This is the personnal note for W1 of CNN at COURSERA

1. Computer vision problem

The procesuss of using a filter or kernal to modifiing the original pictures is called "convolution":

- in Python: conv-forword
- in tensorflow: tf.nn.con2d

e.g

```
[[1 1 -1],
 [1 0 -1],
 [1 0 -1]]
```

is a vertical edge detector #enhence the vertical edges so after the convolution with this filter all the charactor with vertical property will show up more clearly

2. More Edge Detection

Sobel filter

```
[[1 0 -1],
 [2 0 -2],
 [1 0 -1]]
```

Schass filter

```
[[ 3 0 -3 ],
 [10 0 -10],
 [ 3 0 -3 ]]
```

3. Padding

- size of pictures : $n * n$
- size of filters : $f * f$ so the output after the convolution size of $(n-f+1) * (n-f+1)$

So there are clearly two results of doing convolution

- Shrink output
- Throug away a lot of information from edges

In order to fix this problem:
we pad the image $\longrightarrow p = padding$
if we take a padding = $p = 1$
so the size of image is transfered to $(n + 2p) * (n + 2p)$
in return, out put size is still $\longrightarrow n * n$

Valid and Same convolutions

1. Valid $\Leftrightarrow$ no padding
2. Same $\Leftrightarrow n + 2p - f + 1 = n \Rightarrow 2p = f - 1$
   filter size is usually odd #it's nice to have a centre pixel

4. Strided Convolutions

Strip = 2
jump strip times 跳过一行计算，中心直接隔开一个

$$(\frac{n + 2p - f}{s} + 1) * (\frac{n + 2p - f}{s} + 1)$$

We can also note it as cross-correlation

5. Convolutions Over Volume

- **on RGB images there are channels!**
  so we make the filtre of 3 channels, too. we make it like a filter cube and the output is just 2D
- **Multiple filters**
  When we have many different filters at the same time
  $\Rightarrow$ make the output of different filters into different channels of the output.

6. One layer of a convolutional NN

If layer l is a convolutional layer:

- f^l = filter size of layer l
- p^l = padding
- s^l = stride
- n_c^l = number of filters

7. Simple convolutional network

7.1 Types of layers in a convolutional network

- Convolution
- Pooling
- Fully connected

7.2 CNN Examples

- Neural network example

> LeNet - 5

- Pooling layers don't have weight

7.3 Some Excellent Examples

- *Why Convolutions?*
  - Conv layers have much smaller number of parameters
  - 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 connection:
    In each layer, each output value depends only a small numbers of inputs.