Lecture 7 - All about Convolutional Networks

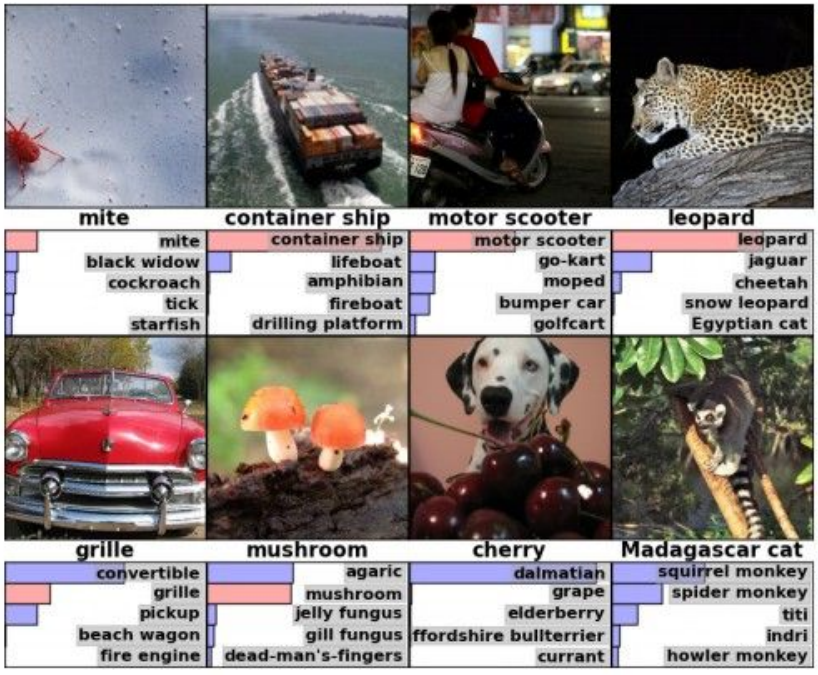
DD2424

April 16, 2018

Fast-forward to today: ConvNets are everywhere

Classification

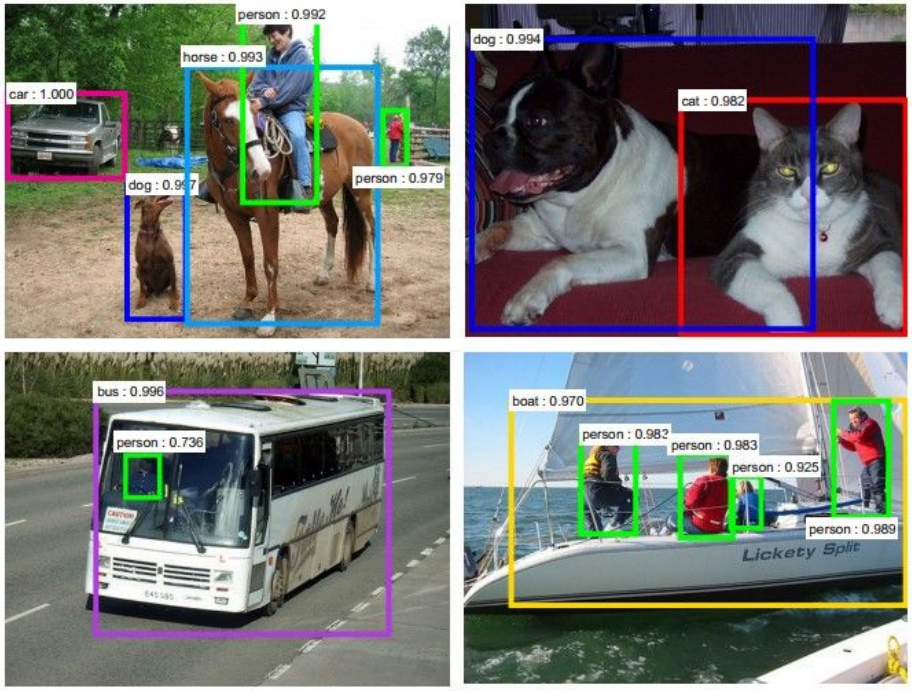
Retrieval



[Krizhevsky 2012]

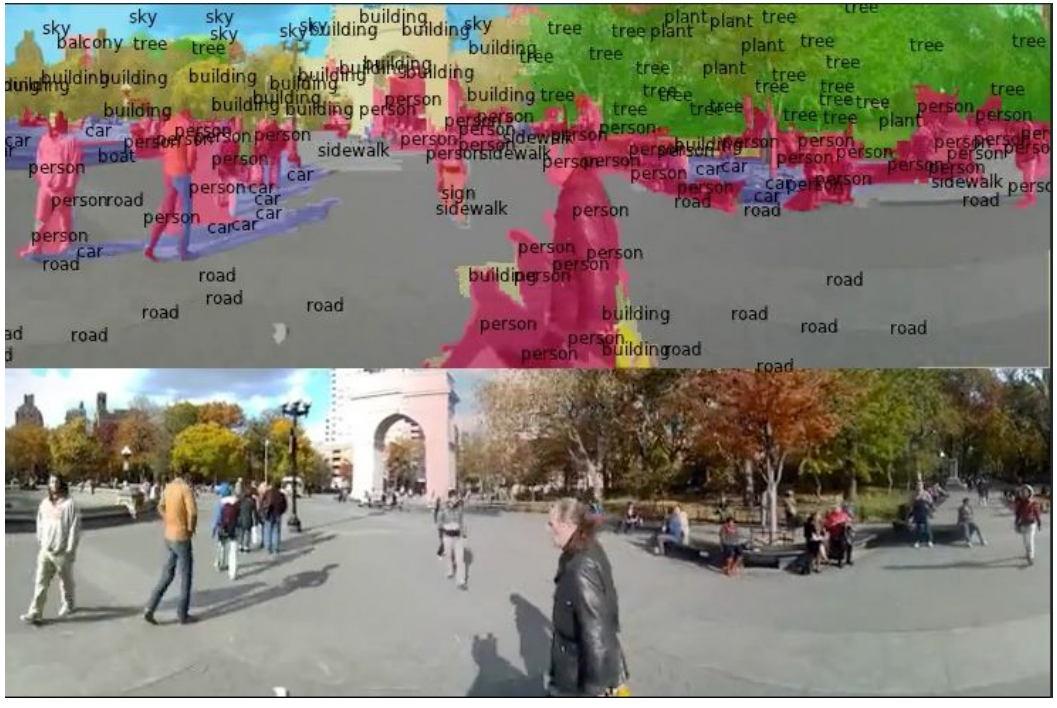
Fast-forward to today: ConvNets are everywhere

Detection



[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



[Farabet et al., 2012]

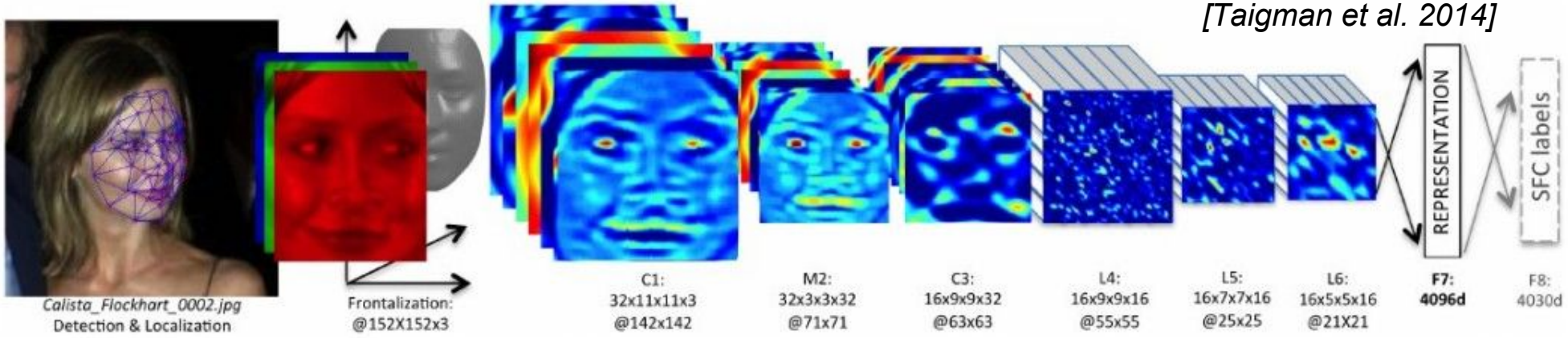
Fast-forward to today: ConvNets are everywhere

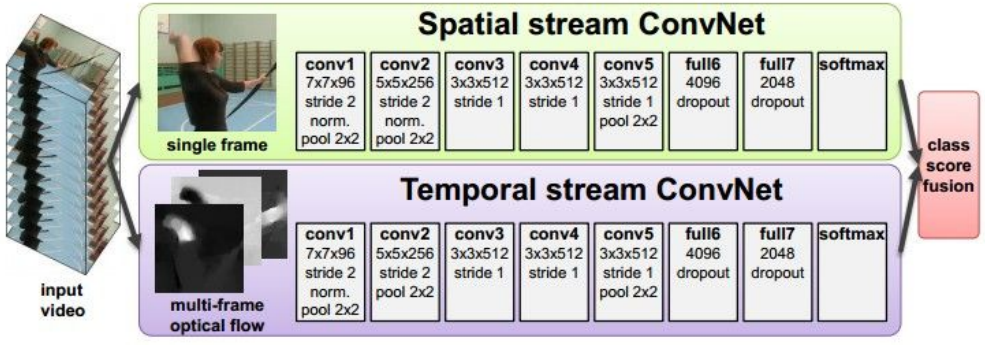
self-driving cars

NVIDIA Tegra X1

Fast-forward to today: ConvNets are everywhere

[Taigman et al. 2014]

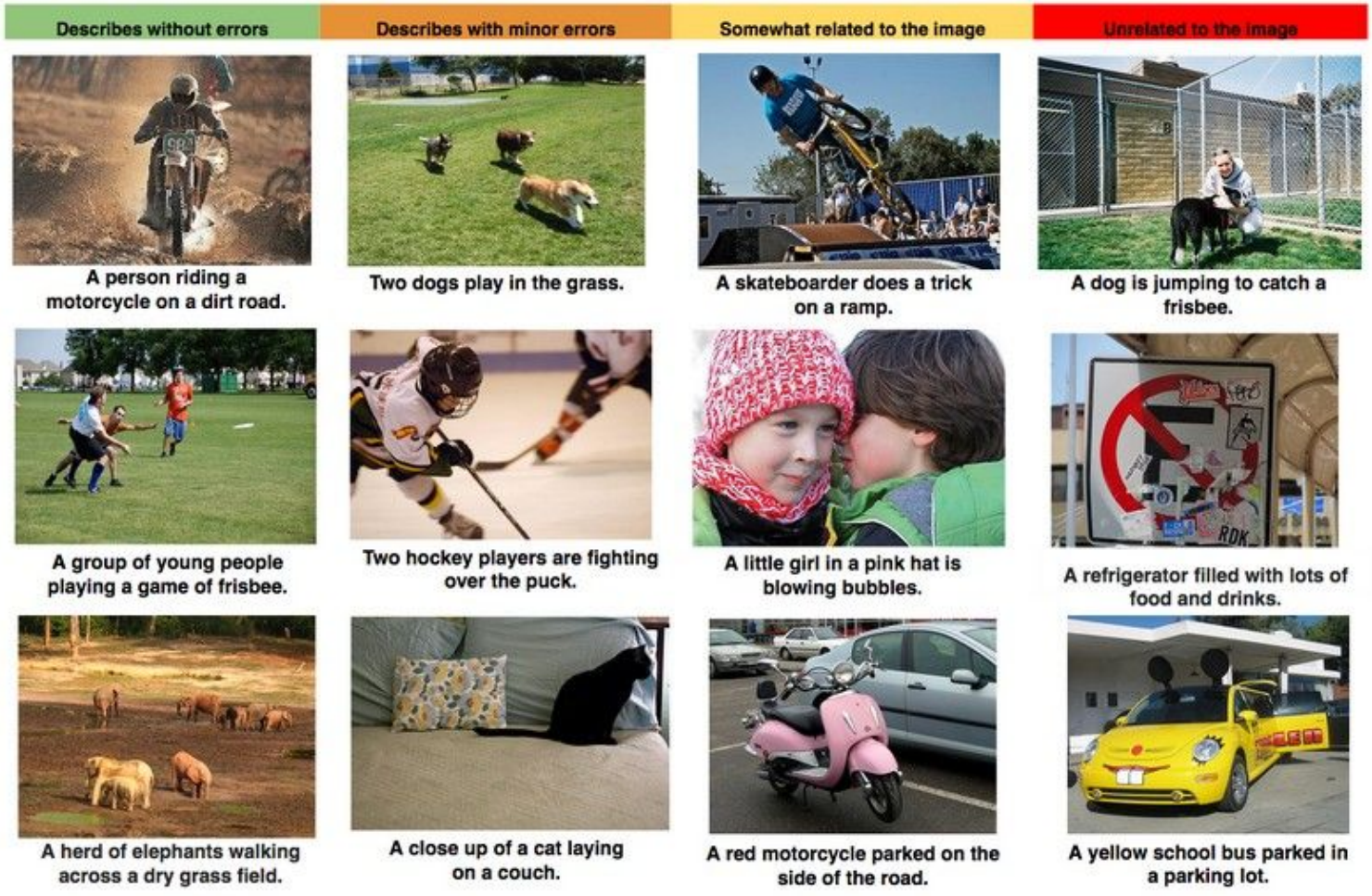




[Simonyan et al. 2014]

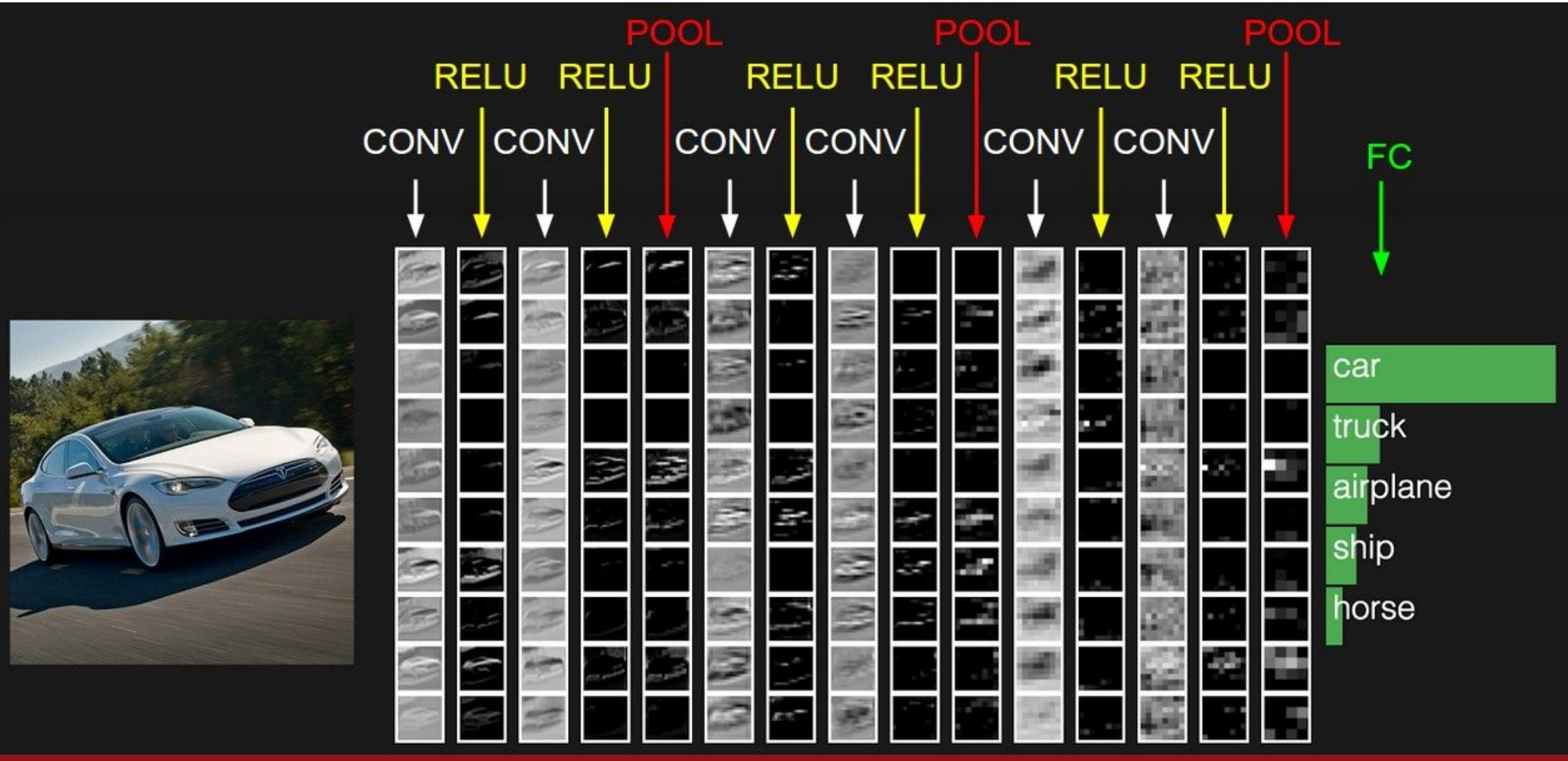
[Goodfellow 2014]

Image Captioning



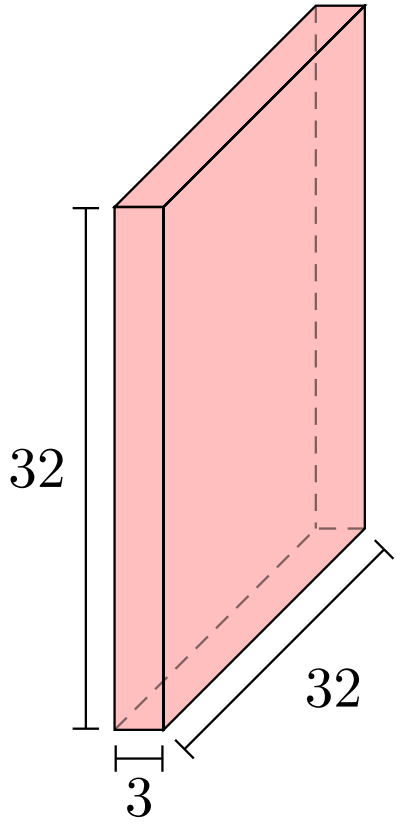
[Vinyals et al., 2015]

preview:

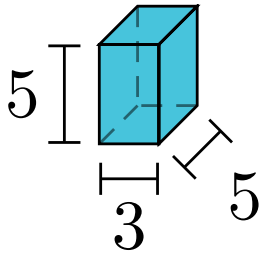

POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

ConvNets for RGB Images: The Convolution Layer

Input Image



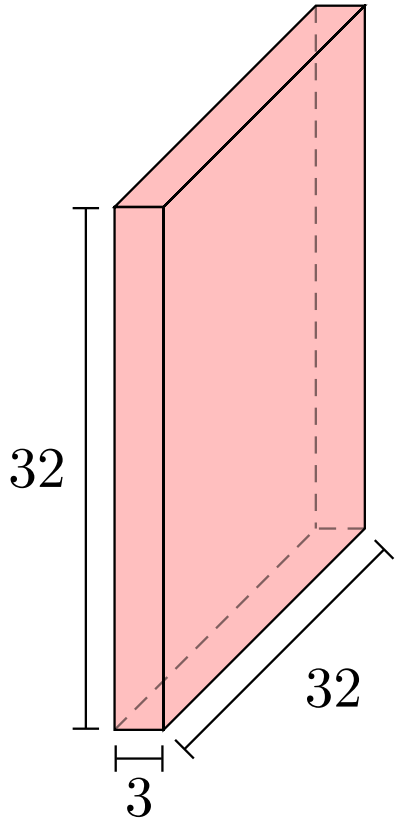
Filter



$\mathbf{X}$ is $32 \times 32 \times 3$

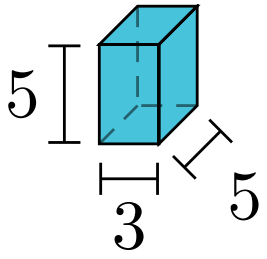
$\mathbf{F}$ is $5 \times 5 \times 3$

Input Image



$\mathbf{X}$ is $32 \times 32 \times 3$

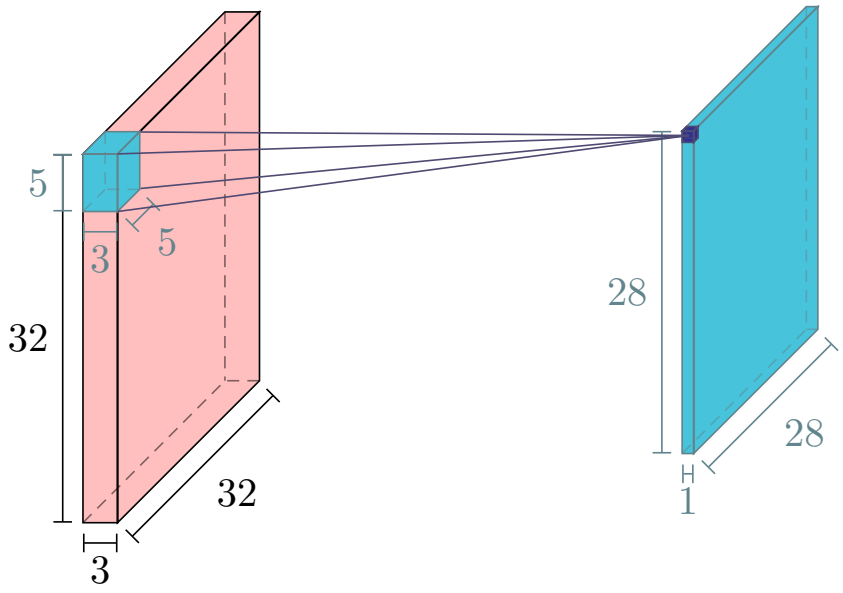
Filter



$\mathbf{F}$ is $5 \times 5 \times 3$

Note: Filter & input image always have the same depth.

Convolution Layer



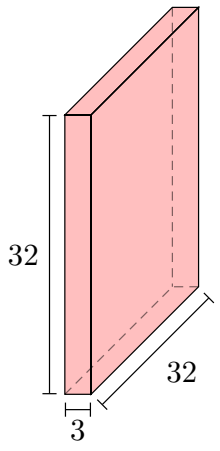
Convolve the image, $\mathbf{X}$, with the filter $\mathbf{F}$.

- Slide filter over all spatial locations in image.
- At each location output 1 number:

  compute **dot product** *between* $\mathbf{F}$ *and a* $5 \times 5 \times 3$ *chunk of* $\mathbf{X}$

Convolution Layer

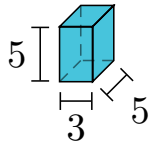
Input Image



$\mathbf{X}$

Size: $32 \times 32 \times 3$

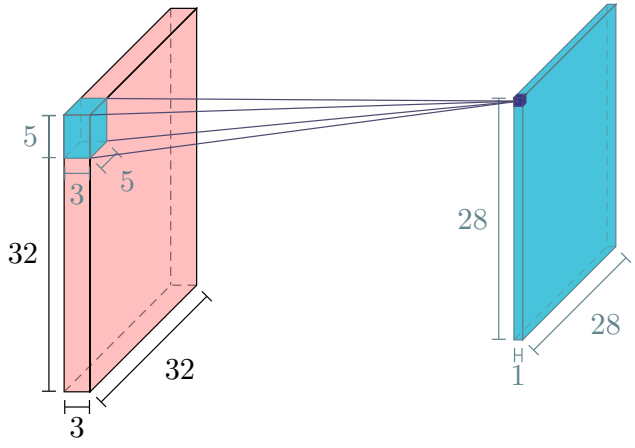
Filter



$\mathbf{F}$

Size: $5 \times 5 \times 3$

Output response map

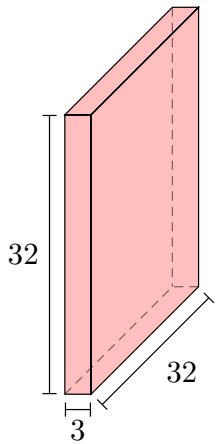


$S = \mathbf{X} * \mathbf{F}$

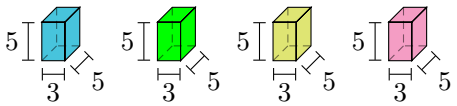
Size: $28 \times 28 \times 1$

Can apply multiple filters.

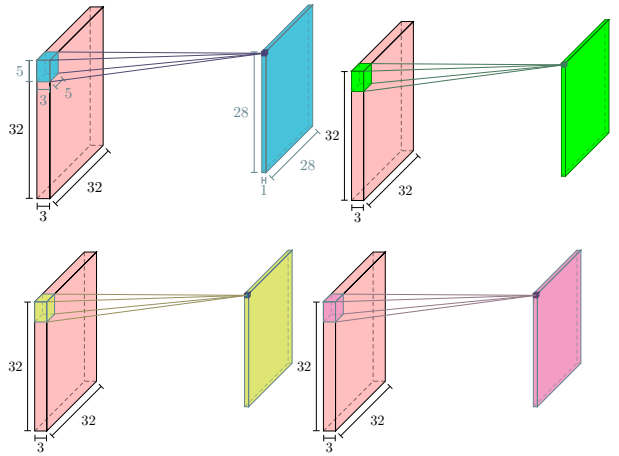
Input Image



Filters



Output response maps



$\mathbf{X}$

Size: $32 \times 32 \times 3$

$\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$

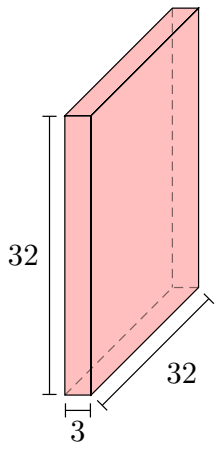
Size $\mathbf{F}_i$: $5 \times 5 \times 3$

$S_i = \mathbf{X} * \mathbf{F}_i$

Size S_i: 28×28

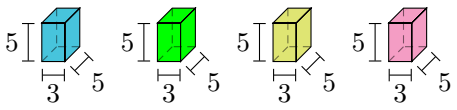
Apply multiple filters and get multiple response maps

Input Image



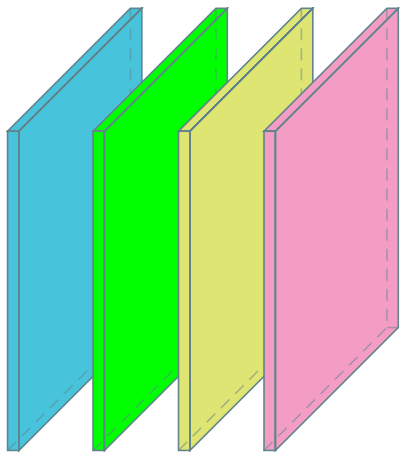
$\mathbf{X}$

Filters



$\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$

Output response maps

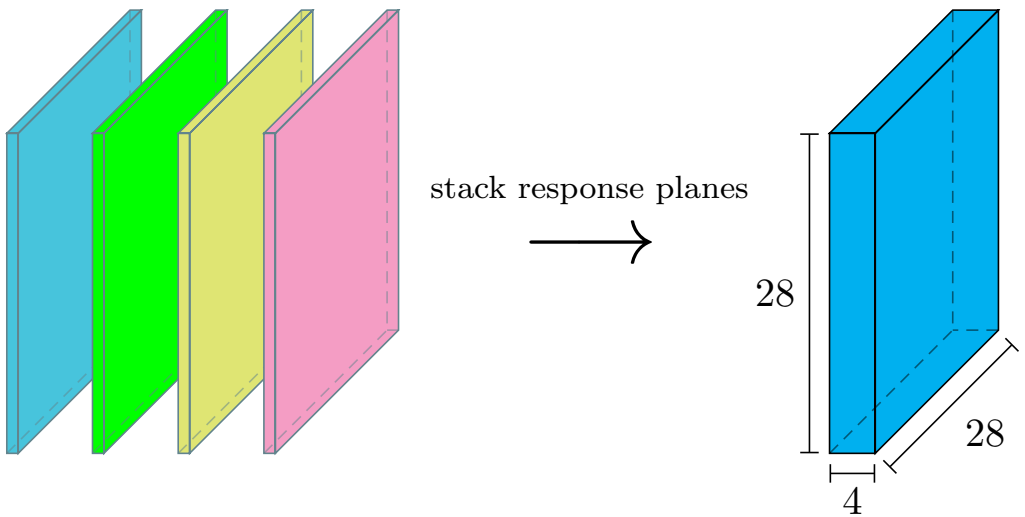


S_1, S_2, S_3, S_4

Each $S_i = \mathbf{X} * \mathbf{F}_i$

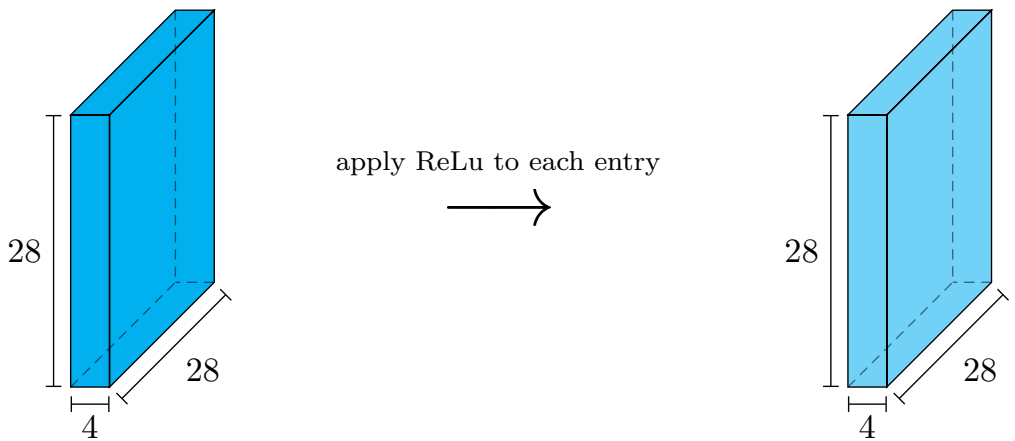
- Stack the multiple response maps to get a *new image* $\mathbf{S}$.

- In our example
  - $\mathbf{S} = \{S_1, S_2, S_3, S_4\}$ and
  - $\mathbf{S}$ has size $28 \times 28 \times 4$



- Apply the non-linear activation function to each element of $\mathbf{S}$.

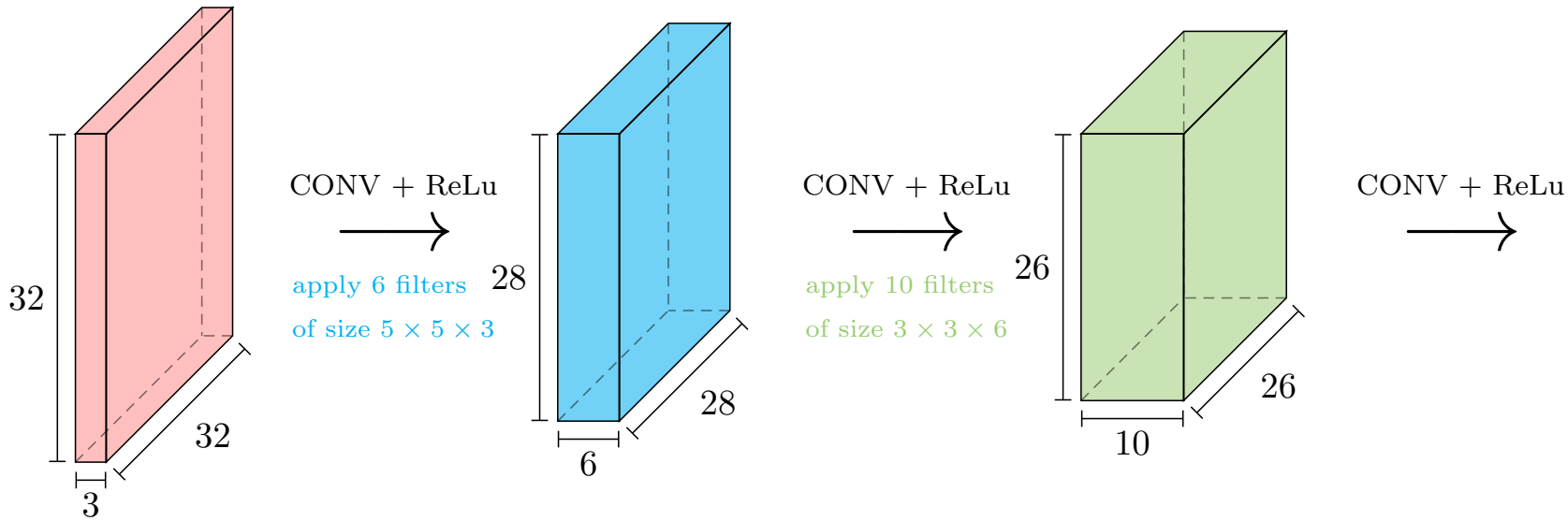
$$\mathbf{H} = \max(0, \mathbf{S})$$



Most basic Convolutional Network layers

Basic **ConvNet** is a composition of

- Convolution Layer
- Activation function



How do we produce final probs for C class labels?

- Add **fully connected layer(s)** after the convolutional layers.
- Example network:
  1 convolutional layer + 1 fully connected layer

$$S_i = \mathbf{X} * \mathbf{F}_i + b_i \quad \text{for } i = 1, \ldots, n_F \quad \leftarrow \text{apply convolution filters}$$

$$\mathbf{S} = \{S_1, \ldots, S_{n_F}\} \quad \leftarrow \text{stack response maps, get new 3D image}$$

$$\mathbf{H} = \max(0, \mathbf{S}) \quad \leftarrow \text{apply ReLu}$$

$$\mathbf{s} = W\text{vec}(\mathbf{H}) + \mathbf{b} \quad \leftarrow \text{fully-connected layer to get } C \text{ scores}$$

$$\mathbf{p} = \text{SoftMax}(\mathbf{s}) \quad \leftarrow \text{turn scores into probabilities}$$

- Dimensions of inputs, outputs and parameters:
  - $\mathbf{X}$ is $w \times h \times 3$
  - Each $\mathbf{F}_i$ is $f \times f \times 3$ and b_i is a scalar
  - Each S_i is $(w - f + 1) \times (h - f + 1)$
  - $\mathbf{S}$ and $\mathbf{H}$ are $(w - f + 1) \times (h - f + 1) \times n_F$
  - W is $C \times (w - f + 1)(h - f + 1)n_F$
  - $\mathbf{b}, \mathbf{s}$ and $\mathbf{p}$ are $C \times 1$

How do we learn the parameters of the network?

- Add **fully connected layer(s)** after the convolutional layers.
- Example network:
  1 convolutional layer + 1 fully connected layer

$$S_i = \mathbf{X} * \mathbf{F}_i + b_i \quad \text{for } i = 1, \ldots, n_F \quad \leftarrow \text{apply convolution filters}$$

$$\mathbf{S} = \{S_1, \ldots, S_{n_F}\} \quad \leftarrow \text{stack response maps, get new 3D image}$$

$$\mathbf{H} = \max(0, \mathbf{S}) \quad \leftarrow \text{apply ReLu}$$

$$\mathbf{s} = W\text{vec}(\mathbf{H}) + \mathbf{b} \quad \leftarrow \text{fully-connected layer to get } C \text{ scores}$$

$$\mathbf{p} = \text{SoftMax}(\mathbf{s}) \quad \leftarrow \text{turn scores into probabilities}$$

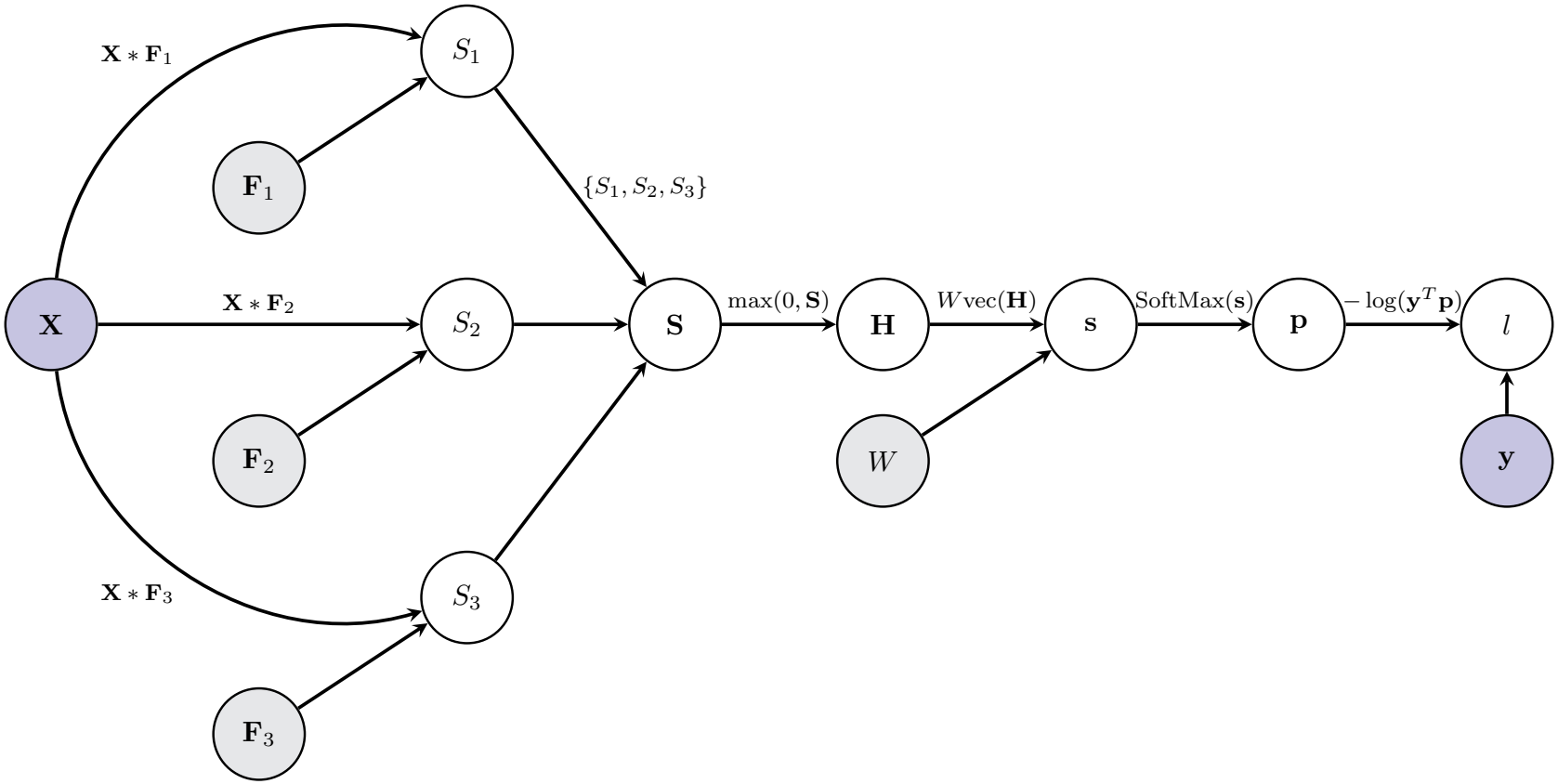
- Dimensions of inputs, outputs and parameters:
  - $\mathbf{X}$ is $w \times h \times 3$
  - Each $\mathbf{F}_i$ is $f \times f \times 3$ and b_i is a scalar
  - Each S_i is $(w - f + 1) \times (h - f + 1)$
  - $\mathbf{S}$ and $\mathbf{H}$ are $(w - f + 1) \times (h - f + 1) \times n_F$
  - W is $C \times (w - f + 1)(h - f + 1)n_F$
  - $\mathbf{b}$, $\mathbf{s}$ and $\mathbf{p}$ are $C \times 1$.

How do we learn the parameters of the network?

- Optimize the usual cross-entropy loss ($+ L_2$ regularization term) on the training data.
- Use mini-batch gradient descent to perform optimization.
- $\Longrightarrow$ need to compute the gradient of the loss w.r.t. the convolutional parameters....

Gradient Computations for one Convolutional layer

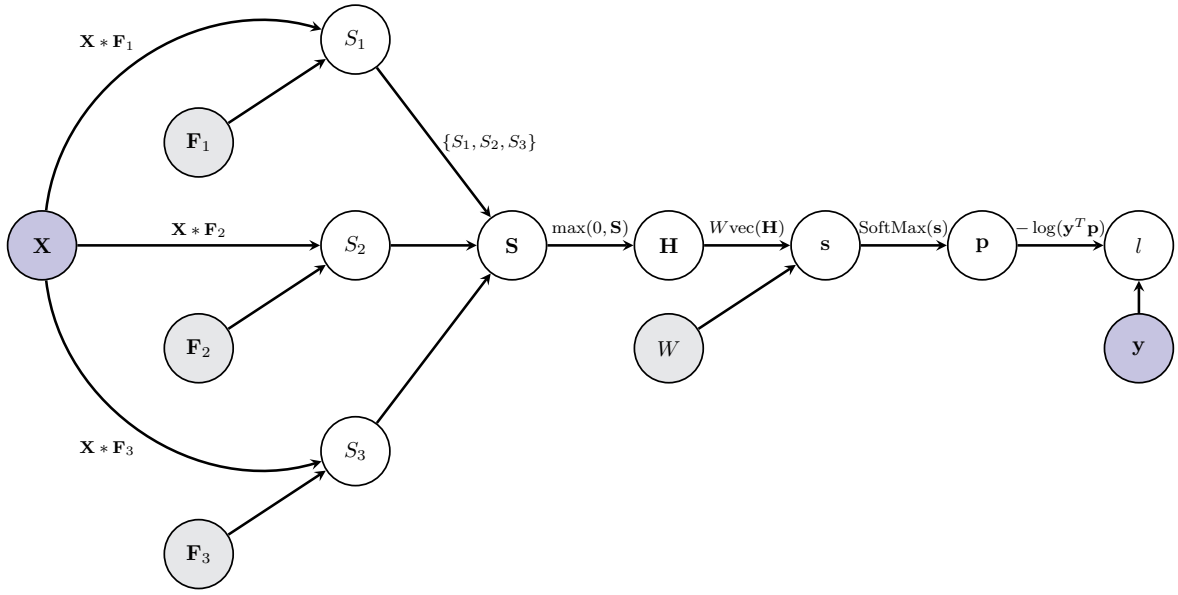
Computational Graph for our simple network



Notes about the above figure

- Apply 3 filters in the convolutional layer ($n_F = 3$).
- $\mathbf{X} = \{X_1, X_2, X_3\}$ and each X_i has size $w \times h$
- Each $\mathbf{F}_i = \{F_{i1}, F_{i2}, F_{i3}\}$ and has size $f \times f \times 3$
- Have omitted the bias weights for clarity.

Computational Graph for our simple network



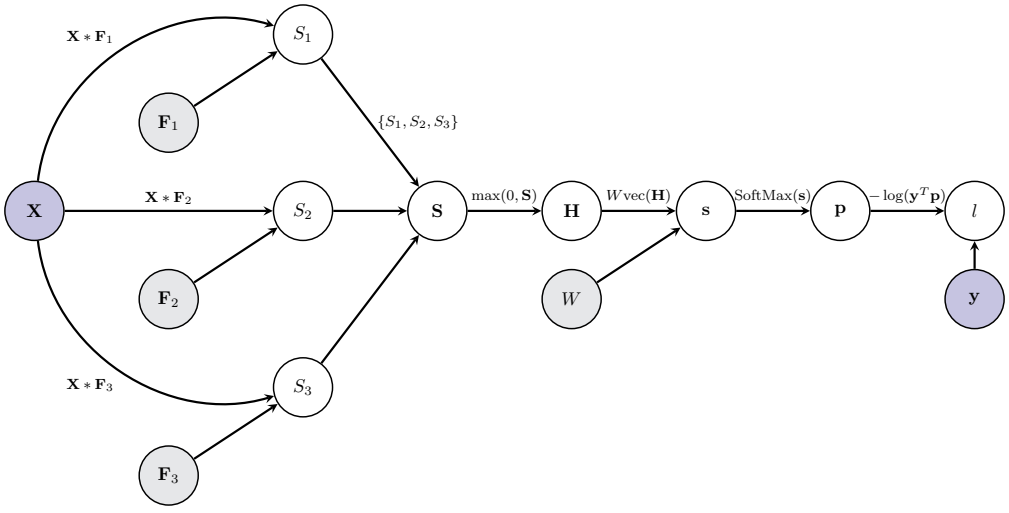
From previous lectures know that

$$\frac{\partial l}{\partial \mathbf{s}} = -(\mathbf{y} - \mathbf{p})^T$$

$$\frac{\partial l}{\partial \text{vec}(\mathbf{H})} = \frac{\partial l}{\partial \mathbf{s}} W$$

$$\frac{\partial l}{\partial \text{vec}(\mathbf{S})} = \frac{\partial l}{\partial \text{vec}(\mathbf{H})} \text{diag}\left(\text{Ind}(\text{vec}(\mathbf{S}) > 0)\right)$$

Computational Graph for our simple network



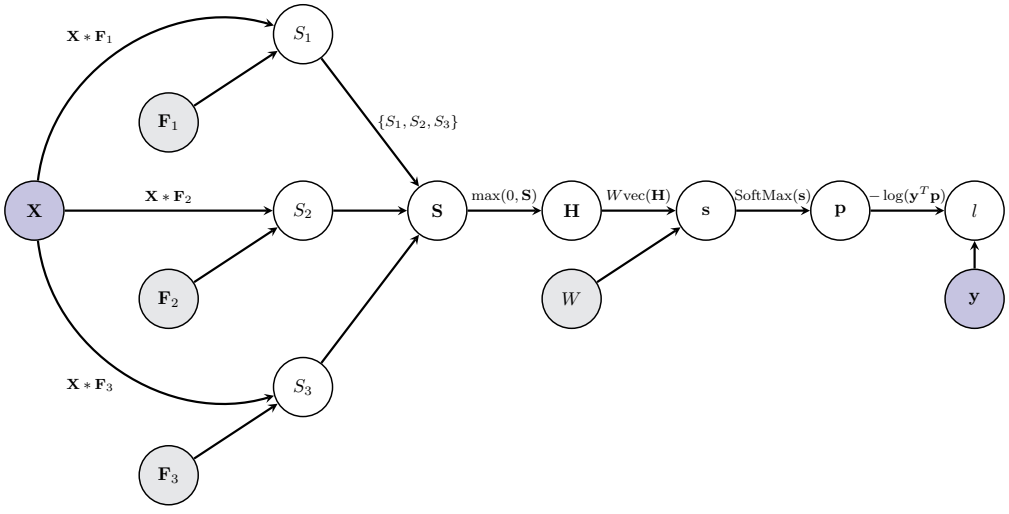
From reading the computational graph we can see that

$$\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)} = \frac{\partial l}{\partial \text{vec}(S_i)} \frac{\partial \text{vec}(S_i)}{\partial \text{vec}(\mathbf{F}_i)}$$

$$= \frac{\partial l}{\partial \text{vec}(\mathbf{S})} \frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_i)} \frac{\partial \text{vec}(S_i)}{\partial \text{vec}(\mathbf{F}_i)}$$

for $i = 1, 2, 3$.

Computational Graph for our simple network



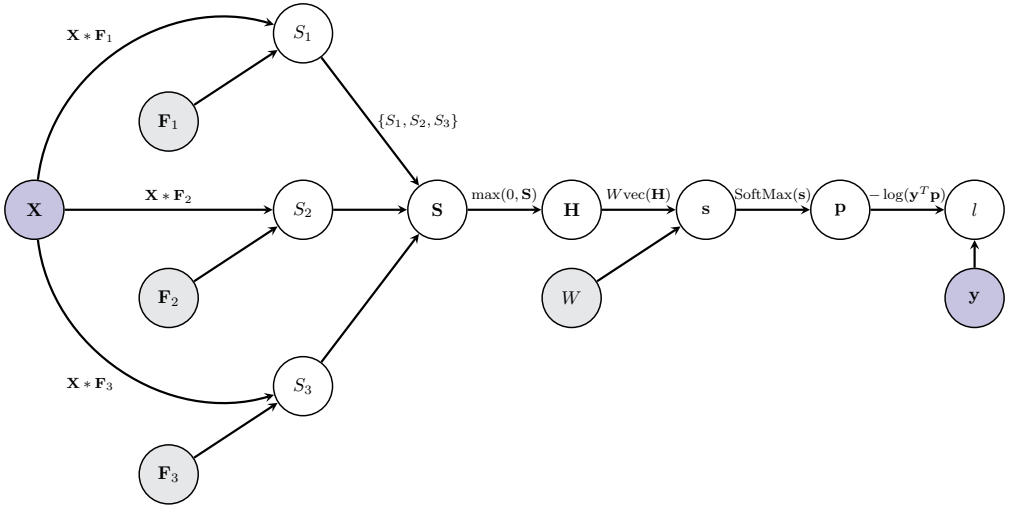
From reading the computational graph we can see that

$$\frac{\partial l}{\partial \text{vec}(F_i)} = \frac{\partial l}{\partial \text{vec}(\mathbf{S})} \frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_i)} \frac{\partial \text{vec}(S_i)}{\partial \text{vec}(\mathbf{F}_i)}$$

$\uparrow$ already know

for $i = 1, 2, 3$.

Computational Graph for our simple network



From reading the computational graph we can see that

$$\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)} = \frac{\partial l}{\partial \text{vec}(\mathbf{S})} \underbrace{\frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_i)}}_{\text{calculate now}} \frac{\partial \text{vec}(S_i)}{\partial \text{vec}(\mathbf{F}_i)}$$

for $i = 1, 2, 3$.

Jacobian of $\text{vec}(S)$ w.r.t. $\text{vec}(S_i)$

- Have $\mathbf{S} = \{S_1, S_2, S_3\} \implies$

$$\text{vec}(\mathbf{S}) = \begin{pmatrix} \text{vec}(S_1) \\ \text{vec}(S_2) \\ \text{vec}(S_3) \end{pmatrix}$$

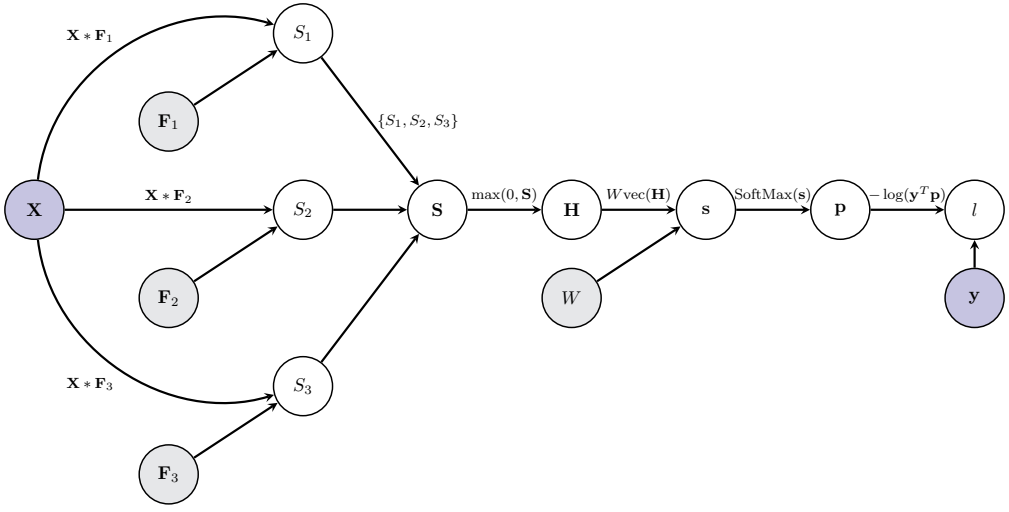
- Then

$$\frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_1)} = \begin{pmatrix} I_t \\ 0_{t\times t} \\ 0_{t\times t} \end{pmatrix}, \quad \frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_2)} = \begin{pmatrix} 0_{t\times t} \\ I_t \\ 0_{t\times t} \end{pmatrix}, \quad \frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_3)} = \begin{pmatrix} 0_{t\times t} \\ 0_{t\times t} \\ I_t \end{pmatrix}$$

  where $t = (w - f + 1) \times (h - f + 1)$ and each $0_{t\times t}$ denotes a matrix of zeros of size $t \times t$.

- Each $\frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_i)}$ has size $3t \times t$

Computational Graph for our simple network



From reading the computational graph we can see that

$$\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)} = \frac{\partial l}{\partial \text{vec}(\mathbf{S})} \frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_i)} \underbrace{\frac{\partial \text{vec}(S_i)}{\partial \text{vec}(\mathbf{F}_i)}}_{\text{calculate now}}$$

for $i = 1, 2, 3$.

Jacobian of $\text{vec}(S_i)$ w.r.t. $\text{vec}(\mathbf{F}_i)$

- Have for $i = 1, 2, 3$:

$$S_i = \mathbf{X} * \mathbf{F}_i$$

- Can write a convolution (in a very memory in-efficient way) as a matrix multiplication

$$\text{vec}(S_i) = M_{\mathbf{X}} \, \text{vec}(\mathbf{F}_i)$$

- $M_{\mathbf{X}}$ has size $(w - f + 1)(h - f + 1) \times (3f^2)$
- What are the entries of $M_{\mathbf{X}}$?

Writing a convolution as a matrix multiplication

Simple Example

- Have an input image X of size 6×6.
- Have a filter F of size 3×3.
- Convolve X by F gives a response map of size 4×4

$$S = X * F$$

- Each entry of S can be written as

$$S_{lm} = \sum_{i=1}^{3} \sum_{j=1}^{3} X_{l+i-1,m+j-1} F_{ij}$$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$S_{11} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to $\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to $\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to $\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ S_{15} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to $\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ S_{21} \\ \vdots \\ S_{44} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to $\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$

Writing a convolution as a matrix multiplication

Simple Example

Want to write this convolution as a matrix multiplication:

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

One Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ S_{21} \\ \vdots \\ S_{44} \end{pmatrix} = \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

new row corresponds to $\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$

$$\text{vec}(S) = M_X \, \text{vec}(F)$$

Multiple planes: Convolution $\rightarrow$ Matrix multiplication

- What about when $\mathbf{X}$ and $\mathbf{F}$ have multiple planes?
- Say $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ has size $6 \times 6 \times 4$,
- $\mathbf{F} = \{F_1, F_2, F_3, F_4\}$ has size $3 \times 3 \times 4$.
- Have

$$S = \mathbf{X} * \mathbf{F} = \sum_{i=1}^{4} X_i * F_i \qquad (S \text{ has size } 4 \times 4)$$

- Then

$$\text{vec}(S) = \sum_{i=1}^{4} M_{X_i} \text{vec}(F_i) = M_{\mathbf{X}} \text{vec}(\mathbf{F})$$

where

$$M_{\mathbf{X}} = \begin{pmatrix} M_{X_1} & M_{X_2} & M_{X_3} & M_{X_4} \end{pmatrix}, \qquad \text{vec}(\mathbf{F}) = \begin{pmatrix} \text{vec}(F_1) \\ \text{vec}(F_2) \\ \text{vec}(F_3) \\ \text{vec}(F_4) \end{pmatrix}.$$

$M_{\mathbf{X}}$ has size 16×36 and $\text{vec}(\mathbf{F})$ has size 36×1.

Back to: Jacobian of $\text{vec}(S_i)$ w.r.t. $\text{vec}(\mathbf{F}_i)$

- Have for $i = 1, 2, 3$:

$$S_i = \mathbf{X} * \mathbf{F}_i$$

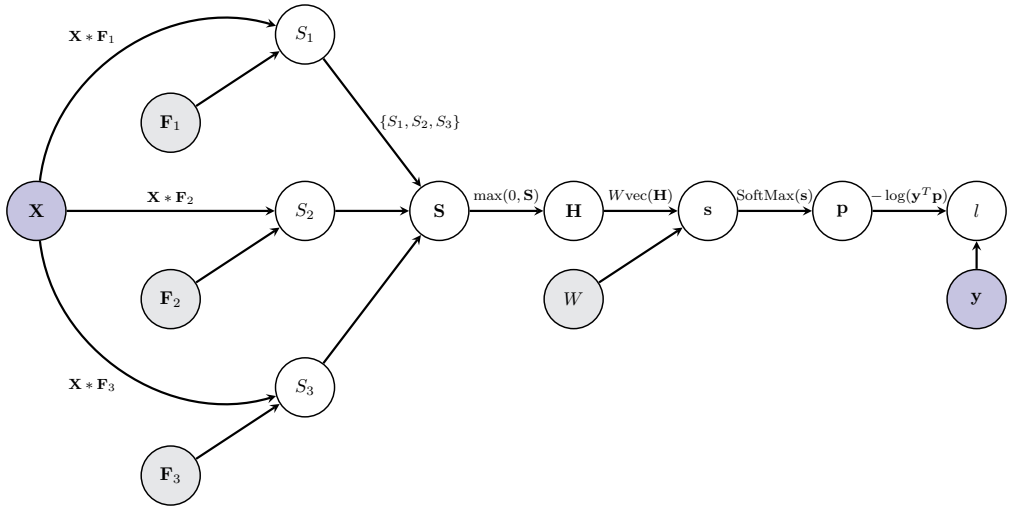
- Can write a convolution (in a very memory inefficient way) as a matrix multiplication

$$\text{vec}(S_i) = M_{\mathbf{X}} \text{vec}(\mathbf{F}_i)$$

- $M_{\mathbf{X}}$ has size $(w - f + 1)(h - f + 1) \times (3f^2)$
- Thus

$$\frac{\partial \text{vec}(S_i)}{\partial \text{vec}(\mathbf{F}_i)} = M_{\mathbf{X}}$$

Gradient of the loss w.r.t. $\text{vec}(\mathbf{F}_i)$



Thus

$$\frac{\partial l}{\partial \text{vec}(\mathbf{F}_1)} = \frac{\partial l}{\partial \text{vec}(\mathbf{S})} \frac{\partial \text{vec}(\mathbf{S})}{\partial \text{vec}(S_1)} \frac{\partial \text{vec}(S_1)}{\partial \text{vec}(\mathbf{F}_1)} = \frac{\partial l}{\partial \text{vec}(\mathbf{S})} \begin{pmatrix} I_t \\ 0_{t\times t} \\ 0_{t\times t} \end{pmatrix} M_{\mathbf{X}}$$

$$= \begin{pmatrix} \frac{\partial l}{\partial \text{vec}(S_1)} & \frac{\partial l}{\partial \text{vec}(S_2)} & \frac{\partial l}{\partial \text{vec}(S_3)} \end{pmatrix} \begin{pmatrix} M_{X_1} & M_{X_2} & M_{X_3} \\ 0_{t\times f^2} & 0_{t\times f^2} & 0_{t\times f^2} \\ 0_{t\times f^2} & 0_{t\times f^2} & 0_{t\times f^2} \end{pmatrix}$$

$$= \left(\frac{\partial l}{\partial \text{vec}(S_1)} M_{X_1}, \frac{\partial l}{\partial \text{vec}(S_1)} M_{X_2}, \frac{\partial l}{\partial \text{vec}(S_1)} M_{X_3} \right)$$

Gradient of the loss w.r.t. $\mathbf{F}_i$

- May want expression for $\frac{\partial l}{\partial \mathbf{F}_i}$ instead of $\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)}$.
- **Option 1**:

  Reshape $\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)}$ (size $1 \times 3f^2$) to $\frac{\partial l}{\partial \mathbf{F}_i}$ (size $f \times f \times 3$).

Gradient of the loss w.r.t. $\mathbf{F}_i$

- May want expression for $\frac{\partial l}{\partial \mathbf{F}_i}$ instead of $\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)}$.
- **Option 2**:

  Return to our simple example ...

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$\begin{pmatrix} v_1 & v_2 & \cdots & v_9 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_{16} \end{pmatrix} \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$\begin{pmatrix} v_1 & v_2 & \cdots & v_9 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_{16} \end{pmatrix} \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

where red column in M_X corresponds to this red block in X

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$(v_1 \quad v_2 \quad \cdots \quad v_9) = (g_1 \quad g_2 \quad \cdots \quad g_{16}) \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

where red column in M_X corresponds to this red block in X

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$\begin{pmatrix} v_1 & v_2 & v_3 & \cdots & v_9 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_{16} \end{pmatrix} \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

where red column in M_X corresponds to this red block in X

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$\begin{pmatrix} v_1 & v_2 & v_3 & v_4 & \cdots & v_9 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_{16} \end{pmatrix} \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

where red column in M_X corresponds to this red block in X

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

. . .

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$\begin{pmatrix} v_1 & v_2 & v_3 & v_4 & \cdots & v_9 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_{16} \end{pmatrix} \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

where red column in M_X corresponds to this red block in X

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Gradient of loss w.r.t. F as opposed to vec(F)

Return to Simple Example

Consider the case

$$\begin{pmatrix} v_1 & v_2 & v_3 & v_4 & \cdots & v_9 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_{16} \end{pmatrix} \underbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} \\ X_{12} & X_{13} & X_{14} & X_{22} & X_{23} & X_{24} & X_{32} & X_{33} & X_{34} \\ X_{13} & X_{14} & X_{15} & X_{23} & X_{24} & X_{25} & X_{33} & X_{34} & X_{35} \\ X_{14} & X_{15} & X_{16} & X_{24} & X_{25} & X_{26} & X_{34} & X_{35} & X_{36} \\ X_{21} & X_{22} & X_{23} & X_{31} & X_{32} & X_{33} & X_{41} & X_{42} & X_{43} \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ X_{44} & X_{45} & X_{46} & X_{54} & X_{55} & X_{56} & X_{64} & X_{65} & X_{66} \end{pmatrix}}_{M_X \text{ size } 16 \times 9}$$

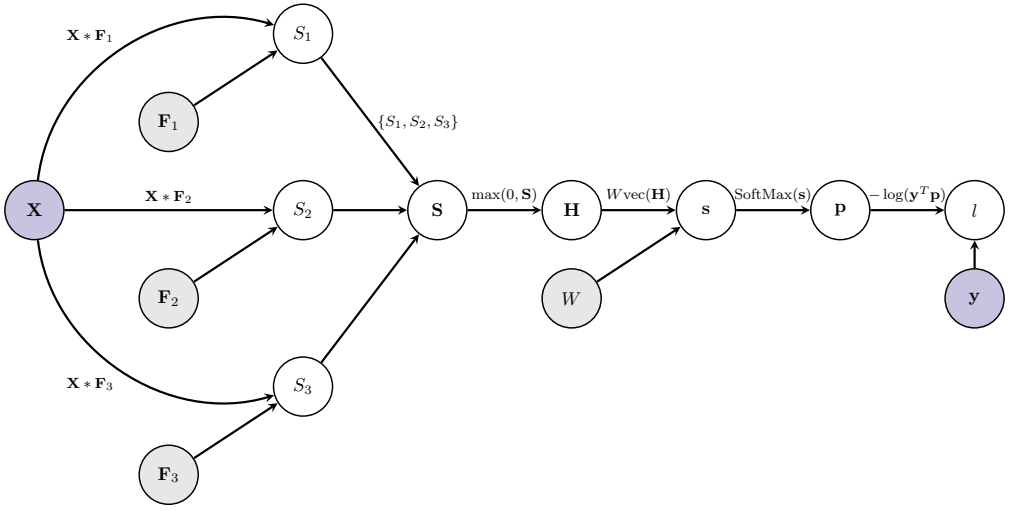
where red column in M_X corresponds to this red block in X

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Thus

$$\begin{pmatrix} v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 \end{pmatrix} = X * \begin{pmatrix} g_1 & g_2 & g_3 & g_4 \\ g_5 & g_6 & g_7 & g_8 \\ g_9 & g_{10} & g_{11} & g_{12} \\ g_{13} & g_{14} & g_{15} & g_{16} \end{pmatrix}$$

Back to Gradient of the loss w.r.t. $\mathbf{F}_i$



Know

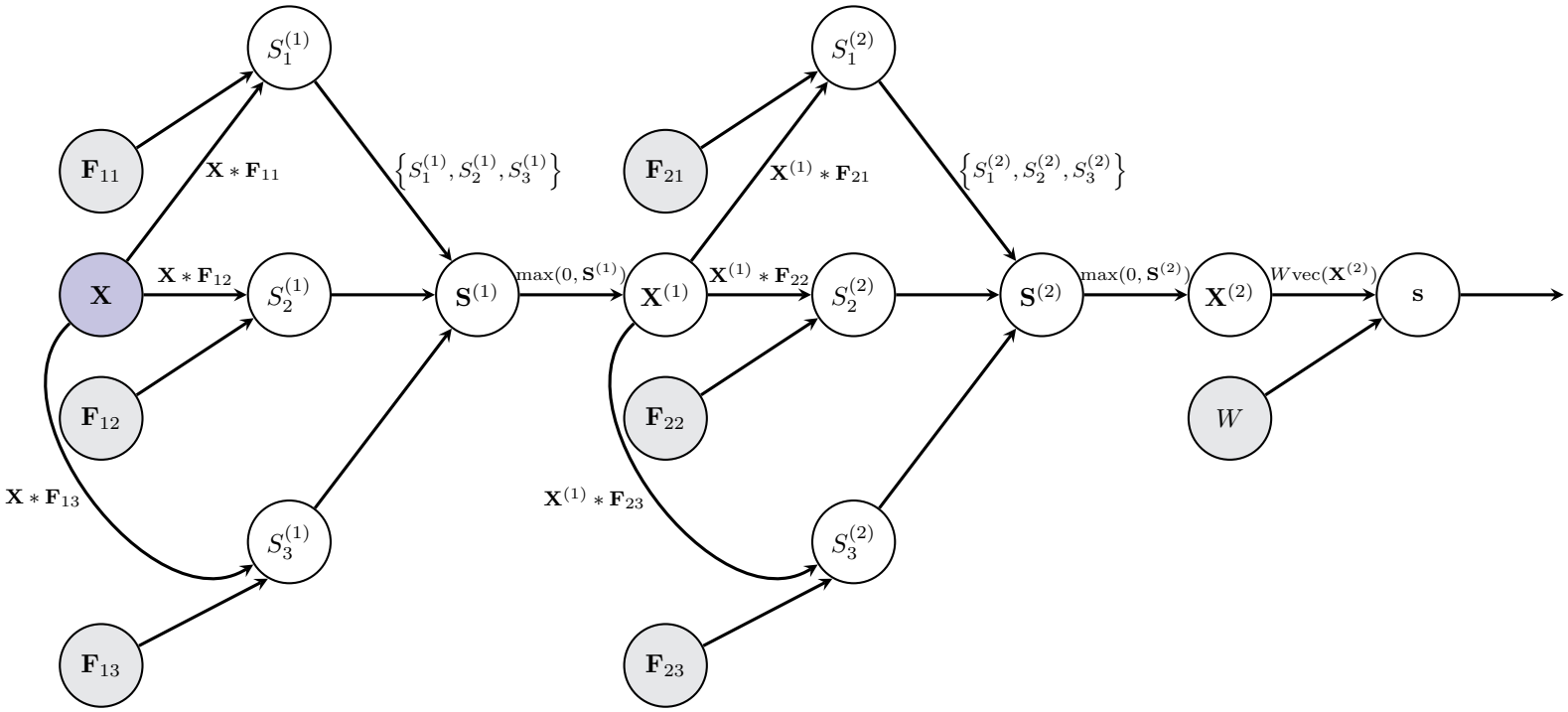
$$\frac{\partial l}{\partial \text{vec}(\mathbf{F}_i)} = \left(\frac{\partial l}{\partial \text{vec}(S_i)} M_{X_1}, \frac{\partial l}{\partial \text{vec}(S_i)} M_{X_2}, \frac{\partial l}{\partial \text{vec}(S_i)} M_{X_3} \right)$$

but our simple example $\Longrightarrow$

$$\frac{\partial l}{\partial \mathbf{F}_i} = \left\{ X_1 * \frac{\partial l}{\partial S_i}, X_2 * \frac{\partial l}{\partial S_i}, X_3 * \frac{\partial l}{\partial S_i} \right\}$$

Gradient Computations for two Convolutional layers

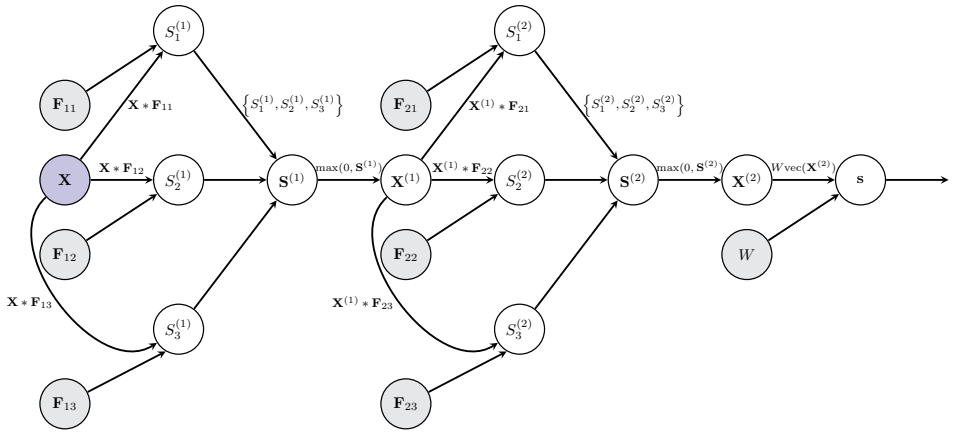
Computational Graph: two convolutional layers



Notes about the figure

- Apply 3 filters at each convolutional layer.
- Have omitted the bias weights for clarity.

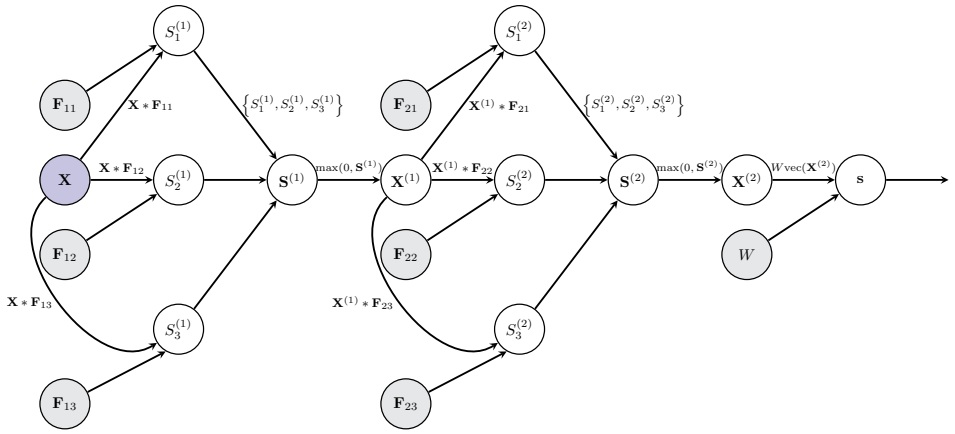
How do we back-propagate the gradient to node $\mathbf{X}^{(1)}$?



- Children of node $\mathbf{X}^{(1)}$ are $S_1^{(2)}, S_2^{(2)}$ and $S_3^{(2)}$
- Thus

$$\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})} = \sum_{i=1}^{3} \frac{\partial l}{\partial \text{vec}(S_i^{(2)})} \frac{\partial \text{vec}(S_i^{(2)})}{\partial \text{vec}(\mathbf{X}^{(1)})}$$

How do we back-propagate the gradient to node $\mathbf{X}^{(1)}$?

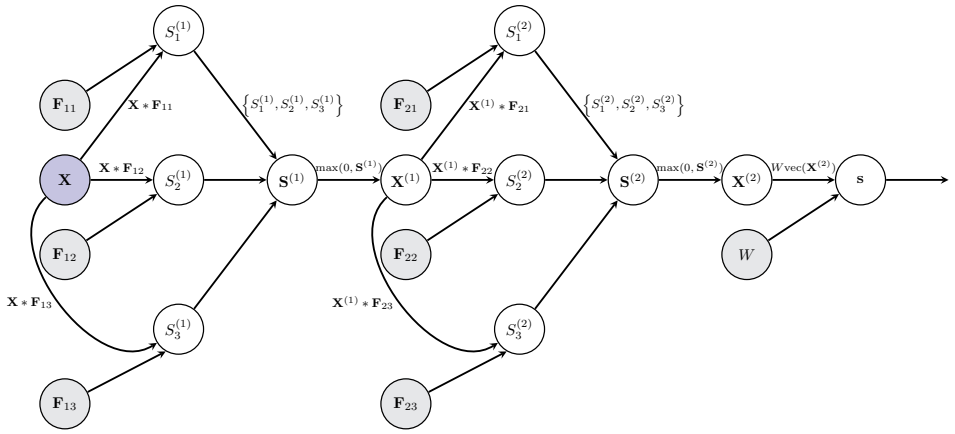


- Children of node $\mathbf{X}^{(1)}$ are $S_1^{(2)}, S_2^{(2)}$ and $S_3^{(2)}$
- Thus

$$\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})} = \sum_{i=1}^{3} \frac{\partial l}{\partial \text{vec}(S_i^{(2)})} \frac{\partial \text{vec}(S_i^{(2)})}{\partial \text{vec}(\mathbf{X}^{(1)})}$$

$\frac{\partial l}{\partial \text{vec}(S_i^{(2)})}$: already know

How do we back-propagate the gradient to node $\mathbf{X}^{(1)}$?



- Children of node $\mathbf{X}^{(1)}$ are $S_1^{(2)}, S_2^{(2)}$ and $S_3^{(2)}$
- Thus

$$\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})} = \sum_{i=1}^{3} \frac{\partial l}{\partial \text{vec}(S_i^{(2)})} \frac{\partial \text{vec}(S_i^{(2)})}{\partial \text{vec}(\mathbf{X}^{(1)})}$$

$\uparrow$ calculate now

Jacobian of $\text{vec}(S_i^{(2)})$ w.r.t. $\text{vec}(\mathbf{X}^{(1)})$

- Have for $i = 1, 2, 3$:

$$S_i^{(2)} = \mathbf{X}^{(1)} * \mathbf{F}_{2i}$$

- Can write a convolution (in a very memory in-efficient way) as a matrix multiplication

$$\text{vec}(S_i^{(2)}) = M_{\mathbf{F}_{2i}}^{\text{filter}} \, \text{vec}(\mathbf{X}^{(1)})$$

- $M_{F_{2i}}$ has size $(w - f + 1)(h - f + 1) \times 3wh$ (assuming $\mathbf{X}^{(1)}$ has size $w \times h \times 3$ and $\mathbf{F}_{2i}$ has size $f \times f \times 3$.)

- What are the entries of $M_{\mathbf{F}_{2i}}^{\text{filter}}$?

Writing convolution as a matrix multiplication II

Simple Example

- Have an input image X of size 6×6.
- Have a filter F of size 3×3.
- Convolve X by F gives a response map of size 4×4

$$S = X * F$$

- Each entry of S can be written as

$$S_{lm} = \sum_{i=1}^{3} \sum_{j=1}^{3} X_{l+i-1,m+j-1} F_{ij}$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$S_{11} = (\overbrace{F_{11} \;\; F_{12} \;\; F_{13} \;\; 0 \;\; 0 \;\; 0}^{\text{entries corresponding to row 1 of } X} \overbrace{F_{21} \;\; F_{22} \;\; F_{23} \;\; 0 \;\; 0 \;\; 0}^{\text{row 2 of } X} \overbrace{F_{31} \;\; F_{32} \;\; F_{33} \;\; 0 \;\; 0 \;\; 0}^{\text{row 3 of } X} \cdots) \, \text{vec}(X)$$

S_{11} is the dot product between F and red entries of X:

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \end{pmatrix} = \begin{pmatrix} \overbrace{F_{11} & F_{12} & F_{13} & 0 & 0 & 0}^{\text{entries corresponding to row 1 of } X} & \overbrace{F_{21} & F_{22} & F_{23} & 0 & 0 & 0}^{\text{row 2 of } X} & \overbrace{F_{31} & F_{32} & F_{33} & 0 & 0 & 0}^{\text{row 3 of } X} & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \end{pmatrix} \text{vec}(X)$$

S_{12} is the dot product between F and red entries of X:

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \end{pmatrix} = \begin{pmatrix} \overbrace{F_{11} & F_{12} & F_{13} & 0 & 0 & 0}^{\text{entries corresponding to row 1 of } X} & \overbrace{F_{21} & F_{22} & F_{23} & 0 & 0 & 0}^{\text{row 2 of } X} & \overbrace{F_{31} & F_{32} & F_{33} & 0 & 0 & 0}^{\text{row 3 of } X} & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \end{pmatrix} \text{vec}(X)$$

S_{13} is the dot product between F and red entries of X:

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \end{pmatrix} = \begin{pmatrix} \overbrace{F_{11} & F_{12} & F_{13} & 0 & 0 & 0}^{\text{entries corresponding to row 1 of } X} & \overbrace{F_{21} & F_{22} & F_{23} & 0 & 0 & 0}^{\text{row 2 of } X} & \overbrace{F_{31} & F_{32} & F_{33} & 0 & 0 & 0}^{\text{row 3 of } X} & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \end{pmatrix} \text{vec}(X)$$

S_{14} is the dot product between F and red entries of X:

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ S_{21} \end{pmatrix} = \begin{pmatrix} \overbrace{F_{11} & F_{12} & F_{13} & 0 & 0 & 0}^{\text{entries corresponding to row 1 of } X} & \overbrace{F_{21} & F_{22} & F_{23} & 0 & 0 & 0}^{\text{row 2 of } X} & \overbrace{F_{31} & F_{32} & F_{33} & 0 & 0 & 0}^{\text{row 3 of } X} & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \end{pmatrix} \text{vec}(X)$$

S_{21} is the dot product between F and red entries of X:

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix}$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ S_{21} \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} \overbrace{F_{11} & F_{12} & F_{13} & 0 & 0 & 0}^{\text{entries corresponding to row 1 of } X} & \overbrace{F_{21} & F_{22} & F_{23} & 0 & 0 & 0}^{\text{row 2 of } X} & \overbrace{F_{31} & F_{32} & F_{33} & 0 & 0 & 0}^{\text{row 3 of } X} & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \text{vec}(X)$$

Writing convolution as a matrix multiplication

Simple Example

Write this convolution as a matrix multiplication involving $\text{vec}(X)$

$$S = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{pmatrix} * \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}$$

Solution:

$$\begin{pmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ S_{21} \\ \vdots \\ \vdots \end{pmatrix} = \overbrace{\begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}}^{M_F^{\text{filter}}} \begin{pmatrix} \\ \\ \\ \\ \text{vec}(X) \\ \\ \\ \end{pmatrix}$$

entries corresponding to row 1 of X | row 2 of X | row 3 of X

Thus

$$\text{vec}(S) = M_F^{\text{filter}} \, \text{vec}(X)$$

Multiple planes: Convolution $\rightarrow$ Matrix multiplication

- What about when $\mathbf{X}$ and $\mathbf{F}$ have multiple planes?
- $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ has size $6 \times 6 \times 4$
- $\mathbf{F} = \{F_1, F_2, F_3, F_4\}$ has size $3 \times 3 \times 4$
- Have

$$S = \mathbf{X} * \mathbf{F} = \sum_{i=1}^{4} X_i * F_i$$

- Then

$$\text{vec}(S) = \sum_{i=1}^{4} M_{F_i}^{\text{filter}} \text{vec}(X_i) = M_{\mathbf{F}}^{\text{filter}} \text{vec}(\mathbf{X})$$

where

$$M_{\mathbf{F}}^{\text{filter}} = \begin{pmatrix} M_{F_1}^{\text{filter}} & M_{F_2}^{\text{filter}} & M_{F_3}^{\text{filter}} & M_{F_4}^{\text{filter}} \end{pmatrix}, \quad \text{vec}(\mathbf{X}) = \begin{pmatrix} \text{vec}(X_1) \\ \text{vec}(X_2) \\ \text{vec}(X_3) \\ \text{vec}(X_4) \end{pmatrix}.$$

$M_{\mathbf{F}}^{\text{filter}}$ has size 16×144 and $\text{vec}(\mathbf{X})$ size 144×1

Back to: Jacobian of $\text{vec}(S_i^{(2)})$ w.r.t. $\text{vec}(\mathbf{X}^{(1)})$

- Have for $i = 1, 2, 3$:

$$S_i^{(2)} = \mathbf{X}^{(1)} * \mathbf{F}_{2i}$$

- Can write a convolution (in a very memory in-efficient way) as a matrix multiplication

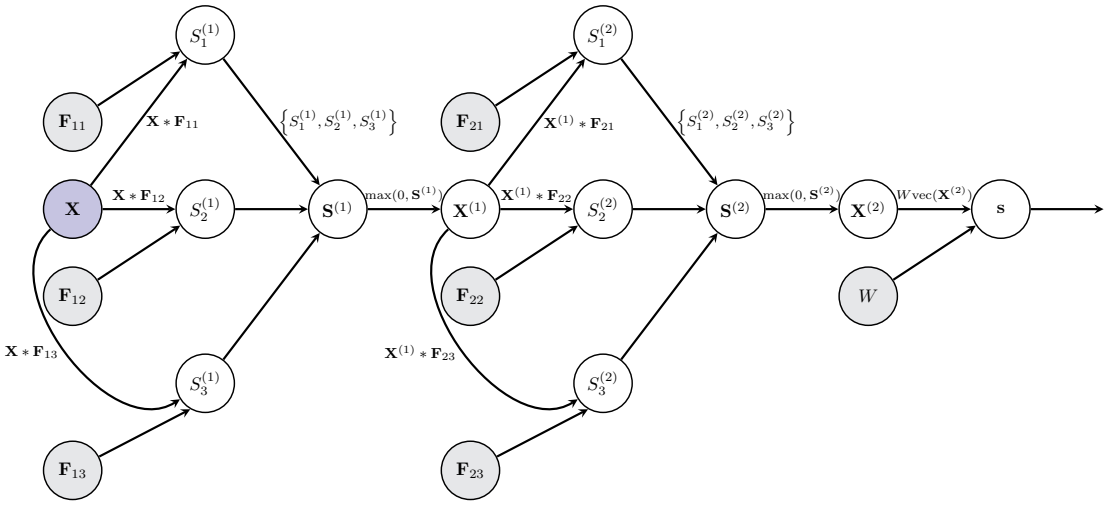
$$\text{vec}(S_i^{(2)}) = M_{\mathbf{F}_{2i}}^{\text{filter}} \text{vec}(\mathbf{X}^{(1)})$$

- $M_{\mathbf{F}_{2i}}^{\text{filter}}$ has size $(w' - f + 1)(h' - f + 1) \times 3w'h'$ (where $w' = w - f + 1$ and $h' = h - f + 1$).

- Thus

$$\frac{\partial \text{vec}(S_i^{(2)})}{\partial \text{vec}(\mathbf{X}^{(1)})} = M_{\mathbf{F}_{2i}}^{\text{filter}}$$

Gradient of the loss w.r.t. $\text{vec}(\mathbf{X}^{(1)})$



- Thus

$$\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})} = \sum_{i=1}^{3} \frac{\partial l}{\partial \text{vec}(S_i^{(2)})} \frac{\partial \text{vec}(S_i^{(2)})}{\partial \text{vec}(\mathbf{X}^{(1)})}$$

$$= \sum_{i=1}^{3} \frac{\partial l}{\partial \text{vec}(S_i^{(2)})} M_{\mathbf{F}_{2i}}^{\text{filter}}$$

Gradient of the loss w.r.t. $\mathbf{X}^{(1)}$

- May want expression for $\frac{\partial l}{\partial \mathbf{X}^{(1)}}$ instead of $\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})}$.
- **Option 1**:

  Reshape $\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})}$ (size $1 \times 3w'h'$) to $\frac{\partial l}{\partial \mathbf{X}^{(1)}}$ (size $w' \times h' \times 3$).

Gradient of the loss w.r.t. $\mathbf{X}^{(1)}$

- May want expression for $\frac{\partial l}{\partial \mathbf{X}^{(1)}}$ instead of $\frac{\partial l}{\partial \text{vec}(\mathbf{X}^{(1)})}$.
- **Option 2**:

  Return to our simple example ...

Remember the simple example

- Have an input image X of size 6×6.
- Have a filter F of size 3×3.
- Convolve X by F gives a response map of size 4×4

$$S = X * F$$

- Each entry of S can be written as

$$S_{lm} = \sum_{i=1}^{3} \sum_{j=1}^{3} X_{l+i-1,m+j-1} F_{ij}$$

- Can re-write the convolution as a matrix multiplication

$$\text{vec}(S) = M_F^{\text{filter}} \text{vec}(X)$$

Propagate the gradient through the convolution

- Have input image X size 6×6 and filter F size 3×3
- Convolve X with F. Can write as

$$\text{vec}(S) = M_F^{\text{filter}} \text{vec}(X)$$

- If this operation part of some network, then the gradient of the loss w.r.t. $\text{vec}(X)$ is given by

$$\frac{\partial l}{\partial \text{vec}(X)} = \frac{\partial l}{\partial \text{vec}(S)} \frac{\partial \text{vec}(S)}{\partial \text{vec}(X)} = \frac{\partial l}{\partial \text{vec}(S)} M_F^{\text{filter}}$$

- Streamlining notation let

$$\mathbf{v} = \mathbf{g}\, M_F^{\text{filter}}$$

where

$$\mathbf{v} = \frac{\partial l}{\partial \text{vec}(X)}, \qquad \mathbf{g} = \frac{\partial l}{\partial \text{vec}(S)}$$

Examine back-prop eqn thru the convolution in more detail

Have

$$(v_1 \quad v_2 \quad \dots \quad v_{36}) = (g_1 \quad g_2 \quad \dots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Examine back-prop eqn thru the convolution in more detail

$$(v_1 \quad v_2 \quad \dots \quad v_{36}) = (g_1 \quad g_2 \quad \dots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$v_1 = g_1 F_{11}$$

Examine back-prop eqn thru the convolution in more detail

$$(v_1 \quad v_2 \quad \dots \quad v_{36}) = (g_1 \quad g_2 \quad \dots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$v_1 = g_1 F_{11}$$

$$v_2 = g_1 F_{12} + g_2 F_{11}$$

Examine back-prop eqn thru the convolution in more detail

$$(v_1 \quad v_2 \quad \dots \quad v_{36}) = (g_1 \quad g_2 \quad \dots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\begin{aligned} v_1 &= g_1 F_{11} \\ v_2 &= g_1 F_{12} + g_2 F_{11} \\ v_3 &= g_1 F_{13} + g_2 F_{12} + g_3 F_{11} \end{aligned}$$

Examine back-prop eqn thru the convolution in more detail

$$(v_1 \quad v_2 \quad \dots \quad v_{36}) = (g_1 \quad g_2 \quad \dots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\begin{aligned} v_1 &= g_1 F_{11} \\ v_2 &= g_1 F_{12} + g_2 F_{11} \\ v_3 &= g_1 F_{13} + g_2 F_{12} + g_3 F_{11} \\ v_4 &= g_2 F_{13} + g_3 F_{12} + g_4 F_{11} \end{aligned}$$

Examine back-prop eqn thru the convolution in more detail

$$(v_1 \quad v_2 \quad \cdots \quad v_{36}) = (g_1 \quad g_2 \quad \cdots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\begin{aligned}
v_1 &= g_1 F_{11} \\
v_2 &= g_1 F_{12} + g_2 F_{11} \\
v_3 &= g_1 F_{13} + g_2 F_{12} + g_3 F_{11} \\
v_4 &= g_2 F_{13} + g_3 F_{12} + g_4 F_{11} \\
v_5 &= g_3 F_{13} + g_4 F_{12} \\
v_6 &= g_4 F_{13} \\
v_7 &= g_1 F_{21} + g_5 F_{11} \\
v_8 &= g_1 F_{22} + g_2 F_{21} + g_5 F_{12} + g_6 F_{11} \\
v_9 &= g_1 F_{23} + g_2 F_{22} + g_3 F_{21} + g_5 F_{13} + g_6 F_{12} + g_7 F_{11} \\
v_{10} &= g_2 F_{23} + g_3 F_{22} + g_4 F_{21} + g_6 F_{13} + g_7 F_{12} + g_8 F_{11} \\
v_{11} &= g_3 F_{23} + g_4 F_{22} + g_7 F_{13} + g_8 F_{12} \\
v_{12} &= g_4 F_{23} + g_8 F_{13} \\
v_{13} &= g_1 F_{31} + g_5 F_{21} + g_9 F_{11} \\
v_{14} &= g_1 F_{32} + g_2 F_{31} + g_5 F_{22} + g_6 F_{21} + g_9 F_{12} + g_{10} F_{11} \\
v_{15} &= g_1 F_{33} + g_2 F_{32} + g_3 F_{31} + g_5 F_{23} + g_6 F_{22} + g_7 F_{21} + g_9 F_{13} + g_{10} F_{12} + g_{11} F_{11} \\
&\vdots
\end{aligned}$$

Examine back-prop eqn thru the convolution in more detail

$$(v_1 \quad v_2 \quad \cdots \quad v_{36}) = (g_1 \quad g_2 \quad \cdots \quad g_{16}) \begin{pmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & 0 & \cdots \\ 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & 0 & \cdots \\ 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & 0 & \cdots \\ 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & F_{31} & F_{32} & F_{33} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & F_{11} & F_{12} & F_{13} & 0 & 0 & 0 & F_{21} & F_{22} & F_{23} & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\begin{aligned}
v_1 &= g_1 F_{11} \\
v_2 &= g_1 F_{12} + g_2 F_{11} \\
v_3 &= g_1 F_{13} + g_2 F_{12} + g_3 F_{11} \\
v_4 &= g_2 F_{13} + g_3 F_{12} + g_4 F_{11} \\
v_5 &= g_3 F_{13} + g_4 F_{12} \\
v_6 &= g_4 F_{13} \\
v_7 &= g_1 F_{21} + g_5 F_{11} \\
v_8 &= g_1 F_{22} + g_2 F_{21} + g_5 F_{12} + g_6 F_{11} \\
v_9 &= g_1 F_{23} + g_2 F_{22} + g_3 F_{21} + g_5 F_{13} + g_6 F_{12} + g_7 F_{11} \\
v_{10} &= g_2 F_{23} + g_3 F_{22} + g_4 F_{21} + g_6 F_{13} + g_7 F_{12} + g_8 F_{11} \\
v_{11} &= g_3 F_{23} + g_4 F_{22} + g_7 F_{13} + g_8 F_{12} \\
v_{12} &= g_4 F_{23} + g_8 F_{13} \\
v_{13} &= g_1 F_{31} + g_5 F_{21} + g_9 F_{11} \\
v_{14} &= g_1 F_{32} + g_2 F_{31} + g_5 F_{22} + g_6 F_{21} + g_9 F_{12} + g_{10} F_{11} \\
v_{15} &= g_1 F_{33} + g_2 F_{32} + g_3 F_{31} + g_5 F_{23} + g_6 F_{22} + g_7 F_{21} + g_9 F_{13} + g_{10} F_{12} + g_{11} F_{11} \\
&\vdots
\end{aligned}$$

There is a pattern here!

Write back-prop eqn through convolution as a convolution

- Reshape vectors $\mathbf{g}$ and $\mathbf{v}$ into matrices

$$G = \begin{pmatrix} g_1 & g_2 & g_3 & g_4 \\ g_5 & g_6 & g_7 & g_8 \\ g_9 & g_{10} & g_{11} & g_{12} \\ g_{13} & g_{14} & g_{15} & g_{16} \end{pmatrix}, \qquad V = \begin{pmatrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 & v_{10} & v_{11} & v_{12} \\ v_{13} & v_{14} & v_{15} & v_{16} & v_{17} & v_{18} \\ v_{19} & v_{20} & v_{21} & v_{22} & v_{23} & v_{24} \\ v_{25} & v_{26} & v_{27} & v_{28} & v_{29} & v_{30} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} & v_{36} \end{pmatrix}$$

- Let

$$G_{\text{zero-pad}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 \\ 0 & 0 & g_5 & g_6 & g_7 & g_8 & 0 & 0 \\ 0 & 0 & g_9 & g_{10} & g_{11} & g_{12} & 0 & 0 \\ 0 & 0 & g_{13} & g_{14} & g_{15} & g_{16} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad F^{\text{rot180}} = \begin{pmatrix} F_{33} & F_{32} & F_{31} \\ F_{23} & F_{22} & F_{21} \\ F_{13} & F_{12} & F_{11} \end{pmatrix}$$

- Then

$$V = G_{\text{zero-pad}} * F^{\text{rot180}} \implies \frac{\partial l}{\partial X} = \left(\frac{\partial l}{\partial S}\right)_{\text{zero-pad}} * F^{\text{rot180}}$$

Multiple planes: Write back-prop eqn as a convolution

- $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ has size $6 \times 6 \times 4$
- $\mathbf{F} = \{F_1, F_2, F_3, F_4\}$ has size $3 \times 3 \times 4$
- Have

$$S = \mathbf{X} * \mathbf{F} \qquad \Longrightarrow \text{vec}(S) = M_{\mathbf{F}}^{\text{filter}} \text{vec}(\mathbf{X})$$

  where
  - S is 4×4,
  - $\text{vec}(S)$ is 16×1,
  - $M_{\mathbf{F}}^{\text{filter}} = (M_{F_1}^{\text{filter}}, M_{F_2}^{\text{filter}}, M_{F_3}^{\text{filter}}, M_{F_4}^{\text{filter}})$ is 16×144 and
  - $\text{vec}(\mathbf{X}) = \begin{pmatrix} \text{vec}(X_1) \\ \vdots \\ \text{vec}(X_4) \end{pmatrix}$ is 144×1.

- Now

$$\frac{\partial l}{\partial \text{vec}(\mathbf{X})} = \left(\frac{\partial l}{\partial \text{vec}(X_1)}, \frac{\partial l}{\partial \text{vec}(X_2)}, \frac{\partial l}{\partial \text{vec}(X_3)}, \frac{\partial l}{\partial \text{vec}(X_4)} \right)$$

Multiple planes: Write back-prop eqn as a convolution

- Now

$$\frac{\partial l}{\partial \text{vec}(\mathbf{X})} = \left(\frac{\partial l}{\partial \text{vec}(X_1)}, \frac{\partial l}{\partial \text{vec}(X_2)}, \frac{\partial l}{\partial \text{vec}(X_3)}, \frac{\partial l}{\partial \text{vec}(X_4)} \right)$$

$$= \left(\frac{\partial l}{\partial \text{vec}(S)} M_{F_1}, \frac{\partial l}{\partial \text{vec}(S)} M_{F_2}, \frac{\partial l}{\partial \text{vec}(S)} M_{F_3}, \frac{\partial l}{\partial \text{vec}(S)} M_{F_4} \right)$$

- Remember for a single plane have

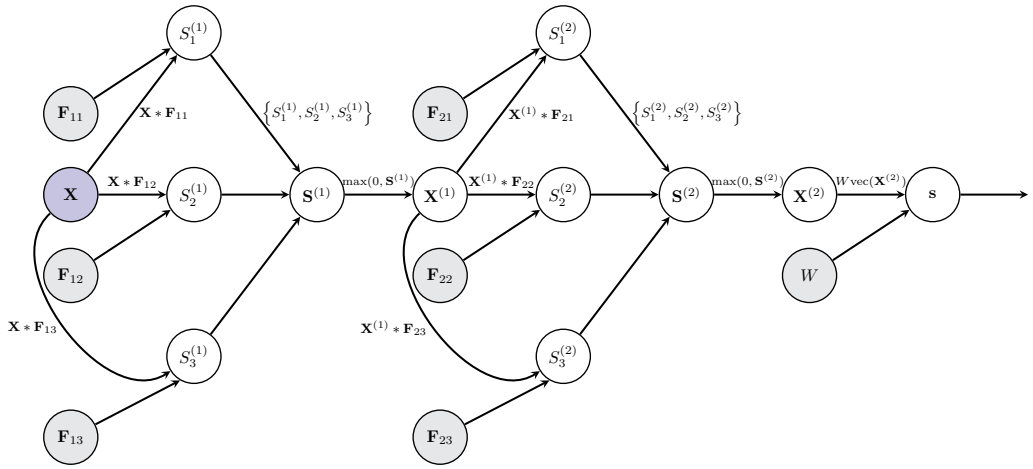
$$\frac{\partial l}{\partial X_i} = G_{\text{zero-pad}} * F_i^{\text{rot180}}$$

where $G_{\text{zero-pad}}$ is the zero-padded version of $\frac{\partial l}{\partial S}$.

- Thus

$$\frac{\partial l}{\partial \mathbf{X}} = \left\{ G_{\text{zero-pad}} * F_1^{\text{rot180}}, G_{\text{zero-pad}} * F_2^{\text{rot180}}, G_{\text{zero-pad}} * F_3^{\text{rot180}}, G_{\text{zero-pad}} * F_4^{\text{rot180}} \right\}$$

Back to gradient of the loss w.r.t. $\mathbf{X}^{(1)}$



Have applied multiple filters with multiple planes and know

$$\frac{\partial l}{\partial \mathrm{vec}(\mathbf{X}^{(1)})} = \sum_{i=1}^{3} \frac{\partial l}{\partial \mathrm{vec}(S_i^{(2)})} M_{\mathbf{F}_{2i}}^{\text{filter}}$$

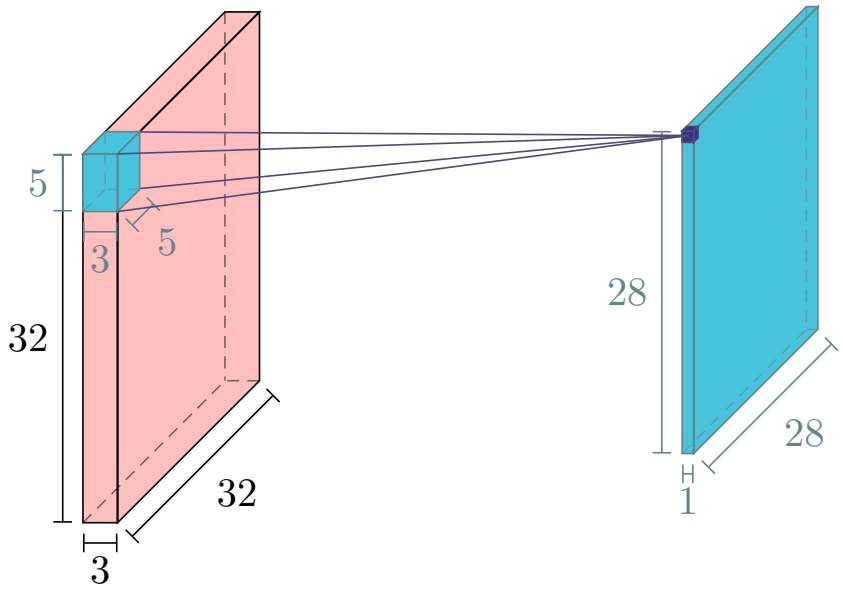
then

$$\frac{\partial l}{\partial X^{(1)}} = \left\{ \sum_{i=1}^{3} G_i^{\text{zero-pad}} * F_{2i,1}^{\text{rot180}}, \sum_{i=1}^{3} G_i^{\text{zero-pad}} * F_{2i,2}^{\text{rot180}}, \sum_{i=1}^{3} G_i^{\text{zero-pad}} * F_{2i,3}^{\text{rot180}} \right\}$$

where $G_i = \frac{\partial l}{\partial S_i^{(2)}}$ and $\mathbf{F}_{2i} = \{F_{2i,1}, F_{2i,2}, F_{2i,3}\}$.

More details on the Convolution layers: **Striding & Zero Padding**

Convolution Layer

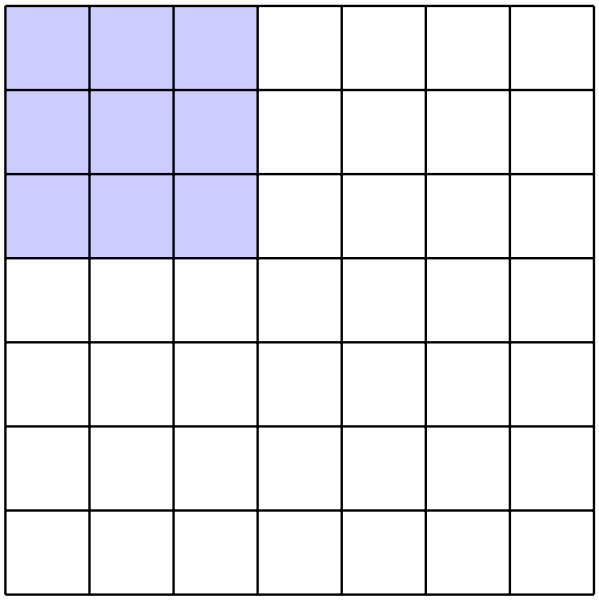


Convolve the image, $\mathbf{X}$, with the filter $\mathbf{F}$.

- Slide filter over all spatial locations in image.
- At each location output 1 number:

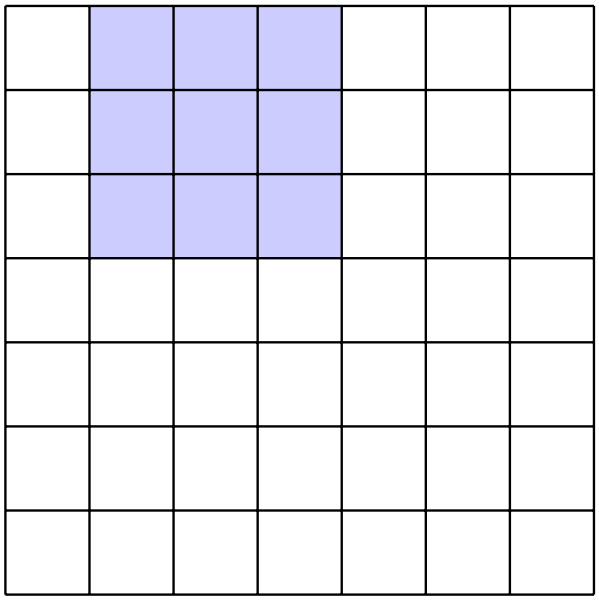
  compute **dot product** *between* $\mathbf{F}$ *and a* $5 \times 5 \times 3$ *chunk of* $\mathbf{X}$

A closer look at the spatial dimensions



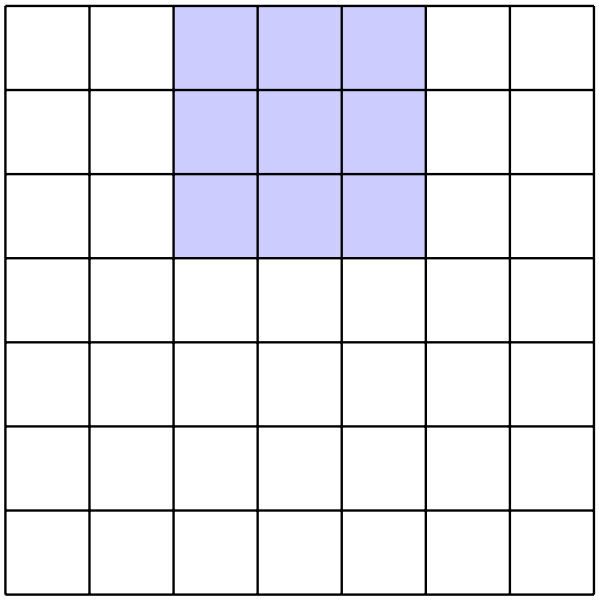
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.

A closer look at the spatial dimensions



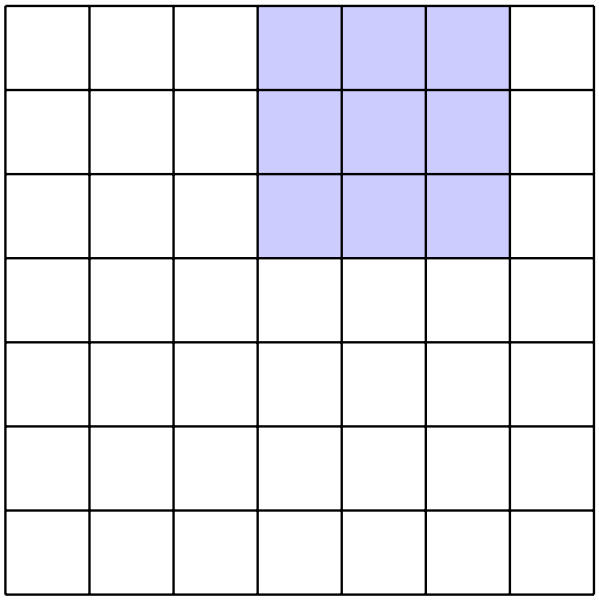
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.

A closer look at the spatial dimensions



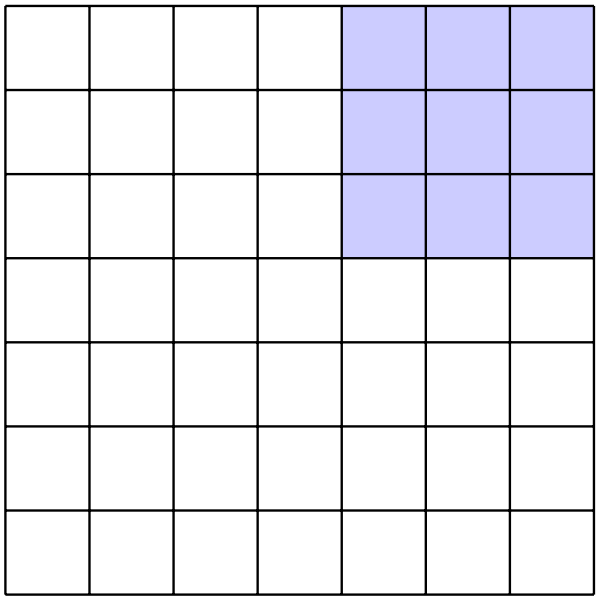
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.

A closer look at the spatial dimensions



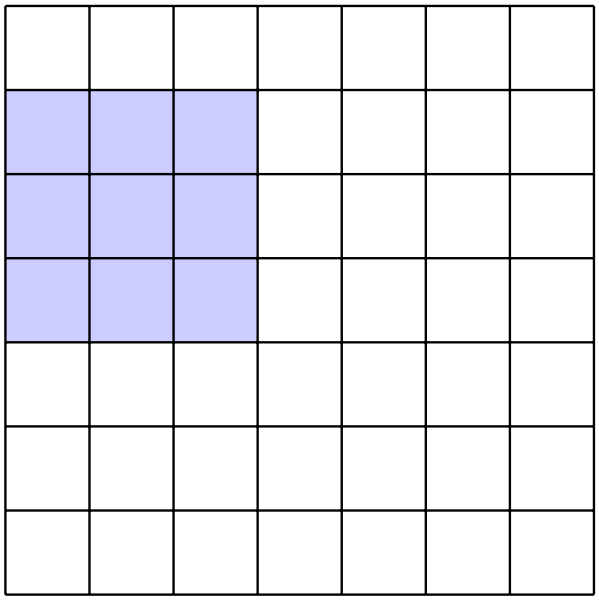
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.

A closer look at the spatial dimensions



- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.

A closer look at the spatial dimensions

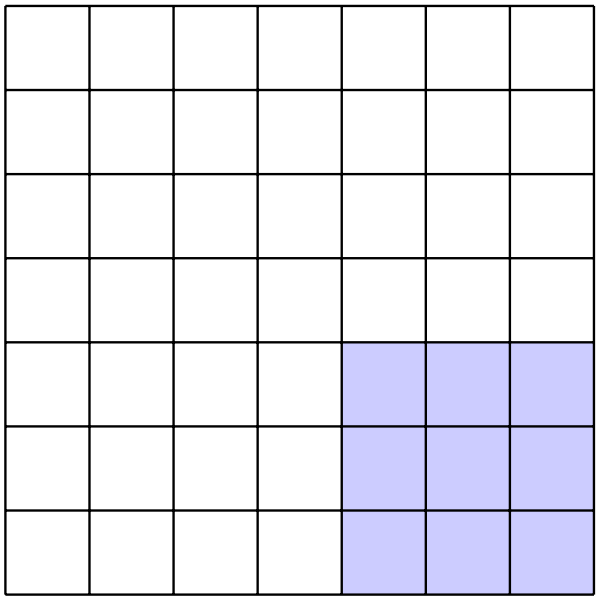


- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.

A closer look at the spatial dimensions

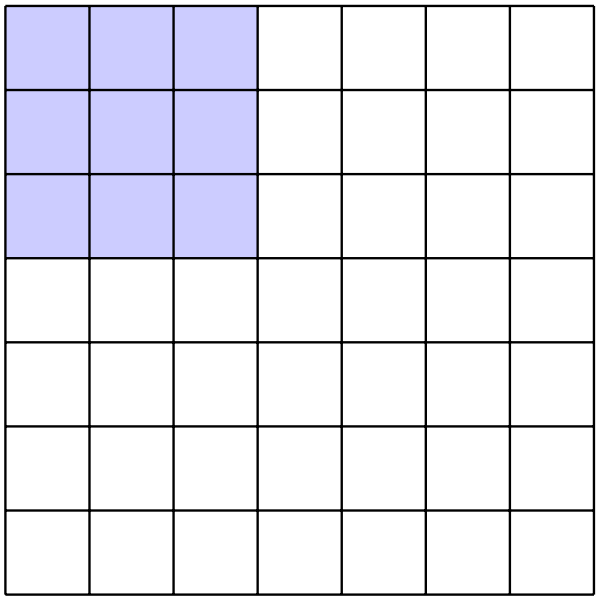
. . .

A closer look at the spatial dimensions



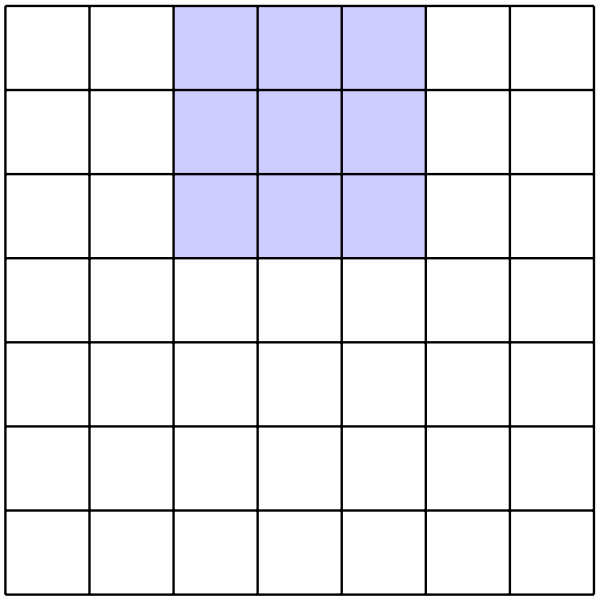
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- $\implies$ 5×5 output.

A closer look at the spatial dimensions



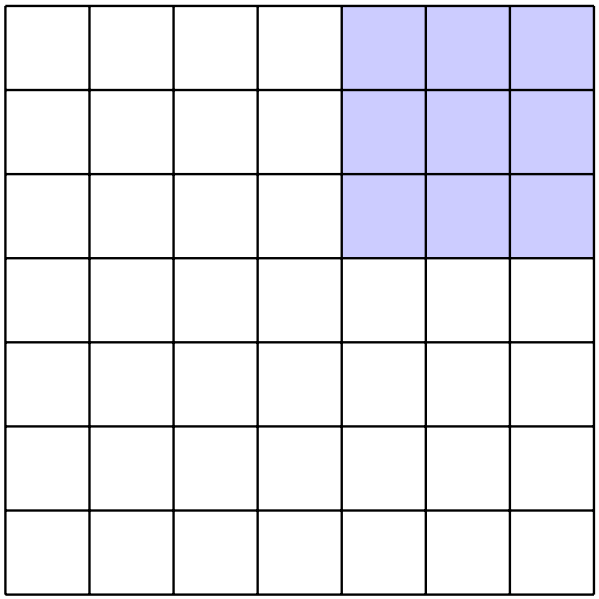
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 2.

A closer look at the spatial dimensions



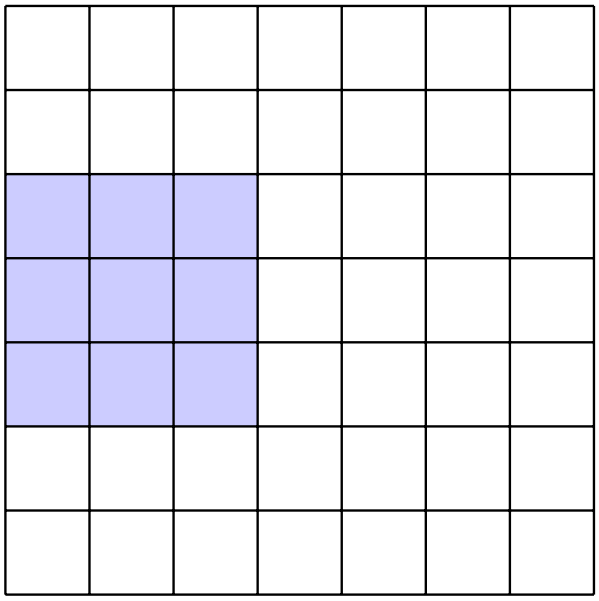
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 2.

A closer look at the spatial dimensions



- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 2.

A closer look at the spatial dimensions

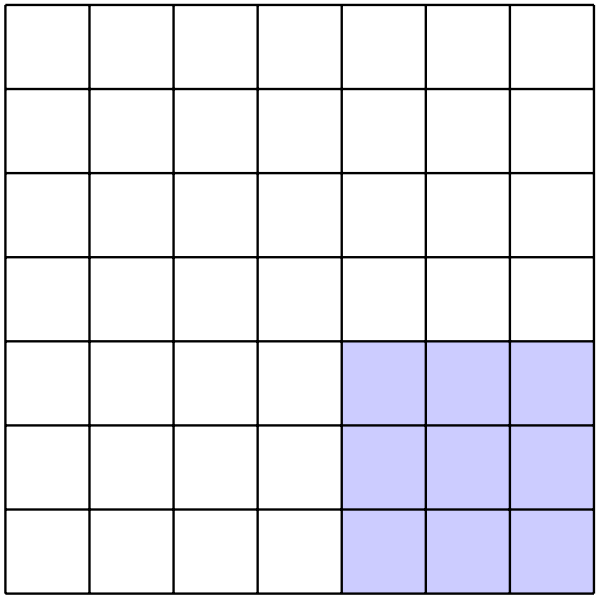


- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 2.

A closer look at the spatial dimensions

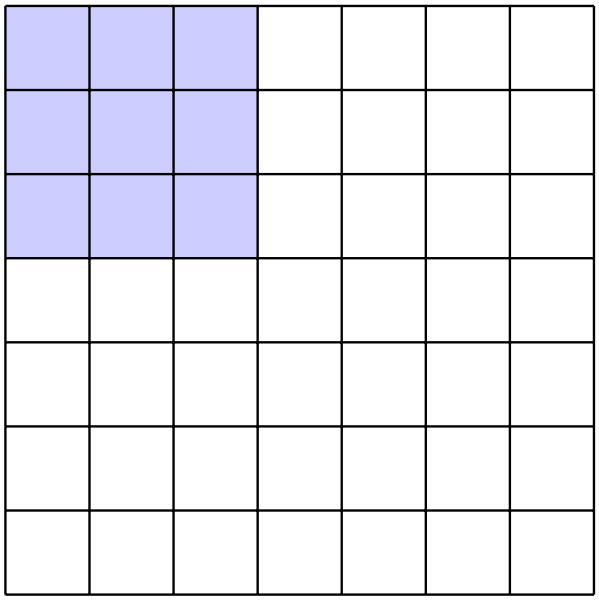
. . .

A closer look at the spatial dimensions



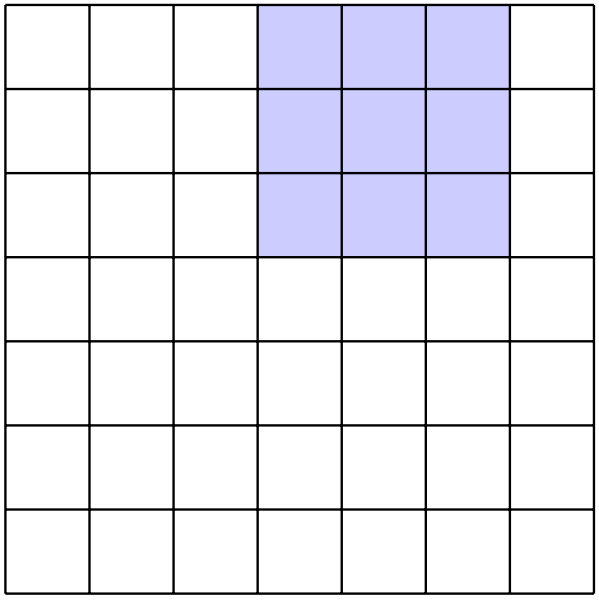
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 2.
- $\implies$ 3×3 output.

A closer look at the spatial dimensions



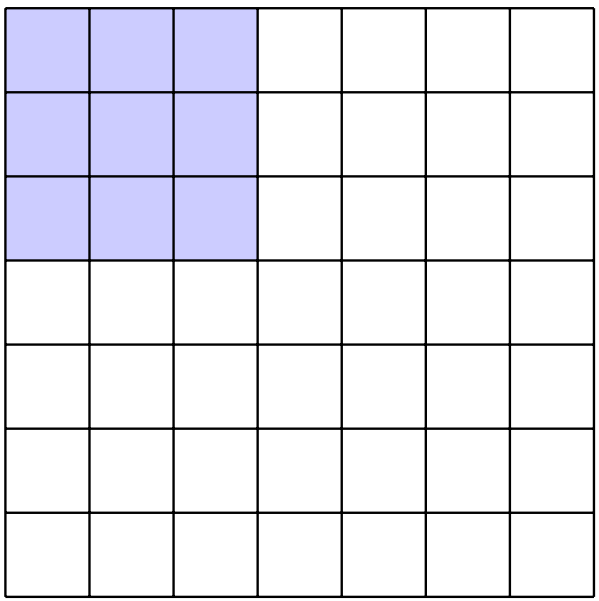
- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 3.

A closer look at the spatial dimensions



- $7 \times 7 \times D$ input (just display one plane).
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** 3.
- **Doesn't fit nicely**! Don't include last column and row.

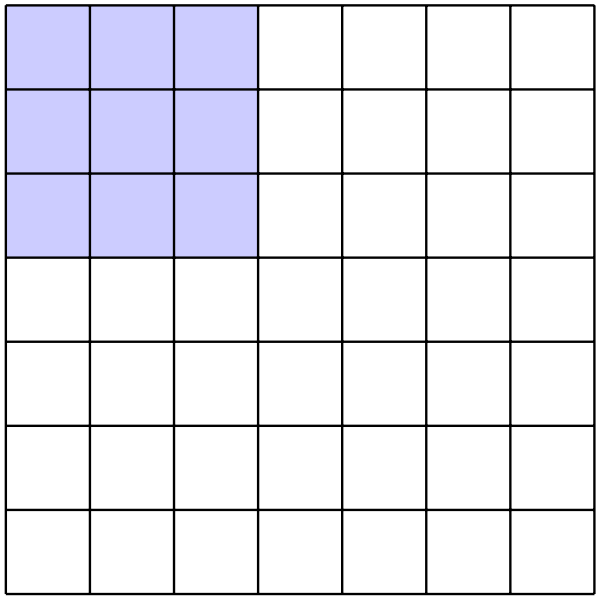
Output volume dimension



- $w \times h \times D$ input.
- Have $f \times f \times D$ filter.
- Apply with **stride** s.
- **Output dimension**: $w' \times h'$

$$w' = (w - f)/s + 1$$
$$h' = (h - f)/s + 1$$

Output volume dimension



- $w \times h \times D$ input.
- Have $f \times f \times D$ filter.
- Apply with **stride** s.
- **Output dimension**: $w' \times h'$

$$w' = (w - f)/s + 1$$
$$h' = (h - f)/s + 1$$

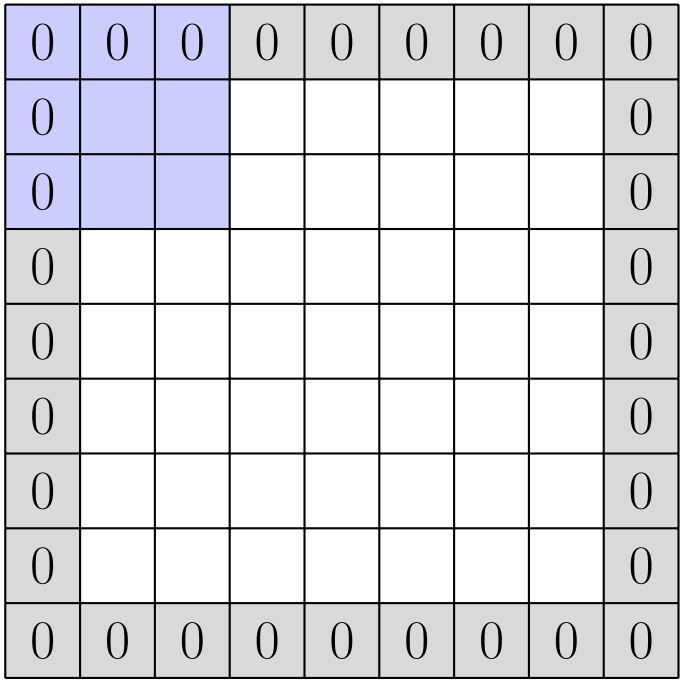
- **Our example**:
  $w = h = 7, f = 3$

  s=1 $\implies$ $w' = (7-3)/1 + 1 = 5$

  s=2 $\implies$ $w' = (7-3)/2 + 1 = 3$

  s=3 $\implies$ $w' = (7-3)/3 + 1 = 2.33$

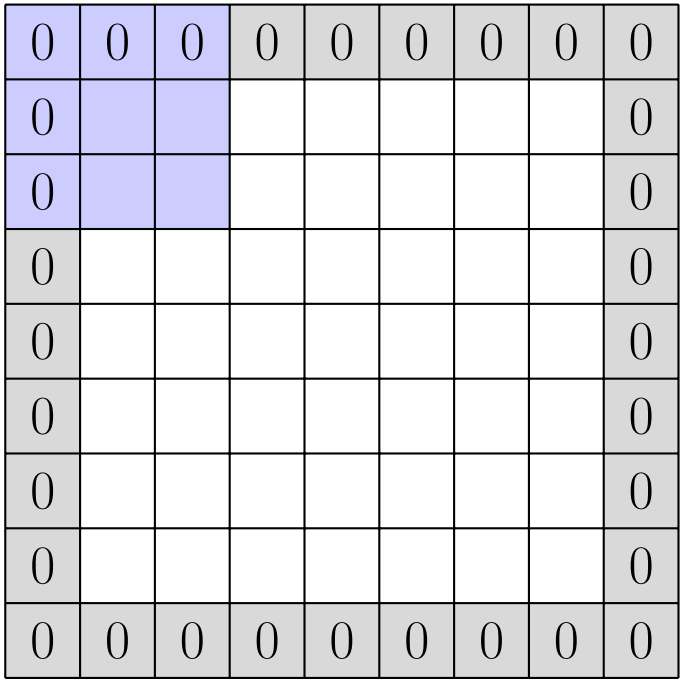
In practice: Common to zero pad the border



- $7 \times 7 \times D$ input.
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** s.
- Pad input with 1 pixel border.
- **Output dimension**: ?
- Remember

$$w' = (w - f)/s + 1$$
$$h' = (h - f)/s + 1$$

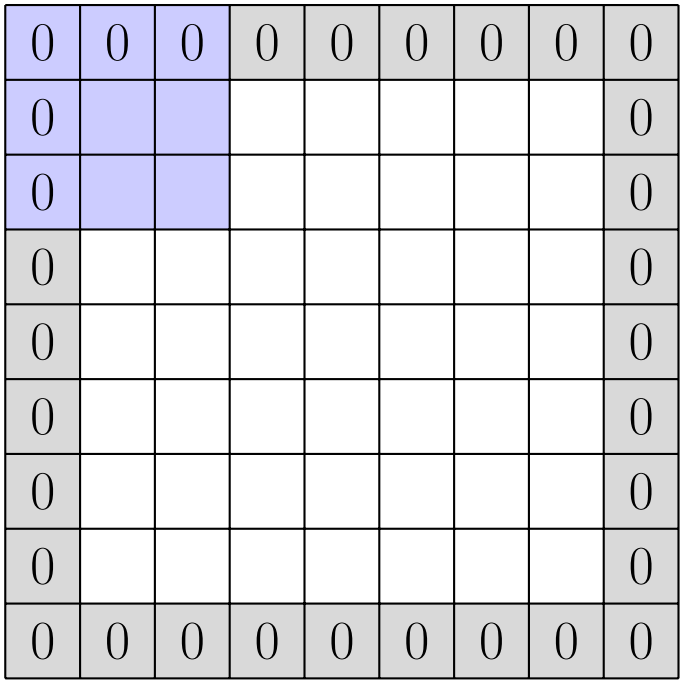
In practice: Common to zero pad the border



- $7 \times 7 \times D$ input.
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** s.
- Pad input with 1 pixel border.
- **Output dimension**: ?
- Remember

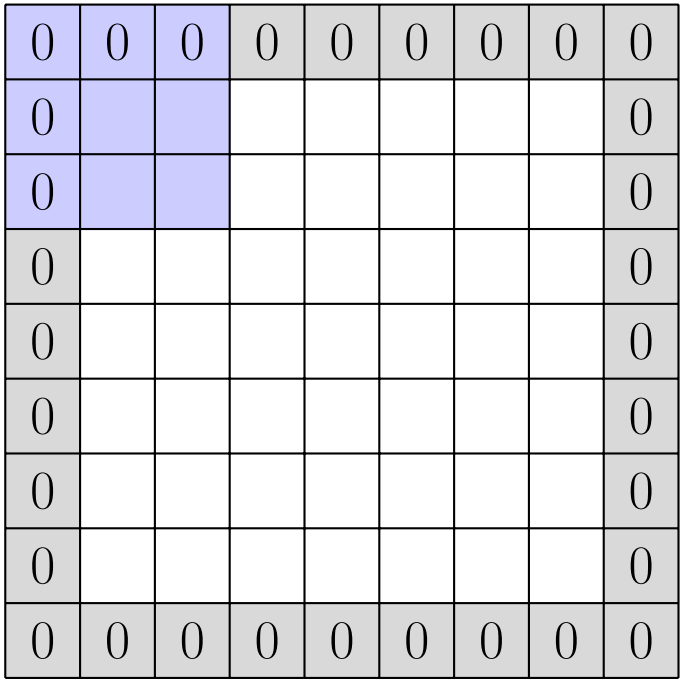
$$w' = (w - f)/s + 1$$
$$h' = (h - f)/s + 1$$

In practice: Common to zero pad the border



- $7 \times 7 \times D$ input.
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** s.
- Pad input with 1 pixel border.
- **Output dimension**: 7×7

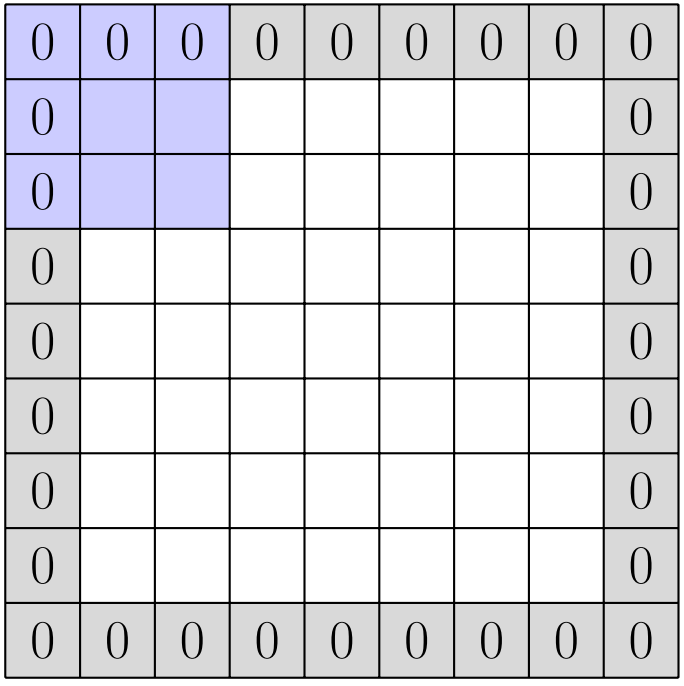
In practice: Common to zero pad the border



- $7 \times 7 \times D$ input.
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** s.
- Pad input with 1 pixel border. $(P = 1)$
- **Output dimension**: 7×7
- In general

$$w' = (w + 2P - f)/s + 1$$
$$h' = (h + 2P - f)/s + 1$$

In practice: Common to zero pad the border



- $7 \times 7 \times D$ input.
- Have $3 \times 3 \times D$ filter.
- Apply with **stride** s.
- Pad input with 1 pixel border. $(P = 1)$
- **Output dimension**: 7×7
- In general, common to have convolutional layers with
  - stride $s = 1$,
  - filters of size $f \times f \times D$,
  - zero-padding $P = (f - 1)/2$

  $\implies w' = w, h' = h$

Example

- **Input volume dimension**: $32 \times 32 \times 3$
- Hyper-parameters of Conv layer:
  - 10 filters of size $5 \times 5 \times 3$
  - Stride: $s = 1$
  - Pad: $P = 2$
- **Output volume dimension**: ?

Example of Dimension Counting

- **Input volume dimension**: $32 \times 32 \times 3$
- Hyper-parameters of Conv layer:
  - 10 filters of size $5 \times 5 \times 3$
  - Stride: $s = 1$
  - Pad: $P = 2$
- **Output volume dimension**: $w' \times h' \times 10$
  where

$$w' = (w - 2P - f)/s + 1 = (32 - 2 * 2 - 5)/1 + 1 = 32$$
$$h' = (h - 2P - f)/s + 1 = (32 - 2 * 2 - 5)/1 + 1 = 32$$

Example of Dimension Counting

- **Input volume dimension**: $32 \times 32 \times 3$
- Hyper-parameters of Conv layer:
  - 10 filters of size $5 \times 5 \times 3$
  - Stride $s = 1$
  - Pad $P = 2$
- **# of parameters in this layer**: ?

Example of Dimension Counting

- **Input volume dimension**: $32 \times 32 \times 3$
- Hyper-parameters of Conv layer:
  - 10 filters of size $5 \times 5 \times 3$
  - Stride: $s = 1$
  - Pad: $P = 2$
- **# of parameters in this layer**: 760
  - Each filter has $5 \times 5 \times 3 + 1 = 76$ parameters.
  - There are 10 filters $\implies$ 76×10 total parameters.

Summary of dimensions in Convolutional Layer

- **Input**:
  Volume, $\mathbf{X}^{(i)} = \{X_1^{(i)}, \ldots, X_D^{(i)}\}$, of size $w \times h \times D$

- Requires 4 hyper-parameters:
  - Number of filters: n_F
  - Spatial size of filters: f
  - Stride: s
  - Amount of zero padding: P

- **Output**:
  Volume $\mathbf{S}^{(i+1)} = \{S_1^{(i+1)}, \ldots, S_{n_F}^{(i+1)}\}$ size $w' \times h' \times n_F$ where
  - $w' = (w - f + 2P)/s + 1$
  - $h' = (h - f + 2P)/s + 1$

  where

$$S_j^{(i+1)} = \mathbf{X}^{(i)} * \mathbf{F}_j + b_j$$

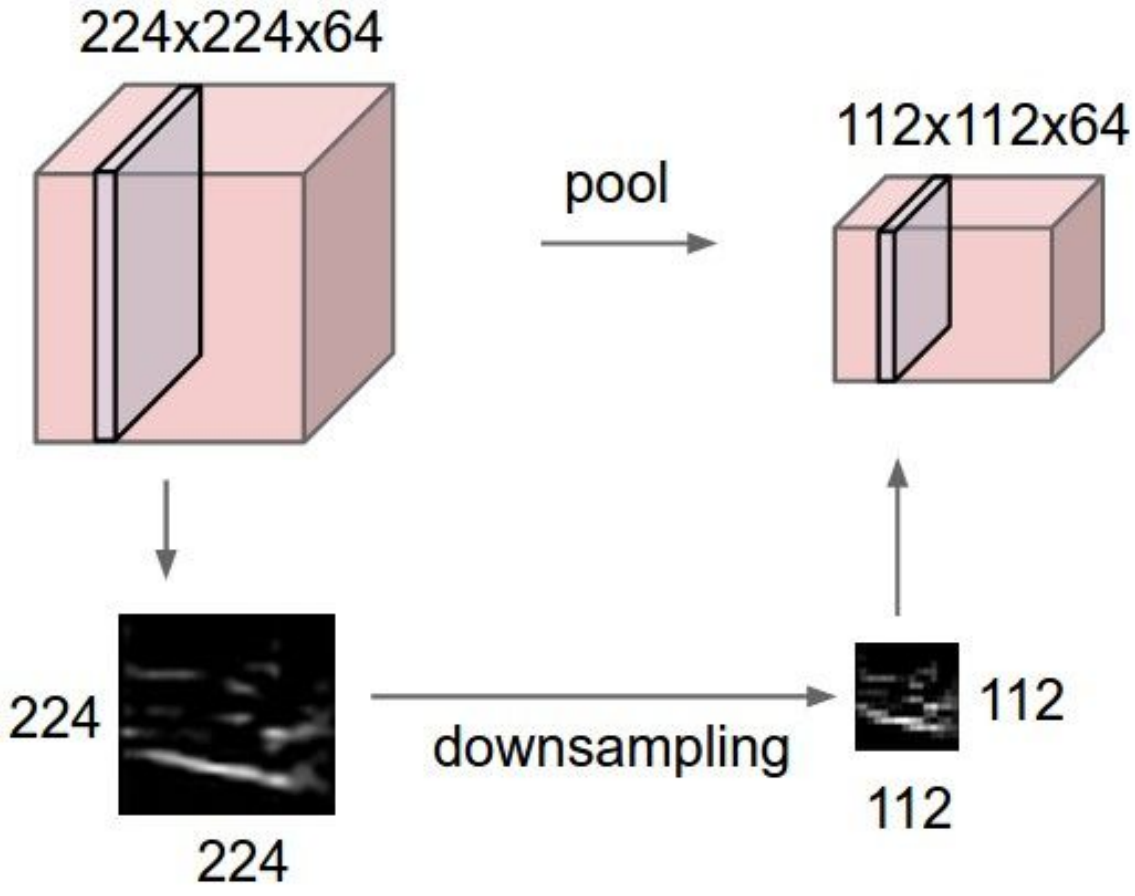
Common Settings

- $n_F =$ powers of 2 (e.g. 32, 64, 128, 512)
  - $f = 3, s = 1, P = 1$
  - $f = 5, s = 1, P = 2$
  - $f = 5, s = 2, P =?$ (whatever fits)
  - $f = 1, s = 1, P = 1$

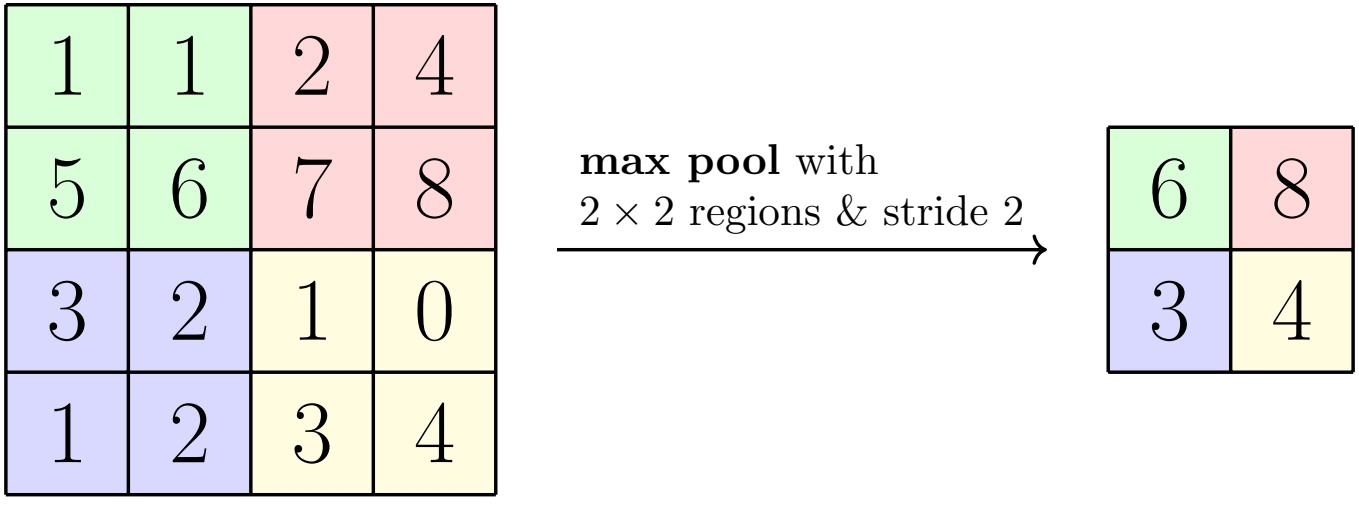
Common Operator in Convolutional Networks: **Pooling Operators**

Pooling layer

- Make the representation smaller and more manageable
- Operates over each response/activation map independently



Simple Example: Max Pooling one response map



Max Pooling

- Denote the max pooling operation with regions of size $r \times r$ and stride s with

$$\tilde{\mathbf{H}} = \text{MaxPool}(\mathbf{H}, r, s)$$

- If $\mathbf{H}$ has size $w \times h \times D$
  $\implies$ $\tilde{\mathbf{H}}$ has size $(w-r)/s+1 \times (h-r)/s+1 \times D$

- Mathematical expression for each entry in $\tilde{\mathbf{H}}$:

$$\tilde{\mathbf{H}}_{ijk} = \max_{\substack{i' \le l \le i'+r-1 \\ j' \le m \le j'+r-1}} \mathbf{H}_{lmk} \quad \text{where } i' = (i-1)s+1, j' = (j-1)s+1$$

- Common settings $r = 2, s = 2$ or $r = 3, s = 2$.

Max Pooling

- Denote the max pooling operation with regions of size $r \times r$ and stride s with

$$\tilde{\mathbf{H}} = \text{MaxPool}(\mathbf{H}, r, s)$$

- If $\mathbf{H}$ has size $w \times h \times D$
  $\implies$ $\tilde{\mathbf{H}}$ has size $(w-r)/s+1 \times (h-r)/s+1 \times D$

- Mathematical expression for each entry in $\tilde{\mathbf{H}}$:

$$\tilde{\mathbf{H}}_{ijk} = \max_{\substack{i' \le l \le i'+r-1 \\ j' \le m \le j'+r-1}} \mathbf{H}_{lmk} \quad \text{where } i' = (i-1)s+1, j' = (j-1)s+1$$

- Common settings $r = 2, s = 2$ or $r = 3, s = 2$.

Jacobian Computations for a Max Pooling layer

- Let

$$\tilde{\mathbf{H}} = \text{MaxPool}(\mathbf{H}, r, r)$$

- For backprop need to calculate:

$$\frac{\partial \text{vec}(\tilde{\mathbf{H}})}{\partial \text{vec}(\mathbf{H})} = ?$$

- Can write MaxPool operation as a matrix operation

$$\text{vec}(\tilde{\mathbf{H}}) = A_{\text{MaxPool}} \text{vec}(\mathbf{H})$$

$$\Longrightarrow$$

$$\frac{\partial \text{vec}(\tilde{\mathbf{H}})}{\partial \text{vec}(\mathbf{H})} = A_{\text{MaxPool}}$$

- What are the entries of A_{MaxPool}?

Jacobian Computations for a Max Pooling layer

- Let

$$\tilde{\mathbf{H}} = \text{MaxPool}(\mathbf{H}, r, r)$$

- For backprop need to calculate:

$$\frac{\partial \text{vec}(\tilde{\mathbf{H}})}{\partial \text{vec}(\mathbf{H})} = ?$$

- Can write MaxPool operation as a matrix operation

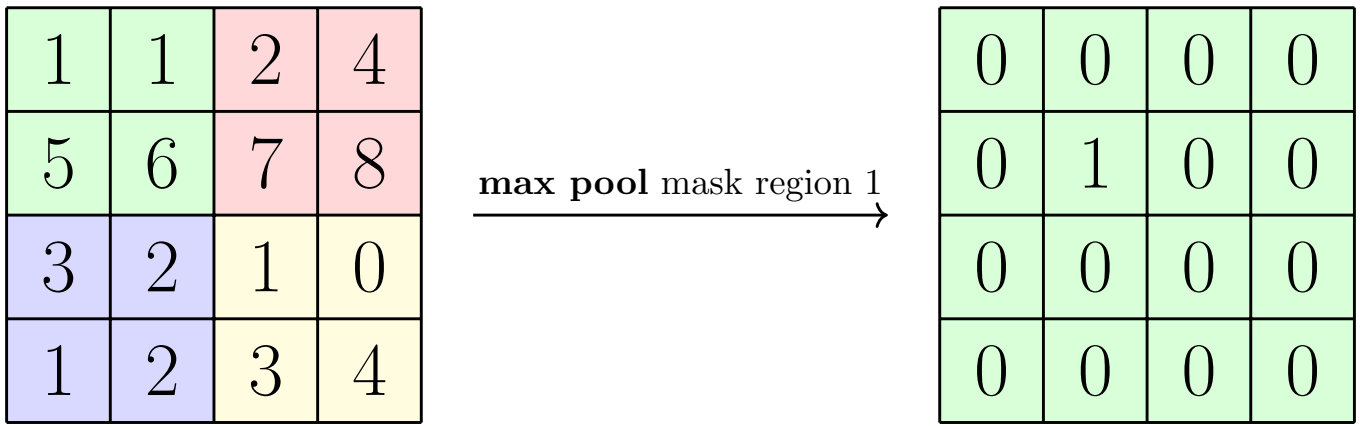
$$\text{vec}(\tilde{\mathbf{H}}) = A_{\text{MaxPool}} \text{vec}(\mathbf{H})$$

$$\Longrightarrow$$

$$\frac{\partial \text{vec}(\tilde{\mathbf{H}})}{\partial \text{vec}(\mathbf{H})} = A_{\text{MaxPool}}$$

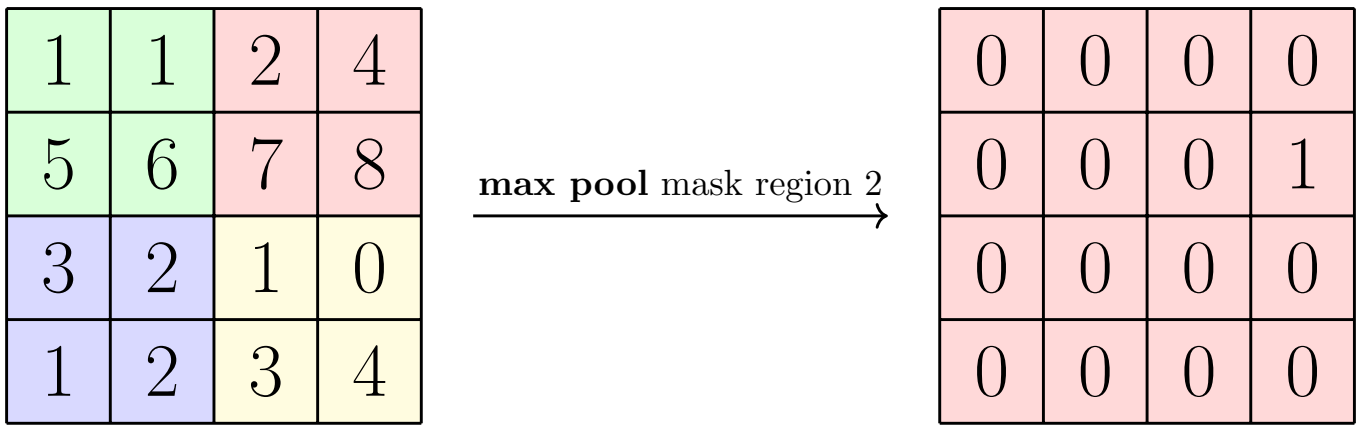
- What are the entries of A_{MaxPool}?

Simple Example: Max Pooling as a matrix multiplication



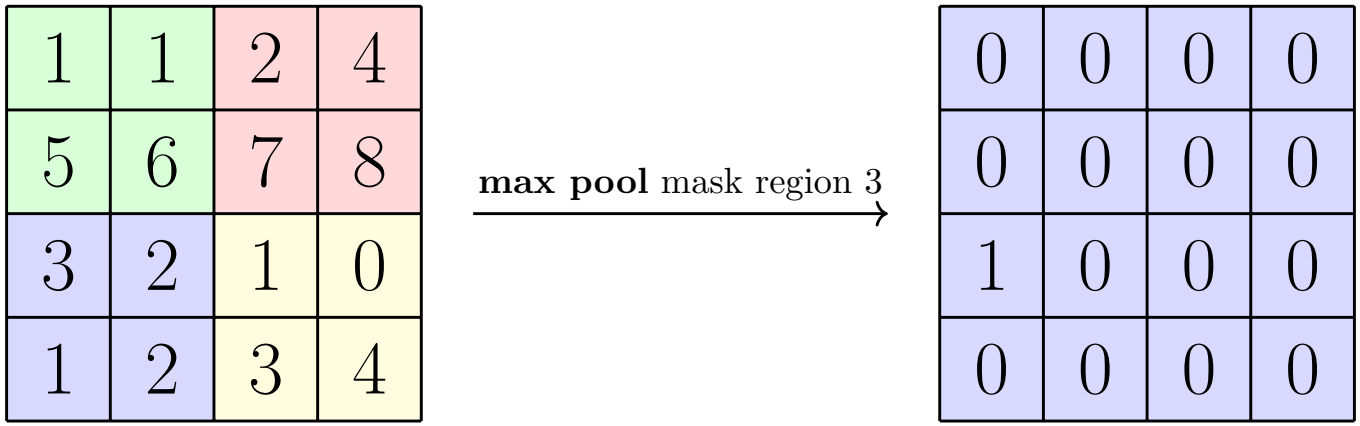
$$(\tilde{H}_{11}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathsf{vec}(H)$$

Simple Example: Max Pooling as a matrix multiplication



$$\begin{pmatrix} \tilde{H}_{11} \\ \tilde{H}_{12} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathsf{vec}(H)$$

Simple Example: Max Pooling as a matrix multiplication



$$\begin{pmatrix} \tilde{H}_{11} \\ \tilde{H}_{12} \\ \tilde{H}_{21} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathsf{vec}(H)$$

Simple Example: Max Pooling as a matrix multiplication

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

$\xrightarrow{\textbf{max pool}\text{ mask region 4}}$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1

$$\begin{pmatrix} \tilde{H}_{11} \\ \tilde{H}_{12} \\ \tilde{H}_{21} \\ \tilde{H}_{22} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathsf{vec}(H)$$

Simple Example: Max Pooling as a matrix multiplication

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

$\xrightarrow{\textbf{max pool} \text{ mask region } 4}$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1

$$\begin{pmatrix} \tilde{H}_{11} \\ \tilde{H}_{12} \\ \tilde{H}_{21} \\ \tilde{H}_{22} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{A_{\text{MaxPool}} \text{ and has size } 4 \times 16} \text{vec}(H)$$

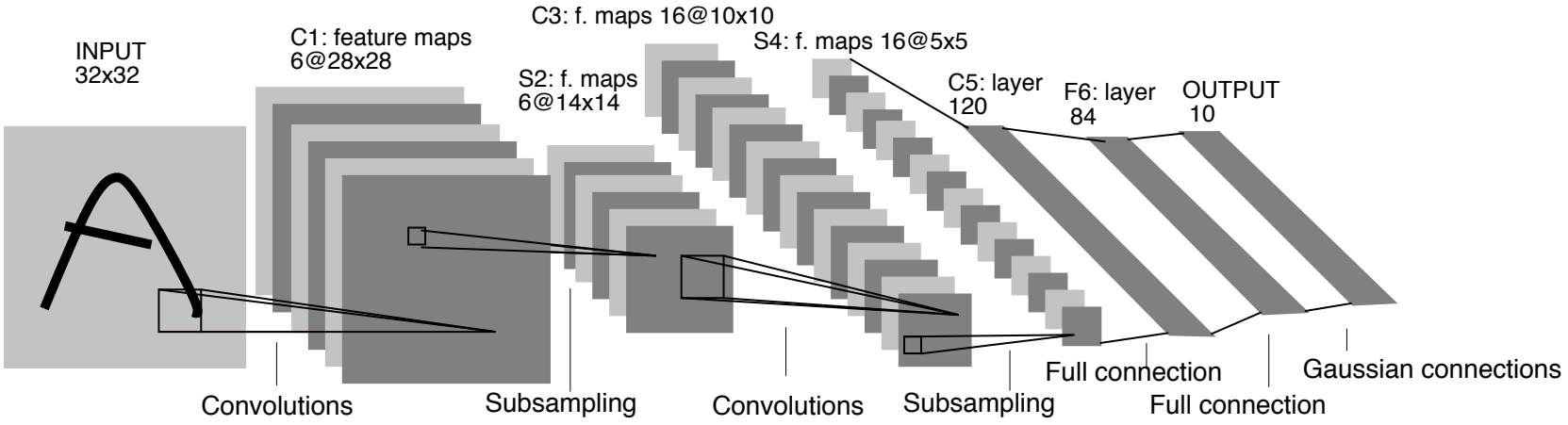
What is A_{AvgPool}?

- Say in each region we pool by **averaging** the responses instead of choosing the max.

- What is A_{AvgPool} for our simple example?

Architecture of Modern ConvNets

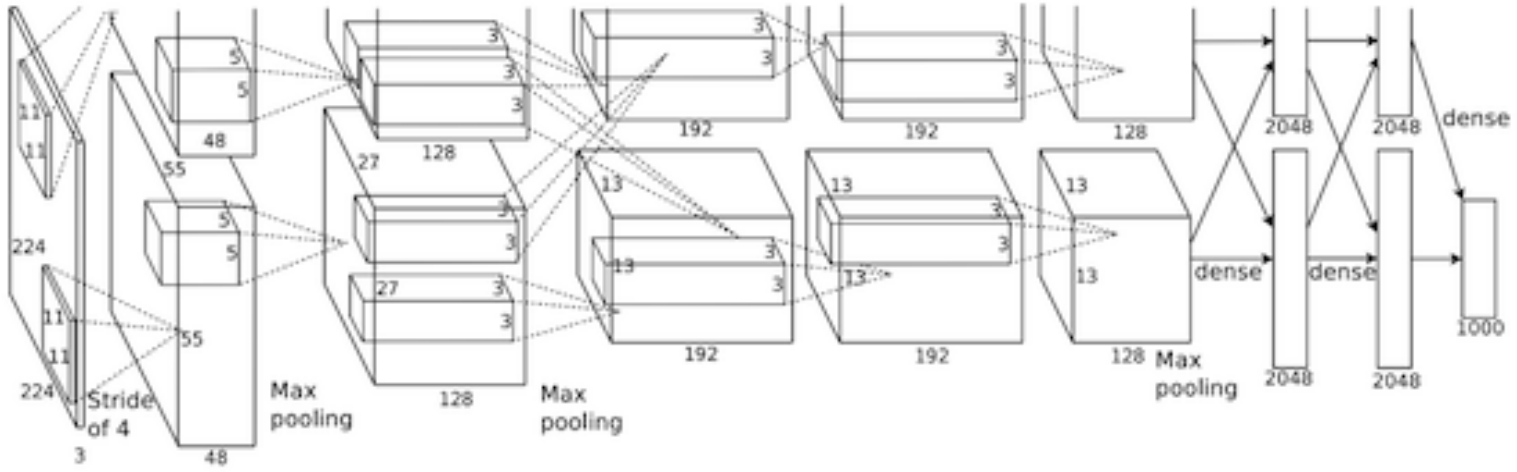
LeNet-5 [LeCun et al., 1998]



- Conv filters are 5×5, applied at stride 1
- Pooling layers are 2×2, applied at stride 2
- Architecture is

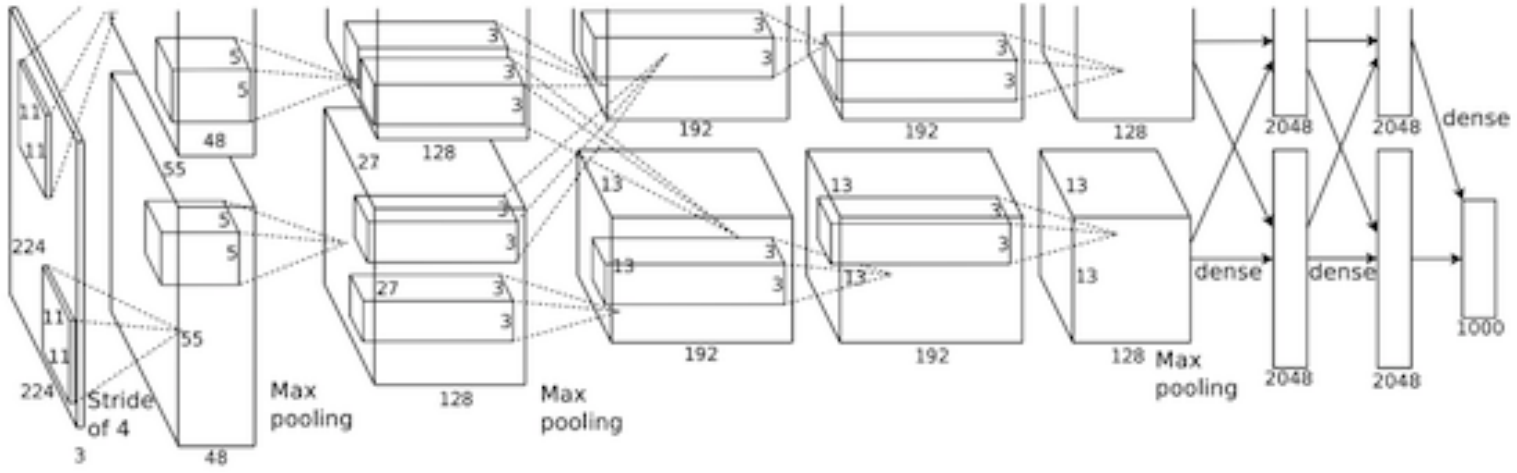
  [CONV-POOL-TANH-CONV-POOL-TANH-FC-TANH-FC-TANH-FC]

AlexNet [Krizhevsky et al. 2012]



- **Input**: $227 \times 227 \times 3$ image
- **First layer**:
  - Convolutional layer
  - 96 filters of size $11 \times 11 \times 3$ applied at stride 4
- What is the output volume size?

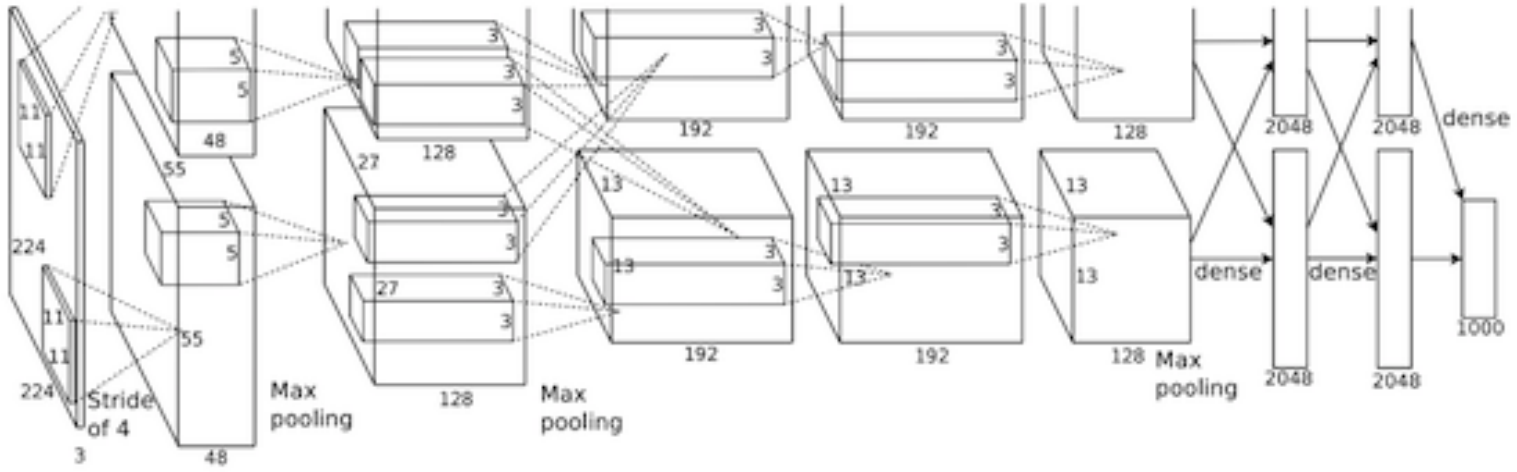
AlexNet [Krizhevsky et al. 2012]



- **Input**: $227 \times 227 \times 3$ image
- **First layer**:
  - Convolutional layer
  - 96 filters of size $11 \times 11 \times 3$ applied at stride 4
- What is the output volume size? $55 \times 55 \times 96$

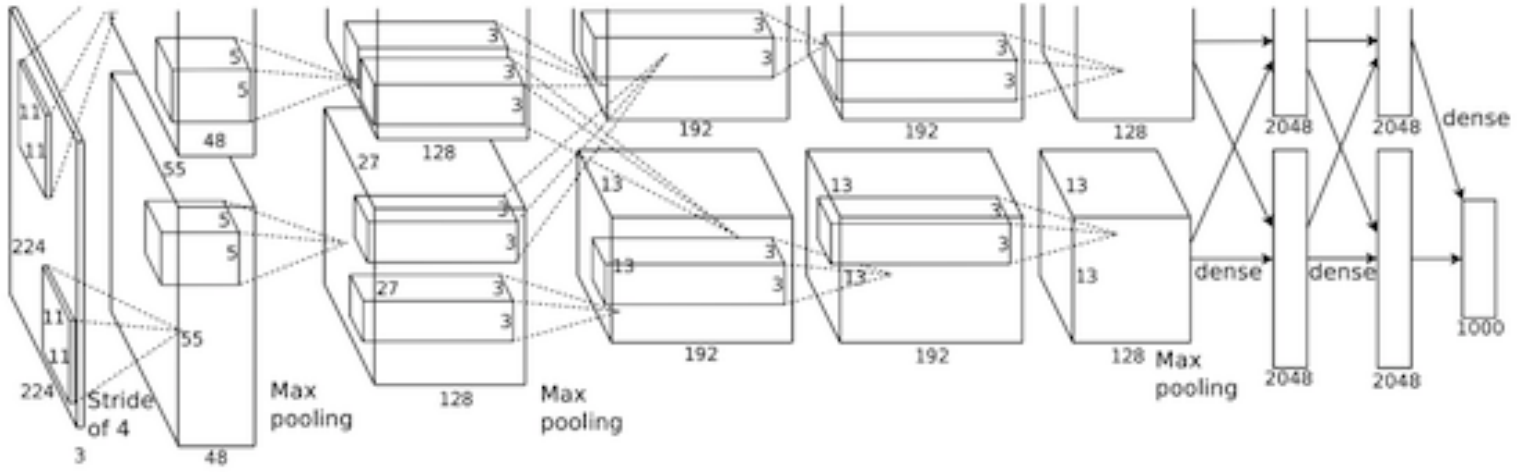
  as (227 -11)/4 + 1 = 55.

AlexNet [Krizhevsky et al. 2012]



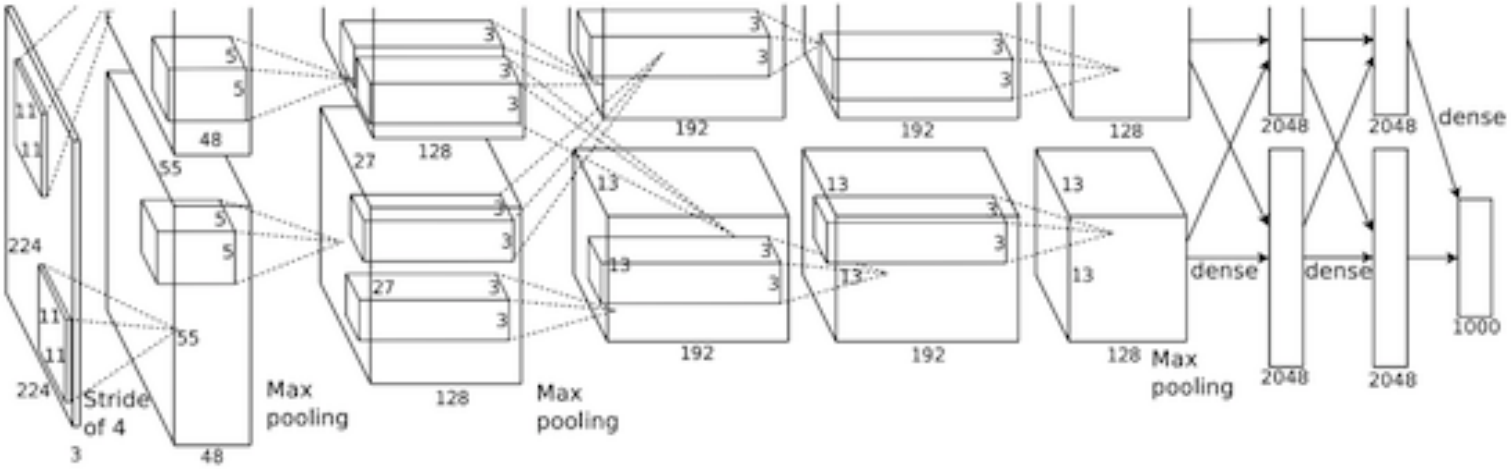
- **Input**: $227 \times 227 \times 3$ image
- **First layer**:
  - Convolutional layer
  - 96 filters of size $11 \times 11 \times 3$ applied at stride 4
- What is the output volume size? $55 \times 55 \times 96$
- # of parameters: $11 \times 11 \times 3 \times 96 = 35k$

AlexNet [Krizhevsky et al. 2012]



- **Input size**: $227 \times 227 \times 3$ image
- **Size after 1st layer**: $55 \times 55 \times 96$ image
- **Second layer**:
  - Pooling layer
  - 3×3 regions applied at stride 2
- What is the output volume size?

AlexNet [Krizhevsky et al. 2012]



- **Input size**: $227 \times 227 \times 3$ image
- **Size after 1st layer**: $55 \times 55 \times 96$ image
- **Second layer**:
  - Pooling layer
  - 3×3 regions applied at stride 2
- What is the output volume size? $27 \times 27 \times 96$

  as (55 - 3)/2 + 1 = 27.

AlexNet [Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

Output size	**Layer type**	**Details**
$227 \times 227 \times 3$	Input	
$55 \times 55 \times 96$	Conv	96 filters, size 11×11 at stride 4, pad 0
$27 \times 27 \times 96$	Max Pool	3×3 regions at stride 2
$27 \times 27 \times 96$	Norm	Normalization layer
$27 \times 27 \times 256$	Conv	256 filters, size 5×5 at stride 1, pad 2
$13 \times 13 \times 256$	Max Pool	3×3 regions at stride 2
$13 \times 13 \times 256$	Norm	Normalization layer
$13 \times 13 \times 384$	Conv	384 filters, size 3×3 at stride 1, pad 1
$13 \times 13 \times 384$	Conv	384 filters, size 3×3 at stride 1, pad 1
$13 \times 13 \times 256$	Conv	256 filters, size 3×3 at stride 1, pad 1
$6 \times 6 \times 256$	Max Pool	3×3 regions at stride 2
4096	Fully connected	4096 neurons
4096	Fully connected	4096 neurons
1000	Fully connected	1000 neurons (class scores)

AlexNet [Krizhevsky et al. 2012]

Details/Retrospectives:

- First use of ReLU
- Used Normalization layers (not common anymore).
- Heavy data augmentation
- Dropout training with $p = 0.5$
- Batch size: 128
- Mini-batch GD + Momentum with $\alpha = 0.9$
- Learning rate initialized: 1e-2; divided by 10 when validation accuracy plateaus.
- L2 weight decay 5e-4
- 18.2% (1 CNN) $\rightarrow$ 15.4% (7 CNN ensemble)

VGGNet [Simonyan and Zisserman, 2014]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB imag)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

The 6 different architecures of VGG Net. Configuration D produced the best results

Best model marked by the yellow box.

VGGNet [Simonyan and Zisserman, 2014]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB imag)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

The 6 different architecures of VGG Net. Configuration D produced the best results

- Only filters of size 3×3 in the convolutional layers, stride 1 and pad 1.
- Relatively sparse use of **Max Pooling** with region size 2×2 and stride 2.
- **Improved ILSVRC Performance** top 5 error:

  2013 best 11.2% $\rightarrow$ 7.3% (VGGNet ensemble)

VGGNet architecture [Simonyan and Zisserman, 2014]

Output size	Layer type	Memory	# parameters
$224 \times 224 \times 3$	Input	224*224*3 = 150K	0
$224 \times 224 \times 64$	Conv	224*224*64 = 3.2M	1,728 = (3*3*3)*64
$224 \times 224 \times 64$	Conv	224*224*64 = 3.2M	36,864 = (3*3*64)*64
$112 \times 112 \times 64$	Max Pool	112*112*64 = 800K	0
$112 \times 112 \times 128$	Conv	112*112*128 = 1.6M	73,728 = (3*3*64)*128
$112 \times 112 \times 128$	Conv	112*112*128 = 1.6M	147,456 = (3*3*128)*128
$56 \times 56 \times 128$	Max Pool	56*56*128 = 400K	0
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	294,912 = (3*3*128)*256
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	589,824 = (3*3*256)*256
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	589,824 = (3*3*256)*256
$28 \times 28 \times 256$	Max Pool	28*28*256 = 200K	0
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	1,179,648 = (3*3*256)*512
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	2,359,296 = (3*3*512)*512
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Max Pool	14*14*512 = 100K	0
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$7 \times 7 \times 512$	Max Pool	7*7*512 = 25K	0
$1 \times 1 \times 4096$	Fully connected	4096	102,760,448 = 7*7*512*4096
$1 \times 1 \times 4096$	Fully connected	4096	16,777,216 = 4096*4096
$1 \times 1 \times 1000$	Fully connected	1000	4,096,000 = 4096*1000

VGGNet architecture [Simonyan and Zisserman, 2014]

Output size	Layer type	Memory	# parameters
$224 \times 224 \times 3$	Input	224*224*3 = 150K	0
$224 \times 224 \times 64$	Conv	224*224*64 = 3.2M	1,728 = (3*3*3)*64
$224 \times 224 \times 64$	Conv	224*224*64 = 3.2M	36,864 = (3*3*64)*64
$112 \times 112 \times 64$	Max Pool	112*112*64 = 800K	0
$112 \times 112 \times 128$	Conv	112*112*128 = 1.6M	73,728 = (3*3*64)*128
$112 \times 112 \times 128$	Conv	112*112*128 = 1.6M	147,456 = (3*3*128)*128
$56 \times 56 \times 128$	Max Pool	56*56*128 = 400K	0
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	294,912 = (3*3*128)*256
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	589,824 = (3*3*256)*256
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	589,824 = (3*3*256)*256
$28 \times 28 \times 256$	Max Pool	28*28*256 = 200K	0
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	1,179,648 = (3*3*256)*512
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	2,359,296 = (3*3*512)*512
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Max Pool	14*14*512 = 100K	0
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$7 \times 7 \times 512$	Max Pool	7*7*512 = 25K	0
$1 \times 1 \times 4096$	Fully connected	4096	102,760,448 = 7*7*512*4096
$1 \times 1 \times 4096$	Fully connected	4096	16,777,216 = 4096*4096
$1 \times 1 \times 1000$	Fully connected	1000	4,096,000 = 4096*1000

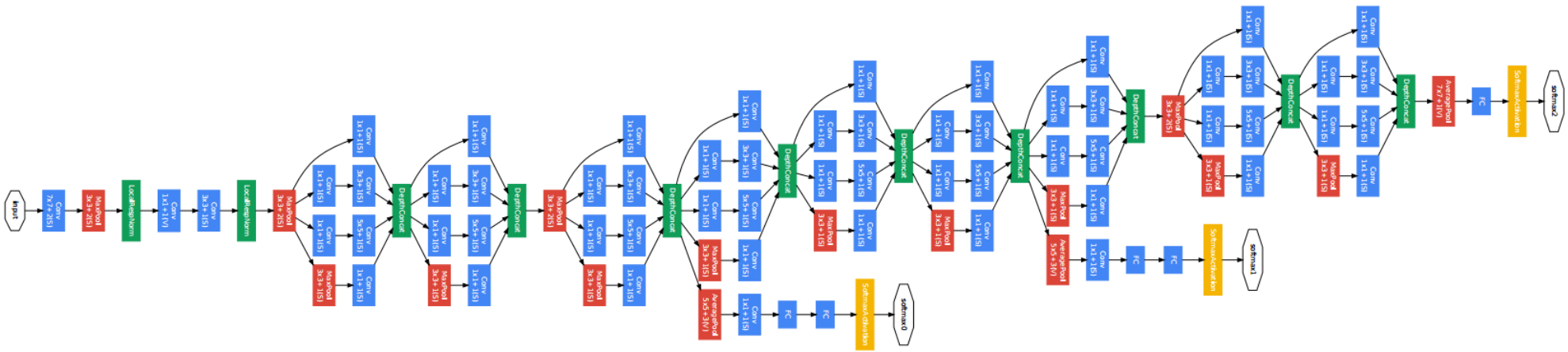
- **Total memory**: 24M $\times$ 4 bytes $\approx$ 93MB per image (for fwd pass, $\approx \times 2$ for bwd pass)
- **Total parameters**: 128 Million parameters

VGGNet architecture [Simonyan and Zisserman, 2014]

Output size	Layer type	Memory	# parameters
$224 \times 224 \times 3$	Input	224*224*3 = 150K	0
$224 \times 224 \times 64$	Conv	224*224*64 = 3.2M	1,728 = (3*3*3)*64
$224 \times 224 \times 64$	Conv	224*224*64 = 3.2M	36,864 = (3*3*64)*64
$112 \times 112 \times 64$	Max Pool	112*112*64 = 800K	0
$112 \times 112 \times 128$	Conv	112*112*128 = 1.6M	73,728 = (3*3*64)*128
$112 \times 112 \times 128$	Conv	112*112*128 = 1.6M	147,456 = (3*3*128)*128
$56 \times 56 \times 128$	Max Pool	56*56*128 = 400K	0
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	294,912 = (3*3*128)*256
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	589,824 = (3*3*256)*256
$56 \times 56 \times 256$	Conv	56*56*256 = 800K	589,824 = (3*3*256)*256
$28 \times 28 \times 256$	Max Pool	28*28*256 = 200K	0
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	1,179,648 = (3*3*256)*512
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	2,359,296 = (3*3*512)*512
$28 \times 28 \times 512$	Conv	28*28*512 = 400K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Max Pool	14*14*512 = 100K	0
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$14 \times 14 \times 512$	Conv	14*14*512 = 100K	2,359,296 = (3*3*512)*512
$7 \times 7 \times 512$	Max Pool	7*7*512 = 25K	0
$1 \times 1 \times 4096$	Fully connected	4096	102,760,448 = 7*7*512*4096
$1 \times 1 \times 4096$	Fully connected	4096	16,777,216 = 4096*4096
$1 \times 1 \times 1000$	Fully connected	1000	4,096,000 = 4096*1000

- Most memory is in early convolutional layers.
- Most parameters in the early fully connected layers.

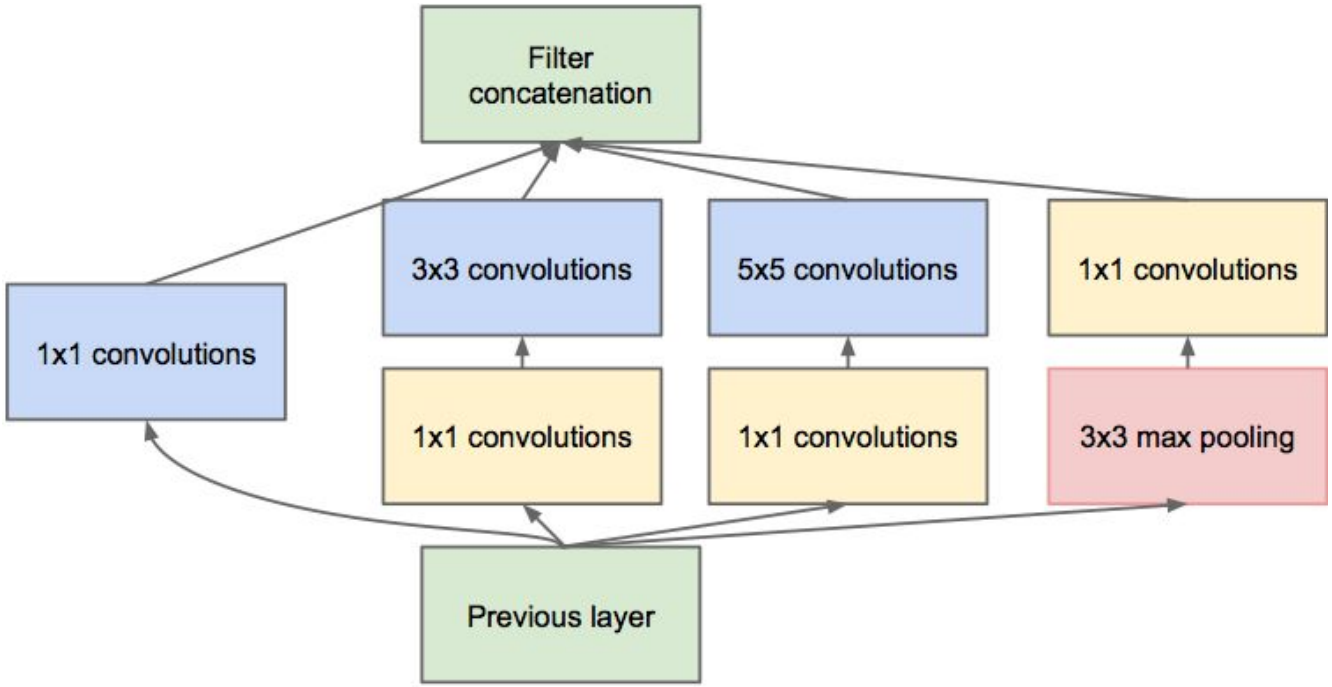
GoogLeNet [Szegedy et al., 2014]



- Convolution layer
- Pooling layer
- SoftMax layer

ILSVRC 2014 winner (6.7% top 5 error)

GoogLeNet: Inception Module



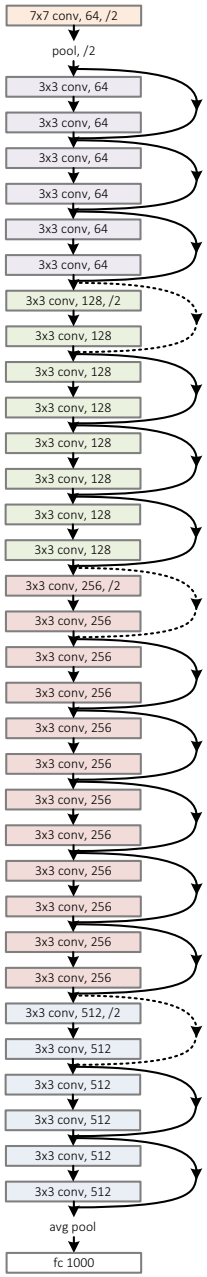
Use $1 \times 1 \times D$ filters to reduce the depth of the response volume before applying the larger more computationally expensive filters.

GoogLeNet: architecture

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

- Only 5 million params! (Only one FC layer for convenience)
- Compared to AlexNet:
  - 12× less params
  - 2× more compute
  - Performance on ILSVRC: 6.67% (vs. 16.4%)

ResNet [He et al., 2015]



- ILSVRC 2015 winner (3.6% top 5 error)
- 2-3 weeks of training on 8 GPU machine
- At runtime: faster than a VGGNet! (even though it has 8× more layers)

← "shallow" version of winning entry.

Next slides from:

Deep Residual Networks, Deep Learning Gets Way Deeper by Kaiming He,

ICML 2016 tutorial.

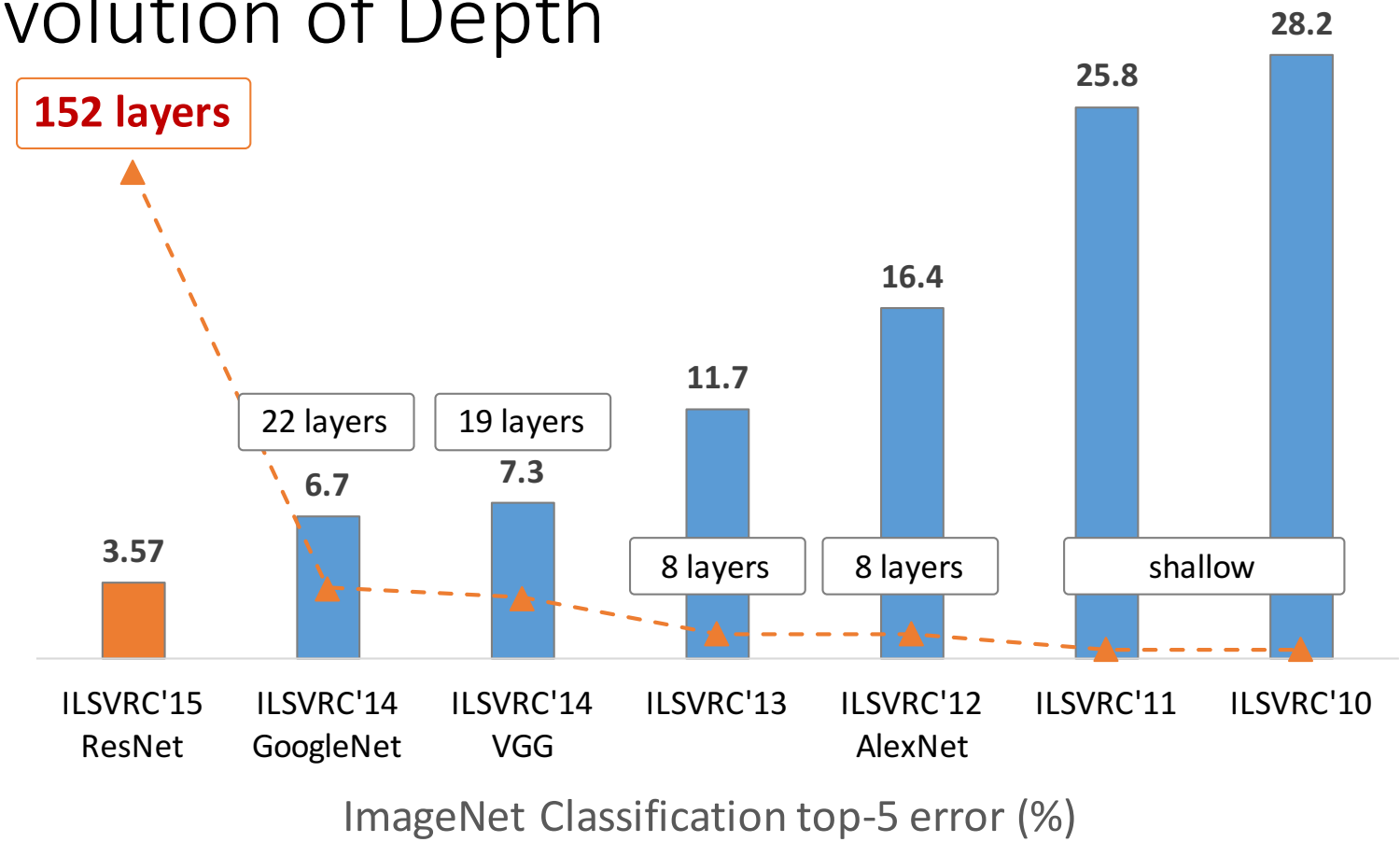
ResNets @ ILSVRC & COCO 2015 Competitions

- **1st places in all five main tracks**
  - ImageNet Classification: "*Ultra-deep*" 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

*improvements are relative numbers

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

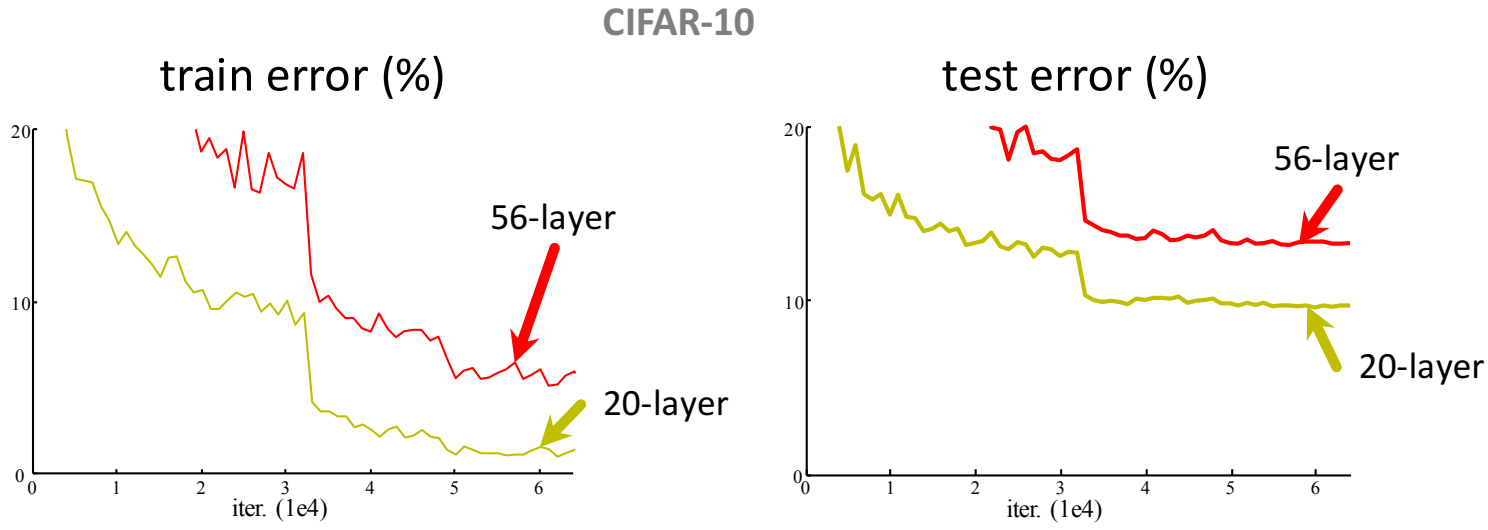
ResNet Motivation [He et al., 2015]

Going Deeper requires

- Care with initialization ✓
- Batch Normalization ✓

Is learning better networks as simple as stacking more layers?

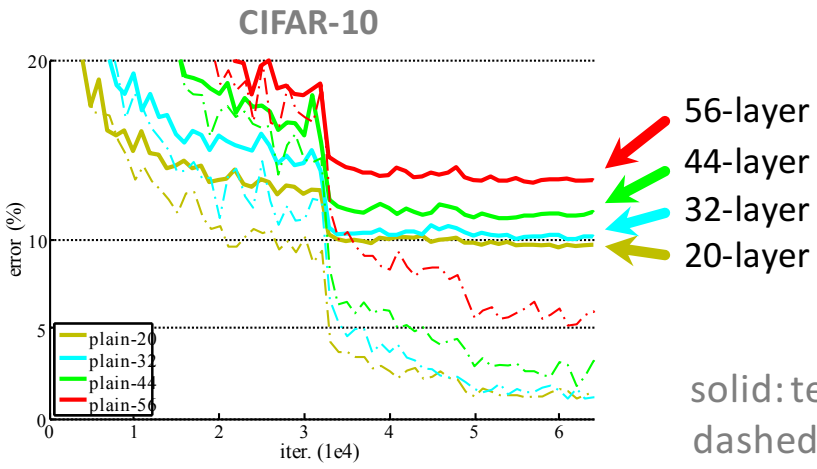
Simply stacking layers?

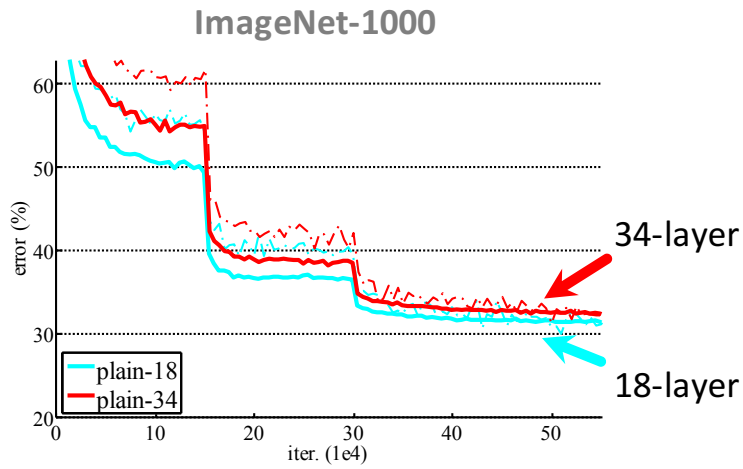


- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Simply stacking layers?

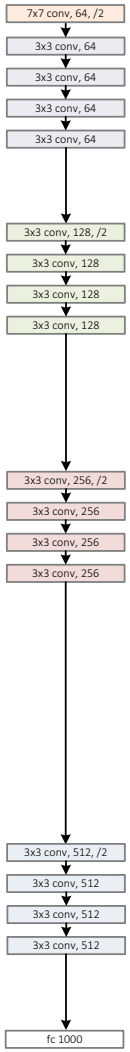


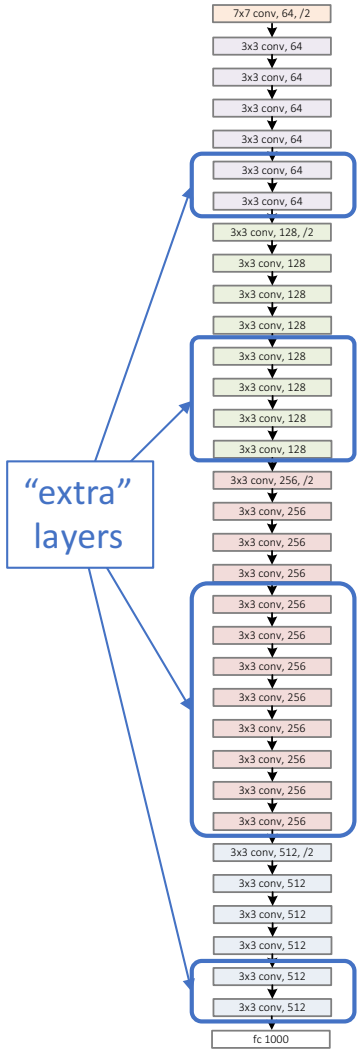


- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

a shallower model (18 layers)





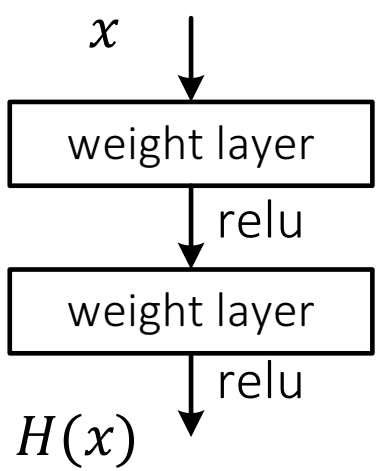
a deeper counterpart (34 layers)

- Richer solution space
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Residual Learning

- Plaint net



any two stacked layers

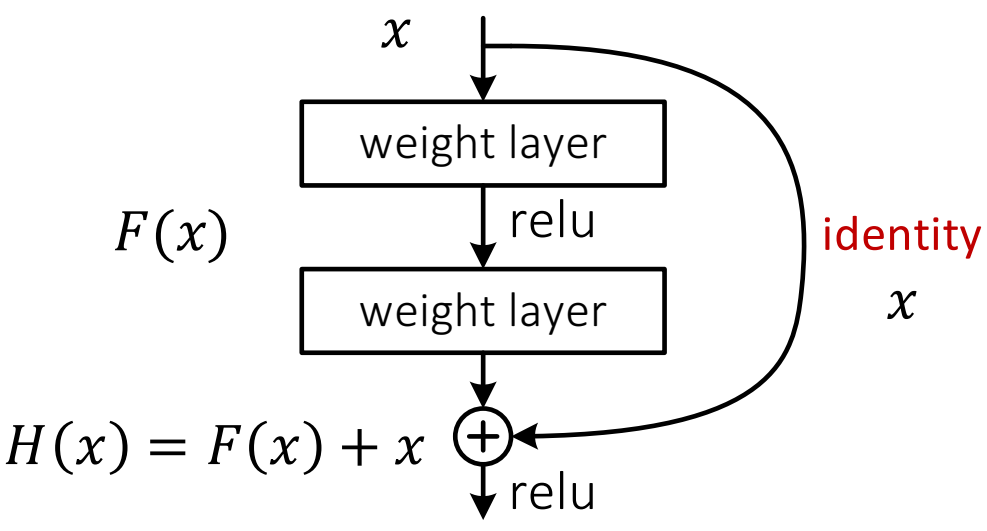
$H(x)$ is any desired mapping,

hope the 2 weight layers fit $H(x)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

~~hope the 2 weight layers fit $H(x)$~~

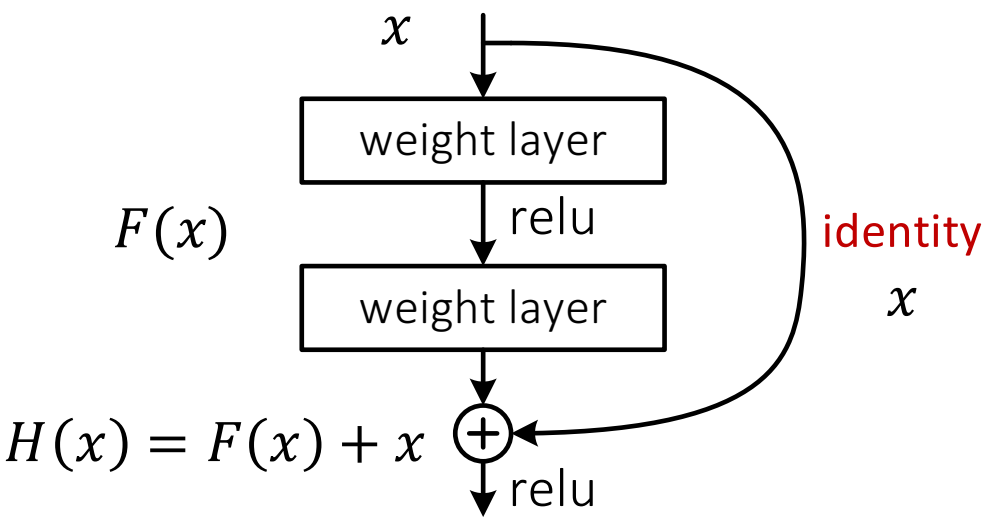
hope the 2 weight layers fit $F(x)$

let $H(x) = F(x) + x$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity



- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

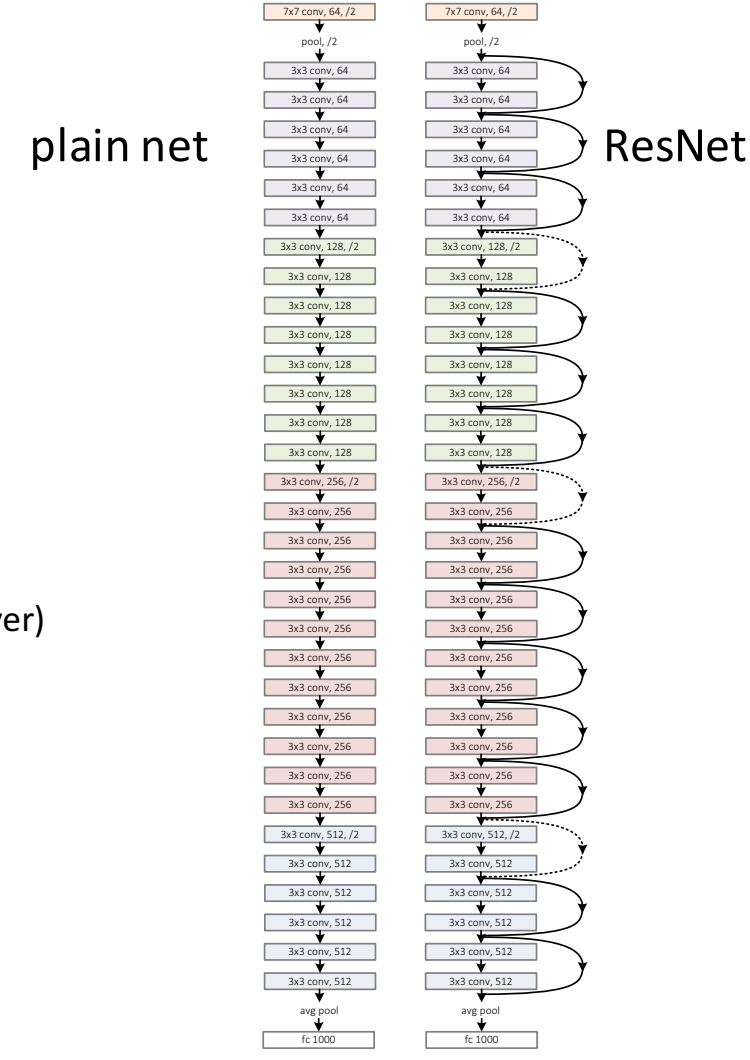
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Network “Design”

- Keep it simple

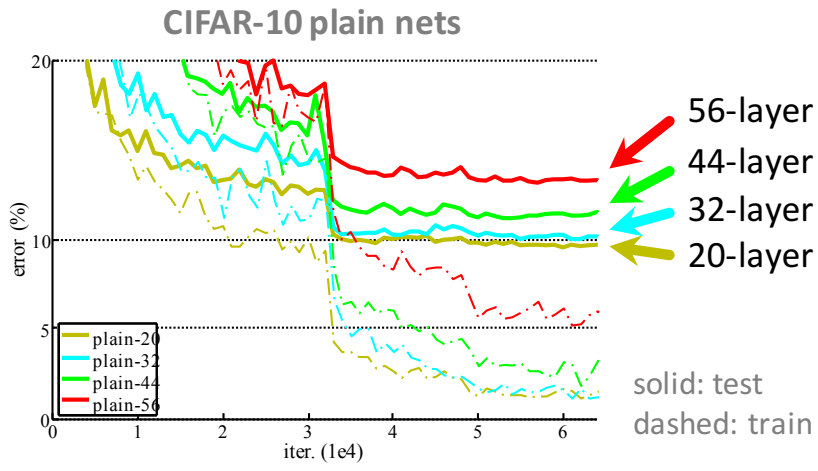
- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2 (~same complexity per layer)
  - Simple design; just deep!

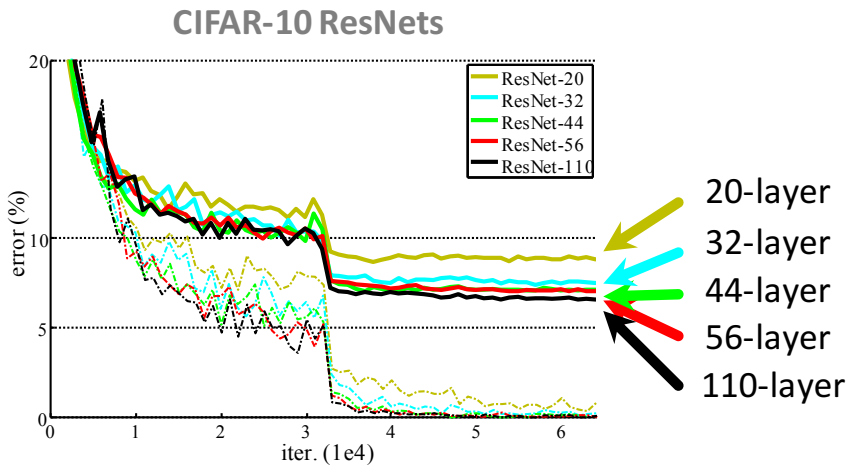
- Other remarks:
  - no hidden fc
  - no dropout



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

CIFAR-10 experiments



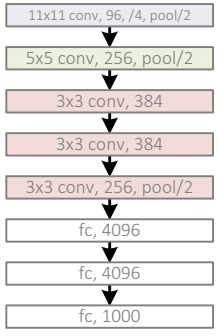


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

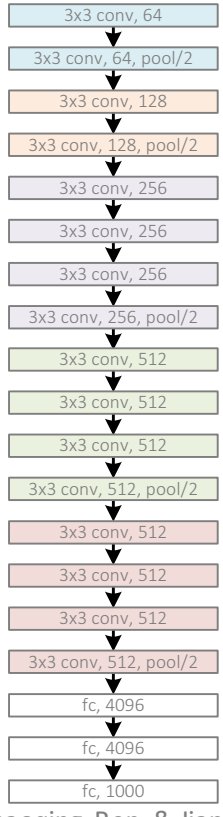
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

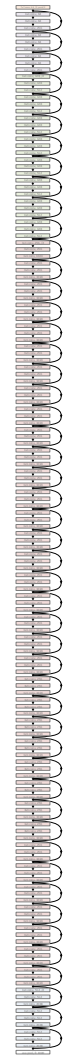
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

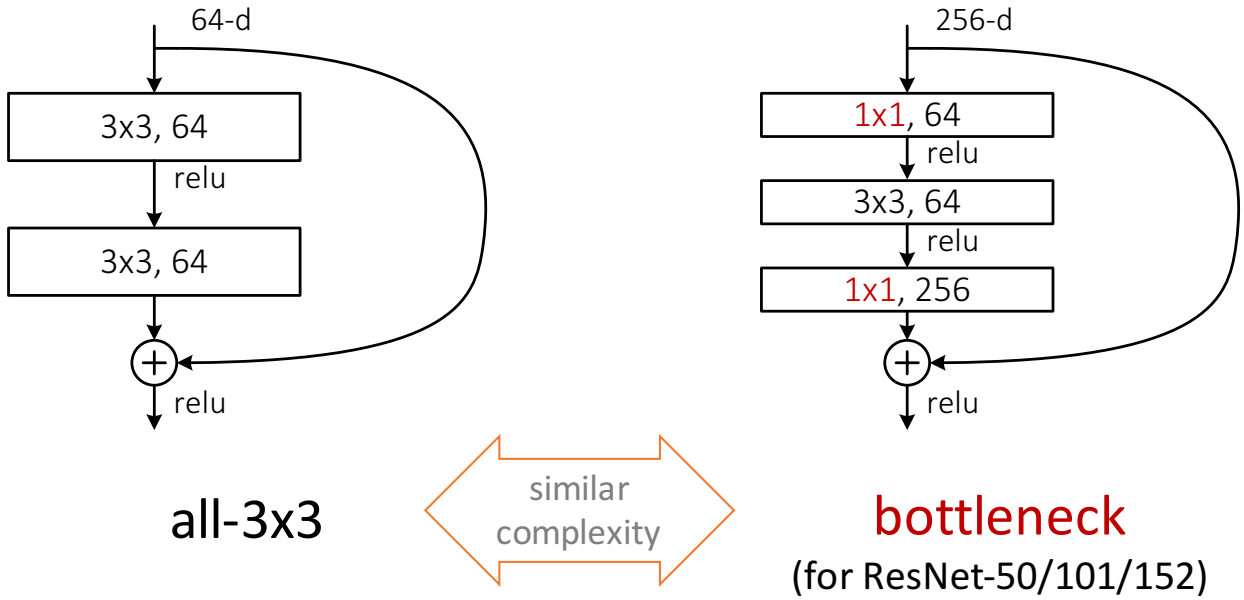


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

- Batch Normalization after every CONV layer.
- Xavier/2 initialization from He et al.
- SGD + Momentum (0.9).
- Learning rate: 0.1, divided by 10 when validation error plateaus.
- Mini-batch size 256.
- Weight decay of 1e-5.
- No dropout used.

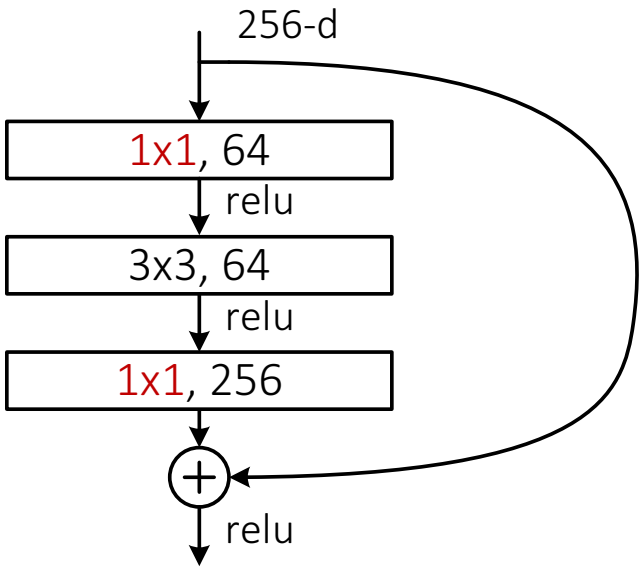
ImageNet experiments

- A practical design of going deeper

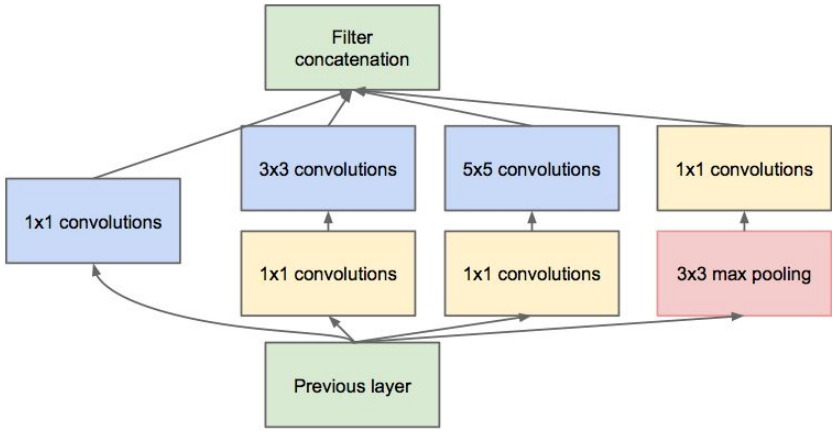


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Same trick as used in GoogLeNet

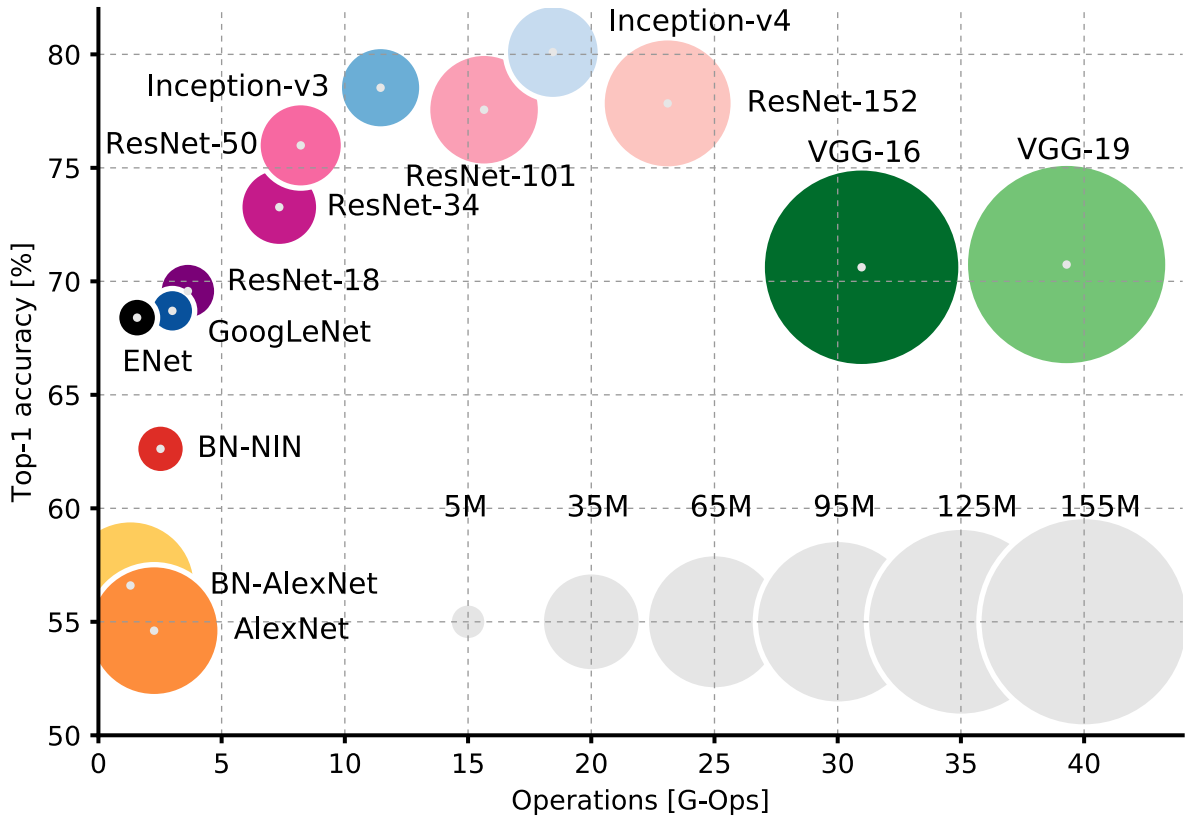


ResNet



GoogLeNet

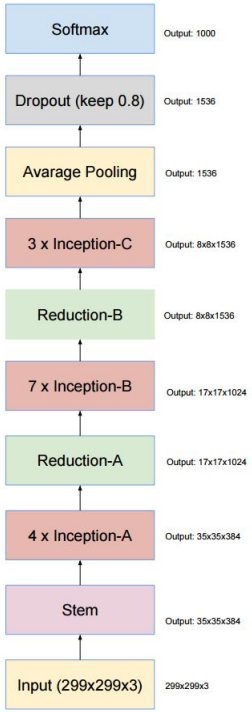
Accuracy on ImageNet Vs network size



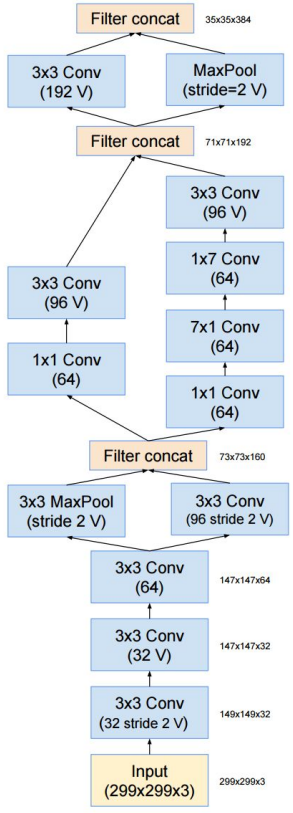
- Top-1 one-crop accuracy Vs amount of operations required for a single forward pass.
- The size of the blobs is proportional to the number of network parameters.

Fig: **An Analysis of Deep Neural Network Models for Practical Applications** by Canziani, Culurciello & Paszke.

Inception-v4



9 layers



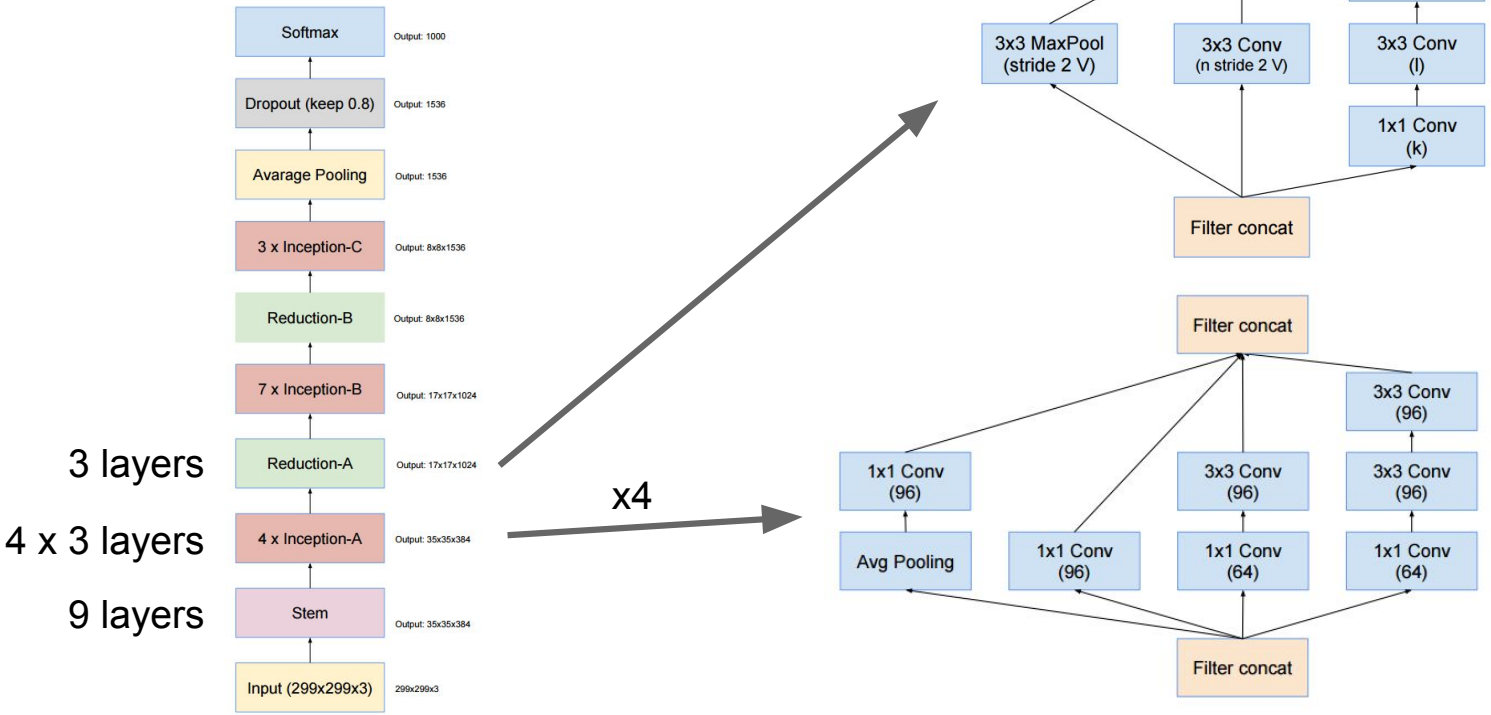
1x7, 7x1 filters

Strided convolution
AND max pooling

V = Valid convolutions
(no padding)

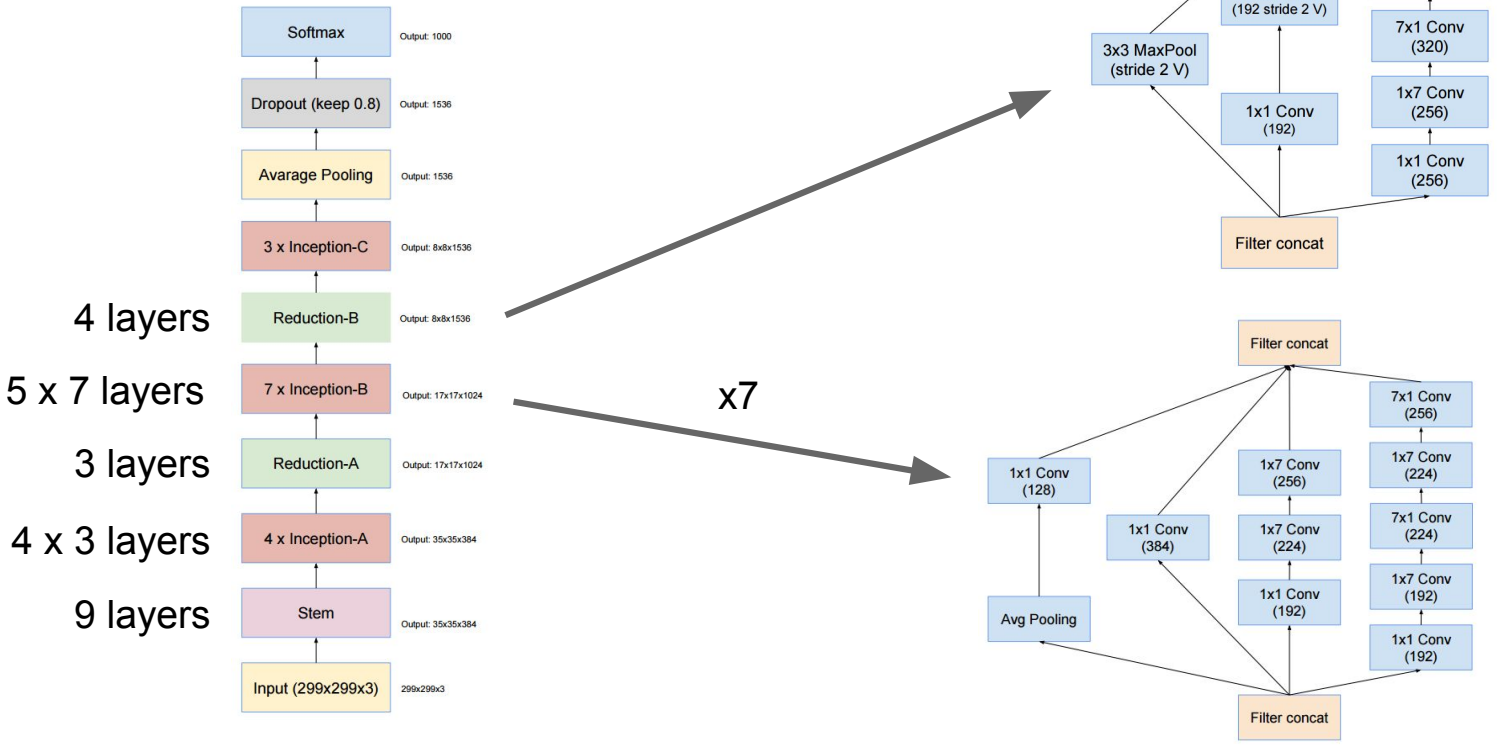
Szegedy et al, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv 2016

Inception-v4



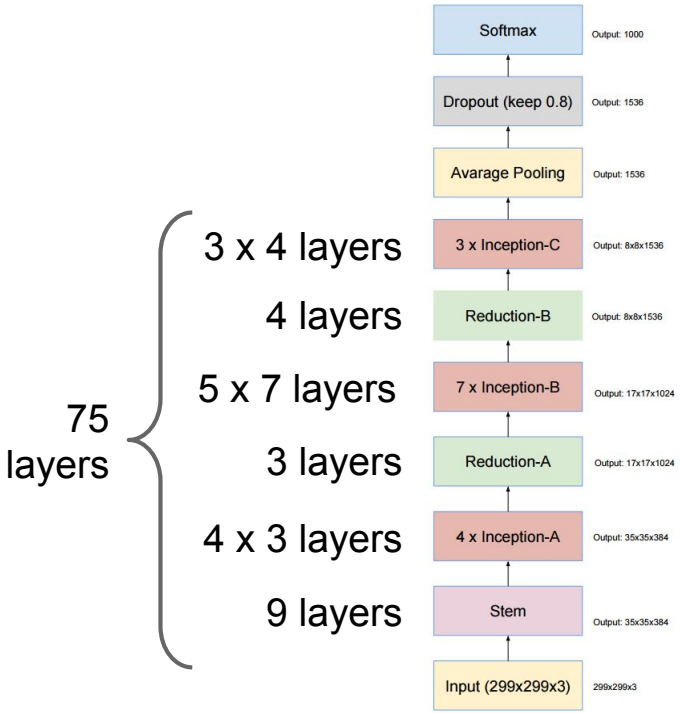
Szegedy et al, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv 2016

Inception-v4

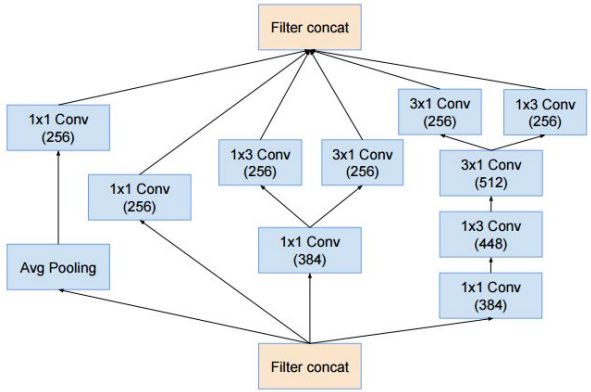


Szegedy et al, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv 2016

Inception-v4

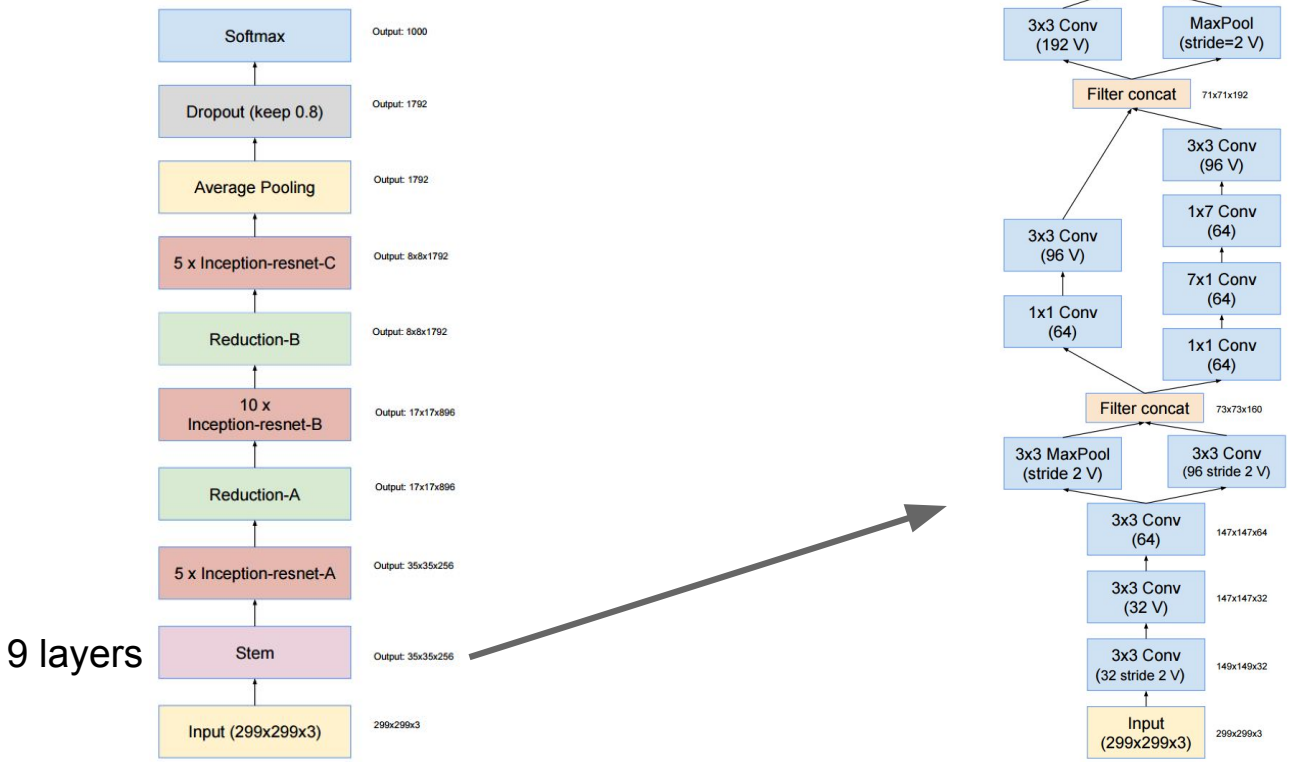


x3

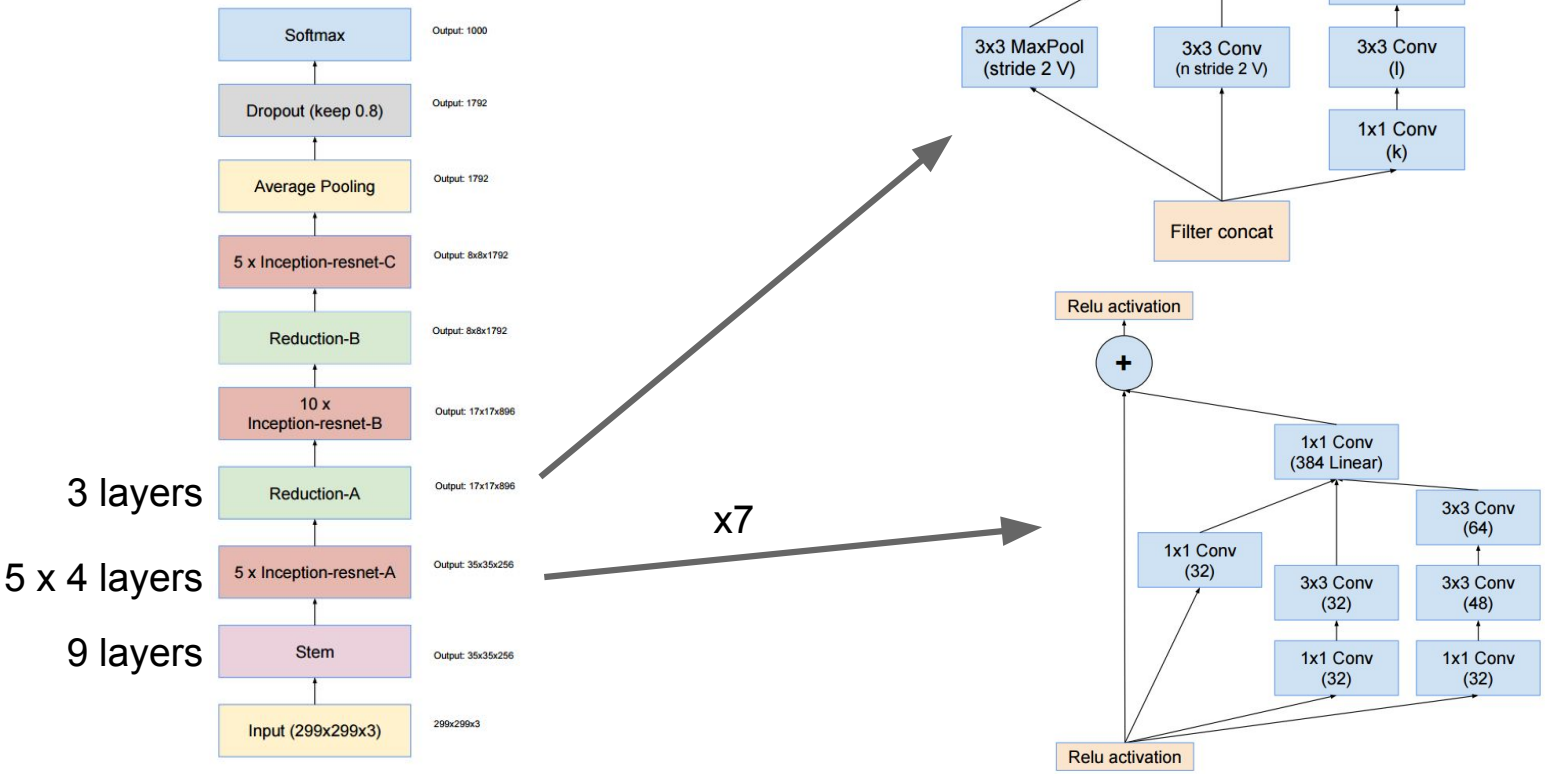


Szegedy et al, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv 2016

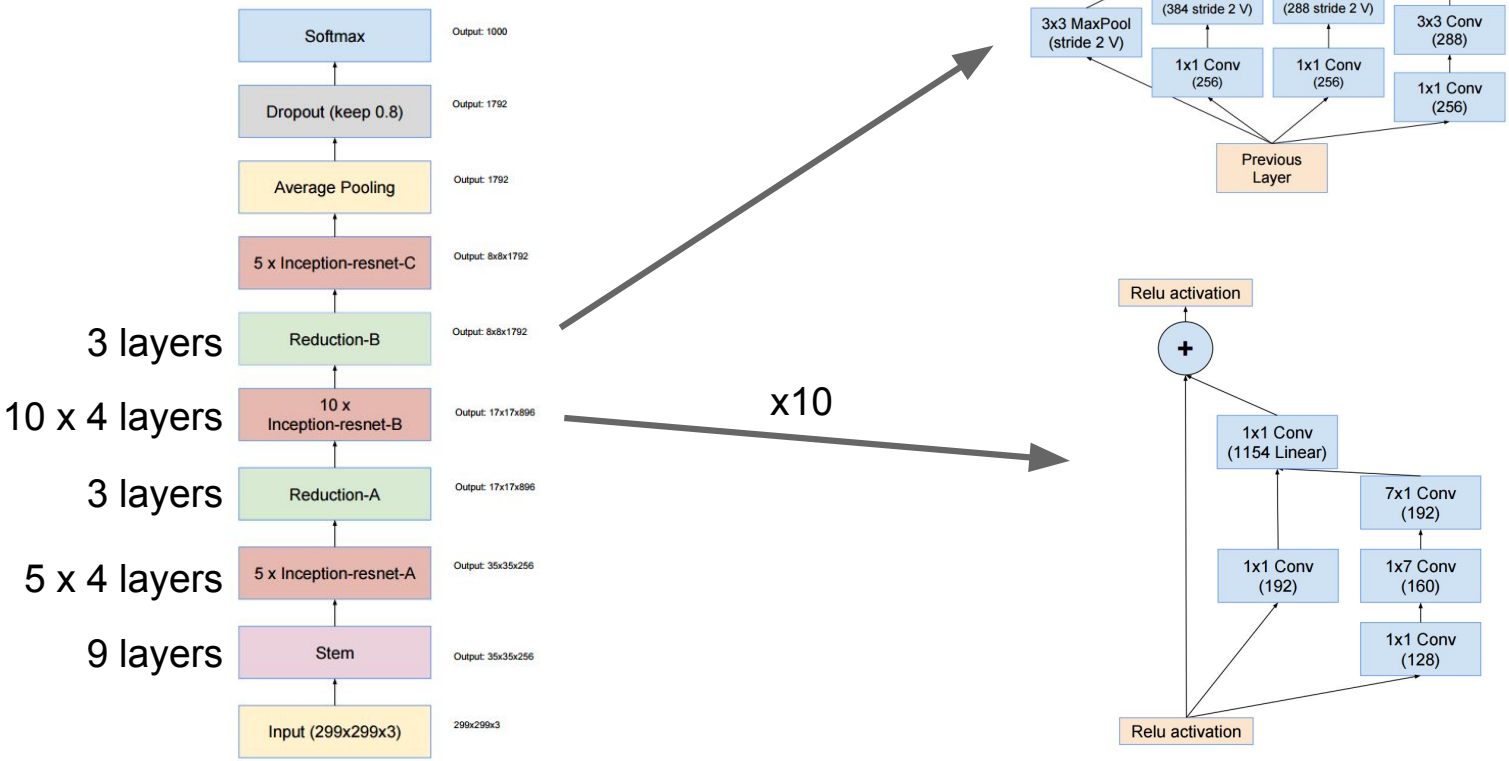
Inception-ResNet-v2



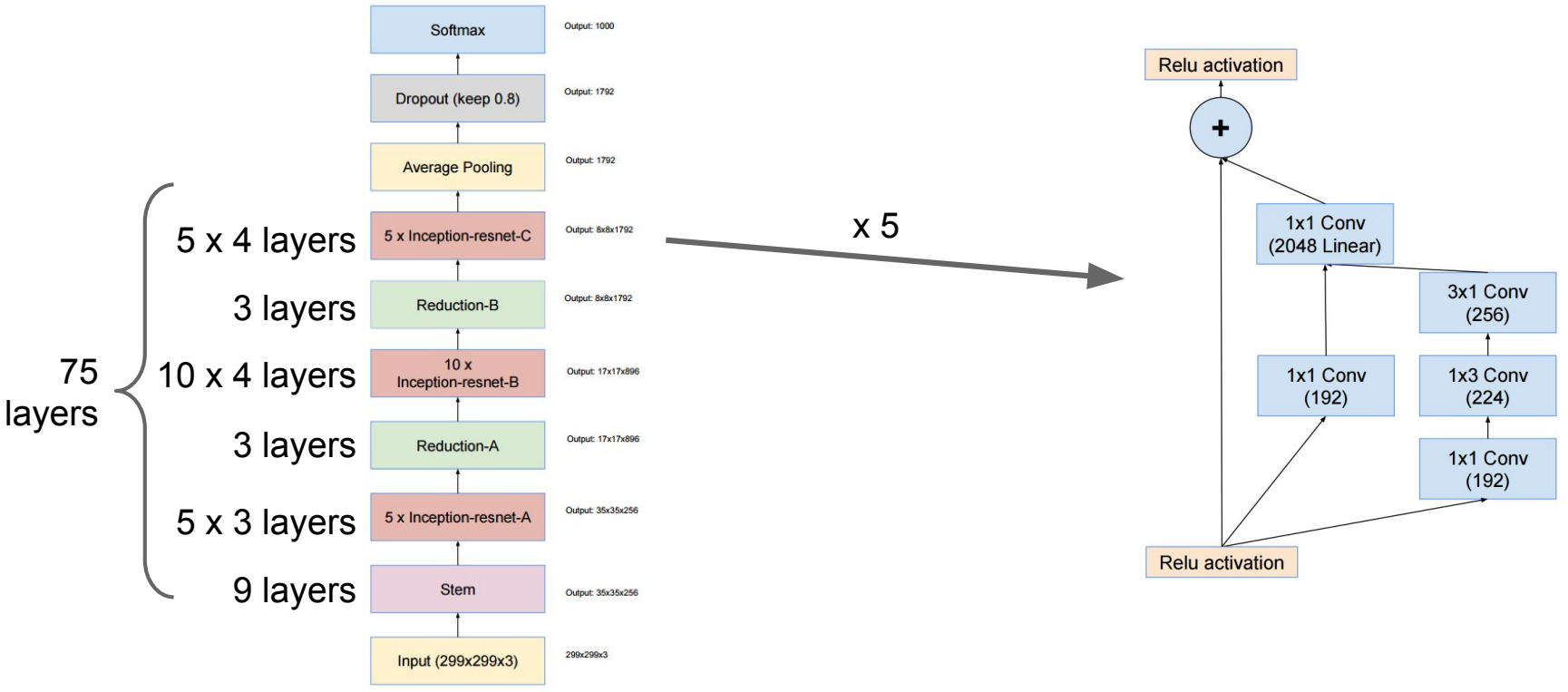
Inception-ResNet-v2



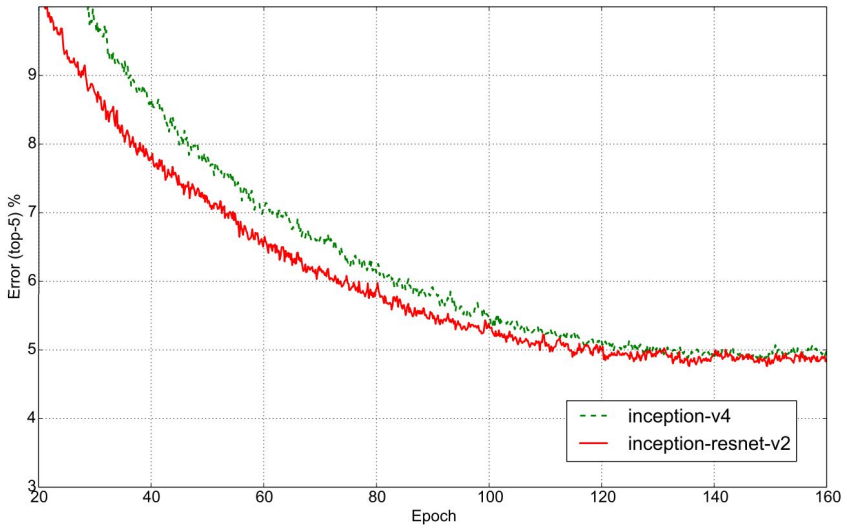
Inception-ResNet-v2



Inception-ResNet-v2



Inception-ResNet-v2



Residual and non-residual converge to similar value, but residual learns faster