Apuntes de clase: Redes Neuronales

Brandon Emmanuel Sánchez Araya
Escuela de Ingeniería en Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
brandon01sanchez@estudiantec.cr
23 Setiembre 2025

***Abstract*—Este documento presenta una formalización de apuntes de clase correspondientes al curso de Inteligencia Artificial. Se abordan los conceptos fundamentales de la regresión logística (binaria y multiclase), el uso del dataset MNIST y la representación de imágenes mediante *flatten*. Asimismo, se introduce la codificación *one–hot*, la formulación matricial con pesos y sesgos, y la relación de estos modelos con la construcción de redes neuronales. Finalmente, se destacan las propiedades esenciales de las redes, como la no linealidad, la organización en capas y su capacidad para resolver problemas complejos a través de la optimización por gradiente.**

I. El Dataset MNIST

El dataset **MNIST** (*Modified National Institute of Standards and Technology*) es uno de los conjuntos de datos más famosos en el área de aprendizaje automático. Fue creado a partir de la recopilación de miles de dígitos manuscritos provenientes de estudiantes de secundaria y empleados de la Oficina del Censo de los Estados Unidos. La idea original era disponer de un conjunto estandarizado que sirviera para probar y comparar algoritmos de reconocimiento de escritura.

- **Conjunto de imágenes**: dígitos escritos a mano (del 0 al 9).
- **Tamaño original**: 128×128 píxeles.
- **Tamaño transformado**: 28×28 pixeles.
- **Flatten**: cada imagen se convierte en un vector de 784 características.
- **1 Channel**: un solo canal, es decir en blanco y negro.
- **Cantidad de ejemplos**: 60,000 para entrenamiento y 10,000 para prueba.

En la Figura 1 se muestra un ejemplo de cómo un dígito manuscrito se representa en MNIST como una matriz de 28×28 píxeles, que luego puede convertirse en un vector de 784 características (*flatten*).

.

II. ¿Cómo diseñar un programa que reconozca todos los números que las personas pueden hacer?

A. Píxeles activos e inactivos & formación de la figura

En una imagen de MNIST, cada píxel tiene una intensidad (0 = "apagado", valores altos = "encendido"). La figura del dígito se forma por el *patrón* de píxeles activos/inactivos. El aprendizaje consiste en ajustar pesos para que ciertas configuraciones de píxeles (patrones) produzcan la clase correcta.

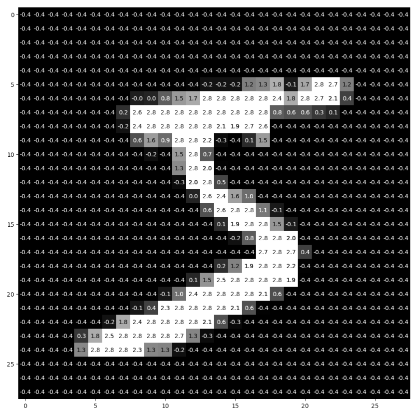

Fig. 1. Ejemplo de representación de un dígito en MNIST y sus píxeles en escala de grises.

B. Por qué esto es un problema complejo

Aunque un dígito "5" tiene una forma reconocible, cada persona lo escribe distinto. La variación en trazo, grosor, inclinación y ubicación hace que sea difícil usar reglas fijas; necesitamos un modelo que aprenda a partir de ejemplos.

C. Clasificación binaria: "¿es un 5 o no?"

La regresión logística es la base de las redes neuronales. Se usa para clasificar entre dos clases (ej: ¿es un 5 o no lo es?).

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

$$h(x) = g(f(x))$$

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

$$f_{w,b}(x) = wx + b$$

En la Figura 2 se observa cómo la regresión logística puede interpretarse como una red neuronal muy simple.

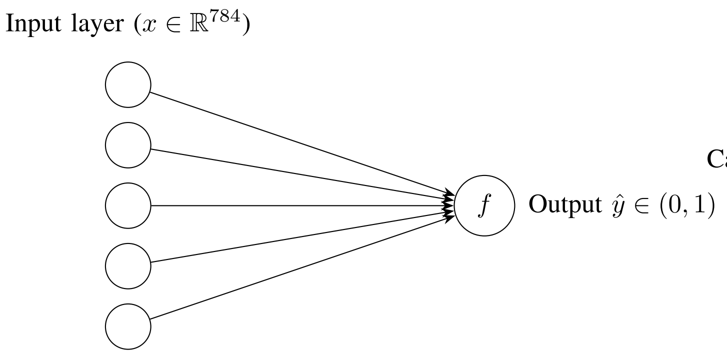

Fig. 2. Modelo de regresión logística como red neuronal: entradas $\rightarrow$ combinación lineal $\rightarrow$ función sigmoide.

D. ¿Cómo alimentar una regresión logística con una matriz*?*

Sea $X \in \mathbb{R}^{28\times 28}$ la imagen (matriz de píxeles). Se aplana (*flatten*) en un vector columna:

$$x = \mathrm{vec}(X) \in \mathbb{R}^{784}.$$

Tamaño de entrada (input layer) y conteo de parámetros

- **Input layer**: 784 *features* (un píxel por entrada).
- **Pesos en binario**: 784 pesos en w + 1 bias **= 785 parámetros** en total.

III. Regresión Logística Multinomial (experimento en clase)

Ejercicio del profe: 10 regresiones que responden "sí/no"

Se eligió a 10 estudiantes, cada uno "especialista" en un dígito (0–9). Cada imagen se le pregunta a los especialista uno por uno y ellos respondieron "sí es mi número" o "no es". Si la respuesta no coincide con la etiqueta verdadera, se hace refuerzo (entrenamiento). Como resultado se obtiene:

One–hot vector

La etiqueta correcta se codifica como un vector con un único 1 en la posición del dígito correcto:

	0	1	2	3	4	5	6	7	8	9
y (one–hot)	0	0	**1**	0	0	0	0	0	0	0

Ese **1** marca cuál estudiante (regresión) debería decir "sí".

La Figura 3 representa la extensión al caso multinomial. En este modelo, las entradas se conectan directamente con múltiples salidas, de manera que cada una corresponde a una clase distinta. De esta forma se pueden reconocer simultáneamente los diez dígitos de MNIST.

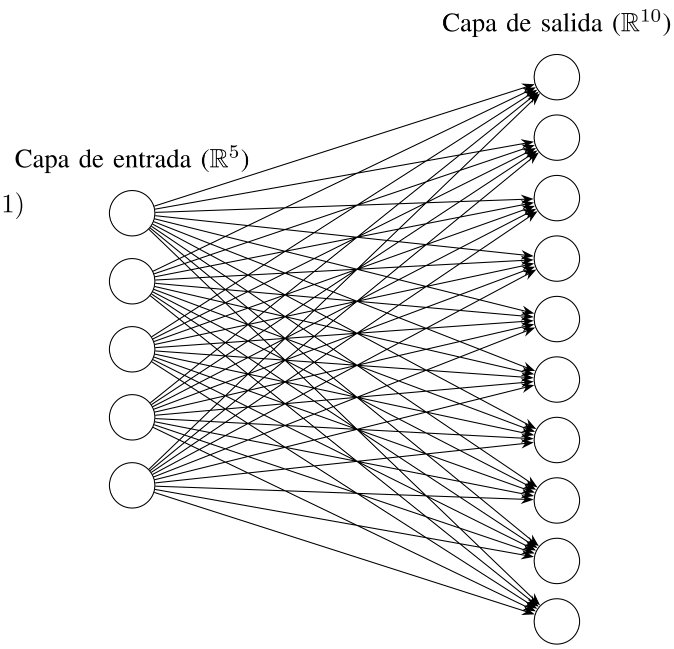

Fig. 3. Regresión logística multinomial: 5 entradas conectadas directamente con 10 salidas.

Compactación: de 10 vectores a una sola matriz

En vez de calcular 10 regresiones por separado, apilamos sus pesos en una matriz:

$$\underbrace{W}_{\in\mathbb{R}^{10\times 784}} = \begin{bmatrix} - \\ w_0^\top \\ - \\ - \\ w_1^\top \\ - \\ \vdots \\ - \\ w_9^\top \\ - \end{bmatrix}, \qquad \underbrace{b}_{\in\mathbb{R}^{10}} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_9 \end{bmatrix}, \qquad z = \underbrace{Wx + b}_{\in\mathbb{R}^{10}}.$$

Índices: $W_{j,i}$ es el peso que conecta el *feature* i (píxel i) con la neurona/clase j.

- **Mi w es una matriz** $W \in \mathbb{R}^{10\times 784}$.
- **Mi b es un vector** $b \in \mathbb{R}^{10}$ (un bias por neurona/clase).

¿Qué sucede con el parámetro $\mathbf{b}$*?*

Cada neurona/clase tiene su propio sesgo: $b = (b_0, \ldots, b_9)^\top$. **Cantidad de neuronas** $=$ tamaño de b.

IV. Ejercicio: de vector a matriz

1) Una sola regresión binaria (vector x)

Sea

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

Entonces

$$z = w^\top x + b = \begin{bmatrix} 3 & 2 & 4 & 5 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + 2 = 67 + 2 = 69, \qquad \hat{y} = \sigma(z).$$

2) Varias regresiones a la vez

Ahora dos regresiones (piensa "dos neuronas de salida"). Apilamos sus pesos en una matriz W y sus sesgos en un vector b:

$$W = \begin{bmatrix} 3 & 2 & 4 & 5 \\ 4 & 3 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 4}, \qquad b = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \in \mathbb{R}^2.$$

Con el mismo x de arriba:

$$z = Wx + b = \begin{bmatrix} 3 & 2 & 4 & 5 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 69 \\ 43 \end{bmatrix}.$$

V. Red Neuronal

Una red neuronal es un modelo matemático inspirado en el funcionamiento del cerebro humano. Está compuesta por unidades llamadas neuronas, organizadas en capas. Las capas están conectadas entre sí, de manera que la salida de una capa sirve como entrada de la siguiente.

Propiedades clave

- **No linealidad**: permite resolver problemas complejos que un modelo lineal no podría.
- **Capas**: la profundidad de la red es un hiperparámetro que define su capacidad, y en cada una de las capas hay neuronas.
- **Diferenciabilidad**: cada capa debe ser diferenciable para que podamos optimizar mediante gradiente descendente.
- **Optimización**: si puedo derivar, puedo optimizar.

Cuando aplicamos una red neuronal después de un clasificador multinomial, la lógica cambia respecto a una clasificación binaria tradicional. En una clasificación binaria simple, la relación es lineal entre las entradas (features) y la salida. En una red neuronal, la salida ya no depende directamente de la imagen original, sino de las activaciones de la capa anterior.

Estructura típica

- **Capa de entrada**: es la que recibe directamente los datos del problema. Cada neurona de esta capa representa una característica (*feature*) de la entrada. Por ejemplo, en el caso de MNIST cada píxel de la imagen se convierte en una neurona de la capa de entrada.
- **Capas intermedias (ocultas)**: son las que procesan la información recibida. Aquí la red va combinando y transformando los datos para encontrar patrones más abstractos. Se llaman "ocultas" porque no interactúan con el mundo exterior: solo comunican información entre la entrada y la salida.
- **Capa de salida**: es la que entrega el resultado final del modelo. Dependiendo del problema, puede ser una sola neurona (para decidir entre dos clases, por ejemplo "sí" o "no") o varias neuronas (para elegir entre múltiples categorías, como los 10 dígitos en MNIST).

Profundidad y complejidad

Entre más capas profundas tenga la red, más puede "desmenuzar" el problema en representaciones intermedias, lo que le permite identificar patrones complejos que una simple regresión logística no podría capturar.

En la Figura 4 se muestra un ejemplo de red neuronal con tres entradas, una capa oculta de cinco neuronas y cuatro salidas. Este esquema ilustra cómo la introducción de capas intermedias permite transformar las representaciones y capturar relaciones no lineales más complejas en los datos.

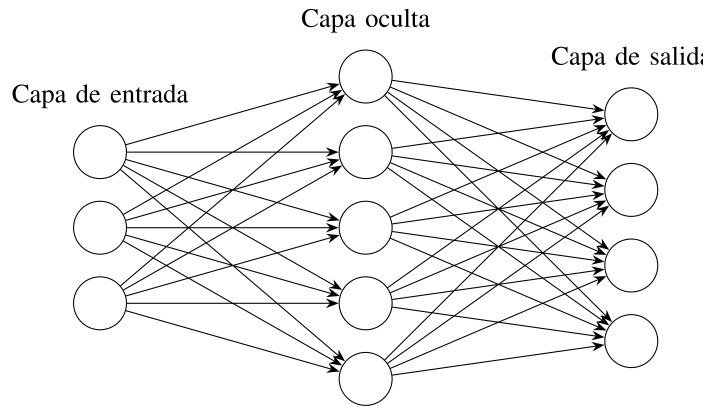

Fig. 4. Red neuronal pequeña: 3 entradas, 1 capa oculta de 5 neuronas y 4 salidas.

VI. Conclusiones

El estudio de la regresión logística permite comprender los cimientos de las redes neuronales modernas. A partir de problemas de clasificación binaria simples se llega de manera natural a la extensión multiclase, donde se introducen la formulación matricial y la codificación *one–hot*. Estos elementos muestran cómo múltiples regresiones pueden integrarse en un solo modelo más general.

El concepto de red neuronal surge al conectar varias de estas operaciones en capas sucesivas, incorporando funciones de activación no lineales que amplían la capacidad de representación. La diferenciabilidad de cada capa asegura la posibilidad de entrenar el modelo mediante optimización, mientras que la profundidad incrementa su habilidad para capturar patrones complejos. En síntesis, las redes neuronales son una evolución directa de la regresión logística, potenciadas por la organización en capas y la introducción de no linealidad.