Descenso de Gradiente en Ciencias Biomédicas

Regresión lineal múltiple, regresión logística y Redes Neuronales

PhD. Pablo Eduardo Caicedo Rodríguez

2026-01-21

Objetivos de aprendizaje

- **Comprender** los fundamentos de la **Regresión Lineal**, **Regresión Logística** y **Perceptrón Multicapa (MLP)**.
- **Aplicar** estos modelos al contexto de **salud fetal** con datos de **cardiotocografía (CTG)**.
- **Evaluar** el desempeño con métricas adecuadas (MSE, AUC/Log-Loss, matriz de confusión, F1).

1. Regresión Lineal
2. Regresión Logística
3. Perceptrón Multicapa
4. Cierre y discusión

Dataset: `fetal_health.csv` (UCI CTG). **Contexto clínico**: interpretación de CTG (*normal*, *sospechoso*, *patológico*).

Contexto clínico (CTG)

Defición

Prueba médica que monitoriza simultáneamente la frecuencia cardíaca del feto y la actividad contráctil del útero. Se realiza generalmente durante el tercer trimestre del embarazo y el parto, colocando dos transductores externos (uno para la frecuencia cardíaca fetal y otro para las contracciones) sobre el abdomen de la madre

Contexto clínico (CTG)

Característica (Variable en CSV)	Cálculo o Descripción
Parámetros Basales	
baseline value	Es la frecuencia cardíaca fetal (FCF) media aproximada en un segmento de 10 minutos, excluyendo aceleraciones, deceleraciones y períodos de variabilidad marcada (>25 lpm). Se redondea a incrementos de 5 latidos por minuto (lpm).[4, 5, 6, 7, 8] El rango normal se considera entre 110 y 160 lpm.[9, 10]
fetal_movement	Número de movimientos fetales detectados por segundo.[1, 11, 12]
uterine_contractions	Número de contracciones uterinas por segundo. Se considera normal tener 5 o menos contracciones en 10 minutos.[1, 4, 11, 12]

Contexto clínico (CTG)

- CTG registra **FCF** y **contracciones uterinas**.
- Clasificación clínica (FIGO): **normal / sospechoso / patológico**.
- Variabilidad, aceleraciones y desaceleraciones son claves.

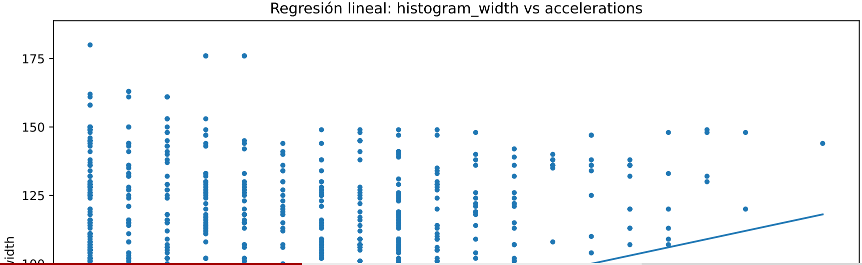
Regresión Lineal

Idea clave

Aproxima una relación **lineal** $\hat{y} = \beta_0 + \sum_j \beta_j x_j$minimizando **MSE**.

Ejemplo didáctico (CTG)

Usamos una **variable continua** de CTG como respuesta (p. ej., *histogram_width*) para ilustrar ajuste y residuales.



Regresión Logística

Idea clave

Modela $P(Y = 1 \mid \mathbf{x}) = \sigma(\beta_0 + \mathbf{x}^\top \beta)$ con **sigmoide** $\sigma(z) = 1/(1 + e^{-z})$.

1. Definición de Regresión Logística

La **Regresión Logística** es un algoritmo de aprendizaje automático supervisado utilizado fundamentalmente para problemas de **clasificación binaria**.

A pesar de su nombre, su objetivo no es predecir un valor continuo, sino modelar la **probabilidad** (P) de que una observación pertenezca a una clase específica (usualmente denotada como $Y = 1$).

El modelo toma variables de entrada (features) $x_1, \ldots, x_n$ y estima $P(Y = 1|\mathbf{x})$.

2. El Mecanismo Central del Modelo

El modelo logístico opera en dos pasos cruciales:

2.1. El Componente Lineal (Logit)

Primero, el modelo calcula una suma ponderada de las entradas, exactamente igual que en una regresión lineal. A este resultado (z) se le conoce como **logit** o, más formalmente, **log-odds**.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- β_0 es el intercepto (sesgo).
- $\beta_{1\ldots n}$ son los coeficientes (pesos) que el modelo aprende.
- El rango de salida de z es el de todos los números reales: $(-\infty, +\infty)$.

2.2. La Función Sigmoide (Logística)

Dado que una probabilidad debe estar en el rango $[0, 1]$, z no puede ser el

3. La Relación Clave: Probabilidad y Log-Odds

El concepto central que conecta el modelo lineal con la probabilidad es el **log-odds**. Esta transformación es necesaria para mapear un espacio acotado $[0, 1]$ a un espacio no acotado $[-\infty, +\infty]$.

3.1. De Probabilidad a Log-Odds

La transformación se realiza en dos pasos:

1. **Probabilidad (P)**: La probabilidad del evento.
   - Rango: $[0, 1]$
2. **Odds (Momios)**: La razón entre la probabilidad de que ocurra (P) y la de que no ocurra ($1 - P$).

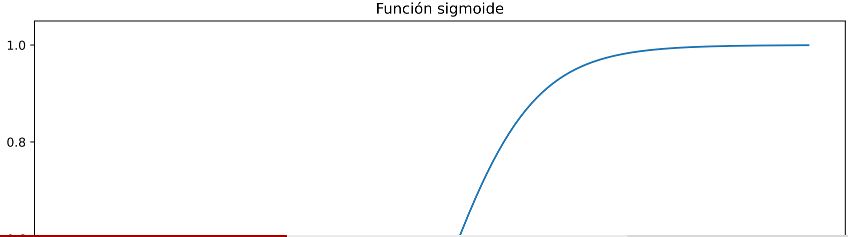
$$Odds = \frac{P}{1 - P}$$

   - Rango: $[0, +\infty]$
3. **Log-Odds (Logit)**: El logaritmo natural de los *odds*.

4. Interpretación de Coeficientes

Debido a esta relación, los coeficientes (β_i) del modelo tienen una interpretación específica:

- **Coeficiente** β_i: Un incremento de una unidad en la variable x_i (manteniendo las demás constantes) genera un cambio de β_i en el **log-odds** de la predicción.
- **Odds Ratio (OR)**: Para una interpretación más intuitiva, se utiliza e^{β_i}. Un incremento de una unidad en x_i **multiplica** los *odds* por un factor de e^{β_i}.



Sección 1

Redes Neuronales

1. El Punto de Partida: Regresión Lineal

El problema de regresión estándar busca encontrar una función f que mapee entradas $\mathbf{x}$ a una salida y.

En la **Regresión Lineal**, asumimos que la relación es *lineal*:

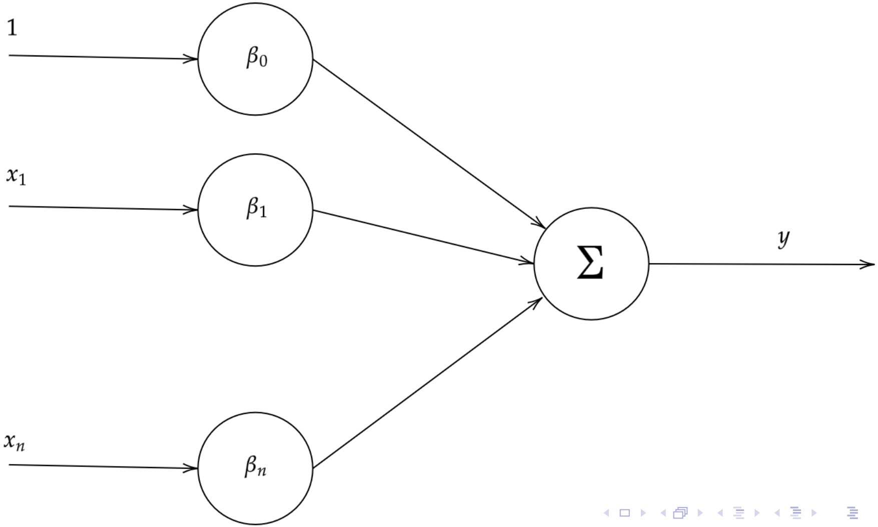
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

O en forma vectorial:

$$y = {}^T\mathbf{x} + \beta_0$$

- **Objetivo**: Encontrar los parámetros óptimos $(, \beta_0)$ que minimizan una función de pérdida (ej. Error Cuadrático Medio, MSE).
- **Limitación**: El modelo está restringido a representar únicamente relaciones lineales.

1. El Punto de Partida: Regresión Lineal



2. Primera Generalización: El GLM

¿Qué pasa si la salida no es lineal o no sigue una distribución normal? (Ej. clasificación binaria).

Usamos un **Modelo Lineal Generalizado (GLM)**, como la **Regresión Logística**:

1. **Predictor Lineal (z)**: Mantenemos el núcleo lineal.

   $$z = {}^{T}\mathbf{x} + \beta_0$$

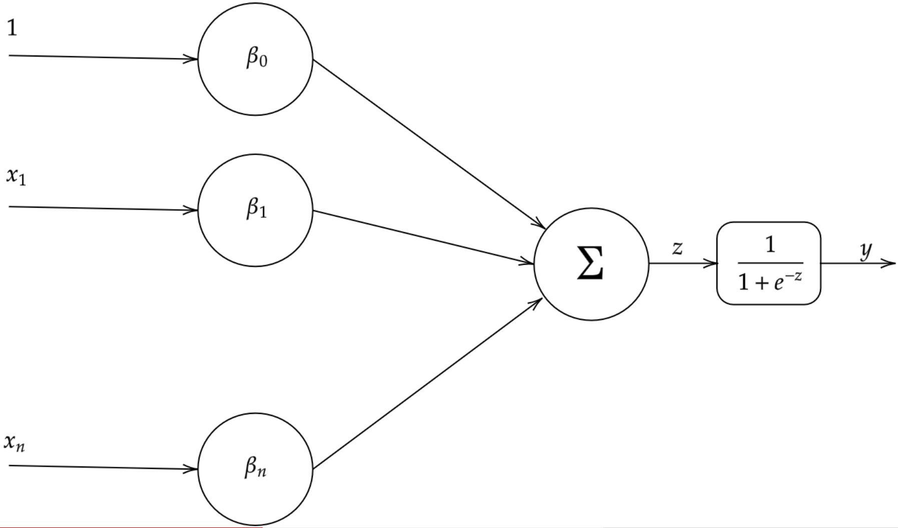
   El resultado z (el *logit*) puede ir de $(-\infty, +\infty)$.

2. **Función de Enlace (Inversa)**: Aplicamos una transformación no-lineal g^{-1} para mapear z al rango deseado (ej. $[0, 1]$ para probabilidad).

   $$\hat{y} = g^{-1}(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

- **Avance**: Hemos *generalizado* la regresión. El modelo sigue siendo lineal en

2. Primera Generalización: El GLM



3. La Neurona: Una Unidad de Regresión

Una **neurona artificial** (o Perceptrón) es conceptualmente idéntica a un modelo de regresión logística (un GLM).

1. **Agregación Lineal**: Calcula el predictor lineal z (la entrada neta).

$$z = \sum_{i=1}^{n} w_i x_i + b$$

- $g(z)$ puede ser $\text{sigmoid}(z)$, $\tanh(z)$, $\text{ReLU}(z)$, etc.
- **Concepto Clave**: Una sola neurona es una unidad de regresión (lineal o generalizada) que aprende un *único* límite de decisión (o una respuesta lineal).

3. La Neurona: Una Unidad de Regresión

Una **neurona artificial** (o Perceptrón) es conceptualmente idéntica a un modelo de regresión logística (un GLM).

1. **Agregación Lineal**: Calcula el predictor lineal z (la entrada neta).

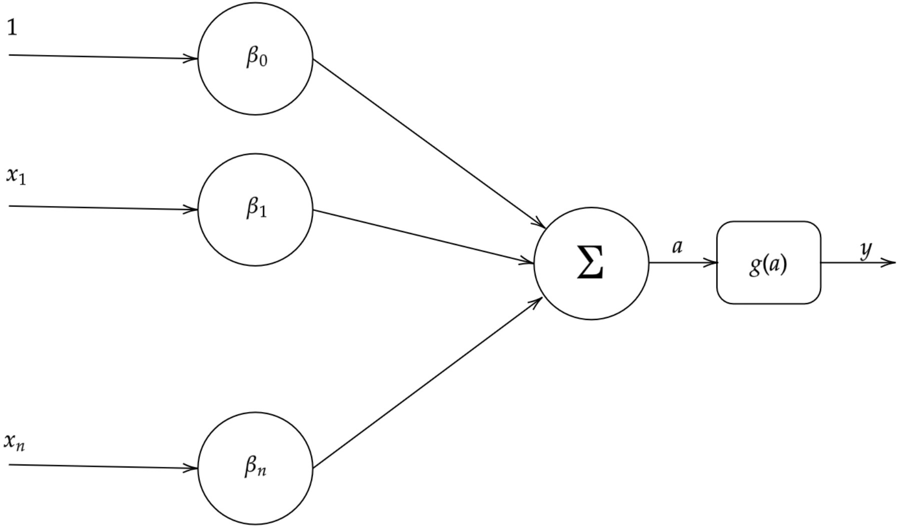
$$z = \sum_{i=1}^{n} w_i x_i + b$$

2. **Función de Activación**: Aplica una transformación no-lineal a (la salida).

$$a = g(z)$$

- $g(z)$ puede ser $\text{sigmoid}(z)$, $\tanh(z)$, $\text{ReLU}(z)$, etc.
- **Concepto Clave**: Una sola neurona es una unidad de regresión (lineal o generalizada) que aprende un *único* límite de decisión (o una respuesta lineal).

3. La Neurona: Una Unidad de Regresión



4. La Red: Composición de Funciones

¿Cómo modelar relaciones *altamente* complejas que una sola neurona no puede capturar?

Respuesta: Apilando las neuronas en capas.

La salida de una capa de “unidades de regresión” se convierte en la *entrada* de la siguiente capa.

$$\text{Capa 1 (Oculta): } \mathbf{a}^{(1)} = g_1(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$$
$$\text{Capa 2 (Salida): } \hat{y} = g_2(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)})$$

Esto es una **composición de funciones** anidada:

$$\hat{y} = g_2\left(\mathbf{W}^{(2)}\left(g_1(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})\right) + \mathbf{b}^{(2)}\right)$$

5. El Teorema de Aproximación Universal

Esta arquitectura de "regresiones apiladas" tiene una propiedad fundamental:

> ***Teorema de Aproximación Universal**: Una red neuronal feedforward con una sola capa oculta (con una función de activación no lineal, como ReLU o Sigmoide) puede aproximar cualquier función continua $f(\mathbf{x})$ a un grado arbitrario de precisión, dado un número suficiente de neuronas.*

- Una regresión lineal solo puede encontrar la *mejor línea*.
- Una red neuronal puede encontrar (aprender) la *función* $f(\mathbf{x})$ en sí misma, sin importar su forma.

La red neuronal es un regresor generalizado en el sentido más amplio.

6. Generalización del Aprendizaje (Optimización)

El proceso de "entrenamiento" también es una generalización de la regresión.

Regresión (Lineal/Logística)

- **Pérdida**: Se define una función de coste (Loss Function) $L(y, \hat{y})$.
  - *Ej. MSE*: $L = (y - \hat{y})^2$
  - *Ej. Cross-Entropy*: $L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$
- **Optimización**: Se encuentran los $\mathbf{w}$ y b que minimizan L (a menudo usando Descenso de Gradiente).

Red Neuronal

- **Pérdida**: Se define la *misma* función de coste $L(y, \hat{y})$ en la salida final.
- **Optimización**: Se encuentran *todos* los $\mathbf{W}^{(l)}$ y $\mathbf{b}^{(l)}$ de *todas* las capas que minimizan L.
- **Mecanismo**: **Descenso de Gradiente** + **Backpropagation** (Retropropagación).
  - Backpropagation es simplemente la aplicación sistemática de la **regla de la**

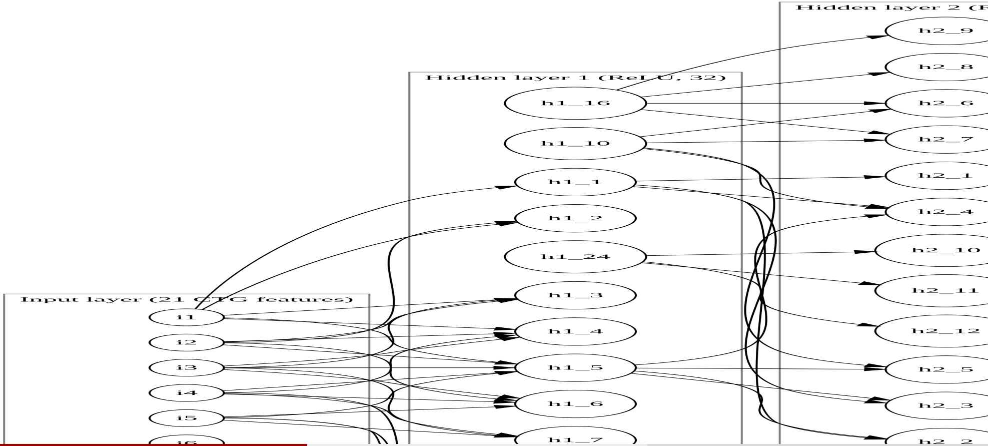
7. Conclusión: La Red como $f_\theta(\mathbf{x})$

1. **Regresión Lineal**:
   - Modelo: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$
   - Asume una forma funcional *fija* (lineal).
2. **GLM (Logística)**:
   - Modelo: $f(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x} + b)$
   - Asume una forma fija (lineal) transformada por una activación fija (ej. sigmoide).
3. **Red Neuronal (MLP)**:
   - Modelo: $f(\mathbf{x}) = f_L(\dots f_2(f_1(\mathbf{x}))\dots)$
   - **No asume una forma funcional fija.**
   - Es un **aproximador de funciones universal** cuyos parámetros θ (todos los W y b) se *aprenden* para que $f_\theta(\mathbf{x})$ imite la verdadera función subyacente $f(\mathbf{x})$ en los datos.

La red neuronal es la generalización última de la regresión: un sistema que **aprende la forma de la función** desde los datos.

Perceptrón Multicapa — MLP

Arquitectura propuesta: 21→32→16→3 con *ReLU* y **softmax** para 3 clases $(1, 2, 3)$.



Cierre y recomendaciones

- La **interpretación clínica** sigue siendo esencial; el ML complementa, no reemplaza, el juicio médico.
- Validar externamente (dominios LMIC vs HIC), sesgos de muestreo y *data shift*.
- Reportar métricas por clase, especialmente la **clase sospechosa**.

Referencias (DOI / ISBN)

- Ayres-de-Campos D, *et al.* **FIGO consensus guidelines on intrapartum fetal monitoring: Cardiotocography.** *Int J Gynaecol Obstet* 2015;131(1):13–24. DOI: 10.1016/j.ijgo.2015.06.020
- Macones GA, *et al.* **The 2008 NICHD workshop report on electronic fetal monitoring.** *J Obstet Gynecol Neonatal Nurs* 2008. (see Obstet Gynecol 2008;112:661–6)
- Hoodbhoy Z, *et al.* **Use of Machine Learning Algorithms for Prediction of Fetal Risk using Cardiotocographic Data.** *Int J Appl Basic Med Res* 2019;9:226–230. DOI: 10.4103/ijabmr.IJABMR_370_18
- James G, Witten D, Hastie T, Tibshirani R. **An Introduction to Statistical Learning (2nd ed.).** Springer, 2021. DOI: 10.1007/978-1-0716-1418-1
- Hosmer DW, Lemeshow S, Sturdivant RX. **Applied Logistic Regression (3rd ed.).** Wiley, 2013. DOI: 10.1002/9781118548387
- Rumelhart DE, Hinton GE, Williams RJ. **Learning representations by**

Apéndice · Carga de datos y EDA mínimo

```
fetal_health
1.0    1655
2.0     295
3.0     176
Name: count, dtype: int64

                                                    count  ...
baseline value                                     2126.0  ...
accelerations                                      2126.0  ...
fetal_movement                                     2126.0  ...
uterine_contractions                               2126.0  ...
light_decelerations                                2126.0  ...
severe_decelerations                               2126.0  ...
prolongued_decelerations                           2126.0  ...
abnormal_short_term_variability                    2126.0  ...
```