# CS 291A: Deep Learning for NLP

## Introduction & Logistics

William Wang
UCSB Computer Science
william@cs.ucsb.edu

Some slides adapted from H. Lee and Y. Chen.

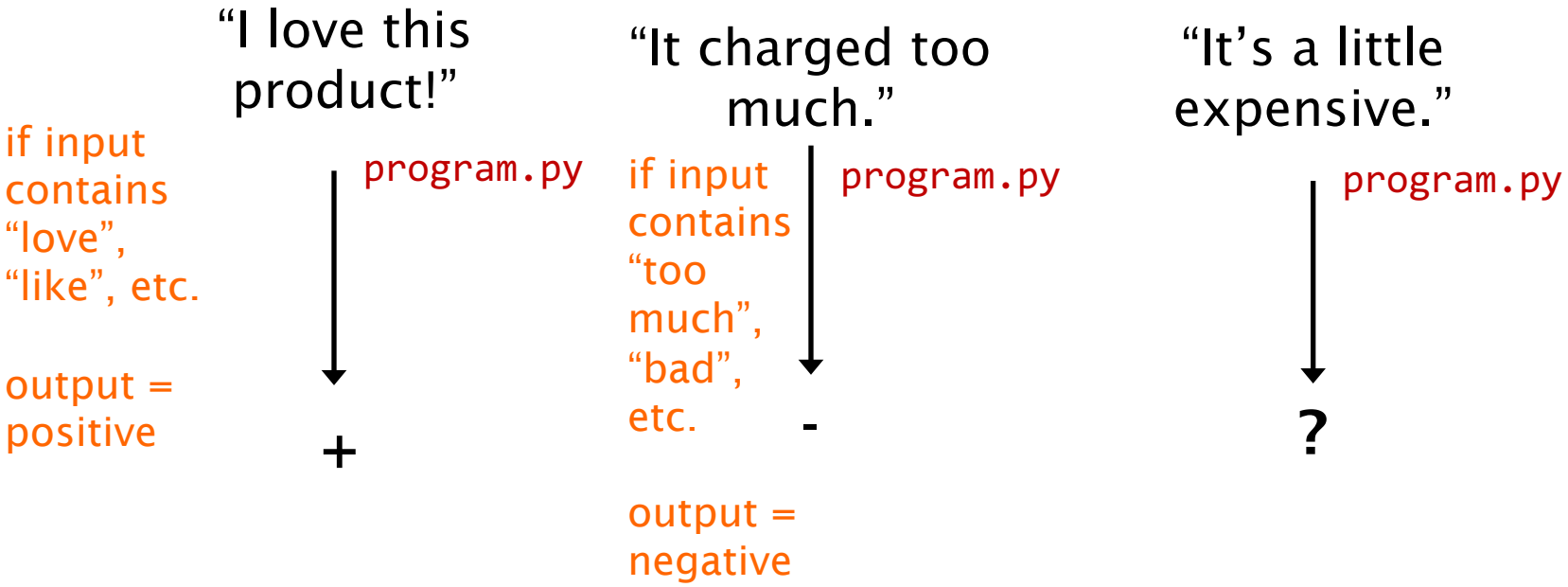
How would you explain "Machine Learning" to a non-technical friend?

# What Computers Can Do?


- Programs that deterministically map inputs to outputs.

# Program for Solving NLP Tasks

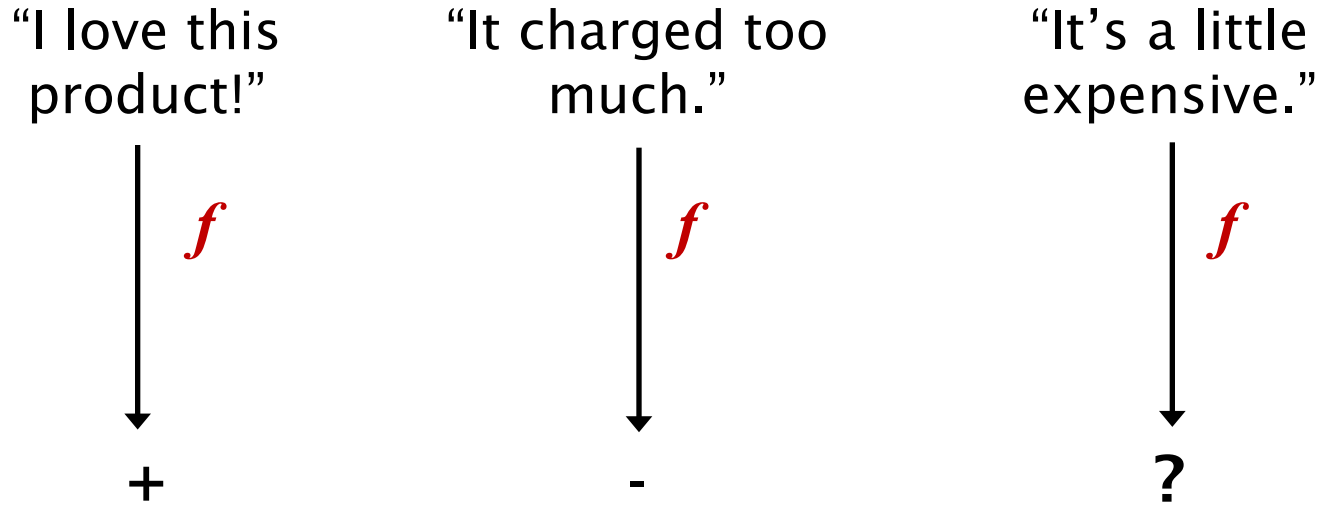
- Task: predicting positive or negative sentiments given a product review.




Some tasks are complex, and we don't know how to write a program to solve them.

# Learning ≈ Find a Function

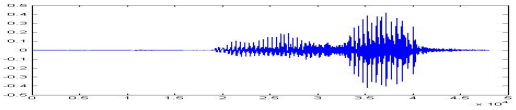
- Task: predicting positive or negative sentiments given a product review.




Given a large amount of data, the machine learns what the function $f$ should be.

# Learning ≈ Find a Function

- Speech Recognition

$f($  $)=$ "你好"

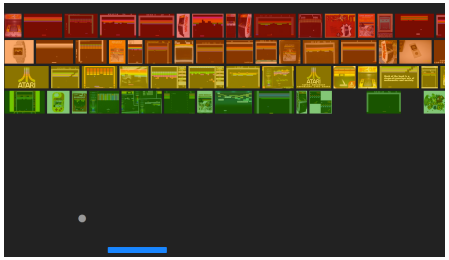
- Handwritten Recognition

$f($  $)=$ "2"

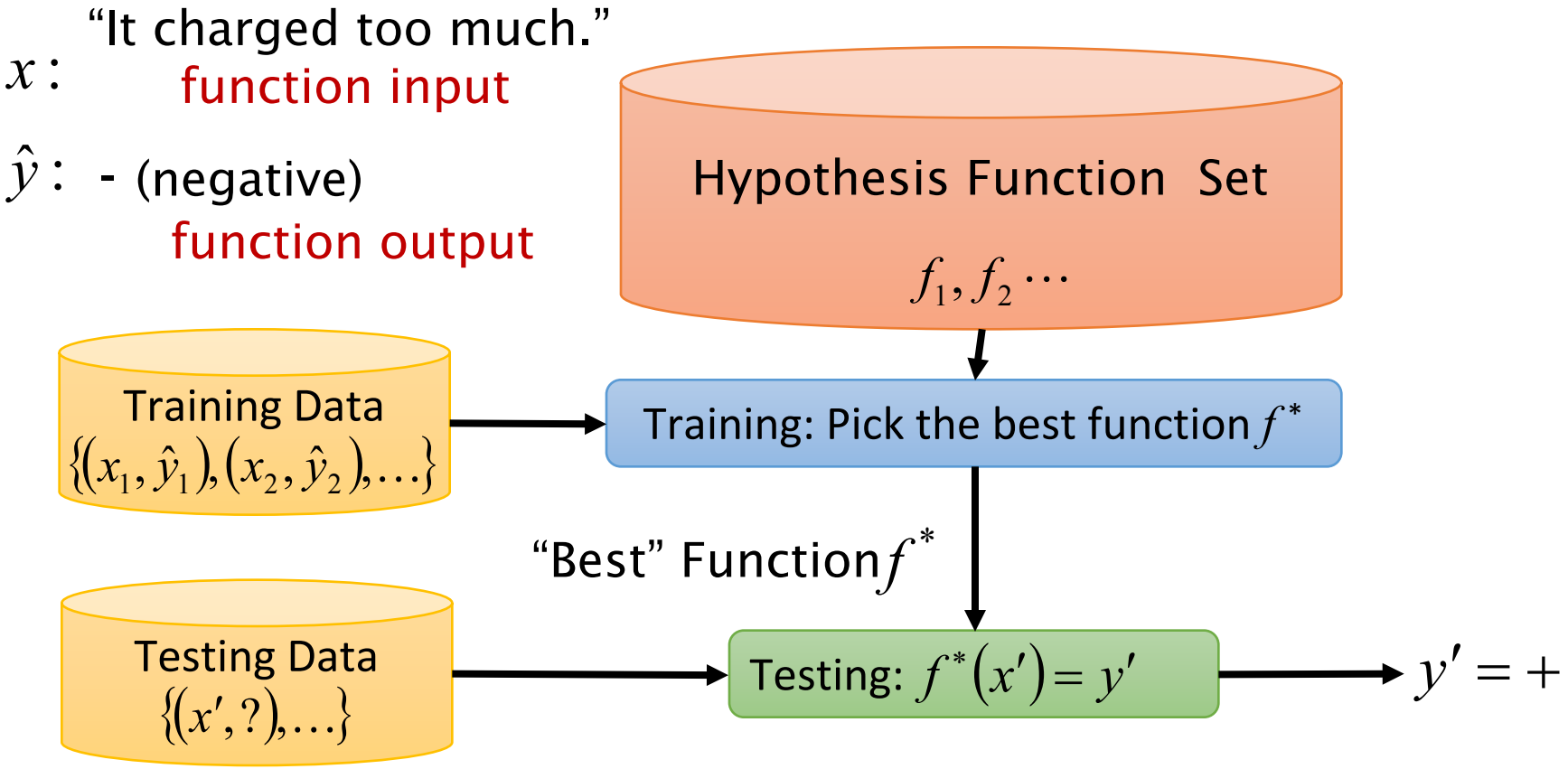
- Weather forecast

$f($  Thursday $)=$ "  Sat"

- Play video games

$f($  $)=$ "move left"

# Machine Learning Framework

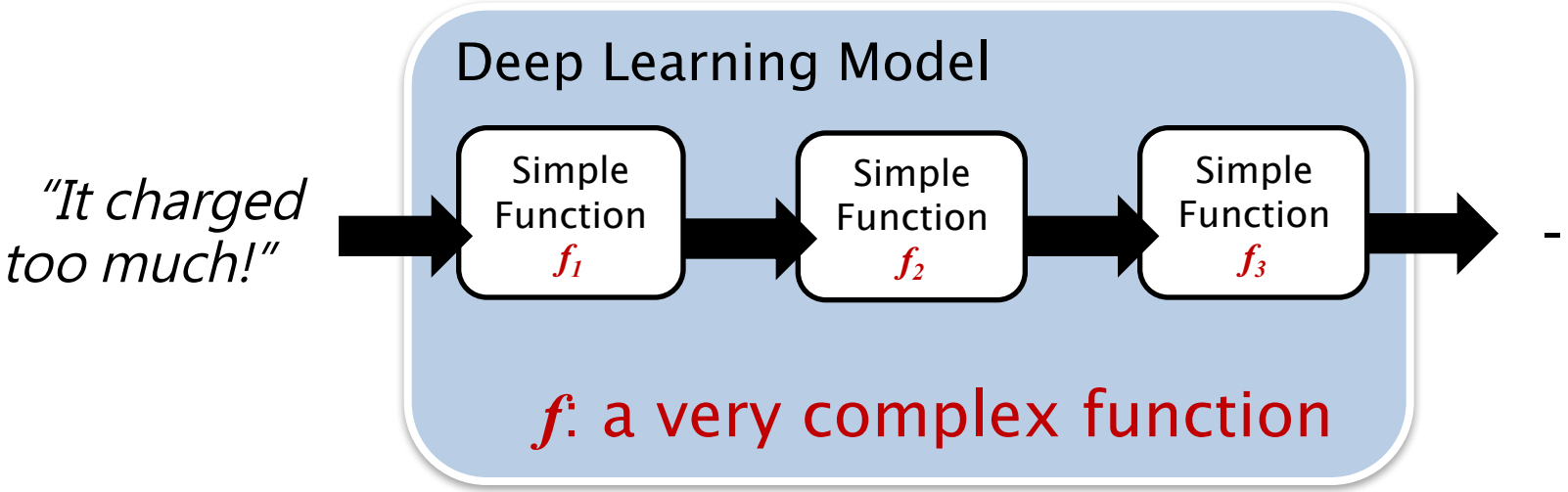



Training is to pick the best function given the observed data
Testing is to predict the label using the learned function

How would you explain
“Deep Learning” to
a non-technical friend?

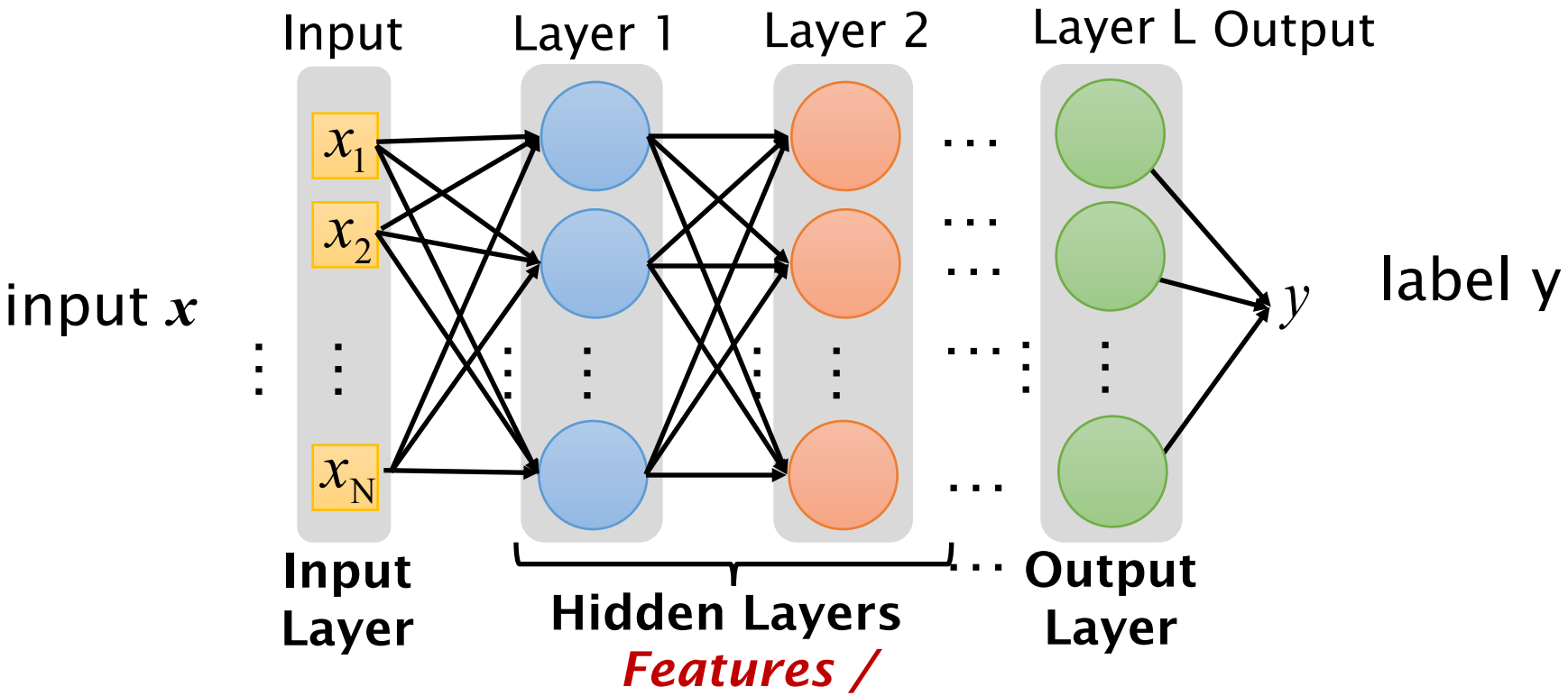
# Nested Functions

- Production line




Deep learning usually refers to *neural network* based model
(rebranding of an old concept)

# Nested Functions

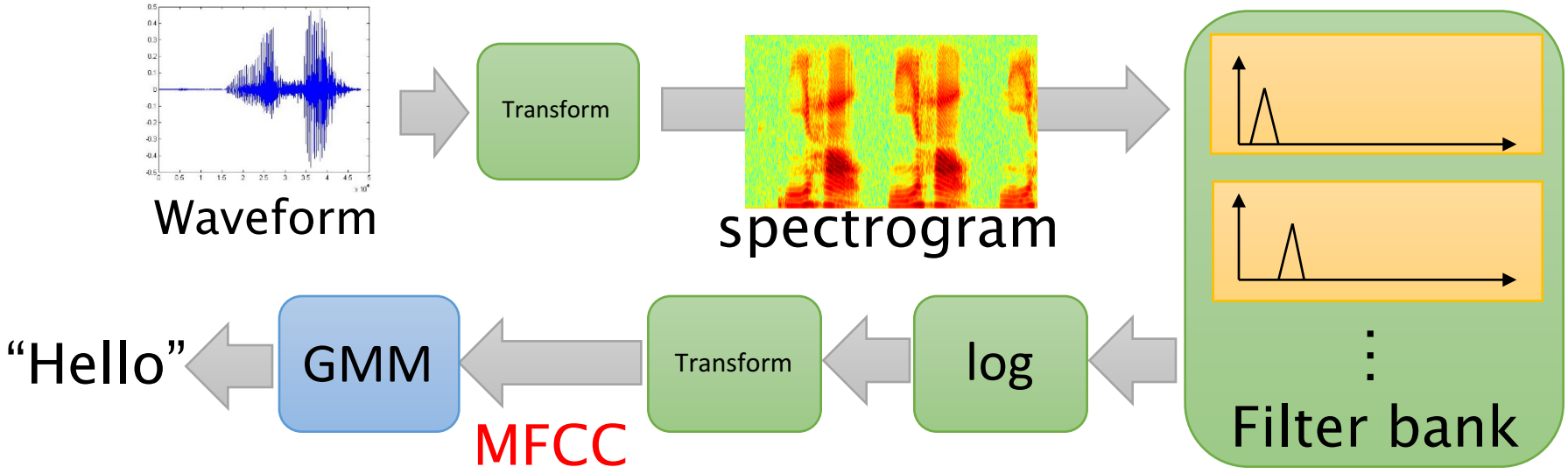



Representation Learning attempts to learn good features/representations

Deep Learning attempts to learn (multiple levels of) representations and an output

# Deep v.s. Shallow – Speech Recognition

- Shallow Model

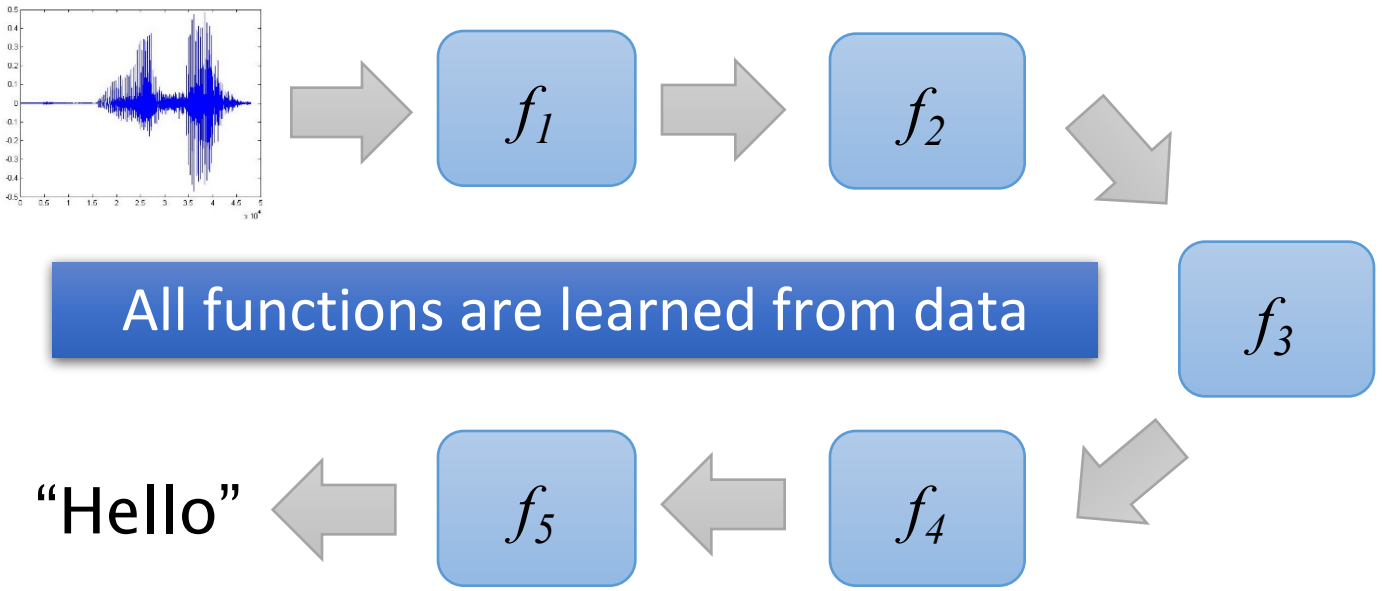



Each box is a simple function in the production line:

:hand-crafted :learned from data

# Deep v.s. Shallow – Speech Recognition

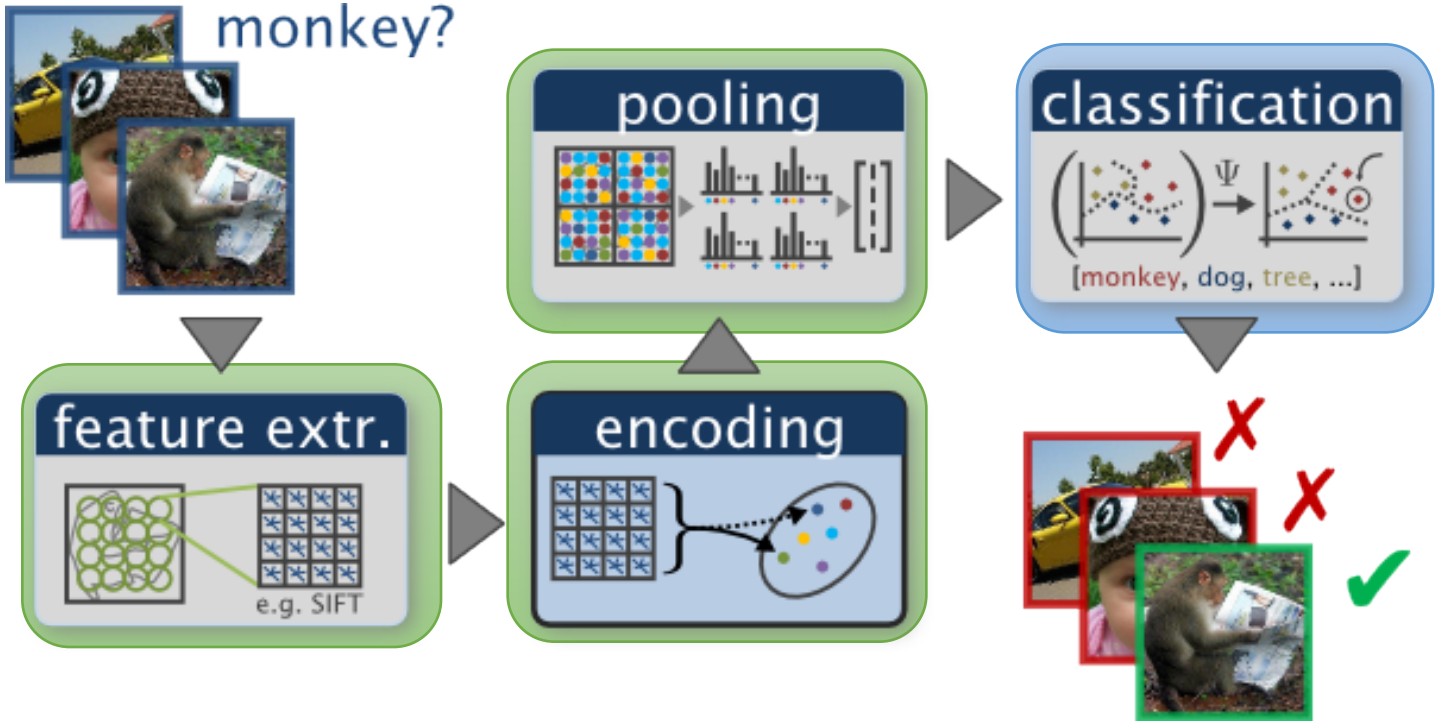
- Deep Model




Feature representation is learned

# Deep v.s. Shallow – Image Recognition

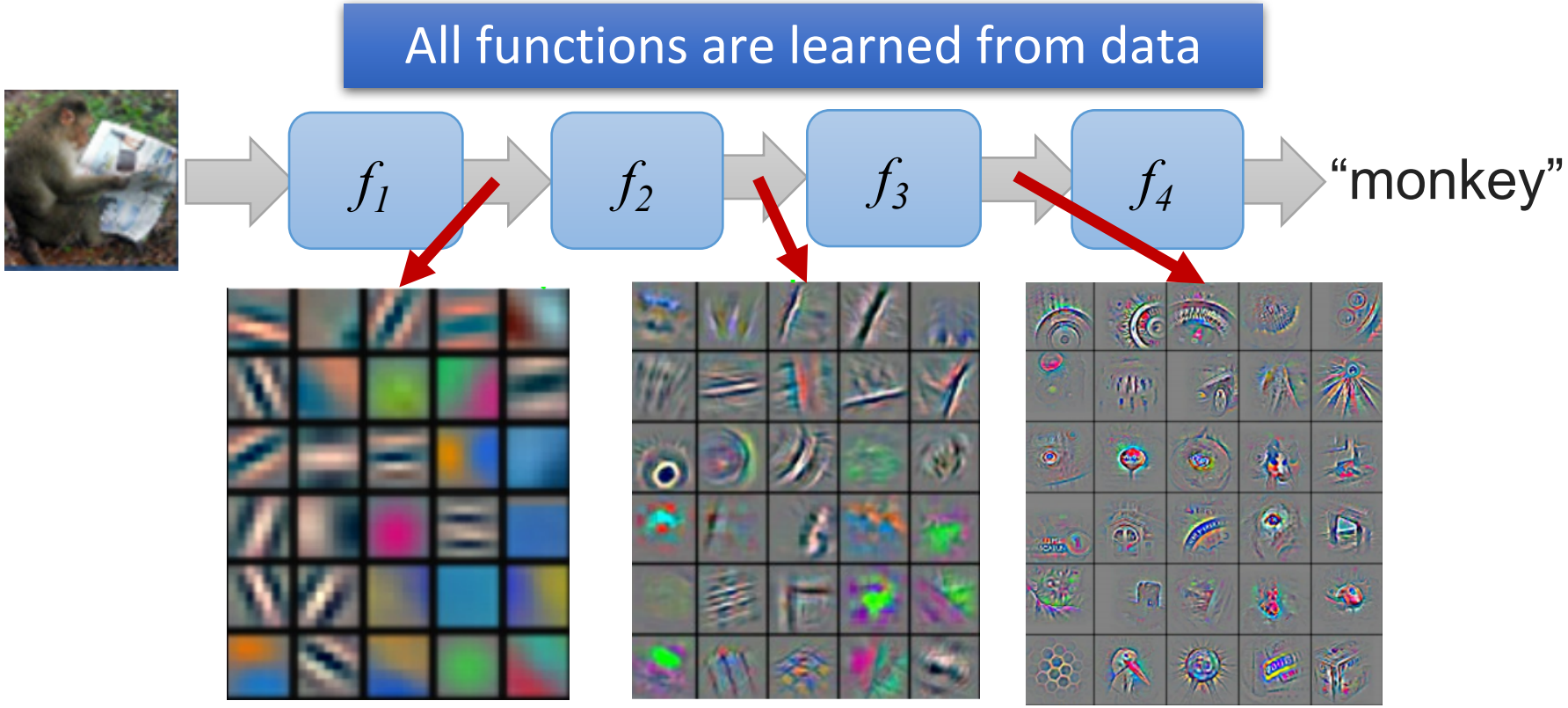
- Shallow Model



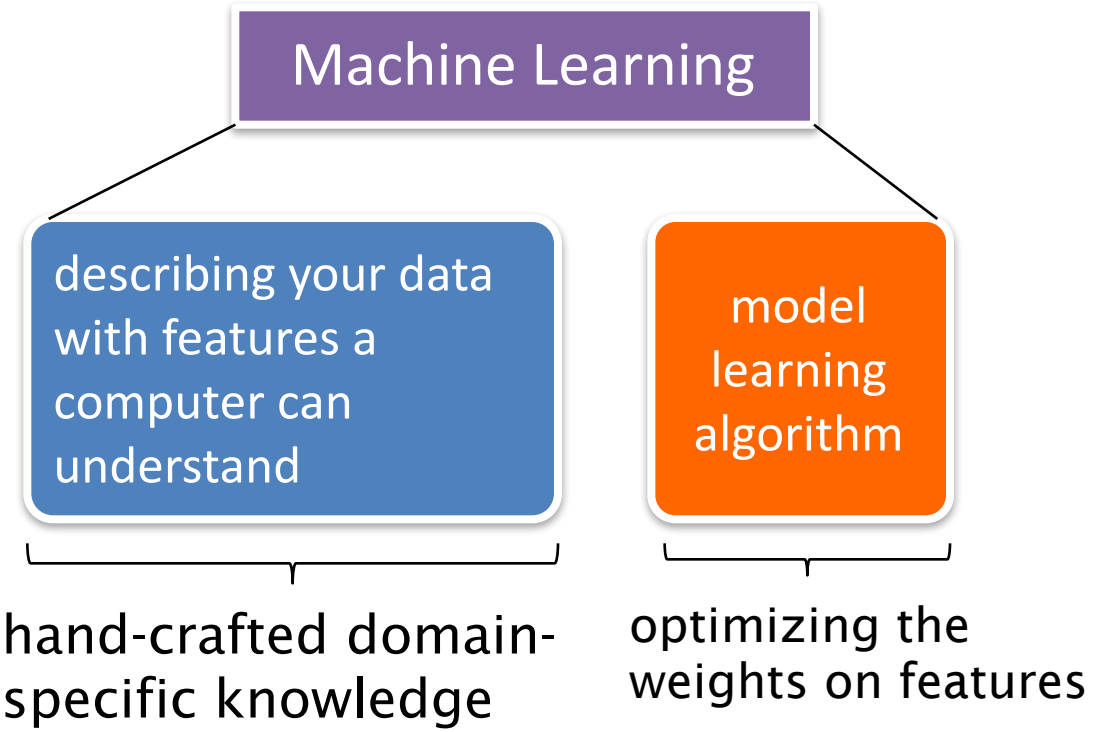

:hand-crafted :learned from data

# Deep v.s. Shallow – Image Recognition

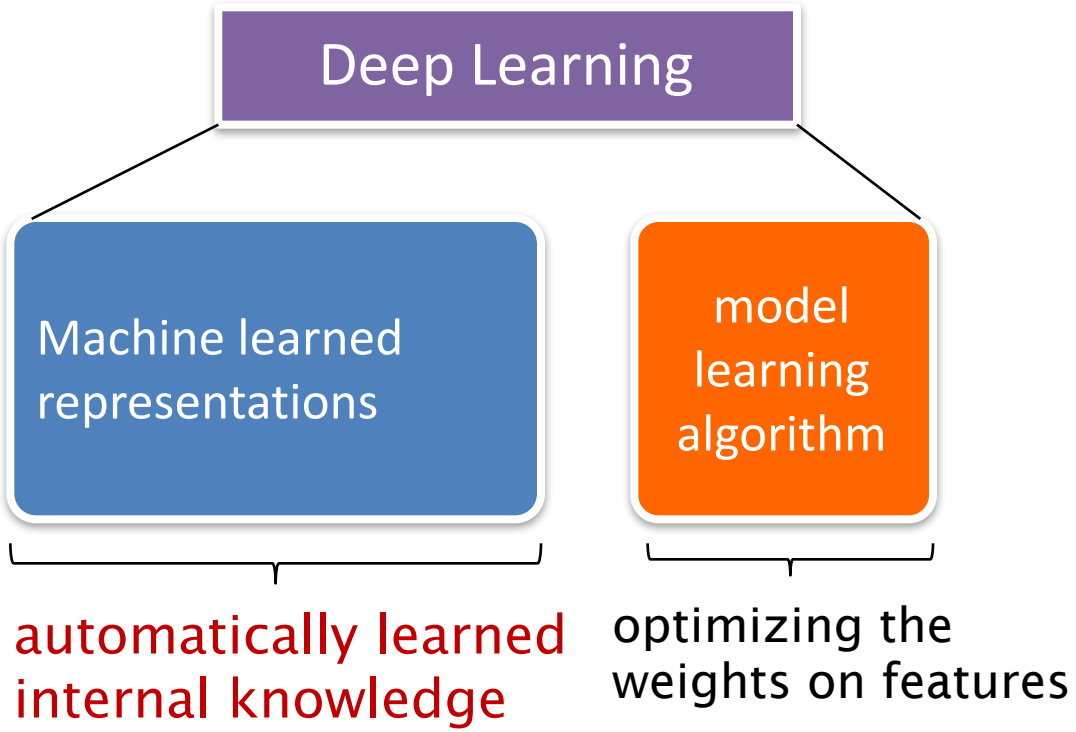
- Deep Model



# Classic Machine Learning v.s. Deep Learning

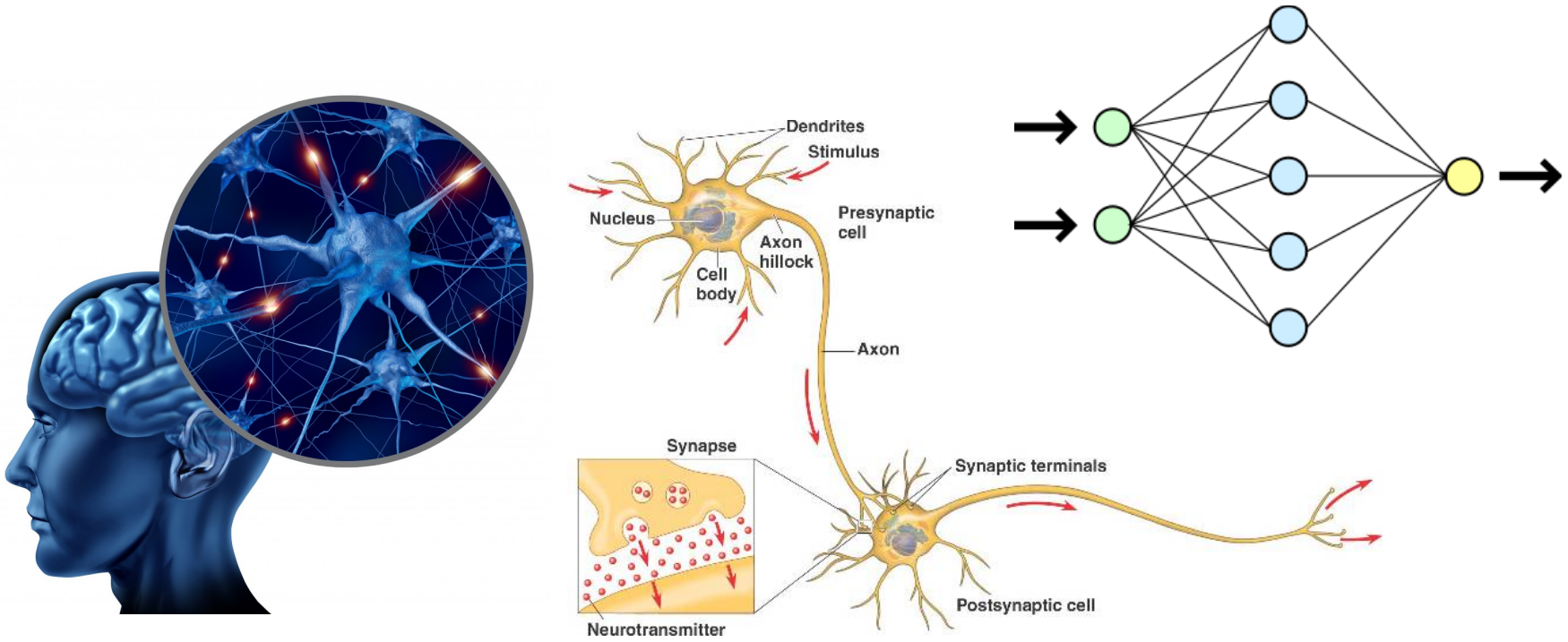


# Classic Machine Learning v.s. Deep Learning

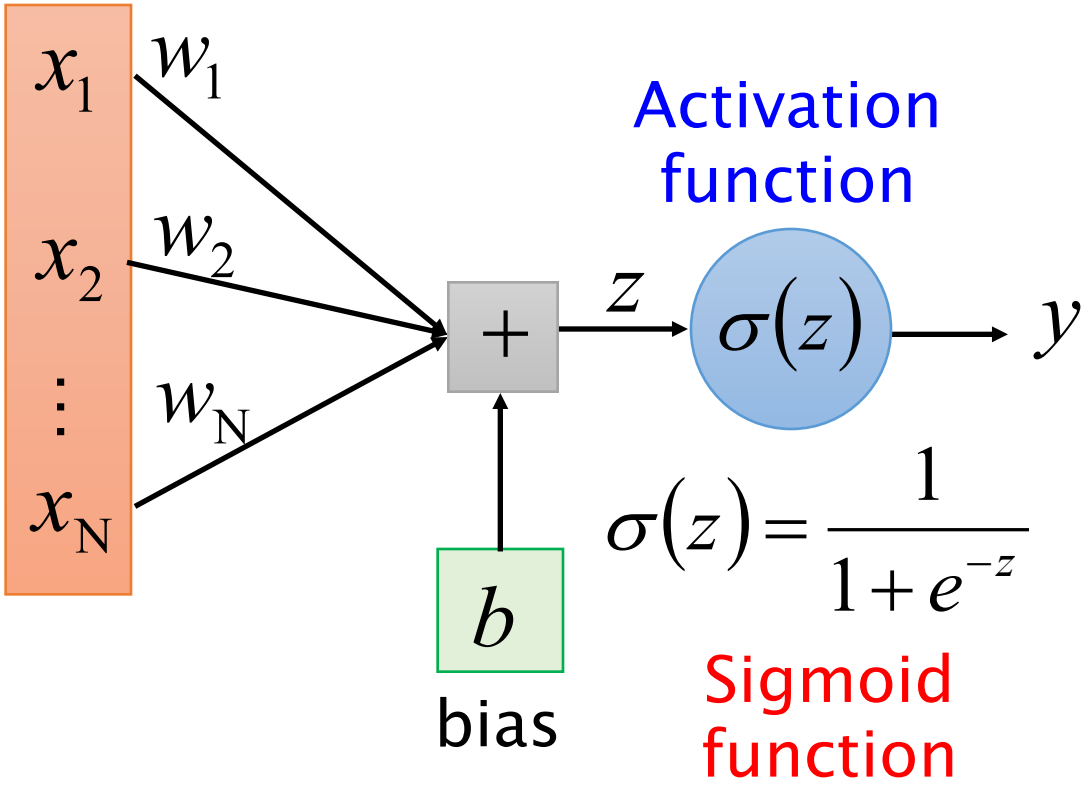


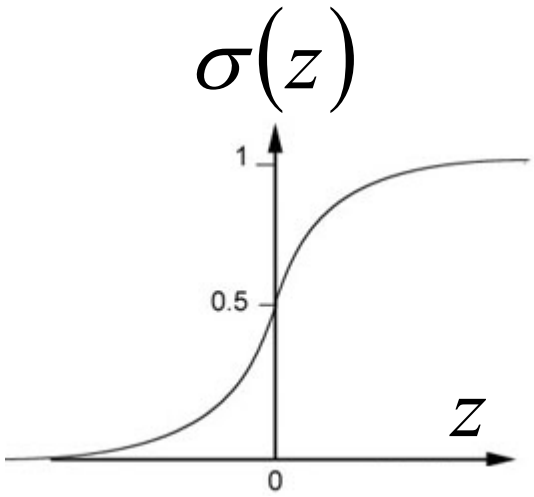

Deep learning usually refers to *neural network* based model

# Inspired by Human Brain



# A Single Neuron: Logistic Regression



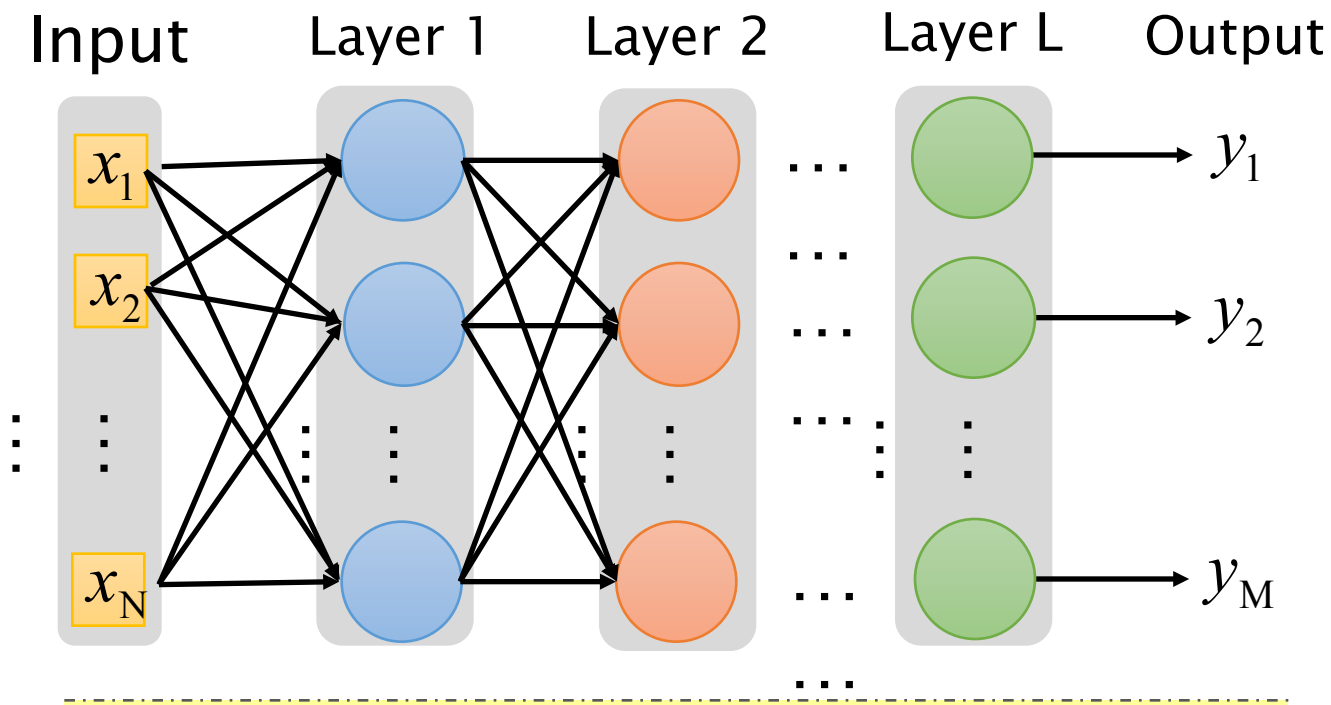

Each neuron is a very simple function

# Deep Neural Network

A neural network is a complex function:

$$f : R^N \rightarrow R^M$$

- Cascading the neurons to form a neural network




Each layer is a simple function in the production line

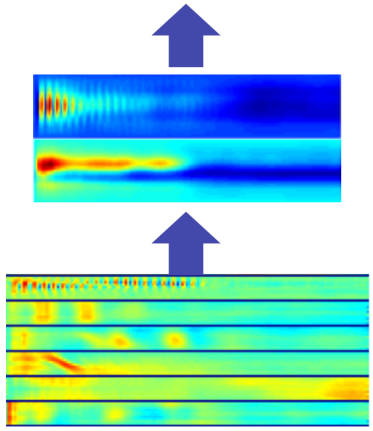
# History of Deep Learning

- 1960s: Perceptron (single layer neural network)
- 1969: Perceptron has limitation
- 1980s: Multi-layer perceptron
- 1986: Backpropagation
- 1989: 1 hidden layer is "good enough", why deep?
- 2006: RBM initialization (breakthrough)
- 2009: GPU
- 2010: breakthrough in Speech Recognition (Dahl et al., 2010)
- 2012: breakthrough in ImageNet (Krizhevsky et al. 2012)
- 2015: "superhuman" results in Image and Speech Recognition

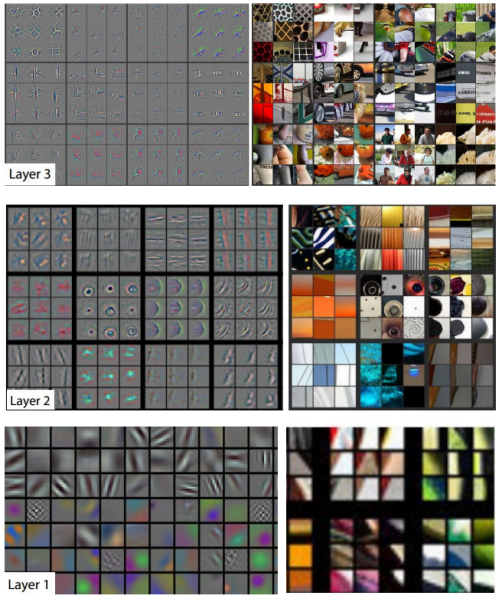
# Deep Learning Breakthrough

- Speech Recognition

| Acoustic Model | WER on RT03S FSH | WER on Hub5 SWB |
|---|---|---|
| Traditional Features | 27.4% | 23.6% |
| Deep Learning | 18.5% (-33%) | 16.1% (-32%) |




- Computer Vision

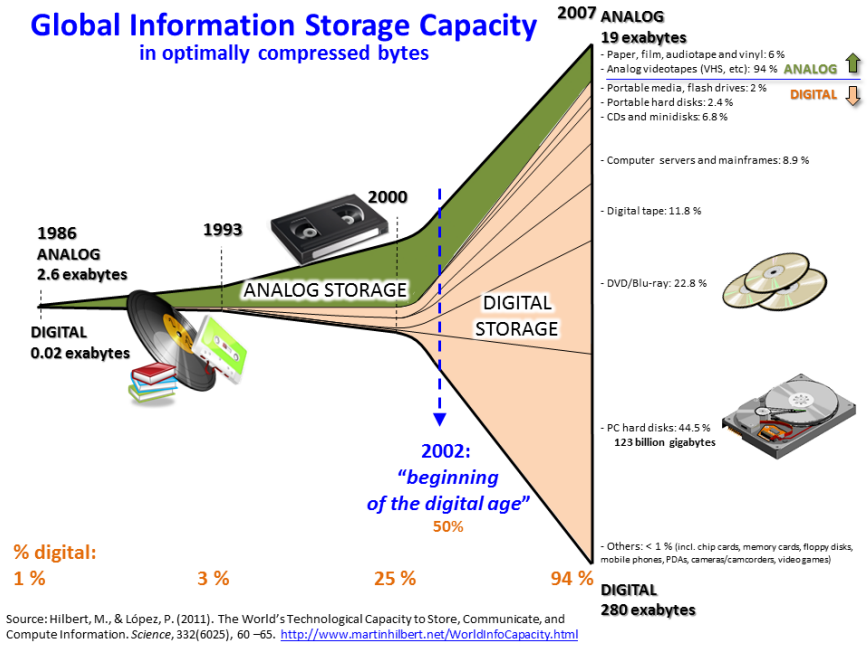
# History of Deep Learning

- 1960s: Perceptron (single layer neural network)
- 1969: Perceptron has limitation
- 1980s: Multi-layer perceptron
- 1986: Backpropagation
- 1989: 1 hidden layer is "good enough", why deep?
- 2006: RBM initialization (breakthrough)
- 2009: GPU
- 2010: breakthrough in Speech Recognition (Dahl et al., 2010)
- 2012: breakthrough in ImageNet (Krizhevsky et al. 2012)
- 2015: "superhuman" results in Image and Speech Recognition

Why does deep learning show breakthrough in applications after 2010?

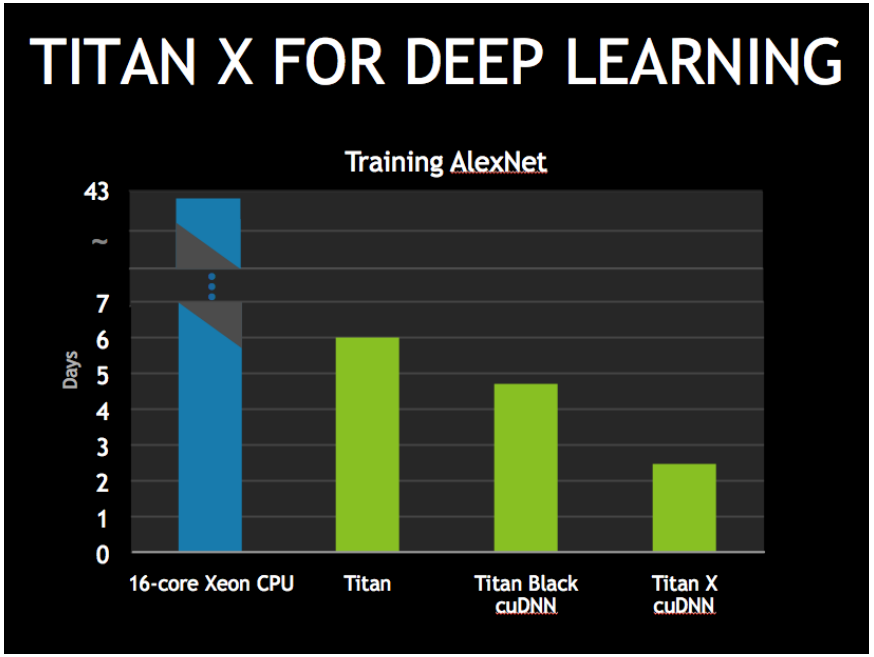
# Reasons why Deep Learning works

- Big Data




Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

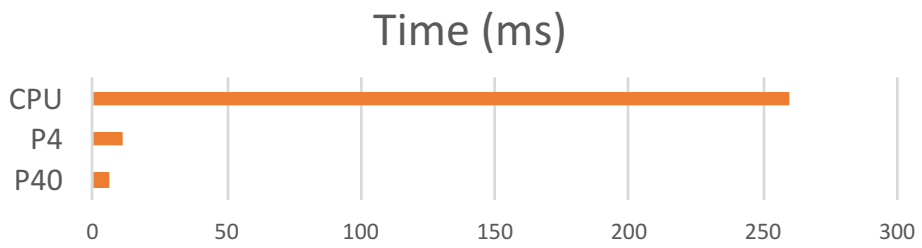
- GPU



# Why Speed Matters?

- Training time
  - Big data increases the training time
  - Long training time is not practical

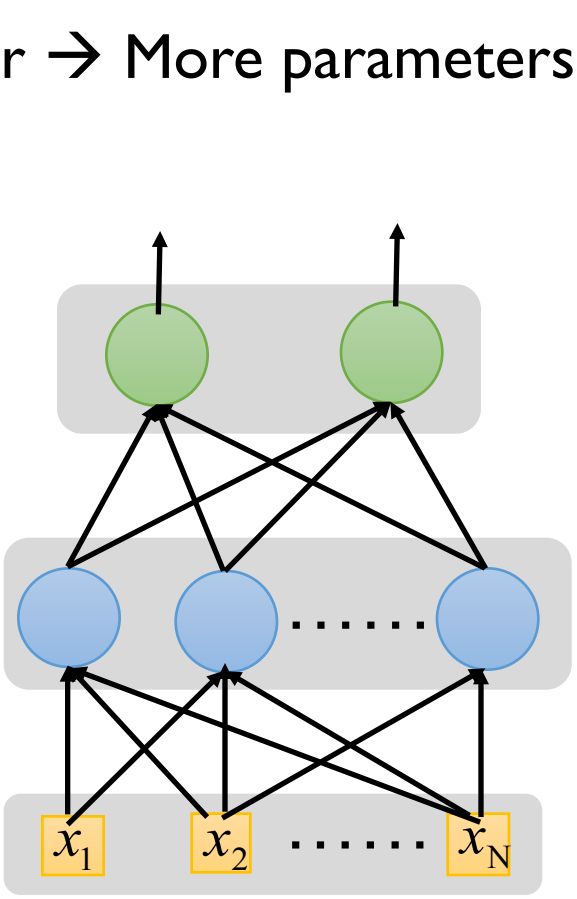
- Inference time
  - Users are not patient to wait for the responses



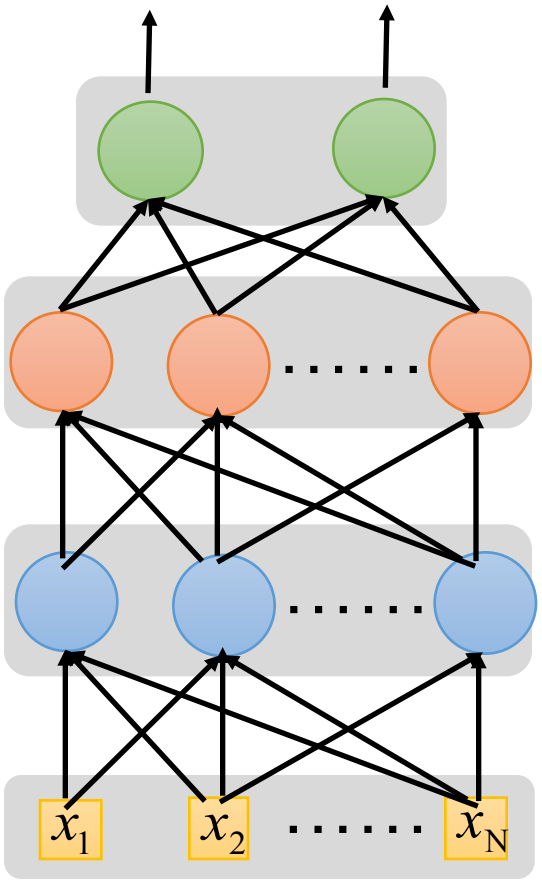

GPU enables the real-world applications using the computational power

# Why Deeper is Better?

- Deeper → More parameters



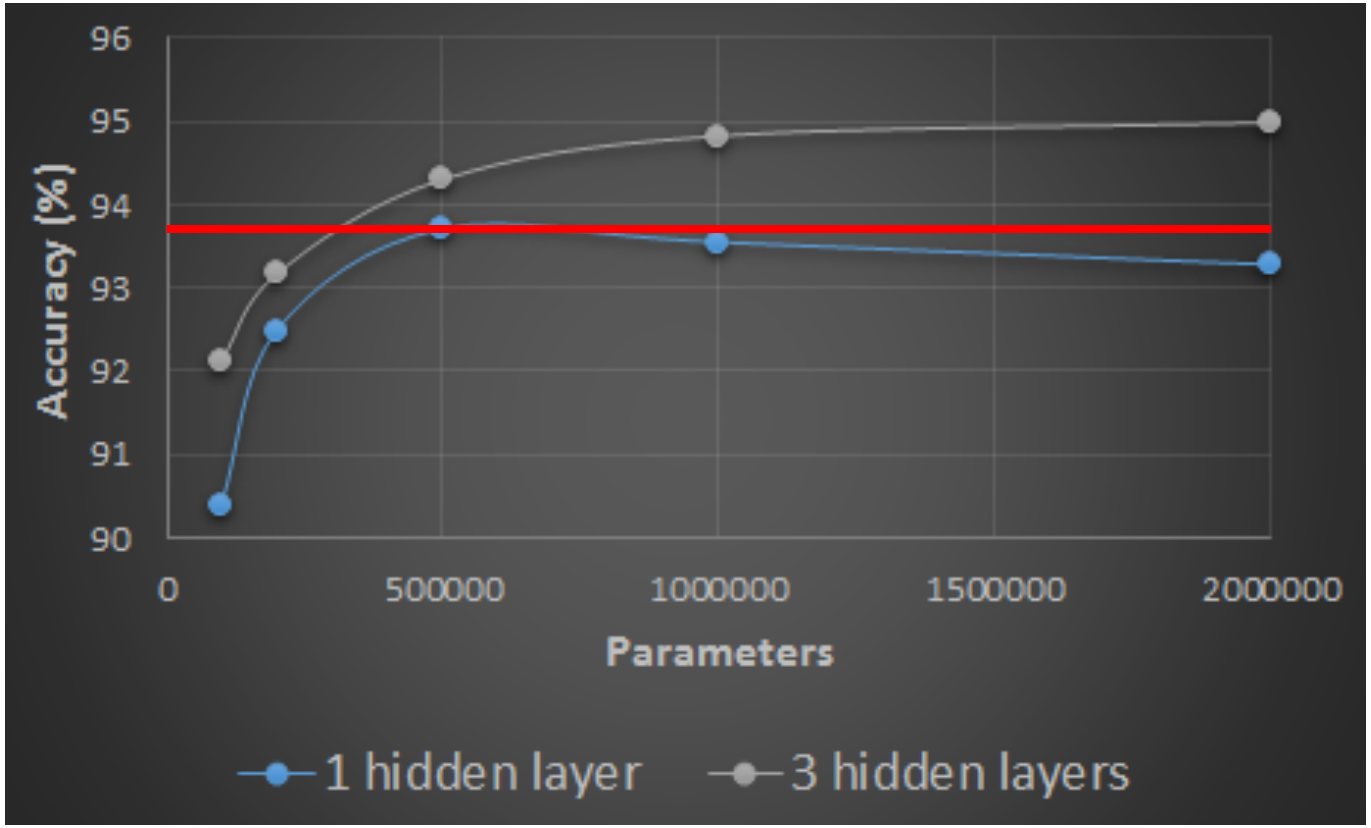

Shallow




Deep

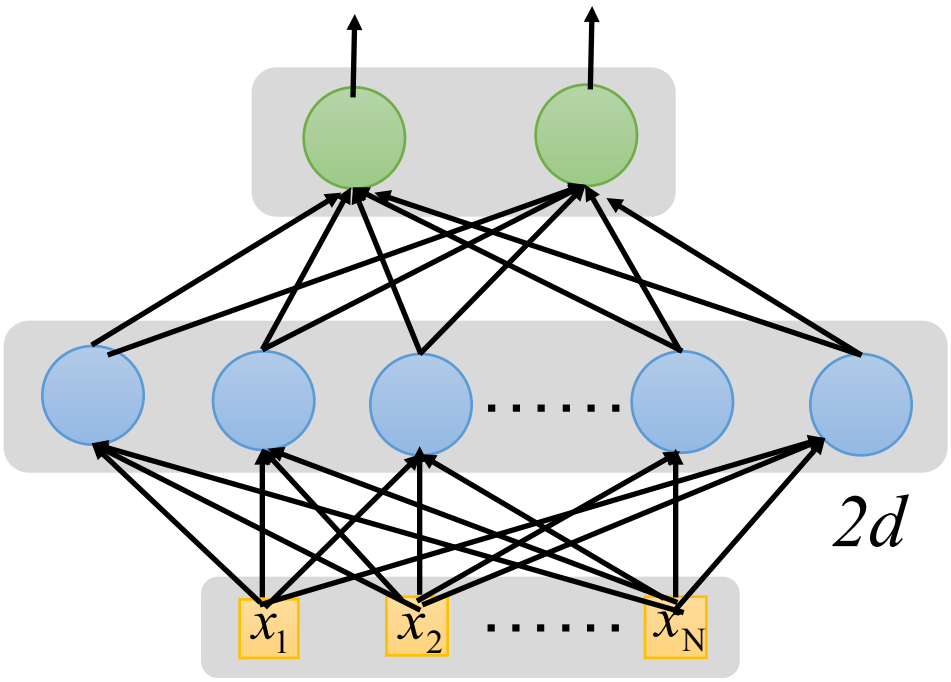
# Wide + Shallow v.s. Thin + Deep

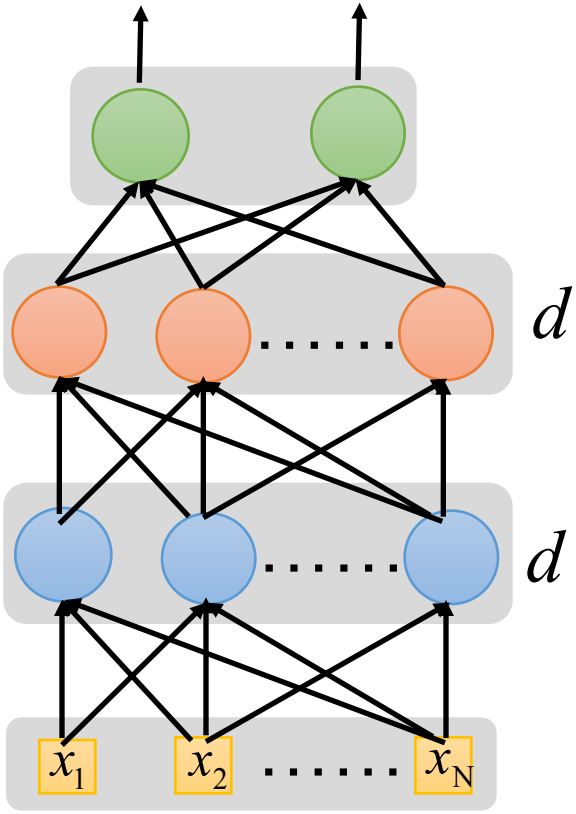
## Hand-Written Digit Classification



# Wide + Shallow v.s. Thin + Deep

- Two networks with the same number of parameters



# HOW TO FRAME THE LEARNING PROBLEM?

# How to Frame the Learning Problem?

- The learning algorithm $f$ is to map the input domain $X$ into the output domain $Y$

$$f : X \rightarrow Y$$

- Input domain: word, word sequence, audio signal, click logs
- Output domain: single label, sequence tags, tree structure, probability distribution

# Output Domain – Classification

- Sentiment Analysis

"This camera is amazing" → +

"This sofa is too pricey." → -

- Speech Phoneme Recognition

 → /h/

- Handwritten Recognition

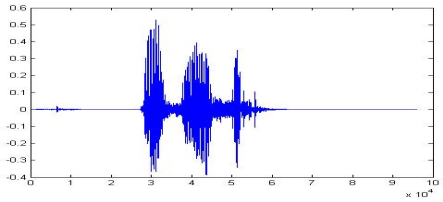
 → 2

# Output Domain – Sequence Prediction

- POS Tagging

"I saw her duck." → I/PRP saw/VBD her/PRP$ duck/NN./.

- Speech Recognition



→ "That silly boy."

- Machine Translation

"How are you doing today?" → "你好吗?"

Learning tasks are decided by the output domains

# Input Domain –
## How to Aggregate Information

- Input: word sequence, image pixels, audio signal, click logs
- Property: continuity, temporal, importance distribution
- Example
  - CNN (convolutional neural network): local connections, shared weights, pooling
    - AlexNet, VGGNet, etc.
  - RNN (recurrent neural network): temporal information

Network architectures should consider the input domain properties

# How to Frame the Learning Problem?

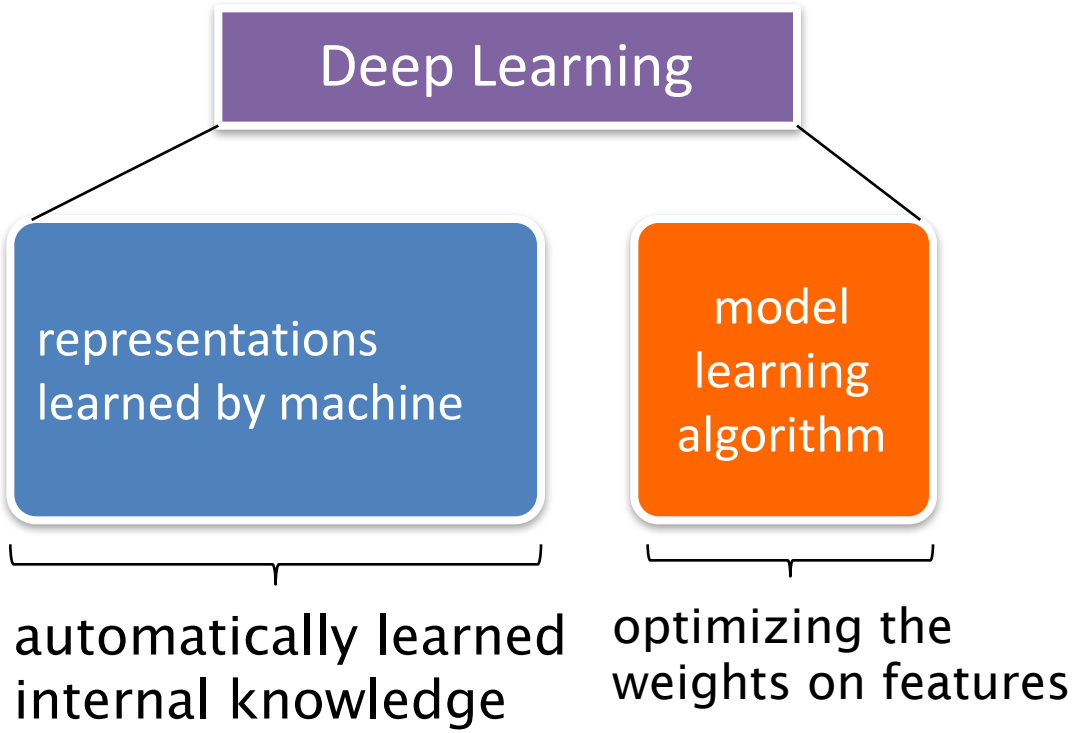
- The learning algorithm $f$ is to map the input domain $X$ into the output domain $Y$

$$f : X \rightarrow Y$$

- Input domain: word, word sequence, audio signal, click logs
- Output domain: single label, sequence tags, tree structure, probability distribution

Network design should leverage input and output domain properties

# Deep Learning for NLP



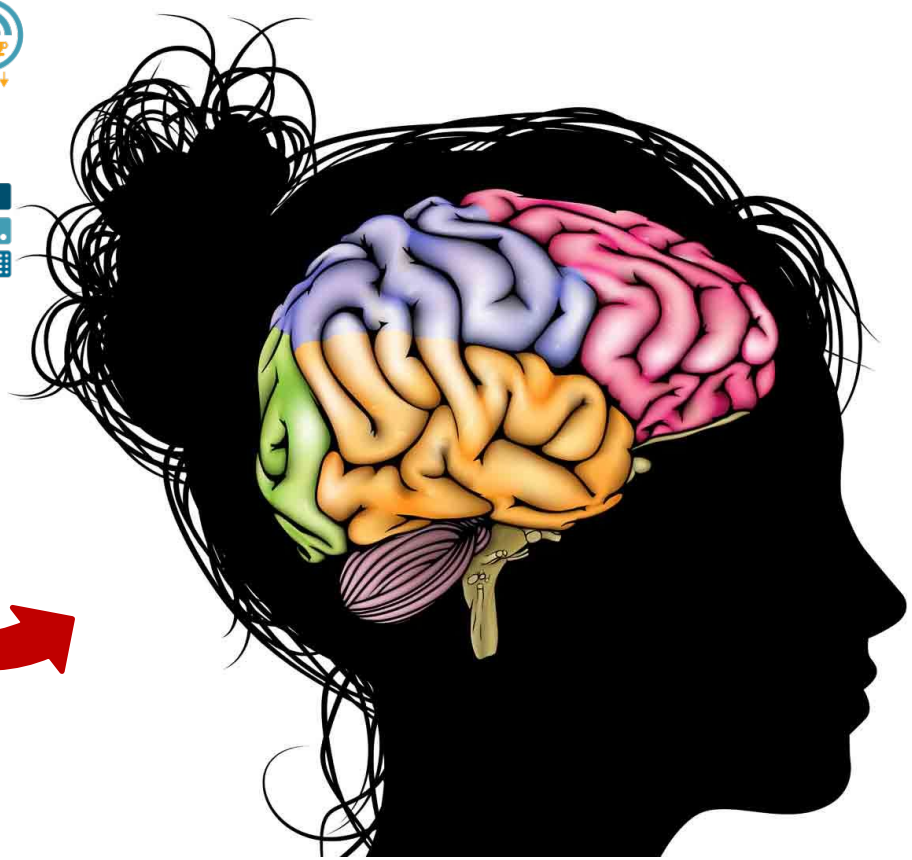

How to frame a task into a learning problem and design the corresponding model

# Core Factors of Deep Learning for NLP

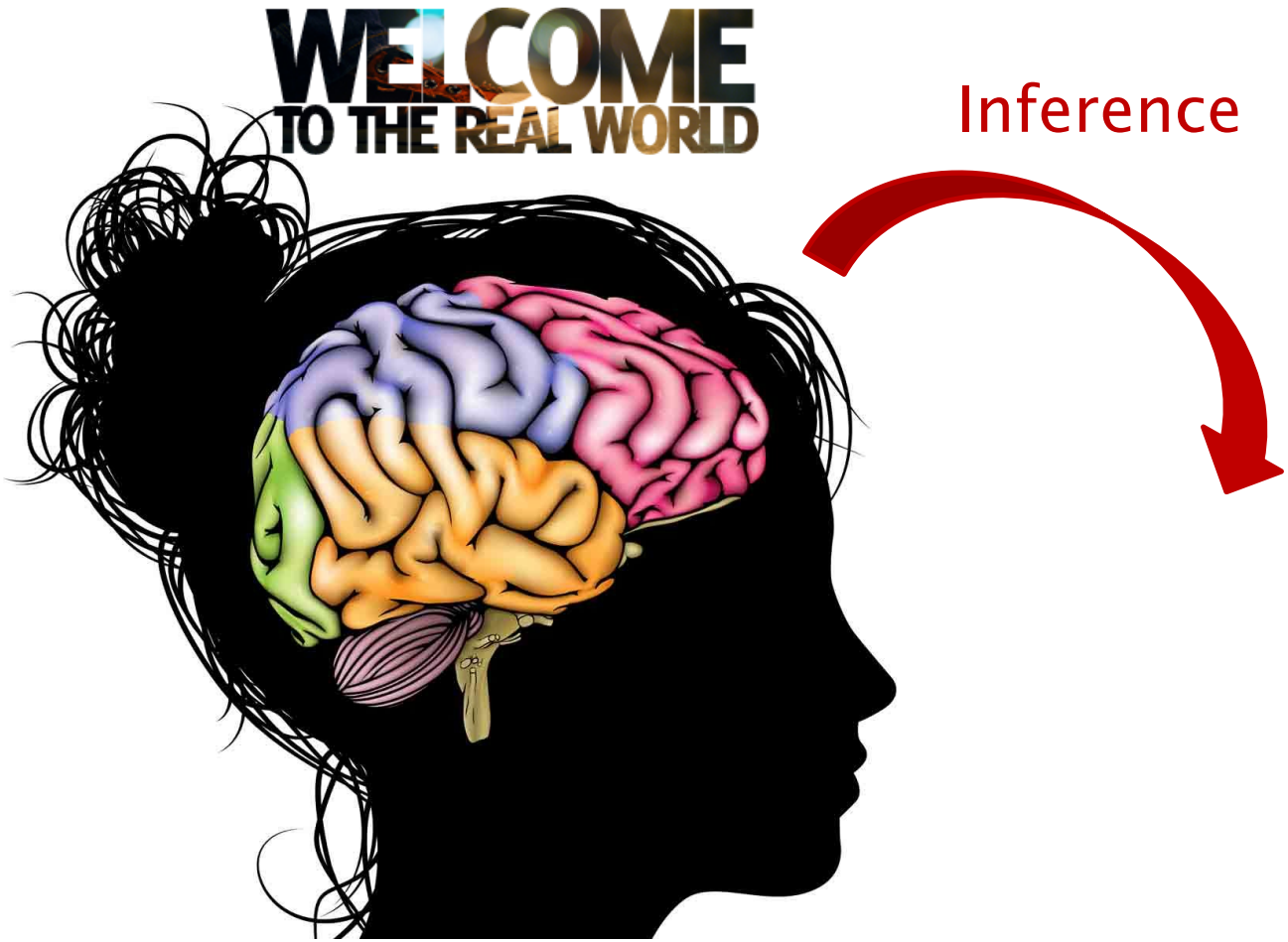
1. Data: big data
2. Hardware: GPU computing
3. Problem solving: design algorithms to allow networks to work for the specific problems.

# Training and Inference

Training

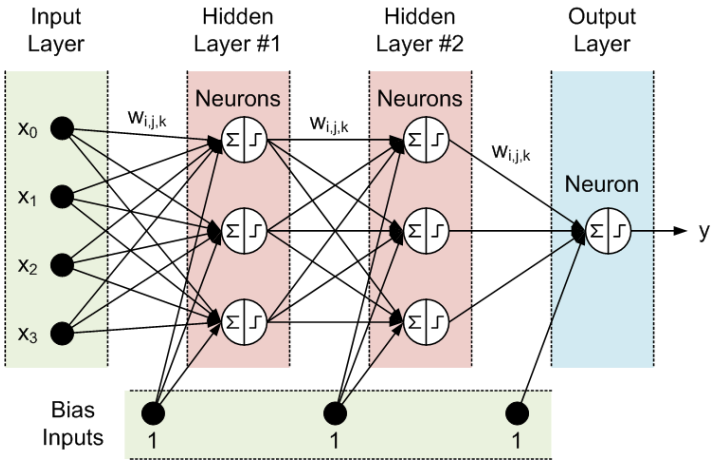


# Training and Inference



# Training and Inference


Inference




Training

Main focus: how to design deep learning algorithms for real-world NLP problems

# Reference

- Reading Materials
  - Academic papers will be updated on the website
- Deep Learning
  - Goodfellow, Bengio, and Courville, "Deep Learning," 2016. http://www.deeplearningbook.org
  - Michael Nielsen, "Neural Networks and Deep Learning" http://neuralnetworksanddeeplearning.com

# Course logistics

- Instructor: William Wang
- Previous taught:
  - Intro. to NLP (UCSB 1901W17)
  - Deep Learning for NLP (UCSB 292F S17)
  - Machine Learning (CMU)
  - Machine Learning for Large Datasets (CMU)
  - Information Extraction, ACL Summer School 2015
  - NAACL+IJCAI Tutorials 2016
- Published 40+ papers in top NLP/AI conferences.

# Course logistics

- Instructor: William Wang
- Time: T R 1:00pm-2:50pm
- Location: PHELPS 2510
- Reader: Ke Ni, ke00@ucsb.edu
  - Proposal + Reports
- Office hour: Tu 3-4pm HFH 1115 starting next week.
- http://william.cs.ucsb.edu/courses

# Course logistics

- Prerequisites:
  - 130A or equivalent data structure course;
  - 165B machine learning;
  - comfortable with deep learning platforms / tools e.g., TensorFlow, (Py)Torch, Keras, MXNet, Theano, Caffe.
  - solid background in machine learning, linear algebra, probability, and calculus;
  - prior experience with AWS or Google Cloud could be helpful.

# Course Objectives

- At the end of the quarter, students should know
  1. how, when, why DL works;
  2. how to frame NLP tasks into DL problems;
  3. how to implement DL algorithms using popular platforms for research;
  4. how to present cutting-edge DL papers;
  5. how to conduct novel DL4NLP research;
  6. how to write a scientific report / conference paper of DL4NLP.

# Text Book

- No official text book is required.
- Here's a recommended one if you ask:
  - Deep Learning, An MIT Press book, Ian Goodfellow and Yoshua Bengio and Aaron Courville.
  - The authors HTML version of the book: http://www.deeplearningbook.org/

# Grading Policy

- There will be two homework assignments (20%), project (65%), and a paper presentation (15%).
- HW: four late days are allowed with no penalty.
- After that 50% will be deducted if it is within 4 days after the due day, unless you have a note from the doctors' office.
- Homework assignment submissions that are five days late will receive zero credits.

# Course Project (65%)

- Two-person teams
- One-page project proposal due 01/30 (10%)
- On 02/20
  - Two-page mid-term report due.
- On 03/08, 03/15, 03/17
  - Final project presentations (8 slides max) + two-minute QA (15%)
- Final report 3-5 pages including references in ICML format (30%).

# Project Expectation

- Proposal:
  - novelty in {problem, task, approach, data & eval}.
  - concrete idea and plan with available dataset.
- Mid-term
  - finish the major implementation of your algorithm.
  - show preliminary results.
- Final report
  - A+: top conference quality (ICML/NIPS/ACL/EMNLP/NAACL)
  - A: standard quality equivalent of a conference paper at a reputable location (COLING/IJCNLP)
  - A-: could be a competitive submission to top conference.
  - B: significant weaknesses in one or more areas.

# Google Cloud Computing Resources

- Once your proposal is approved
  - You will be provided with gift cards if needed.
  - You need to learn how to set up your own DL environment on Google Cloud (tons of references online).
  - In general, you will be provided free cloud research credits up to $100 for reasonable usage.
  - Additionally, AWS also supports students with $100 cloud credits.

# AWS Computing Resources

- What's unreasonable:
  - **Forget to terminate EC2 instances;**
  - Use g2.8xlarge instances to run single-GPU program (most implementations are not parallelized);
  - Frequently transfer large datasets across regions (e.g. N. Cal ←→ VA);
  - Industrial scale experiments (3000 machines).

# Rules of Machine Learning: Best Practices for ML Engineering (Martin Zinkevich, Google)

- http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf

- A very good read.
- We will go over some of the related rules on Thursday.

# Paper Presentation (15%)

- Each of the registered student will be assigned with a DL paper to present in class. (10%)
  - 12 minutes max presentation
  - 12 slides max
  - leave 3 mins for QA.
- Each student will also become **the discussant** of other papers in the same lecture. (5%)
  - prepare 2 questions for each paper and ask.

# Academic Integrity

- We follow UCSB's academic integrity policy from UCSB Campus Regulations, Chapter VII:``Student Conduct and Discipline"). Details see class website.
- In three words: DON'T DO IT!
- If you are not sure, ask the teaching staff.
- Violators will receive an F and will be reported to the Dean of Students Office at UCSB.

# Academic Integrity (cont'd)

- You may discuss course materials and assignments with your classmate, but **you cannot write anything down**.
- Each assignment solution must start by answering the following questions:
- (1) Did you receive any help whatsoever from anyone in solving this assignment? Yes / No.
  - If you answered 'yes', give full details: (e.g. ``Jane explained to me what is asked in Question 3.4")
- (2) Did you give any help whatsoever to anyone in solving this assignment? Yes / No.
  - If you answered 'yes', give full details: (e.g. ``I pointed Joe to section 2.3 to help him with Question 2".

# Active Learning

- It's a graduate course! Ask questions.
- Bring your laptop to every class
- We will do in-class group sessions ("active learning")

# Why the flipped classroom

- Attention span: everyone spaces out during long lectures
  - Middendorf and Kalish, 1995, Johnstone and Percival 1976, Burns 1985
- "the class started 1:00. The student sitting in front of me took copious notes until 1:20. Then he just nodded off… motionless, with eyes shut for about a minute and a half, pen still poised. Then he awoke and continued his rapid note-taking as if he hadn't missed a beat."
- Student remembered only the first 15-20 minutes

# Why the flipped classroom (2)

- **Active learning**: Be in charge of your learning
  - Obviously most important: programming assignments
  - Active learning ("constructivism"), learning by doing
- **Collaborative learning**: Learn from each other
  - Use class time for group activities, worked problems
  - "Small group active learning"

# Piazza

- **CS 291A Piazza**:
  - http://www.piazza.com/ucsb/winter2018/cs291a
- Students are encouraged to answer each other's questions.

- Teaching staff will choose Best Writer(s) with 10% extra credit rewards at the end of the quarter.

# Coming up next class (Thursday)
## Project advice and in-class brainstorming

- Projects:
  - How to propose a novel research project.
  - What to avoid.
  - Project ideas and open-research problems for DL4NLP.
- Group brainstorming:
  - Teaming.
  - Everyone: please bring a piece of paper.
  - I will discuss with you about your initial idea.