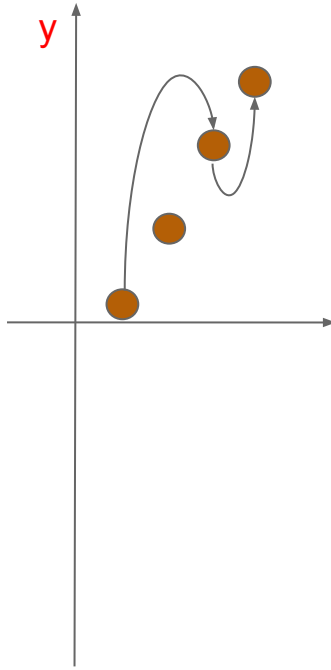
Multilayer Perceptrons

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



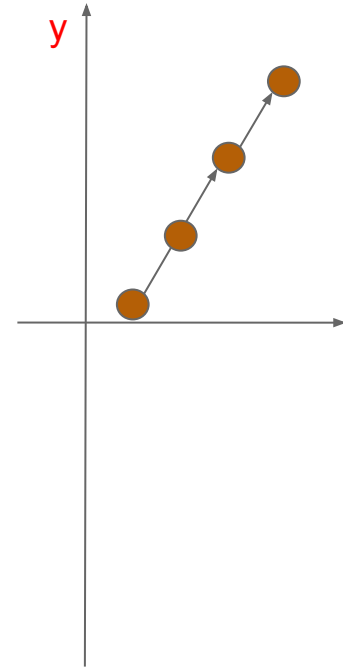
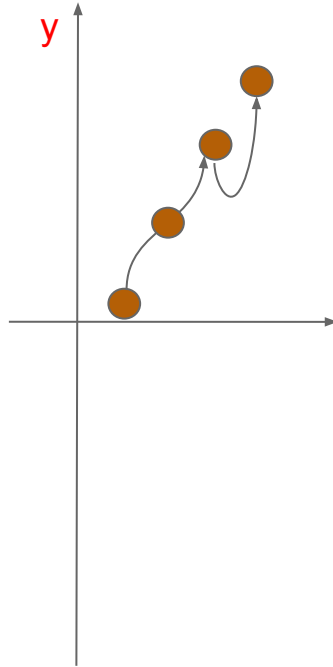
Multilayer Perceptrons

n	x	y
0	1	0
1	5	16
2	6	20
3	2	4

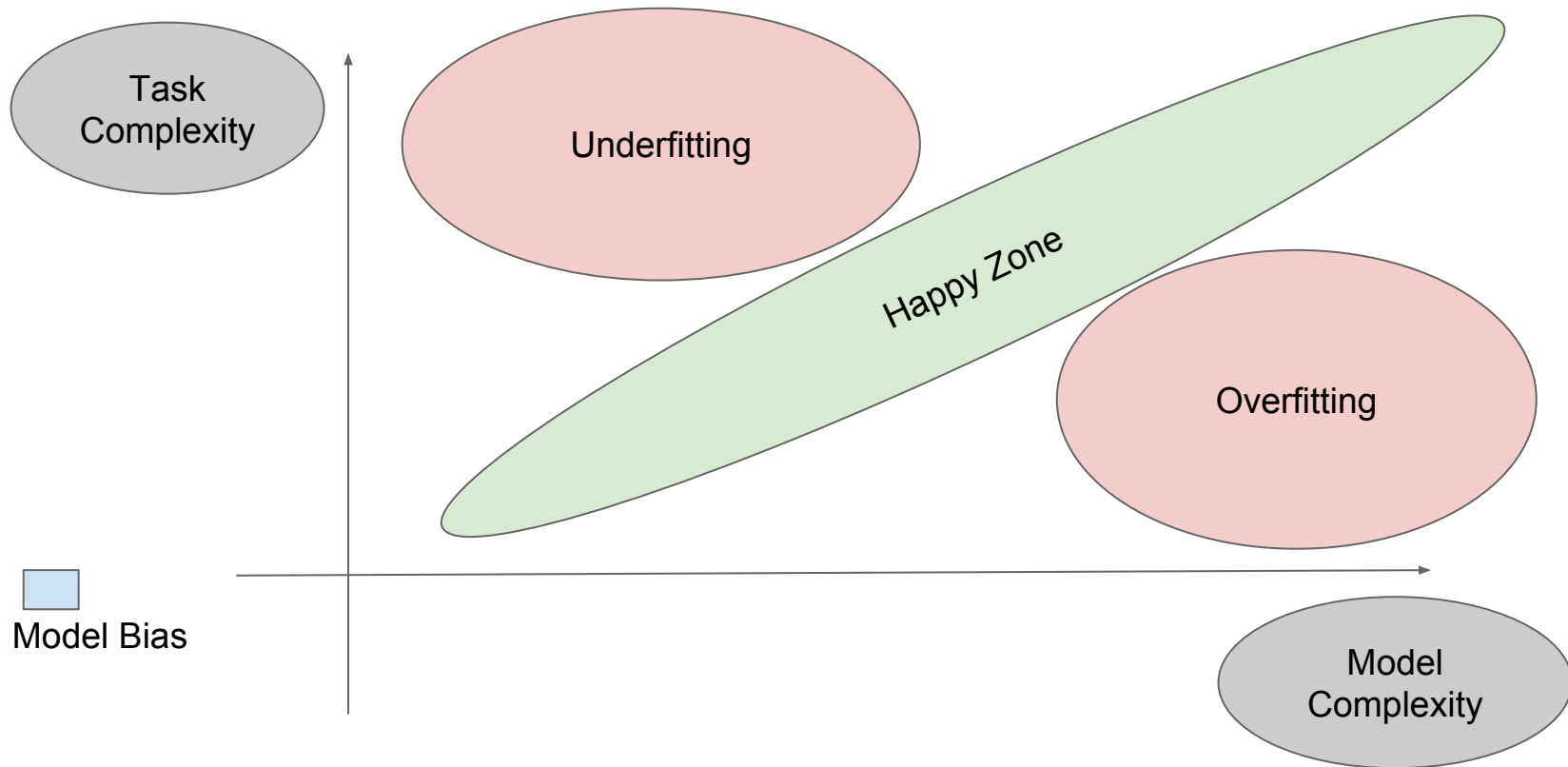


Multilayer Perceptrons

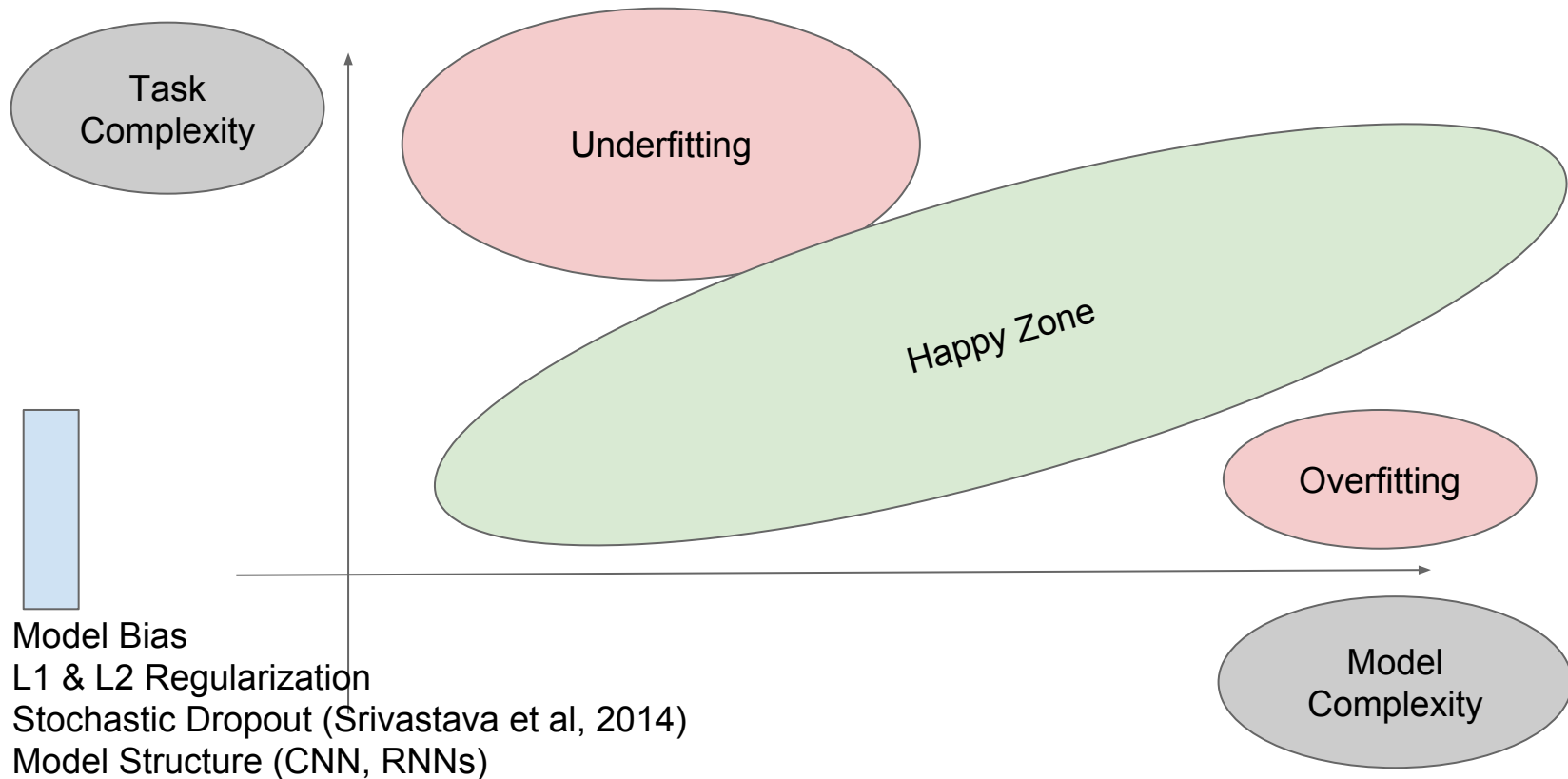
n	x	y
0	1	0
1	5	16
2	6	20
3	2	4



Multilayer Perceptrons



Multilayer Perceptrons



Multilayer Perceptrons

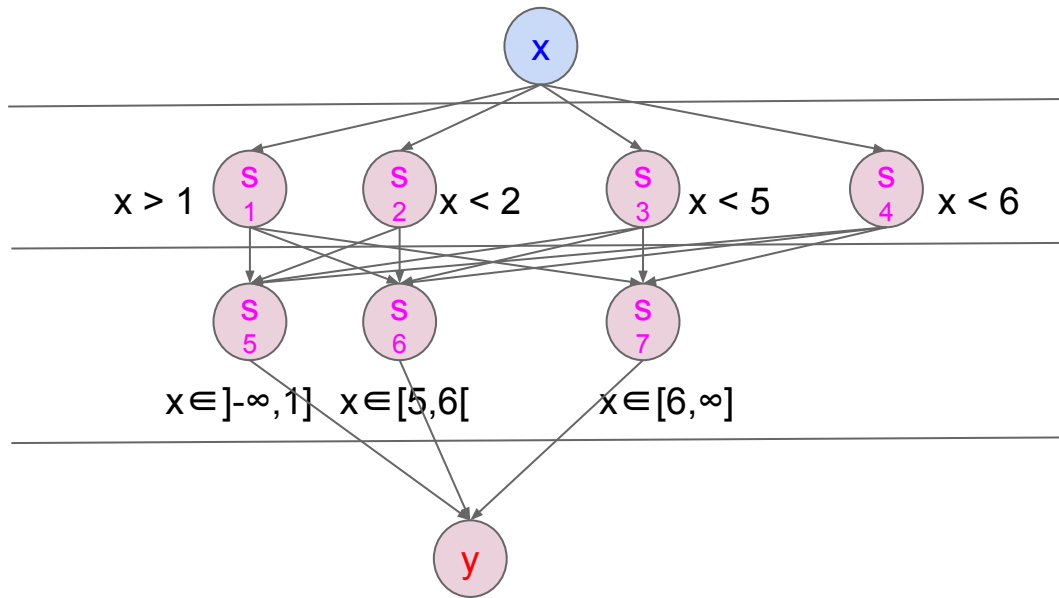
Regularization

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

β = Regularization constant

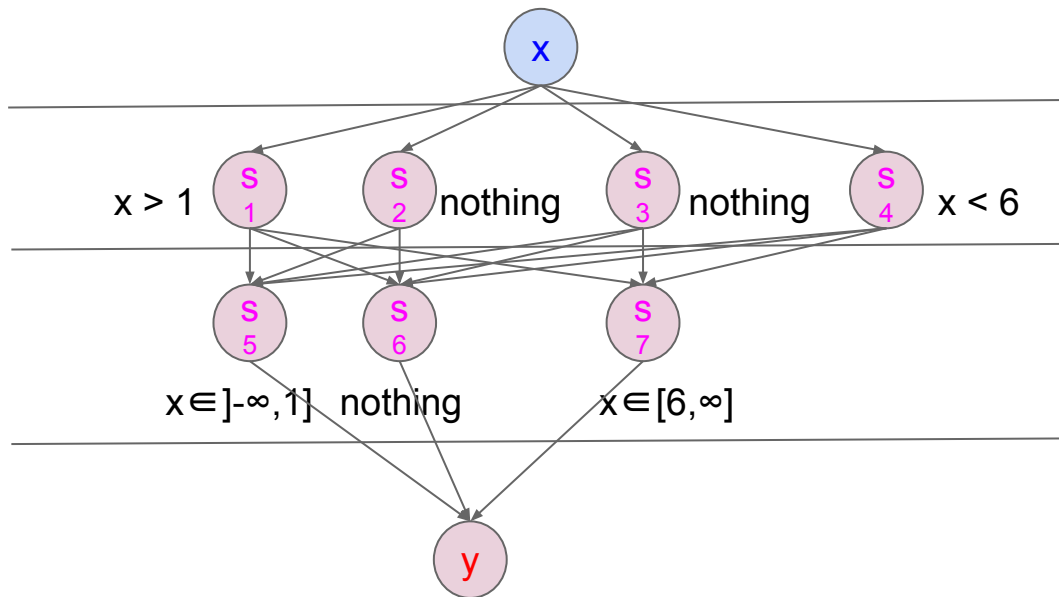
Multilayer Perceptrons

Regularization



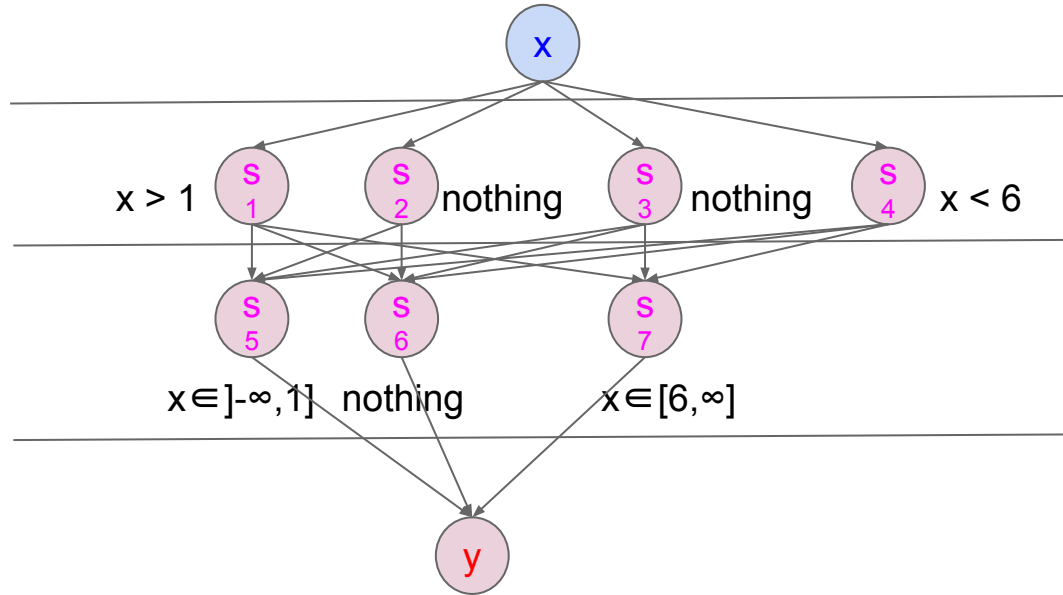
Multilayer Perceptrons

Regularization



Multilayer Perceptrons

Regularization



Find solutions that
require less effort

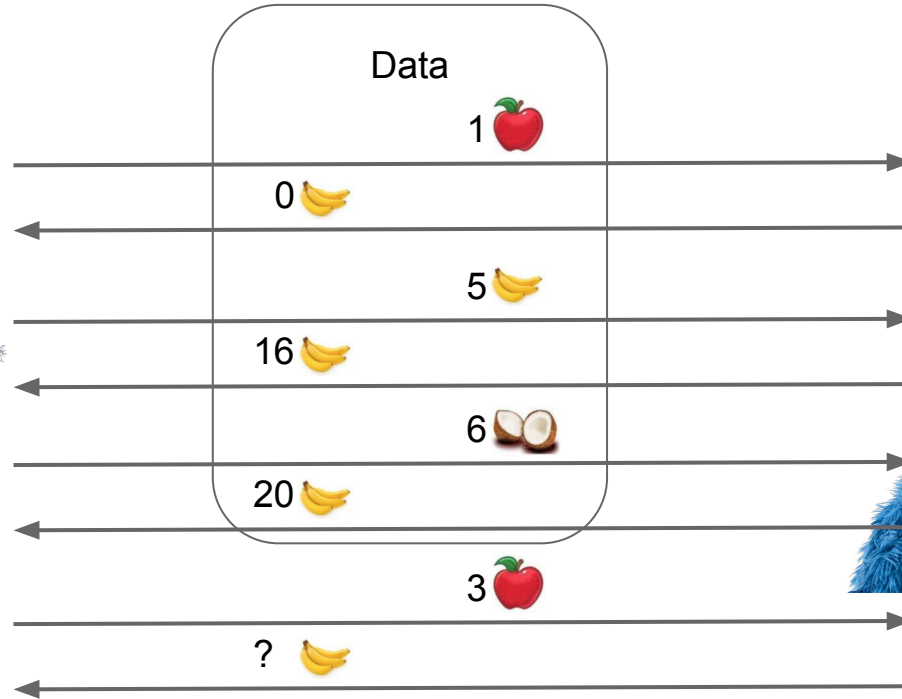
Using Discrete Variables



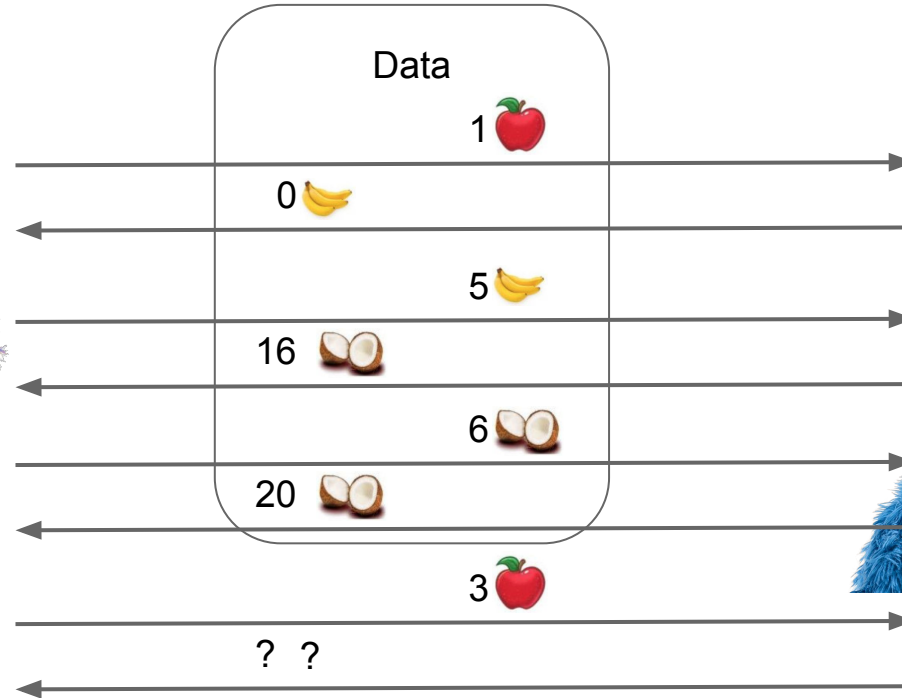
Data	
	1 🍏
0 🍌	
	5 🍏
16 🍌	
	6 🍏
20 🍌	
	3 🍏
? 🍌	



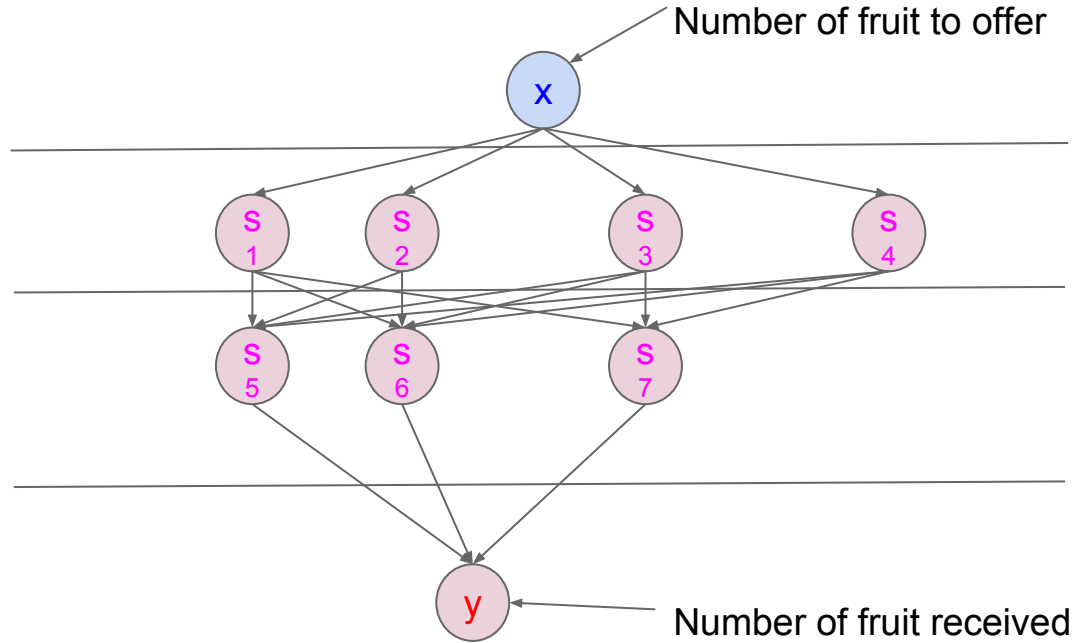
Using Discrete Variables



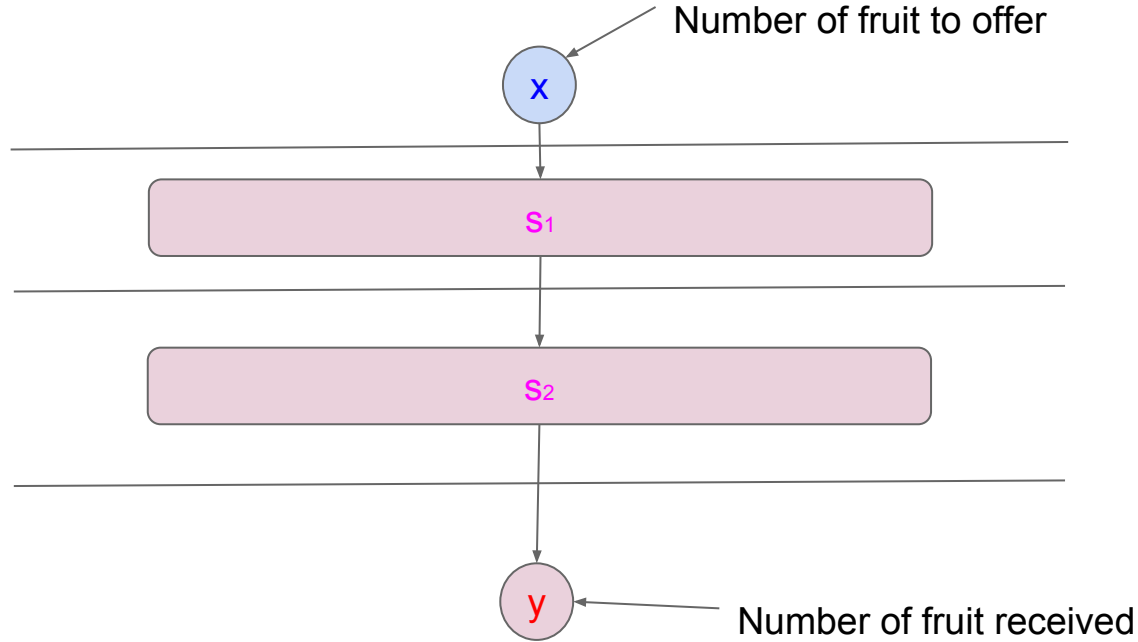
Using Discrete Variables



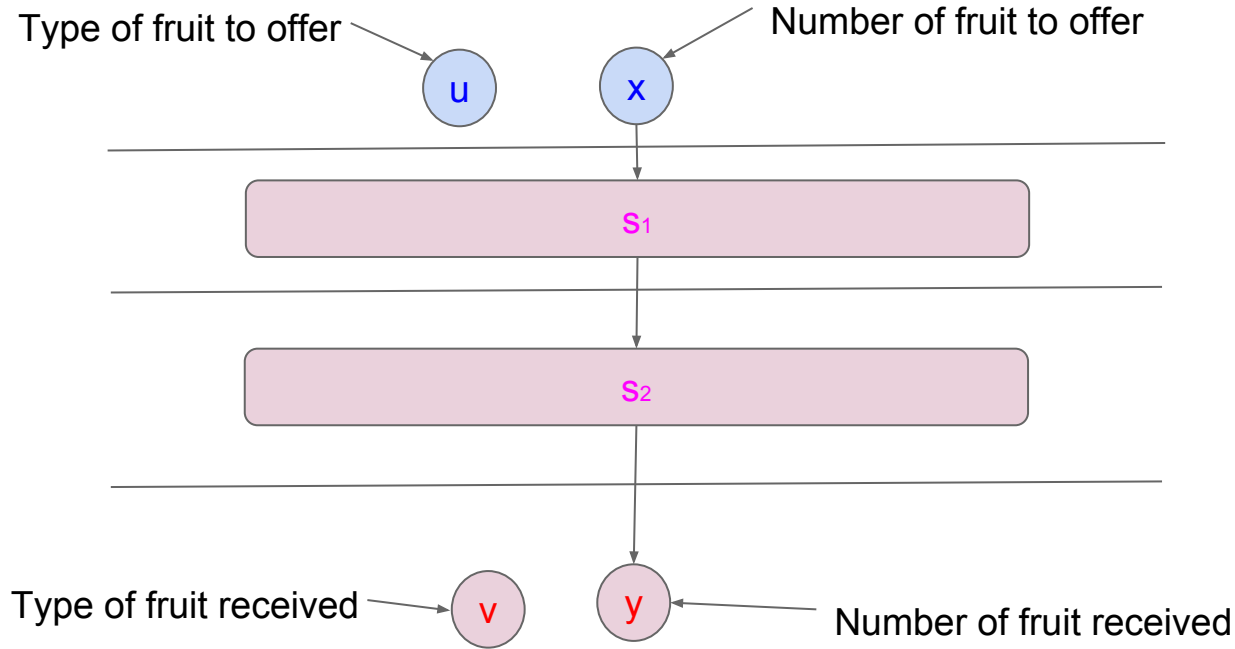
Using Discrete Variables



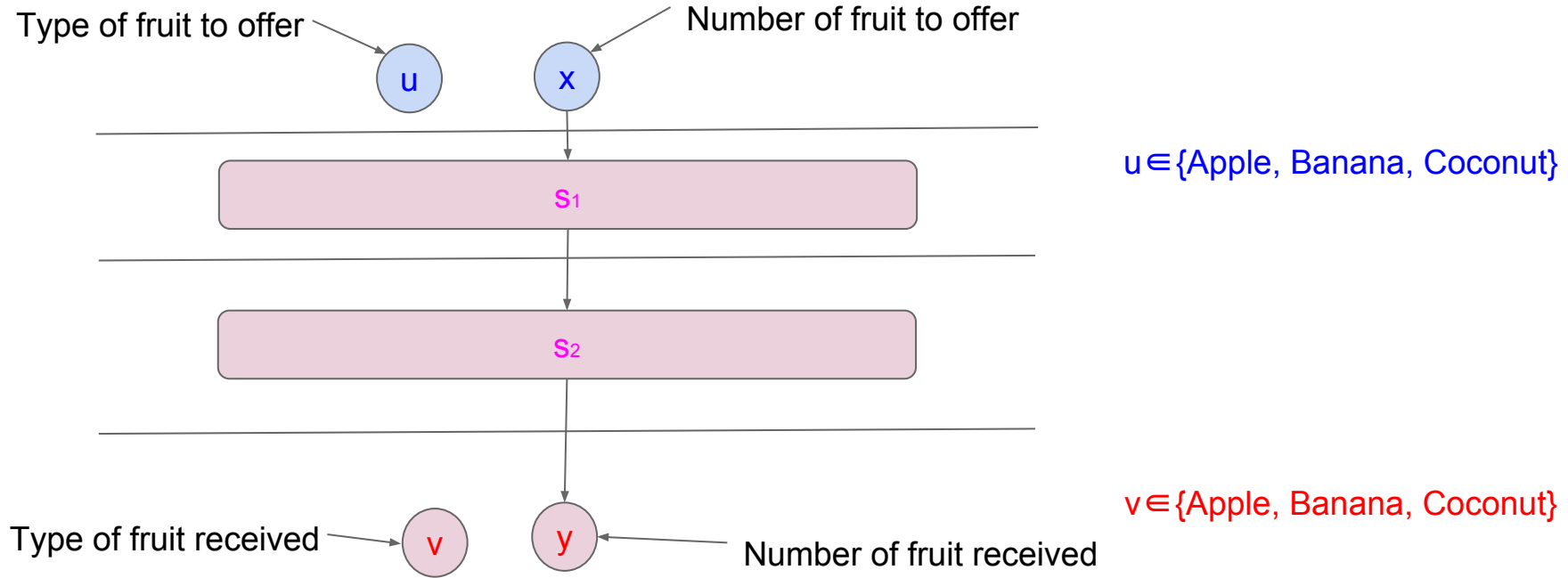
Using Discrete Variables



Using Discrete Variables



Using Discrete Variables



Using Discrete Variables

Lookup Tables

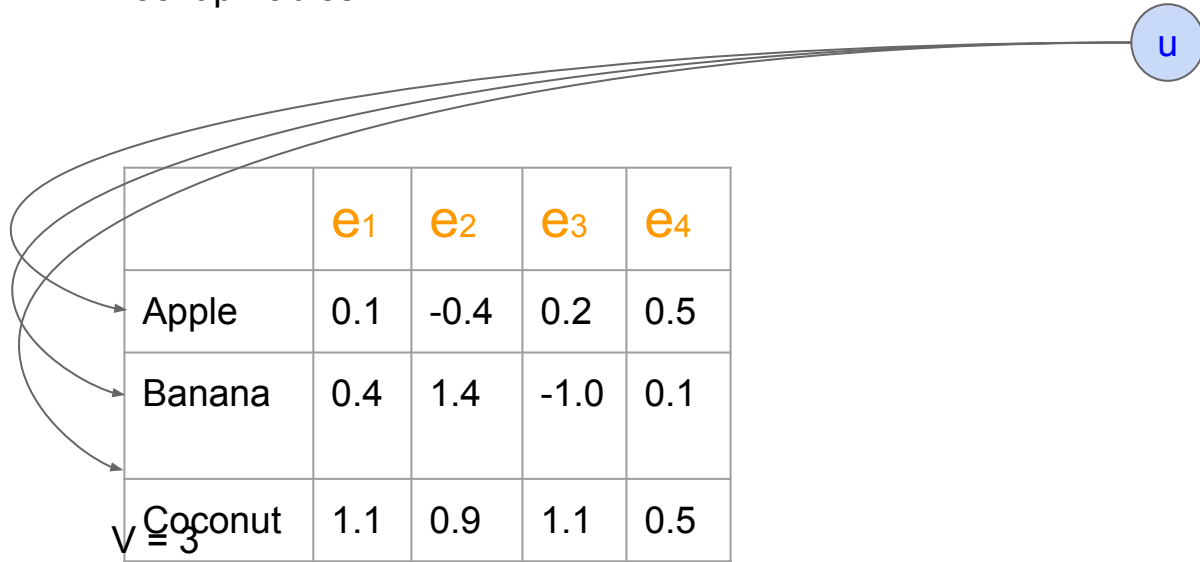


	e ₁	e ₂	e ₃	e ₄
Apple	0.1	-0.4	0.2	0.5
Banana	0.4	1.4	-1.0	0.1
Coconut	1.1	0.9	1.1	0.5

$V = 3$

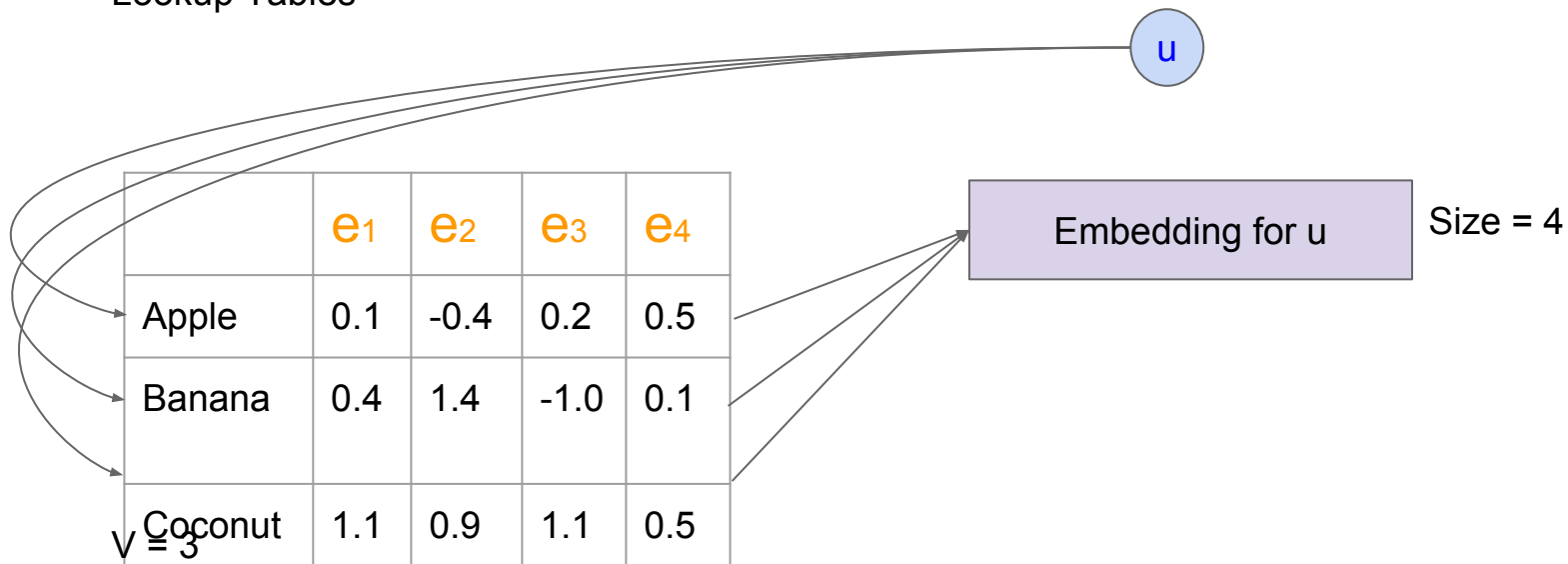
Using Discrete Variables

Lookup Tables



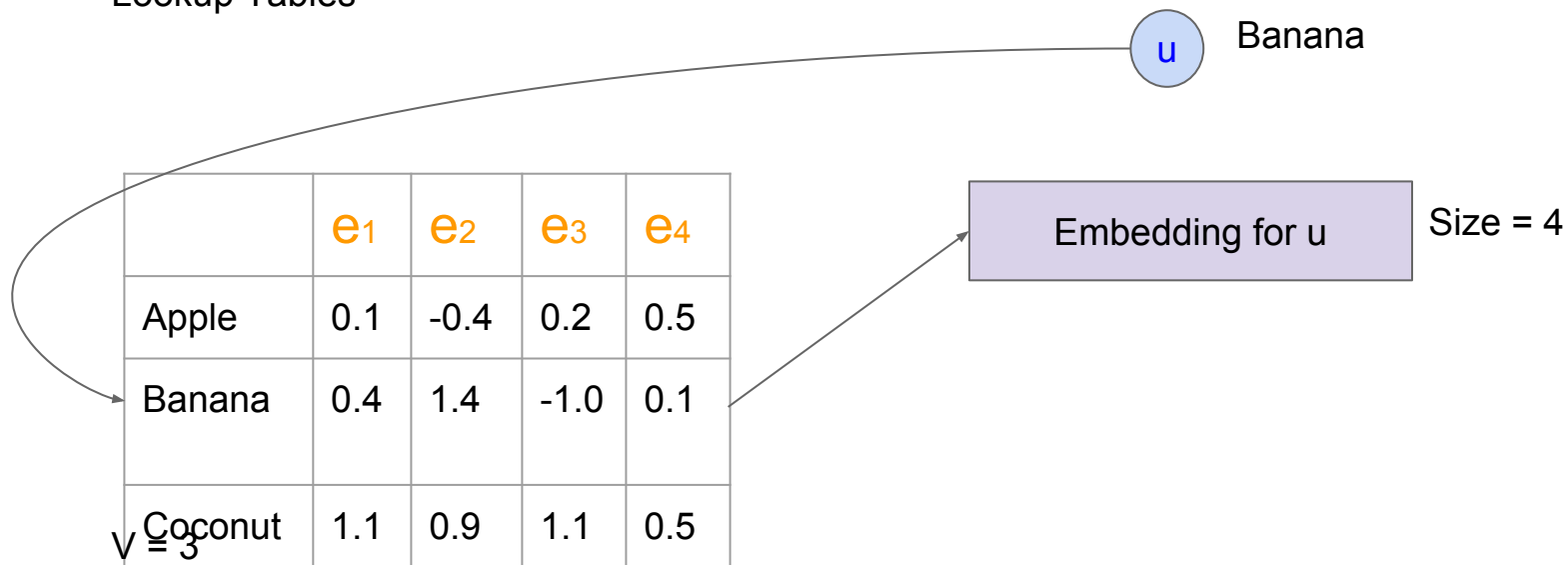
Using Discrete Variables

Lookup Tables



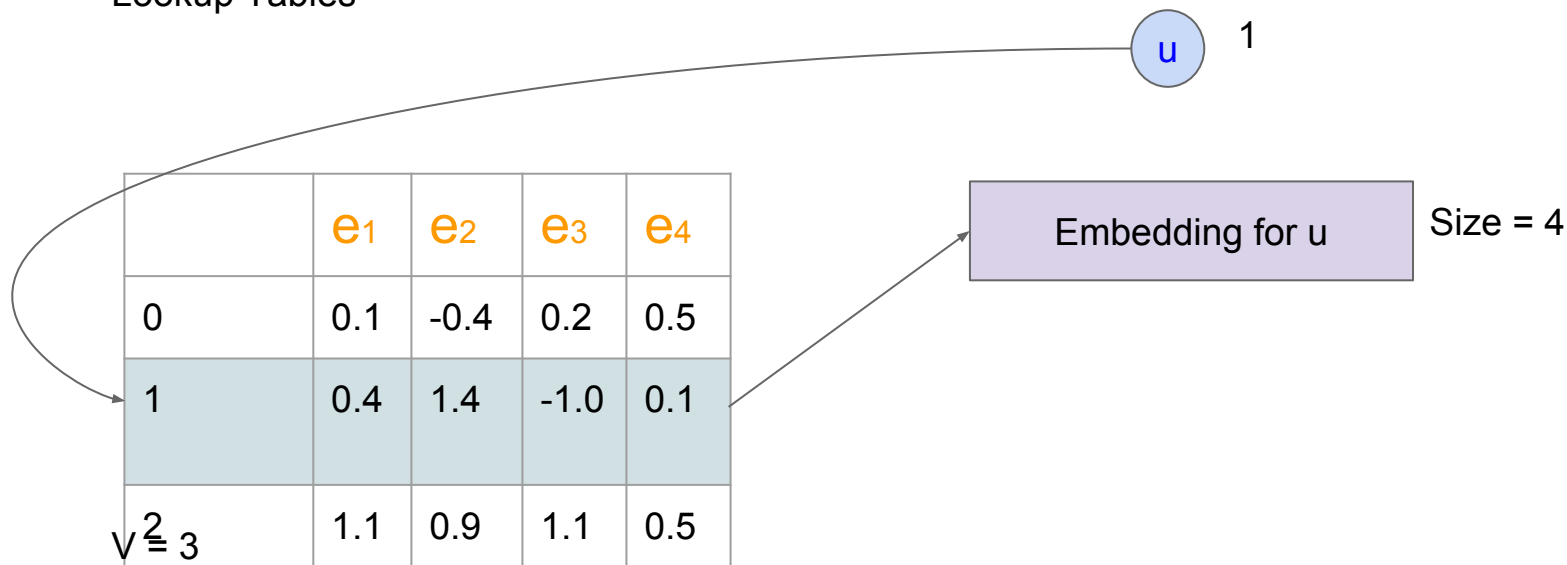
Using Discrete Variables

Lookup Tables



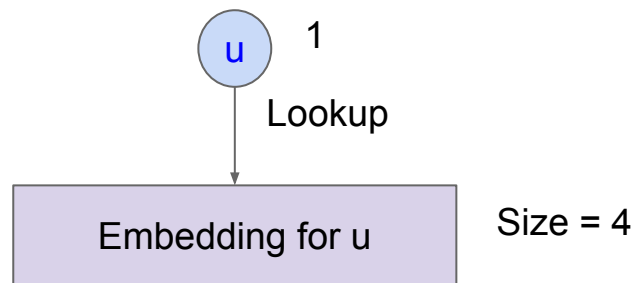
Using Discrete Variables

Lookup Tables

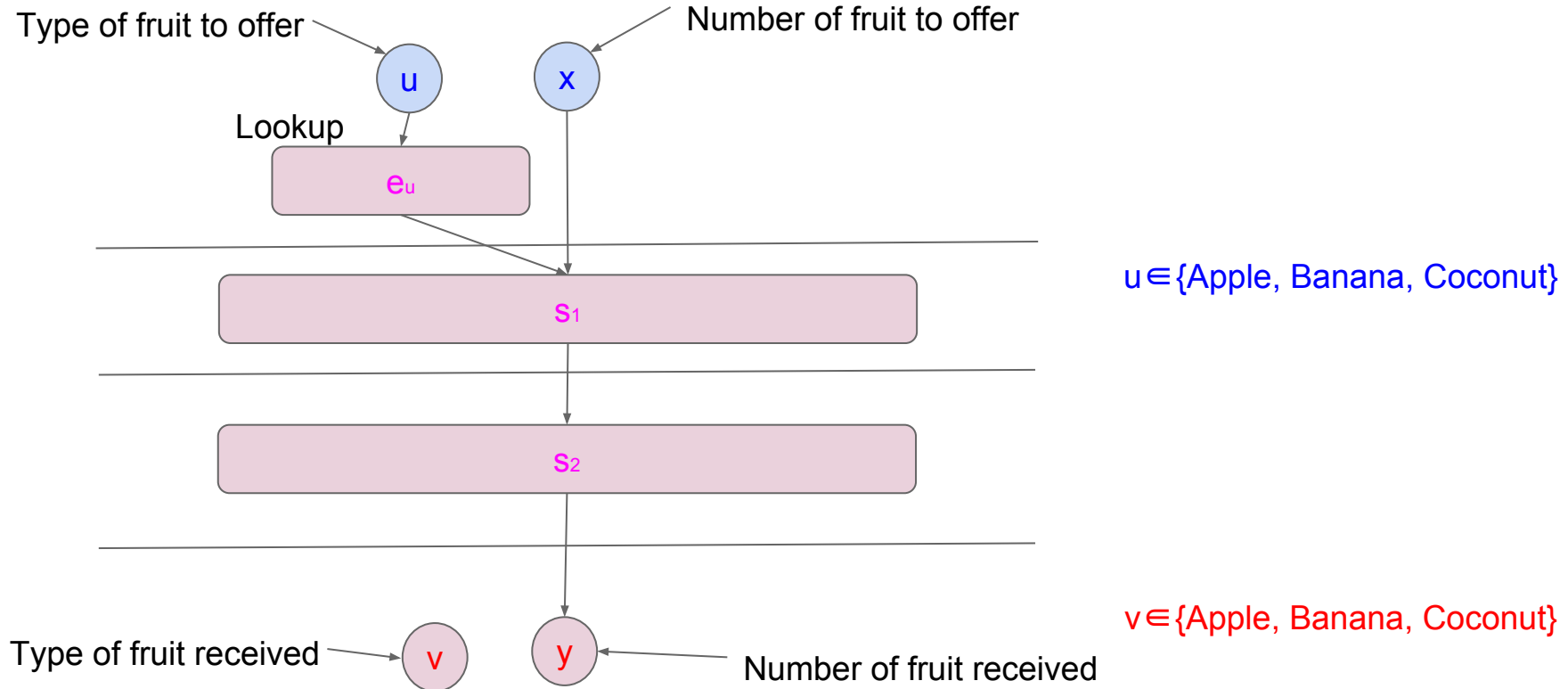


Using Discrete Variables

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

$V = 3$

Input vector

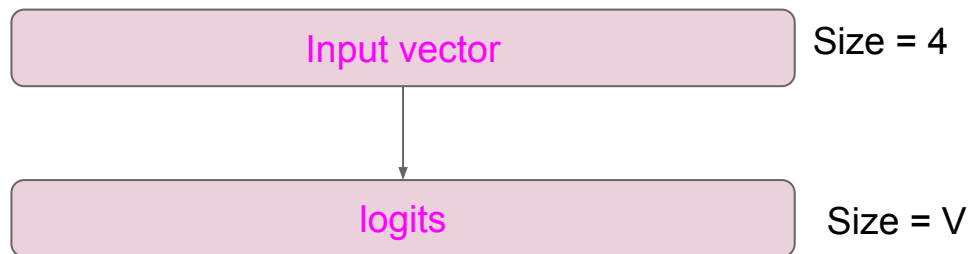
Size = 4

	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

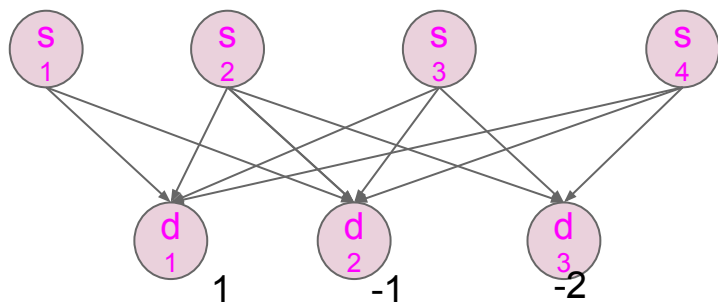
$V = 3$



	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



Input Vector

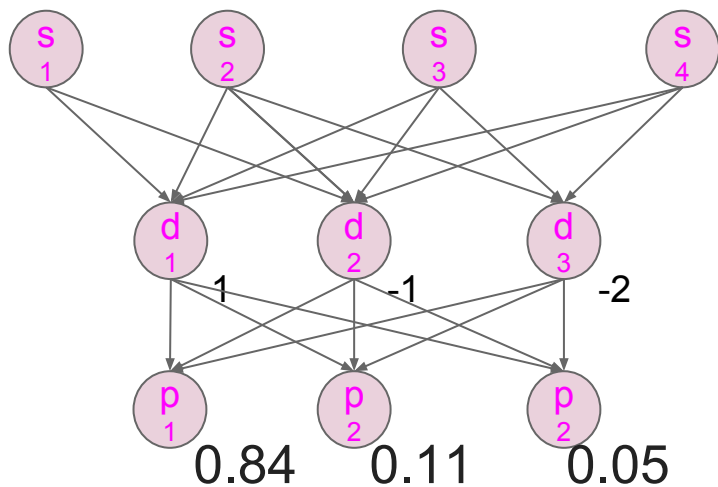
$V = 3$

Logits

	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

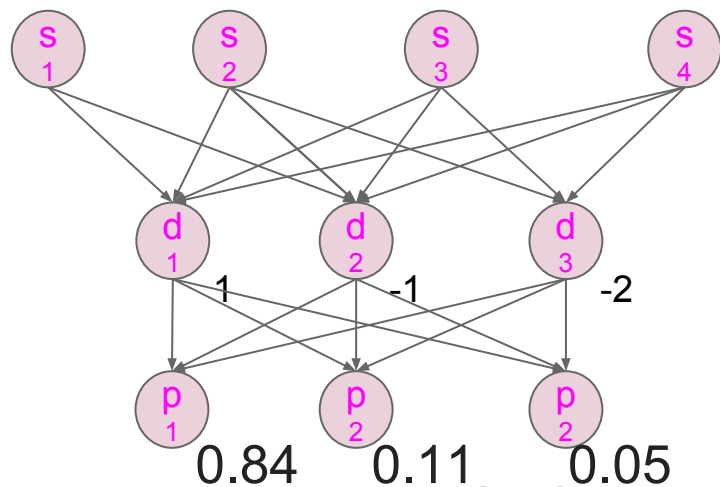


$V = 3$

	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



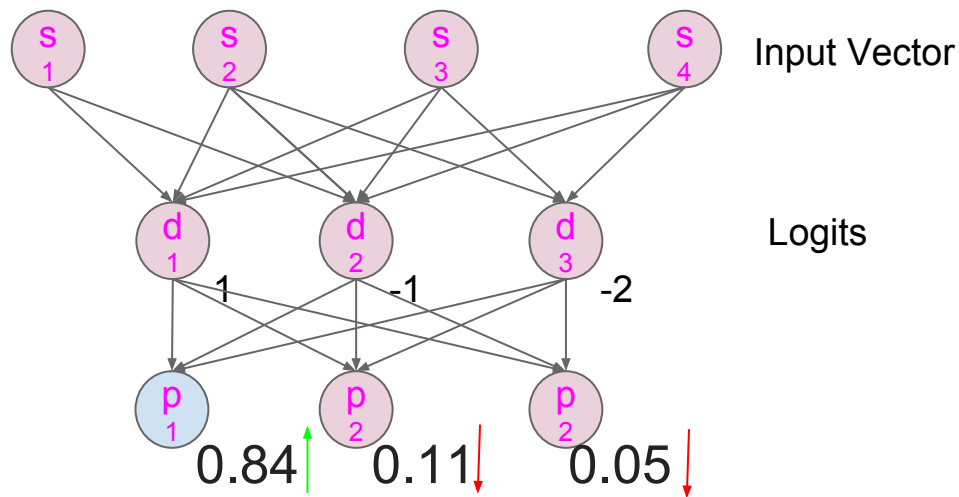
$V = 3$

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

	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

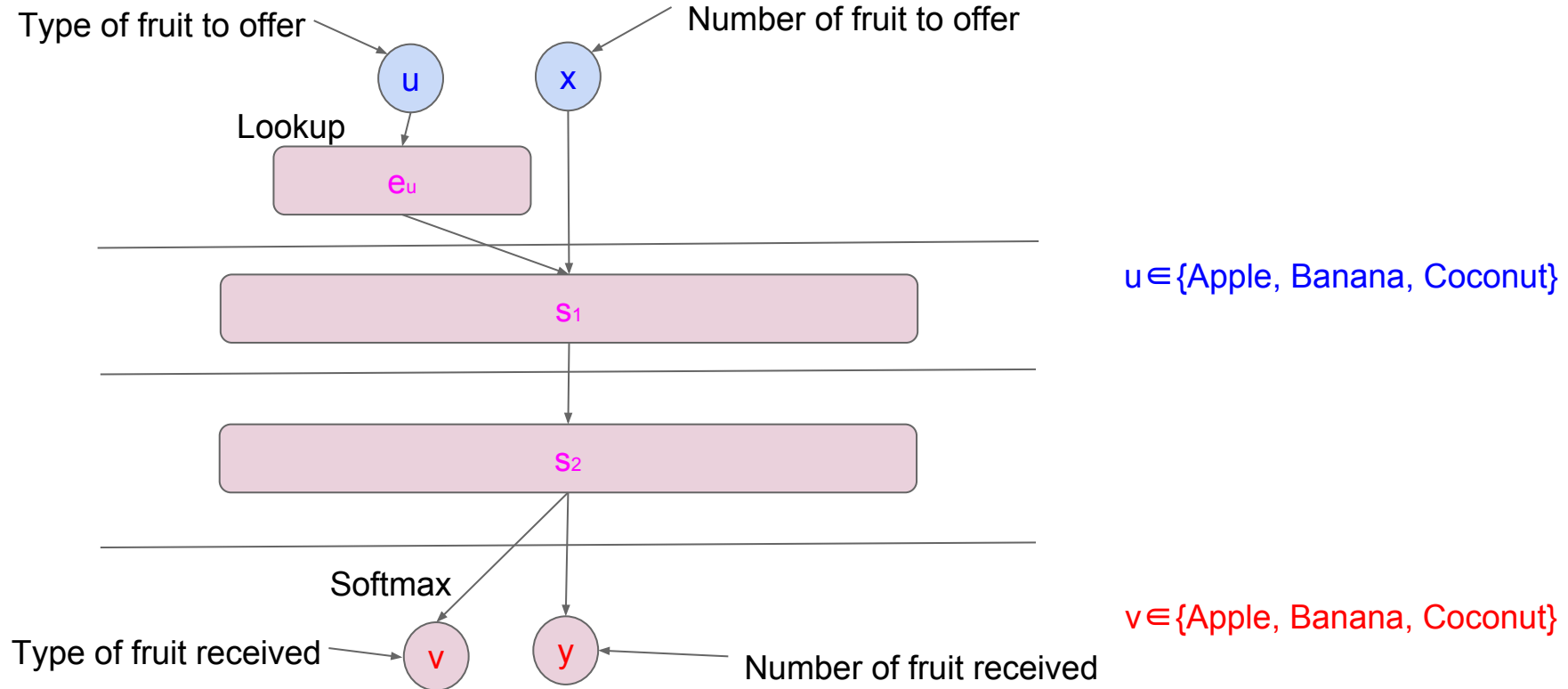


$V = 3$

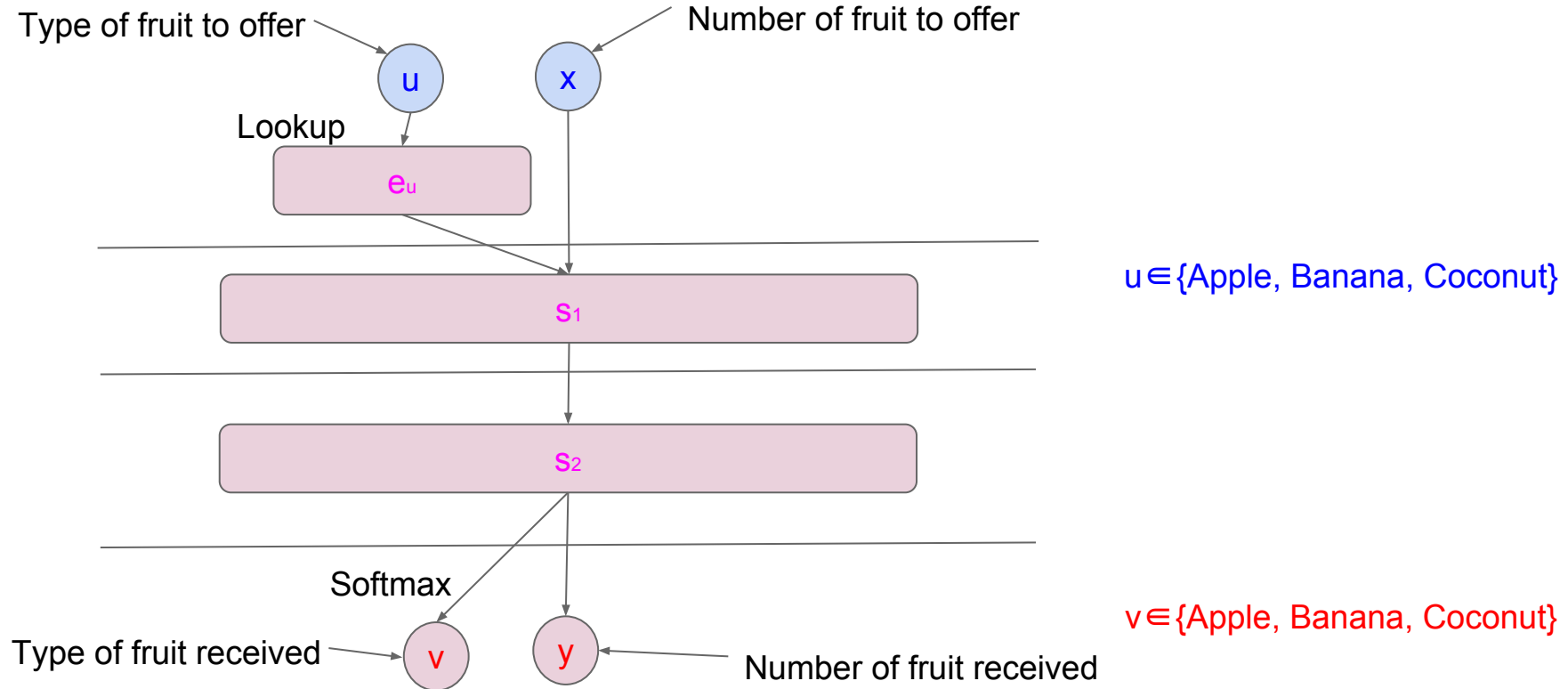
Apple

	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

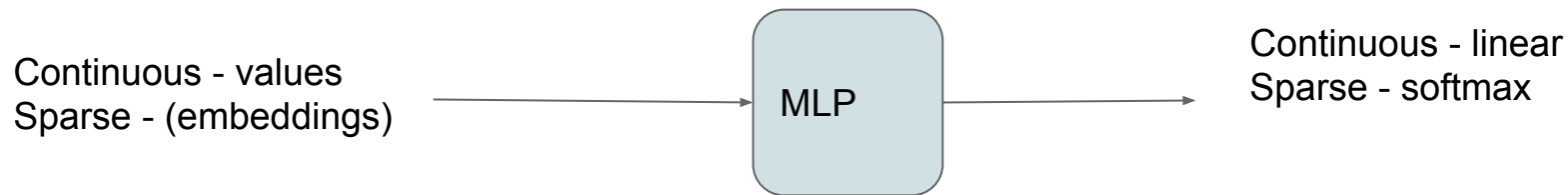
Using Discrete Variables

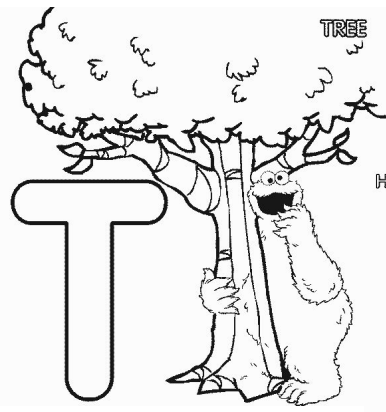


Using Discrete Variables

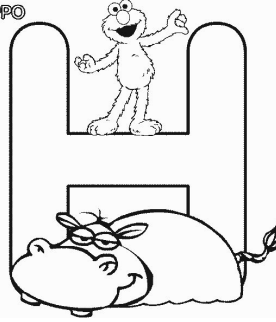


Summary

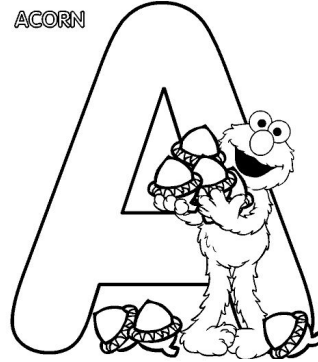




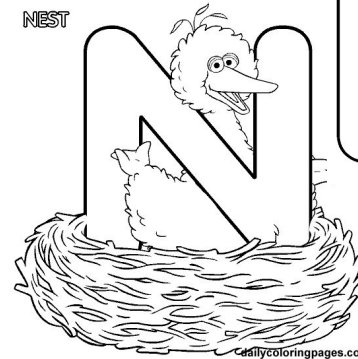
HIPPO



ACORN



NEST



KANGAROO



STRAWBERRY



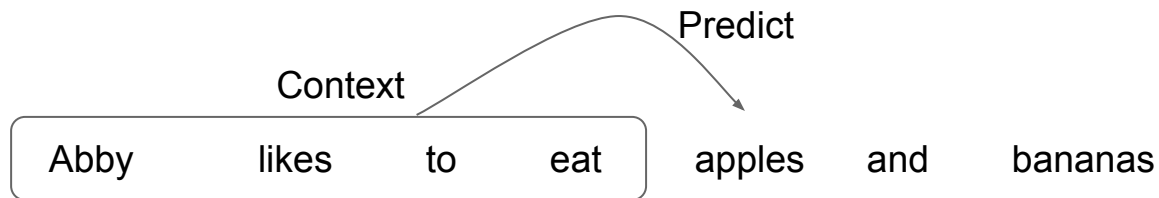
Example Applications

Embedding Pretraining (Collobert et al, 2011)

Abby likes to eat apples and bananas

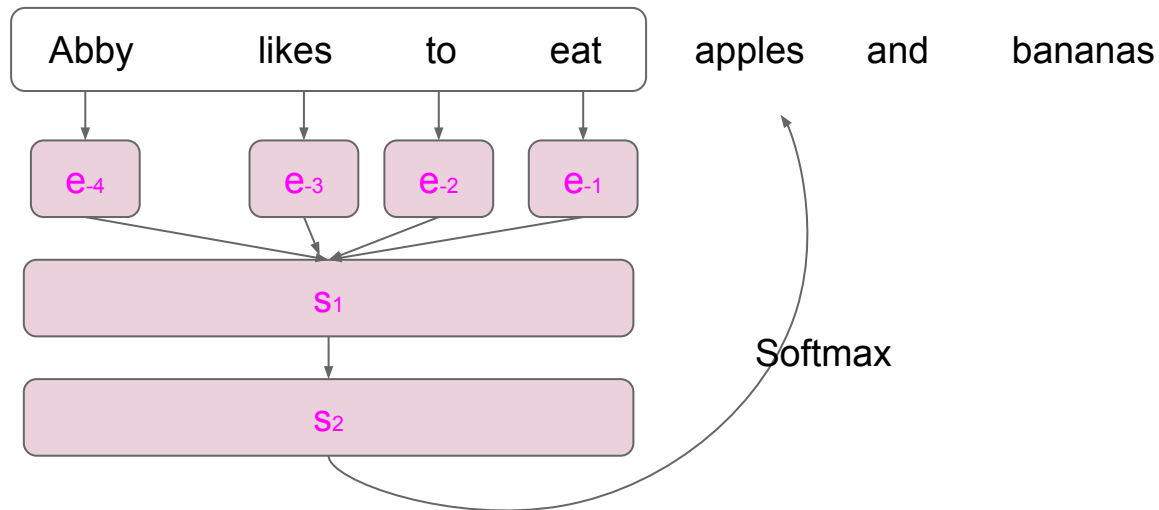
Example Applications

Embedding Pretraining (Collobert et al, 2011)



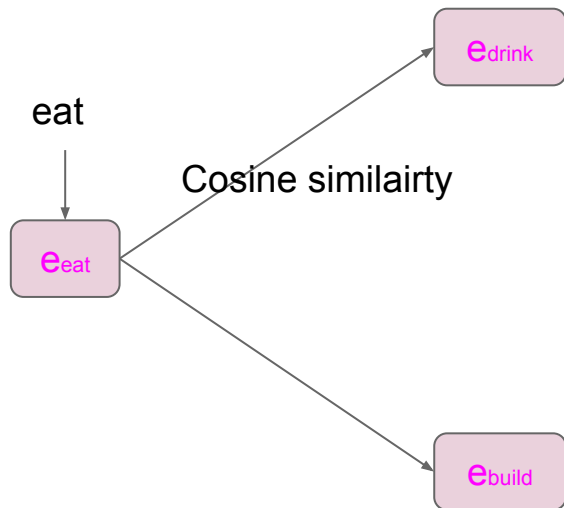
Example Applications

Embedding Pretraining (Collobert et al, 2011)



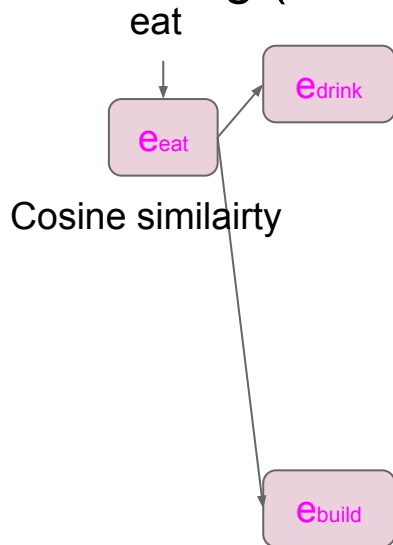
Example Applications

Embedding Pretraining (Collobert et al, 2011)



Example Applications

Embedding Pretraining (Collobert et al, 2011)

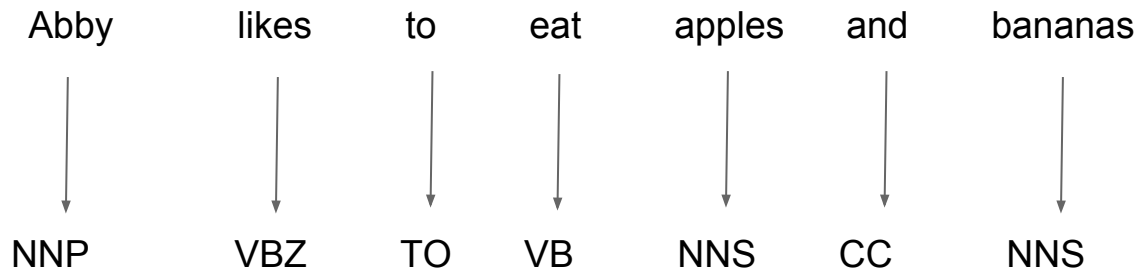


Example Applications

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

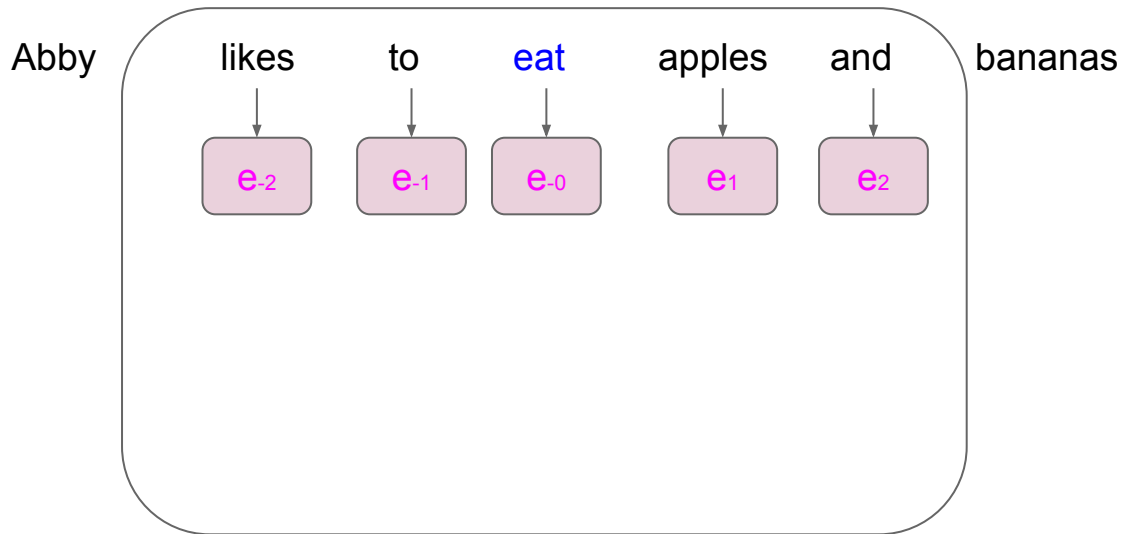
Example Applications

Window-based Tagging (Collobert et al, 2011)



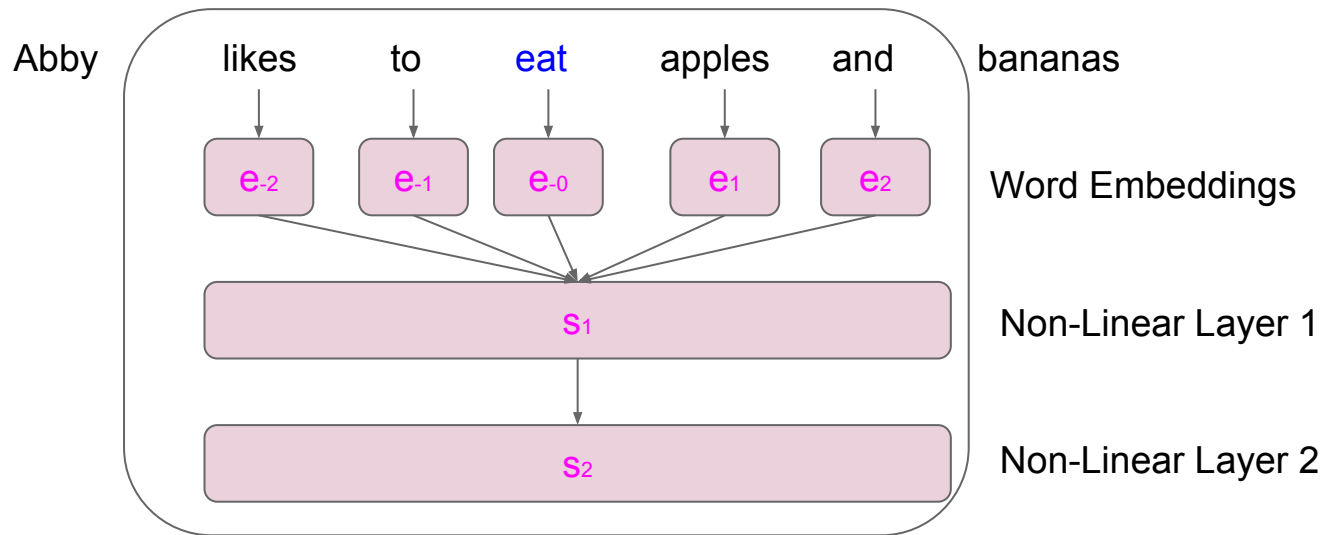
Example Applications

Window-based Tagging (Collobert et al, 2011)



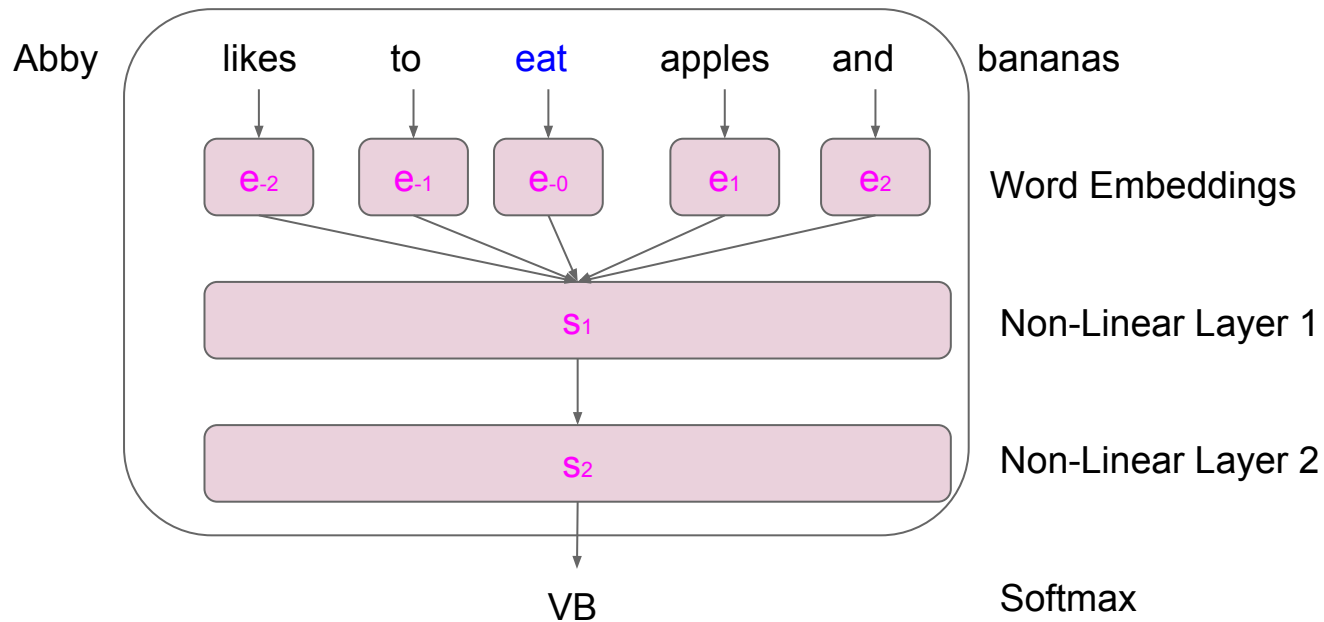
Example Applications

Window-based Tagging (Collobert et al, 2011)



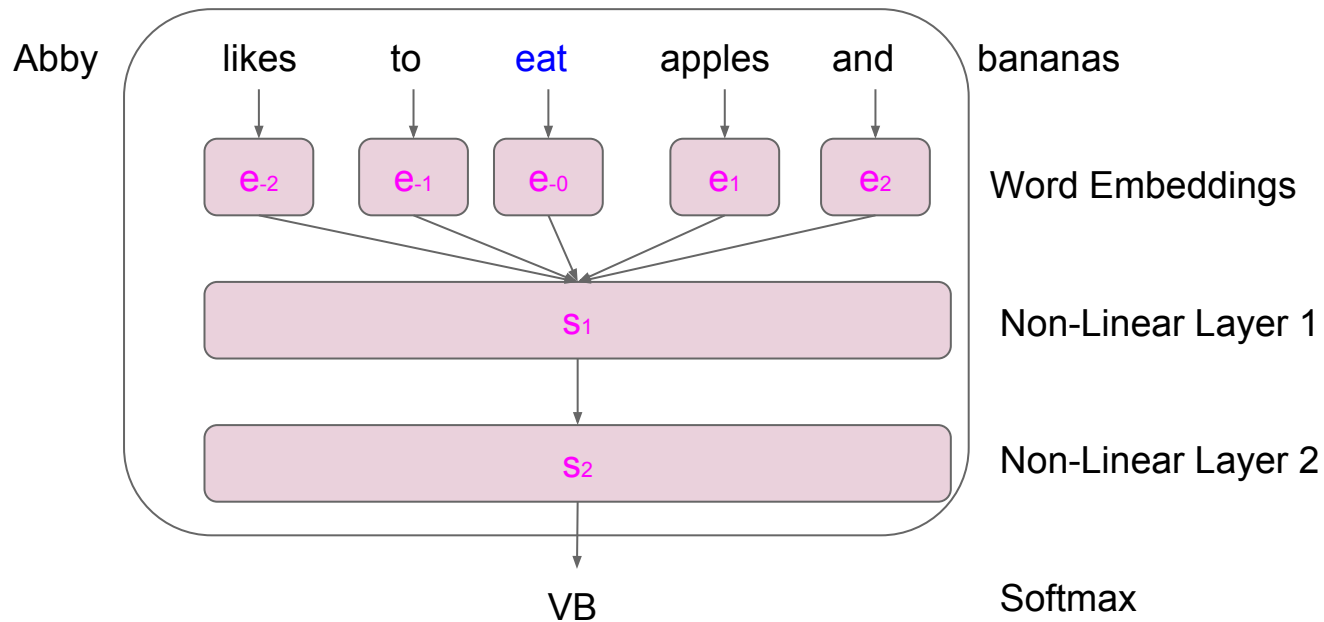
Example Applications

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

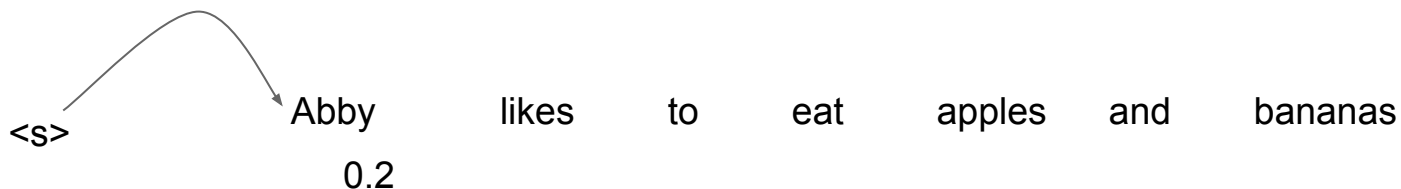
Translation 2 Abby likes to eat apples and bananas

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)



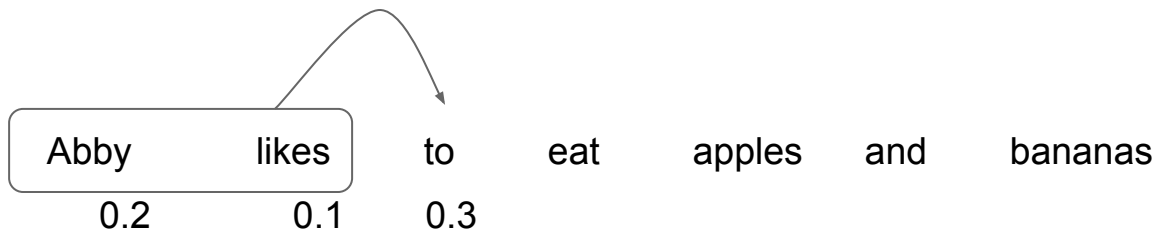
Example Applications

Translation Rescoring (Devlin et al, 2014)



Example Applications

Translation Rescoring (Devlin et al, 2014)



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)

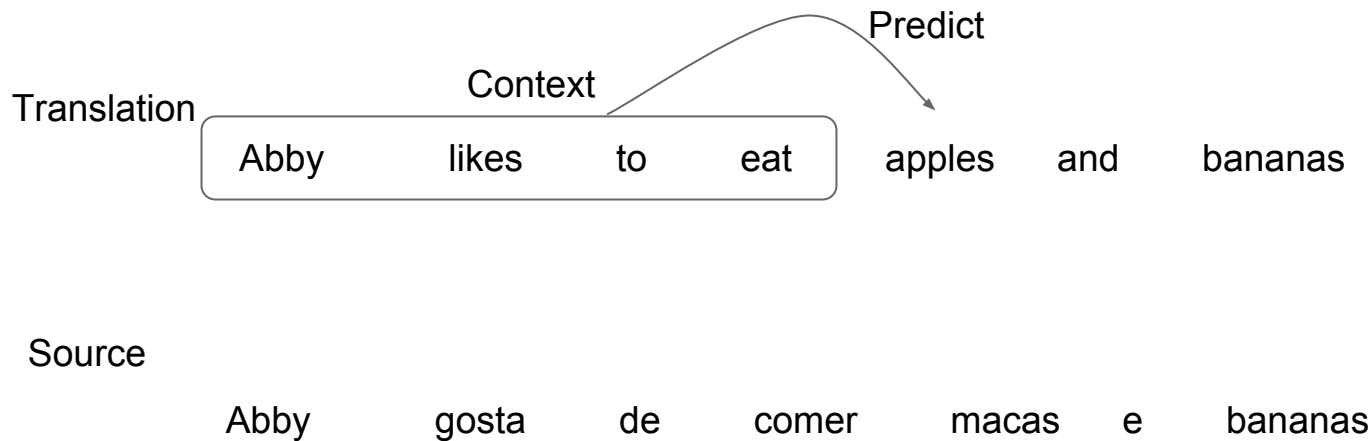
John does to eat coconuts and bananas 0.00003

Abby likes to eat apples and bananas 0.000378

Abby dislikes to drink apples and bananas 0.00012

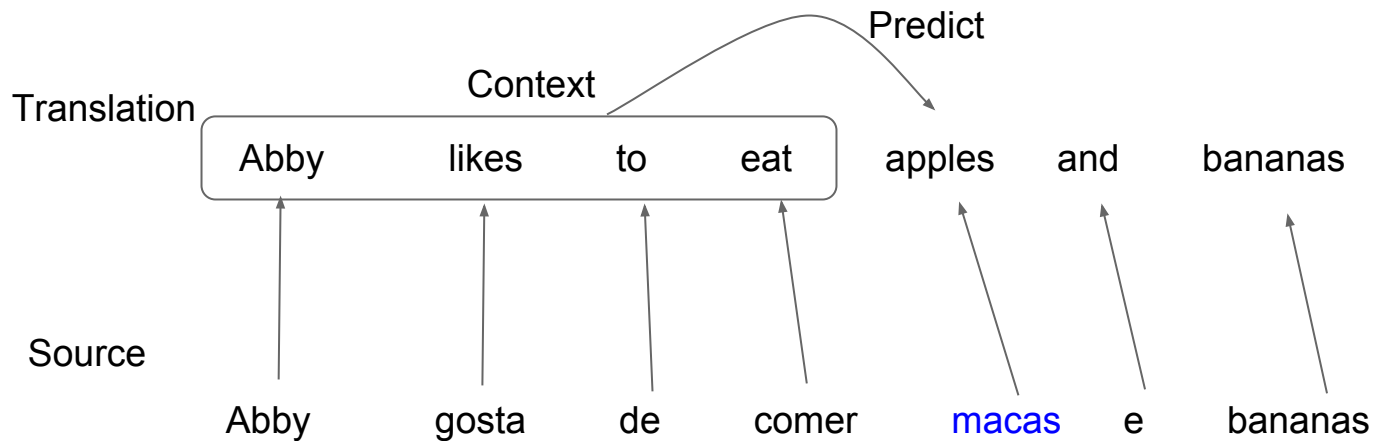
Example Applications

Translation Rescoring (Devlin et al, 2014)



Example Applications

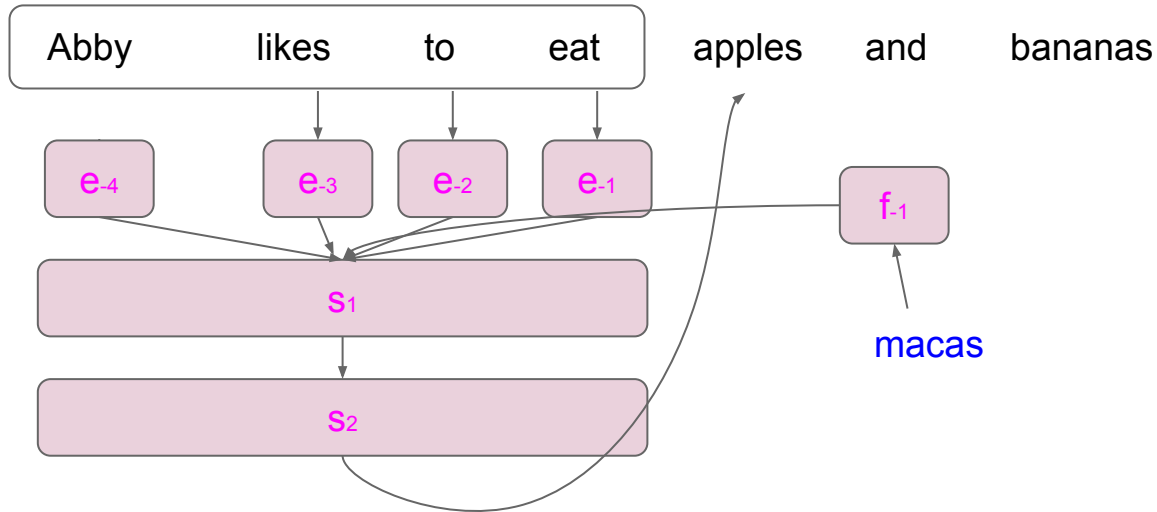
Translation Rescoring (Devlin et al, 2014)



Example Applications

Translation Rescoring (Devlin et al, 2014)

Translation



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

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

$$y = wx + b$$

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

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



Easy!

Computation Graphs are our friends

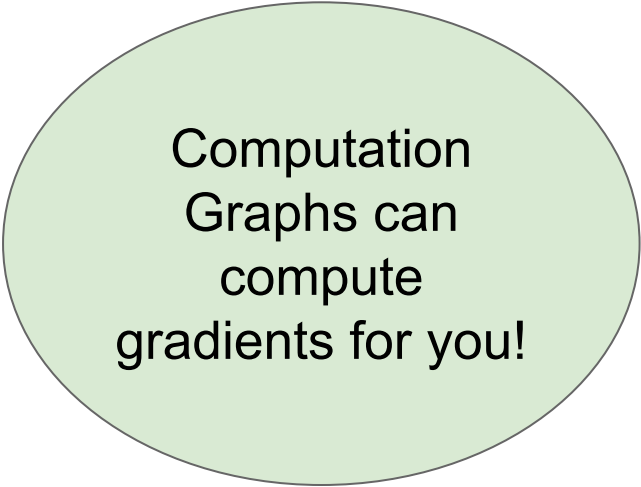
$$y = wx + b + \tanh(yx + b)^2$$



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

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

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

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

Computation Graphs are our friends

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

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

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

Computation Graphs are our friends

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

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

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

Computation Graphs are our friends

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

$$\mathbf{y} = \mathbf{o} + \mathbf{b}$$

$$\mathbf{o} = \mathbf{w}\mathbf{x}$$

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

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

Computation Graphs are our friends

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

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

$$y = o + b$$

$$o = \mathbf{w}x$$

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

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

Computation Graphs are our friends

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

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

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

$$c = d^2$$

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

$$y = o + b$$

$$o = wx$$

Computation Graphs are our friends

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

$$\frac{\partial C}{\partial \mathbf{w}} = \sum_n \frac{\partial c_n}{\partial d_n} \frac{\partial d_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{o}_n} \frac{\partial \mathbf{o}_n}{\partial \mathbf{w}}$$

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

$$c = d^2$$

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

$$\mathbf{y} = \mathbf{o} + \mathbf{b}$$

$$\mathbf{o} = \mathbf{w}\mathbf{x}$$

Computation Graphs are our friends

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

$$\frac{\partial C}{\partial \mathbf{w}} = \sum_n \frac{\partial C_n}{\partial \mathbf{d}_n} \frac{\partial \mathbf{d}_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{o}_n} \frac{\partial \mathbf{o}_n}{\partial \mathbf{w}}$$

$$\frac{\partial C}{\partial \mathbf{b}} = \sum_n \frac{\partial C_n}{\partial \mathbf{d}_n} \frac{\partial \mathbf{d}_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{b}}$$

$$c = d^2$$

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

$$y = o + b$$

$$o = \mathbf{w}x$$

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) \rightarrow z

backward(x,y,dz) \rightarrow 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

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

$$c = d^2$$

Power 2

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

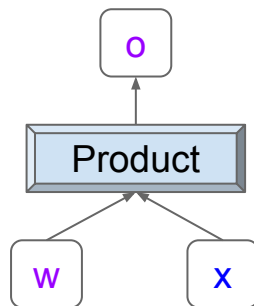
Sub

$$y = o + b$$

Add

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

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



$$o = wx$$

Computation Graphs are our friends

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

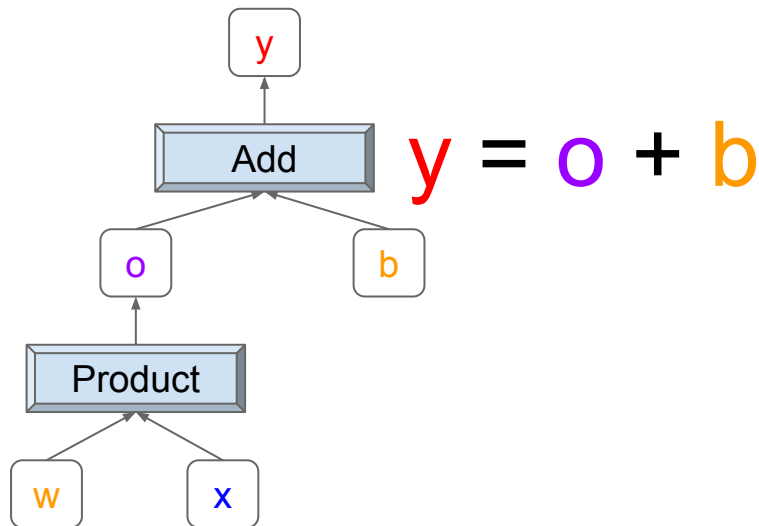
$$c = d^2$$
$$d = y - \hat{y}$$

Power 2

Sub

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

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

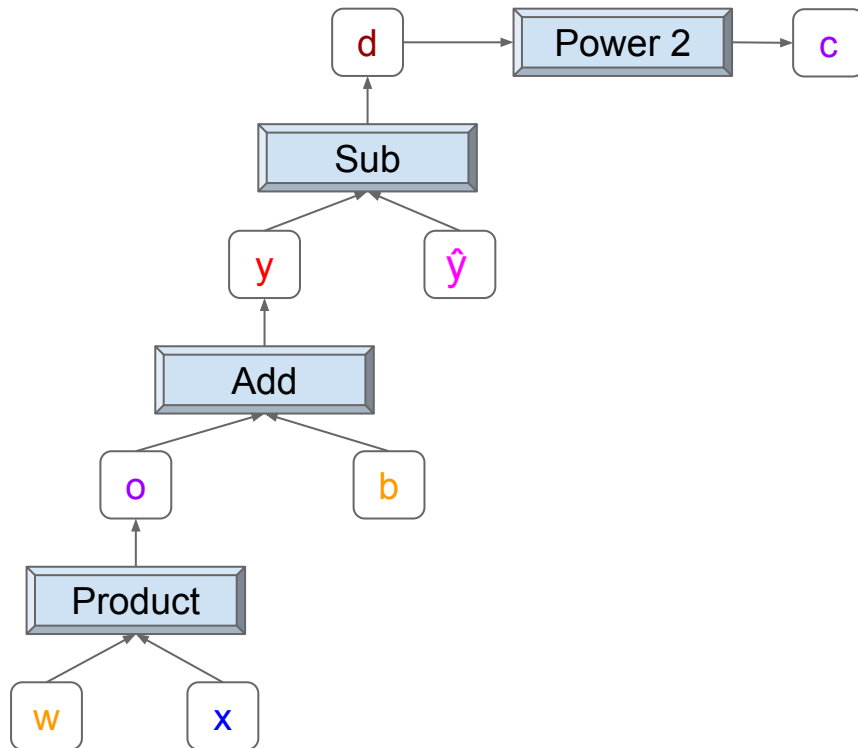


Computation Graphs are our friends

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

$$\frac{\partial C}{\partial \mathbf{w}} = \sum_n \frac{\partial C_n}{\partial \mathbf{d}_n} \frac{\partial \mathbf{d}_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{o}_n} \frac{\partial \mathbf{o}_n}{\partial \mathbf{w}}$$

$$\frac{\partial C}{\partial \mathbf{b}} = \sum_n \frac{\partial C_n}{\partial \mathbf{d}_n} \frac{\partial \mathbf{d}_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{b}}$$

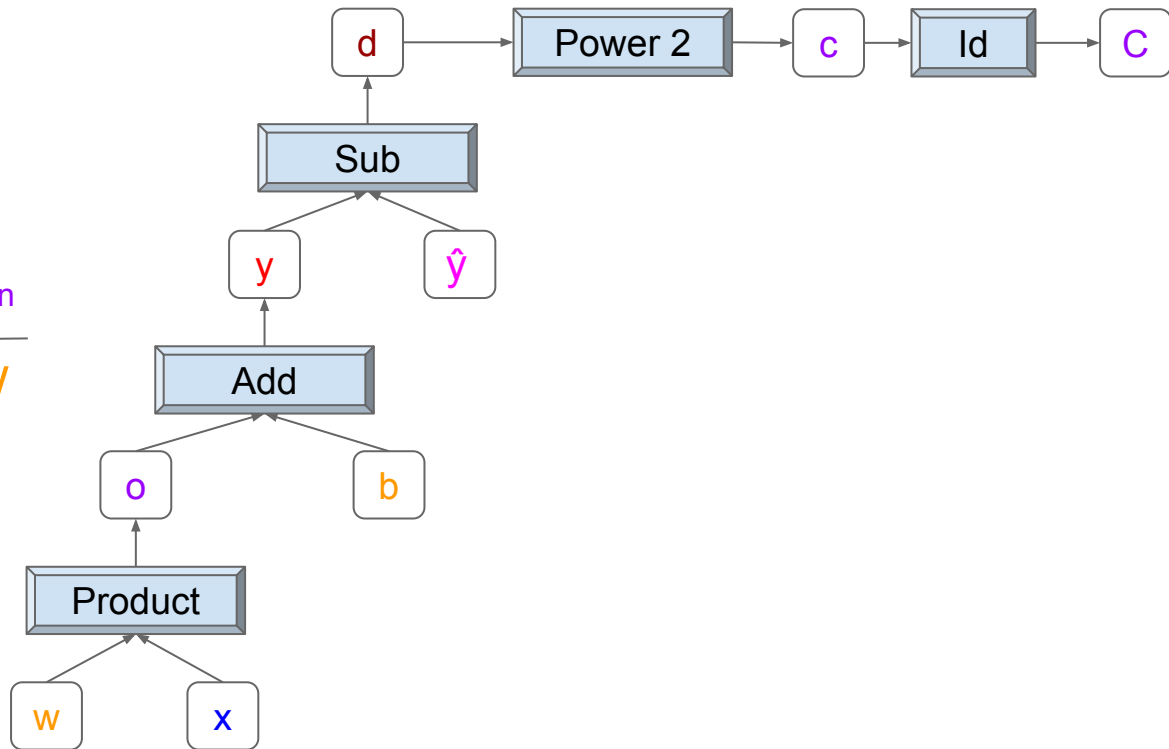


Computation Graphs are our friends

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

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

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

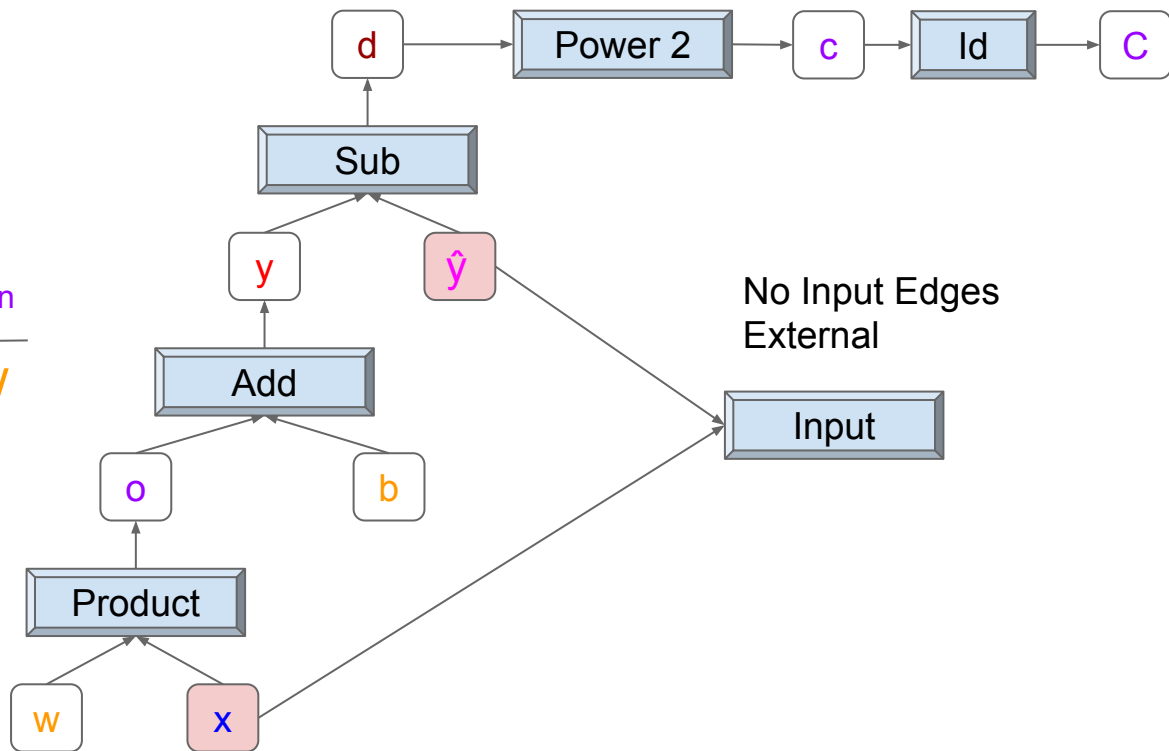


Computation Graphs are our friends

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

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

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

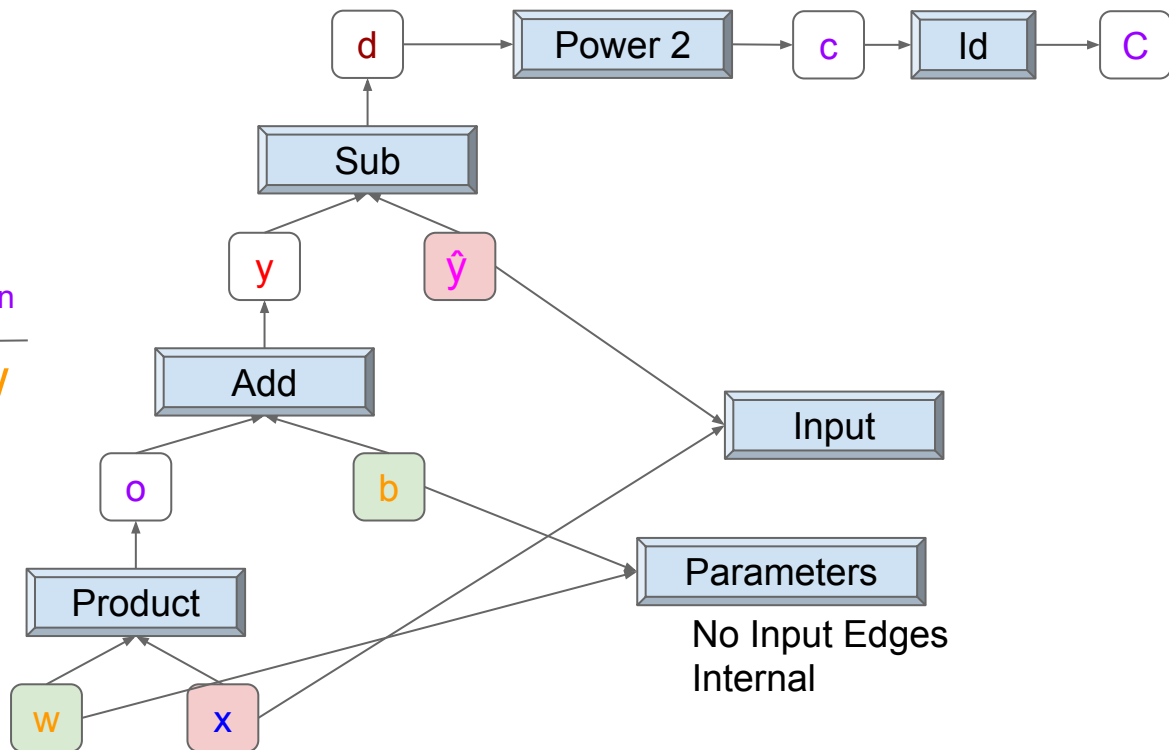


Computation Graphs are our friends

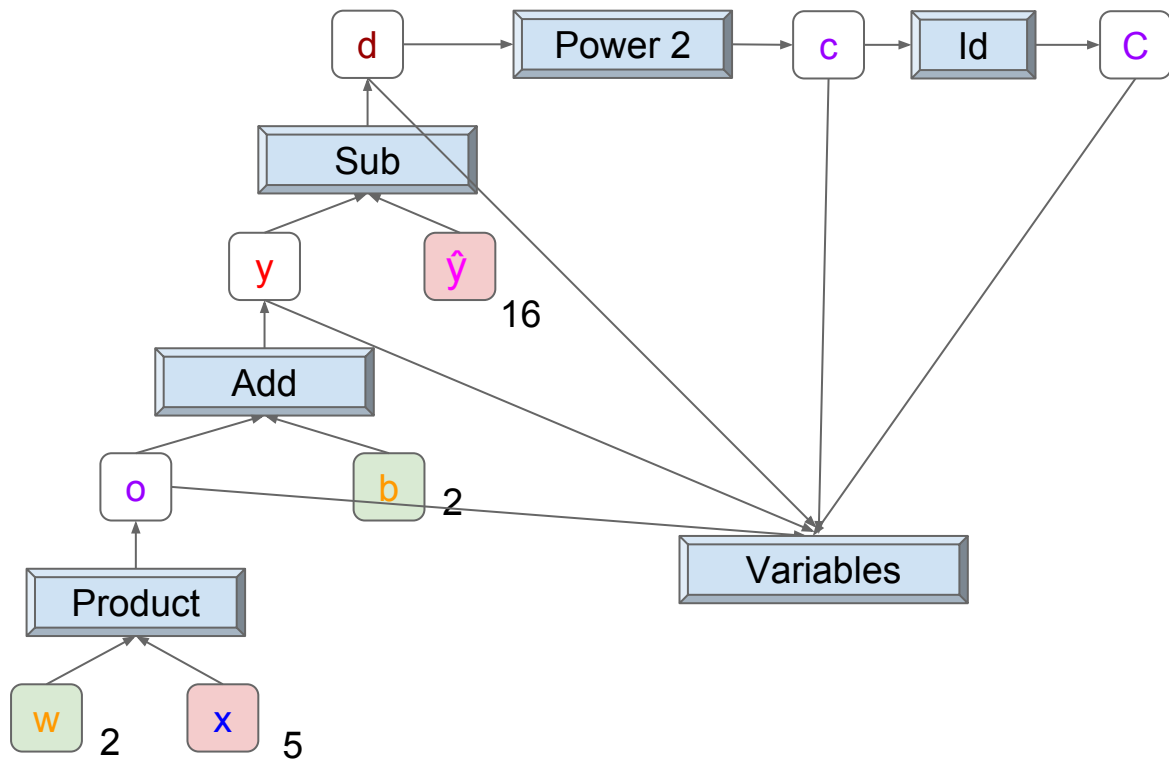
$$C(\mathbf{w}, \mathbf{b}) = \sum_{n \in \{0\}} C_n$$

$$\frac{\partial C}{\partial \mathbf{w}} = \sum_n \frac{\partial C_n}{\partial \mathbf{d}_n} \frac{\partial \mathbf{d}_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{o}_n} \frac{\partial \mathbf{o}_n}{\partial \mathbf{w}}$$

$$\frac{\partial C}{\partial \mathbf{b}} = \sum_n \frac{\partial C_n}{\partial \mathbf{d}_n} \frac{\partial \mathbf{d}_n}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{b}}$$

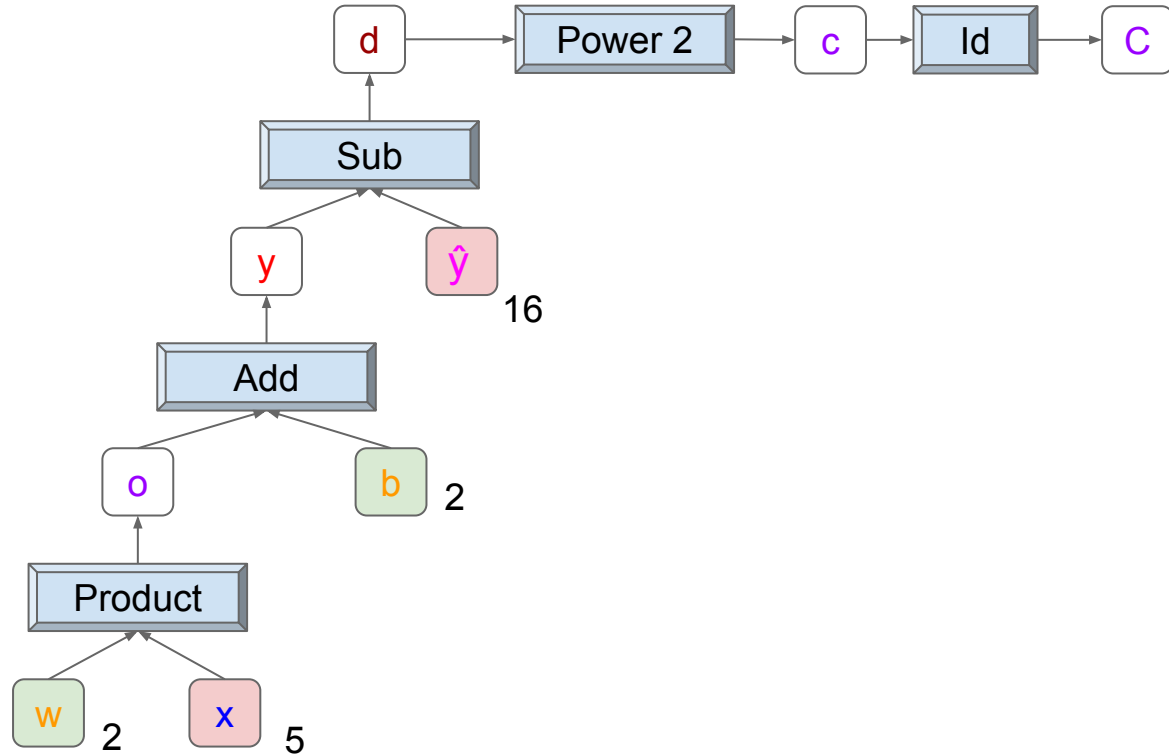


Computation Graphs are our friends



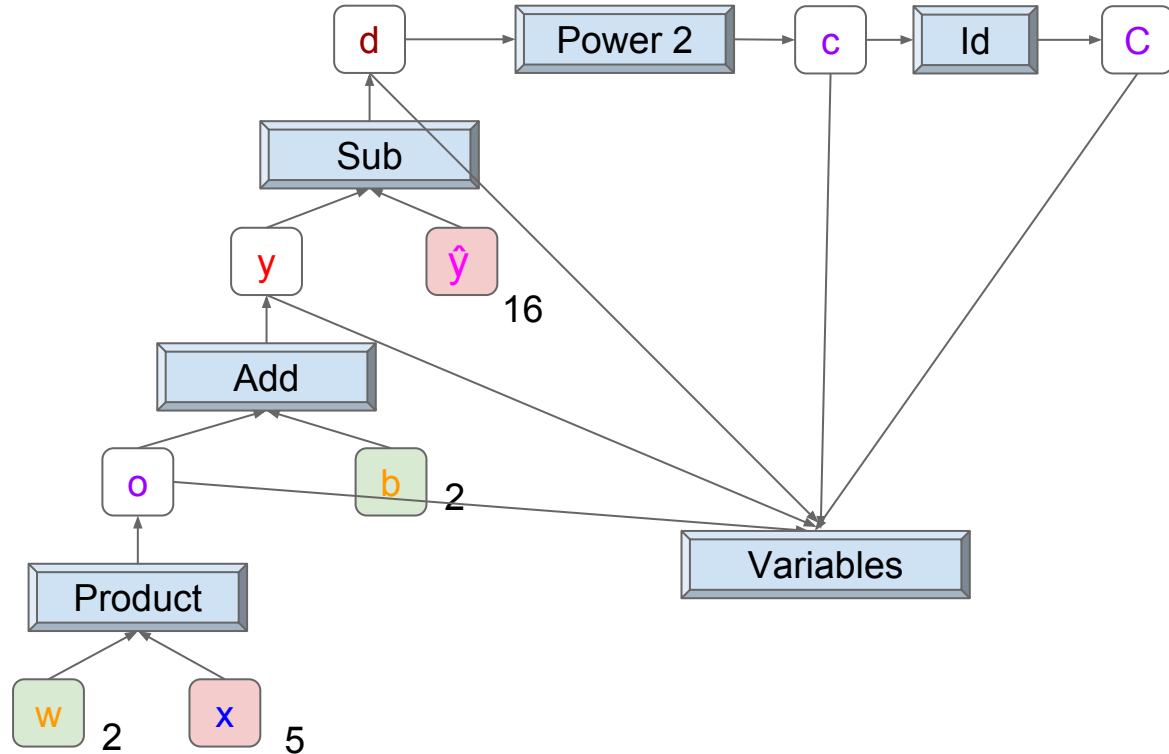
Computation Graphs are our friends

1-Initialize inputs



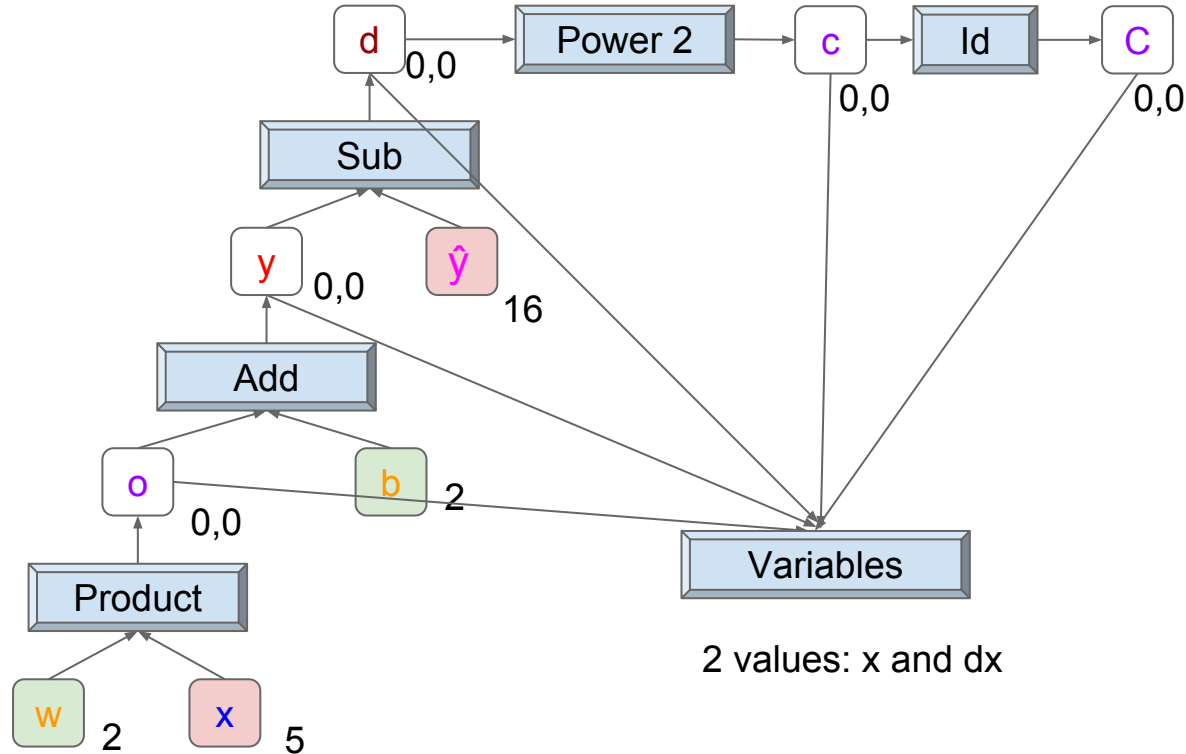
Computation Graphs are our friends

- 1-Initialize inputs
- 2-Initialize variables



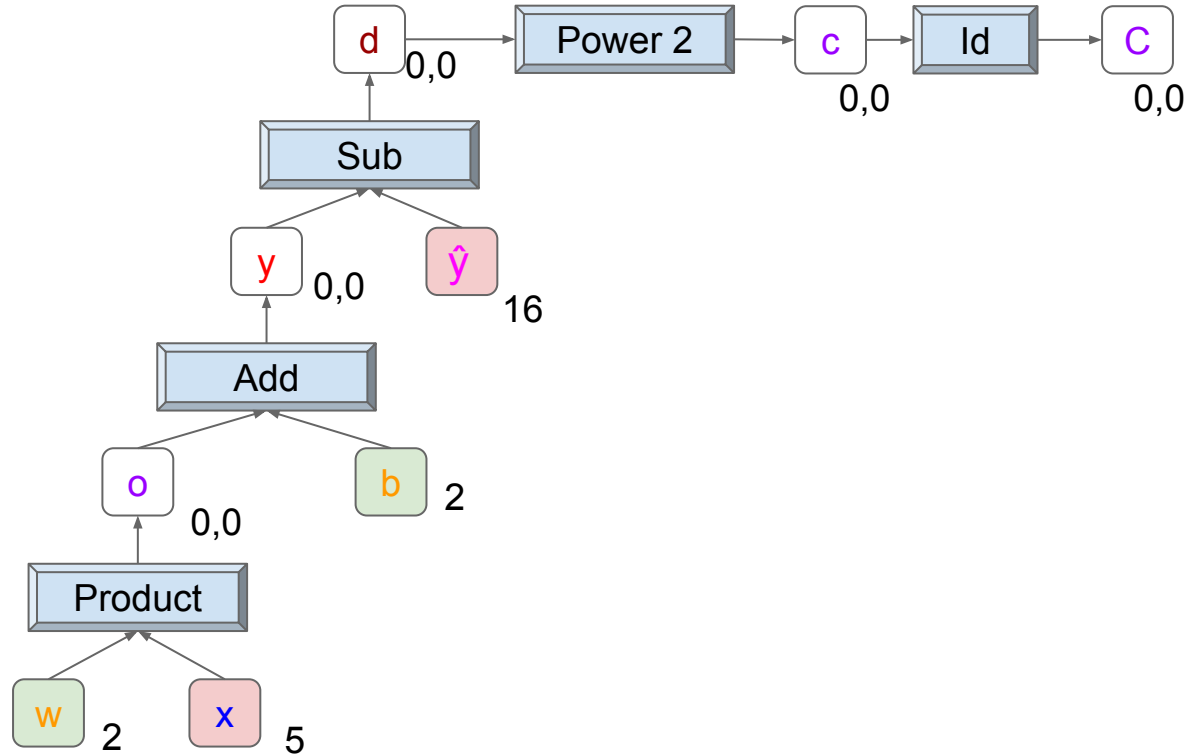
Computation Graphs are our friends

- 1-Initialize inputs
- 2-Initialize variables



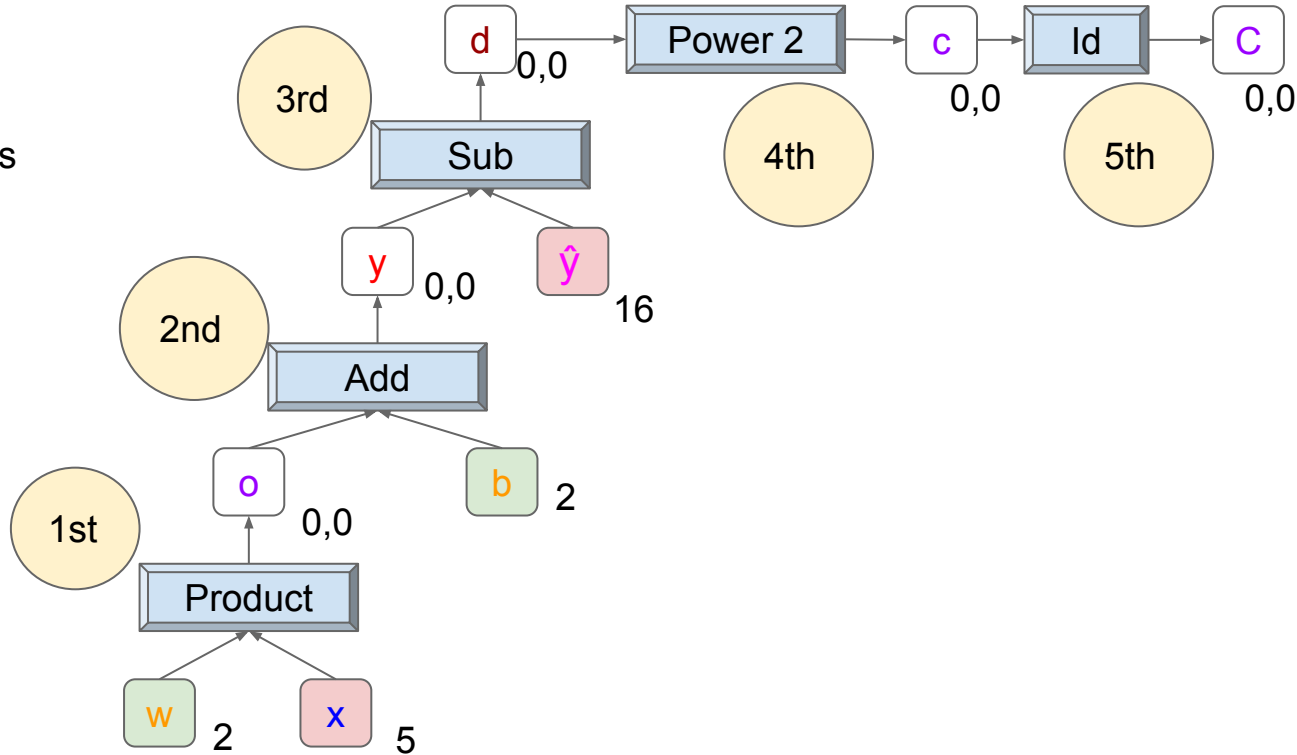
Computation Graphs are our friends

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



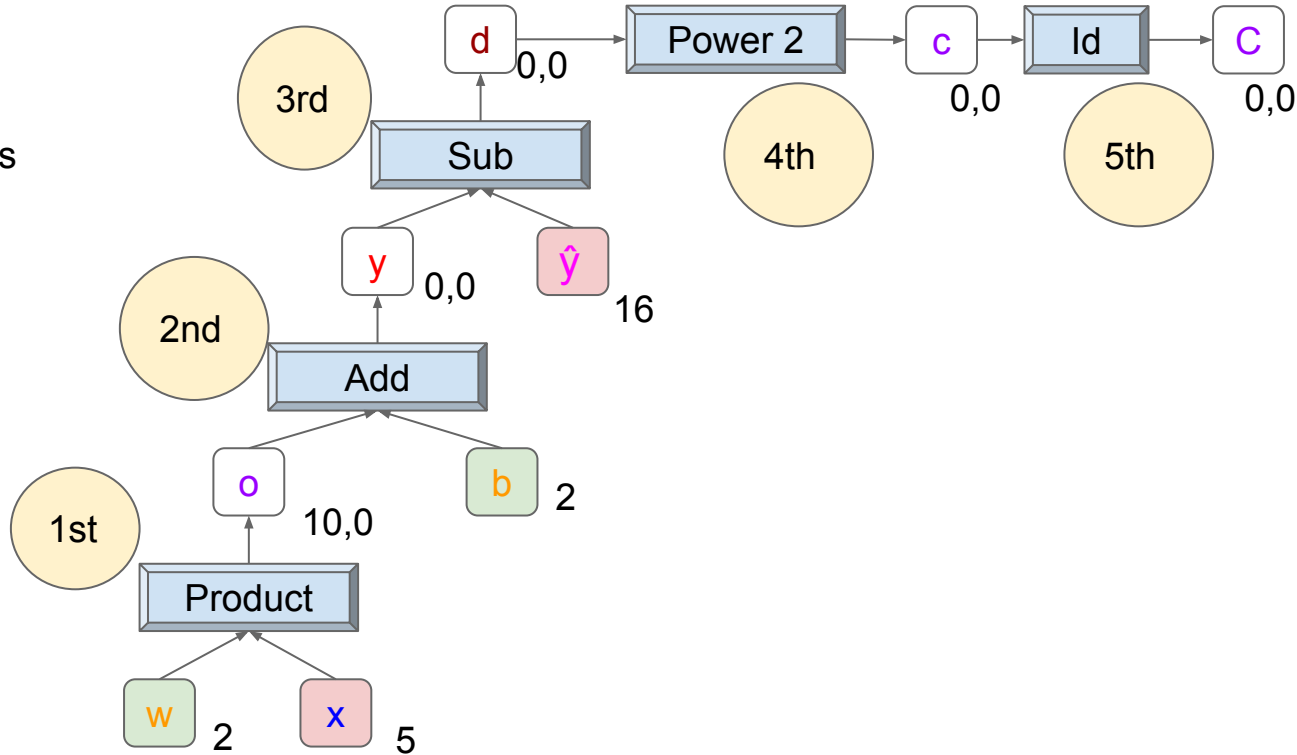
Computation Graphs are our friends

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



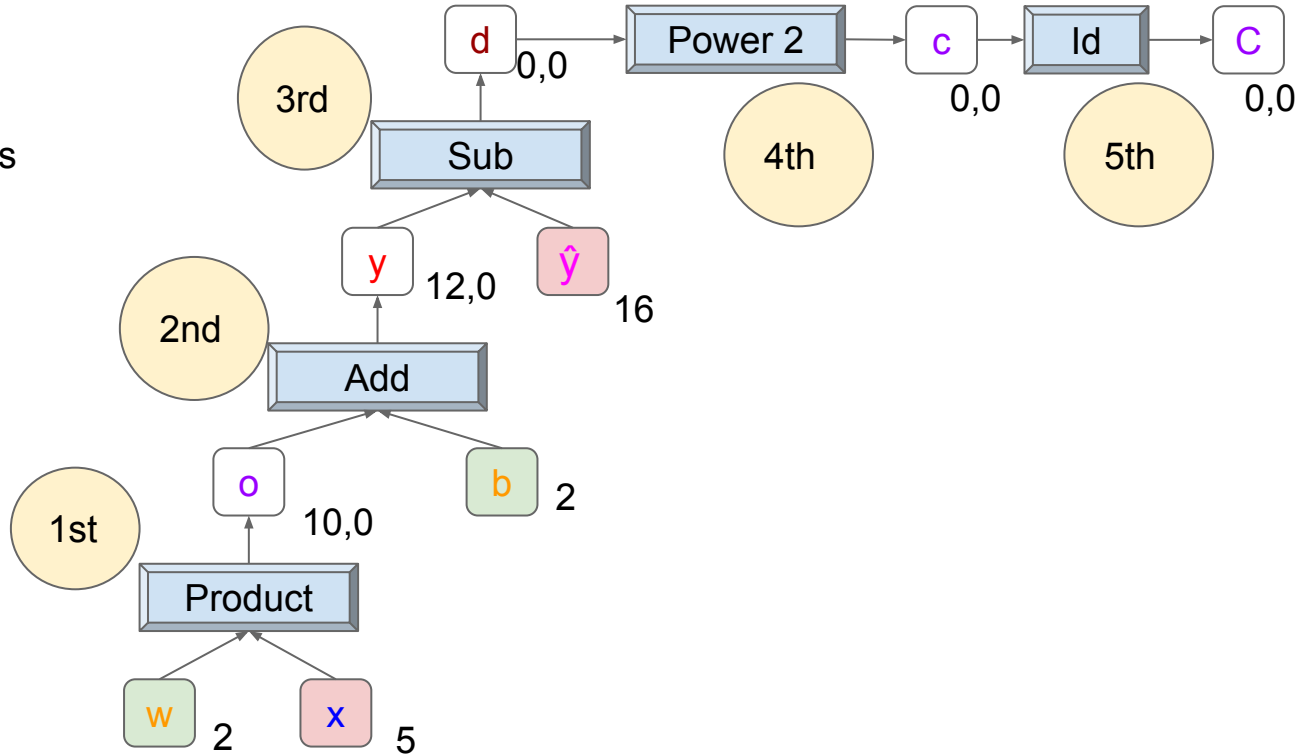
Computation Graphs are our friends

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



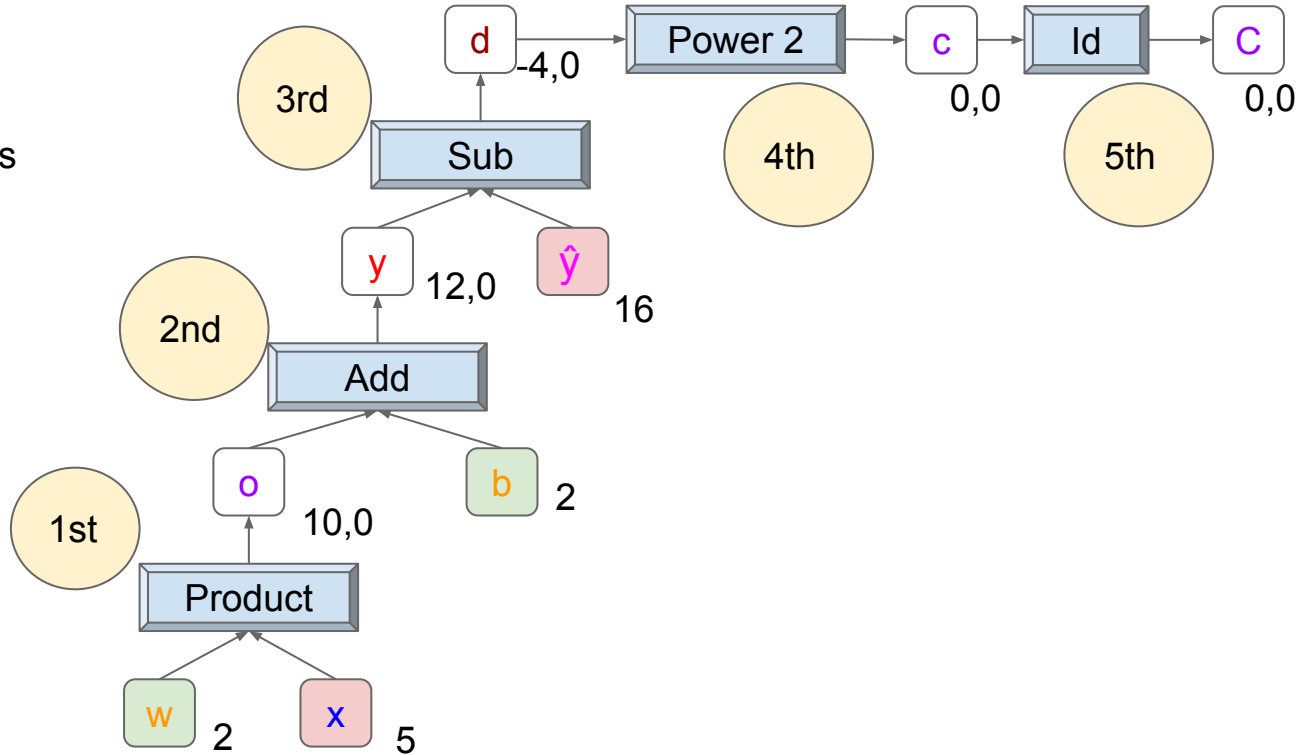
Computation Graphs are our friends

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



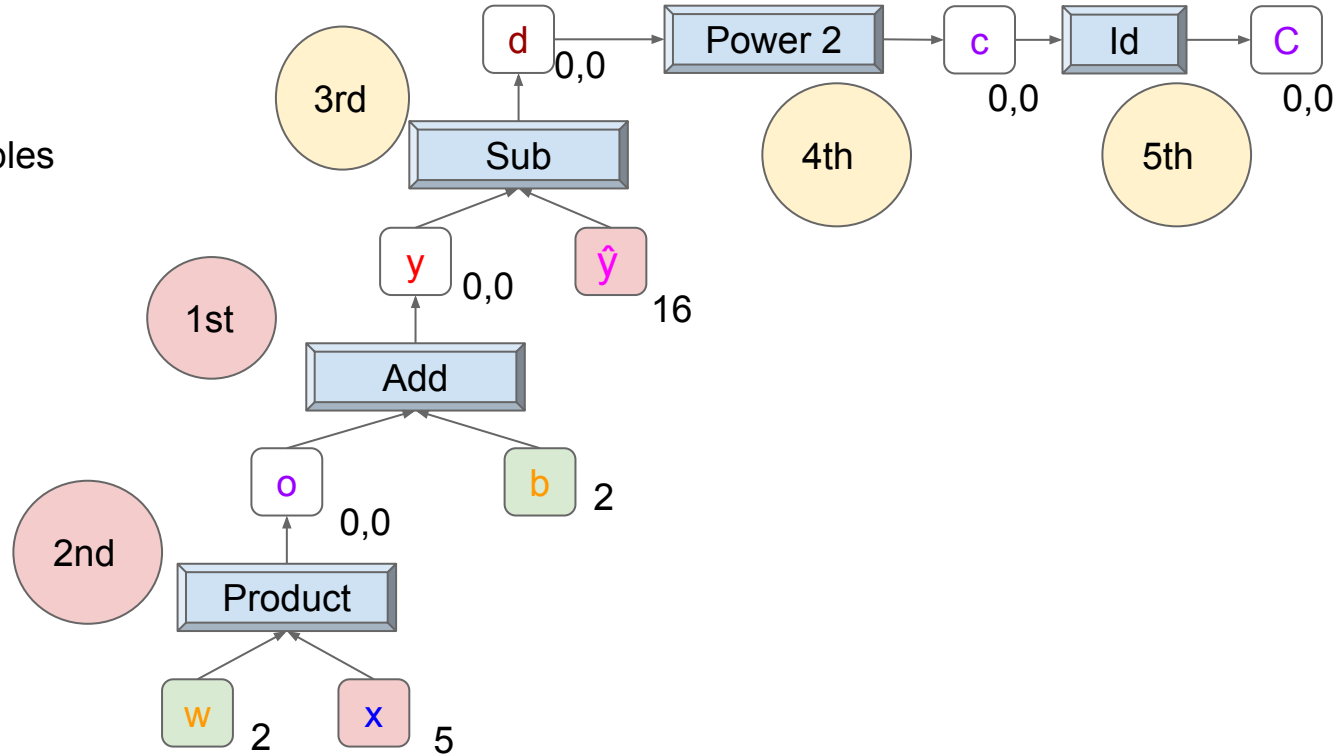
Computation Graphs are our friends

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



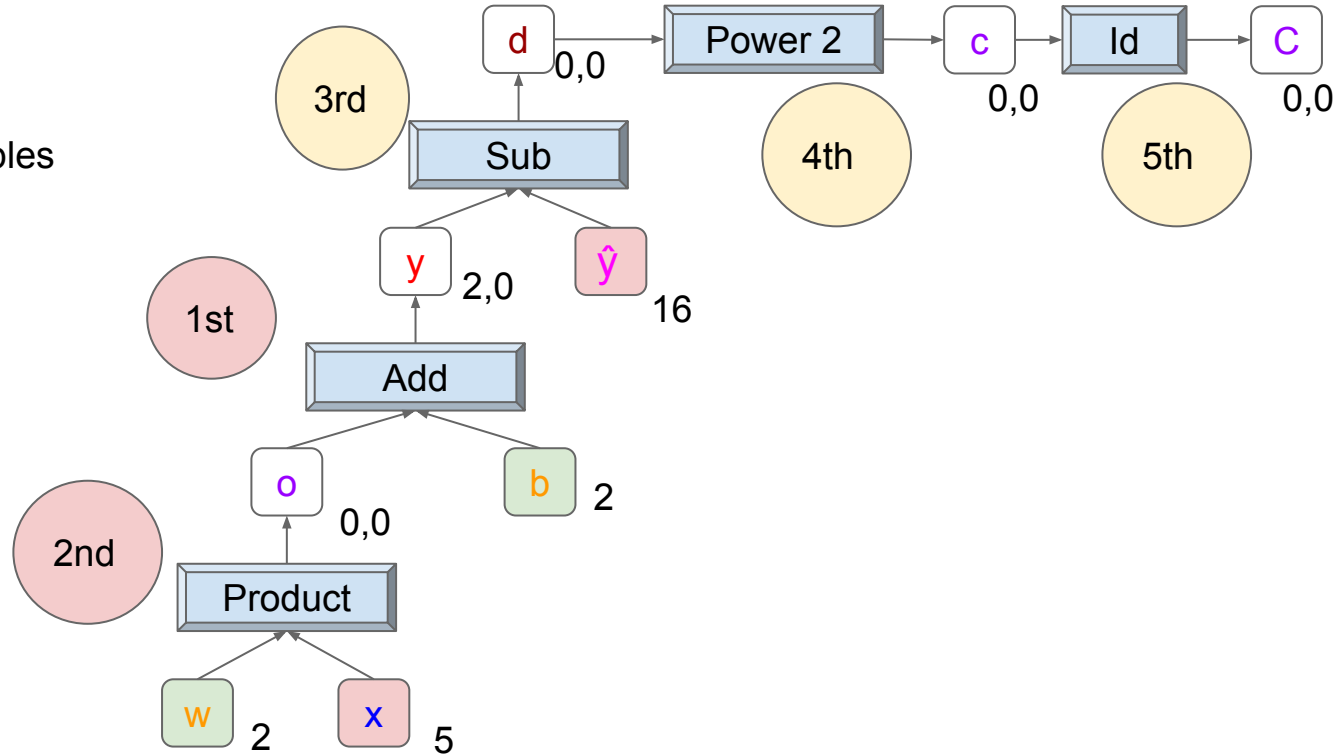
Computation Graphs are our friends

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



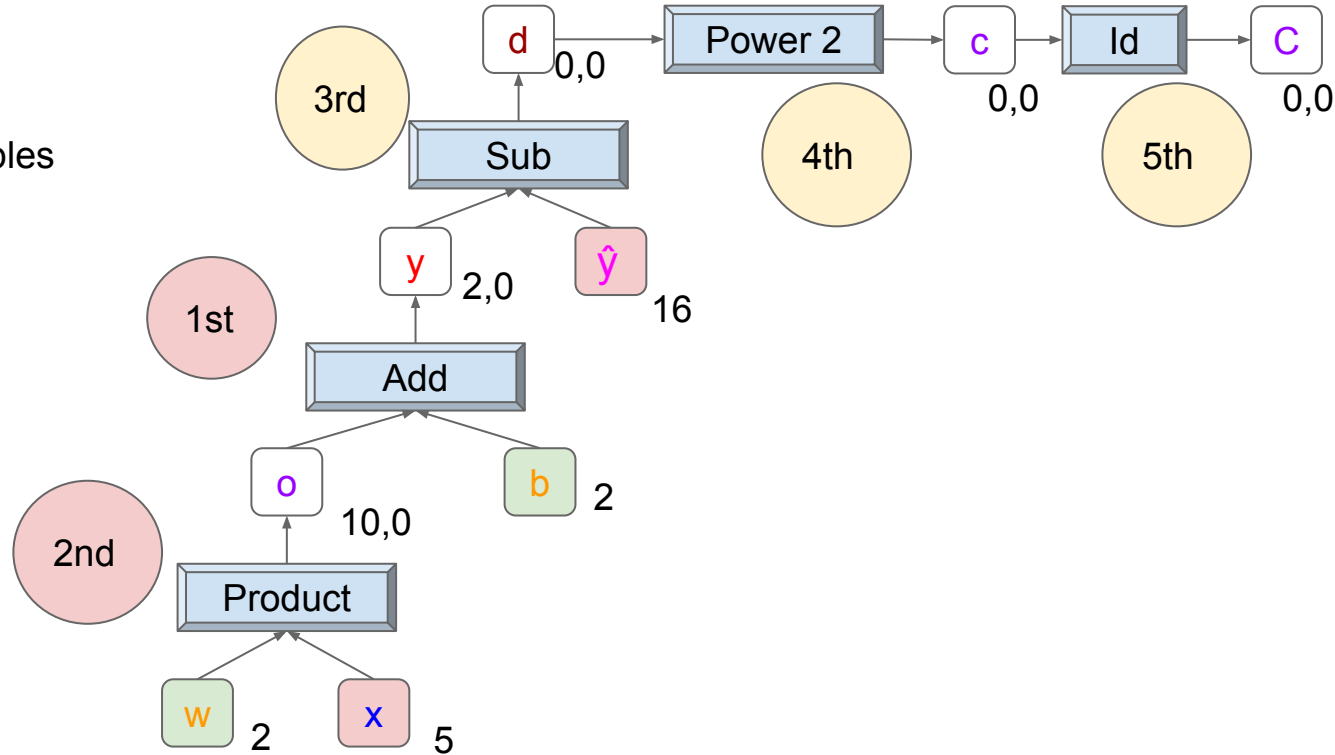
Computation Graphs are our friends

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



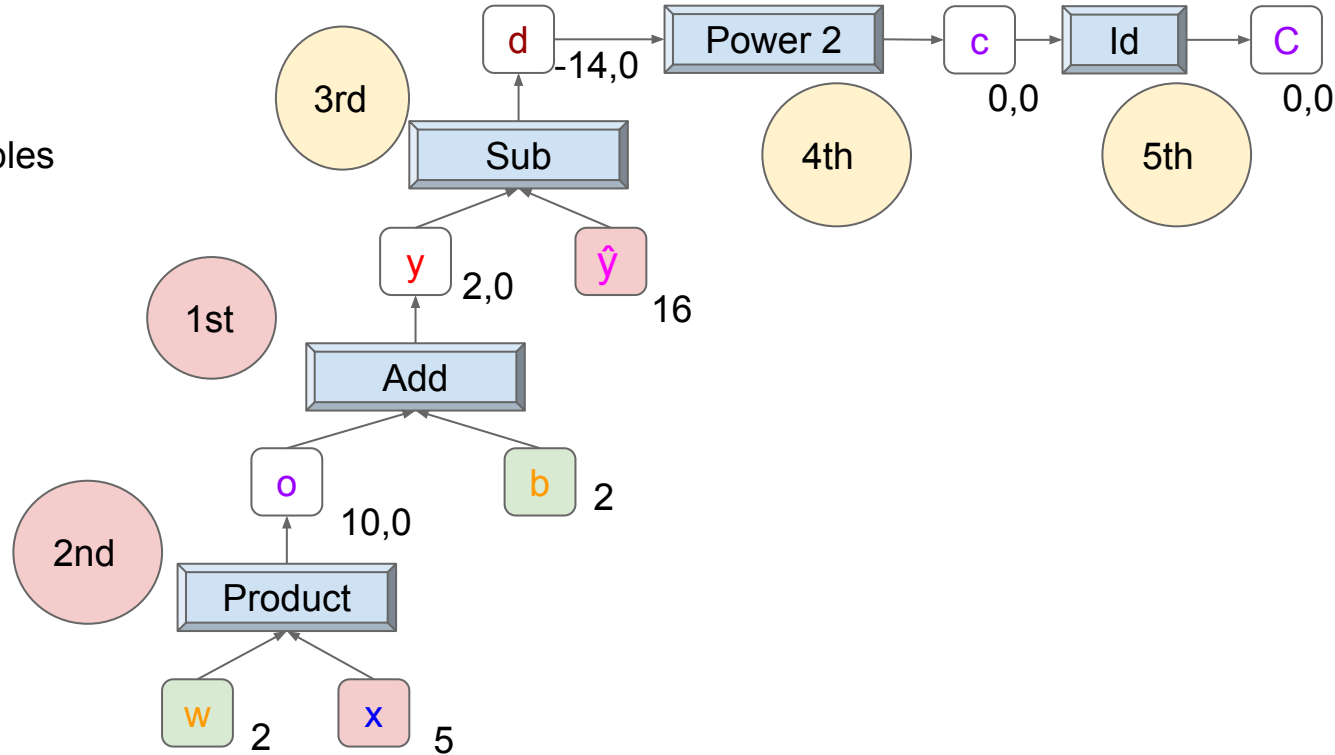
Computation Graphs are our friends

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



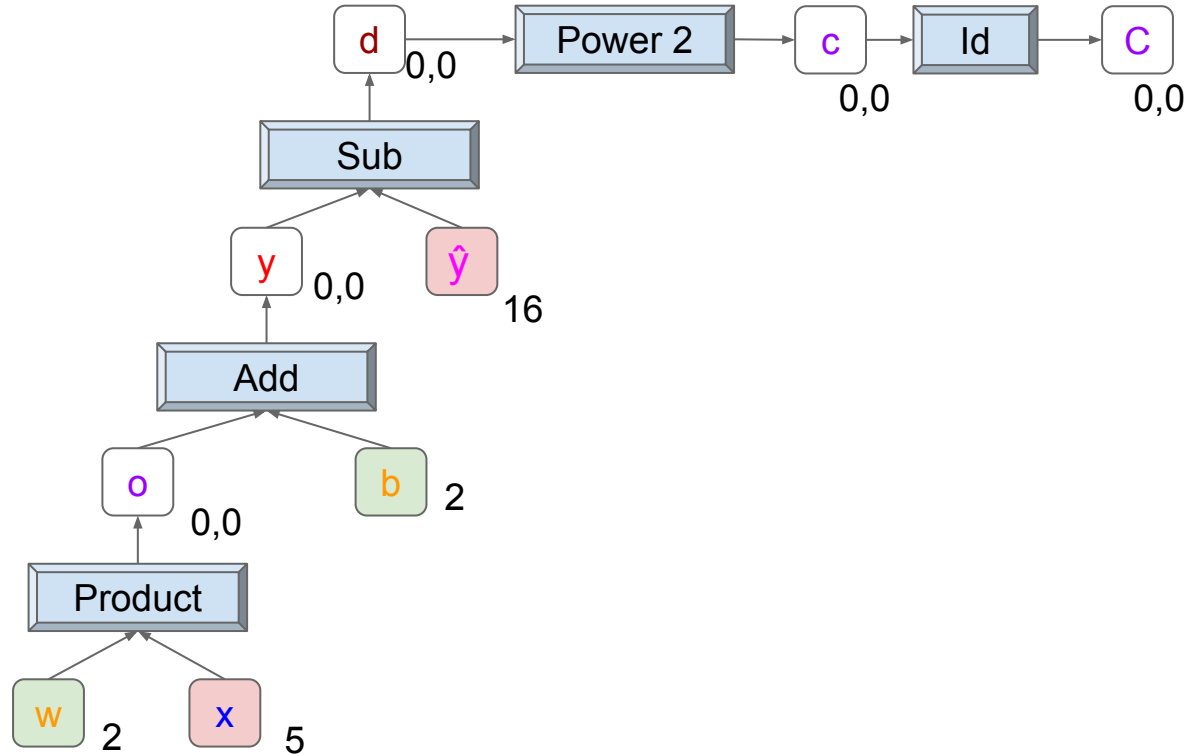
Computation Graphs are our friends

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



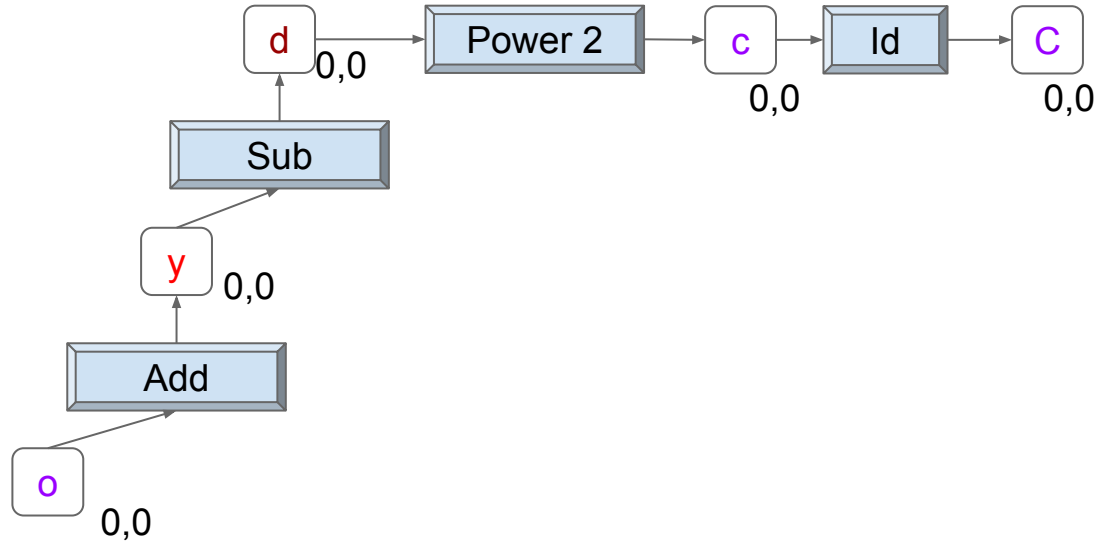
Computation Graphs are our friends

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



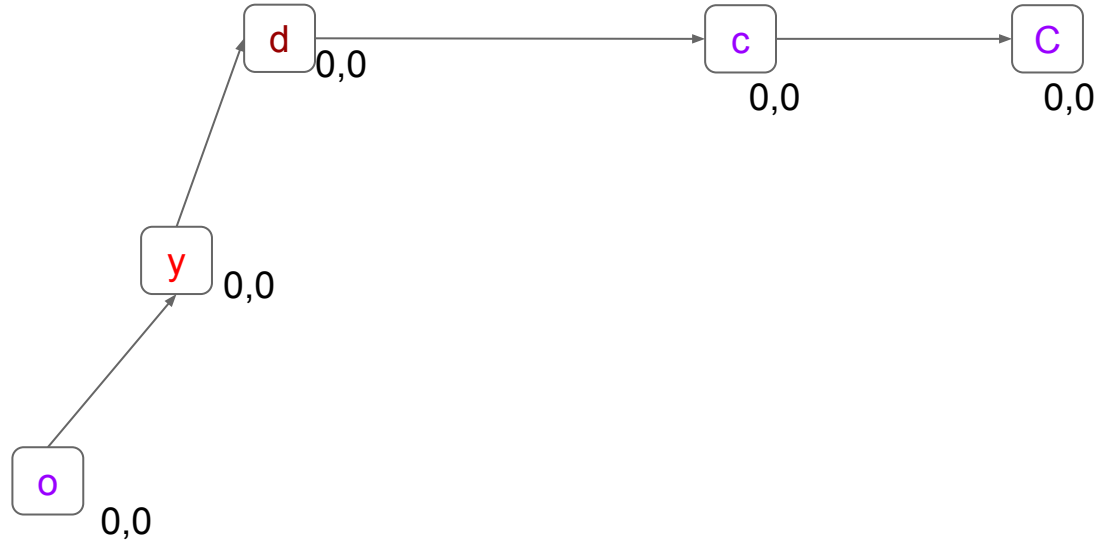
Computation Graphs are our friends

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



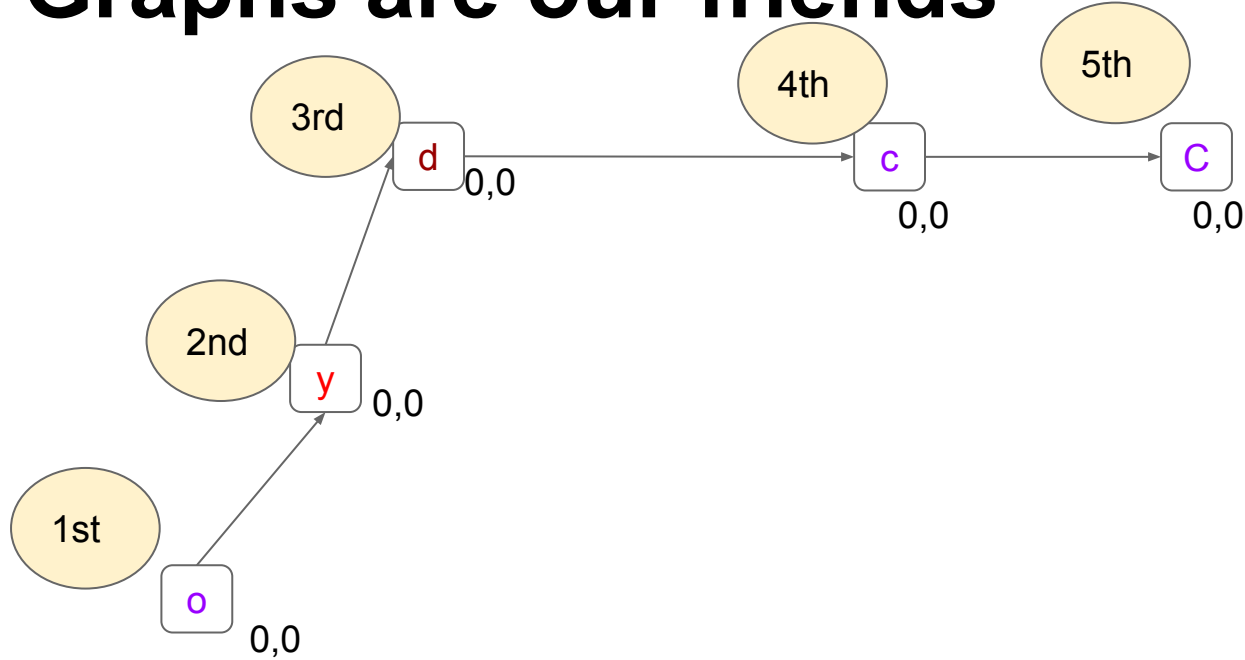
Computation Graphs are our friends

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



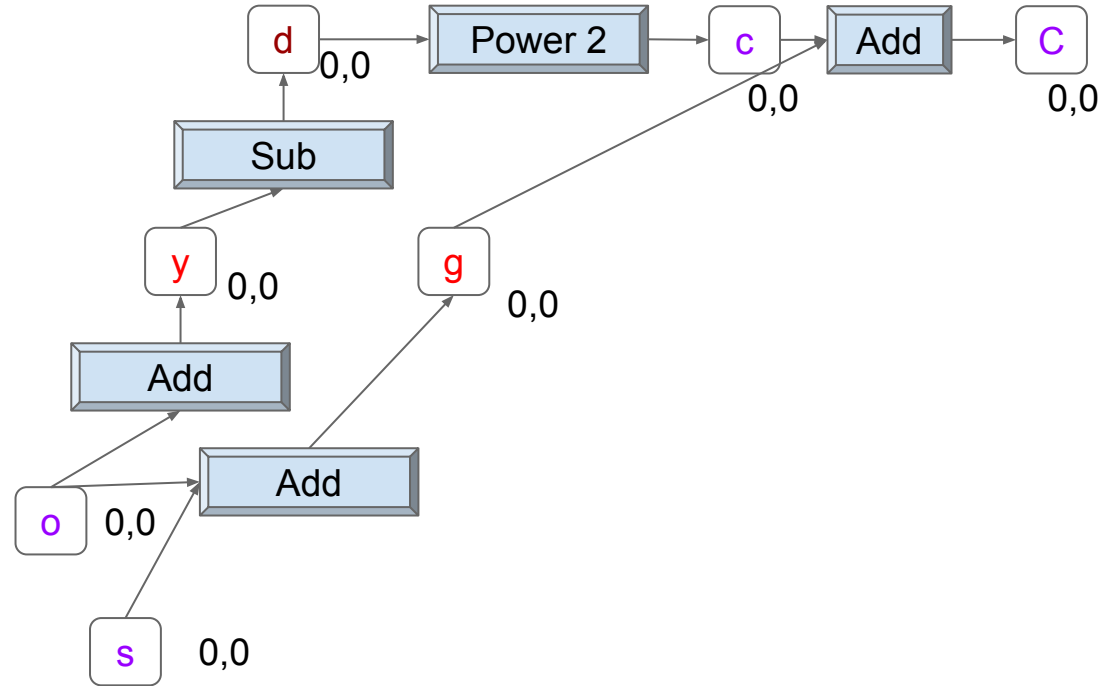
Computation Graphs are our friends

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



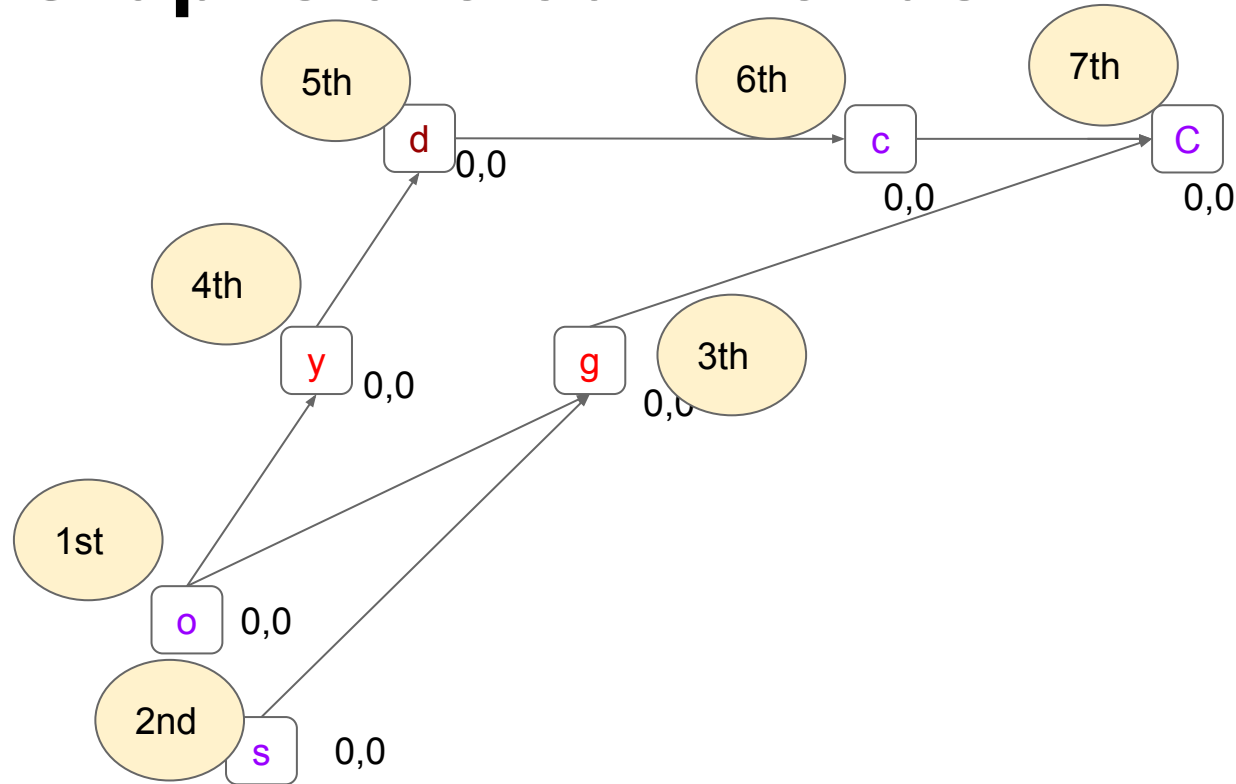
Computation Graphs are our friends

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



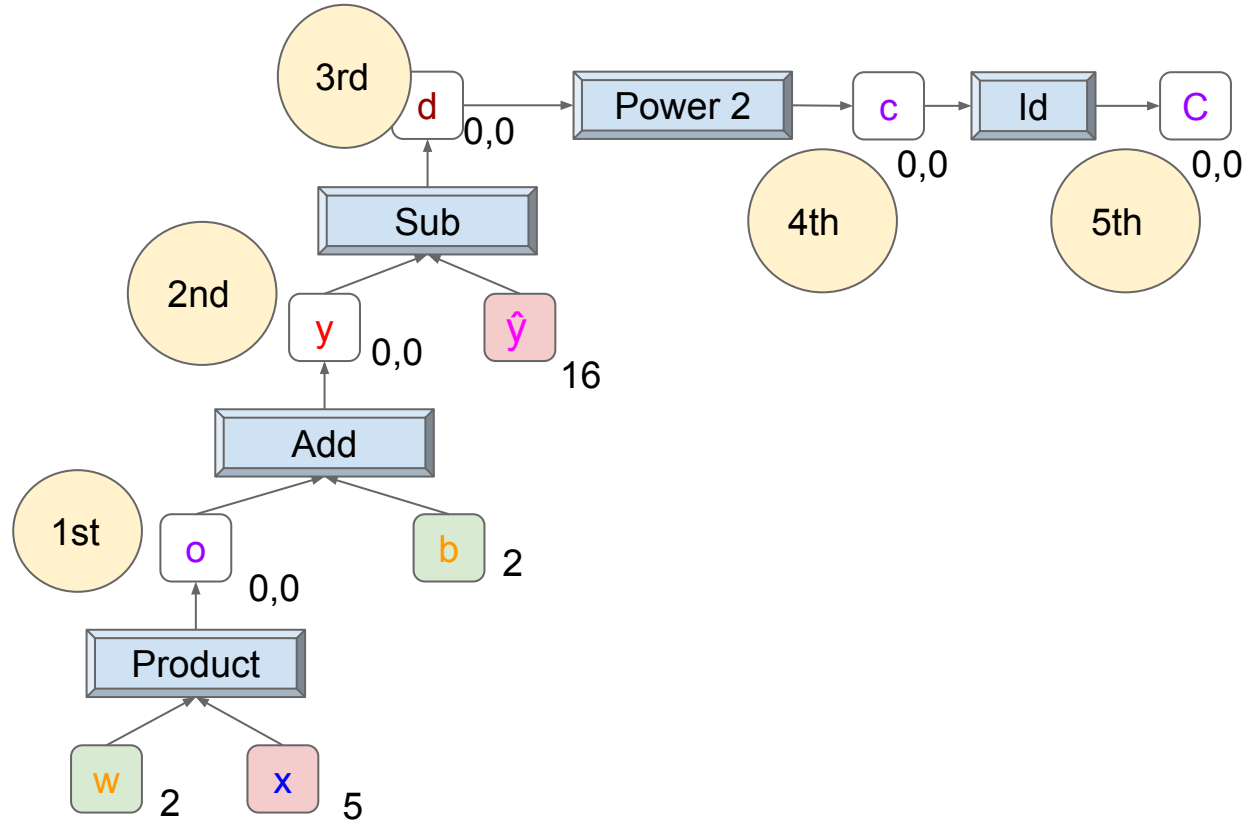
Computation Graphs are our friends

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



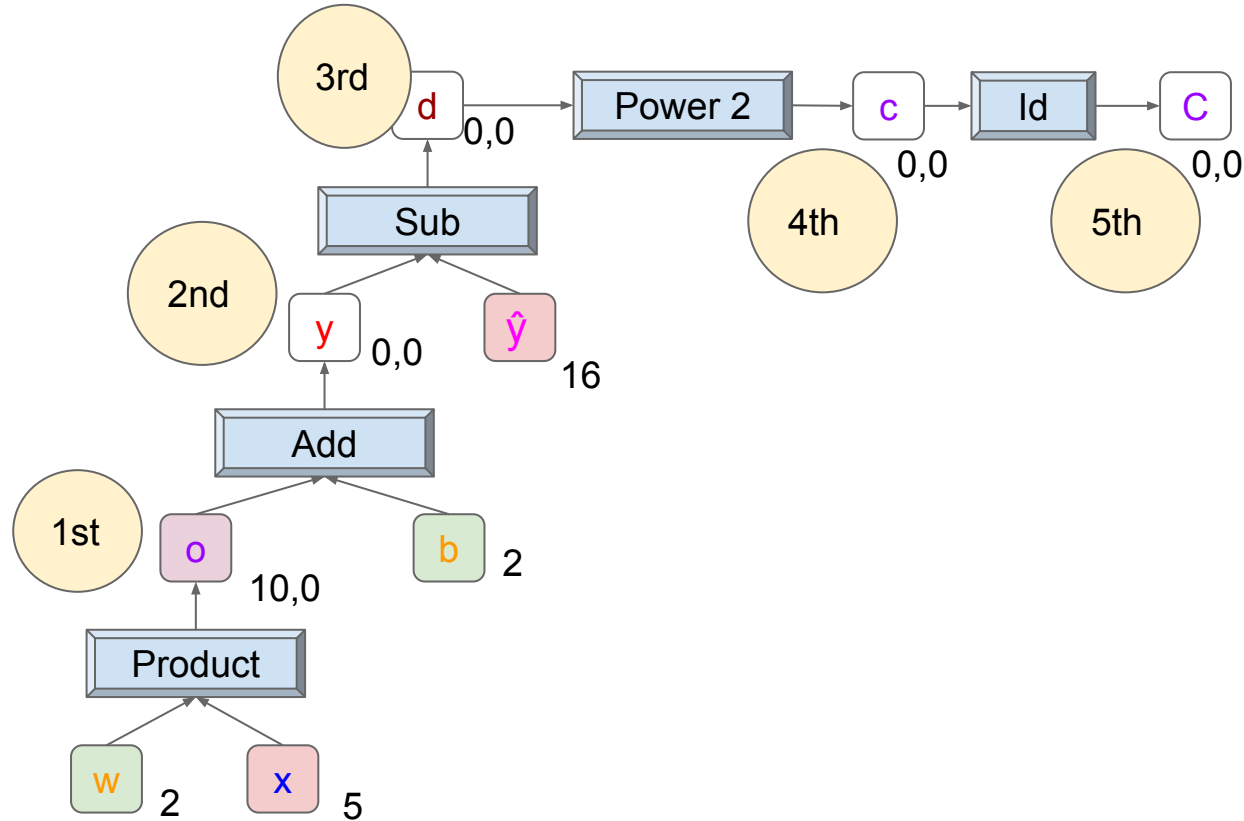
Computation Graphs are our friends

- 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



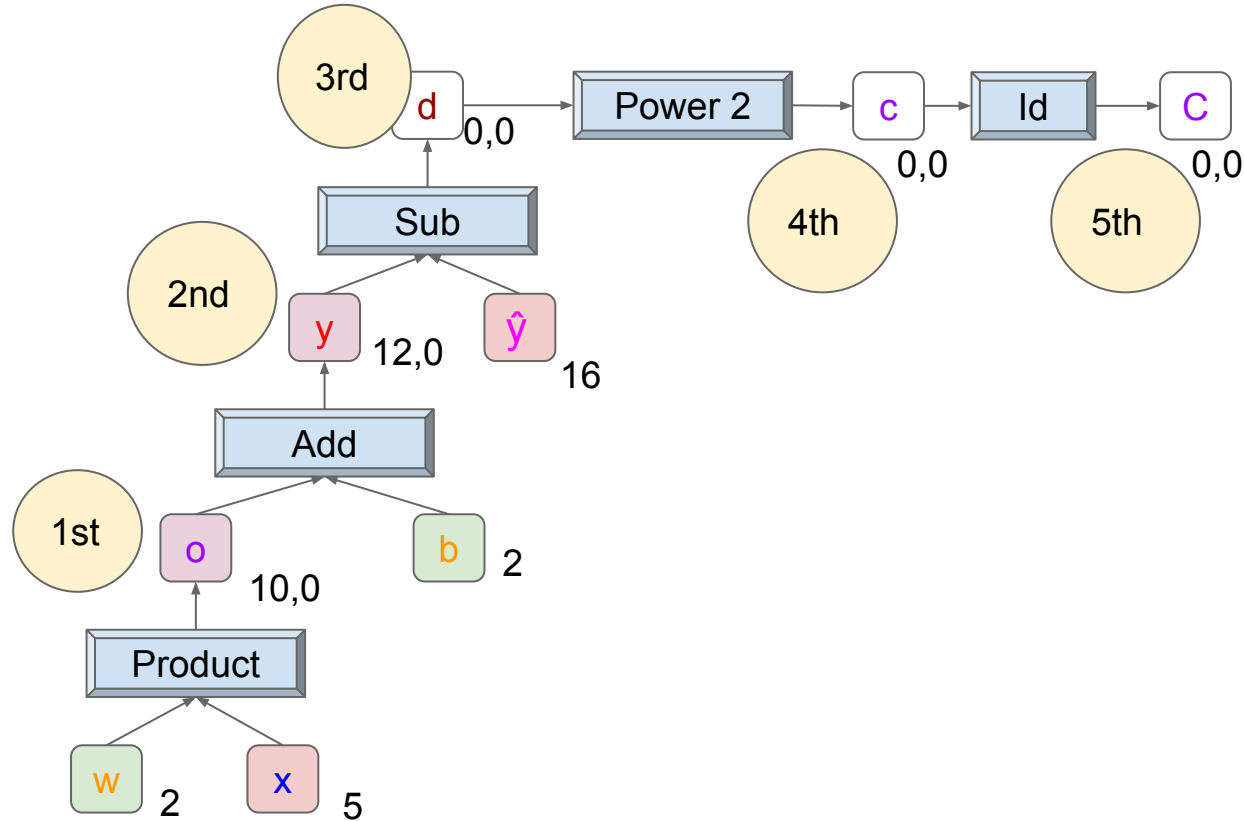
Computation Graphs are our friends

- 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



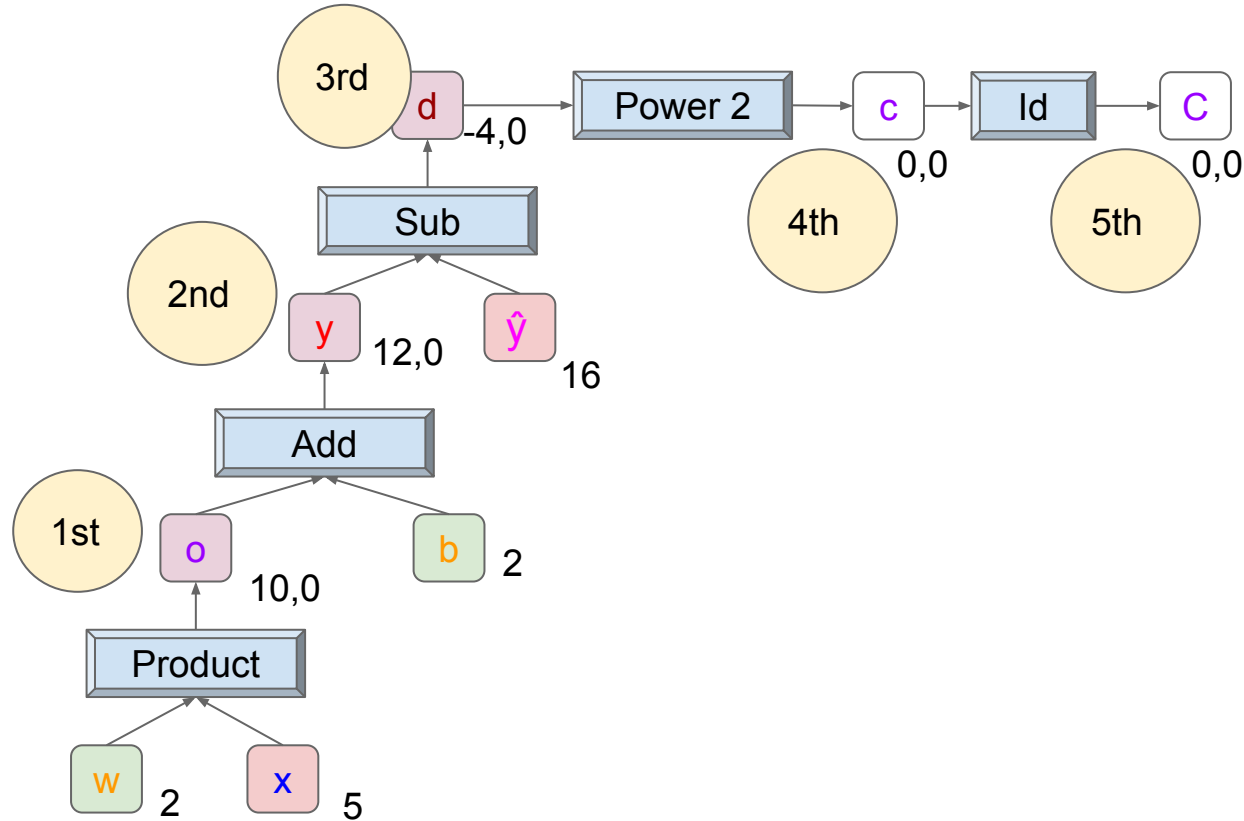
Computation Graphs are our friends

- 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



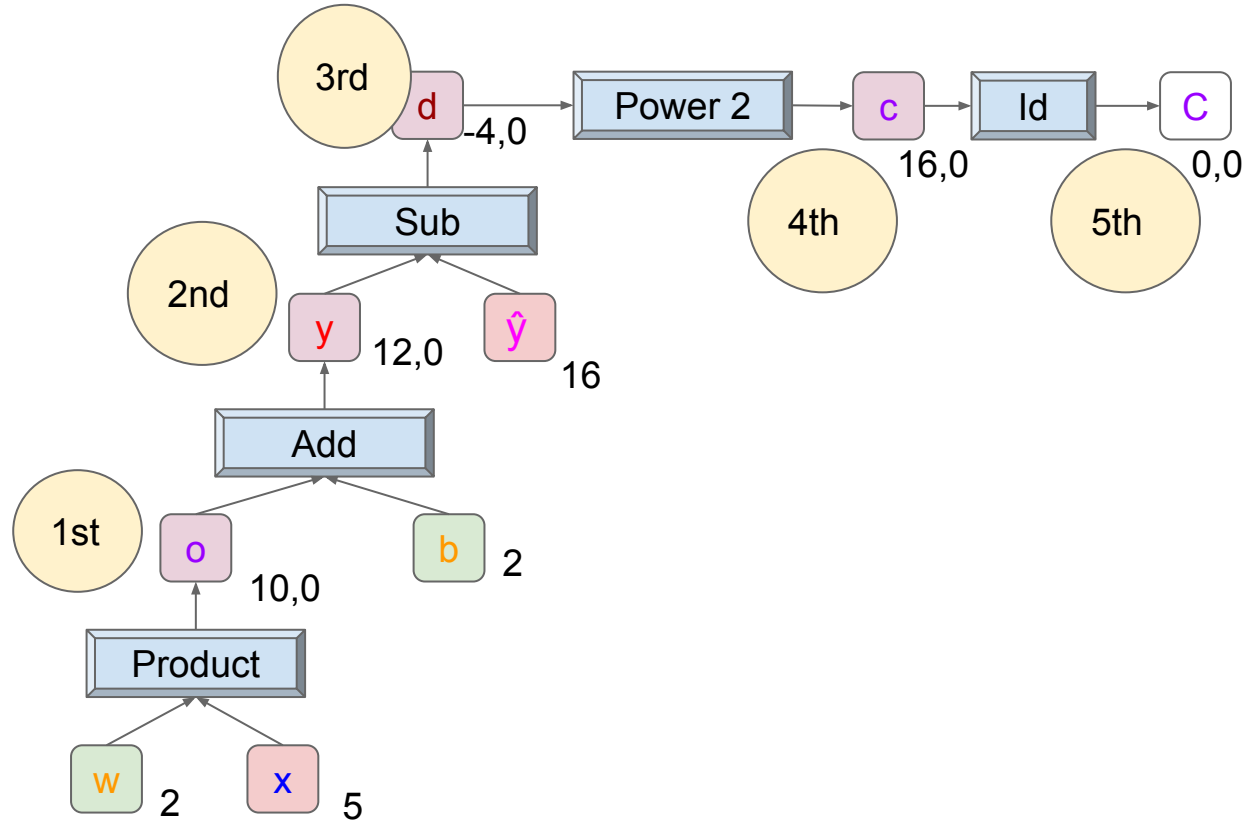
Computation Graphs are our friends

- 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



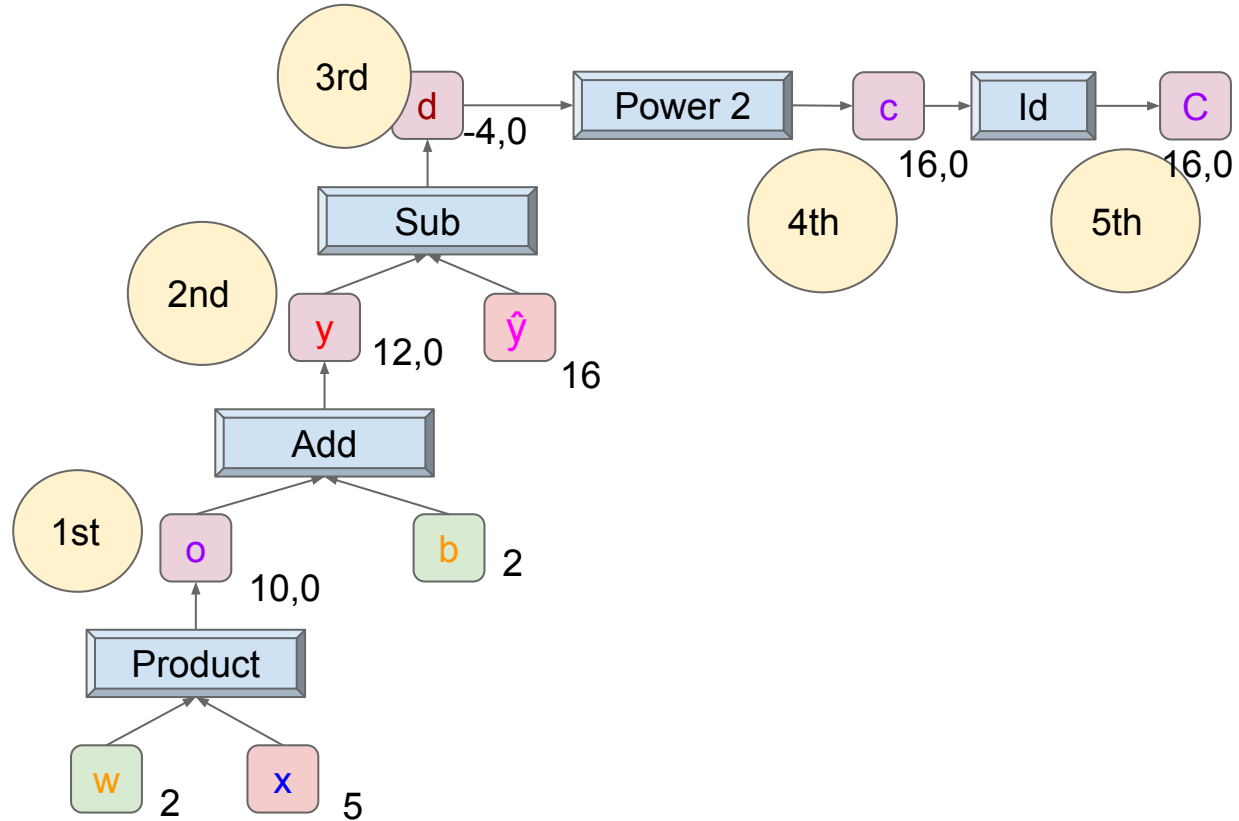
Computation Graphs are our friends

- 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



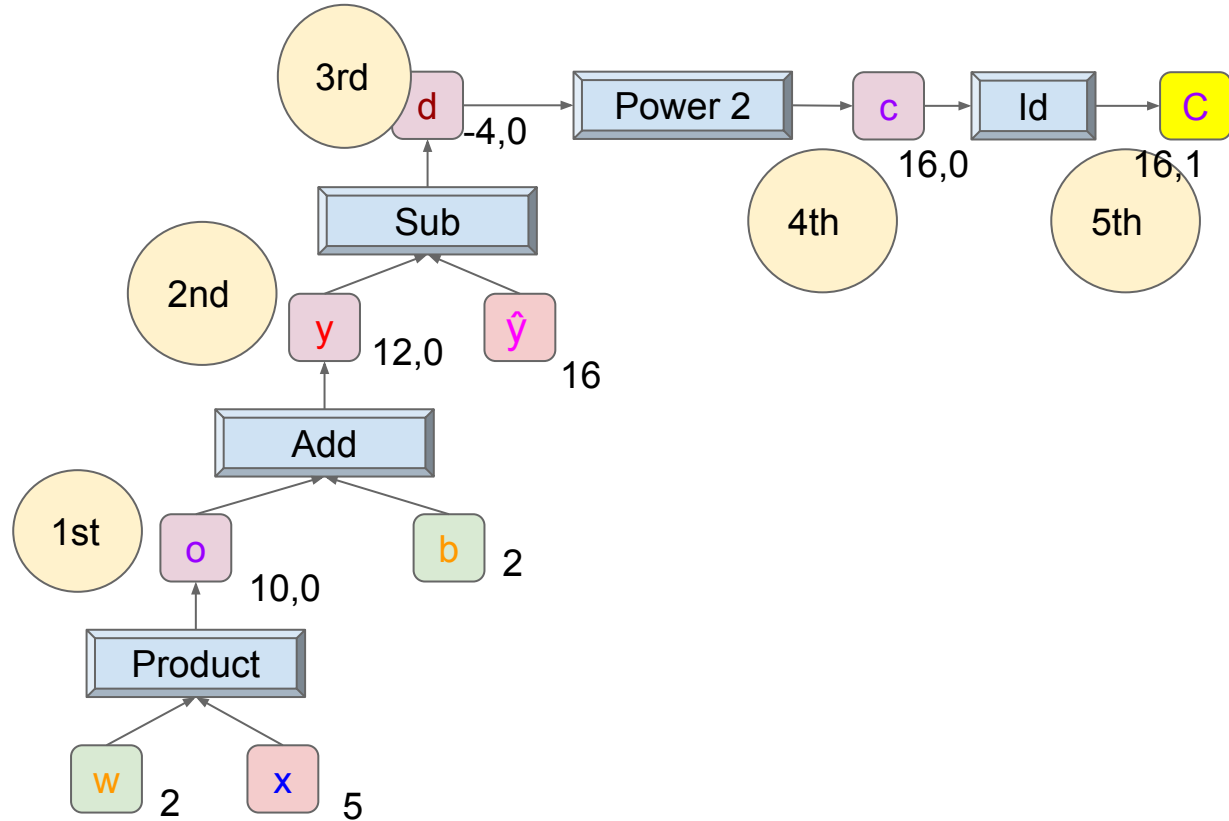
Computation Graphs are our friends

- 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



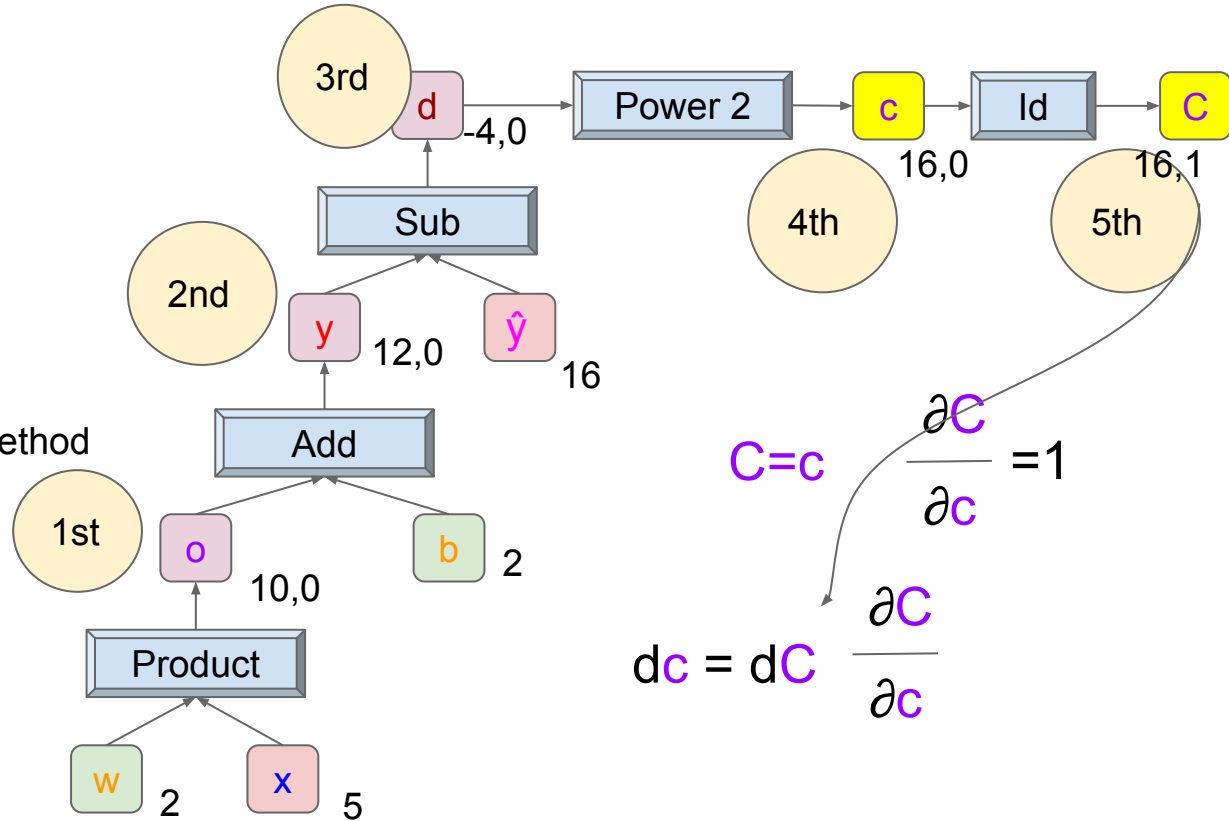
Computation Graphs are our friends

- 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



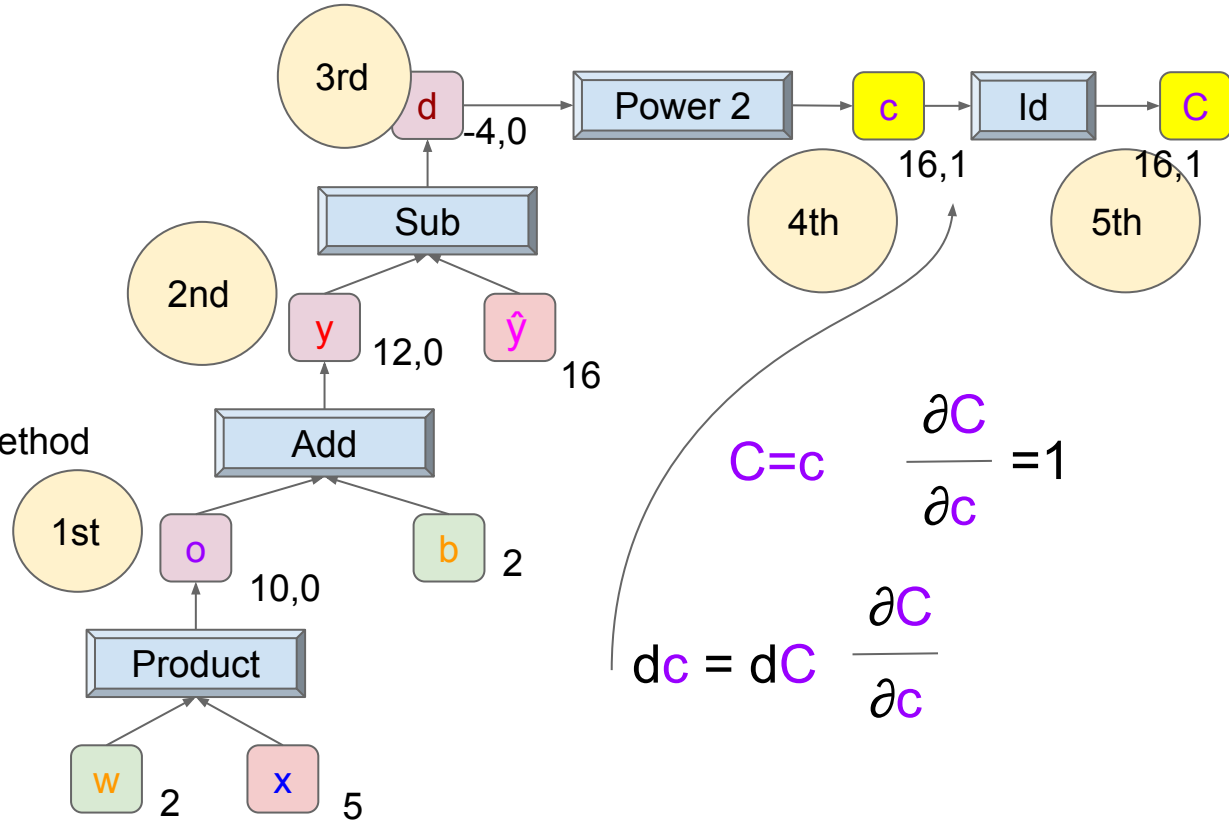
Computation Graphs are our friends

- 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)



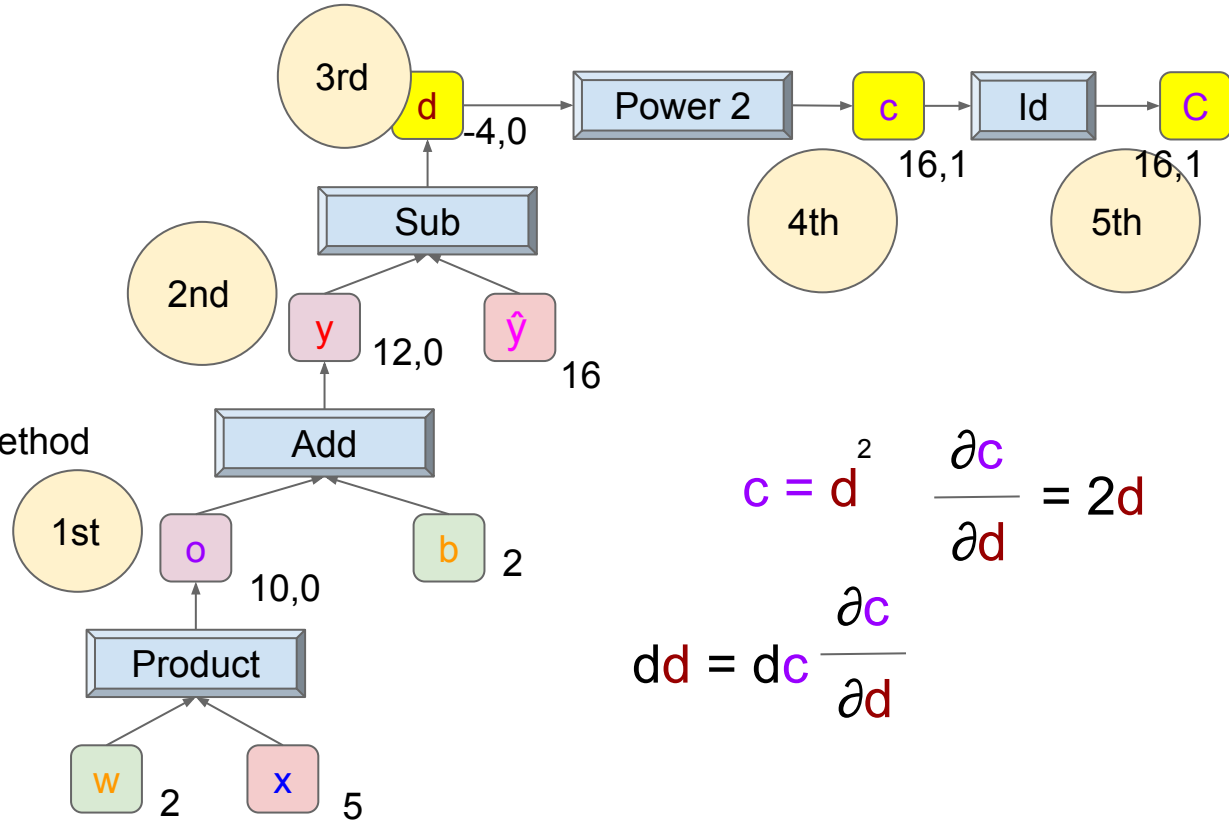
Computation Graphs are our friends

- 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)

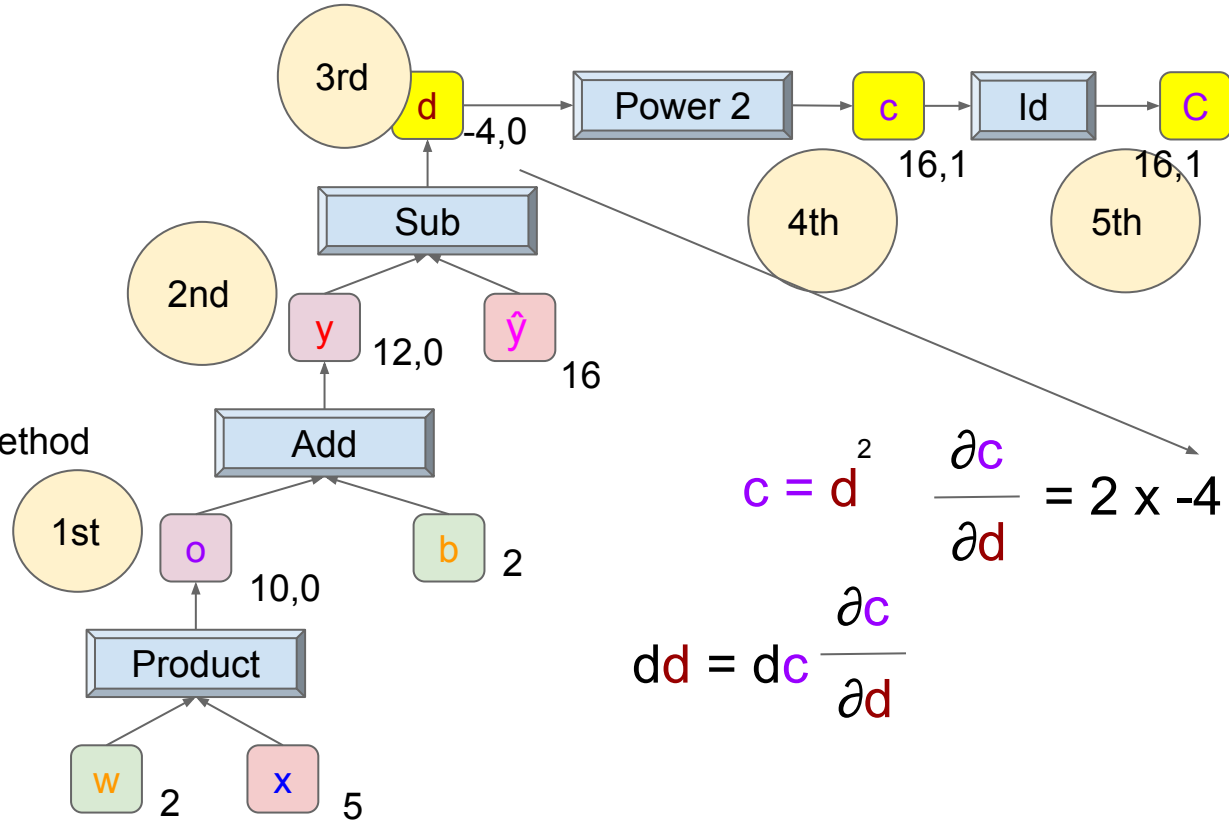


$$c = d^2 \quad \frac{\partial c}{\partial d} = 2d$$

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

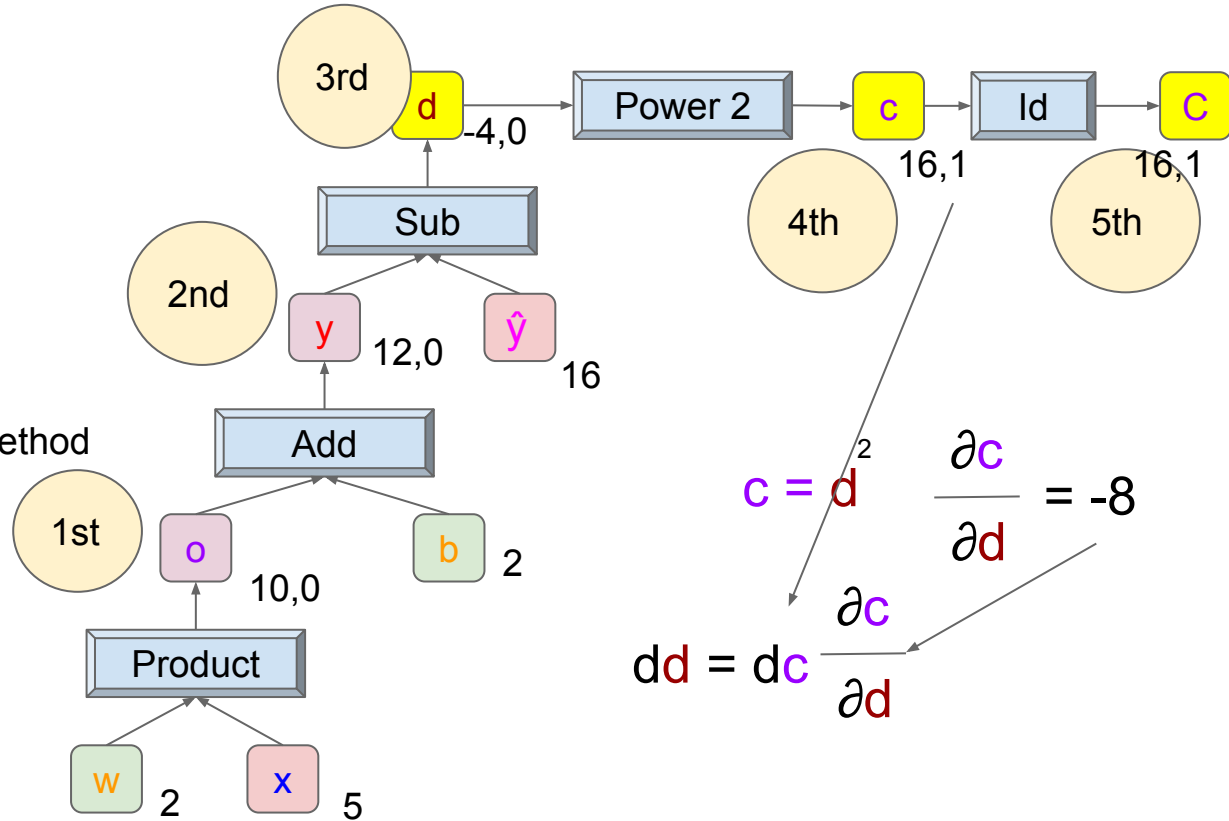
Computation Graphs are our friends

- 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)



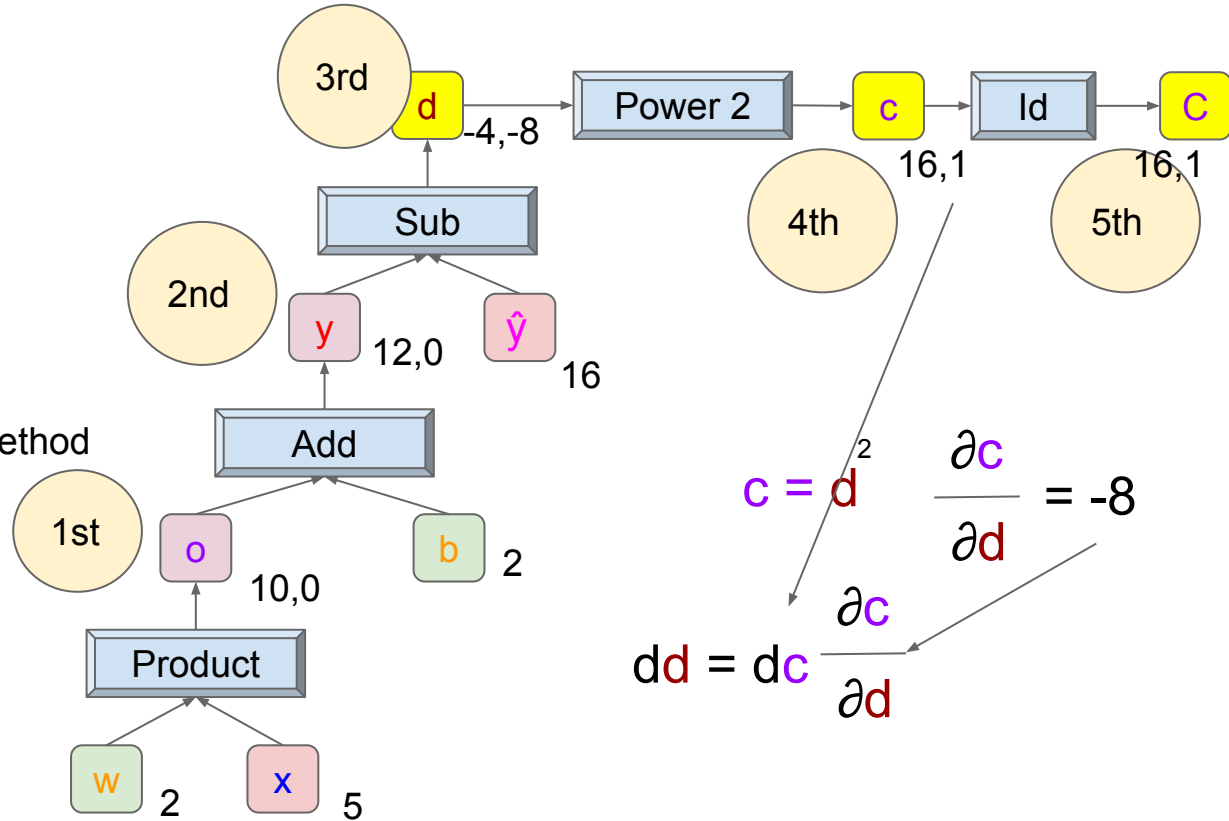
Computation Graphs are our friends

- 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)



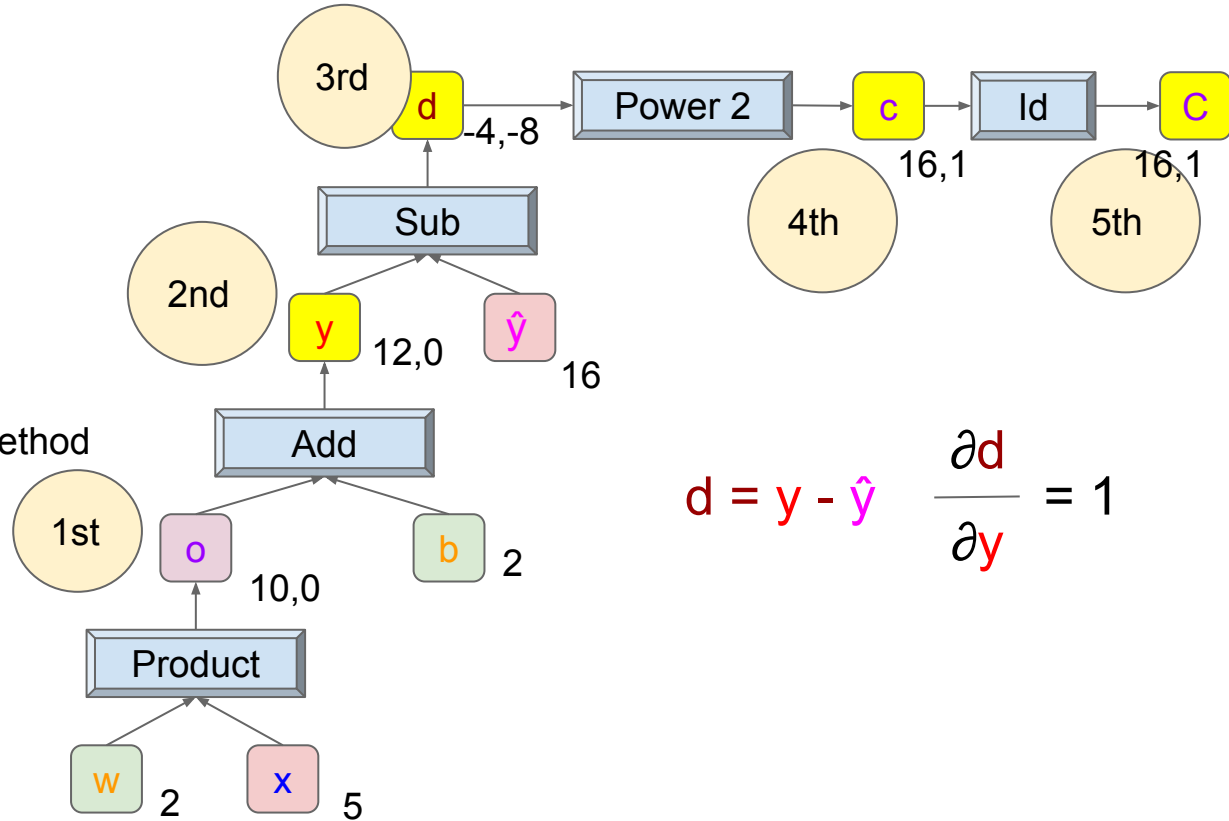
Computation Graphs are our friends

- 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

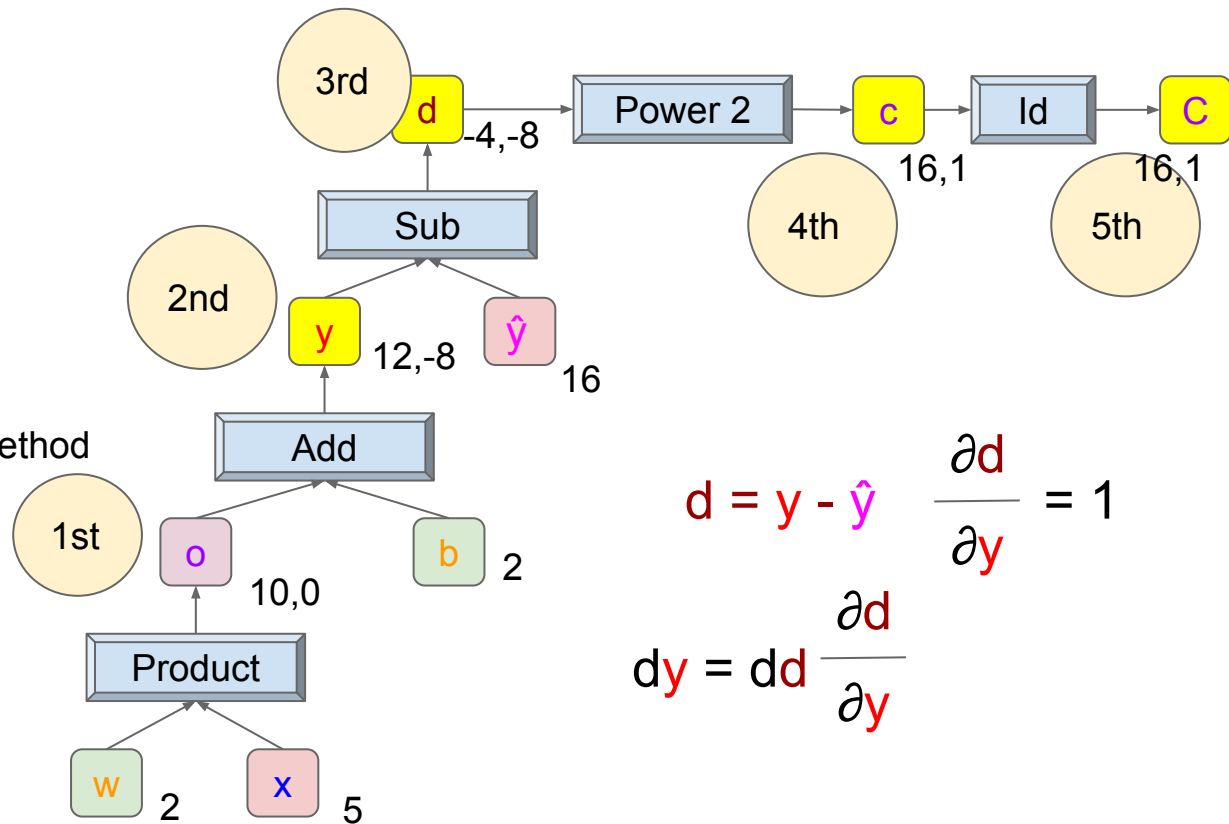
- 1-Initialize inputs
- 2-Initialize variables
- 3-Topological Sort variables
- 4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
- 5-Set gradients to final variables
- 6-run the operations backward method in reverse order (Backward)



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

Computation Graphs are our friends

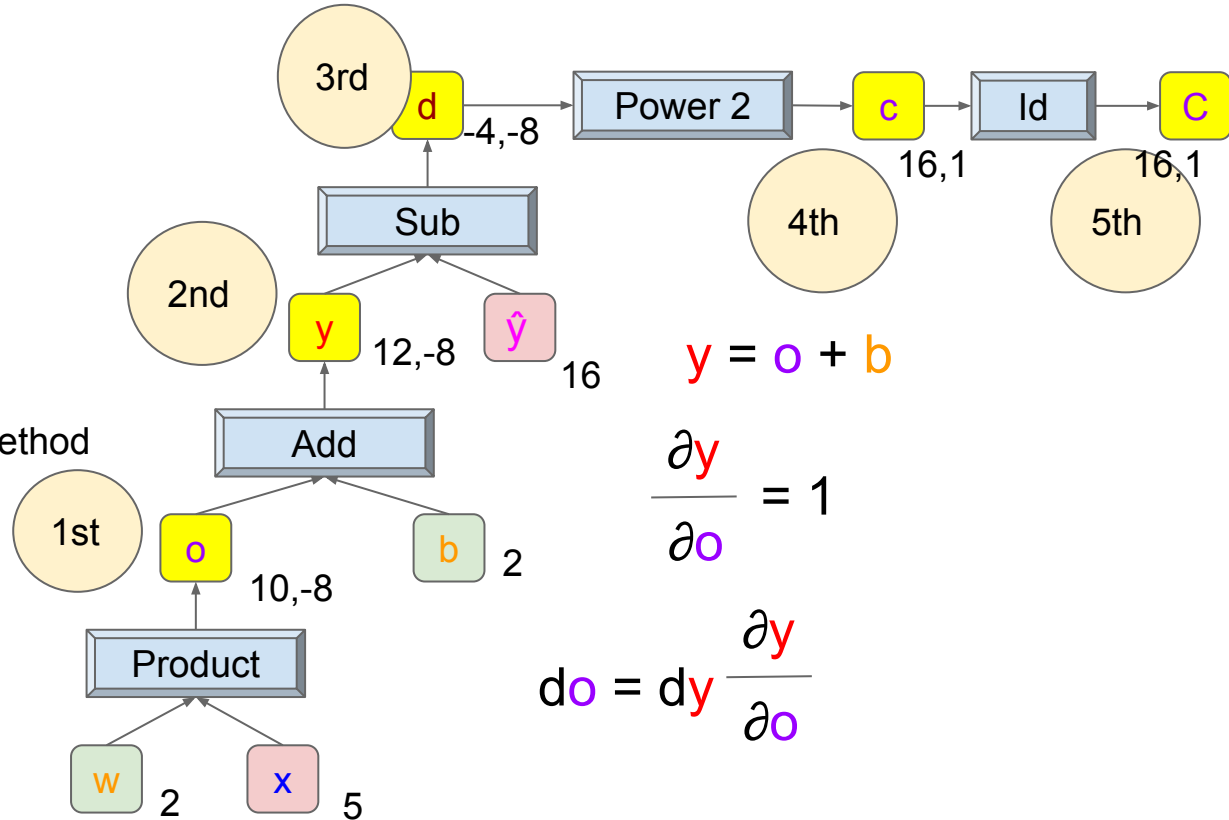
- 1-Initialize inputs
- 2-Initialize variables
- 3-Topological Sort variables
- 4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
- 5-Set gradients to final variables
- 6-run the operations backward method in reverse order (Backward)



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

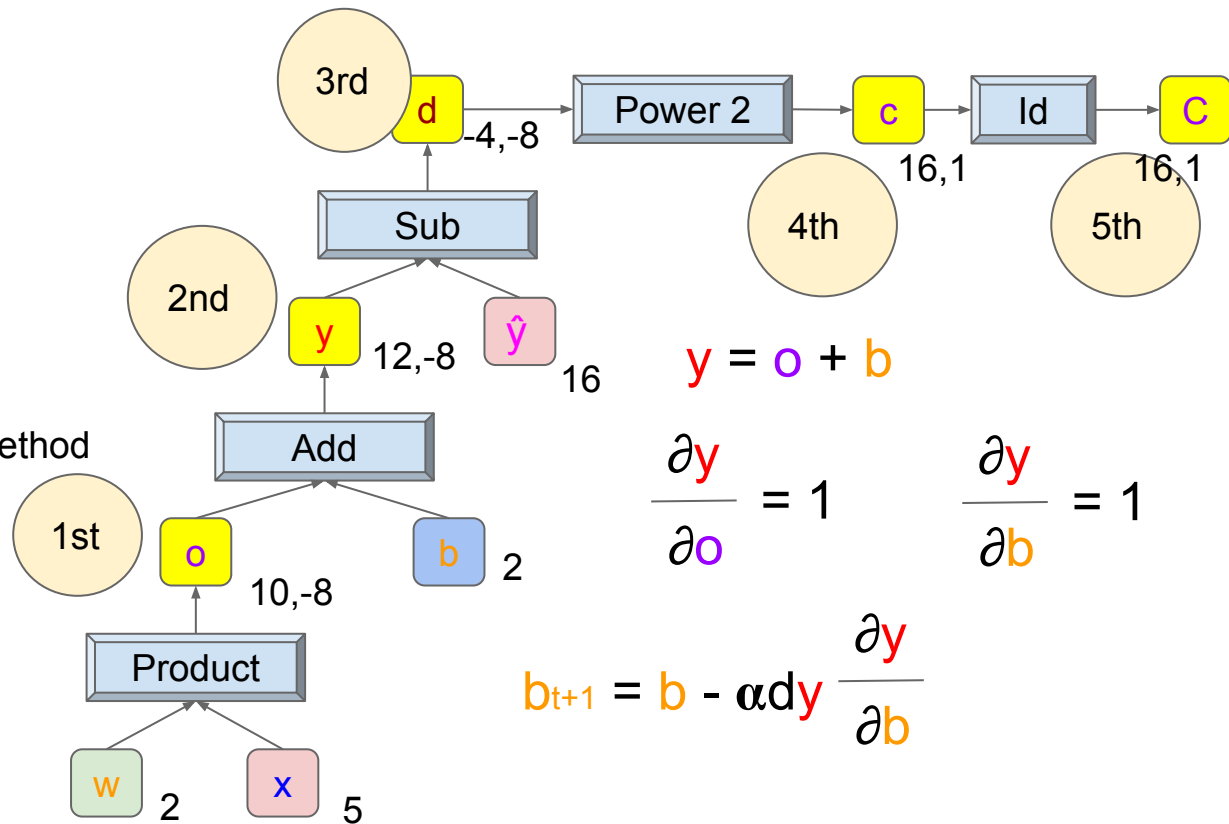
Computation Graphs are our friends

- 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)



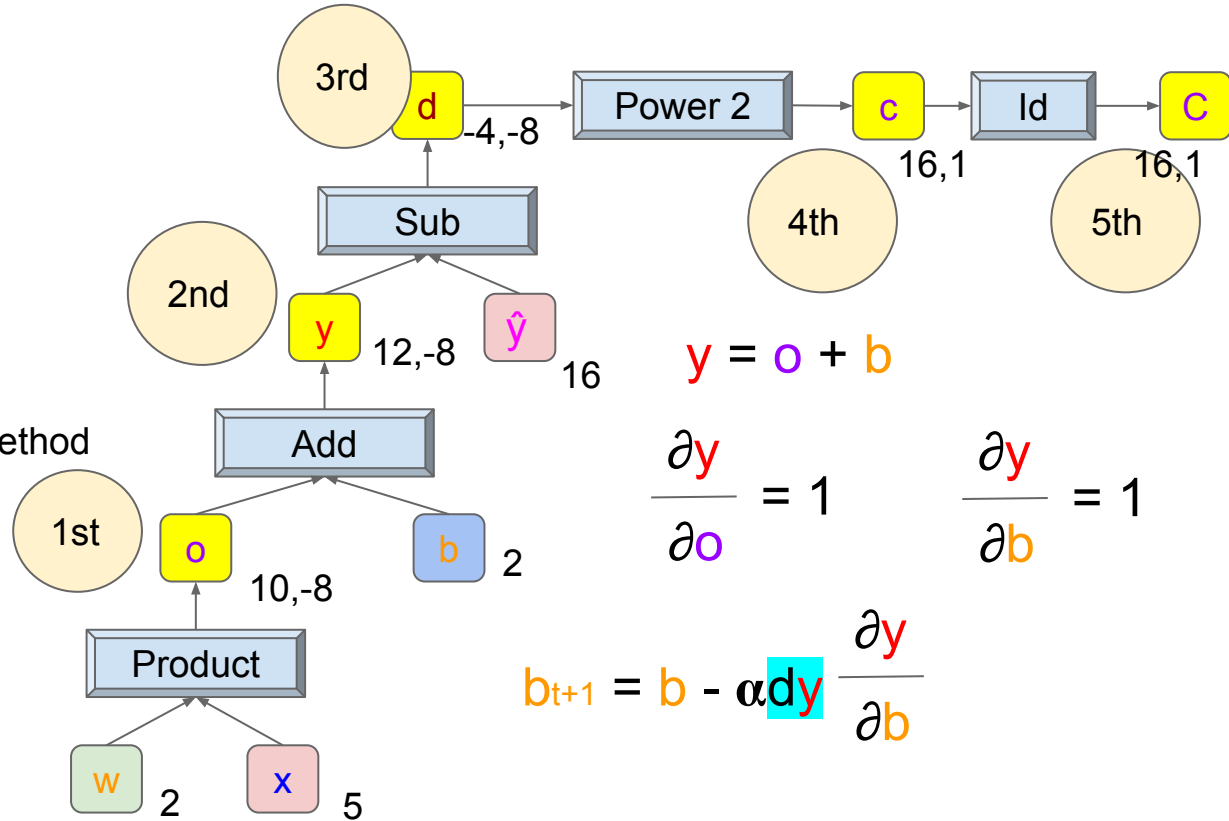
Computation Graphs are our friends

- 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)



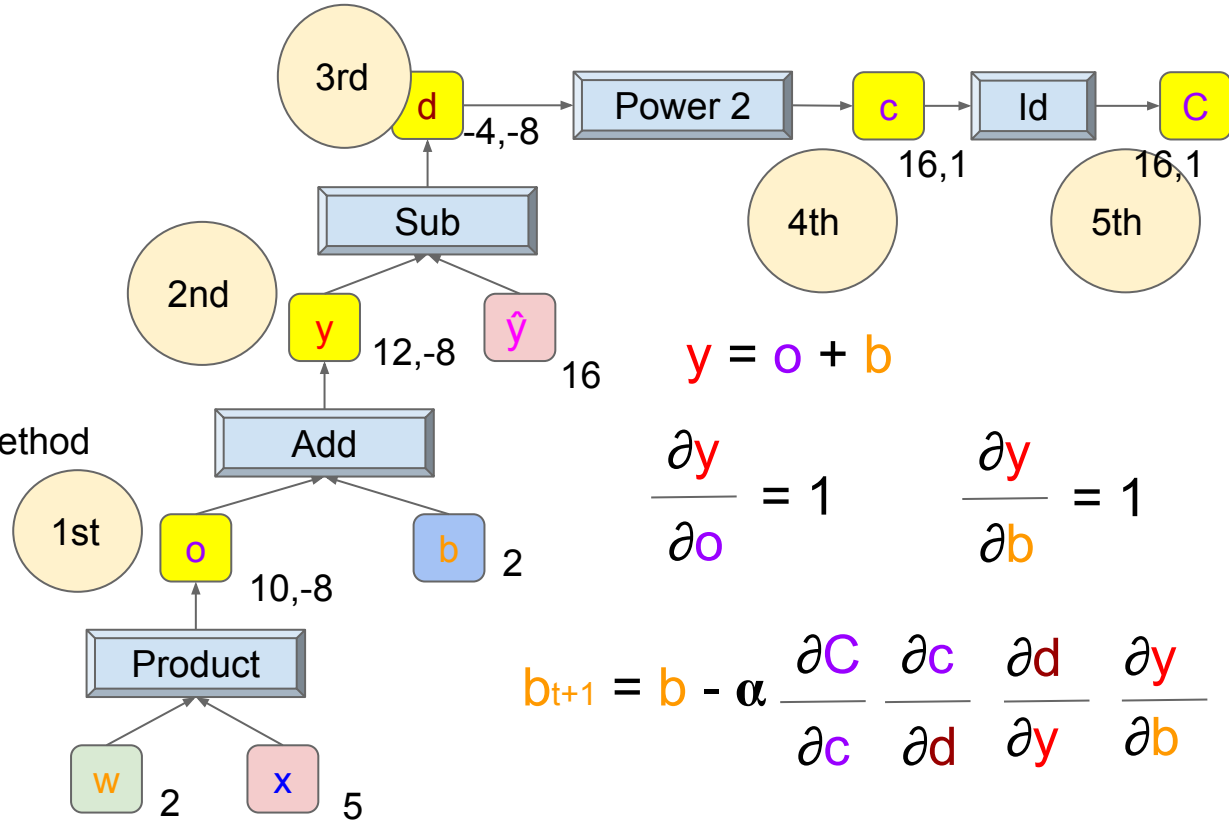
Computation Graphs are our friends

- 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)



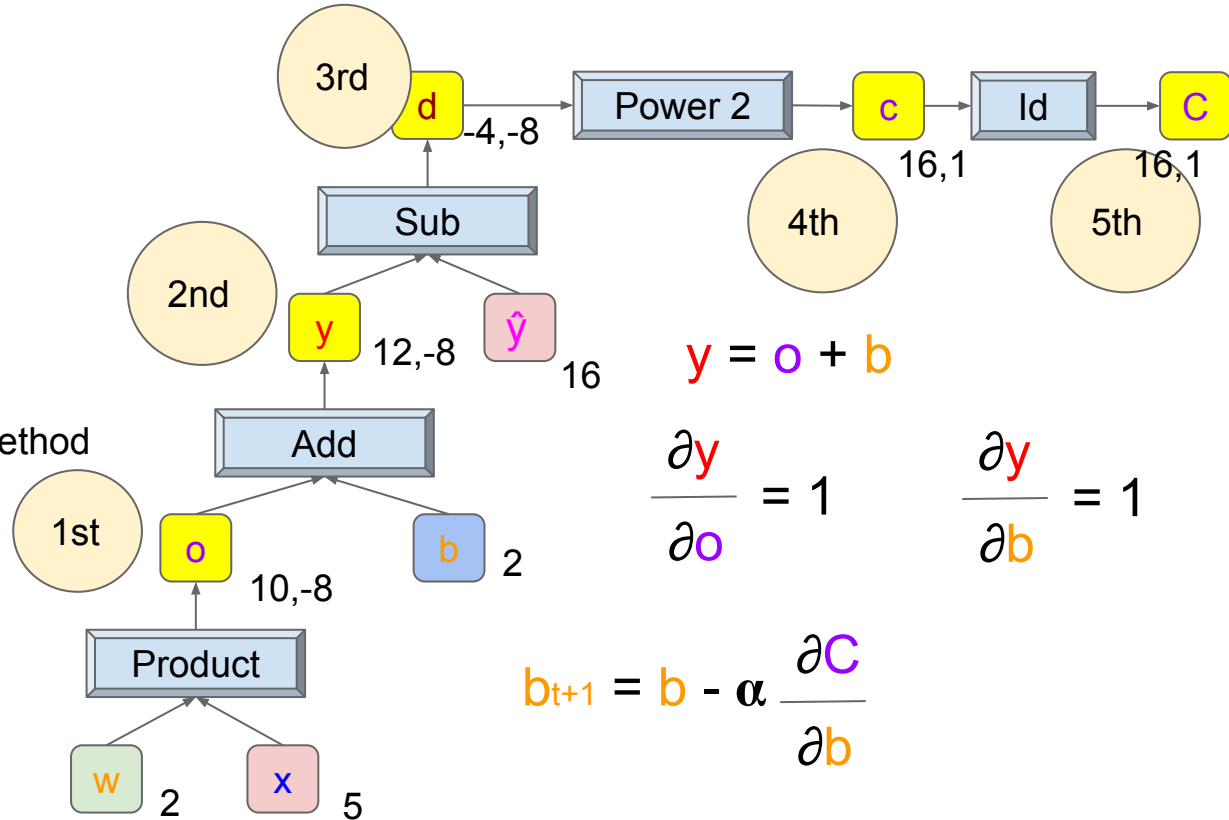
Computation Graphs are our friends

- 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)



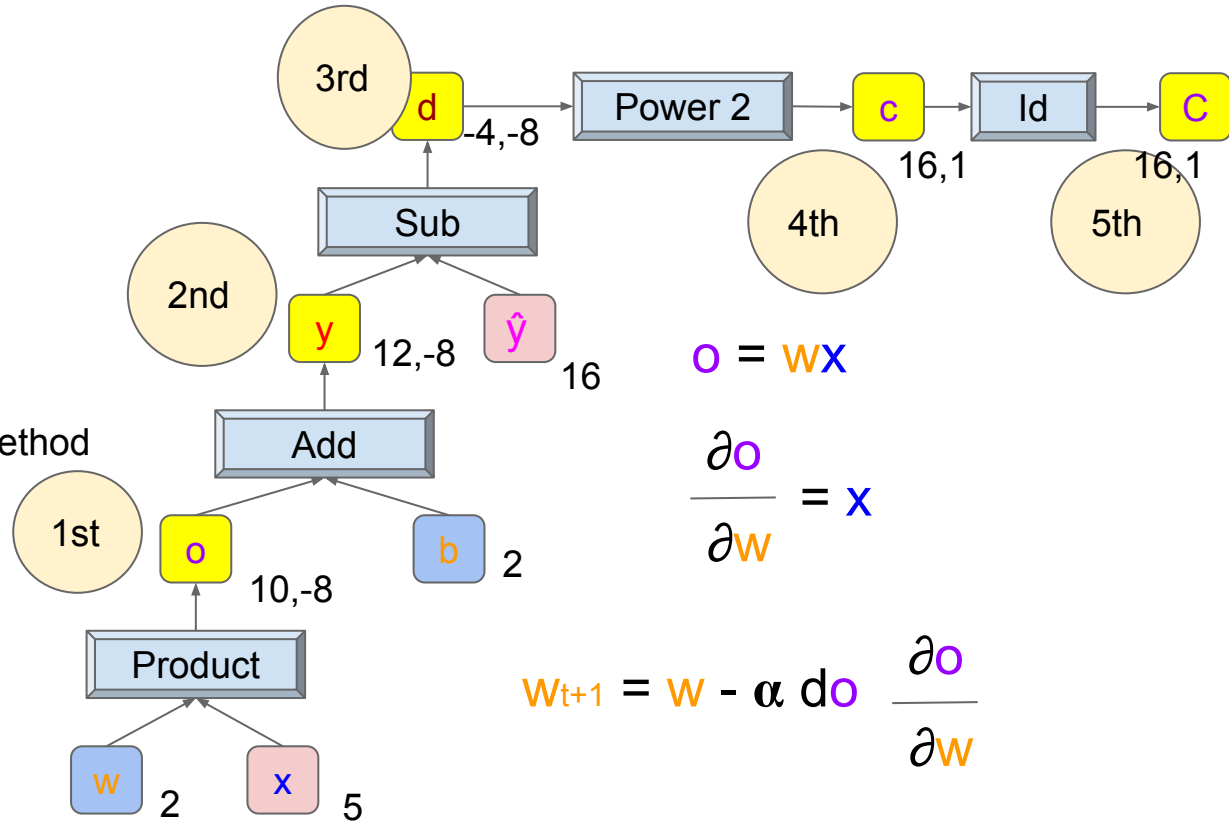
Computation Graphs are our friends

- 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)



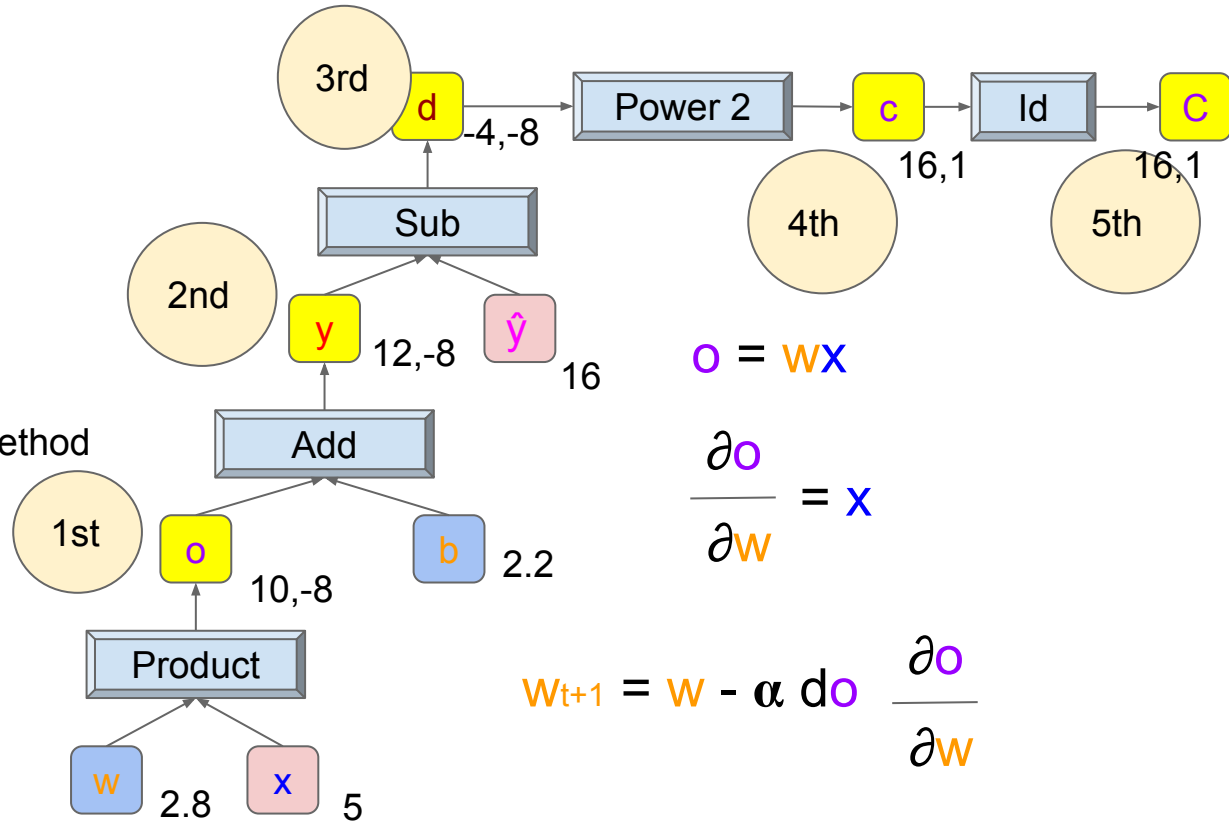
Computation Graphs are our friends

- 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

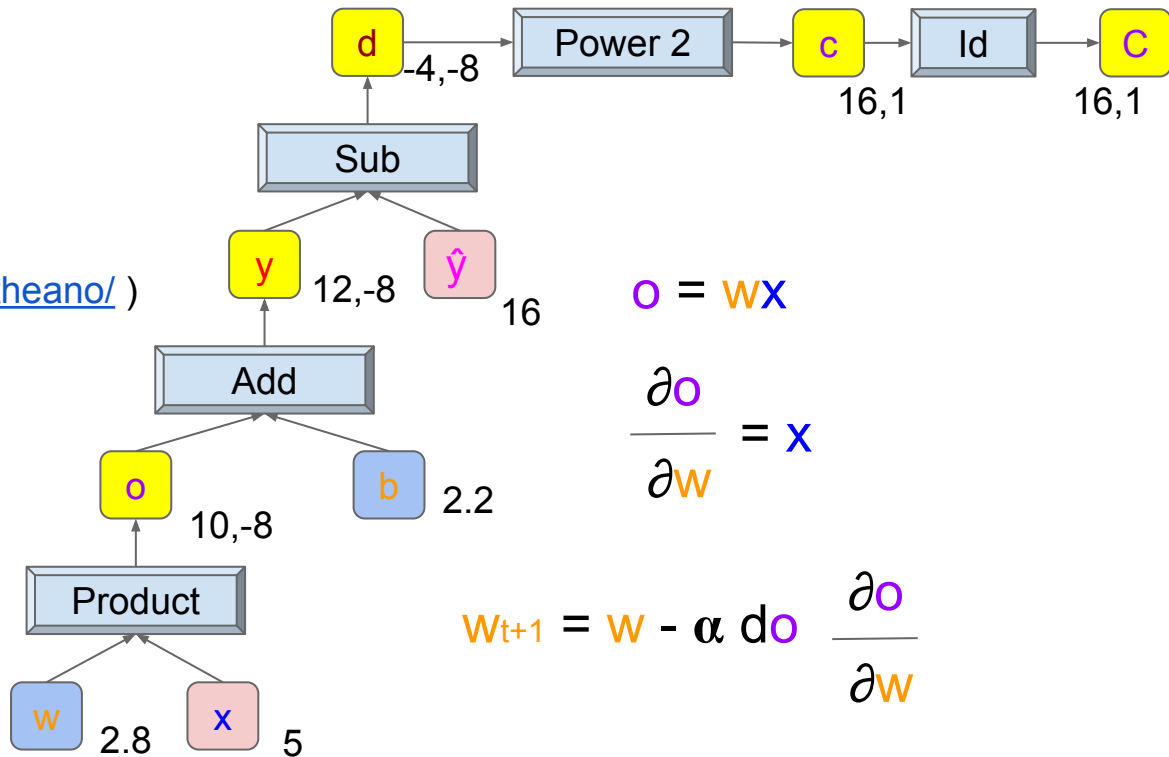
- 1-Initialize inputs
- 2-Initialize variables
- 3-Topological Sort variables
- 4-For each variable in topological order, run the forward method of all operations that link to them (Forward)
- 5-Set gradients to final variables
- 6-run the operations backward method in reverse order (Backward)
- 7-update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



Computation Graphs are our friends

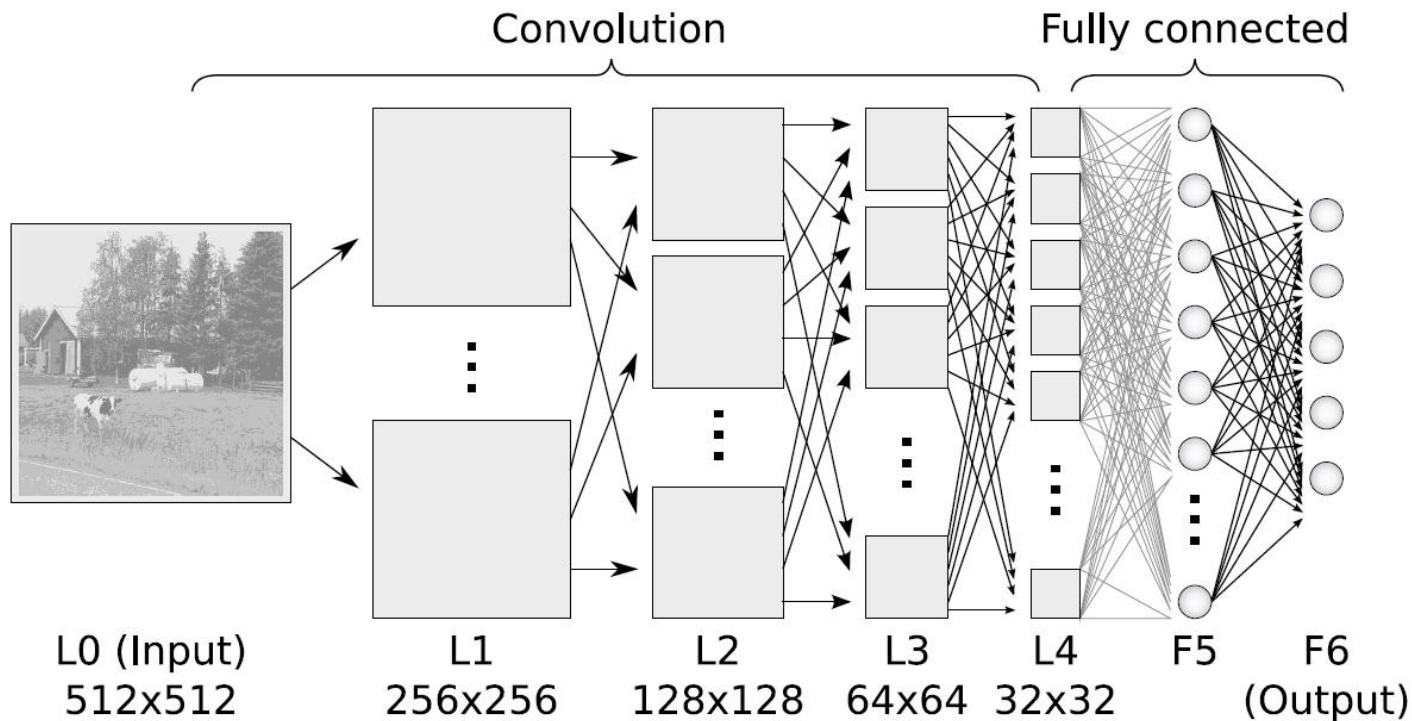
Existing Tools:

- 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/>)



Deep Neural Networks are our friends?

Convolutional Neural Network



Deep Neural Networks are our friends?

Convolutional Neural Network

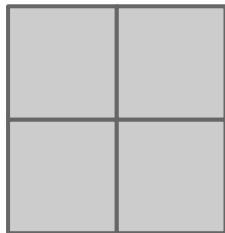
x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12
x13	x14	x15	x16

4x4 image

Deep Neural Networks are our friends?

Convolutional Neural Network

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



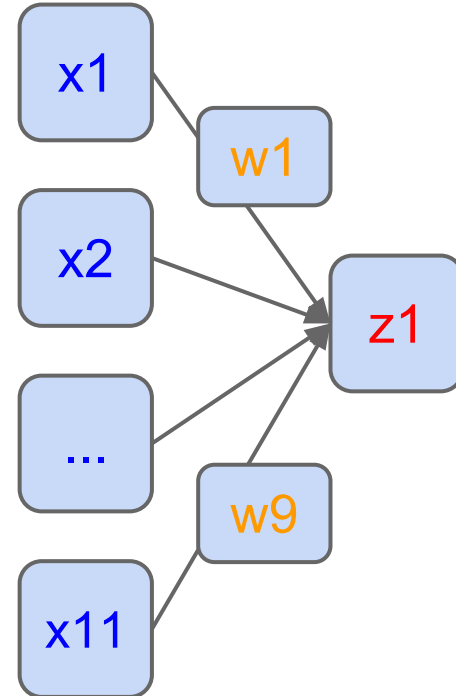
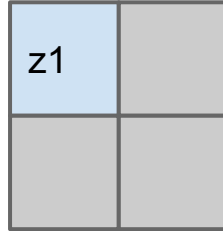
4x4 image

Deep Neural Networks are our friends?

Convolutional Neural Network

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

4x4 image



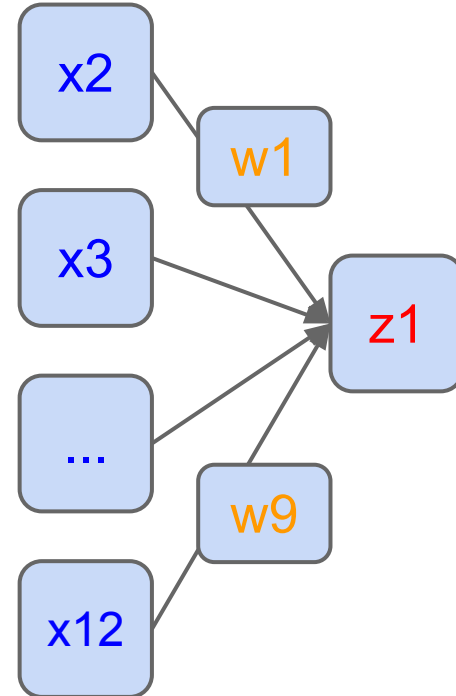
Deep Neural Networks are our friends?

Convolutional Neural Network

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

z1	z2
z3	z4

4x4 image

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

