# Modelos de Deep Learning

## Entrenamiento de un MLP: GD y SGD

Universidad ORT Uruguay

1 de Setiembre, 2025

# Funciones de Pérdida

- Regresión (MSE):

$$L(\widehat{y}, y) = (\widehat{y} - y)^2$$

- Clasificación binaria (BCE): la última activación es **sigmoidea**

$$L(\widehat{y}, y) = -\Big( y \log \widehat{y} + (1 - y) \log(1 - \widehat{y}) \Big)$$

- Clasificación con $K$ clases (CCE): la última activación es **softmax**

$$L(\widehat{\mathbf{y}}, \mathbf{y}) = -\sum_{k=1}^{O} y_k \log \widehat{y}_k$$

con $\mathbf{y} \in \{0, 1\}^O$ (one-hot encoding)

# Costo Empírico

- Dada una función de pérdida $L(\widehat{y}, y)$
- Para un $(\mathbf{W}, \mathbf{b})$ dado, la costo empírico es:

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^{N} L(f(\mathbf{x}_i; \mathbf{W}, \mathbf{b}), y_i) = \frac{1}{N} \sum_{i=1}^{N} L_i$$

  donde $\widehat{\mathbf{y}}_i = f(\mathbf{x}_i; \mathbf{W}, \mathbf{b})$ es la salida de la red para la entrada $\mathbf{x}_i$.

- Entrenar la red significa encontrar el argumento mínimo de $J$ (ERM):

$$(\widehat{\mathbf{W}}, \widehat{\mathbf{b}}) = \arg\min_{(\mathbf{W}, \mathbf{b})} J(\mathbf{W}, \mathbf{b}).$$

# Entrenamiento del MLP: Gradient Descent

- Problema de optimización - Algoritmo de **descenso por gradiente**:

$$(\mathbf{W}_0, \mathbf{b}_0) = \text{inicializar random}$$
$$(\mathbf{W}_{k+1}, \mathbf{b}_{k+1}) = (\mathbf{W}_k, \mathbf{b}_k) - \eta \cdot \nabla J(\mathbf{W}_k, \mathbf{b}_k)$$

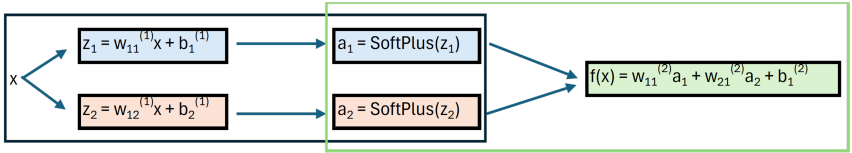
  donde $\eta$ es la **tasa de aprendizaje** (learning rate)

- Debemos calcular el gradiente de $J$ en cada paso.

- Linealidad:

$$\nabla J(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^{N} \nabla L(f(\mathbf{x}_i; \mathbf{W}, \mathbf{b}), y_i) = \underset{(\mathbf{x},y)\sim T}{\text{Promedio}} [\nabla L(f(\mathbf{x}; \mathbf{W}, \mathbf{b}), y)]$$

# Ejemplo: MLP



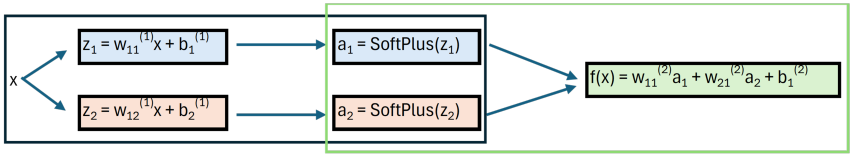

- La red es

$$\widehat{y} = \text{softplus}(x\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\,\mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

- $\mathbf{W}^{(1)} \sim (1, 2)$, $\mathbf{b}^{(1)} \sim (1, 2)$, $\mathbf{W}^{(2)} \sim (2, 1)$, $\mathbf{b}^{(2)} \sim (0)$
- El costo es

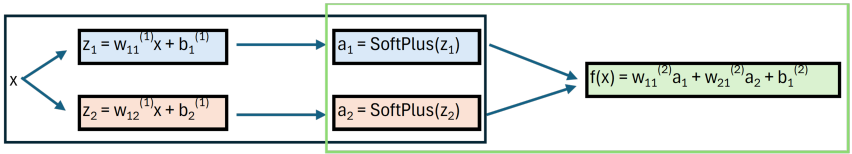
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

# Ejemplo: cálculo de derivadas




- **Objetivo**: Calcular $\nabla L(f(x), y)$ para $(x, y)$ fijo.
- $\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$
- $\frac{\partial \,\text{softplus}(z)}{\partial z} = \sigma(z)$ (sigmoidea)

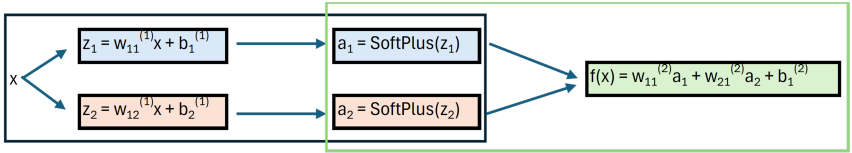
# Ejemplo: cálculo de derivadas




- $\dfrac{\partial \hat{y}}{\partial b_1^{(2)}} = 1$

- $\dfrac{\partial L}{\partial b_1^{(2)}} = \dfrac{\partial L}{\partial \hat{y}} \dfrac{\partial \hat{y}}{\partial b_1^{(2)}} \Longrightarrow \boxed{\dfrac{\partial L}{\partial b_1^{(2)}} = 2(\hat{y} - y)}$

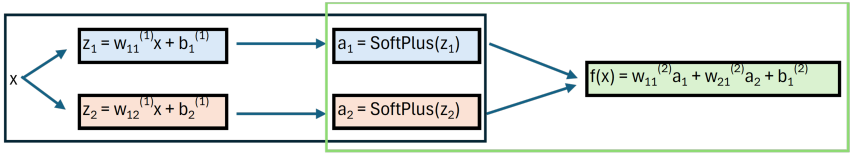
# Ejemplo: cálculo de derivadas




- $\frac{\partial L}{\partial w_{11}^{(2)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_{11}^{(2)}} \Longrightarrow \frac{\partial L}{\partial w_{11}^{(2)}} = 2(\hat{y} - y)a_1.$
- $\frac{\partial L}{\partial w_{21}^{(2)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_{21}^{(2)}} \Longrightarrow \frac{\partial L}{\partial w_{21}^{(2)}} = 2(\hat{y} - y)a_2.$
- El Jacobiano es:

$$\frac{\partial L}{\partial \mathbf{W}^{(2)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}} = 2(\hat{y} - y) \begin{bmatrix} a_1 & a_2 \end{bmatrix} = 2(\hat{y} - y)\mathbf{a}^\top$$

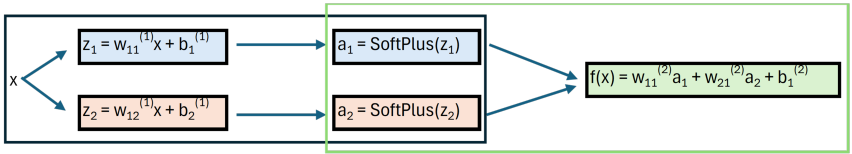
# Ejemplo: cálculo de derivadas




- $\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_1} \Longrightarrow \frac{\partial L}{\partial a_1} = 2(\hat{y} - y) w_{11}^{(2)}.$
- $\frac{\partial L}{\partial a_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \Longrightarrow \frac{\partial L}{\partial a_2} = 2(\hat{y} - y) w_{21}^{(2)}.$
- El Jacobiano es:

$$\boxed{\frac{\partial L}{\partial \mathbf{a}} = 2(\hat{y} - y) \begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} \end{bmatrix} = 2(\hat{y} - y) \left(\mathbf{W}^{(2)}\right)^{\top}}$$

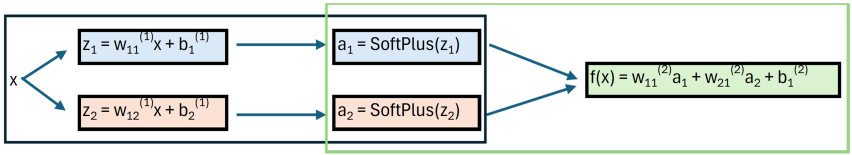
# Ejemplo: cálculo de derivadas




- $\frac{\partial L}{\partial z_1} = \frac{\partial L}{\partial a_1}\frac{\partial a_1}{\partial z_1} \Longrightarrow \frac{\partial L}{\partial z_1} = 2(\hat{y} - y)w_{11}^{(2)}\sigma(z_1).$
- $\frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial a_2}\frac{\partial a_2}{\partial z_2} \Longrightarrow \frac{\partial L}{\partial z_2} = 2(\hat{y} - y)w_{21}^{(2)}\sigma(z_2).$
- El Jacobiano es:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{z}} &= \frac{\partial L}{\partial \mathbf{a}}\frac{\partial \mathbf{a}}{\partial \mathbf{z}} = 2(\hat{y} - y)\begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} \end{bmatrix}\begin{bmatrix} \sigma(z_1) & 0 \\ 0 & \sigma(z_2) \end{bmatrix} \\ &= 2(\hat{y} - y)\left(\mathbf{W}^{(2)}\right)^{\top}\sigma(\mathbf{z})\end{aligned}$$

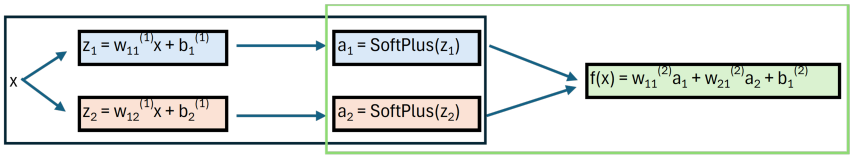
# Ejemplo: cálculo de derivadas




- $\frac{\partial L}{\partial b_1^{(1)}} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial b_1^{(1)}} \Longrightarrow \frac{\partial L}{\partial b_1^{(1)}} = 2(\hat{y} - y)w_{11}^{(2)}\sigma(z_1).$
- $\frac{\partial L}{\partial b_2^{(1)}} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial b_2^{(1)}} \Longrightarrow \frac{\partial L}{\partial b_2^2} = 2(\hat{y} - y)w_{21}^{(2)}\sigma(z_2).$
- El Jacobiano es:

$$\frac{\partial L}{\partial \mathbf{b}^{(1)}} = \frac{\partial L}{\partial \mathbf{z}} \overbrace{\frac{\partial \mathbf{z}}{\partial \mathbf{b}^{(1)}}}^{\mathbf{Id}} = 2(\hat{y} - y)\left(\mathbf{W}^{(2)}\right)^{\top}\sigma(\mathbf{z})$$

# Ejemplo: cálculo de derivadas




- $\dfrac{\partial L}{\partial w_{11}^{(1)}} = \dfrac{\partial L}{\partial z_1}\dfrac{\partial z_1}{\partial w_{11}^{(1)}} \Longrightarrow \dfrac{\partial L}{\partial w_{11}^{(1)}} = 2(\hat{y}-y)w_{11}^{(2)}\sigma(z_1)x.$
- $\dfrac{\partial L}{\partial w_{12}^{(1)}} = \dfrac{\partial L}{\partial z_2}\dfrac{\partial z_2}{\partial w_{12}^{(1)}} \Longrightarrow \dfrac{\partial L}{\partial w_{12}^{(1)}} = 2(\hat{y}-y)w_{11}^{(2)}\sigma(z_1)x.$
- El Jacobiano es:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{W}^{(1)}} &= \frac{\partial L}{\partial \mathbf{z}}\frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} = 2(\hat{y}-y)\begin{bmatrix} w_{11}^{(2)}\sigma(z_1) & w_{21}^{(2)}\sigma(z_2)\end{bmatrix}\begin{bmatrix} x & 0 \\ 0 & x\end{bmatrix} \\ &= 2(\hat{y}-y)\cdot \mathbf{W}^{(2)}\sigma(\mathbf{z})\cdot x\end{aligned}$$

# Ejemplo: Resumiendo el cálculo directo

- $\frac{\partial L}{\partial b_1^{(2)}} = 2(\hat{y} - y)$
- $\frac{\partial L}{\partial \mathbf{W}^{(2)}} = 2(\hat{y} - y)\mathbf{a}$
- $\frac{\partial L}{\partial \mathbf{a}} = 2(\hat{y} - y)\left(\mathbf{W}^{(2)}\right)^\top$
- $\frac{\partial L}{\partial \mathbf{z}} = 2(\hat{y} - y)\left(\mathbf{W}^{(2)}\right)^\top \sigma(\mathbf{z})$
- $\frac{\partial L}{\partial \mathbf{b}^{(1)}} = 2(\hat{y} - y)\left(\mathbf{W}^{(2)}\right)^\top \sigma(\mathbf{z})$
- $\frac{\partial L}{\partial \mathbf{W}^{(1)}} = 2(\hat{y} - y) \cdot \left(\mathbf{W}^{(2)}\right)^\top \sigma(\mathbf{z}) \cdot x$

# Ejemplo: operando en batch (útil para la notebook)

- $\frac{\partial J}{\partial \hat{\mathbf{y}}} = \frac{2}{N}(\hat{\mathbf{y}} - \mathbf{y})^\top \sim (1, N)$
- $\frac{\partial J}{\partial \mathbf{b}^{(2)}} = \frac{2}{N}(\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{1}_N \sim (0)$
- $\frac{\partial J}{\partial \mathbf{W}^{(2)}} = \frac{2}{N}(\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{A} \sim (1, 2)$ donde $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_N^\top \end{bmatrix}$ (!) al implementar trasponer
- $\frac{\partial J}{\partial \mathbf{b}^{(1)}} = \frac{2}{N} \mathbf{1}_N^\top \left[ \left[ (\hat{\mathbf{y}} - \mathbf{y}) \left( \mathbf{W}^{(2)} \right)^\top \right] \odot \sigma(\mathbf{Z}) \right] \sim (1, 2)$ con $\sigma(\mathbf{Z}) = \begin{bmatrix} \sigma(\mathbf{z}_1)^\top \\ \vdots \\ \sigma(\mathbf{z}_N)^\top \end{bmatrix}$
- $\frac{\partial J}{\partial \mathbf{W}^{(1)}} = \frac{2}{N} \mathbf{X}^\top \left[ \left[ (\hat{\mathbf{y}} - \mathbf{y}) \left( \mathbf{W}^{(2)} \right)^\top \right] \odot \sigma(\mathbf{Z}) \right] \sim (1, 2)$ con $\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$

# Ejemplo: volviendo a $\nabla L$ pero en reversa

- $\delta_2 = \frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$
- $\delta_1 = \frac{\partial L}{\partial \mathbf{z}} = 2(\hat{y} - y) \left(\mathbf{W}^{(2)}\right)^\top \sigma(\mathbf{z})$

De aquí se obtienen las derivadas de interés:

- En la capa 2:
  - $\frac{\partial L}{\partial \mathbf{b}^{(2)}} = \delta_2$
  - $\frac{\partial L}{\partial \mathbf{W}^{(2)}} = \delta_2 \, \mathbf{a}^\top$
- En la capa 1:
  - $\frac{\partial L}{\partial \mathbf{b}^{(1)}} = \delta_1$
  - $\frac{\partial L}{\partial \mathbf{W}^{(2)}} = \delta_1 \, x$

# Backpropagation para un MLP

**Objetivo:** Calcular $\nabla L(\widehat{y}, y)$ eficientemente para un MLP con $K$ capas.

- $\mathbf{z}_k = \mathbf{a}_{k-1}^\top \mathbf{W}_k + \mathbf{b}_k^\top$ (pre-activación)
- $\mathbf{a}_k = A(\mathbf{z}_k)$ (post-activación)
- $\mathbf{a}^0 = \mathbf{x}$

**Forward pass:**

- Para $k = 1$ hasta $K$: calcular $\mathbf{z}_k$ y $\mathbf{a}_k$

**Backward pass (errores):**

- Calcular $\delta_K = A'(\mathbf{z}_K)\nabla_{\mathbf{a}_K} L$
- Para $k = K - 1$ hasta 1: $\delta_k = \mathbf{W}_{k+1}^\top A'(\mathbf{z}_k)\delta_{k+1}$

# Backpropagation para un MLP

**Gradientes:**
Para cada capa $k = 1, \ldots, K$:

$$\nabla_{\mathbf{W}_k} L = \delta_k \cdot \mathbf{a}_{k-1}^\top$$
$$\nabla_{\mathbf{b}_k} L = \delta_k$$

**Linealidad:**

$$\nabla J = \frac{1}{N} \sum_{i=1}^{N} \nabla L_i$$

**Actualización:**

$$\mathbf{W}_k \leftarrow \mathbf{W}_k - \eta \, \nabla_{\mathbf{W}_k} J, \quad \mathbf{b}_k \leftarrow \mathbf{b}_k - \eta \, \nabla_{\mathbf{b}_k} J$$

# Entrenamiento del MLP: Stochastic Gradient Descent

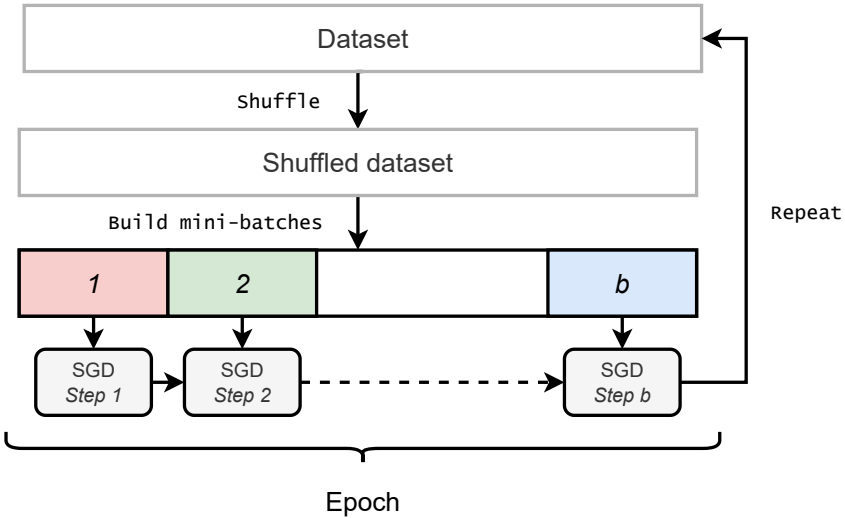
Algoritmo básico: Descenso del gradiente estocástico (SGD)

- Comenzar con $(\mathbf{W}_0, \mathbf{b}_0)$ random
- En cada step $t$: seleccionar $(x, y) \in T$ aleatoriamente y actualizar

$$(\mathbf{W}_{t+1}, \mathbf{b}_{t+1}) = (\mathbf{W}_t, \mathbf{b}_t) - \eta \nabla L(f(\mathbf{x}; \mathbf{W}_t, \mathbf{b}_t), y) \qquad \eta \text{ es el learning rate}$$

En la práctica se actualiza por **mini-batch** (se aprovecha el paralelismo)

# SGD con mini-batches



# Entrenamiento del MLP: SGD con mini-batches

- Comenzar con $(\mathbf{W}_0, \mathbf{b}_0)$ random
- En cada epoch:
  - Seleccionar el batch $B$
  - Actualizar pesos:

$$(\mathbf{W}, \mathbf{b}) \leftarrow (\mathbf{W}, \mathbf{b}) - \eta \nabla J_B(f(\mathbf{x}; \mathbf{W}, \mathbf{b}), y) \qquad \eta \text{ es el learning rate}$$

donde

$$J_B = \frac{1}{n} \sum_{(x_i, y_i) \in B} L_i$$