

# Investigación – Redes Neuronales

---

INTELIGENCIA ARTIFICIAL II

José Pablo Martinez Salazar - 174052  
UPSLP | OTOÑO 2024

## **¿Qué es Tensorflow?**

TensorFlow es una biblioteca de código abierto desarrollada por Google que se utiliza para el aprendizaje automático y la inteligencia artificial. Se ha convertido en una de las herramientas más populares para construir y entrenar modelos de machine learning y deep learning.

TensorFlow permite la construcción de redes neuronales complejas y puede ser utilizado para una amplia gama de tareas, desde la clasificación de imágenes hasta el procesamiento de lenguaje natural y la predicción de series temporales. Es altamente escalable y puede ser utilizado para entrenar modelos pequeños en una sola máquina o modelos muy grandes distribuidos en múltiples máquinas.

## **¿Qué es Keras?**

Keras es una biblioteca de alto nivel para el desarrollo de redes neuronales, diseñada para ser fácil de usar, modular y extensible. Forma parte del ecosistema de TensorFlow, lo que significa que se puede utilizar como una API dentro de TensorFlow para facilitar la creación y entrenamiento de modelos de aprendizaje profundo.

Keras está construida sobre módulos independientes que se pueden combinar fácilmente, incluyendo módulos para capas neuronales, funciones de pérdida, optimizadores, funciones de activación y mucho más.

Está diseñada con la simplicidad en mente, permitiendo a los desarrolladores construir y entrenar redes neuronales complejas con pocas líneas de código. Su interfaz es intuitiva y está orientada a facilitar el trabajo con modelos de deep learning.

## **Para que se utilizan las siguientes funciones de activación:**

### **Sigmoid (sigmoide)**

La función Sigmoid se utiliza principalmente en la última capa de las redes neuronales para problemas de clasificación binaria, ya que convierte cualquier valor en una probabilidad (entre 0 y 1).

### **Tanh (tangente hiperbólica)**

Similar a Sigmoid, pero generalmente se utiliza cuando los datos están centrados alrededor de cero. Es común en capas ocultas de redes neuronales. Proporciona un rango de salida entre -1 y 1, lo cual hace que los valores de activación sean más centrados alrededor de cero, ayudando al aprendizaje más eficiente en ciertas arquitecturas.

## **ReLu (rectified lineal unit)**

Es la función de activación más común en las capas ocultas de redes neuronales profundas debido a su simplicidad y eficiencia computacional.

## **Leaky ReLu**

Similar a ReLU, pero con un pequeño gradiente para entradas negativas, lo que reduce la probabilidad de que las neuronas se "mueran".

## **Softmax**

Utilizado principalmente en la última capa de redes neuronales para tareas de clasificación multiclase, ya que convierte los valores de salida en probabilidades normalizadas que suman 1.

## **¿Qué hace un optimizador en Keras?**

En Keras, un optimizador es un algoritmo utilizado para ajustar los pesos de las redes neuronales durante el entrenamiento con el objetivo de minimizar la función de pérdida. Los optimizadores juegan un papel crucial en el proceso de aprendizaje de los modelos, ya que determinan cómo los pesos y sesgos de la red se ajustan en respuesta al error (diferencia entre las predicciones del modelo y los valores reales).

El objetivo principal del optimizador es minimizar la función de pérdida o error del modelo. Esto se logra ajustando los pesos de las neuronas para reducir la diferencia entre las predicciones del modelo y los valores reales.

## **Referencias**

*Keras. (n.d.). TensorFlow. Retrieved September 3, 2024, from <https://www.tensorflow.org/guide/keras?hl=es-419>*

*MLPRegressor. (n.d.). Scikit-Learn. Retrieved September 3, 2024, from [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPRegressor.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html)*

*TensorFlow Core. (n.d.). TensorFlow. Retrieved September 3, 2024, from <https://www.tensorflow.org/tutorials?hl=es-419>*

*(N.d.). Datacamp.com. Retrieved September 3, 2024, from <https://www.datacamp.com/es/tutorial/introduction-to-activation-functions-in-neural-networks>*