

ResNet

8 de junio de 2025

El incremento de capas en las redes neuronales conlleva un incremento significativo en la dificultad del entrenamiento. Al aumentar la profundidad de la red, la precisión no solo se satura, sino que puede degradarse rápidamente. Para mitigar el problema se propuso el paradigma *Deep Residual Learning*, que da origen a las redes neuronales residuales (ResNet) propuestas por ?.

El concepto clave detrás de ResNet es la introducción de conexiones de atajo o cortocircuito (*identity shortcut connections*), que permiten que las capas “se salten” una o más conexiones dentro de la red. En lugar de forzar a cada conjunto de capas apiladas a ajustarse directamente a la proyección deseada, se permite que estas capas aprendan una proyección residual, Figura 1,

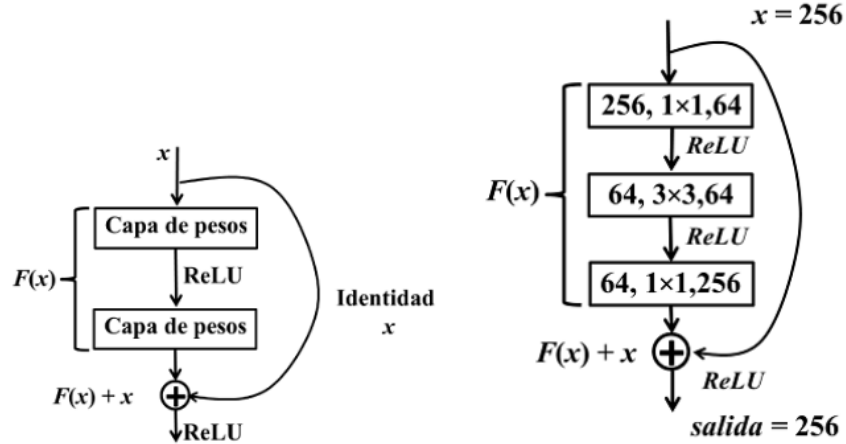


Figura 1: Conexiones de salto

Matemáticamente, si se denota la proyección deseada como $H(x)$, se define la proyección residual como:

$$F(x) = H(x) - x \quad (1)$$

De esta forma, la salida de las capas apiladas se reformula como:

$$H(x) = F(x) + x \quad (2)$$

Esto significa que la red solo necesita aprender el residuo $F(x)$, lo que es más sencillo que aprender $H(x)$ directamente. Si la proyección de identidad es la mejor solución, la red simplemente aprenderá a hacer que $F(x)$ tienda a cero, evitando la degradación del rendimiento con el aumento de capas.

Las conexiones de atajo en ResNet no añaden parámetros adicionales ni aumentan la complejidad computacional, y la red se puede entrenar de manera eficiente mediante descenso del gradiente con retropropagación. En términos de implementación, $F(x) + x$ se realiza mediante una suma elemento a elemento, lo que implica que las dimensiones de $F(x)$ y x deben coincidir. En caso de no ser iguales,

se emplea una proyección lineal W_s para hacer coincidir las dimensiones.

Para comprender mejor el funcionamiento de las conexiones residuales en ResNet, se presenta un ejemplo con un tramo de red que incluye dos cortocircuitos, A y B, cada uno con funciones residuales de 2 y 3 capas, respectivamente. El modelo considera cinco volúmenes con dimensiones específicas: $55 \times 55 \times 96$, $27 \times 27 \times 96$, $27 \times 27 \times 256$, $13 \times 13 \times 256$ y $13 \times 13 \times 384$, Figura 2,

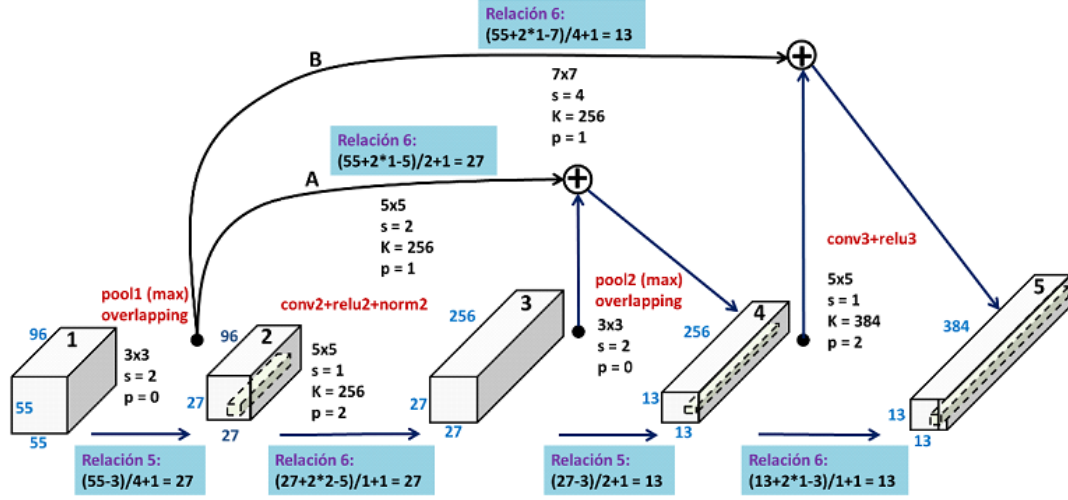


Figura 2: ResNet: ejemplo ilustrativo

En concreto para este trabajo se utiliza el modelo ResNet-50v2, segunda versión de ResNet-50 la cual se crea a partir de la red de 34 capas reemplazando bloques de dos capas por bloques de tres capas, alcanzando un total de 50 capas. Al igual que el resto de modelos con arquitectura ResNet, este modelo se estructura en bloques denominados C0, C1, C2, C3 y C4, cada uno con diferentes dimensiones de salida. La red concluye con una capa *pooling* y una totalmente conectada con 1000 categorías en la salida, usando la función *softmax*. La capa totalmente conectada se puede modificar para variar el número de categorías de salida, Figura 3,

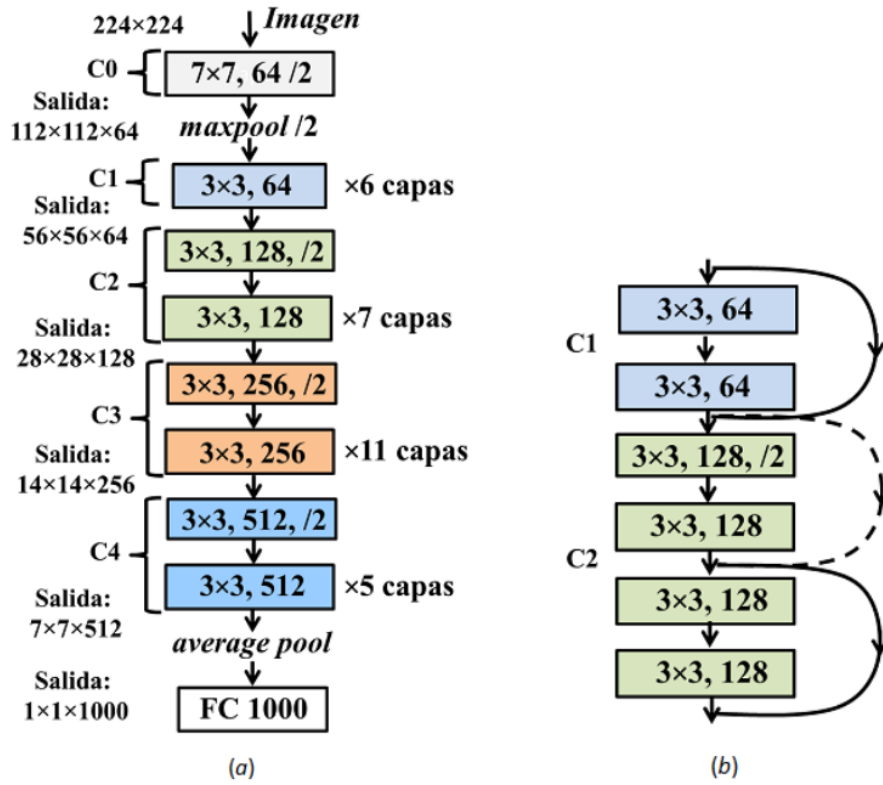


Figura 3: ResNet: (a) Modelo de 34 capas; (b) Modelo de 50 capas