APRENDIZAJE AUTOMÁTICO

Trabajo.1: Programación

Fecha límite de entrega: 28 de Marzo

Valoración máxima: 12.5 puntos

NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Para este trabajo como para los demás es obligatorio presentar un informe escrito con las valoraciones y decisiones adoptadas en el desarrollo de cada uno de los apartados. Incluir en el informe los gráficos generados. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. (obligatorio en pdf). **Sin este informe se considera que el trabajo NO ha sido presentado.**

Normas para el desarrollo de los Trabajos: EL INCUMPLIMIENTO DE ESTAS NORMAS SIGNIFICA PERDIDA DE 2 PUNTOS POR CADA INCUMPLIMIENTO.

- El código de cada ejercicio/apartado de la práctica se debe estructurar en un script Python incluyendo las funciones que se hayan definido. Cada script debe incluirse en un fichero distinto.
- Todos los resultados numéricos o gráficas serán mostrados por pantalla, parando la ejecución después de cada apartado. EL codigo NO DEBE 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 se 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 fijar al comienzo los parámetros por defecto que considere que son los óptimos.
- El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
- Poner puntos de parada para mostrar imágenes o datos por consola.
- Todos los ficheros (*.py, *.pdf) se entregan juntos dentro de un único fichero zip, sin ningún directorio que los contenga.
- ENTREGAR SOLO EL CODIGO FUENTE, NUNCA LOS DATOS.
- **Forma de entrega**: Subir el zip al Tablón docente de CCIA.

1. Ejercicio sobre la búsqueda iterativa de óptimos

Gradiente Descendente (7 puntos).

1. (1 punto) Implementar el algoritmo de gradiente descendente.
2. (2 puntos) Considerar la función $E(u, v) = (ue^v - 2ve^{-u})^2$. Usar gradiente descendente para encontrar un mínimo de esta función, comenzando desde el punto $(u, v) = (1, 1)$ 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^{-14} . (Usar flotantes de 64 bits)
   c) ¿En qué coordenadas (u, v) se alcanzó por primera vez un valor igual o menor a 10^{-14} en el apartado anterior.
3. (2 puntos) Considerar ahora la función $f(x, y) = (x - 2)^2 + 2(y + 2)^2 + 2\sin(2\pi x)\sin(2\pi y)$
   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 η.
   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: $(2{,}1, -2{,}1)$, $(3, -3)$,$(1{,}5, 1{,}5)$,$(1, -1)$. Generar una tabla con los valores obtenidos
4. (2 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)

Este ejercicio ajusta modelos de regresión a vectores de características extraidos de imágenes de digitos manuscritos. En particular se extraen dos característcas concretas: el valor medio del nivel de gris y simetría del número respecto de su eje vertical. Solo se seleccionarán para este ejercicio las imágenes de los números 1 y 5.

1. (2.5 puntos) Estimar un modelo de regresión lineal a partir de los datos proporcionados de dichos números (Intensidad promedio, Simetria) usando tanto el algoritmo de la pseudo-inversa como Gradiente descendente estocástico (SGD). Las etiquetas serán $\{-1, 1\}$, una para 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). (usar $Regress\_Lin(datos, label)$ como llamada para la función (opcional)).
2. (3 puntos) En este apartado exploramos como 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 $[-size, size] \times [-size, size]$
   - 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. (ver función de ayuda)

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 aleatoriamente el signo de un 10 % de las mismas. Pintar el mapa de etiquetas obtenido.

c) Usando como vector de características $(1, x_1, x_2)$ ajustar un modelo de regresion lineal al conjunto de datos generado y estimar los pesos w. Estimar el error de ajuste E_{in} usando Gradiente Descendente Estocástico (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 que 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: $\Phi_2(x) = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)$. 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 ¿Que modelo considera que es el más adecuado? Justifique la decisión.

2.1. BONUS

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

1. (2 puntos) **Método de Newton** Implementar el algoritmo de minimización de Newton y aplicarlo a la función $f(x, y)$ dada en el ejercicio.3. Desarrolle los mismos experimentos usando los mismos puntos de inicio.
   - Generar un gráfico de como desciende el valor de la función con las iteraciones.
   - Extraer conclusiones sobre las conductas de los algoritmos comparando la curva de decrecimiento de la función calculada en el apartado anterior y la correspondiente obtenida con gradiente descendente.