

Taller Keras

Redes neuronales multicapa para procesamiento de imágenes

Mariano Rivera Meraz
Alan G. Reyes Figueroa



PI2018

XII Taller-Escuela de
Procesamiento de Imágenes

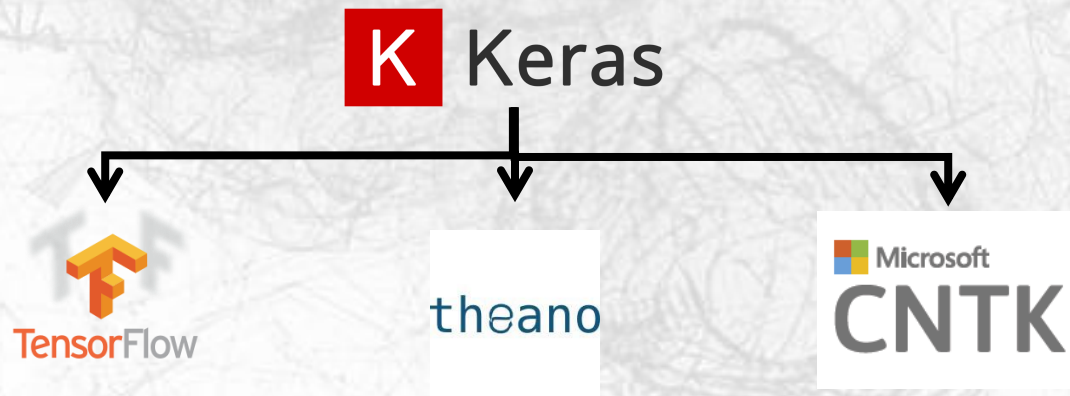
CIMAT, Guanajuato. 20 y 21 de septiembre del 2018

Keras

- Desarrollada por François Chollet (2015-2016).



- Soporte para 3 librerías de NNs



- Documentación: <https://keras.io/>

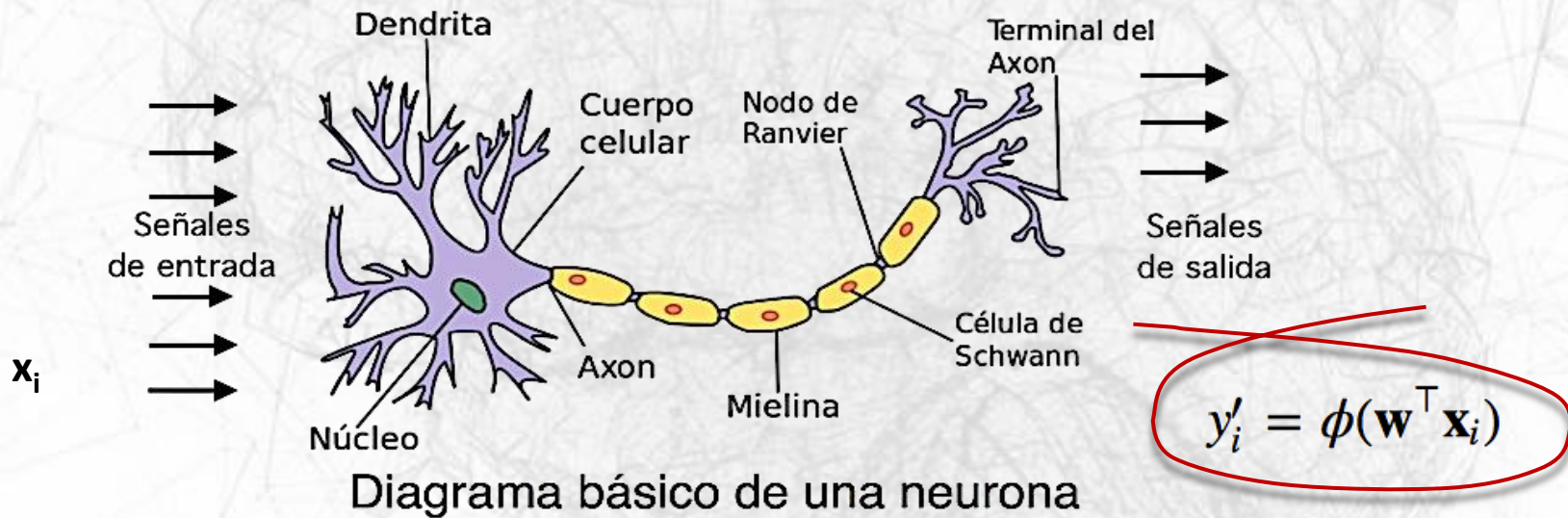
Objetivos del Taller

- Este es un taller introductorio en dos partes:
 - en la primera: vamos a implementar una red neuronal simple (y multicapa) usando Keras.
 - en la segunda: aprenderemos a implementar una red convolucional usando Keras.

El objetivo es que, para quienes nunca hayan implementado una red neuronal, sepan cómo hacerlo usando Keras.

El perceptrón

- Basado en el modelo neuronal de McCulloch-Pitts (1943)



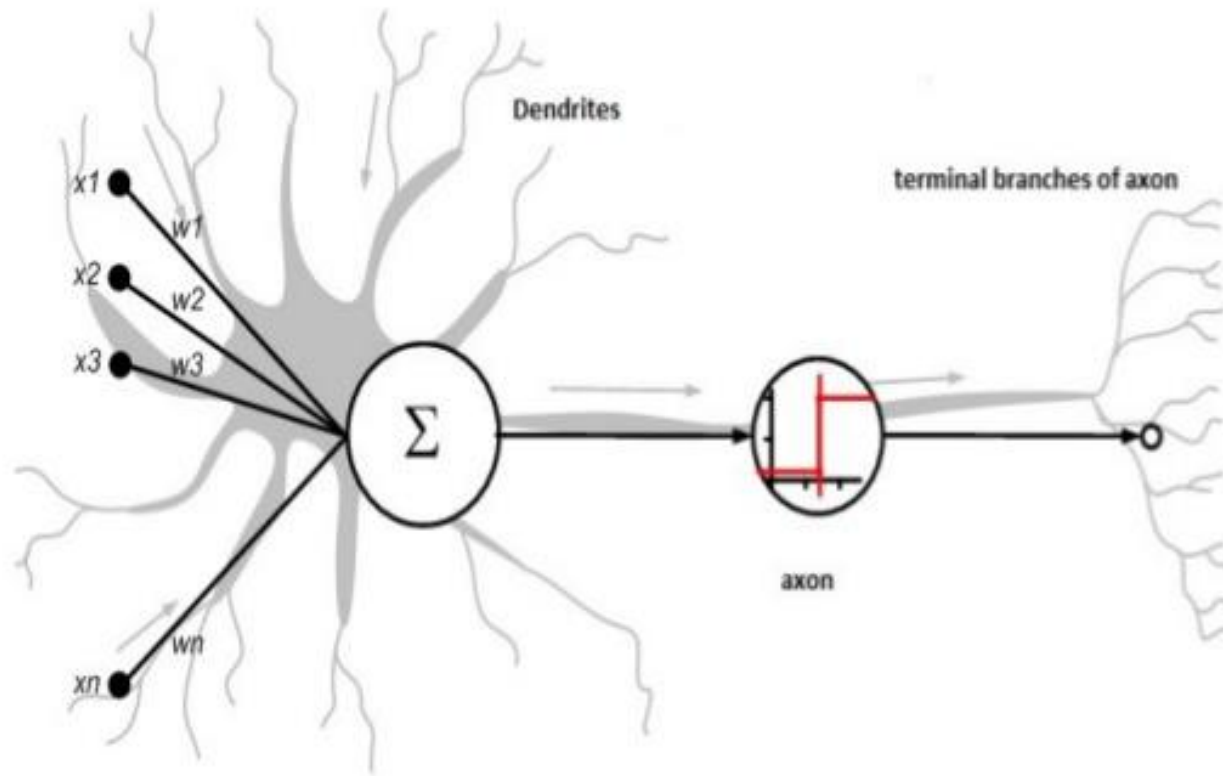
- Rosenblatt (1957) propuso un algoritmo iterativo para el aprendizaje automático de los parámetros.

W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in neuron activity", The Bulletin of the Mathematical Biophysics, 5(4) 115-133, 1943.

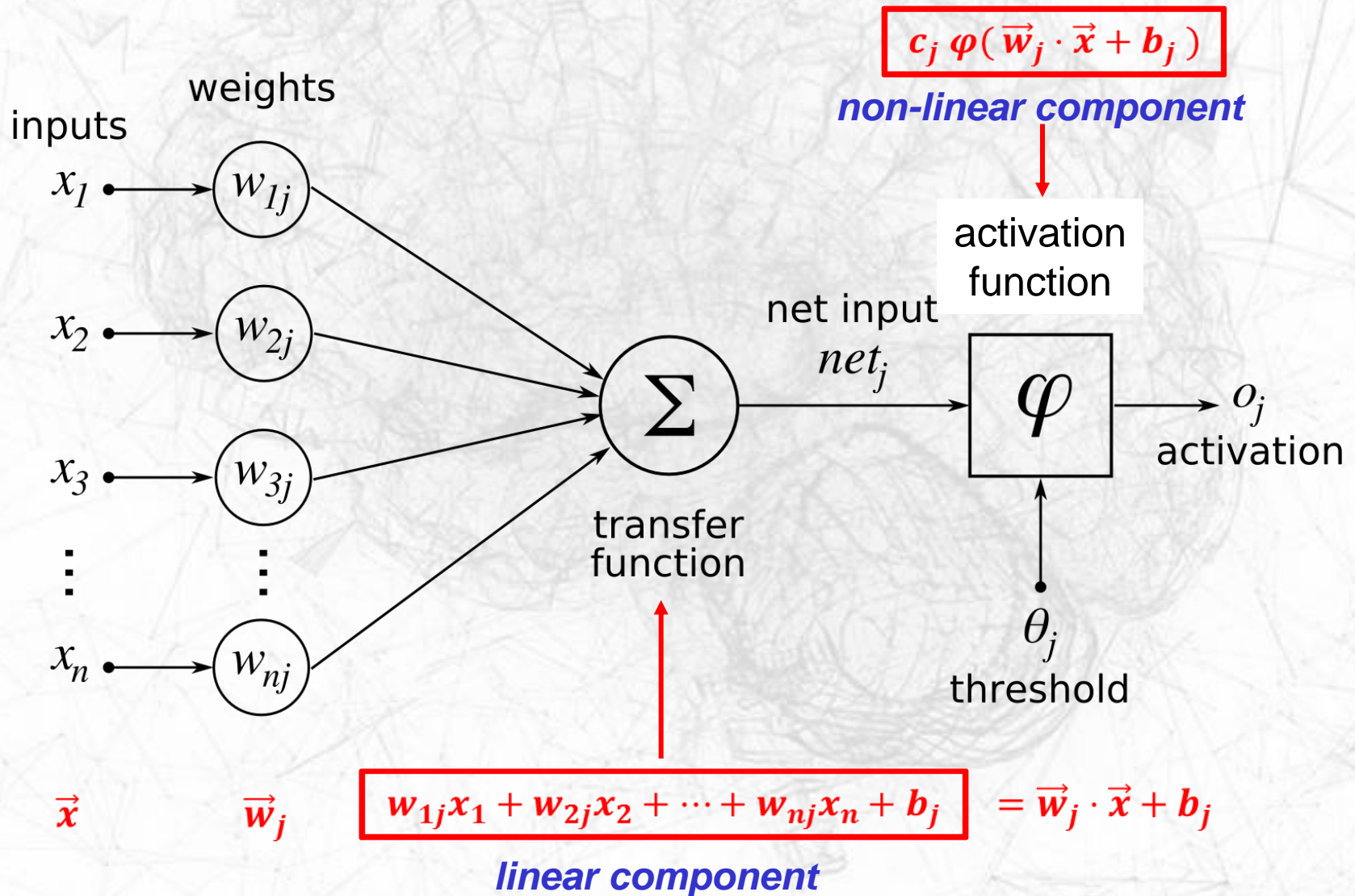
F. Rosenblatt. "The Perceptron, a perceiving and recognizing automaton", Cornell Aeronautical Lab, 1957

Redes Neuronales Artificiales

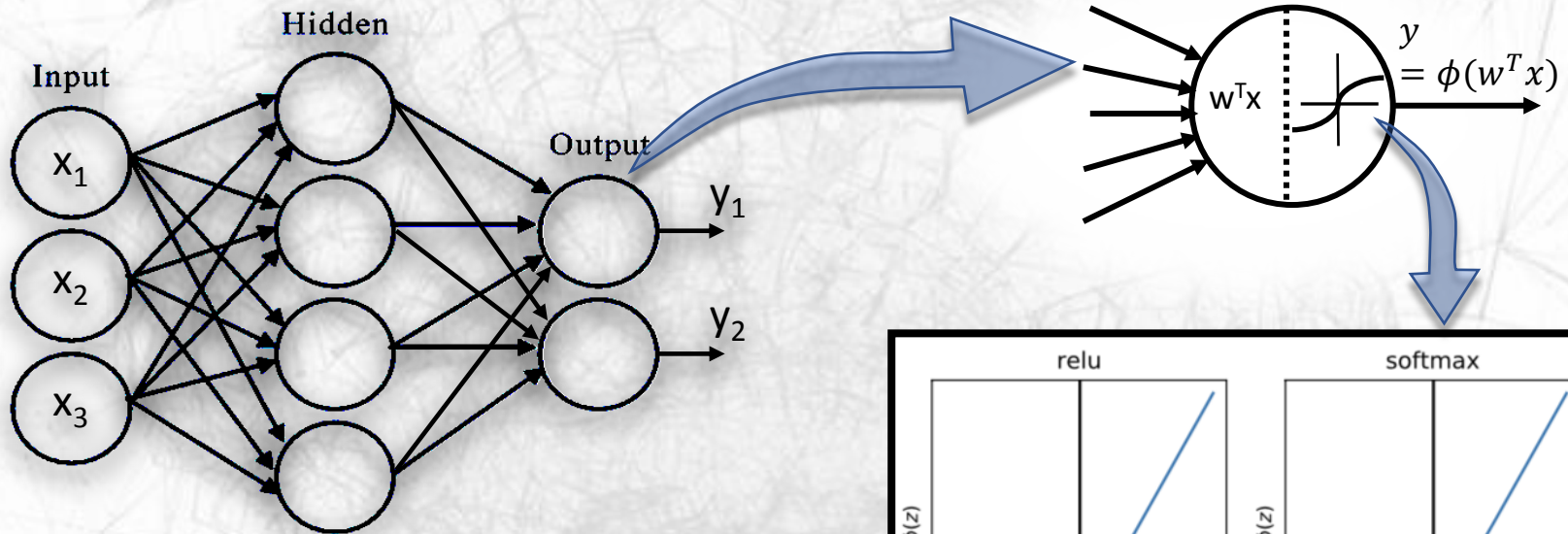
Biologically Inspired Neuron



Redes neuronales artificiales

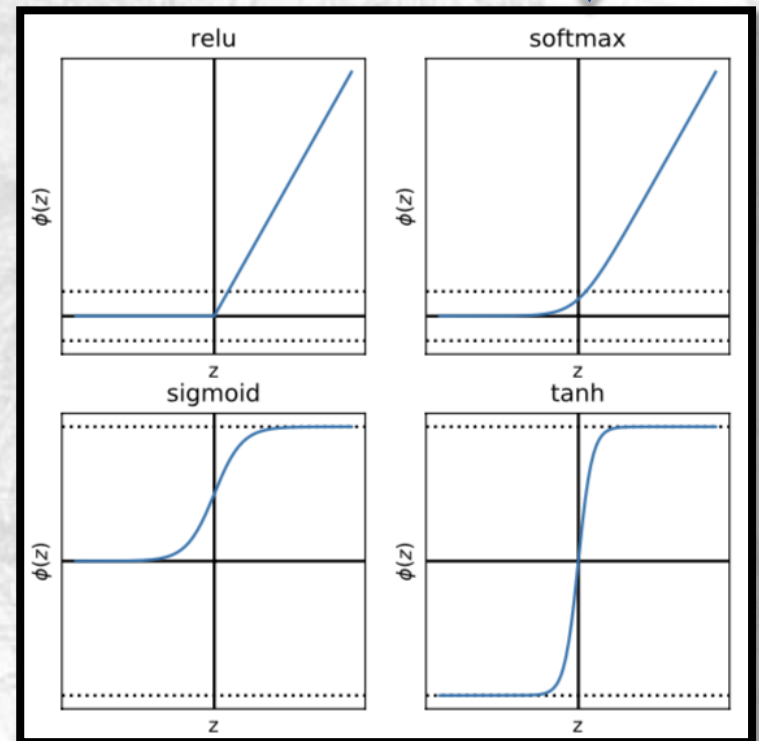


Redes neuronales multicapa



En 1986, Rumelhart, Hinton y Williams proponen el algoritmo **Backpropagation** para entrenar una red neuronal

Este algoritmo estima el **gradiente** de la función de pérdida **de forma numérica** (regla de la cadena).



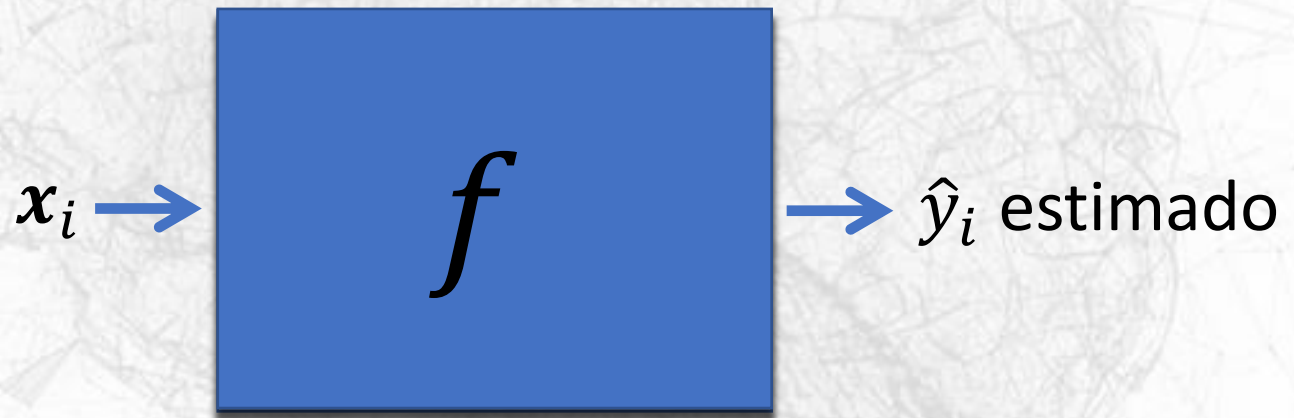
Aprendizaje con redes neuronales

Datos:

x_i : dato

y_i : resultado

(etiqueta, valor esperado, imagen filtrada)



Función de pérdida (*loss function*)

Mide la discrepancia entre la salida estimada (por la red neuronal), y la salida correcta.

x_i : dato

y_i : valor correcto

\hat{y}_i : valor estimado

Ejemplo: Regresión. $L = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$

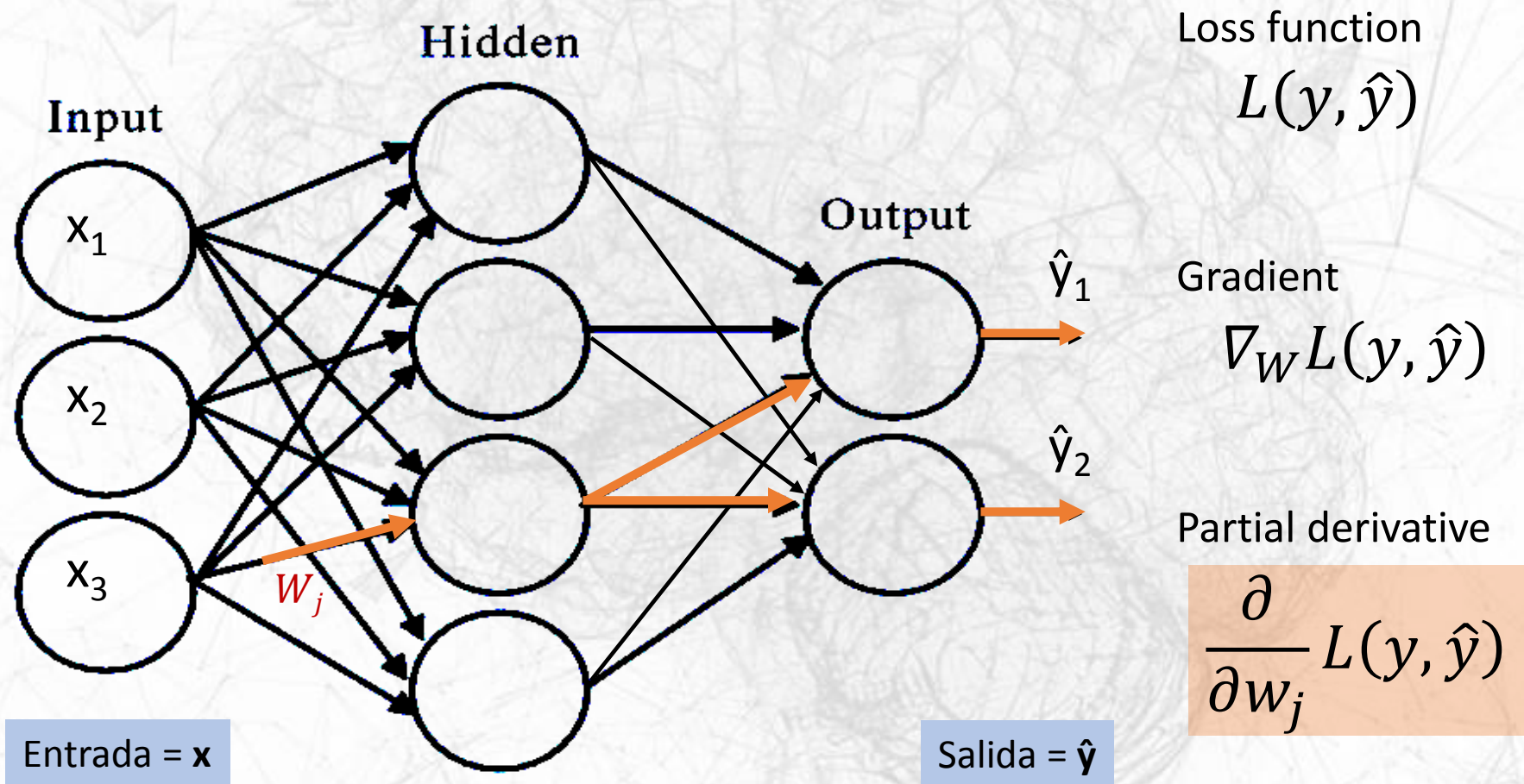
(error medio cuadrático *MSE*) (suma de cuadrados)

Ejemplo: Clasificación. $L = - \sum_{i=1}^n y_i \log \hat{y}_i$

(entropía cruzada, *cross-entropy*)

Aquí, los \hat{y}_i dependen de los pesos, $\hat{y}_i = \hat{y}_i(w_{jk})$.

Backpropagation



Backpropagation

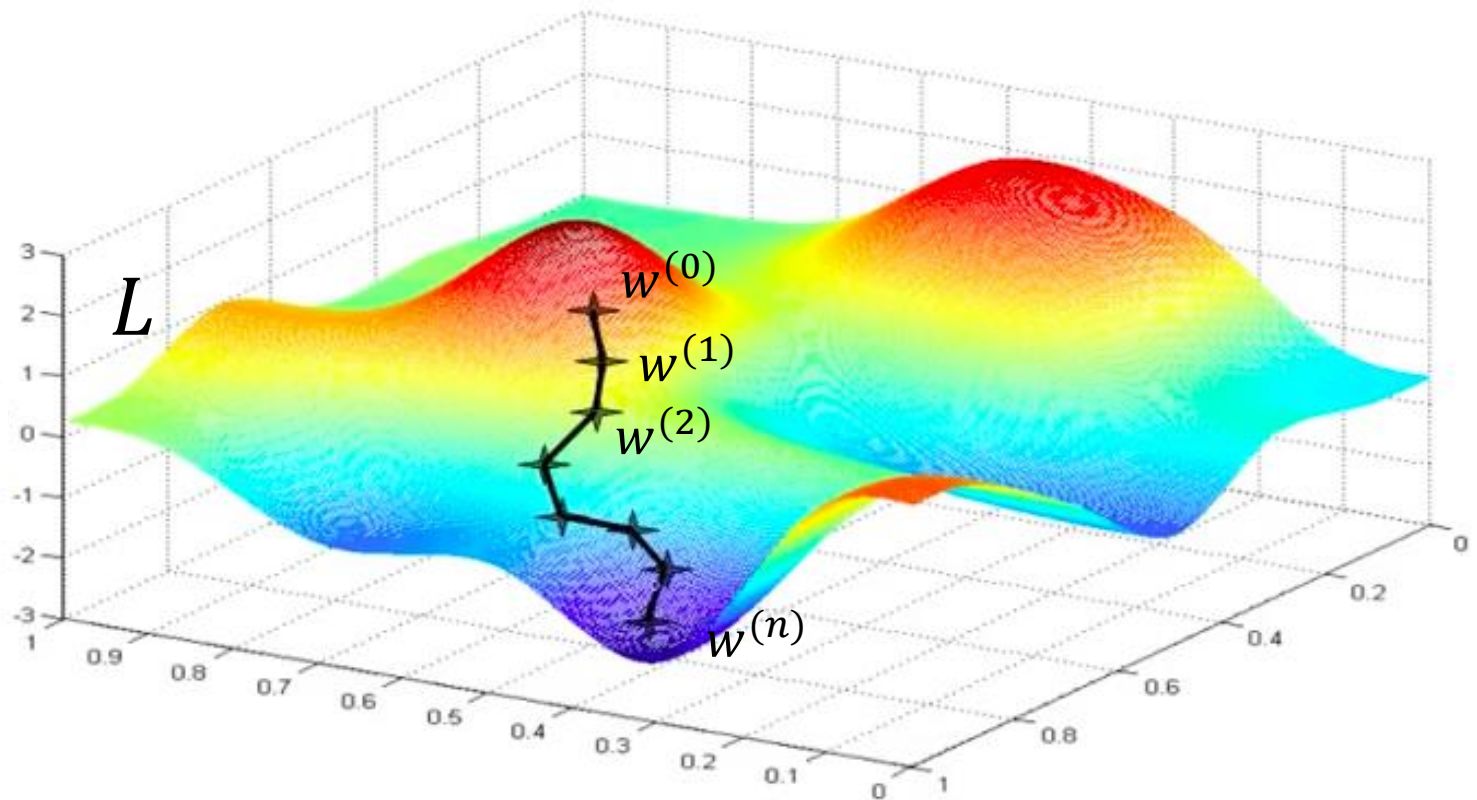
El backpropagation es un algoritmo iterativo.

El mínimo de la función de pérdida L se calcula usando algún método de descenso gradiente

$$W^{(i+1)} = W^{(i)} - \alpha \nabla_W L$$

Los pesos se actualizan en cada iteración, hasta que el algoritmo termina.

Descenso gradiente



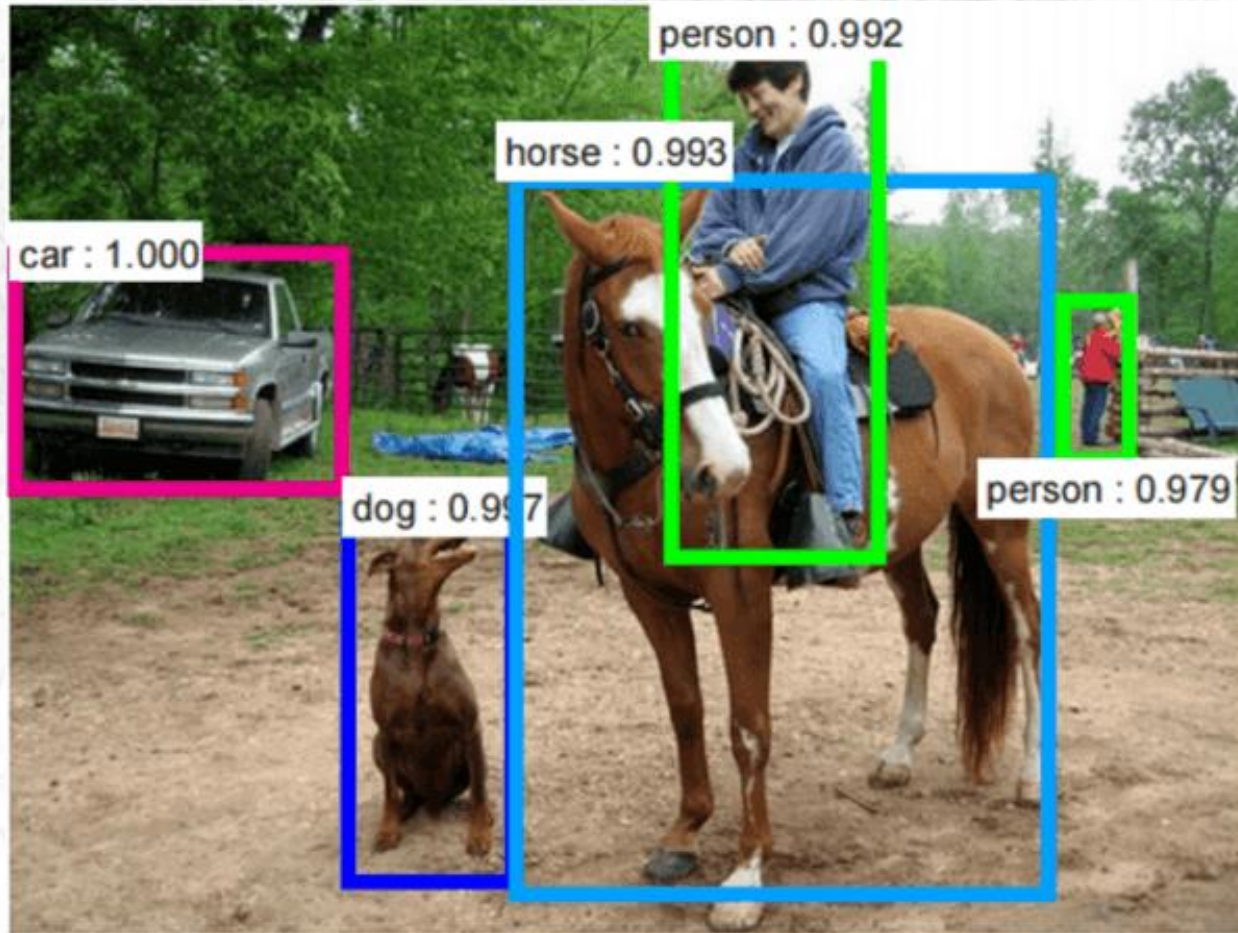
$$W^{(i+1)} = W^{(i)} - \alpha \nabla_W L$$

Aplicaciones: Segmentación de imágenes



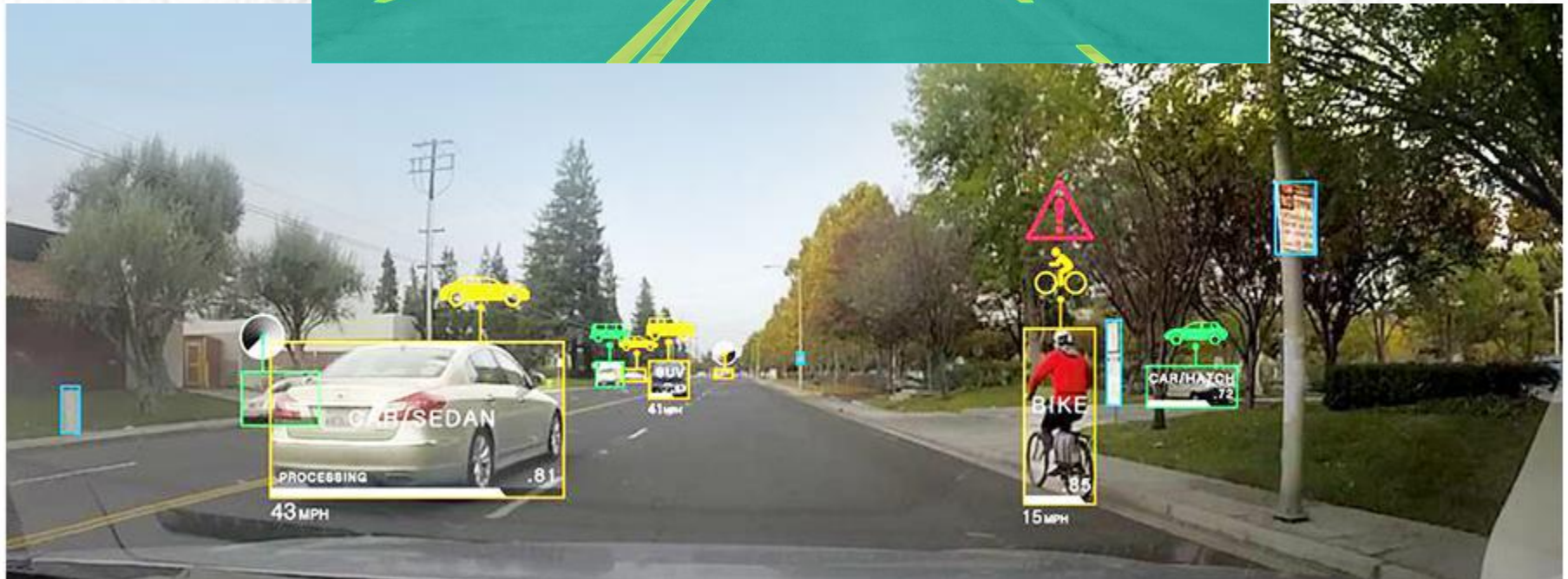
Image Segmentation

Aplicaciones: Detección y reconocimiento



Object Detection

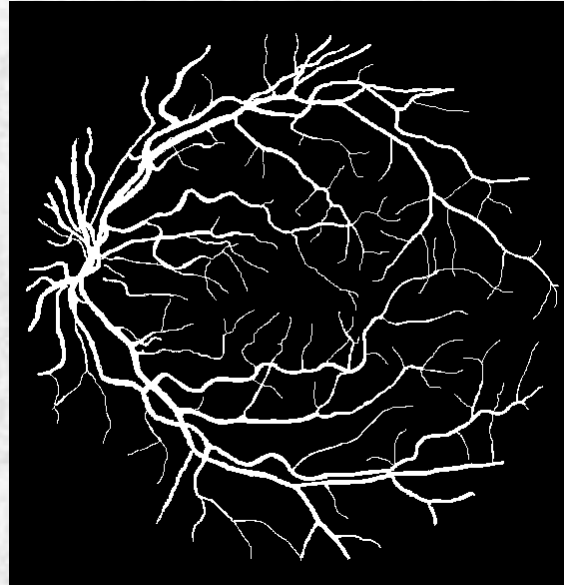
Aplicaciones: Self-Driving Cars



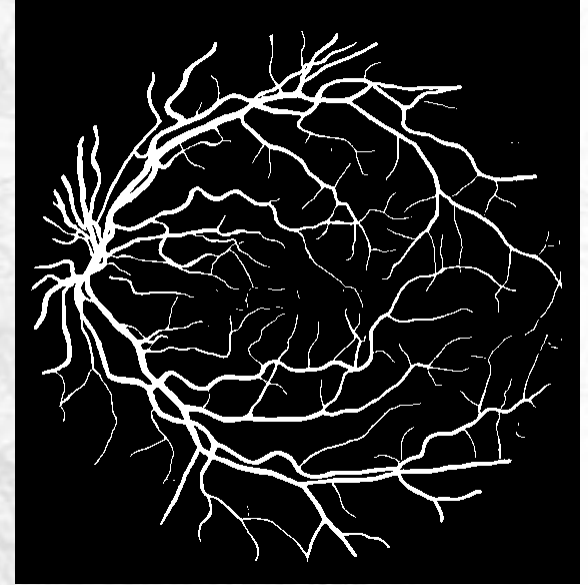
Aplicaciones: Imágenes médicas



Original



Manual Segmentation



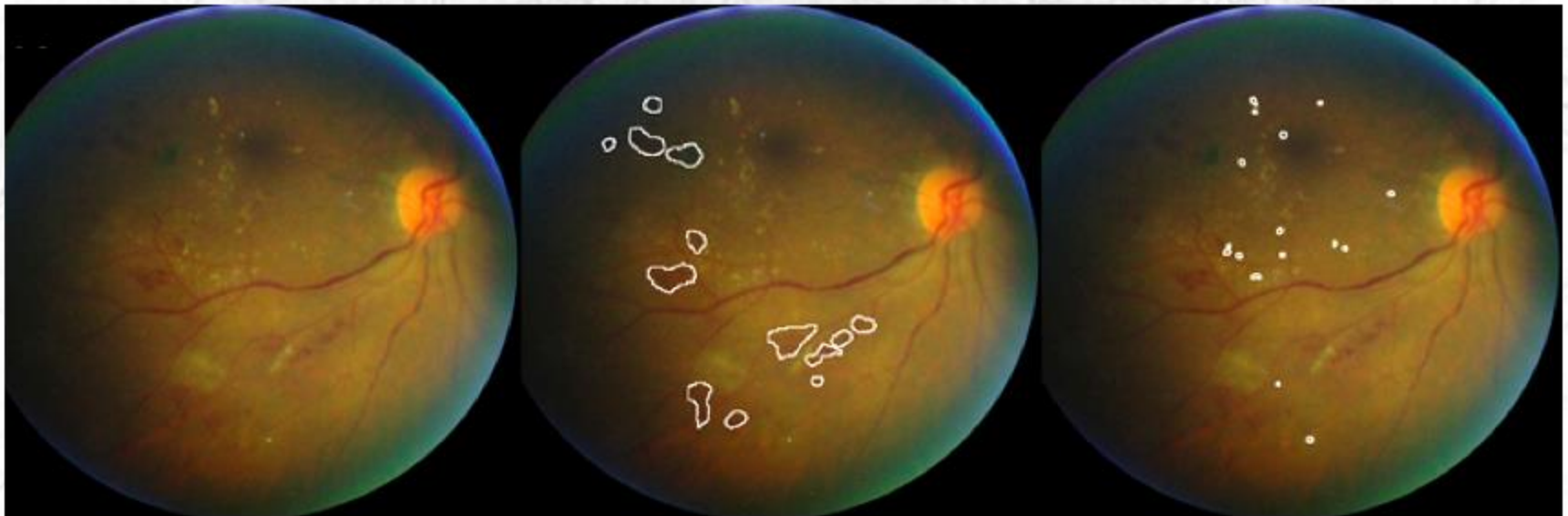
DNN Segmentation

Original retina fundus image, manual segmentation by a human expert and our segmentationc

Aplicaciones: Diagnóstico automatizado

Lesion localization

Robust lesion localization achieved using multi-scale pixel and lesion-level descriptors



Original Image

Red lesion detected

Bright lesions detected

Aplicaciones: diseño de Filtros



Aplicaciones: Super-resolución de una sola imagen

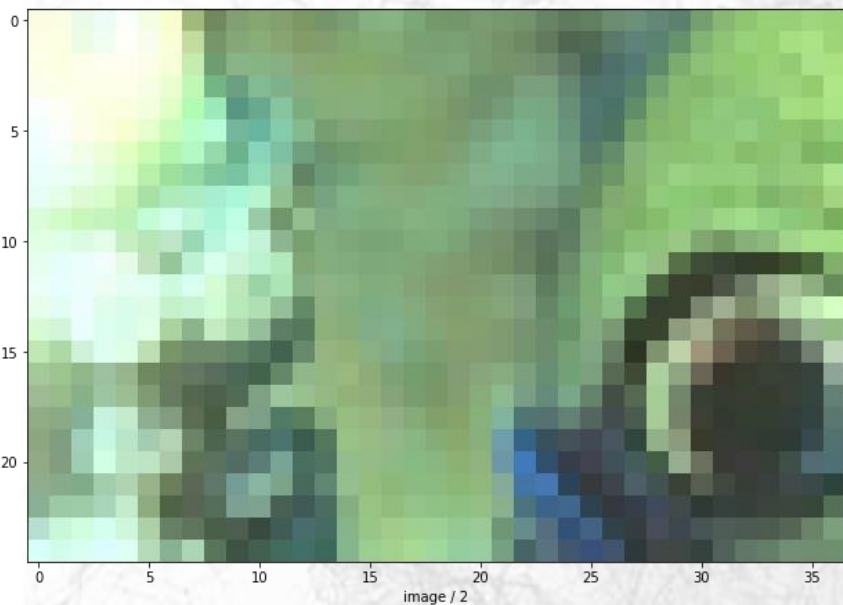


Original

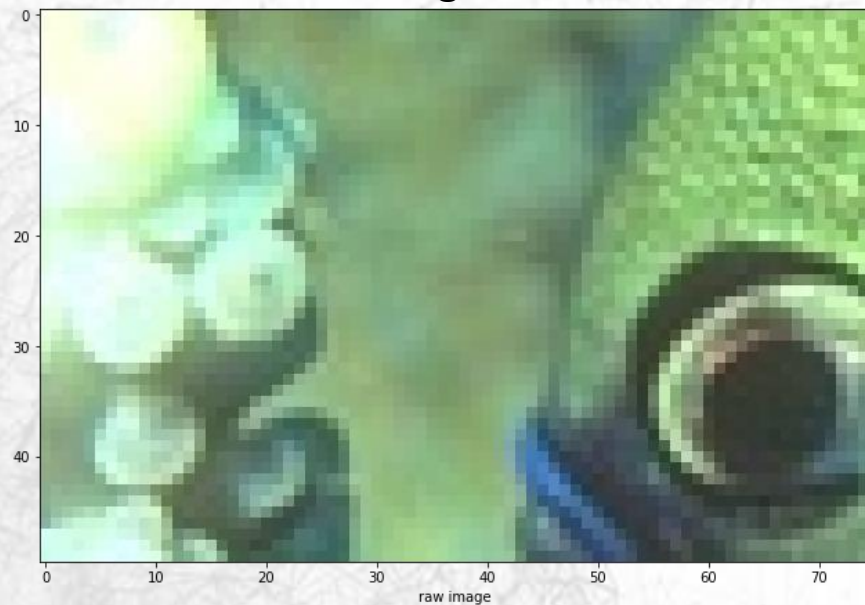


Resized x 2

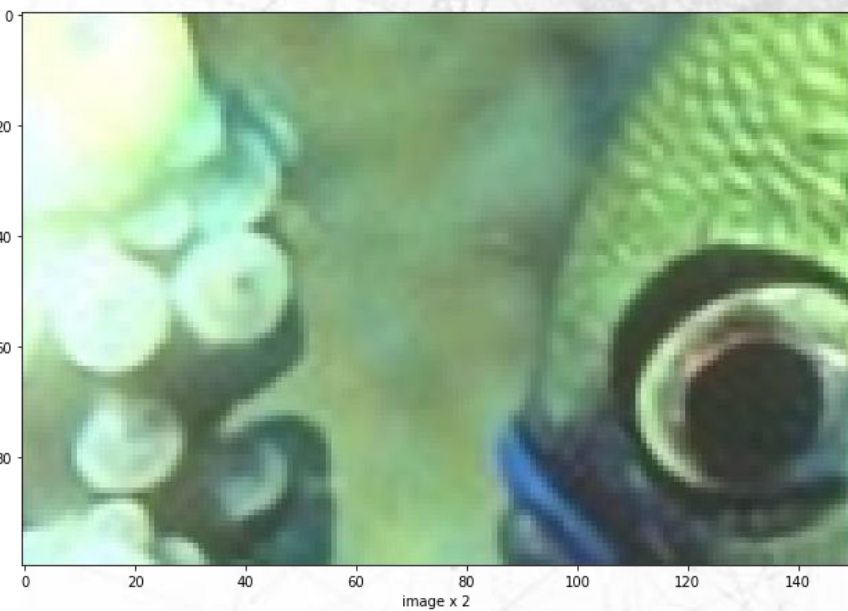
Resized x 0.5



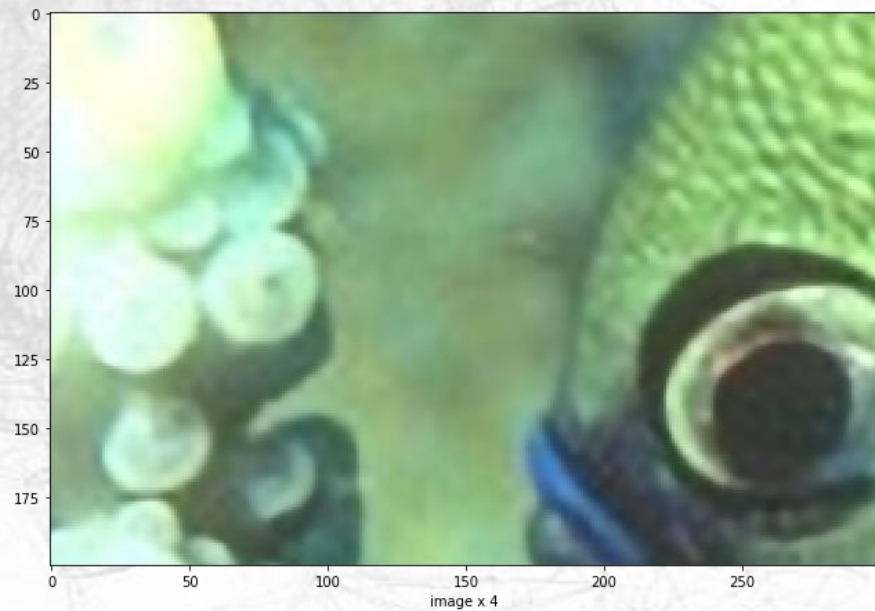
Original



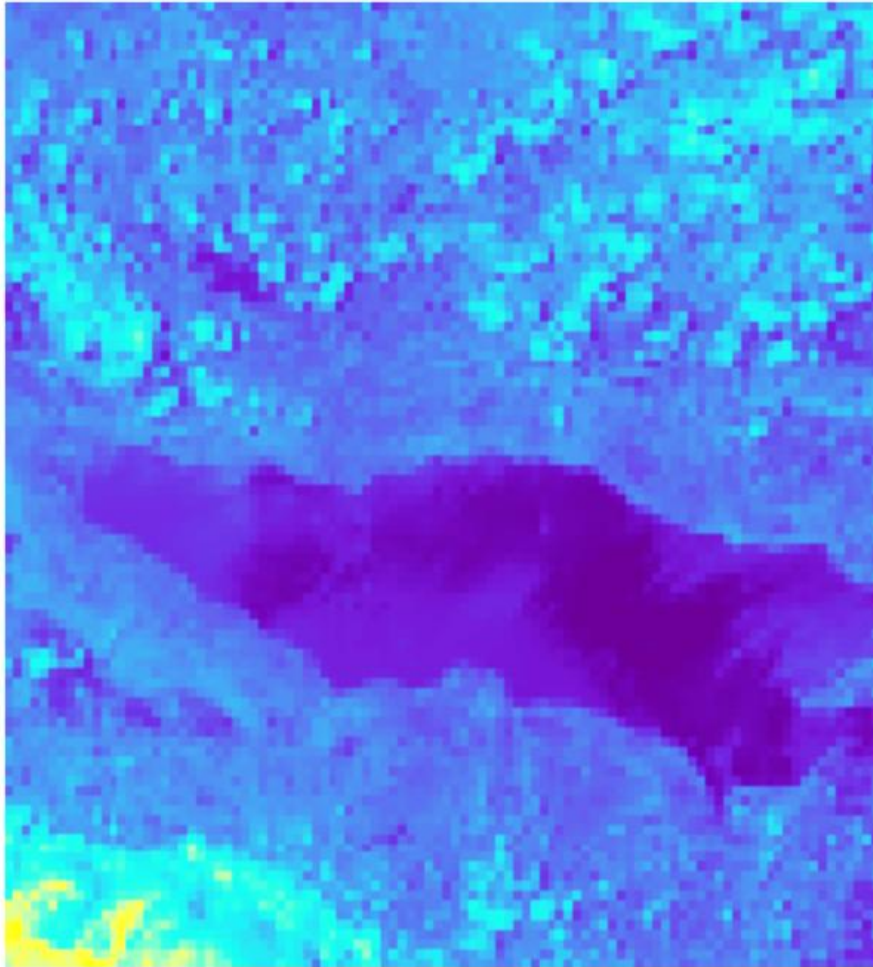
Resized x 2



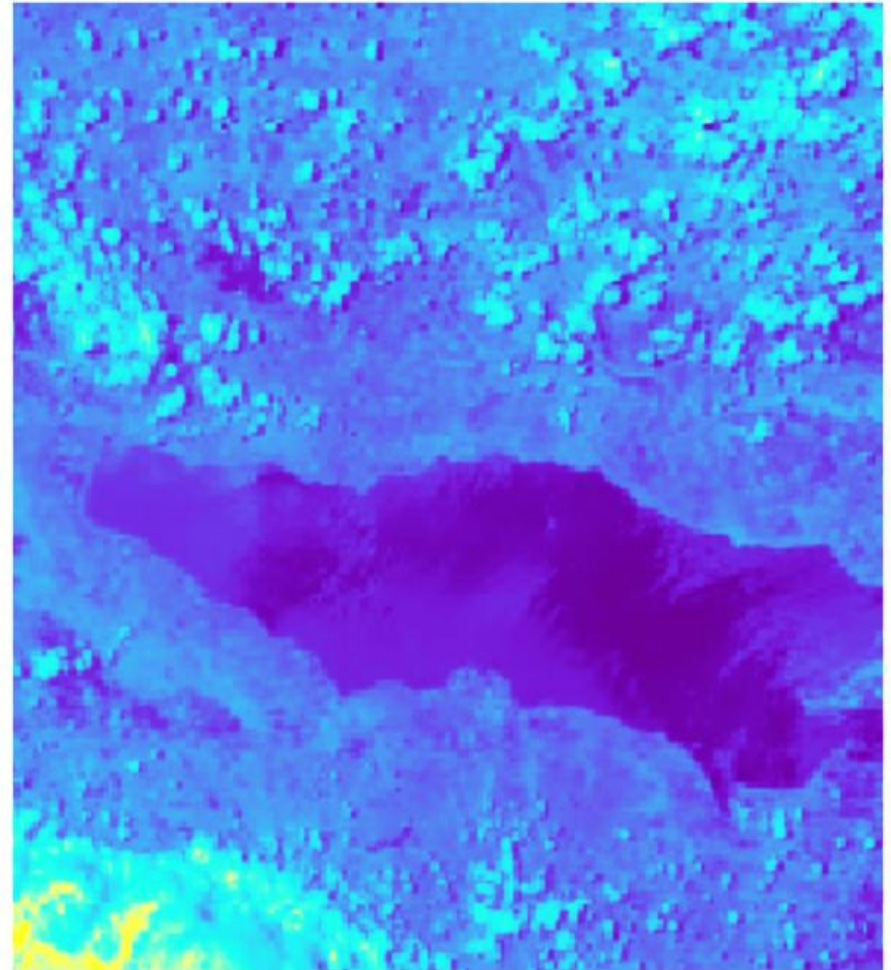
Resized x 4



Aplicaciones: Fusión de imágenes satelitales



Original M7 Band
(750mx750m)



Fused M7 Band
(375mx375m)

Aplicaciones: Colorización

The Lost World (1925)



Automated colorization of films.

Aplicaciones: Art style transfer



"Starry Night", Vincent van Gogh

Aplicaciones: Art style transfer



Objetivos del Taller

- Este es un taller en dos etapas:
 - en la primera: vamos a implementar una red neuronal simple (y multicapa) usando Keras.
 - en la segunda: aprenderemos a implementar una red convolucional usando Keras.
- Para lograr nuestro objetivo, usaremos como ejemplo la base de datos MNIST.

Objetivos del Taller

Base de datos MNIST:



70,000 imágenes (de tamaño 28 x 28).

(de ellos, 60,000 son de entrenamiento y 10,000 son de prueba).

Vamos a implementar redes que aprendan a clasificar los dígitos de MNIST.

¿Qué necesitamos?

- Pre-procesar los datos, para hacerlos adecuados para las redes neuronales.
- Cómo definir el problema de optimización.
- Aprender a construir la red neuronal en Keras (es muy fácil !! Keras = LEGO).
- Buscar los parámetros adecuados.
- Entrenar la red neuronal (para clasificar).

Keras

- Desarrollada por François Chollet (2015-2016).



- Documentación: <https://keras.io/>
- Material de apoyo:

notas del curso “Aprendizaje de máquina” del Prof. Mariano Rivera

[http://personal.cimat.mx:8181/~mrivera/
cursos/temas_aprendizaje.html](http://personal.cimat.mx:8181/~mrivera/cursos/temas_aprendizaje.html)

Vamos a trabajar ejemplos con base en el libro de François Chollet (2018)

