06. Redes Neuronales Convolucionales (CNNs)

Las **redes neuronales convolucionales** (o **CNN**, por sus siglas en inglés: *Convolutional Neural Networks*) son uno de los tipos más poderosos y utilizados de redes neuronales profundas, especialmente para procesamiento de imágenes, vídeo y datos con estructura espacial o temporal.

¿Qué son las redes convolucionales?

Son un tipo especial de red neuronal diseñada para **extraer automáticamente características** (features) de datos con estructura espacial, como imágenes. A diferencia de una red neuronal clásica, donde cada neurona de una capa está conectada con todas las de la capa siguiente (**fully connected**), en una CNN las conexiones están limitadas por la operación de **convolución**.

Conceptos clave

1. **Convolución**
    - Es una operación matemática entre dos funciones (o matrices, en el caso de imágenes) que produce una tercera función que expresa cómo una es modificada por la otra.
    - En imágenes, se utiliza un pequeño filtro o "kernel" que se desplaza sobre la imagen, multiplicando valores y sumando el resultado. Esto permite detectar patrones locales, como bordes, texturas, esquinas, etc.
2. **Filtros/Kernels**
    - Son matrices de pesos pequeños (ej. 3×3, 5×5) que se aprenden durante el entrenamiento y que permiten extraer ciertas características locales.
3. **Mapas de activación (Feature Maps)**
    - Son el resultado de aplicar un filtro a la imagen. Cada filtro genera su propio mapa.
4. **Pooling**

- Es una operación que reduce la dimensión espacial de los mapas de activación (ej. *max pooling* toma el valor máximo de una pequeña ventana). Ayuda a reducir la cantidad de parámetros y controlar el *overfitting*, además de dar cierta invariancia a traslaciones.

5. **Capas totalmente conectadas (Fully Connected)**
    - Al final de la red, suelen añadirse capas tradicionales, donde cada neurona conecta con todas las neuronas de la capa siguiente. Aquí se realiza la clasificación o regresión final.

Arquitectura típica de una CNN

1. **Entrada:** Imagen (por ejemplo, 28×28 píxeles, 3 canales RGB).
2. **Capas de convolución + activación (ReLU):** Extraen características locales.
3. **Capas de pooling:** Reducen la dimensión.
4. **Varias repeticiones de capas conv+pool.**
5. **Flatten:** Convierte el mapa final en un vector.
6. **Capas totalmente conectadas.**
7. **Capa de salida:** Clasificación o regresión.

¿Por qué funcionan tan bien en imágenes?

- **Parámetros compartidos:** El mismo filtro se usa en diferentes partes de la imagen, lo que reduce drásticamente la cantidad de pesos.
- **Invariancia local:** Detectan patrones sin importar su posición exacta.
- **Composición jerárquica:** Las primeras capas detectan patrones simples (bordes), las intermedias patrones más complejos (formas, partes de objetos) y las últimas objetos completos.

Usos comunes

- Reconocimiento de imágenes (ImageNet, MNIST).
- Detección de objetos y reconocimiento facial.
- Segmentación semántica.
- Análisis de vídeo (frame por frame).

- Procesamiento de audio (usando espectrogramas).

Introducción a las Redes Convolucionales

Las redes neuronales convolucionales (CNNs) son un tipo de red neuronal artificial diseñada específicamente para el procesamiento de datos con estructura de cuadrícula, como las imágenes. Estas redes han demostrado un rendimiento excepcional en tareas como la clasificación de imágenes, detección de objetos y reconocimiento facial.

Arquitectura de una CNN

Una CNN está compuesta por varias capas que trabajan en conjunto para extraer características y tomar decisiones. Las principales capas de una CNN incluyen:

1. **Capa de Convolución**:
   - Aplica filtros (kernels) sobre la imagen de entrada para extraer características locales.
   - Cada filtro aprende un patrón específico, como bordes, texturas o formas.
   - Inicialmente, los filtros tienen valores aleatorios, pero a medida que la red se entrena, se ajustan para captar patrones relevantes.
   - Un filtro de **3×3** se desplaza sobre la imagen, multiplicando sus valores por los de los píxeles y sumando el resultado. Esto genera un **mapa de características**, que destaca ciertas zonas de la imagen.
2. **Función de Activación (ReLU)**:
   - Aplica una función no lineal como `ReLU (Rectified Linear Unit)`, que ayuda a la red a aprender relaciones complejas.
   - ReLU transforma valores negativos en cero, lo que mejora la convergencia del entrenamiento y mantiene solo la información relevante.
3. **Capa de Pooling (Submuestreo)**:
   - Reduce la dimensionalidad de la imagen para disminuir la carga computacional y evitar el sobreajuste.

- **Max pooling** toma el valor más alto en una región de la imagen (por ejemplo, de 2×2), mientras que **average pooling** toma el promedio.

4. **Capas completamente conectadas (Fully Connected, FC)**:
   - Aplanan la salida de las capas convolucionales y la conectan con una red neuronal densa.
   - Sirven para clasificar la imagen en diferentes categorías, combinando la información aprendida por las capas anteriores.
5. **Capa de Salida**:
   - Usa funciones de activación como `softmax` para convertir los valores en probabilidades y clasificar la imagen en una de las posibles clases.

Mecanismo de Aprendizaje de Patrones en CNNs

1. Extracción de Características con Filtros

- Al inicio del entrenamiento, los filtros son aleatorios.
- A medida que la red aprende, los filtros se ajustan para detectar patrones específicos como bordes verticales, texturas y formas.
- Capas más profundas combinan patrones simples para identificar partes de objetos, como un ojo o una rueda.

2. Cálculo del Error y Retropropagación

- La red genera una predicción y la compara con la respuesta correcta.
- Se calcula el **error** usando funciones como `categorical crossentropy`.
- La **retropropagación** distribuye el error hacia atrás, ajustando los filtros y pesos en la dirección que minimiza la pérdida.

3. Ajuste de Pesos con Gradiente Descendente

- Se usa un optimizador como `Adam` o `SGD` para modificar los pesos y mejorar la precisión.
- La actualización de los filtros permite que la red refine su capacidad para detectar patrones.

4. Representaciones Jerárquicas

- **Capas iniciales:** Detectan bordes, colores y texturas.
- **Capas intermedias:** Combinan estos elementos en partes más complejas.
- **Capas finales:** Identifican objetos completos en la imagen.

Aplicaciones de las CNNs

- **Reconocimiento de Imágenes**: Clasificación de imágenes en diferentes categorías (e.g., perros, gatos, autos).
- **Detección de Objetos**: Identificación y localización de objetos en una imagen o video.
- **Reconocimiento Facial**: Utilizado en sistemas de seguridad biométrica.
- **Diagnóstico Médico**: Detección de enfermedades a partir de imágenes médicas.
- **Conducción Autónoma**: Detección y clasificación de elementos en la carretera.

Ejemplo de Implementación en Python (con TensorFlow/Keras)

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

# Definir una CNN sencilla
model = keras.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(28, 28, 1)),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')
])

# Compilar la red
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
```

```
metrics=['accuracy'])

# Resumen del modelo
model.summary()
```

Conclusión

Las redes convolucionales han revolucionado el campo de la visión por computadora al permitir la automatización de tareas complejas de reconocimiento visual. Su capacidad para aprender patrones y representaciones jerárquicas las hace ideales para una amplia gama de aplicaciones.