

Apuntes Semana 10 - 29/04/2025

Raúl Sanabria Marroquín
2020182835

Abstract—En este documento se resumen los contenidos vistos durante la Semana 10, enfocados en redes neuronales convolucionales (CNN), la arquitectura ConvNet, técnicas de augmentación de datos en señales de audio, y fundamentos matemáticos como derivadas parciales en redes con funciones sigmoide. También se abordan conceptos clave como stride, padding y campos receptivos.

I. QUIZ

1) Función de activación para clasificación multiclase:

La función *Softmax* se utiliza para calcular distribuciones de probabilidad en tareas de clasificación multiclase:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

2) Técnicas de augmentación en SpecAugment:

- *Time-Frequency Masking*
- *Time Warping*
- *Time Mask*

3) Derivada parcial en red neuronal con sigmoide:

La solución incluye derivadas intermedias utilizando la derivada de la función sigmoide.

$$\frac{\partial L}{\partial a} = 2(a^l - y)$$

$$\frac{\partial a}{\partial z} = \sigma'(z)$$

$$\frac{\partial z}{\partial w} = a_k^{l-1}$$

II. REDES NEURONALES CONVOLUCIONALES

A. Introducción

CNNs son modelos altamente eficientes para procesamiento de imágenes. En datasets como CIFAR-10 (con 10 clases), se trabaja con imágenes de tamaño $32 \times 32 \times 3$, es decir, 3072 entradas por imagen. Las redes completamente conectadas (fully connected) no escalan bien debido a su alta demanda computacional.

B. Arquitectura ConvNet

- Las neuronas están organizadas en tres dimensiones: **largo, ancho y profundidad**.
- Cada neurona está conectada a una pequeña región de la capa anterior (*campo receptivo*), no a toda la imagen.
- Esto permite detectar características como líneas, curvas o bordes.
- La salida final es un vector de clasificación, por ejemplo de tamaño $1 \times 1 \times 10$.

C. Capas Principales de una CNN

1) Convolutional Layer:

- Entrada: [*ancho, largo, canales*]
- Aplica filtros (o kernels) que se aprenden durante el entrenamiento.
- Cada filtro genera una "imagen" de salida, formando una salida de tamaño [*ancho, largo, n*] para *n* filtros.

2) Pooling Layer:

- Realiza *downsampling* sobre ancho y largo, pero mantiene la profundidad.
- Ejemplo: entrada $[32, 32, 12] \rightarrow$ salida $[16, 16, 12]$
- No tiene parámetros entrenables.

3) Fully Connected Layer:

- Toma el vector de características y produce una distribución de probabilidad sobre las clases.

III. CONCEPTOS FUNDAMENTALES EN CNN

A. Filtros o Kernels

- Son matrices (e.g., 3×3 , 5×5) que se deslizan sobre la imagen para aplicar convoluciones.
- **Gaussian Kernel:** suaviza la imagen, útil para eliminar ruido.

B. Campo Receptivo

- Región local de la imagen a la cual está conectada una neurona.
- Ejemplo: para entrada $[32 \times 32 \times 3]$ y campo receptivo $[5 \times 5]$, cada neurona tendrá $5 \times 5 \times 3 = 75$ pesos.

C. Stride, Padding y Cálculo de Dimensiones

- **Stride:** Es la cantidad de pasos que el filtro da sobre la imagen durante la operación de convolución. Un stride mayor reduce la dimensión de la salida.
- **Padding:** Es una técnica que agrega píxeles (generalmente ceros) alrededor del borde de la imagen de entrada. Se utiliza para controlar el tamaño de la salida y, en muchos casos, preservar las dimensiones originales tras la convolución.

Fórmula típica para padding simétrico:

$$p = \frac{k - 1}{2}$$

donde *k* es el tamaño del filtro (kernel).

• Cálculo de las dimensiones de salida:

Sea:

- *m*: tamaño (alto o ancho) de la imagen de entrada,

- k : tamaño del kernel (filtro),
- p : cantidad de padding aplicado,
- s : stride (pasos).

La dimensión de salida se calcula como:

$$\text{Dimensión de salida} = \left\lceil \frac{m - k + 2p}{s} \right\rceil + 1$$

El resultado se redondea al entero más cercano si no es entero.