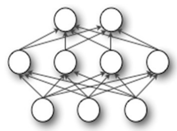
Deep learning for visual recognition

Tues April 23
Kristen Grauman
UT Austin



Last time

- Supervised classification continued
  - Nearest neighbors
  - Support vector machines
    - HoG pedestrians example
    - Kernels
    - Multi-class from binary classifiers

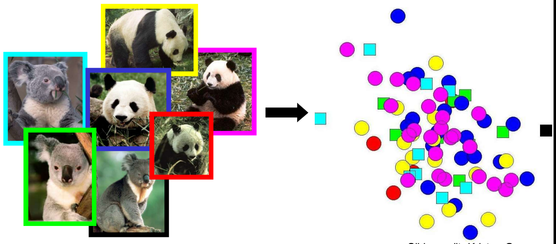
Recalll: Examples of kernel functions

- Linear: $K(x_i, x_j) = x_i^T x_j$
- Gaussian RBF: $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
- Histogram intersection:
  $$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

- Kernels go beyond vector space data
- Kernels also exist for "structured" input spaces like sets, graphs, trees…

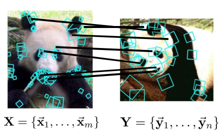
Discriminative classification with sets of features?

- Each instance is unordered set of vectors
- Varying number of vectors per instance



Slide credit: Kristen Grauman

Partially matching sets of features



Optimal match: O(m^3)
Greedy match: O(m^2 log m)
Pyramid match: O(m)

(**m**=num pts)

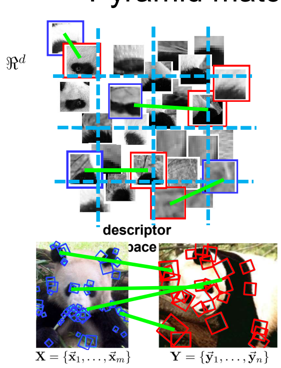
$$\min_{\pi: \mathbf{X} \to \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} ||\mathbf{x}_i - \pi(\mathbf{x}_i)||$$

nate matching kernel that makes it practical to compare large sets of features based on their partial correspondences.

[Previous work: Indyk & Thaper, Bartal, Charikar, Agarwal & Varadarajan, …]

Slide credit: Kristen Grauman

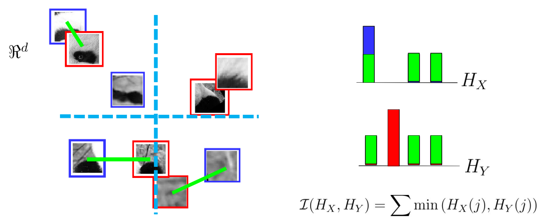
Pyramid match: main idea



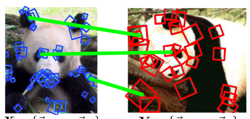
Feature space partitions serve to "match" the local descriptors within successively wider regions.

Slide credit: Kristen Grauman

Pyramid match: main idea



$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$
$$= 3$$



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Histogram intersection counts number of possible matches at a given partitioning.

Slide credit: Kristen Grauman

Pyramid match

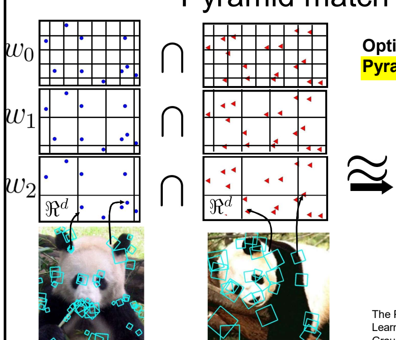
$$K_\Delta(X,Y) = \sum_{i=0}^{L} \underbrace{2^{-i}}_{\text{measures difficulty of a match at level } i} \underbrace{\left\{\mathcal{I}\left(H_X^{(i)}, H_Y^{(i)}\right) - \mathcal{I}\left(H_X^{(i-1)}, H_Y^{(i-1)}\right)\right\}}_{\text{number of newly matched pairs at level } i}$$

- For similarity, weights inversely proportional to bin size (or may be learned)
- Normalize these kernel values to avoid favoring large sets

[Grauman & Darrell, ICCV 2005]

Slide credit: Kristen Grauman

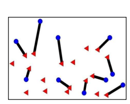
Pyramid match



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Optimal match: O(m³)
Pyramid match: O(mL)

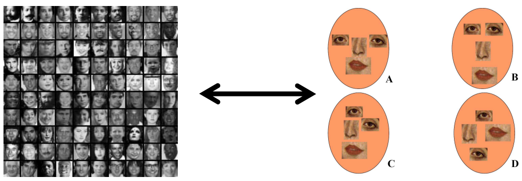
$\approx$



optimal partial matching

The Pyramid Match Kernel: Efficient Learning with Sets of Features. K. Grauman and T. Darrell. Journal of Machine Learning Research (JMLR), 8 (Apr): 725--760, 2007.

BoW Issue:
No spatial layout preserved!

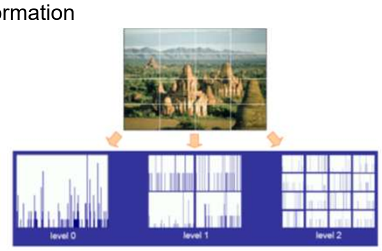


Too much? Too little?

Slide credit: Kristen Grauman

Spatial pyramid match

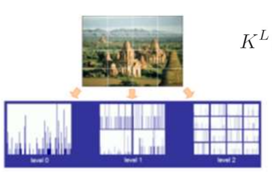
- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



[Lazebnik, Schmid & Ponce, CVPR 2006]

Spatial pyramid match

- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

Sum over PMKs computed in *image coordinate* space, one per word.

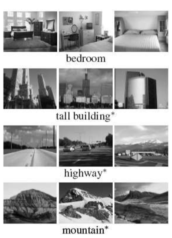
[Lazebnik, Schmid & Ponce, CVPR 2006]

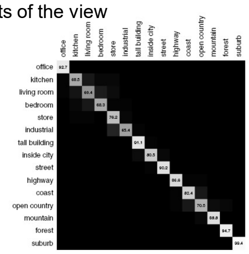
Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local

Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local pieces.
- Sensitive to global shifts of the view



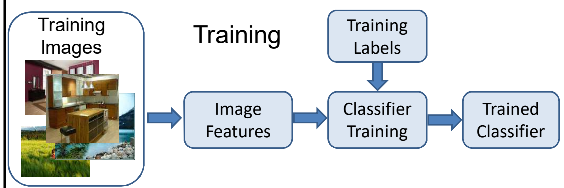


Confusion table

Today

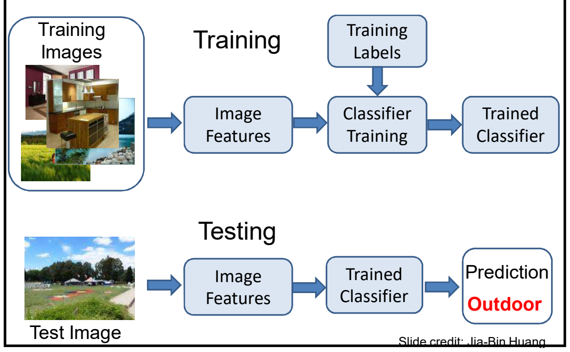
- (Deep) Neural networks
- Convolutional neural networks

Traditional Image Categorization: Training phase



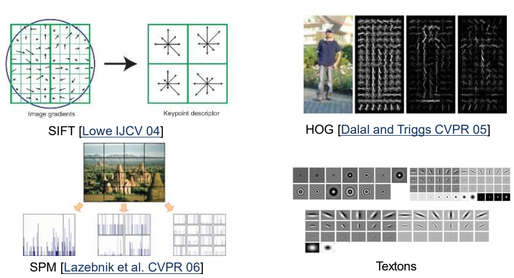
Slide credit: Jia-Bin Huang

Traditional Image Categorization: Testing phase



Slide credit: Jia-Bin Huang

Features have been key

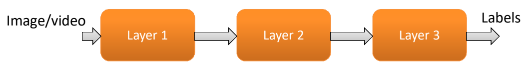


and many others:

SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

Learning a Hierarchy of Feature Extractors

- Each layer of hierarchy extracts features from output of previous layer
- All the way from pixels → classifier
- Layers have the (nearly) same structure

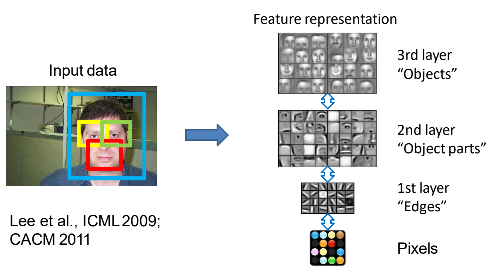


- Train all layers jointly

Slide: Rob Fergus

Learning Feature Hierarchy

Goal: Learn **useful higher-level features** from images



Lee et al., ICML 2009; CACM 2011

Slide: Rob Fergus

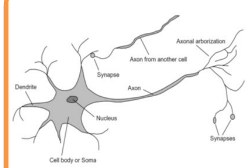
Learning Feature Hierarchy

- Better performance
- Other domains (unclear how to hand engineer):
  - Kinect
  - Video
  - Multi spectral

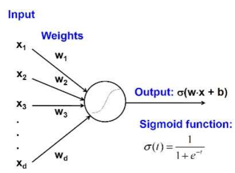
- Feature computation time
  - Dozens of features regularly used [e.g., MKL]
  - Getting prohibitive for large datasets (10's sec /image)

Slide: R. Fergus

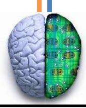
Biological neuron and Perceptrons



A biological neuron



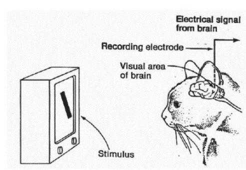
An artificial neuron (Perceptron)
- a linear classifier



Slide credit: Jia-Bin Huang

Simple, Complex and Hypercomplex cells

David H. Hubel and Torsten Wiesel

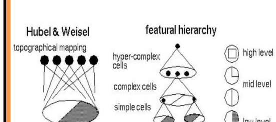


Suggested a **hierarchy** of **feature detectors** in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.

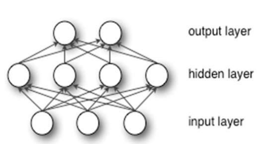
David Hubel's Eye, Brain, and Vision

Slide credit: Jia-Bin Huang

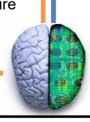
Hubel/Wiesel Architecture and Multi-layer Neural Network



Hubel and Weisel's architecture



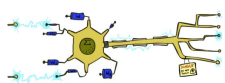
Multi-layer Neural Network
- A *non-linear* classifier



Slide credit: Jia-Bin Huang

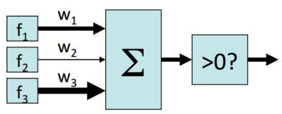
Neuron: Linear Perceptron

- Inputs are feature values
- Each feature has a weight
- Sum is the activation



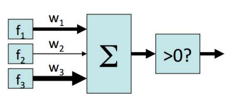
$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive, output +1
  - Negative, output -1



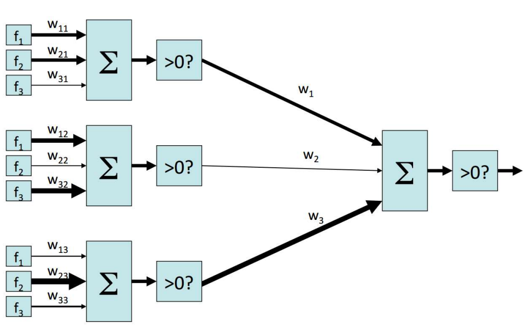
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



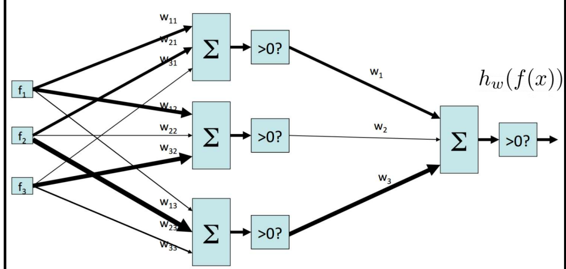
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Learning w

- Training examples

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$$

- Objective: a misclassification loss

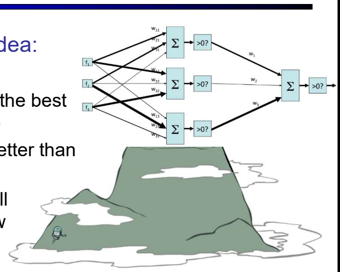
$$\min_w \sum_{i=1}^{m} \left(y^{(i)} - h_w(f(x^{(i)}))\right)^2$$

- Procedure:
  - Gradient descent / hill climbing

Slide credit: Pieter Abeel and Dan Klein

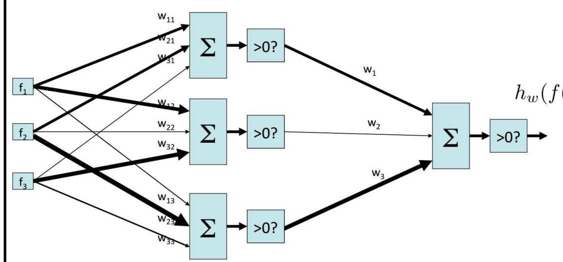
Hill climbing

- Simple, general idea:
  - Start wherever
  - Repeat: move to the best neighboring state
  - If no neighbors better than current, quit
  - Neighbors = small perturbations of w
- What's bad?
  - Complete?
  - Optimal?



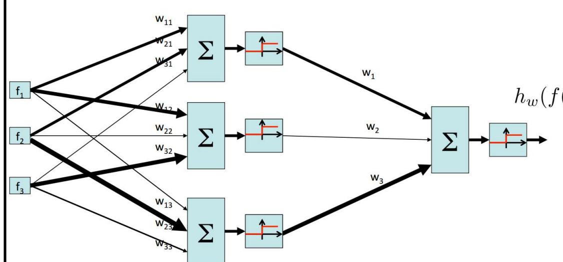
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



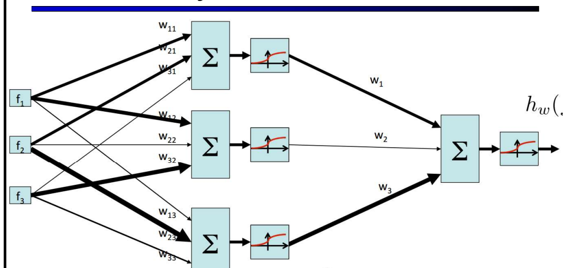
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Two-layer neural network



$$z \to \frac{1}{1+e^{-z}}$$

Slide credit: Pieter Abeel and Dan Klein

Neural network properties

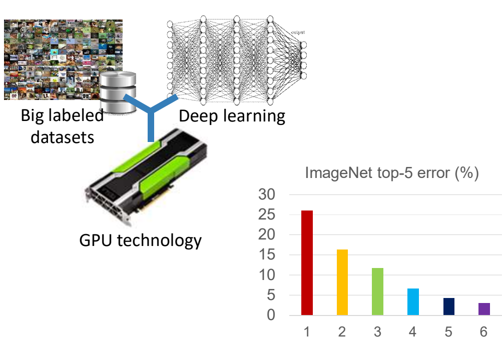
- Theorem (Universal function approximators): A two-layer network with a sufficient number of neurons can approximate any continuous function to any desired accuracy
- Practical considerations:
  - Can be seen as learning the features
  - Large number of neurons
    - Danger for overfitting
  - Hill-climbing procedure can get stuck in bad local optima

Approximation by Superpositions of Sigmoidal Function,1989 Slide credit: Pieter Abeel and Dan Klein

Today

- (Deep) Neural networks
- **Convolutional neural networks**

Significant recent impact on the field



Slide credit: Dinesh Jayaraman

Convolutional Neural Networks (CNN, ConvNet, DCN)

- CNN = a multi-layer neural network with
  - **Local** connectivity:
    - Neurons in a layer are only connected to a small region of the layer before it
  - **Share** weight parameters across spatial positions:
    - Learning shift-invariant filter kernels

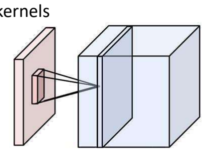
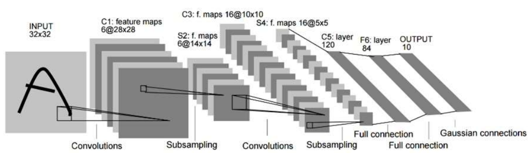


Image credit: A. Karpathy

Jia-Bin Huang and Derek Hoiem, UIUC

LeNet [LeCun et al. 1998]



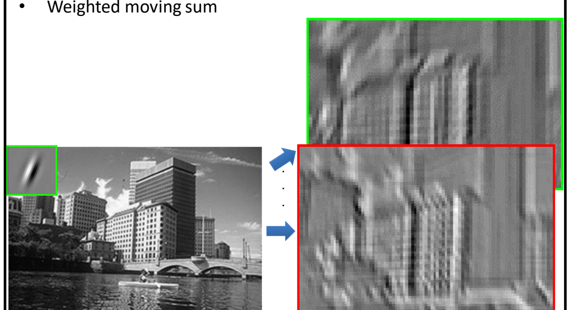
LeNet-1 from 1993

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

Jia-Bin Huang and Derek Hoiem, UIUC

What is a Convolution?

- Weighted moving sum



Input

Feature Activation Map

slide credit: S. Lazebnik

Convolutional Neural Networks

Feature maps

Normalization

Spatial pooling

Non-linearity

Convolution (Learned)

Input Image

slide credit: S. Lazebnik

Convolutional Neural Networks

Feature maps

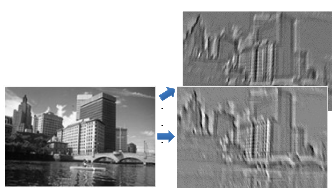
Normalization

Spatial pooling

Non-linearity

Convolution (Learned)

Input Image



slide credit: S. Lazebnik

Convolutional Neural Networks

Feature maps

Normalization

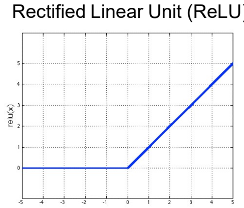
Spatial pooling

Non-linearity

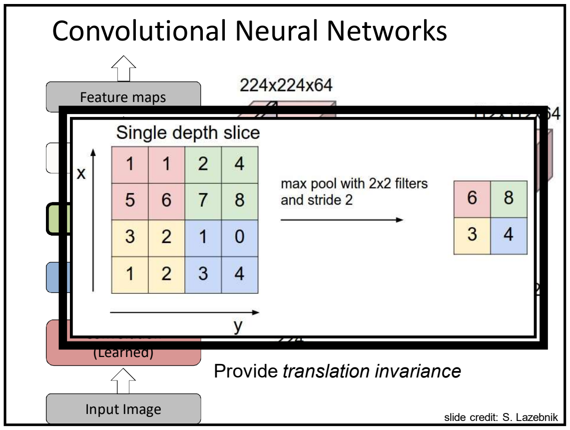
Convolution (Learned)

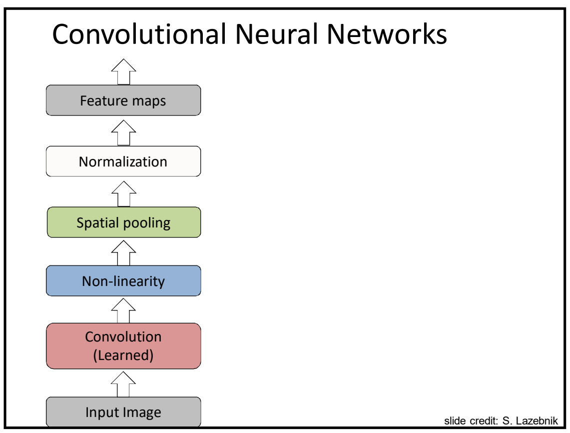
Input Image

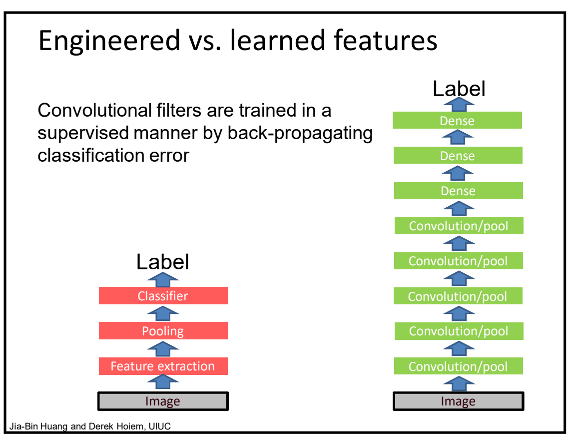
Rectified Linear Unit (ReLU)



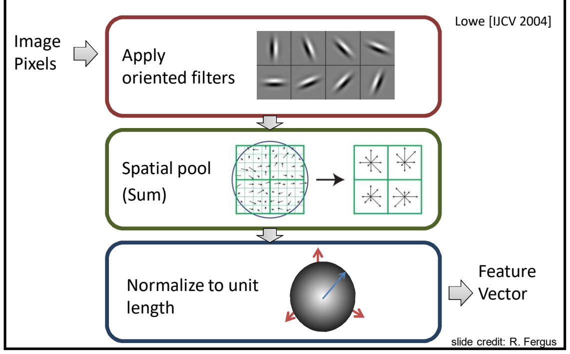
slide credit: S. Lazebnik


Convolutional Neural Networks
Feature maps
224x224x64
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
(Learned)
Provide *translation invariance*
Input Image
slide credit: S. Lazebnik

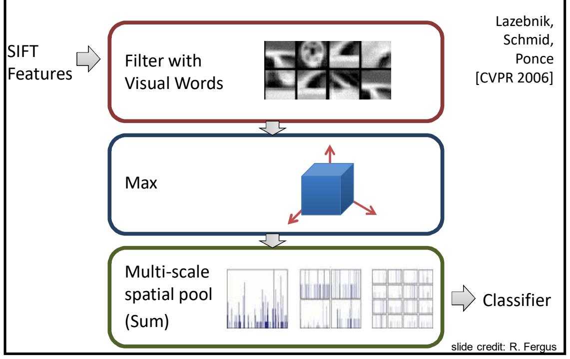

Convolutional Neural Networks
Feature maps
Normalization
Spatial pooling
Non-linearity
Convolution (Learned)
Input Image
slide credit: S. Lazebnik


Engineered vs. learned features
Convolutional filters are trained in a supervised manner by back-propagating classification error
Label
Classifier
Pooling
Feature extraction
Image
Label
Dense
Dense
Dense
Convolution/pool
Convolution/pool
Convolution/pool
Convolution/pool
Convolution/pool
Image
Jia-Bin Huang and Derek Hoiem, UIUC

SIFT Descriptor

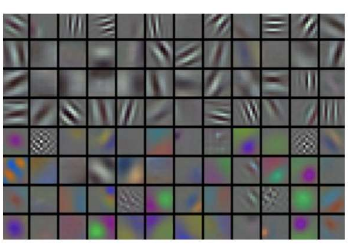


Spatial Pyramid Matching



Visualizing what was learned

- What do the learned filters look like?



Typical first layer filters

WIRED

Google's Artificial Brain Learns to Find Cat Videos

SHARE

SHARE 778

TWEET

COMMENT 0

EMAIL

BY LIAT CLARK. *Wired UK*

When computer scientists at Google's mysterious X lab built a neural network of 16,000 computer processors with one billion connections and let it browse YouTube, it did what

https://www.wired.com/2012/06/google-x-neural-network/

Application: ImageNet

[Deng et al. CVPR 2009]

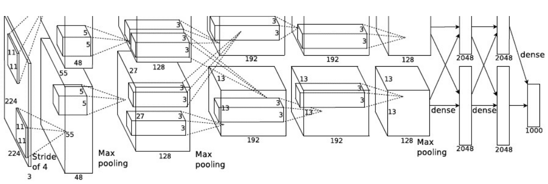
- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

https://sites.google.com/site/deeplearningcvpr2014

Slide: R. Fergus

AlexNet

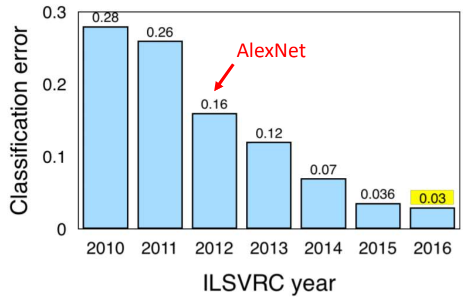
- Similar framework to LeCun'98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data (10^6 vs. 10^3 images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week



A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

Jia-Bin Huang and Derek Hoiem, UIUC

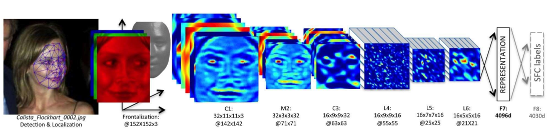
ImageNet Classification Challenge



http://image-net.org/challenges/talks/2016/ILSVRC2016_10_09_clsloc.pdf

Industry Deployment

- Used in Facebook, Google, Microsoft
- Image Recognition, Speech Recognition, ….
- Fast at test time



Taigman et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR'14

Slide: R. Fergus

Recap

- Neural networks / multi-layer perceptrons
  - View of neural networks as learning hierarchy of features
- Convolutional neural networks
  - Architecture of network accounts for image structure
  - "End-to-end" recognition from pixels
  - Together with big (labeled) data and lots of computation → major success on benchmarks, image classification and beyond