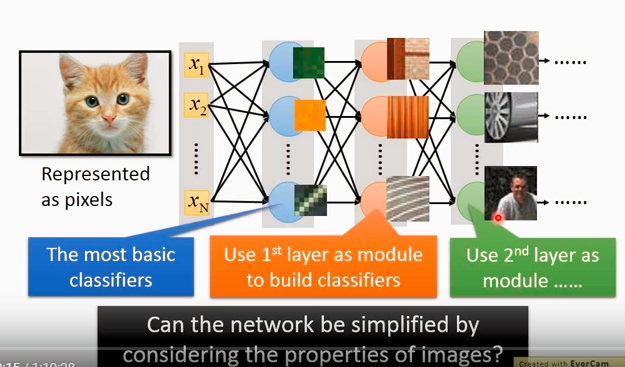
**Why CNN for image?**

Pre: in lecture 7,8,9, it mainly introduce how to use Keras, but I used tensorflow, so I skip those parts.

Supposing we want to classification the image busing neural network.



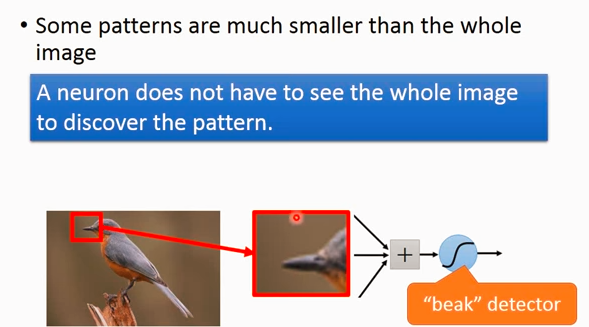
Every neural represent a simple classifier, e.g. if green, it activate the first neural, if orange, the second one. In the 2nd layer, it represent more complicated feature, e.g. texture of wall, of wood or of wheel.

But if the image is 100x100x3 and there are 1000 neurals in the 1st layer, then there are 1000\*30000 parameters. Too much.

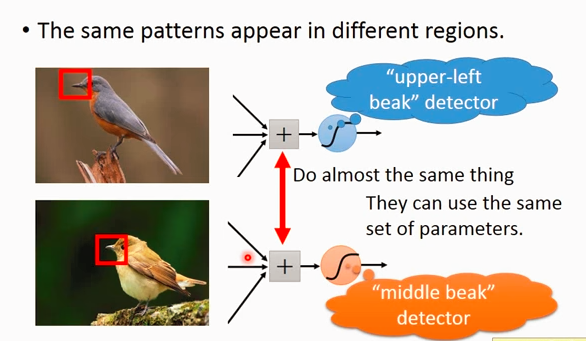
So what CNN do is to simplify this.

Why CNN.

1. When it see some small part of the scene, it know what is that



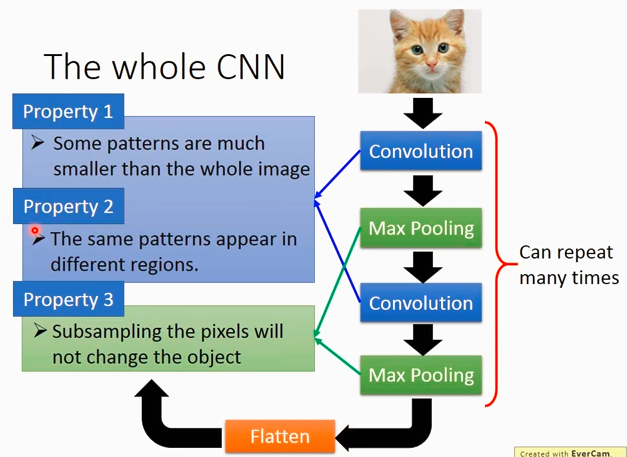
1. We don't care where the beak appear in image, but once it appear, we know it's a bird, so that they can share the same paremeters.



1. Subsampling does not affect the ability to recognition, but less parameters are required.



Finally,

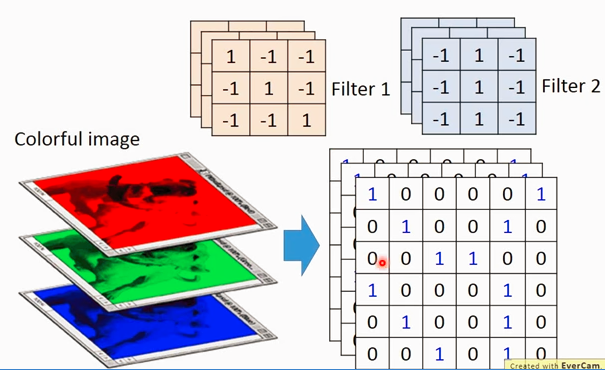


Question:

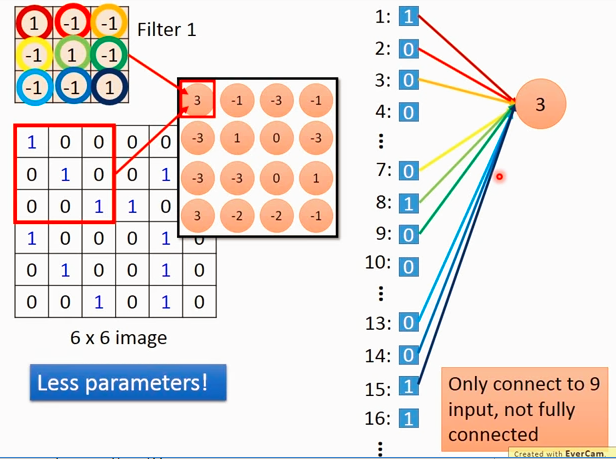
If the beak of the bird have different scales, this may not work well?

Emm..yes, but someone(can’t recognize the name in the lecture) build a network before CNN. That network can rotate, scaling some part of the image, after that, the transformed images are put in CNN, the result is better.

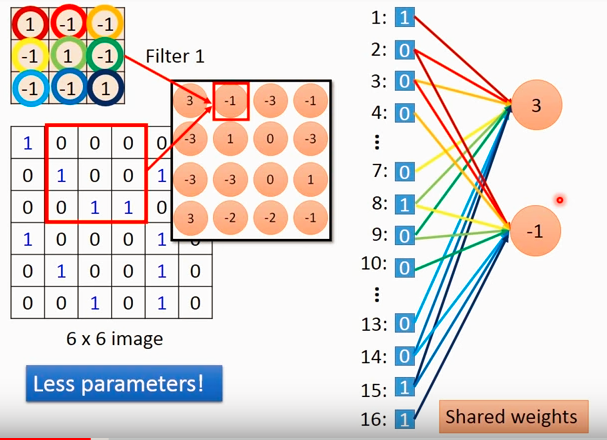
When we doing convolution/filter for RGB images:



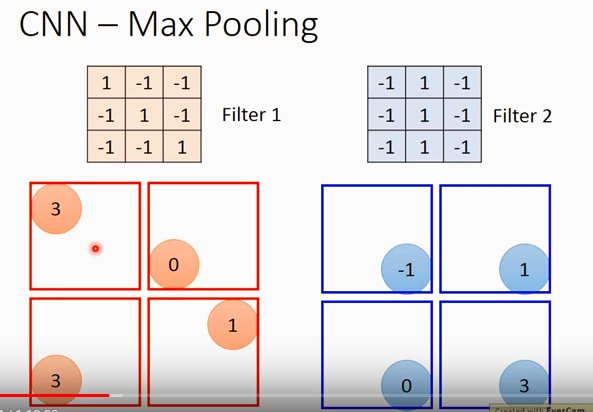
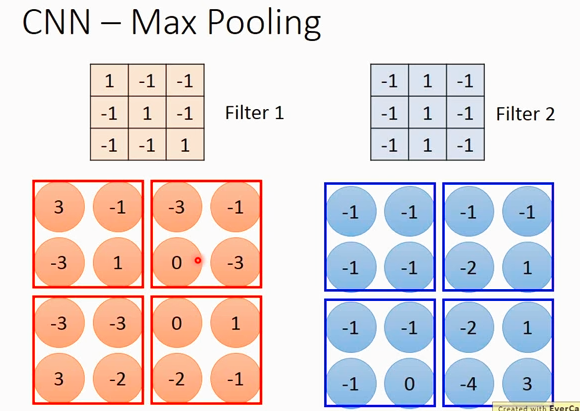
We can reshape the 2D patch to 1D vector.



In fully connected network, every neural has its own weight, but here they share the same weight, thus less parameters are required.



Then, max pooling or mean pooling to subsampling.

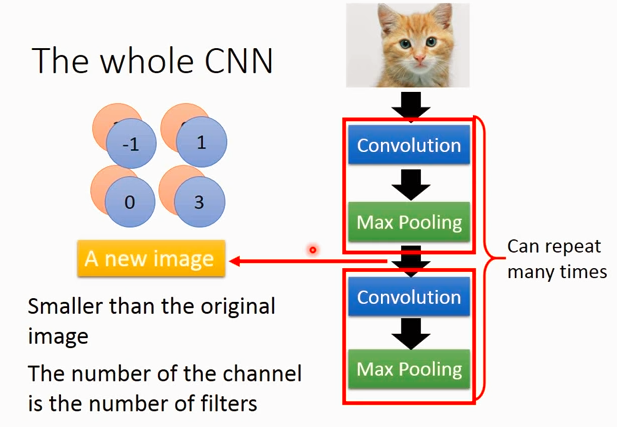


Conv + Max pooling



The depth of the image depend on the number of filters.(if 2 filters, then 2 dimension as shown)

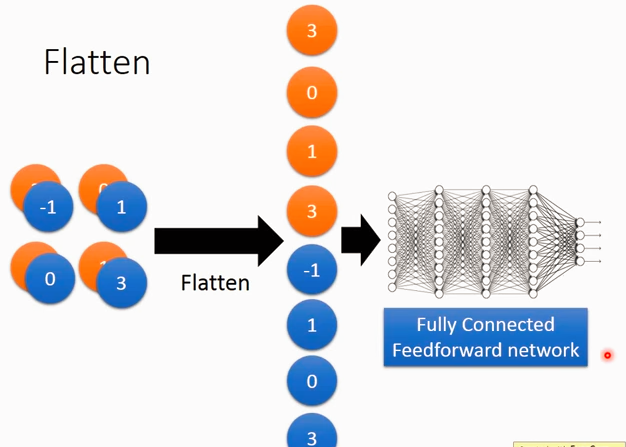
Important!



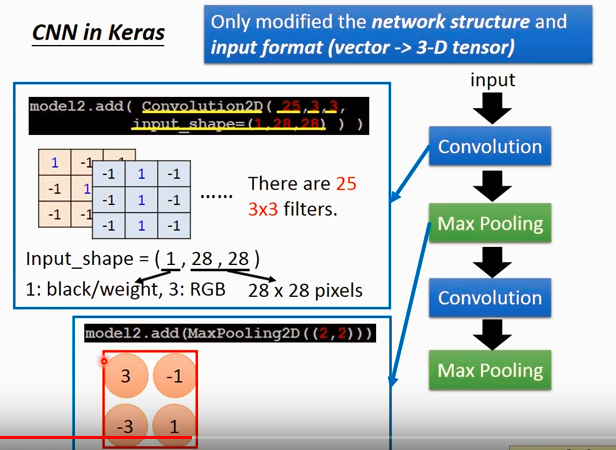
Supposing there are 20 filters in the 1st Conv, so it has 25 depth output.

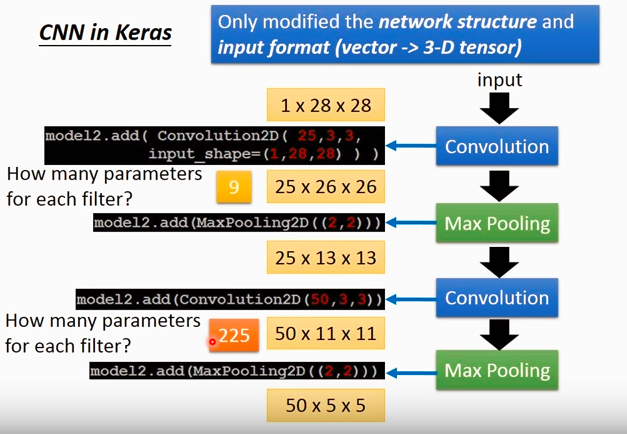
Supposing there are 50 filters in the 2nd Conv, does it have 20\*50 depth output?

NO, the filter is a cube, so the output of 2nd Conv has 50 depth.



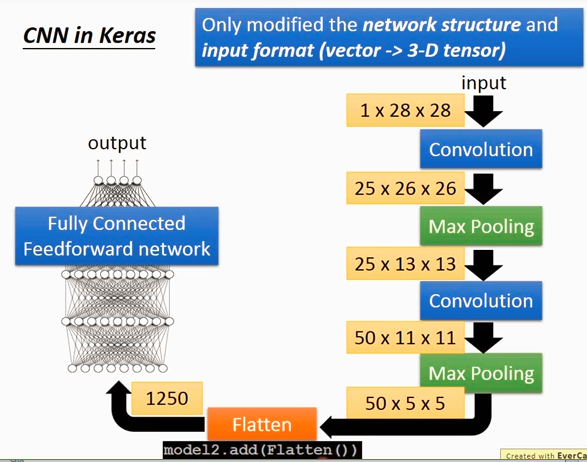
Well, although I don't use Keras, but can get some idea of how data are organized.





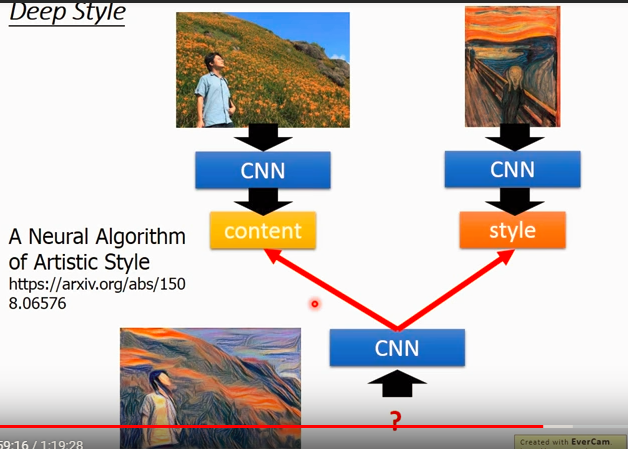
Supposing the input image is 28\*28\*1, the filter size(2D) is 3\*3, after first Conv, each image layer is 26\*26 after convolution, 25 filters in total so output is 25\*26\*26. After max pooling, the size half but depth not change so 25\*13\*13. Similar goes for the 2nd.

What need to be point out is that, the parameters for each filter in the second Conv is 3\*3\*25 since there are 25 layers for each image.



General idea of some interesting applications:

1. Style transform:



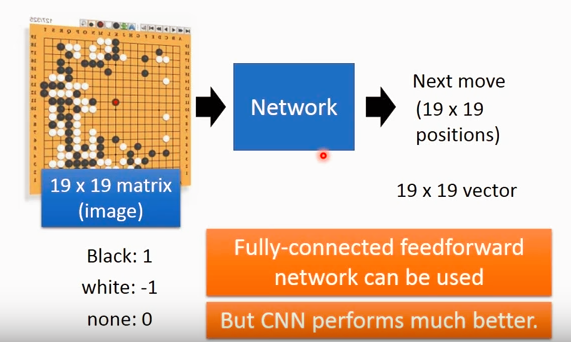
We want the content of the left image and the style of the right image.

The content is represented as the output after filtered.

The style is represented as the correlation between output.

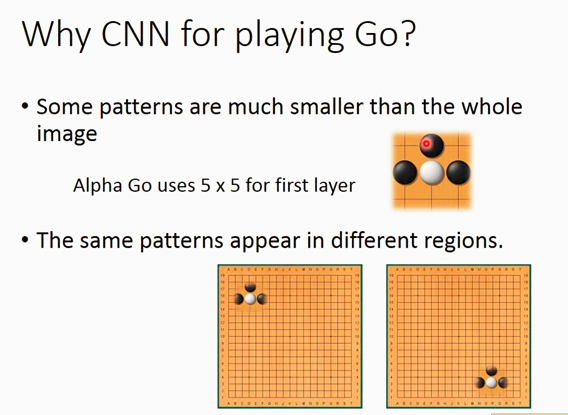
finally, we create a image that have similar output as the left image while have similar correlation between outputs as the right image, then the amazing image can be produced.

1. Weiqi

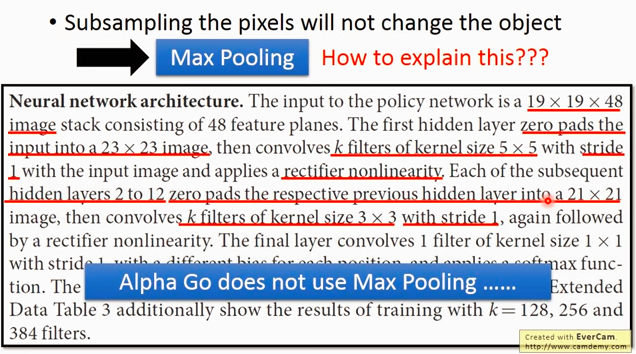


When can we use CNN?

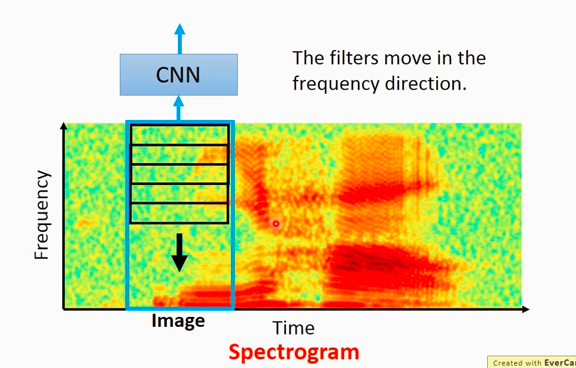
When it share similar features as image.



But, the 3rd point is -> max pooling?(subsampling)



1. Speech

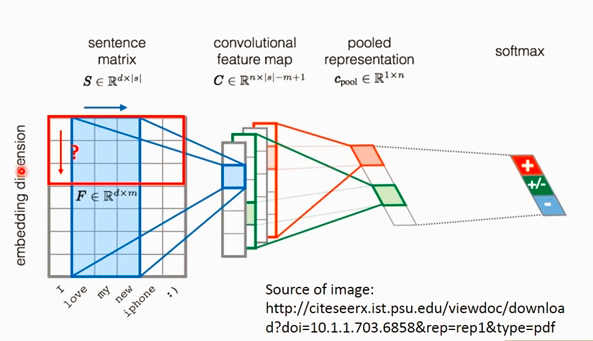


The filter shift along frequency dimension.

Why?

1. Different people saying the same word may have different frequencies, but the “shape” is the same, just have an offset in frequency dimension. (e.g. man may have lower frequency while women higher)
2. After CNN, it always followed with LSTM, which consider the time series, so here just consider frequency series only.
3. Text: if the sentence positive or negative?

Each word is represented by a vector, then we can form a 2D “image” using all these vectors. Then the filter is moving along time series but not along embedding dimension.



That’s all for the class, can’t wait for programming! ;P