

Universidad de los Andes
Maestría en Inteligencia Analítica para la Toma de Decisiones.
Modelos Avanzados para el Análisis De Datos 2.
Prof. Martín Andrade Restrepo.

Taller 5. Redes convolucionales

Observaciones:

En este taller usaremos redes convolucionales (*convnets*) para problemas relacionados con visión por computador. Construiremos los modelos utilizando el API funcional de Keras (*functional API*), más flexible que el API secuencial (*Sequential API*) que utilizamos en los talleres pasados, dado que permite construir distintos “canales” de procesamiento de la información.

Parte I (30 ptos)

En este ejercicio utilizaremos redes convolucionales para el problema de reconocimiento de dígitos de MNIST. Como nuestro input en este caso será procesado como una imagen (una matriz) y no como un vector podemos importarlo y procesarlo usando:

```
1 from tensorflow.keras.datasets import mnist
2 (train_images, train_labels), (test_images, test_labels) = mnist.load_data()
3 train_images = train_images.reshape((60000, 28, 28, 1))
4 train_images = train_images.astype("float32") / 255
5 test_images = test_images.reshape((10000, 28, 28, 1))
6 test_images = test_images.astype("float32") / 255
```

Ya habiendo importado nuestros datos podemos construir nuestra primera red convolucional usando:

```
1 from tensorflow import keras
2 from tensorflow.keras import layers
3 inputs = keras.Input(shape=(28, 28, 1))
4 x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(inputs)
5 x = layers.MaxPooling2D(pool_size=2)(x)
6 x = layers.Flatten()(x)
7 outputs = layers.Dense(10, activation="softmax")(x)
8 model = keras.Model(inputs=inputs, outputs=outputs)
```

Como podrá ver, hemos definido nuestra red sin usar el API secuencial. Por el contrario, hemos definido la red capa por capa utilizando la función “layers” incluyendo al final como parámetro la capa anterior. La acción de “layers” es como dibujar una flecha desde la capa que se recibe como parámetro hasta esta capa que creó.

- a) (5 ptos) Utilice `model.summary()` para describir la red que creó. Describa sus observaciones.

- b) **(5 ptos)** Pruebe el modelo definido arriba para sus datos de prueba. Utilice `sparse_categorical_crossentropy` como función de costo/pérdida. Describa sus observaciones ¿Cuál es el mejor desempeño que logra con su red?
- c) **(15 ptos)** Divida de su conjunto de prueba en 2 para disponer de un conjunto de validación. Incluya un par de capas convolucionales (con sus respectivas capas de pooling) adicionales en su red y un par de capas densas antes del output. Experimente con los hiperparámetros y describa sus mejores resultados (su mejor red debería superar el 98.5% - 99% de acierto).
- d) **(5 ptos)** Experimente con variantes del pooling utilizado (puede probar otros tamaños de ventana y utilizar también L2 pooling). Describa sus resultados.

Parte II (70 ptos)

Construcción de una RNC usando los datos de CIFAR10.

- a) **(10 ptos)** Investigue y describa qué es la base de datos CIFAR10.

En un nuevo código `main` importamos las librerías necesarias y los datos:

```
1 import tensorflow as tf
2 from tensorflow import keras
3 from tensorflow.keras import layers
4 from tensorflow.keras.datasets import cifar10
5
6 (x_train, y_train), (x_test, y_test) = cifar10.load_data()
7 # convert float 32 to have better efficiency
8 x_train = x_train.astype('float32')/255.0
9 x_test = x_test.astype('float32')/255.0
```

- b) **(10 ptos)** Describa preliminarmente los datos que utilizará su red (distribuciones, posibles valores de sus variables, dimensiones, etc.).
- c) **(20 ptos)** Construya una red convolucional a su gusto. Entrene dicha red y evalúela con sus datos. Se recomienda que divida su conjunto de prueba en dos para calibrar sus hiperparámetros. Utilice `model.summary()` para explorar su modelo con más detalle.
- d) **(20 ptos)** Experimente con diferentes arquitecturas y describa sus mejores resultados.
- e) **(10 ptos)** Investigue sobre el parámetro “padding” y sobre el “batch normalization” (que ocasionalmente se utiliza en lugar del pooling). Experimente con ambos y describa sus resultados.