

## Práctica 1: Clasificación de imágenes. Arquitecturas CNN. *Transfer Learning*

Esta práctica se implementa en Google Colab, un entorno que permite desarrollar código Python en Python Notebooks y utilizar aceleración GPU para entrenar modelos de Aprendizaje Profundo. Para usarlo, tendrá que crear una cuenta personal de Google. Para trabajar en Google Colab, siga estas instrucciones:

- Cree una carpeta para las prácticas del curso en su Google Drive personal.
- Abra los enlaces de Colab incluidos en este guion
- Guarde una copia de los notebooks en su carpeta de Google Drive
- Cada notebook de Colab (es decir, los archivos que terminan en .ipynb) corresponde a una parte de la práctica. En Google Drive, haga doble clic en el notebook y seleccione *Abrir con Colab*.
- Una vez que haya completado la práctica (es decir, ha llegado al final del notebook), puede guardar el archivo editado y pasar al siguiente bloc de notas.
- Asegúrese de guardar periódicamente el notebook (Guardar archivo), para no perder el progreso si la máquina virtual de Colab se desconecta. →

El objetivo de esta práctica es presentar al estudiante el problema de clasificación de imágenes, los conceptos básicos de varias arquitecturas CNN, el manejo de datasets de imágenes con PyTorch, y la utilización de estrategias de *Transfer Learning*. Para completar esta práctica, deberá leer la documentación de Pytorch. Puede encontrarla [aquí](#).

Debe completar el código de los notebooks de Colab, y completar un informe de práctica con las respuestas a las preguntas que se incluyen en las secciones siguientes. Una vez haya terminado, debe subir a Moodle una copia de los notebooks completados (archivos .ipynb) y el informe de práctica, combinados en un solo archivo zip. **IMPORTANTE: Tanto el archivo del informe como el archivo zip deben tener el siguiente nombre: 'APELLIDO(s)\_NOMBRE'**

Puedes encontrar [aquí el primer notebook](#). Debe completar el código en el notebook y responder a las preguntas de la Sección 1.1.

### 1.1 Simple CNN

- Tamaños de los conjuntos de entrenamiento y validación descargados del *dataset* MNIST

	Alto de imagen	Ancho de imagen	N.º canales de imagen	N.º muestras
Entrenamiento				
Validación				

- Número de parámetros del modelo Simple CNN

	N.º parámetros entrenables
Simple CNN	

- Incluya las curvas de entrenamiento y validación para 10 épocas. Indique también la mejor precisión obtenida, y en qué época se logra este resultado

	Mejor precisión (validación)	Época con mejor precisión
Simple CNN		

Comentar las conclusiones sobre la evolución de la *loss* de entrenamiento y validación, con respecto a posibles problemas de sesgo (*high-bias*) o sobreajuste (*overfitting*). Indique si considera que continuar con más épocas de entrenamiento mejoraría el rendimiento del modelo

- Incluir la matriz de confusión obtenida. Dada esta matriz de confusión, informe de los 2 casos de confusión entre clases que ocurren con más frecuencia.
- Comente las diferencias entre el gráfico t-SNE de la representación de las capas final e intermedia de la CNN, aplicado a las imágenes del conjunto de validación. Para ello, considere la proximidad y la dispersión entre los clústeres en ambas representaciones, y su relación con la capacidad de realizar una correcta clasificación de las muestras.
- Dadas las diferencias entre la representación t-SNE de ambas capas, y dada la arquitectura de la red implementada, identifique en qué capa de la red se extraen las características, y proponga una forma de reducir la complejidad de la red, con una penalización baja en la precisión de la clasificación.

Puede encontrar [aquí el segundo notebook](#). Debe completar el código en el notebook y responder a las preguntas de las Secciones 6.2 y 6.3.

## 1.2 AlexNet

- Incluya el código que ha utilizado para definir la clase Alexnet
- Número de parámetros del modelo AlexNet

	Nº parámetros entrenables
AlexNet	

- Incluya las curvas de entrenamiento y validación para 15 épocas. Indique también la mejor precisión obtenida, y en qué época se logra este resultado

	Mejor precisión (validación)	Época con mejor precisión
AlexNet		

Comentar las conclusiones sobre la evolución de la *loss* de entrenamiento y validación, y comentar lo que posiblemente está sucediendo después de la época 10. Indique si considera

que continuar con más épocas de entrenamiento mejoraría el rendimiento del modelo

- Incluir la matriz de confusión. Comentar los resultados obtenidos atendiendo a las características de las imágenes de cada clase
- Incluya los resultados t-SNE para la capa última capa de la red: analice estos resultados (proximidad, dispersión, agrupación de clústeres) teniendo en cuenta la apariencia de las imágenes de las diferentes clases, sus características típicas y compare los resultados con los resultados t-SNE en el *dataset* MNIST.

Puedes encontrar [aquí el tercer notebook](#). Debe completar el código en el *notebook* y responder a las preguntas de la Sección 2.1.1.

### 1.3 Transfer Learning

- Precisiones obtenidas para las diferentes alternativas analizadas:

	Entrenado desde cero	Pre-entrenamiento + SVM	Ajuste fino (sin <i>data augmentation</i> )	Ajuste fino (con <i>data augmentation</i> )
Precisión				

- Compare las representaciones t-SNE de las diferentes alternativas: entrenamiento desde cero, pre-entrenamiento + SVM, ajuste fino (sin *data augmentation*) y ajuste fino (con *data augmentation*) A partir de las diferentes representaciones obtenidas, en las cuatro alternativas analizadas., comente sus diferencias en cuanto a la capacidad de separar linealmente ambas clases, y el nivel de muestras clasificadas erróneamente dada esta separación lineal.