

DEEP LEARNING FOR ARTIFICIAL INTELLIGENCE

3rd Master Course UPC ETSETB TelecomBCN Barcelona. Autumn 2019.



Instructors



Xavier
Giró-i-Nieto



Marta R.
Costa-jussà



Noé
Casas



Verónica
Vilaplana



Ramon
Morros



Javier
Ruiz



Albert
Pumarola



Jordi
Torres

Organizers



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Supporters



Google Cloud

GitHub Education

+ info: <http://bit.ly/dlai2019>

[\[course site\]](#)



#DLUPC

Day 2 Lecture 2

Backpropagation



Xavier Giro-i-Nieto

xavier.giro@upc.edu

Associate Professor

Universitat Politècnica de Catalunya
Technical University of Catalonia



Acknowledgements



Kevin McGuinness

kevin.mcguinness@dcu.ie

Research Fellow

Insight Centre for Data Analytics
Dublin City University



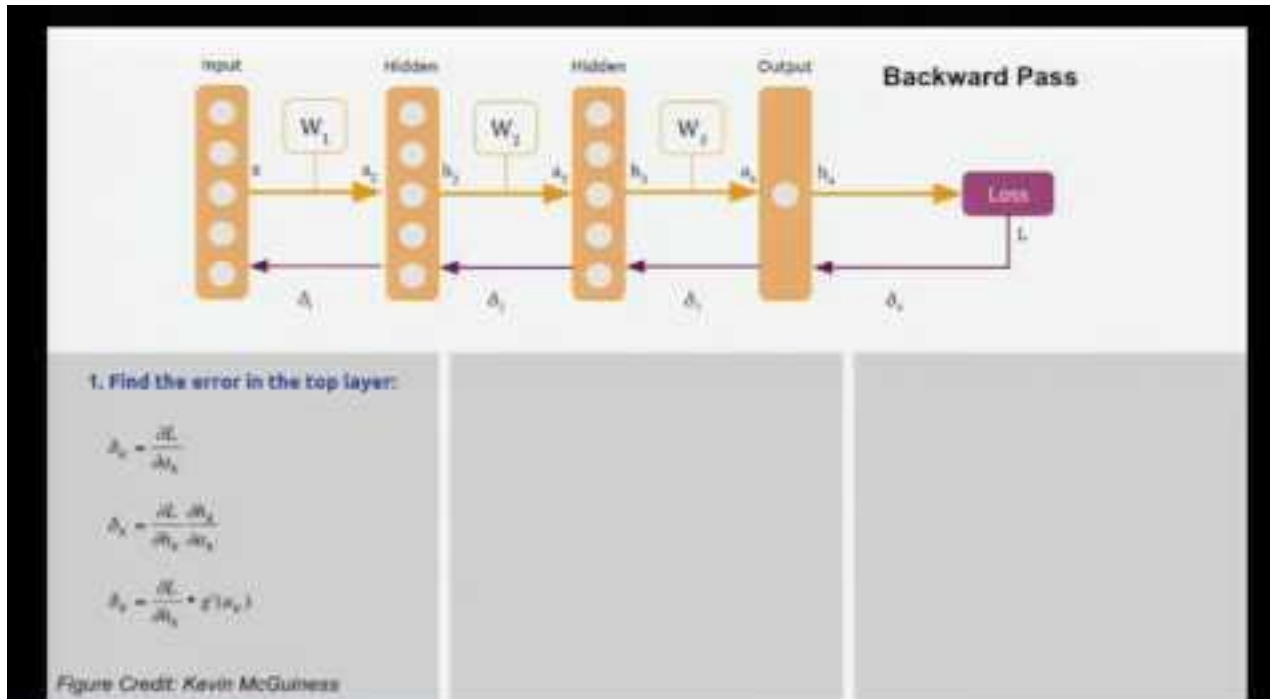
Elisa Sayrol

elisa.sayrol@upc.edu

Associate Professor

ETSETB TelecomBCN
Universitat Politècnica de Catalunya

Video lecture



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament de Teoria del Senyal
i Comunicacions

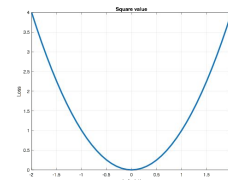
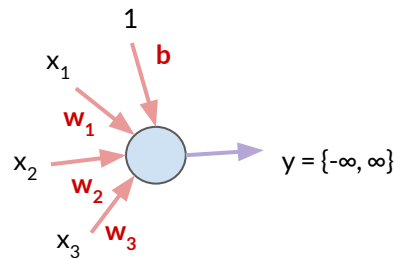
Loss function - $L(y, \hat{y})$

The **loss function** assesses the performance of our model by comparing its predictions (y) to an expected value (\hat{y}), typically coming from ground truth labels.

Example: the predicted price (y) and one actually paid (\hat{y}) could be compared with the Euclidean distance (also referred as L2 distance or Mean Square Error - MSE):

$$y = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b = \mathbf{w}^T \cdot \mathbf{x} + b$$

$$\mathcal{L}_2(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$



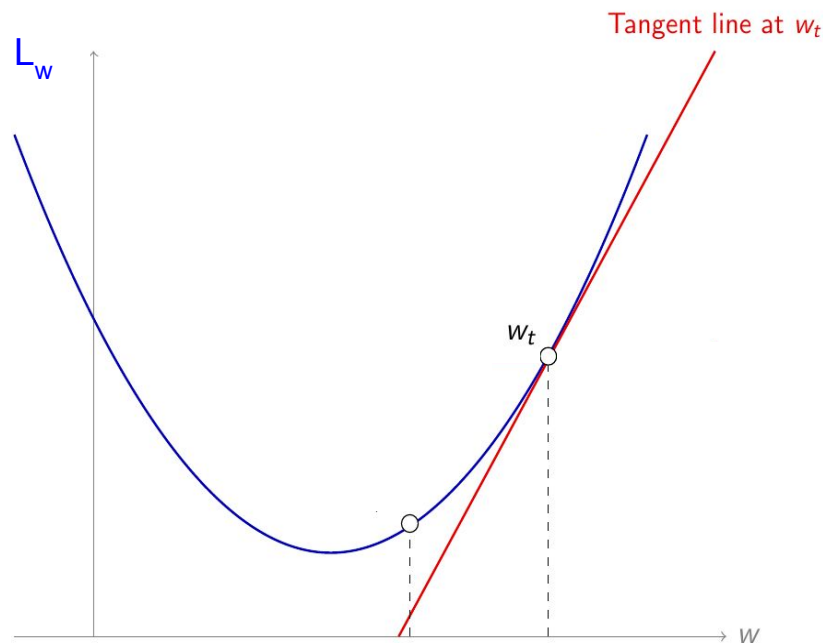
Loss function - $L(y, \hat{y})$

Discussion: Consider the very simple model...

$$y = w \cdot x$$

.....and that, given a pair (y, \hat{y}) , we would like to update the current w_t value to a new w_{t+1} based on the loss function L_w .

- (a) Would you increase or decrease w_t ?
- (b) What operation could indicate which way to go?
- (c) How much would you increase or decrease w_t ?



Gradient Descent (GD)

Motivation for this lecture:

if we had a way to compute the gradient of the loss with respect to the parameter(s), we could use gradient descent to optimize them.

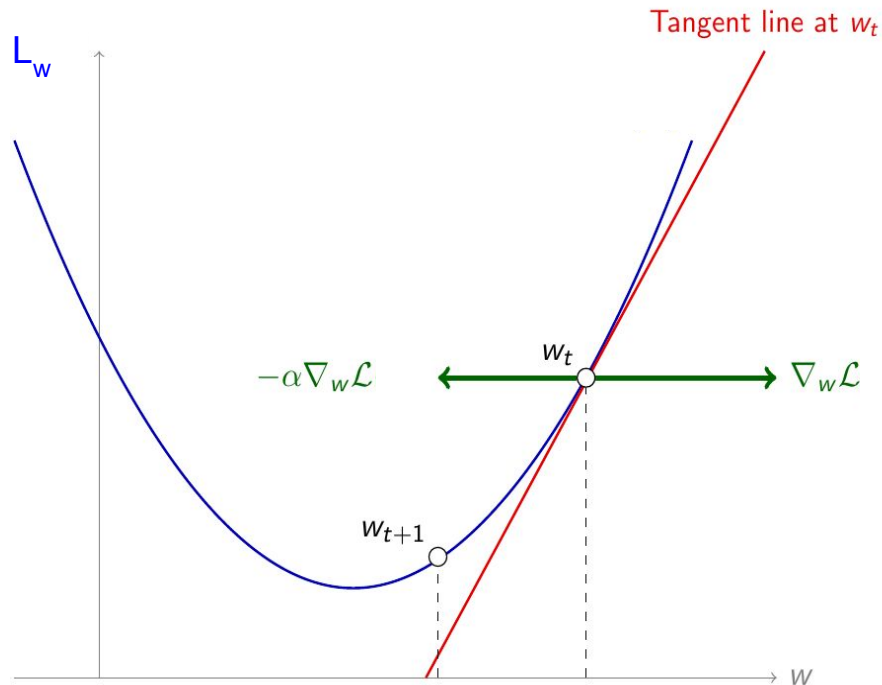
Descend
(minus sign)

↓

$$w_{t+1} \leftarrow w_t - \alpha \nabla \mathcal{L}_w(w_t)$$

↑

Learning
rate (LR)



Gradient Descent (GD)

Backpropagation will allow us to compute the gradients of the loss function with respect to:

- all model parameters (\mathbf{w}) - final goal during training
- input/intermediate data - visualization & interpretability purposes.

Gradients will “**flow**” from the output of the model towards the input (“back”).

Let the Gradient Flo

Celebrate NIPS 2017 with Intel AI

Join us for an exclusive party – and a surprise reveal.

Giveaways, buskers, acrobats, DJ Nostalgia B and a special performance by Flo Rida!

When

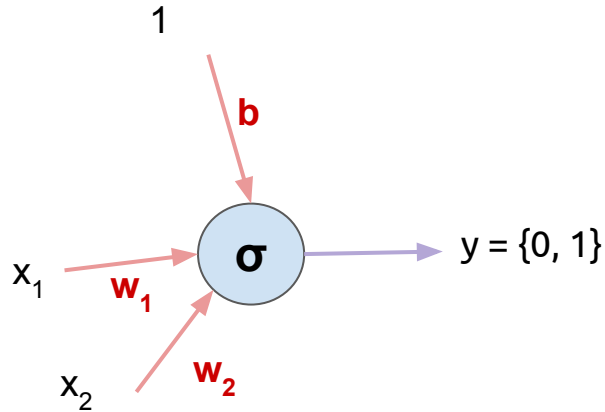
Tuesday, December 5th
9:00 PM – 12:00 AM
Door open at 9:00 PM
Show up early - space is limited

Where

The Loft on Pine
230 Pine Avenue
Long Beach, CA 90802
Near the Long Beach Convention Center

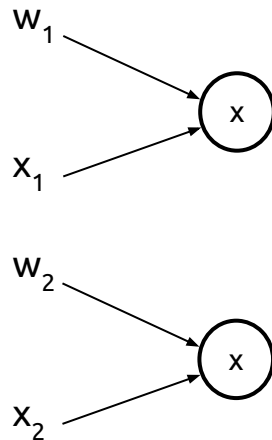
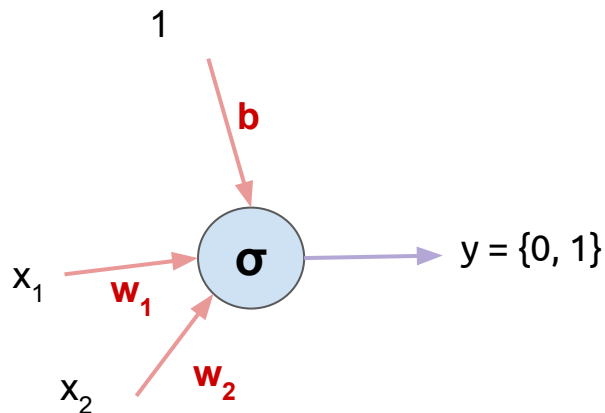


Computational graph of a simple perceptron

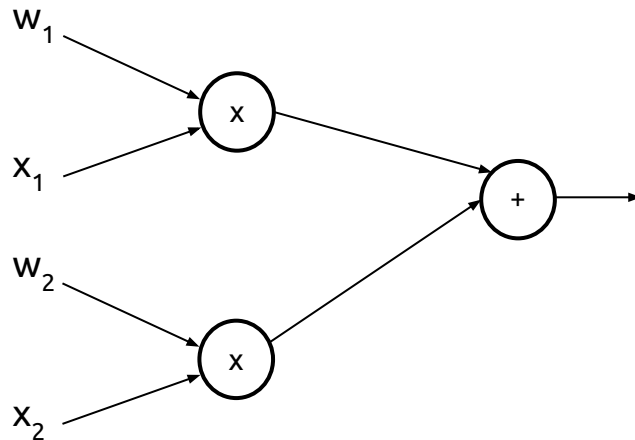
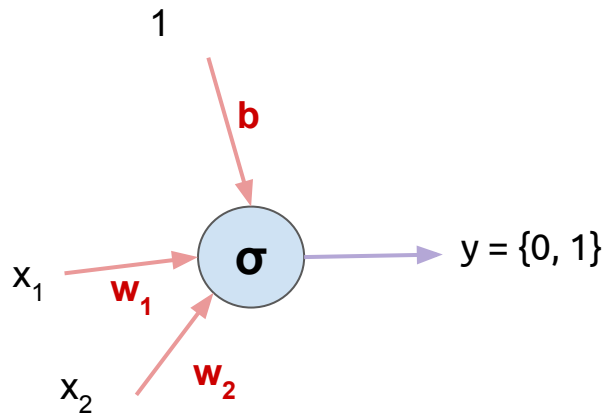


Question: What is the computational graph (operations & order) of this perceptron with a sigmoid activation ?

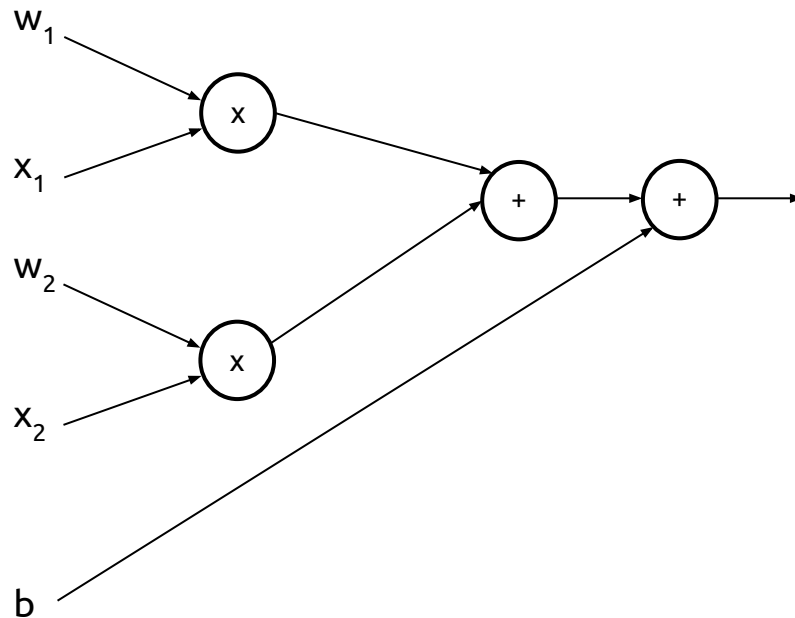
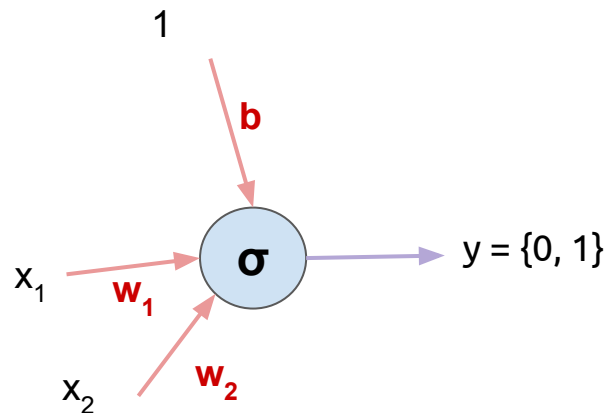
Computational graph of a perceptron



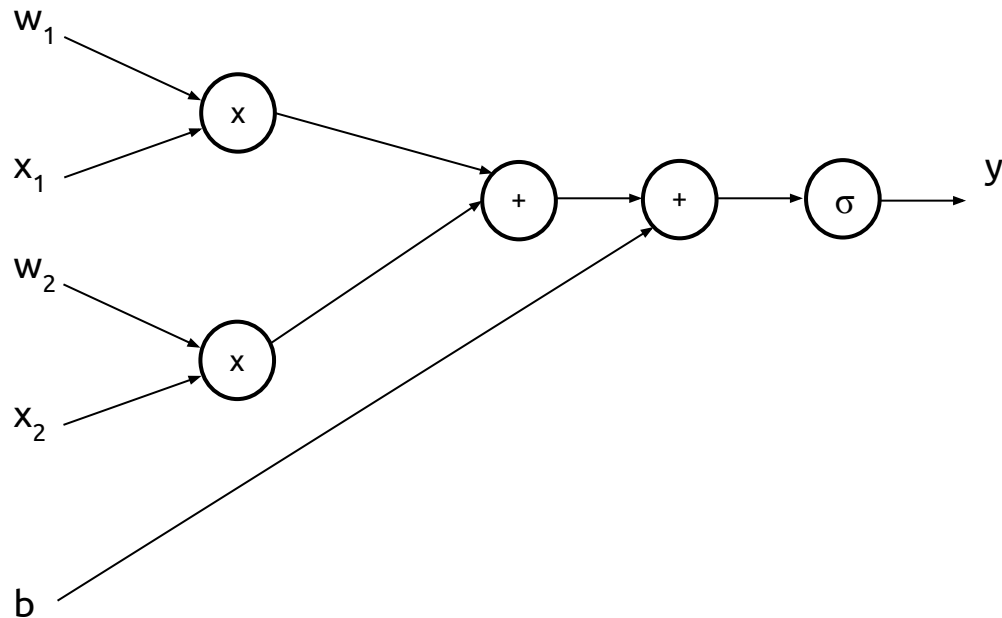
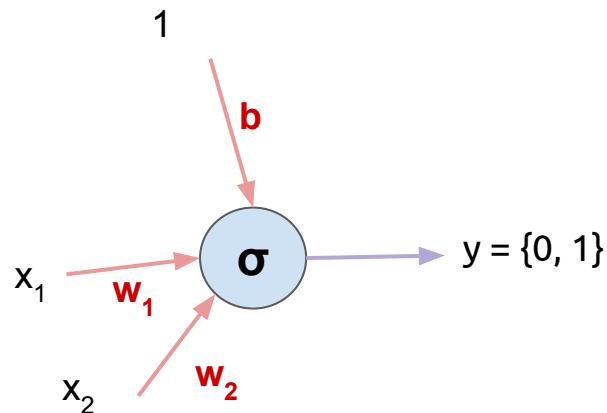
Computational graph of a perceptron



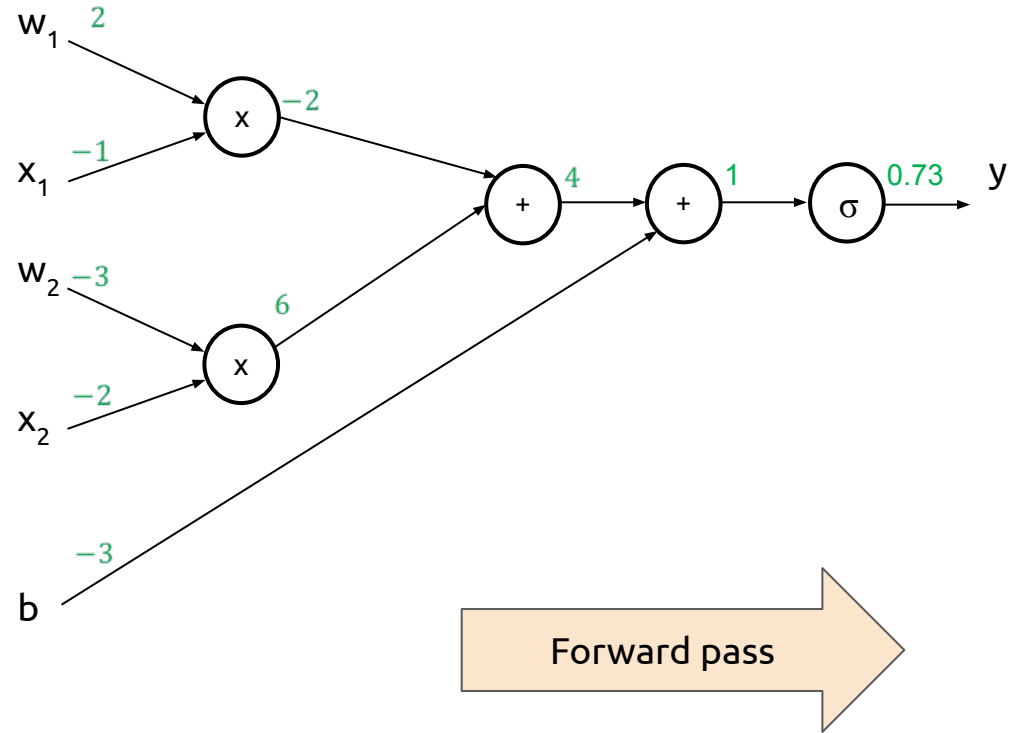
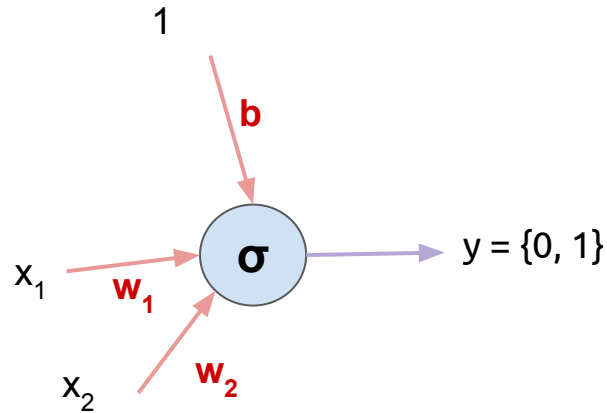
Computational graph of a perceptron



Computational graph of a perceptron



Computational graph of a perceptron

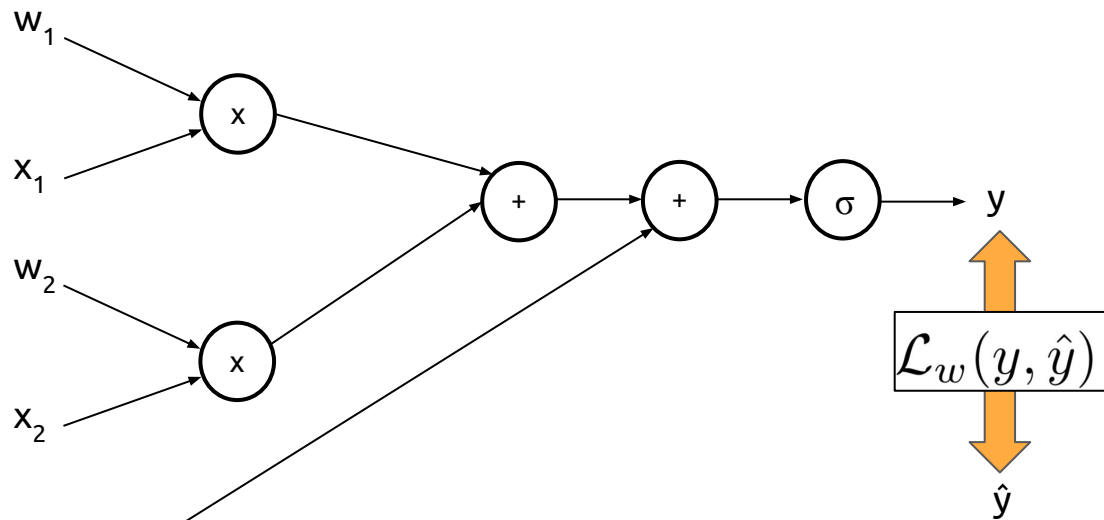


Computational graph of a perceptron

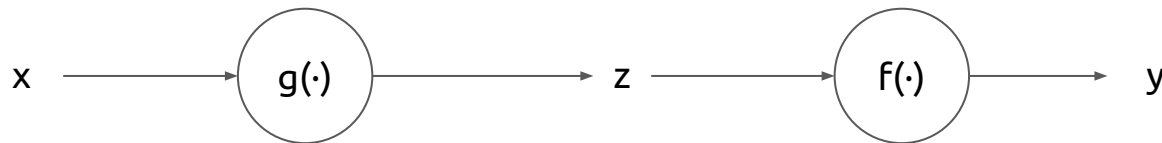
Challenge: How to compute the gradient of the loss function with respect to w_1 or w_2 ?

$$\frac{\partial \mathcal{L}_w(y, \hat{y})}{\partial w_1} = ?$$

$$\frac{\partial \mathcal{L}_w(y, \hat{y})}{\partial w_2} = ?$$

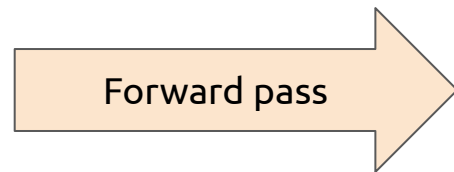


Gradients from composition (chain rule)

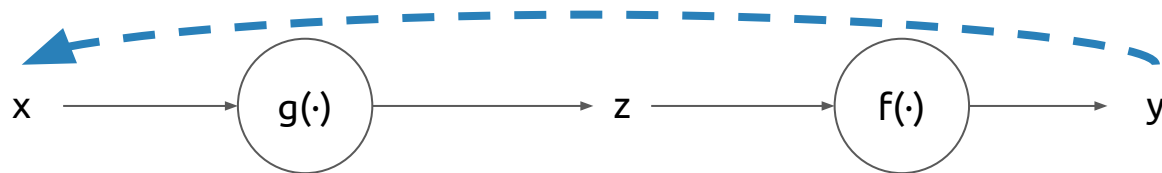


$$y = f(z)$$

$$z = g(x)$$



Gradients from composition (chain rule)



How does a variation
("difference") on x
affect y ?

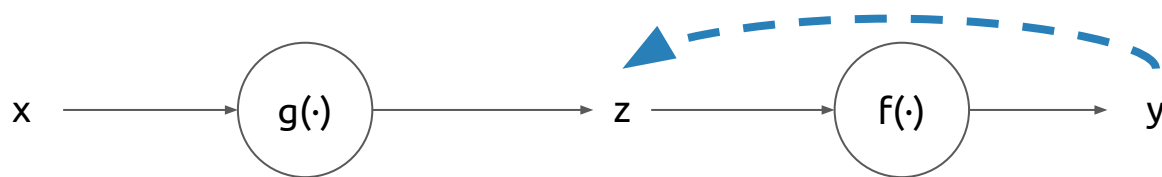
$$y = f(z)$$

$$z = g(x)$$

$$\frac{dy}{dx} = ?$$

Backward pass

Gradients from composition (chain rule)



How does a variation
("difference") on x
affect y ?

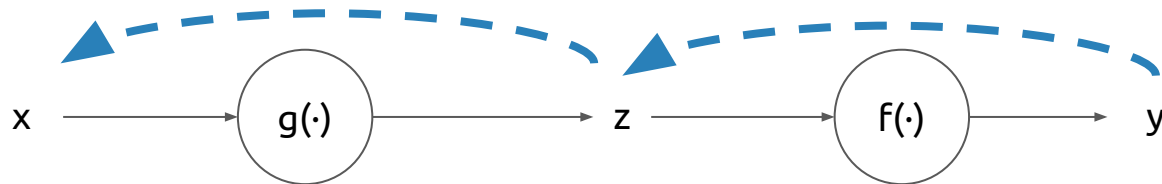
$$y = f(z)$$
$$z = g(x)$$

It will depend on the
how y is affected by
a variation in z ...

$$\frac{dy}{dx} = \frac{dy}{dz} \frac{dz}{dx}$$

Backward pass

Gradients from composition (chain rule)



How does a variation
("difference") on x
affect y ?

$$y = f(z)$$
$$z = g(x)$$

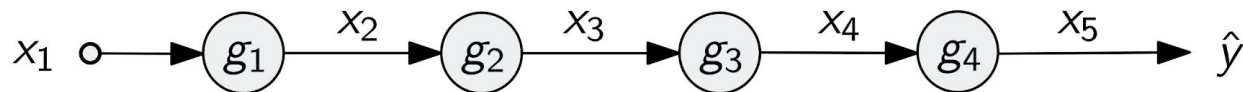
It will depend on the
how y is affected by
a variation in z ...

$$\frac{dy}{dx} = \frac{dy}{dz} \frac{dz}{dx}$$

...scaled (multiplied)
by how z is affected
with a variation in x .

Gradients from composition (chain rule)

$$\hat{y} = g_4(g_3(g_2(g_1(x_1))))$$



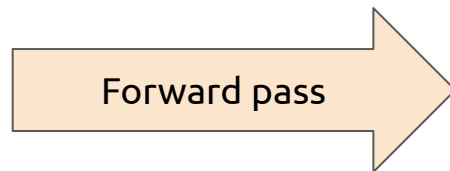
Decompose into steps (**forward propagation**):

$$x_2 = g_1(x_1)$$

$$x_3 = g_2(x_2)$$

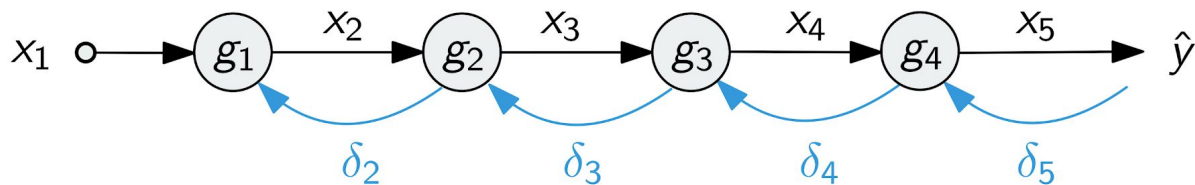
$$x_4 = g_3(x_3)$$

$$\hat{y} = x_5 = g_4(x_4)$$



Gradients from composition (chain rule)

$$\hat{y} = g_4(g_3(g_2(g_1(x_1))))$$



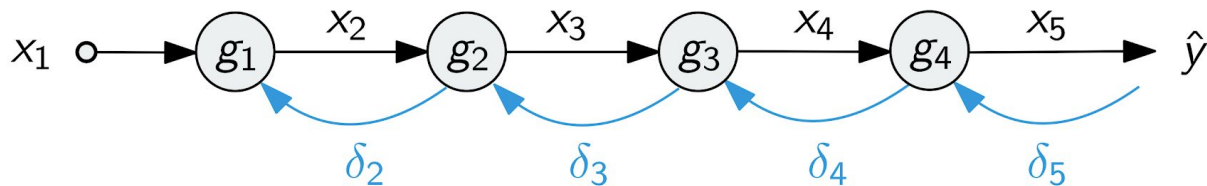
Want to find $\frac{\partial \hat{y}}{\partial x_1}$. Chain rule:

○ ○ ○
$$\frac{\partial \hat{y}}{\partial x_1} = \frac{\partial \hat{y}}{\partial x_4} \frac{\partial x_4}{\partial x_3} \frac{\partial x_3}{\partial x_2} \frac{\partial x_2}{\partial x_1}$$

How does a variation
("difference") on the
input affect the
prediction?

Backward pass

Gradients from composition (chain rule)



Decompose into steps again. Let $\delta_k = \frac{\partial \hat{y}}{\partial x_k}$. **Backpropagation:**

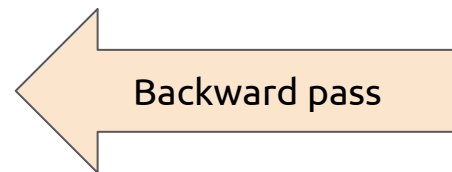
$$\delta_5 = \frac{\partial \hat{y}}{\partial x_5} = 1$$

$$\delta_4 = \frac{\partial \hat{y}}{\partial x_4} = \frac{\partial \hat{y}}{\partial x_5} \frac{\partial x_5}{\partial x_4} = \delta_5 g'_4(x_4)$$

$$\delta_3 = \frac{\partial \hat{y}}{\partial x_3} = \frac{\partial \hat{y}}{\partial x_4} \frac{\partial x_4}{\partial x_3} = \delta_4 g'_3(x_3)$$

$$\delta_2 = \frac{\partial \hat{y}}{\partial x_2} = \frac{\partial \hat{y}}{\partial x_3} \frac{\partial x_3}{\partial x_2} = \delta_3 g'_2(x_2)$$

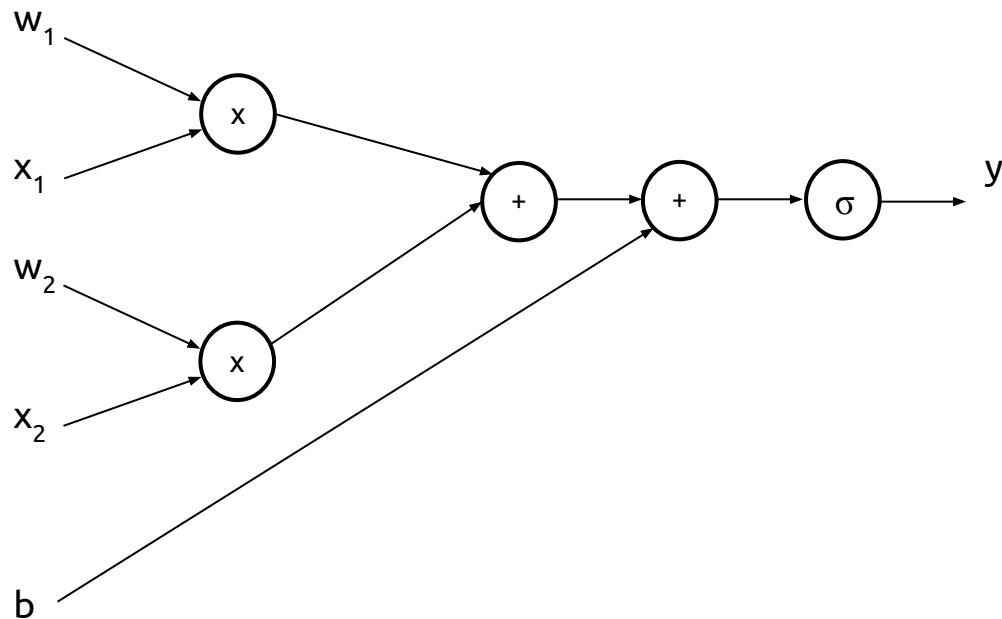
$$\delta_1 = \frac{\partial \hat{y}}{\partial x_1} = \frac{\partial \hat{y}}{\partial x_2} \frac{\partial x_2}{\partial x_1} = \delta_2 g'_1(x_1)$$



Gradients from composition (chain rule)

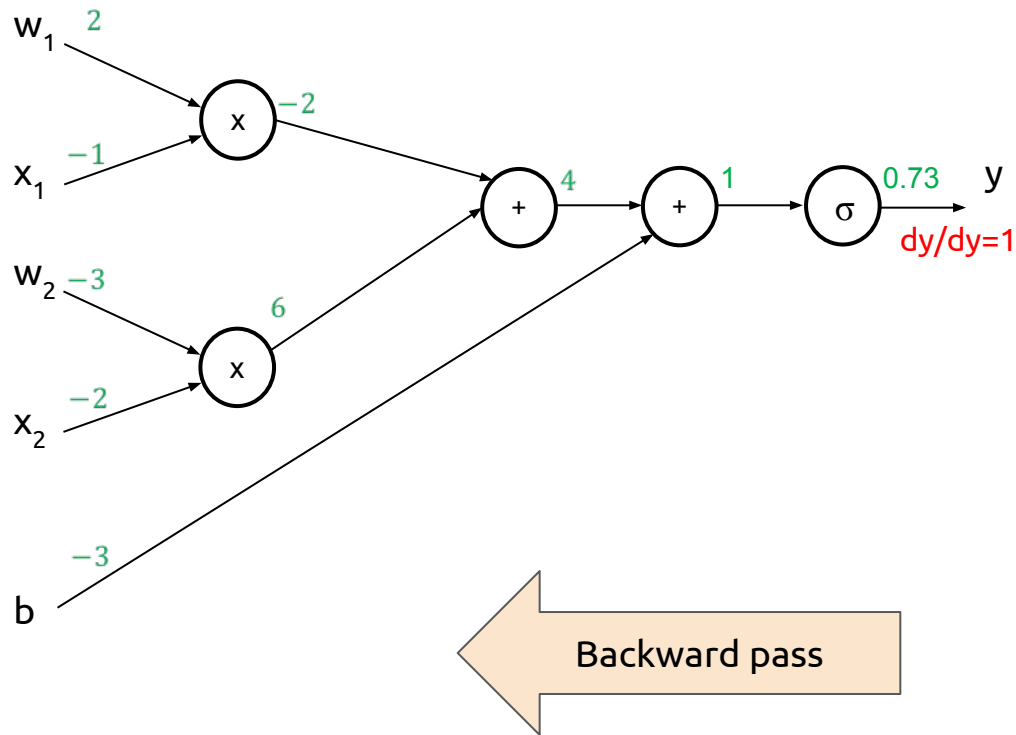
Question: How can gradients be backpropagated for the operations involved in a perceptron?

- PRODUCT (\times)
- SUM (+)
- SIGMOID (σ)



Gradient backpropagation in a perceptron

We can now estimate the sensitivity of the output y with respect to each input parameter w_i and x_i .



Gradient weights for sigmoid σ

(*)

$$\frac{\partial \sigma(x)}{\partial x} = \frac{\partial}{\partial x} \left(\frac{1}{1 + e^{-x}} \right) = \frac{-1}{(1 + e^{-x})^2} \frac{\partial(1 + e^{-x})}{\partial x} = \frac{-1}{(1 + e^{-x})^2} \frac{\partial(e^{-x})}{\partial x}$$

$$(*) f(x) = \frac{g(x)}{h(x)} \quad f'(x) = \frac{g'(x)h(x) - g(x)h'(x)}{h^2}$$

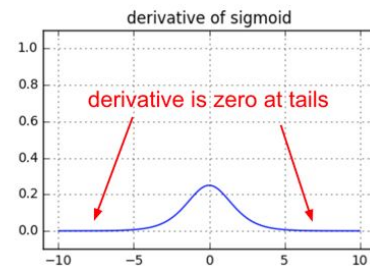
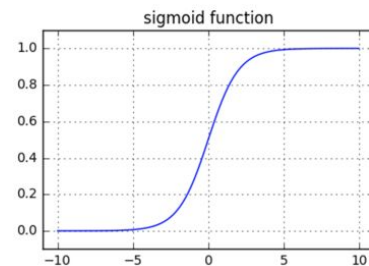
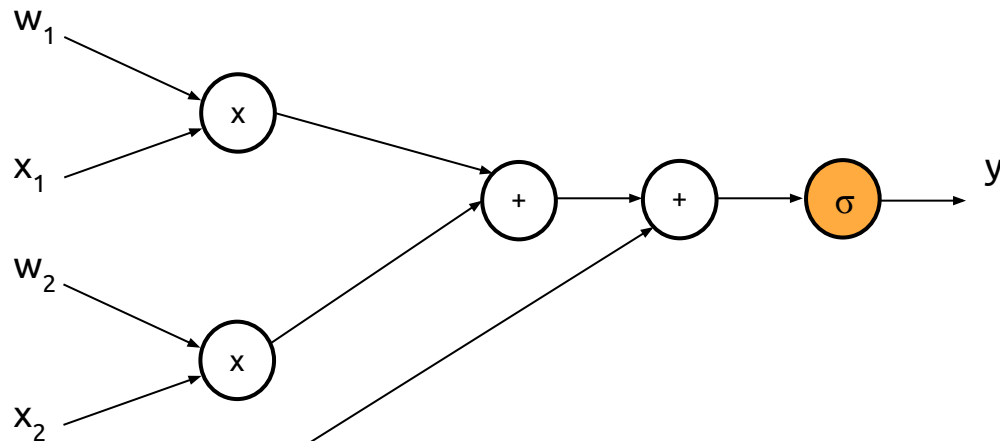
$$\frac{\partial \sigma(x)}{\partial x} = \frac{-1}{(1 + e^{-x})^2} (-e^{-x}) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

...which can be re-arranged as...

$$\frac{\partial \sigma(x)}{\partial x} = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{e^{-x}}{(1 + e^{-x})} \frac{1}{(1 + e^{-x})}$$

$$\frac{\partial \sigma(x)}{\partial x} = \left(\frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} \right) \sigma(x)$$

$$\frac{\partial \sigma(x)}{\partial x} = (1 - \sigma(x)) \sigma(x)$$



Gradient backpropagation in a perceptron

$$\frac{\partial \sigma(x)}{\partial x} = (1 - \sigma(x)) \sigma(x)$$

```
import math

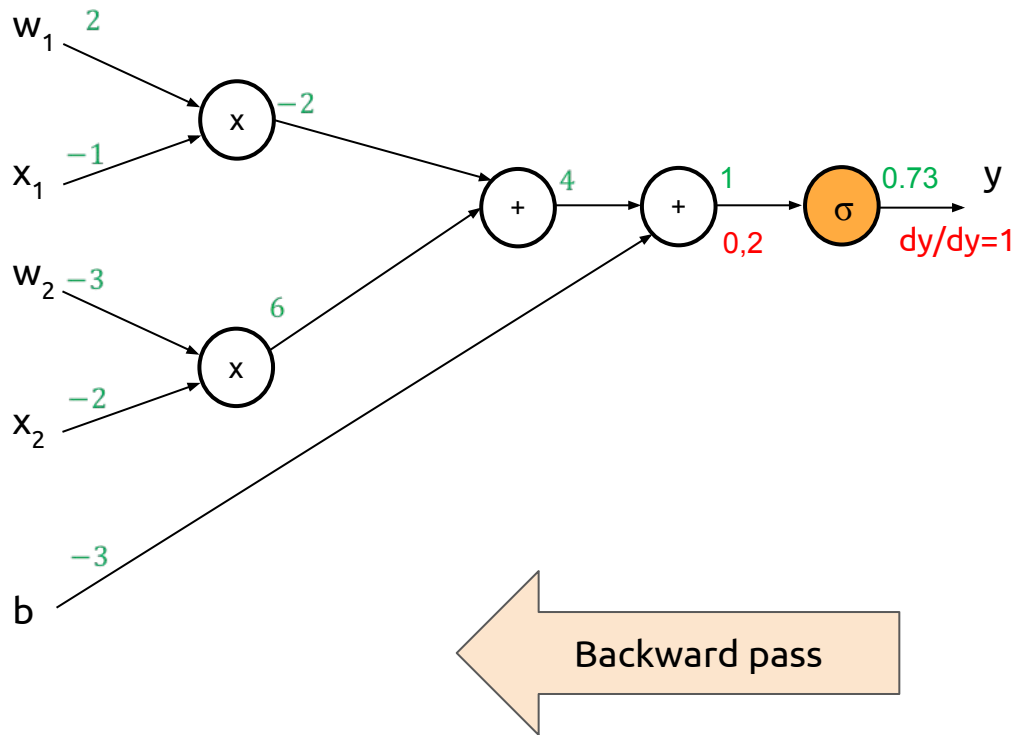
dot=1

# sigmoid function
f = 1.0 / (1 + math.exp(-dot))

# gradient on dot variable,
ddot = (1 - f) * f

print(ddot)
```

0.19661193324148185

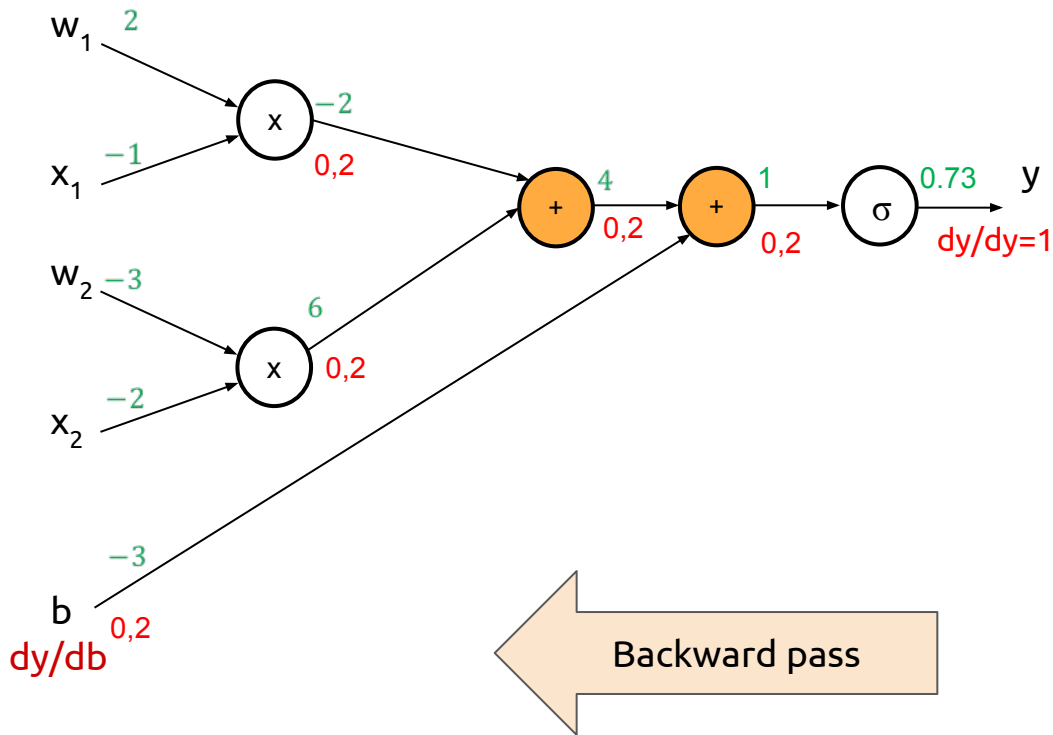


Gradient backpropagation in a perceptron

Sum: Distributes the gradient to both branches.

$$\frac{\partial(a + b)}{\partial a} = 1$$

$$\frac{\partial(a + b)}{\partial b} = 1$$

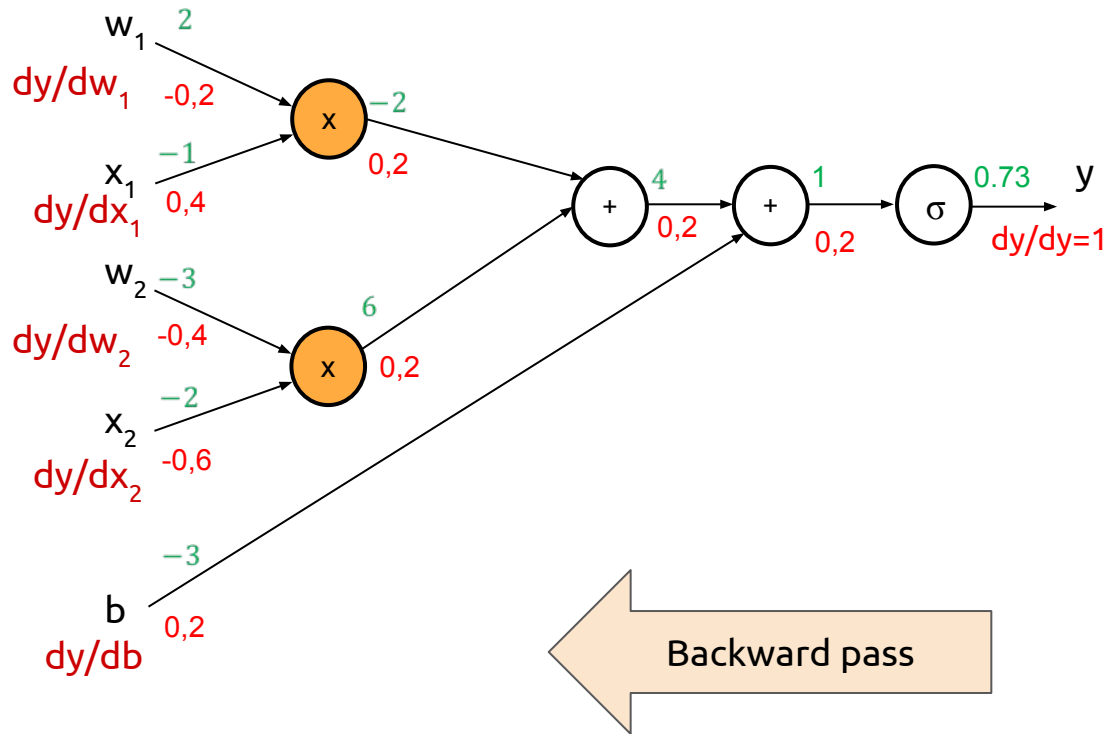


Gradient backpropagation in a perceptron

Product: Switches gradient weight values.

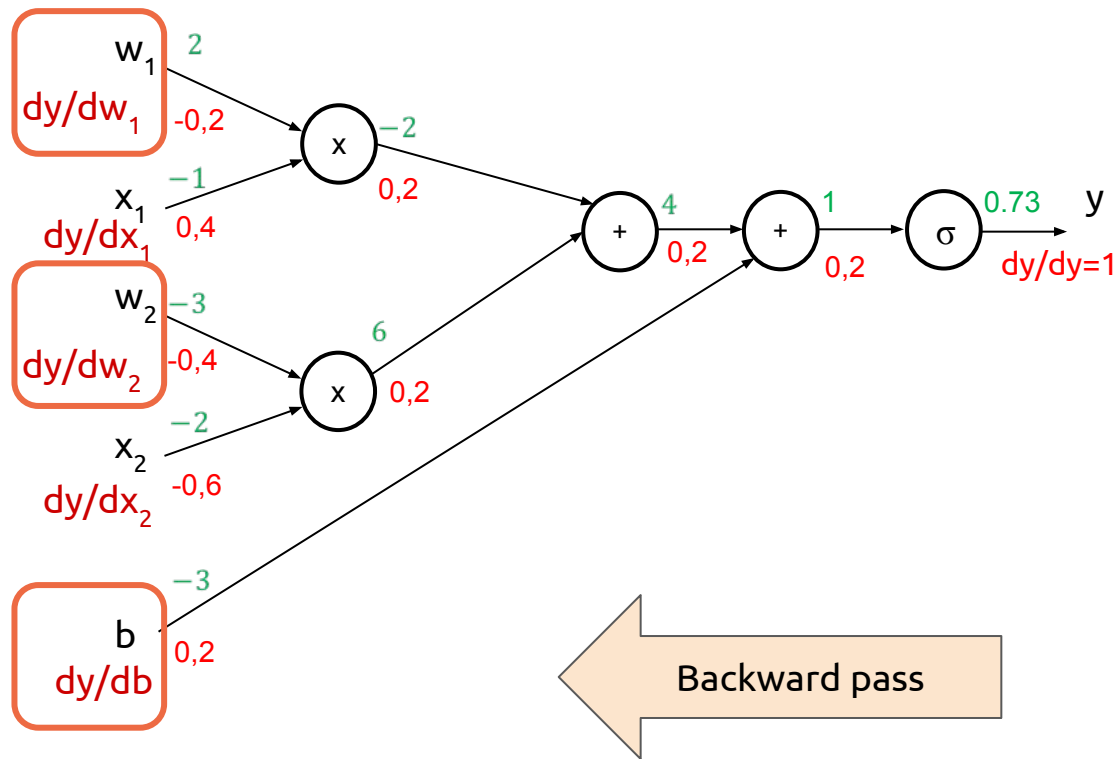
$$\frac{\partial(a \cdot b)}{\partial a} = b$$

$$\frac{\partial(a \cdot b)}{\partial b} = a$$



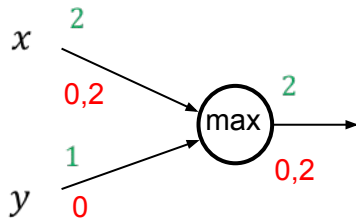
Gradient backpropagation in a perceptron

Normally, we will be interested only on the weights (w_i) and biases (b), not the inputs (x_i). The weights are the parameters to learn in our models.

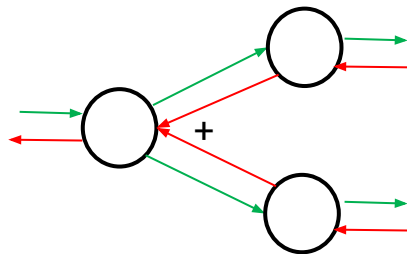


(bonus) Gradients weights for MAX & SPLIT

Max: Routes the gradient only to the higher input branch (not sensitive to the lower branches).



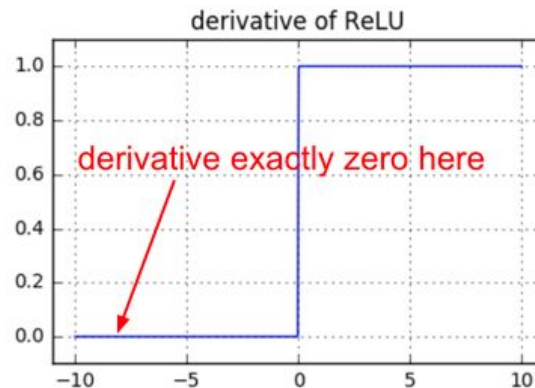
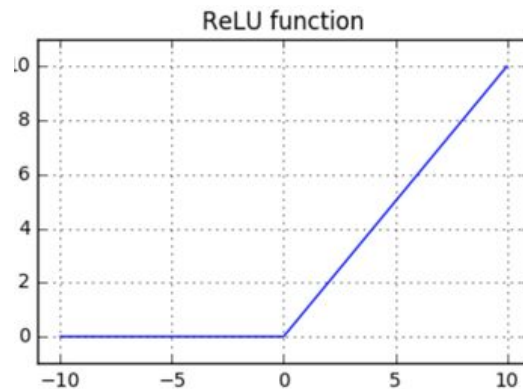
Split: Branches that split in the forward pass and merge in the backward pass, add gradients



(bonus) Gradient weights for ReLU

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

$$\frac{\partial \text{ReLU}(x)}{\partial x} = u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$



Learn more

READ

- Chris Olah, [“Calculus on Computational Graphs: Backpropagation”](#) (2015).
- Andrej Karpathy,, [“Yes, you should understand backprop”](#) (2016), and his [“course notes”](#) at Stanford University CS231n.

WATCH

- Gilbert Strang, [“27. Backpropagation: Find Partial Derivatives”](#). MIT 18.065 (2018)



Josh Gordon
@random_forests



What are the clearest explanations of backprop on the web? The two that I happily point students to are...

[cs231n.github.io/optimization-2/](#)
[colah.github.io/posts/2015-08-...](#)

Any others?

[Tradueix el tuit](#)

9:47 p. m. · 16 jul. 2019 · [Twitter Web App](#)

Undergradese

What undergrads ask vs. what they're REALLY asking

"Is it going to be an open book exam?"

Translation: "I don't have to actually memorize anything, do I?"

"Hmm, what do you mean by that?"

Translation: "What's the answer so we can all go home."

"Are you going to have office hours today?"

Translation: "Can I do my homework in your office?"

"Can i get an extension?"

Translation: "Can you re-arrange your life around mine?"

"Is this going to be on the test?"

Translation: "Tell us what's going to be on the test."

"Is grading going to be curved?"

Translation: "Can I do a mediocre job and still get an A?"

JORGE CHAM © 2008

