Apuntes de clase: Redes Neuronales con el dataset MNIST

Fabián Díaz Barboza
Estudiante Ing. Computación
Tecnológico de Costa Rica
Cartago, Costa Rica
fdiaz@estudiantec.cr
23/09/2025

1 El Dataset MNIST y la Representación de Características

1.1 Descripción del Dataset MNIST

- Imágenes en blanco y negro (1 canal).
- 10 clases (dígitos 0–9).
- Tamaño estándar: 28×28 píxeles (entrada comúnmente utilizada).
- Conjunto: 60 000 ejemplos de entrenamiento y 10 000 de prueba.

1.2 Proceso de Aplanamiento (Flattening)

Una imagen de entrada $X \in \mathbb{R}^{28\times 28}$ se convierte en un vector columna mediante *flatten*:

$$x \in \mathbb{R}^{784}, \qquad 28 \times 28 = 784.$$

Cada uno de los 784 elementos es una característica (feature) que alimenta el modelo.

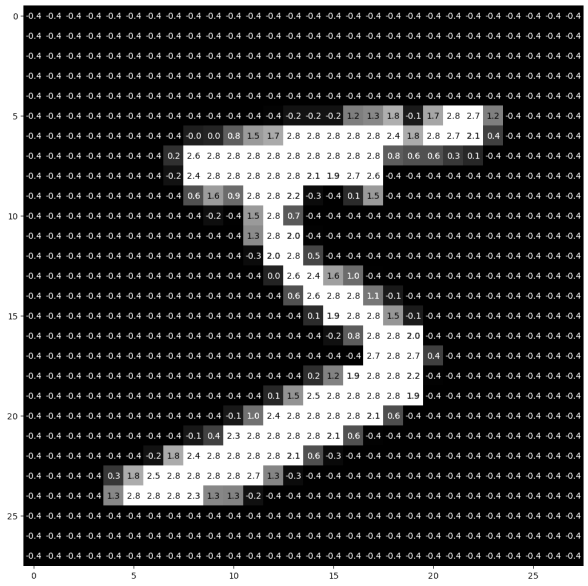

Figura 1: Ejemplo de la representación de un dígito en MNIST como matriz 28×28 y su aplanamiento a un vector de 784 características.

1.3 Píxeles Activos e Inactivos: La Semántica del Input

Un píxel con intensidad 0 se considera "apagado" y valores altos indican un píxel "encendido".

Incluso la regresión logística binaria más simple exige

$$784 \text{ pesos } (w_i) + 1 \text{ sesgo } (b) = 785 \text{ parámetros},$$

lo que muestra la complejidad del espacio de entrada.

2 La Regresión Logística Binaria: La Neurona Fundamental

2.1 Clasificación Binaria como Problema Inicial

La regresión logística estima la probabilidad de que una entrada pertenezca a la clase positiva; la salida está en $(0, 1)$.

2.2 Ecuaciones Fundamentales de la Neurona

Potencial de activación:

$$z = w^\top x + b.$$

Función sigmoide:

$$g(z) = \frac{1}{1 + e^{-z}}.$$

Salida del modelo:

$$\hat{y} = h(x) = g(w^\top x + b).$$

Figura: diagrama esquemático de la neurona (Entradas → combinación lineal → activación → salida)

Figura 2: Diagrama esquemático que interpreta la regresión logística como la neurona más simple.

3 Extensión a la Clasificación Multinomial y la Codificación One-Hot

3.1 Ejemplo de clase: 10 Regresiones Logísticas, una por alumno

Para manejar las 10 clases se puede entrenar una regresión logística por estudiante (una por clase); la capa de salida tendría 10 neuronas (una por clase).

3.2 Codificación One-Hot de las Etiquetas (y)

La etiqueta escalar se codifica como un vector one-hot en $\mathbb{R}^{10}$.

Clase (dígito)	Vector One-Hot ($y \in \mathbb{R}^{10}$)	Esperada
0	$[1,0,0,0,0,0,0,0,0,0]$	Neurona 0
2	$[0,0,1,0,0,0,0,0,0,0]$	Neurona 2
9	$[0,0,0,0,0,0,0,0,0,1]$	Neurona 9

Cuadro 1: Codificación one-hot de etiquetas (ejemplos).

4 Compactación por Álgebra Lineal

4.1 Formulación Matricial de Pesos y Sesgos

Stackeando los vectores de pesos obtenemos la matriz de pesos y el vector de sesgos:

$$W \in \mathbb{R}^{10\times 784}, \qquad b \in \mathbb{R}^{10}.$$

La combinación lineal de la capa de salida se escribe como:

$$z = Wx + b, \qquad z \in \mathbb{R}^{10}.$$

Elemento	Símbolo	Dimensión
Entrada	x	784×1
Matriz de pesos	W	10×784
Sesgos	b	10×1
Potencial de activación	z	10×1

Cuadro 2: Dimensiones en la formulación matricial para MNIST.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}$$

Figura 3: Matriz de pesos W en la capa fully connected: cada fila corresponde a una neurona de salida y cada columna a un píxel de entrada.

4.2 Ejemplo Numérico de Clase: De Vector a Matriz

V.B.1. Cálculo de una sola regresión (vector de 4 features):

$$w = \begin{bmatrix} 3 \\ 2 \\ 4 \\ 5 \end{bmatrix}, \quad b = 2, \quad x = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}.$$

$$z = w^\top x + b = (3\cdot 3) + (2\cdot 4) + (4\cdot 5) + (5\cdot 6) + 2 = 69.$$

$$\hat{y} = \sigma(z).$$

V.B.2. Cálculo de varias regresiones a la vez (2 neuronas):

$$W = \begin{bmatrix} 3 & 2 & 4 & 5 \\ 4 & 3 & 2 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad x = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}.$$

$$z = Wx + b = \begin{bmatrix} 69 \\ 43 \end{bmatrix}.$$

5 Arquitectura de las Redes Neuronales Profundas

5.1 Definición y Estructura Típica

Una **red neuronal artificial** es un modelo de cómputo inspirado en el cerebro humano, compuesto por unidades llamadas **neuronas artificiales**. Cada neurona recibe un conjunto de entradas x, aplica una combinación lineal con sus pesos w y un sesgo b, y luego pasa el resultado por una función de activación g:

$$h(x) = g(w^\top x + b).$$

- **Capa de entrada:** recibe los 784 píxeles (flatten).
- **Capas ocultas:** transforman la información en representaciones abstractas.
- **Capa de salida:** entrega la predicción (10 neuronas para MNIST).

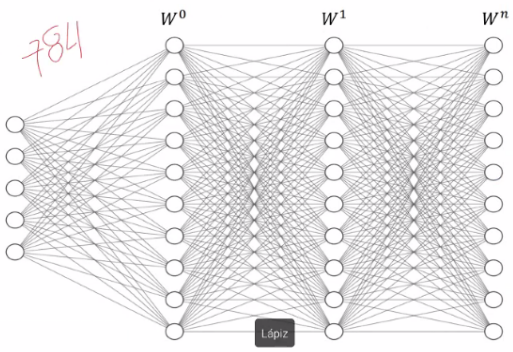

Figura 4: Ejemplo esquemático de una red neuronal con capa de entrada, capa(s) oculta(s) y capa de salida.

5.2 El Rol del Sesgo b

Retomando, el parámetro b (bias o sesgo) podriamos verlo como un **desplazamiento** en la función de activación. Sin b, todas las funciones aprendidas por la red tenderían a pasar por el origen, lo que limita la flexibilidad del modelo.

En el caso de MNIST:

- Tenemos 10 regresiones logísticas (una por cada clase).
- Cada regresión tiene un vector de pesos $w_i \in \mathbb{R}^{784}$ y un sesgo b_i.
- En conjunto, los pesos forman la matriz $W \in \mathbb{R}^{10\times 784}$ y los sesgos forman un vector $b \in \mathbb{R}^{10}$.

Es importante corregir una confusión que se habló en clase: no existe un único b de dimensión 784 por ejemplo. En cambio, hay *un sesgo por neurona de salida*. Cada componente b_i actúa como umbral independiente para la neurona i, permitiendo desplazar su función de activación y ajustar su probabilidad de disparo de forma individual.

5.3 Fully Connected (Completamente Conectadas)

Las capas **fully connected (FC)** son aquellas en las que cada neurona de una capa se conecta con todas las neuronas de la capa anterior.

En nuestro ejemplo de MNIST:

- Cada neurona de salida (de las 10) recibe conexión de los 784 píxeles de entrada.
- Cada conexión tiene su propio peso, y además cada neurona tiene su sesgo b_i.

Esta estructura convierte el modelo en un clasificador mucho más potente que una sola regresión logística binaria, porque permite:

1. Aprender múltiples fronteras de decisión en paralelo.
2. Combinar la información de todos los píxeles de forma diferenciada para cada clase.
3. Ajustar umbrales específicos gracias a los b_i.

En otras palabras, una red fully connected extiende el poder de una regresión logística binaria: al apilar capas con activaciones no lineales, las salidas de una capa se convierten en features no lineales que alimentan la siguiente, permitiendo construir clasificadores mucho más expresivos.

5.3.1 De la Multiclase al Clasificador Binario

Una arquitectura útil consiste en usar primero las 10 regresiones logísticas (capa multiclase) y luego aplicar sobre su salida un clasificador binario adicional. Por ejemplo, para determinar si la imagen corresponde al dígito "5" o no, la decisión puede tomarse a partir de las 10 salidas (o de una combinación entrenada de ellas), en lugar de hacerlo directamente sobre los píxeles. De este modo, las capas previas actúan como extraedores de características no lineales que potencian una decisión binaria final más robusta.

5.4 Propiedades Esenciales de la Red

1. **No linealidad:** las funciones de activación (sigmoide, ReLU, etc.) permiten que la red modelice relaciones no lineales entre entradas y salidas.
2. **Capas y profundidad:** a mayor profundidad, mayor capacidad para representar abstracciones jerárquicas.
3. **Diferenciabilidad:** la diferenciabilidad de las funciones internas es requisito para aplicar retropropagación y optimizar los parámetros mediante gradiente descendente.

6 Conclusiones

En conclusión de la clase, las redes neuronales son como una evolución natural de la regresión logística: partiendo de la clasificación binaria, pasando por la extensión multinomial y compactando parámetros mediante álgebra lineal, se llega a arquitecturas fully connected que permiten mayor expresividad y paralelización. La ecuación $z = Wx + b$ nos sintetiza el paso fundamental hacia la representación matricial; pero el verdadero salto en capacidad proviene de combinar esa formulación con funciones de activación no lineales y con múltiples capas diferenciables, lo que habilita la retropropagación y el entrenamiento eficiente de modelos capaces de abstraer características complejas de datos como MNIST.