

# Práctica 1 - Aprendizaje Automático

## Búsqueda Iterativa de Óptimos y Regresión Lineal

---

Límite de entrega: **3 de Abril de 2022 a las 23:59 (PRADO)**

Valoración máxima: 12 puntos (+2 puntos de Bonus)

Materiales a entregar: un fichero Python (.py o .ipynb) y un informe describiendo y analizando el trabajo desarrollado, los resultados obtenidos y las conclusiones extraídas. En caso de que se opte por entregar un Colab Notebook, el informe debe estar integrado en el mismo cuaderno (intercalando texto, código y resultados).

---

### NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Es obligatorio presentar un informe con las valoraciones y decisiones adoptadas en el desarrollo de cada uno de los apartados. En dicho informe se incluirán los gráficos generados. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. Sin este informe se considera que el trabajo NO ha sido presentado.

El incumplimiento de las normas que se listan a continuación puede implicar la pérdida de 2 puntos por cada norma incumplida:

- Cada ejercicio/apartado de la práctica debe quedar perfectamente identificado en el material entregado (código y memoria).
- Todos los resultados numéricos o gráficas serán mostrados por pantalla, parando la ejecución después de cada apartado.
- El código NO puede escribir nada a disco.
- El path que se use en la lectura de cualquier fichero auxiliar de datos debe ser siempre “datos/nombre\_fichero”. Es decir, se espera que el código lea de un directorio llamado “datos”, situado dentro del directorio donde se desarrolla y ejecuta la práctica.
- Un código es apto para ser corregido si se puede ejecutar de principio a fin sin errores.
- No es válido usar opciones en las entradas. Para ello, se deben fijar al comienzo los valores por defecto que se consideren óptimos.
- El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
- En caso de que haya más de un fichero (por ejemplo, \*.py y \*.pdf), estos se entregarán juntos dentro de un único fichero zip, sin ningún directorio que los contenga.
- Se entrega solamente el código fuente, y no los datos empleados.

---

1. EJERCICIO SOBRE LA BÚSQUEDA ITERATIVA DE ÓPTIMOS (6.5 puntos)

1. (1 punto) Implementar el algoritmo de gradiente descendente.
2. (2 puntos) Considerar la función  $E(u, v) = (u \cdot v \cdot e^{(-u^2 - v^2)})^2$ . Usar gradiente descendente para encontrar un mínimo de esta función, comenzando desde el punto  $(u, v) = (0.5, -0.5)$  y usando una tasa de aprendizaje  $\eta = 0,1$ .
  - a) Calcular analíticamente y mostrar la expresión del gradiente de la función  $E(u, v)$ .
  - b) ¿Cuántas iteraciones tarda el algoritmo en obtener por primera vez un valor de  $E(u, v)$  inferior a  $10^{-8}$ .
  - c) ¿En qué coordenadas  $(u, v)$  se alcanzó por primera vez un valor igual o menor que  $10^{-8}$  en el apartado anterior.
3. (2 puntos) Considerar ahora la función  $f(x, y) = x^2 + 2y^2 + 2\sin(2\pi x)\sin(\pi y)$ 
  - a) Usar gradiente descendente para minimizar esta función. Usar como punto inicial  $(x_0 = -1, y_0 = 1)$ , tasa de aprendizaje  $\eta = 0,01$  y un máximo de 50 iteraciones. Generar un gráfico de cómo desciende el valor de la función con las iteraciones. Repetir el experimento pero usando  $\eta = 0,1$ , comentar las diferencias y su dependencia de  $\eta$ .
  - b) Obtener el valor mínimo y los valores de las variables  $(x, y)$  en donde se alcanzan cuando el punto de inicio se fija en:  $(-0.5, -0.5)$ ,  $(1, 1)$ ,  $(2.1, -2.1)$ ,  $(-3, 3)$ ,  $(-2, 2)$ . Generar una tabla con los valores obtenidos, empleando el máximo número de iteraciones (50) y las tasas de aprendizaje del anterior apartado ( $\eta = 0,01$  y  $\eta = 0,1$ ). Comentar la dependencia del punto inicial.
4. (1.5 punto) ¿Cuál sería su conclusión sobre la verdadera dificultad de encontrar el mínimo global de una función arbitraria?

## 2. EJERCICIO SOBRE REGRESIÓN LINEAL (5.5 puntos)

1. (2.5 puntos) Este ejercicio ajusta modelos de regresión a vectores de características extraídos a partir de imágenes de dígitos manuscritos. En particular, se extraen dos características concretas que miden el valor medio del nivel de gris y la simetría del dígito respecto de su eje vertical. Solo se seleccionarán para este ejercicio las imágenes de los números 1 y 5.

Estimar un modelo de regresión lineal, a partir de los datos proporcionados por los vectores de características dados, usando tanto el algoritmo de la pseudo-inversa<sup>1</sup> como el gradiente descendente estocástico (SGD). Las etiquetas serán  $\{-1, 1\}$ , una por cada vector de cada uno de los números. Pintar las soluciones obtenidas junto con los datos usados en el ajuste. Valorar la bondad del resultado usando  $E_{in}$  y  $E_{out}$  (para  $E_{out}$  calcular las predicciones usando los datos del fichero de test).

2. (3 puntos) En este apartado exploramos cómo se transforman los errores  $E_{in}$  y  $E_{out}$  cuando aumentamos la complejidad del modelo lineal usado. Ahora hacemos uso de la función `simula_unif(N, 2, size)` que nos devuelve N coordenadas 2D de puntos uniformemente muestreados dentro del cuadrado definido por  $[-size, size] \times [-size, size]$ . Se debe realizar el siguiente experimento:

- a) Generar una muestra de entrenamiento de  $N = 1000$  puntos en el cuadrado  $\mathcal{X} = [-1, 1] \times [-1, 1]$ . Pintar el mapa de puntos 2D.
- b) Consideremos la función  $f(x_1, x_2) = \text{sign}((x_1 - 0.2)^2 + x_2^2 - 0.6)$  que usaremos para asignar una etiqueta a cada punto de la muestra anterior. Introducimos ruido sobre las etiquetas cambiando el signo de un 10% de las mismas elegido aleatoriamente. Pintar el mapa de etiquetas obtenido.
- c) Usando como vector de características  $(1, x_1, x_2)$ , ajustar un modelo de regresión lineal al conjunto de datos generado y estimar los pesos  $w$ . Estimar el error de ajuste  $E_{in}$  usando SGD.
- d) Ejecutar todo el experimento definido por (a)-(c) 1000 veces (generamos 1000 muestras diferentes) y
  - Calcular el valor medio de los errores  $E_{in}$  de las 1000 muestras.
  - Generar 1000 puntos nuevos por cada iteración y calcular con ellos el valor de  $E_{out}$  en dicha iteración. Calcular el valor medio de  $E_{out}$  en todas las iteraciones.
- e) Valore qué tan bueno considera que es el ajuste con este modelo lineal a la vista de los valores medios obtenidos de  $E_{in}$  y  $E_{out}$ .
  - Repetir el mismo experimento anterior pero usando características no lineales. Ahora usaremos el siguiente vector de características:  $\phi_2(x) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$ . Ajustar el nuevo modelo de regresión lineal y calcular el nuevo vector de pesos  $\hat{w}$ . Calcular los errores promedio de  $E_{in}$  y  $E_{out}$ .
  - A la vista de los resultados de los errores promedios  $E_{in}$  y  $E_{out}$  obtenidos en los dos experimentos, ¿qué modelo considera que es el más adecuado? Justifique la respuesta.

---

<sup>1</sup>La pseudo-inversa debe ser calculada por medio de la descomposición en valores singulares vista en teoría (esto es, debe emplearse la función `np.linalg.svd()`).

### 3. BONUS (2 puntos)

El BONUS solo se tendrá en cuenta si se ha obtenido al menos el 75 % de los puntos de la parte obligatoria.

(2 puntos) Implementar el algoritmo de minimización de Newton y aplicarlo a la función  $f(x, y)$  dada en el Ejercicio 1.3. Desarrolle los mismos experimentos usando los mismos puntos de inicio. Generar un gráfico de cómo descende el valor de la función con las iteraciones. Extraer conclusiones sobre las conductas de los algoritmos comparando la curva de decrecimiento de la función con el método de Newton y con el gradiente descendente.