

## Laboratorio 3

31/05/2019

### Fundamento teórico

Desde su motivación en la biología hemos visto las bases del aprendizaje profundo mediante dos de las piezas imprescindibles en una red neuronal: el *Perceptrón simple* y la *Retropropagación*. Hoy en día las aplicaciones de esta herramienta de la inteligencia computacional están limitadas a la imaginación de quienes las utilizan y ejemplos como *Sophia*<sup>[1]</sup> nos demuestran su potencial en todo su esplendor.

Para profundizar en las posibilidades científicas que nos brindan las redes neuronales comenzaremos enumerando algunos tipos de redes que actualmente se utilizan:

- LSTM
- Autoencoder
- GAN
- Convolucionales

En esta ocasión utilizaremos las redes convoluciones para realizar un experimento de reconocimiento de imágenes en el cual entrenaremos un modelo para distinguir entre una y otra clase.

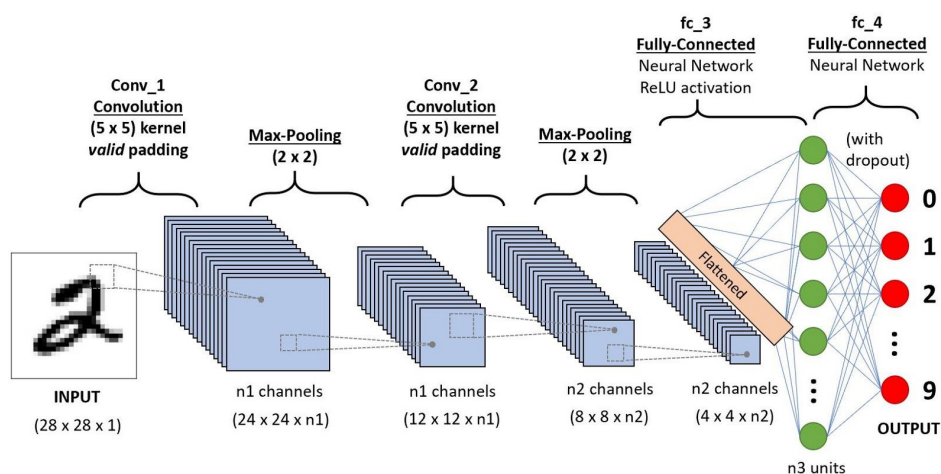
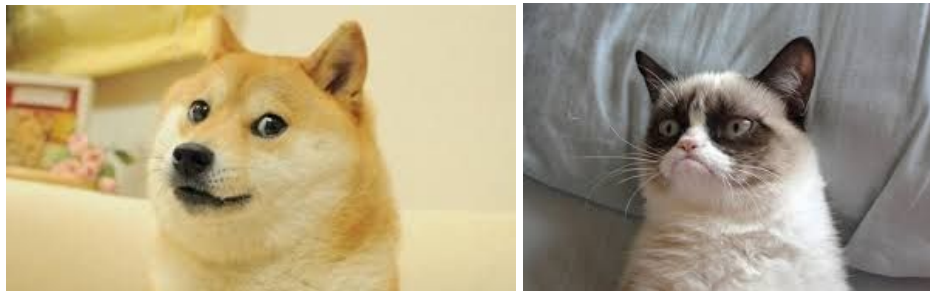


Figura 1: Arquitectura de una red neuronal convolucional.

Fuente: [www.towardsdatascience.com](http://www.towardsdatascience.com)

Una red convolucional es un tipo de red neuronal que contiene una o más capas que realizan la operación de convolución sobre una entrada para procesarla. La operación de la convolución corresponde a la sobreposición entre dos funciones en el tiempo, una de las cuales está invertida y es trasladada para contrastar su similitud con la otra función. Esta operación se utiliza en las capas convolucionales para determinar el grado de similitud que tiene una entrada con el patrón o la característica que intenta identificar la red. A su vez, las redes convolucionales pueden ayudarse de otras capas que mejoren el proceso de aprendizaje como las capas de *Rectificación Lineal (ReLU)*, *Pooling* o de *Normalización*.

Usualmente las redes neuronales son utilizadas para el reconocimiento de imágenes, un ejemplo clásico es la clasificación de imágenes de gatos y perros, caso que utilizaremos como ejemplo para la experiencia de este laboratorio.



### Actividades

1. Cargar el Dataset de imágenes de perros y gatos: [Dataset](#).
2. Implementar una red convolucional con una capa convolucional, una capa de *Max pooling*, una capa *ReLU* y una capa *Softmax*.
3. Dividir el conjunto de imágenes en conjunto de entrenamiento y de validación.
4. Entrenar la red neuronal por +-10 épocas obteniendo un modelo que logre clasificar las imágenes en perros y gatos.
5. Subir una fotografía de su mascota a la carpeta [Repositorio](#) para crear el conjunto de prueba. Debe cargar al menos la foto de un perro y de un gato, de no tener uno o ninguno, fotografiar a las mascotas de un conocido.
6. Utilizar las imágenes de sus compañeros para probar el modelo obtenido.
7. Analizar resultados y concluir al respecto.

### Entregable

Las actividades anteriores pueden ser realizadas individualmente o en parejas. Se debe entregar un archivo .ipynb en un .zip que contenga el código de las actividades mencionadas en el punto anterior.

**Fecha de entrega: 16 de junio del 2019.**

### **Consideraciones generales**

Instalar *Cv2*, *Tensorflow* y *Keras*.

Definir dimensión de las imágenes para trabajar y escalar las imágenes del *Dataset*.

Cargar los datos y darles un orden aleatorio.

Dividir los datos en la entrada y la salida esperada.

Transformar la entrada en un *Numpy Array* (Necesario para trabajar con *keras*).

Normalizar los datos ( 0 - 255).

Definir el modelo, cantidad de capas y configuración de cada una.

### **Documentación**

Tensorflow y Keras:

[Sitio oficial Tensorflow](#)

[Sitio oficial Keras](#)

[Post alternativo para instalación](#)

Configuración del modelo:

[Modelos y capas](#)

[Optimizadores](#)