Preserving Spatial Information: The Importance of Padding in Convolutional Neural Networks

Padding in a Convolutional Neural Network (CNN) is a crucial technique that involves adding extra border pixels (usually set to zero) around the input image or feature map before applying the convolutional filter (kernel). This process is essential for controlling the spatial dimensions of the feature maps, ensuring that information at the borders is not lost, and allowing for the creation of deeper, more stable network architectures.

Why Padding is Applied Before Convolution

Padding is applied **before** the convolution operation in a CNN. The input image or feature map is first surrounded by a border of extra pixels (typically zeros, in a process called "zero padding") to create a larger canvas. Then, the convolutional filter (kernel) is slid over this padded input to perform the convolution operation.

- **Preserving spatial dimensions**: Without padding, each convolution operation causes the output feature map to shrink in size. By adding padding, the output can maintain the same dimensions as the input, allowing for the creation of deeper networks without the image data diminishing too quickly.
- **Retaining edge information**: Pixels at the edges and corners of an image are used less frequently in the convolution process than pixels in the center. Padding ensures that these border pixels are covered by the filter multiple times, giving them equal importance and preventing the loss of crucial edge features.

Types of Padding

The two most common types of padding are generally referred to by their effect on the output dimension:

1. Valid Padding (or No Padding)

- **Definition:** No extra padding (zero-pixels) is added to the input image.
- **Operation:** The filter is convolved only over the valid, legal positions where it completely overlaps the input.
- **Effect:** The output feature map is **smaller** than the input image.

2. Same Padding (or Zero Padding)

- **Definition:** The input image is padded with layers of zeros such that the convolution operation results in an output feature map that has the **same spatial dimensions** (width and height) as the input image.
- **Calculation:** For a N x N input and an F x F filter with a stride of 1, the required padding (P) to achieve "Same" output size is typically P = {F-1}/{2}
- **Effect:** The output feature map maintains the original input size, preventing the volume from shrinking too rapidly.

Less Common Types (Context-Specific)

- **Reflective Padding:** Instead of zeros, the padding pixels are a mirror reflection of the border pixels in the input. This is often used to create a more realistic border for specific image processing tasks.
- **Replicate Padding:** The border pixel values are duplicated to form the padding.

Problem with Convolution Layers Without Padding (Valid Padding)

Without padding, the repeated application of convolution layers causes two major problems:

1. Rapid Dimensionality Reduction

- In a **Valid** convolution, the output dimension is calculated by N(out) = N(in) – F/{S} + 1, where N(in) is the input size, F is the filter size, and S is the stride.
- With typical filter sizes (e.g., 3x3) and stride 1, the size of the feature map shrinks with every layer (e.g., 32x32 to 30x30 to 28x28 etc.).
- In deep networks, this means the feature map size would quickly shrink to 1x1 or even zero after only a few layers, limiting the depth of the network and potentially losing valuable spatial information.

2. Loss of Border Information (Edge Bias)

- Without padding, the pixels on the **edges and corners** of the input image are covered by the convolutional filter **fewer times** than the pixels in the center.
- For example, a pixel in the dead center might be part of the convolution for nine different output pixels (with a 3x3 kernel), but a corner pixel is only used for **one** output pixel.
- This means the weights are updated less frequently for the features derived from border information. As a result, the network learns to pay less attention to the edges,

and the features extracted from the edges are "washed away" too quickly. This leads to a model that is heavily **biased toward the center** of the image.

Effect of Padding on Input Images

The primary effects of padding (specifically **Same Padding**) are to solve the problems mentioned above:

- **Output Size Preservation:** Padding ensures that the spatial dimensions (height and width) of the output feature map remain the **same** as the input image. This allows researchers to build very **deep CNN architectures** without the feature maps shrinking to an unusable size.
- **Preservation of Edge Features:** By adding zero-pixels around the border, padding ensures that the filter can pass over the edge pixels a greater number of times. This helps to adequately utilize the information at the borders and prevents the loss of important features that might be located near the edges of the image.
- **Simplifies Architecture Design:** By guaranteeing that the input and output sizes are the same, padding makes it much easier to design networks, especially complex architectures (like ResNet or U-Net) that involve **skip connections** or **concatenating** feature maps from different layers, as the dimensions are guaranteed to align.