

# Encuentro sincrónico 3

Optimizadores-Early Stopping

# Agenda

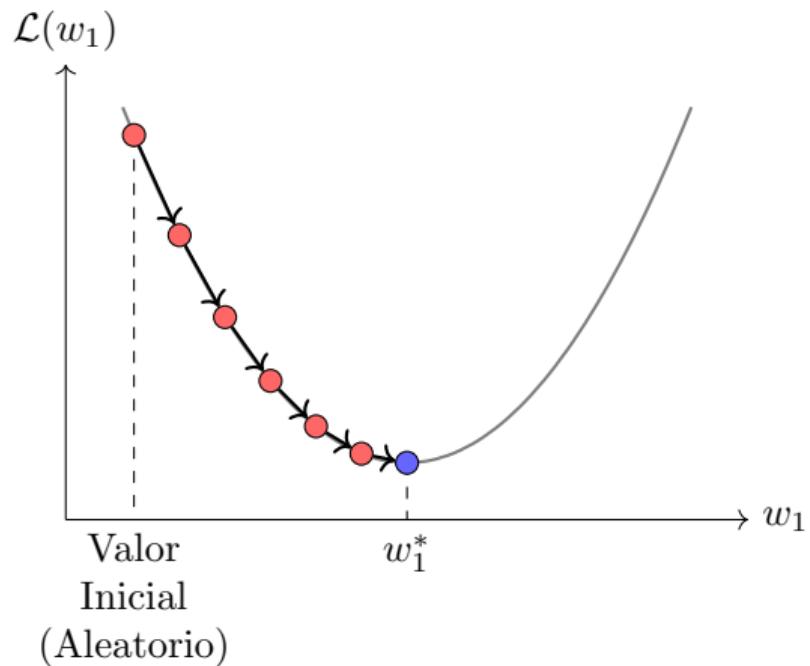
## Temas:

- Gradiente descendente y sus variaciones.
- Parada temprana.

## Table of Contents

- ▶ Gradiente estocástico
- ▶ Para temprana (Early stopping)

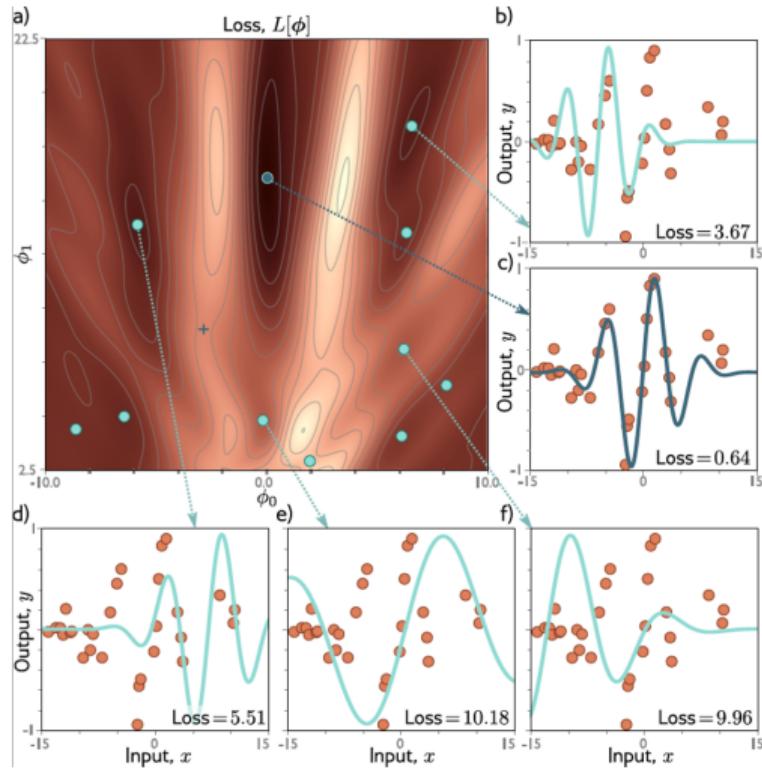
# Gradiente descendente



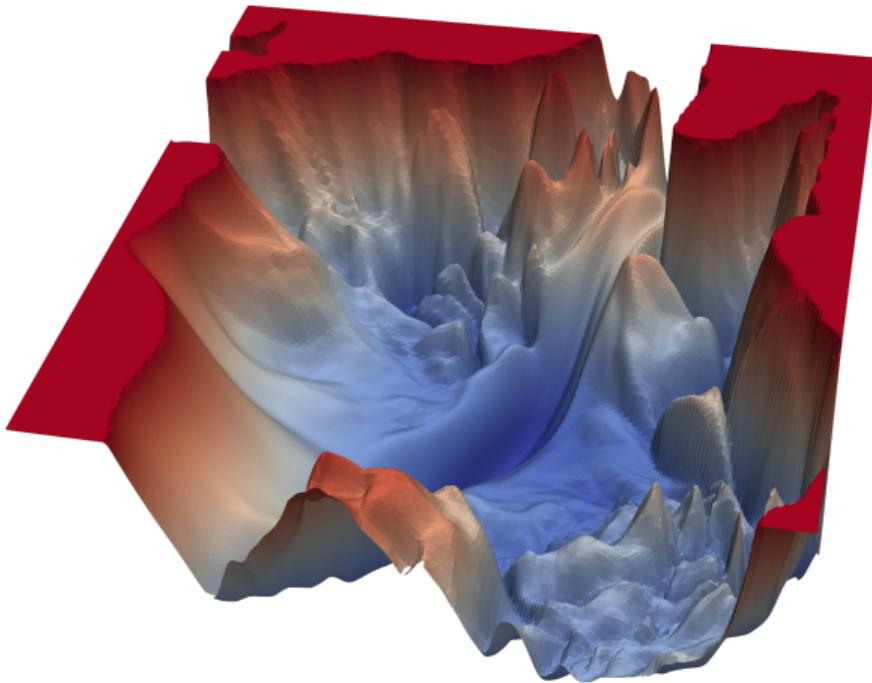
$$w^{(t+1)} = w^{(t)} - \alpha \nabla_w \mathcal{L}(w)$$

- Hay dos factores que controlan el tamaño del paso: el gradiente y la tasa de aprendizaje.
- El gradiente tiende a cero en la medida que la optimización se acerca a un mínimo.
- Es común encontrar que en muchas aplicaciones se usa una tasa de aprendizaje que varía con las iteraciones.

# Gradiente descendente: Mínimos locales I [4]



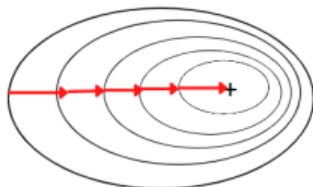
# Gradiente descendente: Mínimos locales II



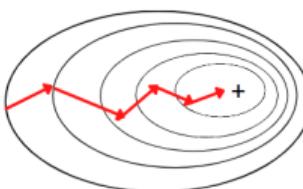
- Por otro lado las funciones de costo son de una dimensión considerablemente alta.
- Si una red tiene un millón de parámetros, la dimensión de la función de costo tendrá ese mismo orden.
- Un artículo reciente [1] presenta una metodología que permite tener una aproximación 3D de las funciones de costo de alta dimensión (VGG-56).

# Gradiente descendente por mini-lotes y gradiente estocástico I

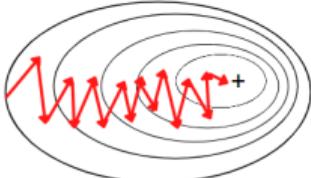
Batch Gradient Descent



Mini-Batch Gradient Descent

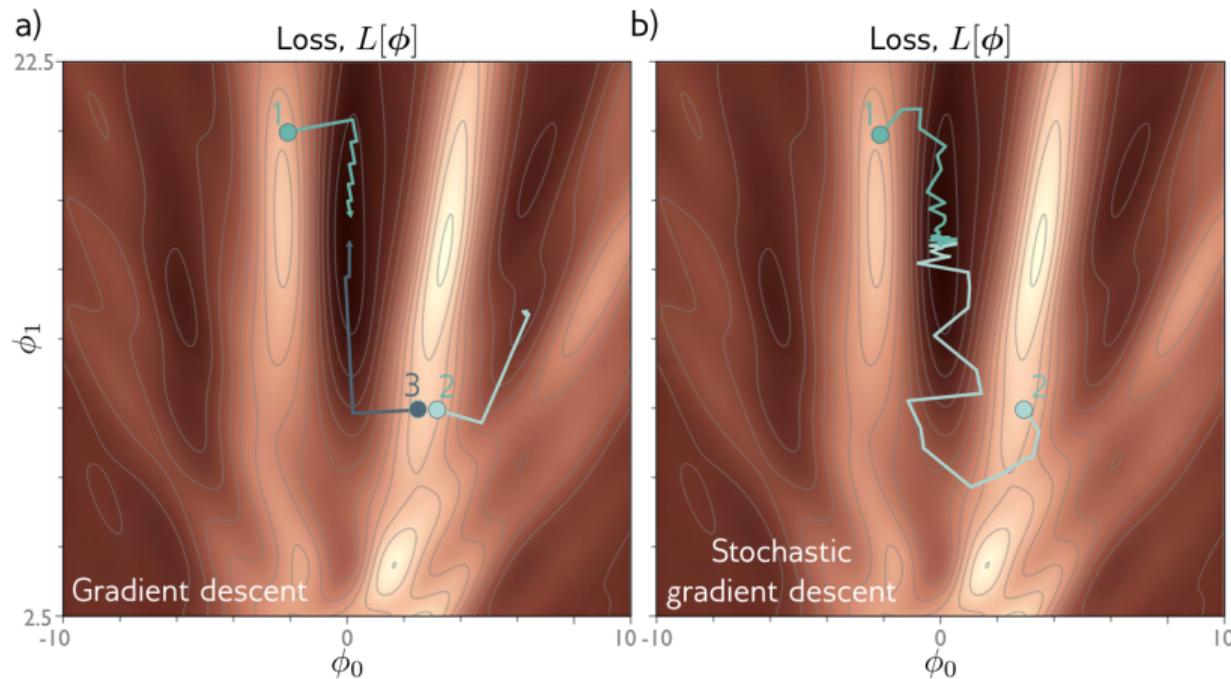


Stochastic Gradient Descent

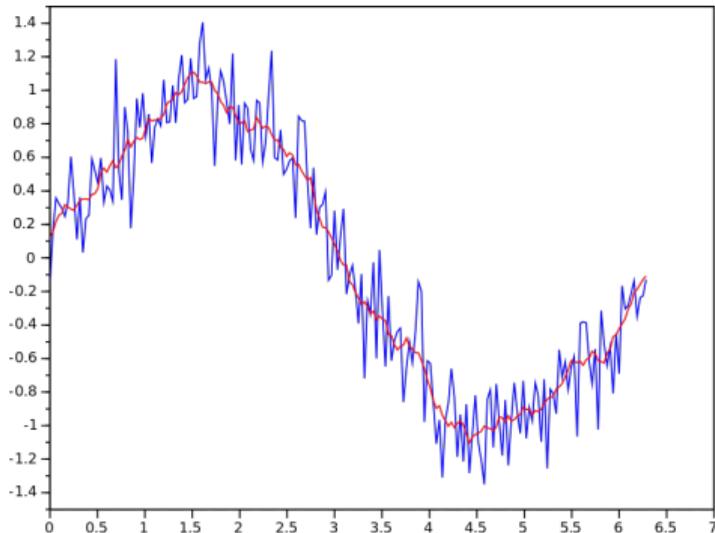


- Usualmente, se usan todos los datos de entrenamiento para calcular el gradiente. Esto conduce a puntos de silla o mínimos locales.
- Una solución está relacionada con el uso de gradientes ruidosos. La aleatoriedad ayuda a salir de mínimos locales.
- Se pueden calcular gradientes ruidosos usando una única muestra del entrenamiento (gradiente estocástico) o usando lotes pequeños (gradiente por mini-lotes).

# Gradiente descendente por mini-lotes y gradiente estocástico II [4]

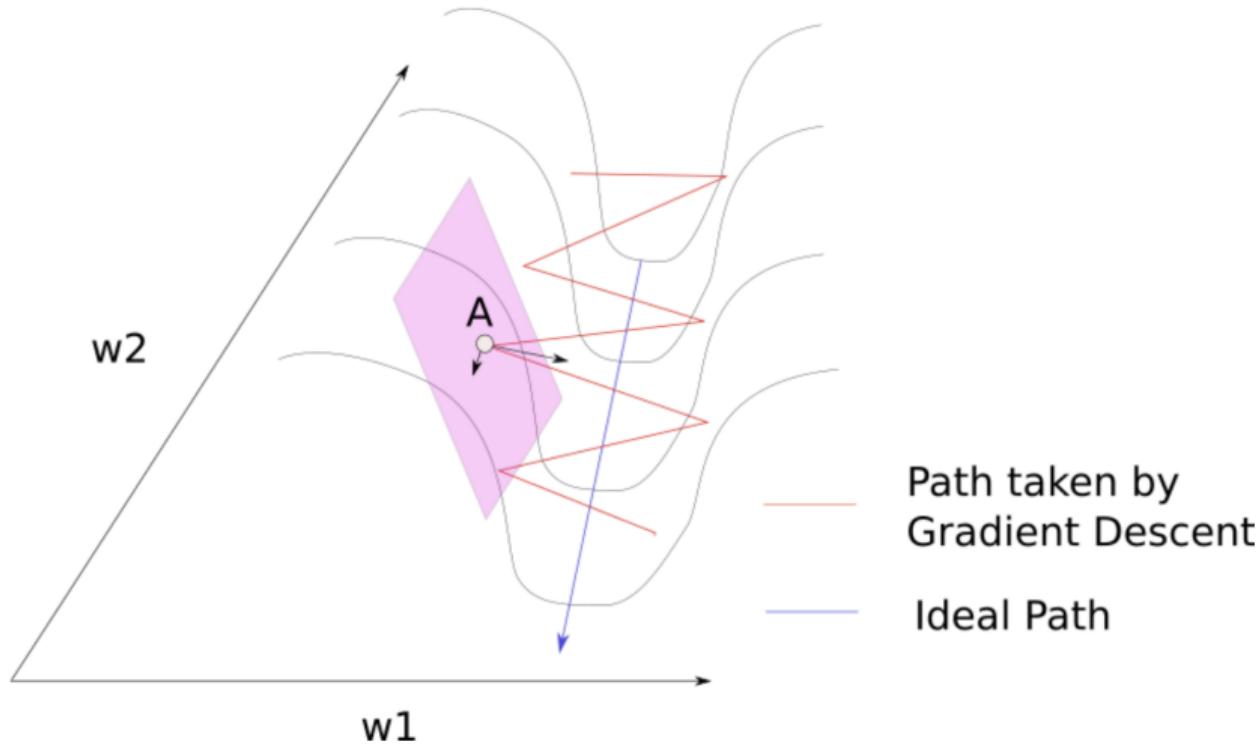


# Gradiente estocástico: Exponentially Weighted Moving Average [5]



- Usar gradiente estocástico minimiza la probabilidad de encontrar mínimos locales.
- Sin embargo, las oscilaciones generadas por el ruido en los gradientes puede ralentizar el aprendizaje.
- Una solución es utilizar los promedios de media móvil, los cuales minimizan las variaciones locales para obtener la tendencia .

# Gradiente estocástico: Momentum I



## Gradiente estocástico: Momentum III

- El momentum es una técnica común para tratar el inconveniente de las variaciones en la función de costo. Se basa en el concepto del Promedio móvil exponencialmente ponderado

$$\begin{aligned}v_j^{(t)} &= \eta v_j^{(t-1)} - \alpha \nabla_w \mathcal{L}(w) \\w_j^{(t)} &= v_j^{(t)} + w_j^{(t-1)},\end{aligned}$$

donde  $\eta$  es el factor del momentum; además,  $v_j^{(t-1)}$  almacena información de los gradientes en iteraciones anteriores, como se nota en el siguiente ejemplo,

$$\eta = 0,9, \quad v_j^{(0)} = 0, \quad \alpha = 1$$

$$v_j^{(1)} = -G_1$$

$$v_j^{(2)} = -0,9G_1 - G_2$$

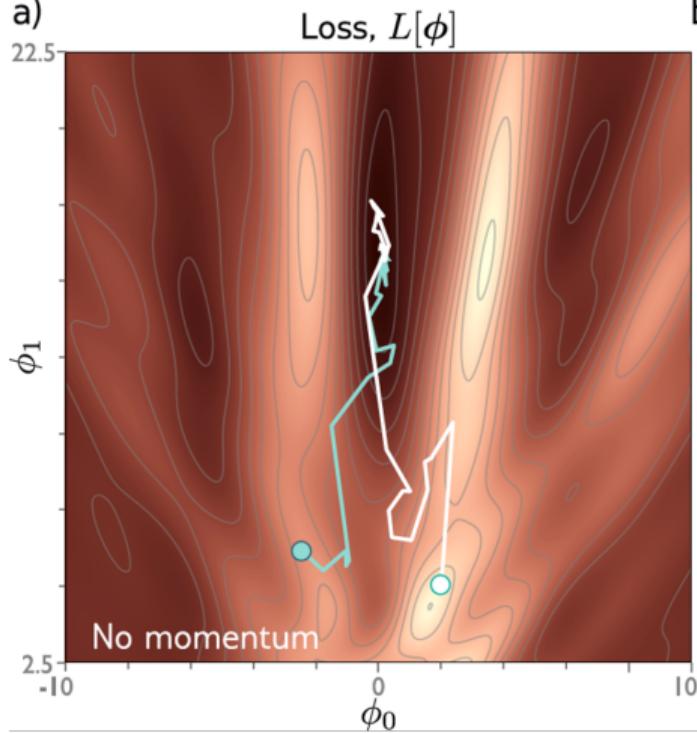
$$v_j^{(2)} = 0,9(-0,9G_1 - G_2) - G_3 = -0,81G_1 - 0,9G_2 - G_3$$



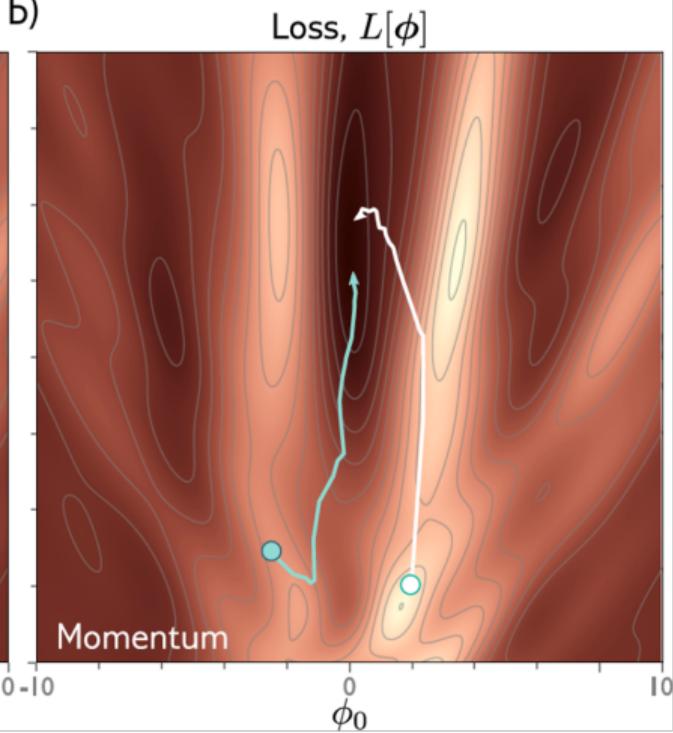
Pontificia Universidad  
JAVERIANA  
Cali

## Gradiente estocástico: Momentum IV [4]

a)



b)



## RMSProp-(Root mean square propagation) I

- Al igual que el momentum, también intenta mejorar el rendimiento del gradiente descendente.
- RMSProp ajusta la tasa de aprendizaje en cada iteración. Además, calcula un learning rate diferente para cada parámetro.

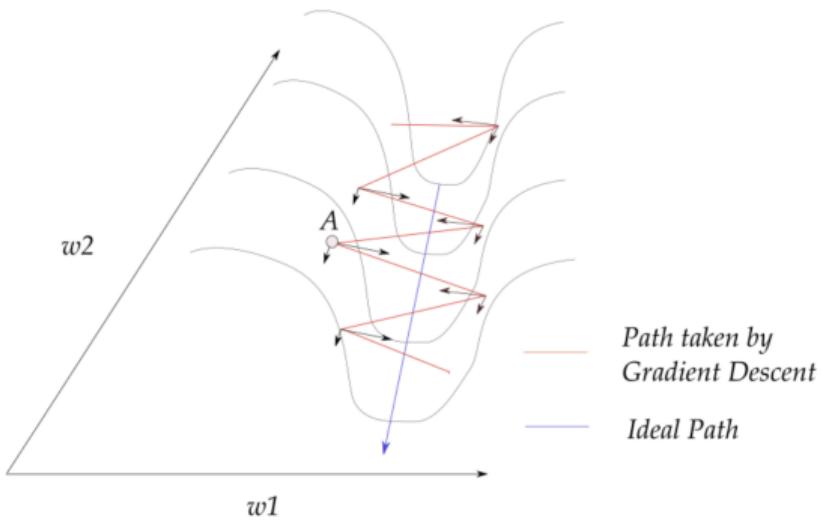
$$v_j^{(t)} = \rho v_j^{(t-1)} + (1 - \rho) (\nabla_w \mathcal{L}(w))^2 \quad \text{Promedio exponencial cuadrático}$$

$$\Delta w^{(t)} = -\frac{\eta}{\sqrt{v_j^{(t)} + \epsilon}} \nabla_w \mathcal{L}(w) \quad \text{Tasa de aprendizaje ajustable}$$

$$w_j^{(t+1)} = w_j^{(t)} + \Delta w^{(t)} \quad \text{Actualización de los parámetros,}$$

donde  $\rho$  es el parámetro para el promedio exponencial (similar al momentum).  $\eta$  valor inicial del learning rate. Similarmente,  $\epsilon$  es una cantidad cercana a cero para la estabilidad numérica.

## RMSProp-(Root mean square propagation) II



- En las ecuaciones anteriores se nota que la razón de aprendizaje depende del promedio exponencial de los gradientes cuadráticos.
- En este sentido, el learning rate se ajustará a la dirección del promedio exponencial y no al gradiente en una iteración particular.
- Así, se priorizan las direcciones que conllevan al mínimo (posiblemente global) y se minimizan las oscilaciones.

## ADAM-(Adaptive Moment Optimization) I

- Como se vio previamente, El momentum intenta conducir la búsqueda en dirección del mínimo, RMSProp procura minimizar las oscilaciones. El algoritmo ADAM combina estas dos heurísticas.

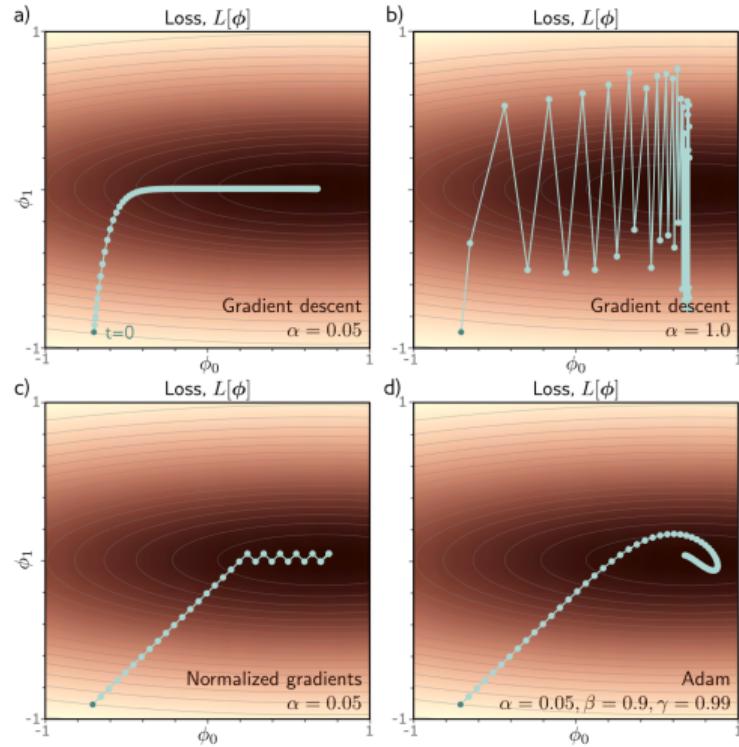
$$v_j^{(t)} = \beta_1 v_j^{(t-1)} + (1 - \beta_1) \nabla_w \mathcal{L}(w)$$

$$s_j^{(t)} = \beta_2 s_j^{(t-1)} + (1 - \beta_2) (\nabla_w \mathcal{L}(w))^2$$

$$\Delta w^{(t)} = -\eta \frac{v_j^{(t)}}{\sqrt{s_j^{(t)} + \epsilon}} \nabla_w \mathcal{L}(w)$$

$$w_j^{(t+1)} = w_j^{(t)} + \Delta w^{(t)},$$

# ADAM-(Adaptive Moment Optimization) II [4]



## Table of Contents

- ▶ Gradiente estocástico
- ▶ Para temprana (Early stopping)

# Para temprana



## Bibliografía

- [1] Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.
- [2] <https://www.analyticsvidhya.com/blog/2022/07/gradient-descent-and-its-types/>.
- [3] <https://blog.paperspace.com/intro-to-optimization-momentum-rmsprop-adam/>
- [4] <https://udlbook.github.io/udlbook/>
- [5] [https://en.wikipedia.org/wiki/Moving\\_average](https://en.wikipedia.org/wiki/Moving_average)