If we have seen further, it is by standing on the shoulders of Giants

- From basic to most recent SotA
- Slightly biased towards Weizmann research
- Intuition
- Hands-on
- Openness
Logistics

• Everything on course website / Piazza

• 4 credit points

• 2 hrs. lecture, 1 hr. tutorial (Tutorial covers new material)

• 4 homework assignments + Final Project

• All communication through Piazza
We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term "Overfitting").

Homework

Theory
From Scratch
Applied

HW1 is online!
Road map

Slide idea: Justin Johnson
Deep Learning is powerful

Andrej Karpathy, Li Fei-Fei, CVPR 2015 Deep Visual-Semantic Alignments for Generating Image Descriptions

Abhishek Bansal- DetectMe (GitHub)
Deep Learning is powerful!

an armchair in the shape of an avocado. an armchair imitating an avocado.

an illustration of an eggplant with a mustache playing a guitar
Supervised Learning

<table>
<thead>
<tr>
<th>Number of new Recipients</th>
<th>Email Length (K)</th>
<th>Country (IP)</th>
<th>Customer Type</th>
<th>Email Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Germany</td>
<td>Gold</td>
<td>Ham</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>Germany</td>
<td>Silver</td>
<td>Ham</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>Nigeria</td>
<td>Bronze</td>
<td>Spam</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Russia</td>
<td>Bronze</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Germany</td>
<td>Bronze</td>
<td>Ham</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>USA</td>
<td>Silver</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>USA</td>
<td>Silver</td>
<td>Spam</td>
</tr>
</tbody>
</table>

Features

- Numeric
- Nominal
- Ordinal

Labels

Hypothesis
Supervised Learning

Regression

E.g. Image denoising

E.g. Object localization

Classification

E.g. Image classification

E.g. Image classification
Supervised Learning

$$\mathbf{A} \{\mathbf{S}\} = \mathbf{h}$$

Hypothesis

Loss

Optimization method

Training set

Hypothesis class

$$\mathcal{H} = \{h_1, h_2 \ldots \}$$

$$h(x) \approx y$$

Inputs

$$\mathbf{X} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_M^T \end{bmatrix}$$

Labels

$$\mathbf{Y} = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_M^T \end{bmatrix}$$

* Diagram by Andrew NG
Linear Regression

Hypothesis class:  Linear

\[ H = \{ h_\theta \mid \theta \in \mathbb{R}^{N+1} \} \]

\[ h_\theta(x) = \theta_0 + \sum_{j=1}^{N} \theta_j x_j = \theta_0 + \bar{\theta}^T x = \theta^T \begin{pmatrix} 1 \\ x \end{pmatrix} \]

Loss:  Mean Squared Error

\[ L = \frac{1}{2M} \sum_{i=1}^{M} (h_\theta(x_i) - y_i)^2 = \frac{1}{2M} \|X \theta - y\|^2 \]

Optimization method:  Normal equations / Gradient Descent

Bias= Just add 1 at top of the input vec!
Normal Equations (intuition)

\[
\hat{\theta} = \arg \min_{\theta} \| y - X\theta \|^2
\]

\[
x \perp e \quad \forall \ x \in \text{span}(X)
\]

\[
X^T(X\hat{\theta} - y) = 0
\]

\[
\hat{\theta} = (X^T X)^{-1} X^T y \quad \ast
\]

Formal proof: HW

Also in HW: is \(X^T X\) invertible?

\(\ast\) if \(X^T X\) invertible
Normal Equations

Q: Will normal equations always be practical?

A: No;

1. $X^TX$ may cost unreasonable memory

IT WAS NICE

WHILE IT LASTED
Gradient descent

Possible solution: Iteratively reduce loss

Can we guarantee global min?
Gradient descent

Parameter space: \( \mathcal{L}(\theta; S) \)

Data space: \( h_\theta(x) \)
Calculus reminder: Directional derivative

\[
\lim_{\varepsilon \to 0} \frac{f(x + \varepsilon v) - f(x)}{\varepsilon \|v\|} = \frac{1}{\|v\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{v^T}{\|v\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|v\|} \langle v, \nabla f \rangle
\]

If differentiable

Gradient! \( \nabla f \)

According to Cauchy-Schwarz inequality:
- Max value is \( \|\nabla f\| \)
- Obtained when \( v \) is parallel to \( \nabla f \)

• Gradient directs to steepest ascent.
• It’s size is the max steepness.
Gradient descent

\[ \nabla \mathcal{L}(\theta_0, \theta_1 \ldots \theta_N) = \begin{pmatrix}
\frac{\partial \mathcal{L}}{\partial \theta_0} \\
\frac{\partial \mathcal{L}}{\partial \theta_1} \\
\vdots \\
\frac{\partial \mathcal{L}}{\partial \theta_N}
\end{pmatrix} \]

1. Initialize \( \theta \sim \text{Random} \)
2. Repeat until convergence:
   \[
   \begin{align*}
   \theta &:= \theta - \alpha \nabla \mathcal{L}(\theta; S) \\
   \end{align*}
   \]

\( \alpha \): Learning rate
Gradient descent

Full batch Gradient Descent

\[ \theta := \theta - \alpha \nabla L(\theta; S) \]

Figure by Z² Little on Medium
Gradient descent for Linear Regression

\[ \mathcal{L} = \frac{1}{2m} \sum_{i=1}^{M} \left( \sum_{j=0}^{N} \theta_j x_{ij} - y_i \right)^2 \]

Repeat until convergence:

\[ \{ \theta: = \theta - \frac{\alpha}{m} X^T e \} \]

Q: Find the relation between convergence and Normal Equations
Feature transform

\[ z = \sqrt{x^2 + y^2} \]
Feature transform

**Feature Transform**

- $x_0 = 1$
- $x_1 = x$
- $x_2 = x^2$
- $x_3 = x^3$
- \vdots
- $x_p = x^p$

Non-linear hypothesis!

(Polynomial)
Error decomposition

Hypothesis - class

Loss

- Optimization gap
- Overfitting
- Underfitting
- True XY relation
- non deterministic

$h$: what we finally get
$h_{ERM}$: Best in training-set
$h_{*}$: Best in class
$h_{Bayes}$: Best possible h

“Zero” loss

Best in class

Best in training-set

Best possible h
Generalization

Error

Model “complexity”

Underfitting

Overfitting

Best Fit

Test Error

Training Error
Overfitting - Data influence

Matrix A

\[
\begin{pmatrix}
3 & 4 \\
6 & 8
\end{pmatrix}
\]
Polynomial fitting

Underfitting → Overfitting

Train
Test

\[ P = 29 \]

Train Error: 0.000000
Test Error: 53467905394032967680.000000
This week’s tutorial:

Linear classification

Niv Granot

Next week’s lecture:

(Me Again) Neural Networks