

ACÀMICA

¡Bienvenidas/os a Data Science!



Agenda

¿Cómo anduvieron?

Explicación: Puesta en Producción

Hands-on training

Break

Actividad: Neural Style Transfer

Explicación: Neural Style Transfer

Cierre



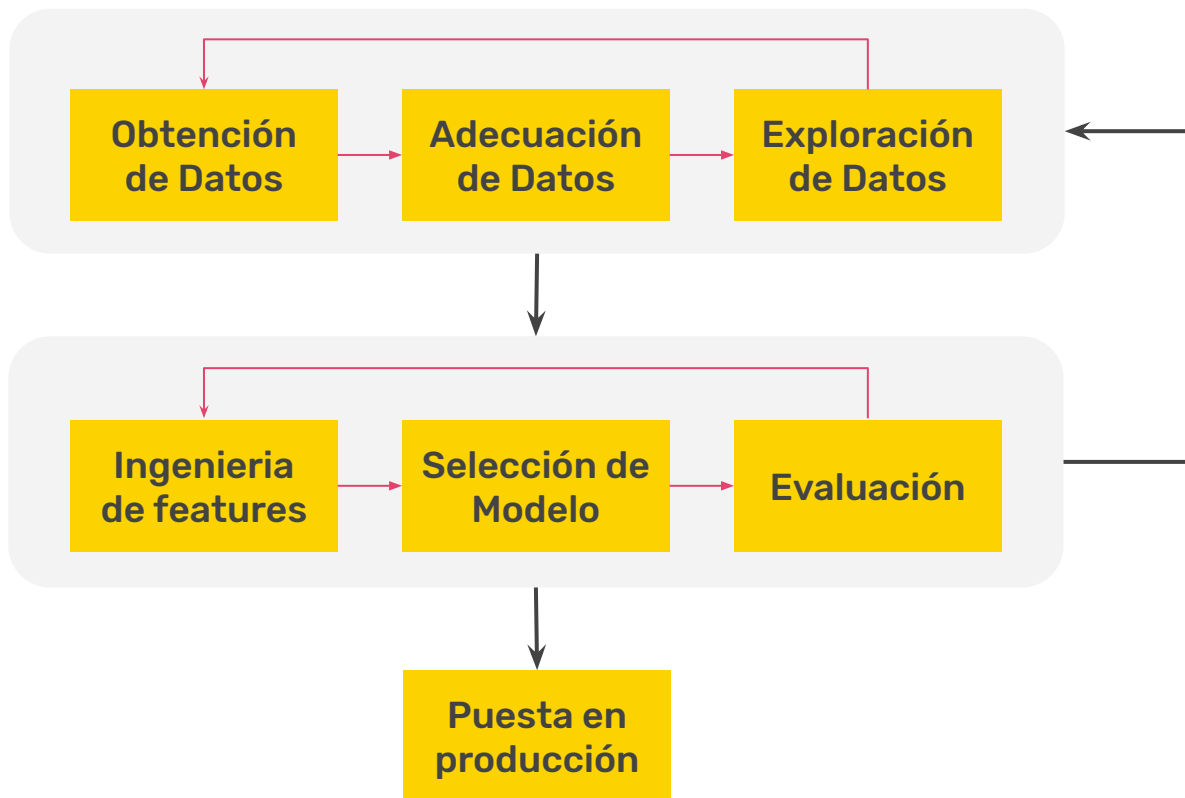
¿Cómo anduvieron?



Puesta en producción



Recorrido completo



¿Por qué Puesta en producción?



Queremos que el modelo que nosotros creamos pueda ser usado por otros usuarios. Con este fin vamos a tener que tener en cuenta **tres aspectos principales** que pueden resultar problemáticos:

- Acceso
- Compatibilidad (Lenguajes, Hardware, Librerías, etc.)
- Escala

SOLUCIÓN 1

Súper usuarios

Todos tienen los modelos y
los recursos para correrlos



SOLUCIÓN 1

Súper usuarios

Todos tienen los modelos y los recursos para correrlos



SOLUCIÓN 2

Division of labor!

Usuario
(Cliente)

Sólo mando
consultas

Servidor

Los modelos
corren acá



SOLUCIÓN 2

Division of labor!

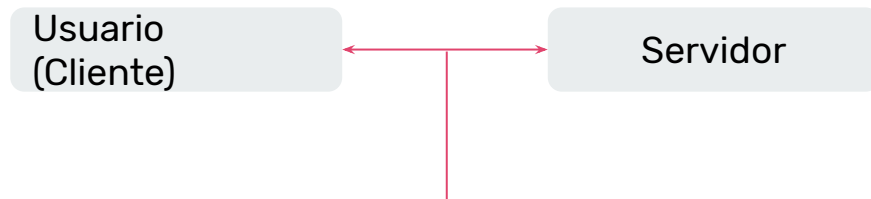


Para el servidor **existen dos posibilidades** dependiendo del uso que le queramos dar:

- Servidor Local (en una empresa o institución) - Red local o intranet
- Nube - Internet

SOLUCIÓN 2

Division of labor!



API:

Application programming interface

Es una librería con una serie de funciones que nos permiten comunicarnos con el servidor

API de Watson: IBM cloud

IBM Watson • ¿Qué es?



Watson es una inteligencia Artificial multipropósito desarrollada y mantenida por IBM. Está especialmente diseñada para lidiar bien con el lenguaje natural y tiene varias funcionalidades. Consta de una serie de modelos muy complejos (poco bias) entrenados con bases de datos muy grandes.

IBM Watson • ¿Qué es?



Watson es una inteligencia Artificial multipropósito desarrollada y mantenida por IBM. Está especialmente diseñada para lidiar bien con el lenguaje natural y tiene varias funcionalidades. Consta de una serie de modelos muy complejos (poco bias) entrenados con bases de datos muy grandes.

¡Se hizo famosa ganando al Jeopardy!

IBM Watson • ¿Qué es?



En concreto, cuando decimos Watson nos referimos a una serie de modelos muy complejos (poco bias), entrenados (o pre-entrenados) con bases de datos muy grandes que se encuentra disponible en IBM Cloud.

Otras alternativas son:

- Google Cloud Platform
- Microsoft Azure

IBM Watson • ¿Cómo usarlo?



Vamos a seguir los pasos detallados [en la plataforma de Acámica](#), y crearnos un usuario de IBM Cloud.

Debemos tener a mano [la documentación](#) para saber cómo conectarnos y usar las funciones disponibles.

Sentiment Analysis



Sentiment Analysis

Vamos a usar Watson para determinar si el contenido de un cuerpo de texto es positivo o negativo.

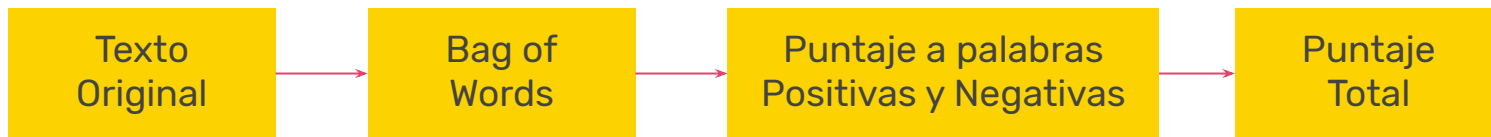


¿Cómo hacer Sentiment Analysis?



Sentiment Analysis • ¿Cómo hacerlo?

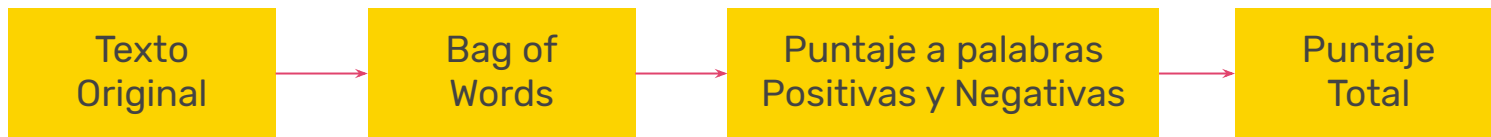
Está muy relacionado a lo que vimos de NLP. Una primera aproximación podría ser:



A pesar de ser muy simple, esto funciona (relativamente bien).

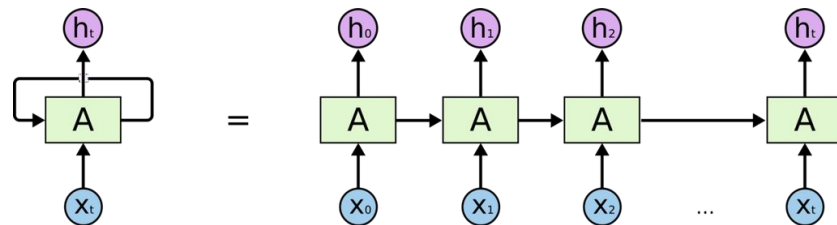
Sentiment Analysis • ¿Cómo hacerlo?

Está muy relacionado a lo que vimos de NLP. Una primera aproximación podría ser:



A pesar de ser muy simple, esto funciona (relativamente bien).

En el caso de Watson, se utilizan modelos más complejos de **Redes Recurrentes**. Esto permite tener en cuenta el orden de las palabras (estructura del texto) y no solo su aparición.



A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup is placed on a matching white saucer. In the background, a white napkin and a silver spoon are visible, though they are out of focus. The overall lighting is soft and even, highlighting the textures of the coffee and the smooth surface of the cup.

¡BREAK!

Ph. Credit: Drew Coffmann



Hands-on training



Hands-on training

DS_Encuentro_46_Watson.ipynb



Actividad: Neural Style Transfer



Neural Style Transfer

A



Neural Style Transfer

A



B



Neural Style Transfer



Hoy nos inspiramos con...

- [Andrew NG Deep Learning Course 4](#)
- “A Neural Algorithm of Artistic Style”:
<https://arxiv.org/abs/1508.06576>
- https://keras.io/examples/neural_style_transfer/



DS_Encuentro_46_style_transfer.ipynb

—

**Resultados
NO garantizados**



Neural Style Transfer



Primer ingrediente: redes convolucionales

¿Cómo funciona?

Primer ingrediente: redes convolucionales

1 <small>x1</small>	1 <small>x0</small>	1 <small>x1</small>	0	0
0 <small>x0</small>	1 <small>x1</small>	1 <small>x0</small>	1	0
0 <small>x1</small>	0 <small>x0</small>	1 <small>x1</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

¿Cómo funciona?



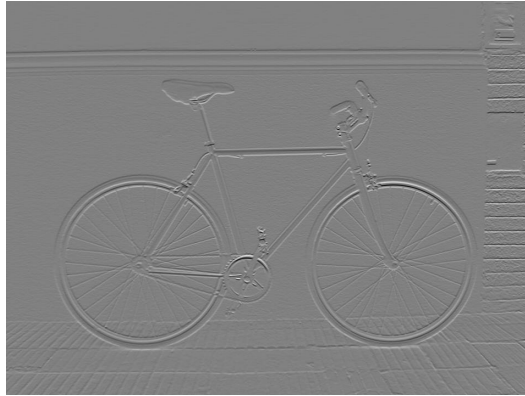
¿Cómo funciona?



1	1	1
0	0	0
-1	-1	-1

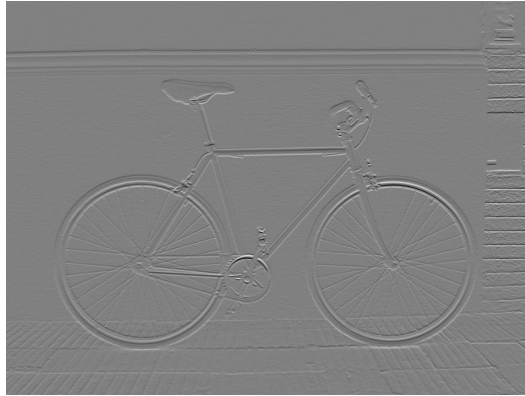
1	0	-1
1	0	-1
1	0	-1

¿Cómo funciona?



1	1	1
0	0	0
-1	-1	-1

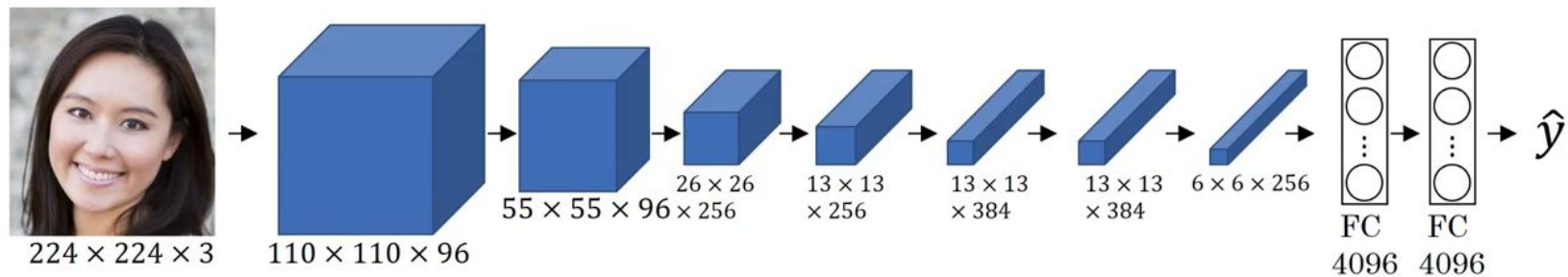
¿Cómo funciona?



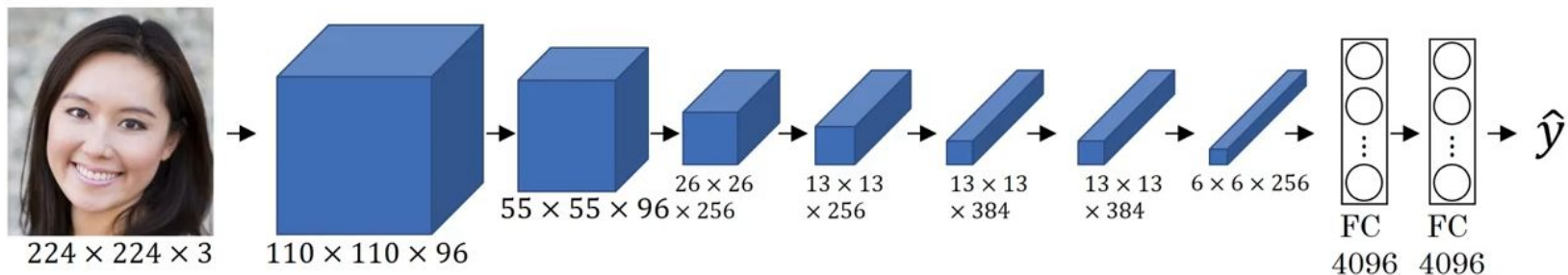
1	1	1
0	0	0
-1	-1	-1

1	0	-1
1	0	-1
1	0	-1

¿Cómo funciona?



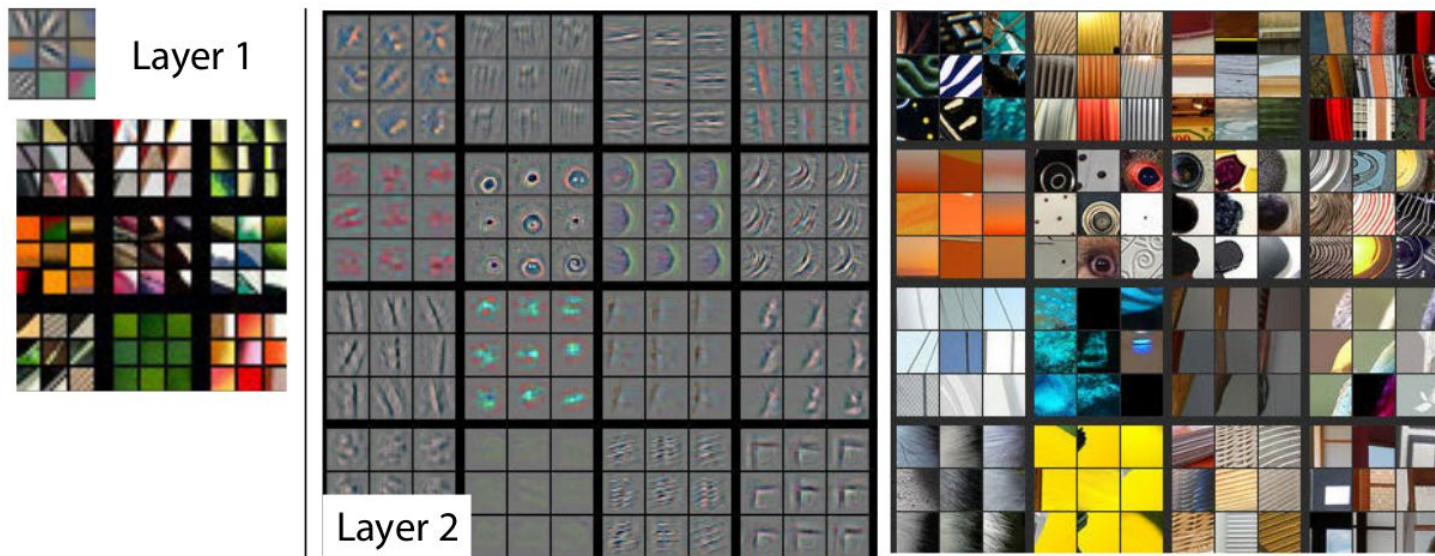
¿Cómo funciona?



- Cada capa de filtros funciona como una etapa de extracción de *features*
- Las últimas capas (Fully connected - Redes Neuronales *convencionales*) clasifican según la existencia o no de algunas características.

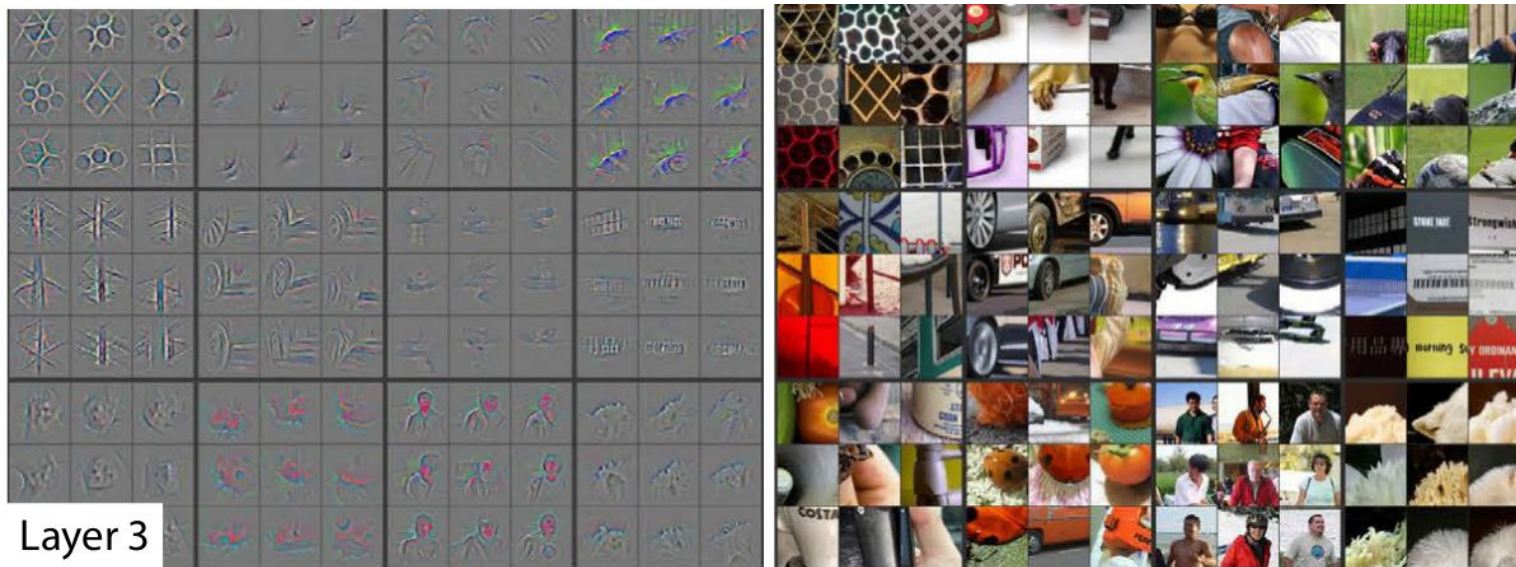
¿Cómo funciona?

Primer ingrediente: redes convolucionales



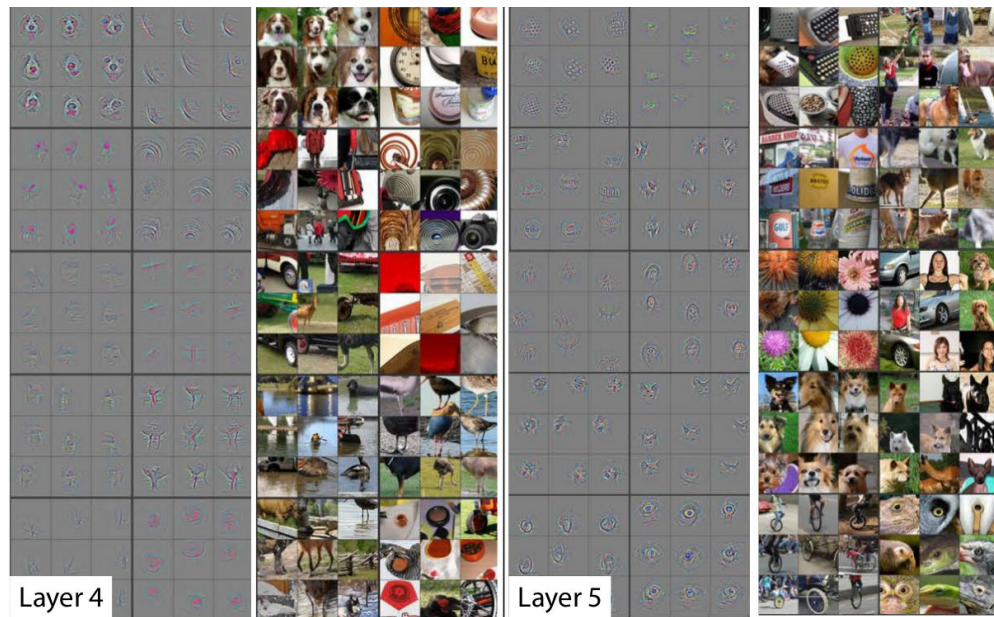
¿Cómo funciona?

Primer ingrediente: redes convolucionales



¿Cómo funciona?

Primer ingrediente: redes convolucionales



Segundo ingrediente: función de costo

¿Cómo funciona?

Segundo ingrediente: función de costo

Contenido **C**



Estilo **S**



¿Cómo funciona?

Segundo ingrediente: función de costo

Contenido **C**



Estilo **S**

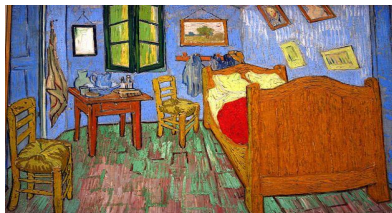


Imagen generada **G**



¿Cómo funciona?

Segundo ingrediente: función de costo

$$J(G) = \alpha * J_{\text{content}}(C, G) + \beta * J_{\text{style}}(S, G)$$

¿Cómo funciona?

Segundo ingrediente: función de costo

$$J(G) = \alpha * J_{\text{content}}(C, G) + \beta * J_{\text{style}}(S, G)$$

Costo
total

Costo asociado al
contenido

Costo asociado al
estilo

¿Cómo funciona?

Segundo ingrediente: función de costo

$$\mathcal{J}(G) = \alpha * \mathcal{J}_{\text{content}}(C, G) + \beta * \mathcal{J}_{\text{style}}(S, G)$$

Costo
total

Costo asociado al
contenido

Costo asociado al
estilo

Método:

1. Inicializar G al azar
2. Usar descenso por gradiente para minimizar $\mathcal{J}(G)$

¿Cómo funciona?

Segundo ingrediente: función de costo

$$J(G) = \alpha * J_{\text{content}}(C, G) + \beta * J_{\text{style}}(S, G)$$



1. Usamos una red neuronal ya entrenada
2. Pasamos la imagen C y la imagen G por la red neuronal.
3. Seleccionamos las activaciones de una capa l , $a^{[l]}(C)$ y $a^{[l]}(G)$

¿Cómo funciona?

Segundo ingrediente: función de costo

$$J(G) = \alpha * J_{\text{content}}(C, G) + \beta * J_{\text{style}}(S, G)$$



1. Usamos una red neuronal ya entrenada
2. Pasamos la imagen C y la imagen G por la red neuronal.
3. Seleccionamos las activaciones de una capa l , $a^{[l]}(C)$ y $a^{[l]}(G)$
4. Si las activaciones son similares, asumimos que las imágenes tienen contenido similar.

$$J_{\text{content}}(C, G) = \| a^{[l]}(C) - a^{[l]}(G) \|^2$$

Para la próxima

1. Terminar de ver los videos de puesta en producción.
2. Completar el notebook de hoy de Watson.
3. Terminar la entrega 06 si aún no lo hicieron.

ACÀMICA