
Introduction
To
Machine Learning

Hi

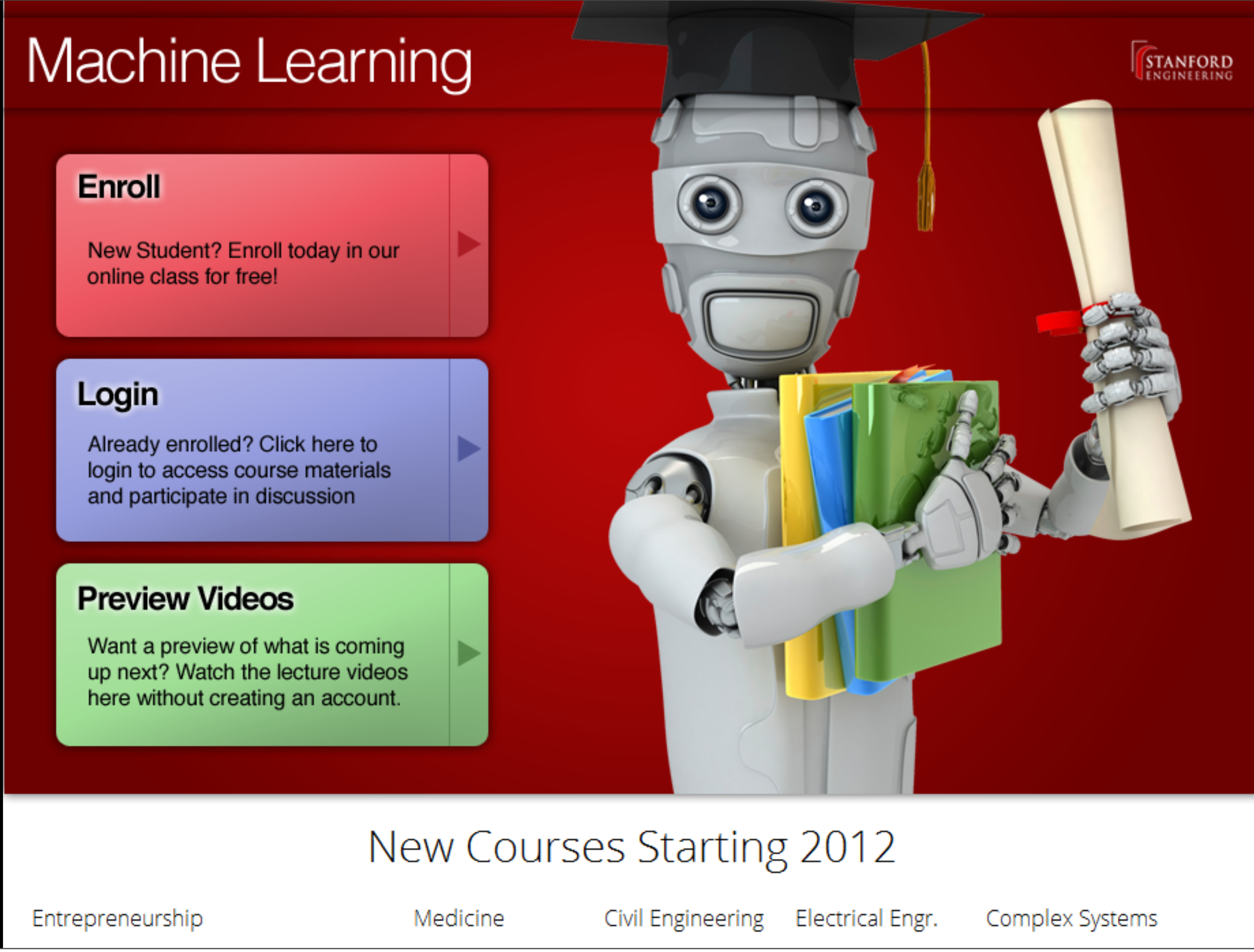
RealGeeks

Real Estate Web Solutions

Tuesday, June 26, 12

Use MongoDB to store millions of real estate properties for sale

Lots of data = fun with machine learning!



Tuesday, June 26, 12

https://www.coursera.org/

Arthur Samuel (1959):

Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed.

Tuesday, June 26, 12

Samuel was one of the pioneers of the machine learning field. He wrote a checkers program that learned from the games it played. The program became better at checkers than he was, but hewas pretty bad at checkers. Checkers is now solved.

Tom Mitchell (1998):

A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T, as measured by P, improves with experience E

Here’s a more formal definition.

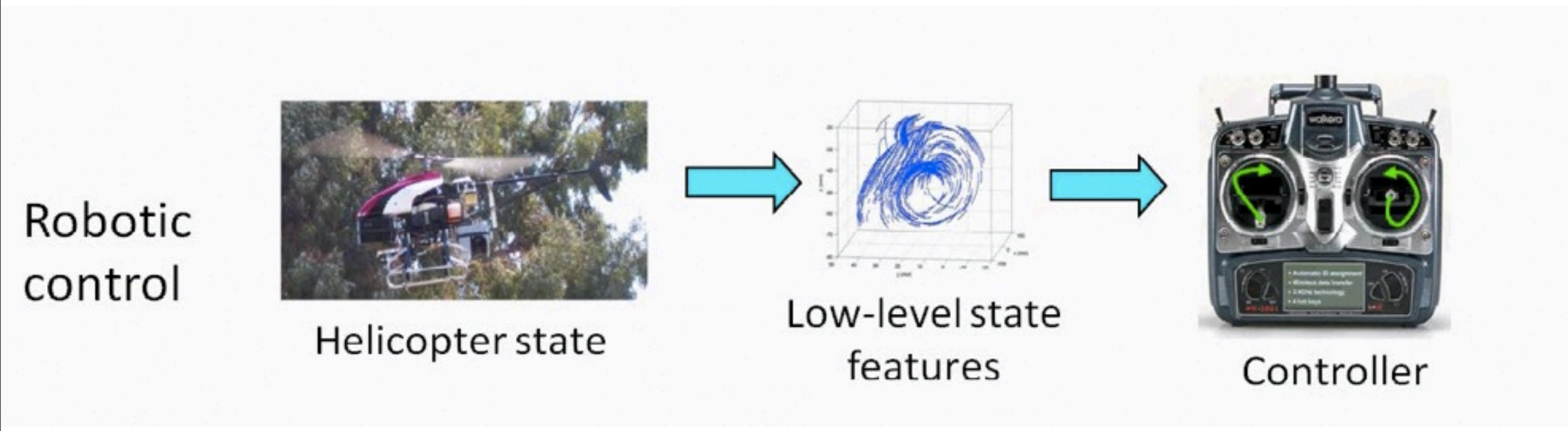
Tuesday, June 26, 12

The guy who taught my class is in the middle. I pretty much ripped off this entire talk from his class.

Here's a pretty cool application of machine learning.

Stanford taught these helicopters how to fly autonomously.

apprenticeship learning



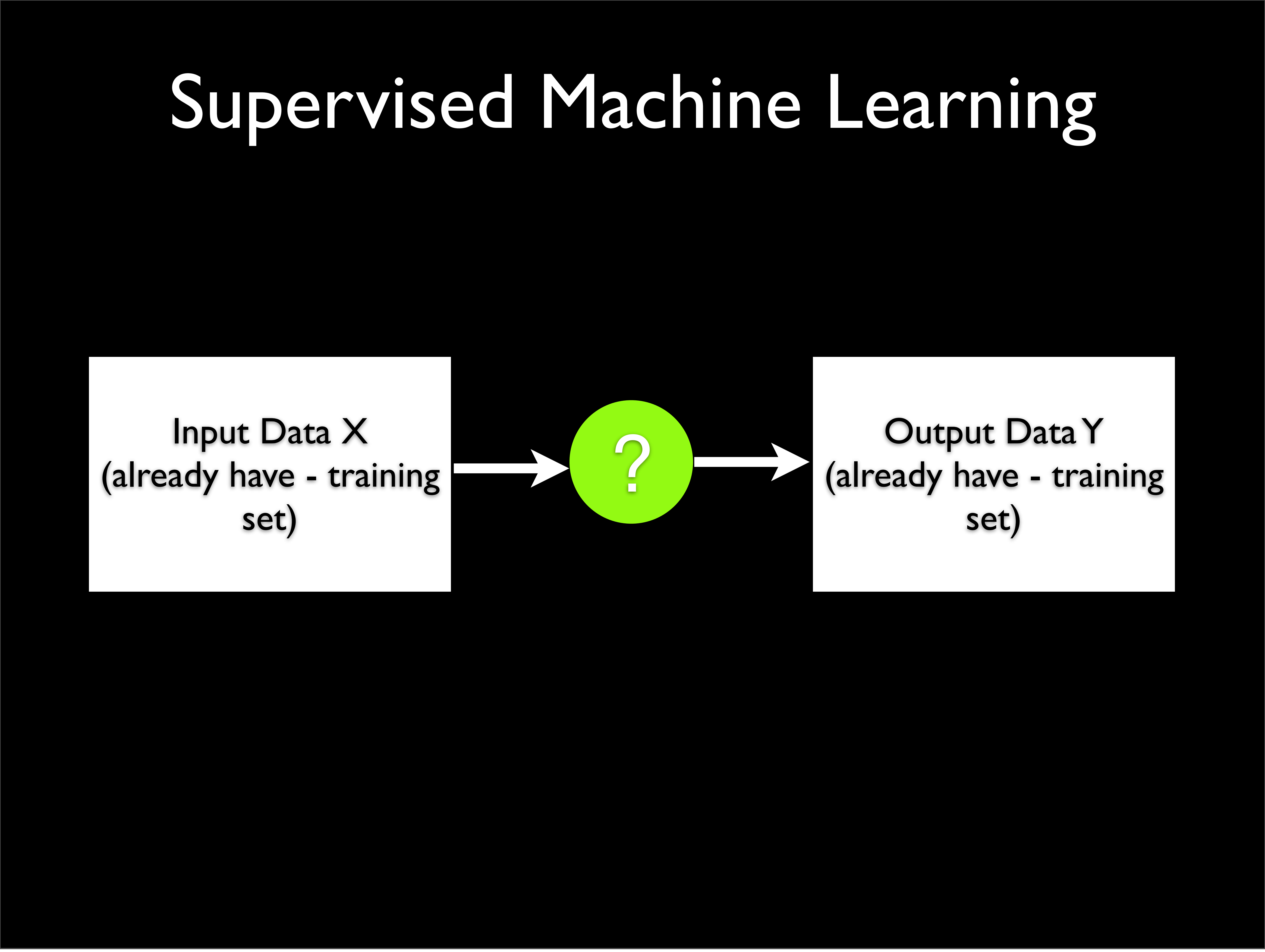
Old attempts at autonomous helicopters

build a model of world and helicopter state

create complex algorithm by hand that tells helicopter what to do

[Courtesy of David Shim]

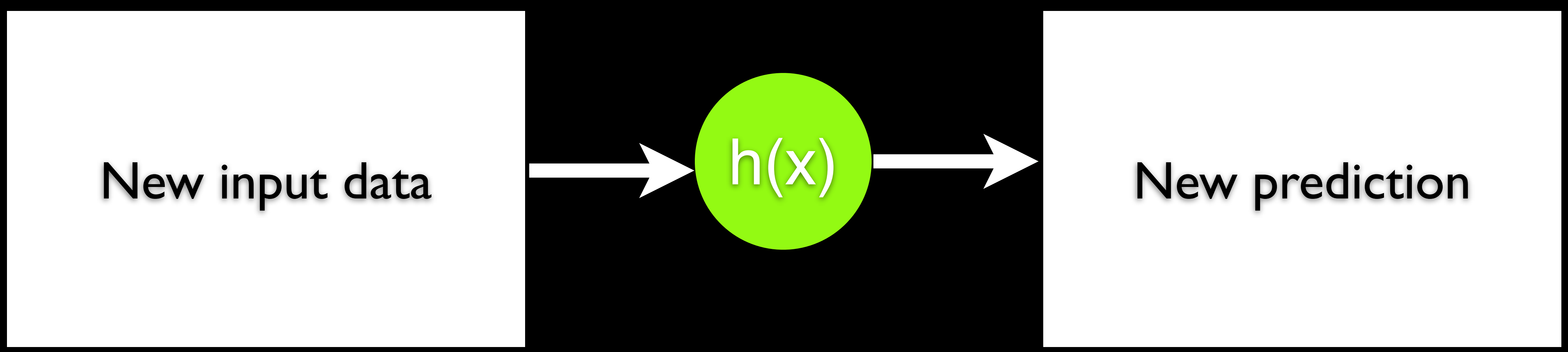
Tuesday, June 26, 12



Tuesday, June 26, 12

? is unknown function that maps our existing input to existing outputs

Supervised Machine Learning Goal

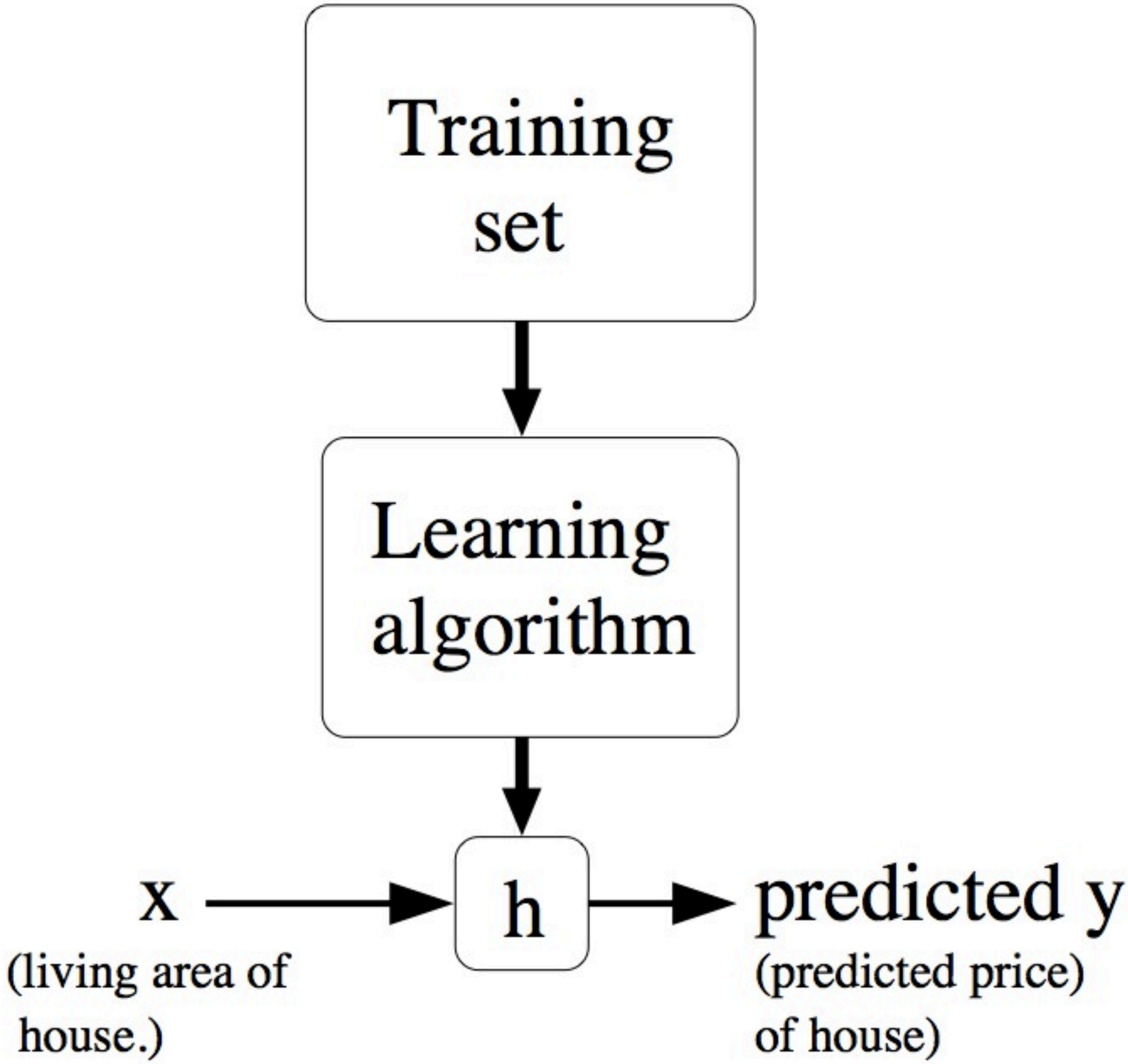


create function h(x)

Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
$\vdots$	$\vdots$

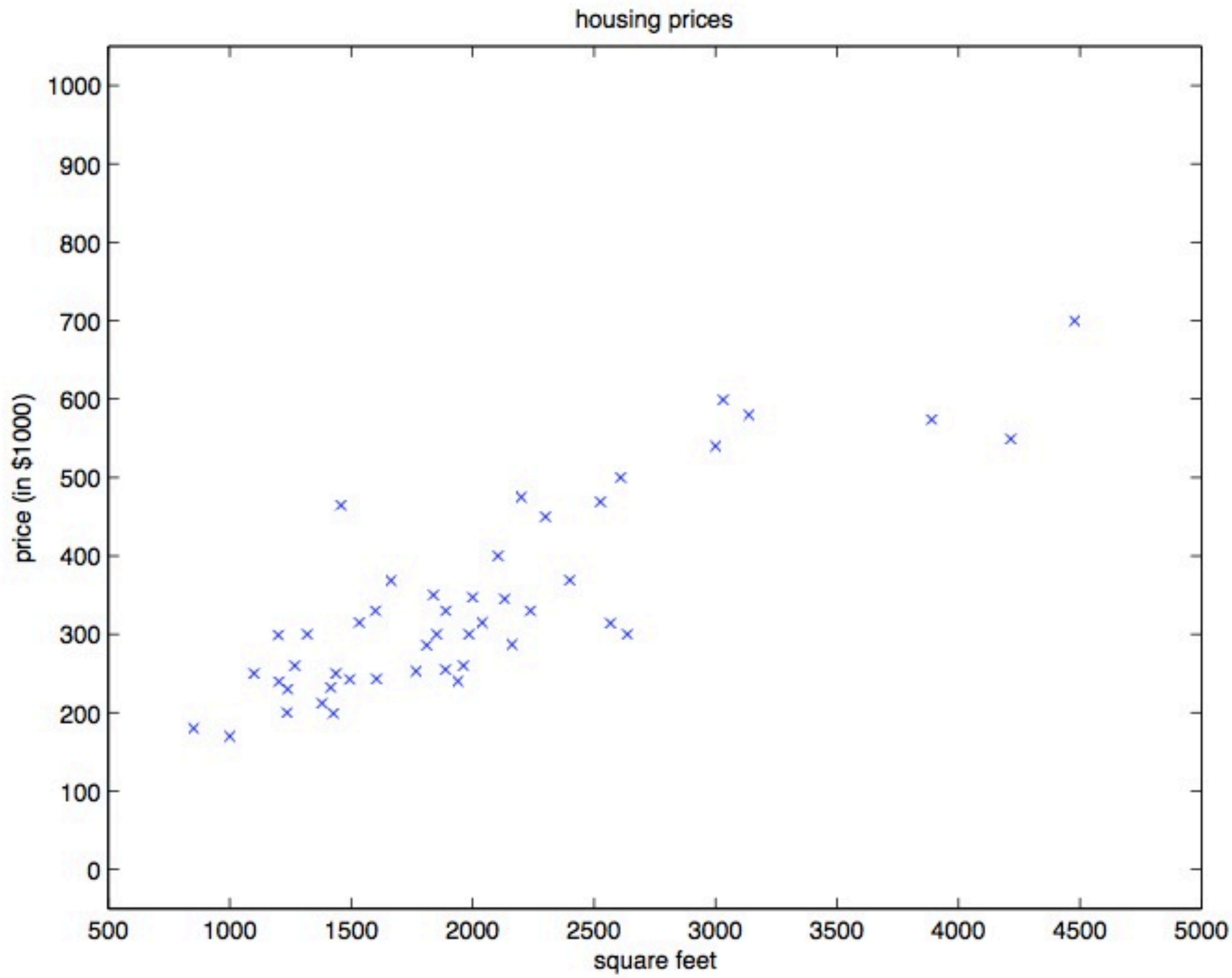
Tuesday, June 26, 12

Here is a new training set



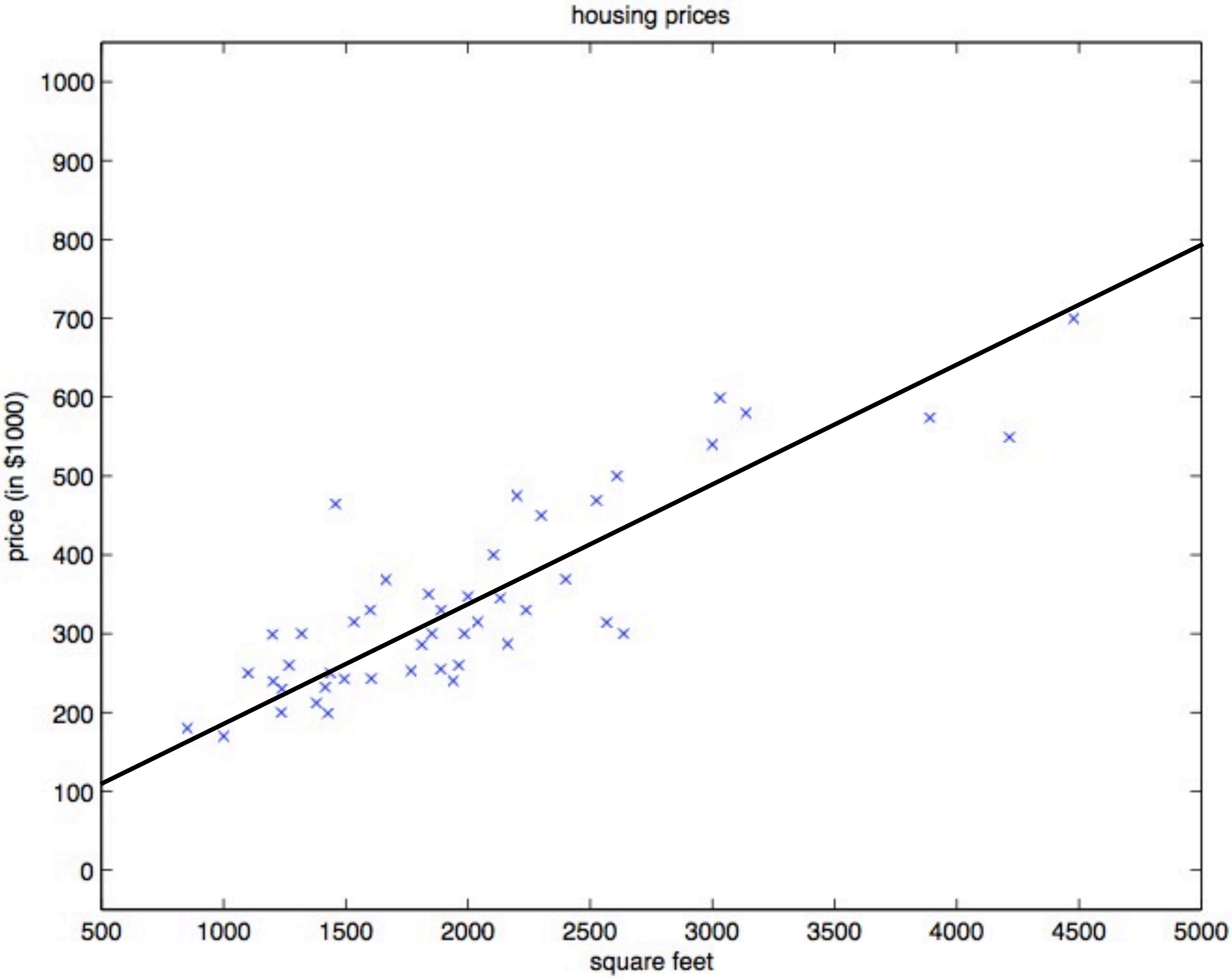
Tuesday, June 26, 12

h is our hypothesis function



Tuesday, June 26, 12

Given data like this, how can we predict new prices for square feet given that we haven't seen before?


housing prices
price (in $1000)
square feet
1000
900
800
700
600
500
400
300
200
100
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

Linear Regression

Tuesday, June 26, 12

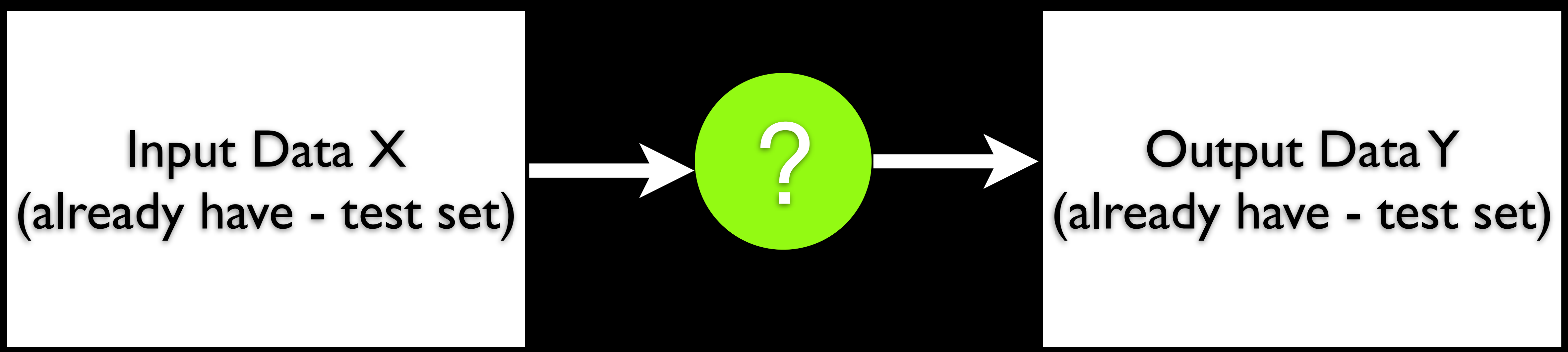
When our target variable y is continuous, like in this example, the learning problem is called a regression problem.

When y can take on only a small number of discrete values, it is called a classification problem. More on that later

show ipython linear regression demo

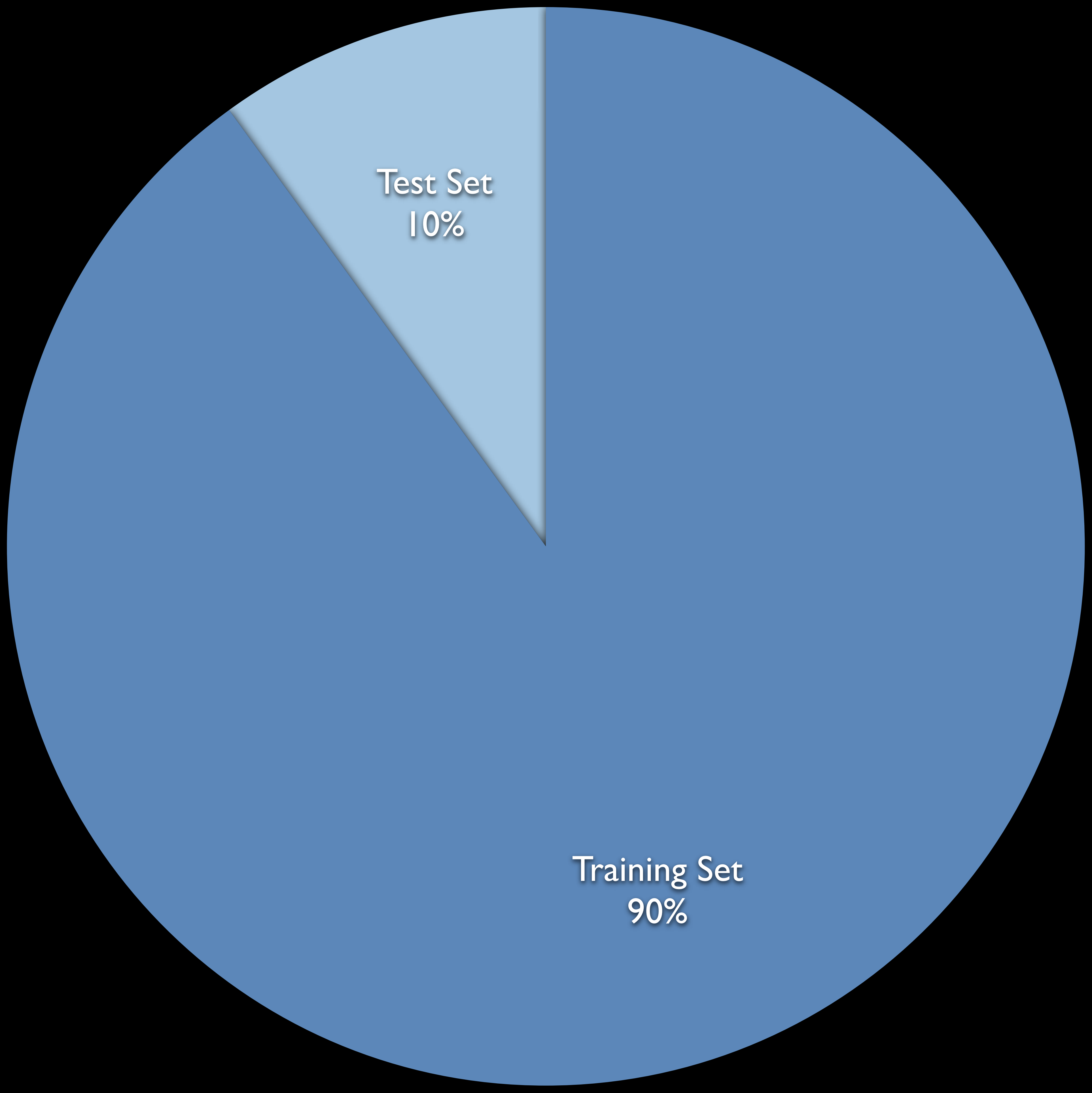
```
source virtualenvwrapper.sh && workon ml_talk && cd ~/work/ml_talk/demos && ipython notebook --pylab inline
```

How did we do?



Tuesday, June 26, 12

? is unknown function that maps our existing input to existing outputs


Test Set
10%
Training Set
90%

How can we give better predictions?

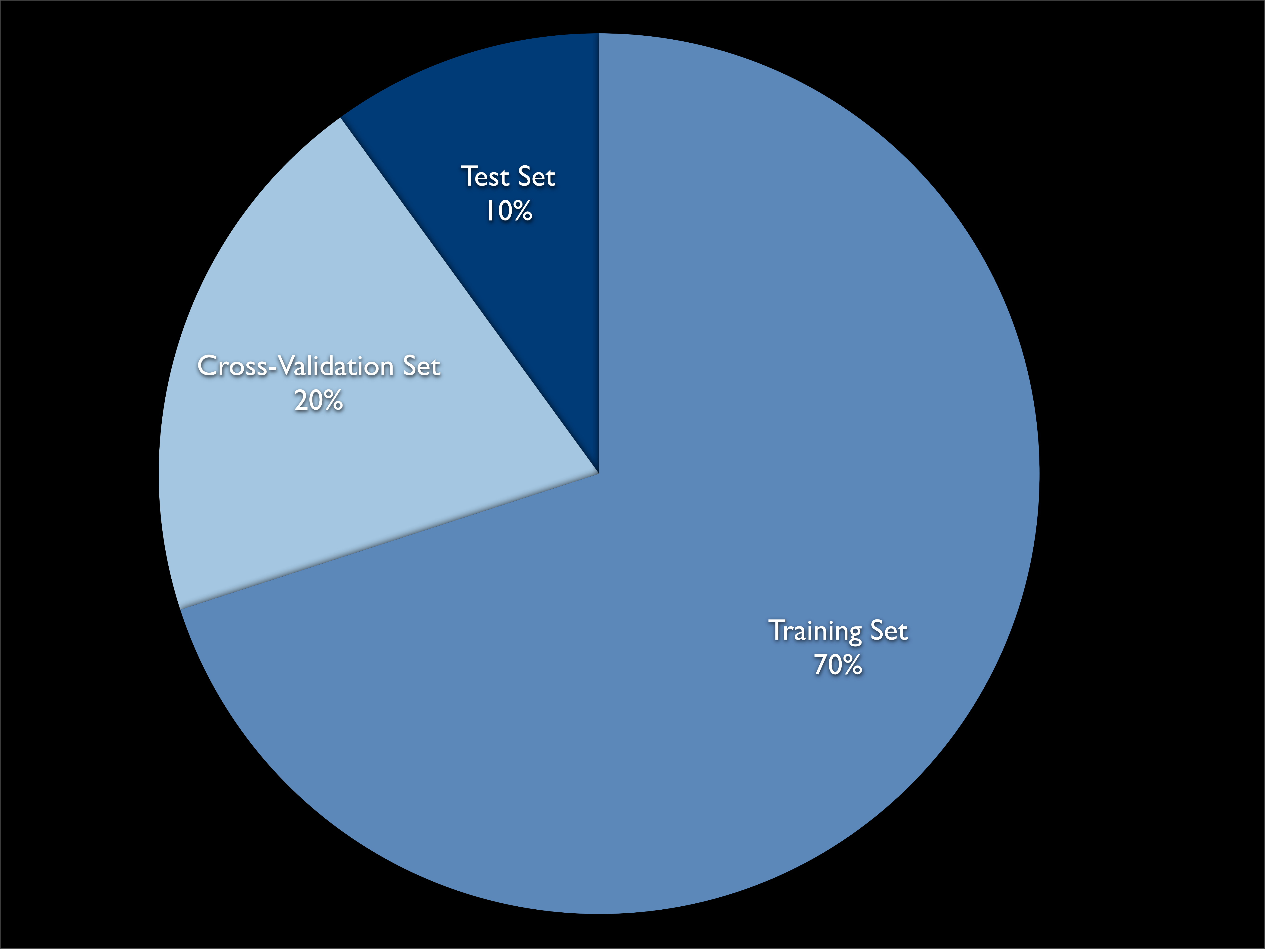
- Add More Data
- Tweak parameters
- Try a different algorithm

Tuesday, June 26, 12

these are 3 ways but there are many more

show second demo

```
source virtualenvwrapper.sh && workon ml_talk && cd ~/work/ml_talk/demos && ipython notebook --pylab inline
```



Tuesday, June 26, 12

This means we get less data in our training set

Why do we need it?

When tweaking model parameters, if we use the test set to gauge our success, we overfit to the test set. Our algorithm looks better than it actually is since we use the same set we use to fine-tune parameters that we use to judge the effectiveness of our entire model.

`numpy.polyfit`, how does it work?

Tuesday, June 26, 12

Let’s lift the covers on this magic function.

Tuesday, June 26, 12

Let’s make our own fitting function.

Hypothesis Function

$$h_\theta(x) = \theta_0 + \theta_1 x_1$$

h is our hypothesis

You might recognize this as Y=mx+b, the slope-intercept formula

x is our input (example 1000 square feet)

Theta is the parameter (actually a matrix of parameters) we are trying to determine. How?

Cost Function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

We are going to try to minimize this thing

Sigma just means “add this stuff up” m is the number of training examples we have

J of Theta is a function that, when given Theta, tells us how close to the training results (y) our prediction function (h) gets when given data from our training set.

The 1/2 helps us take the derivative of this, don’t worry about it.

This is known as the Least Squares cost function. Notice that the farther away our prediction gets from reality, worse the penalty grows.

Gradient Descent

Tuesday, June 26, 12

This is how we will actually go about minimizing the cost function.

Gradient descent seems funky but is simple easy to visualize

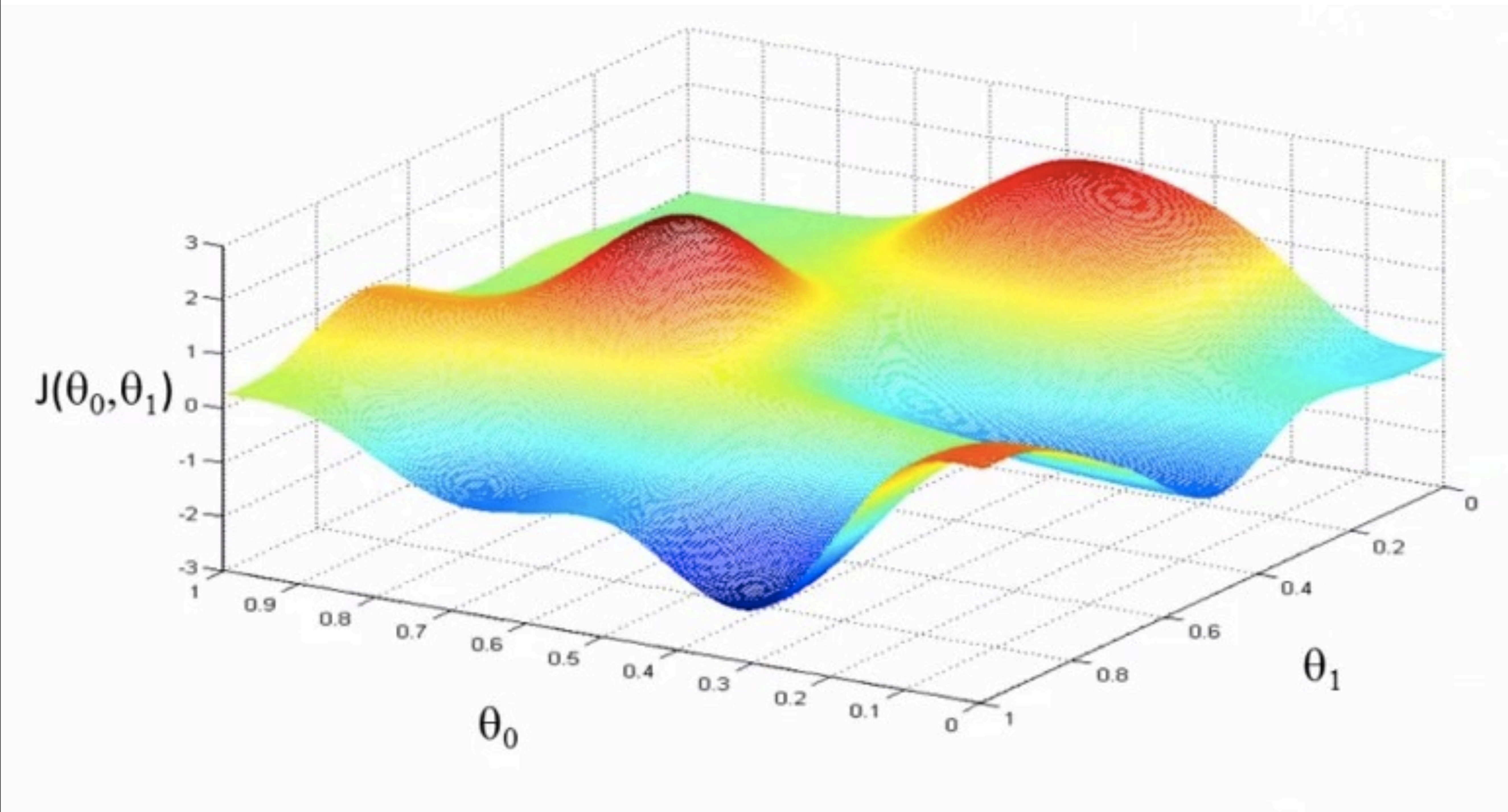
Is used quite a bit in the industry today

Have some function $J(\theta_0, \theta_1)$

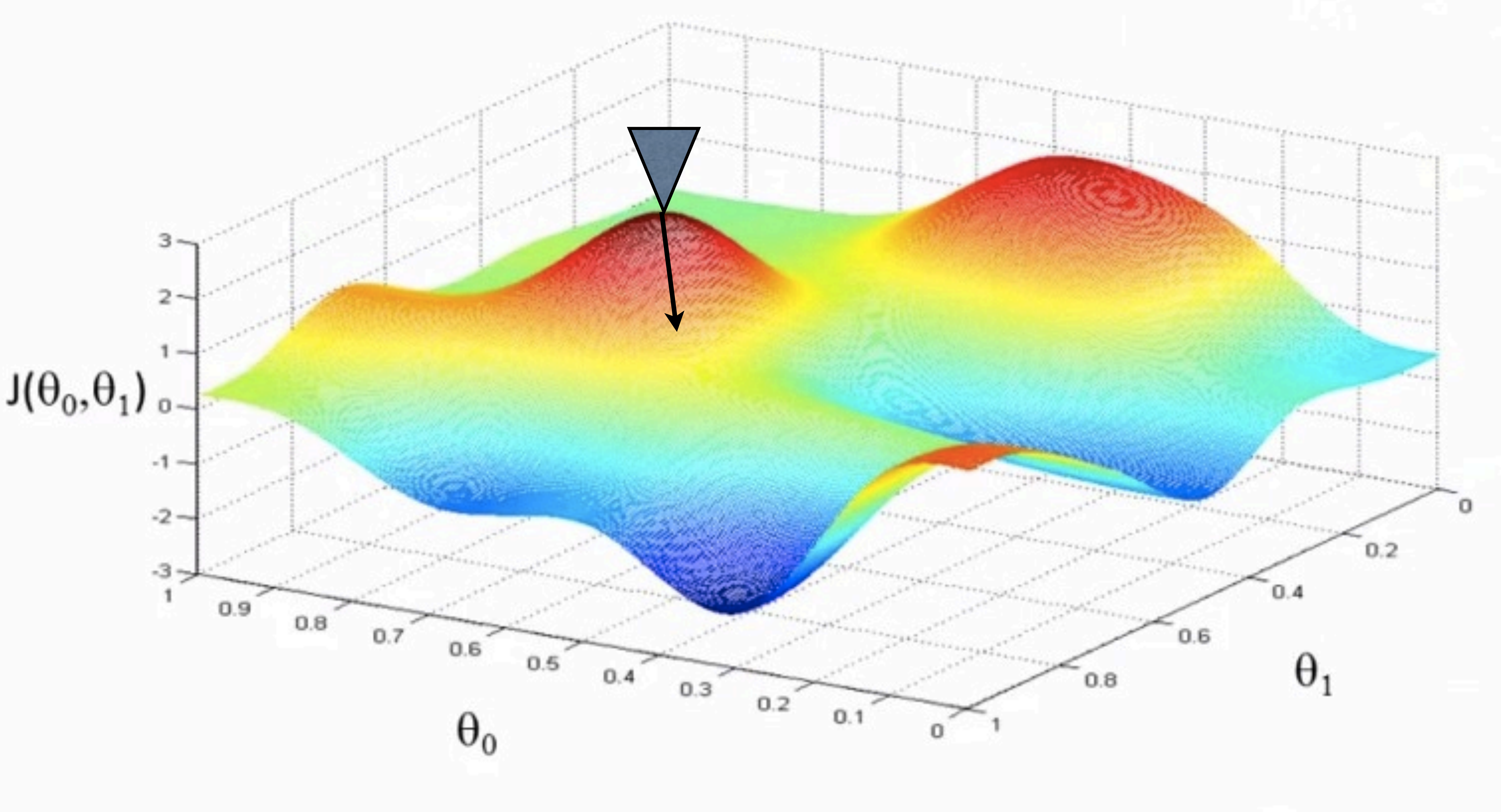
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

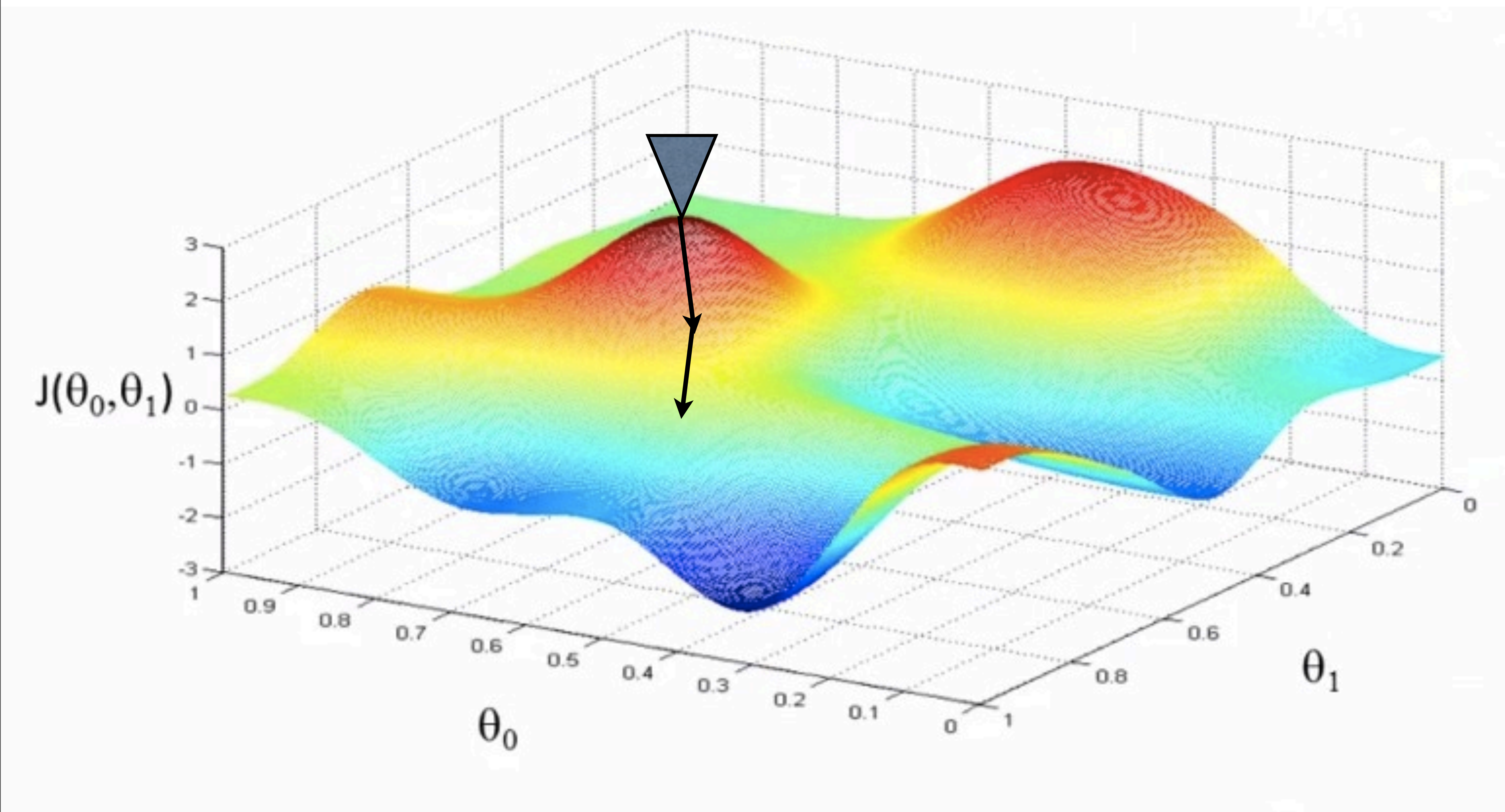
- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$
  until we hopefully end up at a minimum



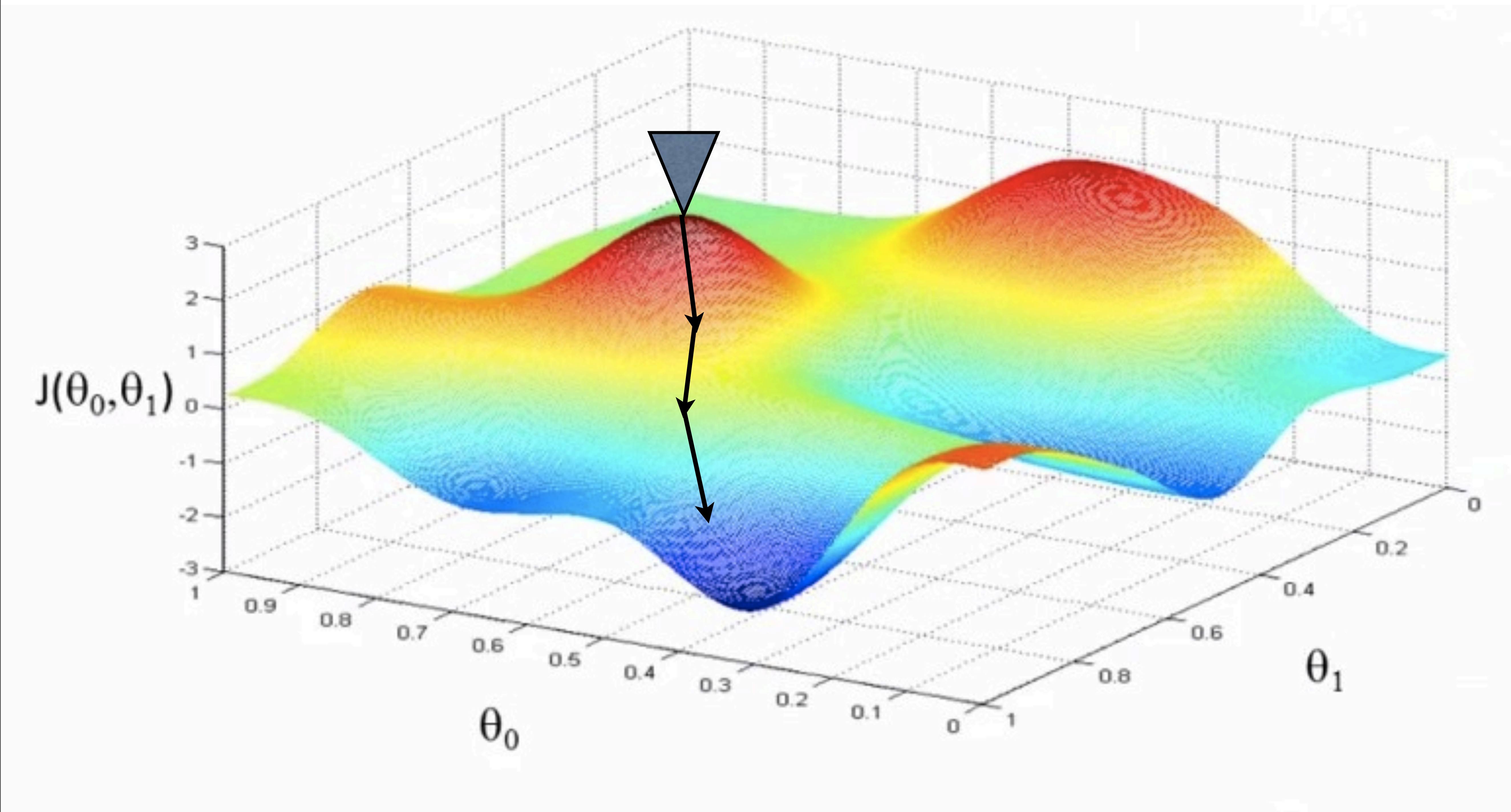
3
2
1
$J(\theta_0,\theta_1)$
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



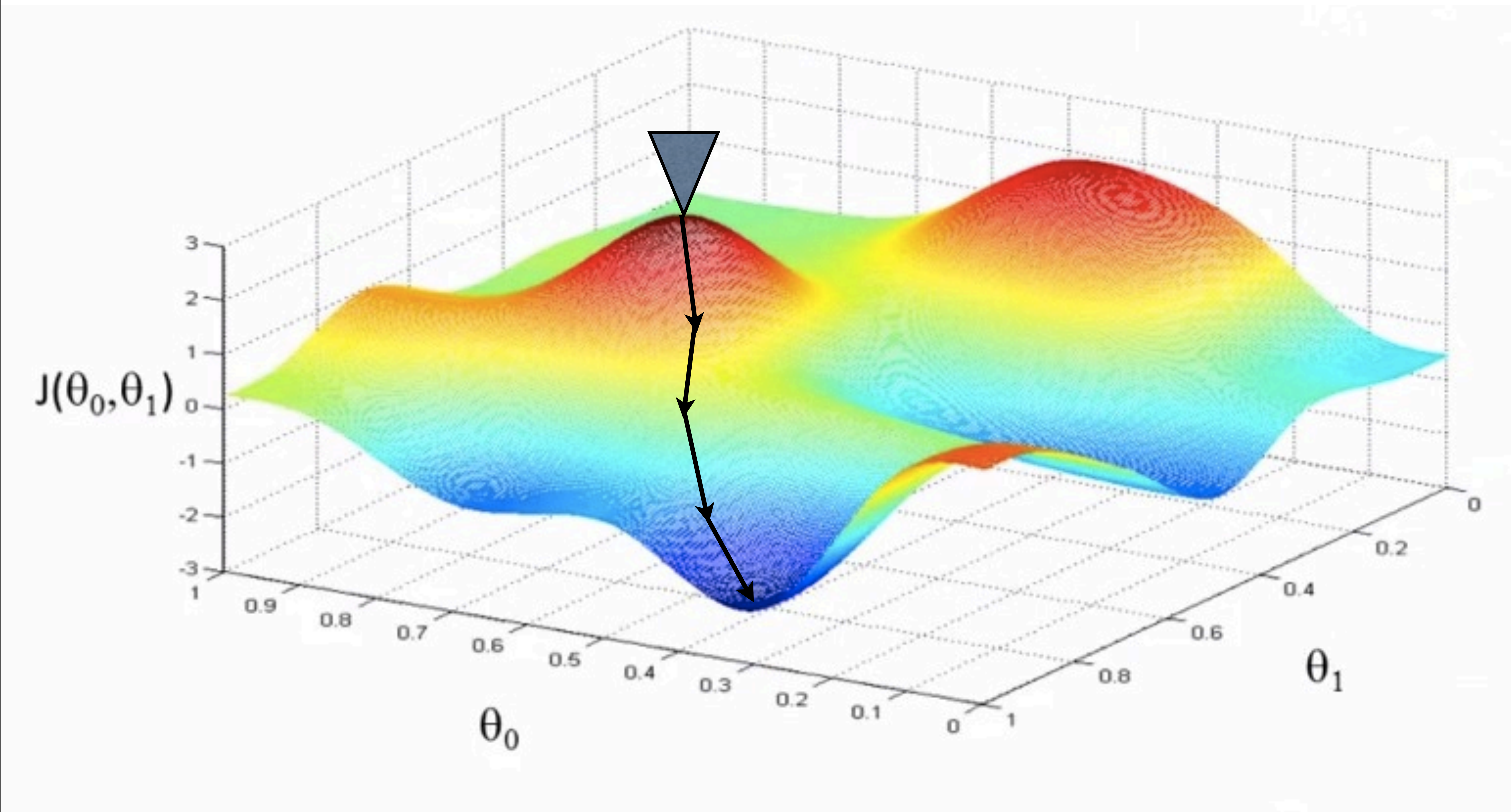
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

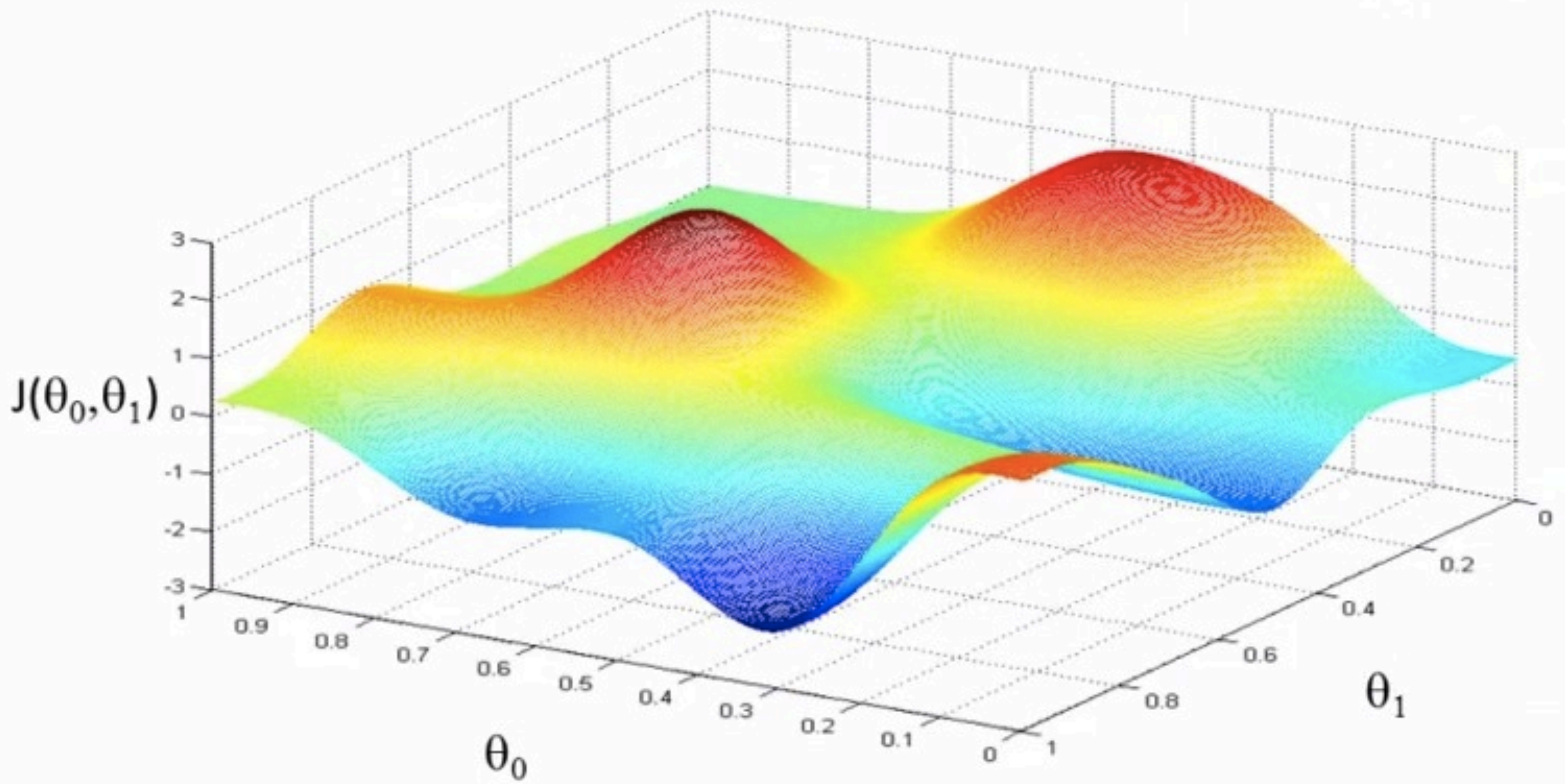


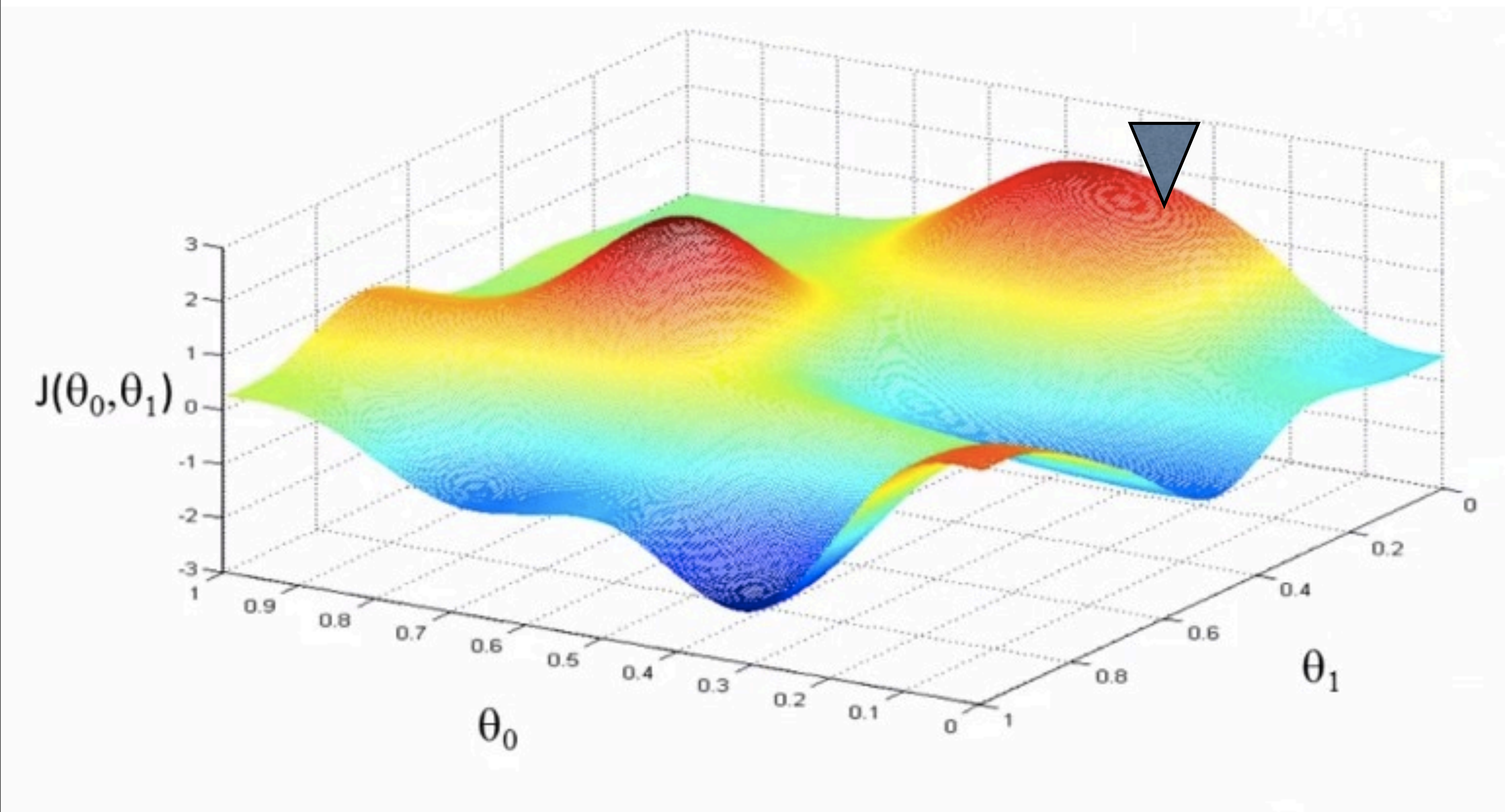
$J(\theta_0,\theta_1)$
θ_0
θ_1
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1



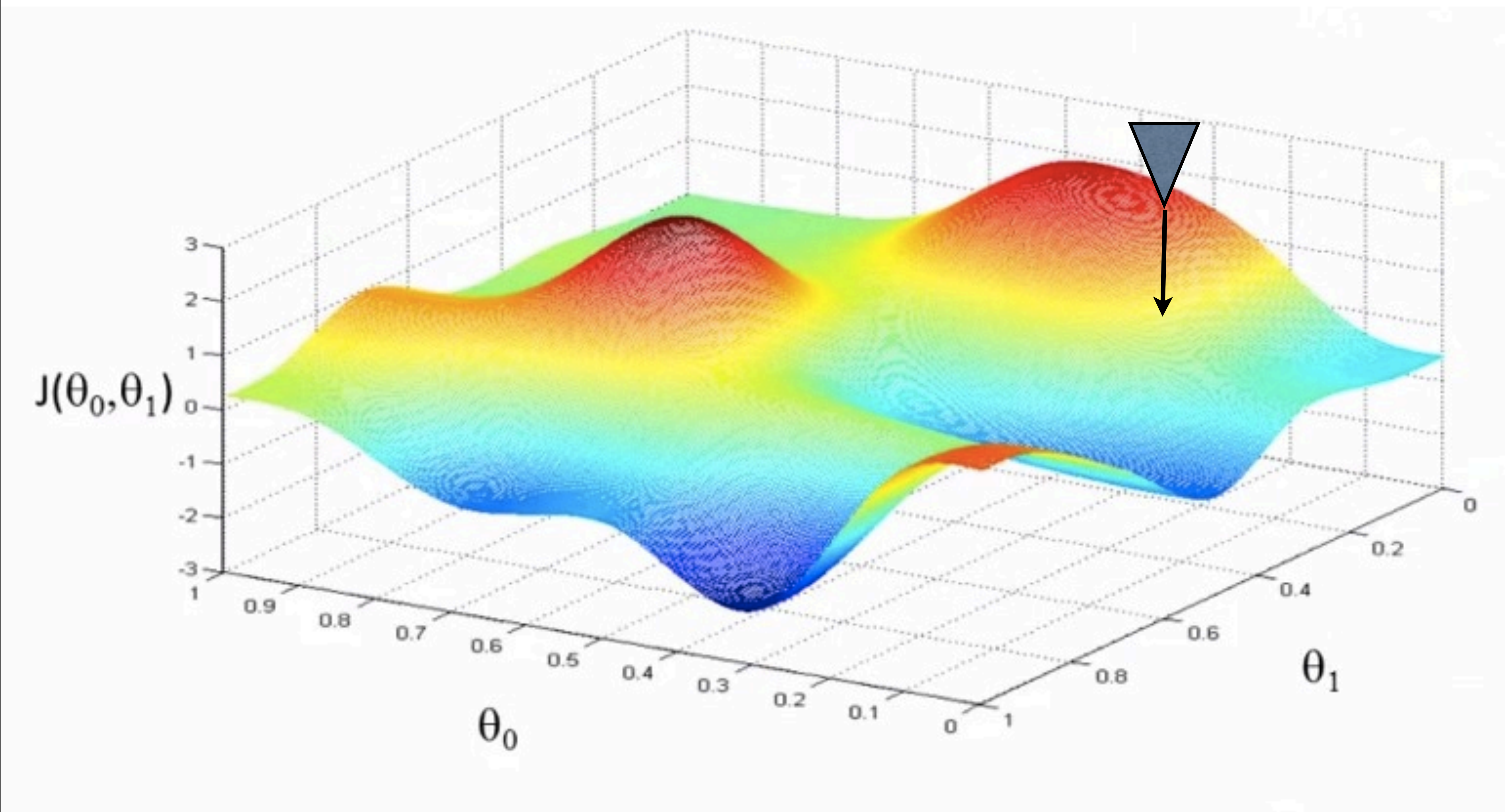
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

But... Local Optima!

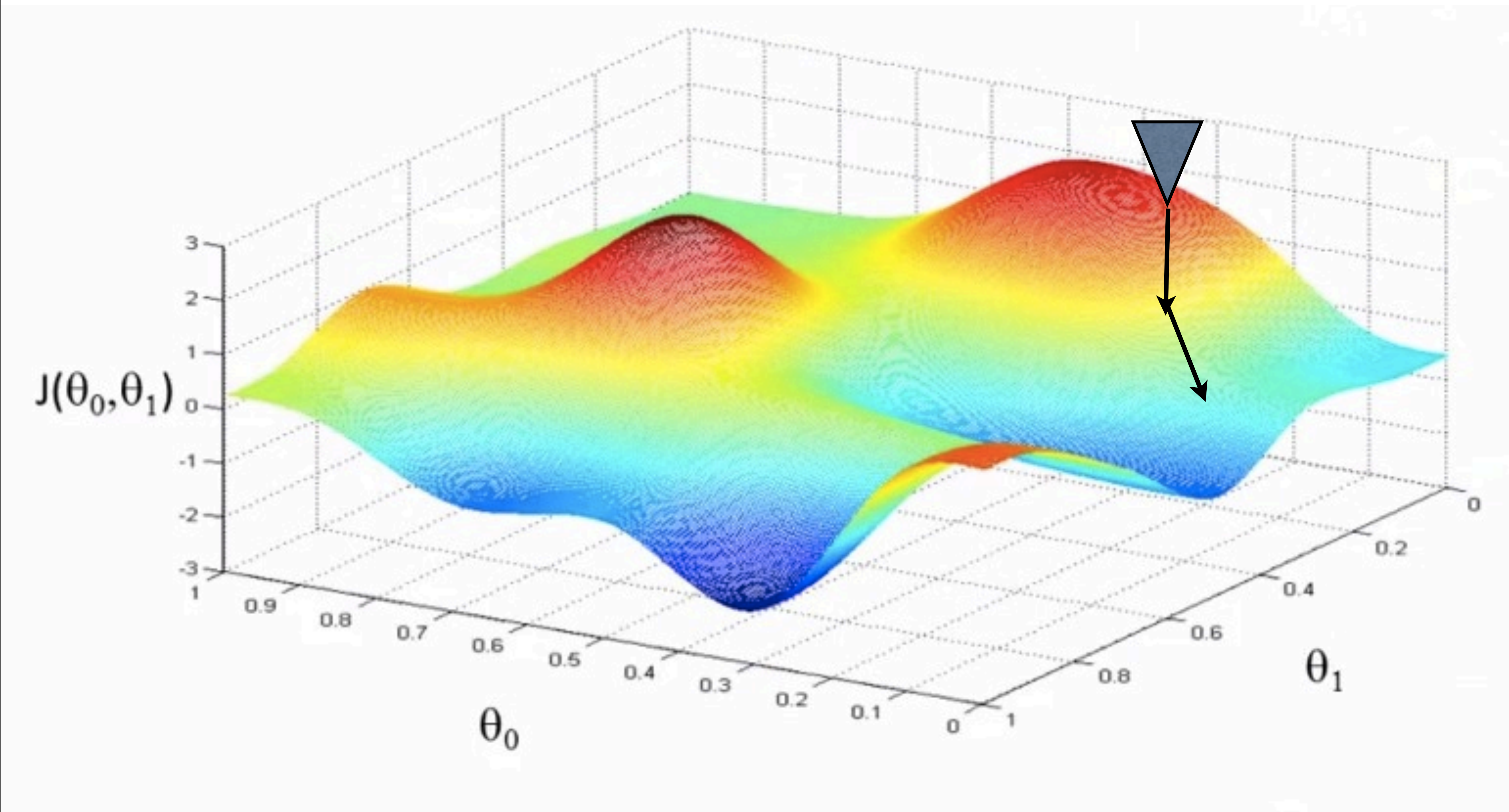

$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



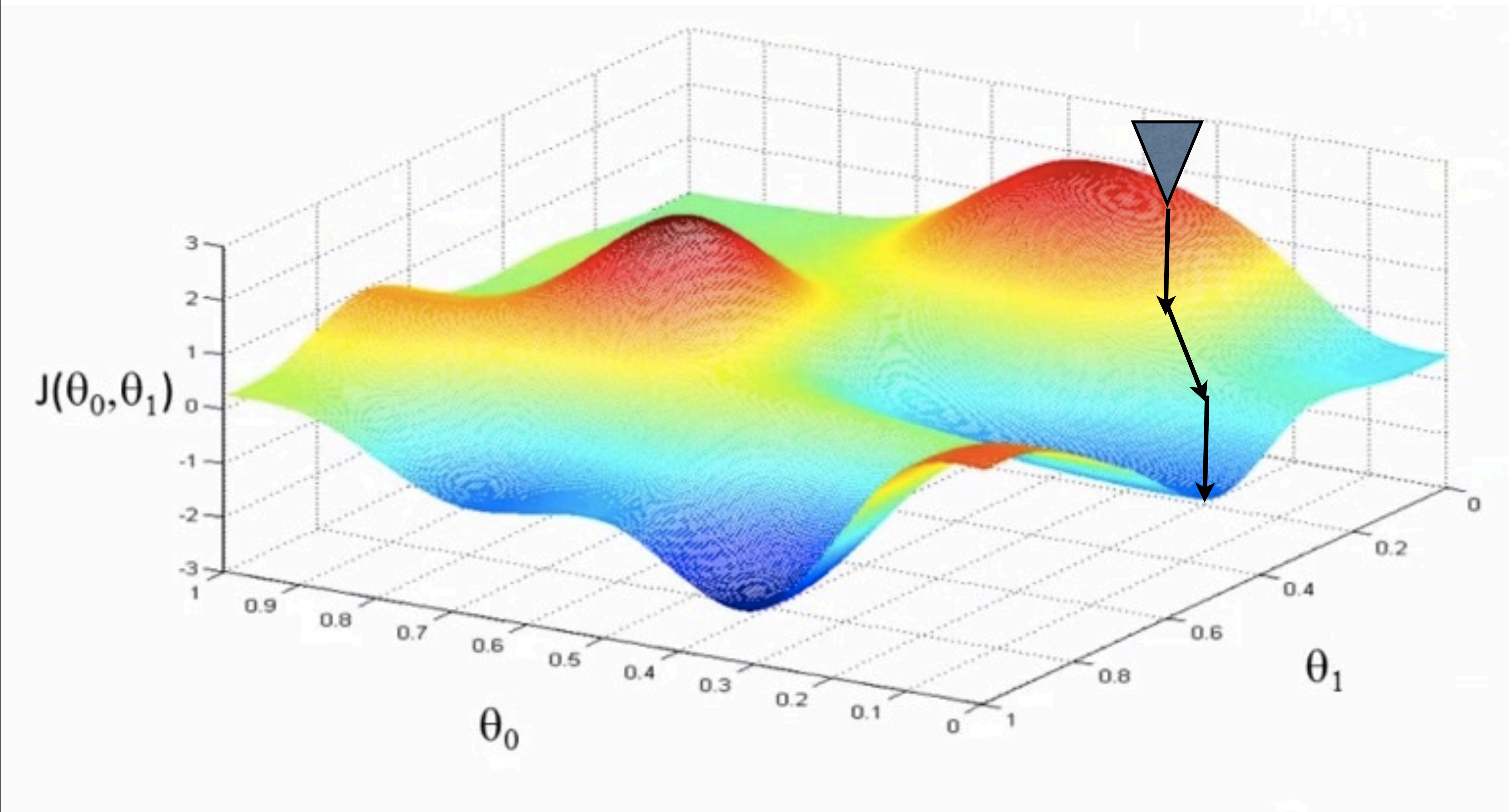
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



3
2
1
0
-1
-2
-3
$J(\theta_0,\theta_1)$
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



3
2
1
$J(\theta_0,\theta_1)$
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

Tuesday, June 26, 12

This is the update step that makes this happen

alpha is the learning rate. This tells it how big of a step to take. Has to be not too big, or it won't converge, not too small, or it will take forever

We keep updating our values for theta by taking the derivative of our cost function. This tells us which way to go to make our cost function smaller.

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y\right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}
$$

Tuesday, June 26, 12

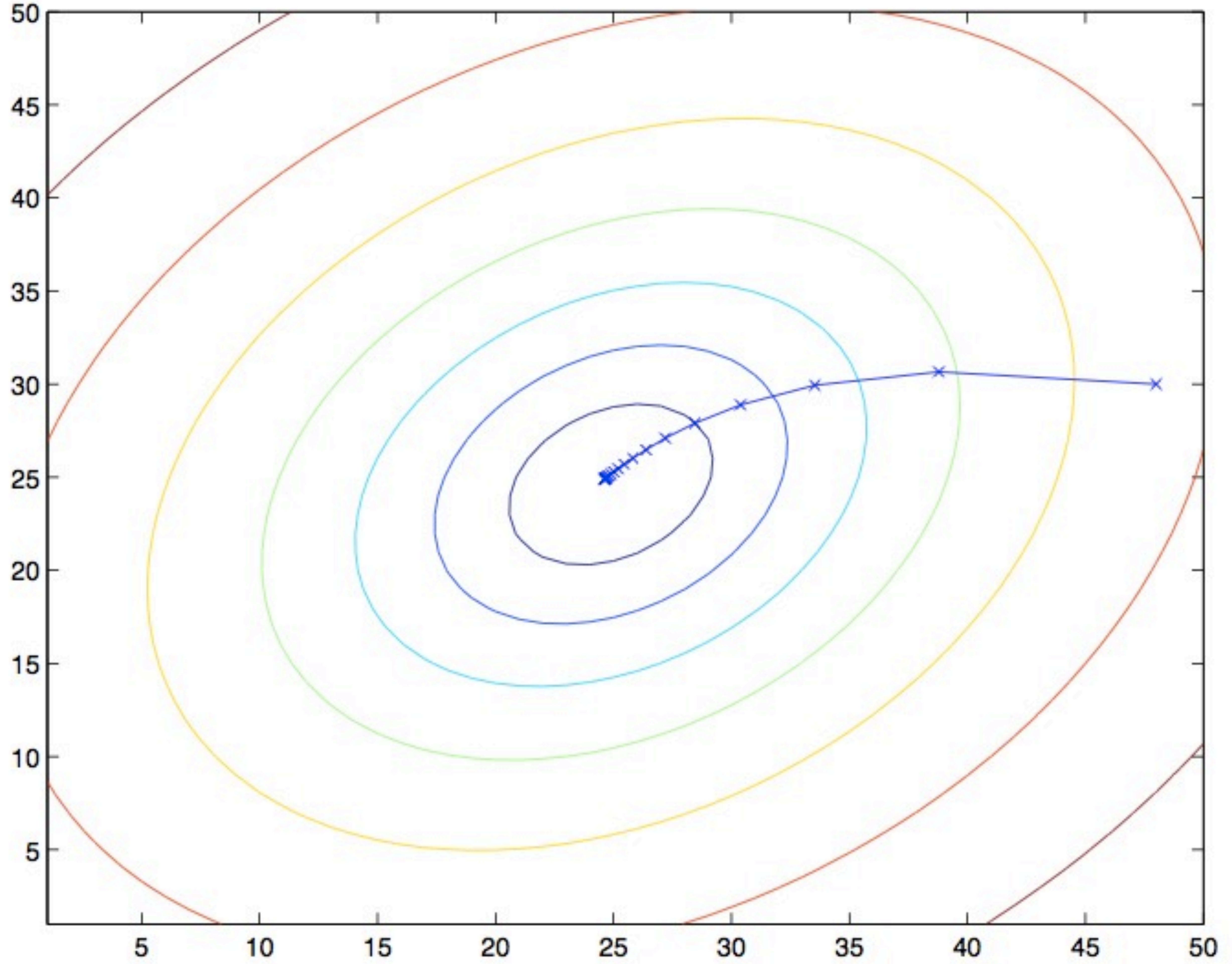
Plug in our cost function

a bunch of scary math happens (this is where the 1/2 came in handy in our cost function)

Update Step

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}.$$

repeat until convergence



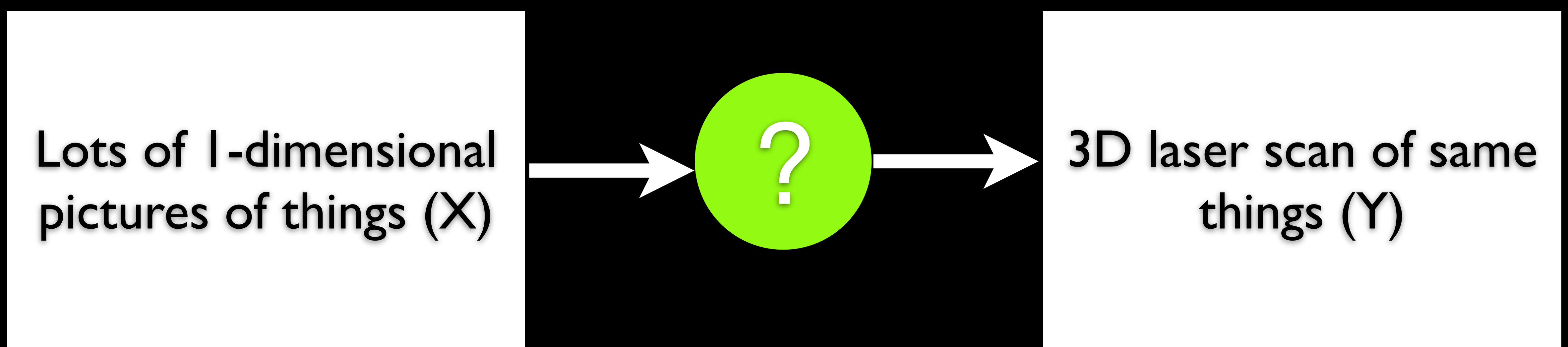
Tuesday, June 26, 12

too bad that’s not actually how numpy.polyfit works

let’s take a break


QUADS.CA

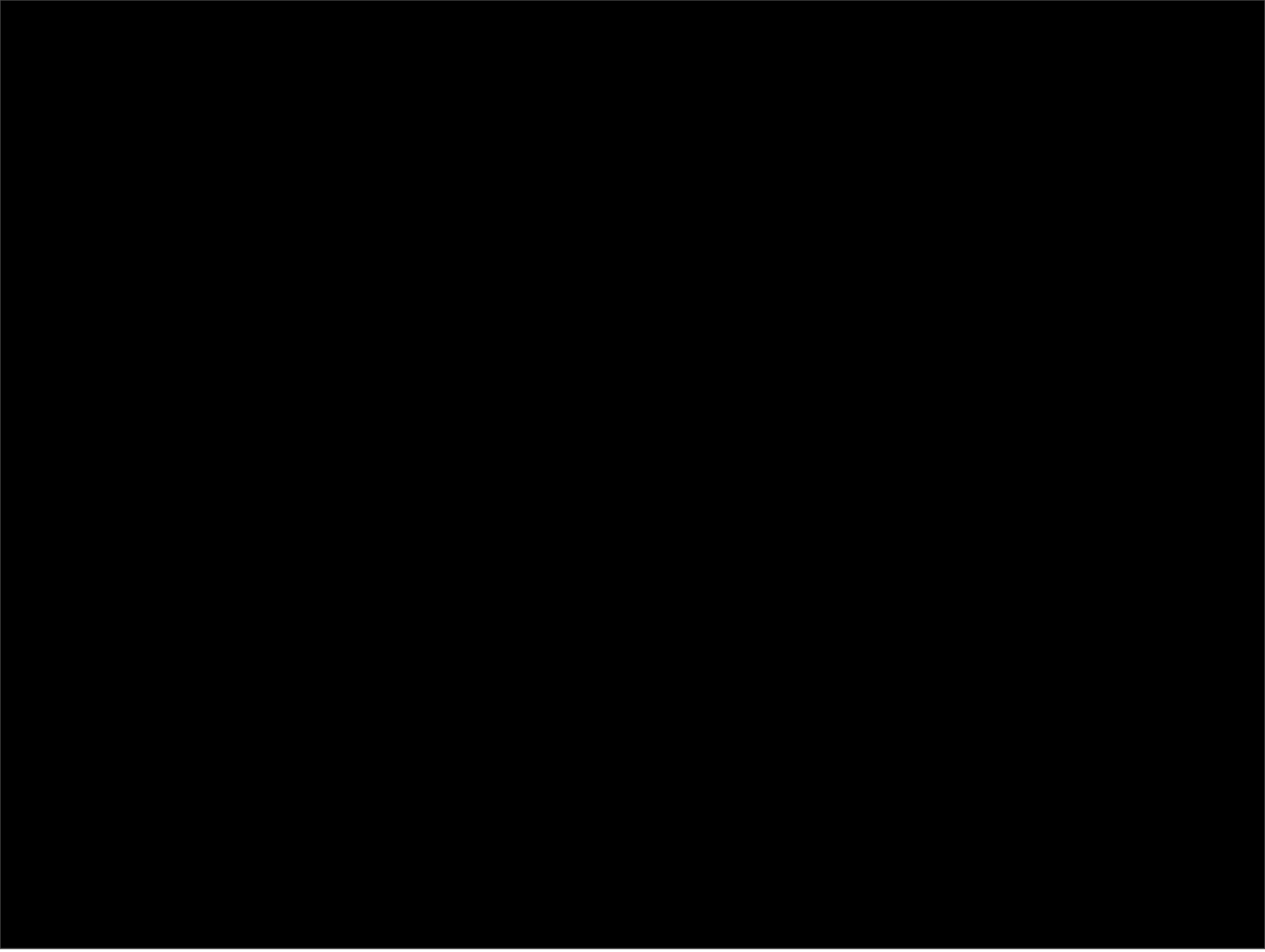
Training Set



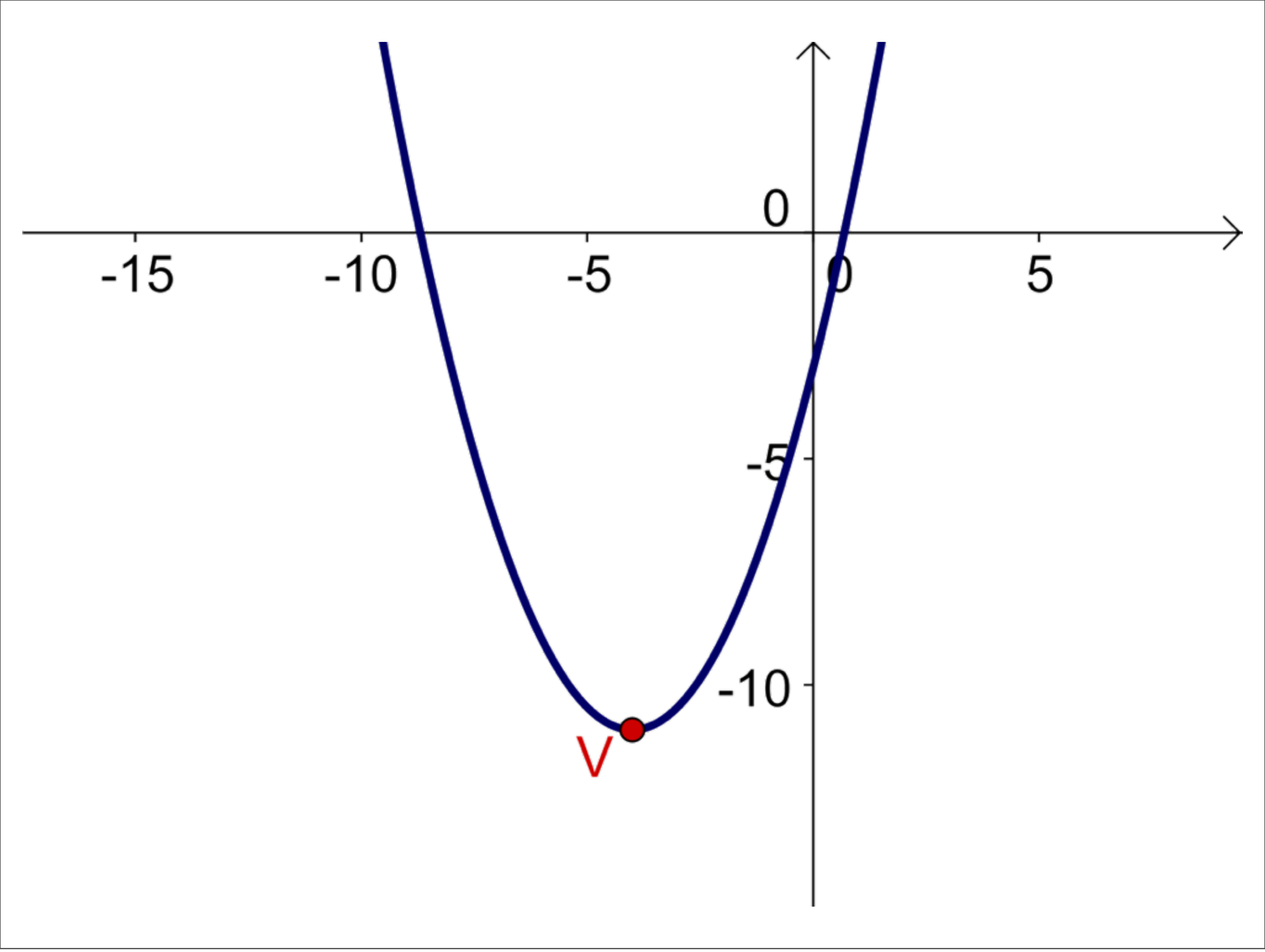

Botrics


Image: 987769.909 Laser match = 987770.996 Image no. = 5700 timeMatchingError= 0.08

Tuesday, June 26, 12



Normal Equation



high school way of minimizing a function (aka Fermat's Theorem)

all maxima and minima must exist at critical point

set derivative of function = 0

$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

Tuesday, June 26, 12

Not going to show the derivation

Closed form

m **training examples,** n **features.**

Gradient Descent

- Need to choose α.
- Needs many iterations.
- Works well even when n is large.

Normal Equation

- No need to choose α.
- Don't need to iterate.
- Need to compute $(X^TX)^{-1}$
- Slow if n is very large.

Logistic Regression

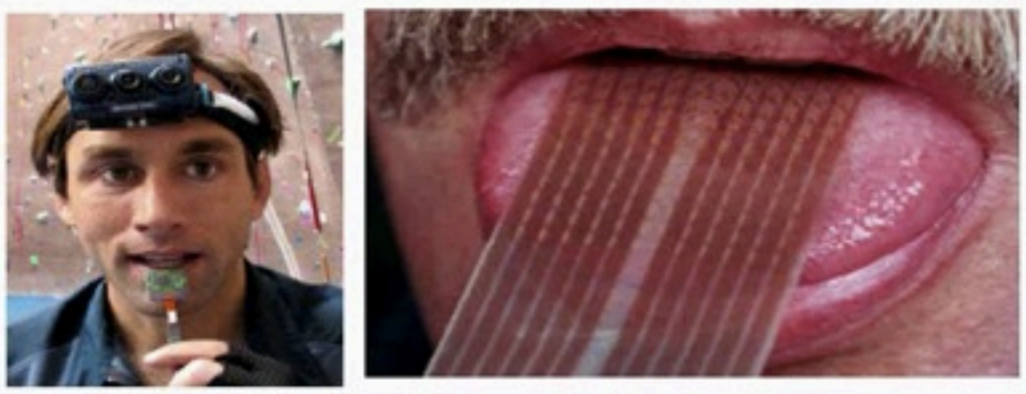
Logistic regression = discrete values

spam classifier

BORING let's do neural nets

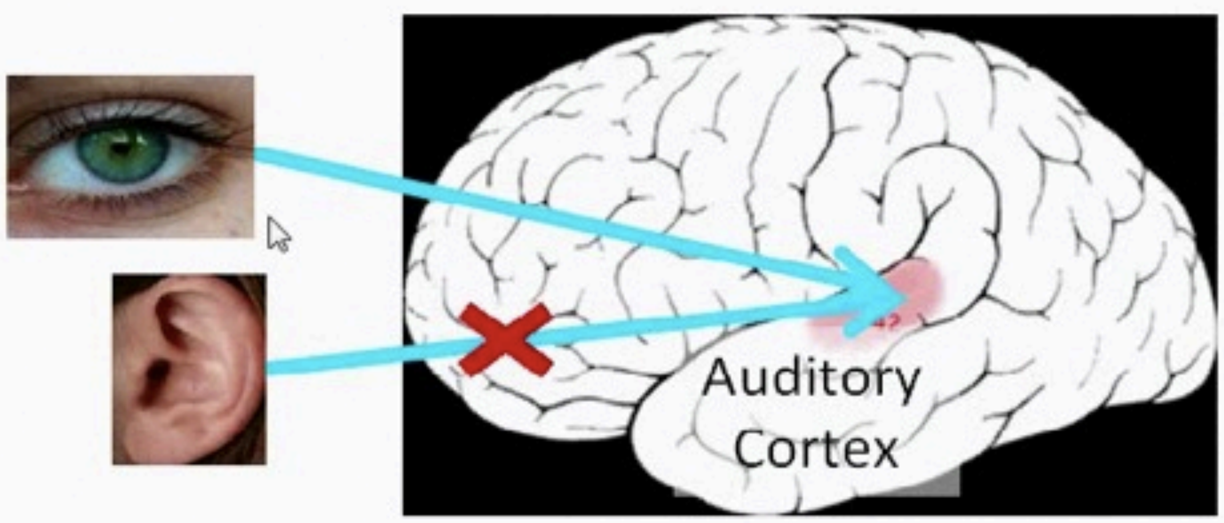
Neural Networks

Sensor representation in the brain



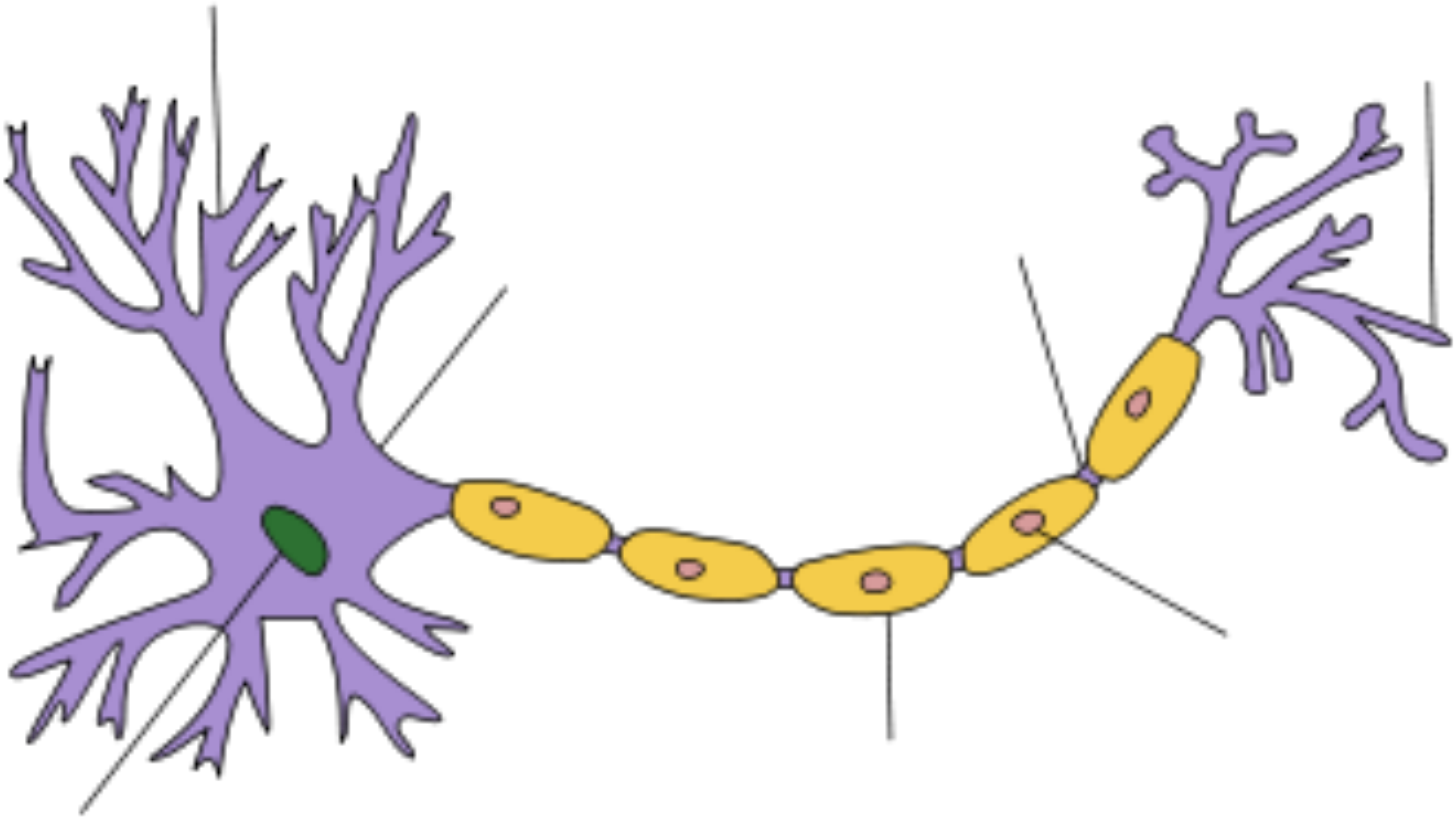
Seeing with your tongue

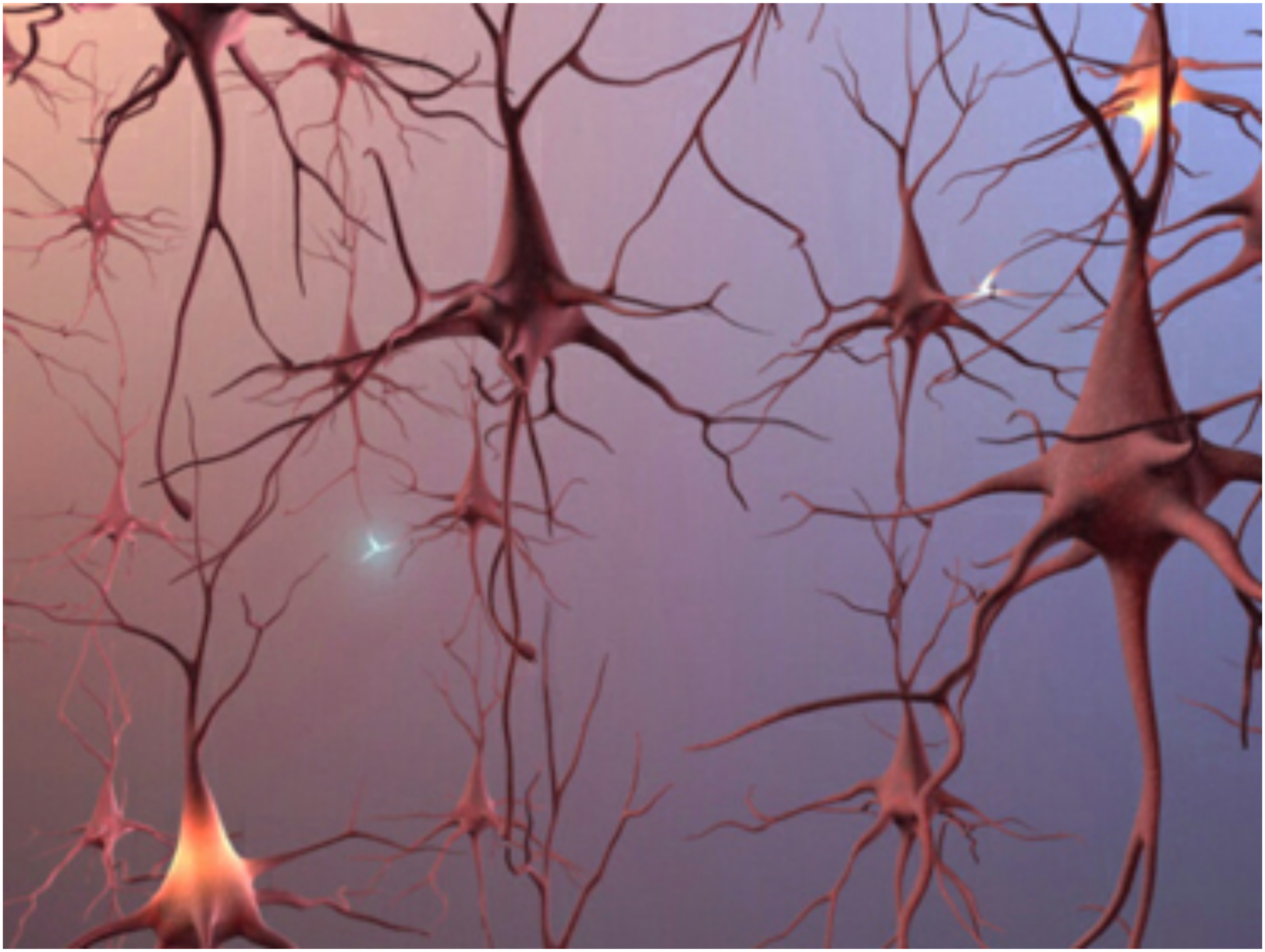
Human echolocation (sonar)

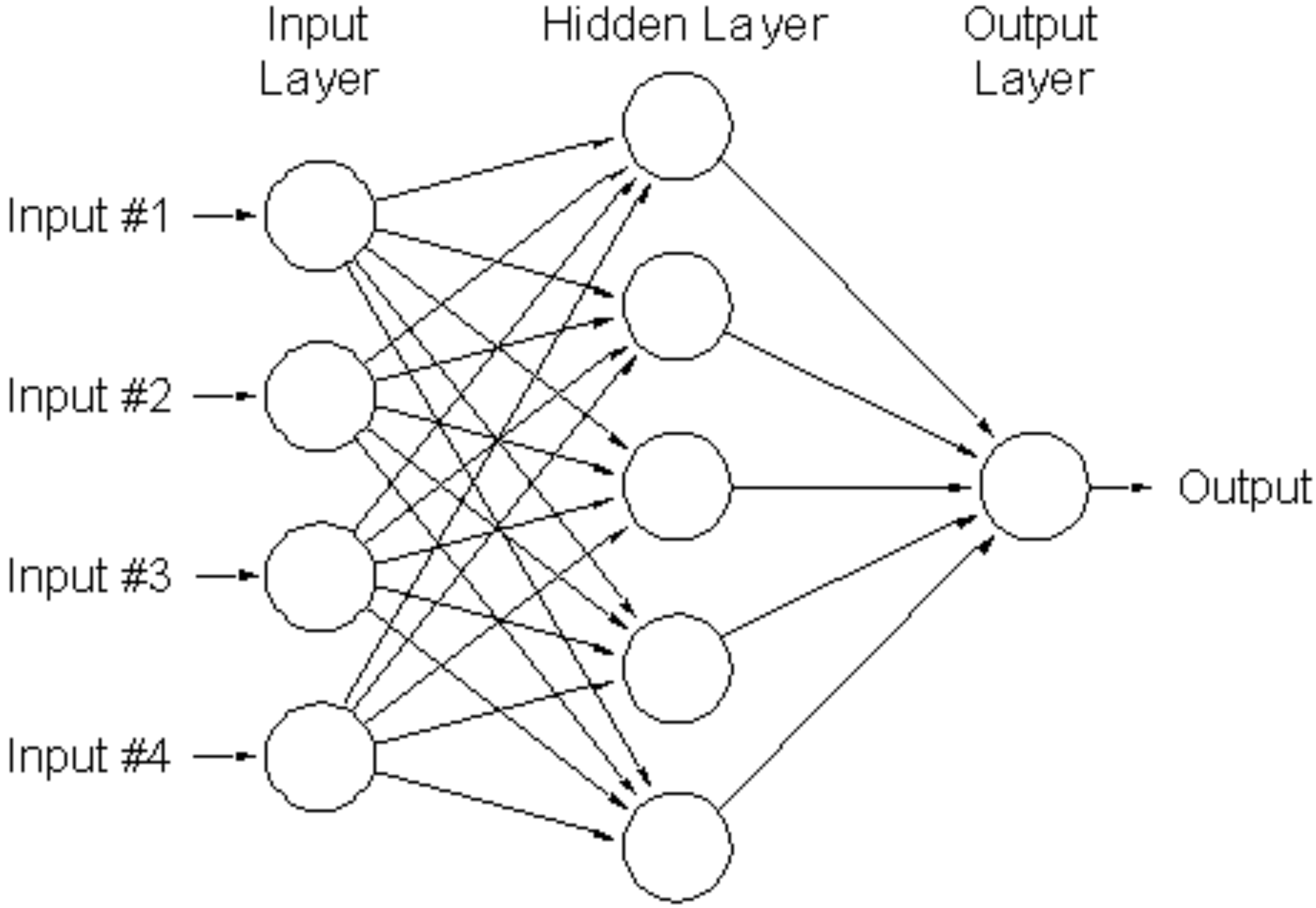


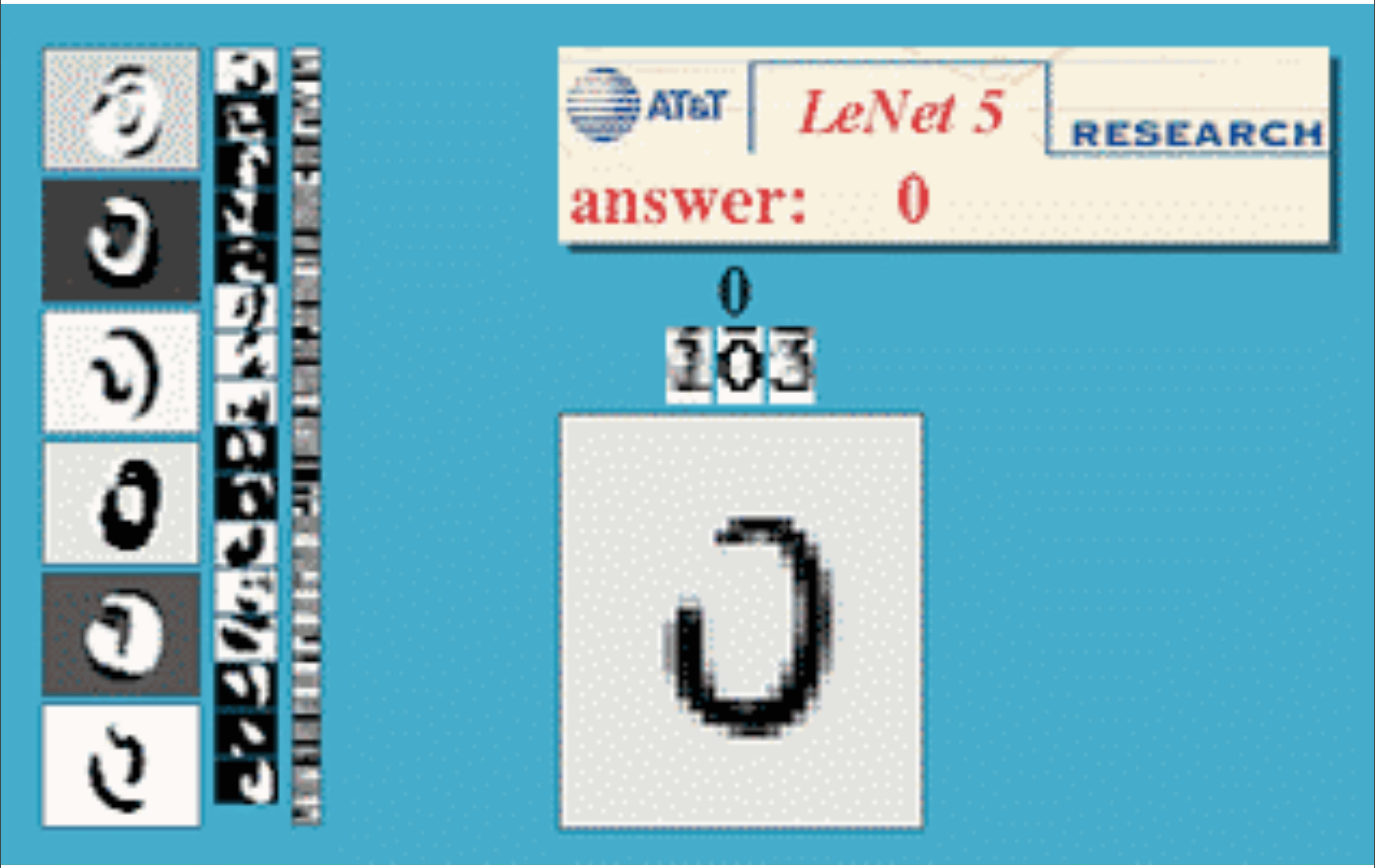
Auditory cortex learns to see.

[BrainPort; Underwood et el.; Roe et al., 1992]






Input Layer
Hidden Layer
Output Layer
Input #1
Input #2
Input #3
Input #4
Output


AT&T
LeNet 5
RESEARCH
answer: 0
0
103


Play
4.9
5.9
STANFORD
UNIVERSITY

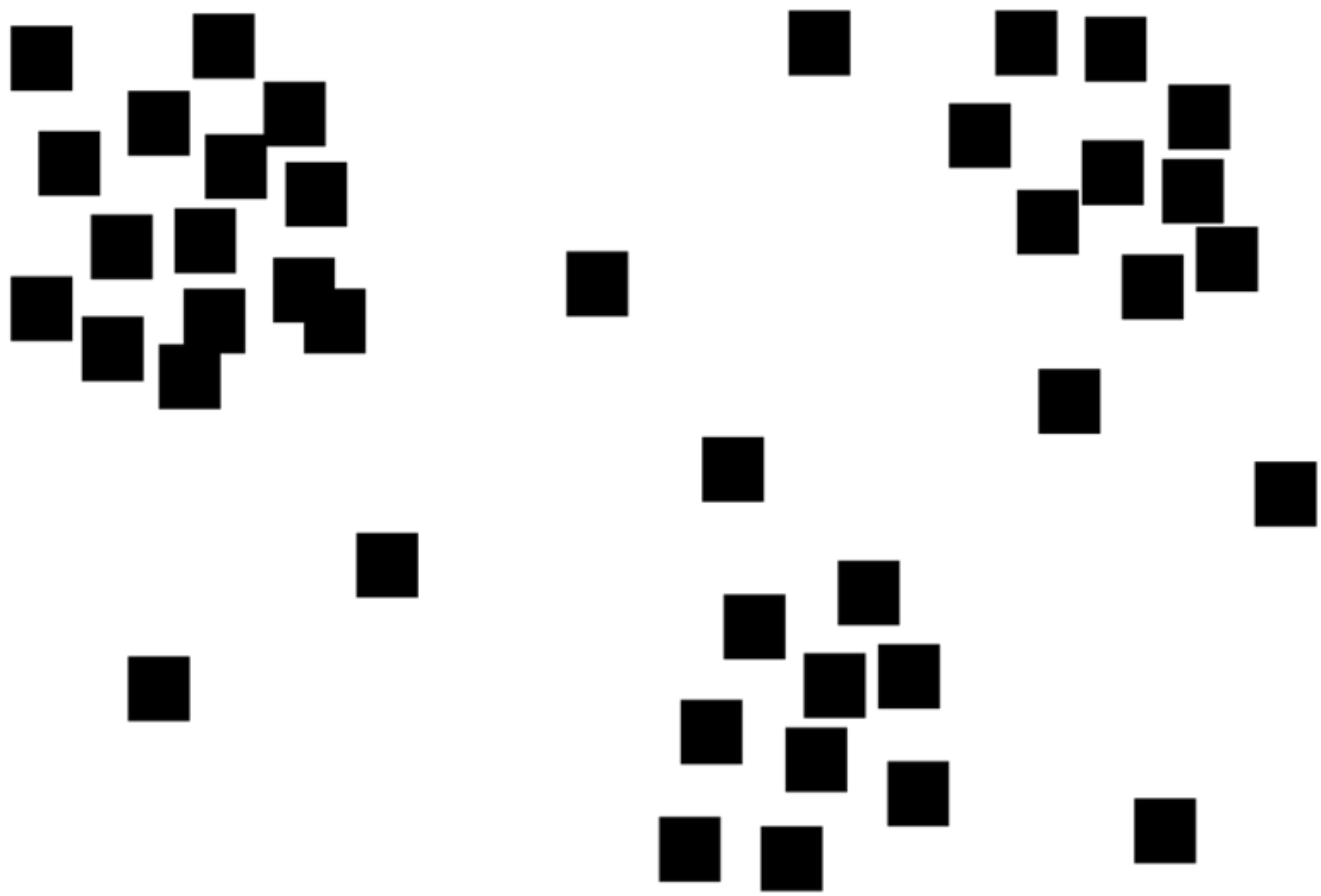
Unsupervised Machine Learning

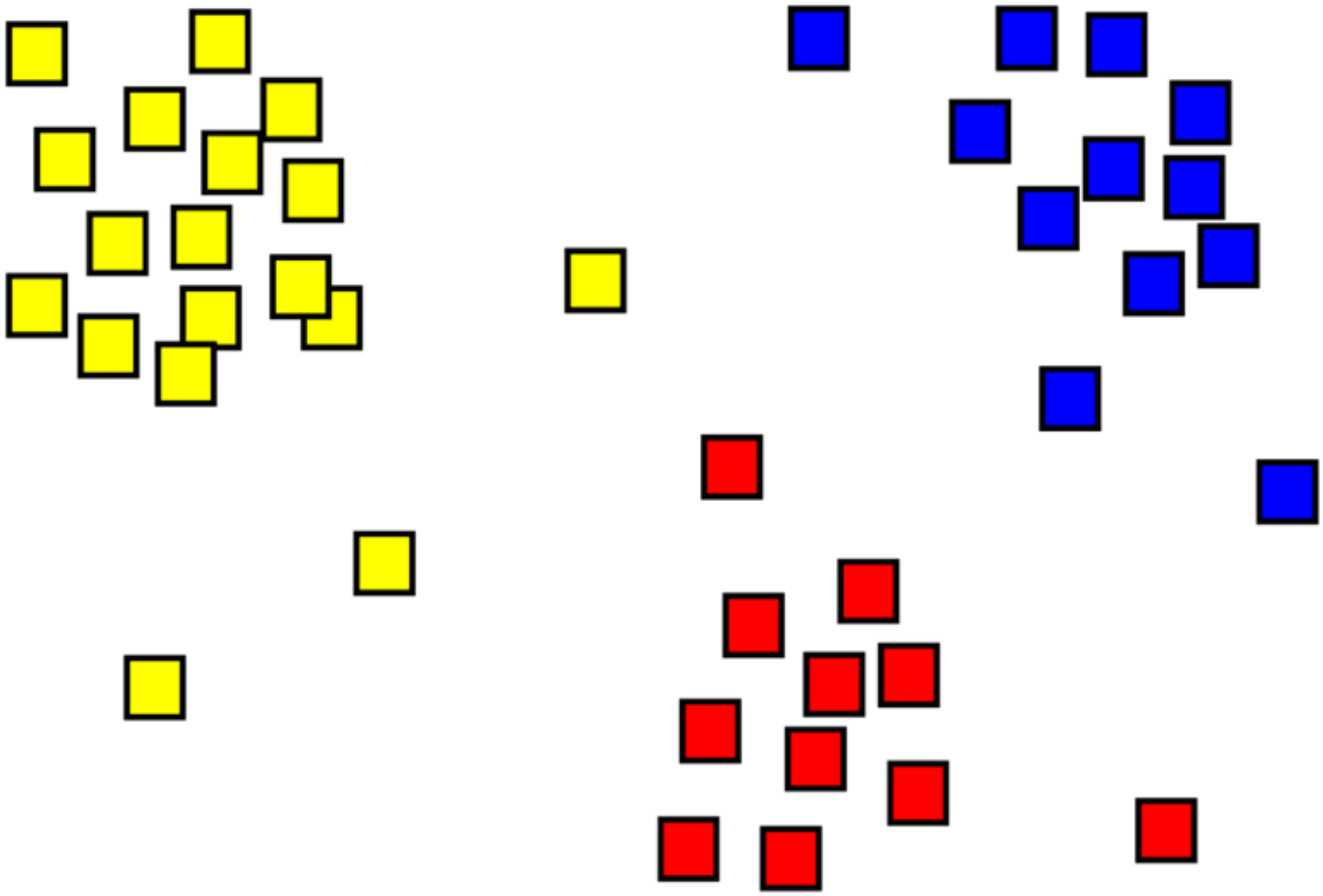
Tuesday, June 26, 12

Supervised learning - you have a training set with known inputs and known outputs

Unsupervised learning - you just have a bunch of data and want to find structure in it

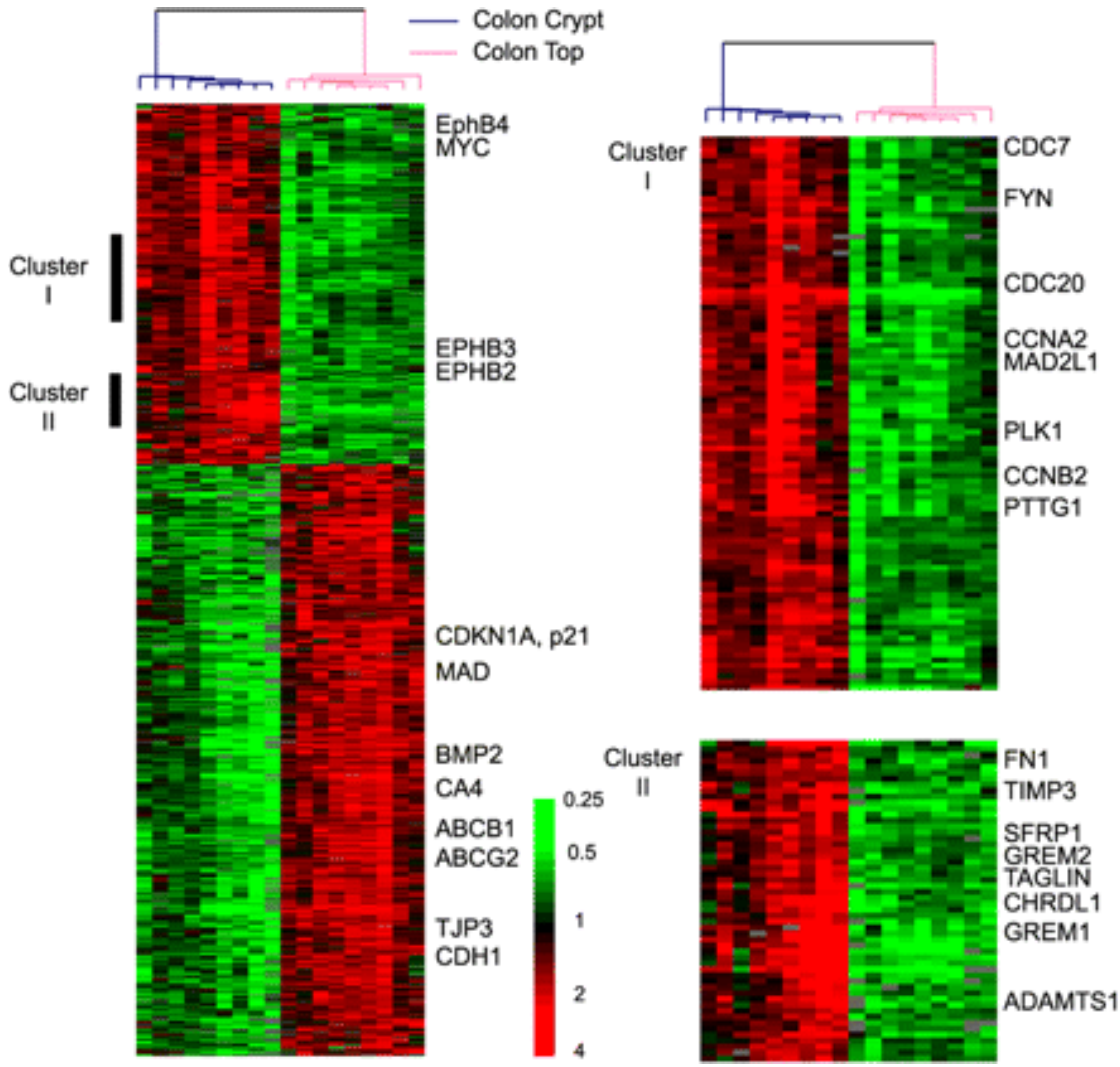
Clustering



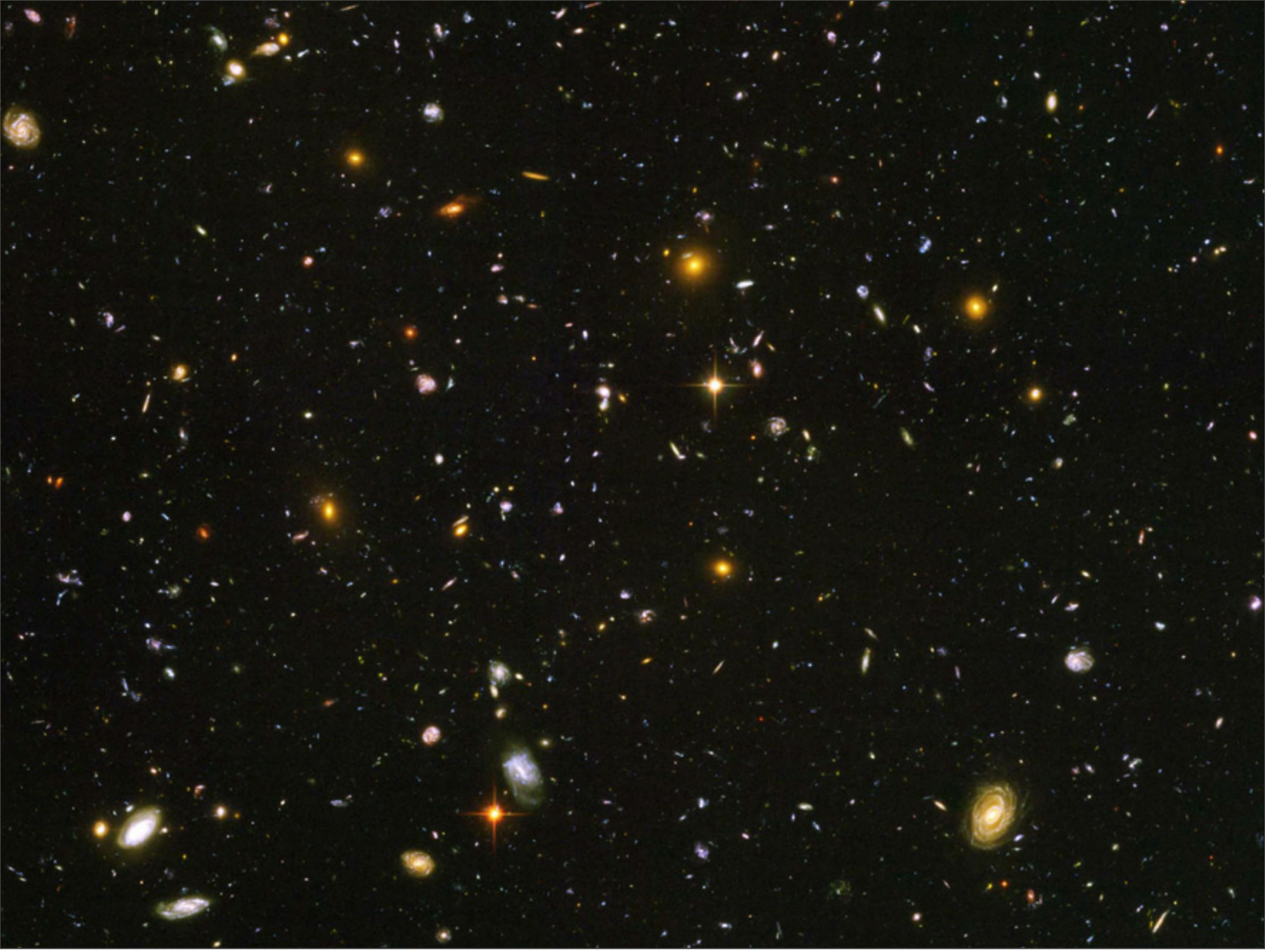


Raw image

K-means quantization

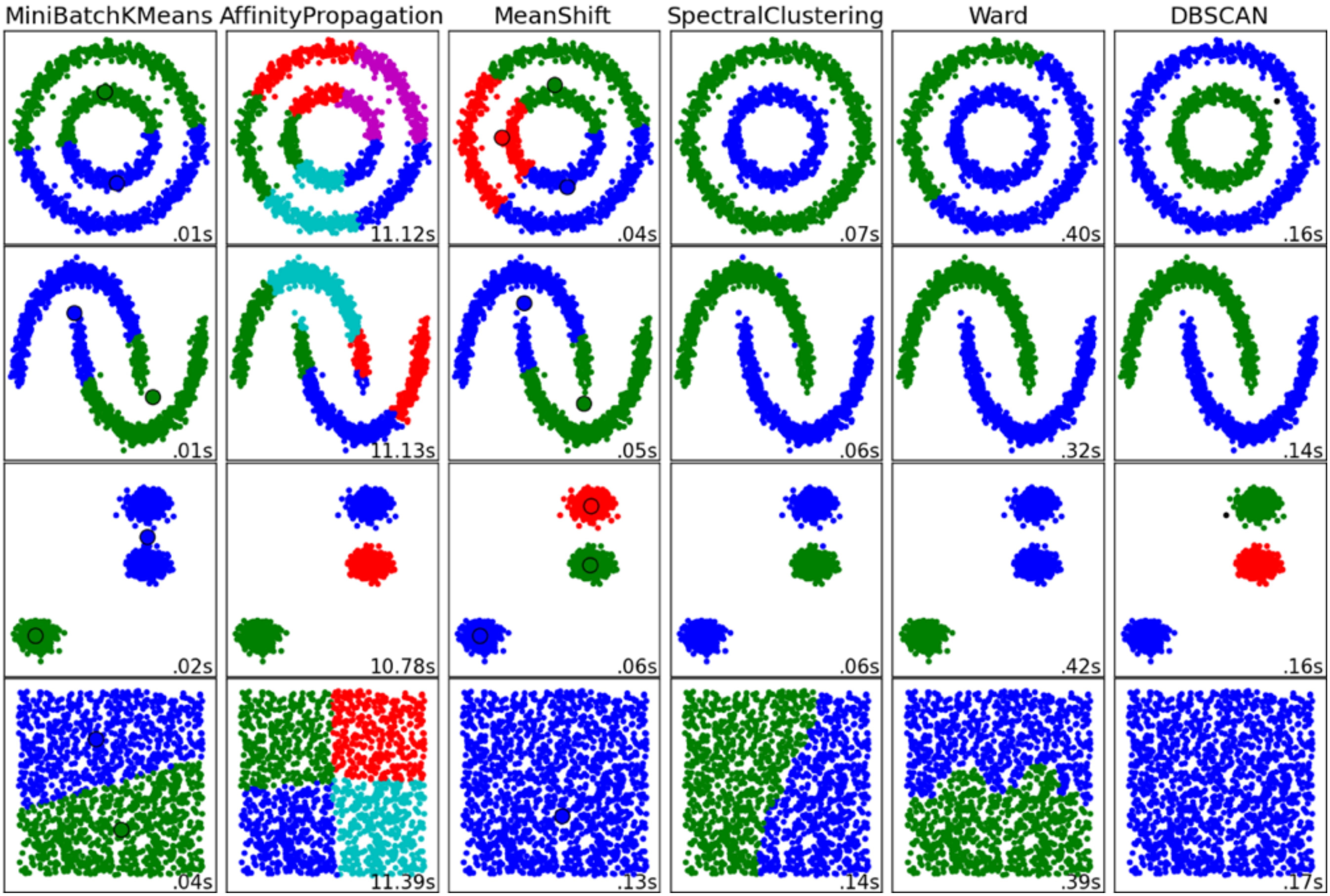


Colon Crypt
Colon Top
Cluster I
Cluster II
EphB4
MYC
EPHB3
EPHB2
CDKN1A, p21
MAD
BMP2
CA4
ABCB1
ABCG2
TJP3
CDH1
0.25
0.5
1
2
4
Cluster I
CDC7
FYN
CDC20
CCNA2
MAD2L1
PLK1
CCNB2
PTTG1
Cluster II
FN1
TIMP3
SFRP1
GREM2
TAGLIN
CHRDL1
GREM1
ADAMTS1



K-Means Clustering

K-Means Clustering


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
DBSCAN
.01s
11.12s
.04s
.07s
.40s
.16s
.01s
11.13s
.05s
.06s
.32s
.14s
.02s
10.78s
.06s
.06s
.42s
.16s
.04s
11.39s
.13s
.14s
.39s
.17s

Tuesday, June 26, 12

Collaborative filtering

netflix prize

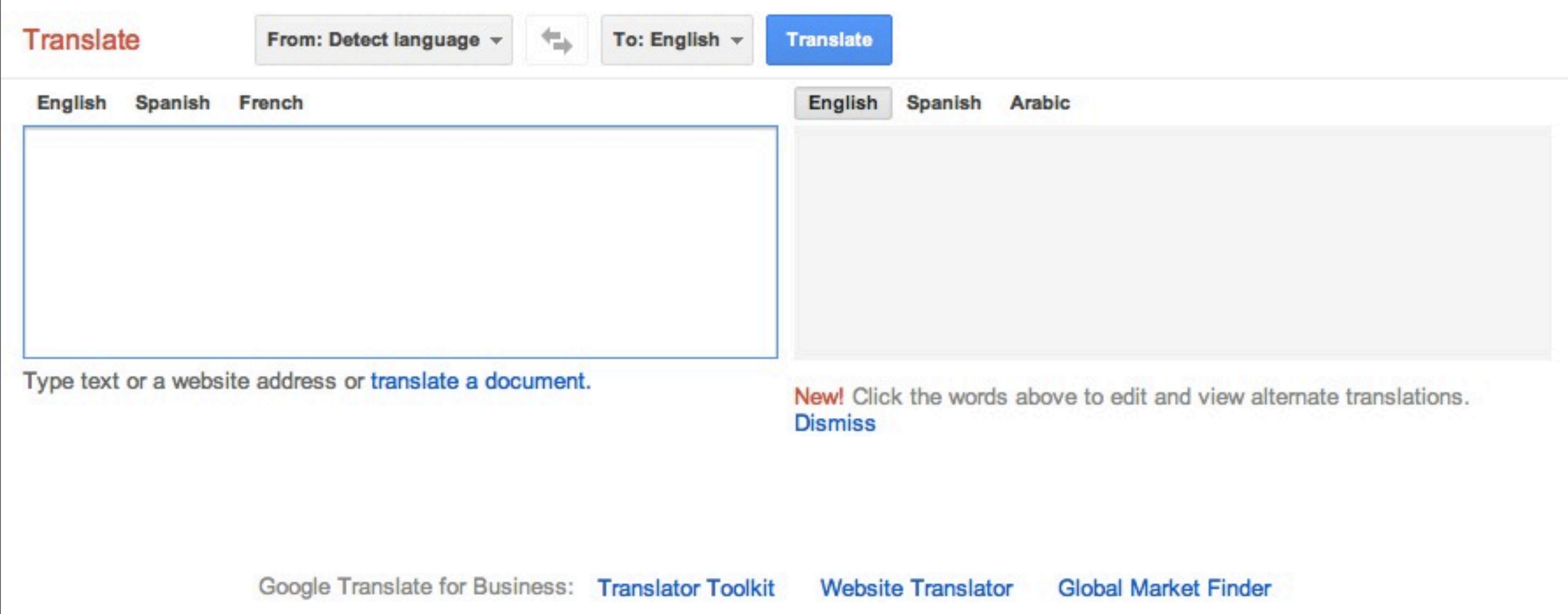
```
CustomerID,MovieID,Rating (1-5), 0 = not rated)
6,16983,0
10,11888,0
10,14584,5
10,15957,0
131,17405,5
134,6243,0
188,12365,0
368,16002,5
424,15997,0
477,12080,0
491,7233,3
508,15929,0
527,1046,2
596,15294,0
```


1. A user expresses his or her preferences by rating items (eg. books, movies or CDs) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.

2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.

3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

Other cool stuff I just thought of

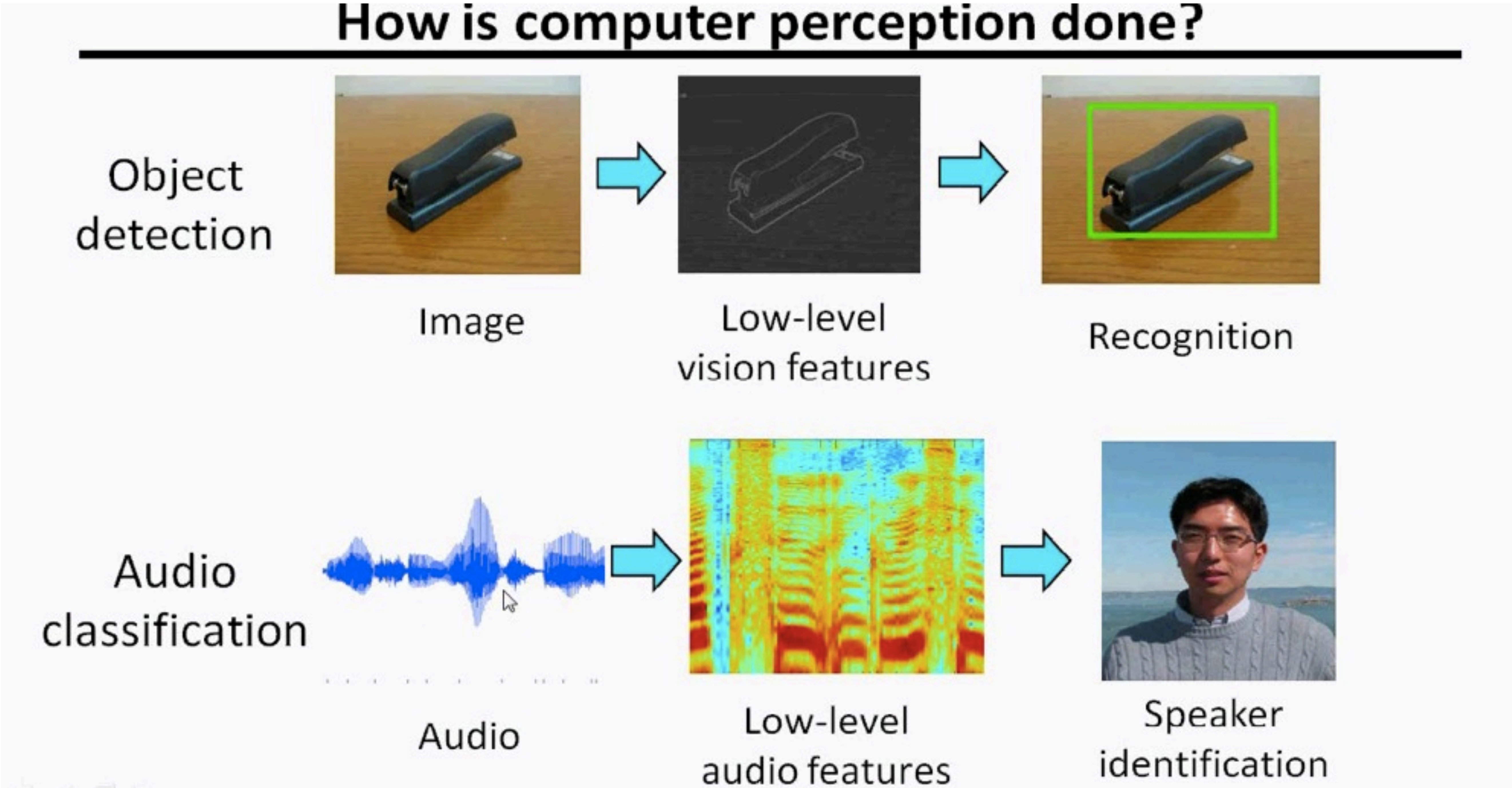

Translate
From: Detect language
To: English
Translate
English
Spanish
French
English
Spanish
Arabic
Type text or a website address or translate a document.
New! Click the words above to edit and view alternate translations.
Dismiss
Google Translate for Business:
Translator Toolkit
Website Translator
Global Market Finder

Deep Learning

Tuesday, June 26, 12

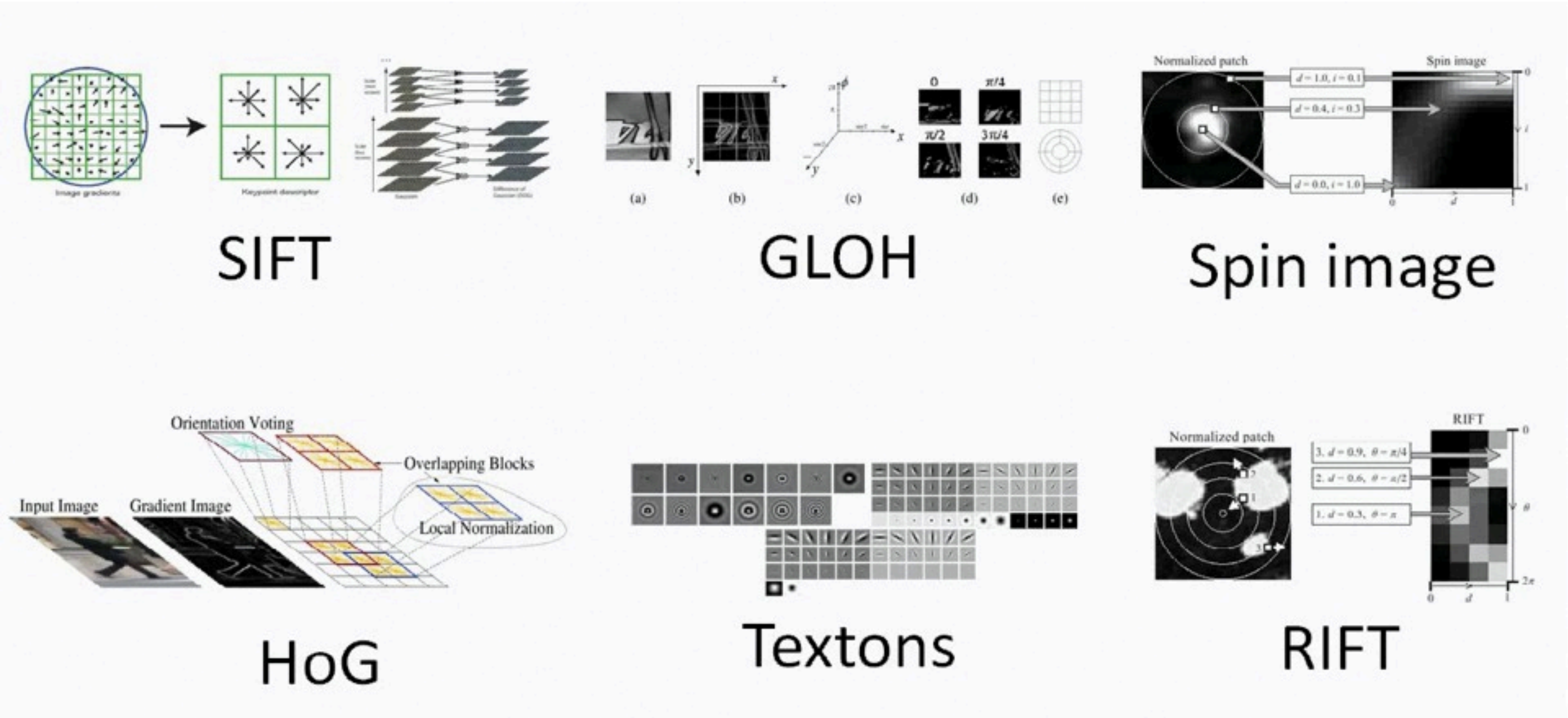
unsupervised neural networks

autoencoder



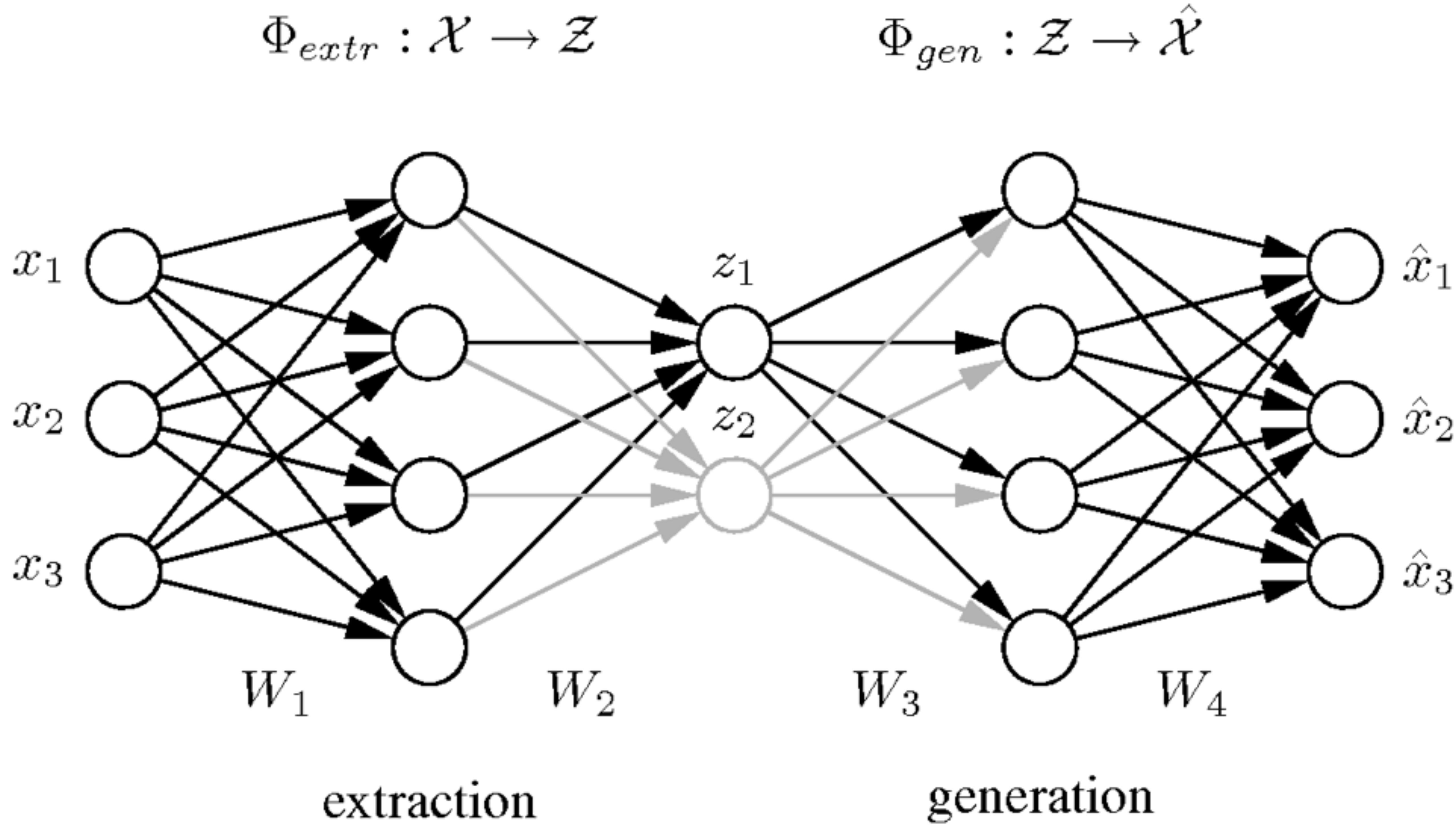
Tuesday, June 26, 12

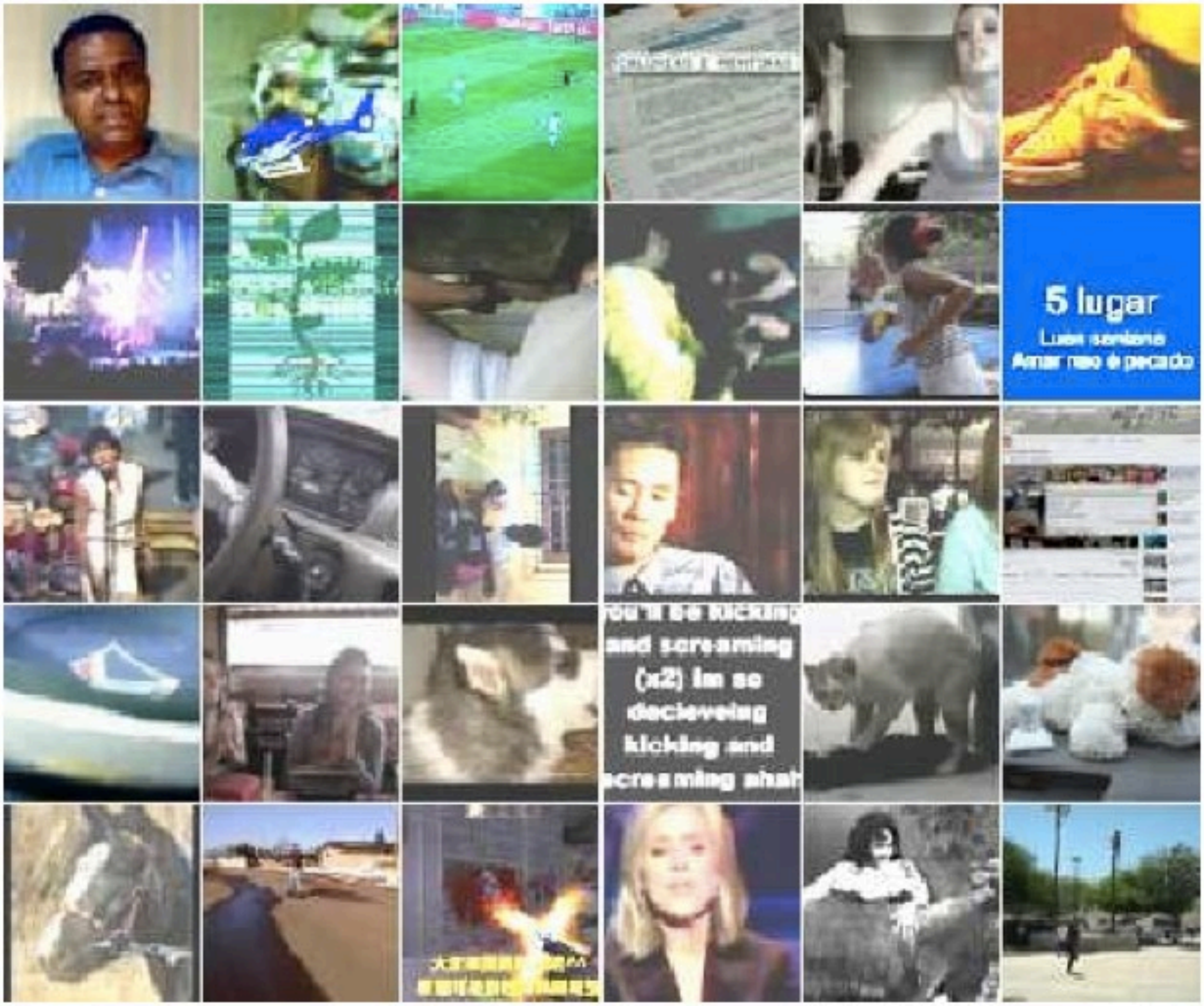
Computer Vision Features

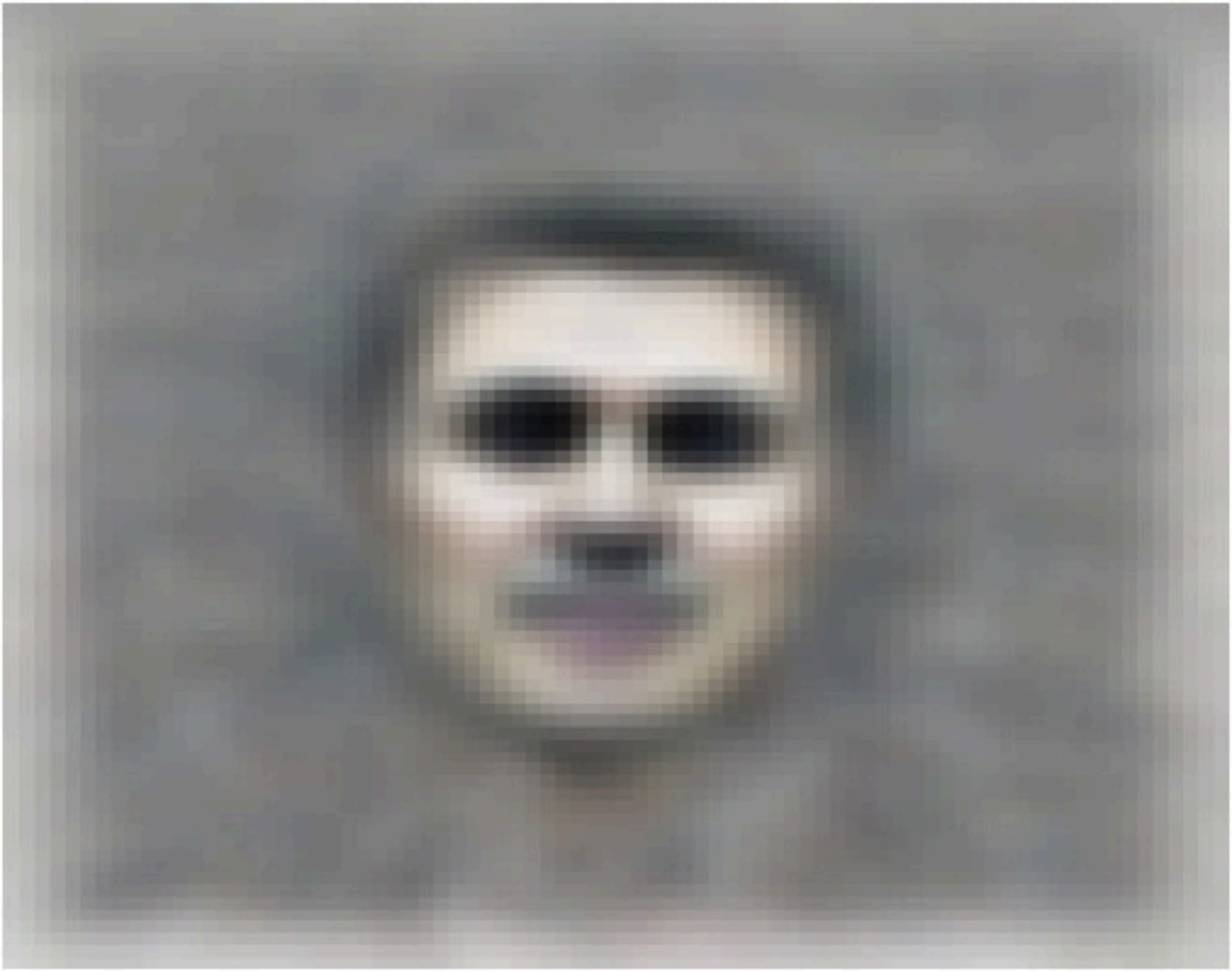


Tuesday, June 26, 12

Autoencoder Neural Network




5 lugar
and screaming
(x2) im so
decieveing
kicking and



THE END