

CSCI 475: Deep Learning

1.	General course information		
1.1	School: Science and Technology	1.6	Credits (ECTS): 8
1.2	Course Title: Deep Learning	1.7	Course Code: CSCI 475
1.3	Pre-requisites: CSCI 353 Programming Paradigms	1.8	Effective from: Fall 2018
1.4	Co-requisites: N/A		
1.5			
<u>Computer Science</u>			

Course Information

	Course leader and teaching staff			
Position	Name	Office #	Contact information	
Course Instructor(s)	Adnan YAZICI,	#7425	adnan.yazici@nu.edu.kz	
	Fatih DEMIRCI	#7424	muhammed.demirci@nu.edu.kz	
Teaching Assistant(s)	Sergey SOLTAN		sergey.soltan@nu.edu.kz	

Course Information

9.	Learning and Teaching Methods			
1	Lecture-demonstration by teacher; Class projects; Homeworks			
2	Formal face-to-face lectures and office hours.			
3	Laboratory sessions to support lecture sections and provide with practical hands on experience with digital circuits and boards.			
10.	Summative Assessments			
#	Activity	Date (tentative)	Weighting (%)	CLOs
	Weekly lab assignments		15%	3-6
	Homework		15%	3-6
	Midterm Exam		20%	1-7
	Attendance (Lecture and Labs)		5%	1-7
	Final Project		20%	6-7
	Final Exam		25%	1-7
	Extra Credit (projects well-written in IEEE conf. format and quality)		10%	3-6
Journals (inc. e-journals)		N/A		
Reference Textbooks		<p>We do not have a regular textbook for the course, though the following may be used as an occasional reference:</p> <ul style="list-style-type: none"> Y. Bengio, I. Goodfellow and A. Courville, "Deep Learning", MIT Press, 2016. Geron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", O'Reilly, 2017. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012. 		

Course Outline			
Session	Topics and Assignments	Course Aims	CLOs
Week 1	Introduction to Machine Learning, Deep Learning, and Applications [Introduction to Machine Learning and Deep Learning; Brief history; Benefits, properties of Deep Learning; Applications]	1, 2	2, 3, 4
Week 2	Machine Learning and Optimization [Linear Regression; Solving Optimization: Gradient Descent, Direct Solution; Generalization; K Nearest Neighbor;]	2, 3	1 - 5
Week 3	Artificial Neural Networks [Linear Classifiers and Their Limits; Perceptron Learning; Training a classifier; Cost Functions: 0-1 loss, linear regression, logistic nonlinearity, logistic regression;]	2, 3	1, 2, 5
Week 4	Artificial Neural Networks [Backpropagation; Single layer and Multilayer Perceptron; Pre-training Steps; Weight Initialization; Overfitting & Extrapolation;]	2, 3	1 – 5
Week 5	Introduction to Image Representation, Image Classification, K-NN and Linear Classifiers for Image Classification [Definitions; image types; linear classification-I]	1,2	3,4
Week 6	Convolutional Neural Networks-I [Convolution layer; Stride; Padding; Spatial Dimensions]	2,3	3,4
Week 7	Convolutional Neural Networks-II [Different layers of processing; Pooling; Fully Connected layer] <i>Midterm</i>	2,3	3,4
Week 8	Convolutional Neural Networks-Working Example, Deep Learning Hardware and Software [Constructing CNN for recognizing symbols, CPUs, GPUs, TPUs, NPUs]	2,3	1,3,4,5
Week 9	Working with CNNs in Practice [Data Augmentation, Dropout, Transfer Learning, Power of Small Filters, Convolutions as Matrix Multiplication]	2, 3	3, 4
Week 10	Recurrent Neural Networks and Applications [RNN, LSTM]	2,3	3,4
Week 11	Case Study-I [Widely-used CNN Architectures; LeNet-5, AlexNet, ZFNet, VGGNet, GoogleNet, ResNet]	2,3	1, 3, 4, 5
Week 12	Case Study-II [Example CNN Applications; Age estimation and gender detection from facial images, image segmentation]	2,3	1, 3, 4, 5
Week 13	Student Presentations - I	2, 3	1, 2, 4
Week 14	Student Presentations - II	2,3	1,2,4
Week 15	Student Presentations - III	2,3	1,2,4

Lecture 1: Introduction to Learning

Many slides from Grosse, Fei, Hinton, Ng

What is Intelligence?

- Intelligence
 - Ability to solve problems
- Examples of Intelligent Behaviors or Tasks
 - Classification of texts based on content
 - Heart disease diagnosis
 - Chess playing

What is Machine Learning?

- Machine Learning (Mitchell 1997)
 - Learn from past experiences
 - Improve the performances of intelligent programs
- Definitions (Mitchell 1997)
 - A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at the tasks improves with the experiences

Solving Real World Problems

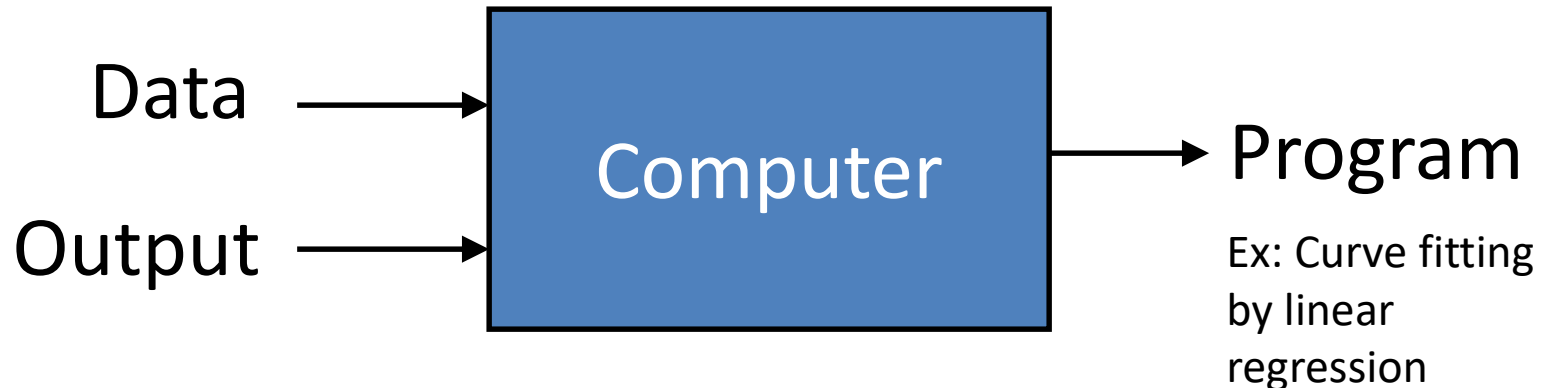
- What is the Input?
 - Features representing the real world data
- What is the Output?
 - Predictions or decisions to be made
- What Is the Intelligent Program?
 - Types of classifiers, value functions, etc.
- How to Learn from experience?
 - Learning algorithms

What is Machine Learning?

Traditional Programming



Machine Learning



Why Machine Learning?

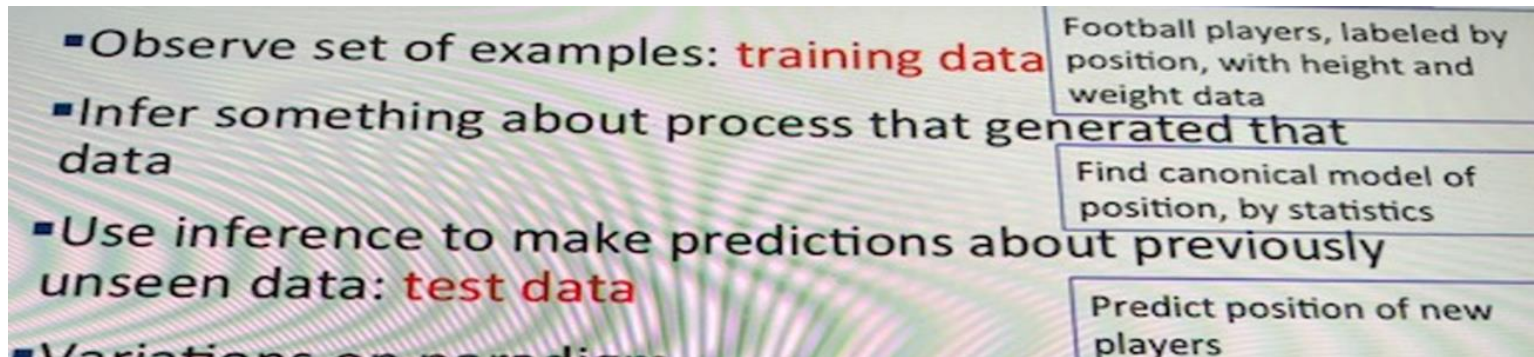
- For many problems, it's difficult to program the correct behavior by hand, e.g.,
 - fraud detection for credit card transactions
 - recommendation systems
 - spam classification
 - etc.
- Instead, we collect lots of examples that specify the correct output for a given input.

Some examples of tasks by machine learning

- Recognizing patterns:
 - Objects in real scenes
 - Facial identities or facial expressions
 - Spoken words
- Recognizing anomalies:
 - Unusual sequences of credit card transactions
 - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
 - Future stock prices or currency exchange rates
 - Which movies will a person like?

Types of Machine Learning

1. **Supervised learning:** have labeled examples of the correct behavior. Given a set of features/label pairs, find a rule that predicts the label associated with a previously unseen input.
 - Prediction
 - Classification (discrete labels), Regression (real values)



Two types of supervised learning

Each training case consists of an input vector x and a target output t .

1. Regression: The target output is a **real number** or a **whole vector of real numbers**. That is, predict a real number associated with a feature vector.

- Use linear regression to fit a curve to data.
 - The price of a stock in 6 months time.
 - The temperature at noon tomorrow.

2. Classification: The target output is a **class label**.

That is, predict a discrete value (label) associated with a feature vector.

- The simplest case is a choice between 1 and 0.
- We can also have multiple alternative labels.

Types of Machine Learning

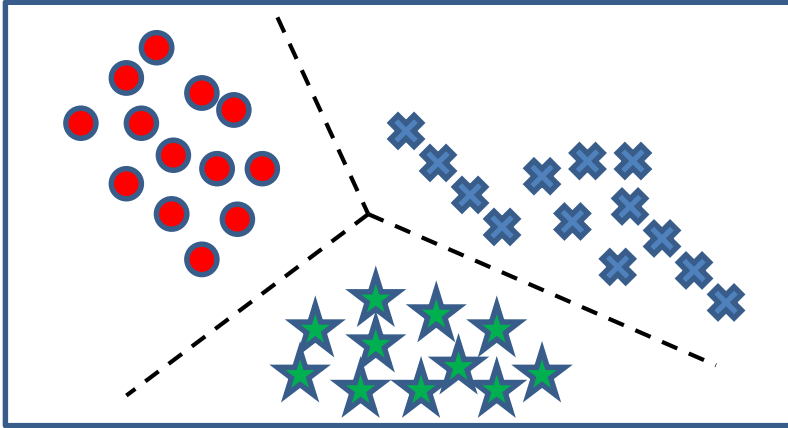
2. Unsupervised learning: no labeled examples. Instead, looking for interesting patterns in the data. Given a set of feature vectors (without labels) group them into “natural clusters” (or create labels for groups).

- Clustering
- Probability distribution estimation
- Finding association (in features)
- Dimension reduction

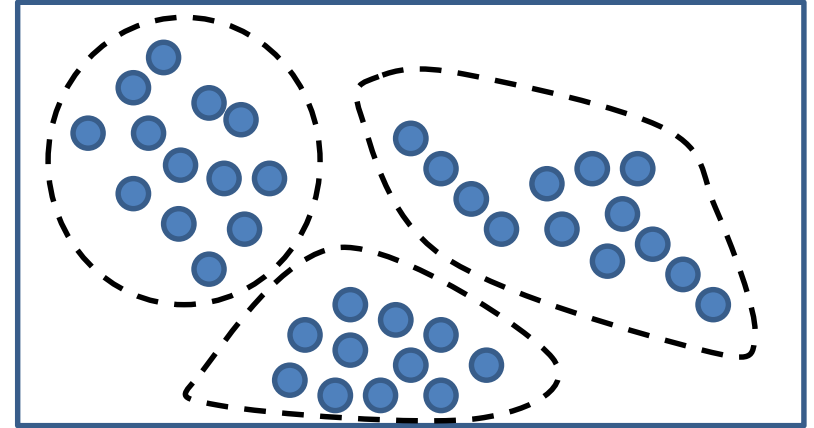
3. Semi-supervised learning: Self-training is a wrapper method for semi-supervised learning. First, a supervised learning algorithm is trained based on the labeled data only. This classifier is then applied to the unlabeled data to generate more labeled examples as input for the supervised learning algorithm.

4. Reinforcement learning: learning system receives a reward signal, tries to learn to maximize the reward signal. Decision making (robot, chess machine)

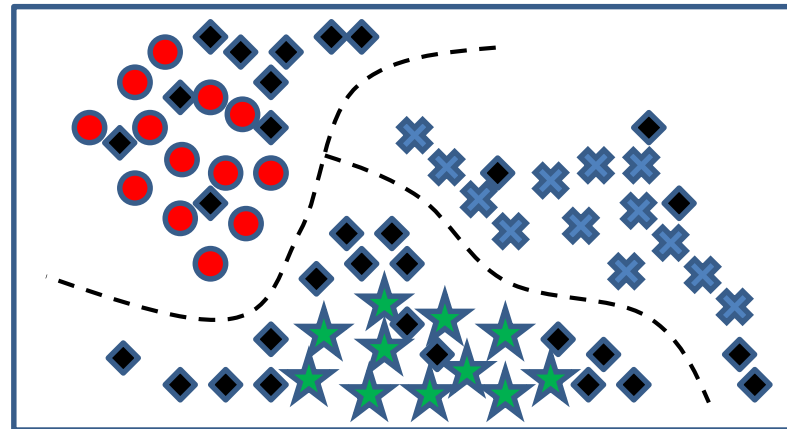
MACHINE LEARNING



Supervised learning

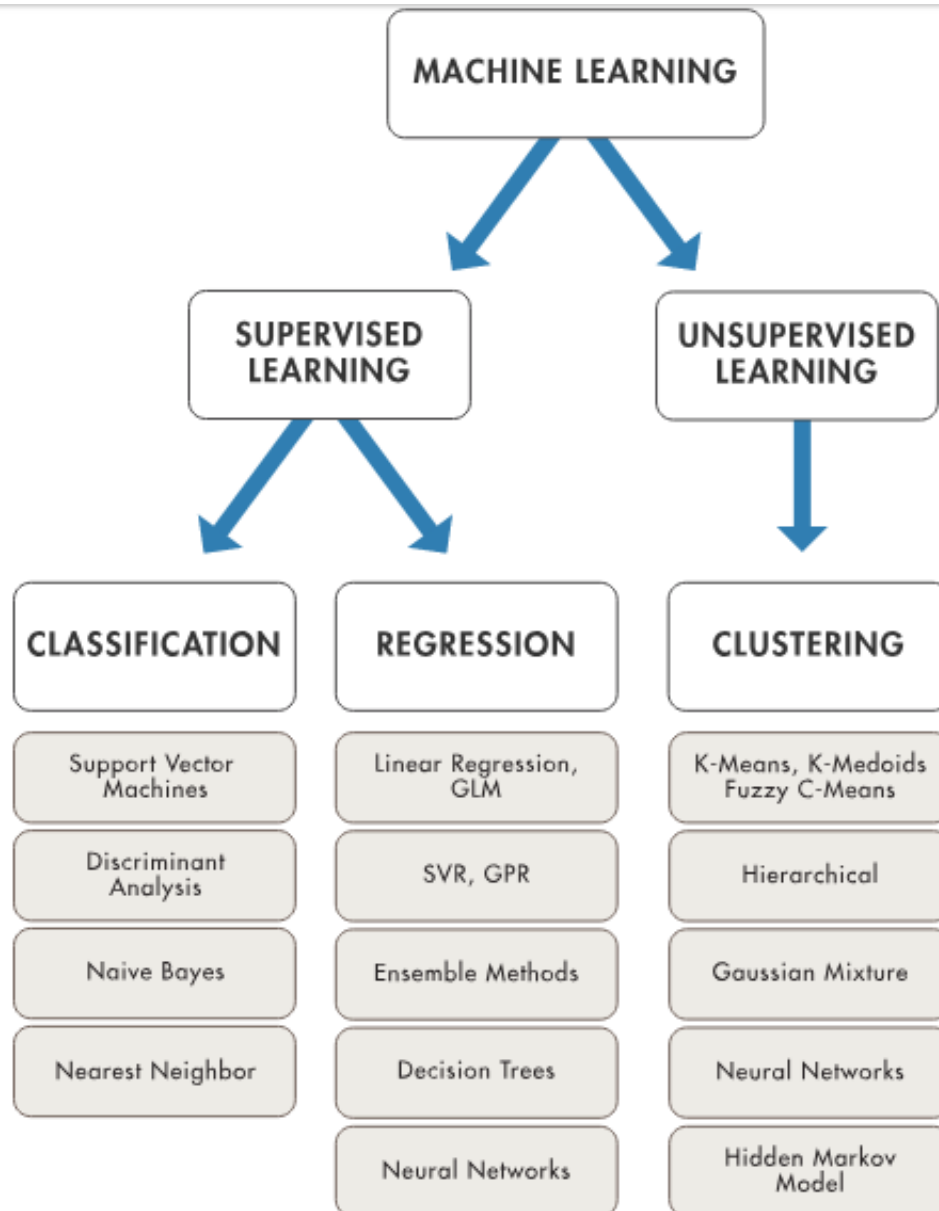


Unsupervised learning



Semi-supervised learning

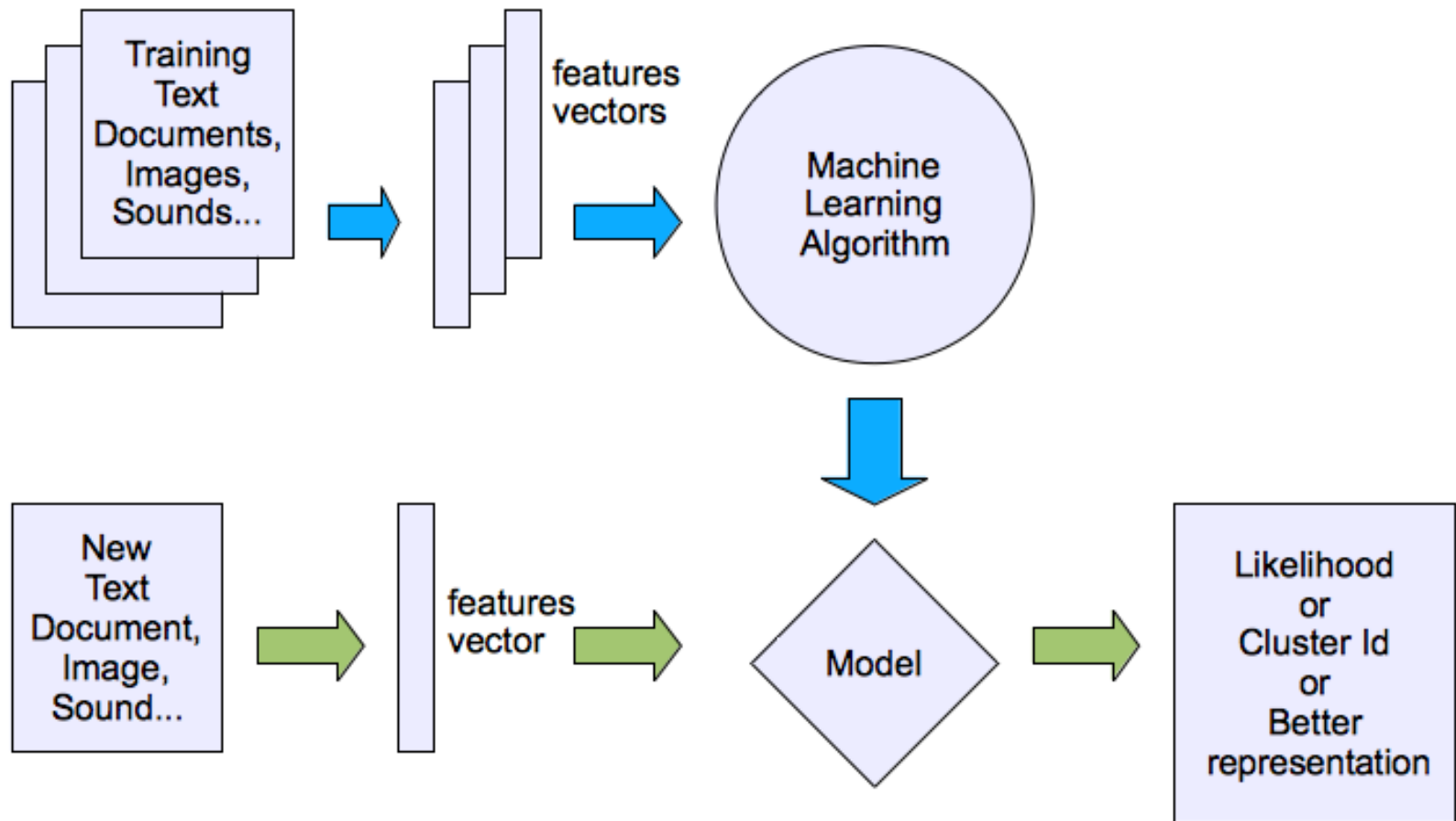
MACHINE LEARNING



Feature Engineering

- Representation of the Real World Data
 - Features: data's attributes which may be useful in prediction
- Feature Transformation and Selection
 - Select a subset of the features
 - Construct new features, e.g.
 - Discretization of real value features
 - Combinations of existing features
- Post Processing to Fit the Classifier
 - Does not change the nature

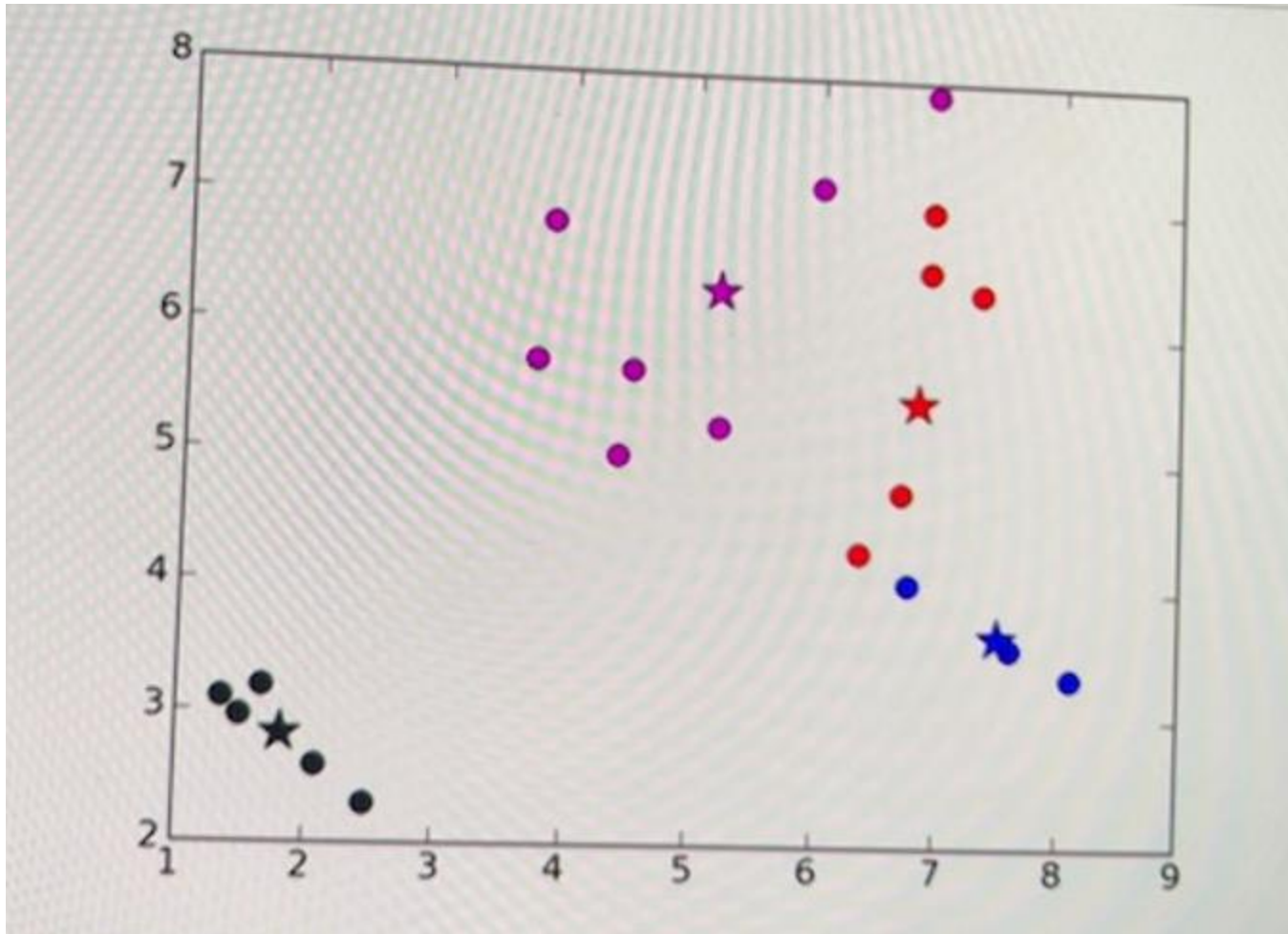
Unsupervised Learning



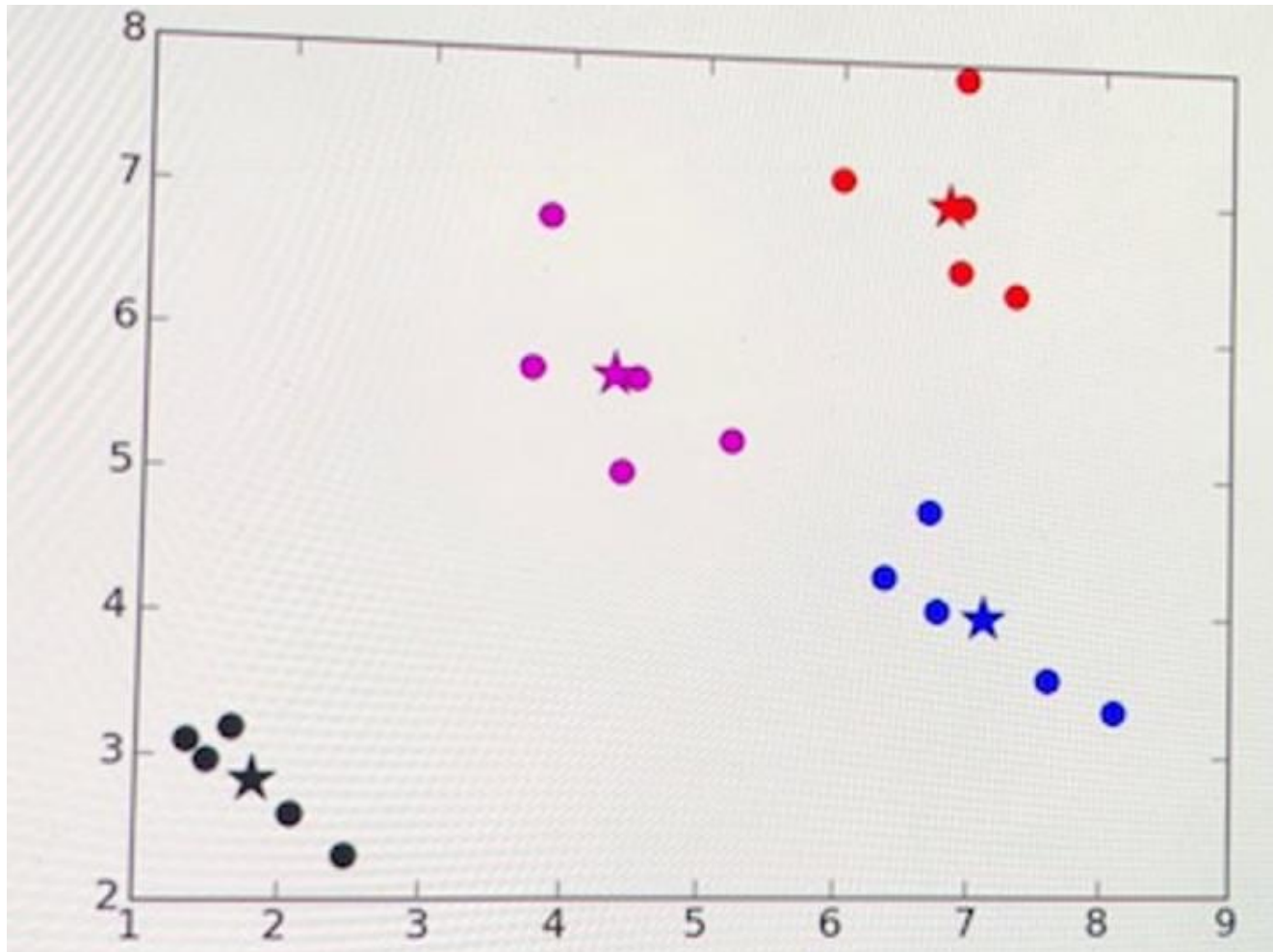
Unsupervised Learning (by Ex.)

- Want to decide on “similarity” of examples, with goal of separating into distinct, “natural”, groups
 - Similarity is a **distance measure**
- Suppose we know that there are k different groups in our training data, but don't know labels (here $k = 2$)
 - Pick k samples (at random?) as exemplars
 - Cluster remaining samples by minimizing distance between samples in same cluster (**objective function**) – put sample in group with closest exemplar
 - Find median example in each cluster as new exemplar
 - Repeat until no change

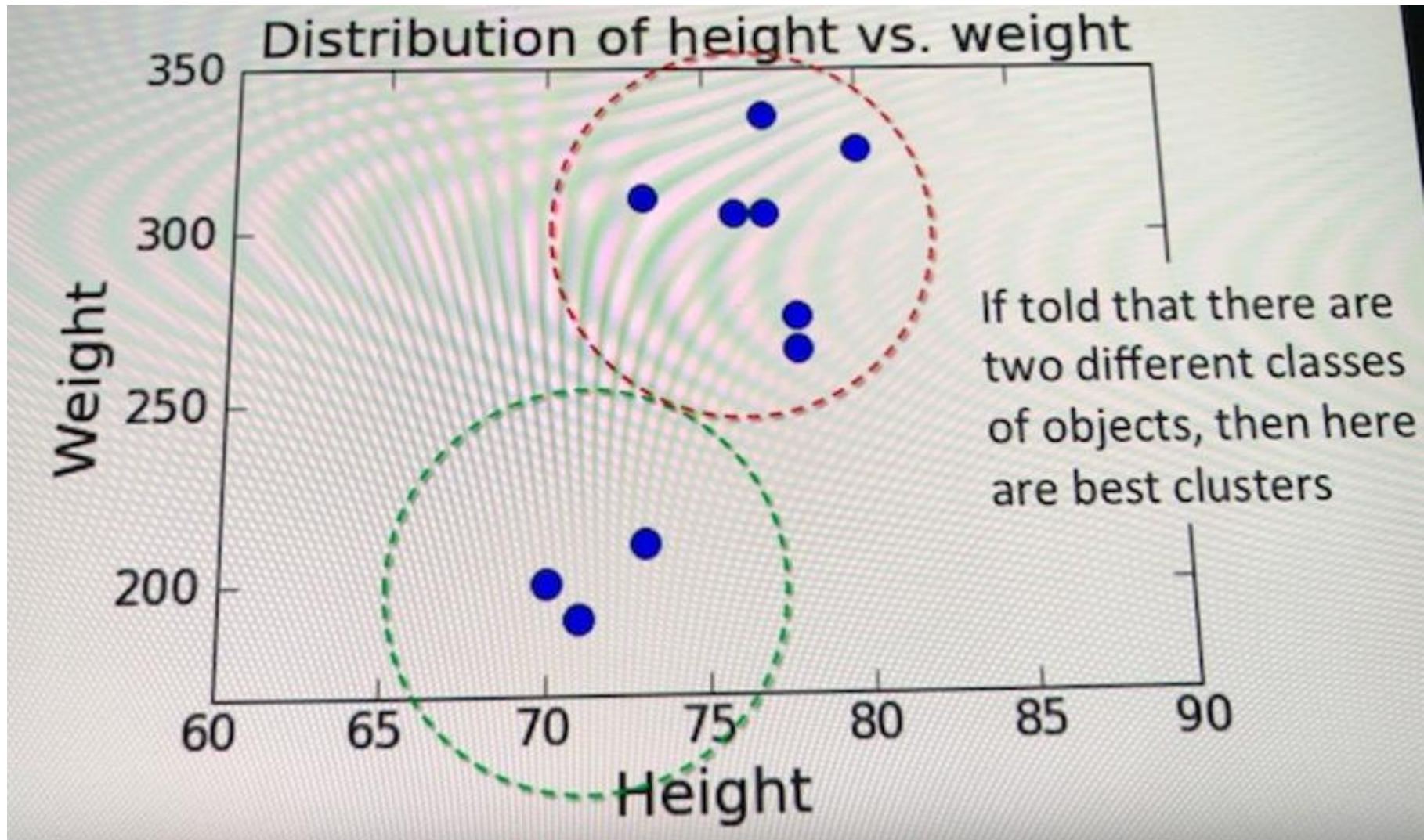
Unsupervised Learning (K-Means)



Unsupervised Learning (K-Means)



Unsupervised Learning (by Ex.)

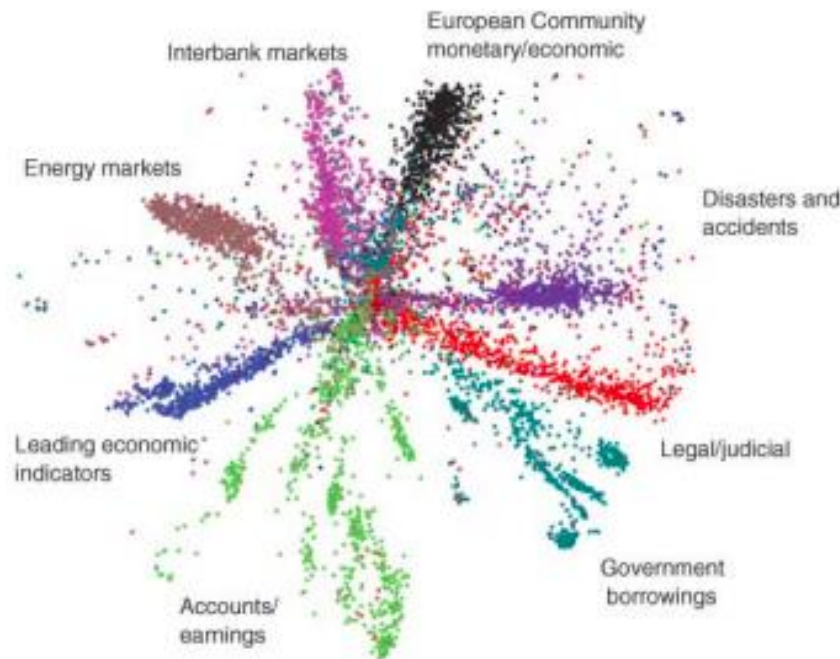


Unsupervised Learning (by Ex.)



Unsupervised Learning

- no labeled examples – instead, looking for interesting patterns in the data
- Example: visualization of documents; algorithm was given 800,000 newswire stories, and learned to represent these documents as points in two-dimensional space



- Colors are based on human labels, but these weren't given to the algorithm

Hierarchical Clustering

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one fewer cluster.
3. Continue the process until all items are clustered into a single cluster of size N .

Hierarchical Clustering

■ Hierarchical:

■ **Agglomerative** (bottom up):

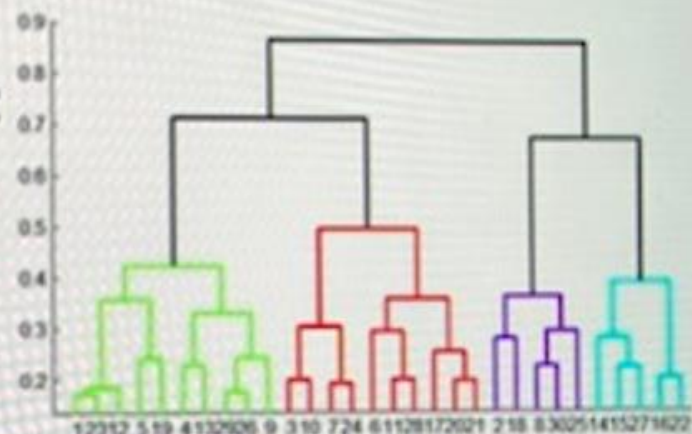
- Initially, each point is a cluster
- Repeatedly combine the two “nearest” clusters into one

■ **Divisive** (top down):

- Start with one cluster and recursively split it

■ This lecture: agglomerative approach

- Same ideas can be used for divisive



Example of Hierarchical Clustering

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{BOS} {NY} {CHI} {DEN} {SF} {SEA}
 {BOS, NY} {CHI} {DEN} {SF} {SEA}
 {BOS, NY, CHI} {DEN} {SF} {SEA}
 {BOS, NY, CHI} {DEN} {SF, SEA}
 {BOS, NY, CHI, **DEN**} {SF, SEA} Single linkage
 or
 {BOS, NY, CHI} {**DEN**, SF, SEA} Complete linkage

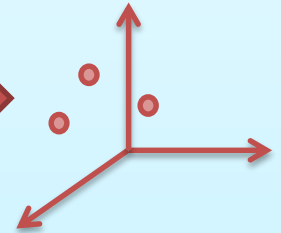
Overview: Supervised Learning

Testing

Test input



Extract
Features

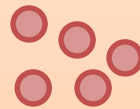


Training

Data with label
(label: human, animal etc.)

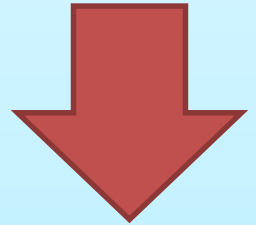


Extract
Features



“Learn”

- Size
- Texture
- Color
- Histogram of oriented gradients
- SIFT
- Etc.



Learned
Models or
Classifiers



Human or Animal or
...

Supervised learning

- have labeled examples of the correct behavior, e.g., handwritten digit classification with the MNIST dataset
 - task: given an image of a digit, predict the digit class
 - 70,000 images of handwritten digits labeled by humans
 - 60,000 used to train the classifier, 10,000 to test its performance
 - This dataset is the “fruit fly” of neural net research
 - Current best algorithm has only 0.23% error rate!

Example1: What makes a 2?

0 0 0 1 1 1 1 1 1 2

2 2 2 2 2 2 2 3 2 3

3 4 4 4 4 4 5 5 5 5

6 6 7 7 7 7 8 8 8

9 9 9 9 9 9 9 9 9

Example2: The ImageNet

IMGENET

www.image-net.org

22K categories and **14M** images

- Animals
 - Bird
 - Fish
 - Mammal
 - Invertebrate
- Plants
 - Tree
 - Flower
 - Food
 - Materials
- Structures
 - Artifact
 - Tools
 - Appliances
 - Structures
- Person
 - Scenes
 - Indoor
 - Geological Formations
 - Sport Activities

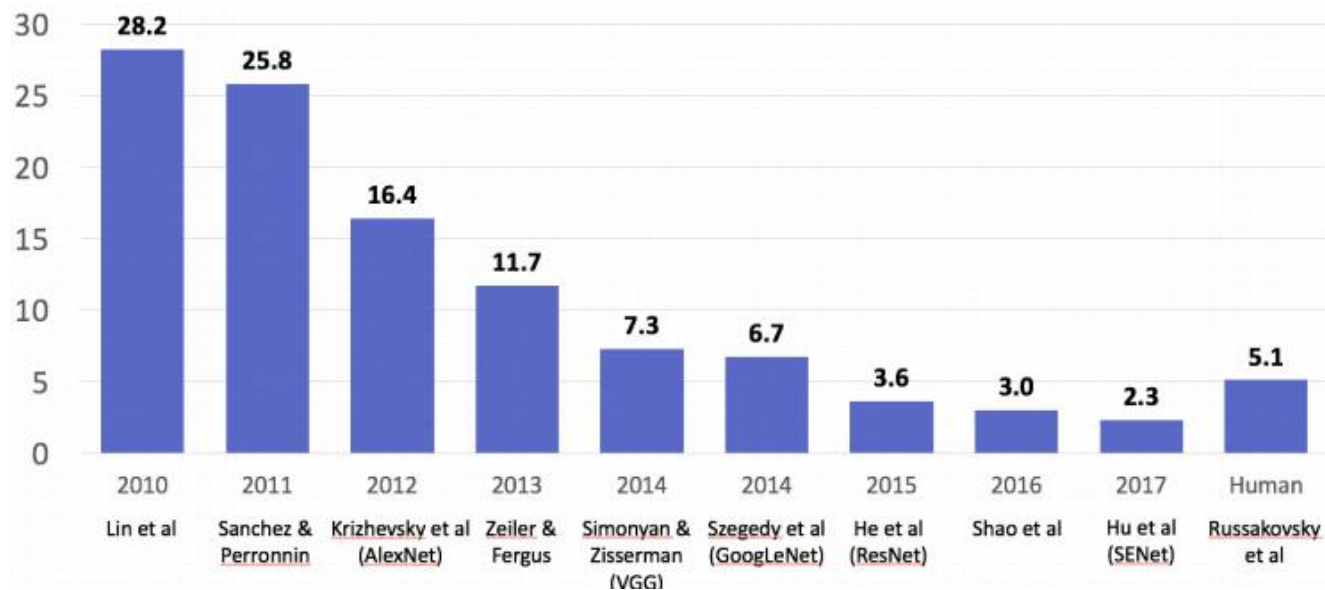
Example2: The ImageNet task

- 1000 different object classes in 1.4 million high-resolution training images from the web.
- Lots of variability in viewpoint, lighting, etc.
 - Best system in 2011 competition got 47% error for its first choice and 25% error for its top 5 choices.



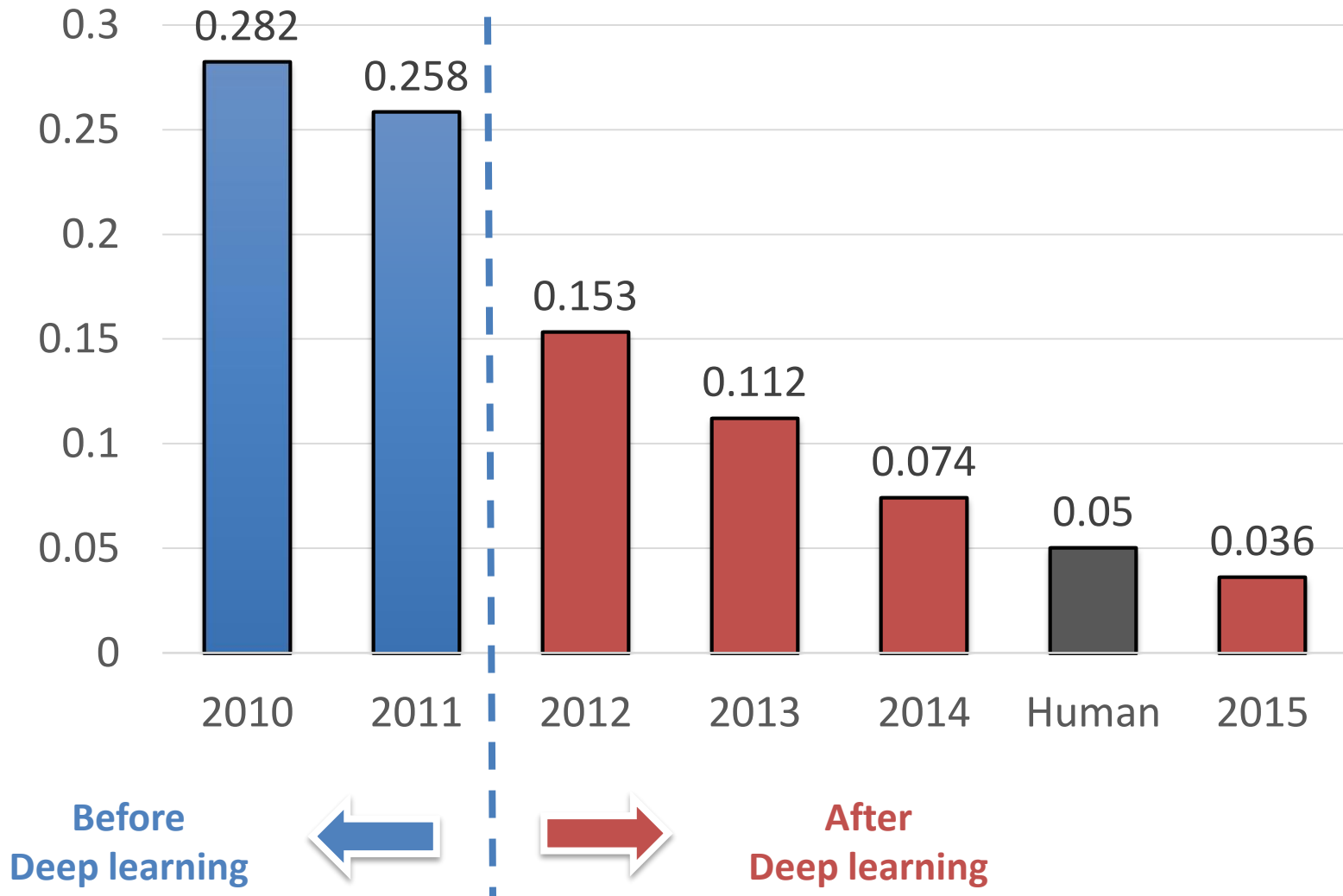
Example2: The ImageNet task

- A very deep neural net (Krizhevsky et. al. 2012 - AlexNet) gets less than 40% error for its first choice and less than 20% for its top 5 choices
- We will talk about this later in the course



Object Recognition

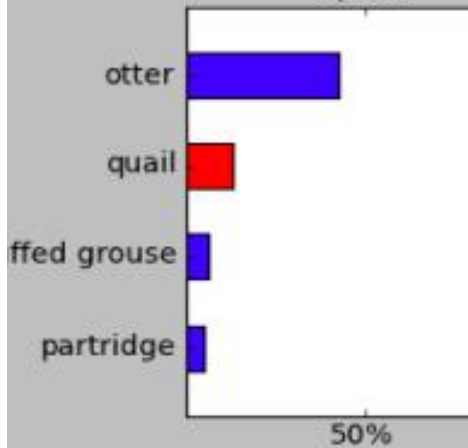
ImageNet Winners



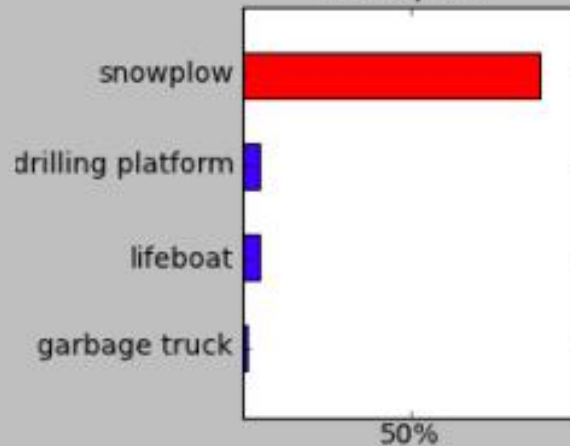
Some Examples of ImageNet



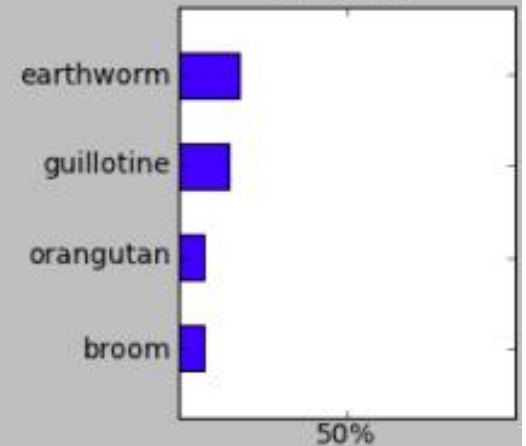
quail



snowplow



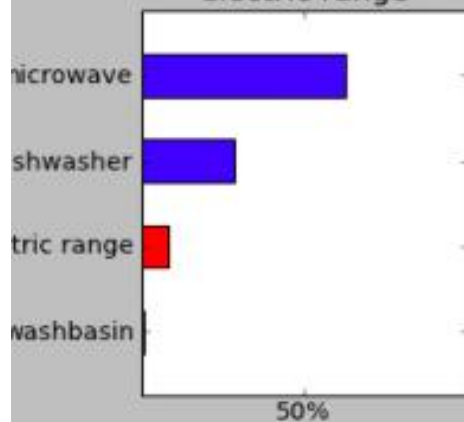
scabbard



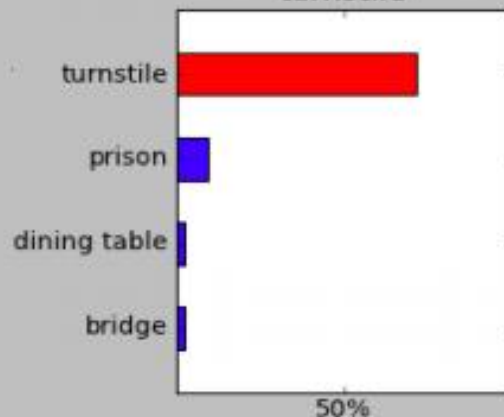
Some Examples of ImageNet



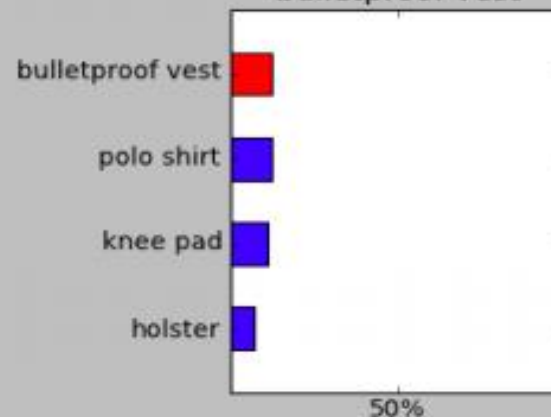
electric range



turnstile



bulletproof vest

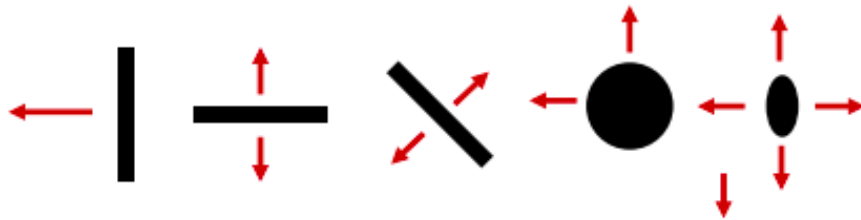


Example3: Speech Recognition

- 2009: Hinton and his group developed and trained a neural network for speech recognition.
- This network outperformed state-of-the-art on a medium sized speech recognition dataset.
- This framework was used in Android OS in 2012.

[Mohamed, Dahl, Hinton: Deep belief networks for phone recognition, 2009.]

Hubel & Wiesel, 1959



Simple cells:
Response to light
orientation

Complex cells:
Response to light
orientation and movement

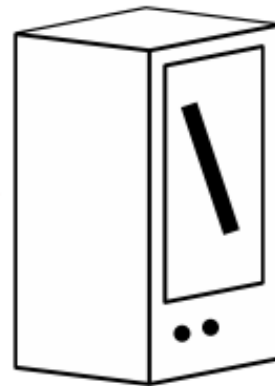
Hypercomplex cells:
response to movement
with an end point



No response



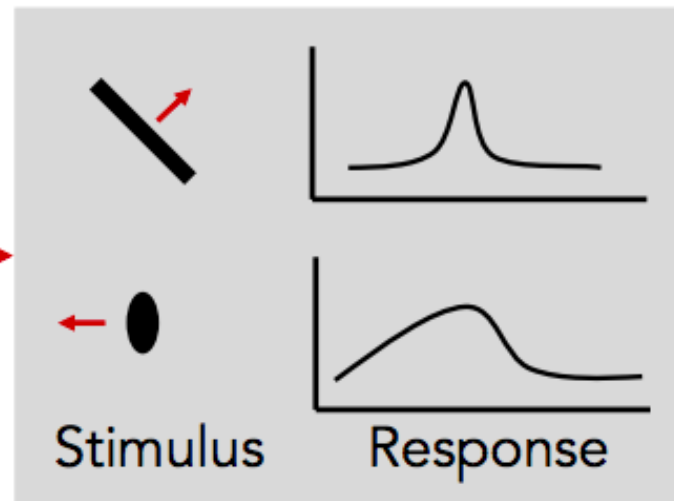
Response
(end point)



Stimulus

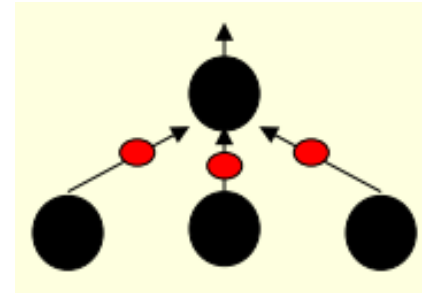


Electrical
signal from
brain



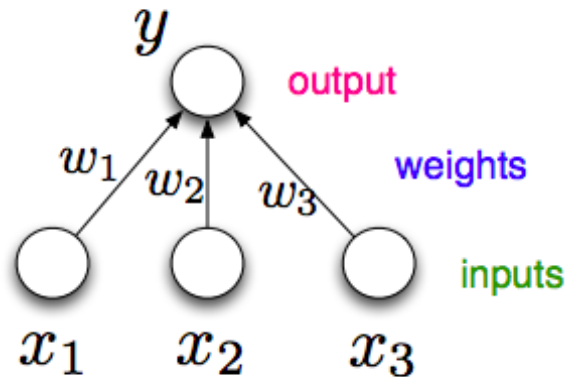
How does the brain work ?

- Each neuron receives inputs from other neurons
- The effect of each input line on the neuron is controlled by a synaptic weight
 - The weights can be positive or negative.
- The synaptic weights adapt so that the whole network learns to perform useful computations
 - Recognizing objects, understanding language, making plans, controlling the body.
- You have about 10^{11} neurons each with about 10^{14} weights.
- A huge number of weights can affect the computation in a very short time.



What are neural networks?

- Most of the biological details aren't essential, so we use vastly simplified models of neurons.
- While neural nets originally drew inspiration from the brain, nowadays we mostly think about math, statistics, etc



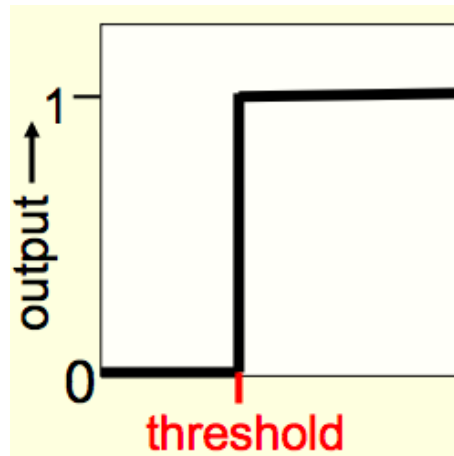
$$y = g \left(b + \sum_i x_i w_i \right)$$

A diagram of the mathematical equation for a neuron's output. The equation is $y = g \left(b + \sum_i x_i w_i \right)$. Colored arrows point to different parts of the equation: a pink arrow points to y (labeled "output"), a red arrow points to g (labeled "nonlinearity"), a blue arrow points to b (labeled "bias"), a green arrow points to x_i (labeled "i'th input"), and a purple arrow points to w_i (labeled "i'th weight").

- Neural networks are collections of thousands (or millions) of these simple processing units that together perform useful computations.

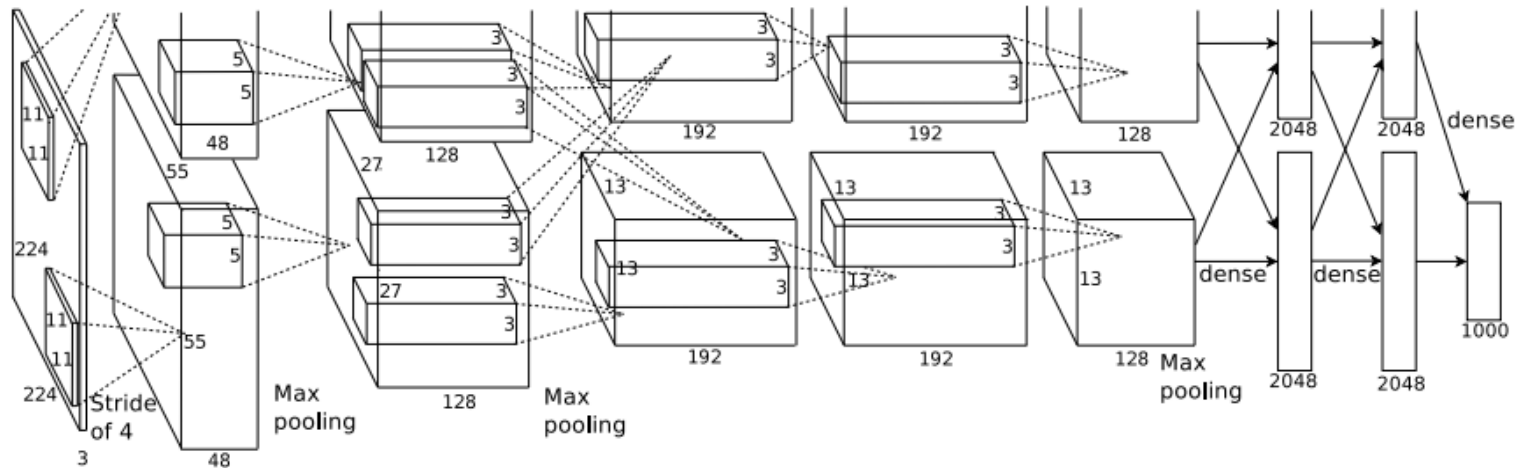
Binary threshold neurons

- McCulloch-Pitts (1943): influenced Von Neumann.
 - First compute a weighted sum of the inputs.
 - Then send out a fixed size spike of activity if the weighted sum exceeds a threshold.



Deep learning

- Deep learning: many layers (stages) of processing
- E.g. this network which recognizes objects in images:



(Krizhevsky et al., 2012)

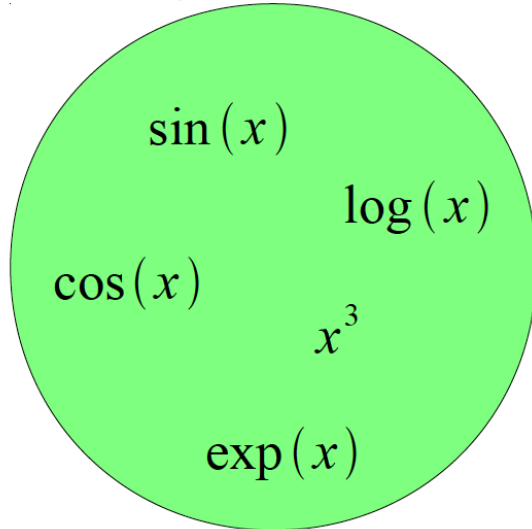
- Each of the boxes consists of many neurons similar to the one on the previous slides


So what *does* Deep (Machine) Learning bring/provide?

- 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

Building a Complicated Function

Given a library of simple functions

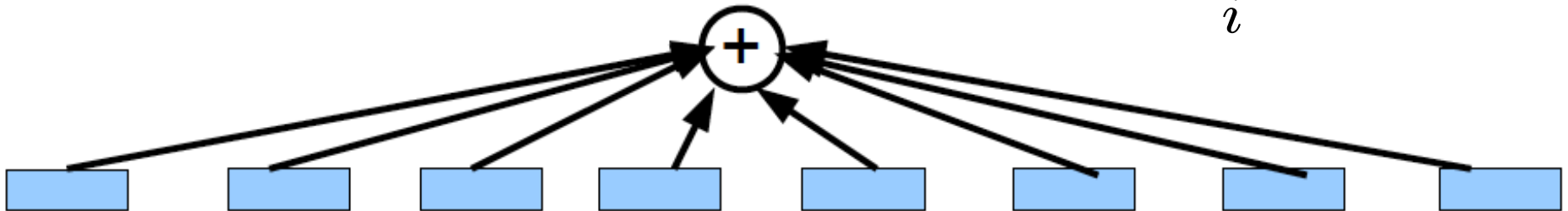


Compose into a

complicate function

Idea 1: Linear Combinations

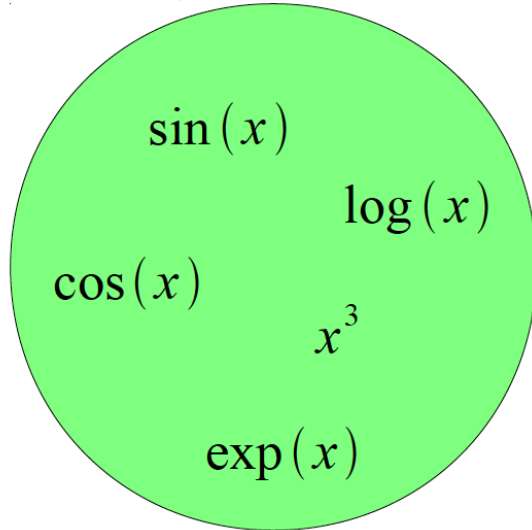
- Boosting
- Kernels
- ...

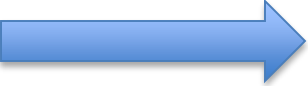
$$f(x) = \sum_i \alpha_i g_i(x)$$



Building A Complicated Function

Given a library of simple functions

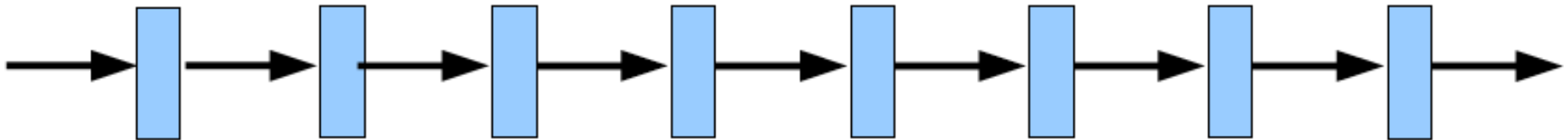


Compose into a

complicate function

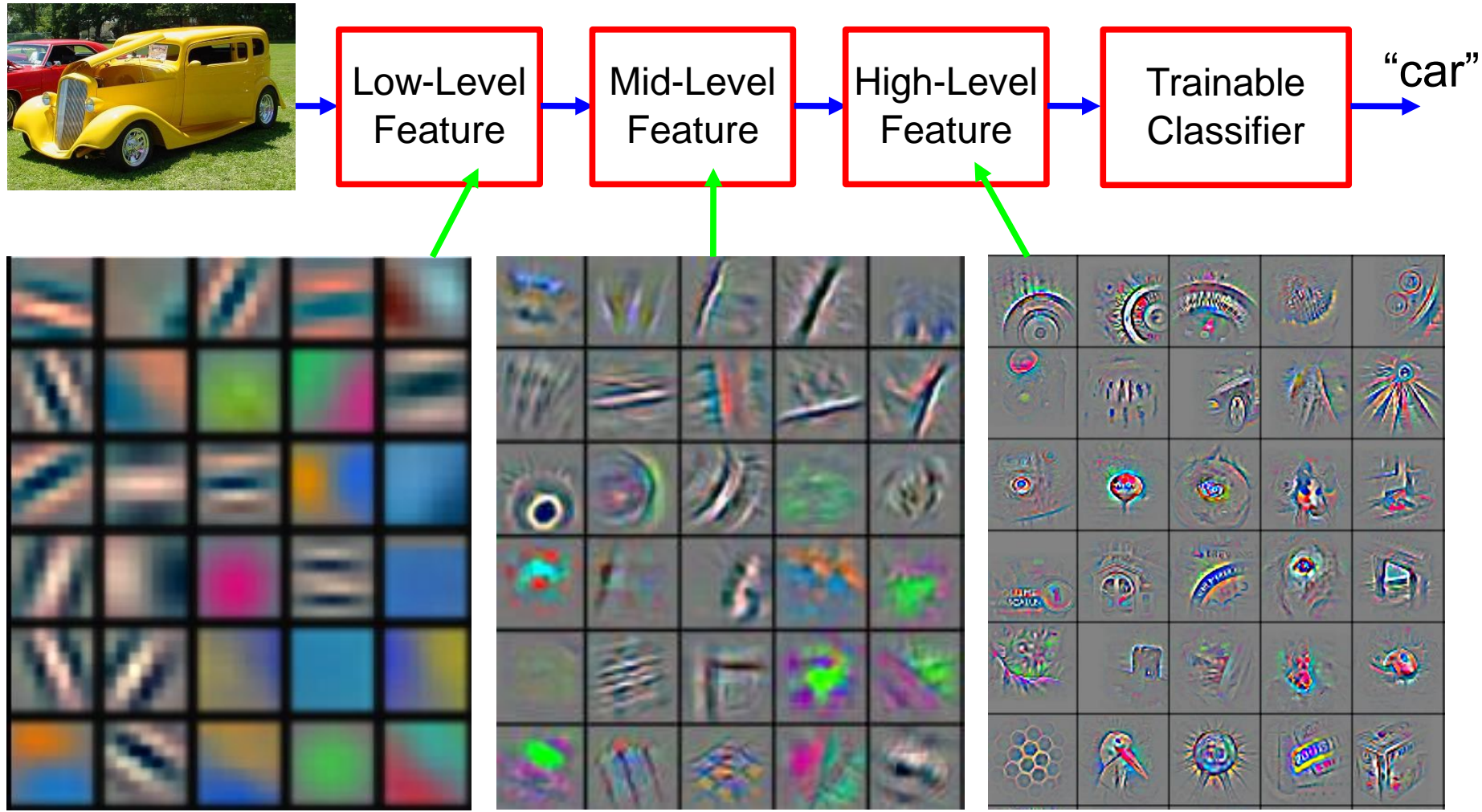
Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms...

$$f(x) = g_1(g_2(\dots(g_n(x)\dots)))$$



Deep Learning = Hierarchical Compositionality



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

Amazing Results and Applications

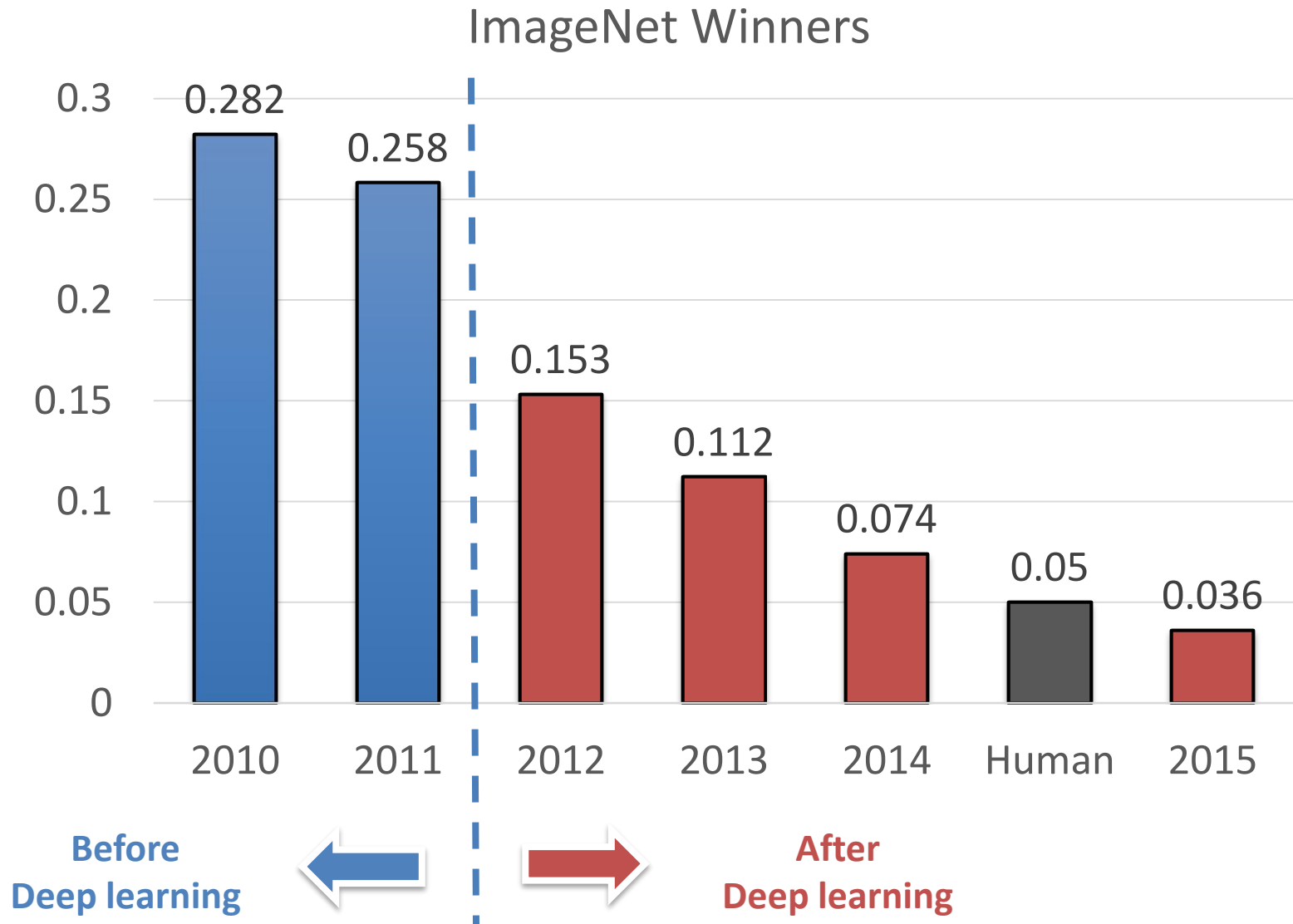
ImageNet Object Recognition Challenge

(<http://image-net.org/challenges/LSVRC/>)



Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012.

Object Recognition





Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]

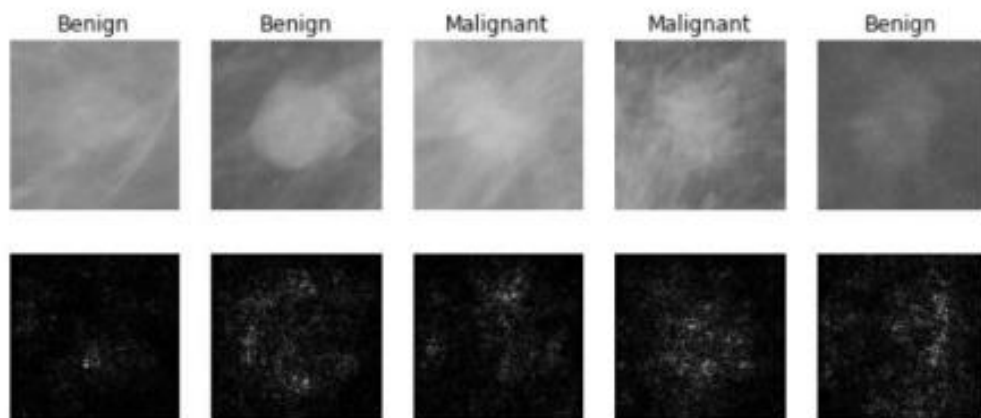


Figure copyright Levy et al. 2016.
Reproduced with permission.

[Levy et al. 2016]



From left to right: [public domain by NASA](#), usage [permitted](#) by ESA/Hubble, [public domain by NASA](#), and [public domain](#).

[Dieleman et al. 2014]



[Sermanet et al. 2011]
[Ciresan et al.]

Photos by Lane McIntosh.
Copyright CS231n 2017.

No errors



A white teddy bear sitting in the grass

Minor errors



A man in a baseball uniform throwing a ball

Somewhat related



A woman is holding a cat in her hand

Image Captioning

[Vinyals et al., 2015]
[Karpathy and Fei-Fei, 2015]



A man riding a wave on top of a surfboard



A cat sitting on a suitcase on the floor

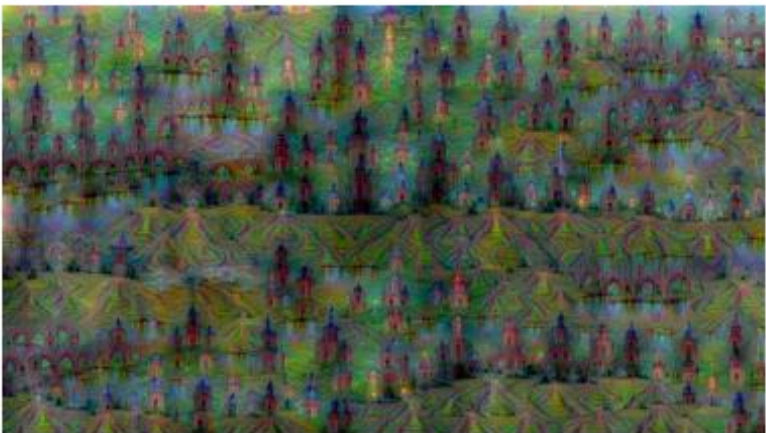
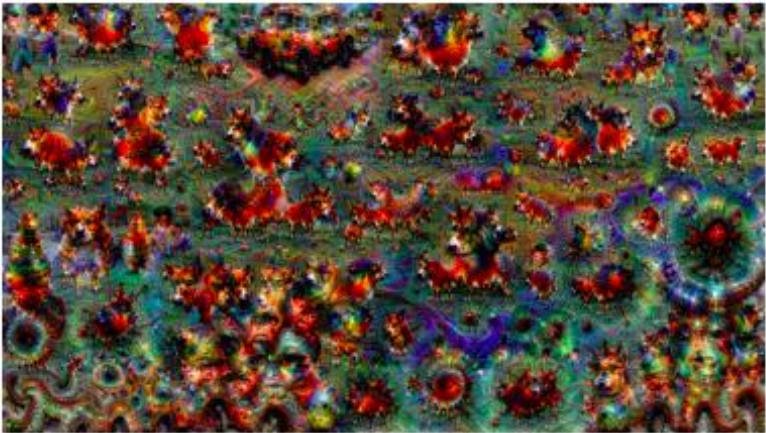


A woman standing on a beach holding a surfboard

All images are CC0 Public domain:

<https://pixabay.com/en/luggage-antique-cat-1643010/>
<https://pixabay.com/en/teddy-plush-bears-cute-teddy-bear-1623436/>
<https://pixabay.com/en/surf-wave-summer-sport-litoral-1666716/>
<https://pixabay.com/en/woman-female-model-portrait-adult-983967/>
<https://pixabay.com/en/handstand-lake-meditation-496008/>
<https://pixabay.com/en/baseball-player-shortstop-infield-1045263/>

Captions generated by Justin Johnson using [NeuralTalk2](#)



Figures copyright Justin Johnson, 2015. Reproduced with permission. Generated using the Inceptionism approach from a [blog post](#) by Google Research.



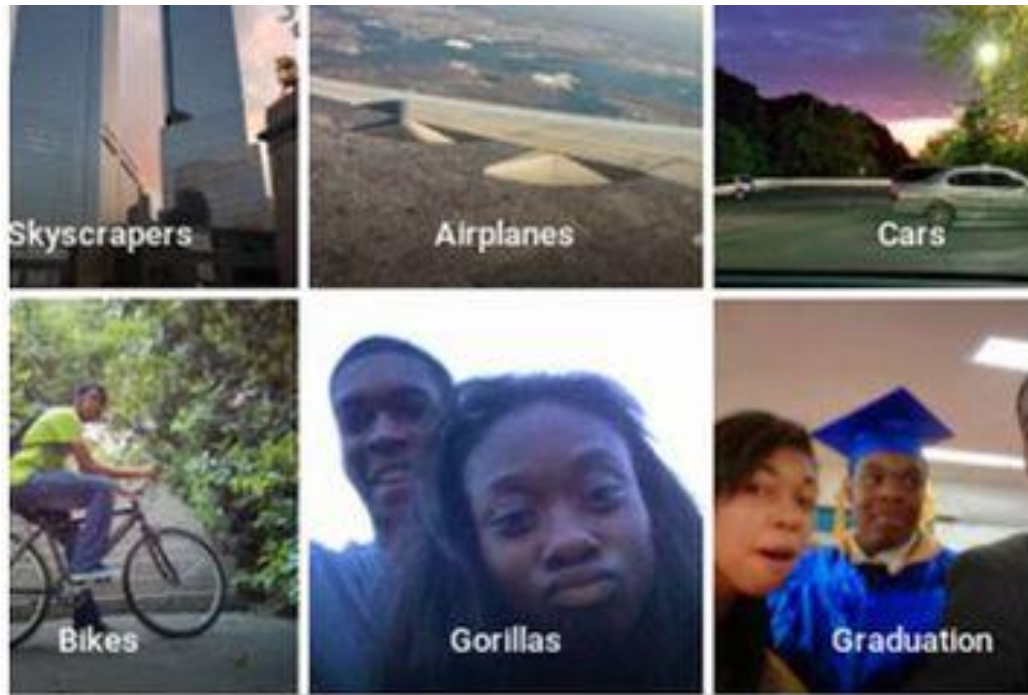
[Original image](#) is CC0 public domain
[Starry Night](#) and [Tree Roots](#) by Van Gogh are in the public domain
[Rokesh image](#) is in the public domain
 Stylized images copyright Justin Johnson, 2017; reproduced with permission



Gatys et al, "Image Style Transfer using Convolutional Neural Networks", CVPR 2016
 Gatys et al, "Controlling Perceptual Factors in Neural Style Transfer", CVPR 2017



It can make mistakes



<http://blogs.wsj.com/digits/2015/07/01/google-mistakenly-tags-black-people-as-gorillas-showing-limits-of-algorithms/>

TWEETS
587

FOLLOWING
18

FOLLOWERS
746

FAVORITES
13



INTERESTING.JPG @INTERESTING_JPG · Feb 20

a surfboard attached to the top of a car .



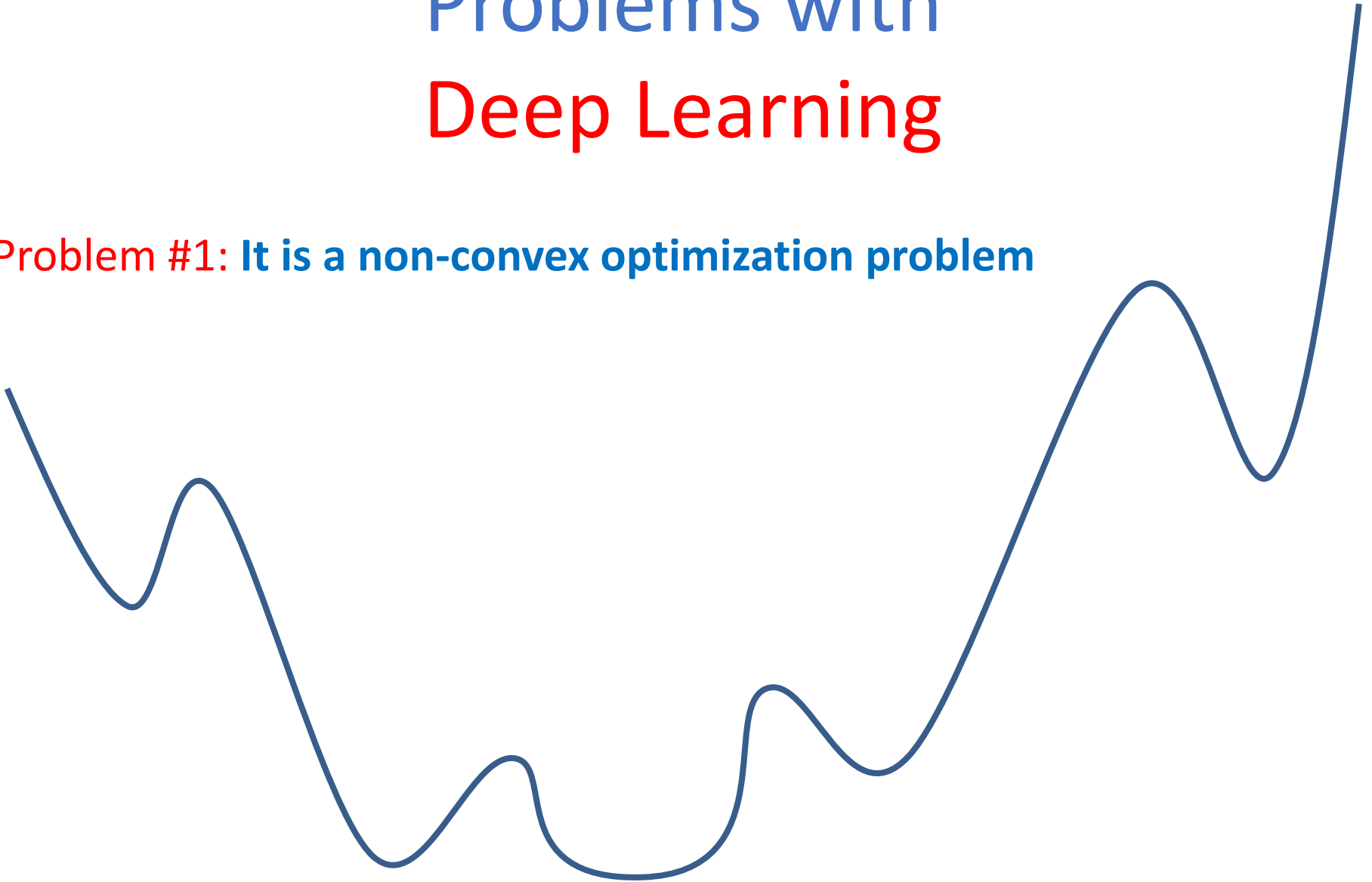
[View more photos and videos](#)

Results from @INTERESTING_JPG via <http://deeplearning.cs.toronto.edu/i2t>

PROBLEMS WITH DEEP LEARNING

Problems with Deep Learning

Problem #1: It is a non-convex optimization problem



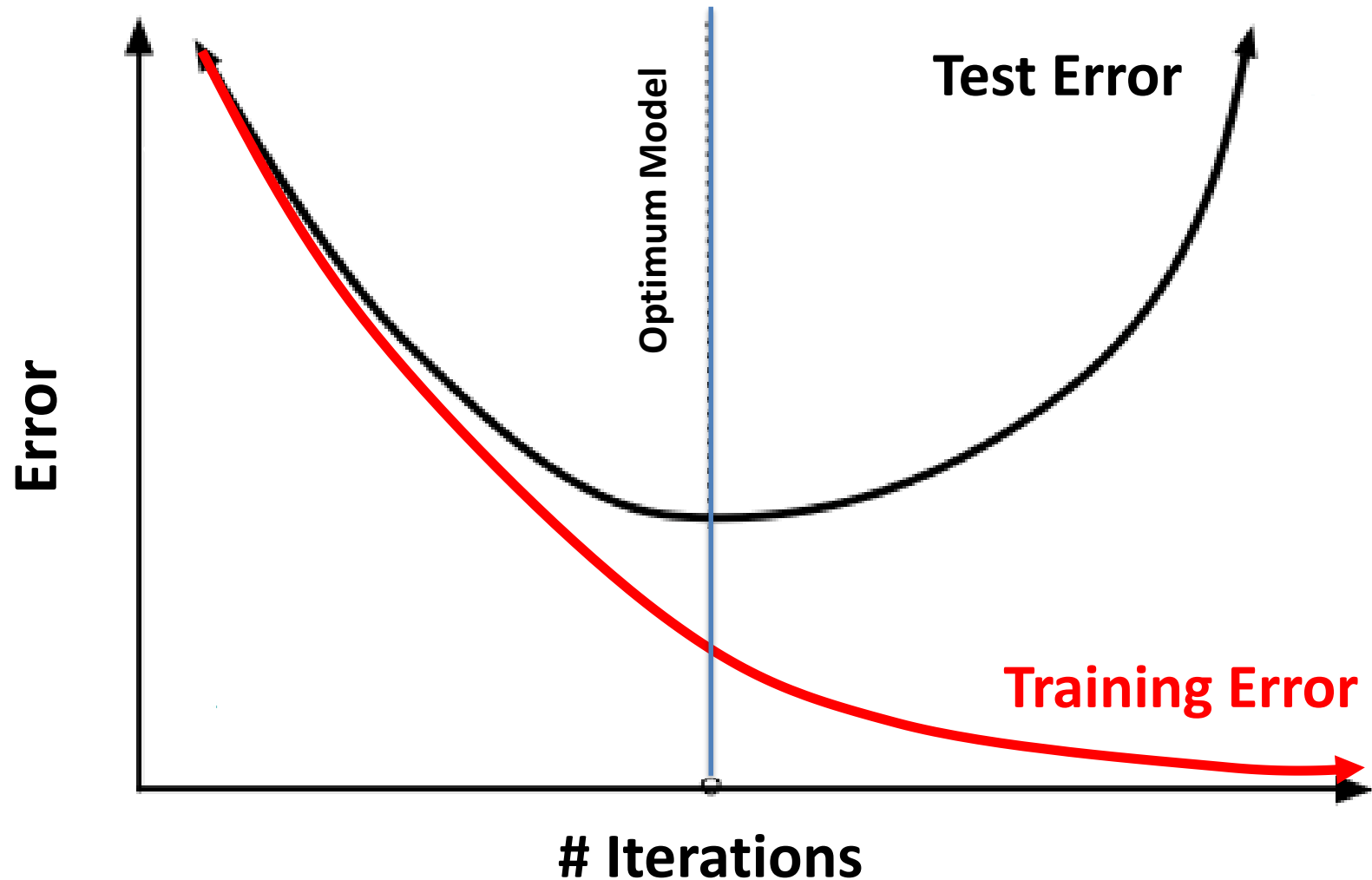
Problems with Deep Learning

- Problem#2: Hard to track down what's failing
 - Pipeline systems have “oracle” performances at each step
 - In end-to-end systems, it's hard to know why things are not working
- Standard response #1
 - Tricks of the trade: visualize features, add losses at different layers, pre-train to avoid degenerate initializations...
 - “We're working on it”
- Standard response #2
 - “Yes, but it often works!”

Problems with Deep Learning

- Problem#3: Lack of easy reproducibility
 - Direct consequence of stochasticity & non-convexity
- Standard response #1
 - It's getting much better
 - Standard toolkits/libraries/frameworks now available
 - Caffe, Theano, Torch
- Standard response #2
 - “Yes, but it often works!”

Problems with Deep Learning: Memorization



Problems with Deep Learning

Requires too much data

Requires too much computational power

Big players

Google, Microsoft, Facebook, Apple, Amazon, Intel, Nvidia

Why does it work now?

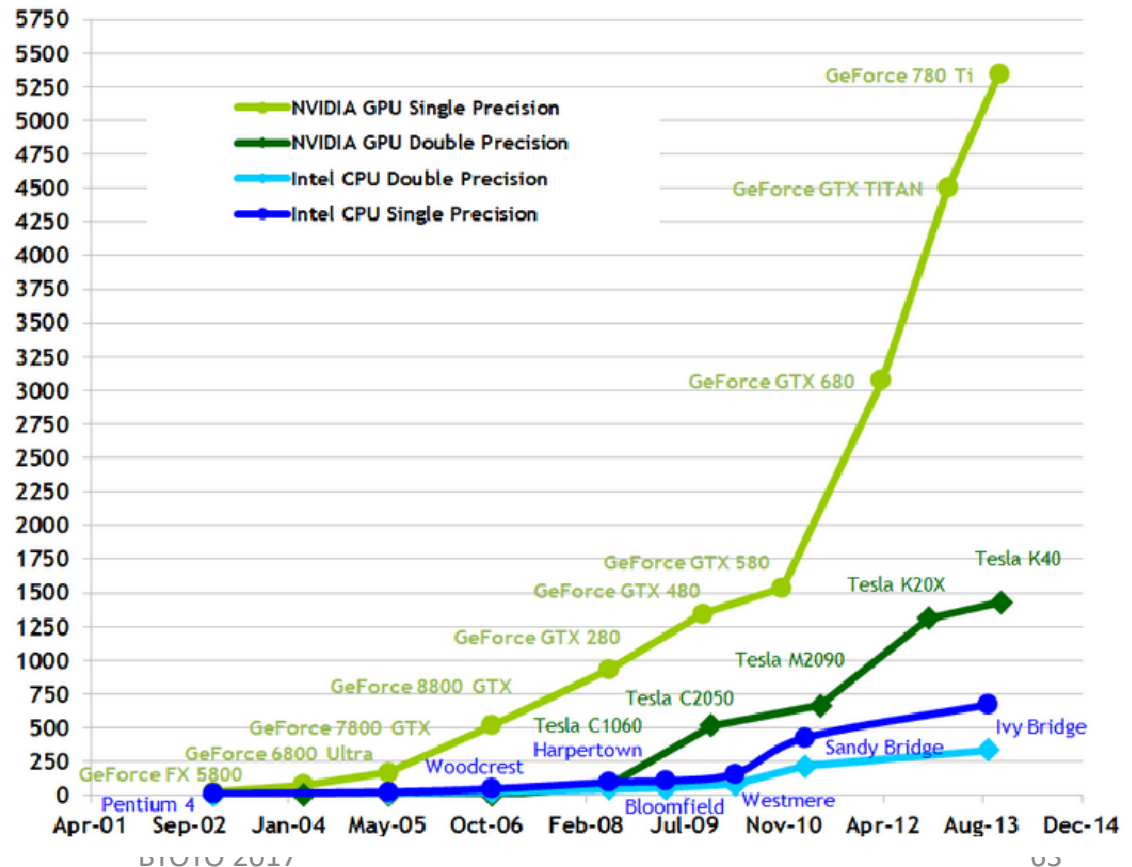
Why does it work now?

We have more computational power now.



Mahmoud Ahmed
Hossam, High
Performance
Hyperspectral Image
Classification using
Graphics Processing
Units, 2015

Theoretical GFLOP/s



Why does it work now?

We have better methods and better understanding of the theory.

Dropout, Restricted
Boltzmann Machines, Adam,
residual networks, memory
networks...

Good and Bad news regarding ML and DL

Good News

- Big players in the game (Google, Microsoft, Facebook, Apple, Amazon, Intel, Nvidia)
 - which have provided libraries and frameworks
 - E.g., Tensorflow, torch, google's colab, decaf/caffe, theano
- More data is available

Bad News

- Big players in the game
- Too much competition
- Too dynamic
- Requires a lot of data
- Requires a lot of tuning
- Requires a lot of computational power