

### **Práctica 5: CNN para clasificación multi-class (CIFAR-10)**

El objetivo de esta práctica es familiarizarse con la biblioteca Keras para python. Concretamente, aprenderemos a definir una red neuronal convolucional (CNN) profunda para atacar el problema de clasificación de imágenes en un escenario multi-clase. También aprenderemos cómo sintonizar algunos de los parámetros más relevantes de la red.

Para esta práctica se utilizará el dataset CIFAR-10, el cual contiene imágenes de 10 distintas clases. Cada imagen está guardada en formato RGB con tamaño 32x32 píxeles.

Se proporciona un notebook de jupyter con código que deberá ser completado, el cual ayudará a responder las siguientes preguntas.

#### **Tareas iniciales**

1. Crea una CNN de, al menos -puede tener más si quieres-, 2 layers convolucionales con sus respectivas operaciones max-pooling, y dos layers fully-connected (FC, i.e., multi-layer perceptron). Tú eliges el número y tamaño de los filtros para los layers convolucionales, el número de neuronas para los FC, y las correspondientes funciones de activación.
2. Define los parámetros de optimización para tu modelo usando “categorical cross entropy” como función de pérdida y el optimizador “adam”. Adicionalmente, calcula la métrica exactitud.
3. Entrena tu modelo durante 10 épocas usando lotes de 64 ejemplos (batch size). Durante el entrenamiento, usa 25% de los datos de entrenamiento para validar. Asegurate que los datos son ordenados aleatoriamente en cada época.
4. Usa como baseline el modelo que acabas de definir y entrenar. Reporta sus detalles de configuración.
5. Grafica la pérdida y la exactitud de clasificación obtenida en cada época para los sets de entrenamiento y validación.
6. Después de entrenar el modelo, úsalo para calcular la pérdida y la exactitud en el set de test. ¿De cuánto es?
7. Comparando la pérdida de entrenamiento, validación, y test, ¿qué puedes concluir?
8. Dado que las imágenes del CIFAR-10 son de 32x32 píxeles, se puede usar fácilmente 4 operadores max-pooling hasta llegar a imágenes de 2x2 píxeles, y posteriormente operadores up-sampling para regresar al tamaño de 32x32, i.e.,  $32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32$ . Si las imágenes de entrada fueran de 28x28 píxeles, las operaciones de max-pooling y up-sampling resultarían en tamaños  $28 \rightarrow 14 \rightarrow 7 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32$ , ¿cómo resolverías el problema?

#### **Otras Preguntas**

9. Evalúa tu modelo usando batch sizes = 4, 16, 128, 256. ¿Disminuyen las pérdidas de entrenamiento y validación? ¿Por cuánto? ¿Cómo afecta esto el tiempo requerido para entrenar?
10. Fija el tamaño del batch en 256. Duplica el número de filtros en cada layer convolucional que hayas definido. ¿Qué impacto tiene esto en el desempeño? ¿Y en el tiempo de entrenamiento?

11. Regresa al número de filtros convolucionales anterior y mantén el tamaño de batch en 256. Ahora duplica el número de layers convolucionales. ¿Qué impacto tiene esto en el desempeño? ¿Y en el tiempo de entrenamiento?
12. Con base en tus resultados hasta ahora, define un nuevo baseline con la combinación de batch size, número de layers convolucionales, y número de filtros en cada layer que creas que mejor desempeño obtendrán. Puedes evaluar también otras funciones de pérdida y otros optimizadores. Entrénalo. Reporta su configuración y su desempeño. ¿Cuáles fueron los resultados?

Usando tu mejor modelo:

13. Muestra la exactitud de clasificación obtenida en el test set.
14. Muestra la matriz de confusión correspondiente.
15. Grafica los filtros del primer layer.
16. Grafica los filtros del último layer.
17. Crea un modelo parcial cuya salida sea la misma que la del primer layer de tu mejor modelo. Grafica la salida producida por cada filtro.
18. Crea un modelo parcial cuya salida sea la misma que la del último layer convolucional de tu mejor modelo. Grafica la salida producida por cada filtro.

Deadline: Miércoles 24 de abril, 8:30 am.

Entrega: Enviar por email, el jupyter notebook donde se muestre cada una de las salidas visuales solicitadas, y un documento PDF que contenga los reporte de configuración de tu baseline y de tu mejor modelo, y que además contenga las respuestas a las preguntas solicitadas.