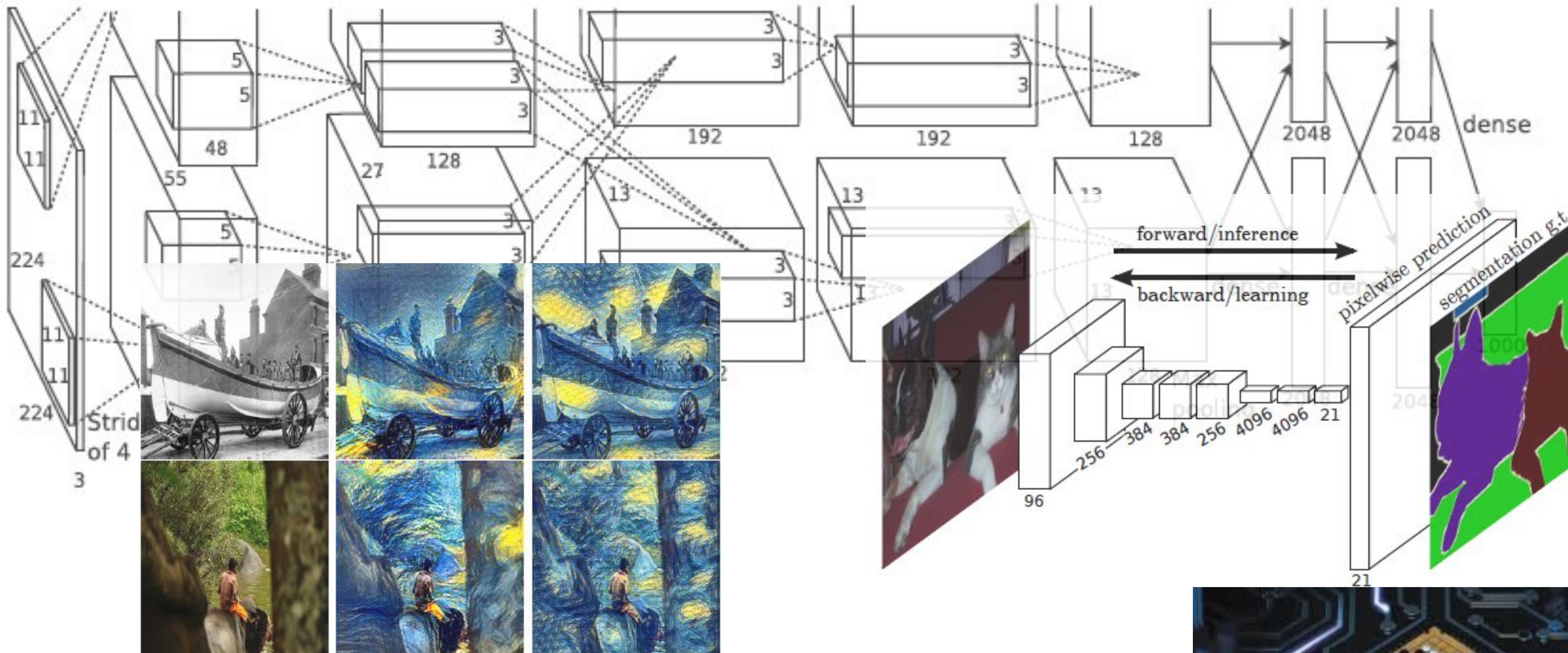
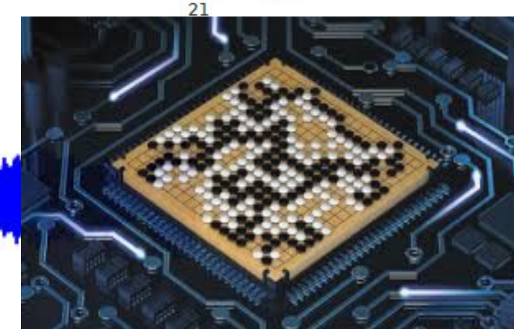
Lecture:

Introduction to Learning Based Vision

The modern world of machine learning





Figures from Krizhevsky et al., Shelhamer et al, Johnson et al, van den Oord et al, Silver et al.

Acquisitions

Google snaps up object recognition startup DNNresearch

Google has acquired a research startup founded within the University of Toronto, whose work includes object recognition.

by Josh Lowensohn @Josh / 13 March 2013, 9:22 am AEDT

2 / 0 / 0 / 0 / / more +

Google has acquired a three-person Canadian research company that specializes in voice and image recognition.

DNNresearch, which was founded last year within the the University of Toronto's computer science department, specializes in object recognition and now belongs to Google.

From left: Ilya Sutskever, Alex Krizhevsky and University Professor Geoffrey Hinton of the University of Toronto's Department of Computer Science. *(photo by John Guatto, University of Toronto)*

Acquisitions

Google snaps up object recognition startup DNNresearch

Google has acquired a research startup founded within the University of Toronto, whose work includes object recognition.

by Josh Lowensohn @Josh / 13 March 2013, 9:22 am AEDT

2 / 0 / 0 / 0 / / more +

Google has acquired a three-person Canadian research company that specializes in voice and image recognition.

DNNresearch, which was founded last year within the the University of Toronto's computer science department, specializes in object recognition and now belongs to Google.

Google Acquires Artificial Intelligence Startup DeepMind For More Than $500M

Posted Jan 26, 2014 by Catherine Shu (@catherineshu)

11.3k SHARES

Next Story

DEEPMIND

CrunchBase

DeepMind

FOUNDED
2011

OVERVIEW
DeepMind is a cutting edge artificial intelligence company. We combine the best techniques from machine learning and systems neuroscience to build powerful general-purpose learning algorithms. Founded by Demis Hassabis, Shane Legg and Mustafa Suleyman, the company is based in London and supported by some of the most iconic technology entrepreneurs and investors of the past decade. Our first commercial ...

Slide by Dhruv Batra

Acquisitions

Google snaps up object recognition startup DNNresearch

Google has acquired a research startup founded within the University of Toronto, whose work includes object recognition.

by Josh Lowensohn @Josh / 13 March 2013, 9:22 am AEDT

2 / 0 / 0 / 0 / / more +

Google has acquired a three-person Canadian research company that specializes in voice and image recognition.

DNNresearch, which was founded last year within

8TH ANNUAL CRUNCHIES AWARDS Celebrate the Best of Tech in 2014 Get Your Tickets Now

Google Acquires Artificial Intelligence Startup DeepMind

Yann LeCun

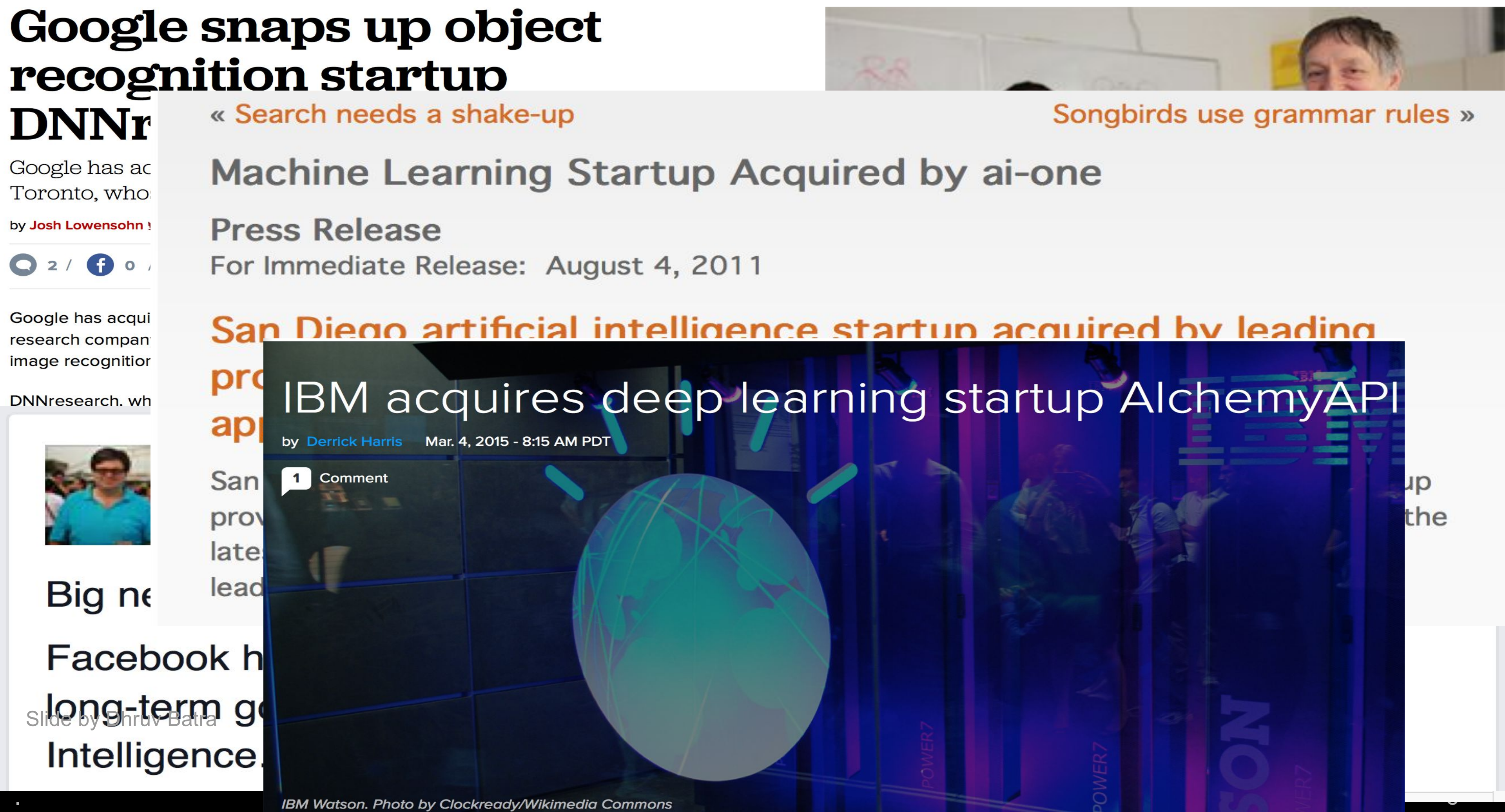
December 9, 2013 ·

Big news today!

Facebook has created a new research laboratory with the ambitious, long-term goal of bringing about major advances in Artificial Intelligence.

Slide by Dhruv Batra

Acquisitions


Google snaps up object recognition startup DNNr

Google has ac
Toronto, who

by Josh Lowensohn

2 / 0

Google has acqui
research compan
image recognitior

DNNresearch. wh

Big ne

Facebook h
long-term g
Intelligence.

« Search needs a shake-up

Songbirds use grammar rules »

Machine Learning Startup Acquired by ai-one

Press Release
For Immediate Release: August 4, 2011

San Diego artificial intelligence startup acquired by leading
pro
ap

San
prov
late
lead

up
the

IBM acquires deep learning startup AlchemyAPI

by Derrick Harris Mar. 4, 2015 - 8:15 AM PDT

1 Comment

IBM Watson. Photo by Clockready/Wikimedia Commons

Slide by Dhruv Batra

What is ~~Machine~~ Learning?

- “the acquisition of knowledge or skills through experience, study, or by being taught.”

- Can be (almost) mapped to reinforcement, unsupervised and supervised machine learning.

Slide by Dhruv Batra

What is Machine Learning?

- [Arthur Samuel, 1959]
  - Field of study that gives computers
  - the ability to learn without being explicitly programmed
- [Kevin Murphy] algorithms that
  - automatically detect patterns in data
  - use the uncovered patterns to predict future data or other outcomes of interest
- [Tom Mitchell] algorithms that
  - improve their performance (P)
  - at some task (T)
  - with experience (E)

Slide by Dhruv Batra

What is Machine Learning?

Slide by Dhruv Batra

ML in Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year

- Decades of ML research oversimplified:
  - All of Machine Learning:
  - Learn a mapping from input to output $f: X \rightarrow Y$
    - e.g. X: emails, Y: {spam, notspam}

Slide by Dhruv Batra

Types of Learning

- Supervised learning
  - Training data includes desired outputs
- Unsupervised learning
  - Training data does not include desired outputs
- Weakly or Semi-supervised learning
  - Training data includes a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

Slide by Dhruv Batra

Tasks

Supervised Learning

x → Classification → y Discrete

x → Regression → y Continuous

Unsupervised Learning

x → Clustering → y Discrete ID

x → Dimensionality Reduction → y Continuous

Slide by Dhruv Batra

Examples for supervised learning

Vision: Image Classification

- http://cloudcv.org/classify/

x → y

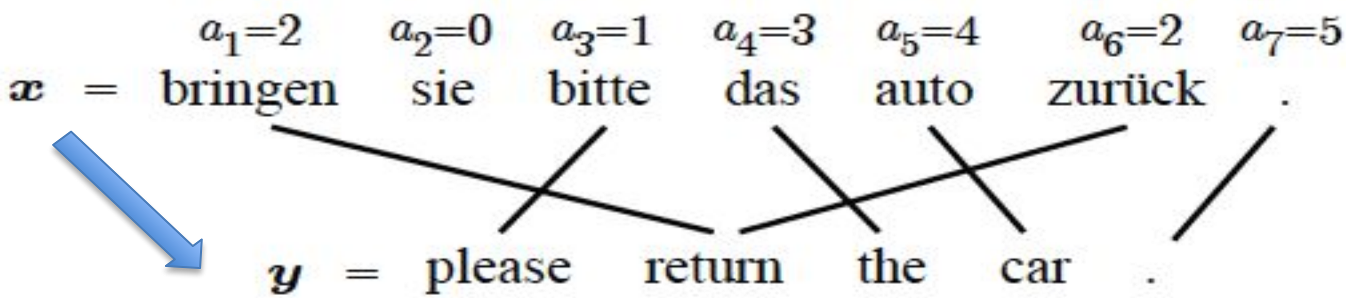
scuba diver

tiger shark

hammerhead shark

Slide by Dhruv Batra

NLP: Machine Translation



Slide by Dhruv Batra

Speech: Speech2Text

Slide by Dhruv Batra

Image captioning

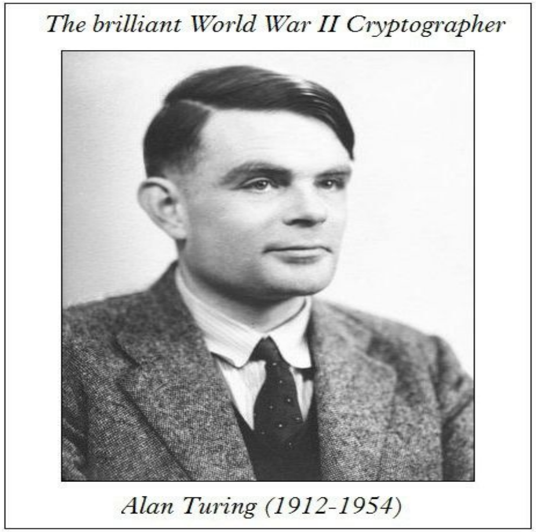
"woman is holding bunch of bananas."

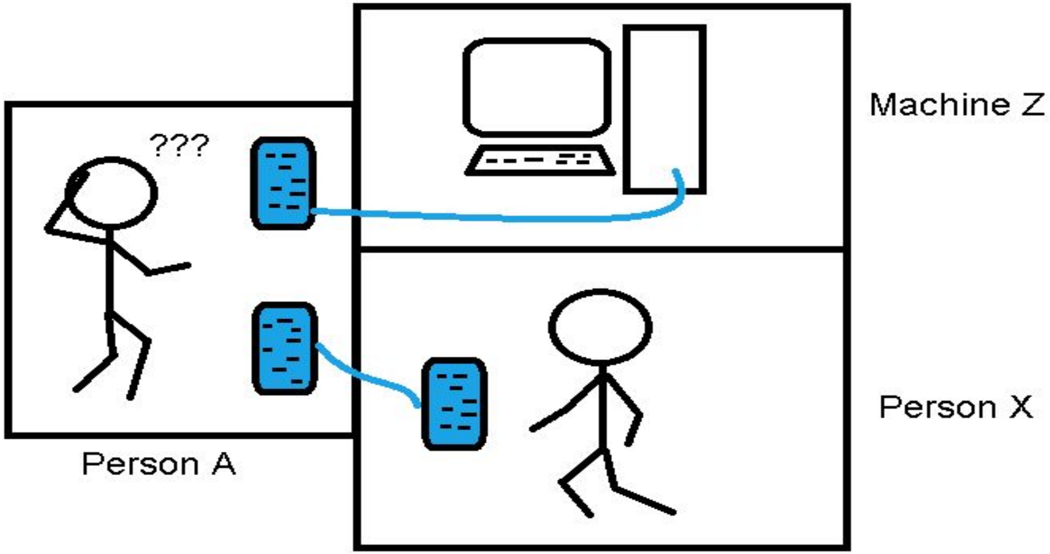
"black cat is sitting on top of suitcase."

http://cs.stanford.edu/people/karpathy/deepimagesent/

AI: Turing Test

"Can machines think"





Q: Please write me a sonnet on the subject of the Forth Bridge.
A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.
A: (Pause about 30 seconds and then give as answer) 105621.

Slide by Dhruv Batra

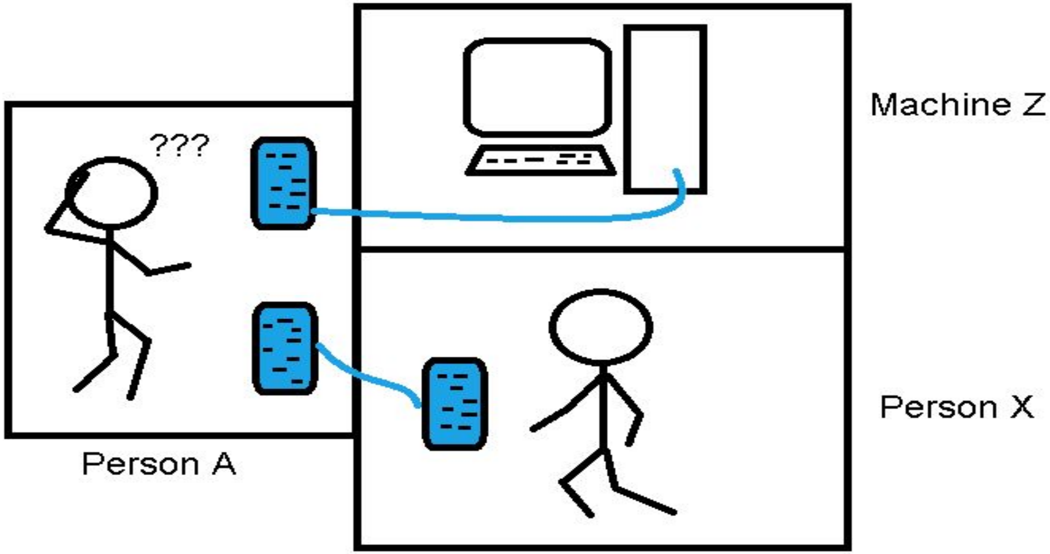
AI: Visual Turing Test

x

Q: How many slices of pizza are there?

y

A: 6



Slide by Dhruv Batra

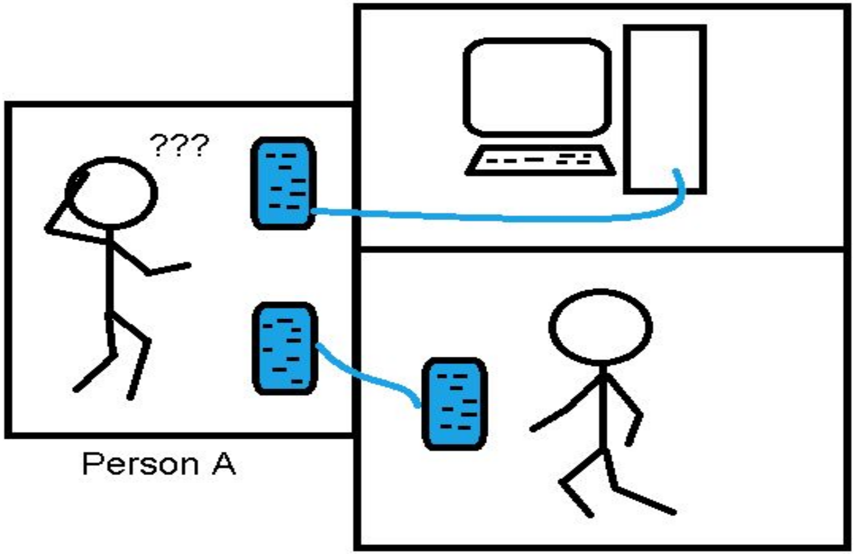
AI: Visual Turing Test

Q: How many slices of pizza are there?

x

y

A: 6



Demo: http://cloudcv.org/vqa/

Slide by Dhruv Batra

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)

Slide by Dhruv Batra

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)
- (Unknown) Target Function
  - f: $X \rightarrow Y$ (the "true" mapping / reality)

Slide by Dhruv Batra

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)
- (Unknown) Target Function
  - f: $X \rightarrow Y$ (the "true" mapping / reality)
- Data: $(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)$

Slide by Dhruv Batra

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)
- (Unknown) Target Function
  - f: X → Y (the "true" mapping / reality)
- Data: $(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)$
- Model / Hypothesis Class
  - g: X → Y
  - $y = g(x) = \text{sign}(w^T x)$
- Learning = Search in hypothesis space
  - Find best g in model class.

Slide by Dhruv Batra

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)
- (Unknown) Target Function
  - f: $X \rightarrow Y$ (the "true" mapping / reality)
- Data: $(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)$
- Model / Hypothesis Class
  - g: $X \rightarrow Y$
  - $y = g(x) = \text{sign}(w^T x)$
- Learning = Search in hypothesis space
  - Find best g in model class.

Slide by Dhruv Batra

Synonyms

- Representation Learning
- Deep (Machine) Learning
- Deep Neural Networks
- Deep Unsupervised Learning
- Simply: Deep Learning

Slide by Dhruv Batra

So what *is* Deep (Machine) Learning?

- A few different ideas:

- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extraction

- Distributed Representations
  - No single neuron "encodes" everything
  - Groups of neurons work together

Slide by Dhruv Batra

Incoming slides and lectures

- Some fundamentals of machine learning
  - A traditional image classification pipeline (Use raw pixels as features and nearest neighbor classifier)
  - Linear classification
  - Loss functions
  - Training by numerical optimization

- Then, deep learning will be a natural extension..

A simple image classification pipeline

Image Classification: a core task in Computer Vision

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

→ cat

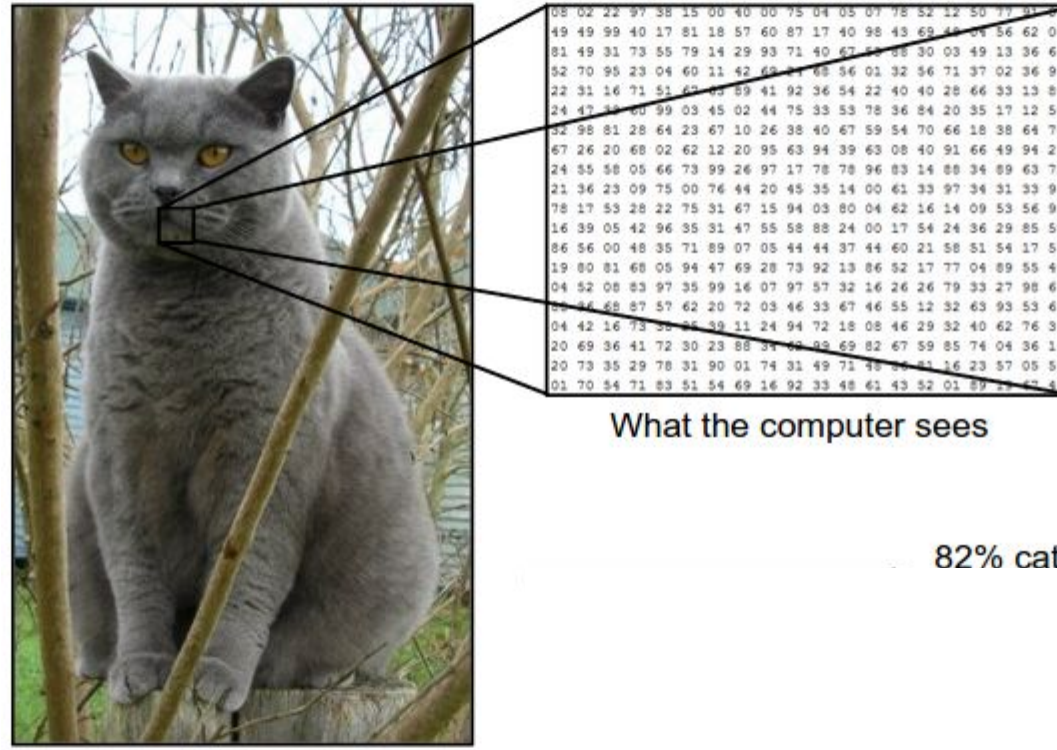
Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

The problem:
semantic gap

Images are represented as 3D arrays of numbers, with integers between [0, 255].

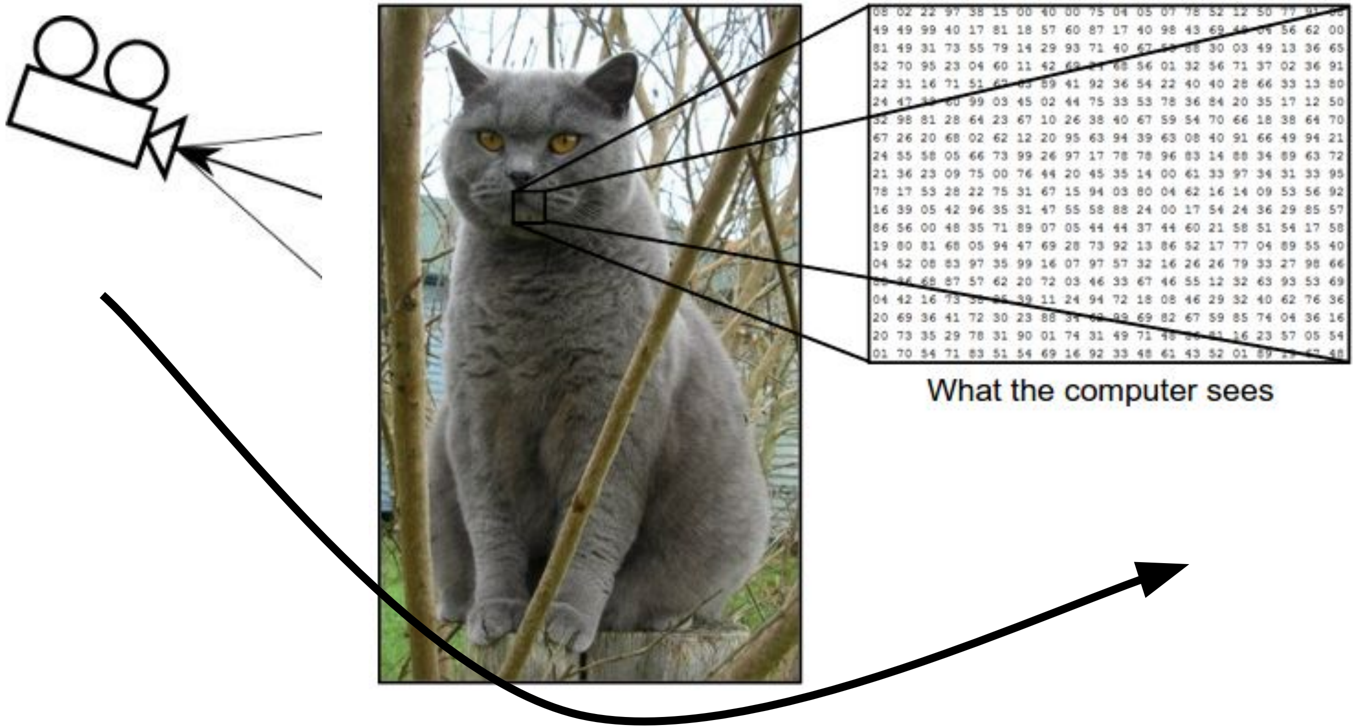
E.g.
300 x 100 x 3

(3 for 3 color channels RGB)



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Challenges: Viewpoint Variation

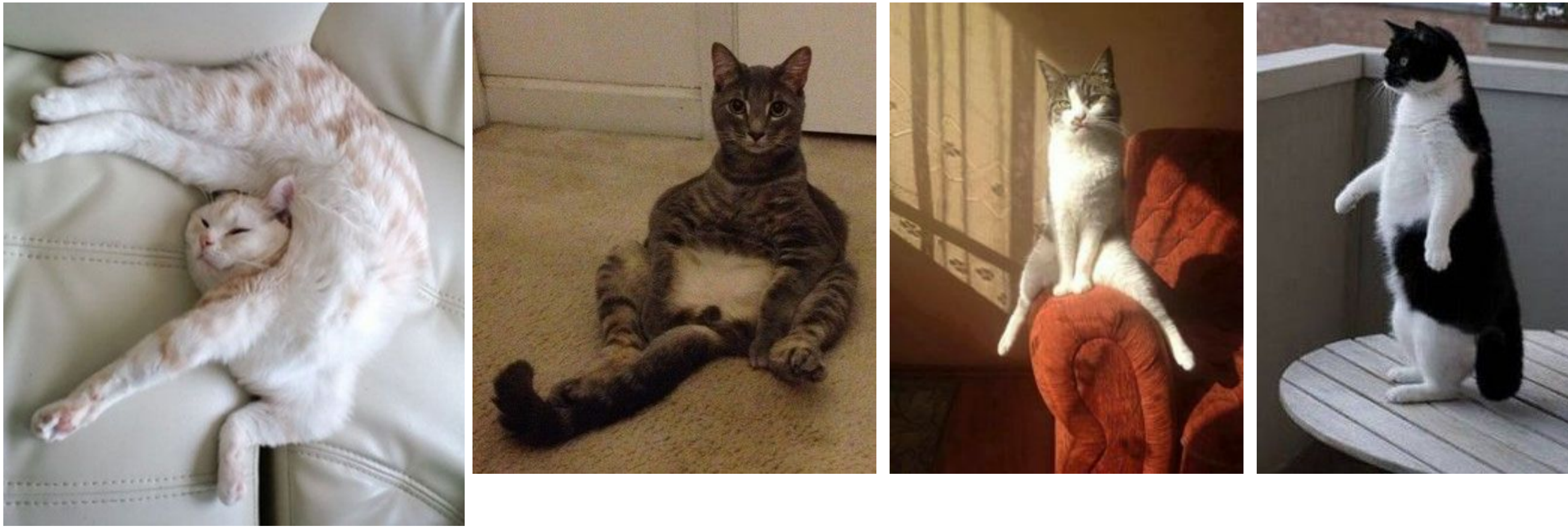


Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Challenges: Illumination

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Challenges: Deformation



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Challenges: Occlusion



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

35

Challenges: Background clutter

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Challenges: Intraclass variation

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

37

An image classifier

```
def predict(image):
    # ????
    return class_label
```

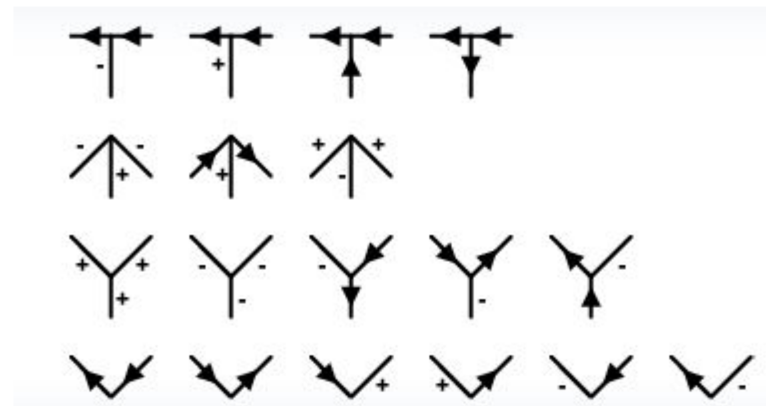
Unlike e.g. sorting a list of numbers,

no obvious way to hard-code the algorithm for recognizing a cat, or other classes.

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Attempts have been made

???



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Data-driven approach:

1. Collect a dataset of images and labels
2. Use Machine Learning to train an image classifier
3. Evaluate the classifier on a withheld set of test images

```python
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

Example training set

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

First classifier: **Nearest Neighbor Classifier**

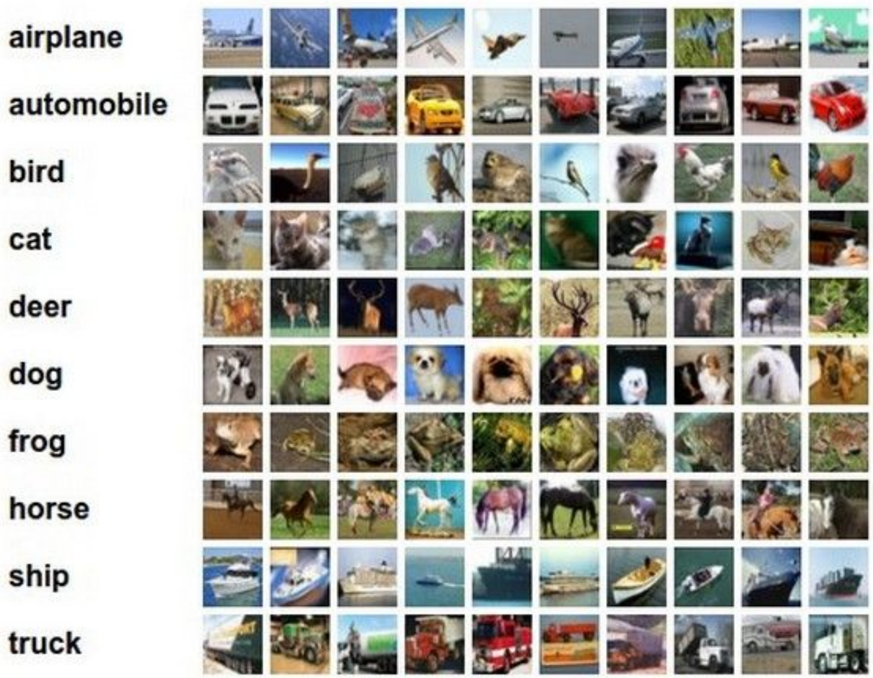
```
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

Remember all training images and their labels

Predict the label of the most similar training image

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

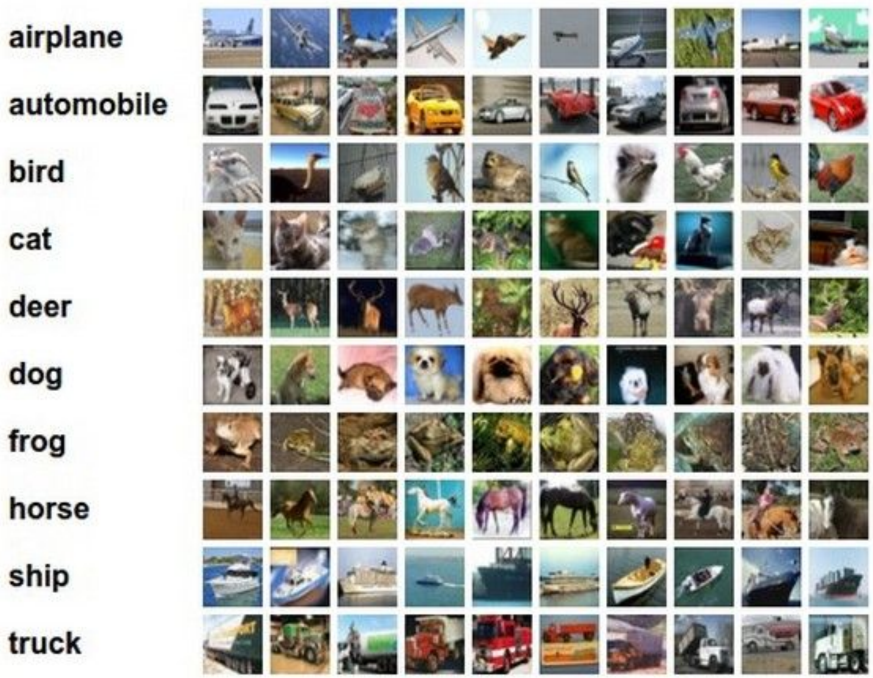
Example dataset: **CIFAR-10**
10 labels
50,000 training images, each image is tiny: 32x32
10,000 test images.



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Example dataset: **CIFAR-10**
10 labels
50,000 training images
10,000 test images.

For every test image (first column), examples of nearest neighbors in rows



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

How do we compare the images? What is the **distance metric**?

L1 distance:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

$$\left| \begin{array}{c} \text{test image} \\ \begin{array}{|c|c|c|c|} \hline 56 & 32 & 10 & 18 \\ \hline 90 & 23 & 128 & 133 \\ \hline 24 & 26 & 178 & 200 \\ \hline 2 & 0 & 255 & 220 \\ \hline \end{array} \end{array} - \begin{array}{c} \text{training image} \\ \begin{array}{|c|c|c|c|} \hline 10 & 20 & 24 & 17 \\ \hline 8 & 10 & 89 & 100 \\ \hline 12 & 16 & 178 & 170 \\ \hline 4 & 32 & 233 & 112 \\ \hline \end{array} \end{array} \right| = \begin{array}{c} \text{pixel-wise absolute value differences} \\ \begin{array}{|c|c|c|c|} \hline 46 & 12 & 14 & 1 \\ \hline 82 & 13 & 39 & 33 \\ \hline 12 & 10 & 0 & 30 \\ \hline 2 & 32 & 22 & 108 \\ \hline \end{array} \end{array} \xrightarrow{\text{add}} 456$$

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Nearest Neighbor classifier

```
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Nearest Neighbor classifier

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

remember the training data

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Nearest Neighbor classifier

for every test image:

- find nearest train image with L1 distance
- predict the label of nearest training image

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Nearest Neighbor classifier

Q: how does the classification speed depend on the size of the training data?

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Nearest Neighbor classifier

Q: how does the classification speed depend on the size of the training data?
linearly :(

This is **backwards**:
- test time performance is usually much more important in practice.
- CNNs flip this: expensive training, cheap test evaluation

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Aside: Approximate Nearest Neighbor

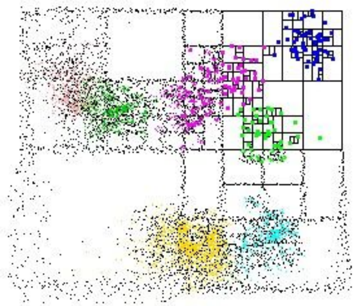
find approximate nearest neighbors quickly

ANN: A Library for Approximate Nearest Neighbor Searching

David M. Mount and Sunil Arya

Version 1.1.2

Release Date: Jan 27, 2010



What is ANN?

ANN is a library written in C++, which supports data structures and algorithms for both exact and approximate nearest neighbor searching in arbitrarily high dimensions.

In the nearest neighbor problem a set of data points in d-dimensional space is given. These points are preprocessed into a data structure, so that given any query point q, the nearest or generally k nearest points of P to q can be reported efficiently. The distance between two points can be defined in many ways. ANN assumes that distances are measured using any class of distance functions called Minkowski metrics. These include the well known Euclidean distance, Manhattan distance, and max distance.

Based on our own experience, ANN performs quite efficiently for point sets ranging in size from thousands to hundreds of thousands, and in dimensions as high as 20. (For applications in significantly higher dimensions, the results are rather spotty, but you might try it anyway.)

The library implements a number of different data structures, based on kd-trees and box-decomposition trees, and employs a couple of different search strategies.

The library also comes with test programs for measuring the quality of performance of ANN on any particular data sets, as well as programs for visualizing the structure of the geometric data structures.

FLANN - Fast Library for Approximate Nearest Neighbors

- Home
- News
- Publications
- Download
- Changelog
- Repository

What is FLANN?

FLANN is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. It contains a collection of algorithms we found to work best for nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset.

FLANN is written in C++ and contains bindings for the following languages: C, MATLAB and Python.

News

- (14 December 2012) Version 1.8.0 is out bringing incremental addition/removal of points to/from indexes
- (20 December 2011) Version 1.7.0 is out bringing two new index types and several other improvements.
- You can find binary installers for FLANN on the Point Cloud Library project page. Thanks to the PCL developers!
- Mac OS X users can install flann though MacPorts (thanks to Mark Moll for maintaining the Portfile)
- New release introducing an easier way to use custom distances, kd-tree implementation optimized for low dimensionality search and experimental MPI support
- New release introducing new C++ templated API, thread-safe search, save/load of indexes and more.
- The FLANN license was changed from LGPL to BSD.

How fast is it?

In our experiments we have found FLANN to be about one order of magnitude faster on many datasets (in query time), than previously available approximate nearest neighbor search software.

Publications

More information and experimental results can be found in the following papers:
- Marius Muja and David G. Lowe: **"Scalable Nearest Neighbor Algorithms for High Dimensional Data"**. Pattern Analysis and Machine Intelligence (PAMI), Vol. 36, 2014. [PDF] [BibTeX]
- Marius Muja and David G. Lowe: **"Fast Matching of Binary Features"**. Conference on Computer and Robot Vision (CRV) 2012. [PDF] [BibTeX]
- Marius Muja and David G. Lowe, **"Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration"**, in International Conference on Computer Vision Theory and Applications (VISAPP'09), 2009 [PDF] [BibTeX]

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

The choice of distance is a **hyperparameter**
common choices:

L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

L2 (Euclidean) distance

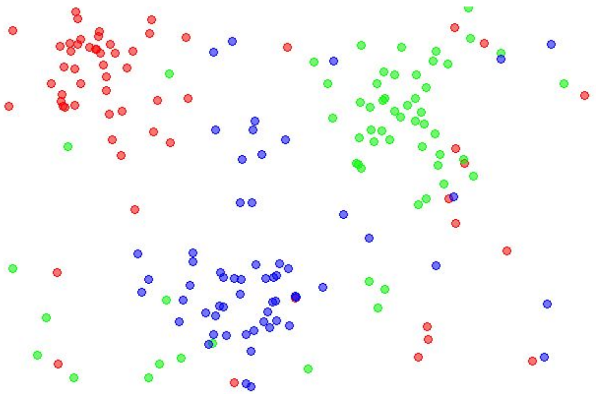
$$d_2(I_1, I_2) = \sqrt{\sum_p \left(I_1^p - I_2^p\right)^2}$$

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

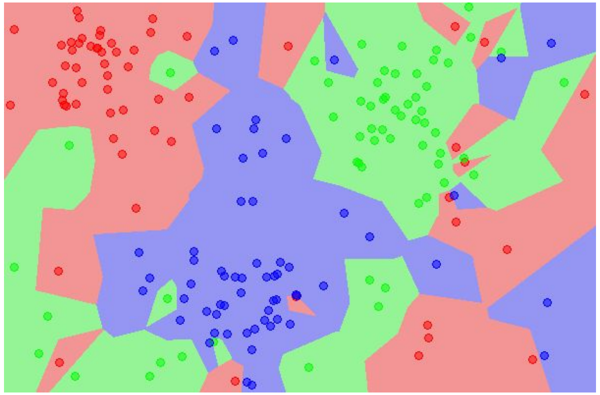
k-Nearest Neighbor

find the k nearest images, have them vote on the label

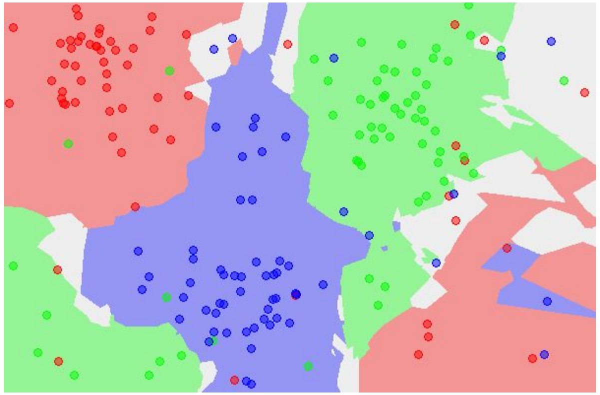
the data



NN classifier



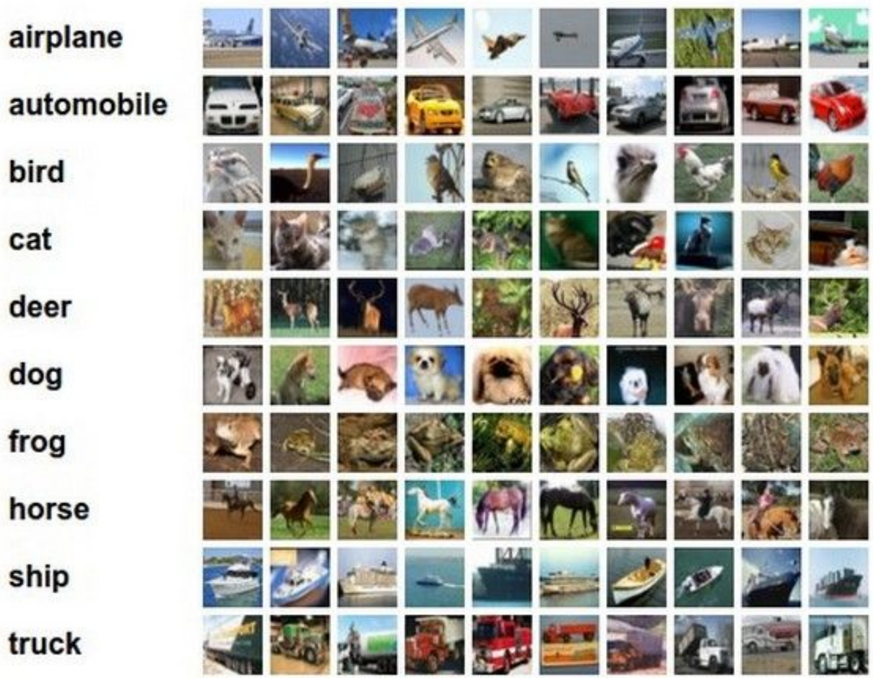
5-NN classifier



http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

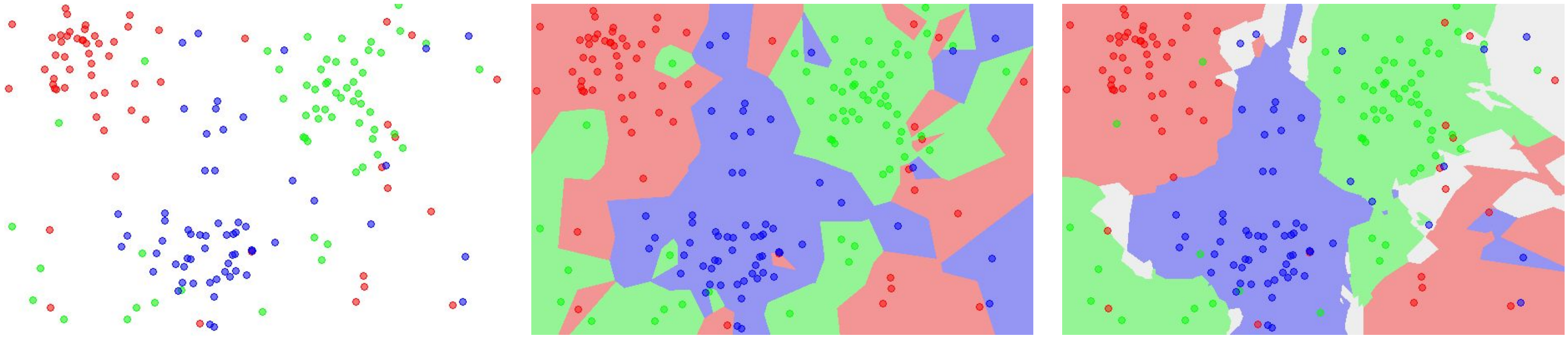
Example dataset: **CIFAR-10**
10 labels
50,000 training images
10,000 test images.



For every test image (first column), examples of nearest neighbors in rows

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

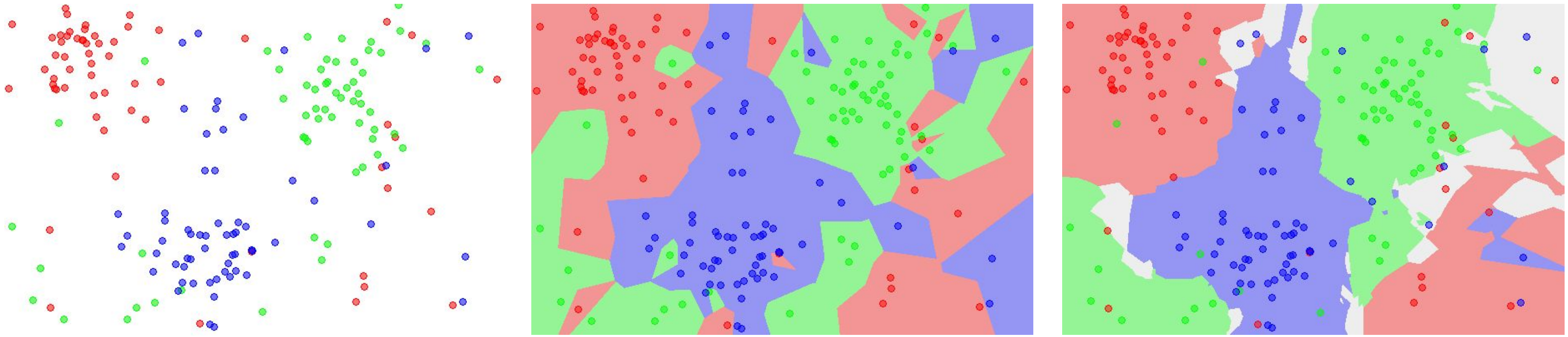
the data	NN classifier	5-NN classifier



Q: what is the accuracy of the nearest neighbor classifier on the training data, when using the Euclidean distance?

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

the data NN classifier 5-NN classifier



Q2: what is the accuracy of the **k-**nearest neighbor classifier on the training data?

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

What is the best **distance** to use?
What is the best value of **k** to use?

i.e. how do we set the **hyperparameters**?

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

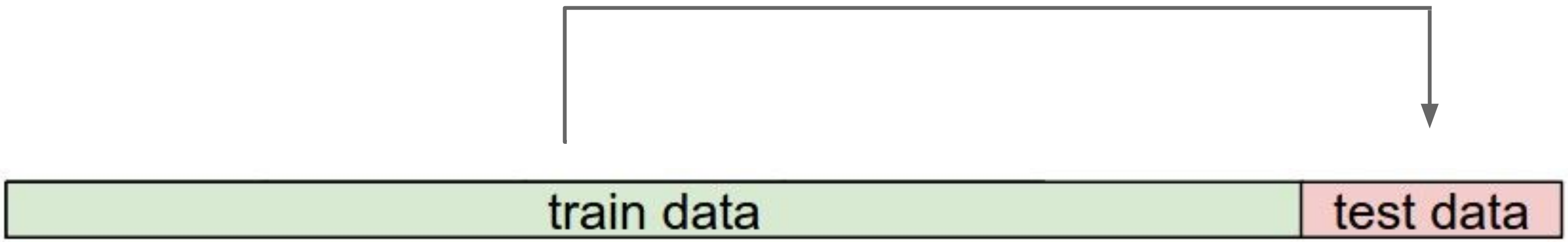
What is the best **distance** to use?
What is the best value of **k** to use?

i.e. how do we set the **hyperparameters**?

Very problem-dependent.
Must try them all out and see what works best.

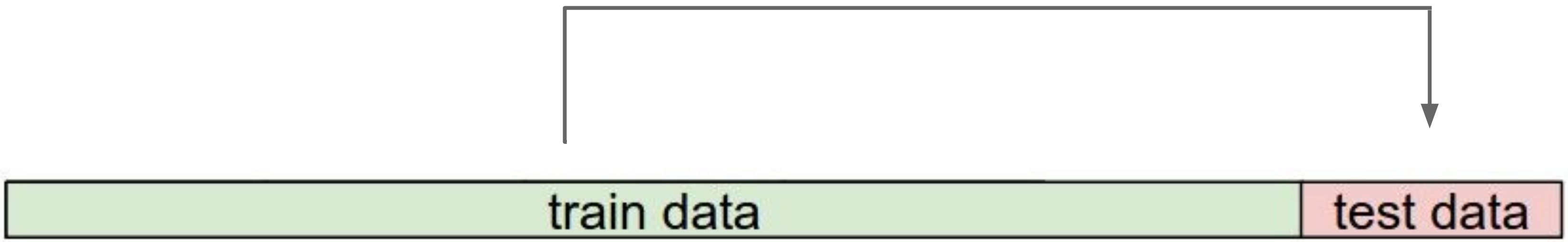
Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Try out what hyperparameters work best on test set.

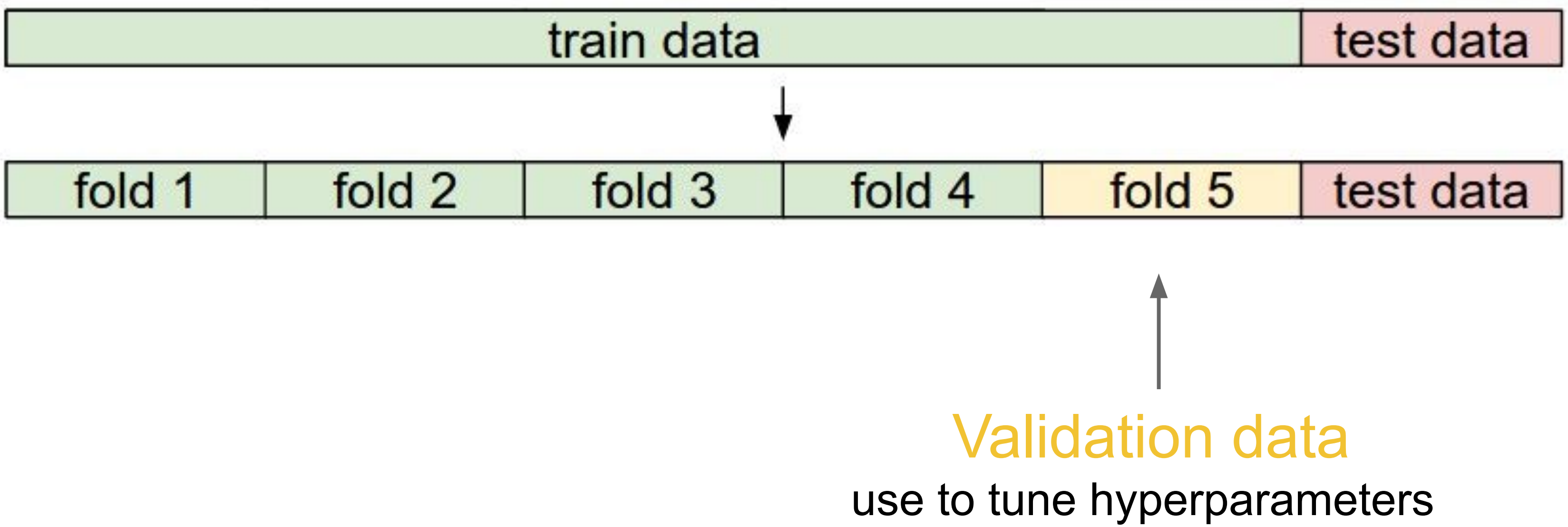


Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

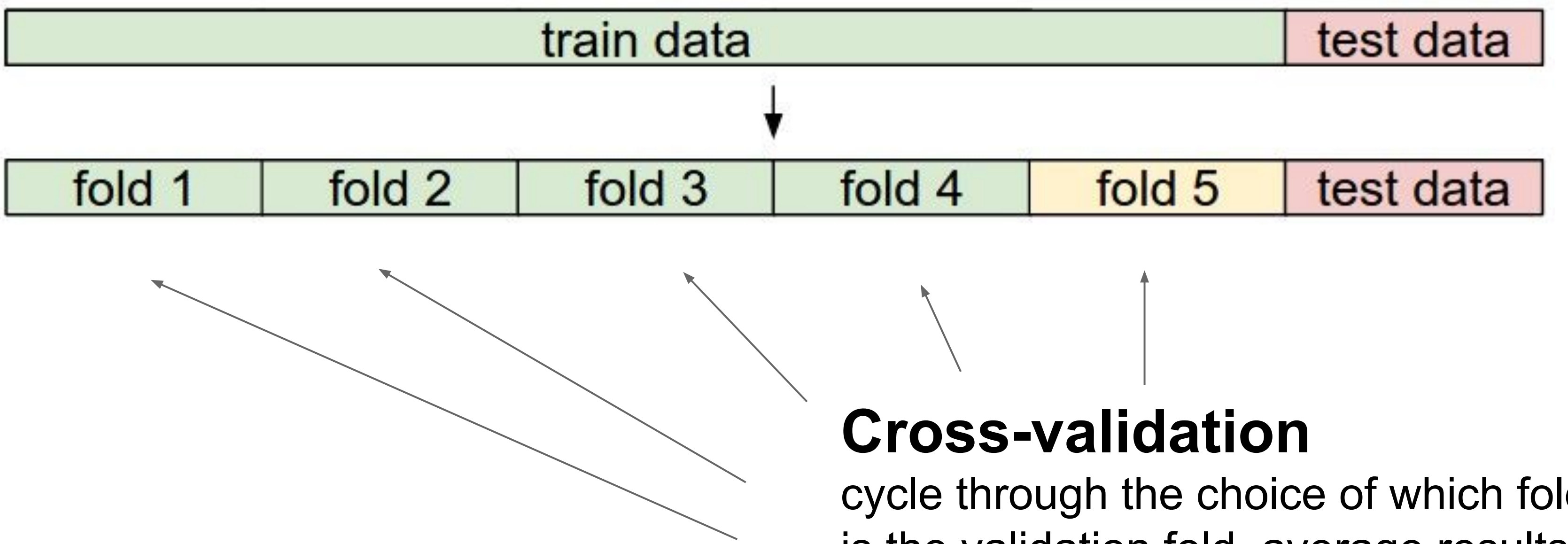
Trying out what hyperparameters work best on test set:
Very bad idea. The test set is a proxy for the generalization performance!
Use only **VERY SPARINGLY,** at the end.



Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson



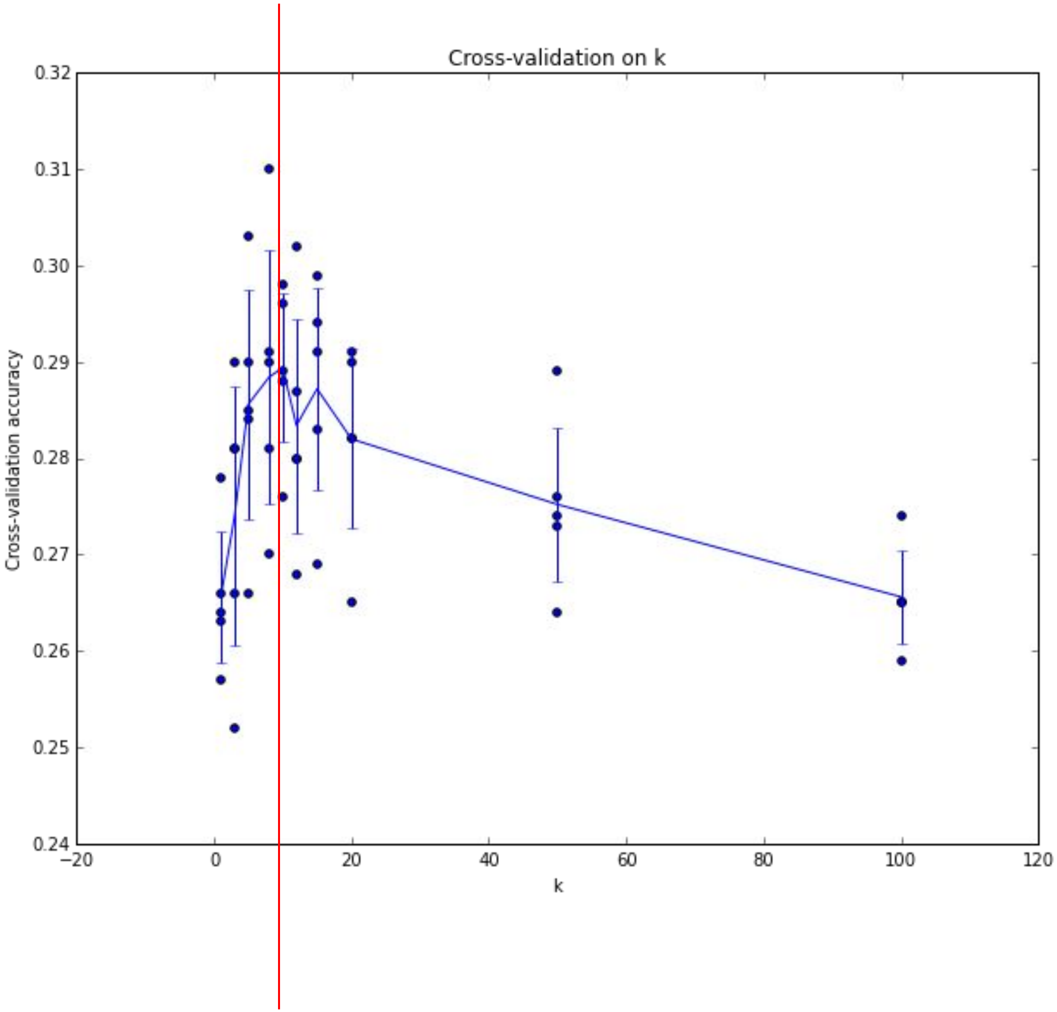
Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson



Cross-validation

cycle through the choice of which fold is the validation fold, average results.

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson



Example of
5-fold cross-validation
for the value of **k.**

Each point: single
outcome.

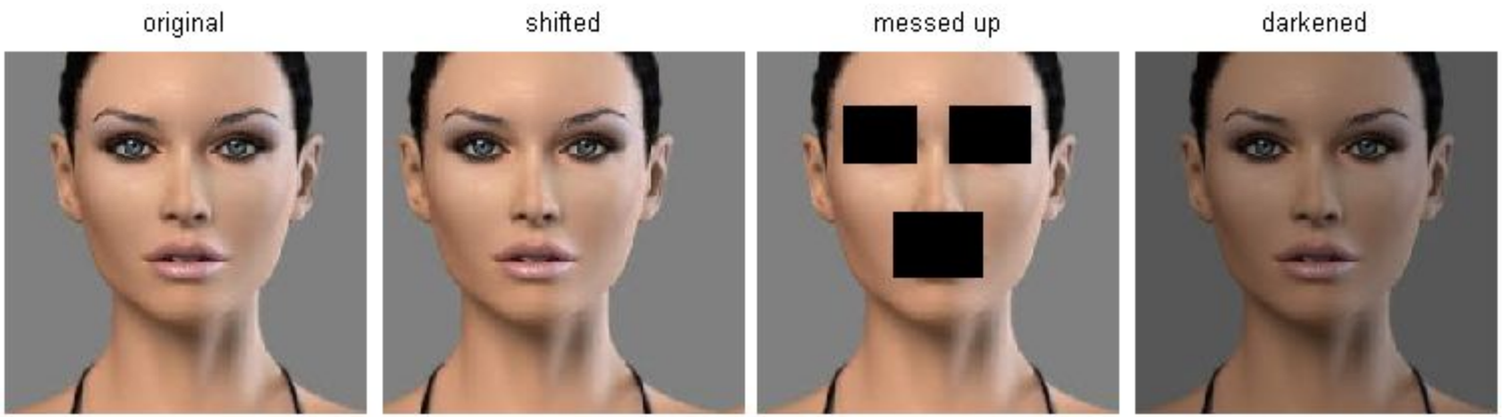
The line goes
through the mean, bars
indicated standard
deviation

(Seems that k ~= 7 works best
for this data)

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

k-Nearest Neighbor on images is **never used.**

- terrible performance at test time
- distance metrics on level of whole images can be very unintuitive



(all 3 images have same L2 distance to the one on the left)

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

Summary

- **Image Classification:** We are given a **Training Set** of labeled images, asked to predict labels on **Test Set**. Common to report the **Accuracy** of predictions (fraction of correctly predicted images)
- We introduced the **k-Nearest Neighbor Classifier**, which predicts the labels based on nearest images in the training set
- We saw that the choice of distance and the value of k are **hyperparameters** that are tuned using a **validation set**, or through **cross-validation** if the size of the data is small.
- Once the best set of hyperparameters is chosen, the classifier is evaluated once on the test set, and reported as the performance of kNN on that data.

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson