

# Congreso

## 1 Introducción

En este trabajo se investiga la relación entre la **energía del grafo asociado a una red neuronal** y la **función de pérdida** durante el proceso de entrenamiento. La energía del grafo es una medida basada en los valores propios de ciertas matrices asociadas a la red neuronal, y su relación con la optimización del modelo es un tema de creciente interés en la comunidad científica.

El estudio se realiza utilizando la base de datos **MNIST**, un conjunto de imágenes de dígitos escritos a mano ampliamente utilizado en el entrenamiento y evaluación de modelos de reconocimiento de patrones. La red neuronal utilizada es una **red densa totalmente conectada** con tres capas ocultas y funciones de activación ReLU.

Para analizar la relación entre la pérdida y la energía del grafo, se implementa la siguiente metodología:

- Se define la **energía del grafo** a partir de diferentes matrices asociadas a la red neuronal: la matriz de adyacencia, la matriz bipartita y la matriz laplaciana.
- Se entrena la red neuronal en la base de datos MNIST, registrando la evolución de la función de pérdida y la energía de las matrices en cada época.
- Se realiza un análisis estadístico mediante **regresión lineal**, **regresión polinómica** y **cálculo del coeficiente de correlación de Pearson** para determinar si existe una relación significativa entre ambas magnitudes.

El objetivo principal de este estudio es evaluar si la energía del grafo puede servir como una métrica alternativa o complementaria para entender el comportamiento del entrenamiento de redes neuronales, proporcionando una nueva perspectiva sobre la dinámica de aprendizaje de estos modelos.

## 2 MNIST

La base de datos MNIST (Modified National Institute of Standards and Technology) es un conjunto de datos ampliamente utilizado en aprendizaje automático y visión por computadora. Consiste en 70,000 imágenes en escala de grises de dígitos escritos a mano (del 0 al 9), cada una con un tamaño de  $28 \times 28$  píxeles. El conjunto de datos está dividido en 60,000 imágenes de entrenamiento y 10,000 de prueba.

## 3 Grafos

Un grafo es una tupla  $G = (V, E)$ , donde  $V$  es un conjunto no vacío de elementos llamados vértices (o nodos) y  $E$  es un conjunto de pares de vértices, llamados aristas.

Un grafo ponderado dirigido es una tripleta  $G = (V, E, \omega)$ , donde  $\omega$  es una función  $\omega : E \rightarrow \mathbb{R}$  y  $E \subset V \times V$ .

Sea  $G$  un grafo, donde  $V(G) = \{v_1, \dots, v_n\}$ . La matriz de adyacencia de  $G$  es una matriz cuadrada  $A(G) = [a_{ij}]$  de  $n \times n$  definida por

$$a_{ij} = \begin{cases} 1, & \text{si } (v_i, v_j) \in E \\ 0, & \text{en caso contrario} \end{cases}$$

Si  $G$  es un grafo ponderado, su matriz de adyacencia ponderada está dada por

$$a_{ij} = \begin{cases} \omega(v_i, v_j), & \text{si } (v_i, v_j) \in E \\ 0, & \text{en caso contrario} \end{cases}$$

## 4 Estructura de la Red Neuronal

La red neuronal implementada en el código es un **Perceptrón Multicapa (MLP, Multi-Layer Perceptron)** con **neuronas completamente conectadas** (fully connected neurons). La arquitectura de la red se describe de la siguiente manera:

- **Capa de entrada:** Recibe imágenes de  $28 \times 28$  píxeles, lo que se traduce en un vector de 784 entradas.
- **Primera capa oculta:** 128 neuronas con activación **ReLU**.
- **Segunda capa oculta:** 64 neuronas con activación **ReLU**.
- **Capa de salida:** 10 neuronas (una por cada dígito del 0 al 9).

Cada capa oculta usa la función de activación **ReLU (Rectified Linear Unit)**, definida como:

$$\text{ReLU}(x) = \max(0, x) \quad (1)$$

La capa de salida no tiene activación explícita, ya que la función de pérdida utilizada es **CrossEntropyLoss**, que incorpora la activación **softmax** implícitamente.

#### 4.1 Grafo asociado a la Red Neuronal

Esta red se puede representar como un **grafo dirigido y ponderado**  $G = (V, E, W)$ , donde:

$$V = V_1 \cup V_2 \cup V_3 \cup V_4$$

$$\begin{aligned} V_1 &= \{v_{1,1}, v_{1,2}, \dots, v_{1,784}\}, & (\text{capa de entrada}) \\ V_2 &= \{v_{2,1}, v_{2,2}, \dots, v_{2,128}\}, & (\text{primera capa oculta}) \\ V_3 &= \{v_{3,1}, v_{3,2}, \dots, v_{3,64}\}, & (\text{segunda capa oculta}) \\ V_4 &= \{v_{4,1}, v_{4,2}, \dots, v_{4,10}\}, & (\text{capa de salida}) \end{aligned}$$

El conjunto de aristas  $E$  es:

$$E = E_1 \cup E_2 \cup E_3$$

$$\begin{aligned} E_1 &= \{(v_{1,i}, v_{2,j}) \mid 1 \leq i \leq 784, 1 \leq j \leq 128\}, & (\text{capa de entrada a primera oculta}) \\ E_2 &= \{(v_{2,i}, v_{3,j}) \mid 1 \leq i \leq 128, 1 \leq j \leq 64\}, & (\text{primera capa oculta a segunda oculta}) \\ E_3 &= \{(v_{3,i}, v_{4,j}) \mid 1 \leq i \leq 64, 1 \leq j \leq 10\}, & (\text{segunda capa oculta a salida}) \end{aligned}$$

La función de pesos  $W$  asigna valores a las conexiones entre neuronas:

$$W(v_{k,i}, v_{k+1,j}) = w_{k,ij}$$

donde  $w_{k,ij}$  representa el peso de la conexión entre la neurona  $i$  en la capa  $k$  y la neurona  $j$  en la capa  $k + 1$ .

## 5 Energía

Sea  $G = (V, E)$  un grafo con matriz de adyacencia  $A(G)$ . Decimos que el polinomio característico de  $G$  es el polinomio característico de su matriz de adyacencia  $A(G)$  dado por  $\det(xI_n - A(G))$ , donde  $I_n$  es la matriz identidad de orden  $n$ . Denotamos al polinomio característico de  $G$  con  $\phi(G, x)$ . Los autovalores del grafo  $G$  son los autovalores de su matriz de adyacencia  $A(G)$ . Dado que  $A(G)$  es una matriz simétrica, sus autovalores son reales. Llamamos espectro de  $G$  al conjunto de autovalores del grafo y lo denotamos por  $\text{Spec}(G)$ .

$$\text{Spec}(G) = \{\lambda \in \mathbb{R} : \phi(G, \lambda) = 0\}$$

Sea  $G$  un grafo con  $n$  vertices. Definimos la energia de  $G$  como

$$\mathcal{E} = \mathcal{E}(G) = \sum_{\lambda \in \text{Spec}(G)} |\lambda|$$

El grado de un vértice  $v$  es el número de aristas conectadas a él y lo denotamos  $\deg(v)$ . Dado un grafo  $G$  con  $V(G) = \{v_1, \dots, v_n\}$  definimos la matriz de grados de  $G$ ,  $\Delta(G)$ , como la matriz diagonal con entradas

$$\delta_{ii} = \deg(v_i)$$

La Matriz Laplaciana de  $G$ , denotada  $L(G)$ , está definida como

$$L(G) = \Delta(G) - A(G)$$

Sea  $G$  un grafo con  $n$  vertices,  $m$  aristas y  $S = \text{Spec}(L(G) - \frac{2m}{n}I_n)$ . La energía Laplaciana de  $G$ , denotada  $\mathcal{E}_L$ , está dada por

$$\mathcal{E}_L = \sum_{\lambda \in S} |\lambda|$$

Una **red neuronal** es un modelo computacional inspirado en el cerebro humano, utilizado en aprendizaje automático e inteligencia artificial para reconocer patrones y tomar decisiones. Consiste en capas de unidades interconectadas llamadas **neuronas**, que procesan datos de entrada y transmiten información a través de conexiones ponderadas.

## 6 Metodología

Con el objetivo de determinar la existencia de una correlación entre la **energía del grafo asociado a una red neuronal** y la **función de pérdida** durante el entrenamiento de la red, empleamos herramientas estadísticas como regresión lineal, regresión polinómica y análisis de correlación.

### 6.1 Definición de Variables

Se consideran las siguientes variables:

- $L$ : Valor de la función de pérdida de la red neuronal después de cada época de entrenamiento.
- $E$ : Energía de la red neuronal, calculada a partir de diferentes matrices asociadas (adyacencia, bipartita y laplaciana).

Dado un conjunto de valores empíricos  $\{(L_i, E_i)\}_{i=1}^n$ , se procede con el análisis de correlación.

## 6.2 Análisis

El procedimiento de análisis comprende cuatro enfoques:

### 6.2.1 Visualización mediante Gráficos

Se grafica la relación entre  $L$  y  $E$  para observar posibles tendencias o patrones.

Se genera un gráfico bidimensional de dispersión con:

Eje  $x : L$  (función de pérdida), Eje  $y : E$  (energía del grafo)

### 6.2.2 Regresión Lineal

Se ajusta un modelo de regresión lineal de la forma:

$$E = \alpha L + \beta + \varepsilon$$

donde:

- $\alpha$  es la pendiente de la regresión.
- $\beta$  es la intersección con el eje  $y$ .
- $\varepsilon$  representa el término de error.

El coeficiente de determinación  $R^2$  se calcula como:

$$R^2 = 1 - \frac{\sum (E_i - \hat{E}_i)^2}{\sum (E_i - \bar{E})^2}$$

donde  $\hat{E}_i$  son los valores predichos y  $\bar{E}$  es el promedio de  $E$ .

### 6.2.3 Regresión Polinómica

Para capturar relaciones no lineales, se ajusta un modelo polinómico de grado  $d$ :

$$E = a_d L^d + a_{d-1} L^{d-1} + \dots + a_1 L + a_0 + \varepsilon$$

El ajuste se realiza usando mínimos cuadrados y se visualiza el polinomio ajustado.

### 6.2.4 Análisis de Correlación

Se calcula el coeficiente de correlación de Pearson:

$$\rho = \frac{\sum (L_i - \bar{L})(E_i - \bar{E})}{\sqrt{\sum (L_i - \bar{L})^2} \sqrt{\sum (E_i - \bar{E})^2}}$$

donde:

- $\rho$  mide la relación lineal entre  $L$  y  $E$ .
- $\rho \approx 1$  indica correlación positiva fuerte.
- $\rho \approx -1$  indica correlación negativa fuerte.
- $\rho \approx 0$  indica ausencia de correlación lineal.

## 7 Conclusión

En resumen, la red neuronal utilizada en el código es un Perceptrón Multi-capas con neuronas completamente conectadas y activaciones ReLU. Además, el código explora la relación entre la pérdida del modelo y la energía de los grafos asociados a la red, proporcionando un análisis espectral que puede ayudar a comprender el comportamiento del entrenamiento.