



UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
APRENDIZAJE PROFUNDO Y APLICACIONES

Ignacio Ibáñez Aliaga
ignacio.ibanez@usach.cl



Laboratorio 1: Convolutional Neural Network (CNN)

Fundamento Teórico

Una de las estructuras de redes neuronales profundas que ha generado un enorme impacto en el uso de imágenes [2], no fue explotado su uso debido a las limitaciones de *hardware* que existía en aquel entonces, dado que el proceso de entrenamiento de estas redes es costoso. Pero años posteriores se usó de forma satisfactoria para la clasificación de dígitos numéricos, al usar una CNN junto al algoritmo de aprendizaje basado en gradientes [3].

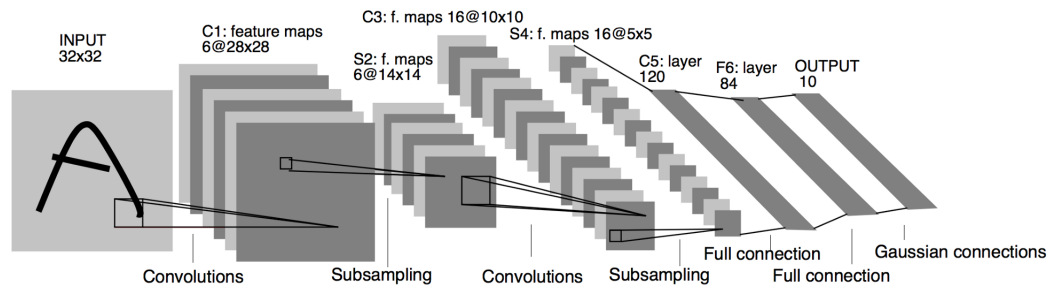
Debido a su gran potencial, los investigadores continuaron mejorando las CNN siendo usadas en gran variedad de tareas de reconocimiento de patrones. Algunas de las ventajas que presentan son un mayor parecido al sistema de procesamiento visual humano, estar altamente optimizadas para procesar imágenes en 2D y 3D, ser eficaz en el aprendizaje y su uso para la extracción de características en 2D [1]. Al estar compuestas por pesos escasos de conexiones, las CNN poseen una cantidad mucho menor de parámetros si se comparan con redes completamente conectadas de tamaño similar, produciendo que en el proceso de entrenamiento basado en el gradiente, sufren menos el problema de disminución del gradiente y por defecto produce pesos altamente optimizados.

La idea central de las CNN es la detección de patrones al igual que otros tipos de redes neuronales, pero la diferencia principal es que los datos de entrada son imágenes las que al ir avanzando dentro de la red son usadas para extraer las diferentes características que la representan. Imitan el comportamiento del cerebro humano y la forma de procesar las imágenes en la corteza visual, compuesta por diferentes capas o grupos de regiones encargadas de realizar funciones específicas. Por ejemplo, si se considera el problema de la detección de rostros las capas se pueden comportar de la siguiente manera:

1. Las primeras capas están especializadas en detectar patrones básicos como líneas o bordes en los diferentes rostros.
2. Las capas siguientes interconectan esos patrones simples y comenzar a identificar diferentes formas dentro de la imagen, pero aún no partes específicas.

3. A medida que se avanza en capas más profundas, estas son capaces de ir interconectando los diferentes patrones y formas detectados en capas anteriores, para poder identificar diferentes figuras como son ojos, nariz, cejas, boca, entre otras.
4. Para terminar en la región inferior con la combinación de la información proveniente de las capas anteriores y determinar la presencia o ausencia de rostros.

Las CNN al recibir como entrada una imagen, progresivamente aplican una serie de capas convolucionales que se encargan de entrenar una gran cantidad de filtros, para extraer las diferentes características de la imagen. Los principales elementos de las CNN son explicados a continuación.



Actividades

El presente laboratorio consiste en utilizar redes neuronales convolucionales para la tarea de clasificación de imágenes, se va a realizar la comparación entre los resultados obtenidos al utilizar un entrenamiento desde cero para una arquitectura propuesta por el alumno, en contraste con un entrenamiento fine-tuning de una arquitectura entrenada con ImageNet.

Para el desarrollo del trabajo es necesario que se siga la estructura dentro del [template](#) dado que esto nos permitirá mantener una organización en el desarrollo del código y un flujo apropiado en las tareas que son necesarias para abordar el problema.

Los pasos que se deben seguir dentro del template de colab son:

1. Parte 1
 - a) Importar las bibliotecas que son usadas.
 - b) Lectura del conjunto de datos, deberá diseñar una estrategia para la selección y lectura del sub conjunto de datos, es decir, se debe crear un conjuntos de datos pequeño que contenga entre 3 a 5 clases y

una cantidad de N ejemplos para cada clase, con el fin de tener un conjunto de datos balanceado con un tamaño que se pueda trabajar con los recursos disponibles.

- c) Realizar la división de los datos en dos conjuntos de datos, Train (70 %) y Test (30 %).
- d) Aplicar técnicas para aumentar la cantidad de imágenes en los datos de entrenamiento, recortando las imágenes, girando, modificando los contrastes, entre otras. Como mínimo, se debe utilizar dos técnicas.

2. Parte 2

- a) Creación de una arquitectura desde cero, para este paso se recomienda adaptar la implementación LeNet vista en clases. Cuya implementación debe funcionar tanto en CPU como en GPU.
- b) Probar el modelo por lo menos con tres configuraciones de parámetros, entre los que se puede variar número de convoluciones, función de activación, número de neuronas en capas densas, entre otros.
- c) Crear los gráficos de entrenamiento, donde se vea el error en los conjuntos de train y test durante cada época de entrenamiento.
- d) Crear matriz de clasificación del modelo que presente el menor error en la clasificación, por lo tanto la mejor clasificación.

3. Parte 3

- a) Importar una arquitectura profunda que se encuentra pre-entrenada con ImageNet
- b) Realizar la modificación en la arquitectura para poder clasificar el número de clases que usted haya escogido.
- c) Aplicar Fine-tuning en la arquitectura profunda, para re-entrenar con el conjunto de datos seleccionado manteniendo los pesos de las arquitecturas ResNet, VGG o DenseNet. Cuya implementación creada debe funcionar tanto en CPU como en GPU.
- d) Probar el modelo por lo menos con tres configuraciones de parámetros.
- e) Crear los gráficos de entrenamiento, donde se vea el error en los conjuntos de train y test durante cada época de entrenamiento.
- f) Crear matriz de clasificación del modelo que presente el menor error en la clasificación, por lo tanto la mejor clasificación.

4. Parte 4 (BONUS)

- a) Escoger el mejor modelo y probar clasificar las imágenes del conjunto test luego de sumarle ligeros niveles de ruido.

Conjuntos de Datos

Para el laboratorio cada grupo deberá seleccionar uno de los conjunto de datos presentados en la siguiente lista o discutir con el profesor un conjunto de datos de similares características, es decir, que tenga 3 o más clases. Los conjuntos de datos se pueden encontrar en el siguiente [link](#)

1. caltech_birds, deberá seleccionar algunas especie de aves como clases y crear un sub conjunto de datos para el desarrollo del laboratorio.
2. cars196, deberá seleccionar algunos modelos de autos como clases y crear un sub conjunto de datos para el desarrollo del laboratorio.
3. stanford_dogs, deberá seleccionar algunas razas de perros como clases y crear un sub conjunto de datos para el desarrollo del laboratorio.

Formato de evaluación

Presentación de avance (30 %)

Durante la presentación de avance se debe exponer un 60 % del trabajo realizado para optar a la nota máxima, es decir, aproximadamente haber completado la parte 1 y 2 o haber completado la parte 1 y 3.

La presentación debe durar máximo 12 minutos y debe contener:

1. Introducción
2. Parte 1
 - a) Explicación del conjunto de datos a trabajar.
 - b) Proceso de selección de datos y técnicas de aumento de datos.
3. Parte 2
 - a) Explicación de la arquitectura creada.
 - b) Explicación de variación de parámetros de entrenamiento.
 - c) Resultados obtenidos.
4. Parte 3
 - a) Explicación de la arquitectura usada en el proceso de fine-tuning.
 - b) Explicación de variación de parámetros de entrenamiento.
 - c) Resultados obtenidos.
5. Parte 4 (BONUS)
6. Análisis de los resultados

7. Conclusiones

Fecha de presentación: 13 de Enero del 2022 durante la clase

Entrega (70 %)

Se debe realizar un informe que sea capaz de plasmar el trabajo realizado. El que debe contener los siguientes puntos.

1. Introducción
2. Explicación del conjunto de datos a trabajar.
3. Proceso de selección de datos y técnicas de aumento de datos.
4. Explicación de la arquitectura creada desde cero.
5. Explicación de variación de parámetros de entrenamiento.
6. Resultados obtenidos.
7. Explicación de la arquitectura usada en el proceso de fine-tuning.
8. Explicación de variación de parámetros de entrenamiento.
9. Resultados obtenidos.
10. Análisis de los resultados
11. Conclusiones

Además, se debe adjuntar el código.

Fecha de entrega: 20 de Enero del 2022 hasta las 23:55 hrs

Notal Final

La nota final del laboratorio se calcula con la siguiente formula:

$$nota_final = nota_presentacion * 0,3 + nota_entrega * 0,7 \quad (1)$$

El desarrollo de esté laboratorio se debe hacer en Colab usando pytorch

Referencias

- [1] Md Zahangir Alom y col. “A state-of-the-art survey on deep learning theory and architectures”. En: *Electronics* 8.3 (2019), pág. 292.
- [2] K Fukushima. “A hierarchical neural network capable of visual pattern recognition”. En: *Neural Network* 1 (1989).
- [3] Yann LeCun y col. “Gradient-based learning applied to document recognition”. En: *Proceedings of the IEEE* 86.11 (1998), págs. 2278-2324.