

Taller 5: Introducción a Redes Neuronales Convolucionales (*CNNs*)

IELE 4922 - Reinforcement Learning
Universidad de los Andes
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica y Electrónica
Semestre 2022-10

¿Qué son las *CNNs*?

Las redes neuronales convolucionales (o *CNNs* por sus siglas en inglés) son un tipo de red neuronal artificial (*ANN*) que se caracteriza por utilizar múltiples capas de filtros convolucionales. Este tipo de red se ha popularizado especialmente en el área de visión por computador debido a su gran éxito en la solución de problemas relacionados con imágenes (procesamiento, clasificación, segmentación, generación, entre otros). Típicamente, la arquitectura de una CNN está constituida principalmente por tres tipos de capas (véase figura 1):

- Capa Convolutiva
- Capa de *Pooling*
- Capa completamente conectada (*fully connected*)

En primer lugar, la capa convolutiva se encarga de tomar las imágenes de entrada y aplicar un total de m filtros para poder hacer un mapeo de características. Luego, el mapa de características ingresa a la capa de *pooling* para comenzar el procedimiento de reducción de dimensionalidad. Mediante este se seleccionan únicamente las características (o *features*) relevantes. Finalmente, todas las características pasan por una capa de *flatten*; esta capa se encarga de transformar una matriz de características en un vector, el cual corresponde a la entrada de la última etapa, la red completamente conectada (*fully connected*). Es en esta última parte de la CNN es donde se lleva a cabo la clasificación.

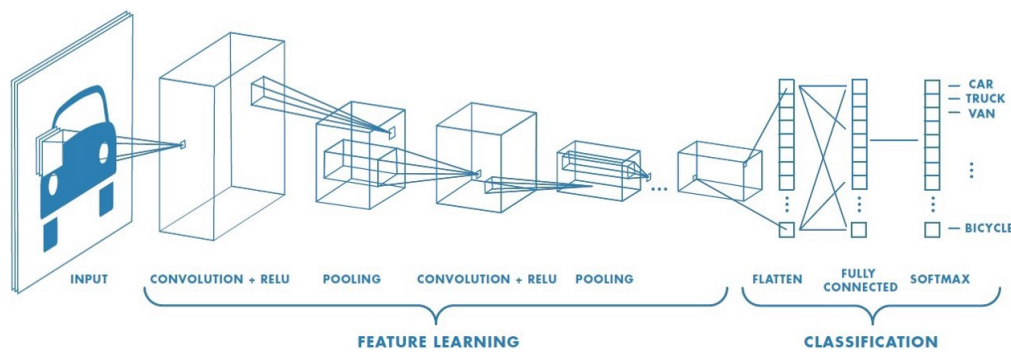


Figura 1: Arquitectura estándar de una red neuronal convolutiva para una tarea de clasificación

A diferencia de los problemas que se han trabajado hasta ahora en el curso, este taller se enfocará en resolver una tarea de **aprendizaje supervisado**. Esto quiere decir que se tendrá una serie de datos de entrada (en este caso imágenes) y sus respectivas etiquetas como salida del modelo. El objetivo es que a partir de unos datos de entrenamiento, el modelo de red neuronal convolucional (CNN) sea capaz de clasificar (asignar la etiqueta) correctamente a las imágenes que no ha visto. Para mayor información en cuanto las CNNs y a su implementación se sugiere revisar los siguientes artículos.

- [Comprehensive Guide to CNNs](#)
- [Convolutional Neural Network implementation with Keras](#)
- [CNN Implementation using Google Colab](#)

Recomendación: Dado el costo computacional que pueden requerir los entrenamientos de estos modelos, se recomienda trabajar en el entorno de *Google Colab*. Esto le permite acceder a una aceleración por hardware a través de una GPU. Actívalo en la barra de herramientas de la siguiente manera: *Entorno de ejecución* → *Cambiar entorno de entorno de ejecución* → *Acelerador por Hardware* (véase figura 2).

Configuración del cuaderno

Acelerador por hardware

GPU  

Para sacar el máximo partido a Colab, evita usar una GPU si no es necesario para tu trabajo. [Más información](#)

☐ Omitir resultado de las celdas de código al guardar este cuaderno

CANCELAR GUARDAR

Figura 2: Activación del acelerador de hardware en *Google Colab*

Clasificación de imágenes

Utilice la base de datos NORB pequeña ¹. En este problema se quiere aprender a clasificar una imagen de un juguete en una de cinco posibles clases: Animales de cuatro patas, Figuras Humanas, Aviones, Camiones y Carros.

Separe aleatoriamente el dataset en grupos de entrenamiento (70 %), validación (15 %) y prueba (15 %)

Su trabajo será, entonces, el de diseñar y entrenar una red convolucional sencilla que les permita clasificar de forma correcta las imágenes.

Para esto, usted tendrá que:

- **Diseñar y construir** la arquitectura de red neuronal convolucional (procure mantener un tamaño no muy grande para evitar *over-fitting* y mantener un costo computacional bajo).

¹Disponible en <http://www.cs.nyu.edu/~ylclab/data/norb-v1.0-small/>

- **Entrenar** dicha red con el set de datos de entrenamiento. Verifique que la red esta entrenando de forma correcta evaluando el error de entrenamiento (*training loss*) conforme avanza el número de épocas.
- **Variar parámetros** como: filtros, número de capas ocultas, funciones de activación, número de neuronas, capas de *dropout*, entre otros. Investigue los efectos que puede tener estos parámetros con el **objetivo de poder mejorar el desempeño de la red**. Para esto evalúe métricas como *accuracy*, *precision* y *recall* realizando pruebas con el set de datos de validación.
- **Evaluar el desempeño** de la arquitectura definitiva de su red neuronal convolucional con el set de datos de prueba. Incluya una [matriz de confusión](#).

Entregable: *Jupyter notebook* (archivo .ipynb) con el entrenamiento, ajuste de parámetros y evaluación de la red neuronal convolucional para el problema de clasificación propuesto.