

### P P3. Redes Neuronales

En este proyecto estarás trabajando con datos generados por el grupo, imágenes de los dígitos del 0 al 9. Se te pedirá que generes un modelo de clasificación utilizando redes neuronales, que evalúes su desempeño en un test ciego, y que evidencies su funcionalidad en tiempo real.

Desarrolla los siguientes puntos en una *Jupyter Notebook*, de forma que se genere un reporte fácilmente comprensible, interpretable y replicable, con información para el lector descrita en formato de markdown.

1. Importa las imágenes a tu ambiente de trabajo (te recomiendo ampliamente usar el método 'flow\_from\_directory' de keras en unión con un objeto de clase 'ImageDataGenerator'). Asegúrate de transformar las imágenes para que estén en escala de grises y de escalarlas al rango  $[0 - 1]$ .
2. Construye una arquitectura de redes neuronales convolucionales para realizar la clasificación de los dígitos. Entrena al modelo en un subconjunto de los datos de entrenamiento y revisa su desempeño usando el subconjunto restante como datos de validación. Explica con detalle la estructura de la arquitectura seleccionada, así como la forma en que los datos van cambiando de tamaño conforme pasan por la red, y las características del entrenamiento realizado.
3. Realiza múltiples modificaciones a la arquitectura y al proceso de entrenamiento (deberás generar resultados para al menos 5 modelos adicionales), de forma que la separación de entrenamiento – validación te sirva para definir la mejor arquitectura (ajuste de hiperparámetros). Explica con detalle las modificaciones realizadas en cada modelo, tanto para la arquitectura como para el proceso de entrenamiento.
4. Para el modelo seleccionado como el mejor del grupo, genera gráficas que comparen la exactitud del modelo entre los datos de entrenamiento y los de validación, a través de las épocas de entrenamiento. Explica con detalle todas las conclusiones que puedas generar a partir de la información mostrada en la gráfica.
5. Utiliza la estructura y características de entrenamiento del modelo “ganador”, para entrenar un nuevo modelo que utilice todo el conjunto de entrenamiento en su proceso de entrenamiento. Con dicho modelo, clasifica las imágenes de test. Reporta métricas de interés y muestra una matriz de confusión. Detalla las conclusiones que generas de los resultados mostrados en la matriz de confusión, así como las obtenidas al comparar los resultados de este ejercicio con los obtenidos por el modelo “ganador” en los datos de validación.
6. Graba el modelo del punto anterior y utilízalo para realizar clasificaciones en tiempo real. En tu computadora deberán observarse a la vez: la imagen original con la probabilidad de cada clase en color rojo, con excepción de la probabilidad de la clase “ganadora” que deberá mostrarse en color verde; la imagen que entra al modelo de CNN, tal y como entra al modelo (no modifiques tamaño, color, etc.). Deberás grabar un video corto (no más de 30 segundos), donde se muestre la clasificación (correcta o incorrecta) de al menos 5 dígitos en 5 formatos distintos (lápiz en hoja de papel blanco, pluma en hoja de cuadros, plumón en pintarrón, etc.)
7. Si deseas que este proyecto se evalúen sobre 110, integra tu modelo a un sistema de OCR (Optical Character Recognition). El proceso típicamente se vería algo así:

- Lee la imagen
- Convierte a escala de grises
- Aplica un filtro de desenfoque (blur) para reducir el ruido (se define el tamaño adecuado del kernel de forma experimental)
- Binariza la imagen usando un umbral estático o dinámico (el umbral estático se define de forma experimental)
- Aplica una función morfológica de dilatación para unir bordes no continuos (se define el tamaño adecuado del kernel, así como la cantidad de iteraciones, de forma experimental).
- Se buscan los contornos de la imagen (típicamente se trata de evitar incluir contornos que están dentro de otros contornos).
- Se itera para cada contorno encontrado un proceso en que, primero, se genera un *bounding box* con las coordenadas del contorno encontrado, se extrae ese pedazo de imagen de la imagen dilatada, se escala la imagen para tener el tamaño y dimensiones esperadas por el modelo, se escala la imagen, y por último, se realiza una predicción usando el modelo.
- Se indica con alguna visualización qué dígitos se encontraron en la imagen y dónde se ubican dentro de la misma.