

Modelos de Deep Learning

Entropía Cruzada y Clasificación

Universidad ORT Uruguay

29 de Setiembre, 2025

Clasificación binaria con redes neuronales

Sea $\mathbf{T} = (\mathbf{X}, \mathbf{y})$ un dataset, con $\mathbf{y} \sim (N)$ la variable a *predecir*:

$$\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \sim (N, d)$$

$$\mathbf{y} = \{y_1, \dots, y_N\} \in \{0, 1\}^N \quad (\text{binaria})$$

Queremos construir una red neuronal tal que:

$$f(\mathbf{X}; \mathbf{W}, \mathbf{B}) = \Pr[\mathbf{y} = \mathbf{1}_N \mid \mathbf{X}; \mathbf{W}, \mathbf{B}] = \hat{\mathbf{p}}$$

Para clasificar:

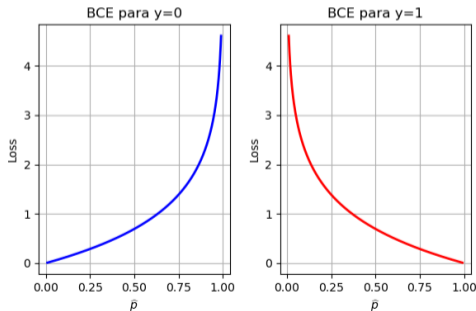
$$\hat{y}_i = \begin{cases} 1 & \text{si } \hat{p}_i \geq 0.5 \\ 0 & \text{si } \hat{p}_i < 0.5 \end{cases}$$

Binary Cross Entropy

- La función de pérdida es la **Binary Cross-Entropy (BCE)**:

$$\text{BCE}(\hat{p}, y) = \text{Loss}(\hat{p}, y) = -\left[y \ln \{\hat{p}\} + (1 - y) \ln \{1 - \hat{p}\}\right]$$

- **Motivación:** máxima verosimilitud



Verosimilitud

- **Verosimilitud** de los weights & biases para el dataset \mathbf{T} :

$$V_{\mathbf{T}}(\mathbf{W}, \mathbf{B}) = \prod_{(\mathbf{x}, y) \in \mathbf{T}} \Pr[y \mid \mathbf{x}, \mathbf{W}, \mathbf{B}]$$

- **Estimación por máxima verosimilitud:**

$$\widehat{\mathbf{W}}, \widehat{\mathbf{B}} = \arg \max_{\mathbf{W}, \mathbf{B}} V_{\mathbf{T}}(\mathbf{W}, \mathbf{B})$$

- Separando por clases:

$$V_{\mathbf{T}}(\mathbf{W}, \mathbf{B}) = \prod_{(\mathbf{x}, 1) \in \mathbf{T}} \Pr[1 \mid \mathbf{x}, \mathbf{W}, \mathbf{B}] \cdot \prod_{(\mathbf{x}, 0) \in \mathbf{T}} \Pr[0 \mid \mathbf{x}, \mathbf{W}, \mathbf{B}]$$

Menos Log-Verosimilitud

■ Aplicando (menos) logaritmo:

$$\begin{aligned}-\ln V_{\mathbf{T}}(\mathbf{W}, \mathbf{B}) &= - \sum_{(\mathbf{x}, 1) \in \mathbf{T}} \ln \Pr[1 \mid \mathbf{x}, \mathbf{W}, \mathbf{B}] + \sum_{(\mathbf{x}, 0) \in \mathbf{T}} \ln \Pr[0 \mid \mathbf{x}, \mathbf{W}, \mathbf{B}] \\&= - \sum_{(\mathbf{x}, 1) \in \mathbf{T}} \ln \hat{p}(\mathbf{x}) - \sum_{(\mathbf{x}, 0) \in \mathbf{T}} \ln(1 - \hat{p}(\mathbf{x})) \\&= - \sum_{(\mathbf{x}, 1) \in \mathbf{T}} \mathbf{1} \cdot \ln \hat{p}(\mathbf{x}) - \sum_{(\mathbf{x}, 0) \in \mathbf{T}} (1 - \mathbf{0}) \cdot \ln(1 - \hat{p}(\mathbf{x}))\end{aligned}$$

■ Reescribiendo con $y \in \{0, 1\}$:

$$-\ln V_{\mathbf{T}}(\mathbf{W}, \mathbf{B}) = - \sum_{(\mathbf{x}, y) \in \mathbf{T}} y \ln \hat{p}(\mathbf{x}) + (1 - y) \ln(1 - \hat{p}(\mathbf{x})) = \sum_{(\mathbf{x}, y) \in \mathbf{T}} \text{BCE}(\hat{p}(\mathbf{x}), y)$$

Resumen

- Minimizamos el negativo del log-likelihood:

$$\widehat{\mathbf{W}}, \widehat{\mathbf{B}} = \arg \min_{\mathbf{W}, \mathbf{B}} -\ln V_{\mathbf{T}}(\mathbf{W}, \mathbf{B})$$

- Que equivale a minimizar la **entropía cruzada**:

$$\widehat{\mathbf{W}}, \widehat{\mathbf{B}} = \arg \min_{\mathbf{W}, \mathbf{B}} - \sum_{(\mathbf{x}, y) \in \mathbf{T}} \left[y \ln \hat{p}(\mathbf{x}) + (1 - y) \ln(1 - \hat{p}(\mathbf{x})) \right]$$

- O su versión promedio:

$$\widehat{\mathbf{W}}, \widehat{\mathbf{B}} = \arg \min_{\mathbf{W}, \mathbf{B}} \mathbb{E}_{\mathbf{T}} \left[\text{BCE}(\hat{p}(\mathbf{x}), y) \right]$$

Clasificación multiclase

- Sea $\mathcal{C} = \{c_1, \dots, c_r\}$ un conjunto de clases.
- Cada clase c_i se codifica mediante un vector one-hot $\mathbf{y}_i \in \{0, 1\}^r$:

$$y_{i,k} = \begin{cases} 1 & \text{si } i = k \\ 0 & \text{en otro caso} \end{cases}$$

- Definimos la función softmax : $\mathbb{R}^r \rightarrow [0, 1]^r$

$$\text{softmax}(u)_i = \frac{e^{u_i}}{\sum_{k=1}^r e^{u_k}}$$

- **Propiedad:** $\sum_{i=1}^r \text{softmax}(u)_i = 1$

Modelo multiclase con softmax

- Sea $\mathbf{T} = (\mathbf{X}, \mathbf{Y})$ con $\mathbf{X} \sim (N, d)$ e $\mathbf{Y} \sim (N, r)$ el OHE del target.
- Una red neural para clasificación multiclase devuelve

$$\hat{\mathbf{P}} = \hat{\mathbf{P}}(\mathbf{X}; \mathbf{W}, \mathbf{B}) \sim (N, r)$$

donde cada componente $\hat{\mathbf{P}}_{i,c}$ representa:

$$\hat{p}_c(\mathbf{x}_i) = \Pr[c \mid \mathbf{x}_i; \mathbf{W}, \mathbf{B}]$$

- En general se implementa usando la función softmax:

$$\hat{\mathbf{P}} = \text{softmax}(\mathbf{Z}) \quad \mathbf{Z} = \text{última salida pre-activación (logits)}$$

- Predicción final:

$$\hat{Y} = \arg \max \left\{ \hat{\mathbf{P}}, \text{axis} = 1 \right\}$$

Ejemplo: AND como clasificación multiclase

$$\underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}}_{\mathbf{X}} \cdot \underbrace{\begin{pmatrix} -1 & 1 \\ -1 & 1 \\ 2 & -1 \end{pmatrix}}_{\mathbf{W}} = \begin{pmatrix} 2 & -1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \xrightarrow{\text{softmax}} \underbrace{\begin{pmatrix} 0.95 & 0.05 \\ 0.73 & 0.27 \\ 0.73 & 0.27 \\ 0.27 & 0.73 \end{pmatrix}}_{\hat{\mathbf{P}}} \Rightarrow \hat{\mathbf{Y}} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Categorical Cross Entropy

Definimos la función de **pérdida** para una observación:

$$\text{Loss}(\hat{\mathbf{p}}, \mathbf{y}) = - \sum_{k=1}^r y_k \ln \hat{p}_k = - \ln \hat{p}_c \quad \text{donde } y_c = 1$$