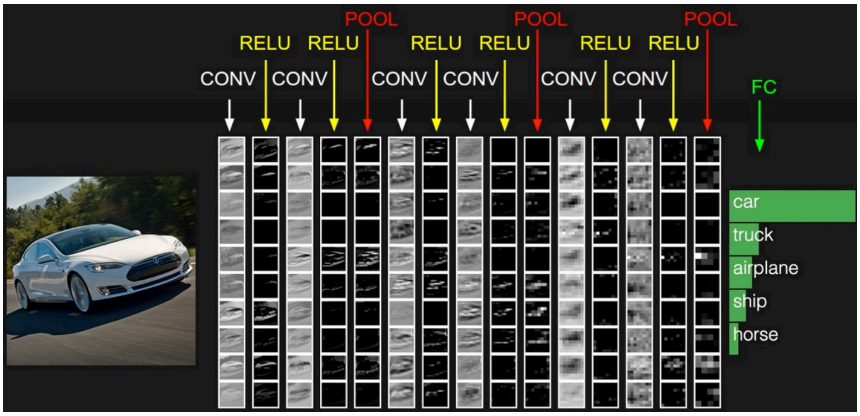
Convolutional Neural Networks (CNNs)

Fei-Fei Li, Justin Johnson, Serena Yeung (http://cs231n.stanford.edu/)

The Convolutional Neural Network in this example is classifying images live in your browser using Javascript, at about 10 milliseconds per image. It takes an input image and transforms it through a series of functions into class probabilities at the end. The transformed representations in this visualization can be losely thought of as the activations of the neurons along the way. The parameters of this function are learned with backpropagation on a dataset of (image, label) pairs. This particular network is classifying CIFAR-10 images into one of 10 classes and was trained with ConvNetJS. Its exact architecture is [conv-relu-conv-relu-pool]x3-fc-softmax, for a total of 17 layers and 7000 parameters. It uses 3x3 convolutions and 2x2 pooling regions. By the end of the class, you will know exactly what all these numbers mean.

Sample stages of a CNN



CONV: “convolution” layer with weights that are learned

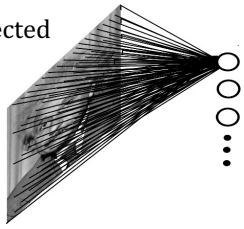
RELU: “rectified linear unit” applies an activation function

POOL: “pooling” selects maximum value in small neighborhoods

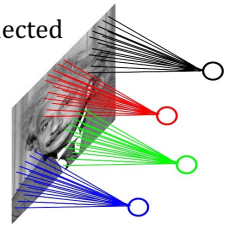
FC: “fully-connected” neural network

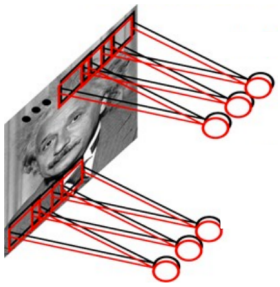
Convolutional layer

fully-connected network:

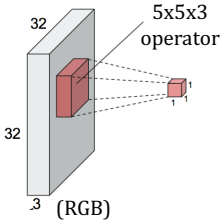


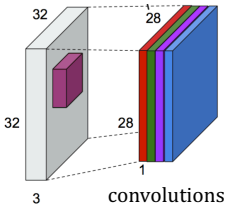
locally-connected network:





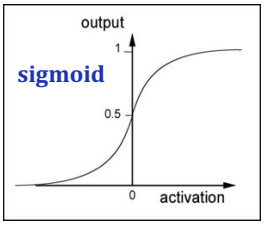
- early layers perform a *convolution* of their inputs
- multiple convolution operators (e.g. red & black)
- weights in convolution operators are *learned*
- convolution operators are typically small (e.g. 5x5)



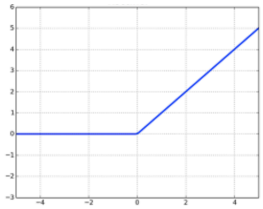


convolutions with multiple operators

ReLU & max pooling layers

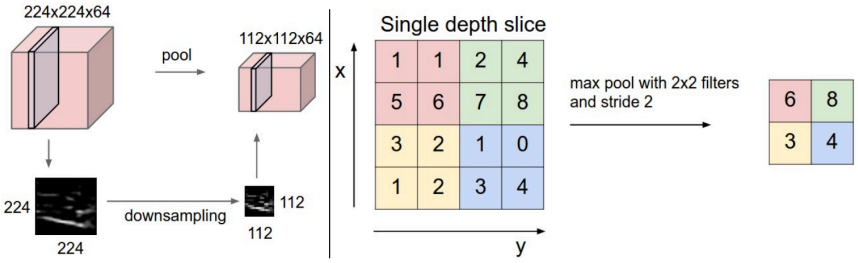


Rectified Linear Activation



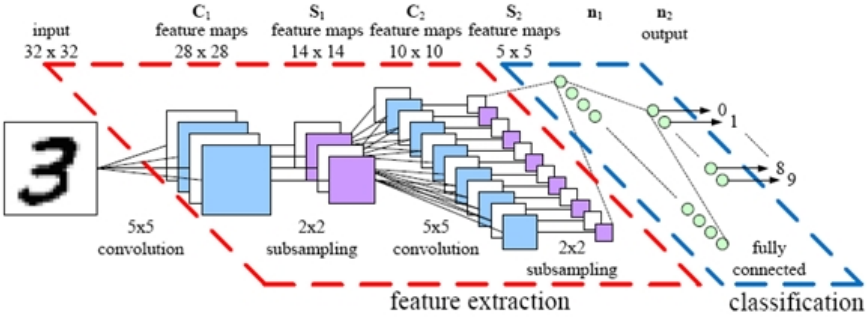
$$RELU(x) = \begin{cases} 0 \text{ if } x < 0 \\ x \text{ if } x >= 0 \end{cases}$$

convolution result



Adding a fully-connected neural net layer

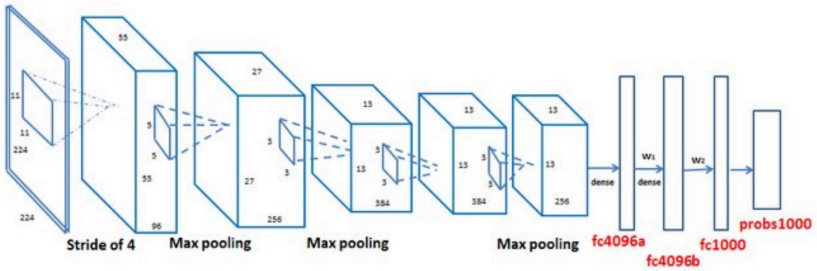
Recognizing digits from the MNIST database with a CNN:



LeNet

LeCun, Bottou, Bengio, Haffner (1998)

AlexNet, ZF Net, GoogLeNet, VGGNet, ResNet, ...

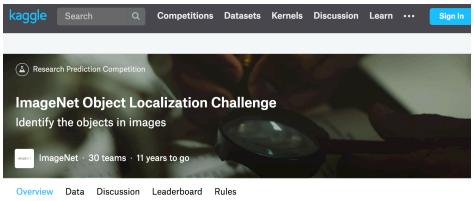


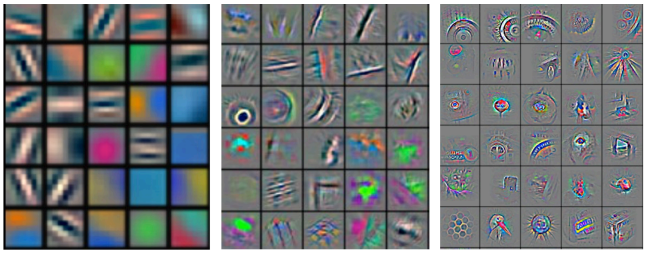
AlexNet: Krizhevsky, Sutskever, Hinton (2012)

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

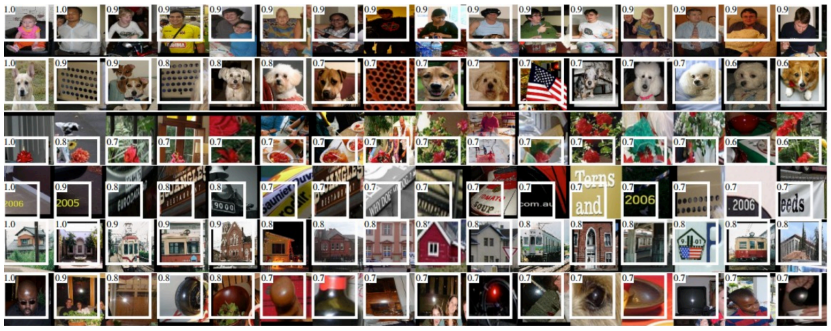
IMAGENET

Annually since 2010





Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

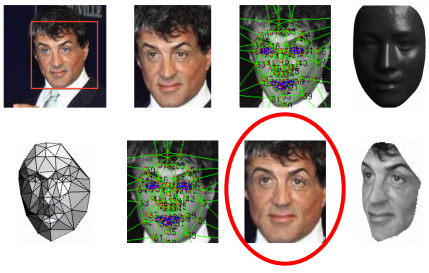


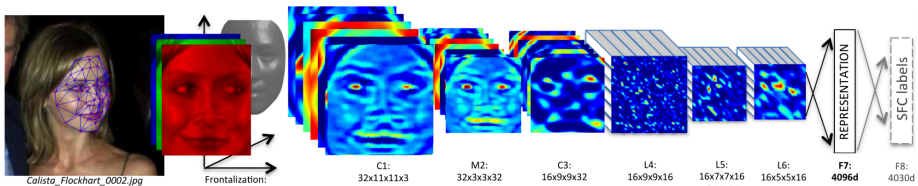
Maximally activating images from some POOL5 neurons of AlexNet (Girshick et al., 2014)

Facebook's DeepFace system

Taigman et al., 2014

- detect face
- 2D align face in crop window using 6 fiducial points
- align to 3D shape model using 67 fiducial points
- use 3D model + image to generate frontal view

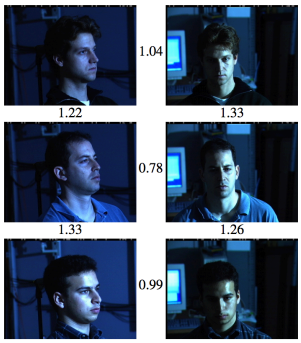




Google's FaceNet system

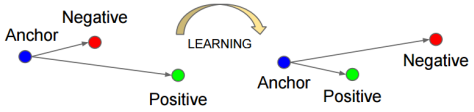
Schroff et al., 2015

FaceNet also uses a deep convolutional network



threshold = 1.1 classifies pairs correctly (smaller value means more similar)

- learns mapping from images to a space where distance between images captures similarity
- training data: triplets of face thumbnails
  - two same ID, one different ID
- learning process: minimize distance between anchor & positive images (same ID), maximize distance between anchor & negative images



OpenFace implementation of FaceNet system

Amos et al., 2015