APRENDIZAJE AUTOMÁTICO

Trabajo.2: Prácticas

Fecha límite de entrega: 1 Mayo

Valoración máxima: 12 puntos

NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Para este trabajo como para los demás es obligatorio presentar un informe escrito con sus 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 Python incluyendo las funciones que se hayan definido.
- 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 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 a PRADO

1. Ejercicio sobre la complejidad de H y el ruido (5 puntos)

En este ejercicio debemos aprender la dificultad que introduce la aparición de ruido en las etiquetas a la hora de elegir la clase de funciones más adecuada. Haremos uso de tres funciones incluidas en el fichero *template_trabajo2.py*:

- $simula\_unif(N, dim, rango)$, que calcula una lista de N vectores de dimensión dim. Cada vector contiene dim números aleatorios uniformes en el intervalo $rango$.
- $simula\_gaus(N, dim, sigma)$, que calcula una lista de longitud N de vectores de dimensión dim, donde cada posición del vector contiene un número aleatorio extraído de una distribucción Gaussiana de media 0 y varianza dada, para cada dimension, por la posición del vector $sigma$.
- *simula_ recta(intervalo)* , que simula de forma aleatoria los parámetros, $v = (a, b)$ de una recta, $y = ax + b$, que corta al cuadrado $[-50, 50] \times [-50, 50]$.

1. (1 punto) Dibujar gráficas con las nubes de puntos simuladas con las siguientes condiciones:
   a) Considere $N = 50$, $dim = 2$, $rango = [-50, +50]$ con $simula\_unif(N, dim, rango)$.
   b) Considere $N = 50$, $dim = 2$ y $sigma = [5, 7]$ con $simula\_gaus(N, dim, sigma)$.
2. Vamos a valorar la influencia del ruido en la selección de la complejidad de la clase de funciones. Con ayuda de la función $simula\_unif(100, 2, [-50, 50])$ generamos una muestra de puntos 2D a los que vamos añadir una etiqueta usando el signo de la función $f(x, y) = y - ax - b$, es decir el signo de la distancia de cada punto a la recta simulada con *simula_ recta()*.
   a) (1 punto) Dibujar un gráfico 2D donde los puntos muestren (use colores) el resultado de su etiqueta. Dibuje también la recta usada para etiquetar. (Observe que todos los puntos están bien clasificados respecto de la recta)
   b) (0.5 puntos) Modifique de forma aleatoria un 10 % de las etiquetas positivas y otro 10 % de las negativas y guarde los puntos con sus nuevas etiquetas. Dibuje de nuevo la gráfica anterior. (Ahora habrá puntos mal clasificados respecto de la recta)
   c) (2.5 puntos) Supongamos ahora que las siguientes funciones definen la frontera de clasificación de los puntos de la muestra en lugar de una recta
   - $f(x, y) = (x - 10)^2 + (y - 20)^2 - 400$
   - $f(x, y) = 0.5x + y + 10$
   - $f(x, y) = y - 20x^2 - 5x + 3$

   Visualizar el etiquetado generado en $2b$ junto con cada una de las gráficas de cada una de las funciones. Comparar las regiones positivas y negativas de estas nuevas funciones con las obtenidas en el caso de la recta. Argumente si estas funciones más complejas son mejores o peores que la función lineal. Observe las gráficas y diga que consecuencias extrae sobre la bondad o maldad de la modificación de etiquetas en el proceso de aprendizaje. Explicar el razonamiento.

Hay pantallazos de los resultados de estas gráfic

Habría que explicar si

Aquí no hay aprendizaje

Cuando visualizamos las funcione

10 % de ruido d

2. Modelos Lineales (7 puntos)

En el percep

a) **(3 puntos) Algoritmo Perceptron:** Implementar la función

$$ajusta\_PLA(datos, label, max\_iter, vini)$$

El perceptron no tiene co

que calcula el hiperplano solución a un problema de clasificación bi algoritmo PLA. La entrada *datos* Cada fila de la matr cada item con su etiqueta está representado por una fila de la matriz, *label* el vector de etiquetas (cada etiqueta es un valor $+1$ o -1), *max_iter* es el número máximo de iteraciones permitidas y *vini* el valor inicial del vector. La función devuelve los coeficientes del hiperplano.

En los ejer

Debemos crear

1) Ejecutar el algoritmo PLA con los datos simulados en los apartados $2a$ de la sección.1. Inicializar el algoritmo con: a) el vector cero y, b) con vectores de números aleatorios en $[0, 1]$ (10 veces). Anotar el número medio de iteraciones necesarias en ambos para converger. Valorar el resultado relacionando el punto de inicio con el número de iteraciones.
2) Hacer lo mismo que antes usando ahora los datos del apartado $2b$ de la sección.1. ¿Observa algún comportamiento diferente? En caso afirmativo diga cual y las razones para que ello ocurra.

e_in error

b) **(4 puntos) Regresión Logística**: En este ejercicio crearemos nuestra propia función objetivo f (una probabilidad en este caso) y nuestro conjunto de datos $\mathcal{D}$ para ver cómo funciona regresión logística. Supondremos por simplicidad que f es una probabilidad con valores 0/1 y por tanto que la etiqueta y es una función determinista de $\mathbf{x}$.

En este eje

Consideremos $d = 2$ para que los datos sean visualizables, y sea $\mathcal{X} = [0, 2] \times [0, 2]$ con probabilidad uniforme de elegir cada $\mathbf{x} \in \mathcal{X}$. Elegir una línea en el plano que pase por $\mathcal{X}$ como la frontera entre $f(\mathbf{x}) = 1$ (donde y toma valores $+1$) y $f(\mathbf{x}) = 0$ (donde y toma valores -1), para ello seleccionar dos puntos aleatorios de $\mathcal{X}$ y calcular la línea que pasa por ambos.

EXPERIMENTO: Seleccione $N = 100$ puntos aleatorios $\{\mathbf{x}_n\}$ de $\mathcal{X}$ y evalúe las respuestas $\{y_n\}$ de todos ellos respecto de la frontera elegida. Ejecute Regresión Logística (ver condiciones más abajo) para encontrar la función solución g y evalúe el error E_{out} usando para ello una nueva muestra grande de datos (> 999). Repita el experimento 100 veces, y

En cada

- Calcule el valor de E_{out} para el tamaño nuestral N=100.
- Calcule cúantas épocas tarda en converger en promedio RL para N=100 en las condiciones fijadas para su implementación.

Implementar Regresión Logística(RL) con Gradiente Descendente Estocástico (SGD) bajo las siguientes condiciones:

accura

- Inicializar el vector de pesos con valores 0.
- Parar el algoritmo cuando $||\mathbf{w}^{(t-1)} - \mathbf{w}^{(t)}|| < 0{,}01$, donde $\mathbf{w}^{(t)}$ denota el vector de pesos al final de la época t. Una época es un pase completo a través de los N datos.
- Aplicar una permutación aleatoria de $\{1, 2, ..N\}$ a los índices de los datos, antes de usarlos en cada época del algoritmo.
- Usar una tasa de aprendizaje $\eta = 0{,}01$

Si no se consigue un 1 de accuracy, probar con otras tasas de apren

Eout 97%Hay más error porque hay más puntos, y los punto

Error tenido en cuenta en cada algorit

Como en el PLA-Pocket se va

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

puntos) Clasificación de Dígitos. Considerar el conjunto de datos de los dígitos manus-
s y seleccionar las muestras de los dígitos 4 y 8. Usar los ficheros de entrenamiento (training)
st que se proporcionan. Extraer las características de **intensidad promedio** y **simetría** en
anera que se indicó en el ejercicio 3 del trabajo 1.

En el aparta

. Plantear un problema de clasificación binaria que considere el conjunto de entrenamiento como datos de entrada para aprender la función g.

. Usar un modelo de Regresión Lineal y aplicar PLA-Pocket como mejora. Responder a las siguientes cuestiones.

La importa

a) Generar gráficos separados (en color) de los datos de entrenamiento y test junto con la función estimada.

b) Calcular E_{in} y E_{test} (error sobre los datos de test).

c) Obtener cotas sobre el verdadero valor de E_{out}. Pueden calcularse dos cotas una basada en E_{in} y otra basada en E_{test}. Usar una tolerancia $\delta = 0{,}05$. ¿Que cota es mejor?