Lecture 1: Introduction

Xuming He
SIST, ShanghaiTech
Fall, 2019

Outline

- Introduction
  - What & Why deep learning?
- Course logistics
  - Overall philosophy
  - Grading policy
  - Pre-requisite / Syllabus
- Background review

Acknowledgement: Bhiksha Raj@CMU's course notes

Introduction

- Our goal: Build intelligent algorithms to make sense of data
  - Example: Recognizing objects in images

red panda (*Ailurus fulgens*)

  - Example: Predicting what would happen next

Vondrick et al. CVPR2016

Introduction

- A broad range of real-world applications
  - Speech recognition
    - Input: sound wave → Output: transcript
  - Language translation
    - Input: text in language A (Eng) → Output: text in language B (Chs)
  - Image classification
    - Input: images → Output: image category (cat, dog, car, house, etc.)
  - Autonomous driving
    - Input: sensory inputs → Output: actions (straight, left, right, stop, etc.)

- Main challenges: difficult to manually design the algorithms

A data-driven approach

- Each task as a mapping function (or a model)

Mapping function

Input data → Expected output

  - input data: images
  - expected output: object or action names

- Building such mapping functions from data

red panda (*Ailurus fulgens*)

→ Mapping function

A data-driven approach

- Building a mapping function (model)

$$y = f(x;\theta)$$

  - x: input data
  - y: expected output
  - θ: parameters to be estimated

- Learning the model from data
  - Given a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$
  - Find the 'best' parameter $\hat{\theta}$, such that

$$y_n \simeq f(x_n;\hat{\theta}) \quad \forall n$$

  - And it can be generalized to unseen input data

What is deep learning?

- Using deep neural networks as the mapping function

- Deep neural networks
  - A family of parametric models
  - Consisting of many ‘simple’ computational units
  - Constructing a multi-layer representation of input

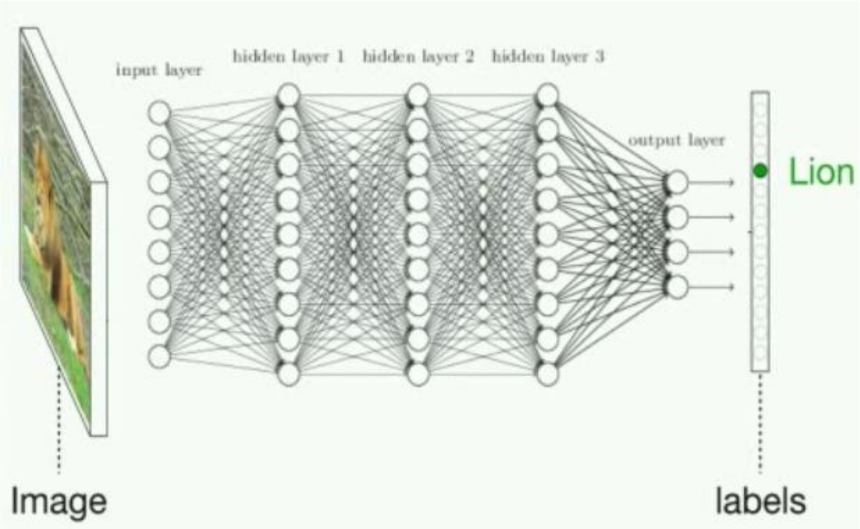
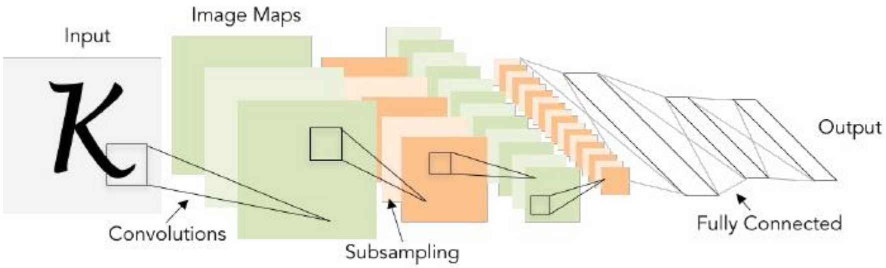


Image from Jeff Clune’s Deep Learning Overview

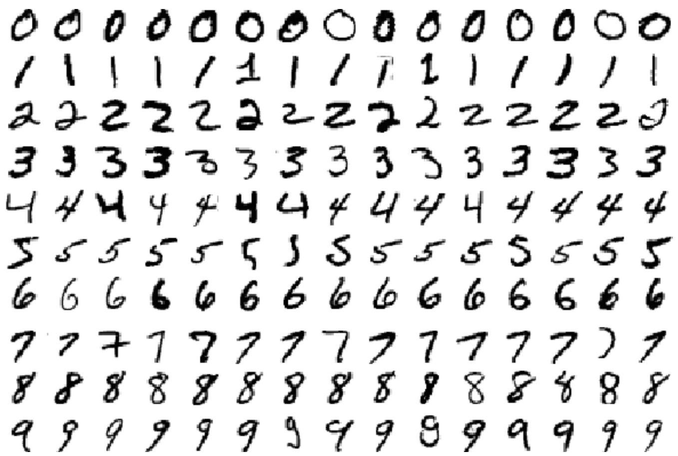
What is deep learning?

- Using deep neural networks as the mapping function

- Parameter estimation from data
  - Parameters: connection weights between units
  - Formulated as an optimization problem
  - Efficient algorithms for handling large-scale models & datasets

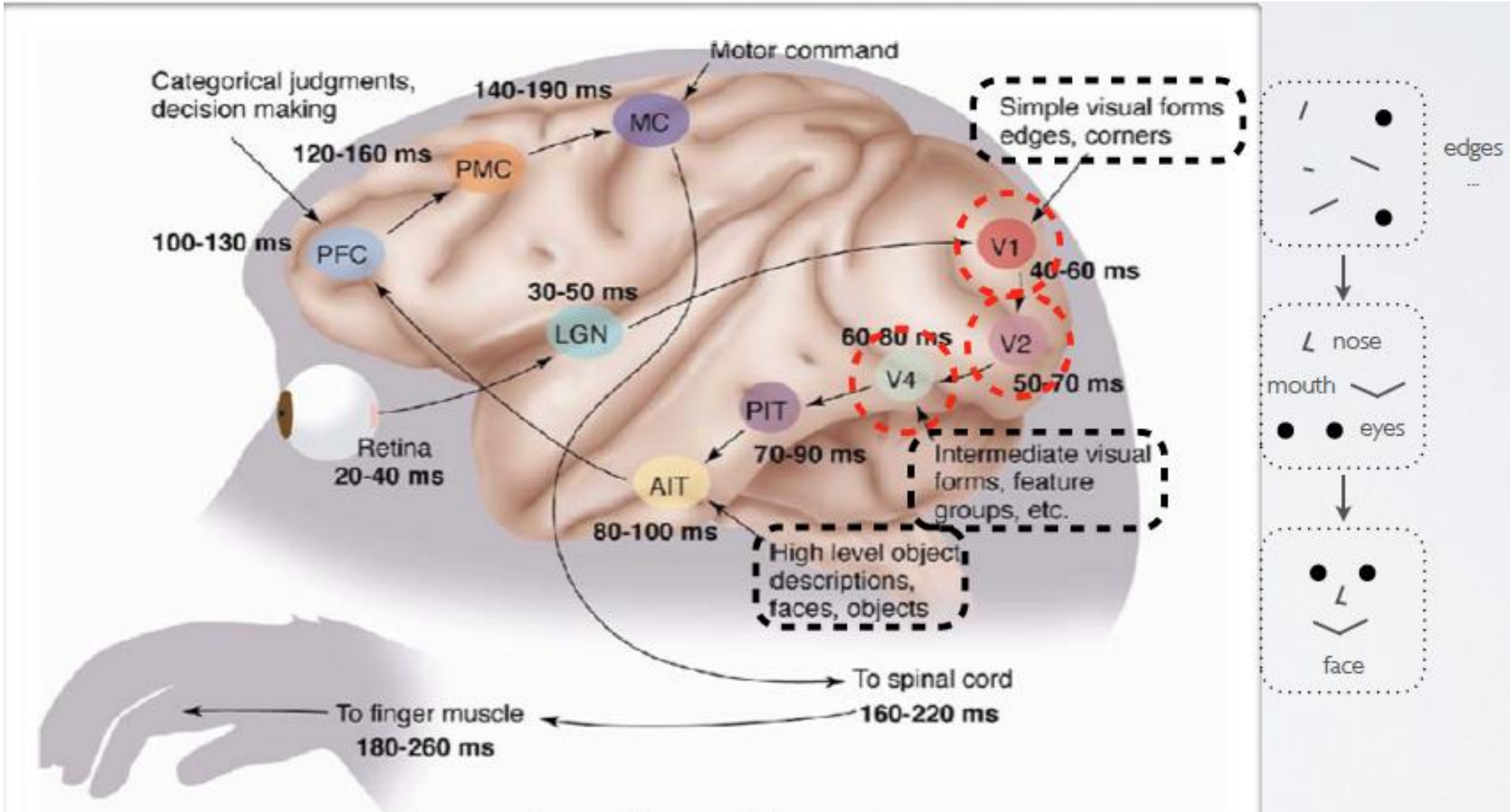


LeNet-5



Why deep networks?

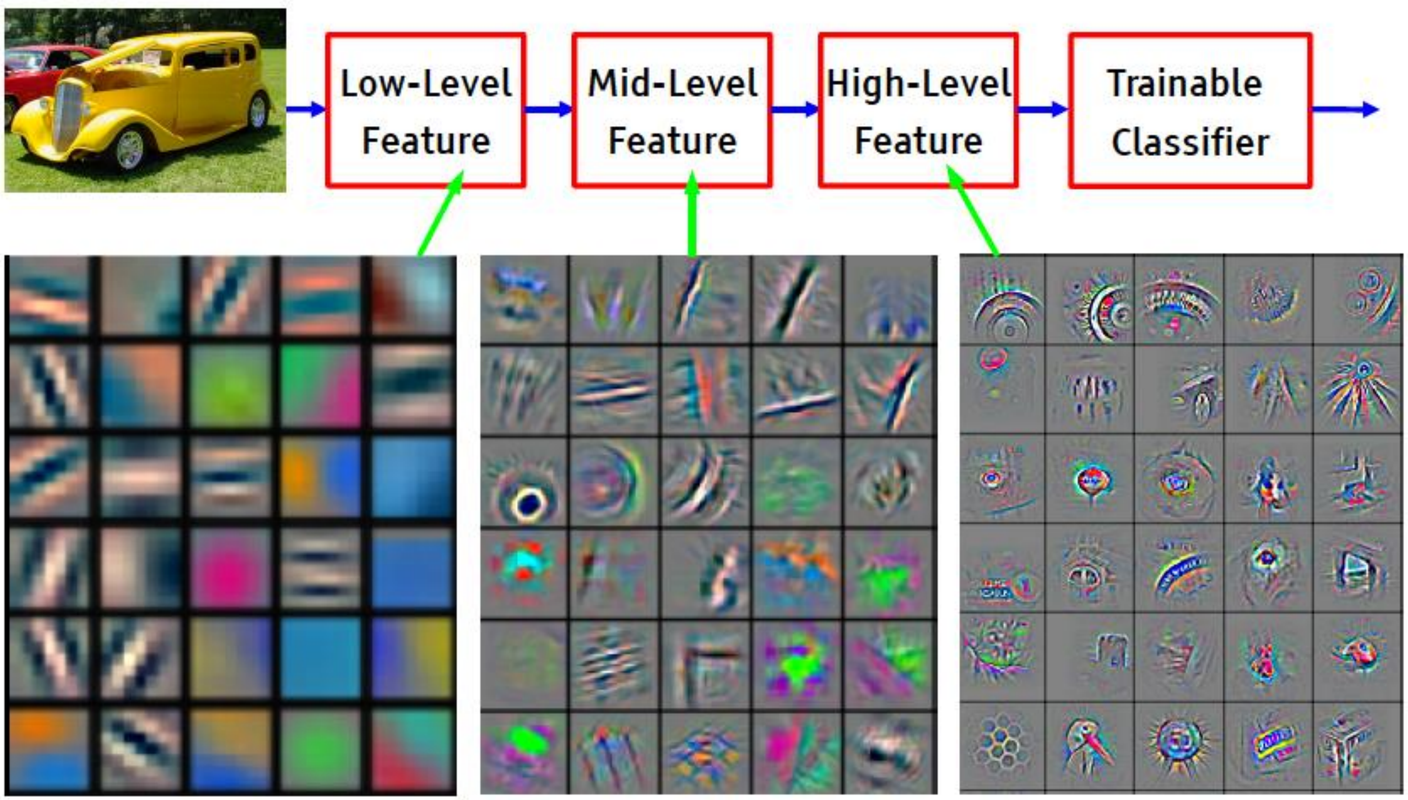
- Inspiration from visual cortex



[picture from Simon Thorpe]

Why deep networks?

- A deep architecture can represent certain functions (exponentially) more compactly
- Learning a rich representation of input data



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

Recent success with DL

- Some recent success with neural networks

TECHNEWSWORLD EMERGING TECH

Computing Internet IT Mobile Tech Reviews Security Technology Tech Blog Reader Services

Microsoft AI Beats Humans at Speech Recognition

By Richard Adhikari
Oct 20, 2016 11:40 AM PT

Microsoft's Artificial Intelligence and Research Unit earlier this week reported that its speech recognition technology had surpassed the performance of human transcriptionists.

Most Popular Newsletters News Alerts

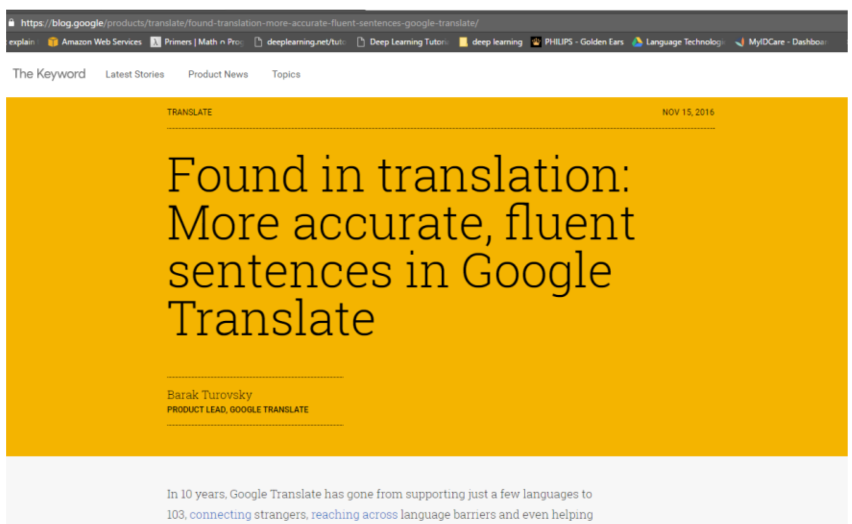
How do you feel about Black Friday and Cyber Monday?

- They're great -- I get a lot of bargains!
- The deals are too spread out -- I'd prefer just one day.
- They're a fun way to kick off the holiday season.
- I don't like the commercialization of Thanksgiving Day.
- They're crucial for the retail industry and the economy.
- The deals typically aren't that good.

Vote to See Results

E-Commerce Times

- Black Friday Shoppers Hungry for New Experiences, New Tech
- Pay TV's Newest Innovation: Giving Users Control
- Apple Celebrates Itself in $300 Coffee Table Tome
- AWS Enjoys Top Perch in IaaS, PaaS Markets
- US Comptroller Gears Up for Blockchain and



The Keyword Latest Stories Product News Topics

TRANSLATE NOV 15, 2016

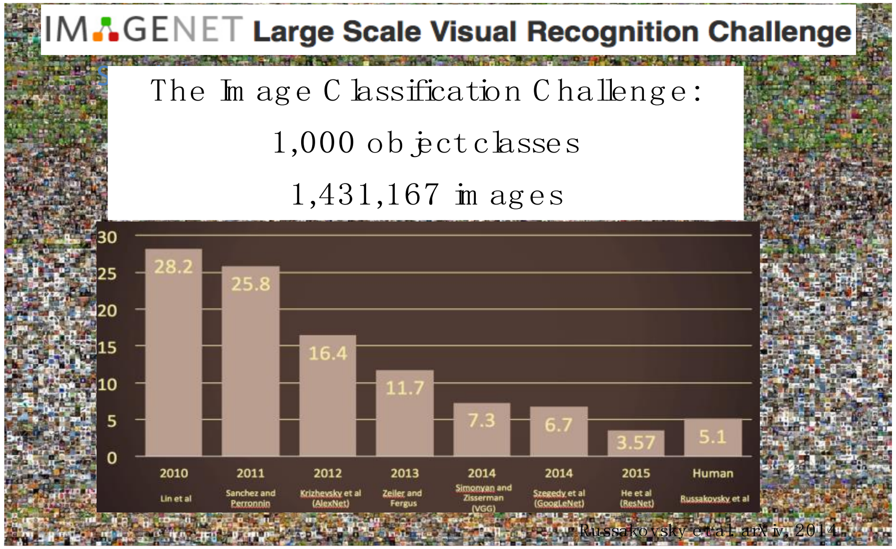
Found in translation: More accurate, fluent sentences in Google Translate

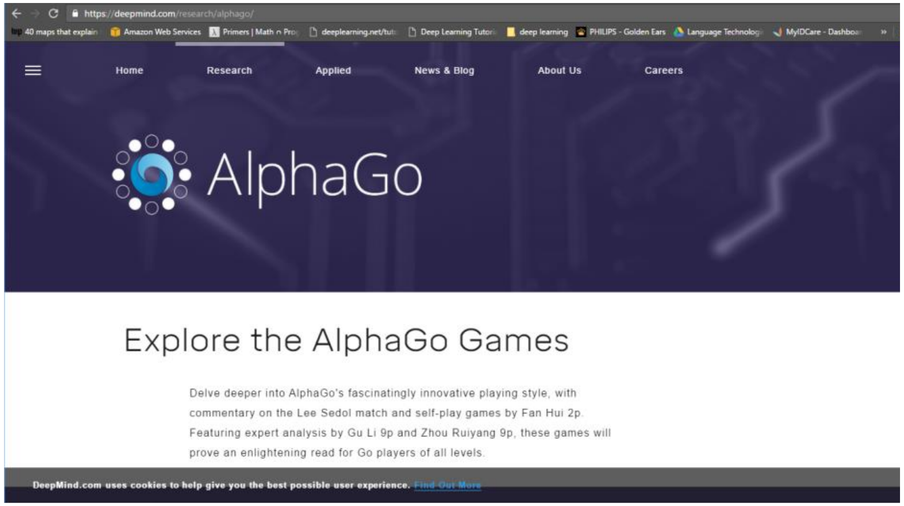
Barak Turovsky
PRODUCT LEAD, GOOGLE TRANSLATE

In 10 years, Google Translate has gone from supporting just a few languages to 103, connecting strangers, reaching across language barriers and even helping

Recent success with DL

- Some recent success with neural networks





Summary: Why deep learning?

- One of the major thrust areas recently in various pattern recognition, prediction and data analysis
  - Efficient representation of data and computation
  - Other key factors: large datasets and hardware

- The state of the art in many problems
  - Often exceeding previous benchmarks by large margins
  - Achieve better performances than human for certain "complex" tasks.

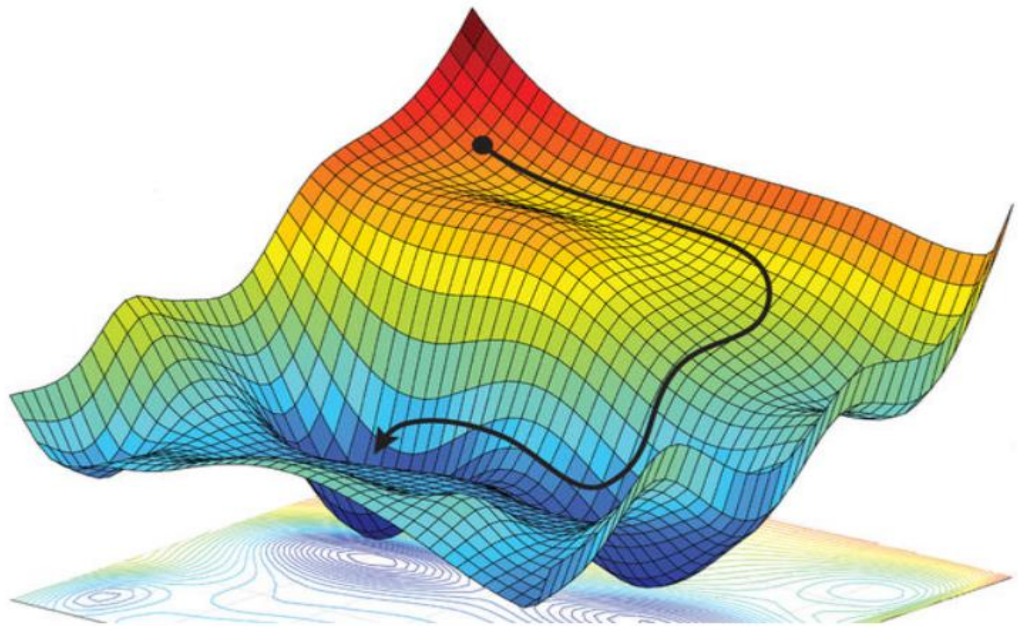
- But also somewhat controversial …
  - Lack of theoretical understanding
  - Sometimes difficult to make it work in practice

Is it alchemy?

Science Home News Journals Topics Careers

Log in | My account

SHARE
26K
1K



Gradient descent relies on trial and error to optimize an algorithm, aiming for minima in a 3D landscape.
ALEXANDER AMINI, DANIELA RUS. MASSACHUSETTS INSTITUTE OF TECHNOLOGY, ADAPTED BY M. ATAROD/*SCIENCE*

AI researchers allege that machine learning is alchemy

Questions behind the scene

- Return of neural networks
  - What is different this time?
  - How it works for specific problems?
  - Why get great performance?
- Future development
  - Its limitation and weakness?
  - The road to general-purpose AI?

Outline

- Introduction
- Course logistics
  - Overall philosophy
  - Grading policy
  - Pre-requisite / Syllabus
- Background review

Course objectives

- Learning to use deep networks
  - How to write from scratch, debug and train neural networks
  - Toolboxes commonly used in practice
- Understanding deep models
  - Key concepts and principles
- State of the art
  - Some new topics from research field
  - Focusing on vision-related problems

Syllabus & Schedule

- Piazza:
  - piazza.com/shanghaitech.edu.cn/fall2019/cs280
  - The schedule for the latter half of the semester may vary a bit
- Part I: Basic neural networks (1~1.5 weeks)
  - Linear models
  - Multiple layer networks
  - Gradient descent and BP
- Part II: Convolutional neural networks
- Part III: Recurrent neural networks/Deep RL
- Part IV: Generative neural networks

Syllabus & Schedule

- Piazza: https://piazza.com/shanghaitech.edu.cn/fall2017/cs280/home
  - The schedule for the latter half of the semester may vary a bit
- Part I: Basic neural networks
- Part II: Convolutional neural networks (4 weeks)
  - CNN basics
  - Understanding CNN
  - CNN in Vision
- Part III: Recurrent neural networks/Deep RL
- Part IV: Generative neural networks

Syllabus & Schedule

- Piazza: https://piazza.com/shanghaitech.edu.cn/fall2017/cs280/home
  - The schedule for the latter half of the semester may vary a bit
- Part I: Basic neural networks
- Part II: Convolutional neural networks
- Part III: Recurrent neural networks/Deep RL (4 weeks)
  - LSTM, GRU
  - Attention modeling
  - RNN in Vision/NLP
  - Deep reinforcement learning
- Part IV: Generative neural networks

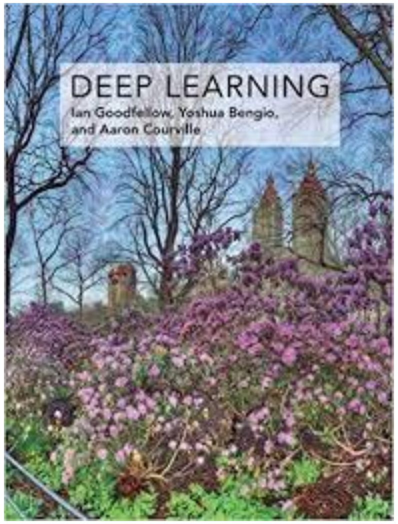
Syllabus & Schedule

- Piazza: https://piazza.com/shanghaitech.edu.cn/fall2017/cs280/home
  - The schedule for the latter half of the semester may vary a bit
- Part I: Basic neural networks
- Part II: Convolutional neural networks
- Part III: Recurrent neural networks/Deep RL
- Part IV: Generative neural networks (2.5~3 weeks)
  - Variational Auto Encoder (VAE)
  - Generative deep nets (GAN)
  - Sequential generative models

- Note: no lectures in the following weeks
  - Oct 28 ~ Nov 1 (ICCV)
  - Nov 18 ~ Nov 22 (Tentative)

Textbook and materials

- Deep learning:
  - http://www.deeplearningbook.org/
  - http://neuralnetworksanddeeplearning.com/



- Online deep learning courses:
  - Stanford: CS230, CS231n
  - CMU: 11-785
  - MIT: 6.S191

- Additional reading materials on Piazza
  - Survey papers, tutorials, etc.

Instructor and TAs

- Instructor: Me
  - hexm at shanghaitech
  - SIST 1A-304D

- TAs (tentative):
  - Rongjie Li
  - Shuailin Li
  - … will be finalized next week

- Office hours: on Piazza

- We will use Piazza as the communication platform

Grading policy

- 3 Problem sets: 15% x 3 = 45%
  - Write-up problem sets + Programming tasks
- Final course project: 35%(+10%)
  - Proposal: 5%
  - Final report (CVPR format): 25%
  - Presentation: 5%
  - Bonus points for novel results: 10%
- 10 Quizzes (in class): 2% x 10 = 20%
- Late policy
  - 25% off per day late
  - Not accepted after 3 late days
  - Does not apply to Final course project/Quizzes
- Collaboration policy
  - Project team: at most 3 students
  - Grading according to each member's contribution

Pre-requisite

- Proficiency in Python
  - All class assignments will be in Python (and use numpy)
  - A Python tutorial available on Piazza
- Calculus, Linear Algebra, Probability and Statistics
  - Undergrad course level
- Equivalent knowledge of Andrew Ng's CS229 (Machine Learning)
  - Formulating cost functions
  - Taking derivatives
  - Performing optimization with gradient descent
- Will be evaluated in next quiz (Thurday)

Outline

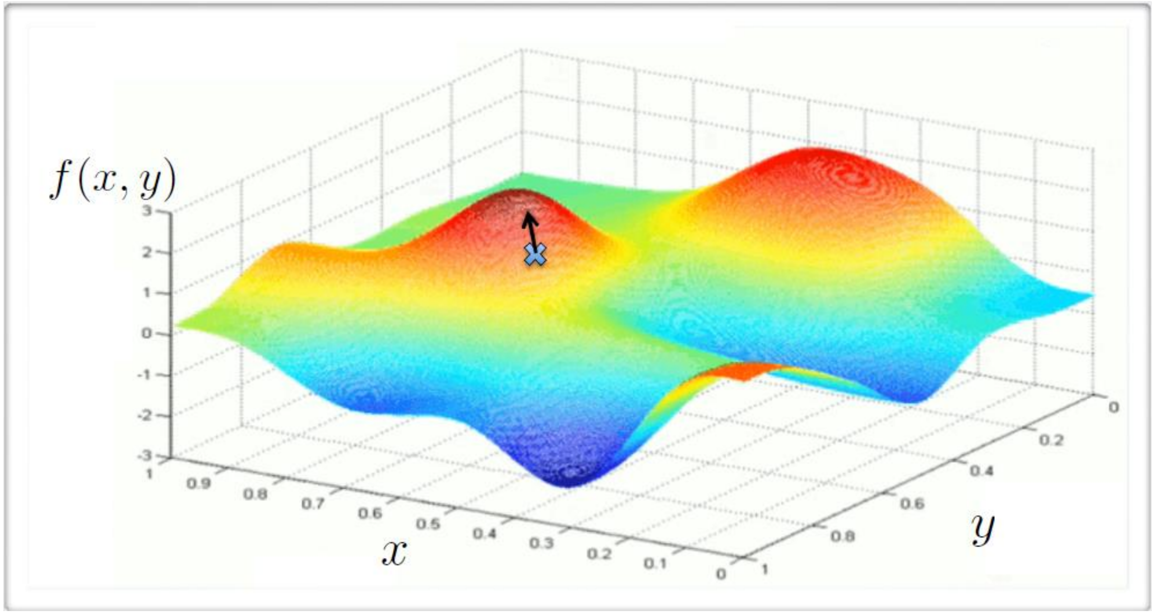
- Introduction
- Course logistics
  - Overall philosophy
  - Grading policy
  - Pre-requisite / Syllabus
- Background review
  - Machine learning basics

Acknowledgement: Hugo Larochelle's, Mehryar Mohri@NYU's & Yingyu Liang@Princeton's course notes

Math review – Calculus

- Gradient

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = [\frac{\partial}{\partial x_1} f(\mathbf{x}), \cdots, \frac{\partial}{\partial x_d} f(\mathbf{x})]^\intercal = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_d} f(\mathbf{x}) \end{bmatrix}$$



Math review – Calculus

- Hessian and Jacobian

• Hessian:

$$\nabla^2_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) & \dots & \frac{\partial^2}{\partial x_1 \partial x_d} f(\mathbf{x}) \\ \vdots & \dots & \vdots \\ \frac{\partial^2}{\partial x_d \partial x_1} f(\mathbf{x}) & \dots & \frac{\partial^2}{\partial x_d^2} f(\mathbf{X}) \end{bmatrix}$$

• If $\mathbf{f}(\mathbf{x}) = [f(\mathbf{x})_1, \dots, f(\mathbf{x})_k]^\top$ is a vector, the Jacobian is:

$$\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x})_1 & \dots & \frac{\partial}{\partial x_d} f(\mathbf{x})_1 \\ \vdots & \dots & \vdots \\ \frac{\partial}{\partial x_1} f(\mathbf{x})_k & \dots & \frac{\partial}{\partial x_d} f(\mathbf{x})_k \end{bmatrix}$$

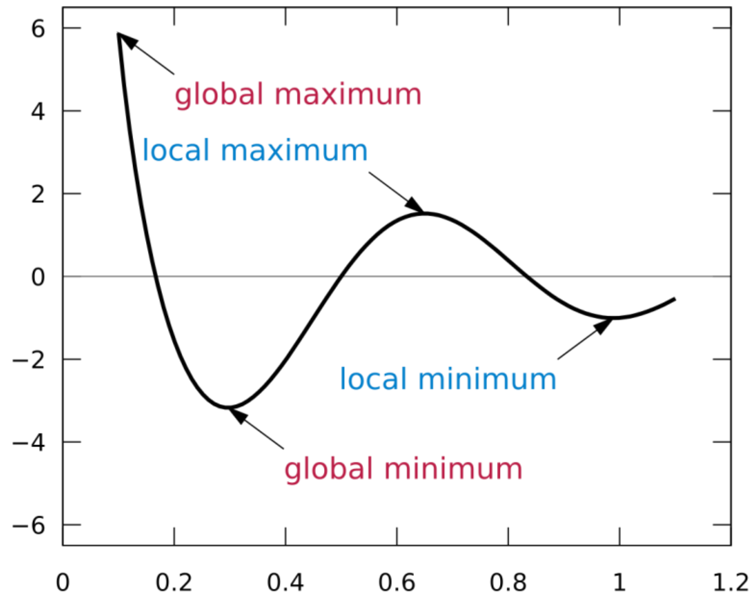
Math review – Calculus

- Local and global minima
  - Necessary condition

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = 0$$

  - Sufficient condition
    - Hessian is positive definite

$$f(\mathbf{x}) \approx f(\mathbf{x}^*) + (\mathbf{x} - \mathbf{x}^*)^\intercal \nabla_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\intercal \nabla_{\mathbf{x}}^2 f(\mathbf{x})(\mathbf{x} - \mathbf{x}^*)$$



Math review – Probability

- Factorization

  - Probability chain rule: $p(s,o) = p(s|o)p(o) = p(o|s)p(s)$
    - in general:

$$p(\mathbf{x}) = \prod_i p(x_i | x_1, \ldots, x_{i-1})$$

  - Bayes rule:

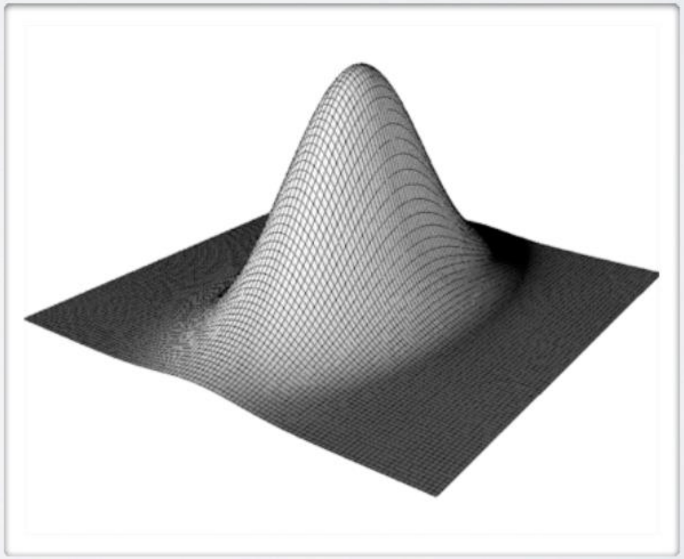
$$p(O=o|S=s) = \frac{p(S{=}s|O{=}o)p(O{=}o)}{\sum_{o'} p(S{=}s|O{=}o')p(O{=}o')}$$

Math review – Probability

- Common distributions

• Gaussian variable: $\mathbf{X} \in \mathbb{R}^d$

- $p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$
- $\mathrm{E}[\mathbf{X}] = \boldsymbol{\mu}$
- $\mathrm{Cov}[\mathbf{X}] = \Sigma$



Math review – Statistics

- Monte Carlo estimation
  - a method to approximate an expensive expectation

$$\mathrm{E}[f(\mathbf{X})] = \sum_{\mathbf{x}} f(\mathbf{x})\, p(\mathbf{x}) \approx \frac{1}{K} \sum_k f(\mathbf{x}^{(k)})$$

  - the $\mathbf{x}^{(k)}$ must be sampled from $p(\mathbf{x})$

- Maximum likelihood

$$\widehat{\theta} = \arg\max_{\theta} p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)})$$

  - Independent and identically distributed

$$p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}) = \prod_t p(\mathbf{x}^{(t)})$$

ML tasks

- Classification: assign a category to each item (e.g., document classification)
- Regression: predict a real value for each item (e.g., prediction of stock values, economic variables)
- Ranking: order items according to some criterion (e.g., relevant web pages returned by a search engine)
- Clustering: partition data into ‘homogenous’ regions (e.g., analysis of very large data sets)
- Dimensionality reduction: find lower-dimensional manifold preserving some properties of the data

Example 1: image classification

Task: determine if the image is indoor or outdoor
Performance measure: probability of misclassification

Example2: clustering images

Task: partition the images into 2 groups

Performance: similarities within groups

Data: a set of images

Standard learning scenarios

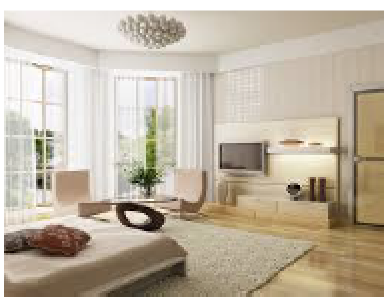
- Unsupervised learning: no labeled data
- Supervised learning: uses labeled data for prediction on unseen points
- Semi-supervised learning: uses labeled and unlabeled data for prediction on unseen points
- Reinforcement learning: uses reward to learn prediction on action policies.
- …

Supervised learning

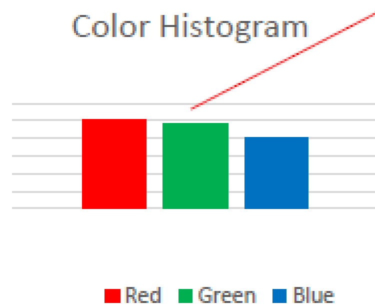
- Task formulation

  - Learning example: $(\mathbf{x}, y)$
  - Task to solve: predict target y from input $\mathbf{x}$
    - classification: target is a class ID (from 0 to nb. of class - 1)
    - regression: target is a real number

Feature vector: x_i



Extract features



Label: y_i

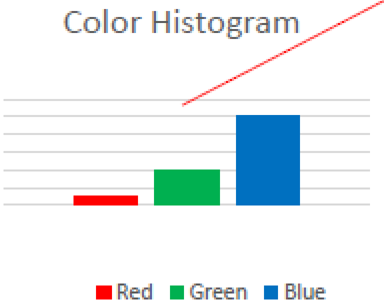
Indoor → 0

Supervised learning

- Task formulation

  - Learning example: $(\mathbf{x}, y)$
  - Task to solve: predict target y from input $\mathbf{x}$
    - classification: target is a class ID (from 0 to nb. of class - 1)
    - regression: target is a real number

Extract features



Feature vector: x_j

Label: y_j

outdoor → 1

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$
  - Find $y = f(x)$ using training data
  - s.t. f correct on test data

What kind of functions?

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. f correct on test data

Hypothesis class

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. f correct on test data

Connection between training data and test data?

Learning problem

- Problem setup

  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. f correct on test data i.i.d. from distribution D

They have the same distribution

i.i.d.: independently identically distributed

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. f correct on test data i.i.d. from distribution D

What kind of performance measure?

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

Various loss functions

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

  - Examples of loss functions:
    - 0-1 loss: $l(f, x, y) = \mathbb{I}[f(x) \neq y]$ and $L(f) = \Pr[f(x) \neq y]$
    - l_2 loss: $l(f, x, y) = [f(x) - y]^2$ and $L(f) = \mathbb{E}[f(x) - y]^2$

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \le i \le n\}$ i.i.d. from distribution D
  - Find $y = f(x) \in \mathcal{H}$ using training data
  - s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

How to use?

Learning problem

- Problem setup
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $y = f(x) \in \mathcal{H}$ that minimizes $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i)$
  - s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

Empirical loss

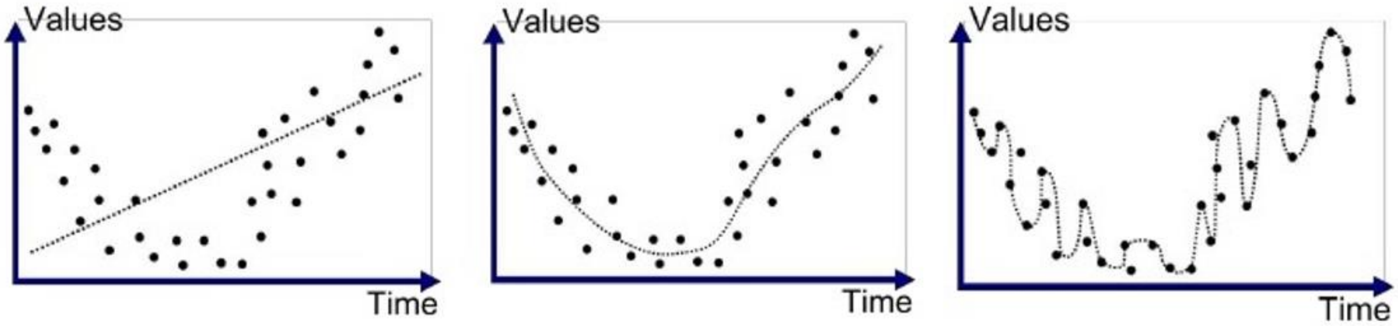
Supervised learning pipeline

- Three steps
  - Collect data and extract features
  - Build model: choose hypothesis class $\mathcal{H}$ and loss function l
  - Optimization: minimize the empirical loss

- Datasets & hyper-parameters
  - Hyper-parameter: a parameter of a model that is not trained (specified before training)
    - Training set $\mathcal{D}^{\mathrm{train}}$ serves to train a model
    - Validation set $\mathcal{D}^{\mathrm{valid}}$ serves to select hyper-parameters
    - Test set $\mathcal{D}^{\mathrm{test}}$ serves to estimate the generalization performance (error)

Generalization

- Model selection for better generalization
  - Capacity: flexibility of a model
  - Underfitting: state of model which could improve generalization with more training or capacity
  - Overfitting: state of model which could improve generalization with less training or capacity
  - Model Selection: process of choosing the best hyper-parameters on validation set



Underfitted Good Fit/Robust Overfitted

Generalization

- Training/Validation curves

Questions

- Generalization
  - Interaction between training set size/capacity/training time and training error/generalization error
- If capacity increases:
  - Training error will ?
  - Generalization error will ?
- If training time increases:
  - Training error will ?
  - Generalization error will ?
- If training set size increases:
  - Generalization error will ?
  - Gap between the training and generalization error will ?

Linear regression

- Formulation
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $f_w(x) = w^T x$ that minimizes $\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2$

l_2 loss; also called mean square error

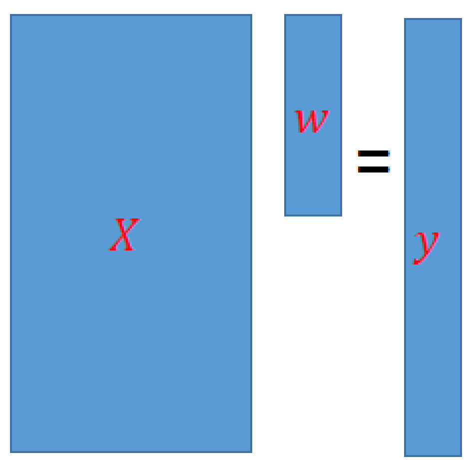
Hypothesis class $\mathcal{H}$

Linear regression

- Optimization

  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $f_w(x) = w^T x$ that minimizes $\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2$
  - Let X be a matrix whose i-th row is x_i^T, y be the vector $(y_1, \dots, y_n)^T$

$$\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2 = \frac{1}{n}\|Xw - y\|_2^2$$



Linear regression

- Optimization
  - Set the gradient to 0 to get the minimizer

$$\nabla_w \hat{L}(f_w) = \nabla_w \frac{1}{n} \|Xw - y\|_2^2 = 0$$

$$\nabla_w [(Xw - y)^T (Xw - y)] = 0$$

$$\nabla_w [\, w^T X^T X w - 2 w^T X^T y + y^T y] = 0$$

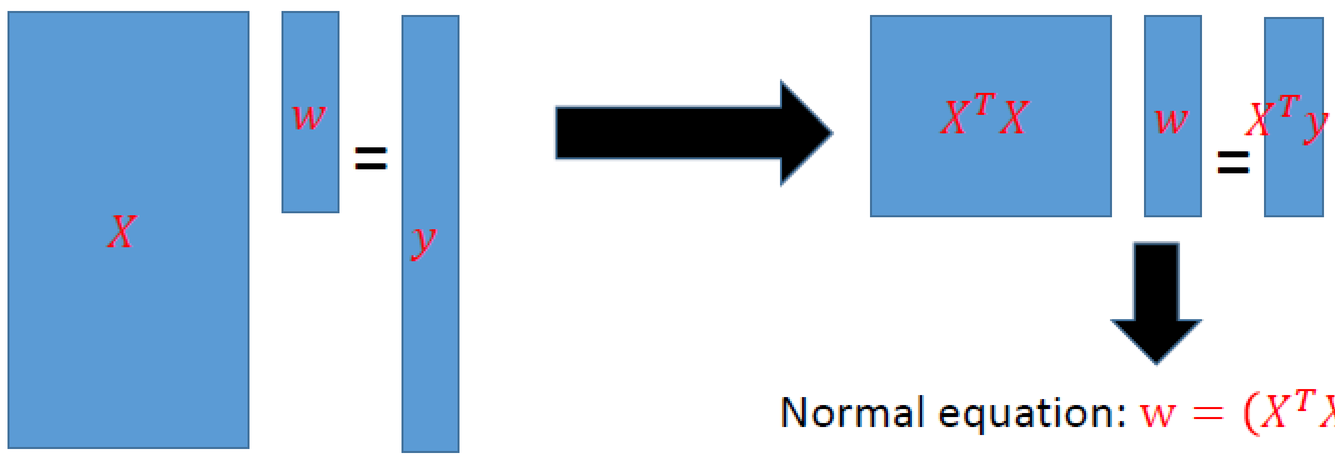
$$2X^T X w - 2X^T y = 0$$

$$\mathrm{w} = (X^T X)^{-1} X^T y$$

Linear regression

- Optimization

  - Algebraic view of the minimizer
    - If X is invertible, just solve $Xw = y$ and get $w = X^{-1}y$
    - But typically X is a tall matrix



Normal equation: $w = (X^T X)^{-1} X^T y$

  - What if not invertible?

Linear regression

- With bias term

Bias term

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $f_{w,b}(x) = w^T x + b$ to minimize the loss

- Reduce to the case without bias:
  - Let $w' = [w;\ b], x' = [x; 1]$
  - Then $f_{w,b}(x) = w^T x + b = (w')^T (x')$

Linear regression

- Why l_2 loss?
  - Why not choose another loss
    - l_1 loss, hinge loss, exponential loss, …
  - Empirical: easy to optimize
    - For linear case: $\mathrm{w} = (X^T X)^{-1} X^T y$
  - Theoretical: a way to encode prior knowledge

Questions:

- What kind of prior knowledge?
- Principal way to derive loss?

A probabilistic view

- Maximum likelihood estimation (MLE)
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(x, y): \theta \in \Theta\}$ be a family of distributions indexed by θ
  - Would like to pick θ so that $P_\theta(x, y)$ fits the data well

A probabilistic view

- Maximum likelihood estimation (MLE)
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(x, y): \theta \in \Theta\}$ be a family of distributions indexed by θ
  - "fitness" of θ to one data point (x_i, y_i)

$$\text{likelihood}(\theta; x_i, y_i) \;:=\; P_\theta(x_i, y_i)$$

A probabilistic view

- Maximum likelihood estimation (MLE)
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(x, y): \theta \in \Theta\}$ be a family of distributions indexed by θ
  - "fitness" of θ to i.i.d. data points $\{(x_i, y_i)\}$

$$\text{likelihood}(\theta; \{x_i, y_i\}) := P_\theta(\{x_i, y_i\}) = \prod_i P_\theta(x_i, y_i)$$

A probabilistic view

- Maximum likelihood estimation (MLE)
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(x, y): \theta \in \Theta\}$ be a family of distributions indexed by θ
  - MLE: maximize "fitness" of θ to i.i.d. data points $\{(x_i, y_i)\}$

$$\theta_{ML} = \mathrm{argmax}_{\theta \in \Theta} \prod_i P_\theta(x_i, y_i)$$

A probabilistic view

- Maximum likelihood estimation (MLE)
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(x, y): \theta \in \Theta\}$ be a family of distributions indexed by θ

  - MLE: maximize "fitness" of θ to i.i.d. data points $\{(x_i, y_i)\}$

$$\theta_{ML} = \text{argmax}_{\theta \in \Theta} \log[\prod_i P_\theta(x_i, y_i)]$$

$$\theta_{ML} = \text{argmax}_{\theta \in \Theta} \sum_i \log[P_\theta(x_i, y_i)]$$

A probabilistic view

- Maximum likelihood estimation (MLE)
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(x, y): \theta \in \Theta\}$ be a family of distributions indexed by θ

  - MLE: negative log-likelihood loss

$$\theta_{ML} = \text{argmax}_{\theta \in \Theta} \sum_i \log(P_\theta(x_i, y_i))$$

$$l(P_\theta, x_i, y_i) = -\log(P_\theta(x_i, y_i))$$

$$\hat{L}(P_\theta) = -\sum_i \log(P_\theta(x_i, y_i))$$

A probabilistic view

- MLE: conditional log-likelihood

  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Let $\{P_\theta(y|x): \theta \in \Theta\}$ be a family of distributions indexed by θ

Only care about predicting y from x; do not care about p(x)

  - MLE: negative conditional log-likelihood loss

$$\theta_{ML} = \operatorname{argmax}_{\theta \in \Theta} \sum_i \log(P_\theta(y_i|x_i))$$

$$l(P_\theta, x_i, y_i) = -\log(P_\theta(y_i|x_i))$$

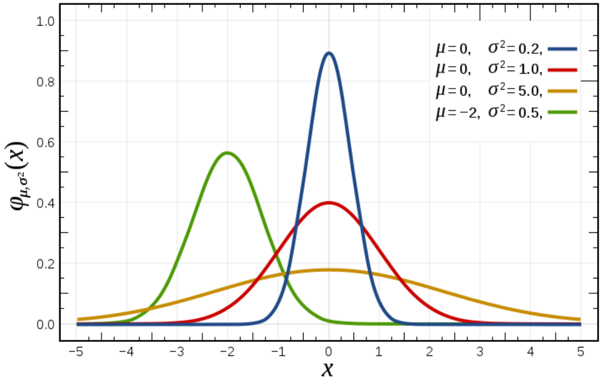
$$\hat{L}(P_\theta) = -\sum_i \log(P_\theta(y_i|x_i))$$

MLE example: Regression

- Regression with l2 Loss
  - Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
  - Find $f_\theta(x)$ that minimizes $\hat{L}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n}(f_\theta(x_i) - y_i)^2$

l_2 loss: Normal + MLE

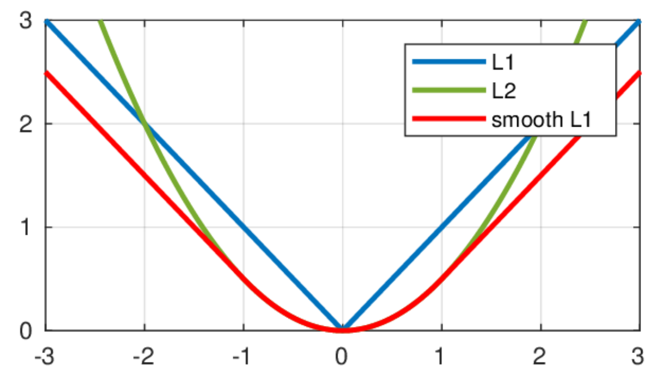
- Define $P_\theta(y|x) = \text{Normal}(y; f_\theta(x), \sigma^2)$
- $\log(P_\theta(y_i|x_i)) = \frac{-1}{2\sigma^2}(f_\theta(x_i) - y_i)^2 - \log(\sigma) - \frac{1}{2}\log(2\pi)$
- $\theta_{ML} = \text{argmin}_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n}(f_\theta(x_i) - y_i)^2$



Outliers

- L2 loss = Gaussian distribution
- What if we have outliers?
  - Some data points lie far away from the linear structure

- Robust estimation
  - L1 loss: Laplace distribution

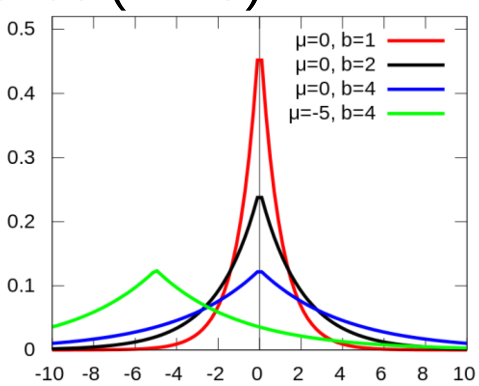


$$\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n} |w^\intercal x_i - y_i|$$

  - Optimization: iteratively reweighted least squares (IRLS)

$$\beta_i^{(0)} = 1, \quad \beta_i^{(t)} = \frac{1}{|{w^{(t)}}^\intercal x_i - y_i|}$$

$$w^{(t+1)} = \arg\min_w \frac{1}{n}\sum_{i=1}^{n} \beta_i^{(t)} \|w^\intercal x_i - y_i\|^2$$



Generalization

- Overfitting or under-determined?
  - Ridge regression

$$\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n} \|w^\intercal x_i - y_i\|^2 + \lambda \sum_{j=1}^{d} w_j^2$$

    - The optimal weights $w^* = (X^\intercal X + \lambda I)^{-1} X^\intercal Y$

  - Lasso regression

$$\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n} \|w^\intercal x_i - y_i\|^2 + \lambda \sum_{j=1}^{d} |w_j|$$

    - Convex optimization: proximal gradient descent

  - The hyper-parameter λ controls the model complexity

Summary

- Introduction to deep learning
- Course logistics
- Review of basic math & ML

- Next time
  - Basic neural networks
  - First Quiz on prerequisite