

CS-568 Deep Learning

Nazar Khan

PUCIT

Variations of Convolutional Neural Networks

CNN Variations

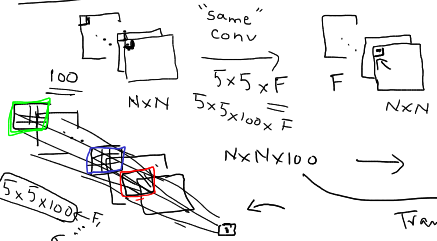
- ▶ There are *lots* of variations of the basic CNN idea.
 - ▶ Fully convolutional networks. No pooling and no fully connected layer.
 - ▶ 1×1 convolutions to reduce computations.
 - ▶ Inception modules to combine multiple filter sizes.
 - ▶ Residual blocks to avoid vanishing gradients.
 - ▶ Depthwise separable convolutions to reduce parameters and computations.
 - ▶ Lightweight and fast models (SqueezeNet, MobileNet, ...) for edge computing.
 - ▶ Fast search over hyperparameters (EfficientNet).
- ▶ A whole course can be dedicated to CNNs alone.

1x1 convolution

Traditional Conv. 5x5

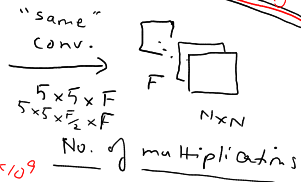
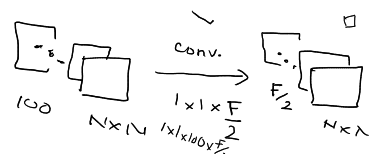
$N = 256$
 $F = 64$

No. of multiplications
 $(N \times N \times F) \times (5 \times 5 \times 100)$



10.9×10^9

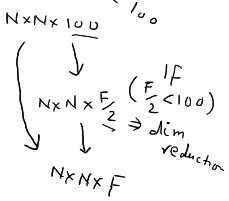
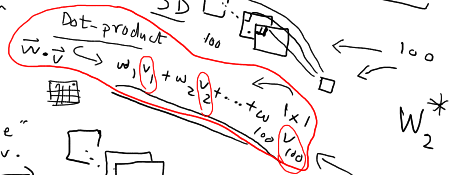
Using 1x1 convolution first



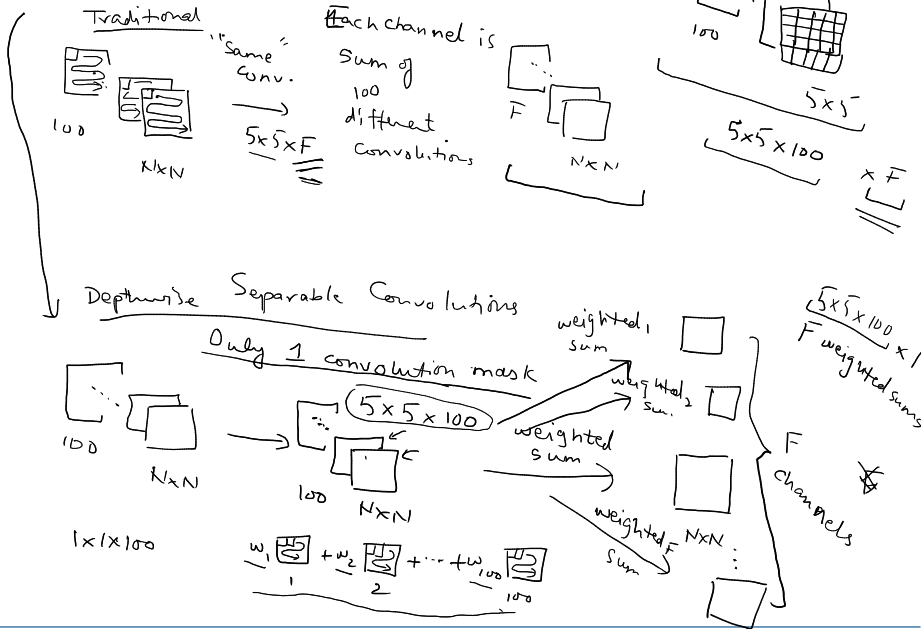
No. of multiplications
 $(N \times N \times \frac{F}{2}) \times (1 \times 1 \times 100)$

No. of multiplications
 $(N \times N \times F) \times (5 \times 5 \times \frac{F}{2})$

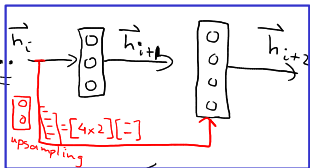
3.8×10^9



Depthwise Separable Convolution



Residual Block



Never let the info. die out

$$W_{i+1} \vec{h}_i + \vec{b}_{i+1} \rightarrow f(\cdot) \vec{h}_{i+1} \rightarrow W_{i+2} \vec{h}_{i+1} + \vec{b}_{i+2} \rightarrow f(\cdot) \vec{h}_{i+2} + \vec{h}_i$$

$$\vec{h}_{i+2} = \text{ReLU} \left(\underbrace{W_{i+2} \vec{h}_{i+1} + \vec{b}_{i+2}}_{\rightarrow 0} + \vec{h}_i \right)$$

If this term $\rightarrow 0$ bcz. of regularization (weight penalty)

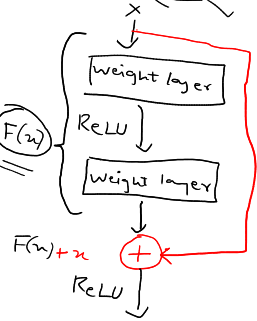
$$= \text{ReLU}(\vec{h}_i)$$

If \vec{h}_i is output of some ReLU then $\vec{h}_i \geq 0$

ResNet Variations

DenseNet

Identity



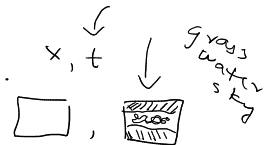
$$F(x) + x \rightarrow \text{ReLU} \rightarrow \text{ReLU}(F(x) + x)$$

So, info. from \vec{h}_i will always be at least carried over to \vec{h}_{i+2} . Info. will not die out.

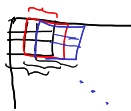


Transposed Convolution

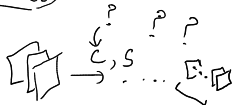
Traditional Conv.



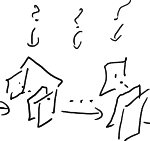
Transposed Convolution



~~Deconvolution~~
DeConv Net

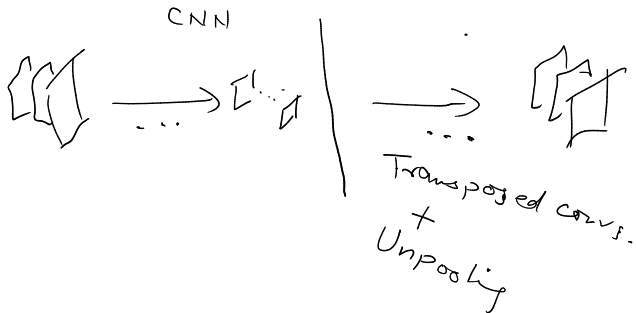
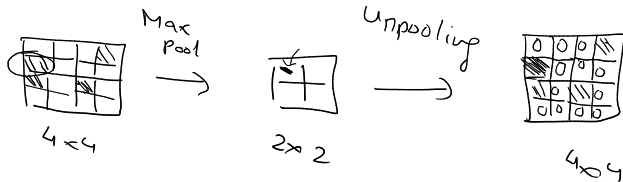


Reover



Convolution \rightarrow "reverse"
+ sub-sample

Unpooling



Fully Convolutional Networks

