Lecture Notes for
Machine Learning in Python

⟦👨‍🏫, 👨‍💻, 🐍, 👨‍🔬⟧

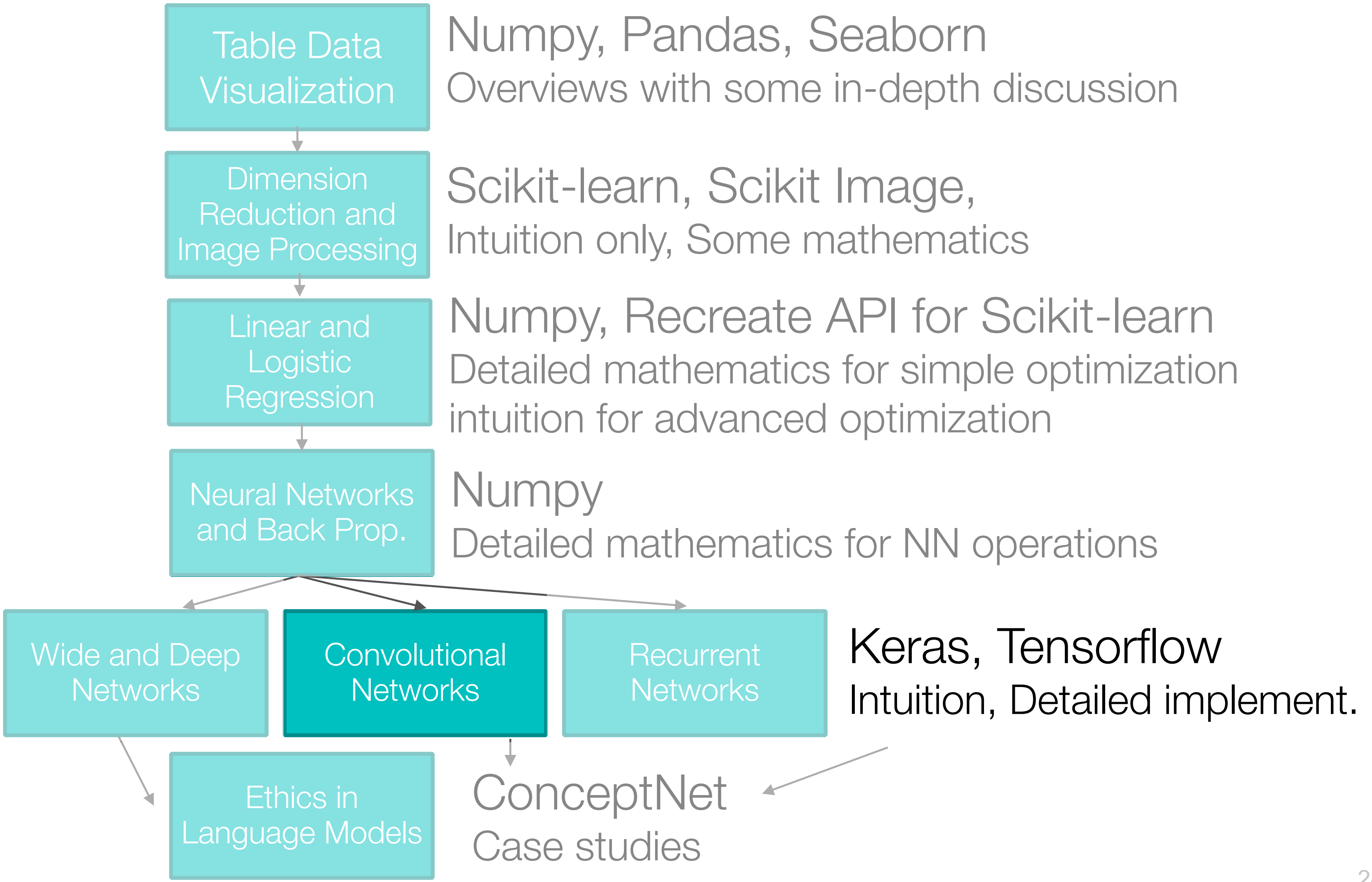
Professor Eric Larson

Convolutional Networks

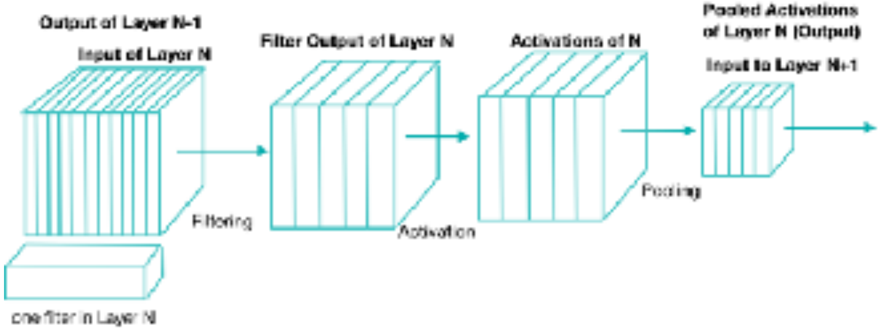
Class logistics and Agenda

- Logistics:
  - Next week, Thanksgiving break (lecture)
  - Lab due dates
- Agenda:
  - CNN Review
  - CNN Demo
  - CNN Town Hall
- Next Time:
  - More Advanced CNNs

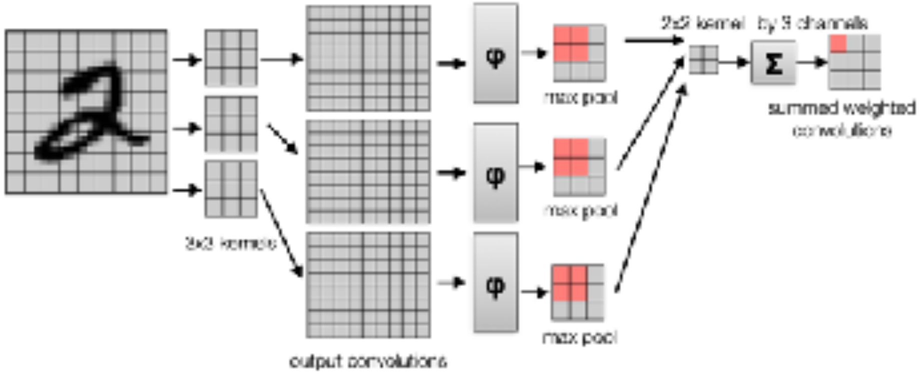
Class Overview, by topic

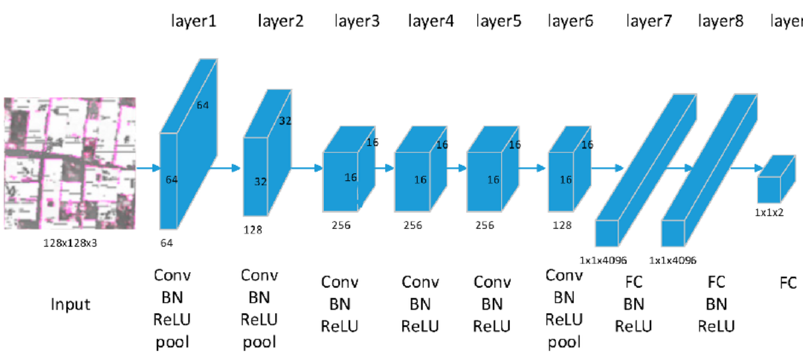


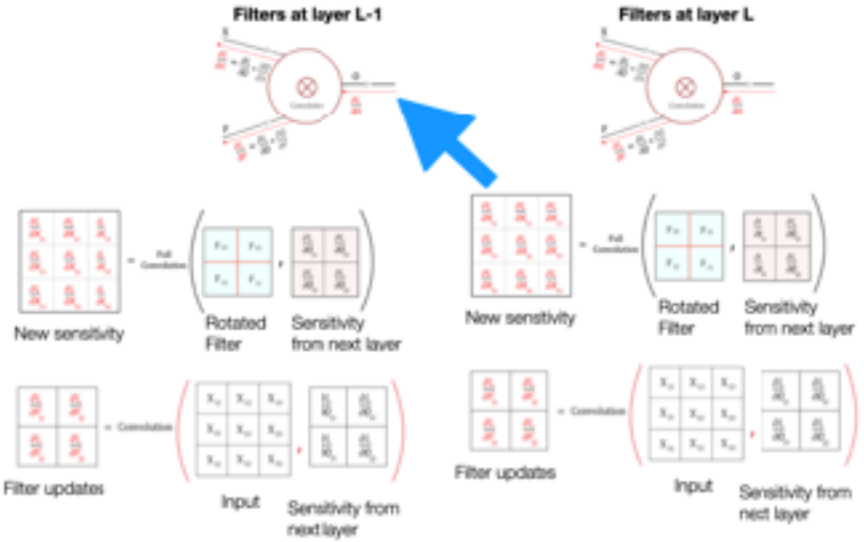
Last Time:



Structure of Each Tensor: Channels x Rows x Columns



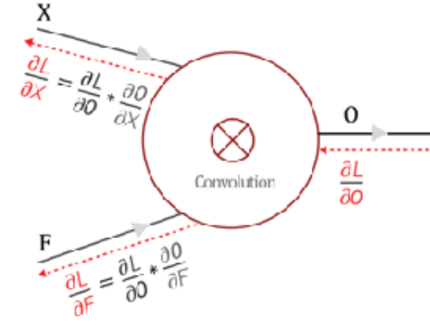


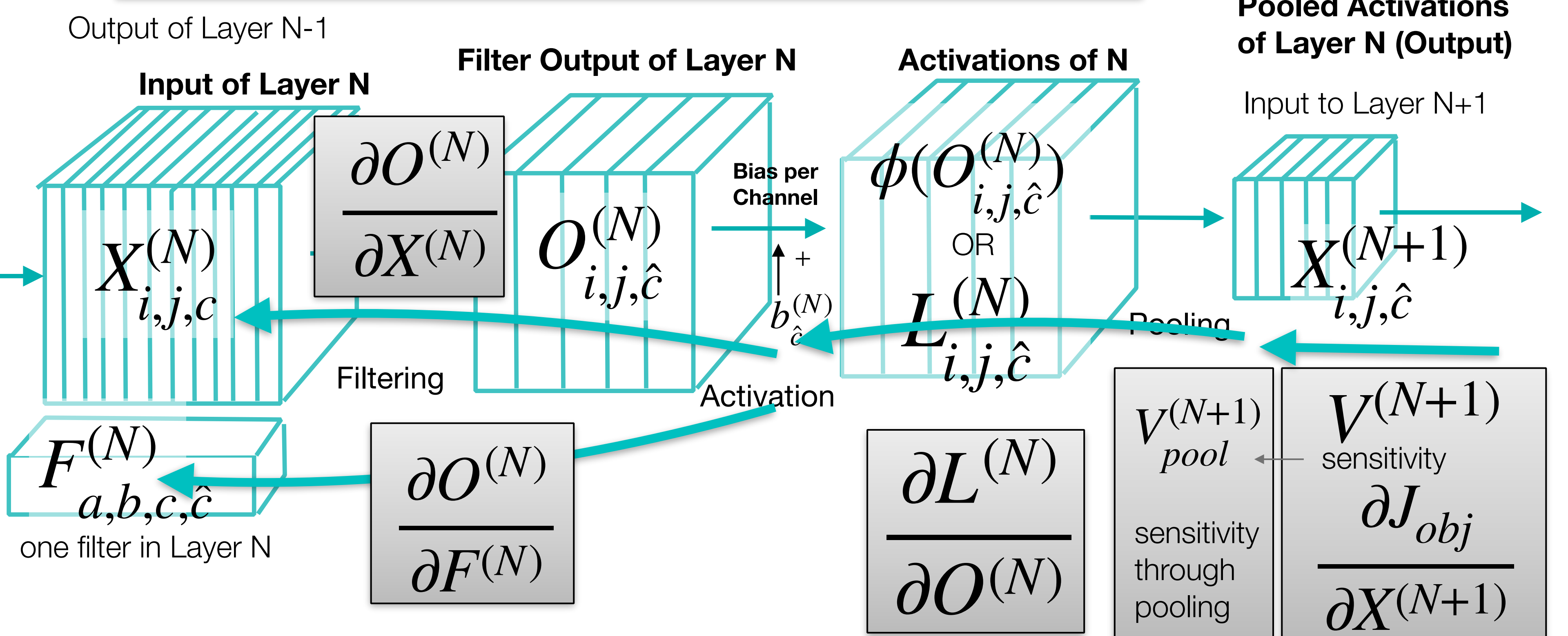


CNNs Back Propagation

Sensitivity to layer in back propagation

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





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: C

Just incorporate sensitivity, to get weight update

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

Self-test

- For each traditional convolutional layer in a CNN, there are two convolutions per filter in the layer during back propagation.
    - **False**. There is a single convolution needed during back propagation for each filter.
    - **False**. The first convolutional layer has only one convolution needed, but all others have two per filter.
    - **True**. The filter weights are updated through two convolutions, one for each dimension.
    - **True**. One convolution is needed for pooling, and the other for filter weight updates.

Demo

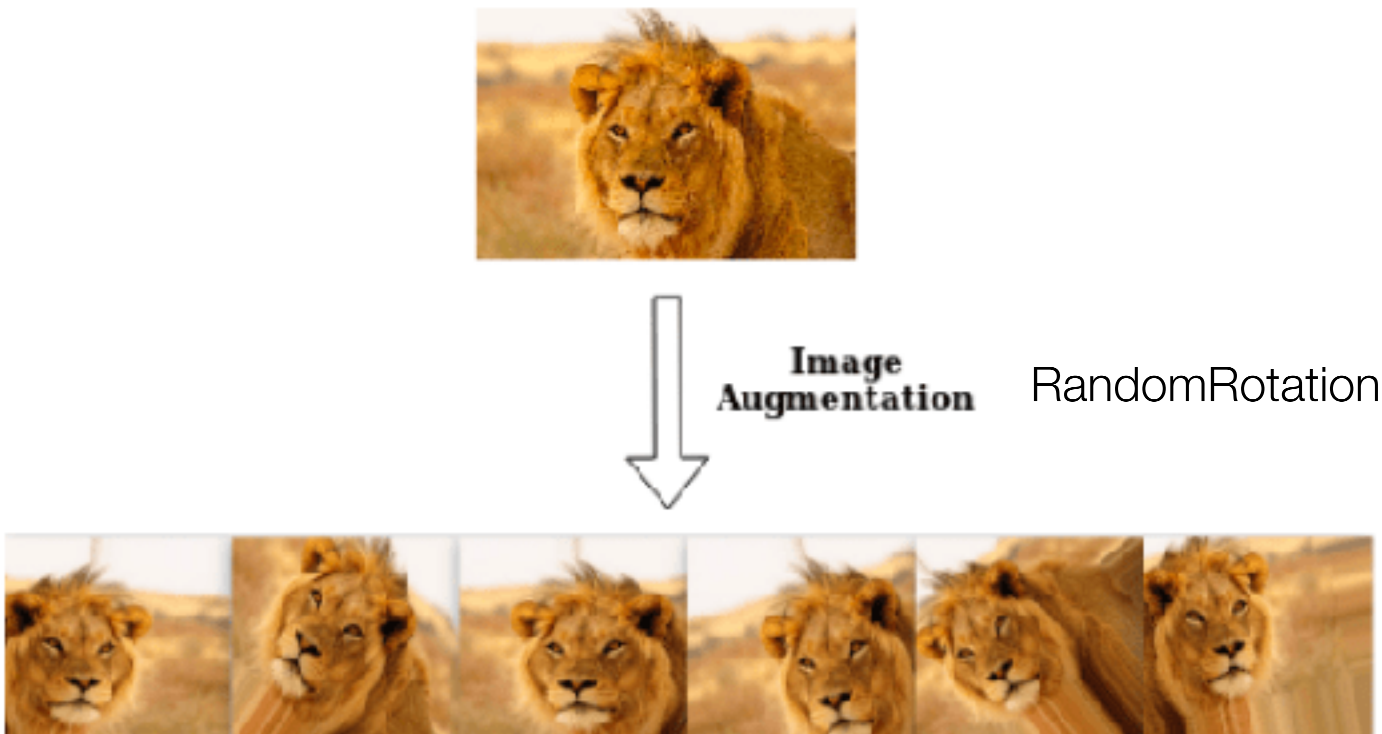
Convolutional Neural Networks
in TensorFlow
with Keras

`11. Convolutional Neural Networks.ipynb`

Image Data Augmentation

```python
cnn = Sequential()
```

```python
# add in augmentations directly
cnn.add( RandomFlip("horizontal") ) # flip horizontally
cnn.add( RandomRotation(0.05) ) # rotate by 5%
cnn.add( RandomTranslation(height_factor=0.1, width_factor=0.1) )
cnn.add( RandomBrightness(factor=0.1, value_range=(0.0, 1.0)) ) #
cnn.add( RandomContrast(0.1) ) # add or decrease contrast
```

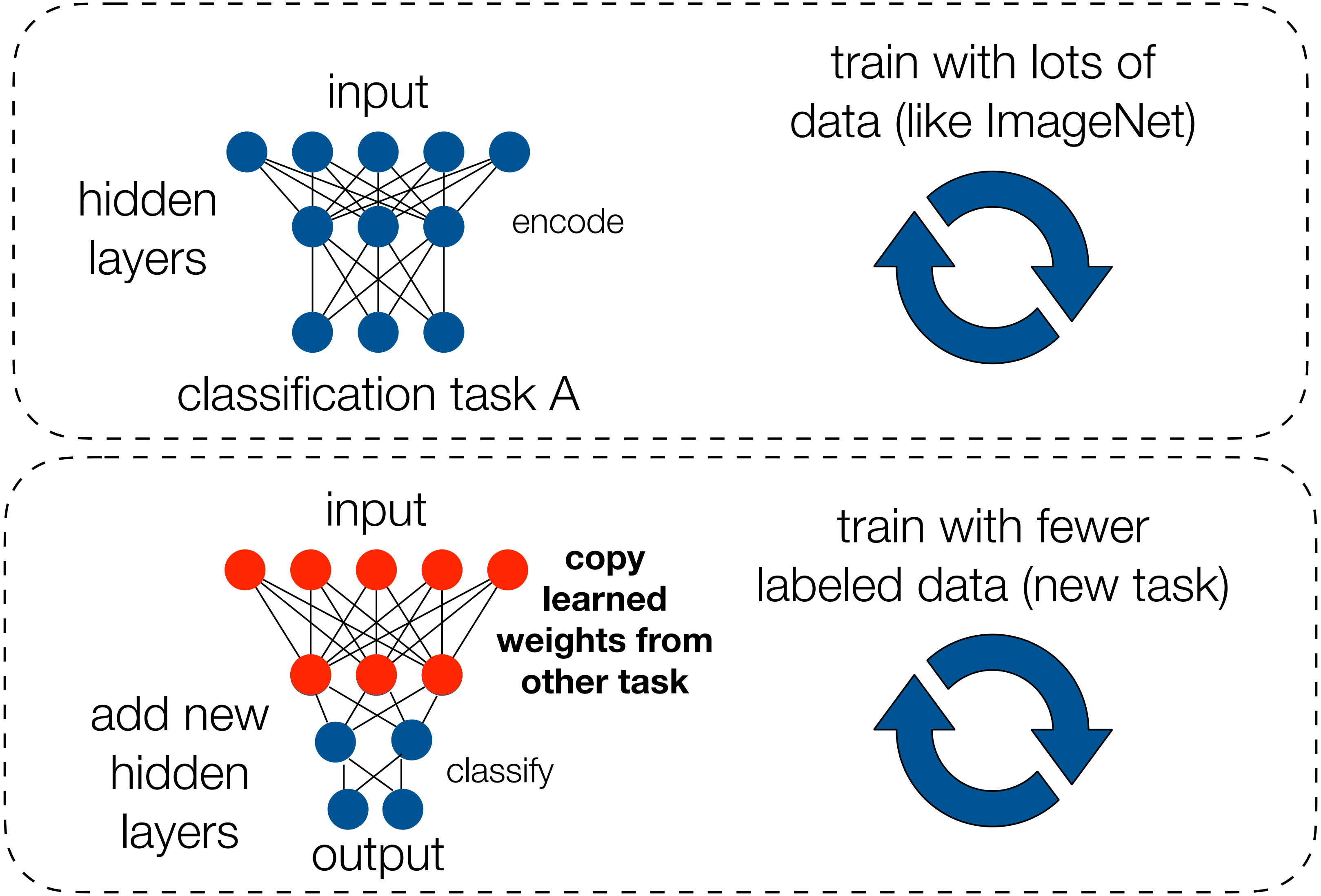


RandomRotation( )

https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/

Transfer Learning

- transfer learning: a basic primer



Many Pre-trained Models to choose from!

AlexNet

A landmark in computer vision, this 2012 winner of ImageNet has over 50,000 citations.

AlexNet (Places)

The same architecture as the classic AlexNet model, but trained on the Places365 dataset.

Inception v1

Also known as GoogLeNet, this network set the state of the art in ImageNet classification in 2014.

Inception v1 (Places)

The same architecture as the classic Inception v1 model, but trained on the Places365 dataset.

VGG 19

Introduced in 2014, this network is simpler than Inception variants, using only 3x3 convolutions and no branches.

Inception v3

Released in 2015, this iteration of the Inception architecture improved performance and efficiency.

Inception v4

Released in 2016, this is the fourth iteration of the inception architecture, focusing on uniformity.

ResNet v2 50

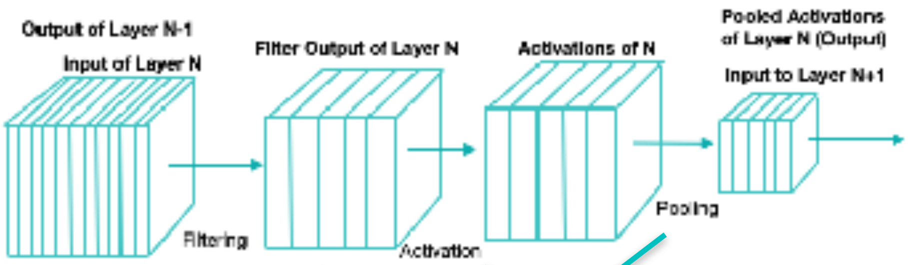
ResNets use skip connections to enable stronger gradients in much deeper networks. This variant has 50 layers.

CNN Town Hall

Thanks to
Machine Learning the
robot apocalypse was
short lived!

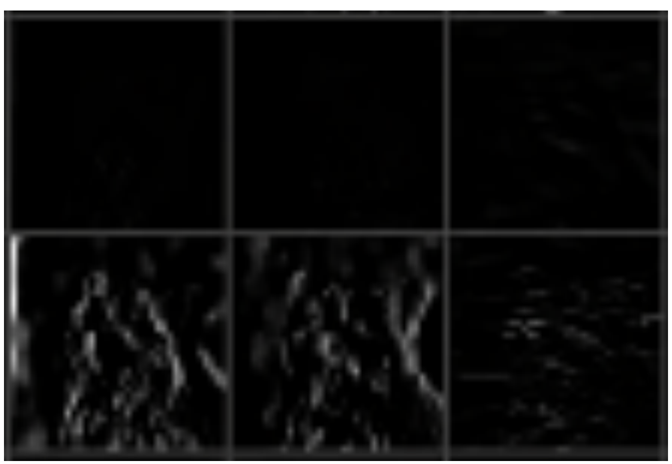
What has a CNN learned?

Early Conv Layers in CNN



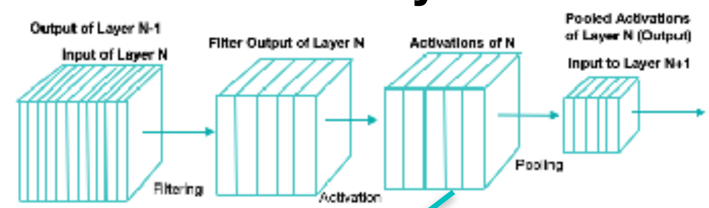
(i, j, channels)

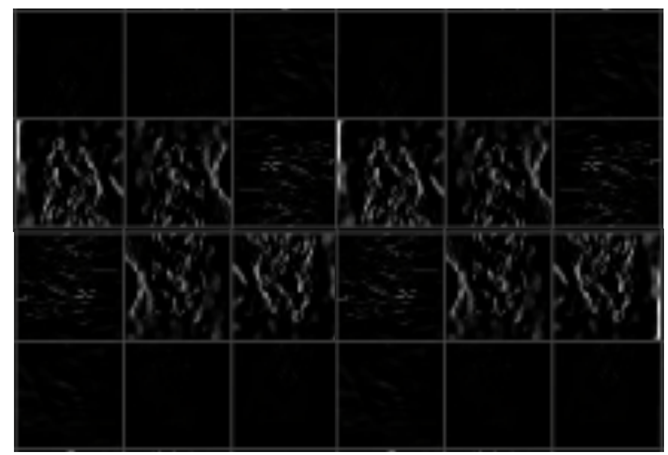
Visualize
each channel



Separate channels. Early layers have fewer filters, but larger activations

Later Conv Layers in CNN





Later layers have more filters,
but smaller activations

Naming in Video:

conv1 (output activations of conv)
p1 (output of pooling)
n1 (output of normalization)

CNN: Visuals

Deep Visualization Toolbox

yosinski.com/deepvis

#deepvis

Jason Yosinski | Jeff Clune | Anh Nguyen | Thomas Fuchs | Hod Lipson

Cornell University | University of Wyoming | Jet Propulsion Laboratory, California Institute of Technology

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

Next Lecture

- More CNN architectures and CNN history