ECE 6504: Deep Learning for Perception

Dhruv Batra
Virginia Tech

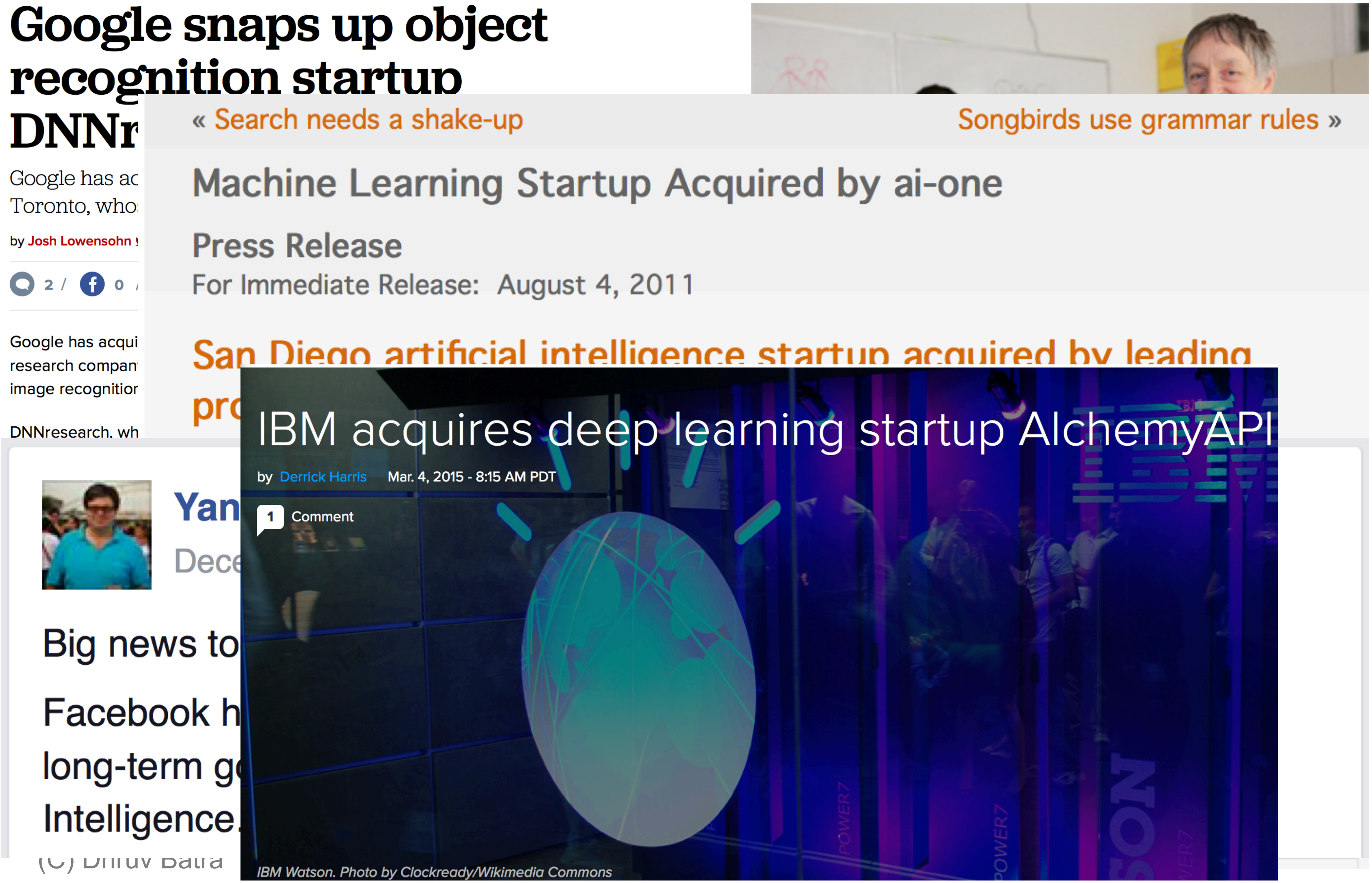
What is this class about?

Some of the most exciting developments in

**Machine Learning,
Vision, NLP, Speech, Robotics
& AI in general**

in the last decade!

Acquisitions



First Caveat

- This is an ADVANCED Machine Learning class
  - This should NOT be your first introduction to ML
  - You will need a formal class; not just self-reading/coursera

  - If you took ECE 4984/5984 @VT, you're in the right place
  - If you took an equivalent class elsewhere, see list of topics taught in ECE 4984/5984 to be sure.

Topics Covered in Intro to ML

- Basics of Statistical Learning
  - Loss function, MLE, MAP, Bayesian estimation, bias-variance tradeoff, overfitting, regularization, cross-validation
- Supervised Learning
  - Nearest Neighbour, Naïve Bayes, Logistic Regression, Support Vector Machines, Kernels, Neural Networks, Decision Trees
  - Ensemble Methods: Bagging, Boosting
- Unsupervised Learning
  - Clustering: k-means, Gaussian mixture models, EM
  - Dimensionality reduction: PCA, SVD, LDA
- Perception
  - Applications to Vision, Natural Language Processing

What is ~~Machine~~ Learning?

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

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)

What is Machine Learning?

ML in a 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 → Y
    - e.g. X: emails, Y: {spam, notspam}

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

Tasks

Supervised Learning

x → Classification → y Discrete

x → Regression → y Continuous

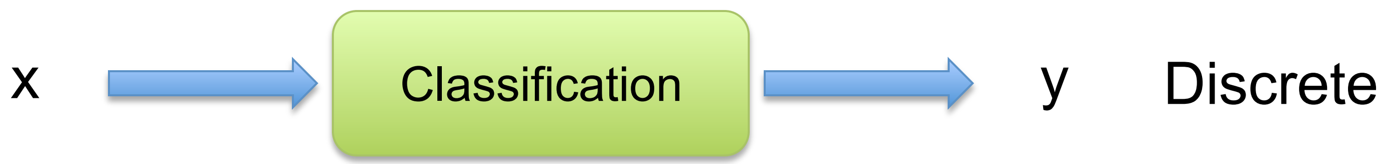
Unsupervised Learning

x → Clustering → y Discrete ID

x → Dimensionality Reduction → y Continuous

Supervised Learning

Classification



Vision: Image Classification

- http://cloudcv.org/classify/

x → y

scuba diver

tiger shark

hammerhead shark

NLP: Machine Translation

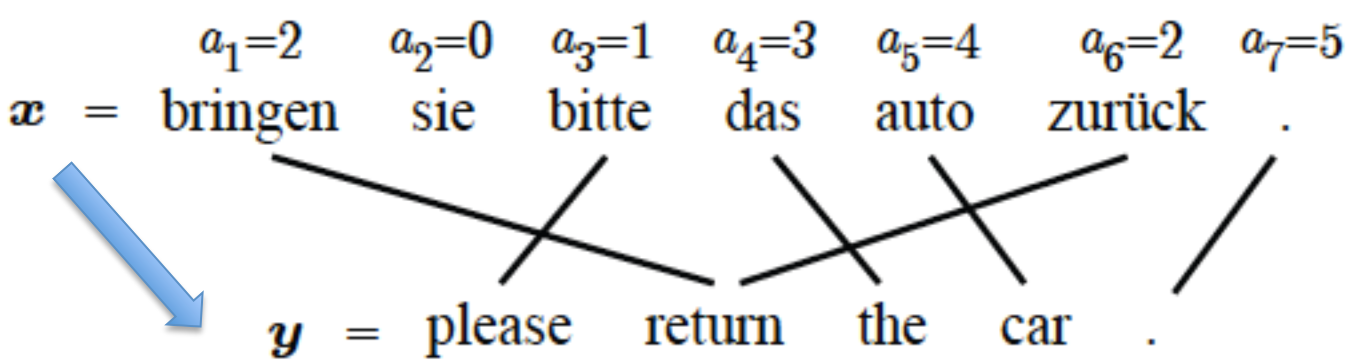
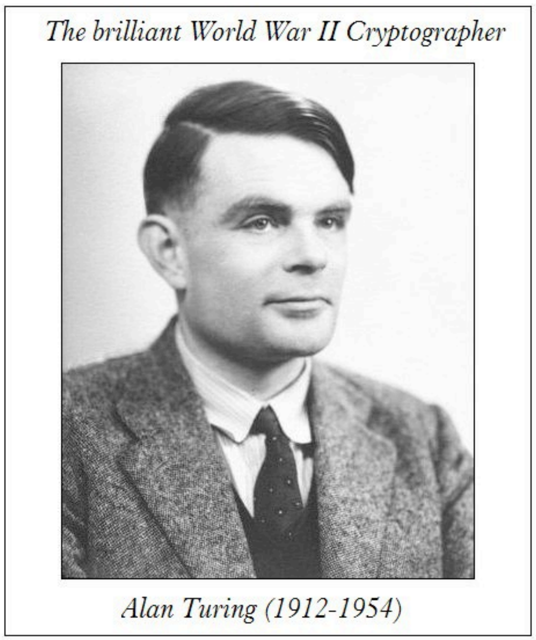


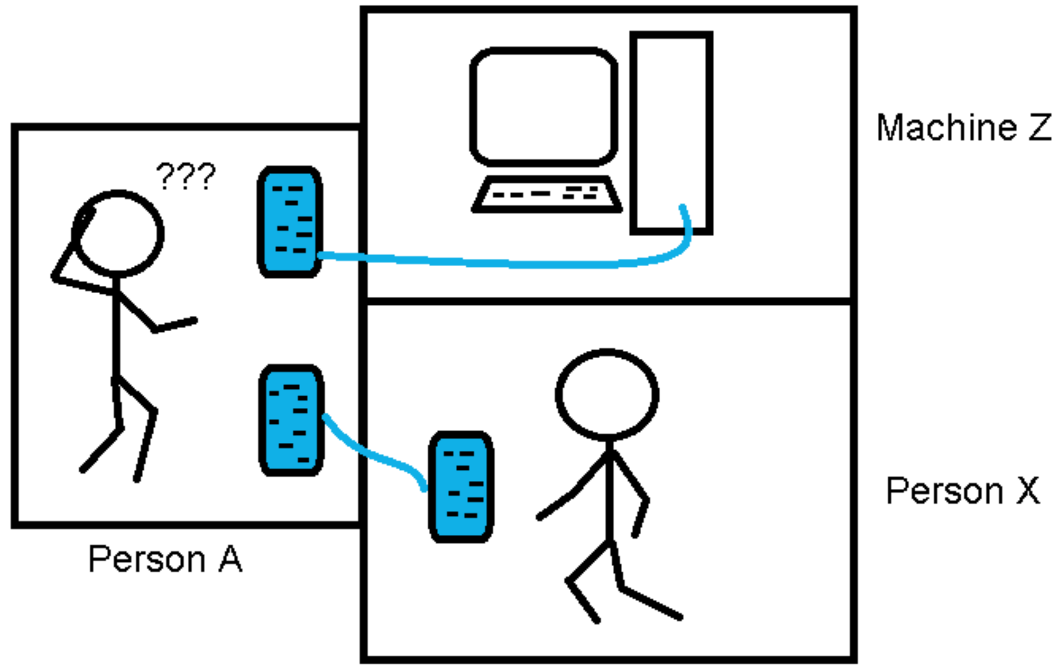
Figure Credit: Kevin Gimpel

Speech: Speech2Text

AI: Turing Test

"Can machines think"





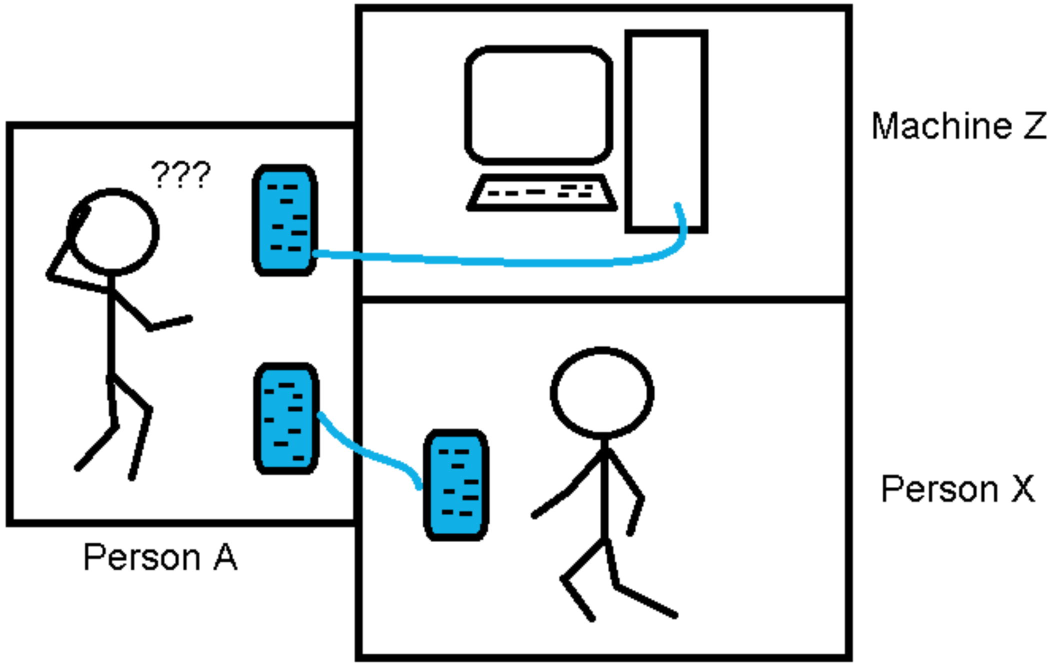
Q: Please write me a sonnet on the subject of the Forth Bridge.
A: Count me out on this one. I never could write poetry.

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

AI: Visual Turing Test

x Q: How many slices of pizza are there?

y A: 6



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.

Synonyms

- Representation Learning

- Deep (Machine) Learning
- Deep Neural Networks

- Deep Unsupervised Learning

- Simply: Deep Learning

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

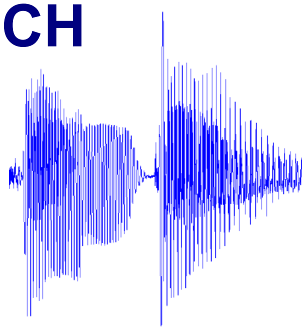
- Distributed Representations
  - No single neuron “encodes” everything
  - Groups of neurons work together

Traditional Machine Learning

VISION

→ **hand-crafted features SIFT/HOG** (fixed) → **your favorite classifier** (learned) → "car"

SPEECH



→ **hand-crafted features MFCC** (fixed) → **your favorite classifier** (learned) → \ˈd ē p\

NLP

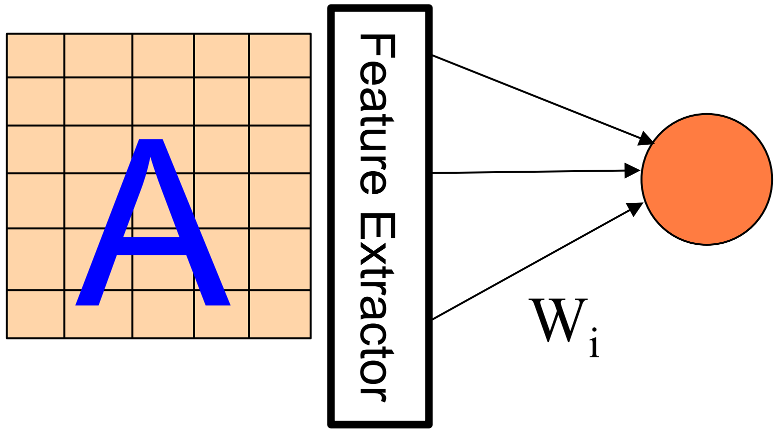
This burrito place is yummy and fun!

→ **hand-crafted features Bag-of-words** (fixed) → **your favorite classifier** (learned) → "+"

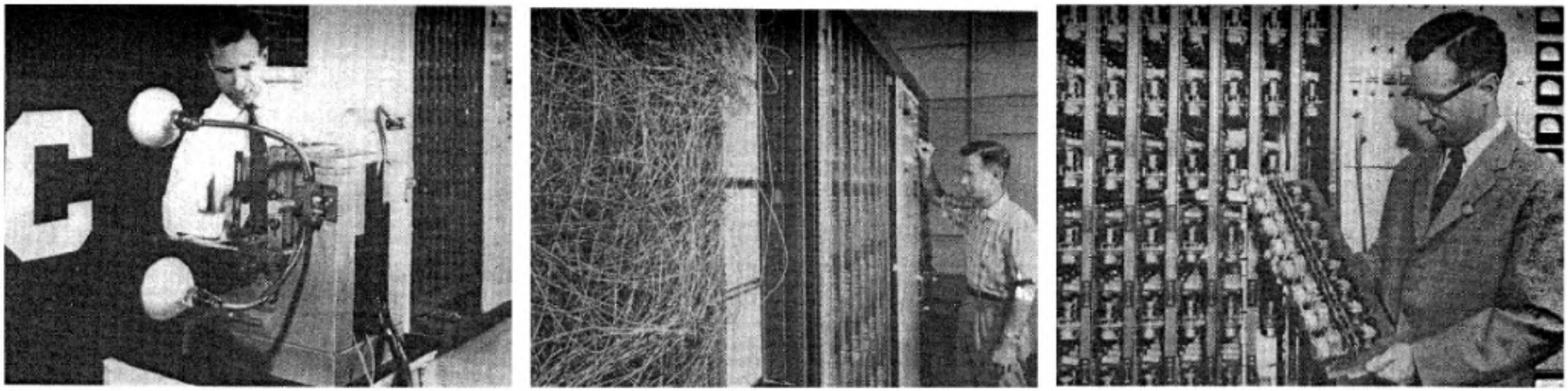
Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

It’s an old paradigm

- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching.
- Designing a feature extractor requires considerable efforts by experts.



$$y = sign\left(\sum_{i=1}^{N} W_i F_i(X) + b\right)$$



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Hierarchical Compositionality

VISION

pixels → edge → texton → motif → part → object

SPEECH

sample → spectral band → formant → motif → phone → word

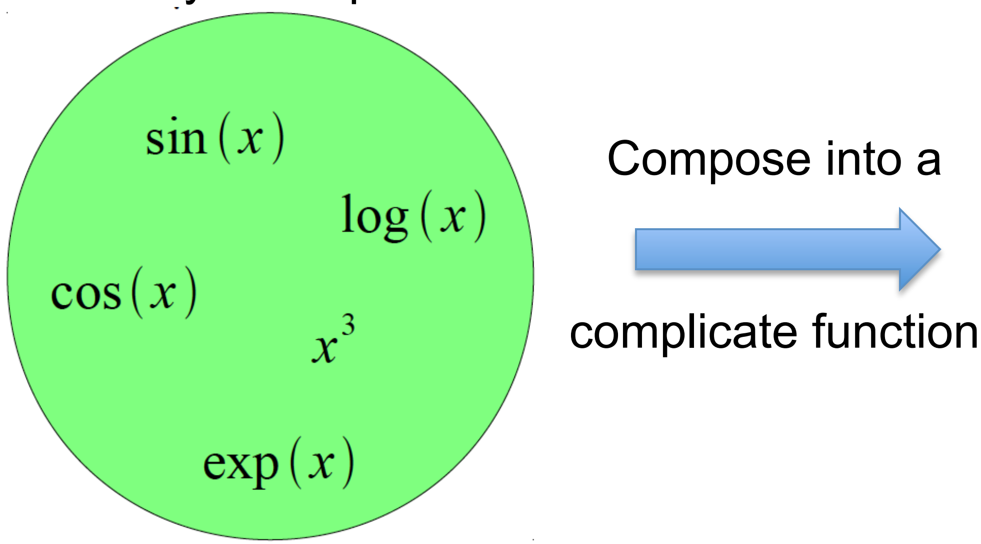
NLP

character → word → NP/VP/.. → clause → sentence → story

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

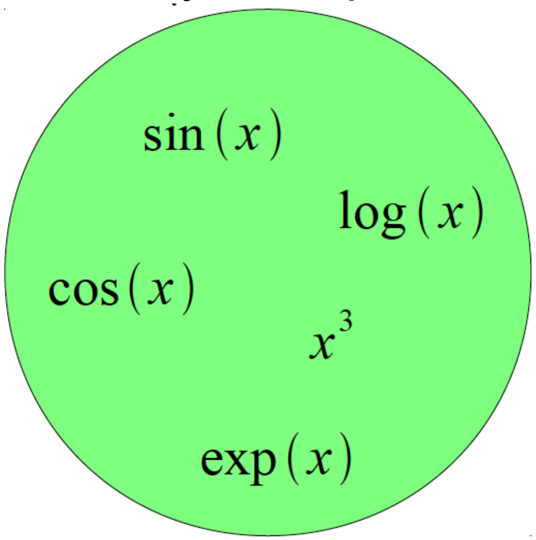
Building A Complicated Function

Given a library of simple functions



Building A Complicated Function

Given a library of simple functions



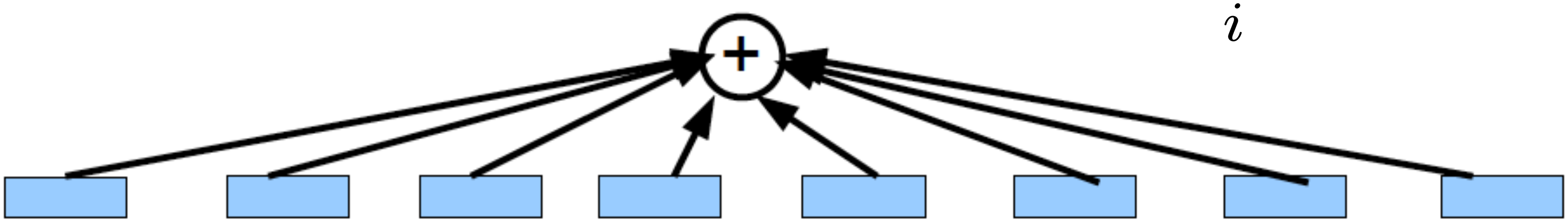
Compose into a

complicate function

Idea 1: Linear Combinations

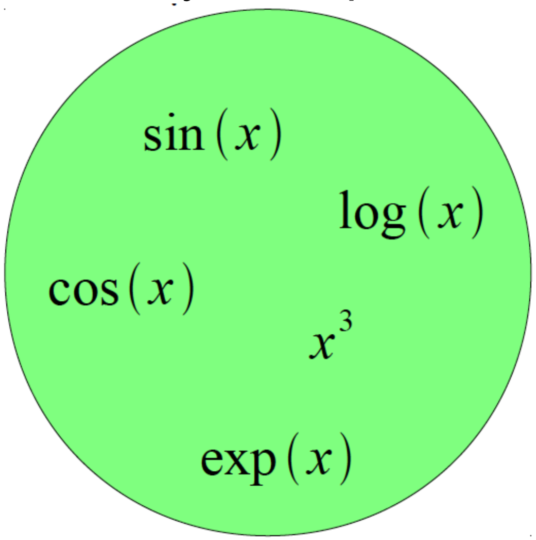
- Boosting
- Kernels
- …

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



Building A Complicated Function

Given a library of simple functions



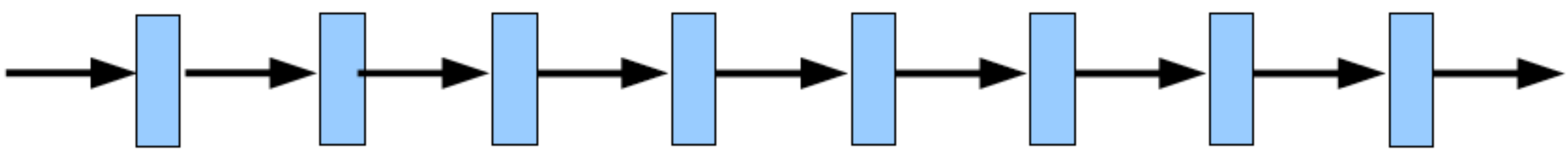
Compose into a

complicate function

Idea 2: Compositions

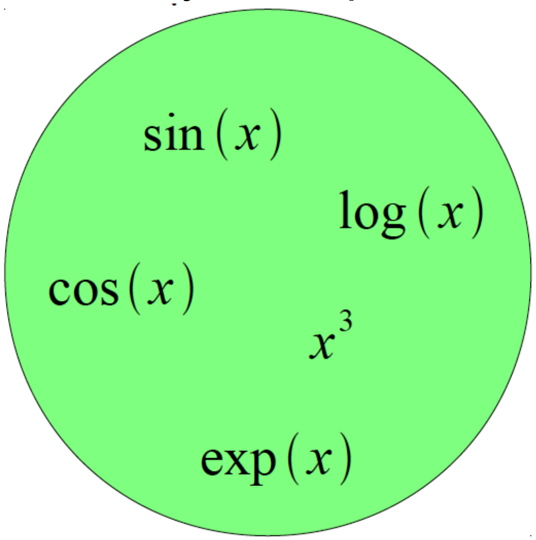
- Deep Learning
- Grammar models
- Scattering transforms…

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



Building A Complicated Function

Given a library of simple functions

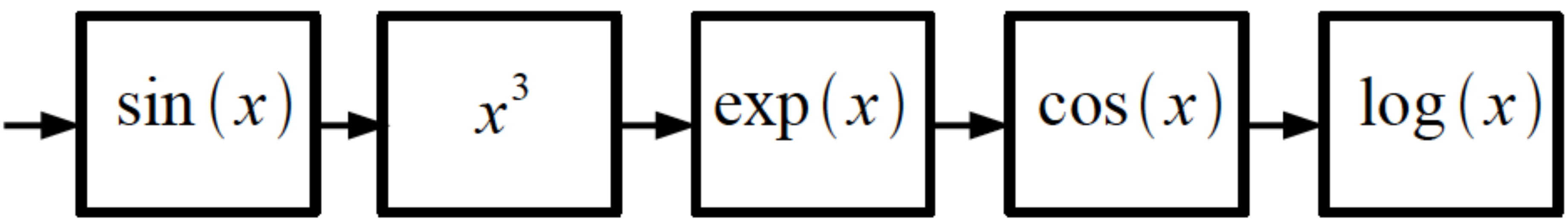


Compose into a complicate function

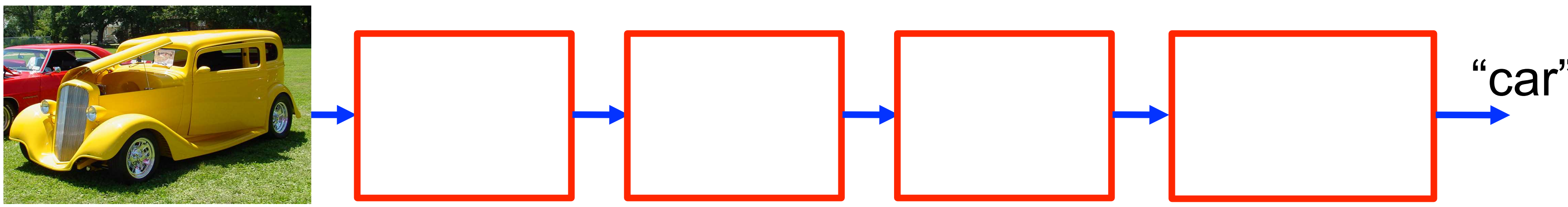
Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms…

$$f(x) = \log(\cos(\exp(\sin^3(x))))$$

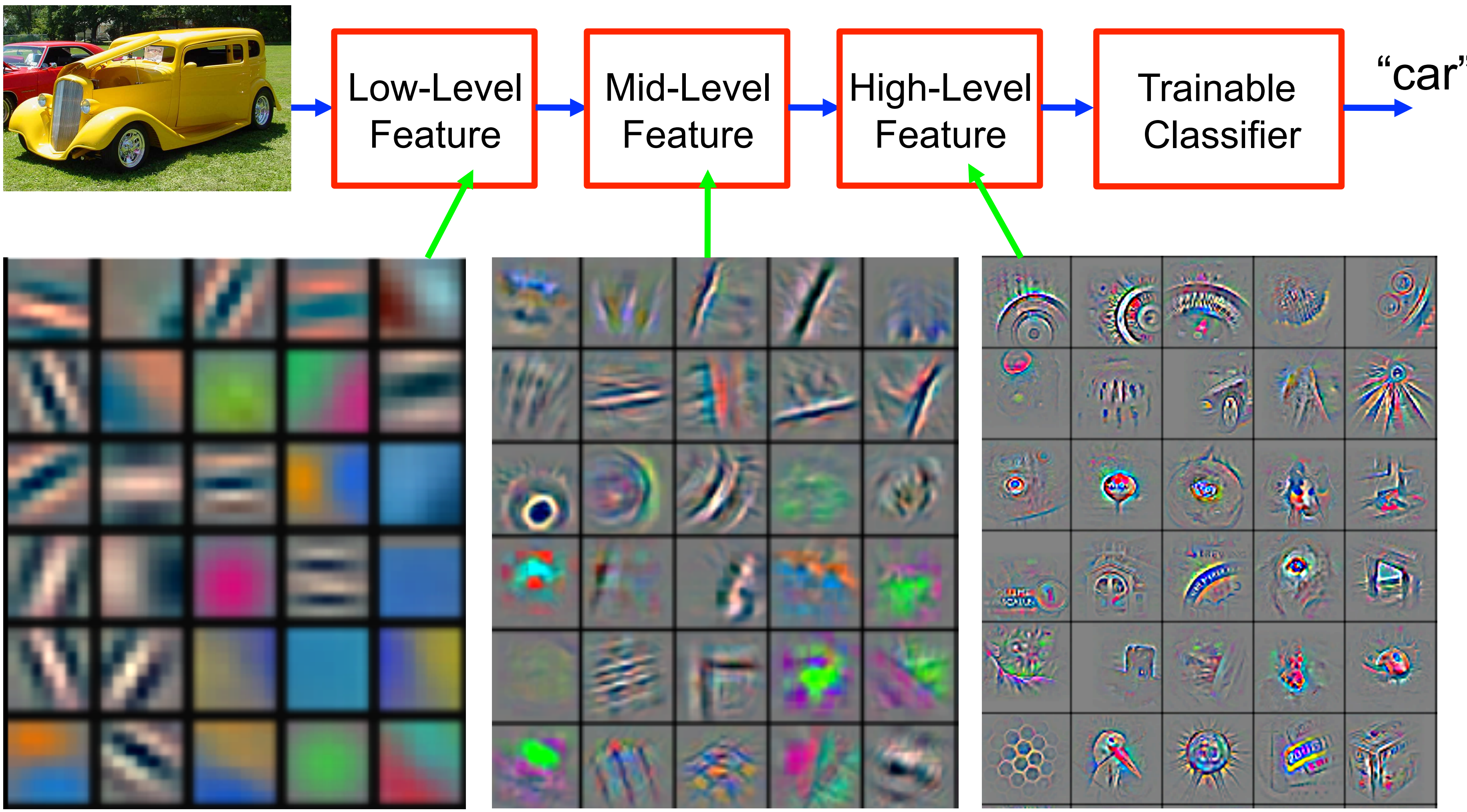


Deep Learning = Hierarchical Compositionality



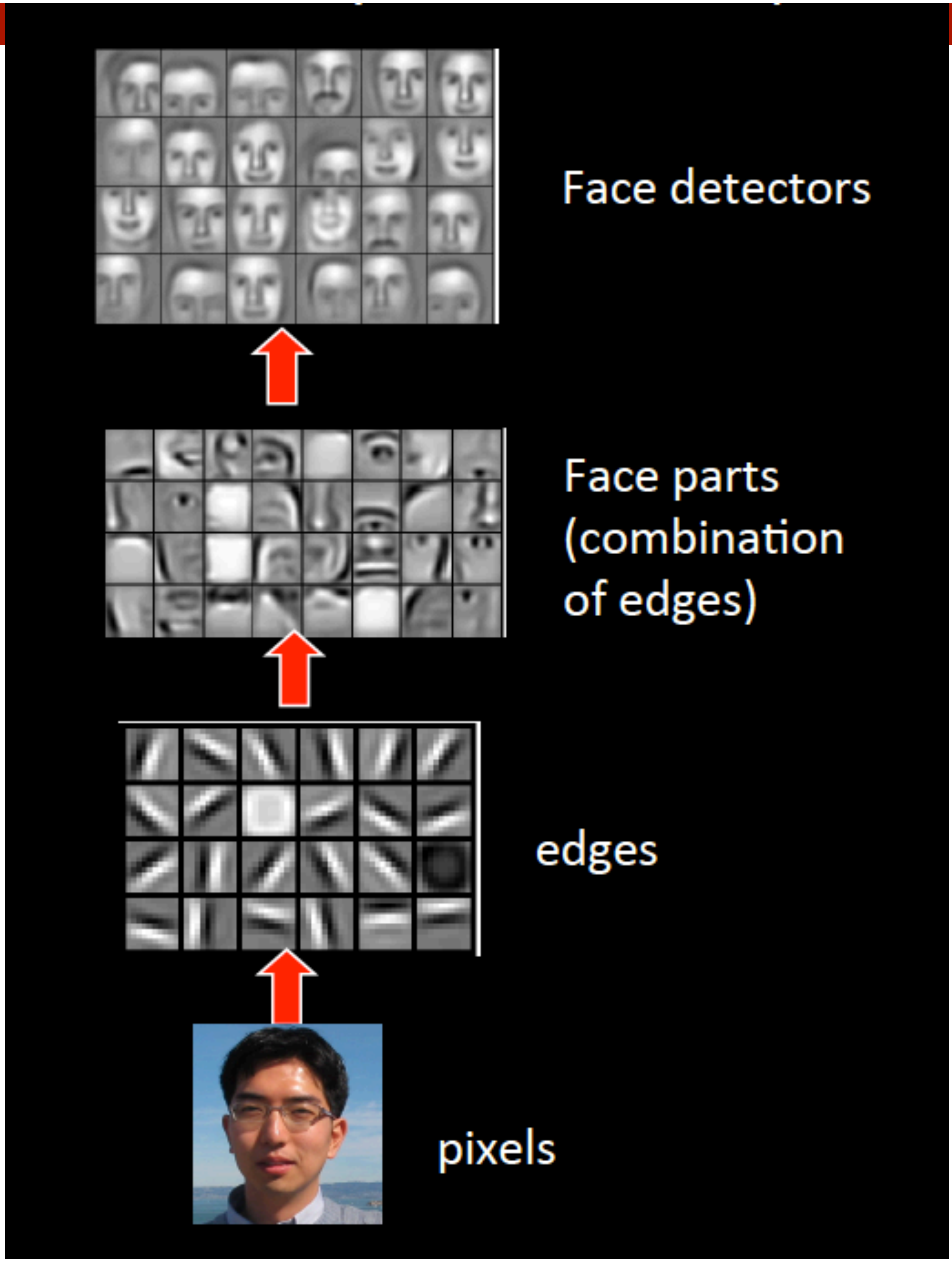
Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Deep Learning = Hierarchical Compositionality



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

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun



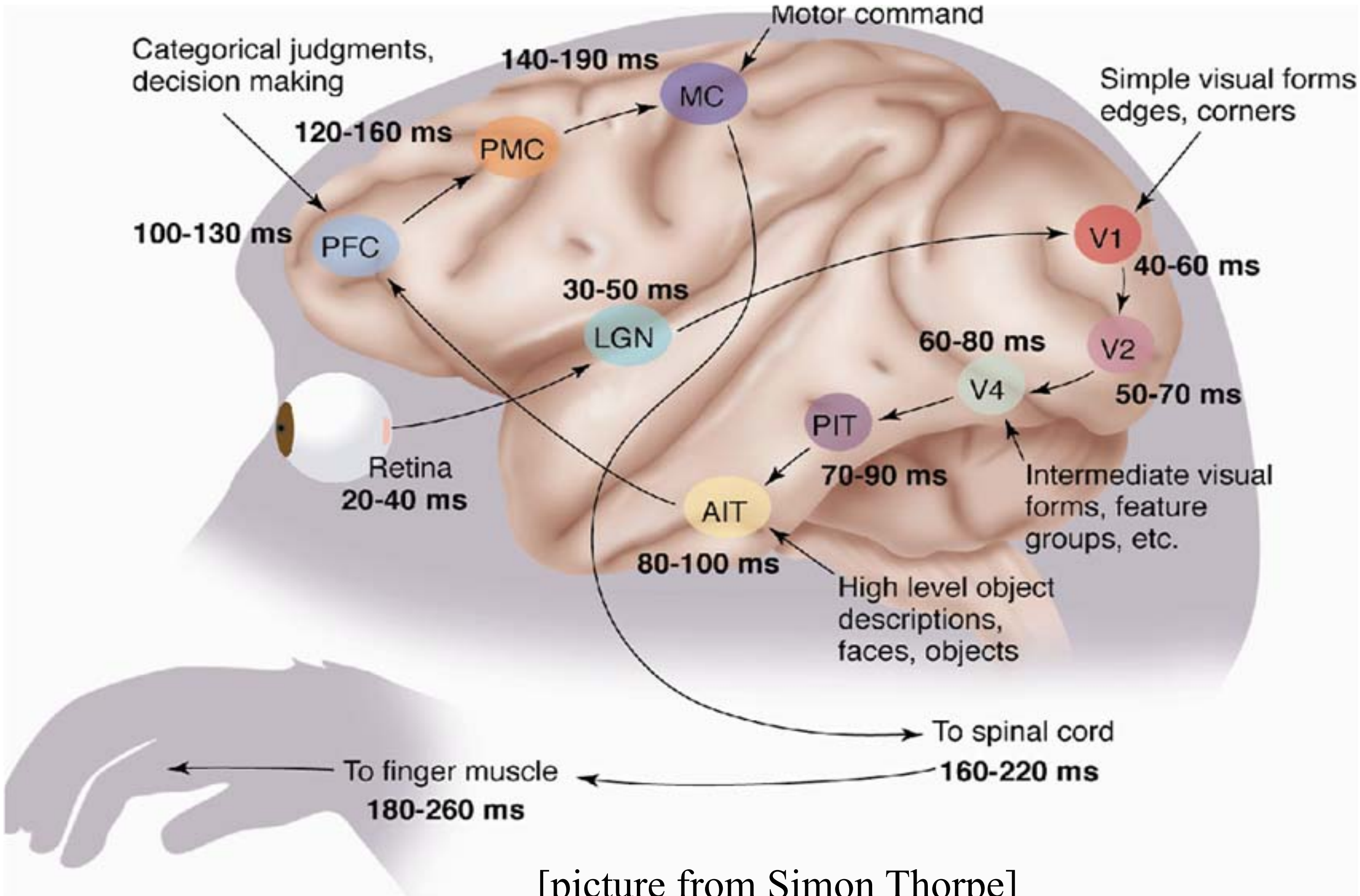
Sparse DBNs

[Lee et al. ICML '09]

Figure courtesy: Quoc Le

The Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex



[picture from Simon Thorpe]

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

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

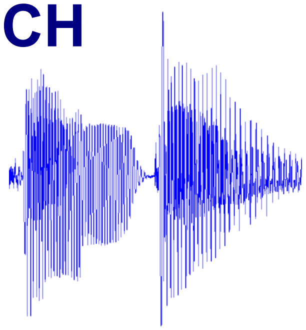
- Distributed Representations
  - No single neuron "encodes" everything
  - Groups of neurons work together

Traditional Machine Learning

VISION

→ **hand-crafted features** **SIFT/HOG** (fixed) → **your favorite classifier** (learned) → “car”

SPEECH



→ **hand-crafted features** **MFCC** (fixed) → **your favorite classifier** (learned) → \ˈd ē p\

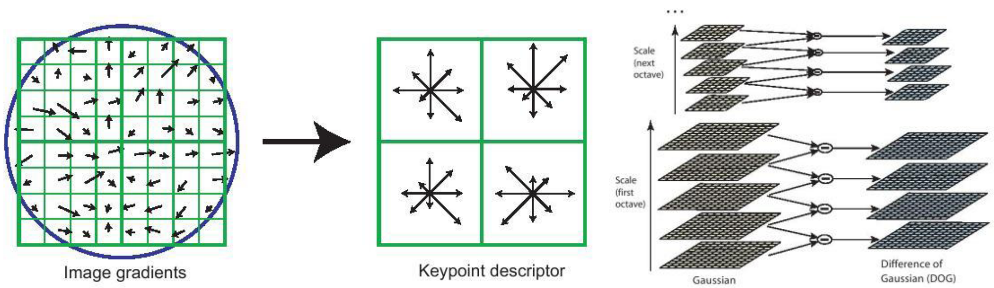
NLP

This burrito place is yummy and fun!

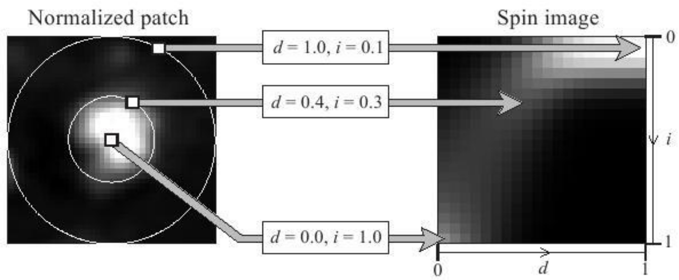
→ **hand-crafted features** **Bag-of-words** (fixed) → **your favorite classifier** (learned) → “+”

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

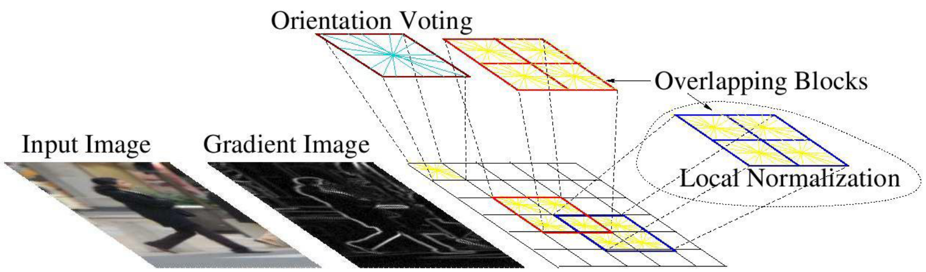
Feature Engineering



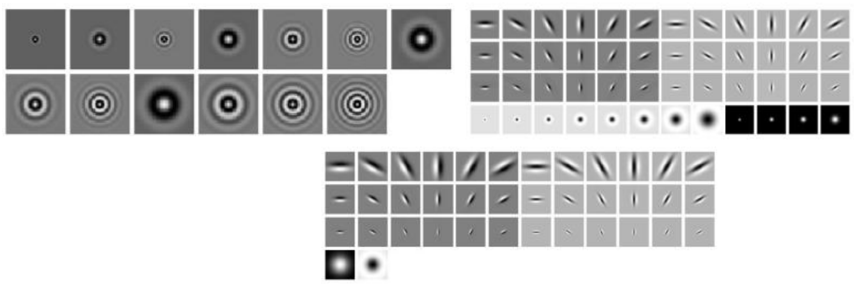
SIFT



Spin Images



HoG



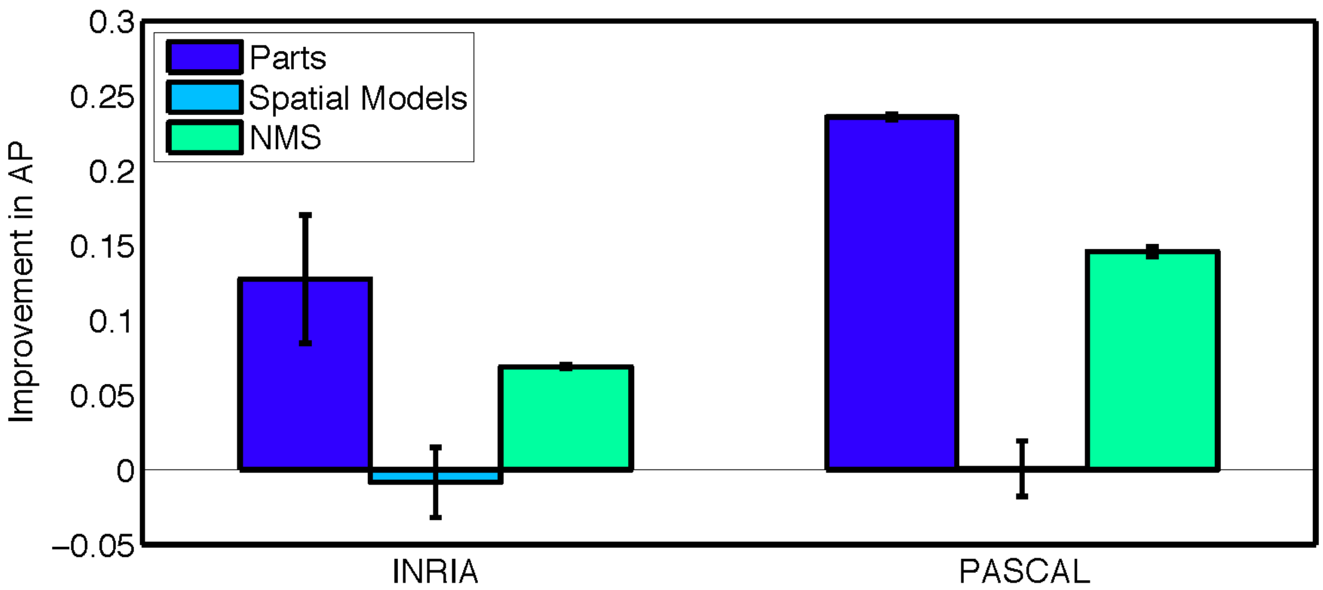
Textons

and many many more….

What are the current bottlenecks?

- Ablation studies on DPM [Parikh & Zitnick, CVPR10]
  - Replace every “part” in the model with a human

- Key takeaway: “parts” or features are the most important!



Seeing is *worse* than believing

- [Barbu et al. ECCV14]

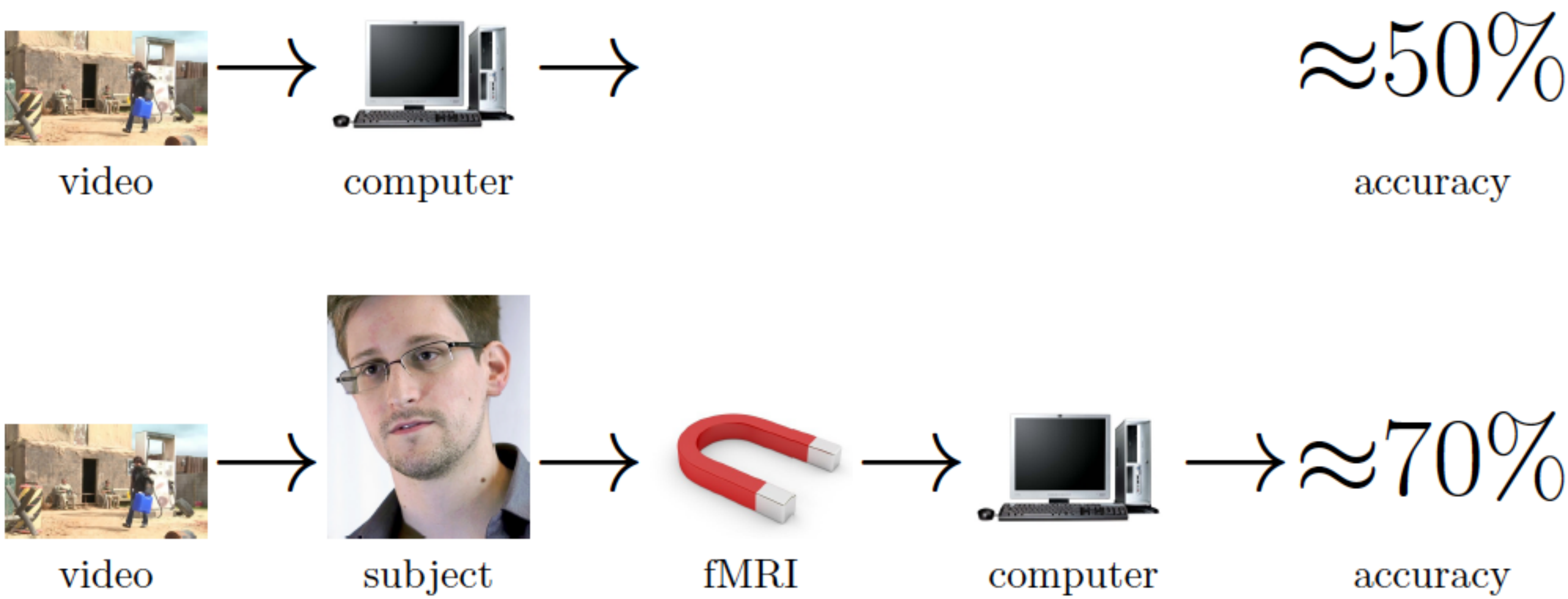
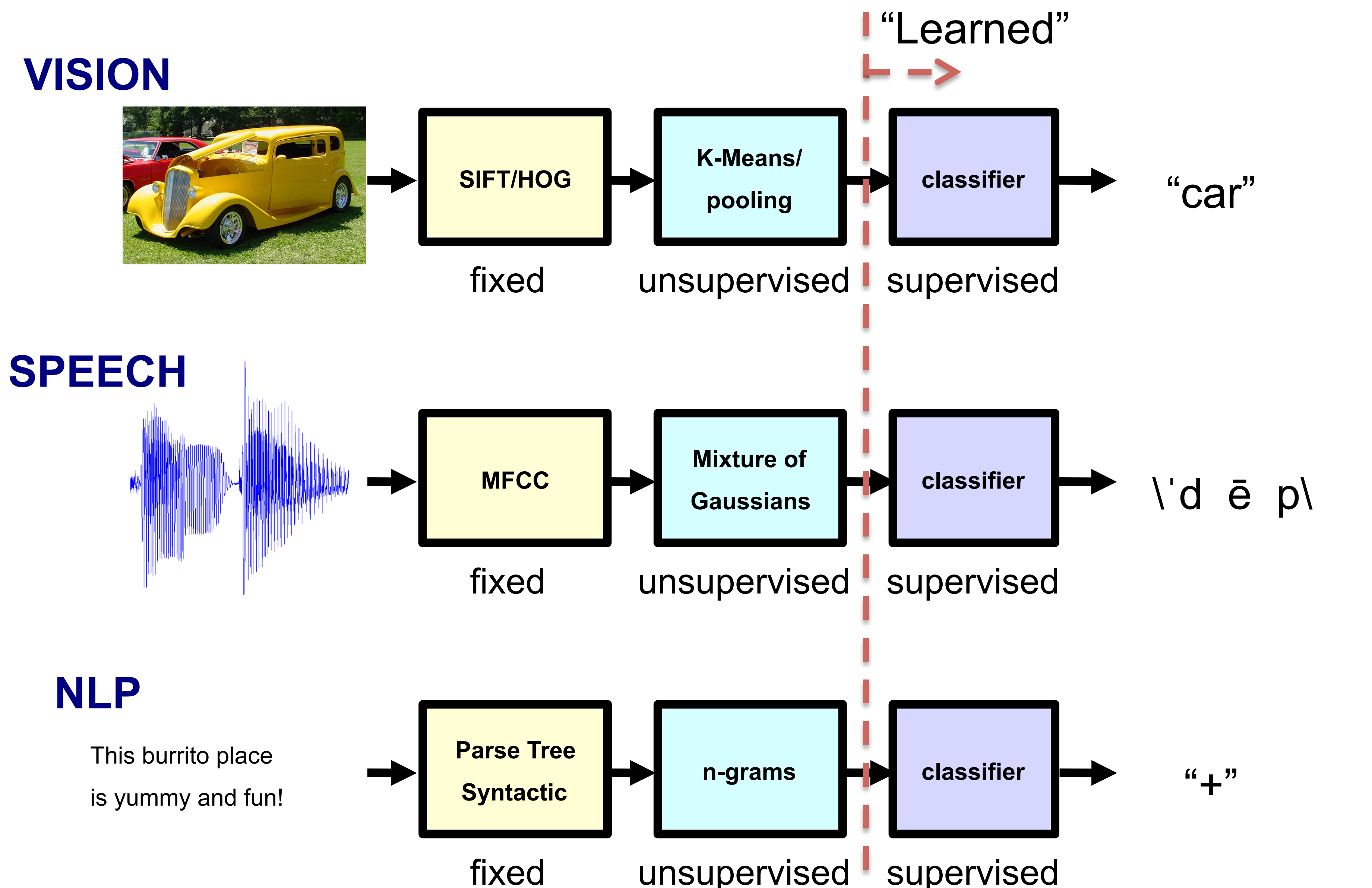
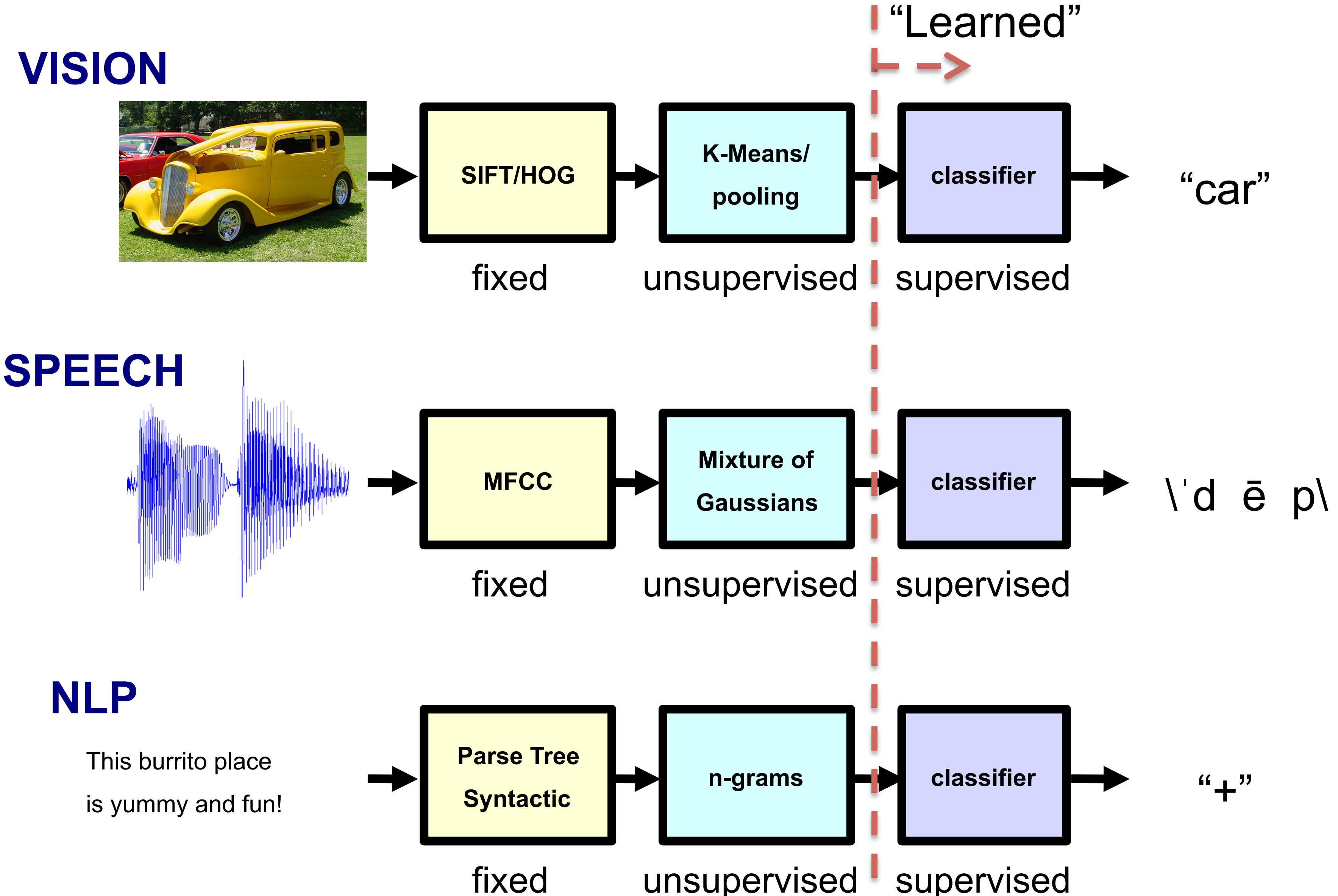


Image Credit: Barbu et al.

Traditional Machine Learning (more accurately)



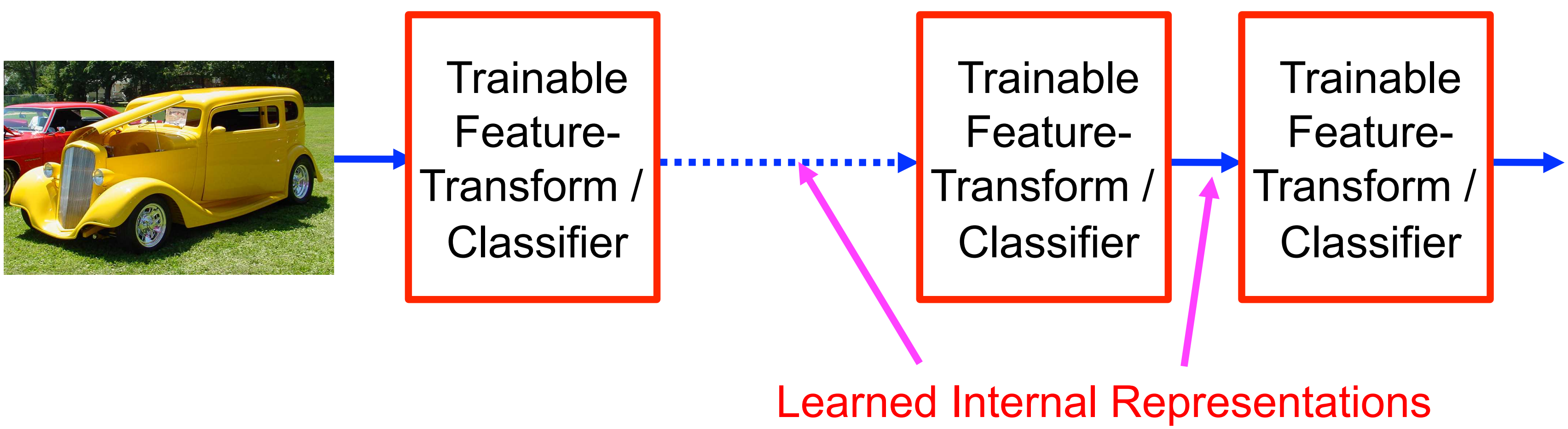
Deep Learning = End-to-End Learning



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Deep Learning = End-to-End Learning

- A hierarchy of trainable feature transforms
  - Each module transforms its input representation into a higher-level one.
  - High-level features are more global and more invariant
  - Low-level features are shared among categories



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

"Shallow" vs Deep Learning

- "Shallow" models

hand-crafted Feature Extractor → "Simple" Trainable Classifier

fixed | learned

- Deep models

Trainable Feature-Transform / Classifier → Trainable Feature-Transform / Classifier → Trainable Feature-Transform / Classifier

Learned Internal Representations

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Do we really need deep models?

So what *is* Deep (Machine) Learning?

- A few different ideas:

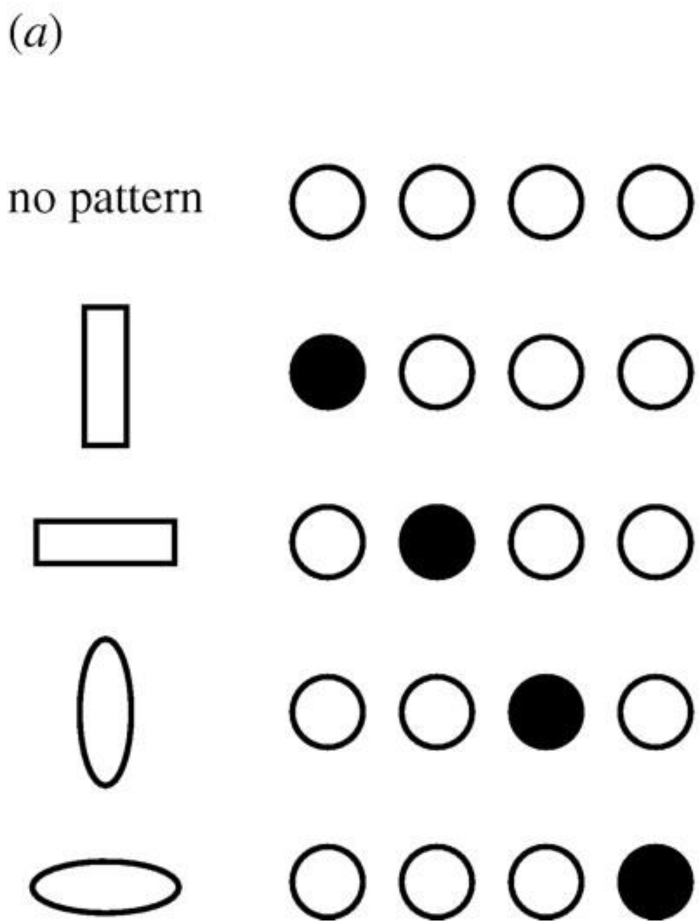
- (Hierarchical) Compositionality
  – Cascade of non-linear transformations
  – Multiple layers of representations

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

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

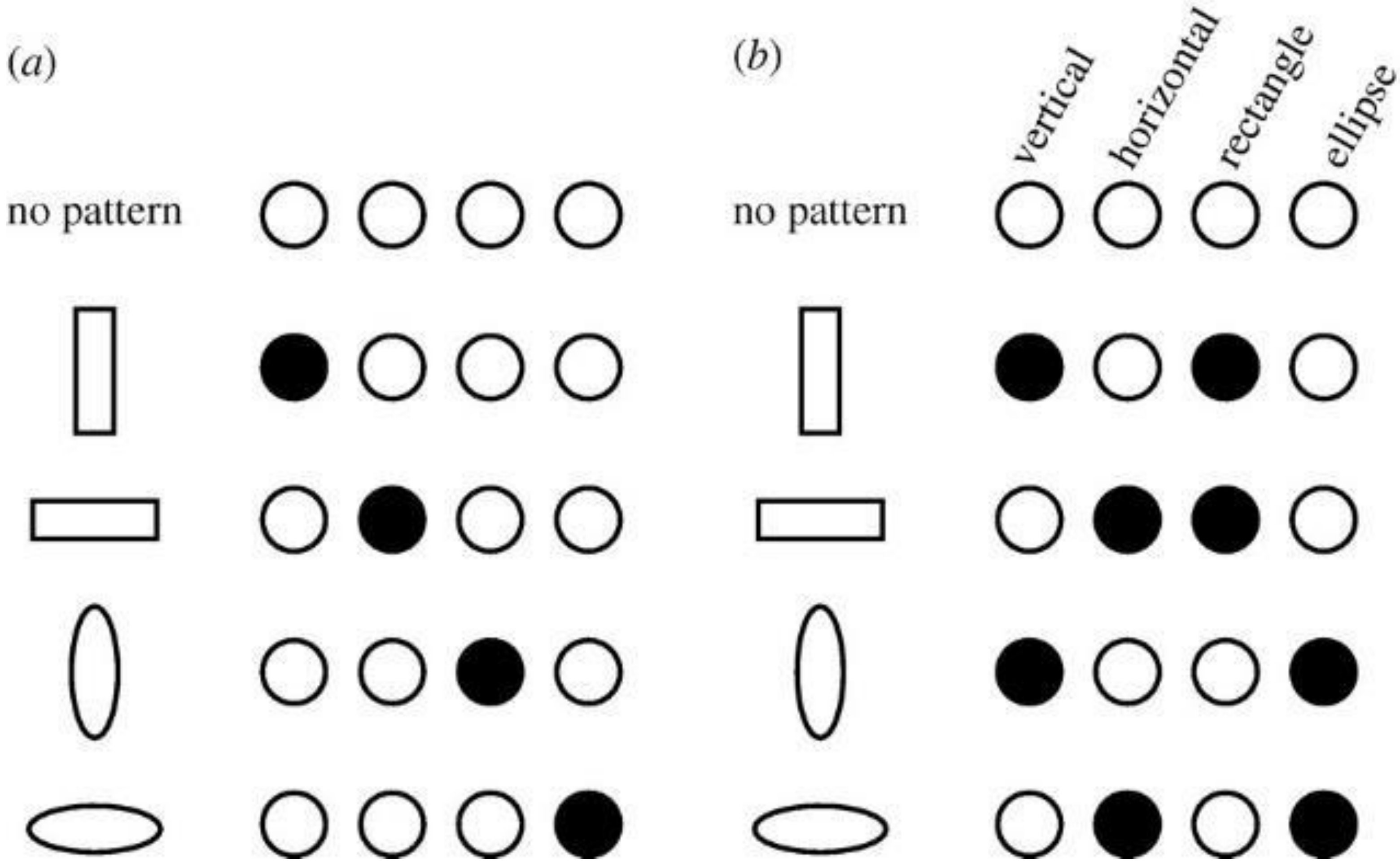
Distributed Representations Toy Example

- Local vs Distributed



Distributed Representations Toy Example

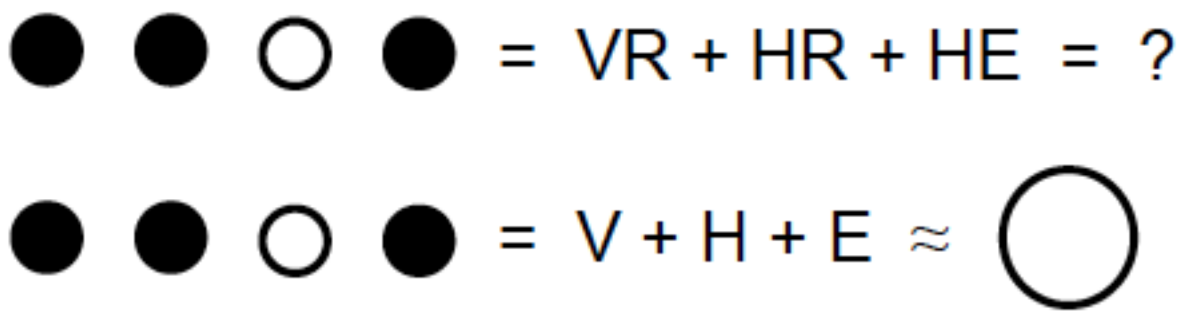
- Can we interpret each dimension?



Power of distributed representations!

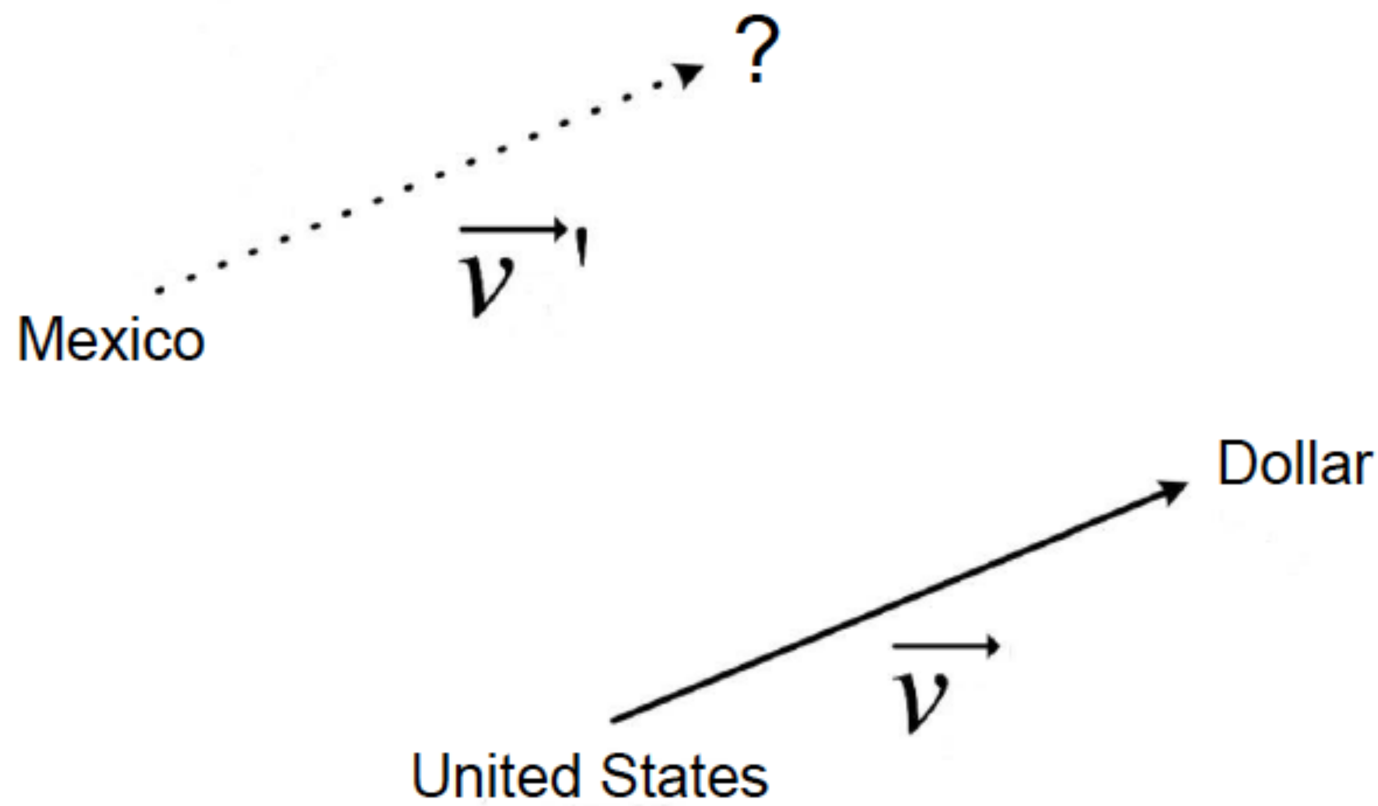
Local

Distributed



Power of distributed representations!

- United States:Dollar :: Mexico:?



Power of distributed representations!

- Example: all face images of a person
  - 1000x1000 pixels = 1,000,000 dimensions
  - But the face has 3 cartesian coordinates and 3 Euler angles
  - And humans have less than about 50 muscles in the face
  - Hence the manifold of face images for a person has <56 dimensions
- The perfect representations of a face image:
  - Its coordinates on the face manifold
  - Its coordinates away from the manifold

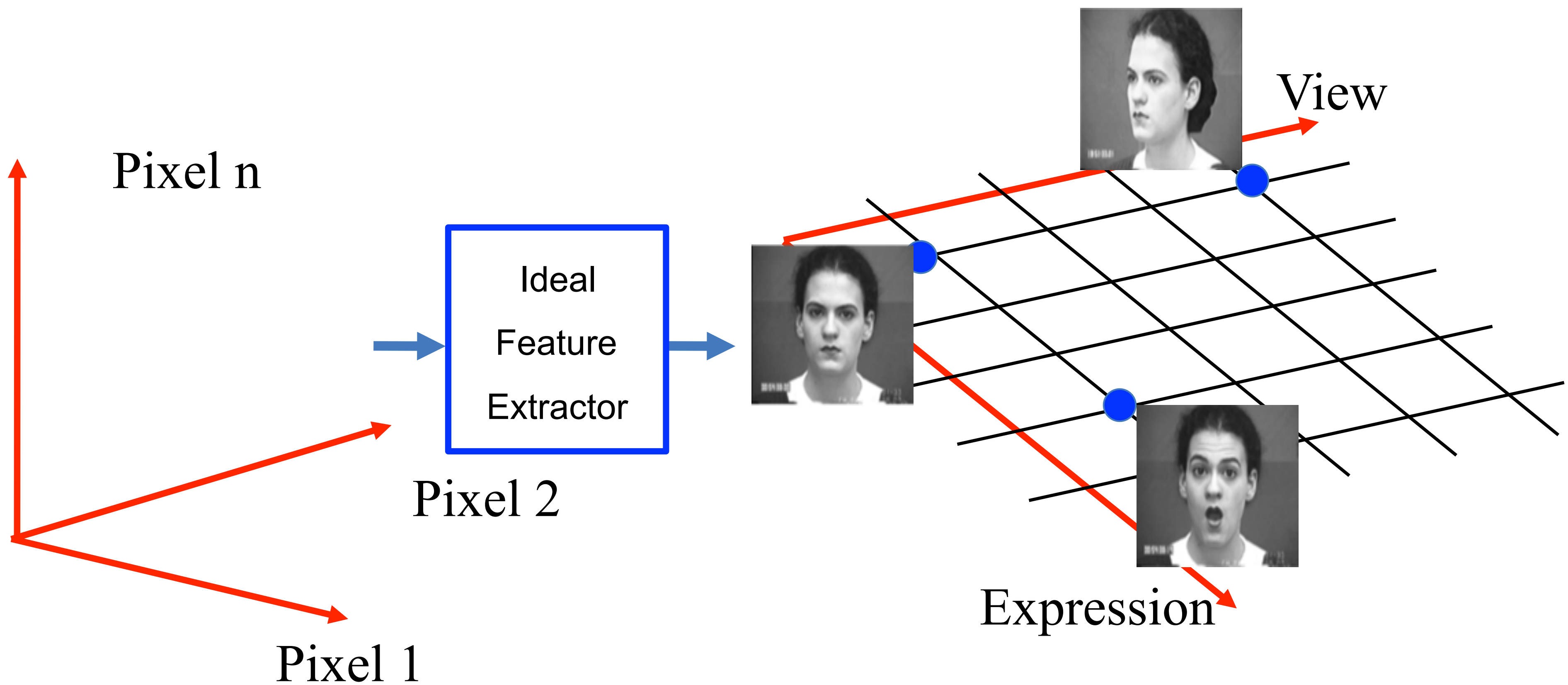
$\longrightarrow$ **Ideal Feature Extractor** $\longrightarrow$

$$\begin{bmatrix} 1.2 \\ -3 \\ 0.2 \\ -2\ldots \end{bmatrix}$$

Face/not face
Pose
Lighting
Expression

Power of distributed representations!

The Ideal Disentangling Feature Extractor



Distributed Representations

- Q: What objects are in the image? Where?

Ideal Feature Extractor

- window, top-left
- clock, top-middle
- shelf, left
- drawing,middle
- statue, bottom left
- …
- hat, bottom right

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Power of distributed representations!

DecorNets: Encouraging Decorrelated Activations in Deep Neural Networks

Michael Cogswell
cogswell@vt.edu

Faruk Ahmed
faruk@vt.edu

Ross Girshick
rbg@microsoft.com

Larry Zitnick
larryz@microsoft.com

Dhruv Batra
dbatra@vt.edu

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

Benefits of Deep/Representation Learning

- (Usually) Better Performance
  - *"Because gradient descent is better than you"*
    Yann LeCun

- New domains without "experts"
  - RGBD
  - Multi-spectral data
  - Gene-expression data
  - Unclear how to hand-engineer

"Expert" intuitions can be misleading

- *"Every time I fire a linguist, the performance of our speech recognition system goes up"*
  – Fred Jelinik, IBM '98

- *"Maybe the molecule didn't go to graduate school"*
  – Will Welch defending the success of his approximate molecular screening algorithm, given that he's a computer scientist, not a chemist

Database Screening for HIV Protease Ligands: The Influence of Binding-Site Conformation and Representation on Ligand Selectivity", Volker Schnecke, Leslie A. Kuhn, Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, Pages 242-251, AAAI Press, 1999.

Problems with Deep Learning

- Problem#1: Non-Convex! Non-Convex! Non-Convex!
  - Depth>=3: most losses non-convex in parameters
  - Theoretically, all bets are off
  - Leads to stochasticity
    - different initializations → different local minima

- Standard response #1
  - “Yes, but all interesting learning problems are non-convex”
  - For example, human learning
    - Order matters → wave hands → non-convexity

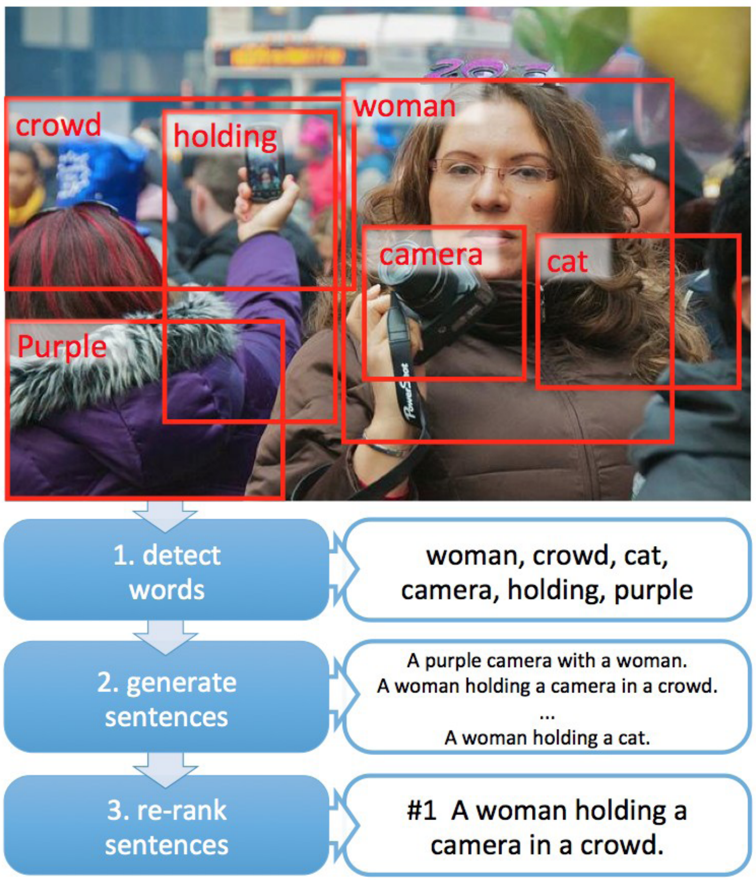
- Standard response #2
  - “Yes, but it often works!”

Problems with Deep Learning

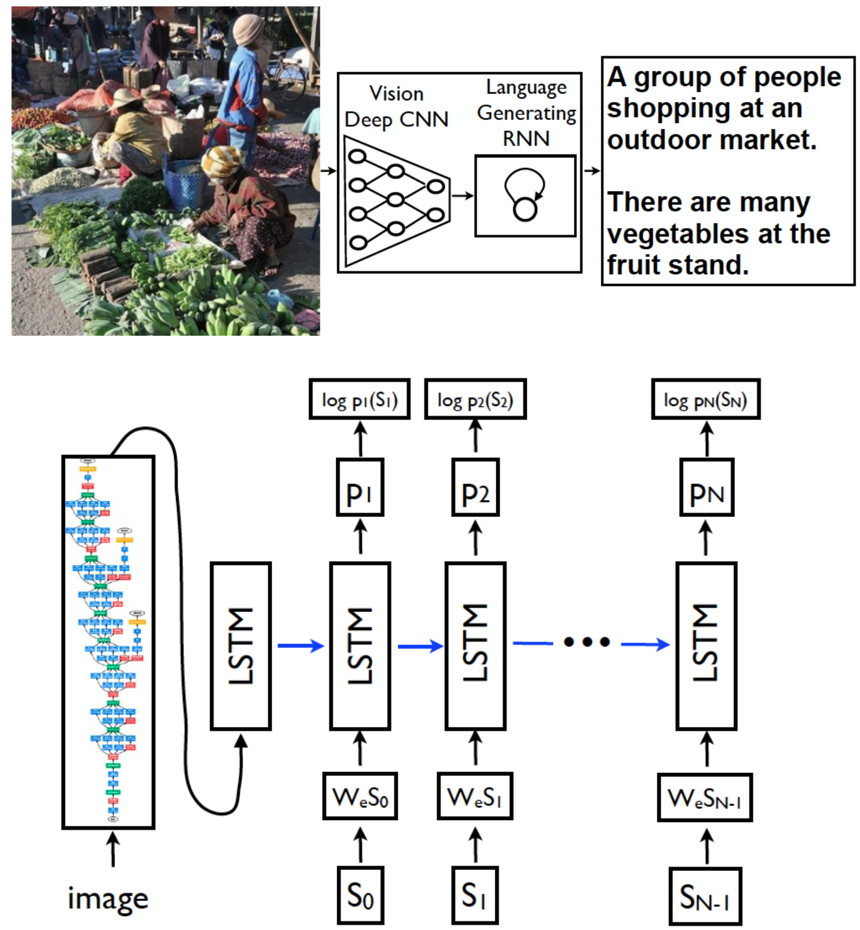
- Problem#2: Hard to track down what's failing
  - Pipeline systems have "oracle" performances at each step
  - In end-to-end systems, it's hard to know why things are not working

Problems with Deep Learning

- Problem#2: Hard to track down what's failing



[Fang et al. CVPR15]



[Vinyals et al. CVPR15]

Pipeline — End-to-End

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!"

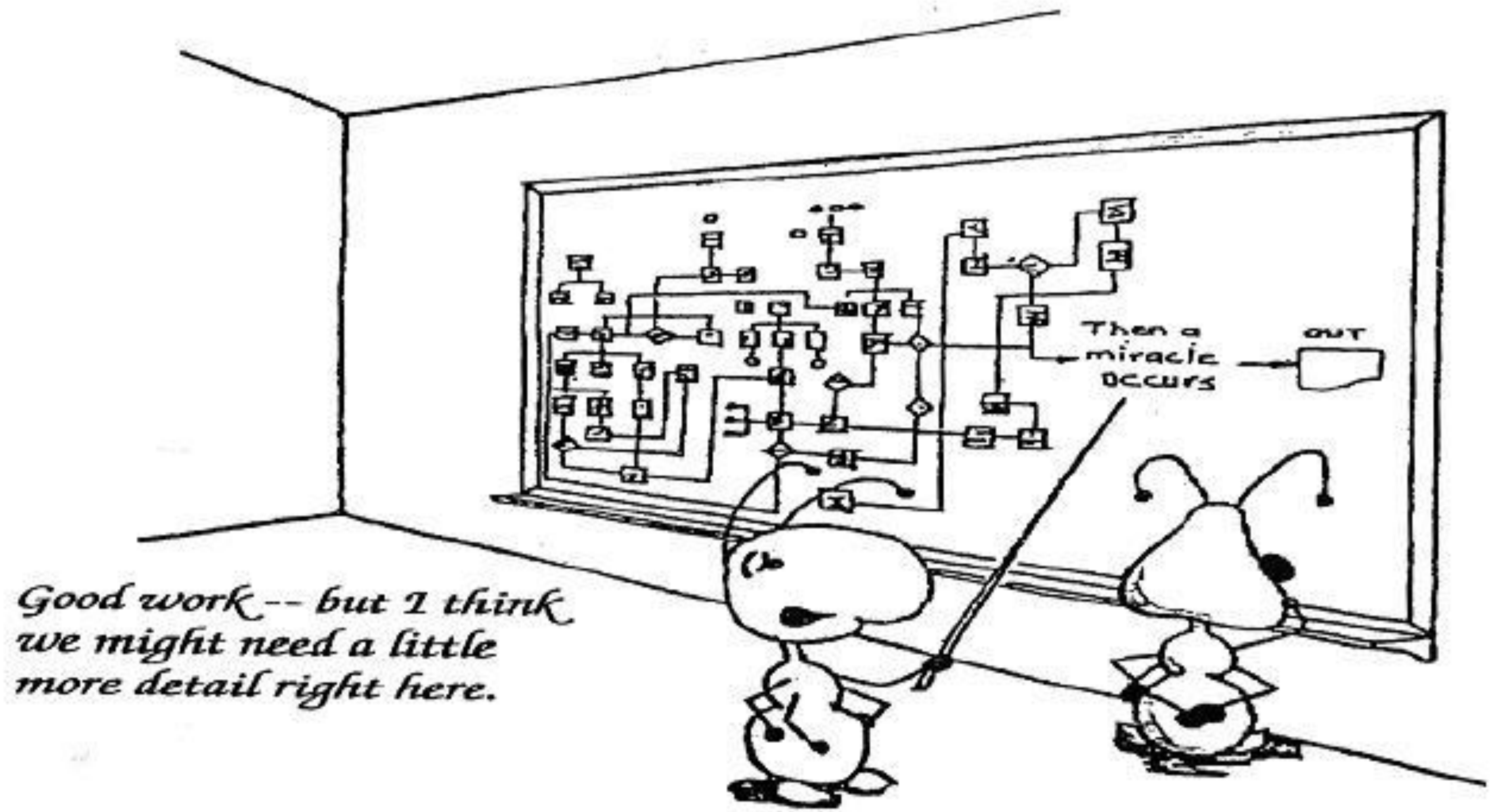
Problems with Deep Learning

- 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!"

Yes it works, but how?



1958 New York Times…

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

The New York Times

Science

COMPUTER SCIENTISTS STYMIED IN THEIR QUEST TO MATCH HUMAN VISION

By WILLIAM J. BROAD

Published: September 25, 1984

EXPERTS pursuing one of man's most audacious dreams - to create machines that think - have stumbled while taking what seemed to be an elementary first step. They have failed to master vision.

After two decades of research, they have yet to teach machines the seemingly simple act of being able to recognize everyday objects and to distinguish one from another.

Instead, they have developed a profound new respect for the sophistication of human sight and have scoured such fields as mathematics, physics, biology and psychology for clues to help them achieve the goal of machine vision.

SCIENCE

Researchers Announce Advance in Image-Recognition Software

By **JOHN MARKOFF** NOV. 17, 2014

Email

Share

Tweet

Save

More

MOUNTAIN VIEW, Calif. — Two groups of scientists, working independently, have created artificial intelligence software capable of recognizing and describing the content of photographs and videos with far greater accuracy than ever before, sometimes even mimicking human levels of understanding.

Until now, so-called computer vision has largely been limited to recognizing individual objects. The new software, described on Monday by researchers at Google and at Stanford University, teaches itself to identify entire scenes: a group of young men playing Frisbee, for example, or a herd of elephants marching on a grassy plain.

The software then writes a caption in English describing the picture. Compared with human observations, the researchers found, the computer-written descriptions are surprisingly accurate.

Captioned by Human and by Google's Experimental Program

Human: "A group of men playing Frisbee in the park."
Computer model: "A group of young people playing a game of Frisbee."

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

INTERESTING.JPG @INTERESTING_JPG · 10h

a man holding a mirror up to his face .

1

View more photos and videos

(

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

TWEETS 587 FOLLOWING 18 FOLLOWERS 746 FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · 18h

a man carrying a bucket of his hands in a yard .

2

View more photos and videos

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

TWEETS 587 FOLLOWING 18 FOLLOWERS 746 FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · Feb 20

a surfboard attached to the top of a car .

8 8

View more photos and videos

(C

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

TWEETS 587 | FOLLOWING 18 | FOLLOWERS 746 | FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · Feb 19

a man dressed in uniform is looking at his cell phone .

2

View more photos and videos

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

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

INTERESTING.JPG @INTERESTING_JPG · 16h

this appears to be a small bedroom in the snow .

6

View more photos and videos

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

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Classification:

1000 object classes **1.4M/50k/100k images**

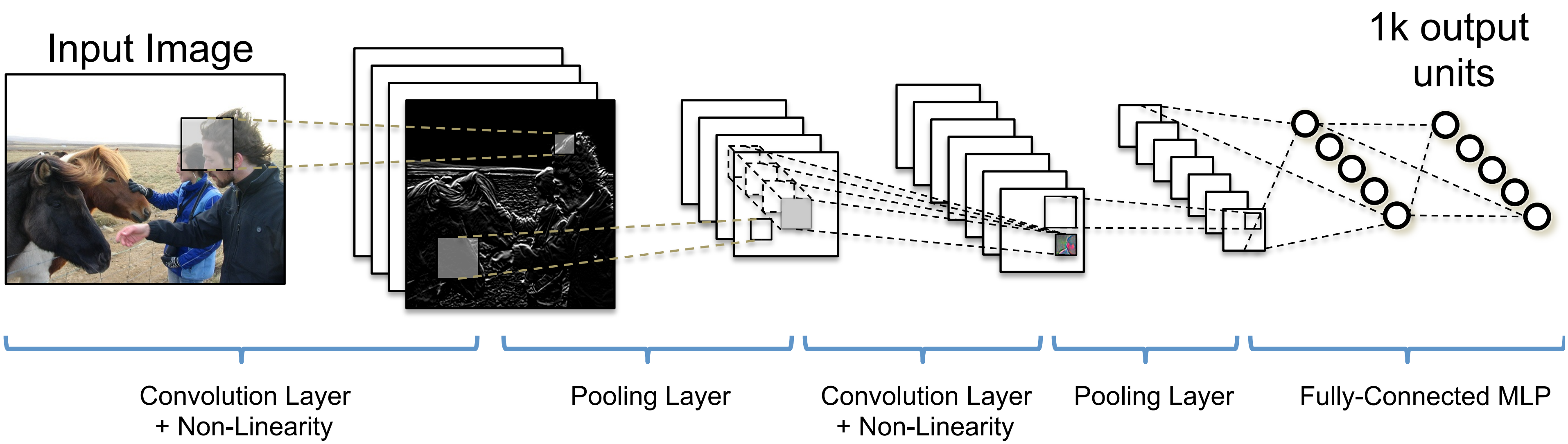
Detection:

200 object classes **400k/20k/40k images**

http://image-net.org/challenges/LSVRC/{2010,…,2014}

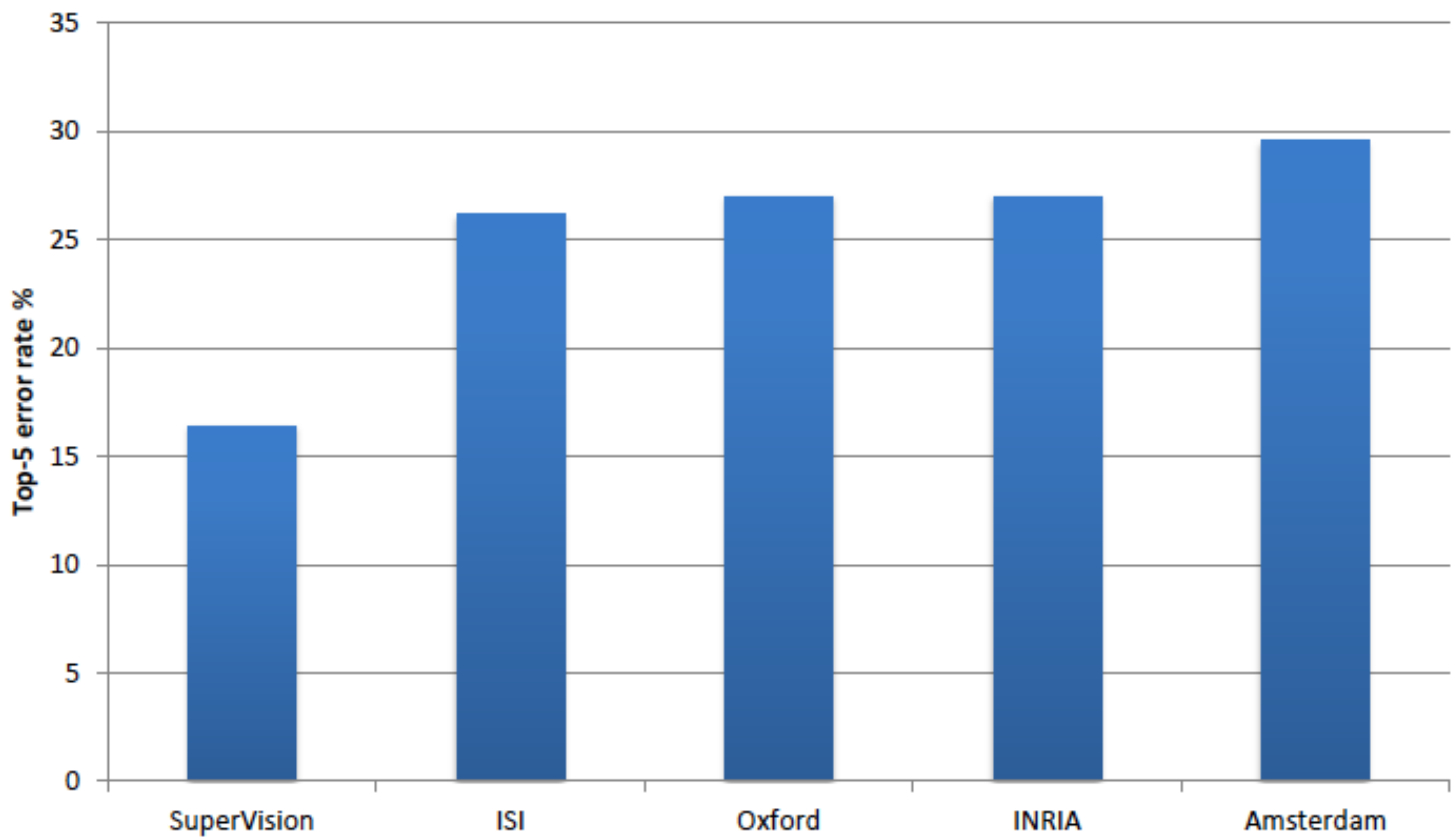
Data Enabling Richer Models

- [Krizhevsky et al. NIPS12]
  - 54 million parameters; 8 layers (5 conv, 3 fully-connected)
  - Trained on 1.4M images in ImageNet
  - Better Regularization (Dropout)



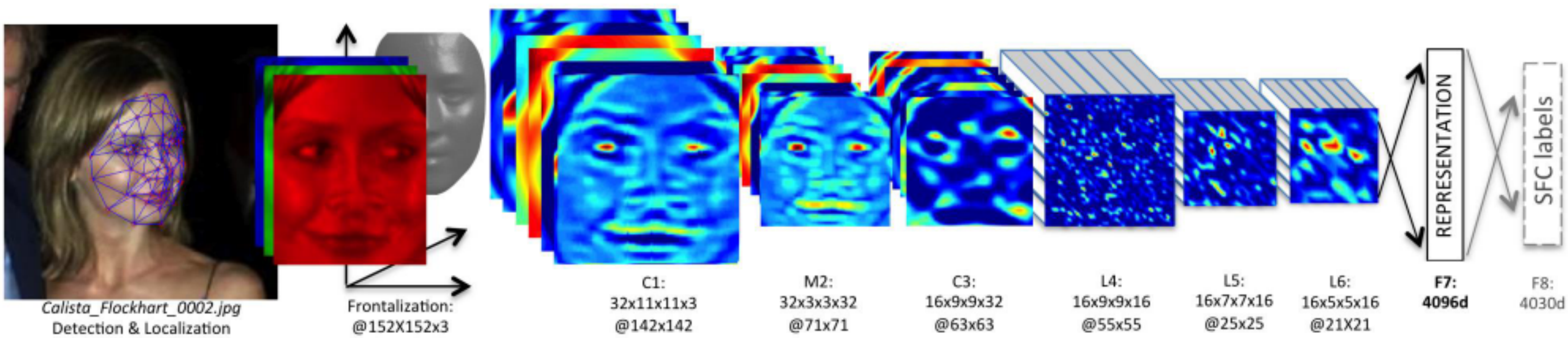
ImageNet Classification 2012

- [Krizhevsky et al. NIPS12]: 16.4% error
- Next best team: 26.2% error



Other Domains & Applications

- Vision
- Natural Language Processing
- Speech
- Robotics
- Game playing



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

Why are things working today?

- More compute power
  - GPUs are ~50x faster

- More data
  - 10^8 samples (compared to 10^3 in 1990s)

- Better algorithms/models/regularizers
  - Dropout
  - ReLu
  - Batch-Normalization
  - …

THE SPACE OF

MACHINE LEARNING METHODS

Recurrent
Neural Net

Boosting

Convolutional
Neural Net

Neural Net

Perceptron

SVM

Deep (sparse/
denoising) Autoencoder

Autoencoder
Neural Net

Sparse Coding

ΣΠ

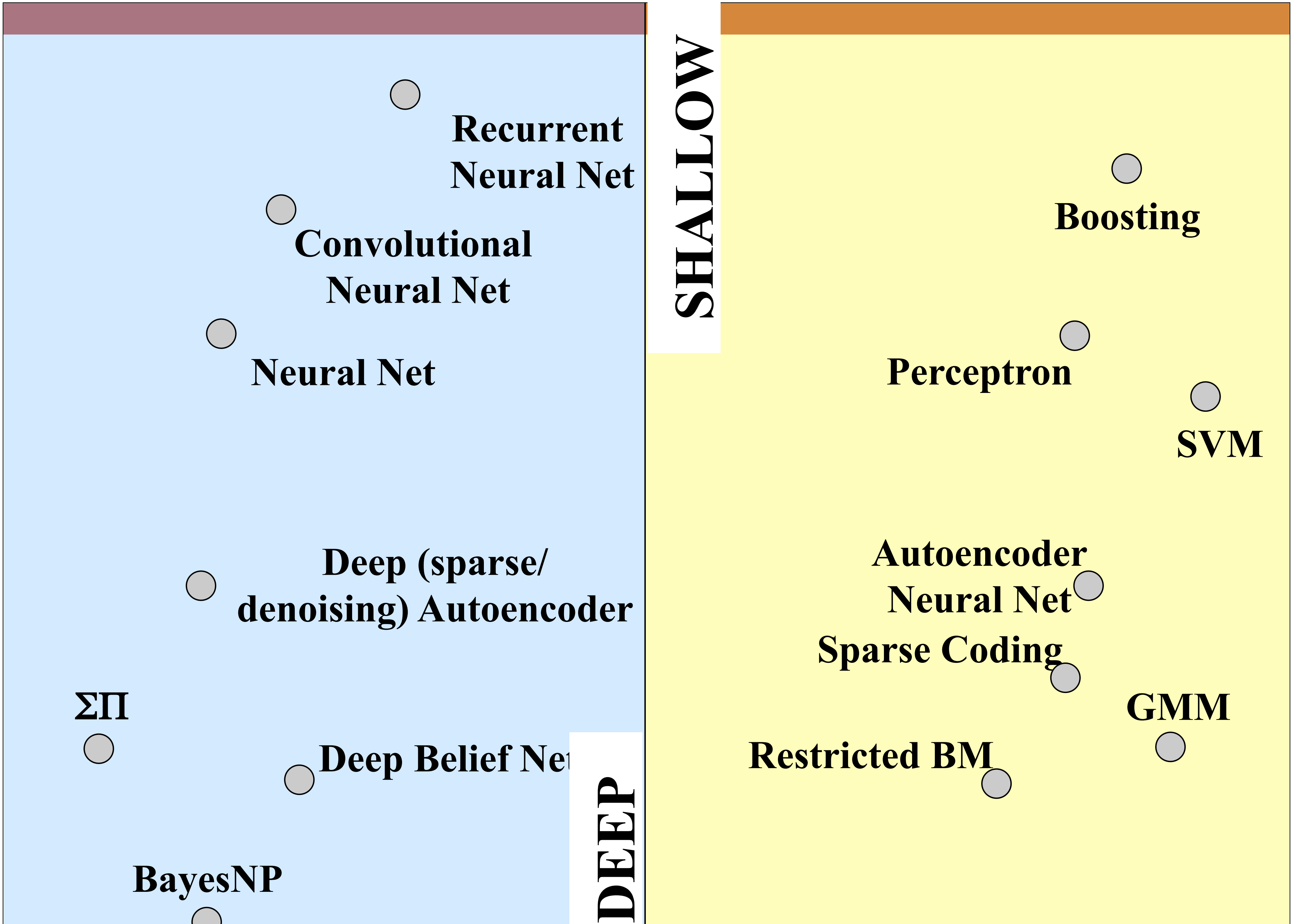
GMM

Deep Belief Net

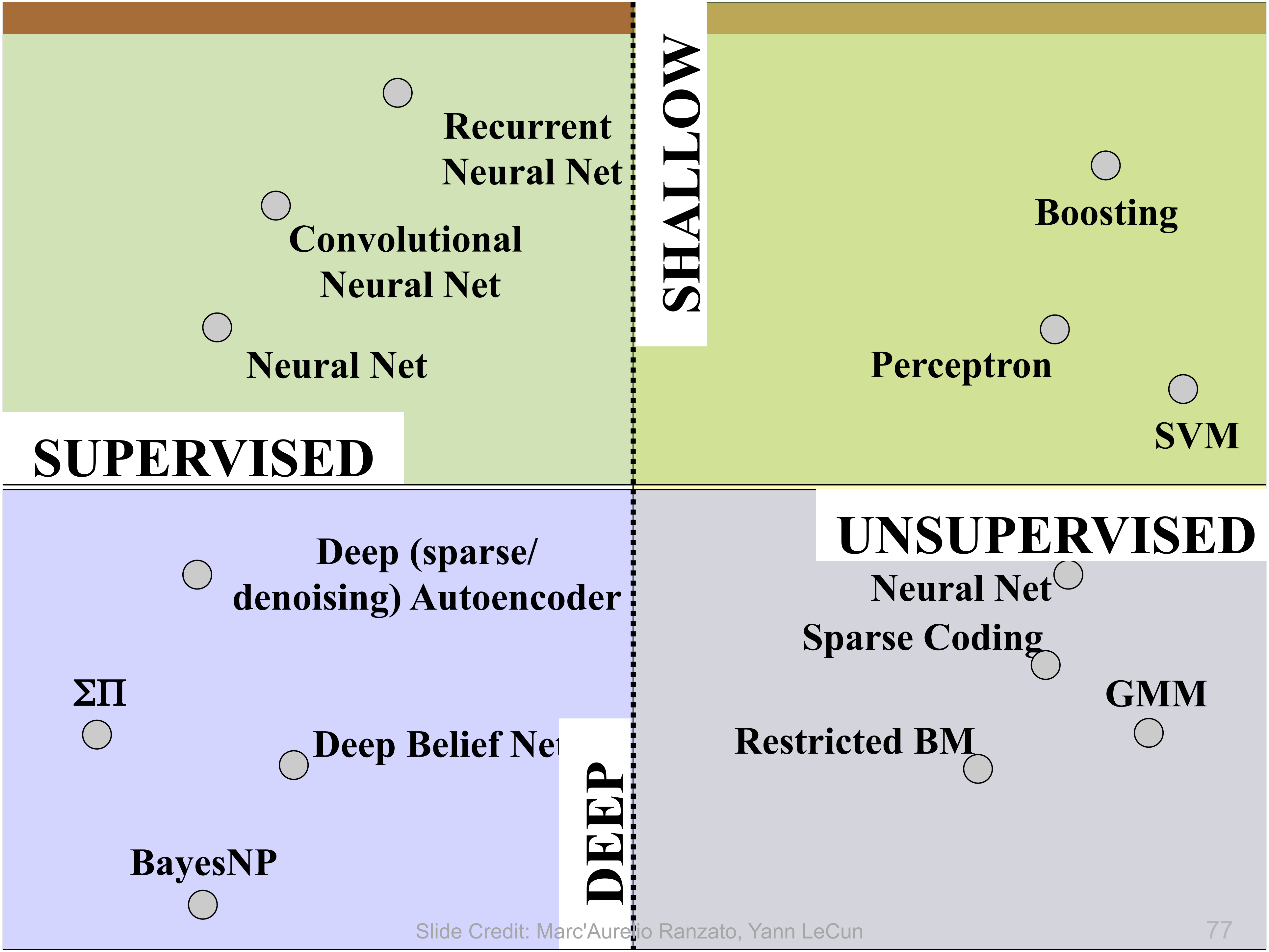
Restricted BM

BayesNP

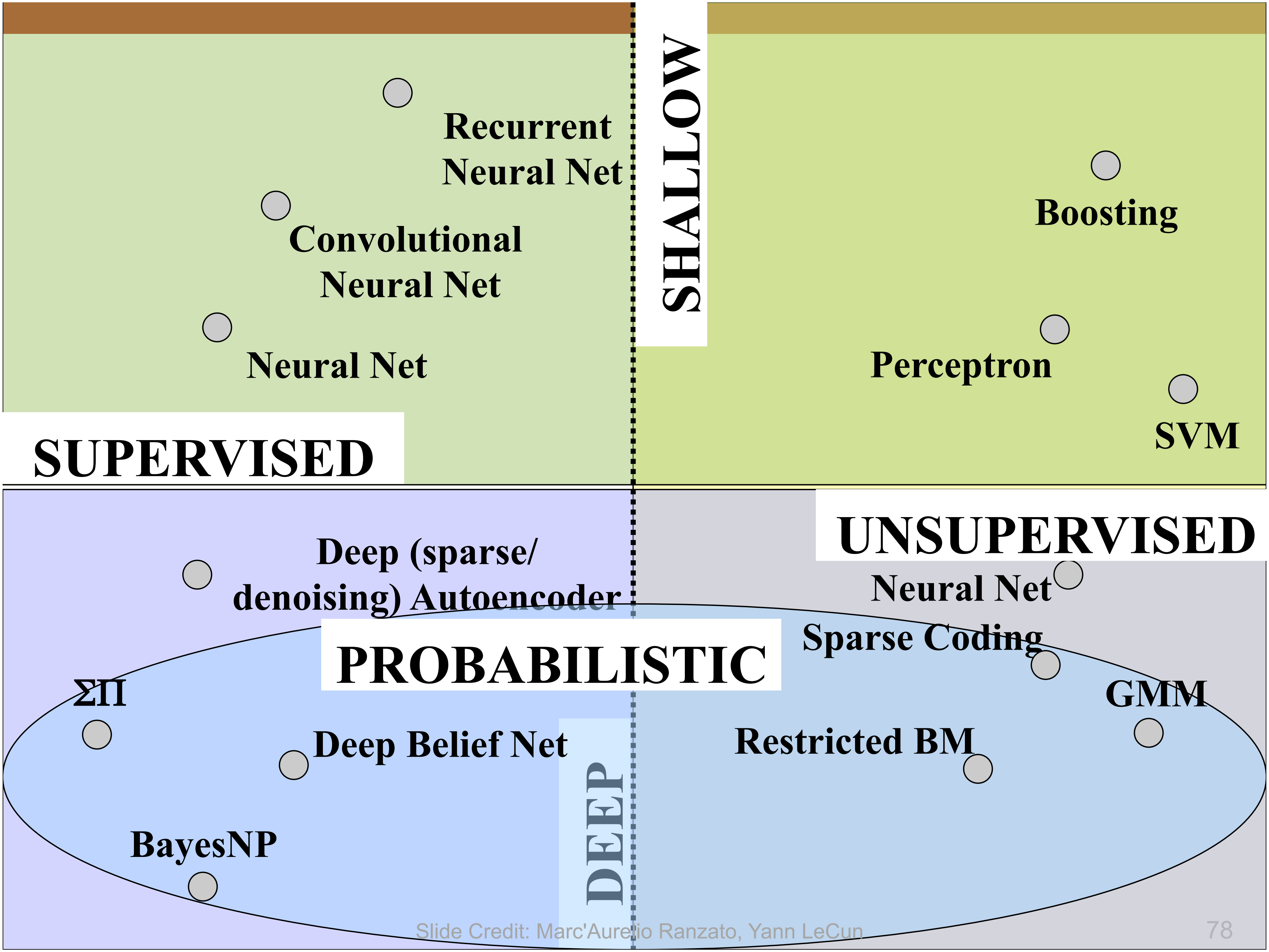
Disclaimer: showing only a subset of the known methods



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

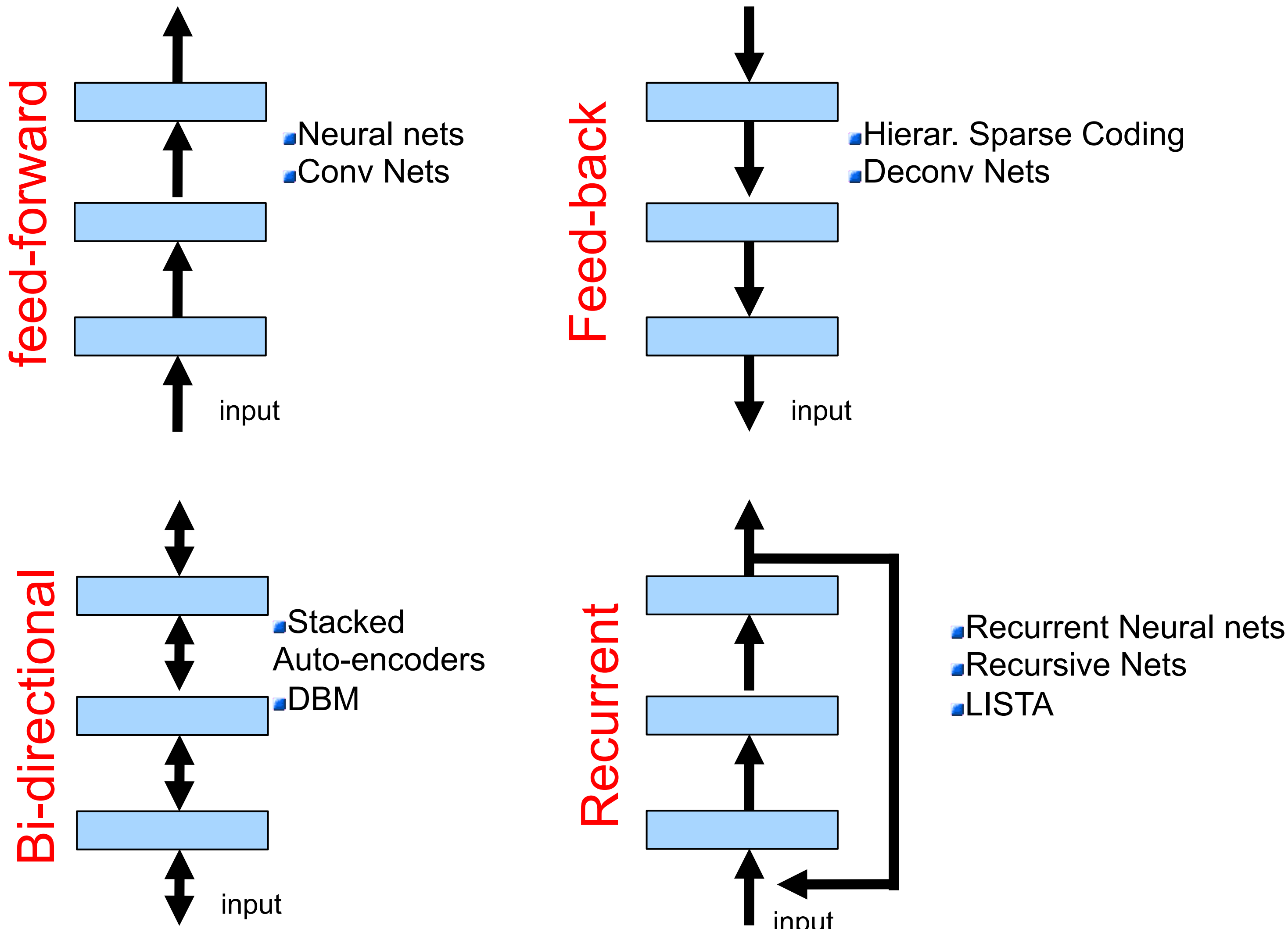


Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

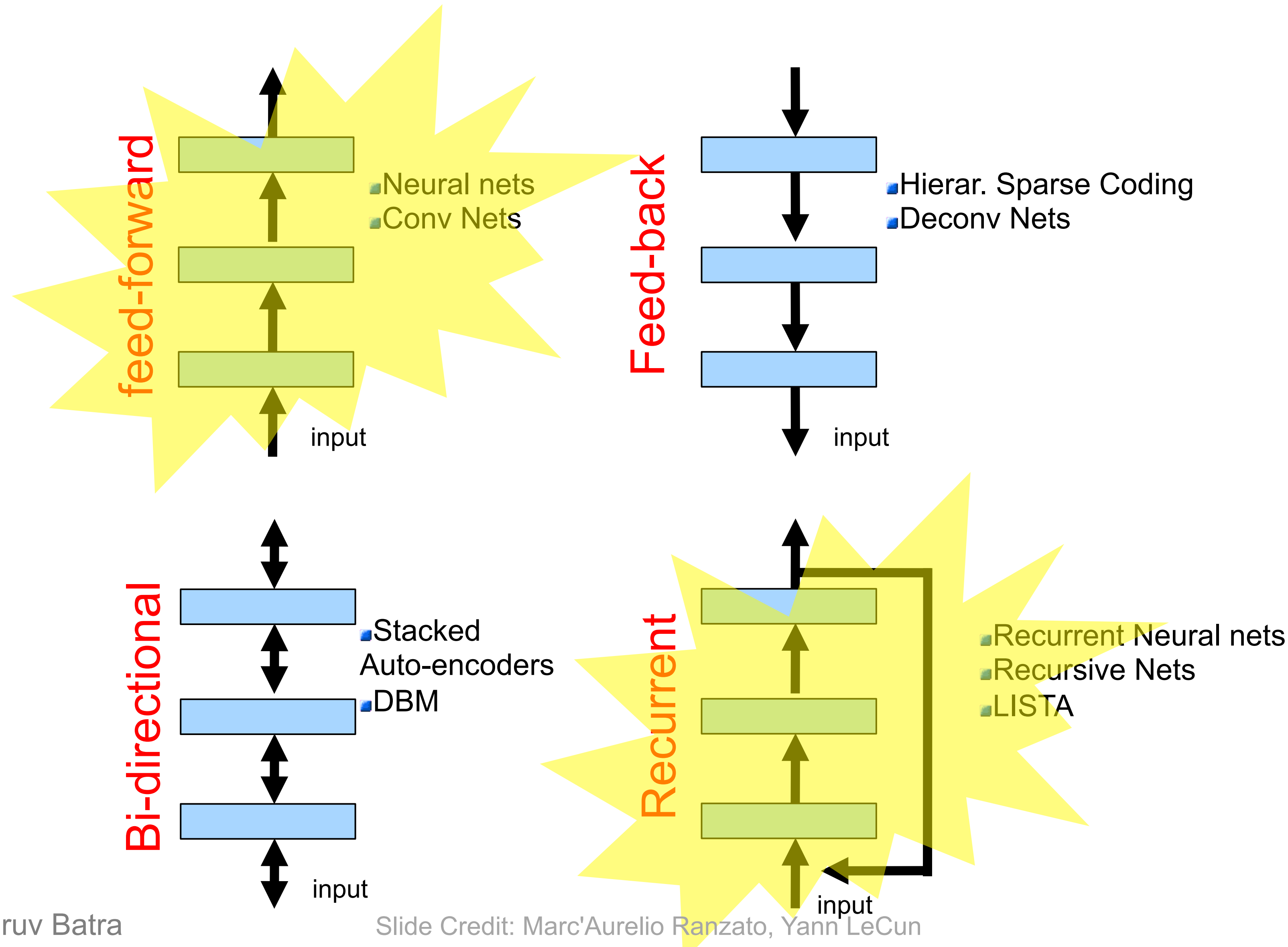


Recurrent
Neural Net
Convolutional
Neural Net
Neural Net
SHALLOW
Boosting
Perceptron
SVM
SUPERVISED
UNSUPERVISED
Deep (sparse/
denoising) Autoencoder
Neural Net
Sparse Coding
PROBABILISTIC
ΣΠ
GMM
Deep Belief Net
Restricted BM
BayesNP
DEEP

Main types of deep architectures



Focus of this class



Main types of learning protocols

- Purely supervised
    - Backprop + SGD
  - Good when there is lots of labeled data.

- Layer-wise unsupervised + superv. linear classifier
    - Train each layer in sequence using regularized auto-encoders or RBMs
    - Hold fix the feature extractor, train linear classifier on features
  - Good when labeled data is scarce but there is lots of unlabeled data.

- Layer-wise unsupervised + supervised backprop
    - Train each layer in sequence
    - Backprop through the whole system
  - Good when learning problem is very difficult.

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Focus of this class

- Purely supervised
    - Backprop + SGD
  - Good when there is lots of labeled data.
- Layer-wise unsupervised + superv. linear classifier
    - Train each layer in sequence using regularized auto-encoders or RBMs
    - Hold fix the feature extractor, train linear classifier on features
  - Good when labeled data is scarce but there is lots of unlabeled data.
- Layer-wise unsupervised + supervised backprop
    - Train each layer in sequence
    - Backprop through the whole system
  - Good when learning problem is very difficult.

Linear Classifier: SVM

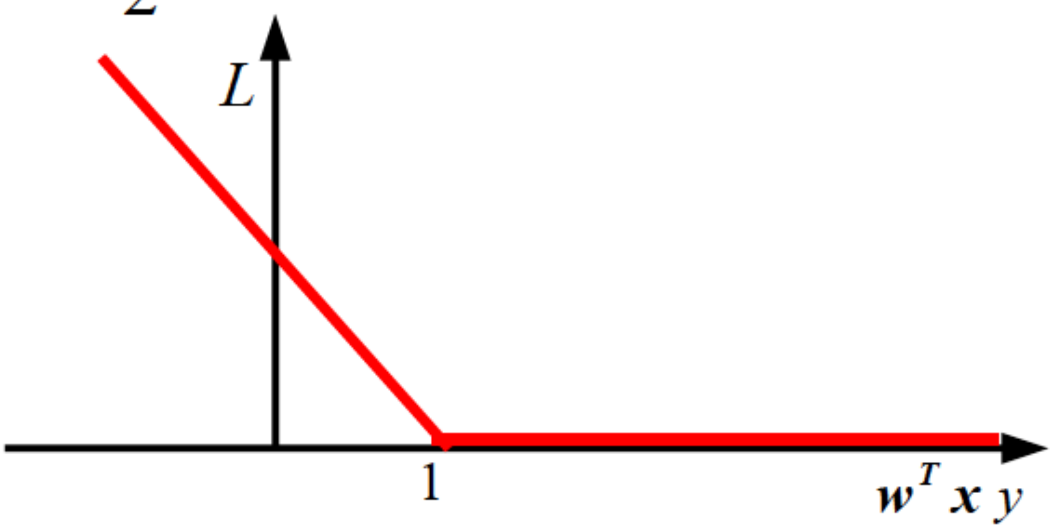
Input: $\boldsymbol{x} \in R^D$

Binary label: $y \in \{-1, +1\}$

Parameters: $\boldsymbol{w} \in R^D$

Output prediction: $\boldsymbol{w}^T \boldsymbol{x}$

Loss: $L = \frac{1}{2}\|\boldsymbol{w}\|^2 + \lambda \, max\,[0, 1 - \boldsymbol{w}^T \boldsymbol{x}\, y]$



Hinge Loss

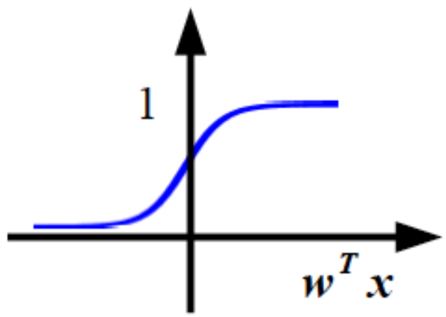
Ranzato

Linear Classifier: Logistic Regression

Input: $\boldsymbol{x} \in R^D$

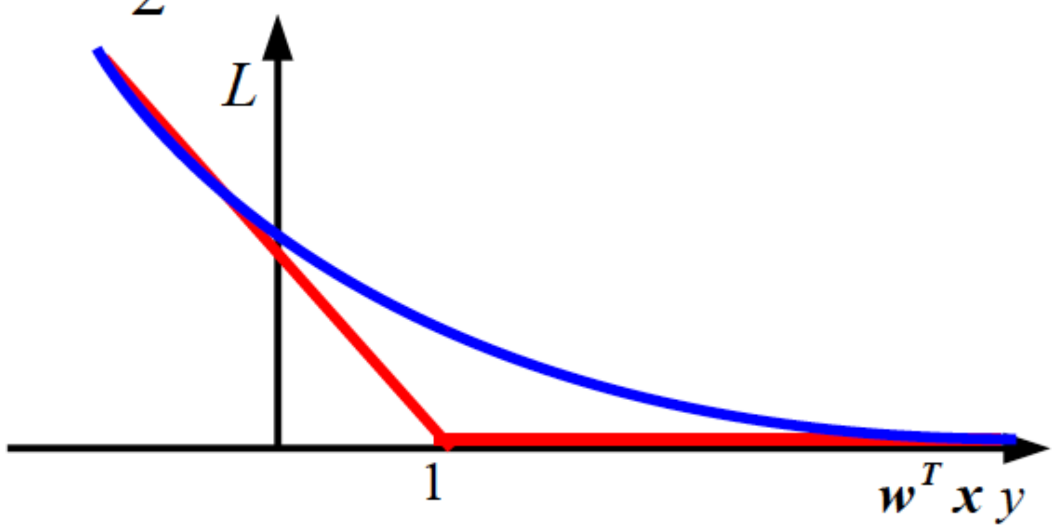
Binary label: $y \in \{-1, +1\}$

Parameters: $\boldsymbol{w} \in R^D$



Output prediction: $p(y=1|\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{w}^T \boldsymbol{x}}}$

Loss: $L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \lambda \log(p(y|\boldsymbol{x}))$

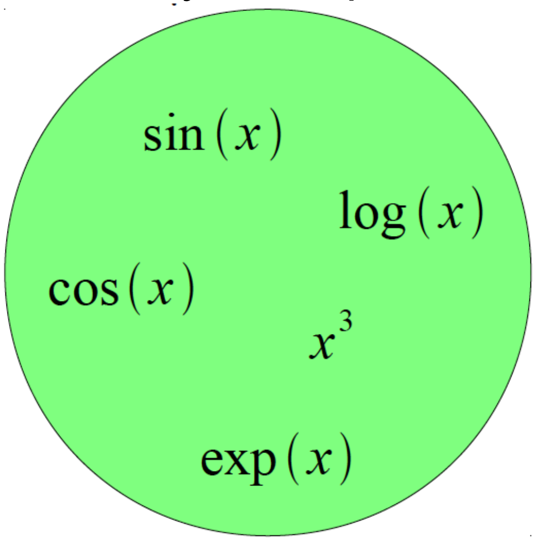


Log Loss

Ranzato

Logistic Regression as a Cascade

Given a library of simple functions



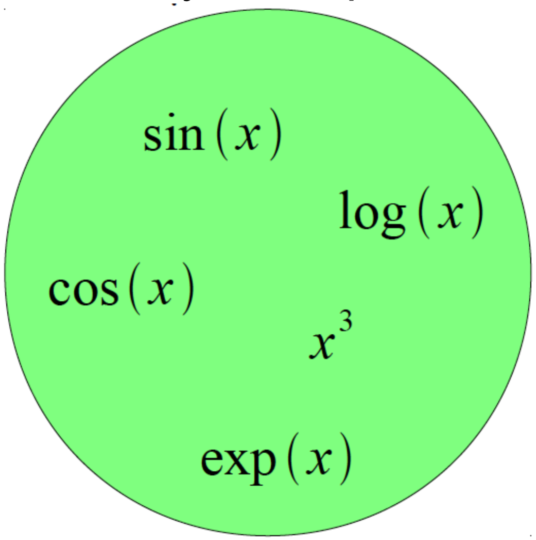
Compose into a

complicate function

$$-\log\left(\frac{1}{1+e^{-\mathbf{w}^\intercal \mathbf{x}}}\right)$$

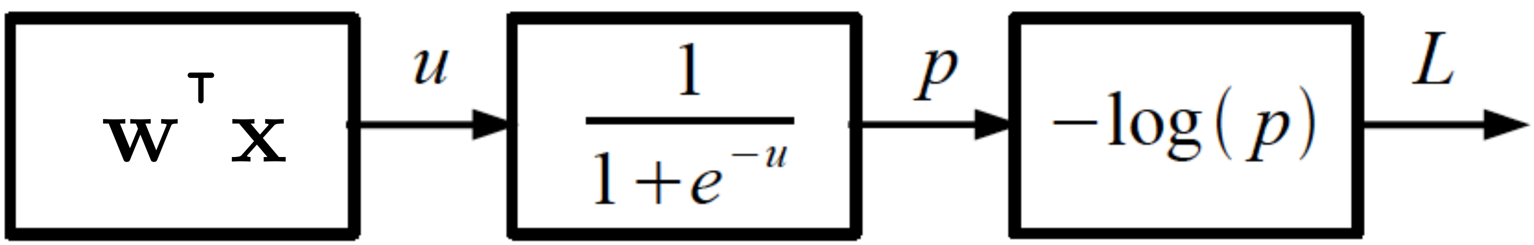
Logistic Regression as a Cascade

Given a library of simple functions

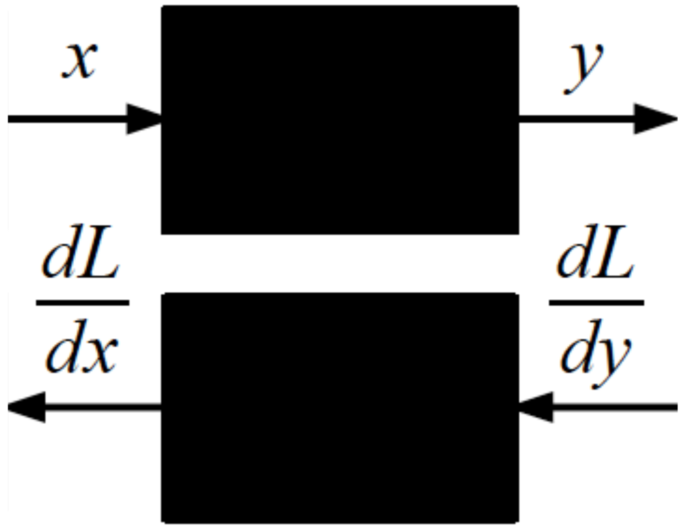


Compose into a complicate function

$$-\log\left(\frac{1}{1+e^{-\mathbf{w}^\intercal \mathbf{x}}}\right)$$

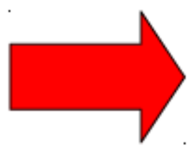


Chain Rule

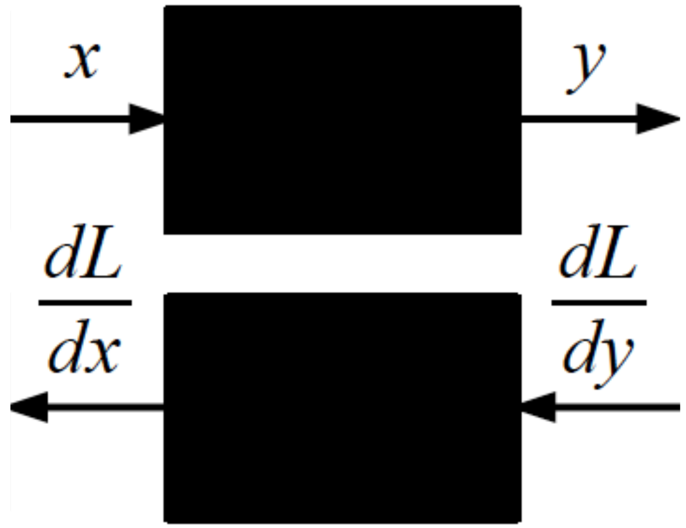


Given $y(x)$ and dL/dy,

What is dL/dx ?



Chain Rule

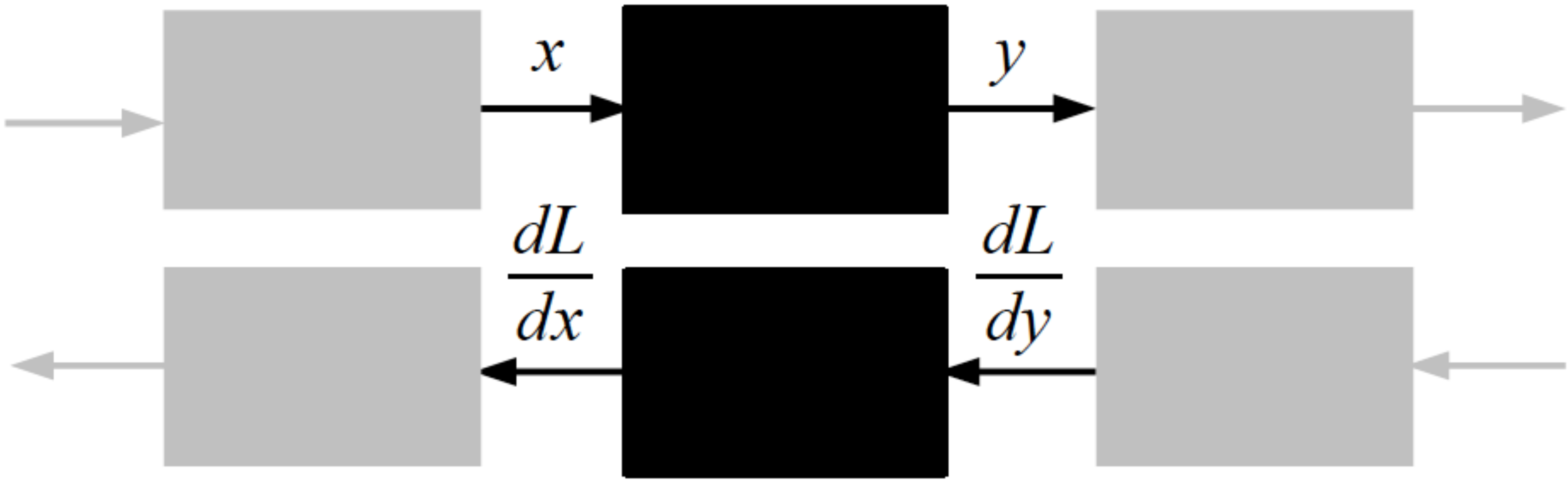


Given $y(x)$ and dL/dy,

What is dL/dx ?

$$\frac{dL}{dx} = \frac{dL}{dy} \cdot \frac{dy}{dx}$$

Chain Rule: All local

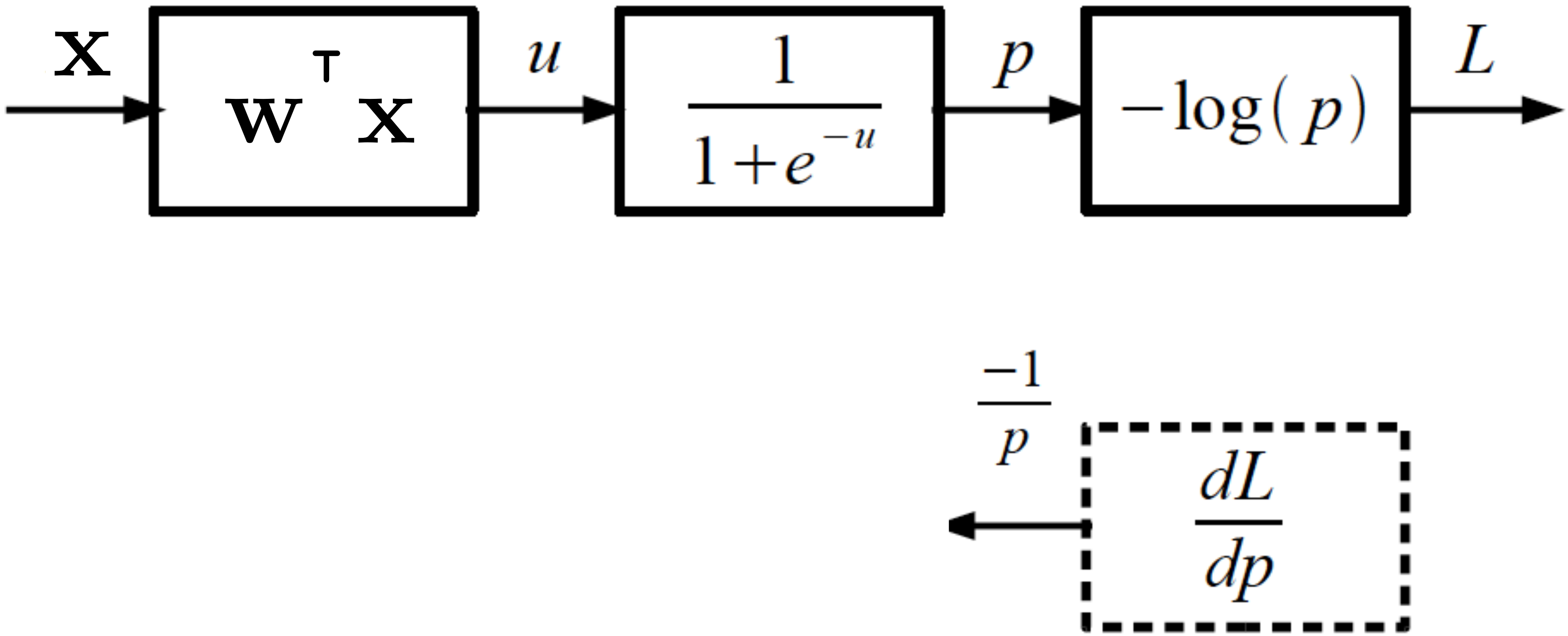


Given $y(x)$ and dL/dy,

What is dL/dx ?

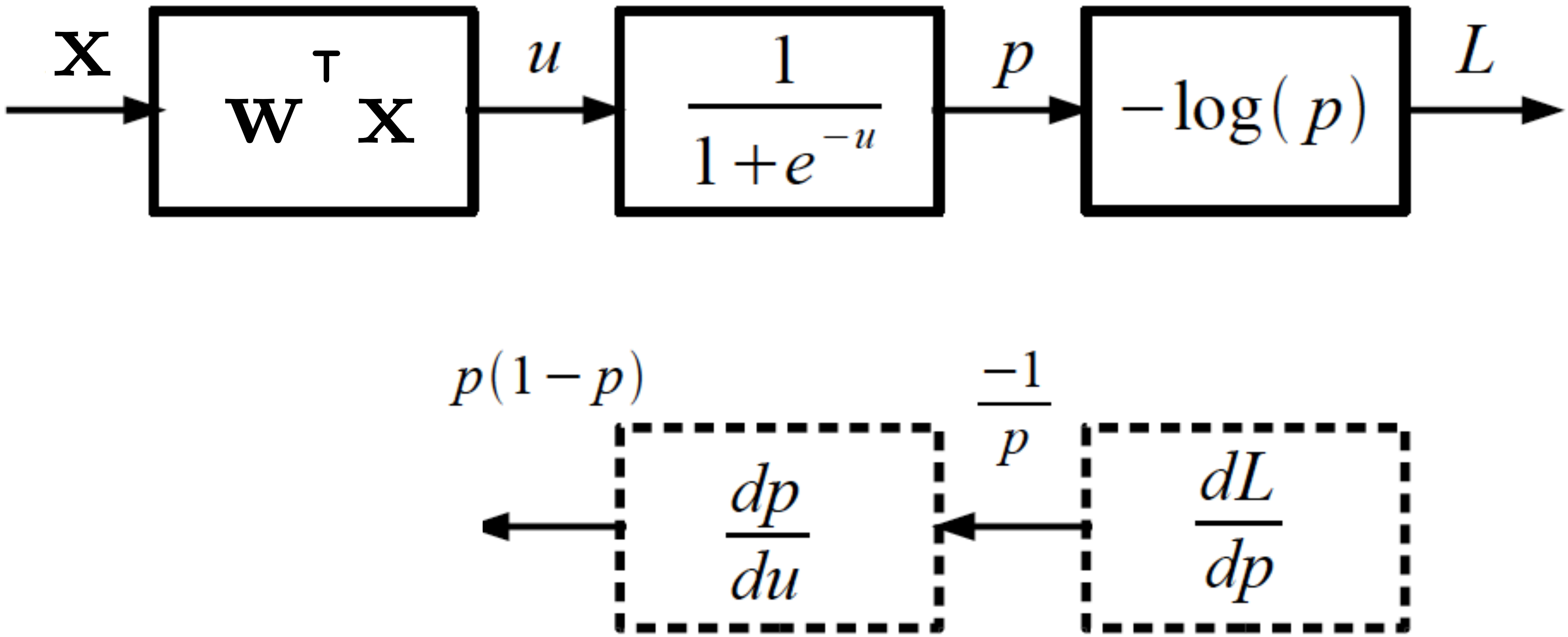
$$\frac{dL}{dx} = \frac{dL}{dy} \cdot \frac{dy}{dx}$$

Logistic Regression as a Cascade

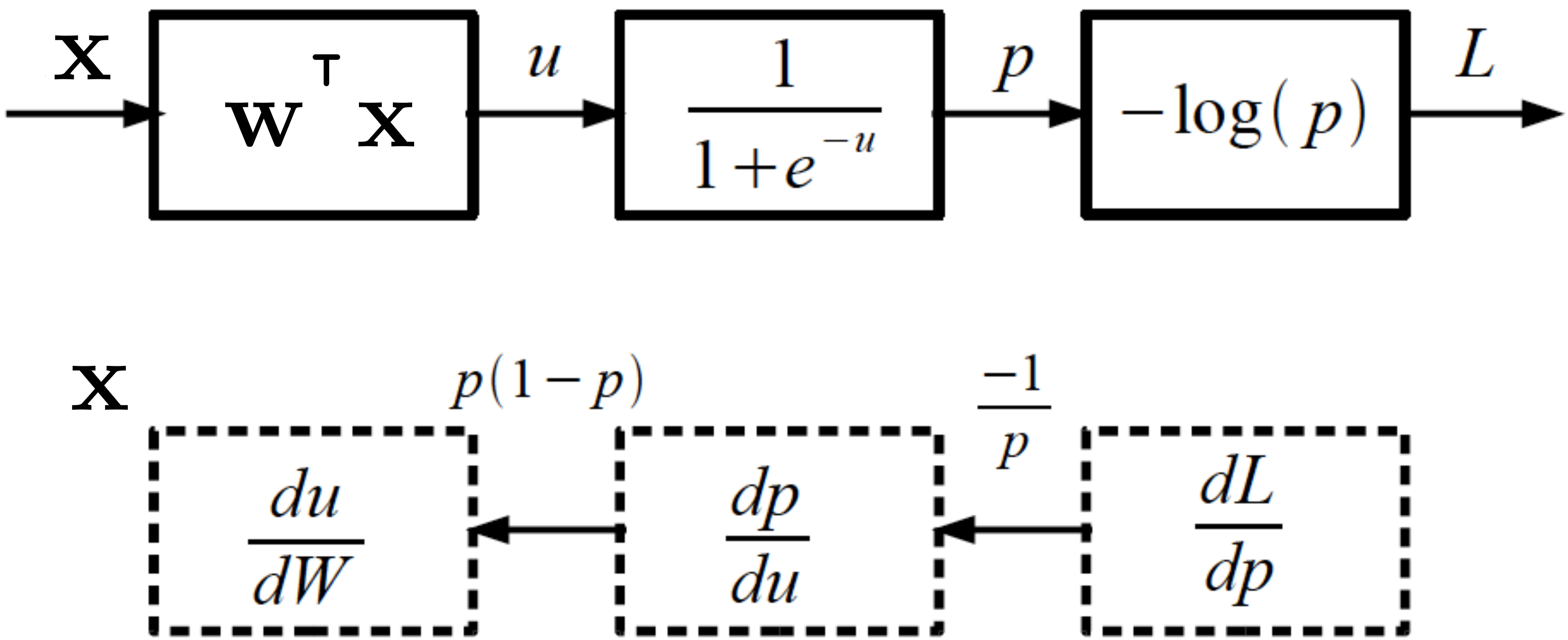


Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Logistic Regression as a Cascade



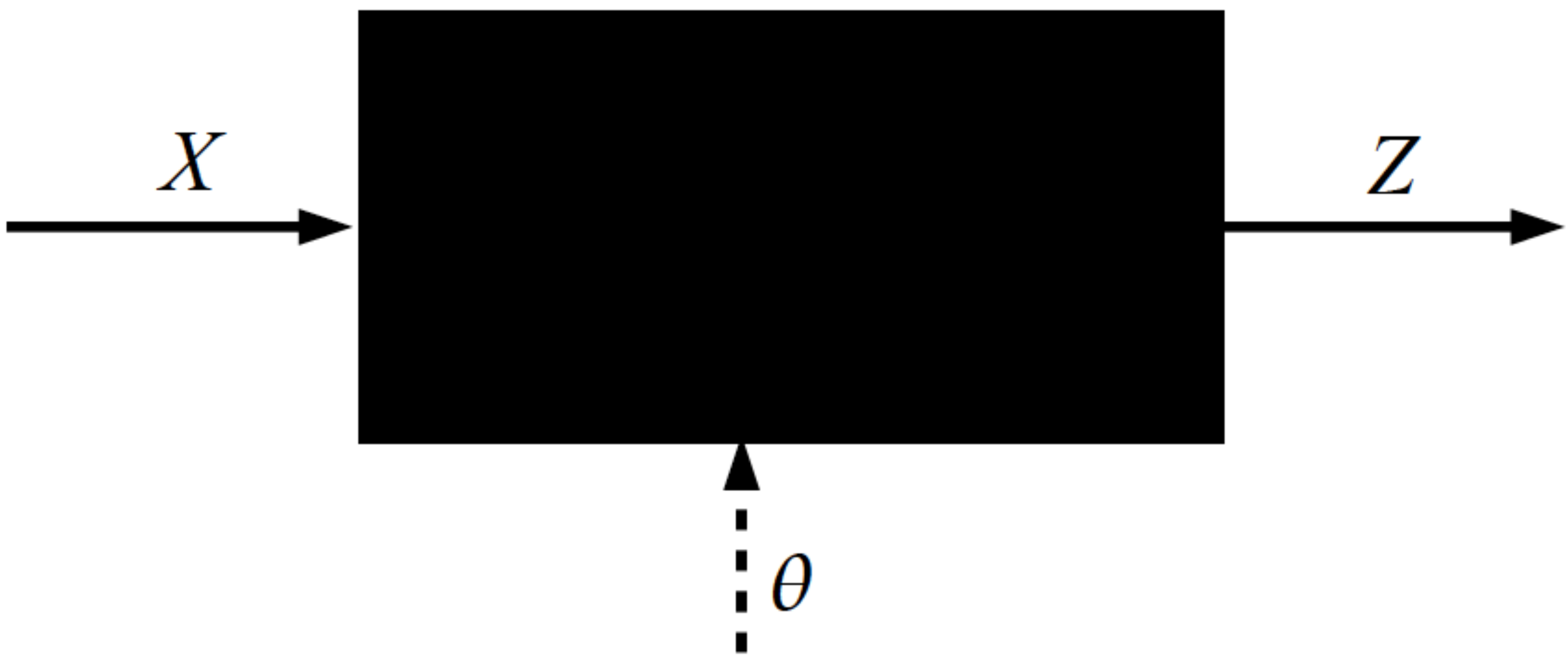
Logistic Regression as a Cascade



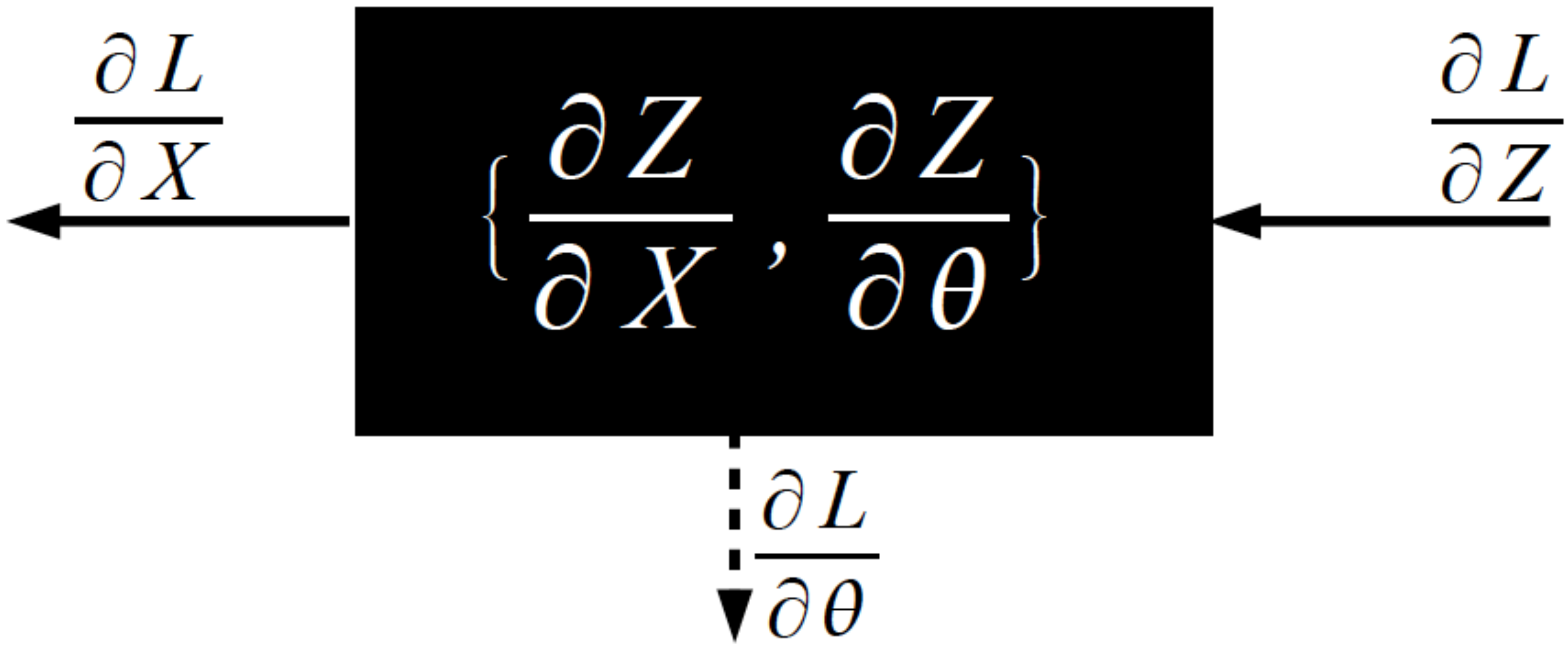
$$\frac{dL}{dW} = \frac{dL}{dp} \cdot \frac{dp}{du} \cdot \frac{du}{dW} = (p-1)\mathbf{x}$$

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Key Computation: Forward-Prop

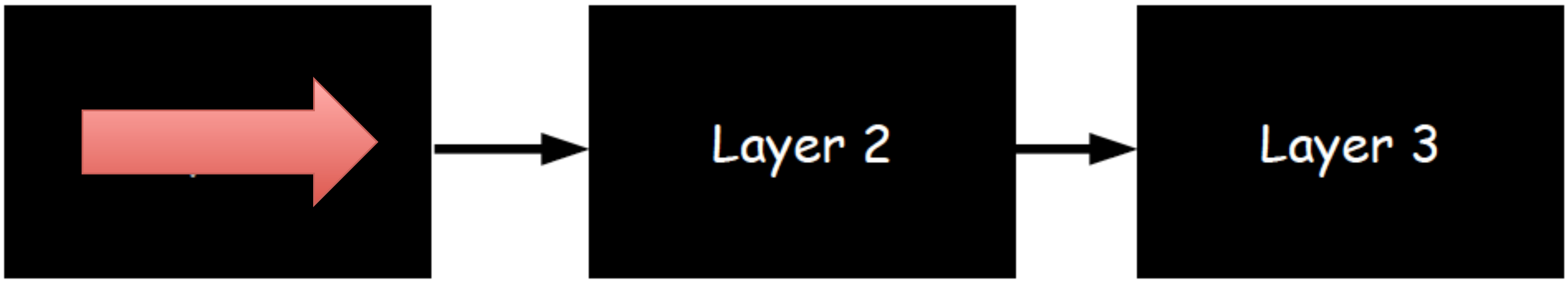


Key Computation: Back-Prop



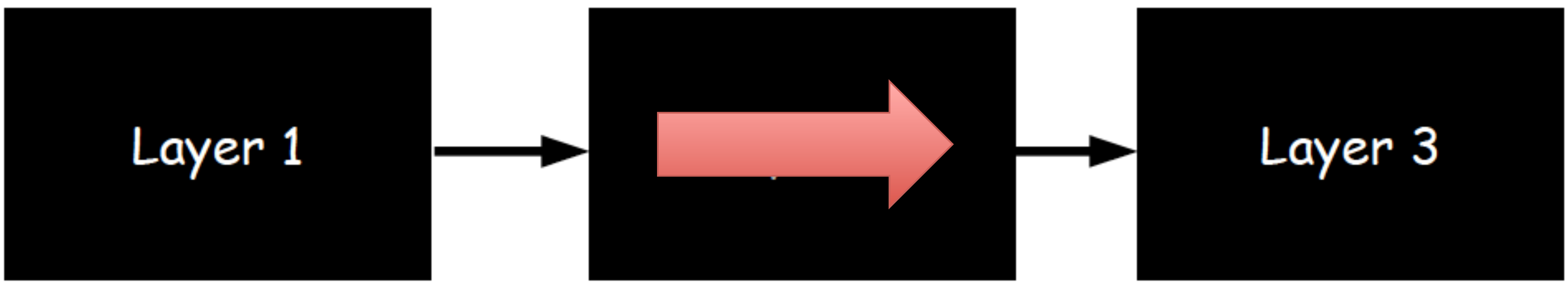
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]



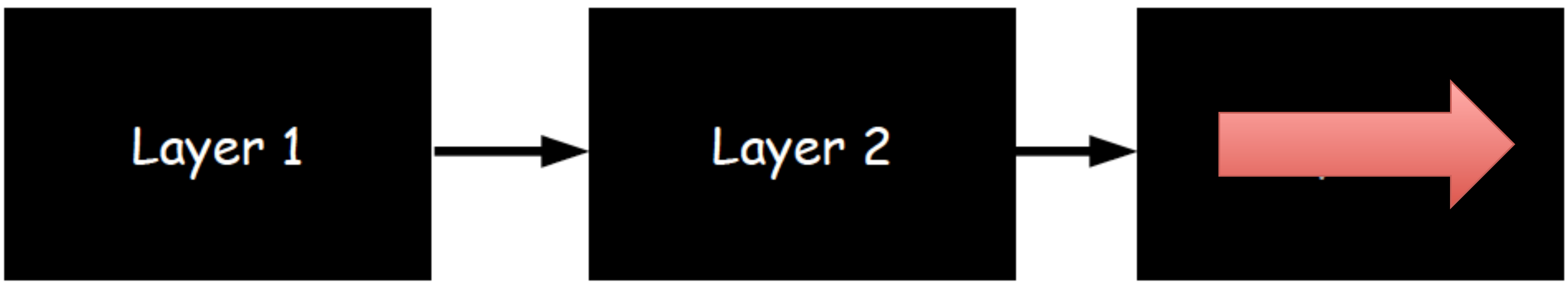
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]



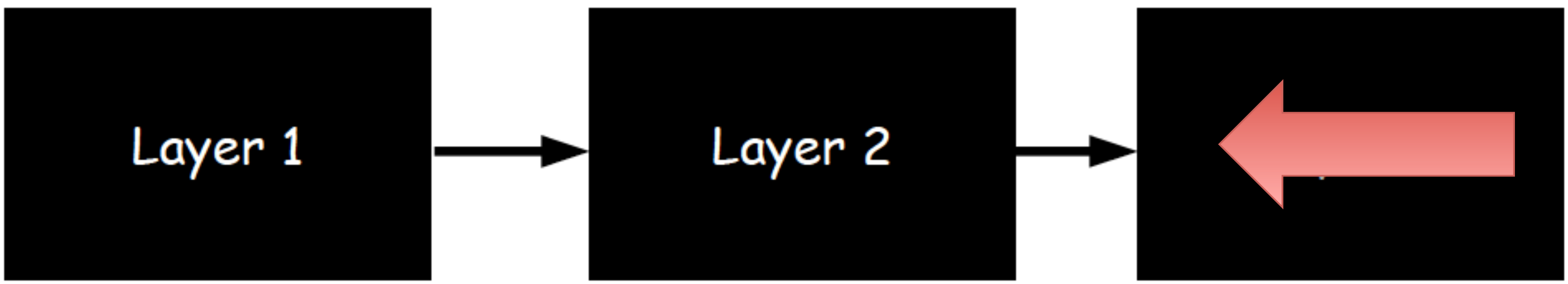
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]



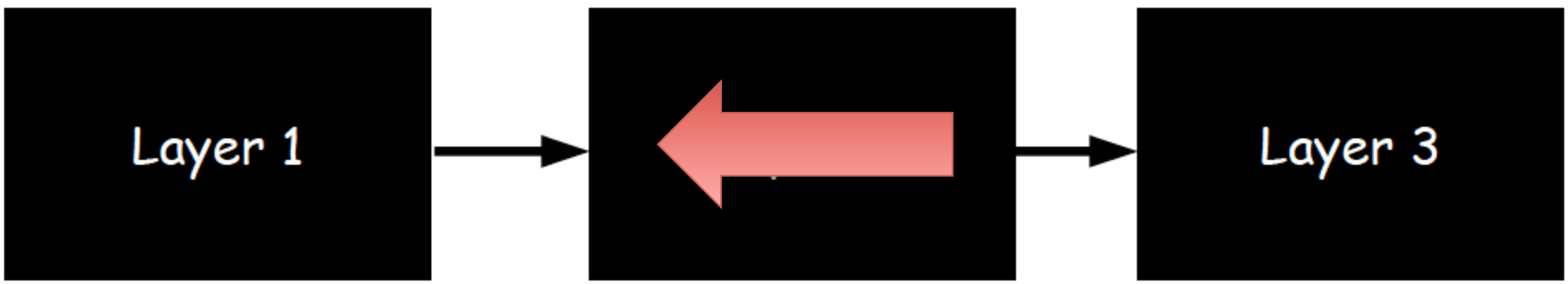
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]



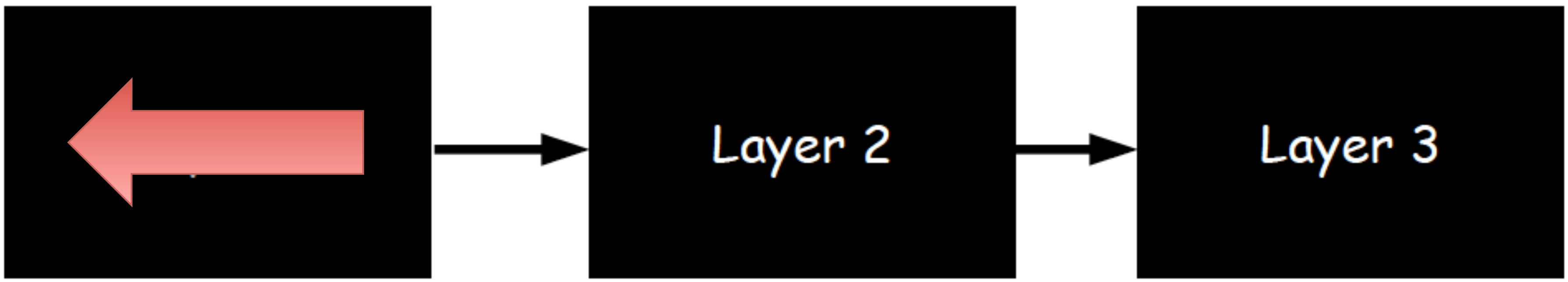
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]



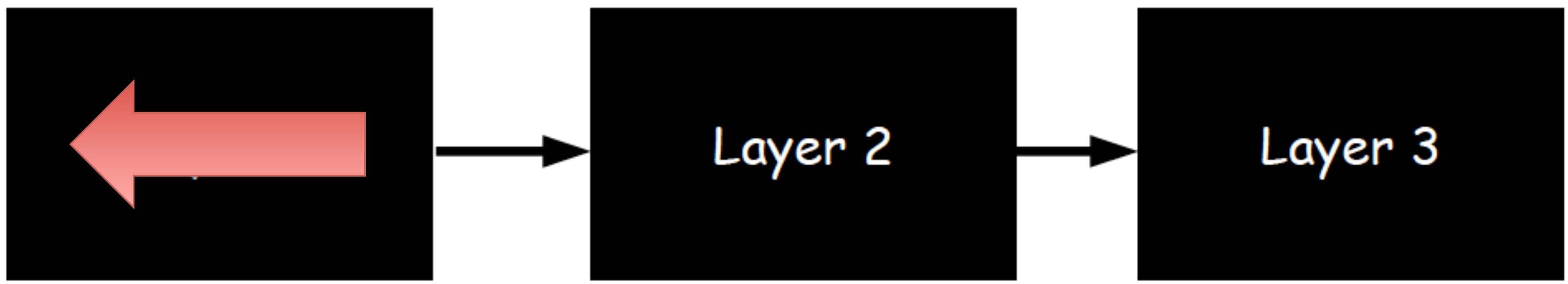
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]



Neural Network Training

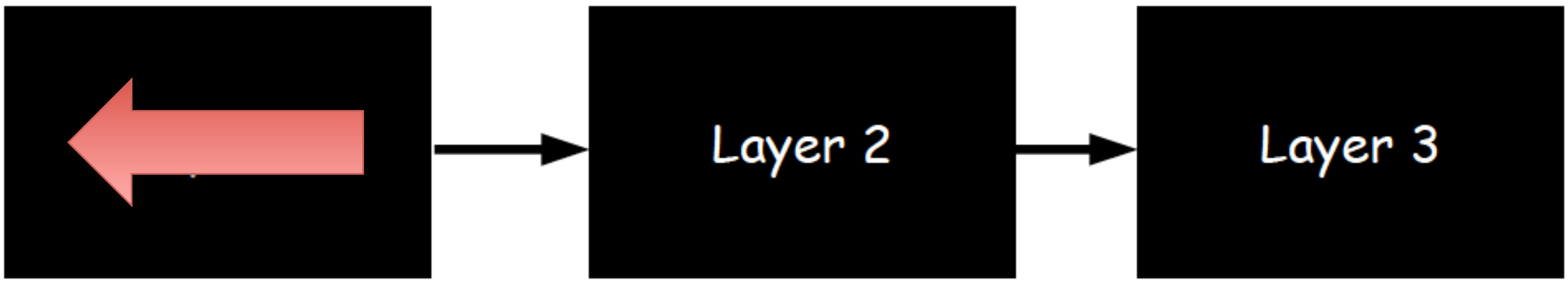
- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]
- Step 3: Use gradient to update parameters



$$\theta \leftarrow \theta - \eta \frac{dL}{d\,\theta}$$

Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]
- Step 3: Use gradient to update parameters
  - With momemtum



$$\theta \leftarrow \theta - \eta \, \Delta$$

$$\Delta \leftarrow 0.9 \, \Delta + \frac{\partial L}{\partial \theta}$$

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Course Information

- Instructor: Dhruv Batra
  - dbatra@vt
  - Office Hours: Fri 3-4pm
  - Location: 468 Whittemore

TAs

Ashwin Kalyan

B. Tech., NIT Surathkal

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

Abhishek Das

B. Tech., IIT Roorkee

http://abhishekdas.com/

Syllabus

- Background & Basics
  - Neural Networks, Backprop, Optimization (SGD)
- Module 1: Convolutional Neural Networks (CNNs)
  - Architectures, Pre-training, Fine-tuning
  - Visualizations, Fooling ConvNets, Adversarial examples, Inverting Representations
  - Different tasks: segmentation ConvNets
- Module 2: Recurrent Neural Networks (RNNs)
  - Difficulty of learning; “Vanilla” RNNs, LSTMs, GRU
  - RNNs for Sequence-to-Sequence (machine translation & image captioning)
- Module 3: Beyond RNNs
  - CNNs + RNNs for Visual Question Answering (VQA)
  - Learning to execute, Memory Networks
- Module 4: Advanced Topics
  - Bayesian Neural Networks, Hyper-parameter optimization
  - Different regularizers

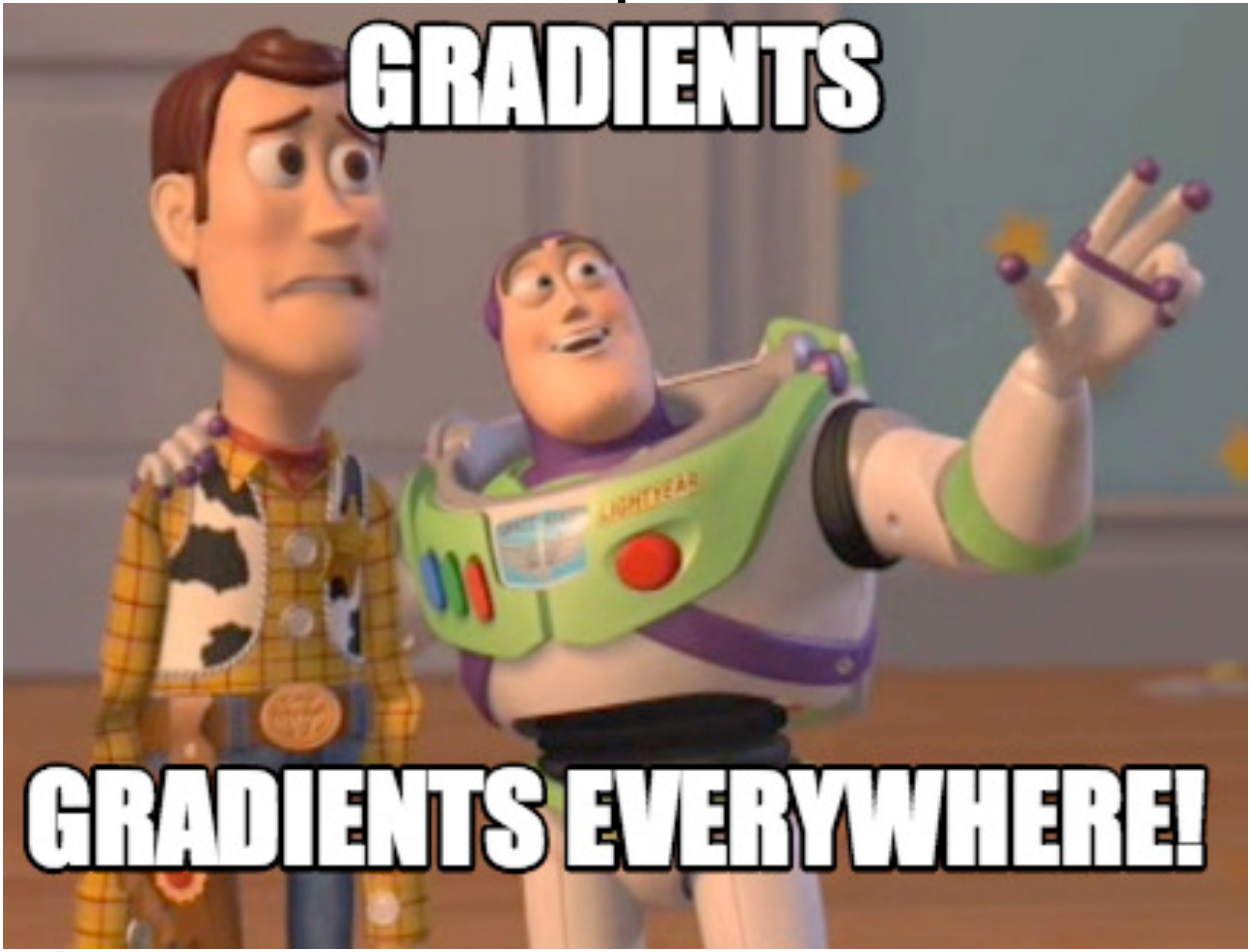
Syllabus

- You will learn about the methods you heard about
- But we are not teaching "how to use a toolbox"
- You will understand algorithms, ~~theory~~, applications, and implementations
- **It's going to be FUN and HARD WORK ☺**

Prerequisites

- Intro Machine Learning
  - Classifiers, regressors, loss functions, MLE, MAP
- Linear Algebra
  - Matrix multiplication, eigenvalues, positive semi-definiteness…
- Calculus
  - Multi-variate gradients, hessians, jacobians…

Prerequisites



Prerequisites

- Intro Machine Learning
  - Classifiers, regressors, loss functions, MLE, MAP
- Linear Algebra
  - Matrix multiplication, eigenvalues, positive semi-definiteness…
- Calculus
  - Multi-variate gradients, hessians, jacobians…
- Programming!
  - Homeworks will require Python, C++, and Lua!
  - Libraries/Frameworks: Caffe and Torch
  - HW0 (pure python), HW1 (python + Caffe), HW2 (Caffe), HW3+4 (Torch)
  - Your language of choice for project


FINISHED DEEP LEARNING
HOMEWORK!!

Organization & Deliverables

- 4 homeworks (40%)
  - First one goes out next week
    - Start early, Start early, Start early, Start early, Start early, Start early, Start early, Start early, Start early, Start early
- Paper Reviews (15%)
  - Read 1 paper per class
  - Submit reviews before class
- Paper Presentations (15%)
  - [Tentative] 1 presentation in the semester
  - Practice run with a TA 3-4 days before scheduled date
- Final project (25%)
  - Projects done individually, or groups of two students
- Class Participation (5%)
  - Contribute to class discussions on Scholar
  - Ask questions, answer questions

Homeworks

- Homeworks are hard, start early!
  - Due in 2 weeks via Scholar (Assignments tool)
  - Pure Implementation
  - Kaggle Competitions:
    - https://inclass.kaggle.com/c/VT-ECE-Machine-Learning-HW1

- "Free" Late Days
  - 5 late days for the semester
    - Use for HW, project proposal/report
    - Cannot use for HW0, reviews, or presentations

  - After free late days are used up:
    - 25% penalty for each late day

HW0

- Out today; due Monday (08/31)
  - Available on class webpage + Scholar

- Grading
  - Does not count towards grade.
  - BUT Pass/Fail.
  - <=75% means that you might not be prepared for the class

- Topics
  - Implement a multi-class SVM and soft-max classifier
  - SGD on two different losses
  - Hyperparameter optimization with a standard package

Paper Reviews

- Length
  - <=1 page, 11 pt Times New Roman, 1 inch margins
- Due: Midnight before class

- Organization
  - Summary:
    - What is this paper about? What is the main contribution? Describe the main approach & results. Just facts, no opinions yet.
  - List of positive points / Strengths:
    - Is there a new theoretical insight? Or a significant empirical advance? Did they solve a standing open problem? Or is a good formulation for a new problem? Or a faster/better solution for an existing problem? Any good practical outcome (code, algorithm, etc)? Are the experiments well executed? Useful for the community in general?
  - List of negative points / Weaknesses:
    - What would you do differently? Any missing baselines? missing datasets? any odd design choices in the algorithm not explained well? quality of writing? Is there sufficient novelty in what they propose? Has it already been done? Minor variation of previous work? Why should anyone care? Is the problem interesting and significant?
  - Reflections
    - How does this relate to other papers we have read? What are the next research directions in this line of work?

Presentations

- Frequency
  - [Tentative] Once in the semester

- Expectations
  - Read all papers for that day
  - Overview of that day's theme (e.g. Visualizing ConvNets)
  - Present details of at least 2 papers in detail
    - Describe formulation, experiment, approaches, datasets
    - Encouraged to present a broad picture
    - Show results; demo code if possible
  - How do different papers related to each other?
  - Please clearly cite the source of each slide that is not your own.
  - No review needed
  - Meet with TA 3-4 days before class to dry run presentation
    - Worth 40% of presentation grade

Project

- Goal
  - Chance to try Deep Learning
  - Encouraged to apply to your research (computer vision, NLP, UAVs, computational biology…)
  - Must be done this semester. No double counting.
  - Can combine with other classes
    - get permission from both instructors; delineate different parts
  - Extra credit for shooting for a publication

- Main categories
  - Application/Survey
    - Compare a bunch of existing algorithms on a new application domain of your interest
  - Formulation/Development
    - Formulate a new model or algorithm for a new or old problem
  - Theory
    - Theoretically analyze an existing algorithm

Spring 2013 Projects

- Poster/Demo Session

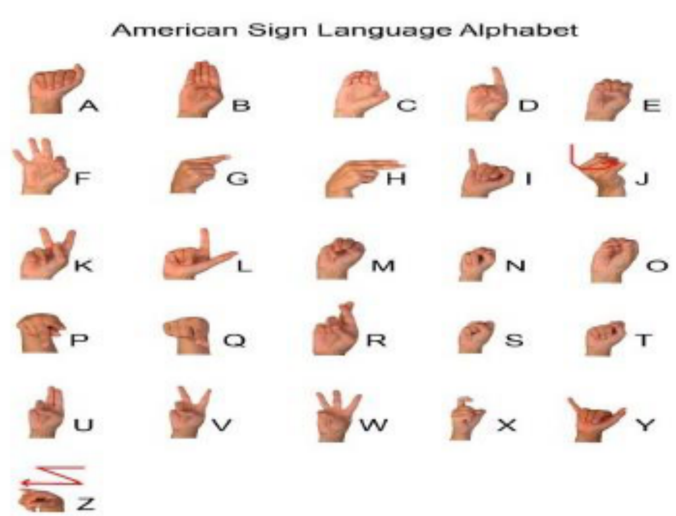
Spring 2013 Projects

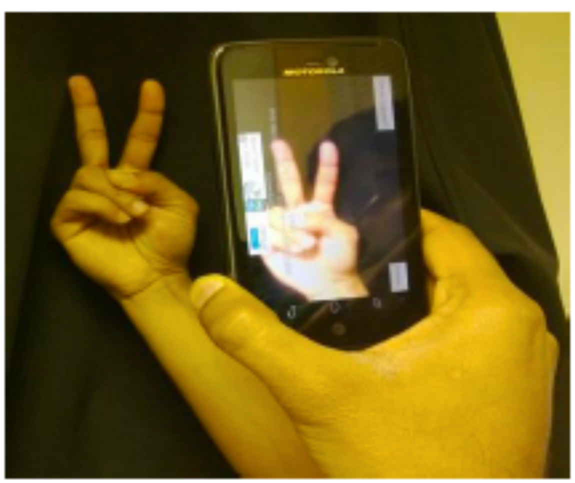
- Gesture Activated Interactive Assistant
  - Gordon Christie & Ujwal Krothpalli, Grad Students
  - http://youtu.be/VFPAHY7th9A?t=42s

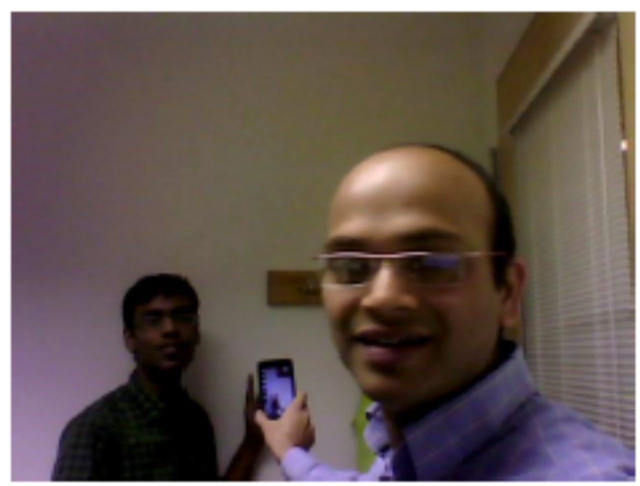
Figure 7: A simple 2D pose estimation in a controlled setting.

Spring 2013 Projects

- American Sign Language Detection
  - Vireshwar Kumar & Dhiraj Amuru, Grad Students







Re-grading Policy

- Homework assignments
  - **Within 1 week** of receiving grades: see the TAs

- This is an advanced grad class.
  - The goal is understanding the material and making progress towards our research.

Collaboration Policy

- Collaboration
  - Only on HWs and project (not allowed in HW0).
  - You may discuss the questions
  - Each student writes their own answers
  - Write on your homework anyone with whom you collaborate
  - Each student must write their own code for the programming part

- Zero tolerance on plagiarism
  - Neither ethical nor in your best interest
  - Always credit your sources
  - Don’t cheat. We will find out.

Waitlist / Audit / Sit in

- Waitlist
  - Do HW0. Come to first few classes.
  - Let's see how many people drop.

- Audit
  - Make presentation

- Sitting in
  - Talk to instructor.

Communication Channels

- Primary means of communication -- Scholar Forum
  - No direct emails to Instructor unless private information
  - Instructor/TAs can provide answers to everyone on forum
  - Class participation credit for answering questions!
  - No posting answers. We will monitor.

- Staff Mailing List
  - f15ece6504-staff-g@vt.edu

- Class websites:
  - https://scholar.vt.edu/portal/site/f15ece6504
  - https://computing.ece.vt.edu/~f15ece6504/

- Office Hours

How to do well in class?

- Come to class!
  - Sit in front; ask question
  - This is the most important thing you can do

- One point
  - No laptops or screens in class

Other Relevant Classes

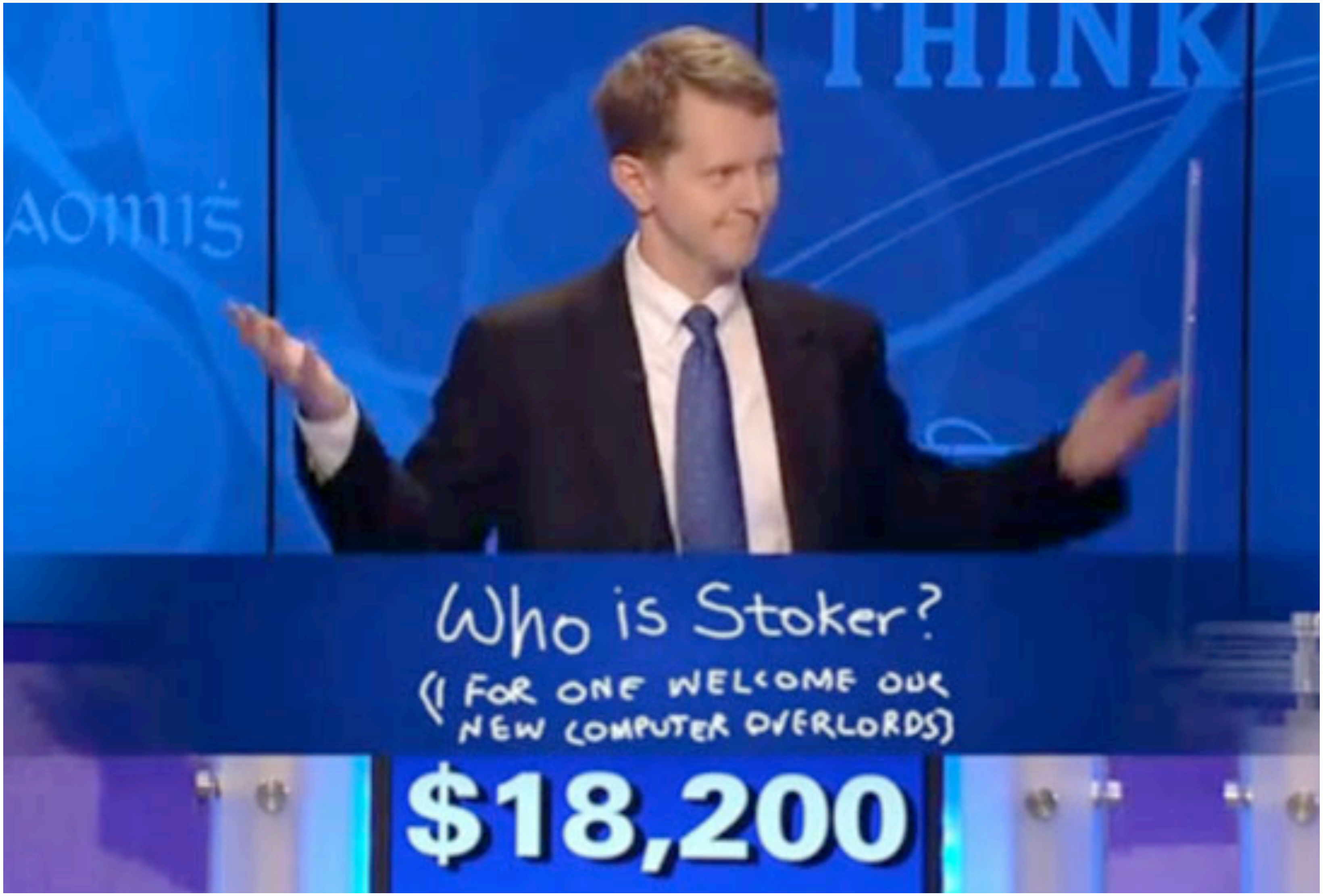
- https://filebox.ece.vt.edu/~dbatra/faq.html

Todo

- HW0
  - Due Monday 11:55pm

- Paper Presentations
  - Start looking at schedule
  - Find papers you are interested in

Welcome


THINK
Who is Stoker?
(I FOR ONE WELCOME OUR NEW COMPUTER OVERLORDS)
$18,200