

## Práctico 6

### Regresión lineal. Regresión logística. Redes Neuronales.

#### Ejercicio 1

Considere al siguiente conjunto de entrenamiento, utilizado para aprender la función de hipótesis  $h_{\theta}(x) = \theta_0 + \theta_1 x_1$  mediante regresión lineal.

#	$x_1$	$y$
1	-1.0	2,6
2	0.0	2.0
3	1.0	2,4
4	2.0	4,1
5	3.0	6,5
6	4.0	9,8

- Utilice el método de las ecuaciones normales para calcular los valores de  $\theta$  que minimizan la función de costo.
- Implemente el algoritmo de descenso por gradiente y utilícelo para calcular  $\theta$ . Compare los resultados con los obtenidos en el punto anterior.
- ¿Cómo mejoraría la función de hipótesis para este problema en particular? Repita los pasos a) y b) con la nueva función de hipótesis.
- ¿Cuál es el objetivo de utilizar regularización en la función de costo?

#### Ejercicio 2

Considere el siguiente conjunto de entrenamiento:

#	$x_1$	$x_2$	Clase
1	-2	1	C
2	-1	1	C
3	0	1	C
4	1	1	B
5	2	1	A
6	-2	-1	C
7	-1	-1	C
8	0	-1	B
9	1	-1	A
10	2	-1	A

- Utilice regresión logística multiclase para obtener los parámetros de la función de hipótesis elegida. Determine las fronteras de decisión correspondientes.
- Indique la clase asignada al ejemplo:

#	$x_1$	$x_2$	Clase
1	-2.5	1.5	?

**Ejercicio 3**

- a) Explique la diferencia entre descenso por gradiente batch y descenso por gradiente estocástico. Enumere ventajas y desventajas comparativas.
- b) ¿Cuál es el rol del hiperparámetro  $\alpha$  en el descenso por gradiente? ¿Cuáles son las ventajas y desventajas de utilizar un valor muy grande o muy pequeño para este hiperparámetro?
- c) ¿Cuál es el propósito de normalizar las variables de entrada en la regresión lineal y en la regresión logística?

**Ejercicio 4 [\*]**

Considere el corpus «A quién voto» utilizado en la entrega anterior. Utilizaremos ese corpus para obtener dos clasificadores basados en regresión logística que, a partir de las preguntas, predigan el partido uno y el candidato el otro. A diferencia de las entregas anteriores, no hay restricciones en cuanto a utilización de bibliotecas como *scikit-learn* o *numpy*.

La metodología a seguir será la siguiente:

- a) Utilice para todo el ejercicio solamente los partidos y candidatos que tienen más de mil votos. Elimine al resto.
- b) Separe el corpus de entrada en dos conjuntos, seleccionando las instancias de forma aleatoria, reservando 80% para entrenamiento y 20% para evaluación.
- c) Construya los clasificadores a partir de las preguntas. Seleccione entre diferentes métodos de penalización utilizando validación cruzada sobre el corpus de entrenamiento, utilizando *accuracy* como medida de performance.
- d) Utilizando el método definido en (b), utilice PCA para seleccionar los  $n$  componentes principales, ajustando  $n$  nuevamente a través de validación cruzada.
- e) Evalúe los métodos obtenidos en (b) y (c) sobre el corpus de evaluación. Reporte *accuracy*, precisión, *recall* y medida-f para cada caso. Construya en cada caso la matriz de confusión entre las diferentes categorías.
- f) Utilizando el clasificador de candidato, construya un clasificador de partido que, simplemente, devuelva el partido asociado al candidato. Repita el paso d. ¿Se obtienen los mismos resultados? Comente.

**Ejercicio 5**

- a) Diseñe unidades sigmoide para las operaciones lógicas AND, OR y NOT.
- b) Diseñe una red multicapa para la operación XOR.
- c) Diseñe una red multicapa para la función booleana:  $((A \wedge B) \vee C) \wedge \neg D$ .
- d) ¿Cómo implementaría un problema de clasificación multiclase utilizando una red neuronal?

**Ejercicio 6**

- a) Diseñe e implemente una red neuronal con una única capa oculta pero cantidad arbitraria de neuronas.
- b) Implemente el algoritmo de Backpropagation (realice chequeo de gradiente durante el desarrollo para verificar la corrección del algoritmo implementado)
- c) Considere las siguientes funciones con las entradas en el rango  $[-1,1]$ :

$$\begin{aligned} f(x) &= x^3 - x^2 + 1 \\ g(x,y) &= 1 - x^2 - y^2 \\ h(x,y) &= x + y \end{aligned}$$

Utilice su implementación para ajustar estas funciones, inicializando los pesos con valores aleatorios y utilizando como ejemplos de entrenamiento 40 puntos uniformemente distribuidos en el espacio. Grafique:

- i. el error cometido en el conjunto de entrenamiento en cada iteración;
- ii. las aproximación de las funciones obtenidas luego de  $10^2$ ,  $10^3$ ,  $10^4$  y  $10^5$  iteraciones.

### Ejercicio 7

- a) ¿Qué ventaja tiene utilizar una función de costo de entropía cruzada respecto de una función de costo cuadrático para una red neuronal?
- b) ¿Cuándo se utiliza una capa softmax como última capa de una red neuronal?

### Ejercicio 8 [\*]

Implemente un nuevo jugador para el problema del ejercicio 2 del práctico 1, utilizando aprendizaje Q. A diferencia de lo visto en el teórico, la función Q se representa por una red neuronal en lugar de una tabla. Evalúe a este nuevo jugador contra los obtenidos en la entrega previa.