



# Tarea 1

## Fundamentos de Redes Neuronales Profundas

Profesor: Fernando Crema Garcia

Fecha: 06 de Junio de 2025

La nota final será basada el promedio ponderado de cada tarea (70 %) y 30 % del proyecto final. La nota de los informes será multiplicada por un **extra** ( $\alpha \in [0,8,1,2]$ ) dependiendo de la calidad del reporte asociado.

Tarea 1	Tarea 2	Tarea 3	Tarea 4
Fundamentos	Componentes	Arquitecturas	Optimizando
30 %	30 %	30 %	10 %

### Fecha de entrega soft:

21/06/2025 23:59 pm Caracas.

### Fecha de entrega todos:

01/08/2025 23:59 pm Caracas.

### Penalizaciones:

2 pts / día. Máximo 10 pts.

Luego del 05/08/2025 tienen 0.

### Nota final:

$$\min \{ \alpha (0,3T1 + 0,3T2 + 0,3T3 + 0,1T4) , 20 \}$$

- **Notebook 6.3** - Stochastic gradient descent
- **Notebook 6.4** - Momentum
- **Notebook 6.5** - Adam
- **Notebook 7.1** - Backpropagation in toy model
- **Notebook 7.2** - Backpropagation
- **Notebook 7.3** - Initialization

2. Integrar la pregunta en la sección Evaluación a los notebooks 7.2 y 7.3. Verifique que los resultados obtenidos son los mismos a los del notebook.

### Objetivos específicos

Aunado a la resolución de los notebooks<sup>1</sup> se desea la elaboración de un informe de no más de 4 páginas donde resuelvan dos preguntas prácticas usando solamente el código generado dentro de sus notebooks.

1. Documentar el proceso de resolución de los notebooks en un Jupyter Notebook explicativo, incluyendo reflexiones personales sobre los conceptos fundamentales cubiertos y cómo se relacionan entre sí a lo largo del flujo de aprendizaje profundo.
2. Diseñar un experimento final donde se integre lo aprendido: entrenar un modelo simple utilizando los principios vistos (activaciones, arquitectura, función de pérdida, optimización, etc.) y almacenarlo como modelo óptimo.

## I. INTRODUCCIÓN

### I-A. Objetivos

El objetivo de esta actividad es consolidar los conocimientos fundamentales sobre redes neuronales profundas a través de la resolución de una serie de notebooks del libro Understanding Deep Learning de Simon Prince. Los notebooks seleccionados cubren aspectos clave del aprendizaje profundo, desde funciones de activación y diseño de redes profundas hasta funciones de pérdida, algoritmos de optimización y retropropagación (backpropagation).

#### Objetivos específicos

1. Resolver los siguientes notebooks del libro UDL de Simon Prince, utilizando Jupyter/Colab:
  - **Notebook 3.4** - Activation functions
  - **Notebook 4.1** - Composing networks
  - **Notebook 4.2** - Clipping functions
  - **Notebook 4.3** - Deep networks
  - **Notebook 5.1** - Least squares loss
  - **Notebook 5.2** - Binary cross-entropy loss
  - **Notebook 5.3** - Multiclass cross-entropy loss
  - **Notebook 6.2** - Gradient descent

## II. EVALUACIÓN

### II-A. Backpropagation

Considere la red neuronal de la Figura 1.

Cada neurona oculta utiliza una función de activación sigmoide:

$$\sigma(t) = \frac{1}{1 + \exp(-t)}.$$

La función de pérdida es el *error cuadrático medio* (MSE), y los puntos de datos son:

$$(x_1, x_2, y)_i, \quad i = 1, 2, 3 = \{(1, 1, 1), (0, 2, 1), (-1, 1, 0)\}.$$

Las unidades de entrada son negras, los *biases* son verdes, las unidades ocultas son naranjas, y la unidad de salida es roja. Los pesos están indicados en las flechas correspondientes.

## II-B. Extra Postgrado

Escriba la expresión de la ecuación 7.25 línea 3 para las siguientes funciones de activación:

- TanH
- Selu
- LeakyRelu

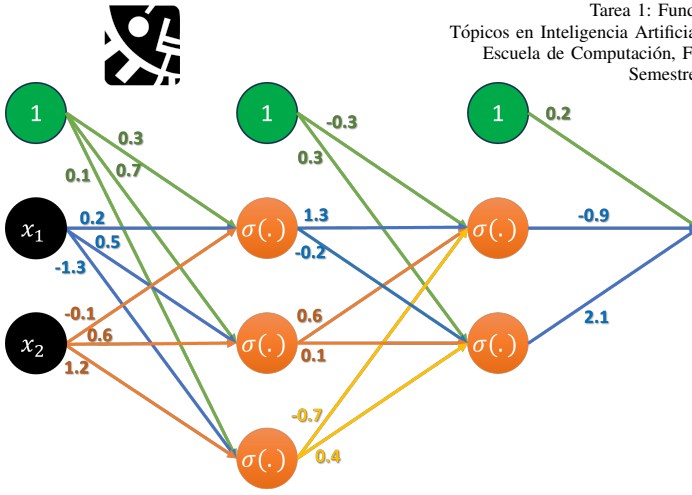


Figura 1. Red Neuronal  $f(x) = y$  con  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  los componentes de los sesgos en color verde, la de las neuronas ocultas en color naranja. Hay dos capas ocultas 0 y 1. Hay una sola salida en color rojo.

- Calcula analíticamente el gradiente mediante **backpropagation**, desarrollando todos los pasos correspondientes.
- Identifica todas las componentes explicadas en la sección 7.4 ecuación 7.24 en el caso del paso hacia adelante.

$$\mathbf{f}_0 = \beta_0 + \Omega_0 \mathbf{x}_i$$

$$\mathbf{h}_k = \mathbf{a}[\mathbf{f}_{k-1}] \quad k \in \{1, 2, \dots, K\}$$

$$\mathbf{f}_k = \beta_k + \Omega_k \mathbf{h}_k. \quad k \in \{1, 2, \dots, K\}$$

- Modifique la ecuación 7.25 para la nueva función de activación

$$\frac{\partial \ell_i}{\partial \beta_k} = \frac{\partial \ell_i}{\partial \mathbf{f}_k} \quad k \in \{K, K-1, \dots, 1\}$$

$$\frac{\partial \ell_i}{\partial \Omega_k} = \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T \quad k \in \{K, K-1, \dots, 1\}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} = \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left( \Omega_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), \quad k \in \{K, K-1, \dots, 1\}$$

- Identifica todas las componentes explicadas en la sección 7.4 de la ecuación modificada 7.25 en el caso del paso hacia atrás
- Luego, realiza un paso de descenso de gradiente (batch, simple) con **tasa de aprendizaje igual a 1**.
- ¿Cuáles son los nuevos pesos?

**Nota:** No es necesario usar una computadora. Todo puede resolverse con lápiz y papel.

**Sugerencia:**

$$\frac{d}{dt} \sigma(t) = \sigma(t)(1 - \sigma(t)).$$

Verificar que la nueva pérdida efectivamente ha disminuido también es una buena pista.



- Intente analizar la diferencia entre épocas y el tiempo de convergencia de los algoritmos.

### II-C. Algoritmos de optimización

El objetivo en este ejercicio es comprender mejor las especificidades, ventajas e inconvenientes de (algunos de) los algoritmos de optimización basados en gradientes disponibles para entrenar redes neuronales. Se proporciona una implementación en pytorch como material suplementario para jugar con ella.

### II-D. Extra Postgrado

En el caso de los alumnos de postgrado, elabore una página extra en el informe donde explique usando la notación del libro UDL los siguientes algoritmos de optimización

- AdamW
- AdaDelta
- RMSDrop
- LBFGS

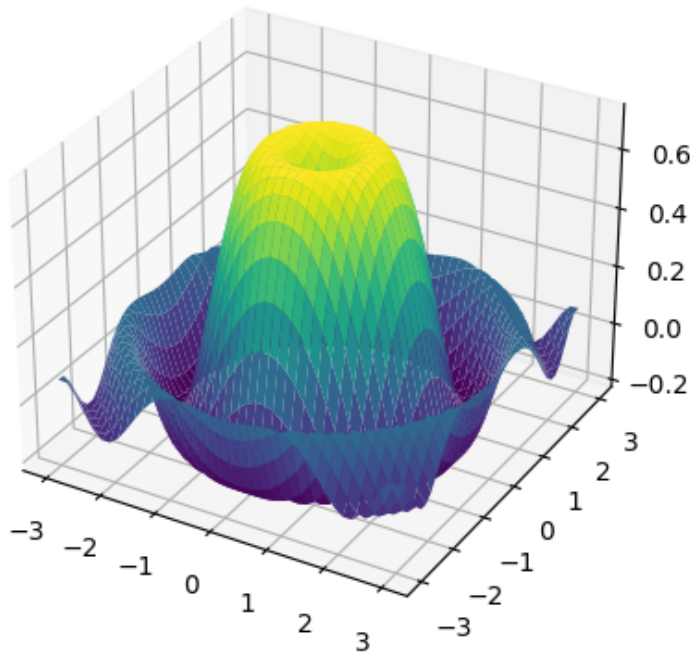


Figura 2. Superficie generada por la función  $f(x, y)$

Considere la siguiente superficie de la figura 2 descrita de la siguiente manera

$$f(x, y) = \frac{\sin(0.8(x^2 + y^2))}{(x^2 + y^2)^{0.9}}$$

El conjunto de datos consiste en puntos muestreados en esta superficie con o sin ruido, a partir de una cuadrícula regular en el plano bidimensional.

Para responder a las siguientes preguntas, puedes utilizar el código proporcionado en el cuaderno y modificarlo para adaptarlo a tus necesidades. Tendrás que generar gráficos para apoyar tus conclusiones y agregarlas en el informe.

- Cuál es el impacto del parámetro ‘ruido’ en el proceso de optimización?
- Cómo se diferencia el método del descenso del gradiente con el descenso estocástico y las versiones aceleradas?
- Cómo impacta la dimensión de la red neuronal en el rendimiento del optimizador?