- Structure helps learning

- Statistical Invariance

  - Translation invariance

    e.g. detect cat in image at different locations

  - weight sharing

    share ~~the~~ weights of inputs ~~that~~ contain ~~the~~ same information

    statistical invariance

    Image → CNN Text → RNN

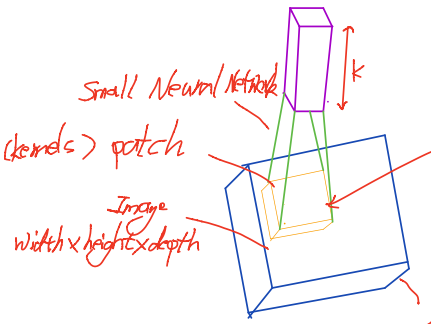
- Convolutional Networks (CONVETS)

  neural networks ~~that~~ share ~~their~~ parameters across space

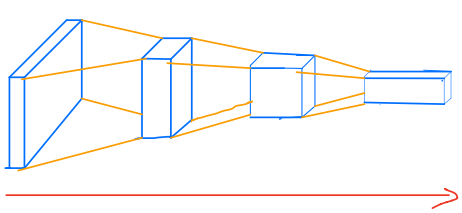
  e.g. Image has width, height, and depth (color channel)

  - Convolutions

    Instead of having stack of neural network layers, having stach of convolution layers ⟹ CNN



patch with smaller size → fewer weights (shared across space)



→ classifier

spatial info squeezed out
only parameters that map
to content of image remain

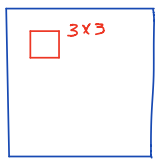
Regressively squeeze ~~the~~ spatial dimentions while increasing the depth, by copply patch with same parameters on different part of the data with fixed step size.

- stride:

  step size ~~that~~ ~~the~~ patch is moved

  - Valid Padding: moving patch doesn't go pass ~~the~~ boundrary
  - Same Padding: patch go across edge, pad with 0 at out of edge data points, such that ~~the~~ output size is exactly ~~the~~ same as input

- Strides, depth & Padding example



Input depth: 3
output depth: 8

output

padding	stride	width	Height	Depth
same	1	28	28	8
valid	1	26	26	8
valid	2	13	13	8

- Convolutional Network

  classifier: multi-layer neural networks

  chain rule with shared weights. $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w}(x_1) + \frac{\partial L}{\partial w}(x_2)$

- Advanced Convnet-ology
  - pooling

    stride of 2 with valid padding is very aggressive, which loss a lot of information. Consider skip 1 in every 2 convolutions: Take small strides (e.g. 1), and the take neiboring convolutions and combine them.

    - max pooling

      At every point on the feature map, look at small neighborhood around that map, and compute the maximum of all the responses arounded it.

      Advs:
      - Don't add extral parameters
      - often more accurate

      cons:
      - more expensive
      - more hyper parameters, e.g. pooling size. pooling stride

- Typical CONVNET structure of max pooling

CLASSIFIER
FULLY CONNECTED NN
FULLY CONNECTED NN
MAX POOLING
CONVOLUTION
MAX POOLING
CONVOLUTION
IMAGE

First famous model using above architecture is "LENET-5" by YAN LECUN, 1998

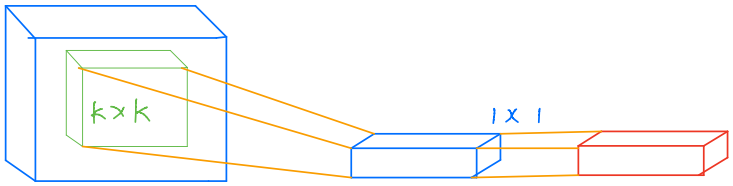
Similar modern version: "ALEXNET" by ALEX KRIZHEVSKY In 2012 won ImageNET challenge

- Average Pooling
  Similar to max pooling, but instead taking max, take average

- 1 x 1 Convolutions

  patch size : 1 x 1

  classical patch (k x k). only a linear classifier



Average Pooling + 1 x 1 convolutions ⇒ successful

- Inception Module

  Instead of having a single convolution, you have composition of: average pooling followed by 1 x 1 convolution;

  1 x 1 convolution;

  1 x 1 followed by 3 x 3

  1 x 1 followed by 5 x 5

  At the top, simply concatenate them together

  You can choose these parameters in such a way that the total parameters of your model is yet very small, and the model performs better than you have a simple convolution.

Reading: V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. ArXiv.