

## Desafío - Identificación de escenarios naturales del mundo

En este desafío validaremos nuestros conocimientos de redes neuronales convolucionales. Para lograrlo, necesitarás desarrollar modelos de redes neuronales convolucionales para clasificación de imágenes para seis diferentes clases.

Lee todo el documento antes de comenzar el desarrollo **individual**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

**Tiempo asignado: 2 horas**

### Descripción

En esta oportunidad buscaremos entrenar redes neuronales convolucionales para un conjunto de imágenes que corresponden a escenarios naturales del mundo, con seis diferentes clases:

- **buildings:** Para escenarios con edificios
- **forest:** Escenarios de bosques
- **glacier:** Glaciares
- **mountain:** Escenarios con montañas
- **sea:** Imágenes con mar
- **street:** Escenarios de ciudad

La base de datos se encuentra en cinco archivos serializados (pickle):

- **cnn\_train\_X.npy:** imágenes destinadas para entrenar los modelos
  - **cnn\_train\_y.npy:** etiquetas asociadas a las imágenes de entrenamiento
  - **cnn\_test\_X.npy:** imágenes para test
  - **cnn\_test\_y.npy:** etiquetas correspondiente a las imágenes de test
  - **cnn\_pred\_X.npy** conjunto de imágenes sin etiquetar destinadas a predicción.
1. Importa correctamente los conjuntos de datos, desplegando la cantidad de registros de cada categoría para cada conjunto. Despliega ocho imágenes aleatorias del conjunto de entrenamiento y codifica las etiquetas con OneHotEncoder para los conjuntos de entrenamiento y Test.
  2. Implementa una red neuronal multicapa Fully Connected para clasificar las imágenes, las que deben contener un mínimo de cinco capas ocultas. Despliega las

métricas (aplicadas al conjunto de test) que estimes conveniente para medir el rendimiento del modelo.

3. Desarrolla una red neuronal Convolutiva usando tres o más capas convolucionales y dos o más capas densas, de manera que la cantidad de parámetros a estimar se encuentre en el intervalo [3MM, 8MM], con 25 épocas. Muestra la curva de la pérdida para cada época y la curva de la métrica accuracy (conjunto test).
4. Implementa una red neuronal Convolutiva que mejore la capacidad de generalización alcanzada en el modelo anterior. Para esto usa 16 o más capas (entre convolucionales y densas) y mide su rendimiento.

Luego, con el modelo entrenado, aplícalo sobre el conjunto de predicción, mide su rendimiento y compáralo con el rendimiento calculado en el modelo anterior. ¿Para qué etiqueta el modelo se equivoca más? Muestra en forma aleatoria ocho imágenes con la etiqueta de predicción otorgada por el modelo.

## Requerimientos

1. Importa, codifica y despliega adecuadamente las imágenes y variables **(1 punto)**
2. Implementa y evalúa redes neuronales multicapa, y evalúa su rendimiento. **(3 puntos)**
3. Implementa redes convolucionales con parámetros determinados, y evalúa sus curvas de pérdida y accuracy. **(3 puntos)**
4. Implementa mejoras a redes convolucionales, y las compara de acuerdo con sus desempeños. **(3 puntos)**



¡Mucho éxito!

### Consideraciones y recomendaciones

- Para resolver este desafío se sugiere utilizar Google Colab, con procesamiento de GPU
- Para cargar los dataset use np.load con el argumento allow\_pickle=True