M2177.0043 Introduction to Deep Learning

Lecture 1: Introduction

Hyun Oh Song[1]

[1] Dept. of Computer Science and Engineering, Seoul National University

March 17, 2020

Outline

For those who are waitlisted

- Following people should submit the override form to Seohyun Park in CSE office:
  - Anyone who's majoring in Computer science, Electrical engineering, Mathematics, Statistics, Liberal studies.
  - Or anyone who's minoring in Computer science

Logistics

- Teaching Staff: (Instructor) Prof. Hyun Oh Song, (TAs) Wonho Choo, Janghyun Kim
- Lectures: Tuesdays & Thursdays 5:00-6:15 PM
- Material a bit different from last years course. Minimal review of probability & linear algebra. More on the underlying principles behind the state of the art machine learning.

Logistics

- Textbook: No official textbook. (optional-1) Deep learning-Ian Goodfellow and Yoshua Bengio and Aaron Courville-MIT Press-2016, free online; (optional-2) Introduction to Probability by Bertsekas and Tsitsiklis
- Prerequisites: Students are expected to have taken courses in linear algebra, multivariate calculus, and probability and statistics as prerequisites. Also familiarity with Numpy or a related matrix-oriented programming language is strongly recommended.
- Course website: On ETL. I strongly recommend everyone to engage questions and discussions in the Q&A board on ETL.

Grading

- Homeworks should be submitted on ETL as LaTex generated pdf files. Use the LaTex template posted on ETL.

Grading

- Homeworks should be submitted on ETL as LaTex generated pdf files. Use the LaTex template posted on ETL.
- Grade: Homework (20%), Midterm (30%), Final (40%), Participation/Quiz (10%)

Grading

- Homeworks should be submitted on ETL as LaTex generated pdf files. Use the LaTex template posted on ETL.
- Grade: Homework (20%), Midterm (30%), Final (40%), Participation/Quiz (10%)
- Homeworks: Problems will be posted on ETL. No late homeworks will be accepted (no exceptions). There's no need to submit doctor's notes, conference travel invitations, etc. We will drop your lowest homework instead.

$$\texttt{total\_hw\_score} = \frac{1}{n-1}\left(\sum_{i}^{n} \texttt{score}(i) - \min_{i} \ \texttt{score}(i)\right)$$

Collaboration

- Collaboration on solving the homework is allowed. Discussions are encouraged.
- However, you should think about the problems on your own. If you do collaborate with someone or use a book or website, you are expected to write up your solution independently.

Outline

What is this course about?

- This class is not about learning how to use Tensorflow/Caffe/Theano/Torch.
- The goal of this class is to provide thorough grounding in the algorithms, mathematics, theories, and the underlying intuition needed to do in-depth research and applications in machine (shallow and deep) learning.

Machine learning

- Study of algorithms that learn from the data x

$$x \longrightarrow \boxed{f(x)} \longrightarrow y$$

Supervised learning

- Binary/Multiclass classification
  Given x find y in $\{1, \ldots, K\}$.
- Regression
  Given x find y in $\mathbb{R}^d$
- Sequence annotation
  Given sequence $x_1, \ldots, x_n$, find $y_1, \ldots, y_m$
- Prediction
  Given x_t and $y_1, \ldots, y_{t-1}$, find y_t

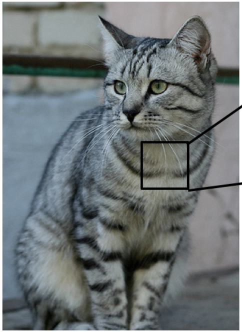
Example: Classification

This image by Nikita is licensed under CC-BY 2.0

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

$\longrightarrow$ cat

Example: Classification



This image by Nikita is
licensed under CC-BY 2.0

```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

What the computer sees

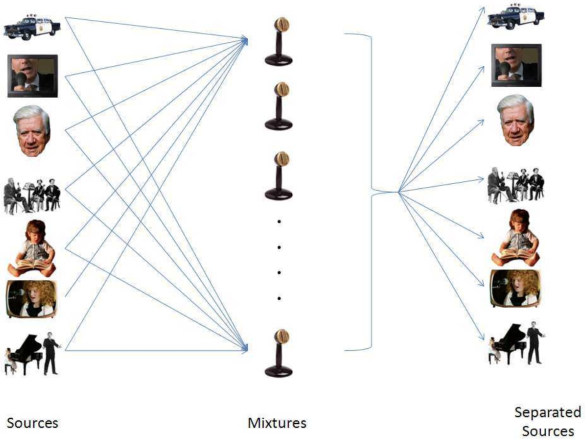
An image is just a big grid of numbers between [0, 255]:

e.g. 800 x 600 x 3
(3 channels RGB)

Unsupervised learning

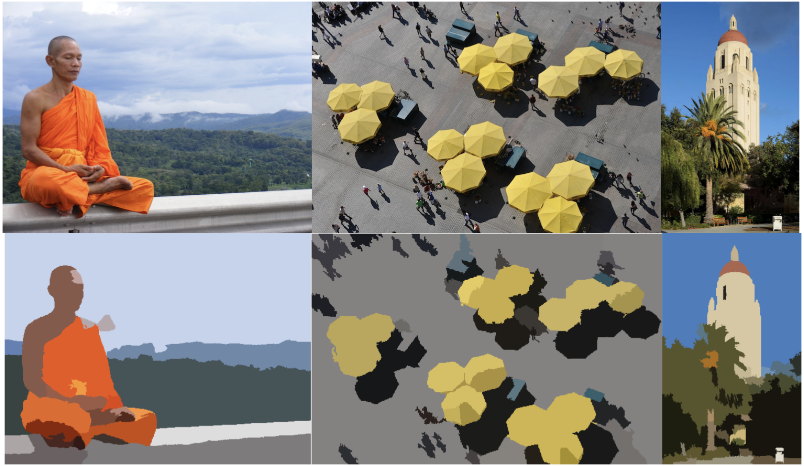
- Clustering
  Find a set of prototypes representing the data
- Sequence Analysis
  Find a latent casual sequence for observations (HMM, Kalman Filter, etc)
- Independent components / dictionary learning
  Find a set of factors for observation
- Novelty detection
  Find the odd one out

Example: Independent Component Analysys

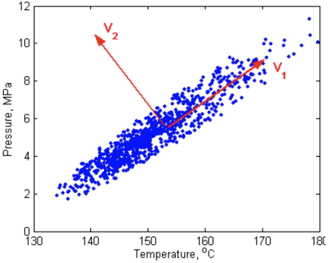


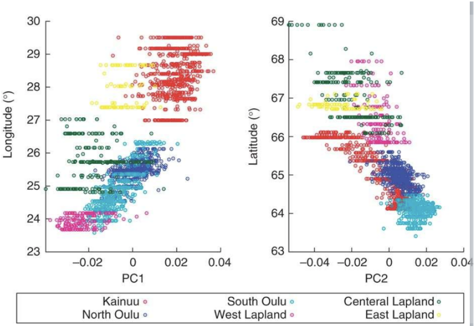
find them automatically

Example: Normalized Cuts



Example: Principal component analysis



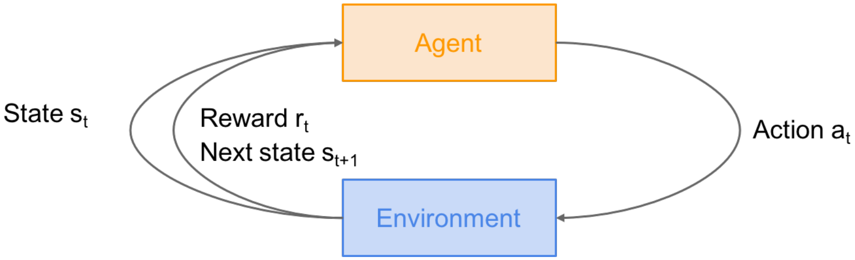


[Variance component model to account for sample structure in genome-wide association studies](), **Nature Genetics 2010**

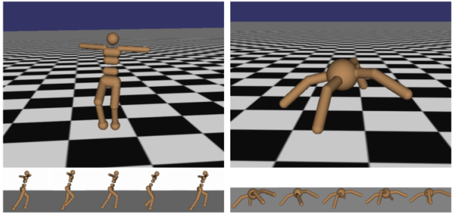
Interaction with Environments

- Online
  Observe x_t, predict $f(x_t)$, observe y_t
- Active learning
  Query y_t for x_t, improve model, pick new x
- Bandits
  Pick arm, get reward, pick new arm. Also viewed as one-state or stateless reinforcement learning
- Reinforcement learning
  Take action, environment responds, take new action

Reinforcement learning



Robot locomotion



Objective: Make the robot move forward

State: Angle and position of the joints
Action: Torques applied on joints
Reward: 1 at each time step upright + forward movement

Atari Games

[Mnih et al. NIPS Workshop 2013; Nature 2015]

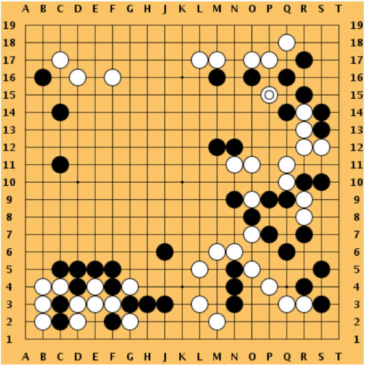
Objective: Complete the game with the highest score

State: Raw pixel inputs of the game state
Action: Game controls e.g. Left, Right, Up, Down
Reward: Score increase/decrease at each time step

Figures copyright Volodymyr Mnih et al., 2013. Reproduced with permission.

- How many possible configurations of image states are there?

Go



Objective: Win the game!

State: Position of all pieces
Action: Where to put the next piece down
Reward: 1 if win at the end of the game, 0 otherwise

- How many possible configurations of board states are there?

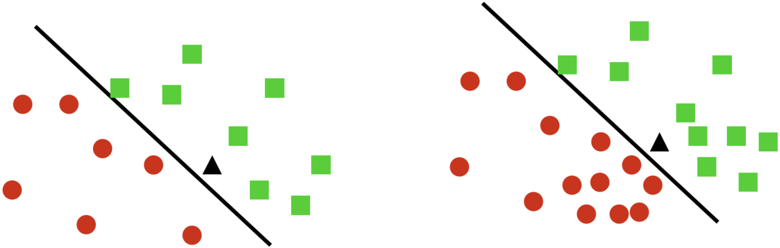
Discriminative vs. Generative

- Discriminative models
  Estimate $p(y|x)$ directly. Often leads to better convergence and simpler solutions.

- Generative models
  - Estimate the joint distribution over $p(y,x)$.
  - Use conditional probability to infer $p(y|x) = \frac{p(y,x)}{p(x)}$.
  - Can use $p(x)$ to generate data.

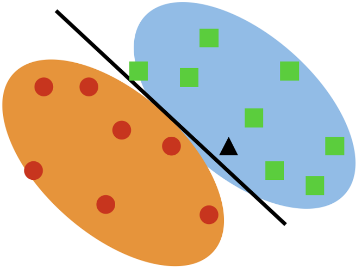
Discriminative

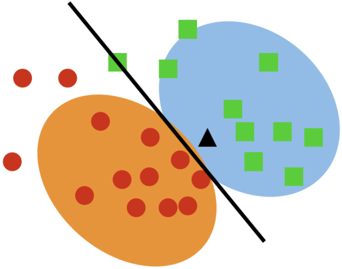
- Only care about estimating the conditional probabilities.
- Very good when underlying distribution of data is really complicated.



Generative

- Model observation $p(x, y)$ first.
- Then infer $p(y|x)$.
- Good for missing variables.





Outline

Tentative Agenda

- Score functions and Loss functions
- Theoretical analysis of optimization algorithms
- Introduction to neural networks
- Convolutional neural networks
- Practical issues in training networks
- CNN architectures
- Recurrent networks
- Deep metric learning
- Generative models - GANs, VAEs
- Disentanglement
- Adversarial attacks and defense
- Data mixup augmentation
- Network pruning and efficient inference
- Deep reinforcement learning

General tips

- Learning how to learn coursera (haven't checked but heard it's good) https://www.coursera.org/learn/learning-how-to-learn
- Take the homeworks seriously and start early.
- Engage in discussions with your peers.
- Actively utilize the Q&A board on ETL.