

# Apuntes Semana 11 - 06/05/2025

Jose Carlos Umaña Rivera  
Instituto Tecnológico de Costa Rica  
Cartago, Costa Rica

**Abstract**—Este documento recopila los apuntes correspondientes a la clase del día. Se presentan algunas observaciones respecto a la Tarea 2 y el Proyecto 2 además de un repaso teórico sobre redes neuronales convolucionales. Luego se realiza la introducción a las arquitecturas convolucionales, mencionando algunas de las arquitecturas más utilizadas actualmente.

## I. DUDAS TAREA 2

De parte del grupo no hubo dudas de la Tarea, el profesor preguntó en general que cambios eran necesarios para lograr cambiar la red para resolver un problema de regresión.

## II. RECOMENDACIONES DEL PROYECTO 2

El profesor preguntó al grupo que tal iban avanzando en cuanto al proyecto, surgieron algunas dudas y entre las recomendaciones que se dieron fueron las de generar las imágenes necesarias de los dataset antes y no durante el entrenamiento del modelo, pues esto podía ser un proceso muy pesado si se hace durante la ejecución, entonces de esta manera solo se tendría que realizar una vez y no múltiples veces. Además el profesor dió la recomendación de probar utilizar una parte reducida del dataset, para así observar si igualmente con menos datos igual se logran muy buenos resultados y de esta manera reducir la carga.

## III. REPASO CONVOLUTIONAL NEURAL NETWORKS

### A. Fully Connected

Se recibe un input de características que son transformadas con una serie de capas ocultas y se tiene una capa de salida. Pero se tiene un problema en el cual se podría llegar a tener muchos pesos o parámetros, lo cual hace el proceso lento y poco escalable.

### B. ConvNet

ConvNet permite tener una neurona en 3 dimensiones (largo, ancho y profundidad), donde estas neuronas estarán conectadas a pequeñas regiones de la capa anterior, en lugar de fully-connected. Permitiendo reducir la imagen a un solo vector.

#### 1) Arquitectura ConvNet:

- Convolutional Layer: Se producen filtros que extraen cierta información necesaria.
- Pooling Layer: Es la capa encargada de realizar el down-sampling a lo largo y ancho de la imagen. Esta se puede decidir cada cuanto se realiza, no es solamente después de realizar toda la convolución.
- Fully Connected: Calcula las probabilidades para realizar la clasificación final.

### C. Capa de Convolución

1) *Filtro o Kernel*: Son matrices que puede variar en dimensiones y están encargadas de realizar la convolución y así generar otras imágenes. Cada uno extrae cierta información de la imagen.

2) *Campo receptivo*: Es equivalente al filtro, siendo la region local de la imagen a la cual está conectada una neurona.

3) *Stride*: Es el número de pasos en el que se aplicará el filtro.

4) *Padding*: Técnica para agregar pixeles alrededor de la imagen donde:

$$p = \frac{k-1}{2}, \quad k = \text{tamaño del filtro}$$

y la dimensión de salida se calcula de la manera:

$$\text{Dimensión} = \frac{m-k+2p}{s} - 1$$

donde m es la cantidad de pesos, k el tamaño del filtro, p el tamaño del padding y s la cantidad de pasos.

### D. Pesos Compartidos

Para reducir los pesos se usan pesos compartidos, utilizando el mismo filtro para toda la imagen.

### E. Pooling Layer

Tiene como función reducir el tamaño espacial de las imágenes a lo largo y ancho. Y se introduce periódicamente en medio de capas convolucionales, teniendo como objetivo reducir la cantidad de parámetros y computación de la red. Se utilizan técnicas como el Max pooling, que toma la información más valiosa.

1) *Dimensionalidad del Pooling*: Entrada  $W \times H \times D$ , recibe el tamaño del kernel k, el Stride s y mantiene la profundidad D.

$$W_2 = \frac{W-k}{s} + 1$$

#### 2) Técnicas:

- Average pooling: Toma el promedio de la región.
- L2-Norm pooling: Usa el valor según la norma L2.

### F. Fully-Connected Layer

Se encarga de unir las arquitecturas y es el clasificador final.

#### IV. ARQUITECTURAS CONVOLUCIONALES

Se prefiere stacks con convoluciones pequeñas. Y algunas de las reglas que se tendrán:

- Imágenes divisibles por 2 muchas veces. (32, 64, 96, 224)
- Campos receptivos pequeños, Stride = 1.
- Padding que no altere el espacio. (usar 0)
- Capa de pooling (max pooling suele ser el más común)

##### A. Arquitecturas

1) *LeNet*: Básicamente la arquitectura vanilla. Desarrollada por Yann LeCun. Consta de 5 capas (2 conv, 2 pooling, 1 fully connected).

2) *AlexNet*: Tenía un problema y es que tiene muchos filtros grandes (11x11, 5x5 y 3x3). Usa 5 capas convolucionales y 3 densas.

3) *ZFNet*: Reduce los kernels de AlexNet. Utilizando filtros de 7x7 primeramente y luego aplicando de 3x3.

4) *GoogleNet*: Reduce la cantidad de parametros de 60 Millones en AlexNet a 4 Millones, utiliza average pooling. Usa módulos Inception, que combinan múltiples tipos de convolución y pooling en paralelo para extraer distintas características en una misma capa.

5) *VGG16*: El 16 en su nombre es porque posee 16 capas, además aplica deep learning.

6) *ResNet*: Permite entrenar redes muy profundas gracias a las conexiones residuales.

7) *DenseNet*: Parecido a ResNet, en lugar de hacer conexiones entre bloques, hace conexiones entre todas las capas. Cada capa recibe como entrada la salida de todas las capas anteriores.