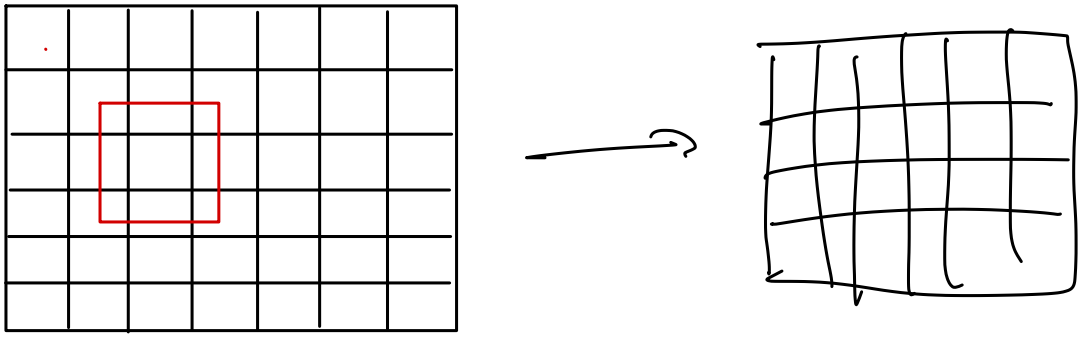
Recall



Typical workflow

- Take in a set of n images
- For each image, apply same convolutional filter (either slightly reduces size, or retains size)
- Apply ReLU (or other activation) at each pixel (same size)
- Max pool (reduce size)

Do this k times (k new channels)

These steps are iterated to form a "deep" CNN.

Eventually, with all the pooling, the images become small (e.g. 5x5 or 4x4), at which point we "<u>flatten</u>" them:

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \\ \hline \end{array} \longrightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix}$$

and then do a final one (or a few) regular fully connected layers. If the response is quantitative, the final output is numeric; if Y is qualitative, the final output is a vector of the # of classes, and a softmax (or similar) function is applied.

Remark Images are usually color, not grayscale.

This is often encoded as 3 <u>channels</u> (AKA #s)

(R, G, B) = (red, green, blue) for each pixel, with values in $\{0, 1, 2, \ldots, 255\}$.

(0,0,0) = black, (255, 255, 255) = white

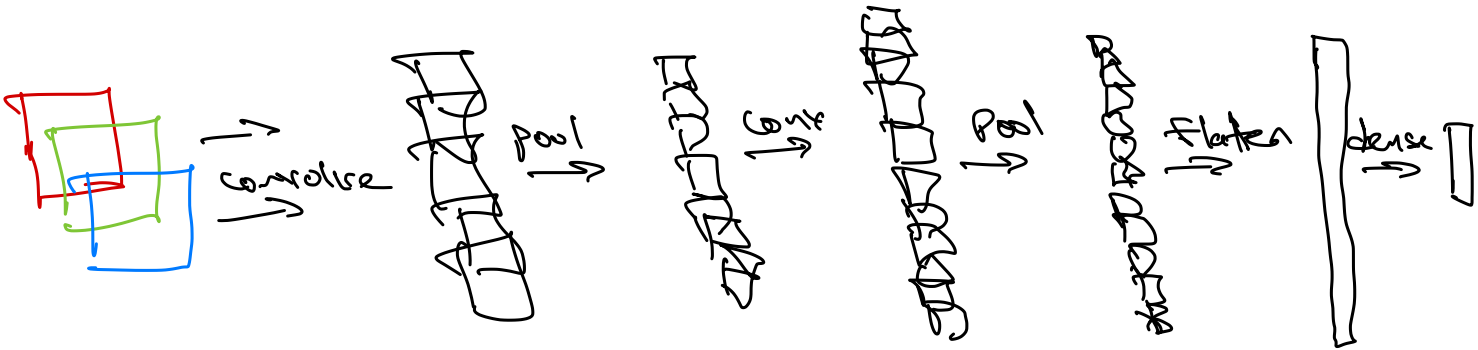
(0, 0, 255) = blue (5, 71, 42) = evergreen

(0, 128, 128) = teal etc...

So each original image has 3 "<u>channels</u>", i.e. there are really 3n original images at the input stage.

A CNN applies a different filter to each channel, and adds them together at the 1st stage (before ReLU or pooling).

A typical representation



$\boxed{\text{Ex}}$ Classification (10 classes) of 32x32 color images.

Define + count parameters of a "deep" CNN:

- Input arrays of dimension $(\underbrace{32, 32}_{\text{spatial image}}, \underbrace{3}_{\text{\# of channels} = (R, G, B)})$

- Do a 3x3 convolutional layer 32 times

  output of stage: $(\underbrace{30, 30}_{\text{image size}}, 32)$ ↳ # of channels

  Count parameters: (3 input channels) × (9 params/conv) × (32 convs) + (32 biases) = 896 parameters

- ReLU previous step, no change
- Max-pool, output dimension (15, 15, 32)
- Do another 3x3 conv layer, 64 times,

  output dim: $(\underbrace{13, 13}_{\text{image}}, 64)$ ↳ channels

Count param: (32 input channels) × (9 params/conv)
× (64 convs) + (64 biases) = 18496

- ReLU, no change
- Max pool, output dim: (6, 6, 64)
  image channels
- Final convolutional layer of 3x3, 64 times
  output dim: (4, 4, 64)
  param count: (64 input channels) × (9 params/conv)
  × (64 output channel) + (64 biases) = 36,928
- ReLU, no change
- Flatten into vector of length 4 × 4 × 64 = 1024

- Dense layer with 64 nodes,

  output : vector of length 64

  param count: (1024 inputs) x (64 params) + (64 biases)

  = 65600

- ReLU, no change

- Final stage dense layer w/ 10 outputs

  dim : 10

  param: (64 hidden nodes) x (10 outputs) + (10 biases) = 650

Total # of params: 122,570

- Softmax prior stage, classifier is class with highest prob.

$\boxed{\text{Note}}$ For "natural images" <u>data augmentation</u> is often used

- rotation
- stretching
- zooming
- cropping
- flipping