深度学习

Lecture1 Introduction

YMSC, Pang Tongyao

About This Course

- It is
  - An introduction to deep learning methods, from basics to advanced topics.
  - Focused on algorithms and applications, emphasizing the development of strong coding skills. (Math is still important!)
- It isn't
  - Focused on rigorous mathematical definitions and derivations.
  - A comprehensive overview of all topics within deep learning.

About This Course

- Grading Policy
  - Assignments: 3*25%
  - Final Projects: 25%
  - Late policy: 10% off per day late; not accepted after 10 days
- Office hour
  - by appointment
  - email: typang@tsinghua.edu.cn
  - office: 双清综合楼B533

Reference

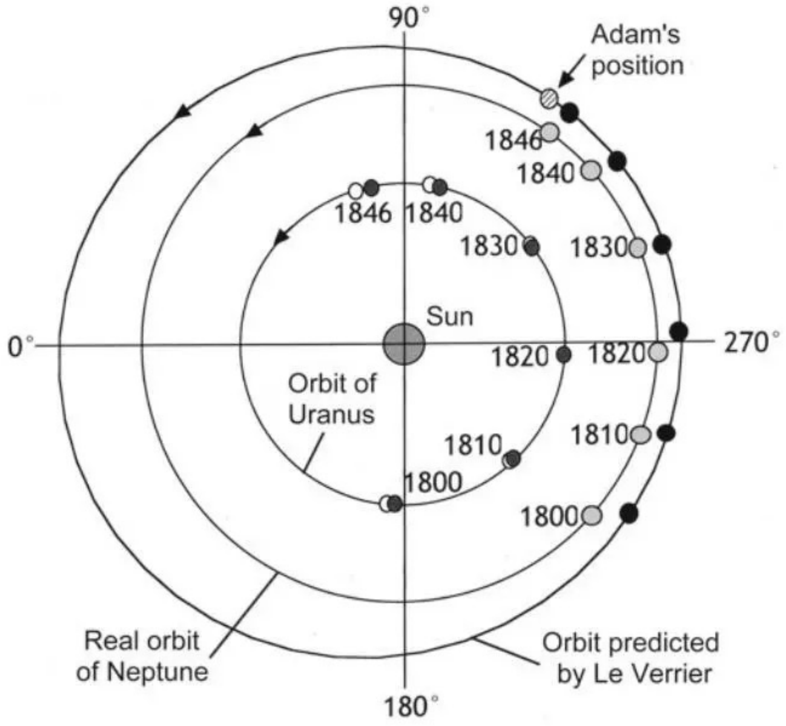
- Books
  - Bishop, Christopher M., and Nasser M. Nasrabadi. **Pattern recognition and machine learning**, Springer, 2006.
  - Bengio, Yoshua, Ian Goodfellow, and Aaron Courville. **Deep learning**, MIT press, 2017.
  - Zhang, Aston and Lipton, Zachary C. and Li, Mu and Smola, Alexander J., **Dive into Deep Learning**, Cambridge University Press, 2023
- Related Courses:
  - 6.8300/6.8301: Advances in Computer Vision, MIT
  - 6.S978: Deep Generative Models, MIT
  - CS231n: Deep Learning for Computer Vision, Stanford
  - Reinforcement Learning, by David Silver, UCL

What is Learning?

Motivating Examples

Year	Housing Price
2021	2
2022	3
2023	4
2024	?

Predict the Housing Price of a City

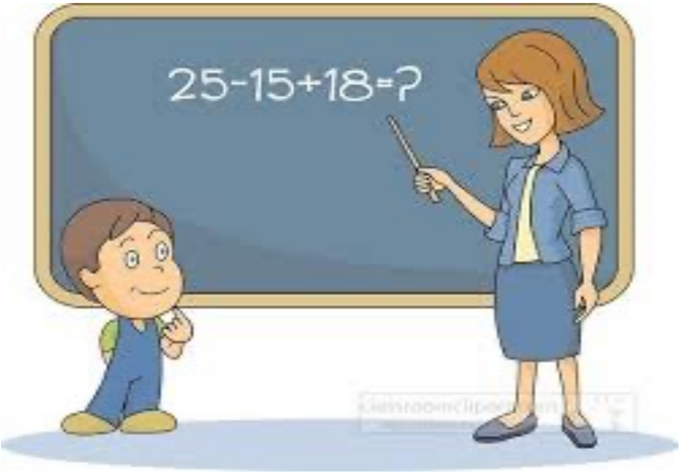


The Discovery of Neptune

Astronomers noticed Uranus' motion differed from predictions and used its trajectory to predict the location and mass of an unknown celestial body.

Motivating Examples

A doggie is learning

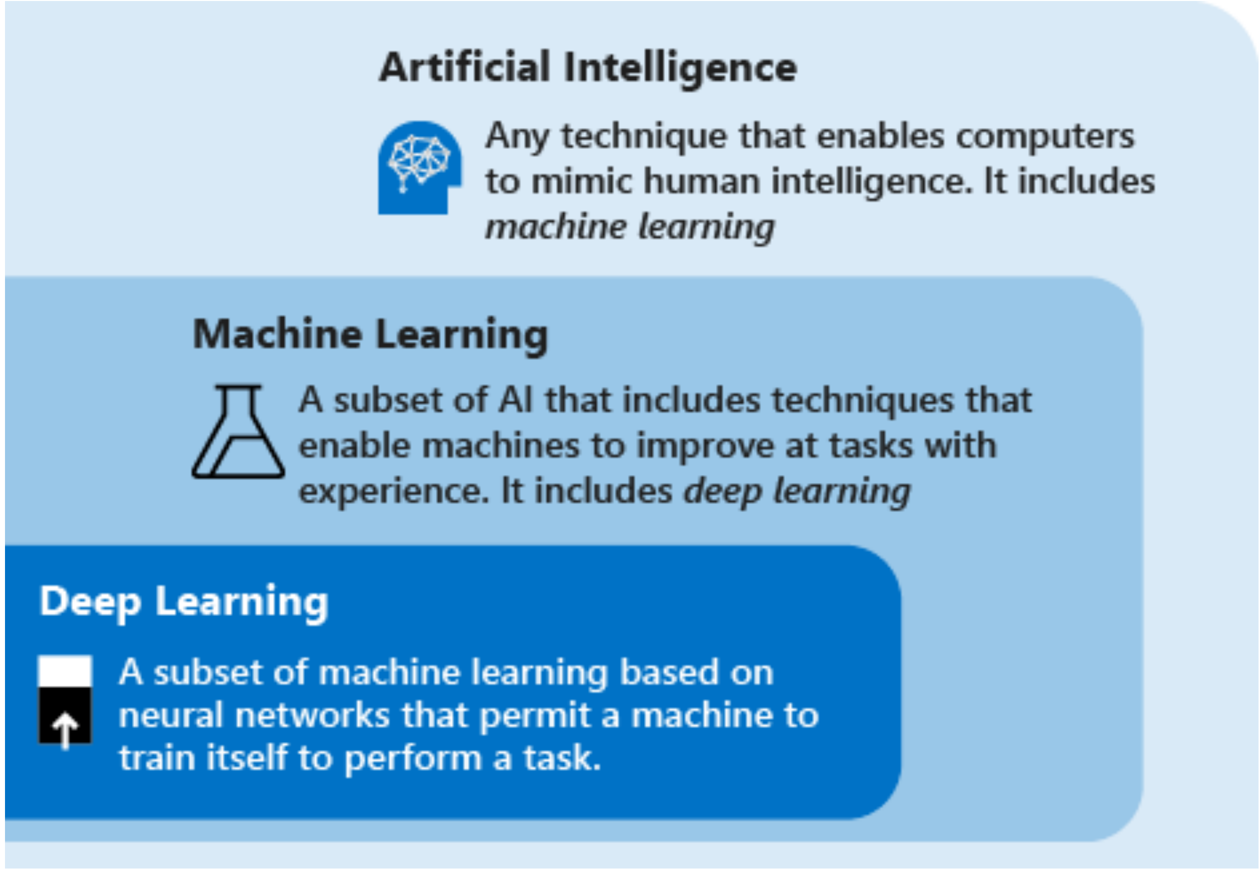


A kid is learning

Also learning……

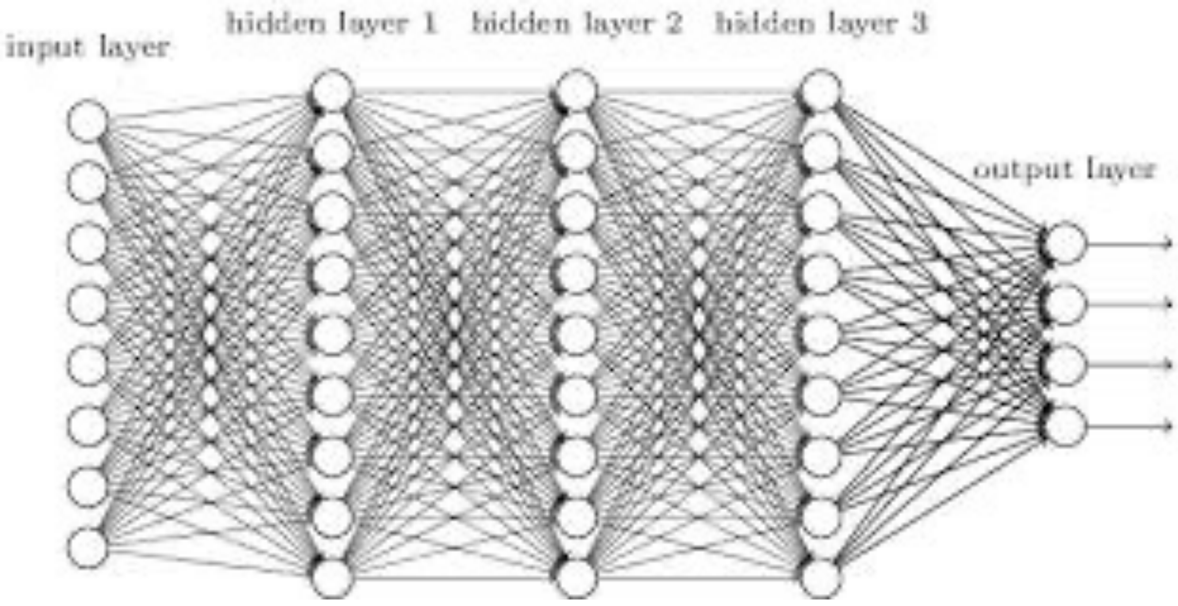
Machine/Deep Learning

- ***Machine learning*** is the study of algorithms that can learn from experience.
- Experience: typically in the form of observational data or interactions with an environment.
- As a machine learning algorithm accumulates more experience, its performance improves.



Deep Learning Methods

- Supervised Learning
- Unsupervised Learning
  - Self-supervised Learning
  - Generative Models
- Reinforcement Learning
- Meta/Transfer Learning
- ……

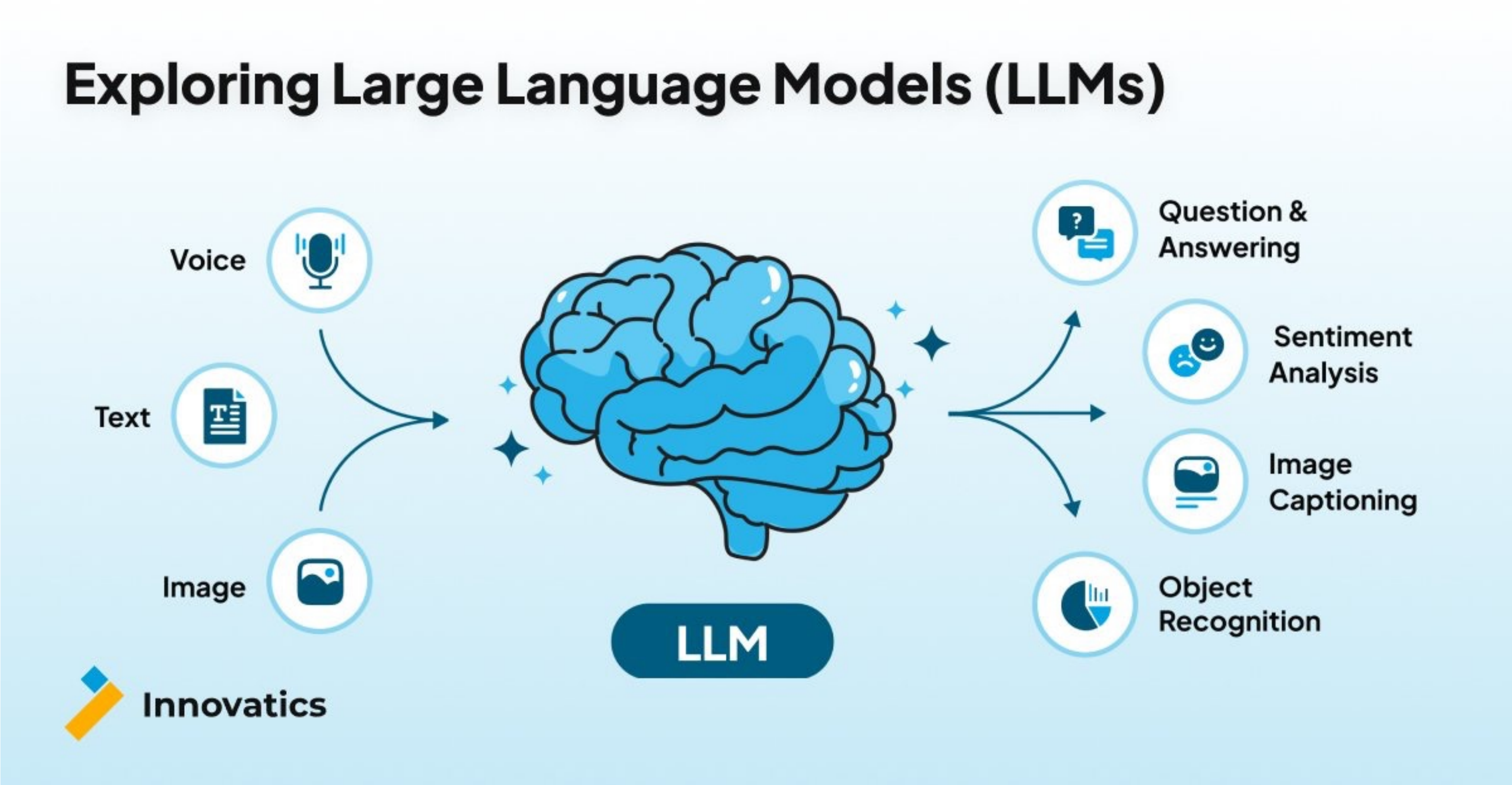


Deep Neural Networks

Applications

Text-to-image Generation

Applications



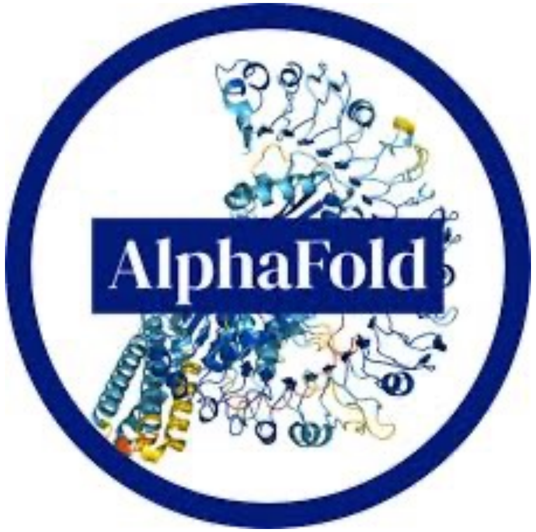
Gemini

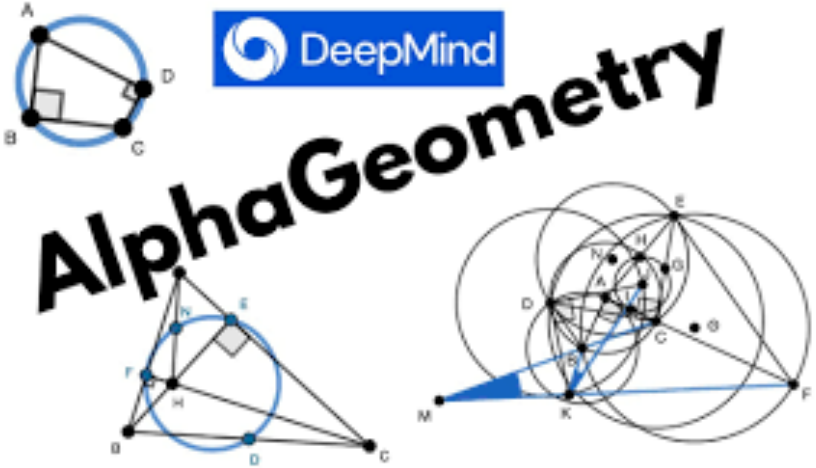
ChatGPT

Claude

deepseek

Applications





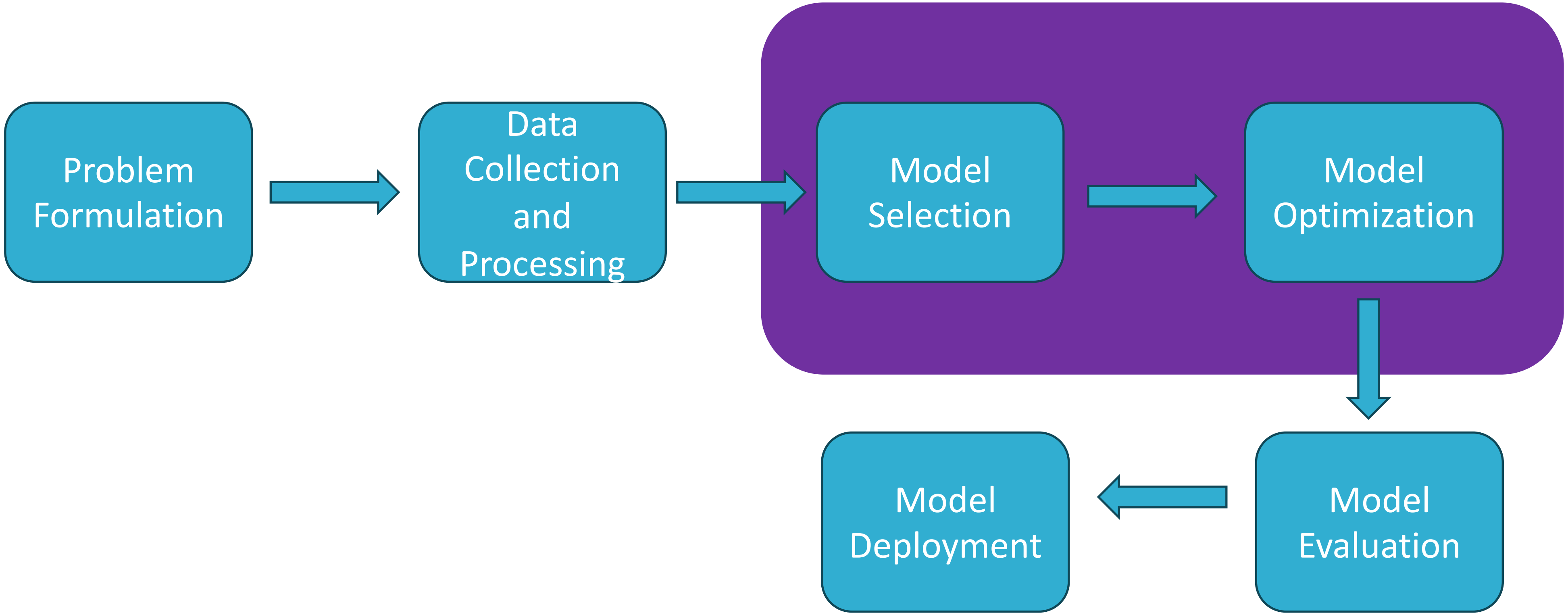
Applications

Embodied Intelligence

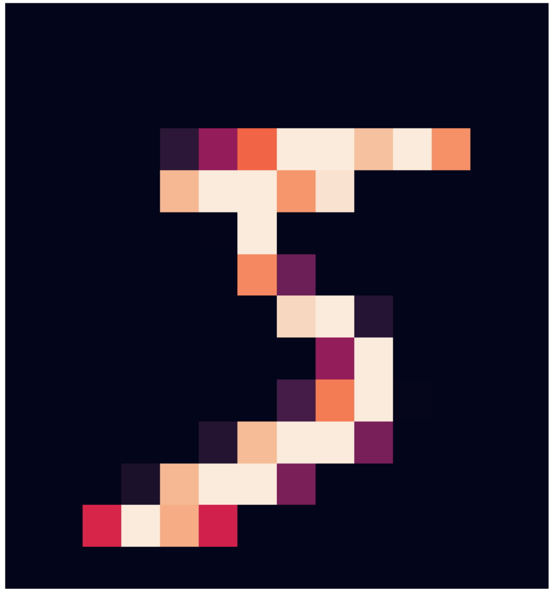
Self-Driving

Start from the very beginning

Typical Process

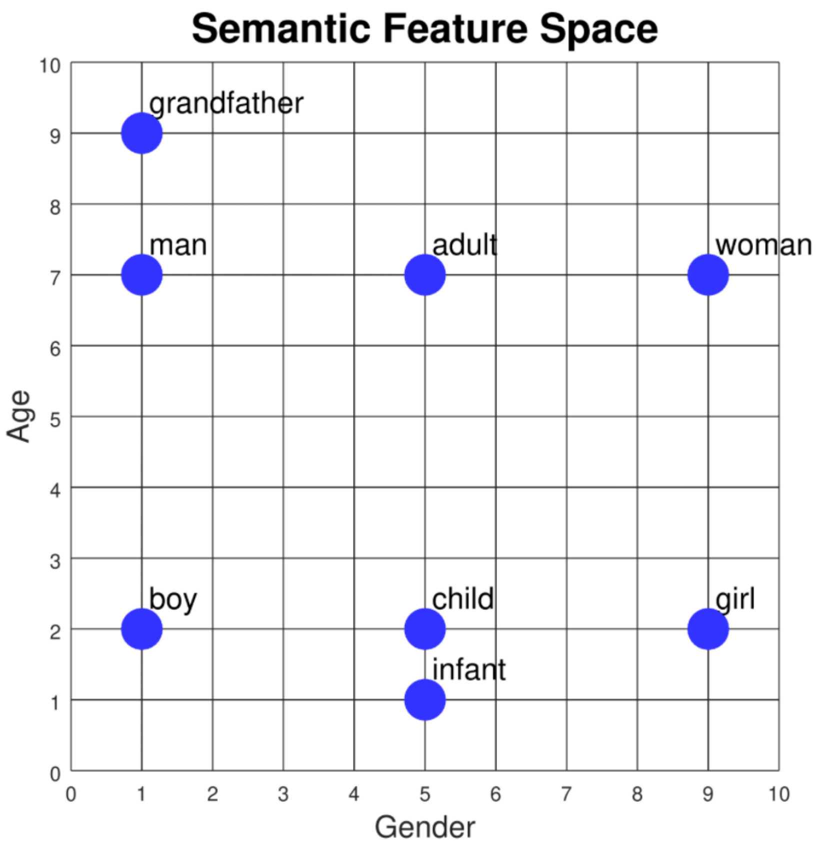


Data Format



0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	30	94	170	253	253	225	253	195	0	0
0	0	0	0	219	253	253	198	247	0	0	0	0	0
0	0	0	0	0	1	253	0	0	0	0	0	0	0
0	0	0	0	0	0	190	70	0	0	0	0	0	0
0	0	0	0	0	0	0	240	253	25	0	0	0	0
0	0	0	0	0	0	0	0	93	253	0	0	0	0
0	0	0	0	0	0	0	46	183	253	2	0	0	0
0	0	0	0	0	24	221	253	253	78	0	0	0	0
0	0	0	18	219	253	253	80	0	0	0	0	0	0
0	0	136	253	212	132	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Data Format



Word Coordinates

	Gender	Age
grandfather	[1,	9]
man	[1,	7]
adult	[5,	7]
woman	[9,	7]
boy	[1,	2]
child	[5,	2]
girl	[9,	2]
infant	[5,	1]

--Dave Touretzky

Data Format

One-hot Encoding

apple = [1 0 0 0 0 ......]

bag = [0 1 0 0 0 ......]

cat = [0 0 1 0 0 ......]

dog = [0 0 0 1 0 ......]

elephant = [0 0 0 0 1 ......]

Data Format

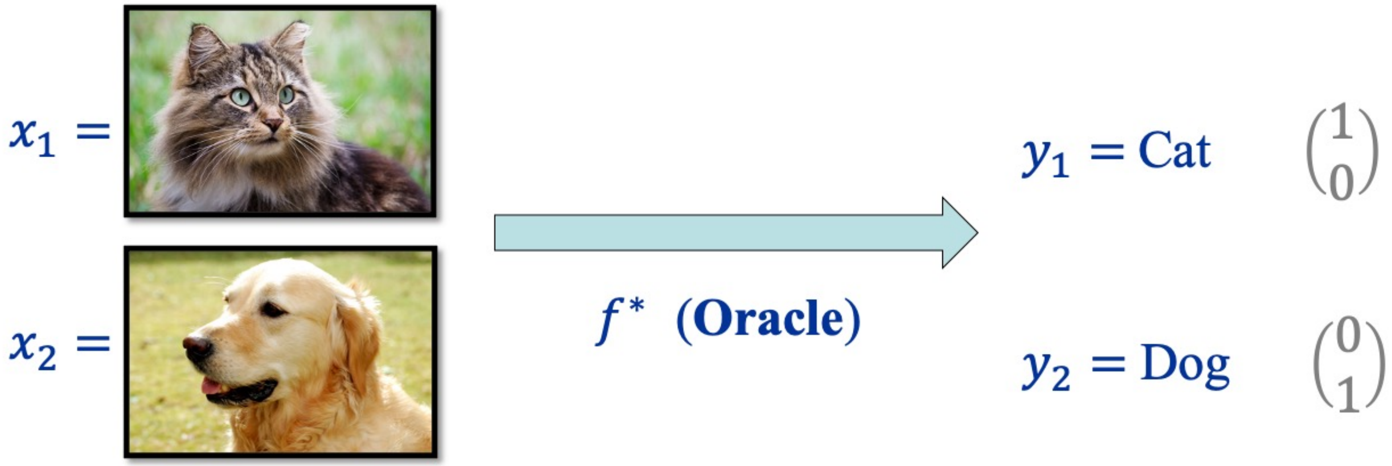
- In deep learning algorithms, most data is in the form of tensors, which can be better utilized for accelerated computation on GPUs.
- Tensor data can still have various structures, such as time series data, graph data, etc.
- Dataset Decomposition

$$\mathcal{D} = \mathcal{D}_{\boldsymbol{train}} \cup \mathcal{D}_{\boldsymbol{test}} \cup \mathcal{D}_{\boldsymbol{validation}}$$

$\mathcal{D}_{\boldsymbol{train}}$: for training; $\mathcal{D}_{\boldsymbol{test}}$: for evaluation; $\mathcal{D}_{\boldsymbol{validation}}$:for selection

Supervised Learning

- Training data with labels: $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, input: x_i , output/label: y_i ,
- Goal: learn the mapping $x_i \to y_i$



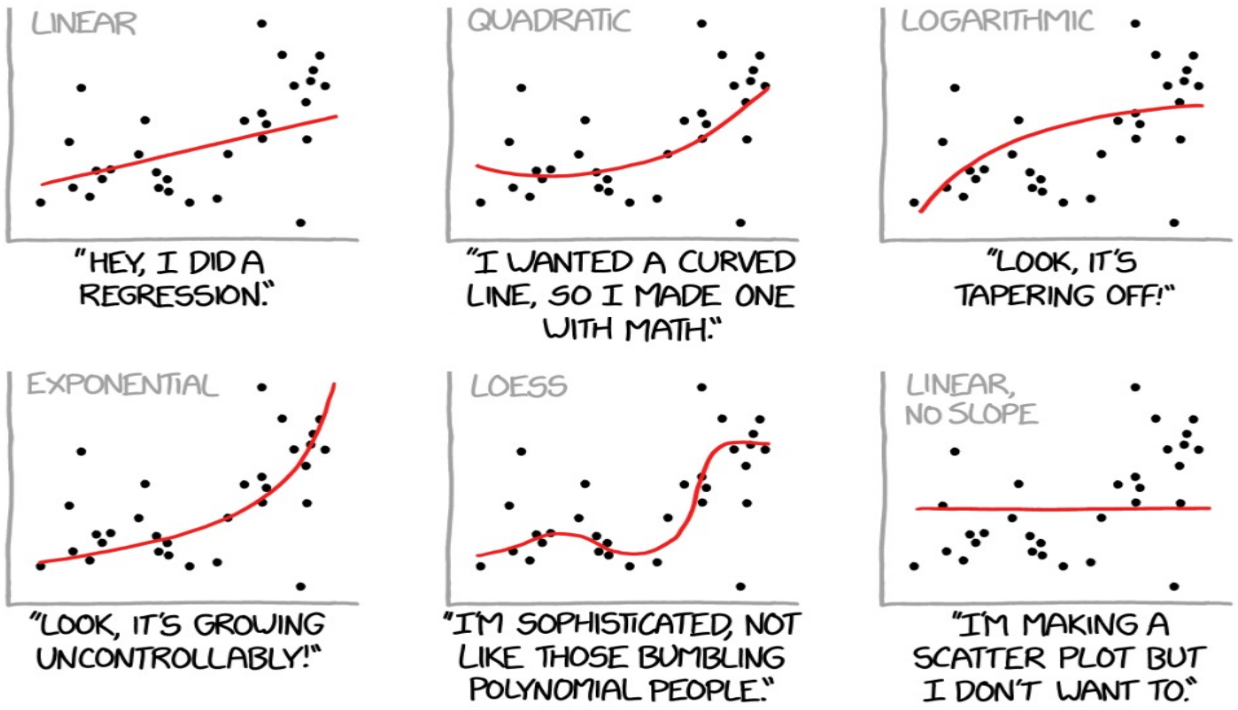
- The oracle f^* is unknown to us, except through the dataset

$$\mathcal{D} = \{x_i, y_i = f^*(x_i)\}_{i=1}^{N}$$

Hypothesis Space

Supervised Learning:

1. Define a **hypothesis space** $\mathcal{H}$ consisting of a set of candidate functions, e.g.
$\mathcal{H} = \{f: f(x) = w_0 + w_1 x\}$
2. Find the "**best**" function $\hat{f}$ in $\mathcal{H}$ that **approximates** f^*



What you get depends on $\mathcal{H}$!

https://xkcd.com/2048/

Optimization

- Define a **loss function** to measure the distance between the model solution and the oracle solution. Then, we can find the best approximation by an optimization problem

$$\min_{f \in \mathcal{H}} R_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), f^*(x_i))$$

The empirical loss

- Optimization algorithms to be explored in the following lectures……

Generalization

- **Generalization ability:** to do well on new, unseen data.

What we **can** solve

Empirical risk minimization

$$\min_{f \in \mathcal{H}} R_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), f^*(x_i)) \quad \Longrightarrow \quad \hat{f}$$

$$x_i \sim \mu$$

Population risk minimization

$$\min_{f \in \mathcal{H}} R_{pop}(f) = \mathbb{E}_{x \sim \mu}[L(f(x), f^*(x))] \quad \Longrightarrow \quad \tilde{f}$$

What we **really** want to solve

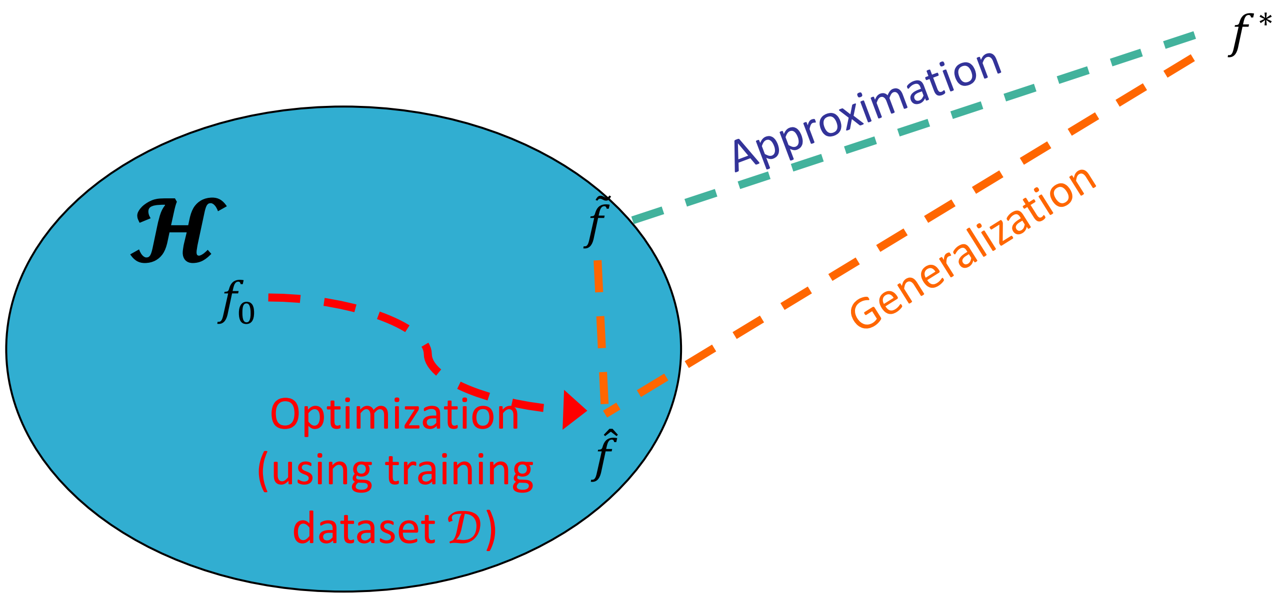
Generalization Gap

$$\hat{f} \neq \tilde{f}$$

Generalization error depends on the training dataset, hypothesis spaces, and optimization algorithms etc.

Paradigms of Supervised Learning

- **Approximation**: the distance between the hypothesis space and the oracle solution.
- **Optimization**: Seeks the best solution within the hypothesis space based on training data.
- **Generalization**: Examines the difference between the optimized solution and the true solution on unseen data.



Linear Regression

Linear Regression

- Linear Model with Continuous Output
- Step 1: Define the hypothesis space as the set of linear models.

$$\mathcal{H} = \{f : f(x) = w_0 + w_1 x, w_0 \in \mathbb{R}, w_1 \in \mathbb{R}\}$$

- Step 2: Find the best approximation.

First, define the loss function, for example:

$$L(f_1, f_2) = \int p(x) ||f_1 - f_2||_2^2 d\mu$$

Minimize the empirical loss over the training dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$

$$\min_{f \in \mathcal{H}} R_{emp}(f) = \min_{w_0, w_1} \frac{1}{2N} \sum_{i=1}^{N} (w_0 + w_1 x_i - y_i)^2$$

Solution:

$$\frac{\partial R_{emp}}{\partial w_0}(\widehat{w}_0, \widehat{w}_1) = 0 \text{ and } \frac{\partial R_{emp}}{\partial w_1}(\widehat{w}_0, \widehat{w}_1) = 0$$

$$\widehat{w}_0 = \bar{y} - \widehat{w}_1 \bar{x} \qquad \widehat{w}_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \qquad \bar{x} = \frac{1}{N} \sum_i x_i \qquad \bar{y} = \frac{1}{N} \sum_i y_i$$

Test process: $y_{test} = \widehat{w}_0 + \widehat{w}_1 x_{test}$

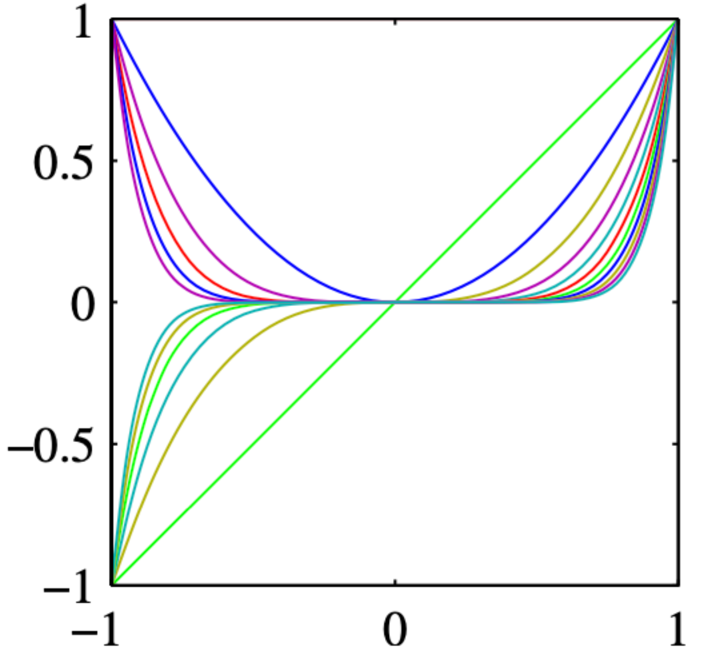
Linear Basis Models

- Q: Can linear models only represent linear relationships between x and y?
- No! Introduce basis functions or feature mappings can extend its ability to model more complex relationships.

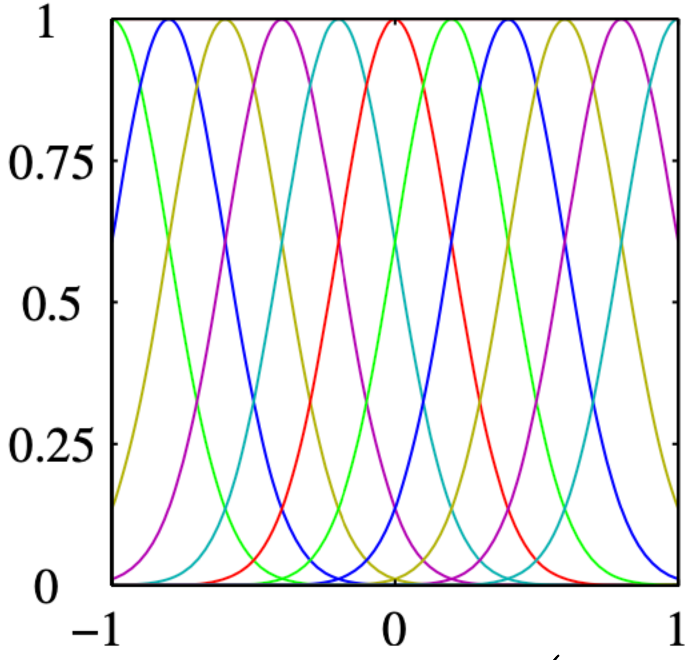
$$\mathcal{H}_M = \left\{ f : f(x) = \sum_{j=0}^{M-1} w_j \phi_j(x) \right\}$$

$\mathcal{H} = \{f : f(x) = w_0 + w_1 x, w_0 \in \mathbb{R}, w_1 \in \mathbb{R}\}$ corresponds to $d = 1, M = 2, \phi_0(x) = 1, \phi_1(x) = x$

Example of Basis Functions



Polynomials: $\phi_j(x) = x^j$



Gaussian: $\phi_j(x) = \exp\left(-\frac{(x-m_j)^2}{2s^2}\right)$

-- Bishop

Linear Basis Models

- Optimization:

$$\min_{f \in \mathcal{H}_M} R_{emp}(f) = \min_{w \in \mathbb{R}^M} R_{emp}(w)$$

$$= \min_{w \in \mathbb{R}^M} \frac{1}{2N} \sum_{i=1}^{N} \left(f(x_i) - y_i\right)^2$$

$$= \min_{w \in \mathbb{R}^M} \frac{1}{2N} \sum_{i=1}^{N} \left(\sum_{j=0}^{M-1} w_j \, \phi_j(x_i) - y_i \right)^2$$

$$\Longleftrightarrow$$

$$\min_{w \in \mathbb{R}^M} \frac{1}{2N} \|\Phi w - y\|^2$$

where

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \cdots & \phi_{M-1}(x_2) \\ \vdots & \ddots & \vdots \\ \phi_0(x_N) & \cdots & \phi_{M-1}(x_N) \end{pmatrix}$$

$$w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Linear Regression

- To solve $\min_{w \in \mathbb{R}^M} \frac{1}{2N} ||\Phi w - y||^2$,

we can do this by setting $\nabla R_{emp}(\hat{w}) = 0 \Leftrightarrow \Phi^{\mathrm{T}}(\Phi \hat{w} - y) = 0$

Solve the linear equation systems:

- When $\Phi^{\mathrm{T}}\Phi$ is invertible, there is an unique solution $\hat{w} = \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}y$
- When $\Phi^{\mathrm{T}}\Phi$ is not invertible, there are many solutions

$$\hat{w}(u) = \Phi^{\dagger}y + \left(I - \Phi^{\dagger}\Phi\right)u \,, \forall\, u \in \mathbb{R}^M, \Phi^{\dagger} = \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}$$

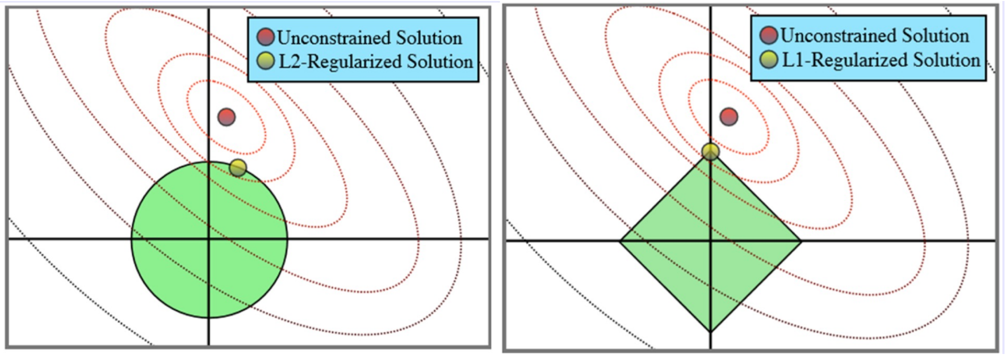
Q: how to select the desirable one?

Linear Regression

- Add regularization

$$\min_{w \in \mathbb{R}^M} \frac{1}{2N} \|\Phi w - y\|^2 + \lambda \underbrace{C(w)}_{\text{the regularization term}}$$

  - ℓ_2 regularization (ridge regression): $C(w) = ||w||_2^2$.
  - ℓ_1 regularization (**lasso**): $C(w) = ||w||_1$, pursuit sparsity
  - ……



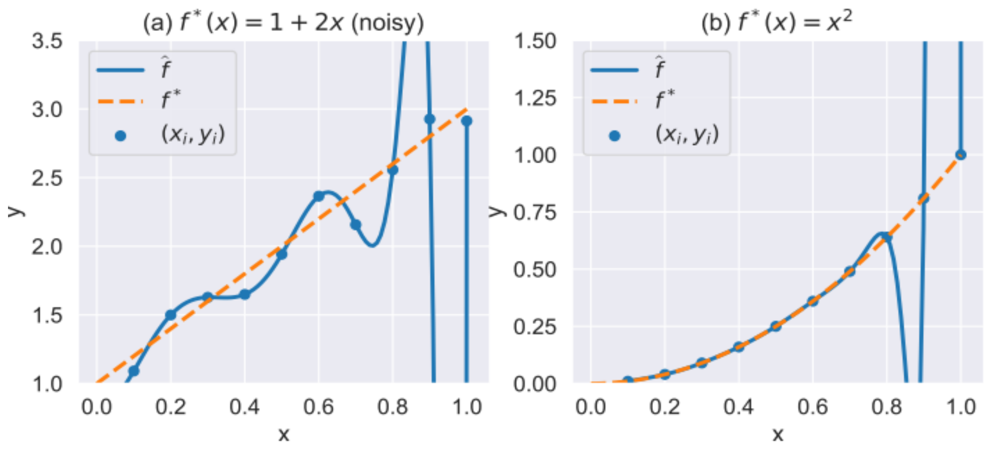
-- Mark Schmidt

Regularization and generalization

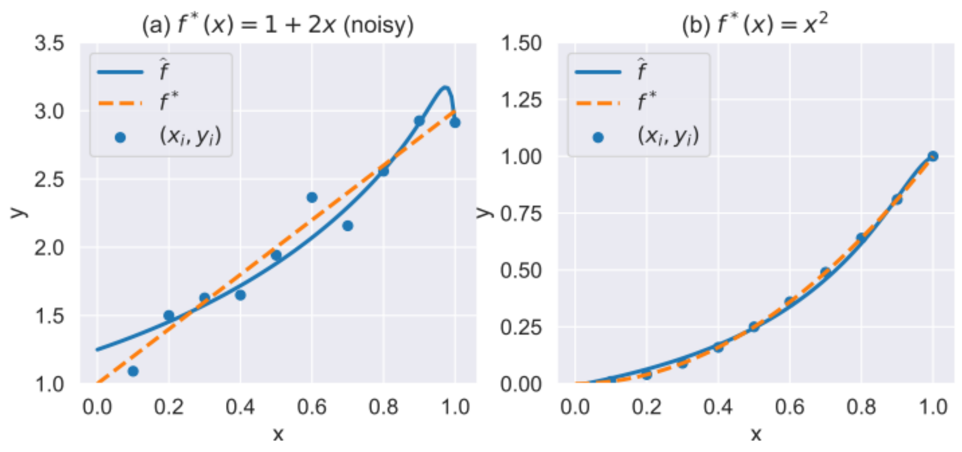
- We apply ℓ^2 regularization on the over-fitting examples
- Consider

$$\mathcal{H}_M = \left\{ f : f(x) = \sum_{i=0}^{99} w_j x^j \right\} \text{ so } M = 100 \text{ , but } N = 10$$

<u>Without regularization</u>



<u>With regularization</u>



Linear Regression

- Ridge regression

$$\min_{w\in\mathbb{R}^M} \frac{1}{2N} \|\Phi w - y\|^2 + \lambda ||w||_2^2$$

$$\hat{w} = \left(\Phi^{\mathrm{T}}\Phi + 2N\lambda I\right)^{-1} \Phi^{\mathrm{T}} y$$

- Lasso: no analytical solution

$$\min_{w\in\mathbb{R}^M} \frac{1}{2N} \|\Phi w - y\|^2 + \lambda ||w||_1$$

Optimization Algorithms

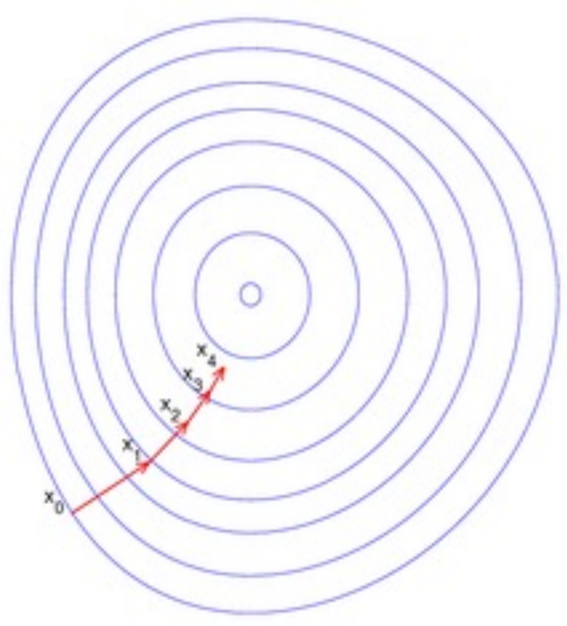
- A necessary first-order optimality condition for $\min f(x)$

$$\nabla f(x^*) = 0$$

- Solve it iteratively:

Gradient Descent : $x_{n+1} = x_n - \gamma \nabla f(x_n)$

- Provided $\gamma < ||\nabla^2 f||$, it can be shown that $||\nabla f(x_n)|| \to 0$

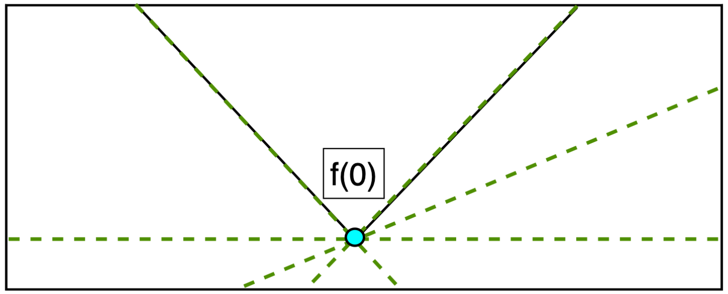


Optimization Algorithms

- $||\cdot||_1$ is not differentiable at 0. Consider subgradient

$$\partial f(x_0) = \{g | f(x) \geq f(x_0) + g^T(x - x_0)\}$$

$$\partial |x| = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ [-1,1], & x = 0 \end{cases}$$



- Subgradient methods: $w_{k+1} = w_k - \gamma_k g_k,\ g_k \in \partial f(w_k)$
- Due to the slow convergence rate of subgradient methods, many advanced algorithms have been proposed to solve Lasso, e.g. ADMM, proximal gradient descent.

Classification

This image by Nikita is licensed under CC-BY 2.0

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

→ cat

(Discrete output)

Logistic Regression

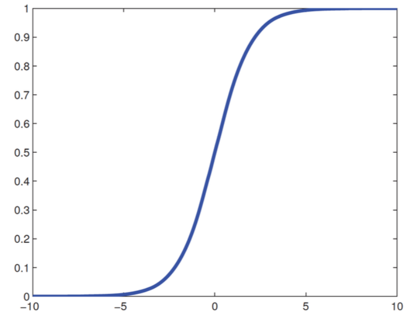
- Binary classification problem:

$$\text{one-hot encoding for the output } \left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right\}$$

This two-dimensional vector can be understood as the probability for each class and can take continuous values.

- A linear hypothesis space: $\{u(x): u = w^T x, x \in \mathbb{R}^n, \mathrm{w} \in \mathbb{R}^n\}$.
- Softmax: Map the extracted feature u to the space of one-hot codes

$$\mu = \frac{1}{1+e^{-u}}, 1-\mu = \frac{e^{-u}}{1+e^{-u}}$$



Logistic Regression

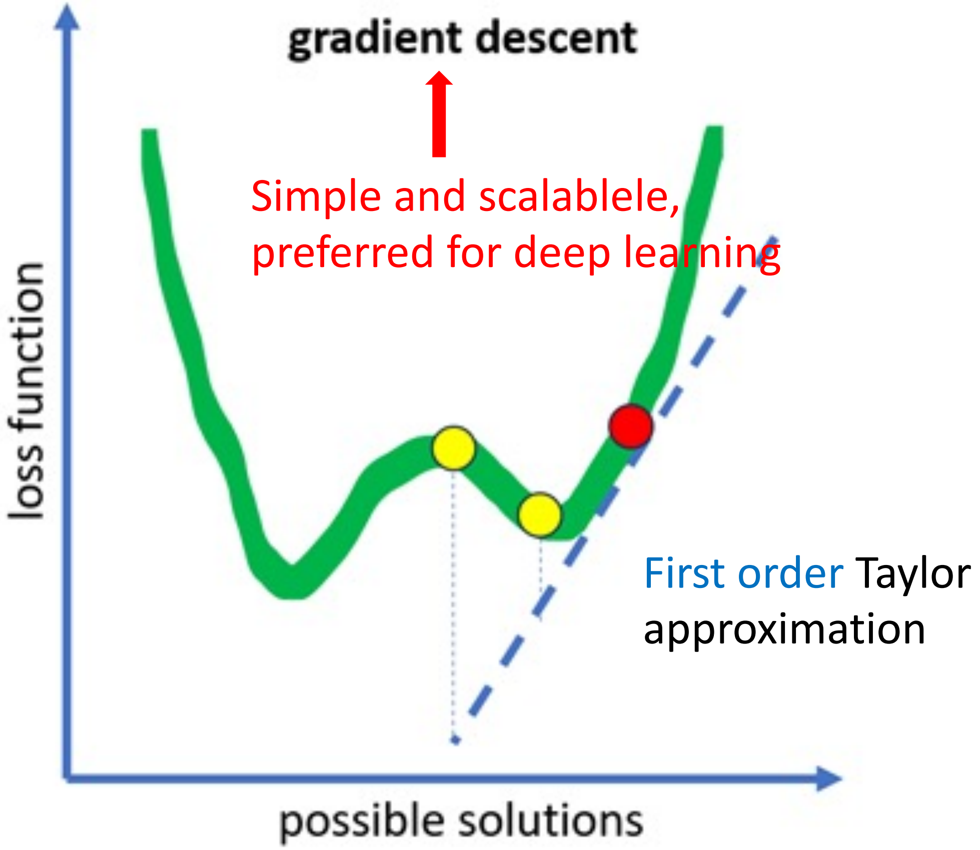
- Loss function
  - ℓ_2-loss
  - Maximum likelihood (KL divergence)

$$\max \Pi_{i=1}^{N} p(y_i|x_i,w) \Leftrightarrow \min -\sum_{i=1}^{N} \log p(y_i|x_i,w) \ \ (\textit{cross entropy})$$

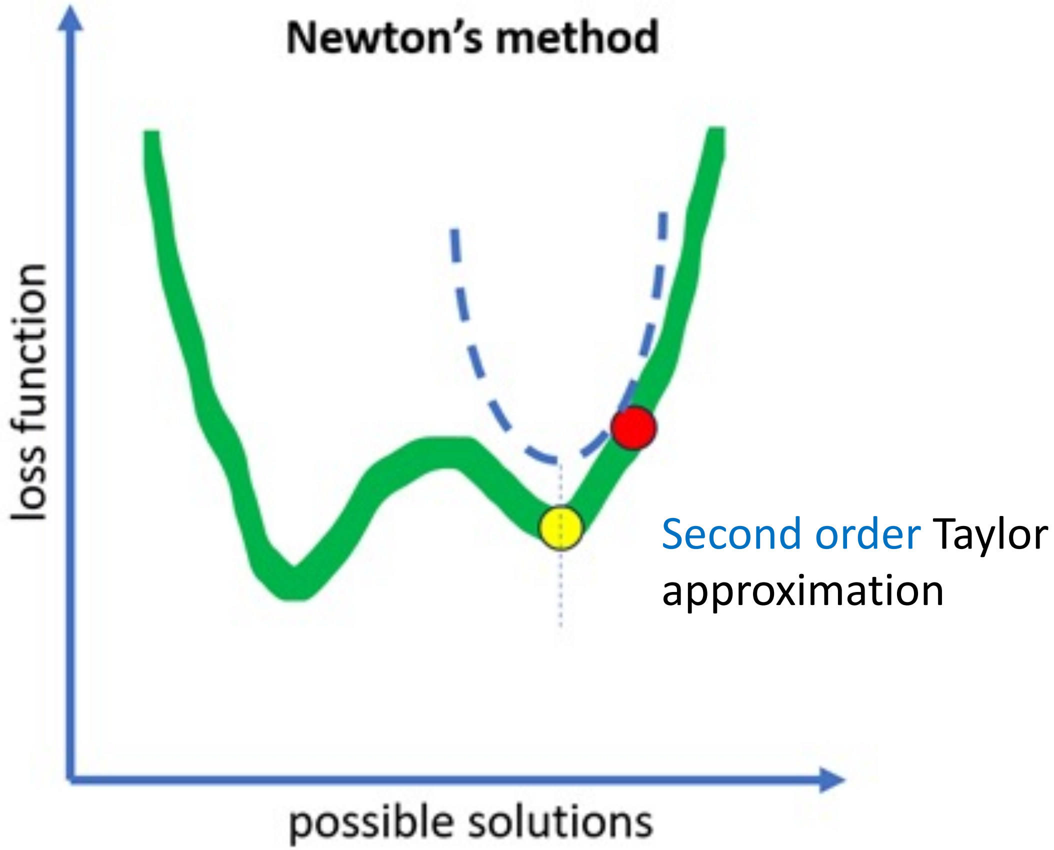
$$-\sum_{i=1}^{N} \log p(y_i|x_i,w) = \sum_{i=1}^{N} -y_i \log \mu_i - (1-y_i)\log(1-\mu_i)\,,\mu_i = \frac{1}{1+e^{-w^T x}}$$

- Optimization: smooth and convex loss function; gradient descent, newton methods…

Optimization



lower convergence, low memory
requirements and computational cost



faster convergence, high memory
requirements and computational cost

Summary

- Machine Learning: learning from experience (or data).
- Supervised learning is to learn the mapping from x to y given paired training dataset $\{(x_i, y_i)\}$. Key steps include
  1. Define the hypothesis space
  2. Define the empirical loss
  3. Find the best approximation under the given empirical loss