



Clase 9:

# Reconstrucción del espacio de fases

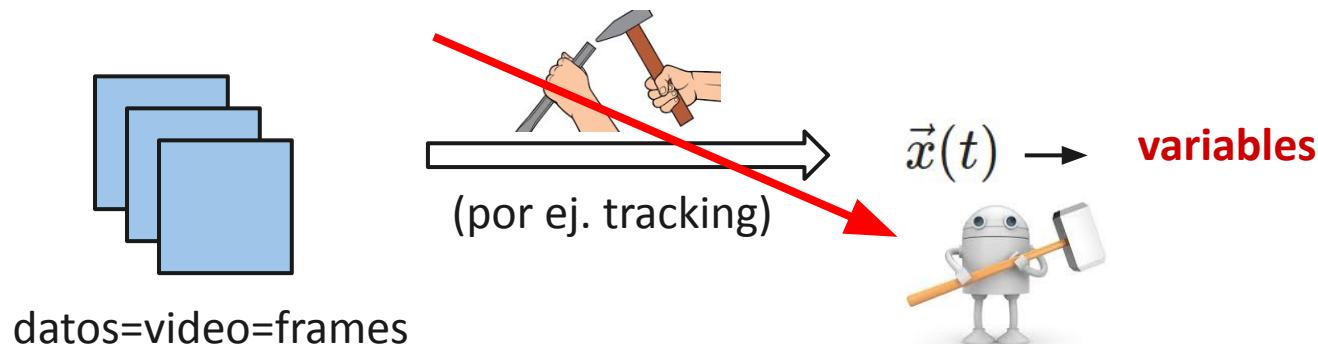


# Reconstrucción del espacio de fases

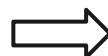
- Motivación en el marco de la materia
- Redes neuronales
  - Neurona artificial
  - Redes neuronales
  - Autoencoders
- Vamos a hacer en el Colab
- Bibliografía

## Motivación en el marco de la materia

- Si teníamos un video que refleja la dinámica

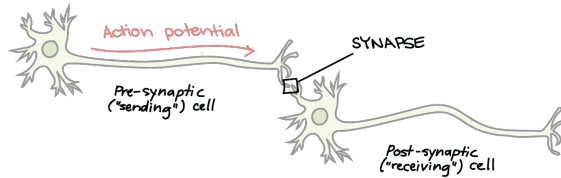


- SVD
- Autoencoders



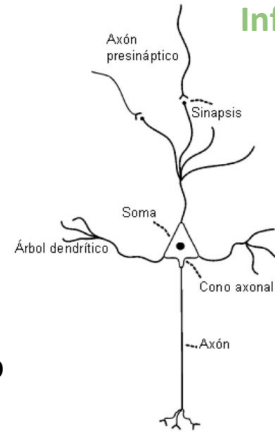
**Reconstrucción de espacio de fases**

## Neurona artificial

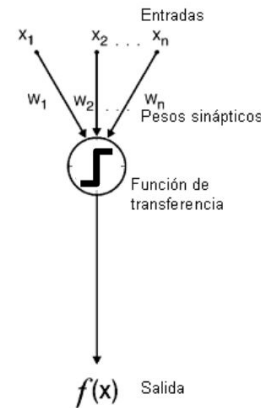


Inspiración biológica  
Neurona y sinapsis

Una neurona biológica recibe estímulos de otras neuronas  
Si los estímulos superan un umbral, la neurona dispara  
Transmite información a otra neurona, que recibe el estímulo



Información

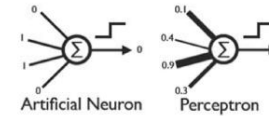


Una neurona artificial recibe entradas pesadas  
Si la suma de ellas supera un umbral, la neurona se activa  
Transmite información a otra neurona, como entrada de ella



Pitts McCulloch Rosenblatt

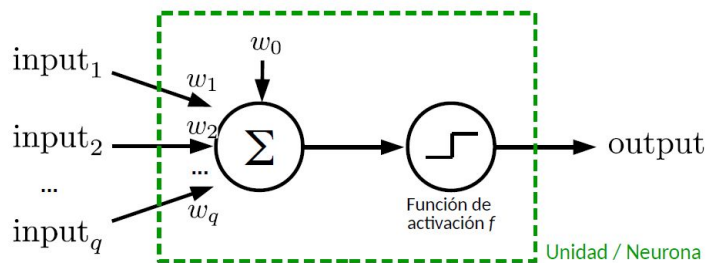
1943 1960



Perceptrón simple

# Neurona artificial

## Perceptrón simple



Las entradas son valores (números)

Se multiplican por sus respectivos pesos ( $w_j$ )

Se agrega una entrada independiente (bias o  $w_0$ )

Se suman todas las entradas pesadas y el bias

La salida es función no lineal de esa suma (número)

Es un modelo lineal generalizado

Activación  $\leftarrow$  Entradas

$$y = f\left(\sum_j x_j w_j\right) = f(x^T \cdot w)$$

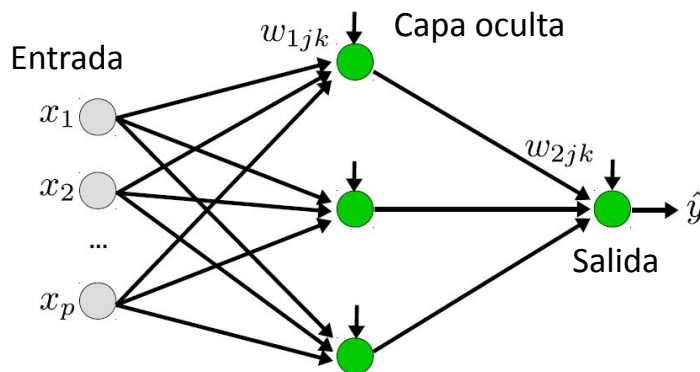
Salida  $\leftarrow$  Pesos

Se puede resolver con  
álgebra de matrices

$$f\left(\underset{1 \times q}{\text{Input}} \cdot \underset{q \times 1}{W}\right) = \underset{1 \times 1}{\text{Output}}$$

# Redes neuronales

## Perceptrón Multi-Capa



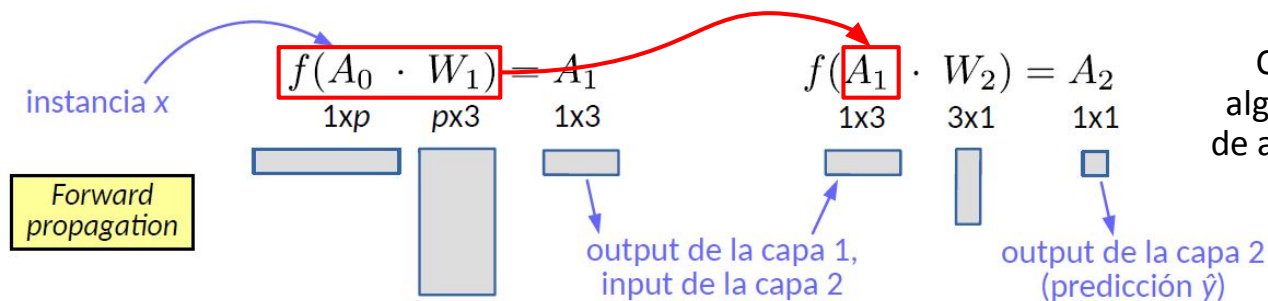
Las entradas son valores (números)

Los pesos de las conexiones ( $w_j$ ) son los parámetros

Se agrega una capa oculta con neuronas

Cada neurona es función no lineal de la suma del bias y las activaciones anteriores pesadas (número)

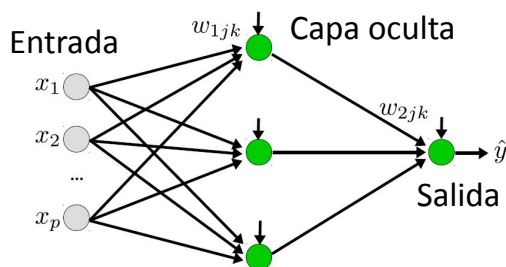
Cada salida es función no lineal de activaciones pesadas de neuronas de la capa anterior (número)



Cadena de problemas algebraicos con funciones de activación no lineal en el medio (anidados)

# Redes neuronales

Perceptrón Multi-Capa = **redes feed-forward**



**Feed-forward** porque la información **fluye hacia adelante**

Se las llama **fully-connected** cuando cada neurona se **conecta** con **todas** las neuronas de la siguiente capa

Si tiene **más de una capa oculta** se habla de **redes profundas**

La salida es una función compleja de la entrada

$f: \mathbb{R}^n \rightarrow \mathbb{R}^p$  Función global de n entradas y p salidas

$f(x) = g \circ f_K \circ \dots \circ f_2 \circ f_1(x)$  Composición de funciones

$f_i(x) = a(w_i x + b_i)$  Función de cada una neurona

**Cómo ajusto los pesos?**

**Back-propagation**

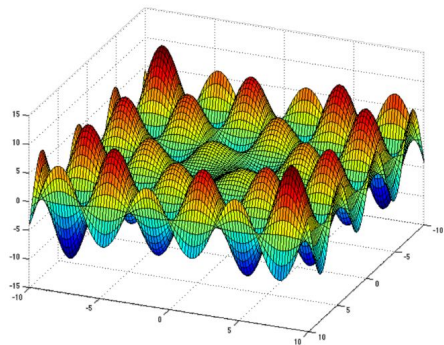
Una buena opción es usar el **descenso por el gradiente**, pero...

La **relación** de la salida con el gradiente de todos los parámetros es **complicada**

Aunque, si considero **únicamente la capa anterior**, es **fácil** estimar el gradiente

Estimar el gradiente de adelante hacia atrás, de a una capa a la vez

# Redes neuronales



Perceptrón multi-capas  
Función costo no convexa

The loss surfaces of multilayer networks

[A Choromanska](#), [M Henaff](#), [M Mathieu](#)... - Artificial intelligence ..., 2015 - proceedings.mlr.press

Para **redes neuronales grandes**, la mayoría de los **mínimos locales son similares** y presentan performance similar en los dataset de test

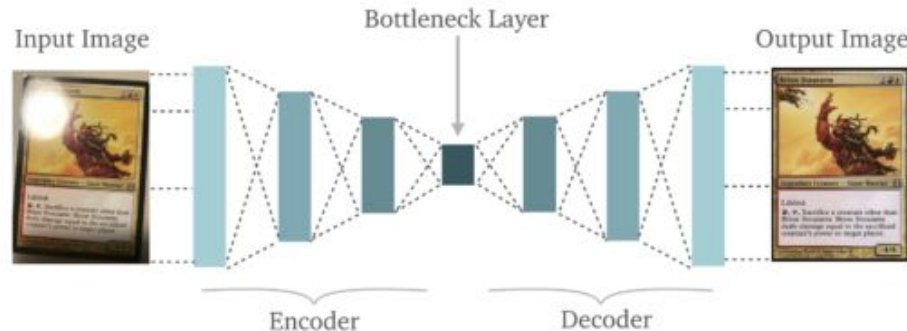
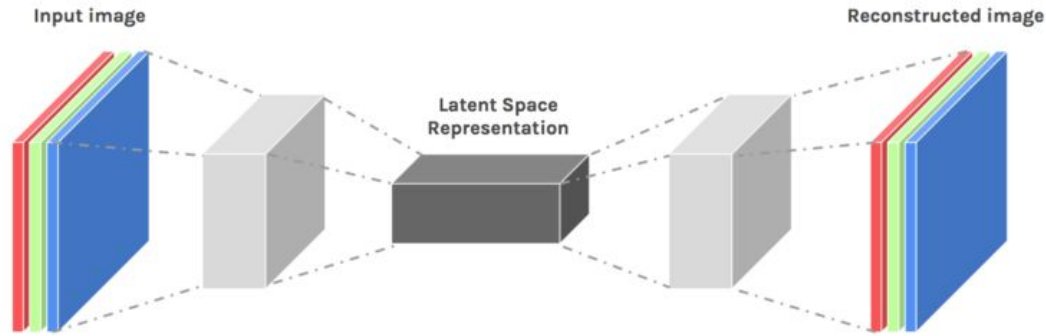
La **probabilidad** de encontrar **mínimos locales 'malos'** **decrece** con el tamaño de la red

**Focalizarse demasiado** en la búsqueda del **mínimo global** en el dataset de entrenamiento **no es útil** en la práctica, y **puede terminar en sobreajuste** del modelo

Esto lleva a que tengamos **distintas redes** (distintos pesos) pero que me resuelven bien el problema (predicción y generalización)



# Autoencoders



La red tiene una parte que codifica (encoder) la entrada y otra que decodifica (decoder)

Contiene la información suficiente de los datos

Se obtiene activación en una capa oculta de baja dimensionalidad que se llama representación en el espacio latente

Minimiza la distancia entre el input y la imagen reconstruida (auto-supervisado)

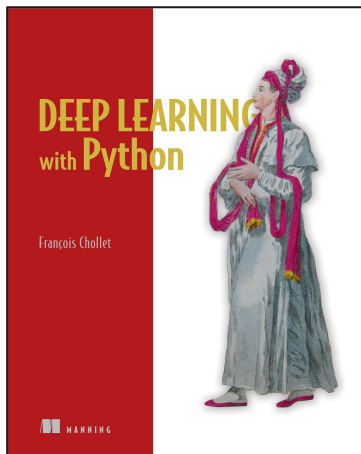
## Vamos a hacer en el Colab



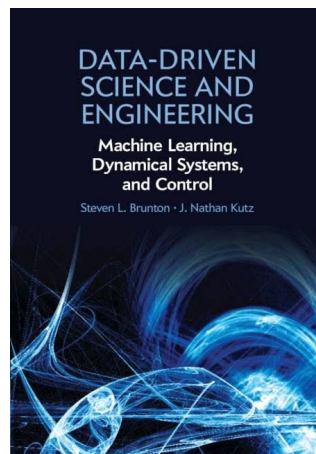
- Reconstrucción del espacio de fases
  - SVD
    - GIF Elemans
  - Autoencoders
    - GIF Elemans
    - Oscilador amortiguado (video)
    - + videos

## Bibliografía recomendada

---



Chollet 2017



Brunton & Kutz 2019

Google

 stackoverflow

towards  
data science

 YouTube