

# Redes Neuronales y Aprendizaje Profundo para Visión Artificial

Dr. Ariel H. Curiale

Cuatrimestre: 2do de 2019

---

## PRÁCTICA 3: INTRODUCCIÓN A TENSORFLOW

---

1. Usando los datos provisto por sklearn sobre el precio de los inmuebles en Boston (`from sklearn.datasets import load_boston`), entrenar un modelo de regresión lineal que permita predecir el precio de los inmuebles ( $Wx + b$ ). Graficar la función de costo sobre los datos de validación y el ajuste final utilizando la función de matplotlib `scatter`. Normalizar los datos de entrada y utilizar  $\sim 25\%$  para testing.
2. Implementar los problemas 3, 4 y 6 de la práctica 2 utilizando TensorFlow.
3. Predecir si los cliente de un banco sacarán un plazo un fijo utilizando una red neuronal compuesta por cuatro capas ocultas de 300 neuronas mas una capa de salida. Los datos se corresponden con una campaña de marketing de un banco portugués. El resultado de la campaña se encuentra codificado en la variable `y` del archivo `bank-additional-full.csv`. Para más información sobre los datos visitar<sup>1</sup>. Excluir la característica `duration` ya que se corresponde con el tiempo de duración de la llamada. Este dato es el único que no se conoce antes de hacer la llamada.

Como pueden observar si abren el archivo de texto, los datos no están preparados para ser la entrada de una red neuronal ya que algunas características contienen valores categóricos (se encuentran definidos en una categoría). Para preprocesar los datos de forma adecuada se puede utilizar `scikit-learn` (`preprocessing.LabelEncoder()`), `pandas` (`pd.get_dummies()`) o hacerlo directamente en `numpy`. Normalizar los datos y utilizar  $\sim 25\%$  para validación (testing). Además introduzca una regularización mediante dropouts según le parezca. Defina el optimizador y su tasa de aprendizaje, la función de costo a utilizar y estudie diferentes funciones de activación no lineales.

4. Imagine que es un botánico que busca automatizar la clasificación de las flores de la especie Iris según las medidas de largo y ancho de sus sépalos y pétalos. Proponga una arquitectura de red que permita resolver dicha problemática considerando las siguientes tres especie de iris: setosa, virginica y versicolor. Los datos y la información de las características se pueden obtener en <sup>2</sup> utilizando `scikit-learn`.
5. Implementar el problema MNIST para una red con 5 capas densas de 200, 100, 60 y 30 neuronas en sus capas ocultas. Defina el optimizador y su tasa de aprendizaje y la función de costo. Utilizar el método `glorot_uniform` como inicialización de los pesos de la red. Comparar el rendimiento de dicha arquitectura utilizando como función de activación la función

---

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets/bank+marketing>

<sup>2</sup>[https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)

sigmoid, relu y relu+dropout. Graficar la función de costo y la precisión de cada arquitectura.

6. Implementar la red propuesta por LeCun (LeNet-5) [1] para resolver el problema de la clasificación de dígitos manuscritos. La arquitectura original de LeNet-5 es la siguiente:

Layer	Features	size	Kernel size	Strides	Activation
Imagen	1	32x32	-	-	-
Conv	6	28x28	5x5	1	tanh
Avg. Pooling	6	14x14	2x2	2	-
Conv	16	10x10	5x5	1	tanh
Avg. Pooling	16	5x5	2x2	2	-
Conv	120	1x1	5x5	1	tanh
FC	-	84	-	-	tanh
FC	-	10	-	-	softmax

Luego comparar la precisión en la detección de dígitos manuscritos si se cambia la función de activación por relu y se aplica un max pooling. Nosotros vamos a trabajar con imágenes de 28x28 maneje el padding de las convoluciones de forma adecuada para que el tamaño de las capas de salida coincidan con la arquitectura original que se muestra en la tabla.

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, pp. 2278–2324, Nov. 1998.