Deber 02

Rede Neuronales Artificales

Preguntas.

1. Un modelo de ajuste polinomial de grado n es una fórmula $y = p(x) = a_0 + a_1 x + \ldots a_n x^n$. Para un conjunto de datos $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ construya en una hoja de trabajo de Jupyter un modelo que utilice el método del gradiente descendiente, con una función que realice un ajuste polinomial de el grado n que se requiera. El factor α de descenso debe ser un parámetro de la función, así como la tolerancia, el valor de α por defecto será 0.001 y el valor por defecto de la tolerancia será 10^{-8}. La función debe devolver los valores de los coeficientes a_j, $j = 0, \ldots, n$

2. Utilice la función generada en el literal anterior para hacer ajustes de la tabla de datos *nolineal.csv* con polinomios de grado $n = 2, 3, \ldots, 7$. En cada caso grafique los datos y el polinomio de ajuste. Presente una tabla que muestre todos los gráficos de forma ordenada., utilice un layout 2×3, para hacerlo.

3. Utilice los datos del archivo *Tribus.csv* paragenerar un modelo de clasificación de individuos en las tribus de acuerdo a su localización. Siga el esquema descrito a continuación:

   (a) Generación de variables 'dummies'. Genere tres variables Y_{En}, Y_E y Y_T de a siguiente manera:

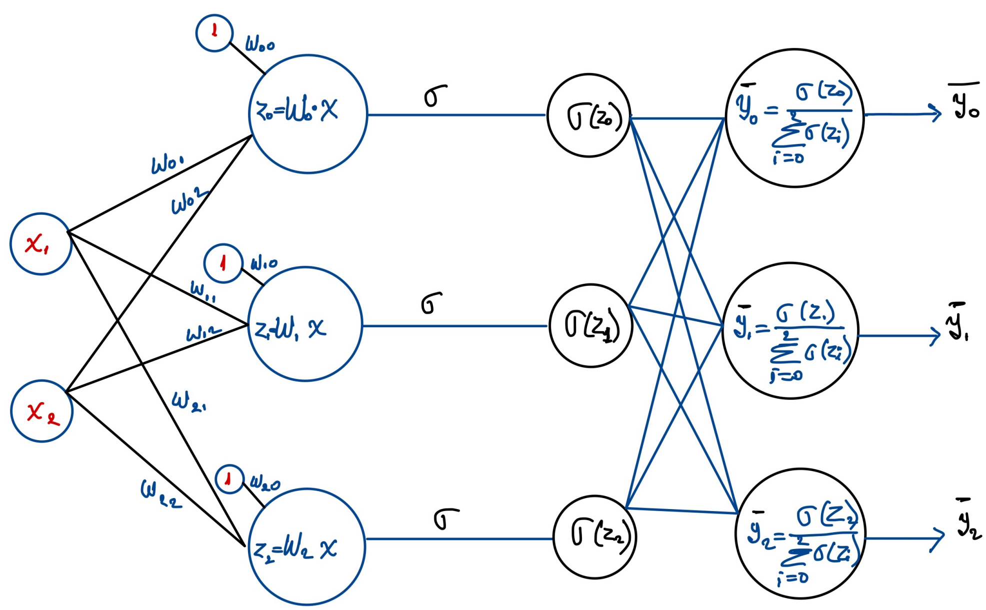
      i. Y_{En} es 1 si el individuo es un Enano caso contario Y_{En} es 0.

      ii. Y_E es 1 si el individuo es un Elfo caso contario Y_E es 0.

      iii. Y_T es 1 si el individuo es un Trol caso contario Y_T es 0.

   (b) Genere un modelo de clasificación neuronal, de una sola neurona, con la función de activación logísitica para cada variable respuesta generada en el literal anterior. Note que va a producir 3 modelos, cada uno decide si el individuo está o no en la clase de interés.

   (c) Note que la aproximación de clasificación del literal anterior resulta "incómodo" a la hora de clasificar nuevos individuos. Para evitar este problema, combine los tres modelos de la siguiente manera:



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

(d) Minimice la función de pérdida $\dfrac{1}{k+1}\displaystyle\sum_{j=0}^{k}\left\|Y_j-\overline{Y}_j\right\|^2$, donde $k+1$ es el número total de individuos , $Y_j = [\ Y_{En_j} \quad Y_{E_j} \quad Y_{T_j}\ ]$ y $\overline{Y}_j = [\ \overline{Y}_{0_j} \quad \overline{Y}_{1_j} \quad \overline{Y}_{2_j}\ ]$. Note que $\|\cdot\|$ es la norma usual de $\mathbb{R}^3$. El trabajo fuerte está en calcular el gradiente de la función de pérdida.

(e) Entrene el modelo eligiendo los pesos de manera eleatoria y calcule el porcentaje de bien clasificados de cada clase.

(f) Extra: Calcule y grafique la frontera para cada grupo.

4. Demuestre que si $\sigma(x)=\frac{1}{1+e^{-x}}$, entonces $\tanh(x)=2\sigma(2x)-1$. Grafique el resultado en $[-3,3]$, haga una captura de pantalla para incluirla en su redacción.

5. Otra aproximación a un modelo de clasificación binaria. Para un conjunto de datos:

i	V_1	V_2	$\ldots$	V_n	Y
1	x_{11}	x_{12}	$\ldots$	x_{1n}	y_1
2	x_{21}	x_{22}	$\ldots$	x_{2n}	y_2
3	x_{31}	x_{32}	$\ldots$	x_{3n}	y_3
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	x_{m1}	x_{m2}	$\ldots$	x_{mn}	y_m

con $y_i\in\{-1,1\}$, se tiene que la función de perdida asociada es la función $L=-\sum_{i=1}^{m}\log(\left|\frac{y_i}{2}-\frac{1}{2}-\hat{y}_i\right|)$, donde $\hat{y}_i=\sigma\left(\sum_{j=1}^{n}w_jx_{ij}\right)$ y $y_i\in\{-1,1\}$.

(a) Encuentre de forma explícita $\frac{\partial L}{w_j}$, $j=1,\ldots,n$.

(b) Plantee la forma vectorial general el método del gradiente descendiente.

(c) Programe en Python un script que permita encontrar los valores óptimos de los pesos w_j en un perceptrón para la base de datos *iris2.csv.*

6. Suponga que en la tabla de datos

i	V_1	V_2	$\ldots$	V_n	Y
1	x_{11}	x_{12}	$\ldots$	x_{1n}	y_1
2	x_{21}	x_{22}	$\ldots$	x_{2n}	y_2
3	x_{31}	x_{32}	$\ldots$	x_{3n}	y_3
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	x_{m1}	x_{m2}	$\ldots$	x_{mn}	y_m

la respuesta Y son rótulos de más de 2 clases digamos $\{c_1,c_2,..,c_l\}$, $l>2$. Para construir un clasificador automático, utilicemos k modeles logísticos de la siguiente forma:

- Para $k\in\{1,...,l\}$ construya el vector de datos Y_k donde $y_{ik}=\begin{cases}1 & si \quad y_i=c_k\\ -1 & si \quad y_i\neq c_k\end{cases}$.
- Calcule el modelo logístico correspondiente para cada tabla de datos:

i	V_1	V_2	$\ldots$	V_n	Y_k
1	x_{11}	x_{12}	$\ldots$	x_{1n}	y_{1k}
2	x_{21}	x_{22}	$\ldots$	x_{2n}	y_{2k}
3	x_{31}	x_{32}	$\ldots$	x_{3n}	y_{3k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	x_{m1}	x_{m2}	$\ldots$	x_{mn}	y_{mk}

- Con los k modelos anteriores, un nuevo individuo X defininido por los valores de las variables V_j con $j=1,\ldots,n$ se clasifica en la clase donde $\sigma(W_k\cdot X)$ sea máxima. W_k es el vector de pesos del $k-$ésimo modelo., $k=1,\ldots,l$.

Cree una hoja de Jupyter que utilizando la idea descrita calcule un modelo para clasificar los datos de la base *iris.csv*.

7. Para el conjunto de datos *plano.csv*, que son las coordenadas de puntos aleatorios en $\mathbb{R}^2$. Usando una hoja de trabajo de Jupyter construya una variable Y clasificando los puntos de la siguiente manera:

$$y_i = \begin{cases} \text{"circleIn"} & \text{si} \quad x_1^2 + x_2^2 \leq 4 \\ \\ \text{"circleOut"} & \text{si} \quad x_1^2 + x_2^2 > 4 \end{cases}$$

   - Haga un gráfico donde los puntos "circleIn" se marquen con cruces y los puntos "circleOut" se marque con diamantes.
   - Construya un modelo losgístico en base de los datos y de la clasificación realizada. Calcule el porcentaje de puntos bien clasificados en cada clase.
   - Utilizando np.random construya 100 puntos de manera aleatoria en el rectángulo $[-5, 5] \times [-5, 5]$.
   - Clasifique los datos utilizando el modelo construido, calcule el valor de la validación cruzada, grafique la clasificación y comente los resultados.

8. Repita el ejercicio anterior incluyendo como columnas de la tabla x_1^2 , x_2^2 y $x_1 * x_2$.