Convolutional Neural Networks

EE698V - Machine Learning for Signal Processing

Vipul Arora

Announcements

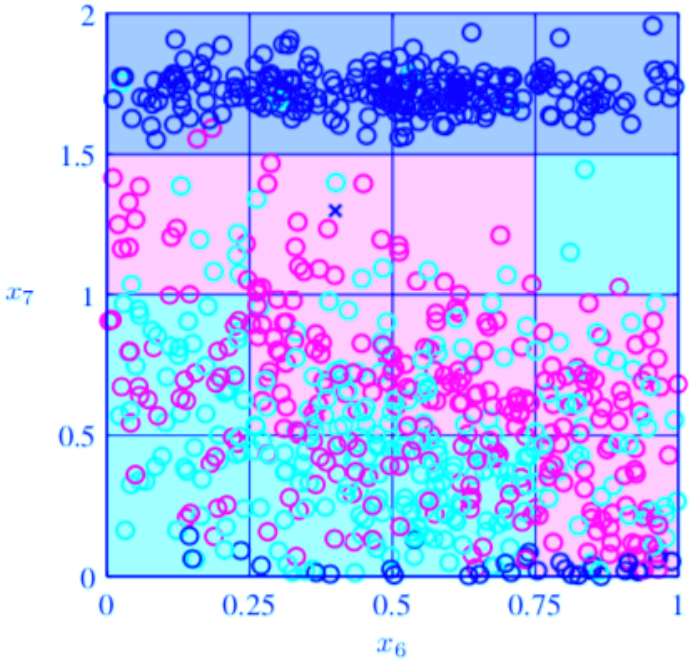
- Theory Quiz on Oct 31 (Thurs) or Nov 1 (Fri)
- Coding Quiz on Nov 3 (Sun)

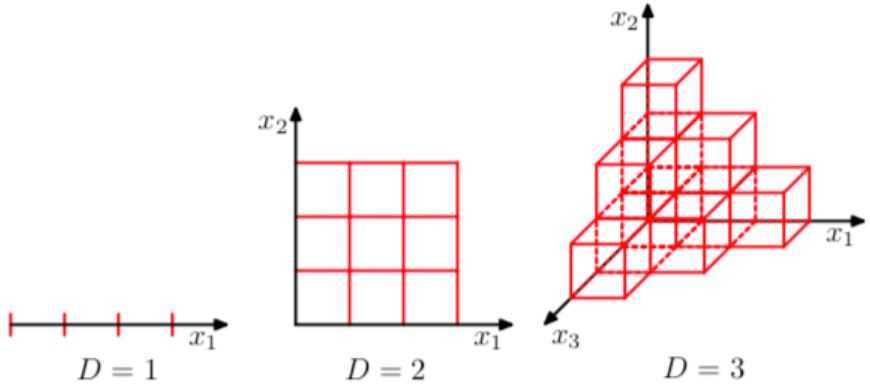
Curse of Dimensionality

- We have been studying about dimensionality reduction
- Let us try to appreciate the scale of the problem of large dimensionality

- **Reference**: PRML Section 1.4

For Classification



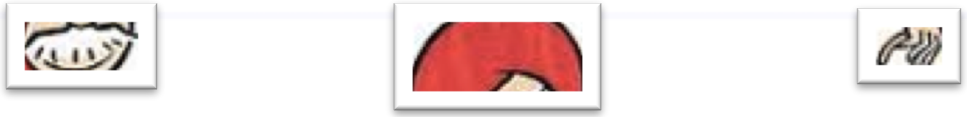


D	$N = 4^D$ (no. of cells)
2	16
3	64

For Images

Template Matching

- We can find blobs/regions that are likely to classify the image



- Cross-correlation of these templates with the test image will give a matching score

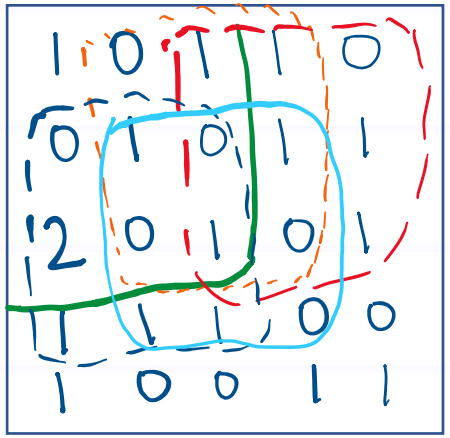
Finding optimal templates

- We can find these templates optimally from the data discriminatively
- Recall, cross-correlation
- $H = X \otimes W$

2D Convolution: actually cross-correlation

- $H[i_1, j_1] = \sum_{k=0}^{K} \sum_{l=0}^{K} X[i_1 s + k, j_1 s + l]\, W[k, l]$
- s is the hop size (stride)
- K is the kernel size (template size)
- $H[i_1, j_1]$ will be maximum when match occurs

2D Convolution



$\otimes$

1	0	0
0	1	0
0	0	1

=

3	0	3
1	2	0
3	2	2

hop size, $h = 1$

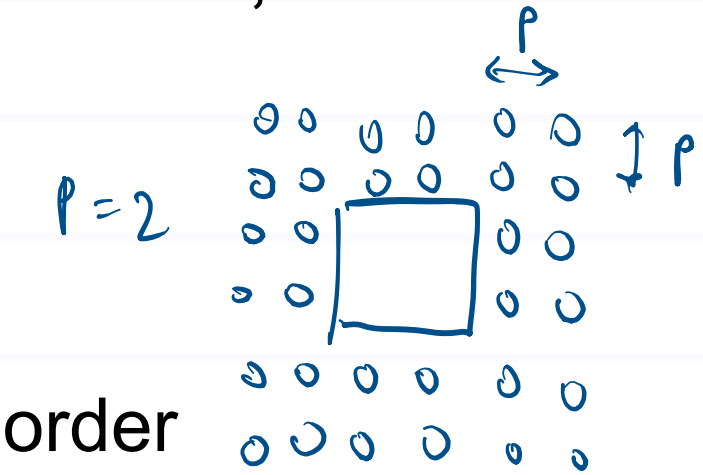
- Use hop size = 1 or 2

3	3
3	2

$h = 2$

Question

- Given an image of size $I \times J$ and a kernel of size $K \times L$, with a hop of s and s' along the two dimensions, what is the size of the output:
  - without any zero padding
  - with a zero padding of width P along each border

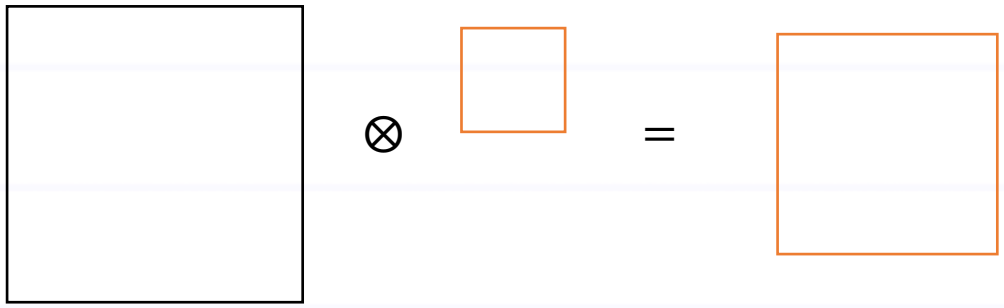


How to optimize kernel for classification

- Define multiple kernels, say N
- Find H for each kernel
- Give these as input to a neural network classifier
- Train end to end !!

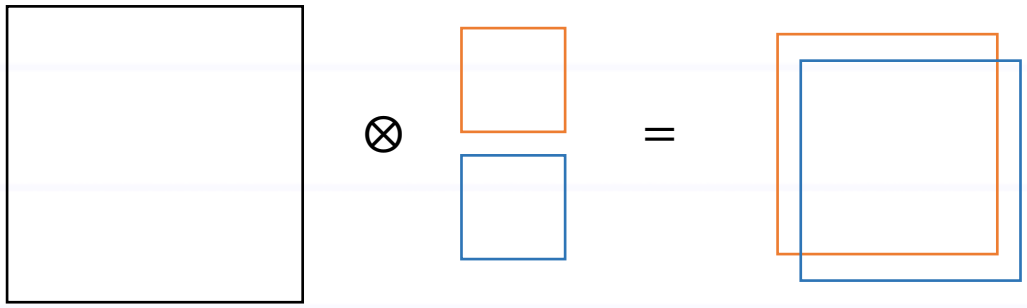
How to optimize kernel for classification

- Define multiple kernels, say N
- Find H for each kernel
- Give these as input to a neural network classifier
- Train end to end !!



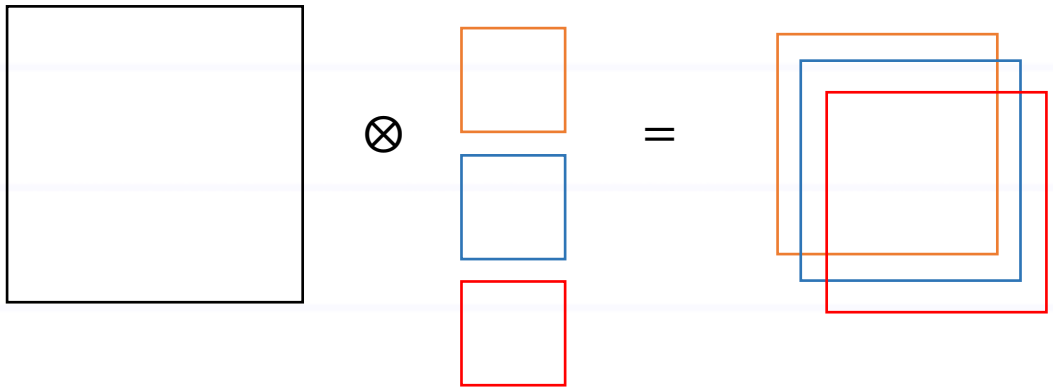
How to optimize kernel for classification

- Define multiple kernels, say N
- Find H for each kernel
- Give these as input to a neural network classifier
- Train end to end !!



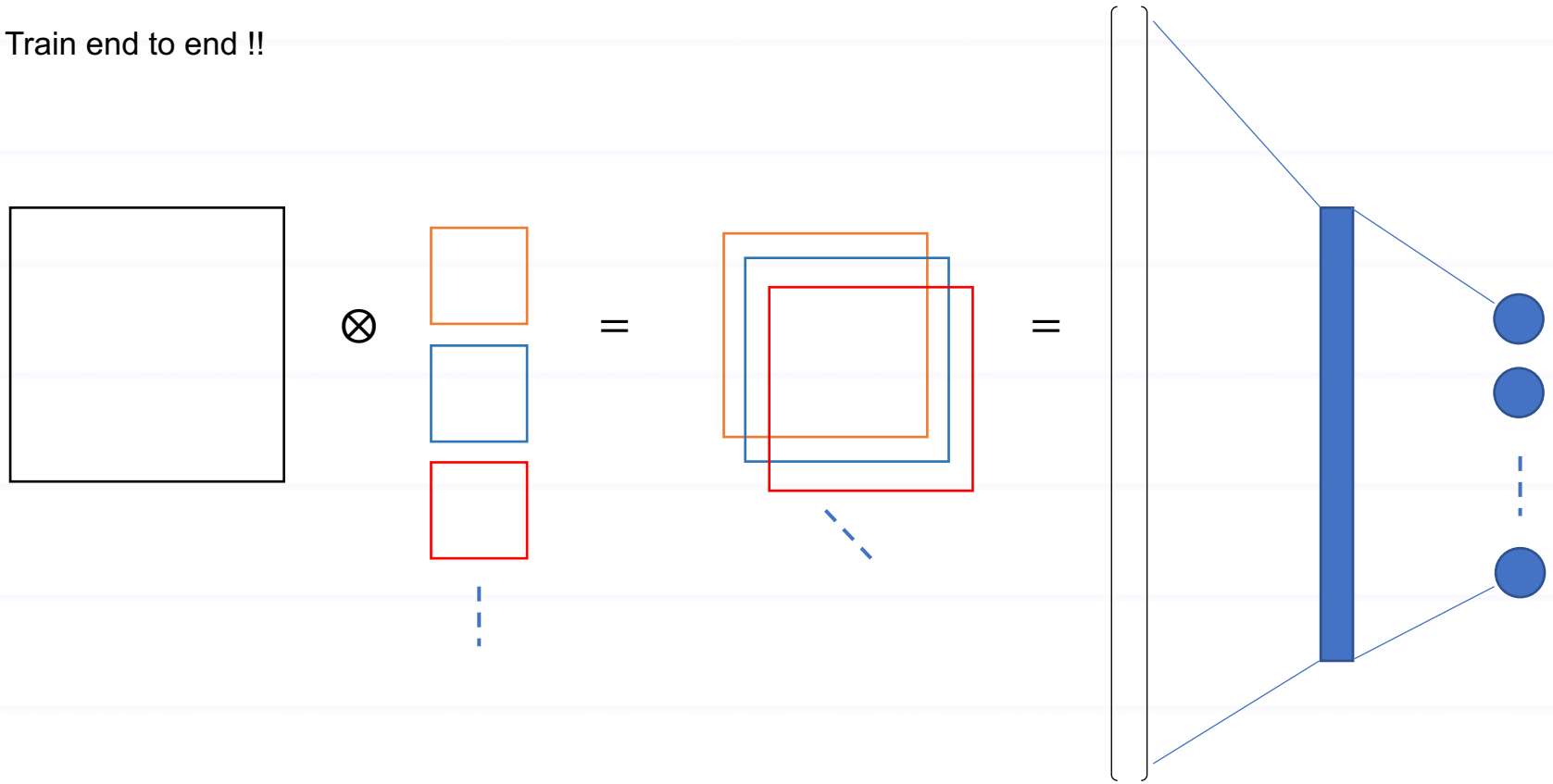
How to optimize kernel for classification

- Define multiple kernels, say N
- Find H for each kernel
- Give these as input to a neural network classifier
- Train end to end !!



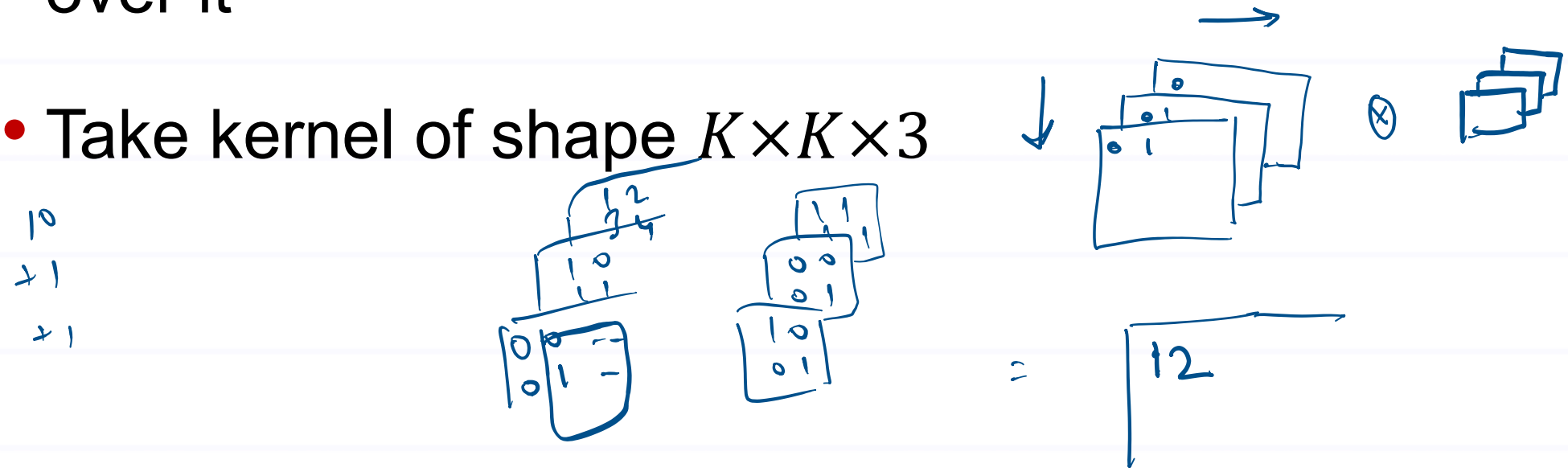
How to optimize kernel for classification

- Define multiple kernels, say N
- Find H for each kernel
- Give these as input to a neural network classifier
- Train end to end !!

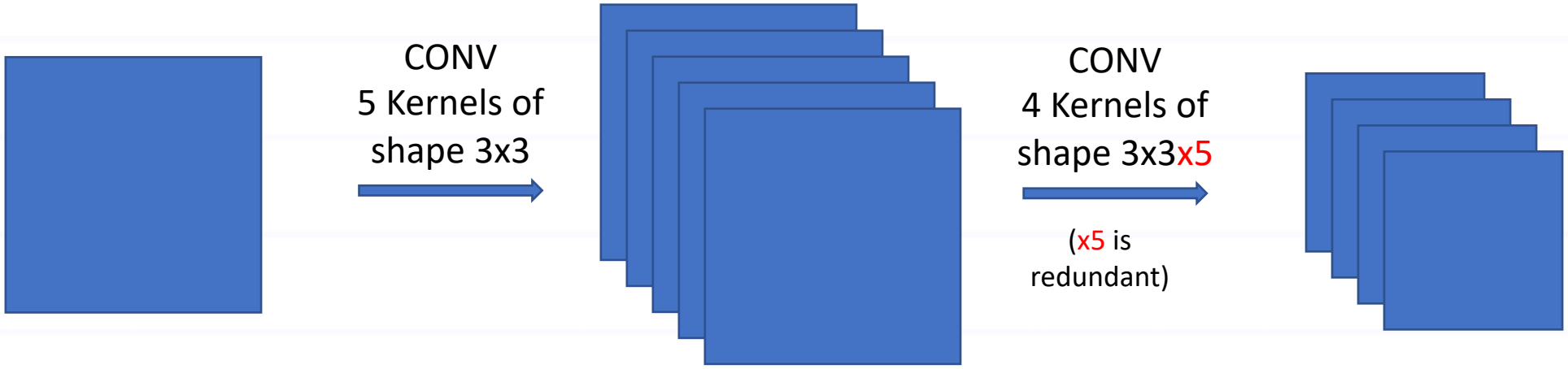


How to deal with RGB images?

- Input is of size $I \times J \times 3$
- The 3 channels are correlated, so no point in convolving over it
- Take kernel of shape $K \times K \times 3$



We can stack layers of kernels



- This is known as
  - Convolutional Neural Network
  - ConvNet
  - CNN

CNNs

References: (highly recommended)

- https://youtu.be/j41cU9ad7SU

- https://youtu.be/P5OgPz_h9CE

Standard concepts

Learn the following concepts from the reference videos. No need to memorize the architectures of the popular CNNs.

- Pooling
- Drop out
- Flattening
- Find the number of parameters in each layer
- Find the computational complexity of each layer
- The problem of vanishing gradients

Standard practices

- Generally odd sized kernel are used. E.g. K=3 or K=5 or K=7
- Use padding P=(K-1)/2 and s=1 for output size to be same as input size
- ReLU activation is used to take care of vanishing gradients