



Universidad  
Carlos III de Madrid

Grado en Ingeniería Informática

Curso 2020/2021

**Redes de Neuronas Artificiales**

# **Problema de Regresión**

*Predicción del precio medio de la vivienda en California*

**Autores:**

Alba Reinders Sánchez  
Alejandro Valverde Mahou

100383444  
100383383

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Preparación de Datos</b>	<b>3</b>
2.1. Normalización . . . . .	3
2.2. Aleatorización . . . . .	3
2.3. Separación en conjuntos de datos . . . . .	3
<b>3. Adaline</b>	<b>4</b>
3.1. Experimentación . . . . .	4
3.2. Resultados Obtenidos . . . . .	4
3.3. Análisis . . . . .	4
<b>4. Perceptron Multicapa</b>	<b>4</b>
4.1. Experimentación . . . . .	4
4.2. Resultados Obtenidos . . . . .	4
4.3. Análisis . . . . .	4
<b>5. Comparación de Modelos</b>	<b>4</b>
<b>6. Conclusiones</b>	<b>4</b>

## 1. Introducción

El problema a resolver es la **predicción del precio medio de la vivienda en California**, usando dos modelos diferentes: *Adaline* y *Perceptron Multicapa*.

El *Adaline* es un modelo **lineal**, mientras que el *Perceptron Multicapa* es un modelo **no lineal**. El objetivo de esta práctica es realizar una comparativa entre estos dos modelos mediante la experimentación y análisis de los resultados, para averiguar cuál de los dos es capaz de encontrar la solución más cercana a la solución óptima.

## 2. Preparación de Datos

Los datos proporcionados son: *longitude*, *latitude*, *housingMedianAge*, *totalRooms*, *totalBedrooms*, *population*, *households*, *medianIncome*, ***medianHouseValue***.

La salida de los modelos deberá ser ***medianHouseValue*** en función del resto de atributos.

El conjunto de ejemplos proporcionados es de **17000**.

### 2.1. Normalización

El primer paso en el preprocesado de los datos es la **normalización**. Esta técnica consiste en acotar todos los datos en un rango de 0 a 1. Se ha decidido para este problema, normalizar exclusivamente los atributos de entrada, y no la salida.

La normalización de los datos se realiza cuando los distintos atributos de entrada no están en la misma escala, ya que tener atributos con escalas muy diferentes puede dar lugar al cálculo erróneo de los pesos, lo que deriva en modelos ineficaces.

La transformación lineal que se aplica a cada atributo es:

$$atr'_i = \frac{atr_i - \min(atr)}{\max(atr) - \min(atr)}$$

### 2.2. Aleatorización

Para evitar un entrenamiento inadecuado, es necesario otorgar a los modelos una lista de datos desordenados, o con orden aleatorio. De esta forma el modelo no se ajusta a un rango concreto de valores, que podrían darse seguidos si no se organizan aleatoriamente.

### 2.3. Separación en conjuntos de datos

Dado que este problema tiene una cantidad suficientemente grande de datos, se puede realizar la división del conjunto de datos en 3 subconjuntos:

- **Conjunto de Entrenamiento:**

Con él se realiza el aprendizaje (ajuste de pesos) del modelo. Es el conjunto más grande, pues tiene el 60 % de los datos (10200 instancias).

- **Conjunto de Test:**

Se usa para evaluar la precisión y capacidad de generalización del modelo. Este conjunto tiene el 20 % de los datos (3400 instancias).

- **Conjunto de Validación:**

Se usa para determinar los mejores hiper-parámetros del modelo. Este conjunto tiene el 20 % de los datos (3400 instancias).

### **3. Adaline**

#### **3.1. Experimentación**

#### **3.2. Resultados Obtenidos**

#### **3.3. Análisis**

### **4. Perceptron Multicapa**

#### **4.1. Experimentación**

#### **4.2. Resultados Obtenidos**

#### **4.3. Análisis**

### **5. Comparación de Modelos**

### **6. Conclusiones**