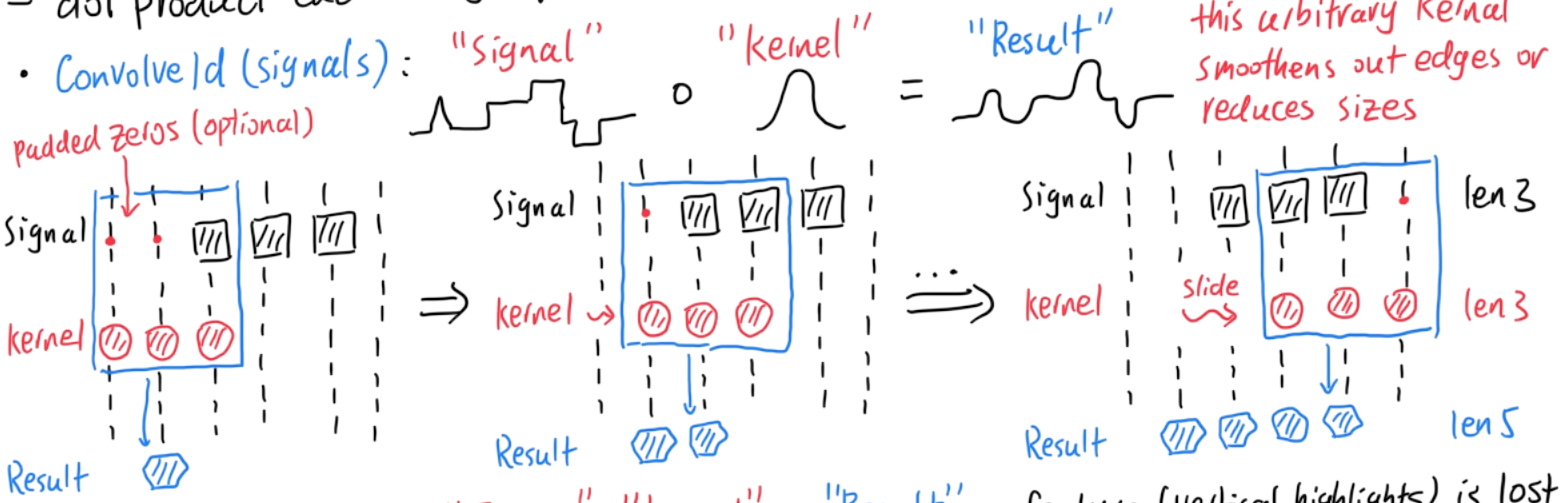
18) Convolution & Transformation

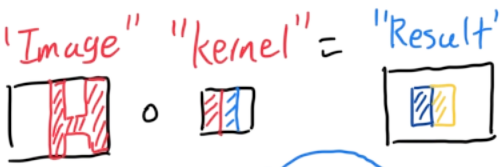
Convolution

- dot product each image patch by a kernel

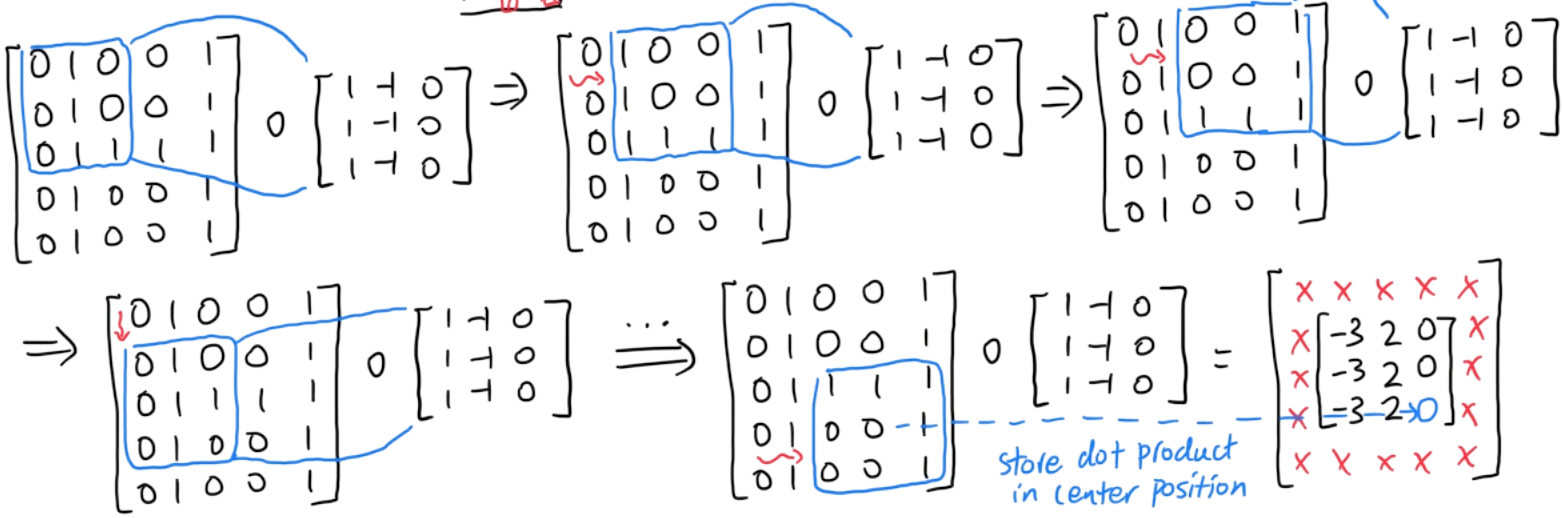
- Convolve1d (signals):



- Convolve2d (images):



feature (vertical highlights) is lost at the edges due to no padding



- Convolve3d (Images): process is similar to 1D & 2D

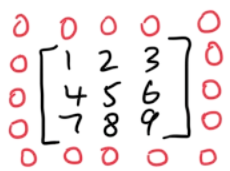
- In DL, a bias term will be added to the "result"

More on Kernels & Feature Maps

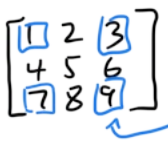
- kernel: small filter that extracts one feature across different images → which will then be used for classification
- In DL, kernels are learned through GD rather than scientific handcrafting
- Applying N-kernels outputs N feature maps, or layers/channels
  - shape: (# channels, height, width) in PyTorch

Convolution Parameters

① Zero Padding: insert P rows & cols of zeros to the image to increase the size of convolution result



② Stride: skip convolving the image by S units to decrease the size of convolution result



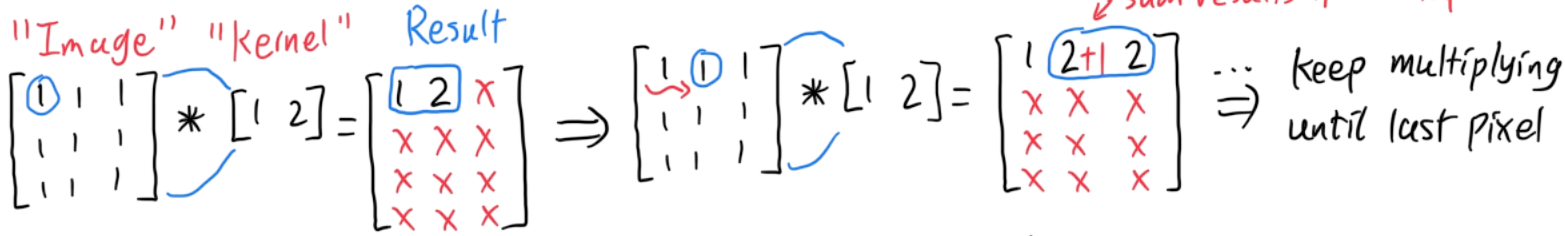
$S_h = S_w = 2$ (default: 1)

- Formula on result size: $N_h = \left\lfloor \frac{M_h + 2P - k_h}{S_h} \right\rfloor + 1$, $N_w = \left\lfloor \frac{M_w + 2P - k_w}{S_w} \right\rfloor + 1$

. Denote N as output, M as input, h = (height), w = (width), k = kernel size ∝ 2n+1

Transpose Convolution

- Scalar-multiply a kernel by each image pixel; deconvolution



· Used for CNN autoencoders & Super-resolution CNN

- Formula on result size: $N_h = S_h(M_h - 1) + k_h - 2P$, $N_w = S_w(M_w - 1) + k_w - 2P$

stride spaces out input row/col by 0s

padding cuts off output edges with 0s

Pooling

- Take each non-overlapping patch of image & compute their ___:

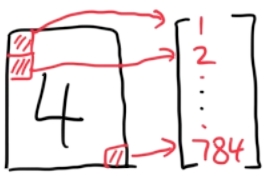
① Avg: result image is smoothened (prevent outliers/noisy data)

② Max: sharpened with important features (assist sparse data)

- Purpose: ① Downsampling & Feature selection

② Increase Receptive Field: # pixels that are mapped to a node

· FFN unit only has RF = 1 (treat each pixel independently)



Each pooling layer in CNN increases RF (attention) of image

- nn.MaxPool2d vs. nn.MaxPool3d on 3D-Image (latter removes depth dimension)

Pooling Parameters

① kernel size: # pixels in the pooling window (usually 2)

② Stride: # pixels to skip for window (usually same as ① to prevent overlapping)

- Formula on result size: $N = \left\lfloor \frac{M-K}{S} \right\rfloor + 1$ ← 'floor' mode: drops leftover window

$$N = \left\lceil \frac{M-K}{S} \right\rceil + 1$$ ← 'ceil' mode: operate as normal

} only matters if $K \neq S$

Image Transform

— Grayscale, Color change, flip, Zoom/crop, ...

- Why?: ① Adapt image size to pre-trained CNN ② Data augmentation
- Workflow: