

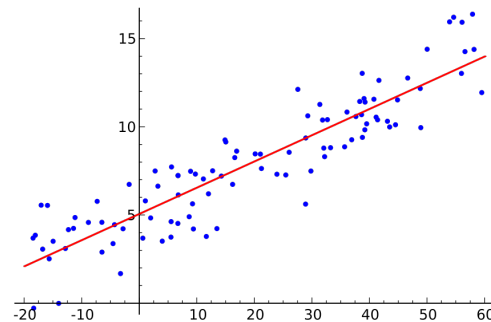
Redes Neuronales y Aprendizaje Profundo para Visión Artificial

Dr. Ariel H. Curiale

Cuatrimestre: 2do de 2019

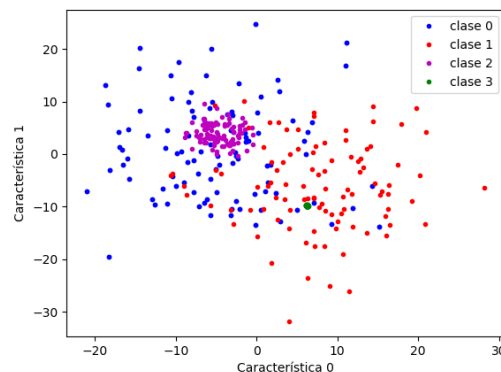
PRÁCTICA 1: INTRODUCCIÓN A LAS TÉCNICAS DE ML

1. A partir de un conjunto de datos n -dimensionales ($n \geq 5$), $\{(x_1, \dots, x_n)_i\}$, implementar una regresión lineal utilizando mínimos cuadrados para calcular los parámetros b, a_1, \dots, a_n de forma que un ajuste lineal de la forma $y = a_1x_1 + \dots + a_nx_n + b$ representa una aproximación razonable. Genere el conjunto de datos de forma aleatoria tal que los datos y el posible ajuste sean como el que se observa en la siguiente figura para el caso 1D:



Estudiar el efecto del error respecto la cantidad de datos y la dimensionalidad del problema. Relacionarlo con el problema conocido como la maldición de la dimensionalidad.

2. Generar un conjunto de datos n -dimensionales ($n \geq 3$) utilizando p distribuciones normales con $p \geq 4$ de forma similar al que se muestra en la siguiente figura:



Para la generación de los datos, fijar el centro y la varianza de las distribuciones de forma aleatoria en un rango razonable de los mismos para que en particular dos clases se solapen ligeramente. A partir de estos datos implementar el algoritmo de k-means ($k \geq 4$) para clasificar los datos. Presentar tres gráficos para observar la evolución de la clasificación y

sus medias (instantes inicial, t y cuándo alcanza el estado estacionario). Además, mostrar un ejemplo para el caso en que el método k-means no resuelve bien la clasificación.

3. Implementar el método de clasificación k-nearest neighbors para clasificar las imágenes de la base de datos MNIST y CIFAR-10. Se espera tener una exactitud del 100% para el caso de MNIST con los primeros 20 ejemplos de test (usar keras para cargar los datos).
4. Generar un conjunto de datos bidimensionales formado por 5 o más clases. Luego, utilizar el método k-nearest neighbors para clasificar los datos tomando 1, 3 y 7 vecinos. Por último, graficar las fronteras de decisión.
5. Implementar el método de clasificación lineal SVM y SoftMax con una regularización L_2 utilizando el paradigma de objetos. Evaluar la implementación con la base de datos MNIST y CIFAR-10. Para implementar los clasificadores hacer uso de la herencia. En particular hacer que hereden la funcionalidad de la clase LinearClassifier: fit, predict y loss_gradient. Implementar también la métrica accuracy para ir viendo la precisión durante el entrenamiento. Por cuestiones de estabilidad numérica, ya que aparece una exponencial en el método SoftMax, es recomendable que los scores sean pequeños al utilizar este método. Comparar la exactitud de los métodos y graficar las curvas de loss y accuracy para las distintas épocas tanto para el entrenamiento como la evaluación del método. ¿Cuál es mejor? ¿Alguno es mejor para algún problema en particular?

Para el entrenamiento utilizar el método mini-batch o BGD.