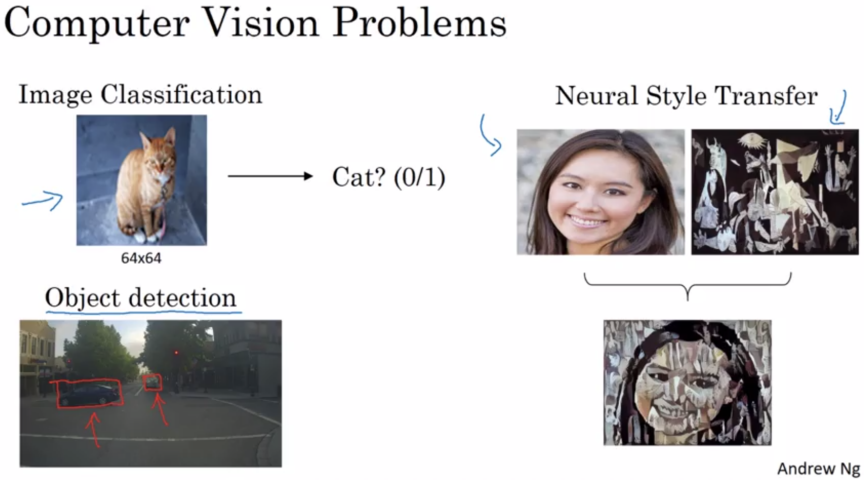
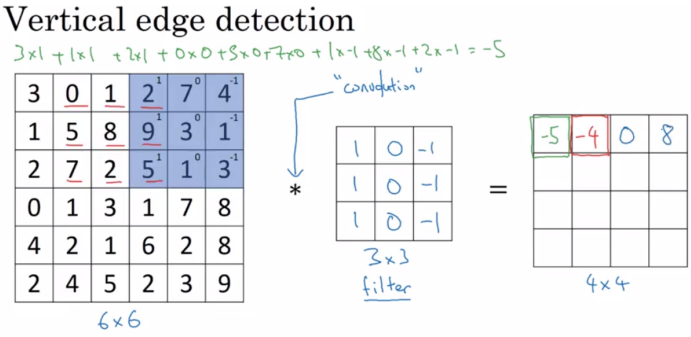
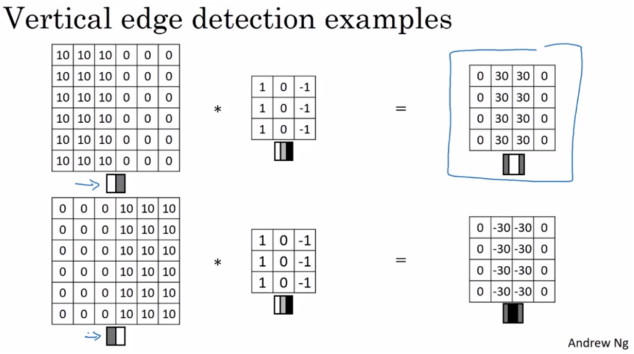
deep Learning_c4_week1

Created:	19/02/21 17:19	**Updated:**	19/02/22 13:26

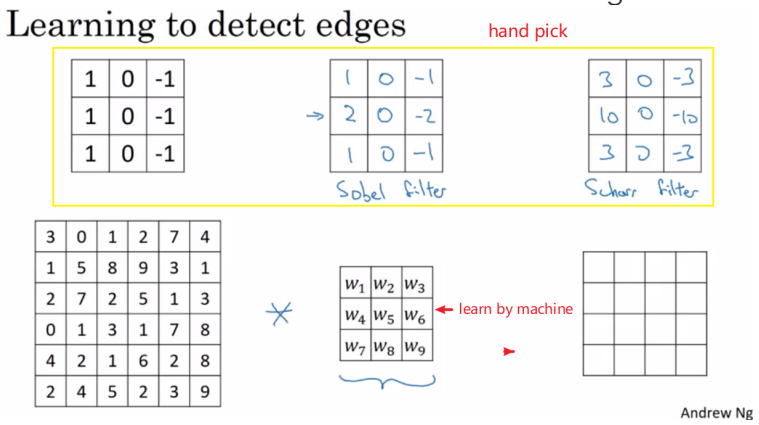
- # introduction
  - 3 main works in computer vision
    1. Image classification.
    2. Object detection.
       - Detect object and localize them.
    3. Neural style transfer
       - Changes the style of an image using another image.
    4. 

- # edge detection
  - 

  - 

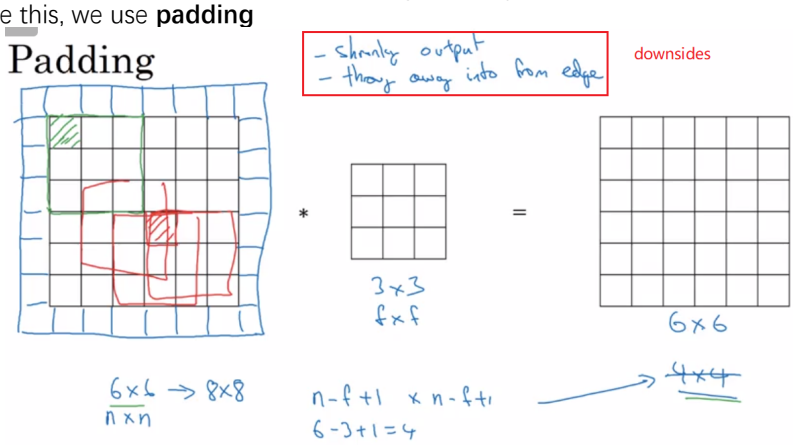
  - convolution operation
    - TensorFlow `tf.nn.conv2d`.
    - keras: `Conv2d`
  - filter

- we don't need to hand craft but instead using DL to learn them.

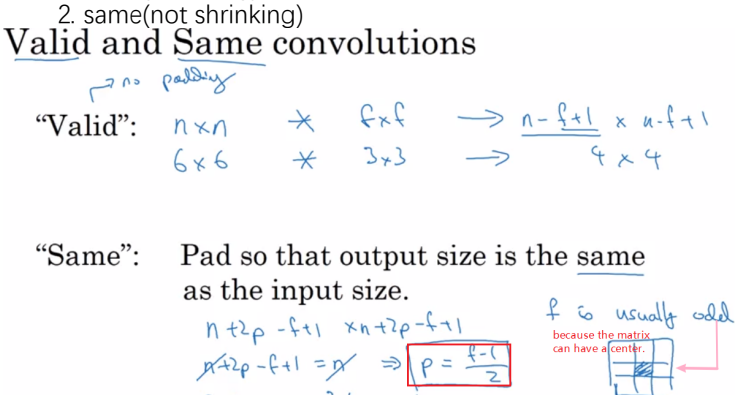


padding

- in previous slide, we see two **downsides** of convolution
  - the matrix keep shrinking.
  - throwing away a lot of information that are in the edges. (central units would be counted more times than the edges units.)
- to solve this, we use **padding**



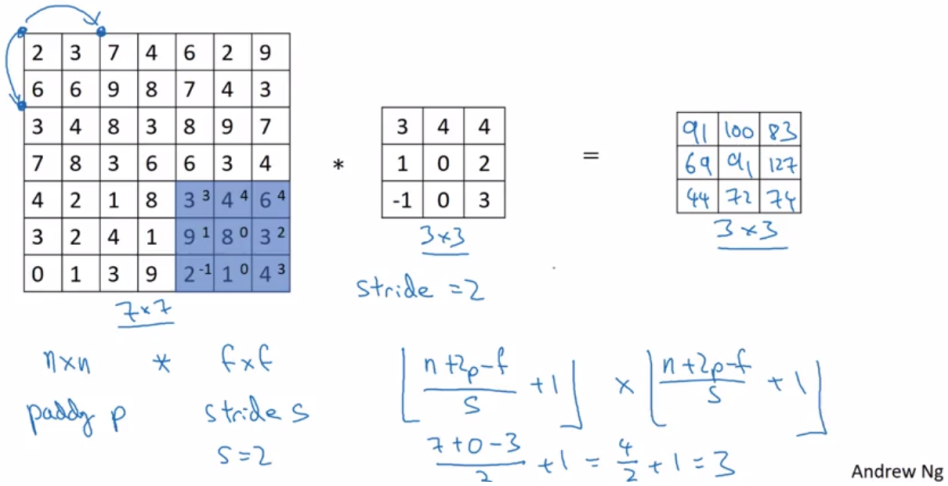
- formula here:
  - padding amount p
  - two kinds of convolutions
    1. valid (shrinking)
    2. same(not shrinking)



strided convolution

- stride s

- Strided convolution



Andrew Ng

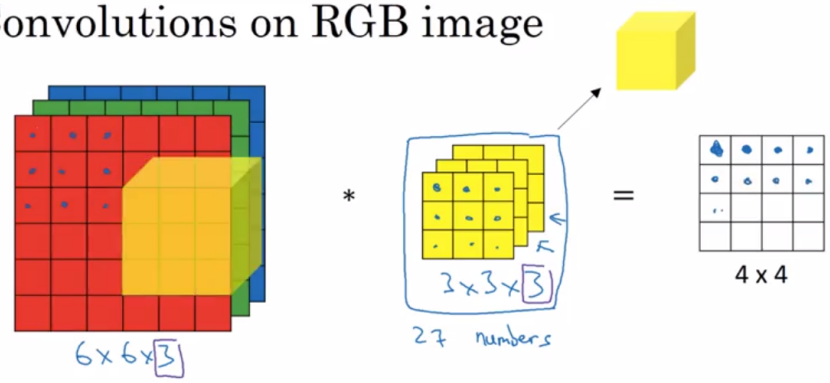
- Summary of convolutions

$n \times n$ image $\quad f \times f$ filter

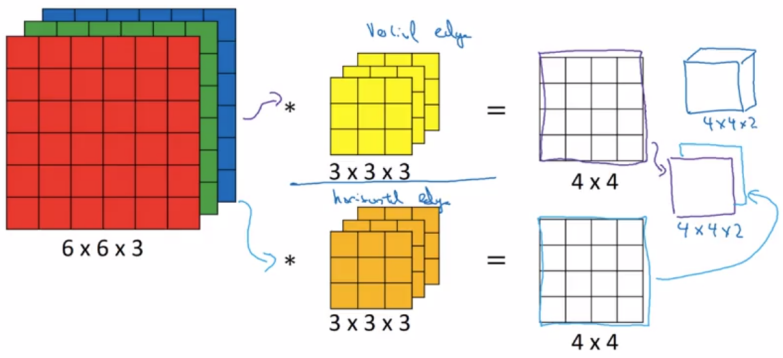
padding p $\quad$ stride s

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

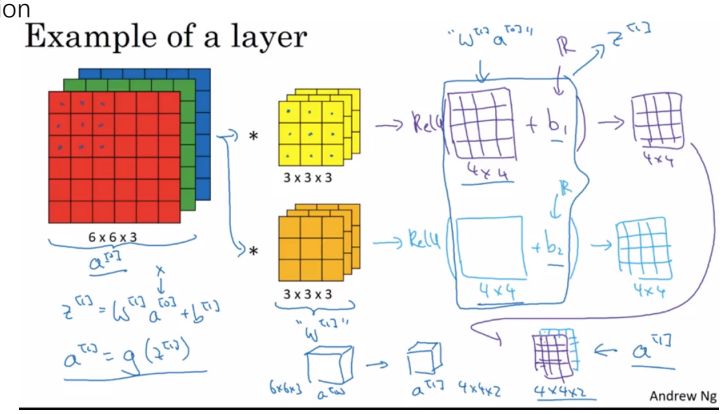
- over volume
  - we call each layer in the third D as **channel**
  - Convolutions on RGB image

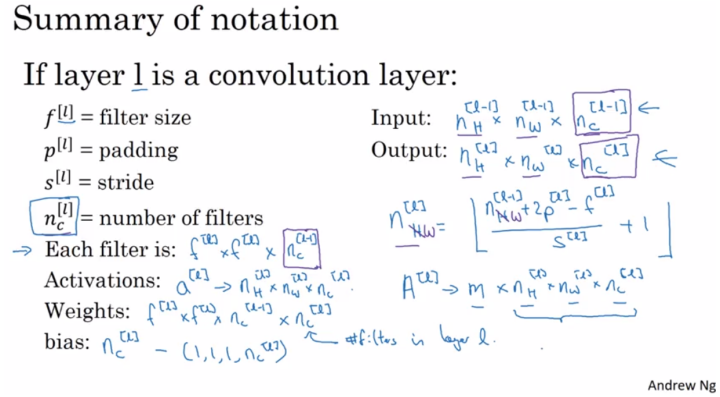


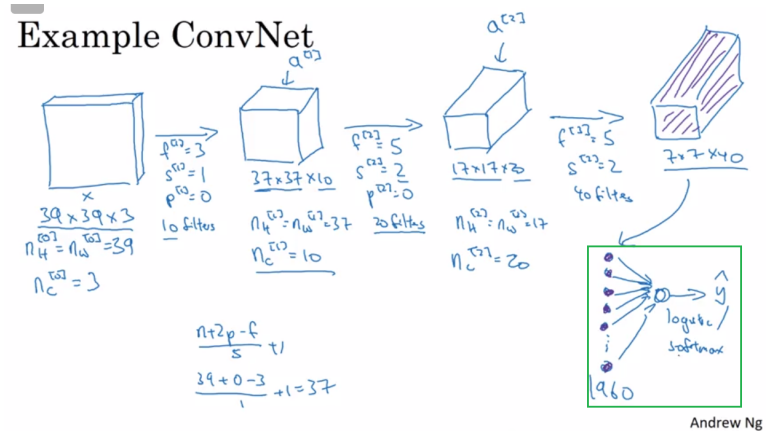
  - Multiple filters



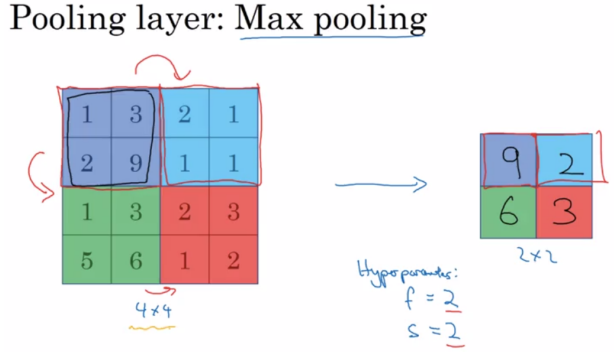
- one layer of convolutional NN

- intuition
  - 

- notation
  - 

- example
  - 

pooling

- intuition of **Max pooling**
  - **select the biggest values in each region**
  - 

- **average pooling**
  - taking the averages of the values instead of taking the max values.
- summary: fixed layer. No parameter to learn.

Summary of pooling

Hyperparameters:

f : filter size — f=2, s=2 / f=3, s=2

s : stride

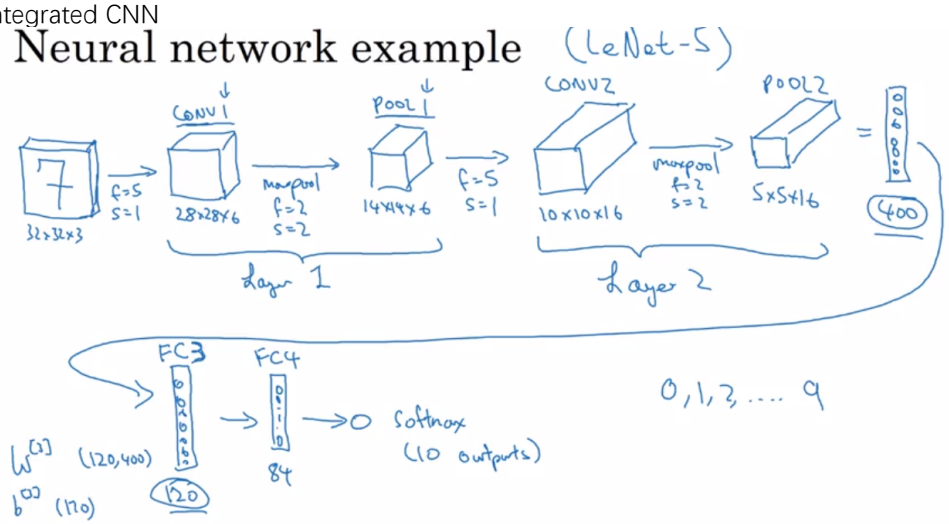
Max or average pooling

conventional values

~~p: padding~~

No parameters to learn!

- Ex. a integrated CNN



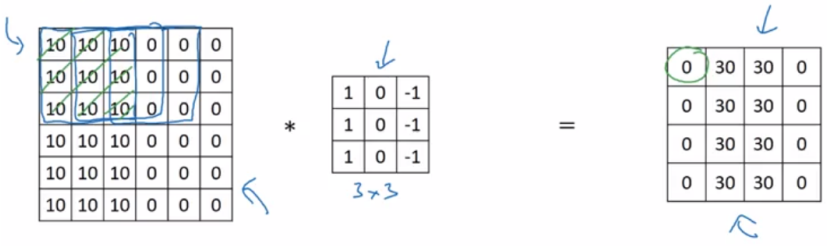
Andrew Ng

- # *FC* here means fully connected layer

why using convolutions

- if using fully ANN, their are too many parameters to learn for image
- convolutions
  - Sharing
  - Sparsity

Why convolutions



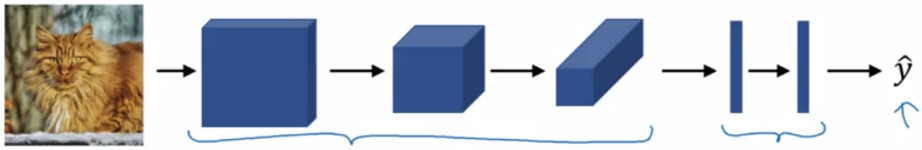
Parameter sharing: A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.

→ **Sparsity of connections:** In each layer, each output value depends only on a small number of inputs.

Andrew Ng

Putting it together

Training set $(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})$.



Cost $J = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

Use gradient descent to optimize parameters to reduce J