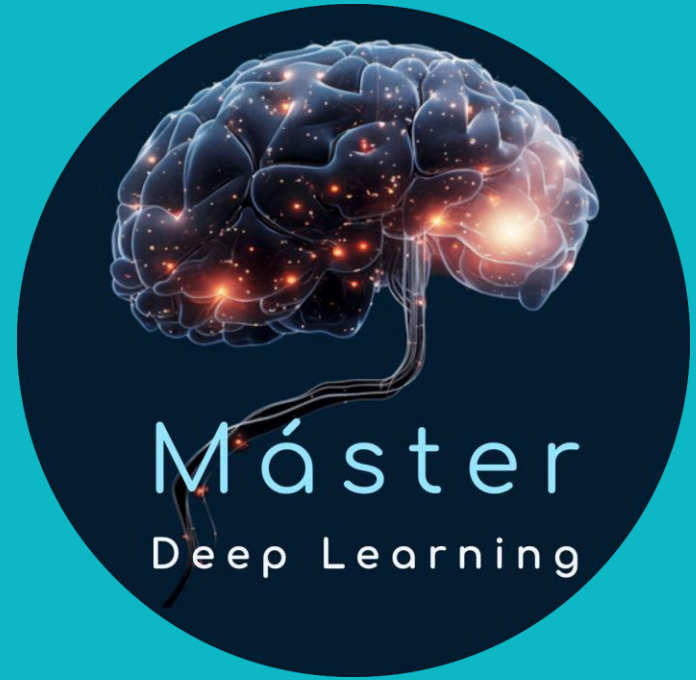


# Deep Learning

Pytorch 4 – Logging & Inference



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



Máster  
Deep Learning

1. Guardar y cargar modelos
2. Evaluación del modelo y visualización de métricas
3. Precisión Mixta Automática
4. Callbacks en pytorch
5. Gráficas de entrenamiento
6. Guardar métricas

# 1. Guardar y cargar modelos

## 4 ¿Por qué guardar modelos?

- Reutilizar modelos
- Reentrenar el modelo en el futuro
- Inferencia



## ¿Qué partes del modelo podemos guardar?

	¿Qué incluye?	Ventajas	Desventajas	Uso común
<b>Pesos del modelo</b>	<ul style="list-style-type: none"><li>• Parámetros aprendidos</li></ul>	Ligero y eficiente. Flexible: puedes reutilizar en distintas arquitecturas	Necesitas reconstruir la arquitectura al cargar.	Transfer learning, reentrenamiento
<b>Modelo completo</b>	<ul style="list-style-type: none"><li>• Arquitectura del modelo.</li><li>• Parámetros aprendidos.</li></ul>	Listo para inferencia.  Fácil de usar: no necesitas redefinir el modelo.	Menos flexible.  Más pesado en almacenamiento.	Inferencia rápida, compartir el modelo.
<b>Checkpoint</b>	<ul style="list-style-type: none"><li>• Parámetros del modelo</li><li>• Estado del optimizador</li><li>• Épocas y pérdida</li></ul>	Permite reanudar el entrenamiento.  Guarda todo el estado de entrenamiento.	Más pesado.  Requiere optimizador y modelo definidos al cargar	Entrenamientos largos o interrumpidos.

## 2. Evaluación del modelo y visualización de métricas



## Evaluación del modelo

- Los resultados obtenidos del entrenamiento y validación ***no muestran el rendimiento del modelo en situaciones reales***, con muestras desconocidas.
- Permite detectar problemas en el entrenamiento.
  - Problemas de generalización
  - Sesgos

## Métricas en problemas de clasificación

- Precisión (Accuracy)
- Tasa de Verdaderos Positivos (TPR)
- Tasa de Verdaderos Negativos (TNR)
- F1-score
- Curva ROC y área bajo la curva ROC (AUC)
- Matrices de confusión



# 3. Precisión Mixta Automática

## Precisión Mixta Automática (AMP)

- Es una técnica utilizada para acelerar el entrenamiento de modelos de deep learning al realizar cálculos con una mezcla de precisión de 16 bits (float16) y 32 bits (float32) de forma automática y segura.
- Implica realizar ciertas operaciones con menor precisión (float16) mientras otras se mantienen en mayor precisión (float32), ya que ***no todas las operaciones son seguras en float16*** debido a posibles pérdidas de precisión o problemas



## Precisión Mixta Automática (AMP)

- Las dos principales ventajas son:
  - **Mejora del rendimiento:** Reduce el uso de memoria y aumenta la velocidad de las operaciones en GPU.
  - **Menor consumo de memoria:** Requiere menos memoria para almacenar tensores y gradientes, lo que permite entrenar modelos más grandes o aumentar el tamaño de los batch.

## 4. Callbacks en pytorch



## Callbacks en pytorch

- Son herramientas que permiten **gestionar tareas** comunes **durante el entrenamiento** de modelos de forma más eficiente y estructurada.
- Facilitan la integración de funciones como métricas, visualización y acciones específicas:
  - Visualizaciones del entrenamiento
  - Parada temprana
  - Crear puntos de control
  - Etc.



## Ejemplos Callbacks:

- **EarlyStopping**: es una técnica utilizada durante el entrenamiento de modelos para ***detener el proceso de entrenamiento automáticamente*** si una métrica monitorizada deja de mejorar después de un número especificado de épocas.
- **ModelCheckpoint**: herramienta que permite ***guardar automáticamente los mejores modelos*** durante el entrenamiento basándose en una métrica específica. También se puede usar para guardar checkpoints periódicamente.



## Ventajas de usar Callbacks:

- **Flexibilidad:** Pueden personalizarse para adaptarse a flujos de trabajo específicos.
- **Compatibilidad:** Suelen integrarse fácilmente con frameworks como torchmetrics, tensorboard o herramientas de visualización.
- **Modularidad:** Permiten añadir y combinar funcionalidades sin modificar el núcleo del modelo o el entrenamiento.
- **Automatización:** Reducen el código repetitivo al manejar procesos como checkpoints, registro de métricas o programación de eventos.

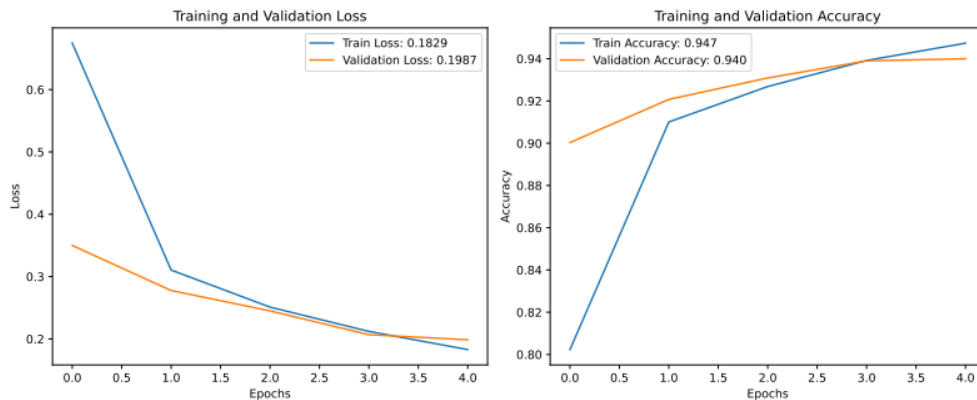
## 5. Gráficas de entrenamiento





## Gráficas de entrenamiento

- Nos permite registrar el proceso del entrenamiento.



## 6. Logging

## Logging en pytorch

- Nos permite guardar métricas, hiperparámetros, etc, en ficheros.

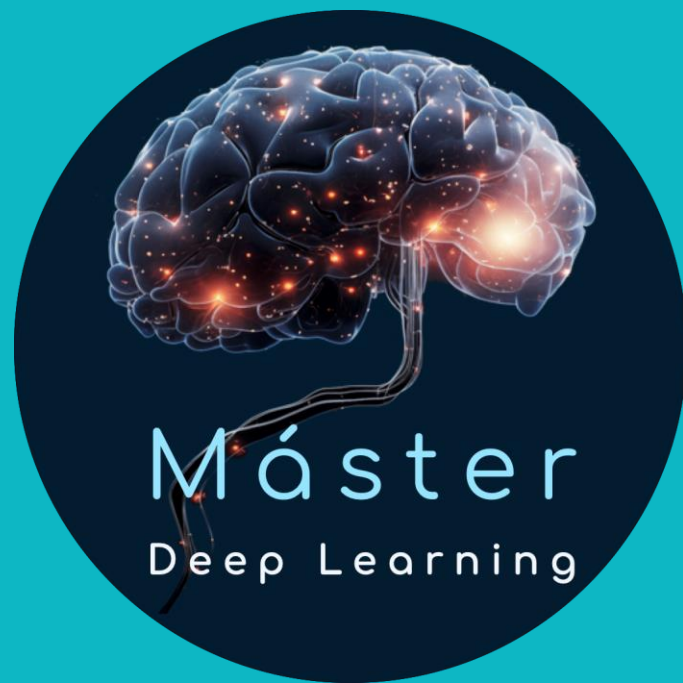
# 7.EXTRA RESOURCES

## Documentación

- ▶ [https://pytorch.org/tutorials/beginner/saving\\_loading\\_models.html](https://pytorch.org/tutorials/beginner/saving_loading_models.html)
- ▶ <https://pytorch.org/docs/stable/amp.html>

# Deep Learning

Pytorch 4 – Logging & Inference



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



Máster  
Deep Learning