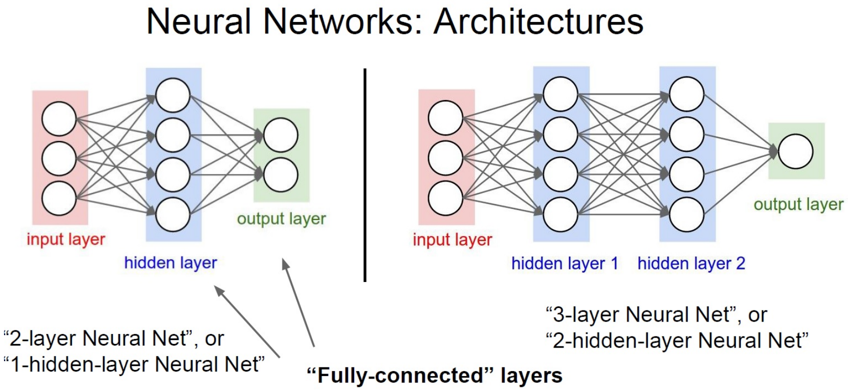
Deep Learning 2

Farzaneh Mirzazadeh
From Stanford C231 Course by
Fei Fei Li, Andrej Karpathy, Justin Johnson

University of California, Santa Cruz

Winter' 17

Multilayer Perceptron



- Forward propagation: compute the output
- Backpropagation: compute derivative of error with respect to parameters: a systematic chain rule

Deep nets have many parameters: prune to overfitting.

Preventing overfitting in deep nets

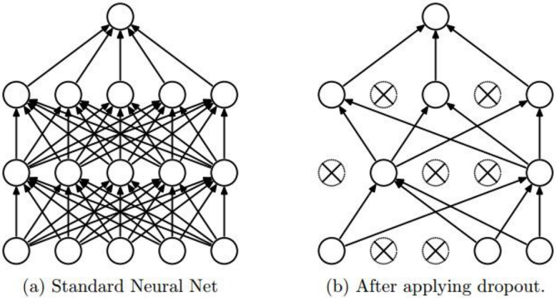
- **Parameter L2 and L1 norm regularization**
  Standard, as before, leave bias unregularized L2 regularization for NNs is called weight decay
- **Dataset augmentation**
  use for object recognition in images, Create fake data by rotating, scaling, translating
- **Noise injection**
  Add random noise to input
- **Early stopping:**
  Stop training as soon as the validation set error increases
- **Dropout**
  Important: Add noise to hidden layers. Discussed in this lecture.

See Chapter 7 of Deep Learning book.

Part 1: Dropout

Regularization: **Dropout**

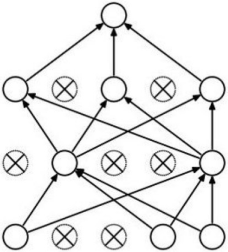
"randomly set some neurons to zero in the forward pass"



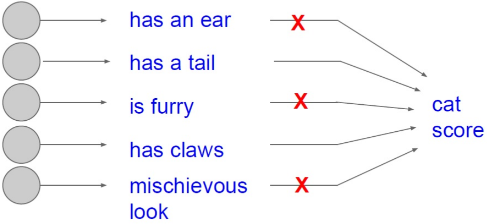
[Srivastava et al., 2014]

Take this into account in backward pass too. (Zero out the gradient from the neurons that are dropped out

Waaaait a second…
How could this possibly be a good idea?



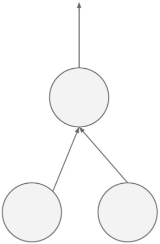
Forces the network to have a redundant representation.



At test time….

Can in fact do this with a single forward pass! (approximately)

Leave all input neurons turned on (no dropout).



(this can be shown to be an approximation to evaluating the whole ensemble)

Multiply the activation to p (the dropout probability)

Ensemble Methods

- Ensemble Methods: Combine different methods to reduce generalization error
- Typically ensemble methods are winners of challenges!
- Examples: Bagging and boosting

Bagging

- Bagging: short for bootstrap aggregation
- Idea: train different models separately, then have them all vote on output for test samples
- Generate different datasets by sampling with replacement from original dataset, then train on those
- Rationale: Different models will usually not make the same error on test set
- Dropout can be considered as a form of BAGGING

Part 2

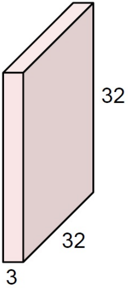
Convolutional Neural Networks (CNNs):
Network for images

CNN

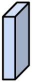
- Still Feed forward Deep Net
- State of the art results in computer vission
- 2012 Imagenet Challenge Winner, Hinton's group
- Stacks of
  - Convolutional layer
  - RELU layer
  - Pooling layer
- Followed by a fully connected layer
- Number of parameters in a fuly connected feed forward neural net for images would be large

Convolution Layer

32x32x3 image



5x5x3 filter

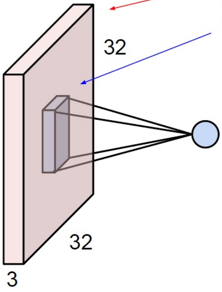


Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolution Layer

32x32x3 image

5x5x3 filter w

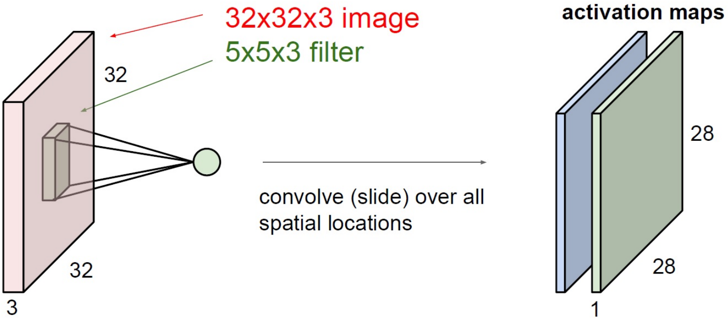


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

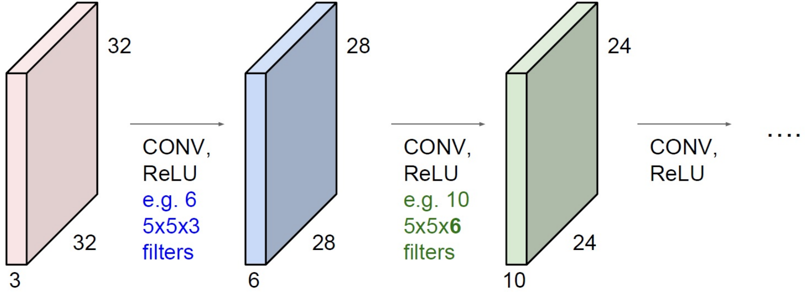
$$w^T x + b$$

Convolution Layer

consider a second, green filter

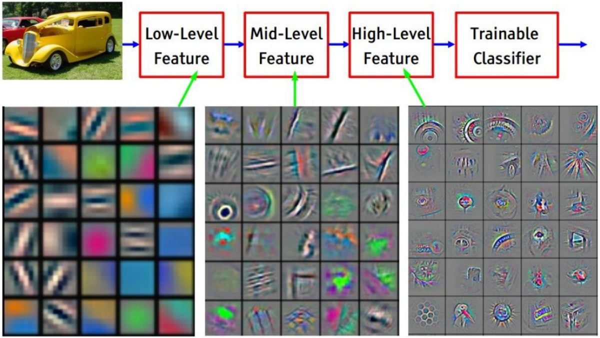


Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions

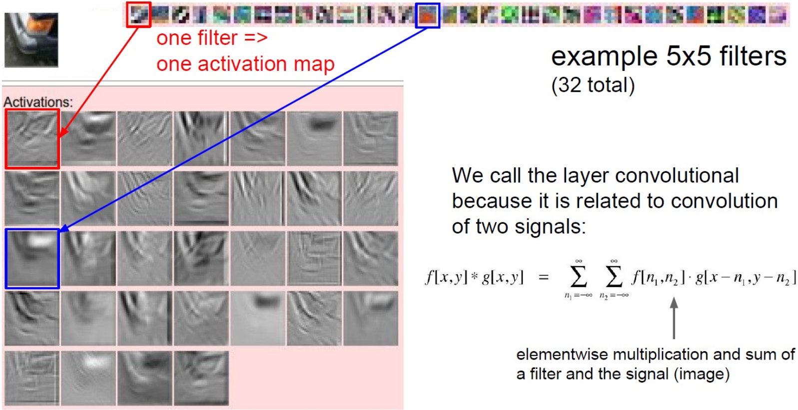


Preview

[From recent Yann LeCun slides]



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



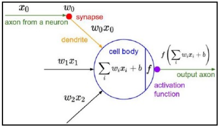
example 5x5 filters
(32 total)

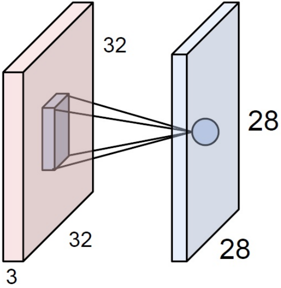
We call the layer convolutional because it is related to convolution of two signals:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

elementwise multiplication and sum of a filter and the signal (image)

The brain/neuron view of CONV Layer





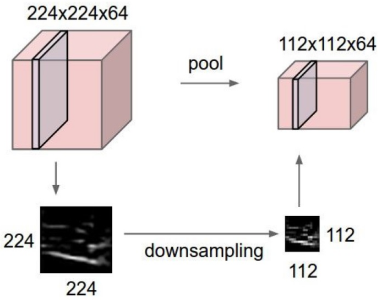
An activation map is a 28x28 sheet of neuron outputs:

1. Each is connected to a small region in the input
2. All of them share parameters

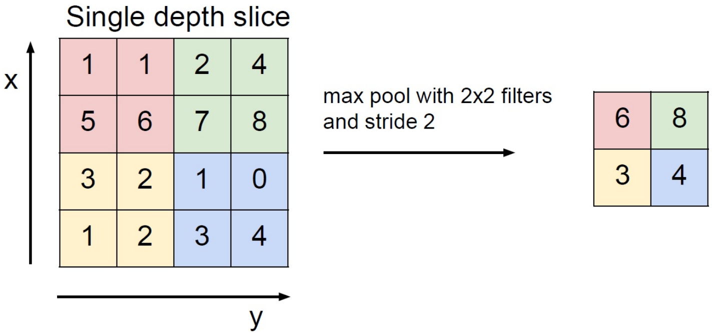
"5x5 filter" -> "5x5 receptive field for each neuron"

Pooling layer

- makes the representations smaller and more manageable
- operates over each activation map independently:

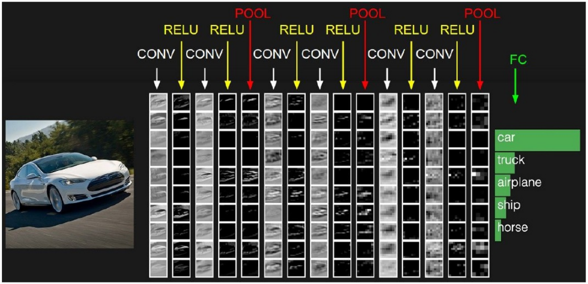


MAX POOLING


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

Fully Connected Layer (FC layer)

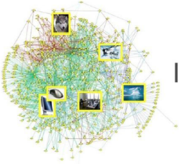
- Contains neurons that connect to the entire input volume, as in ordinary Neural Networks



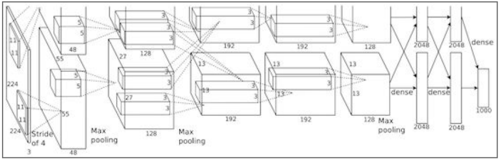
A bit of history:

ImageNet Classification with Deep Convolutional Neural Networks

[Krizhevsky, Sutskever, Hinton, 2012]



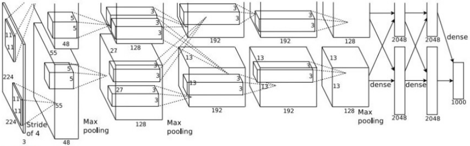
IMAGENET



“AlexNet”

Case Study: AlexNet

[Krizhevsky et al. 2012]



Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

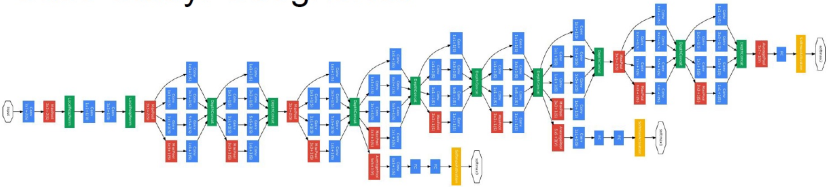
Fast-forward to today: ConvNets are everywhere

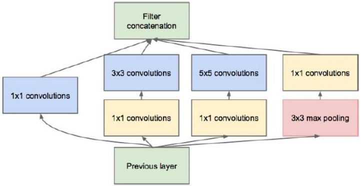
self-driving cars

NVIDIA Tegra X1

Case Study: GoogLeNet

[Szegedy et al., 2014]





Inception module

ILSVRC 2014 winner (6.7% top 5 error)