Lecture Notes for

Machine Learning in Python

⟦👨‍🏫, 👨‍💻, 🐍, 👨‍🔬⟧

Professor Eric Larson

Convolutional Neural Networks

Logistics and Agenda

- Logistics
  - Wide/Deep due soon!
  - Town Hall, if needed
- Agenda
  - Basic CNN Structures
  - CNN Gradient overview

Wide and Deep Town Hall

"And how does it make you feel when she jumps over you and calls you a lazy dog?"

Class Overview, by topic

Table Data Visualization

Numpy, Pandas, Seaborn
Overviews with some in-depth discussion

Dimension Reduction and Image Processing

Scikit-learn, Scikit Image,
Intuition only, Some mathematics

Linear and Logistic Regression

Numpy, Recreate API for Scikit-learn
Detailed mathematics for simple optimization
intuition for advanced optimization

Neural Networks and Back Prop.

Numpy
Detailed mathematics for NN operations

Wide and Deep Networks

Convolutional Networks

Recurrent Networks

Keras, Tensorflow
Intuition, Detailed implement.

Ethics in Language Models

ConceptNet
Case studies

Convolutional Neural Networks

STOP making fun of different programming languages

C is FAST

Java is POPULAR

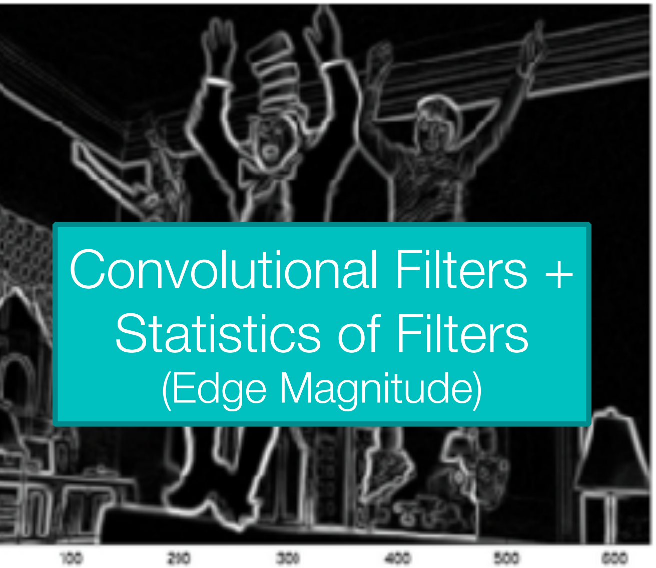
Ruby is COOL

Python is BEAUTIFUL

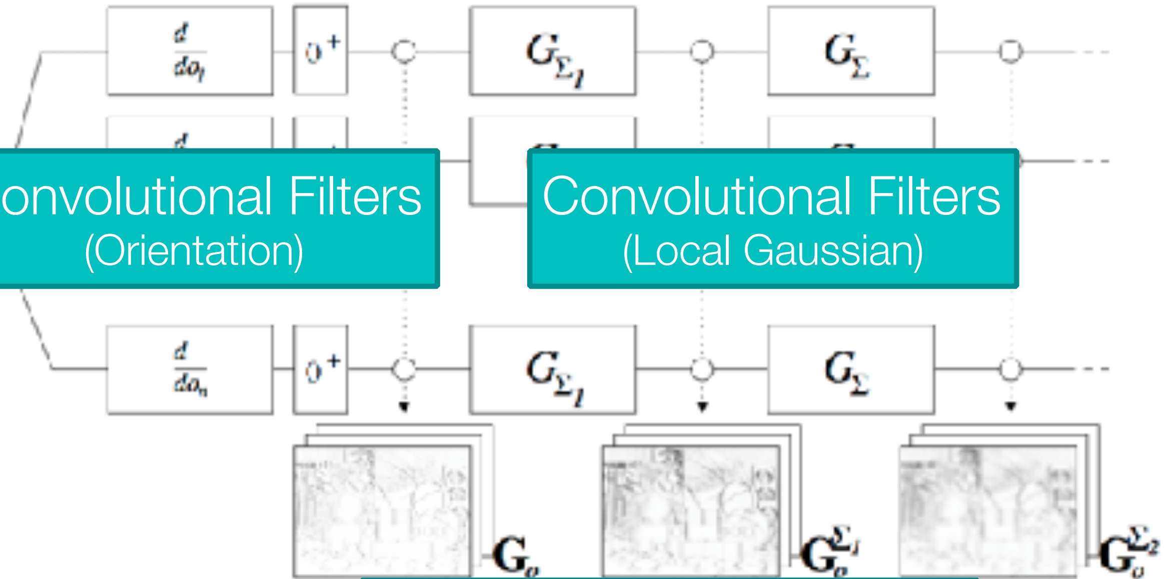
Javascript

Haskell is INTRIGUING

What we did before (Daisy)



Convolutional Filters +
Statistics of Filters
(Edge Magnitude)



Convolutional Filters
(Orientation)

Convolutional Filters
(Local Gaussian)

Statistics of Filter Outputs
(Histograms)

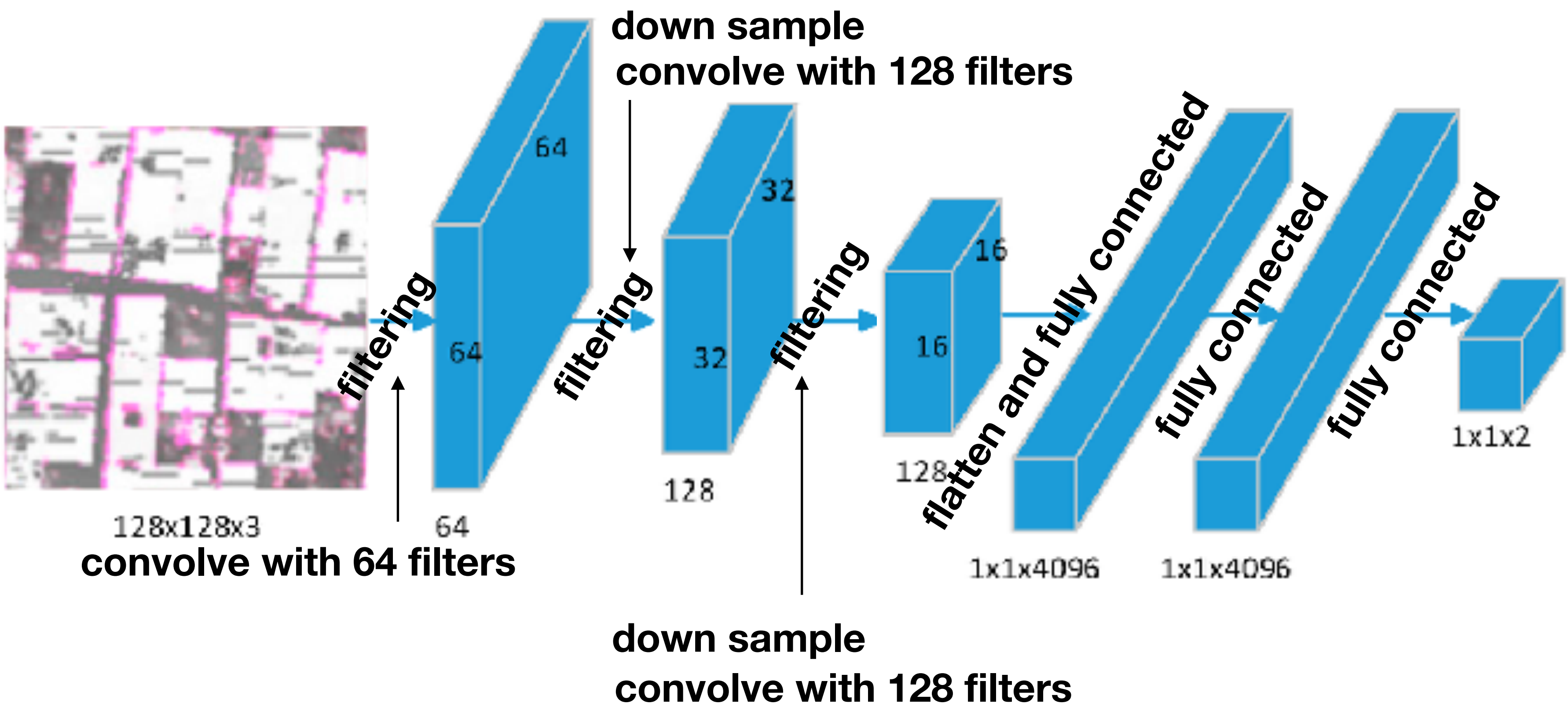
take norma ... nt u,v

$$\widetilde{\mathbf{h}}_\Sigma(u,v) = \left\| \left[\mathbf{G}_1^\Sigma(u,v), \ldots, \mathbf{G}_H^\Sigma(u,v) \right]^\top \right\|$$

$$\begin{aligned}
\mathcal{D}(u_0,v_0) = \\
\Big[\quad & \widetilde{\mathbf{h}}^\top_{\Sigma_1}(u_0,v_0), \\
& \widetilde{\mathbf{h}}^\top_{\Sigma_1}(\mathbf{l}_1(u_0,v_0,R_1)), \cdots, \widetilde{\mathbf{h}}^\top_{\Sigma_1}(\mathbf{l}_T(u_0,v_0,R_1)), \\
& \widetilde{\mathbf{h}}^\top_{\Sigma_2}(\mathbf{l}_1(u_0,v_0,R_2)), \cdots, \widetilde{\mathbf{h}}^\top_{\Sigma_2}(\mathbf{l}_T(u_0,v_0,R_2)),
\end{aligned}$$

Tola et al. “Daisy: An efficient dense descriptor applied to wide- baseline stereo.” Pattern Analysis and Machine Intelligence, IEEE Transactions

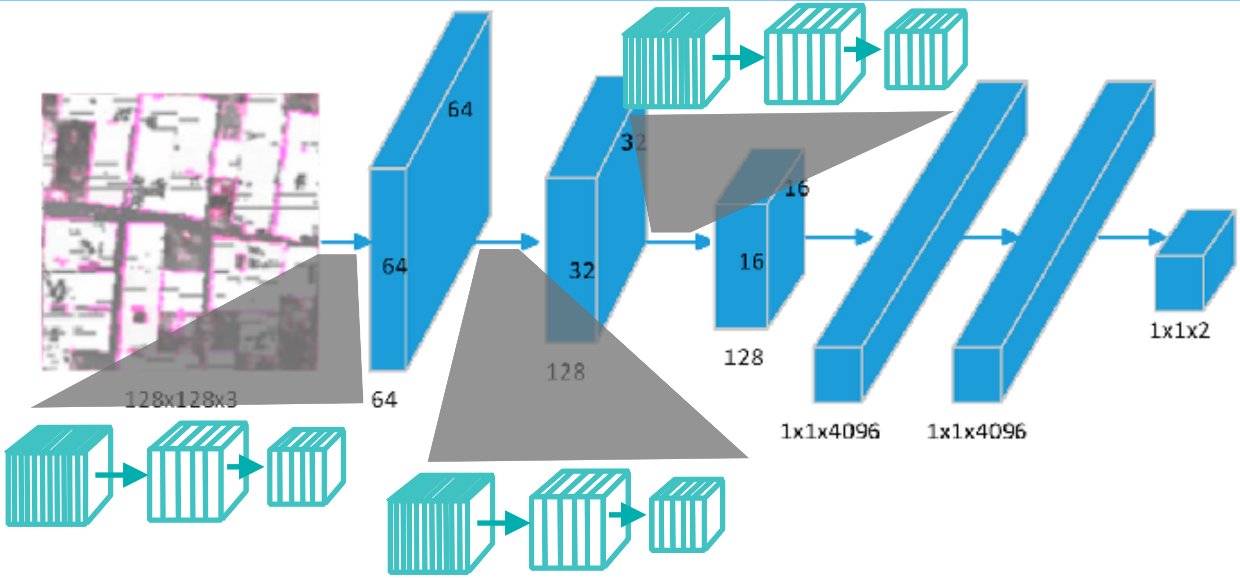
CNN Overview



Blue Tensors: Outputs tensors of Each Layer

Learned Params: Weights in Each Arrow

CNN Overview, per layer processing



- **Conv. layer(s):**
  - filtering
  - activation
  - pooling
    - Each pooling layer *can* make the input image “smaller”
    - less dependence on exact pixel locations
- **Final layers are densely connected**
  - typically multi-layer perceptrons (logistic regression)

Reminder: Convolution

$$\sum \left(\mathbf{I}\left[i \pm \frac{r}{2}, j \pm \frac{c}{2} \right] \odot \mathbf{f} \right) = \mathbf{O}[i,j]$$

output image at pixel $\mathbf{i,j}$

input image at r x c range of pixels centered in i,j

kernel of size, r x c usually r=c

0	0	0	0	0	0	0	0	0
0	1	2	3	4	12	9	8	0
0	5	2	3	4	12	9	8	0
0	5	2	1	4	10	9	8	0
0	7	2	1	4	12	7	8	0
0	7	2	1	4	14	9	8	0
0	5	2	3	4	12	7	8	0
0	5	2	1	4	12	9	8	0
0	0	0	0	0	0	0	0	0

input image, $\mathbf{I}$

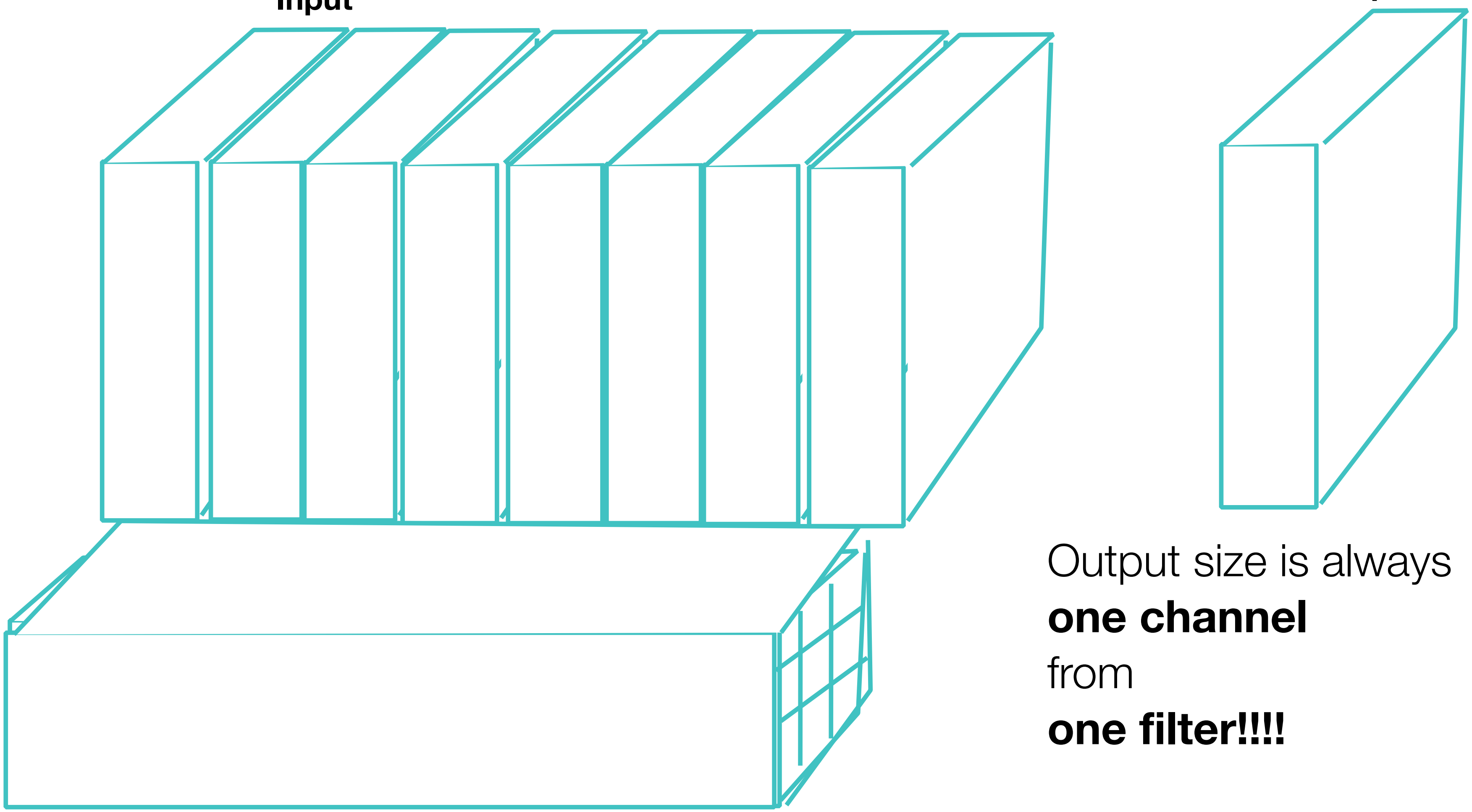
1	2	1
2	4	2
1	2	1

kernel filter, $\mathbf{f}$ 3x3

20	21	36	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…

output image, $\mathbf{O}$

Convolution in a CNN

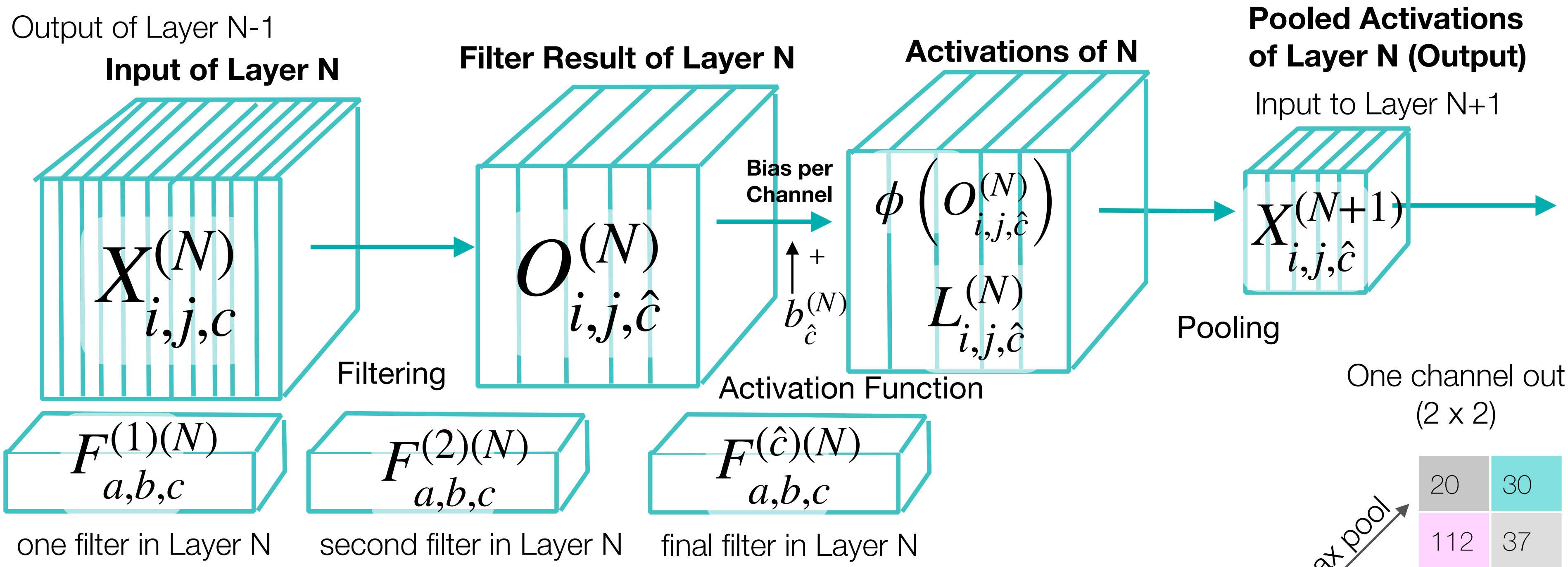


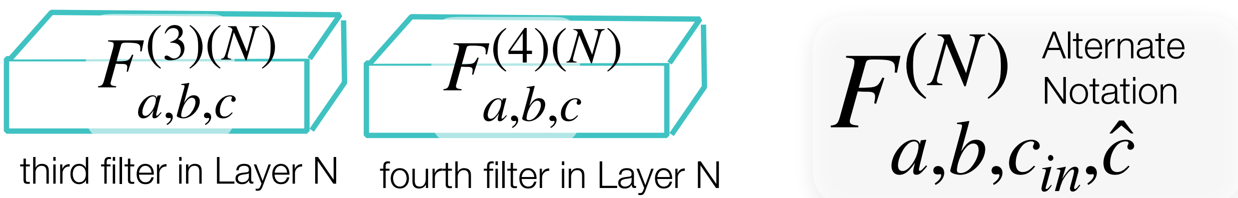
Output size is always
one channel
from
one filter!!!!

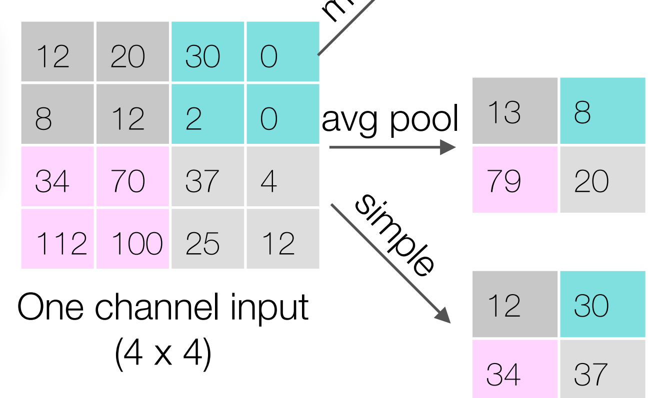
3x3 Filter (Kernel) one filter in Layer N

Convolutional Layers

Structure of Each Tensor:
Rows x Columns x Channels







Self Test: What are the learned parameters?

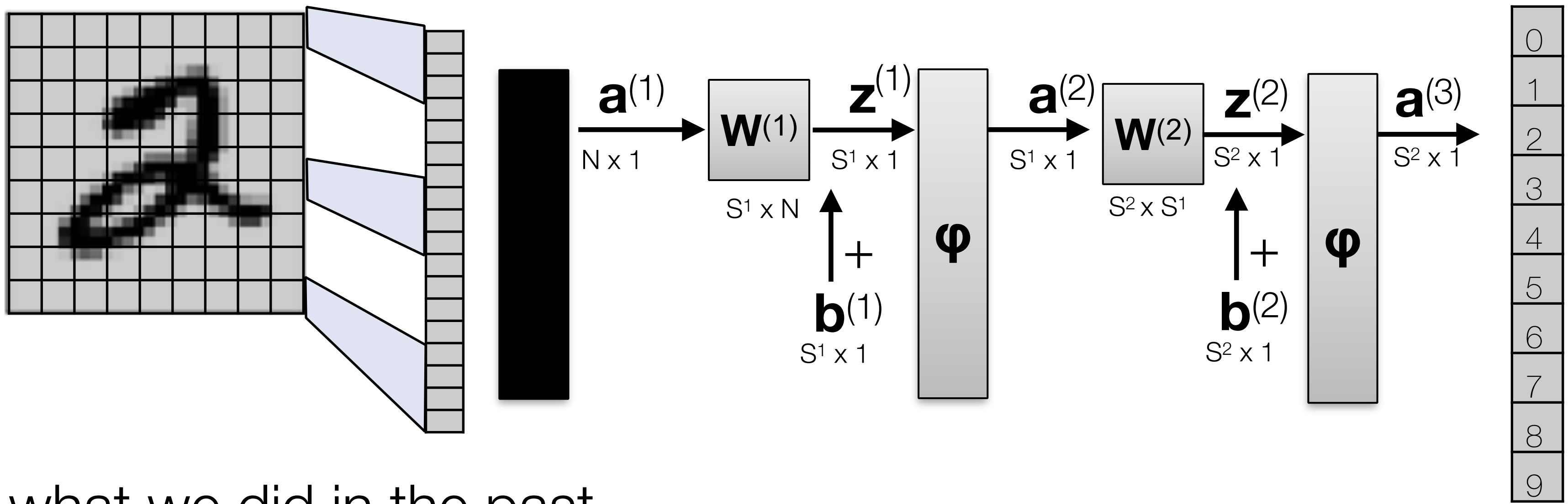
A. $O^{(N)}$

B. $L^{(N)}$

C. $F^{(N)}$ and $b^{(N)}$

D. All of these

Simple Example: From Fully Connected to CNN



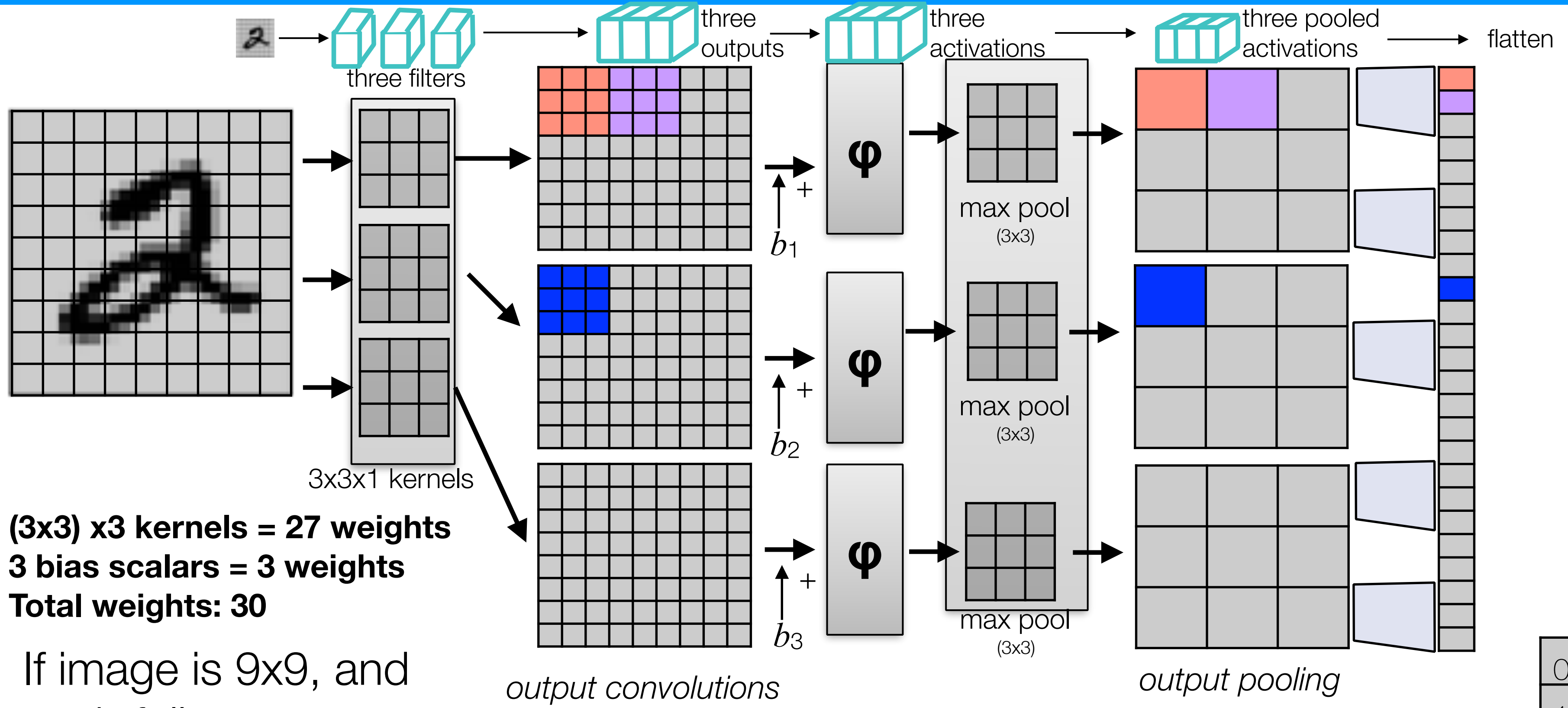
what we did in the past

If image is 9x9, and each fully connected layer is 20 hidden neurons wide, how many parameters are in this NN?

for 9x9, $\underbrace{9^2\text{x}20 + 20}_{\mathbf{W}^{(1)} \quad \mathbf{b}^{(1)}} + \underbrace{(20\text{x}10) + 10}_{\mathbf{W}^{(2)} \quad \mathbf{b}^{(2)}} =$ 1,850 parameters

first layer second layer

Simple Example: From Fully Connected to CNN



(3x3) x3 kernels = 27 weights
3 bias scalars = 3 weights
Total weights: 30

If image is 9x9, and each fully connected layer is 20 hidden neurons wide, how many parameters are in this NN?

$$30 + \underbrace{(27 \times 20) + 20}_{\mathbf{W}^{(1)}\ \ \mathbf{b}^{(1)}} + \underbrace{(20 \times 10) + 10}_{\mathbf{W}^{(2)}\ \ \mathbf{b}^{(2)}} = 800$$

first conv layer

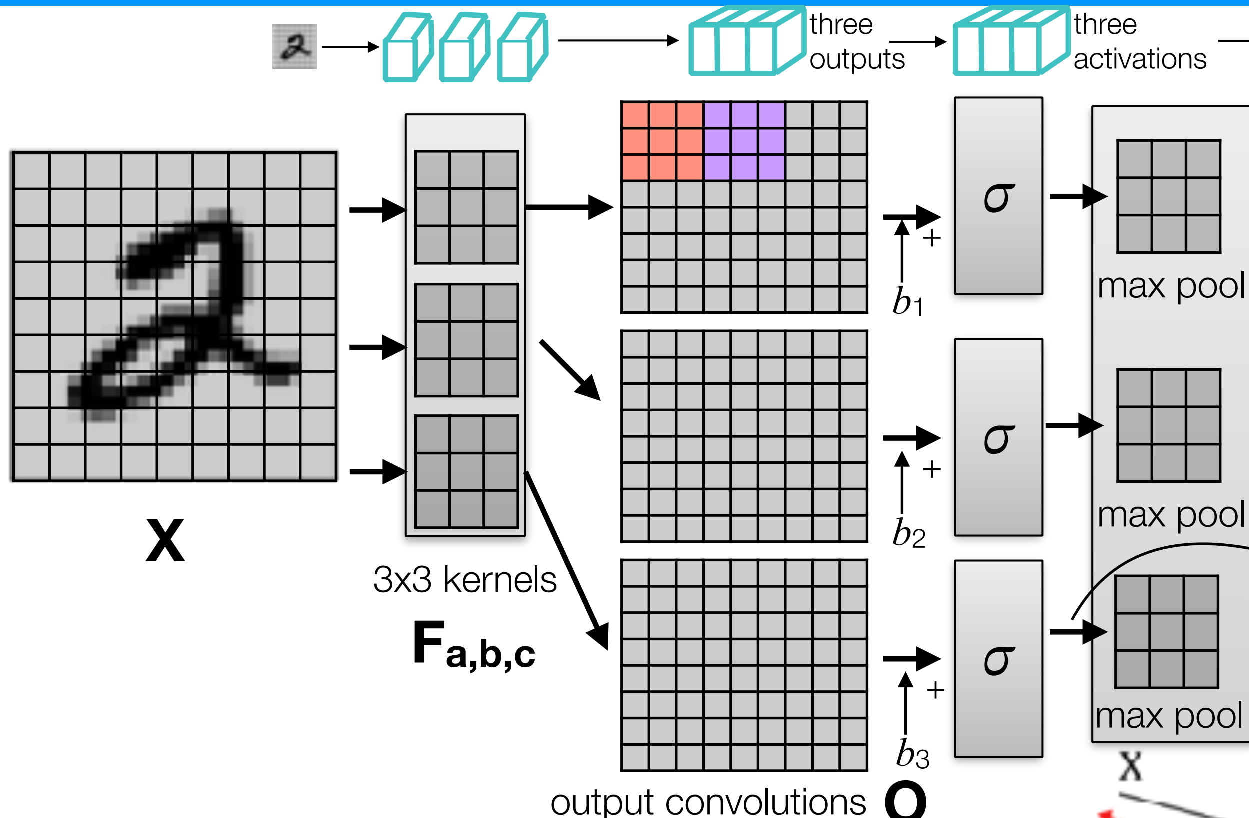
first dense layer

second dense layer

$\mathbf{b}^{(1)}$ $S^1 \times 1$

$\mathbf{b}^{(2)}$ $S^2 \times 1$

CNN gradient setup



Derivative of max pool is easy to compute:

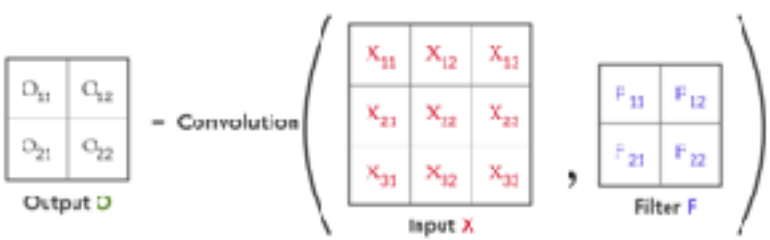
$$\frac{\partial}{\partial L_i}\text{pool}(L_i) = \begin{cases} 1, & \text{if } L_i \text{ is max} \\ 0, & \text{else} \end{cases}$$

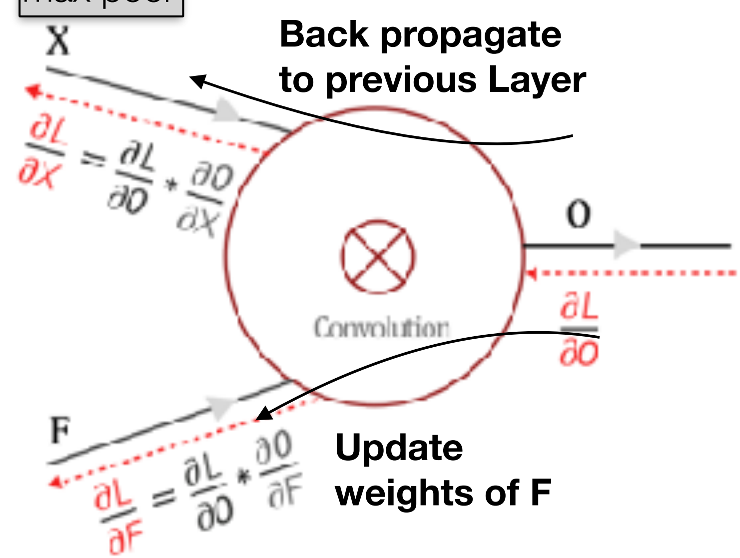
output activations

$\mathbf{L} = \sigma(\mathbf{O})$

$\mathbf{X}^{(\text{Next})} = \text{pool}(\ \sigma(\mathbf{O})\ )$

Derivative of convolution is more involved:



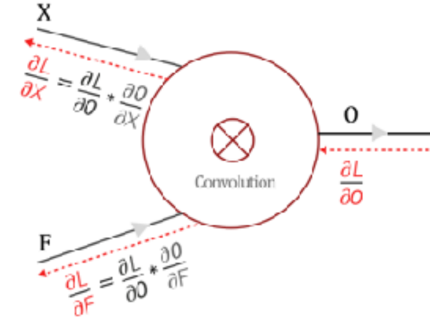


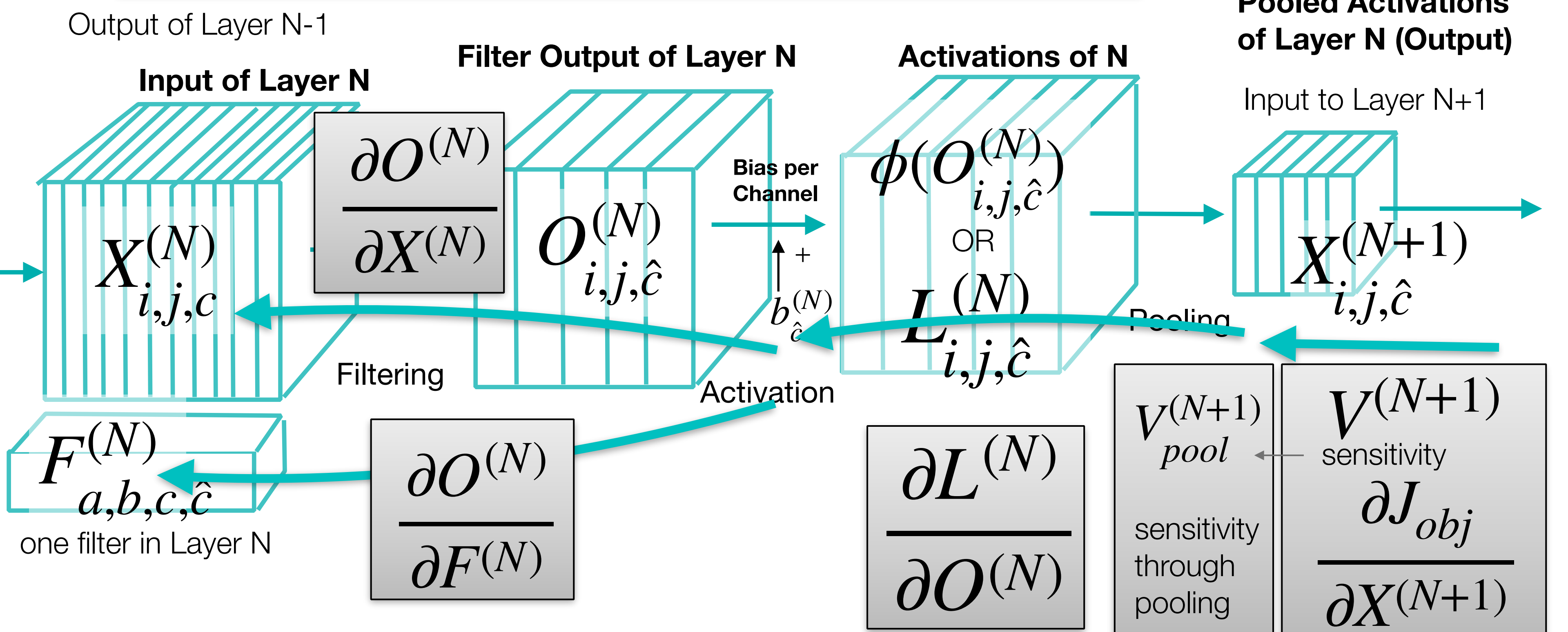
https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c

CNNs Back Propagation

Sensitivity to layer in back propagation

$$V^{(N)} = \frac{\partial O^{(N)}}{\partial X^{(N)}} \cdot \frac{\partial L^{(N)}}{\partial O^{(N)}} \cdot V_{pool}^{(N+1)} = \frac{\partial J_{obj}}{\partial X^{(N)}}$$





Now we can calc partial derivative

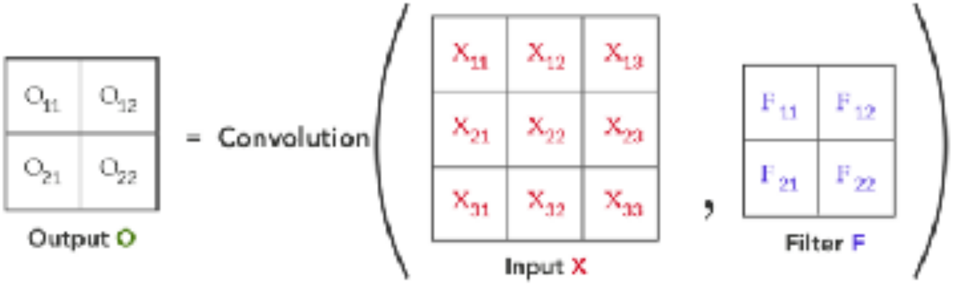
$$\frac{\partial L^{(N)}}{\partial F^{(N)}} = \frac{\partial O^{(N)}}{\partial F^{(N)}} \cdot \frac{\partial L^{(N)}}{\partial O^{(N)}}$$

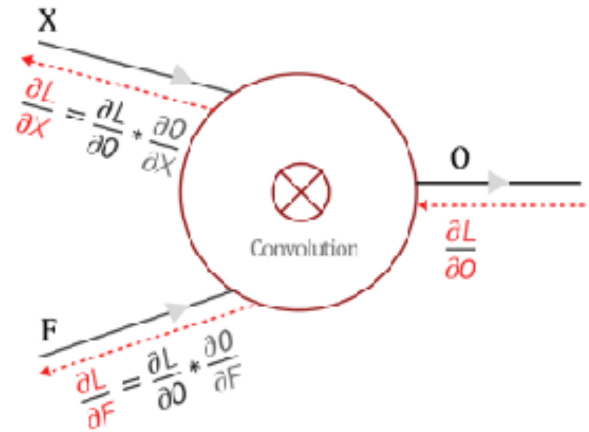
sor: (

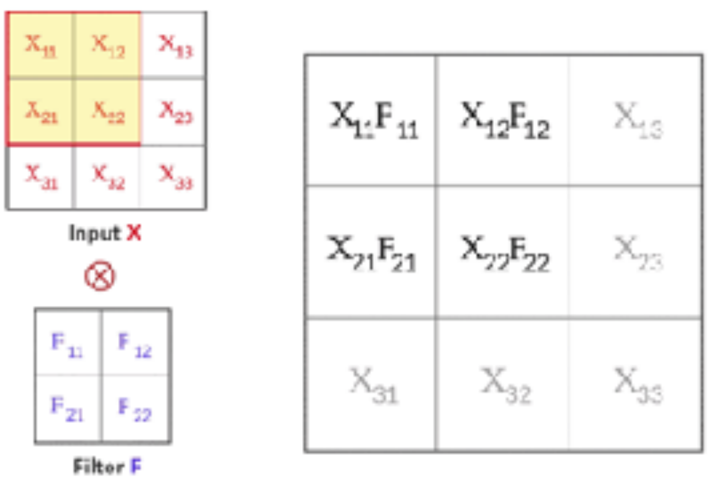
Just incorporate sensitivity, to get weight update

$$\frac{\partial J_{obj}}{\partial F^{(N)}} = \frac{\partial O^{(N)}}{\partial F^{(N)}} \cdot \frac{\partial L^{(N)}}{\partial O^{(N)}} \cdot V_{pool}^{(N+1)}$$

Breaking Apart Convolution Operations







$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

$$O_{11} = X_{11} \cdot F_{11} + X_{12} \cdot F_{12} + X_{21} \cdot F_{21} + X_{22} \cdot F_{22}$$

$$O_{12} = X_{12} \cdot F_{11} + X_{13} \cdot F_{12} + X_{22} \cdot F_{21} + X_{23} \cdot F_{22}$$

$$O_{21} = X_{21} \cdot F_{11} + X_{22} \cdot F_{12} + X_{31} \cdot F_{21} + X_{32} \cdot F_{22}$$

$$O_{22} = X_{22} \cdot F_{11} + X_{23} \cdot F_{12} + X_{32} \cdot F_{21} + X_{33} \cdot F_{22}$$

$$O_{**} = X_{**} \cdot F_{11} + X_{**} \cdot F_{12} + X_{**} \cdot F_{21} + X_{**} \cdot F_{22}$$

Filter is consistent on columns, input increases indices

https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c

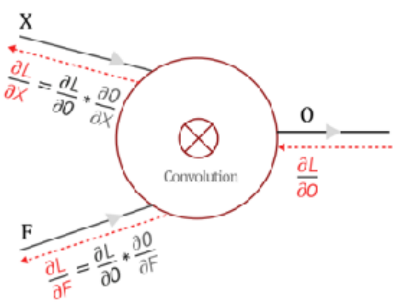
Gradient of Convolution

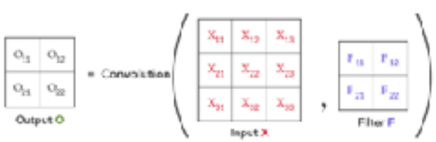
$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial X}$$

for sensitivity

$$\boxed{\frac{\partial L}{\partial F} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial F}}$$

for weight updates





$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$

$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22}$

$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22}$

$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22}$

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

Finding derivatives with respect to F_{11}, F_{12}, F_{21} and F_{22}

$$\frac{\partial O_{11}}{\partial F_{11}} = X_{11} \quad \frac{\partial O_{11}}{\partial F_{12}} = X_{12} \quad \frac{\partial O_{11}}{\partial F_{21}} = X_{21} \quad \frac{\partial O_{11}}{\partial F_{22}} = X_{22}$$

derivative of every O_{ij} w.r.t. F_{11}

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{22}}$$

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * X_{11} + \frac{\partial L}{\partial O_{12}} * X_{12} + \frac{\partial L}{\partial O_{21}} * X_{21} + \frac{\partial L}{\partial O_{22}} * X_{22}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * X_{12} + \frac{\partial L}{\partial O_{12}} * X_{13} + \frac{\partial L}{\partial O_{21}} * X_{22} + \frac{\partial L}{\partial O_{22}} * X_{23}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * X_{21} + \frac{\partial L}{\partial O_{12}} * X_{22} + \frac{\partial L}{\partial O_{21}} * X_{31} + \frac{\partial L}{\partial O_{22}} * X_{32}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * X_{22} + \frac{\partial L}{\partial O_{12}} * X_{23} + \frac{\partial L}{\partial O_{21}} * X_{32} + \frac{\partial L}{\partial O_{22}} * X_{33}$$

$$\begin{bmatrix} \frac{\partial L}{\partial F_{11}} & \frac{\partial L}{\partial F_{12}} \\ \frac{\partial L}{\partial F_{21}} & \frac{\partial L}{\partial F_{22}} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

Filter updates

Input

Derivative
From activation!

https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c

Gradient of Convolution

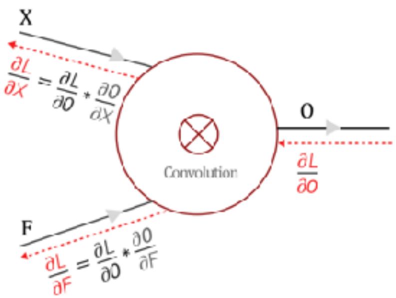
$$\frac{\partial L}{\partial X_{11}} = \frac{\partial L}{\partial O_{11}}\frac{\partial O_{11}}{\partial X_{11}} + \frac{\partial L}{\partial O_{12}}\cancel{\frac{\partial O_{12}}{\partial X_{11}}} + \frac{\partial L}{\partial O_{21}}\cancel{\frac{\partial O_{21}}{\partial X_{11}}} + \frac{\partial L}{\partial O_{22}}\cancel{\frac{\partial O_{22}}{\partial X_{11}}}$$

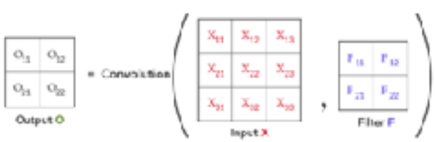
$$\boxed{\frac{\partial L}{\partial X} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial X}}$$

for sensitivity

$$\frac{\partial L}{\partial F} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial F}$$

for weight updates





$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$

$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22}$

$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22}$

$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22}$

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

Differentiating with respect to X_{11}, X_{12}, X_{21} *and* X_{22}

$$\frac{\partial O_{11}}{\partial X_{11}} = F_{11} \quad \frac{\partial O_{11}}{\partial X_{12}} = F_{12} \quad \frac{\partial O_{11}}{\partial X_{21}} = F_{21} \quad \frac{\partial O_{11}}{\partial X_{22}} = F_{22}$$

Similarly, we can find local gradients for O_{12}, O_{21} *and* O_{22}

$$\frac{\partial L}{\partial X_{11}} = \frac{\partial L}{\partial O_{11}} * F_{11}$$

$$\frac{\partial L}{\partial X_{12}} = \frac{\partial L}{\partial O_{11}} * F_{12} + \frac{\partial L}{\partial O_{12}} * F_{11}$$

$$\frac{\partial L}{\partial X_{13}} = \frac{\partial L}{\partial O_{12}} * F_{12}$$

$$\frac{\partial L}{\partial X_{21}} = \frac{\partial L}{\partial O_{11}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{11}$$

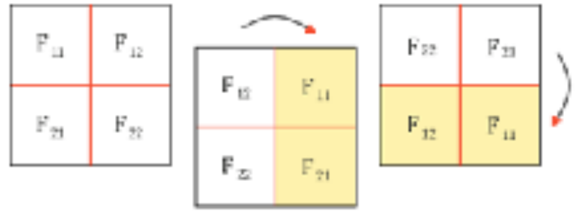
$$\frac{\partial L}{\partial X_{22}} = \frac{\partial L}{\partial O_{11}} * F_{22} + \frac{\partial L}{\partial O_{12}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{12} + \frac{\partial L}{\partial O_{22}} * F_{11}$$

$$\frac{\partial L}{\partial X_{23}} = \frac{\partial L}{\partial O_{12}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{12}$$

$$\frac{\partial L}{\partial X_{31}} = \frac{\partial L}{\partial O_{21}} * F_{21}$$

$$\frac{\partial L}{\partial X_{32}} = \frac{\partial L}{\partial O_{21}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{21}$$

$$\frac{\partial L}{\partial X_{33}} = \frac{\partial L}{\partial O_{22}} * F_{22}$$



$$\begin{bmatrix} \frac{\partial L}{\partial X_{11}} & \frac{\partial L}{\partial X_{12}} & \frac{\partial L}{\partial X_{13}} \\ \frac{\partial L}{\partial X_{21}} & \frac{\partial L}{\partial X_{22}} & \frac{\partial L}{\partial X_{23}} \\ \frac{\partial L}{\partial X_{31}} & \frac{\partial L}{\partial X_{32}} & \frac{\partial L}{\partial X_{33}} \end{bmatrix} = \text{Full Convolution}\left(\begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

New sensitivity

Rotated Filter

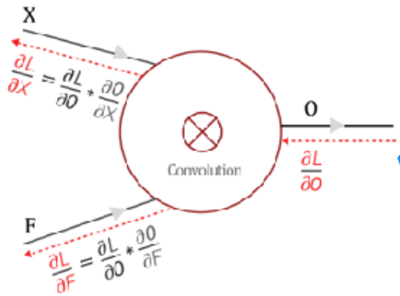
Derivative From activation! (zero padded)

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} & 0 \\ 0 & \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c

Summary

Filters at layer L-1



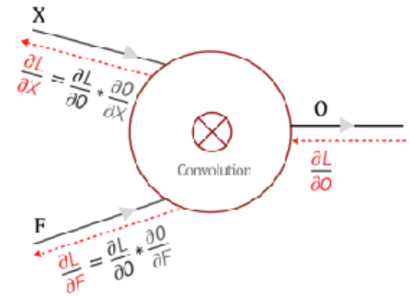
$$\begin{bmatrix} \frac{\partial L}{\partial X_{11}} & \frac{\partial L}{\partial X_{12}} & \frac{\partial L}{\partial X_{13}} \\ \frac{\partial L}{\partial X_{21}} & \frac{\partial L}{\partial X_{22}} & \frac{\partial L}{\partial X_{23}} \\ \frac{\partial L}{\partial X_{31}} & \frac{\partial L}{\partial X_{32}} & \frac{\partial L}{\partial X_{33}} \end{bmatrix} = \text{Full Convolution}\left(\begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

New sensitivity | Rotated Filter | Activation Derivative

$$\begin{bmatrix} \frac{\partial L}{\partial F_{11}} & \frac{\partial L}{\partial F_{12}} \\ \frac{\partial L}{\partial F_{21}} & \frac{\partial L}{\partial F_{22}} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

Filter updates | Input | Activation Derivative

Filters at layer L



$$\begin{bmatrix} \frac{\partial L}{\partial X_{11}} & \frac{\partial L}{\partial X_{12}} & \frac{\partial L}{\partial X_{13}} \\ \frac{\partial L}{\partial X_{21}} & \frac{\partial L}{\partial X_{22}} & \frac{\partial L}{\partial X_{23}} \\ \frac{\partial L}{\partial X_{31}} & \frac{\partial L}{\partial X_{32}} & \frac{\partial L}{\partial X_{33}} \end{bmatrix} = \text{Full Convolution}\left(\begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

New sensitivity | Rotated Filter | Activation Derivative

$$\begin{bmatrix} \frac{\partial L}{\partial F_{11}} & \frac{\partial L}{\partial F_{12}} \\ \frac{\partial L}{\partial F_{21}} & \frac{\partial L}{\partial F_{22}} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

Filter updates | Input | Activation Derivative

CNN Gradient Takeaways

- Derivative of a convolutional layer is calculated through two additional convolutions
  - One for filter updates
  - One for calculating a new sensitivity
- We need to run convolution fast in order to speed up both:
  - feedforward operations (inference and training)
  - back propagation (training)
- Another great resource:
  - https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199

Next Lecture

- More CNN architectures and CNN history