Lecture Notes for

Machine Learning in Python

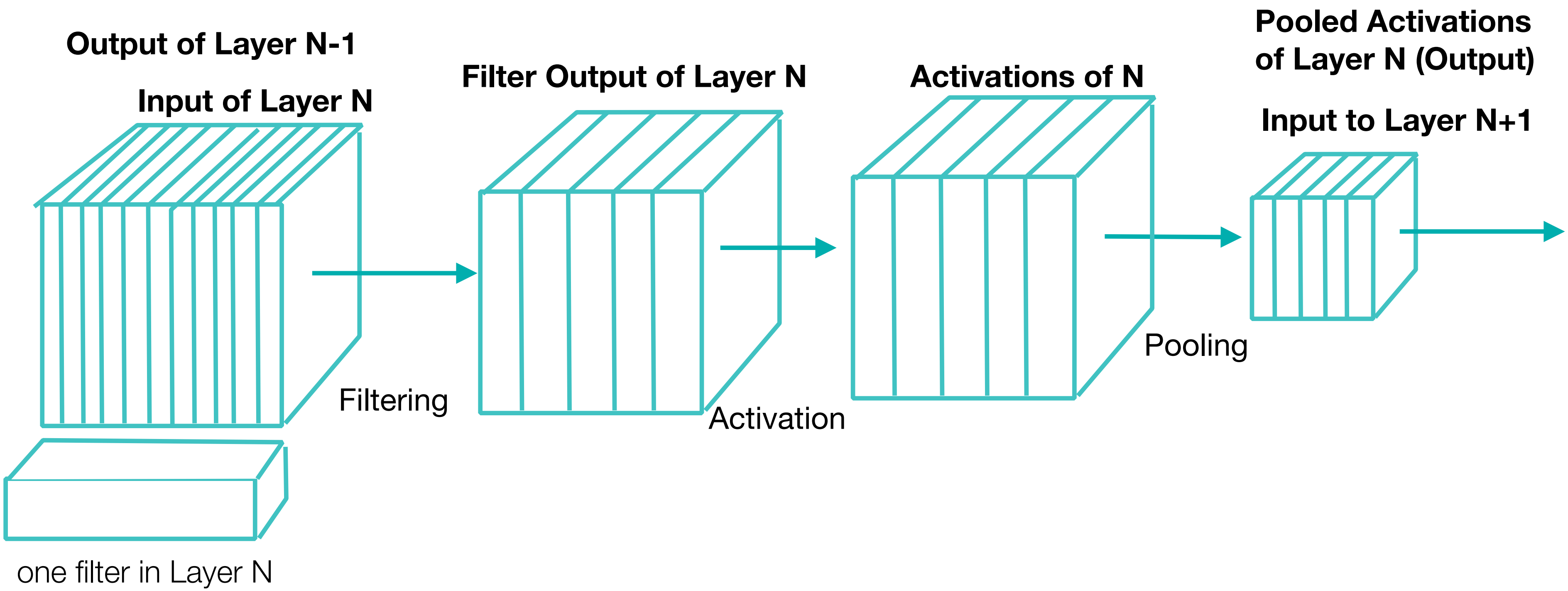
Professor Eric Larson

An Ongoing History of Convolutional Networks

Class logistics and Agenda

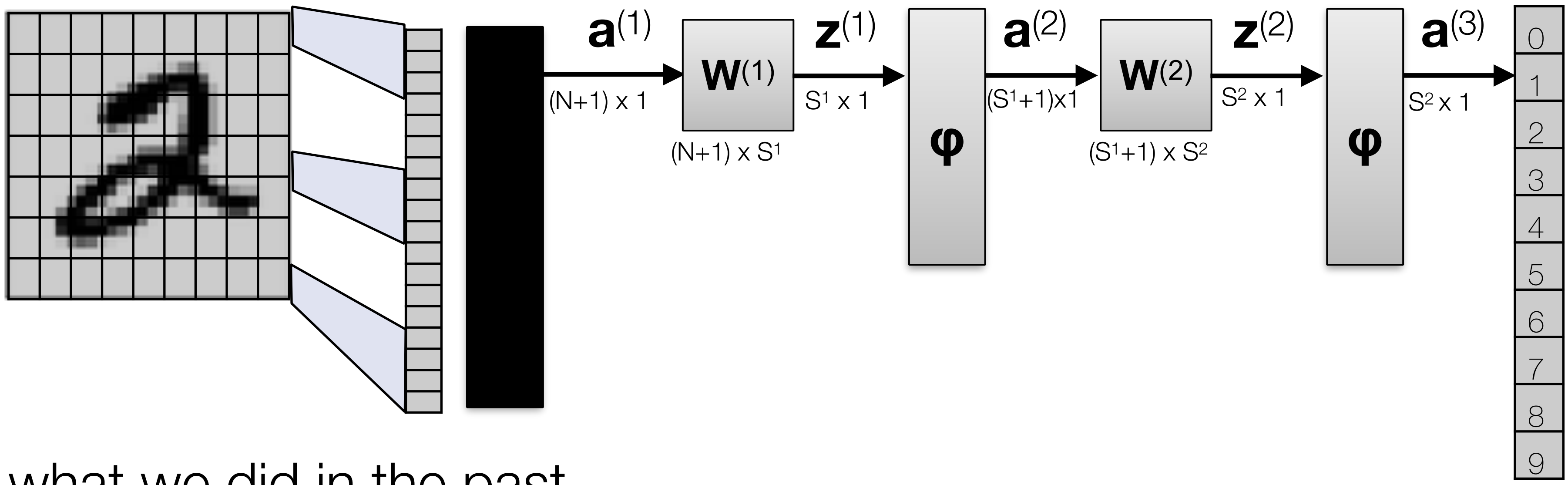
- Wide/Deep Lab due soon!
- Agenda:
  - Finish CNN Discussion
  - CNN Demo
  - History of CNNs
    - with Modern CNN Architectures
- Next Time:
  - More Advanced CNN Demo

Last Time: CNNs, Putting it together



Structure of Each Tensor: Channels x Rows x Columns

Simple Example: From Fully Connected to CNN



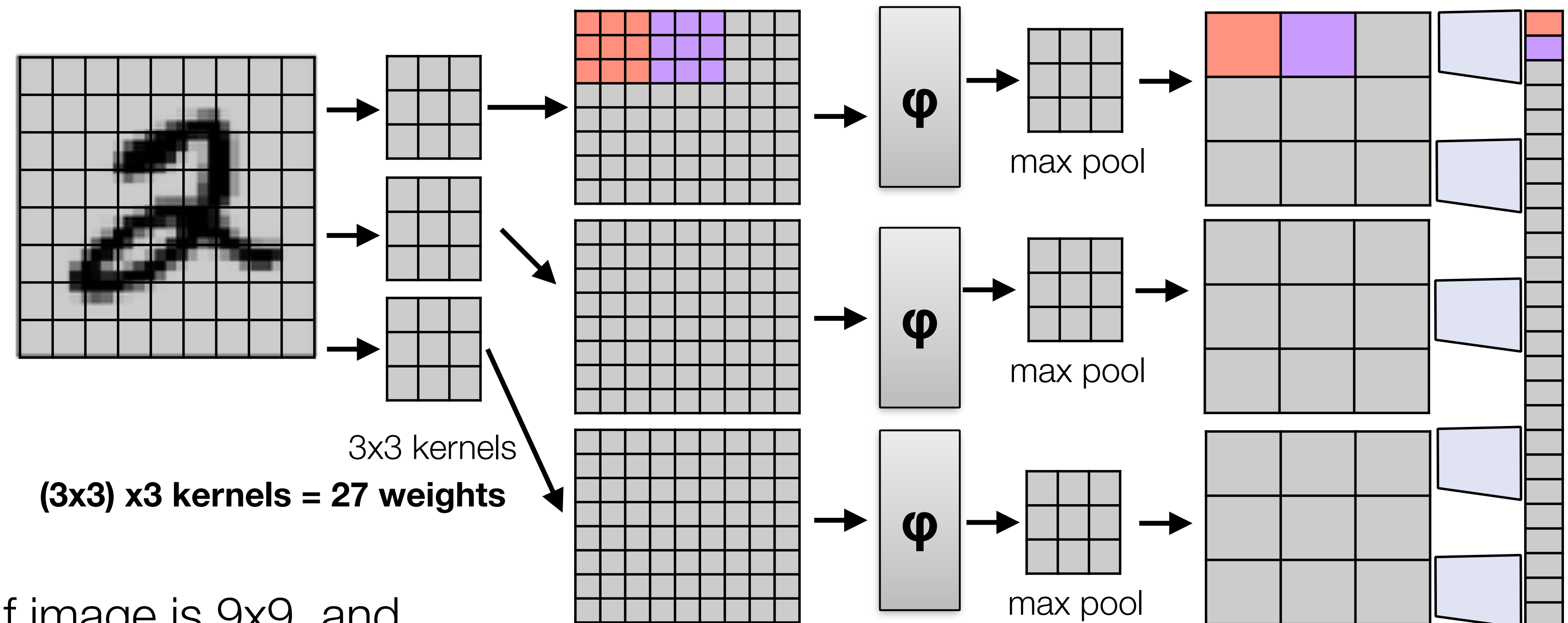
what we did in the past

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

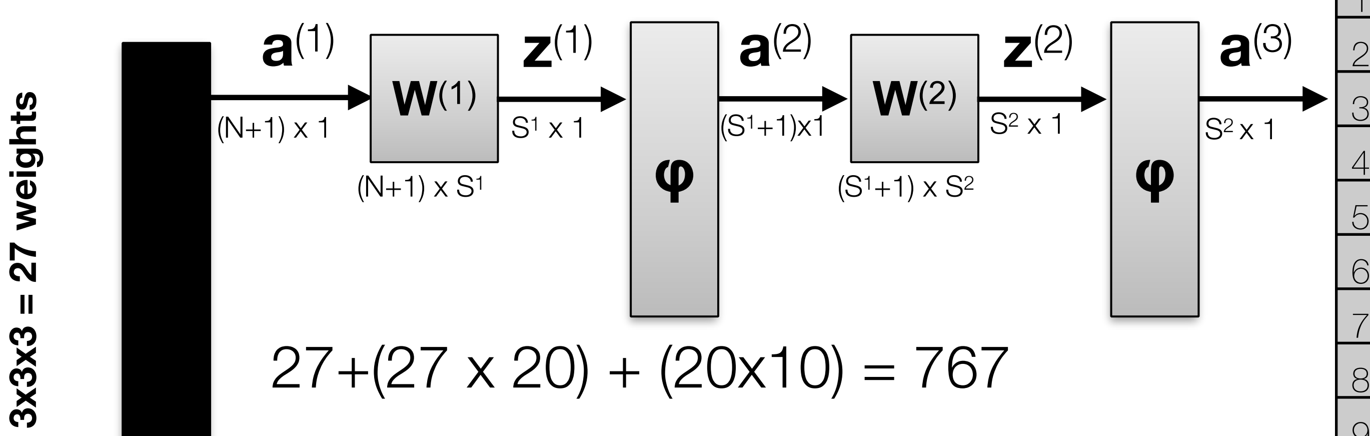
$(K^2 \times 20) + (20 \times 10) = 200 + 20\,K^2$

for 9x9 $= 200 + 20 \times 9^2 =$ 1,820 parameters

Simple Example: From Fully Connected to CNN

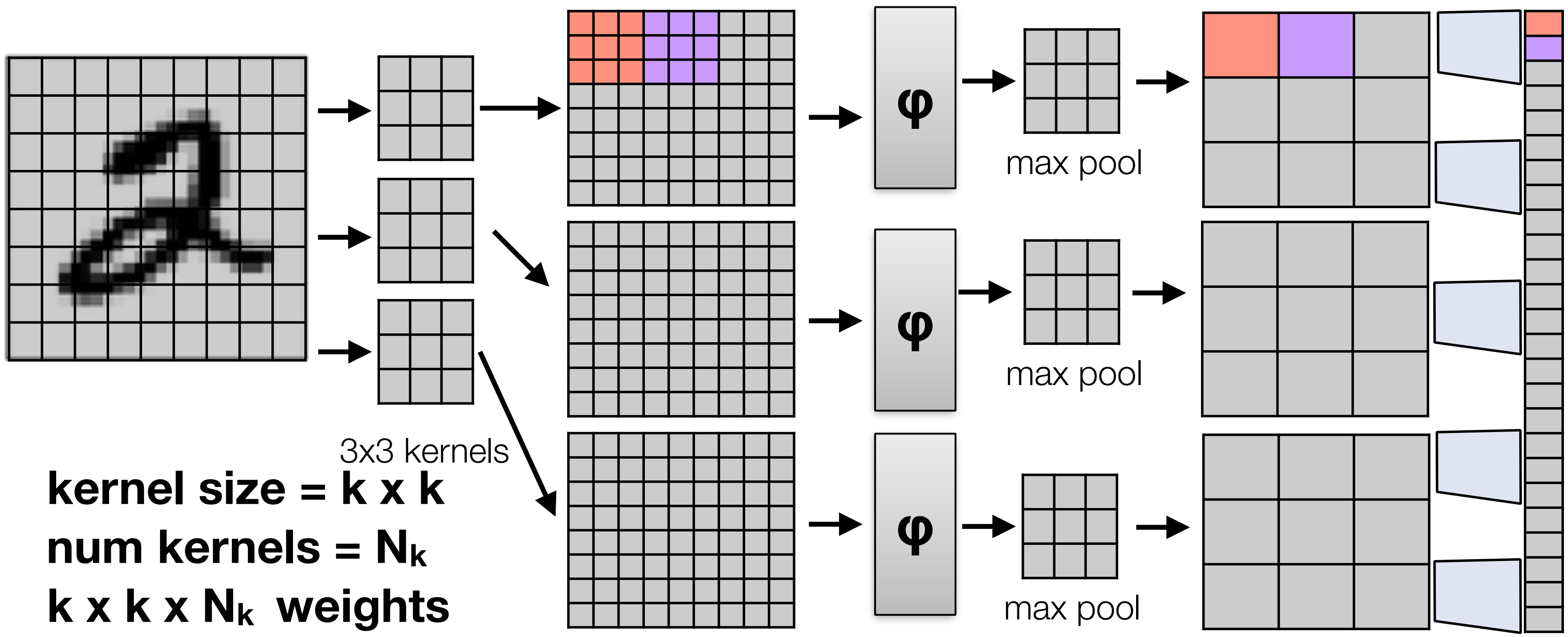


If image is 9x9, and each fully connected layer is 20 hidden neurons wide, how many parameters are in this NN (ignore bias)?



$$27+(27 \times 20) + (20 \times 10) = 767$$

Simple Example: From Fully Connected to CNN



kernel size = k x k

num kernels = N_k

$k \times k \times N_k$ weights

convolutional params

$N_k \times k^2$ ← filter dimension

num filters

pooling outputs

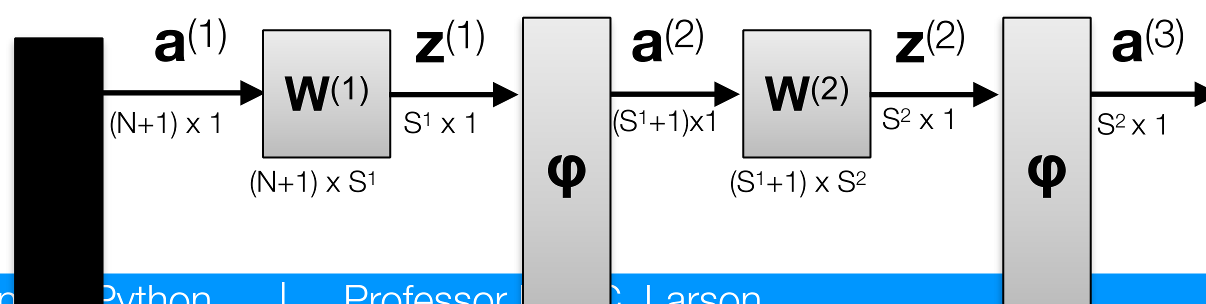
if stride = k

$(K/k) \times (K/k) \times N_k$

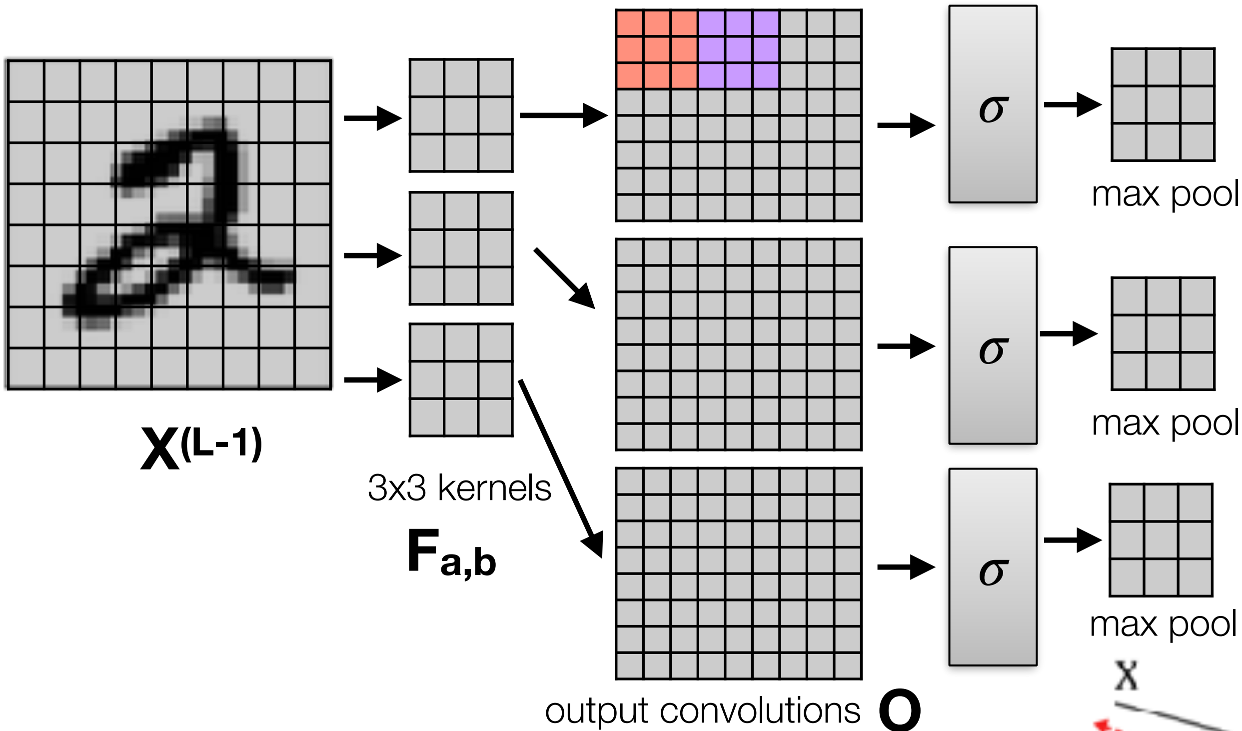
Input to MLP

$N_k \times (K^2/k^2)$

image dimension



CNN gradient



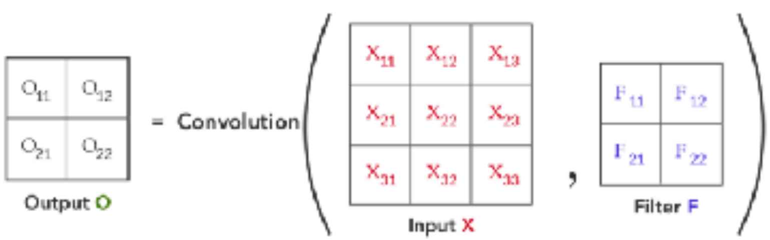
Derivative of max pool is easy:

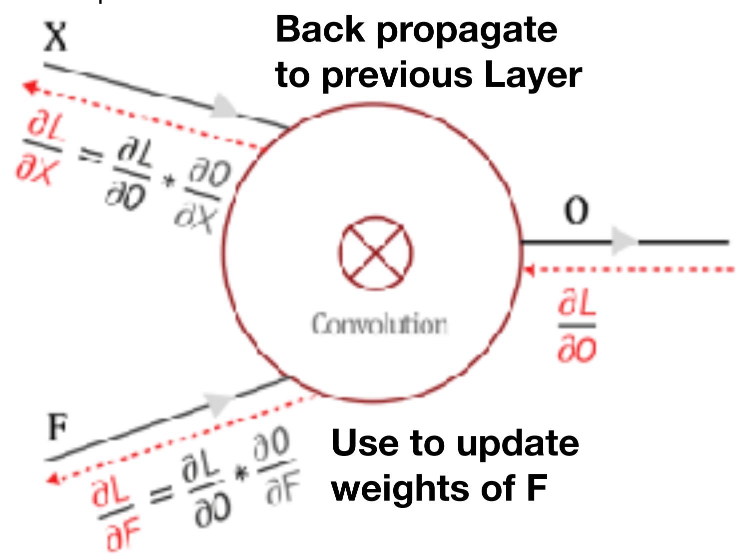
for each input X_i

$$\text{pool}'(X_i) = 1 \text{ if } X_i \text{ is max}$$
$$0 \text{ else}$$

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

Derivative of convolution is more involved:





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

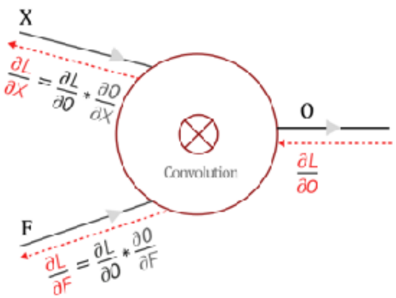
Gradient of Convolution

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

for back propagation

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

for weight updates



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

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

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

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

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

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

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

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

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

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

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

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

Filter updates

Output from convolution

Sensitivity from next layer

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

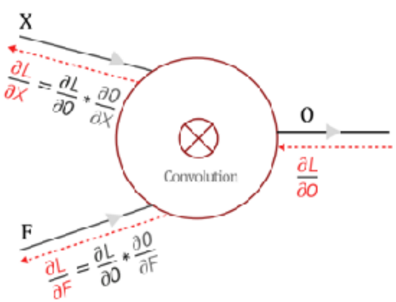
Gradient of Convolution

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

for back propagation

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

for weight updates



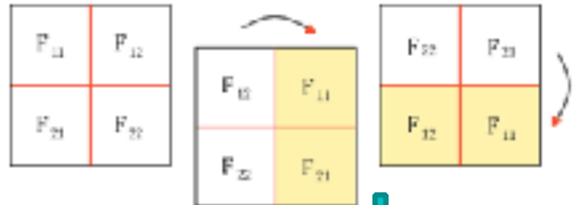
$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

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

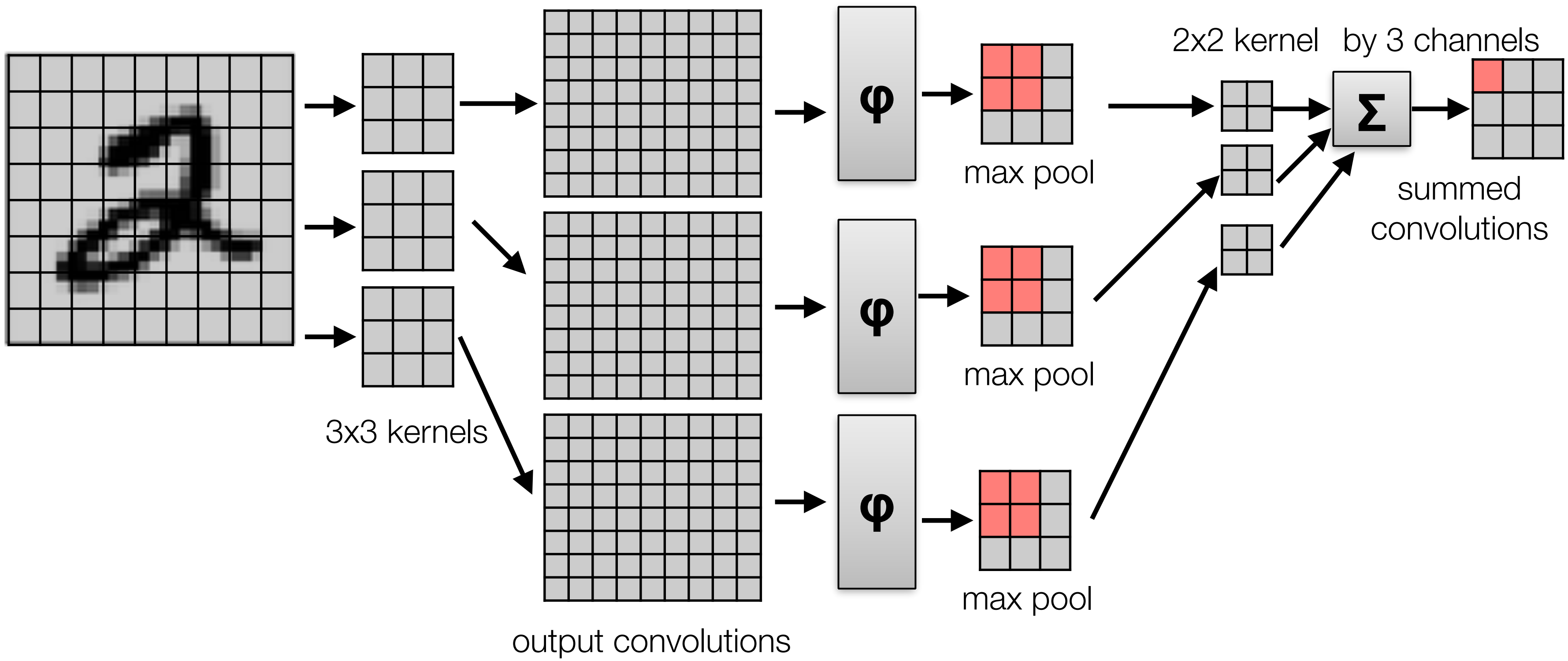
Sensitivity from next layer

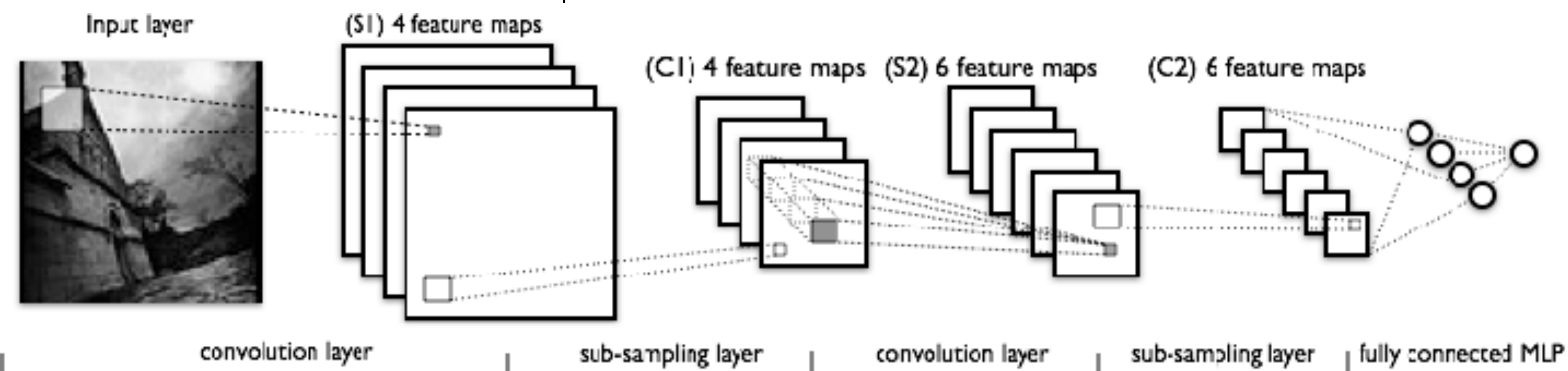
https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c

CNN Gradient

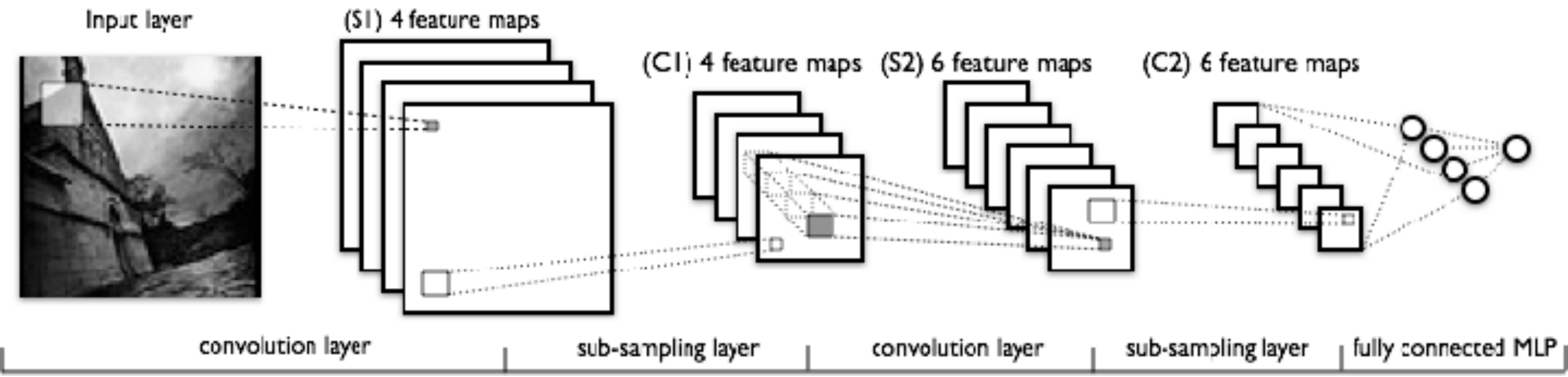
- Takeaways:
  - Derivative of a convolutional layer is calculated through two additional convolutions
    - One for filter updates
    - One for calculating a new sensitivity
  - We need to run convolution fast in order to speed up both:
    - feedforward operations (inference and training)
    - back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

CNN adding more convolutional layers



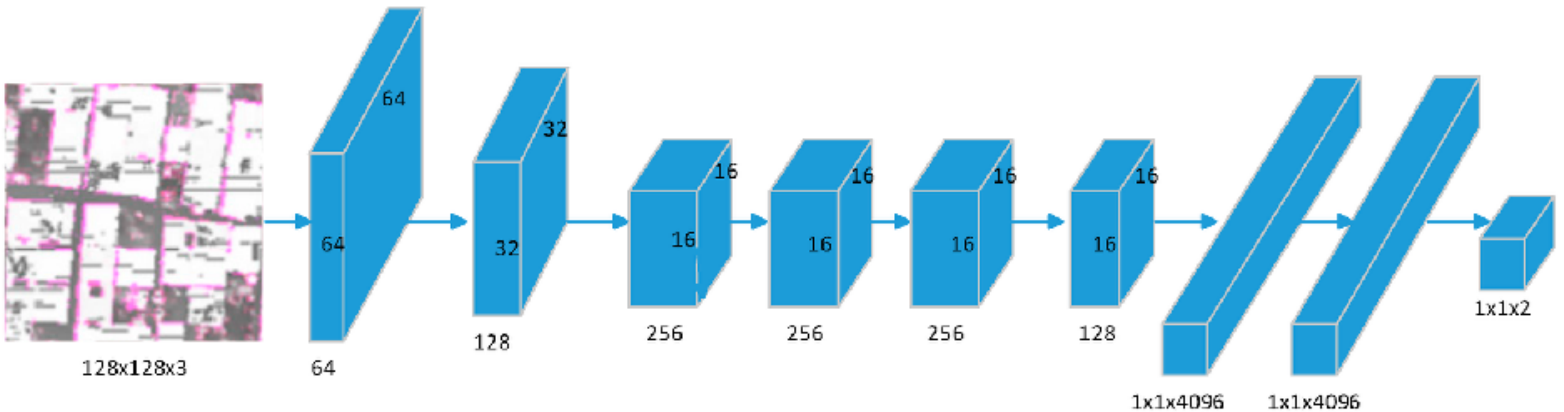


Some Example CNN Architectures

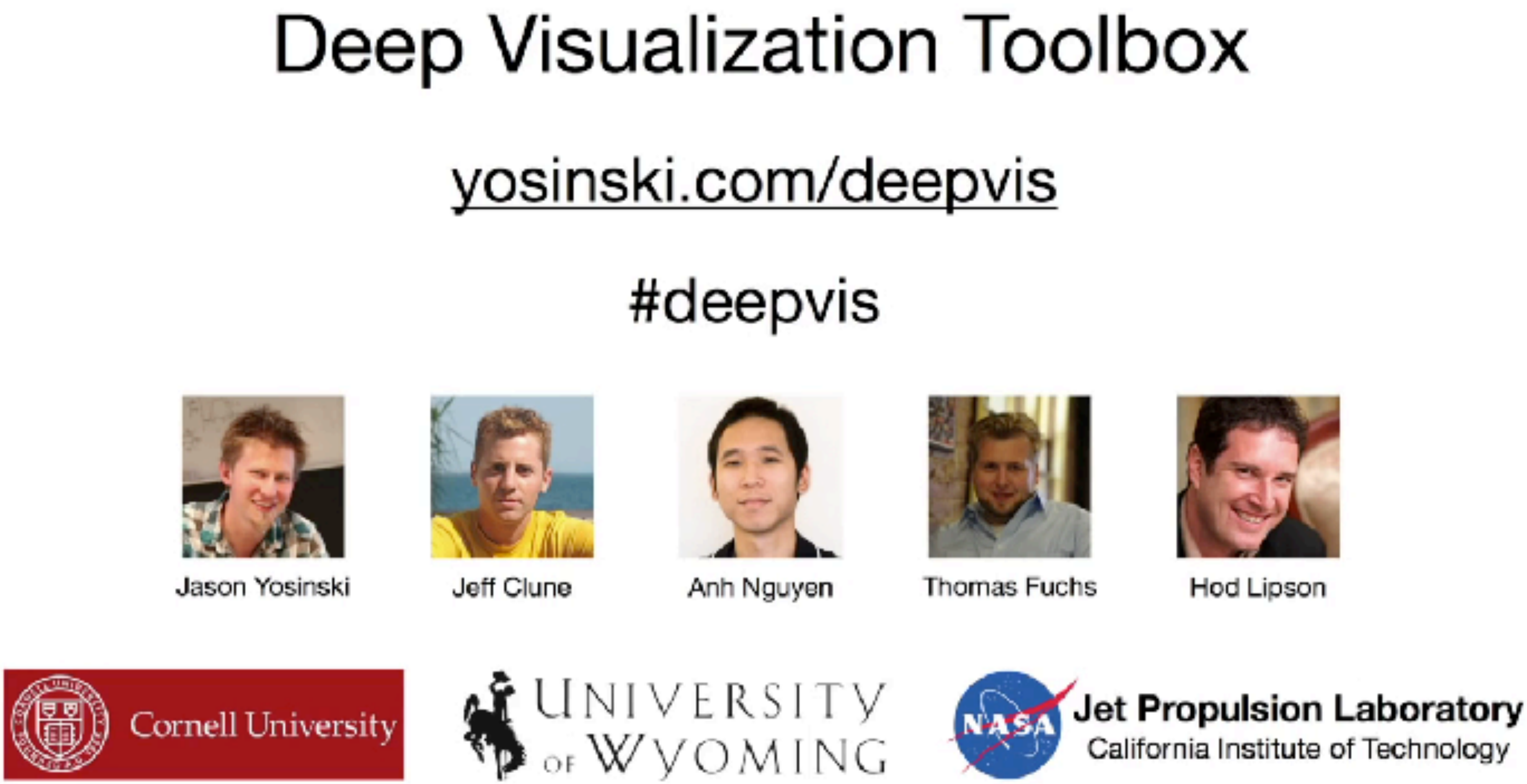


pooling

pooling



CNN: What does it all mean?



https://github.com/yosinski/deep-visualization-toolbox

Demo

Convolutional Neural Networks
in TensorFlow
with Keras

```
11. Convolutional Neural Networks.ipynb
```

Next Lecture

- More CNN architectures and CNN history