

---

## RESULTADO DE APRENDIZAJE

---

### RdA de la asignatura:

- **RdA 2:** Aplicar modelos de aprendizaje automático supervisado y no supervisado, así como su validación y optimización, en la resolución de problemas tanto reales como simulados.

### RdA de la actividad:

- Comprender las funciones de combinación, activación y pérdida del perceptrón sigmoide y multiclase.
- Derivar los gradientes asociados a las funciones de pérdida de cada perceptrón.
- Implementar en Python los algoritmos del perceptrón sigmoide y multiclase.

---

## INTRODUCCIÓN

---

**Pregunta inicial:** ¿Qué pasa si le pedimos a una máquina que nos diga si una imagen contiene un gato, un perro o un ave? ¿Cómo lo decide?

---

## DESARROLLO

---

### Actividad 1: Perceptrón Sigmoide

**Descripción:** En esta actividad, se explorará el funcionamiento del perceptrón sigmoide, abordando sus funciones de combinación, activación y pérdida. Se derivará el gradiente necesario para el entrenamiento mediante la metodología de clase magistral, combinada con aprendizaje práctico para la implementación.

### ¿Cómo lo haremos?

- **Clase magistral:** Se explicará la teoría que sustenta el perceptrón sigmoide:
  - Funciones de combinación y activación (sigmoide).
  - Pérdida por entropía cruzada binaria.
  - Derivación del gradiente para los pesos y el sesgo.
- **Implementación en Python:** Los estudiantes accederán a un cuaderno de Jupyter previamente preparado.

Enlace al cuaderno: [12-Perceptron-Sigmoide.ipynb](#).

**Verificación de aprendizaje:**

- ¿Cuál es el propósito de la función de activación sigmoide?
- ¿Cómo se calcula el gradiente para el entrenamiento del perceptrón sigmoide?
- ¿Qué tipos de problemas son adecuados para este modelo?

**Actividad 2: Perceptrón Multiclase**

Esta actividad se centra en el perceptrón multiclase, destacando las funciones de combinación, activación (softmax) y pérdida categórica. Se derivará el gradiente para su entrenamiento, utilizando metodologías de clase magistral y aprendizaje basado en proyectos.

**¿Cómo lo haremos?**

- **Clase magistral:** Se cubrirá la teoría que sustenta el perceptrón multiclase:
  - Funciones de combinación y activación (softmax).
  - Pérdida categórica cruzada.
  - Derivación del gradiente para los pesos y sesgos.
- **Implementación en Python:** Los estudiantes accederán a un cuaderno de Jupyter previamente preparado.

Enlace al cuaderno: [12-Perceptron-Multiclase.ipynb](#).

**Verificación de aprendizaje:**

- ¿Cuál es la función de activación utilizada en el perceptrón multiclase y por qué?
- ¿Cómo se deriva el gradiente para el entrenamiento en problemas multiclase?
- ¿Cuándo deberíamos preferir el perceptrón multiclase frente a otros modelos?

---

**CIERRE**

---

**Tarea:** Realizar los ejercicios planteados en los cuadernos de Jupyter.

**Pregunta de investigación:**

1. ¿Cómo se compara el perceptrón multiclase con otros métodos de clasificación, como regresión logística multinomial?
2. En el perceptrón multiclase, ¿por qué se utiliza la función softmax junto con la pérdida categórica cruzada? ¿Cómo trabajan juntas para mejorar la clasificación?
3. ¿Qué problemas podrían surgir al usar combinaciones incorrectas de funciones de activación y pérdida?

**Para la próxima clase:** Investigar qué son los frameworks de aprendizaje profundo como PyTorch y TensorFlow, y responder a las siguientes preguntas:

- ¿Qué es un framework? ¿Qué es una API en este sentido?
- ¿Qué características y ventajas ofrecen estos frameworks para implementar redes neuronales?
- ¿Cuáles son las diferencias clave entre PyTorch y TensorFlow en términos de facilidad de uso, flexibilidad y rendimiento?
- ¿Qué ejemplos prácticos existen del uso de PyTorch y TensorFlow en la resolución de problemas de clasificación y predicción?
- ¿Cómo podrías instalar y configurar uno de estos frameworks en tu entorno de trabajo?

Se proporcionará material de lectura y enlaces sugeridos:

- [Introducción a PyTorch](#)
- [Documentación oficial de TensorFlow](#)