Introduction to Machine Learning

Introduction

SHENG-JYH WANG

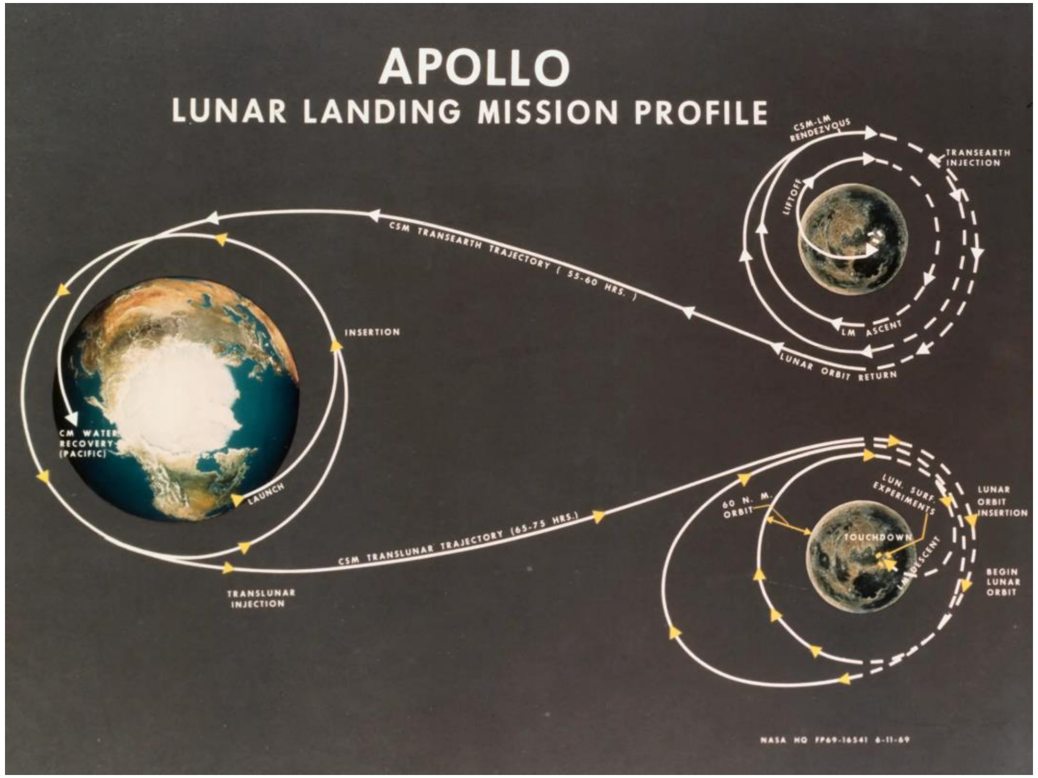
NATIONAL YANG MING CHIAO TUNG UNIVERSITY, TAIWAN

SPRING, 2024

Conventional Techniques

Apollo 11 Mission in 1969

- Knowledge based
- Model based



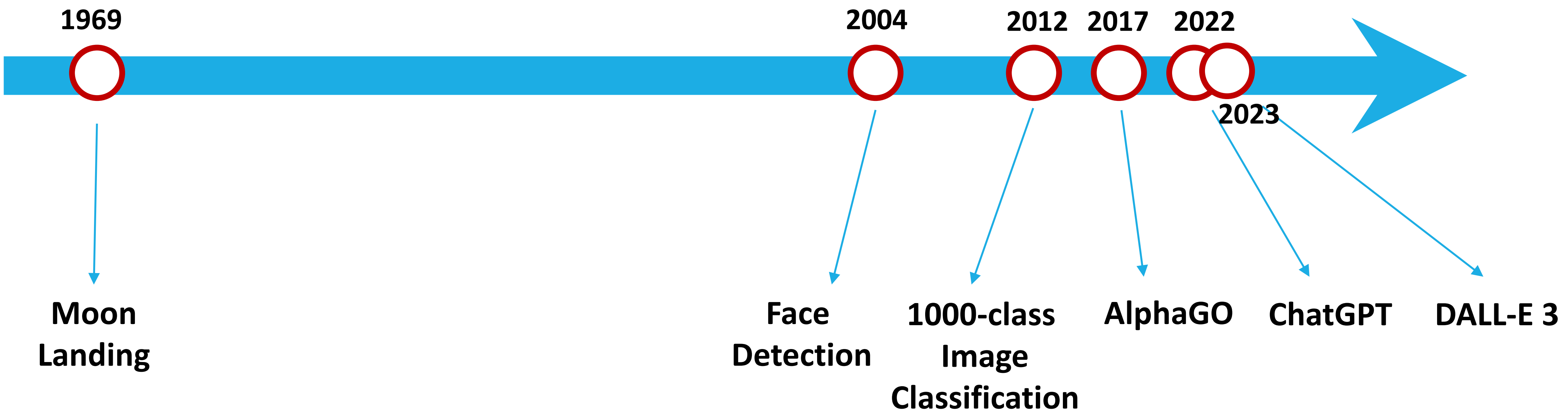
https://www.history.com/news/apollo-11-moon-landing-timeline

Problems that are Difficult to Model

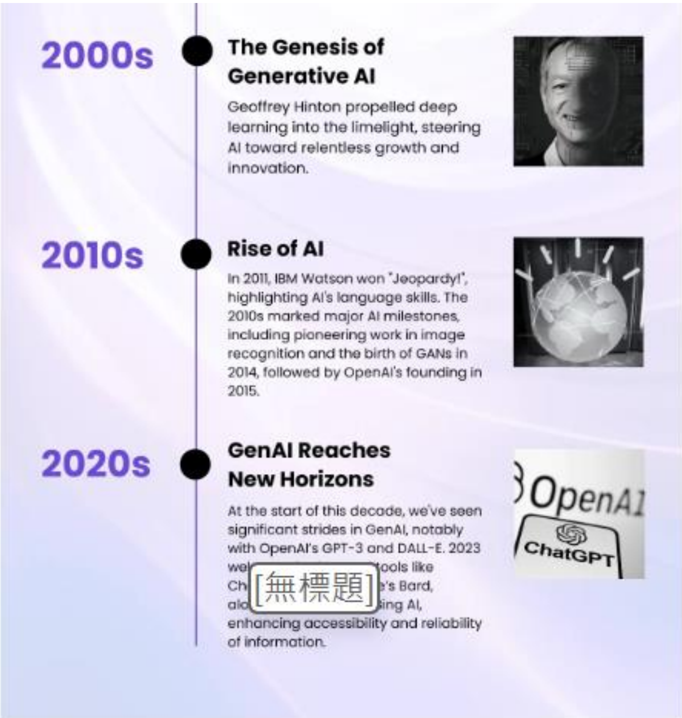
Face Detection

Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57 (2004): 137-154.

Recent History of Human Technologies

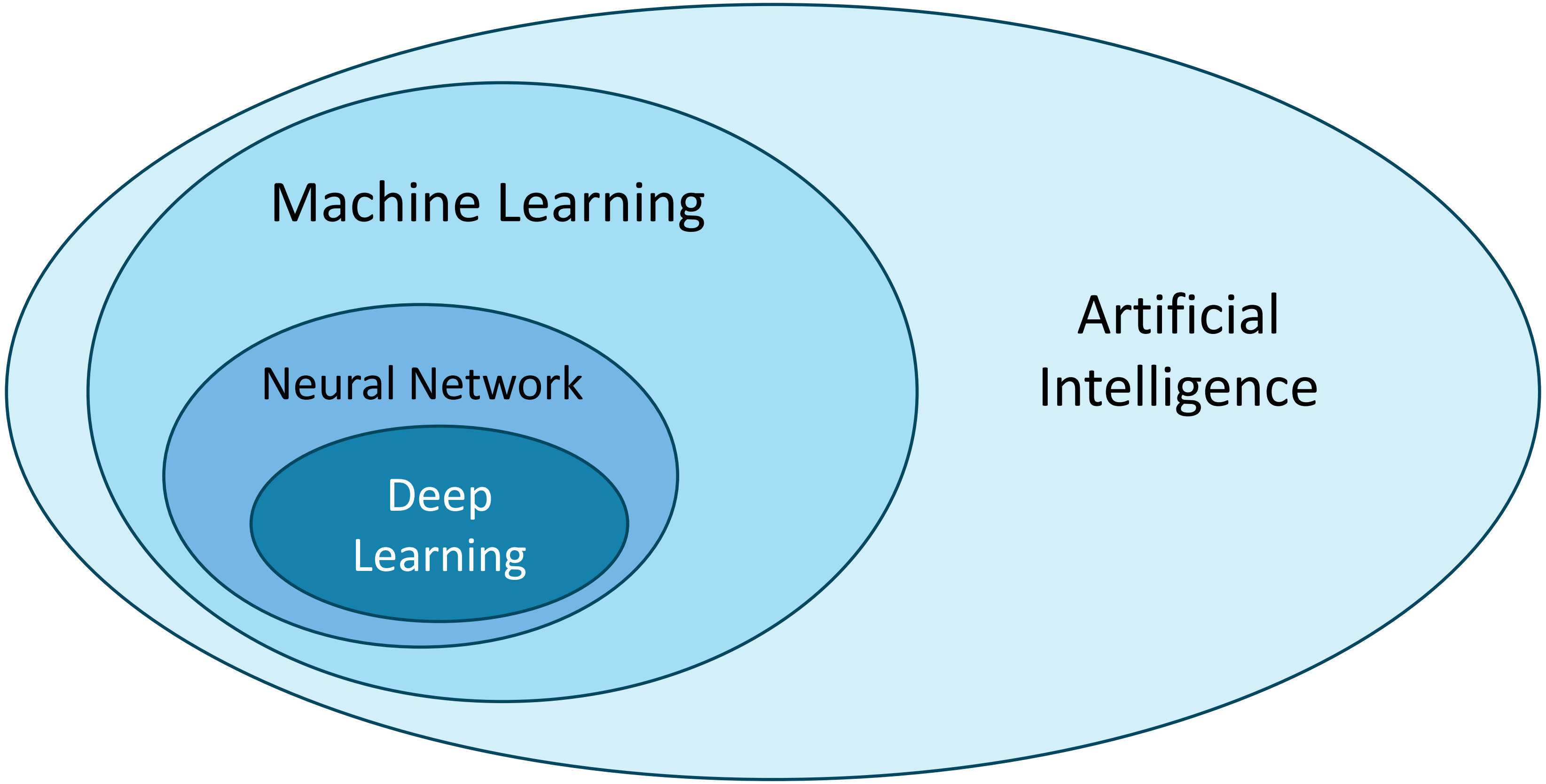


History of AI



https://www.calls9.com/blogs/the-history-of-ai-a-timeline-from-1940-to-2023

Scope of Machine Learning



What is machine learning?

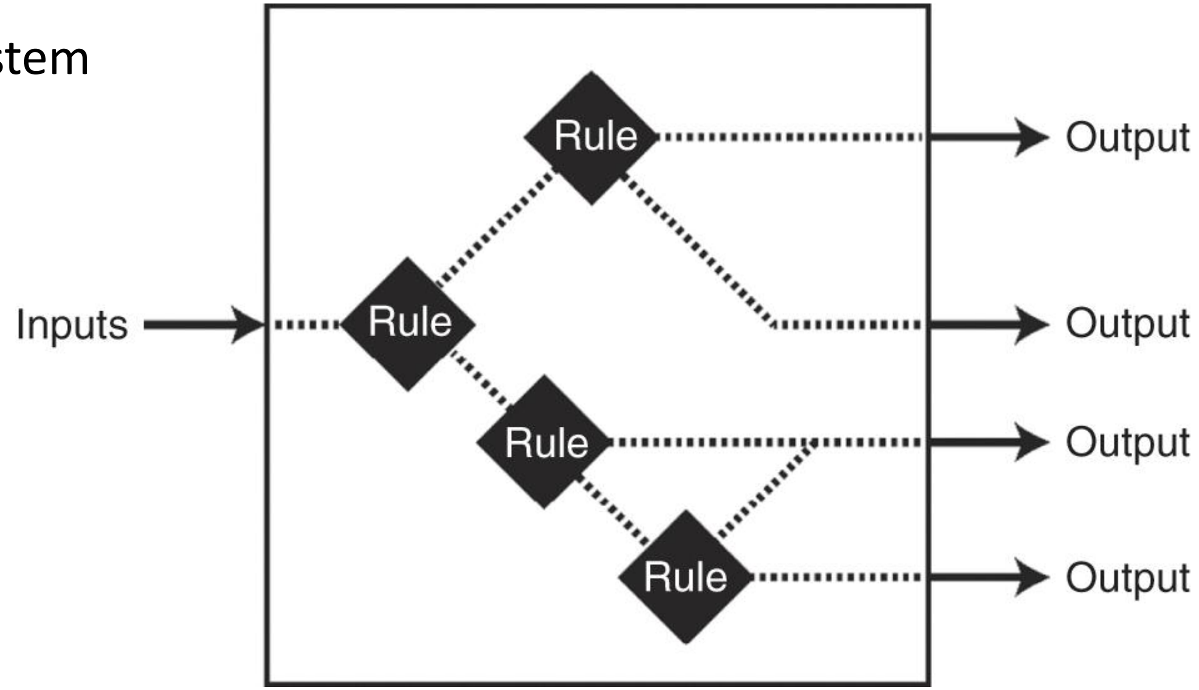
To design and develop algorithms that allow computers to evolve behaviors based on **empirical data**.

- ✓ Try to explore certain patterns or regularities.
- ✓ Learn models from the given data.
- ✓ Based on the given data, the learner produces a useful output in new cases.

Introduction

Knowledge-based AI

Rule-based Expert System

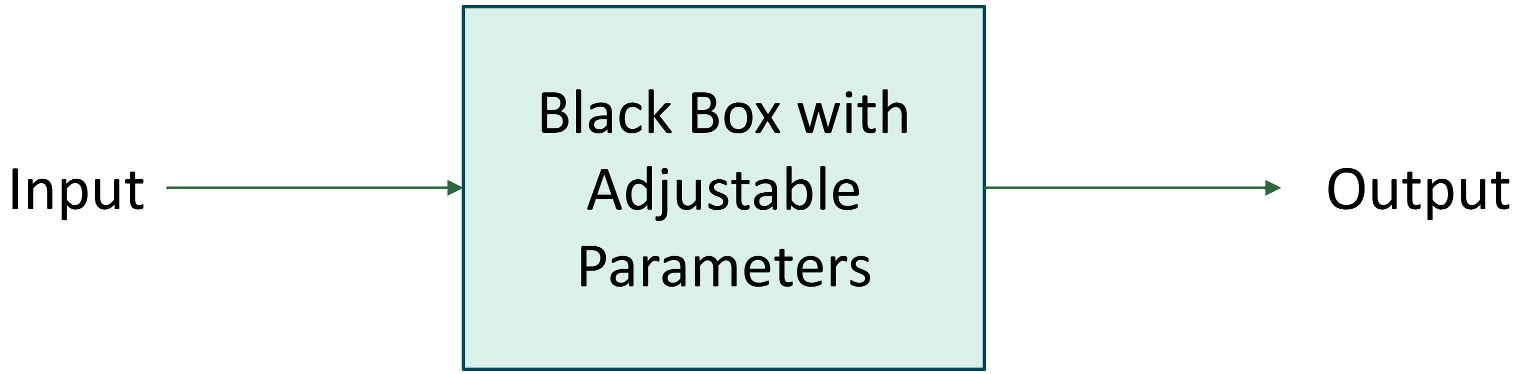


Ref: https://aneskey.com/20-epilogue-artificial-intelligence-methods/

Introduction

Data-Driven AI

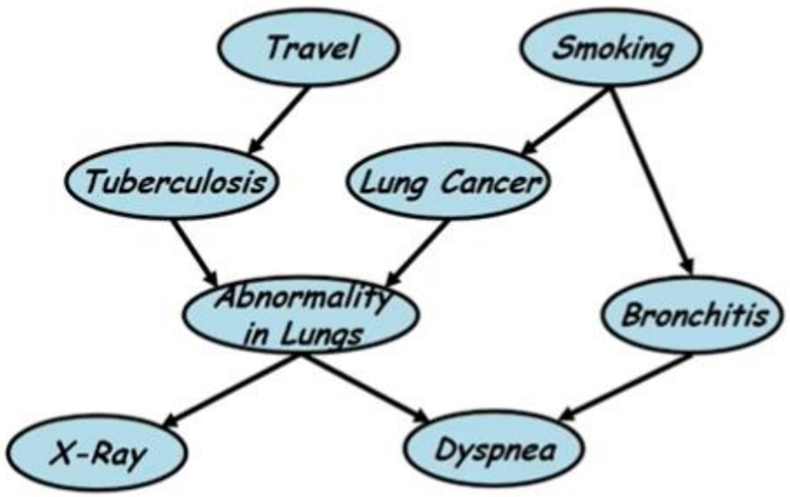
Black-Box Statistical Model
(e.g., neural networks, support vector machine)



Introduction

Data-Driven AI

Integration of Domain Knowledge and Statistical Learning
(e.g., Bayesian framework, probabilistic graphical models)

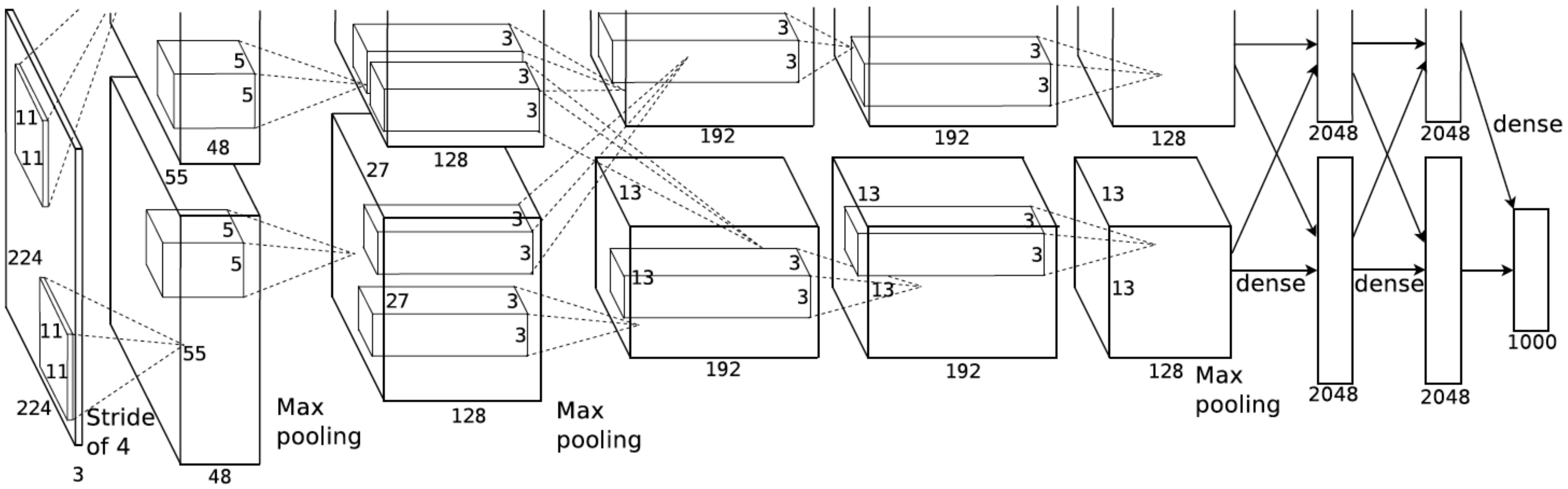


Ref: https://www.youtube.com/watch?v=WKAcfXUSaeA

Introduction

Data-Driven AI

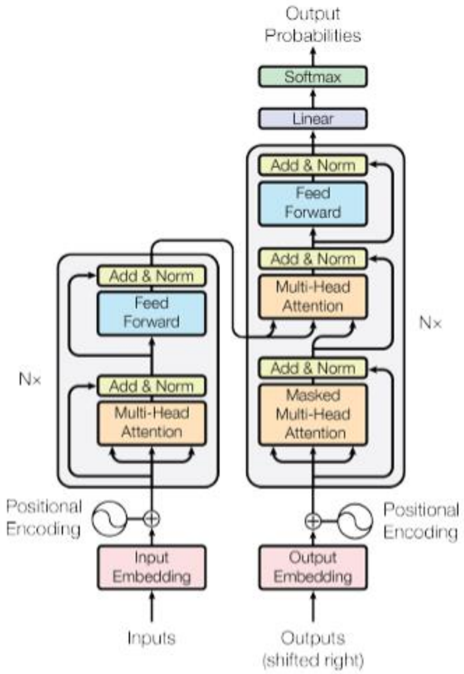
Deep Learning Models



Introduction

Data-Driven AI

Combination of Deep Learning Modules and Attention Mechanism



Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Introduction

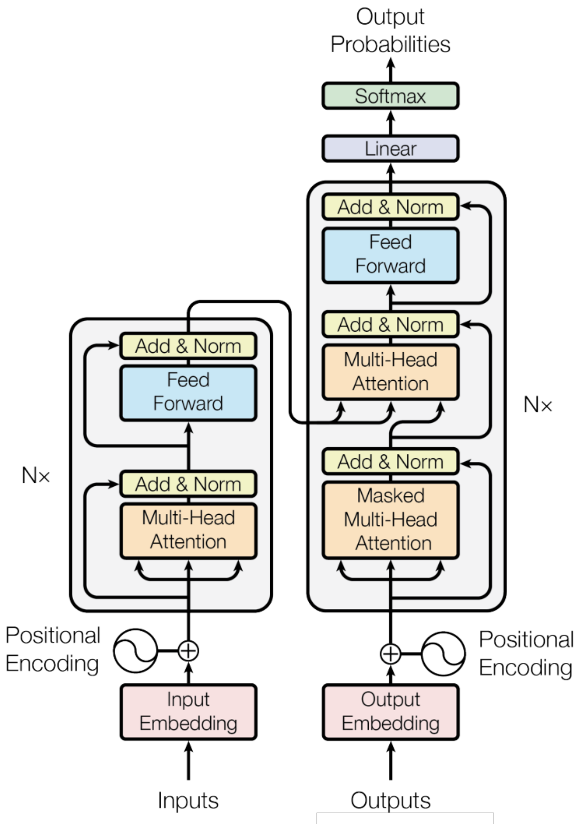
Data-Driven AI

Foundation Models

BERT

Encoder

Bidirectional Encoder Representations from Transformers



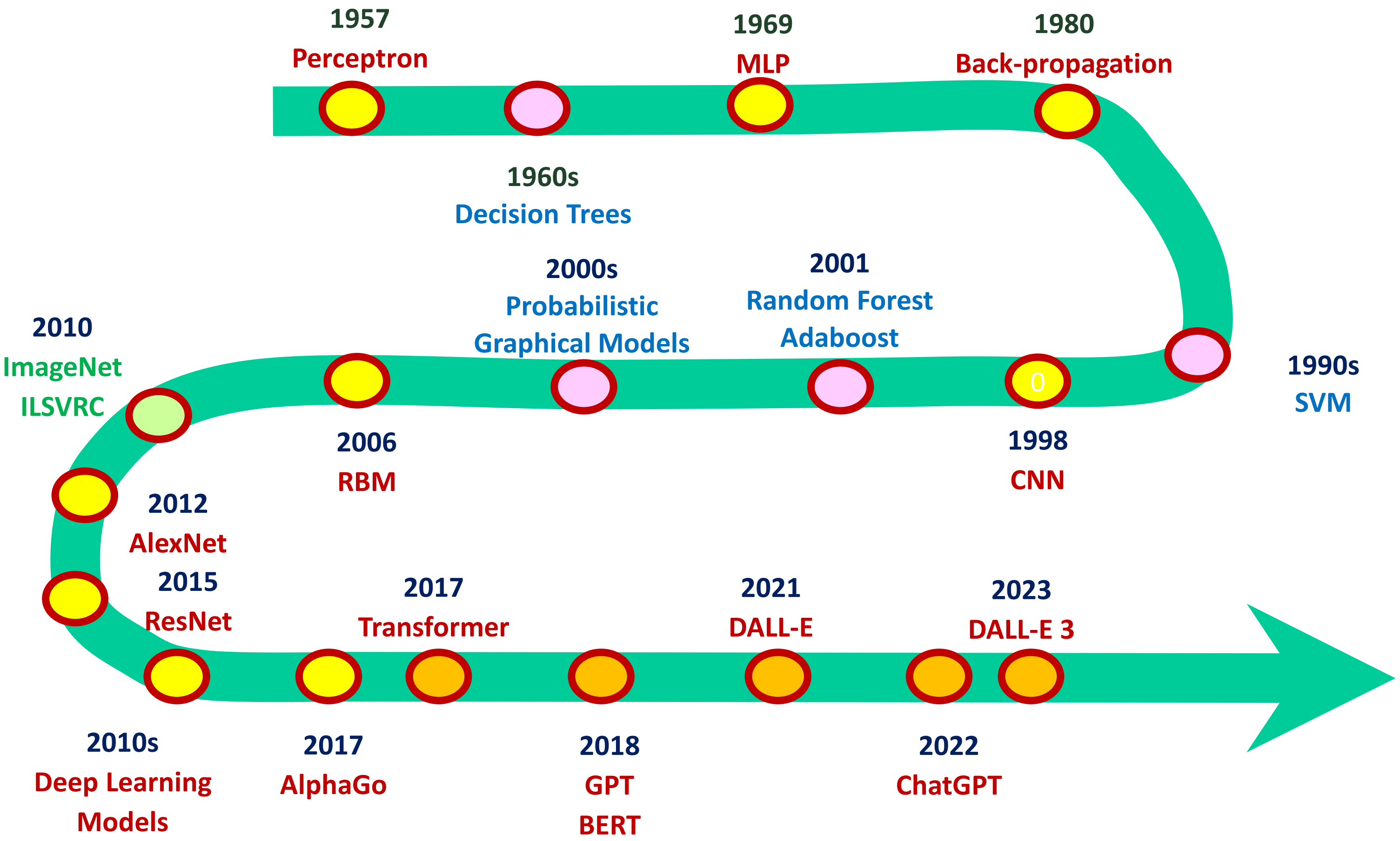
GPT

Decoder

Generative Pre-trained Transformers

https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/

History of Machine Learning

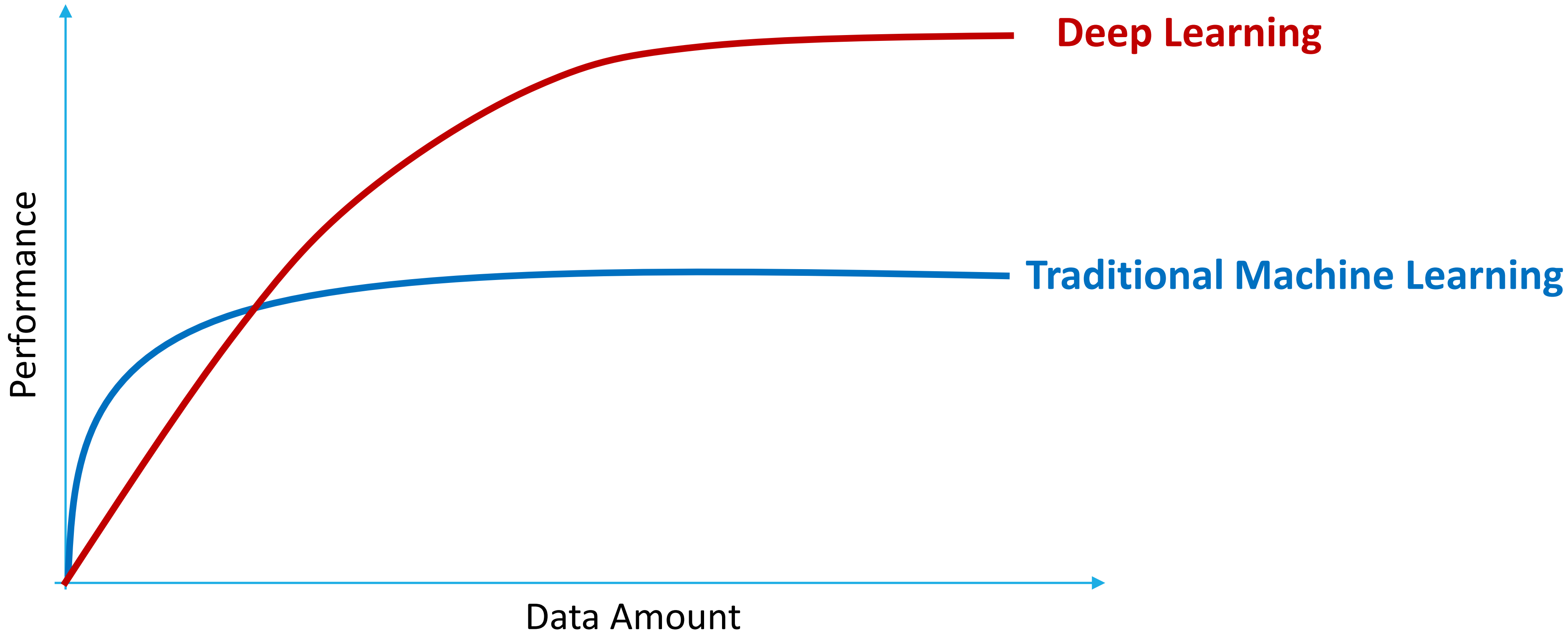


History of Machine Learning

	Rules ~1950	Machine learning ~1960	Representation learning ~2000	Deep learning ~2012	Foundation models ~2017
Decision Rules	Hand-designed	Learnt	Learnt	Learnt	Learnt
Features	Hand-designed	Hand-designed	Learnt (Simple features)	Learnt (Hierarchy of features)	Learnt (Hierarchy of features)
Learning	None	Supervised	Supervised	Supervised	Self-supervised + in-context learning
Data	None	Few labeled-data	Mid-sized labeled-data	Large labeled-data	Very large unlabeled data + small labeled data
Adaptability	None	None	Little	Medium	Large

Schneider, Johannes. "Foundation models in brief: A historical, socio-technical focus." arXiv preprint arXiv:2212.08967 (2022).

Performance vs Data Amount



Major issues in machine learning?

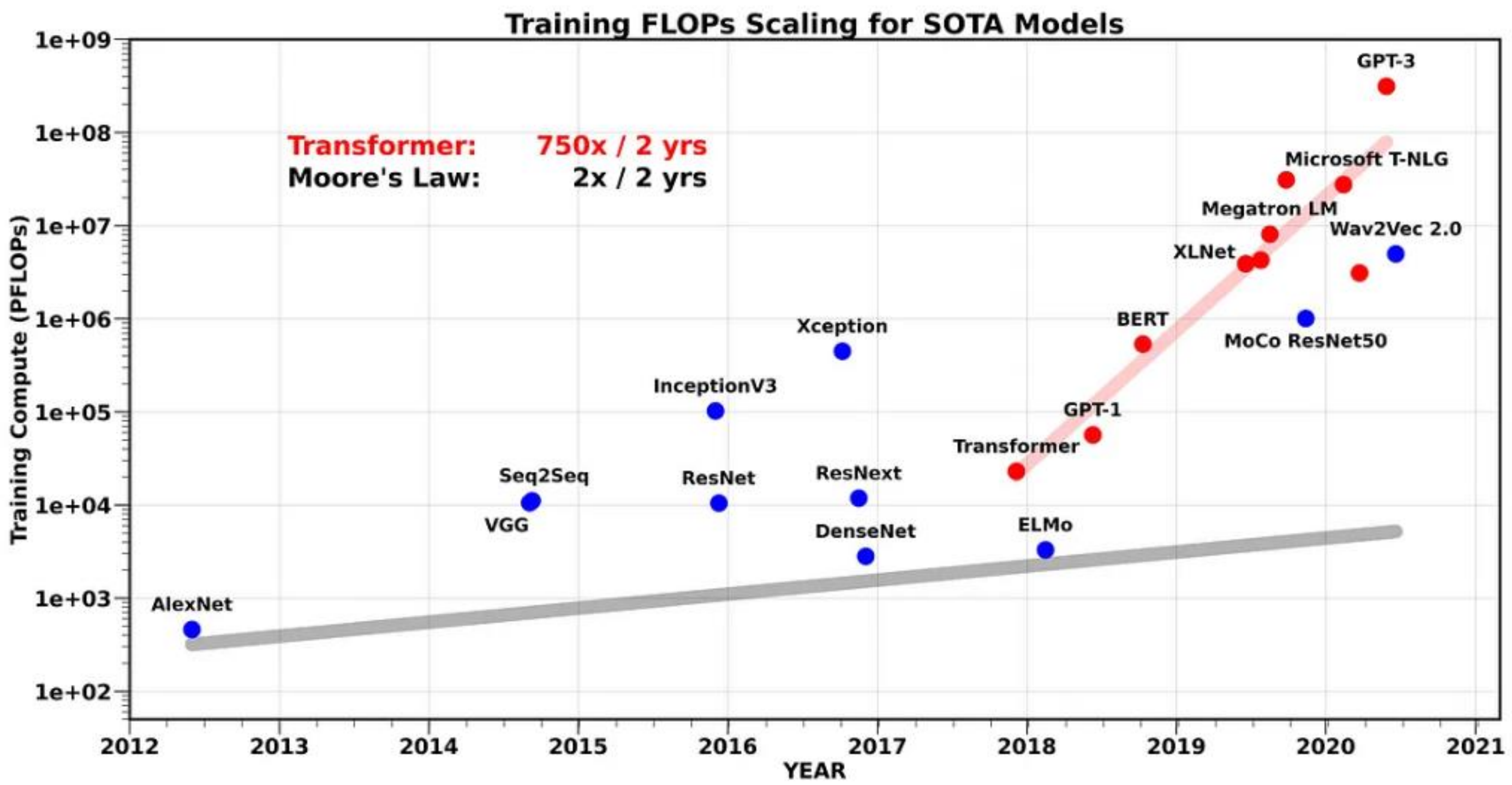
In training,

- ✓ We need efficient algorithms to build the models, to process the data, and to store the data.
- ✓ For certain problems, we need to collect a huge amount of data.

After training

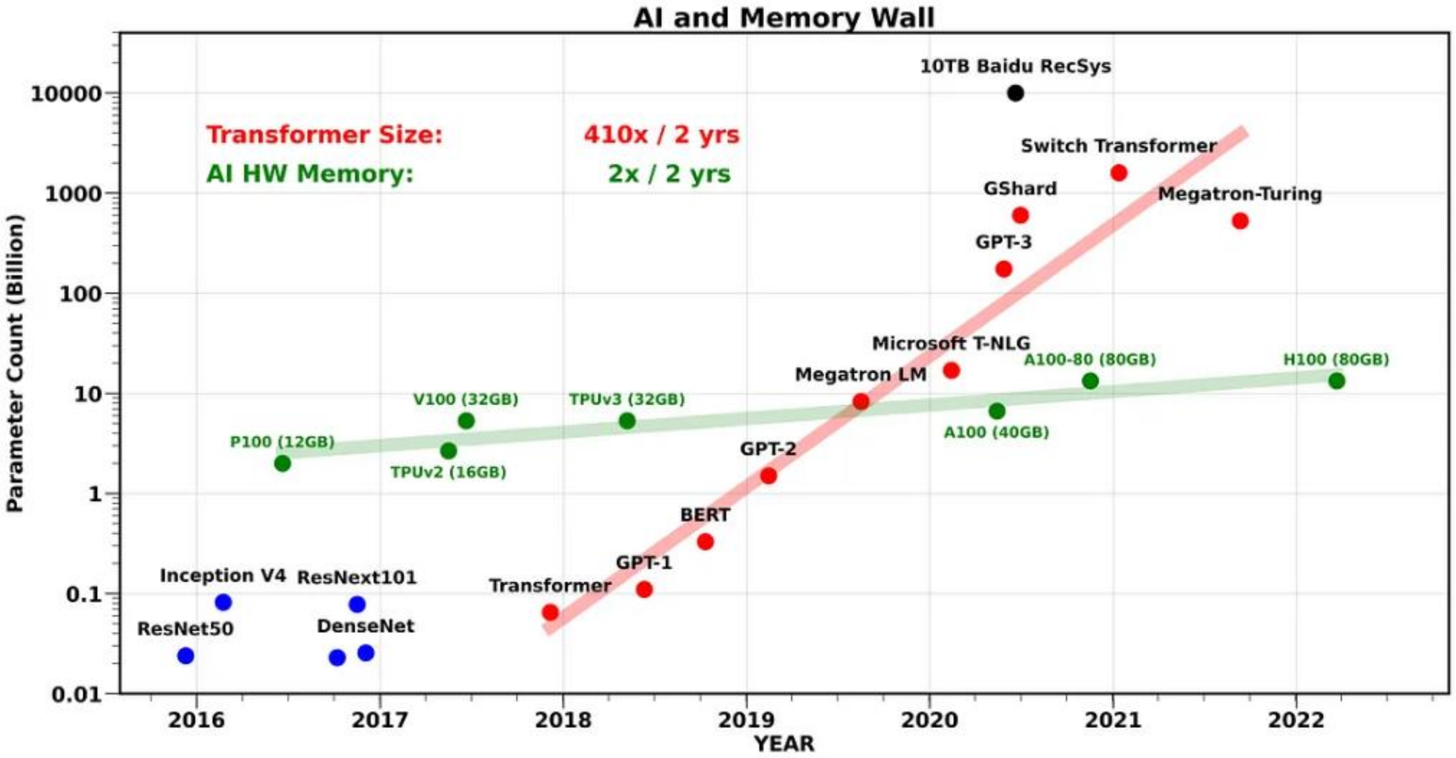
- ✓ We need efficient algorithms for **inference** or **generalization**.

Trend of Required Computations



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Trend of Model Size



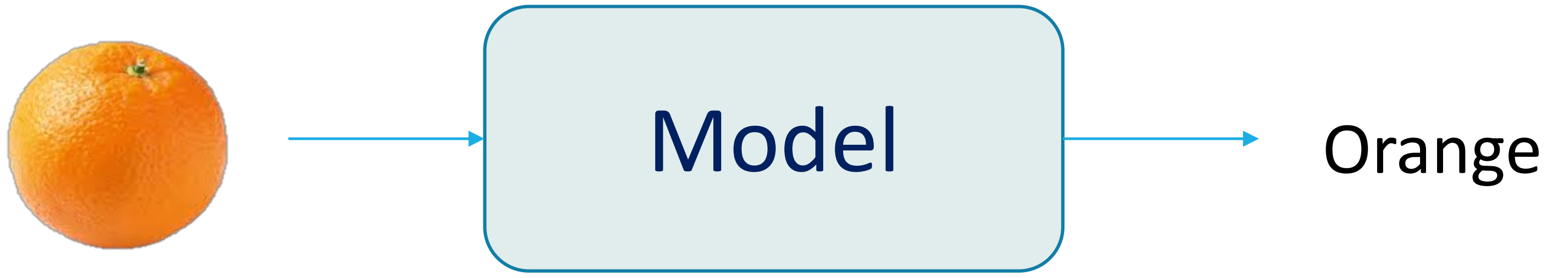
https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Major Topics of Machine Learning (1/5)

- Supervised Learning
- Unsupervised Learning
  - ✓ Self-supervised Learning
- Semi-supervised Learning
- Reinforcement Learning

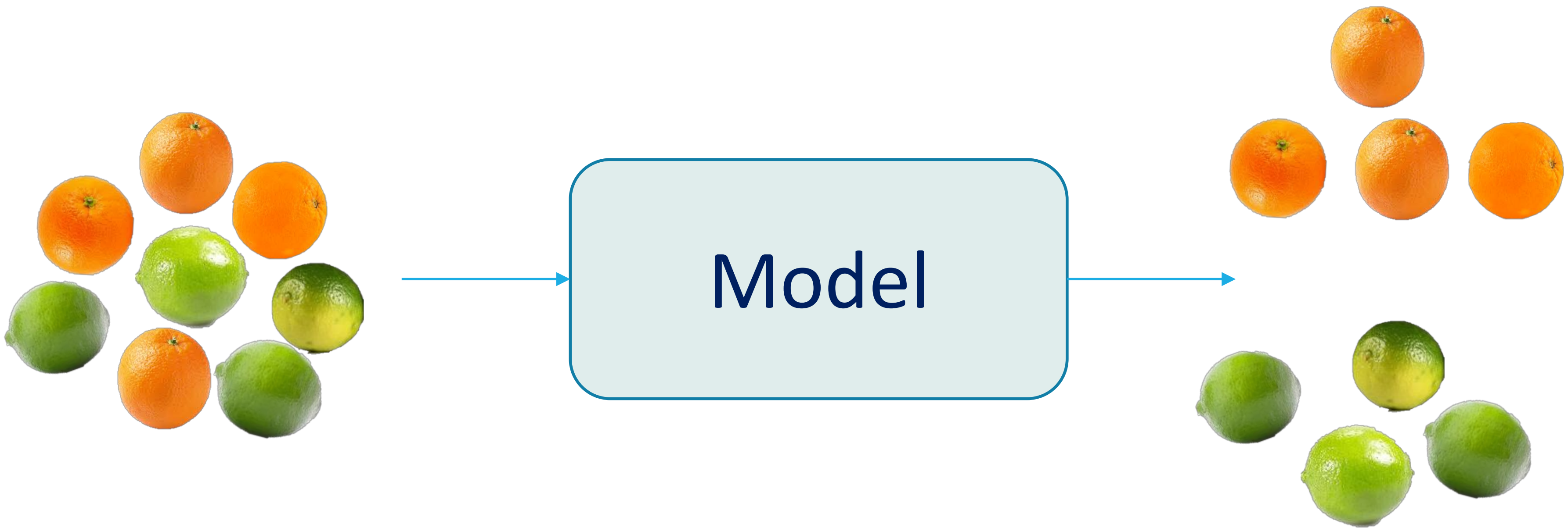
Major Topics of Machine Learning (2/5)

- ***Supervised learning***: to learn a model to classify data or predict outcomes.



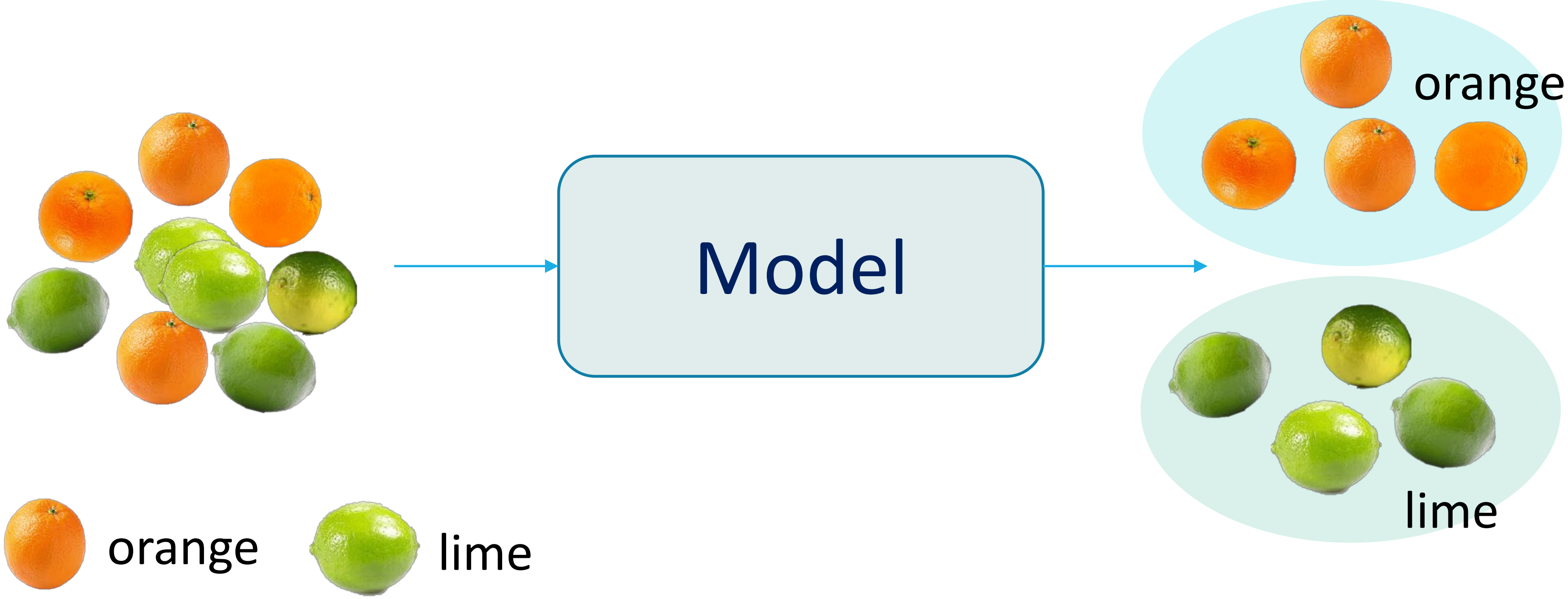
Major Topics of Machine Learning (3/5)

- ***Unsupervised learning***: to analyze and cluster unlabeled datasets



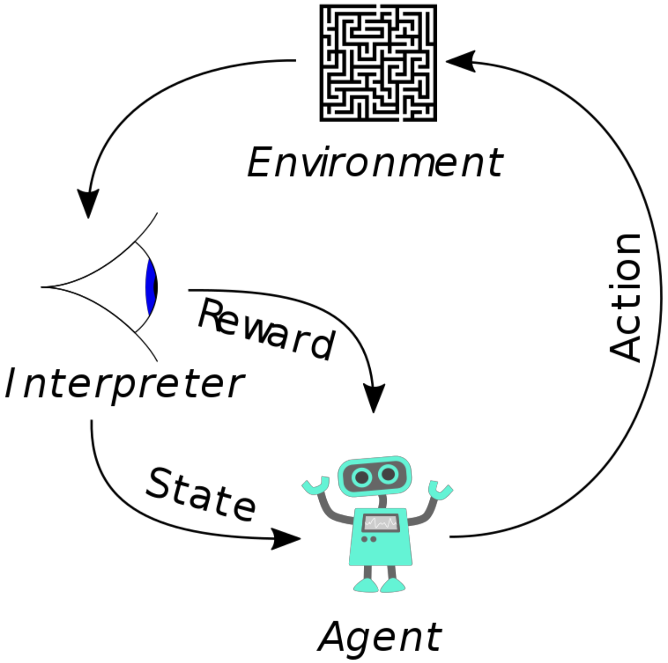
Major Topics of Machine Learning (4/5)

- ***Semi-supervised learning***: use a small amount of labeled data and a large amount of unlabeled data to label all the unlabeled data.



Major Topics of Machine Learning (5/5)

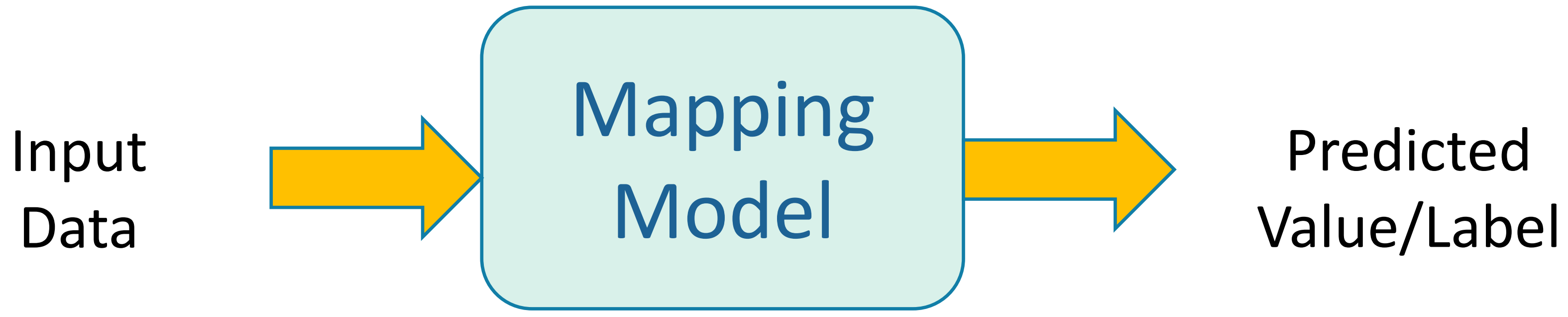
- ***Reinforcement Learning***: learn how to take actions in an environment in order to maximize the cumulative reward.



Supervised Learning (1/7)

- ***Training data***: examples of the input vectors along with their corresponding target vectors.
- ***Types of supervised learning***
  - ✓ ***Classification***: assign each input vector to one of a finite number of discrete categories.
  - ✓ ***Regression:*** assign each input vector to one or more continuous variables.
- ***Methods***: Linear Regression, Linear Classification, Neural Networks, Support Vector Machine, Ensemble Learning, ……..

Supervised Learning (2/7)



Supervised Learning (3/7)

Regression

A large (or infinite) number of values

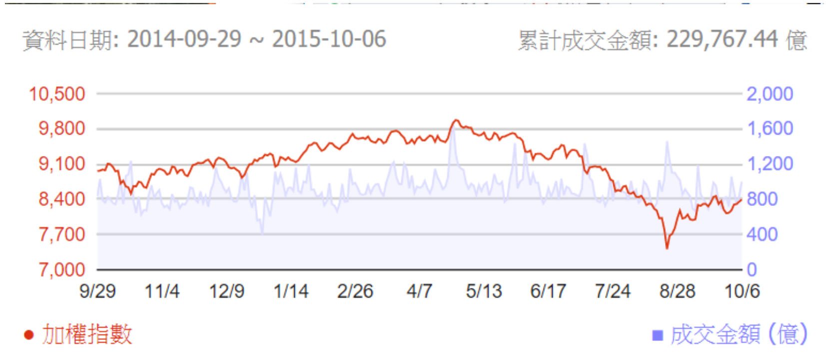
Input Data

x

Mapping Model

y

Predicted Value



Estimated TAIEX on 11/5

Supervised Learning (4/7)

Classification

A finite number of labels

Input Data $\mathbf{x}$

Mapping Model

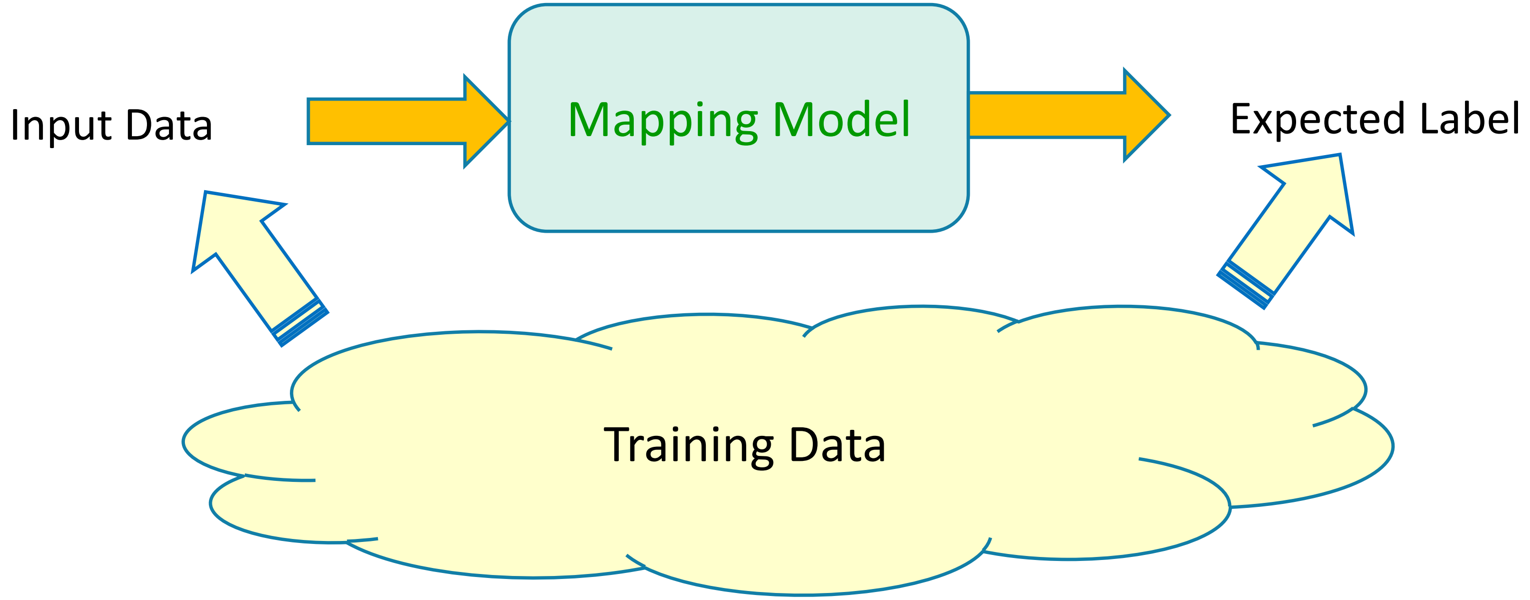
$\mathbf{C_k}$

Predicted Label

Flower

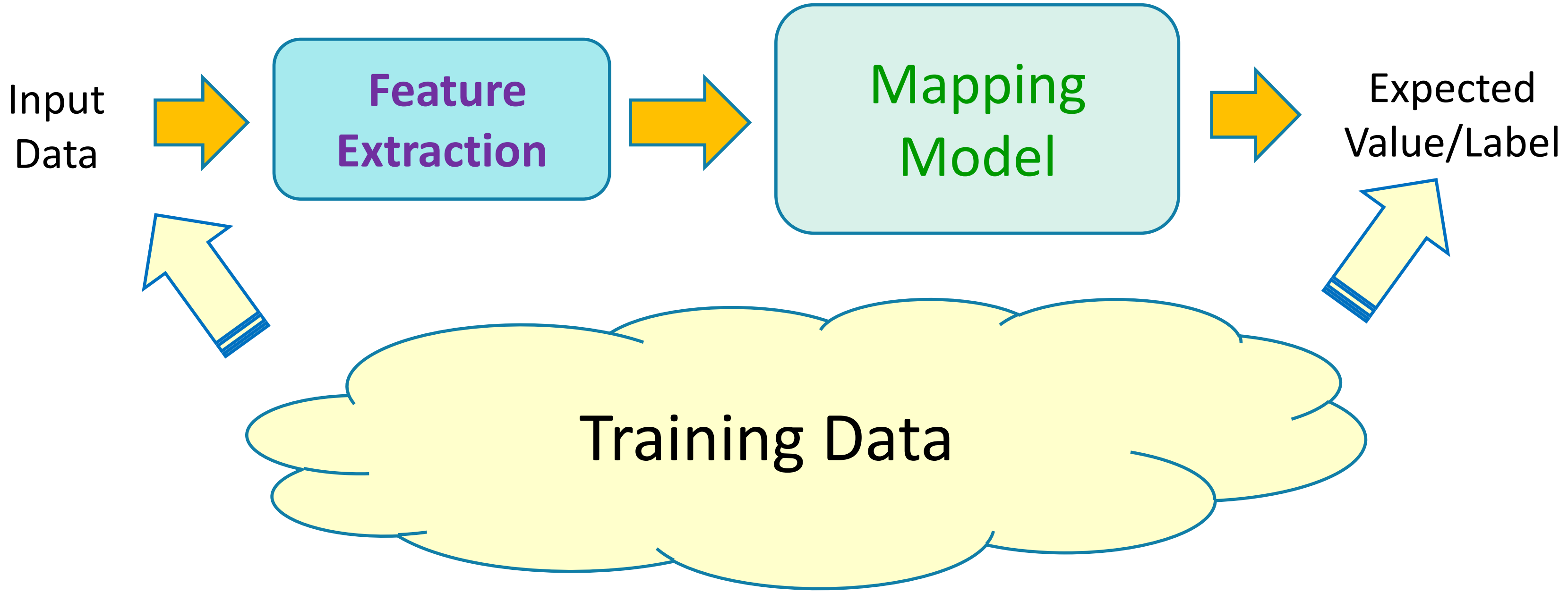
Supervised Learning (5/7)

Model Training: use a set of N digits {$\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$}, sometimes together with their target vectors {t_1, t_2, …, t_N}, to learn a proper model for the problem.



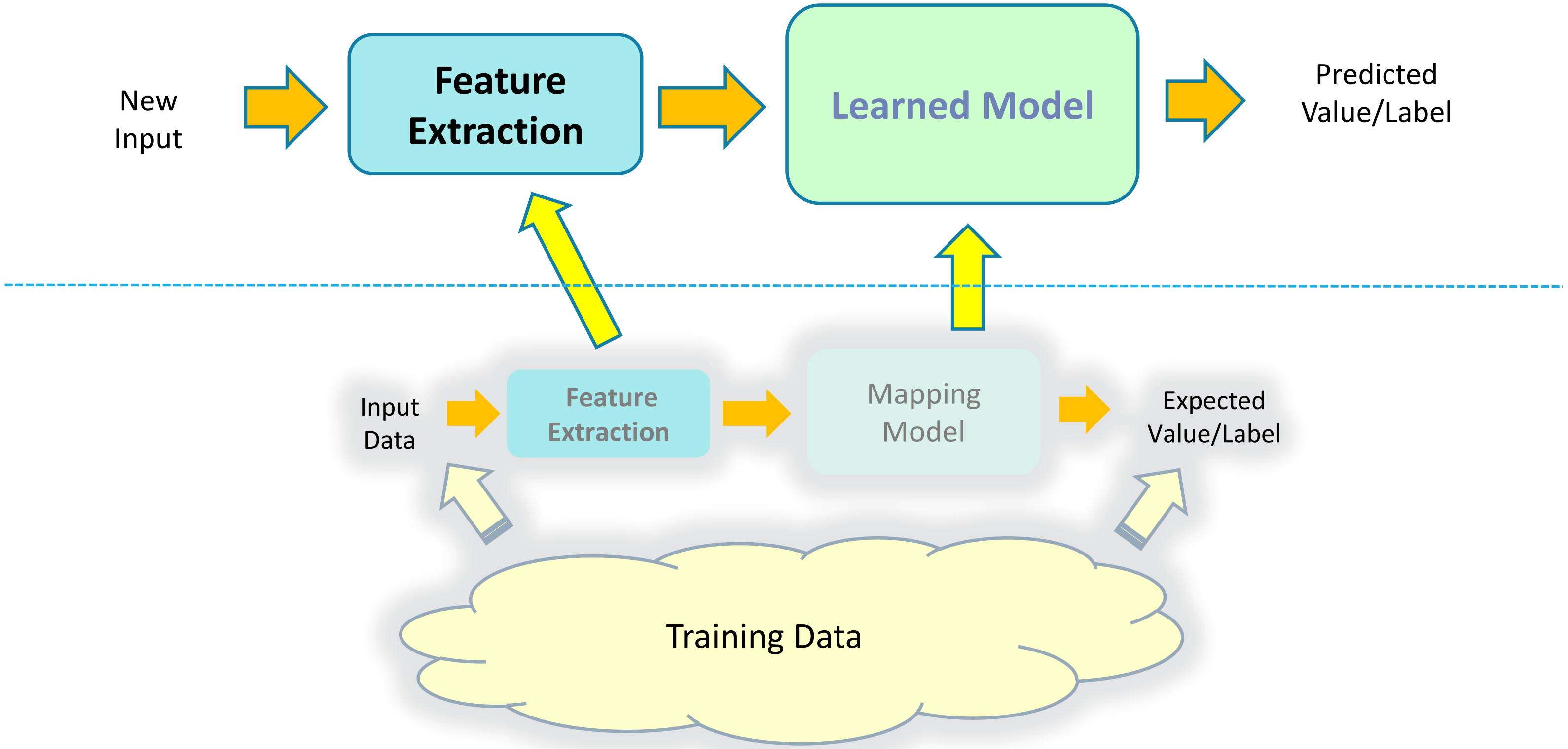
Supervised Learning (6/7)

Feature Extraction: The original input variables are usually transformed into some new space of variables, where the problem can be handled in an easier or more efficient way.



Supervised Learning (7/7)

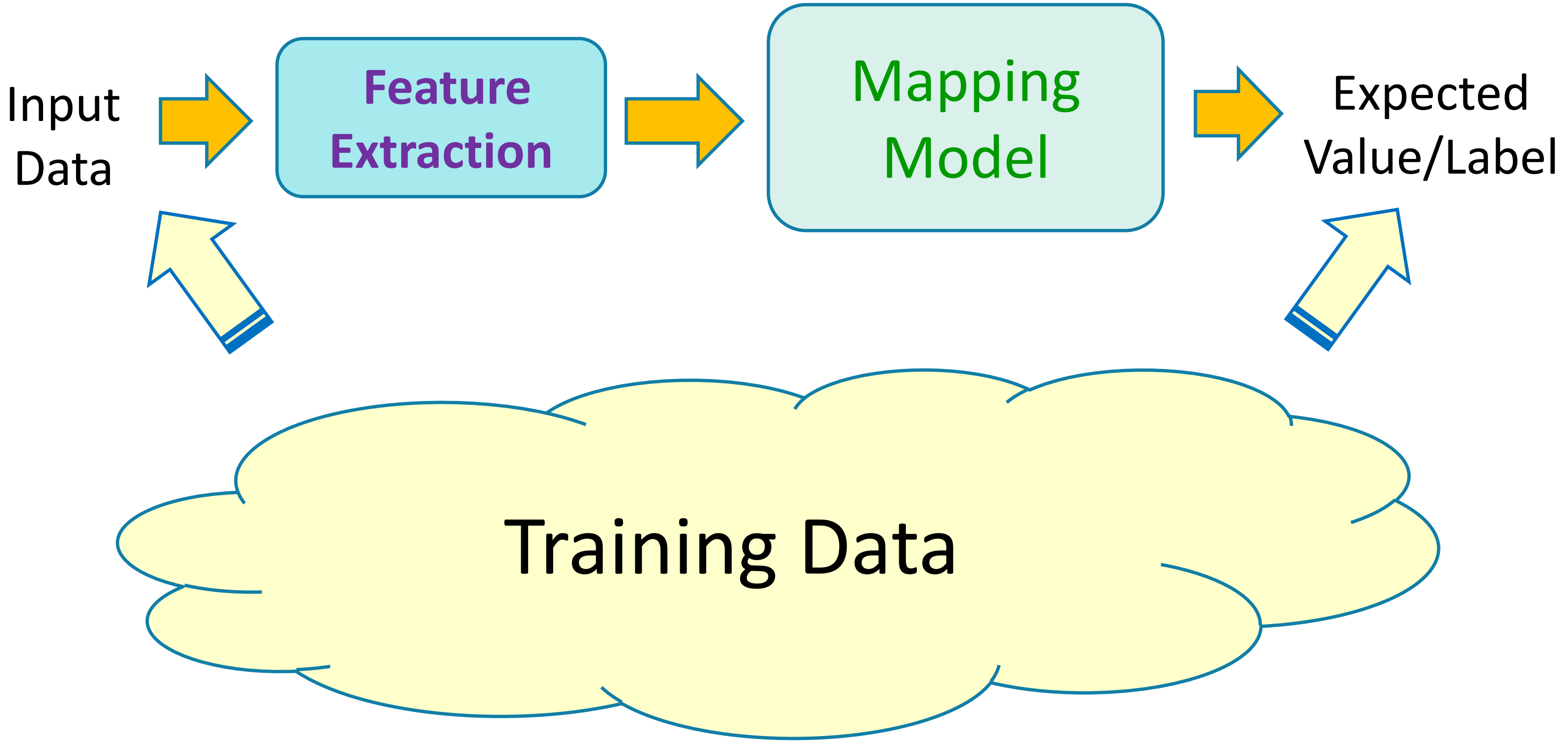
Generalization: the ability to categorize correctly new examples that differ from those used for training.



Preparation of Training Data

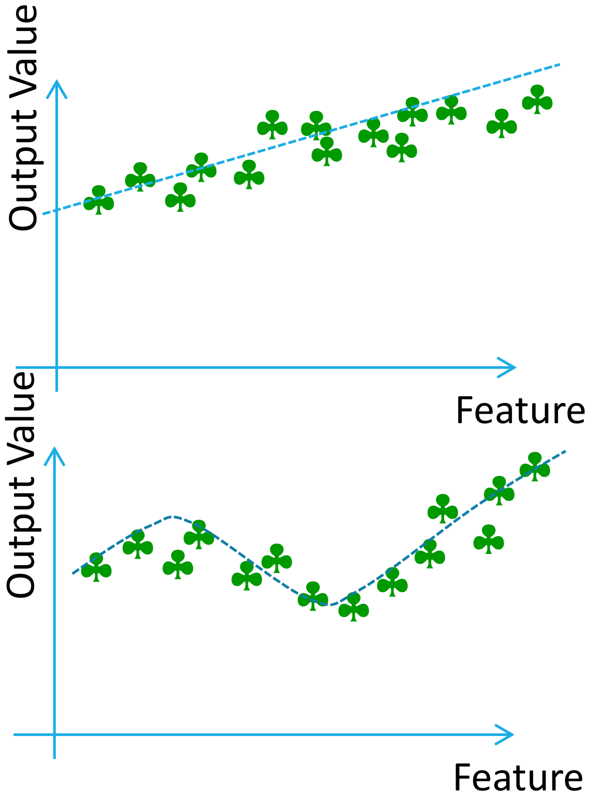
	Input					Expected Label
Palm					…	C_1
Flower					…	C_2
						⋮
Else					…	C_{N+1}

Feature Extraction (1/4)

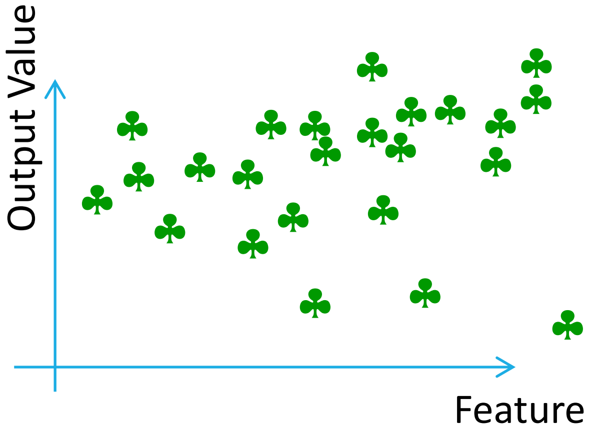


Feature Extraction (2/4)

Regression



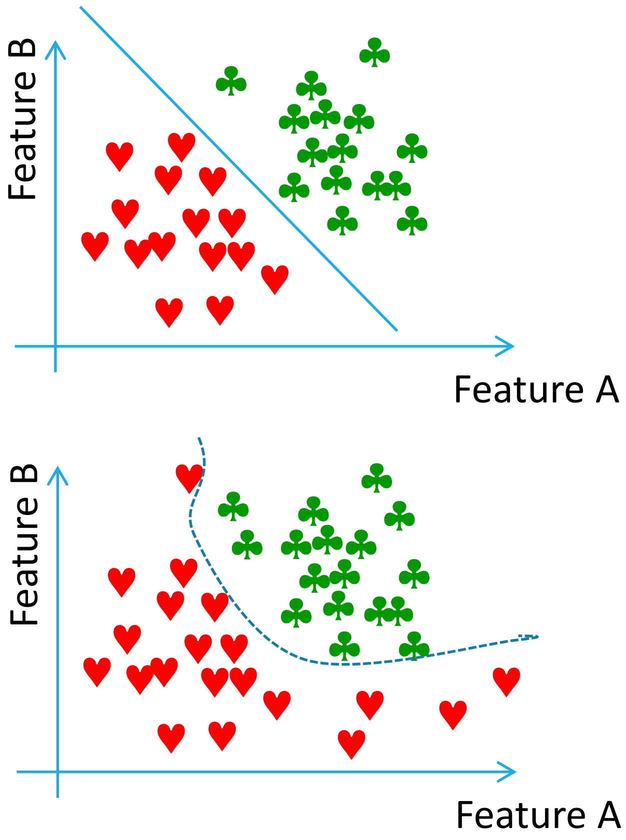
Good Features



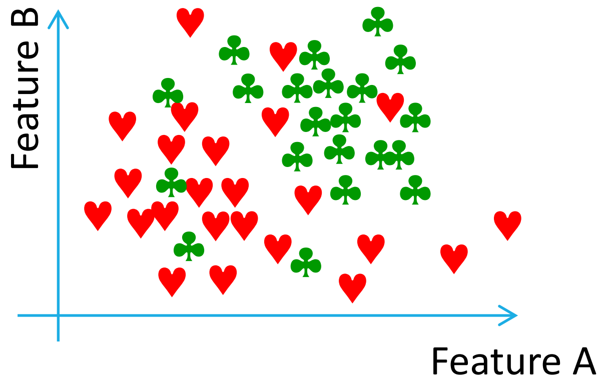
Bad Features

Feature Extraction (3/4)

Classification



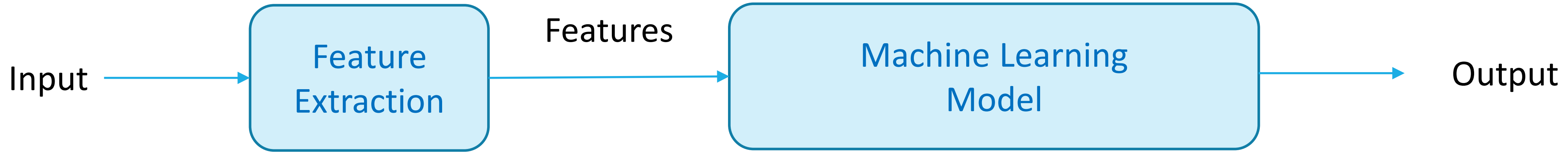
Good Feature



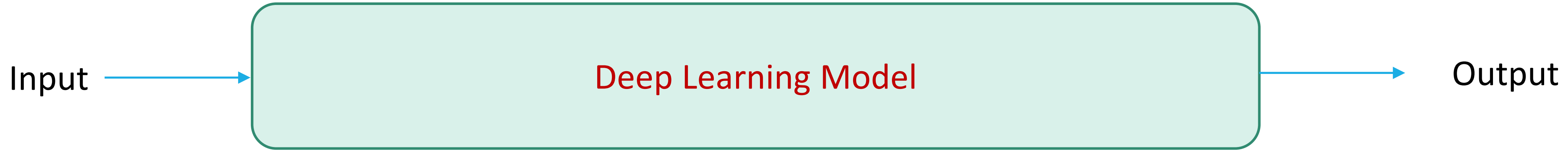
Bad Feature

Feature Extraction (4/4)

Hand-crafted Features

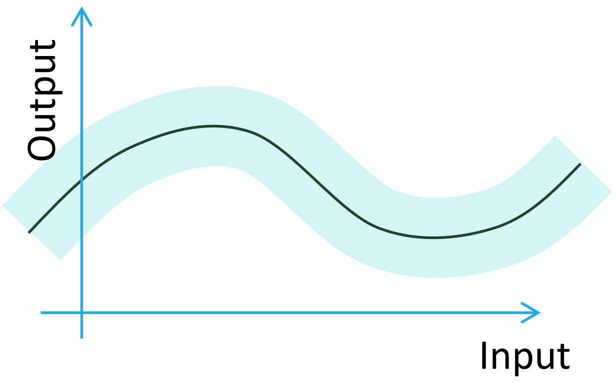


Learned Features

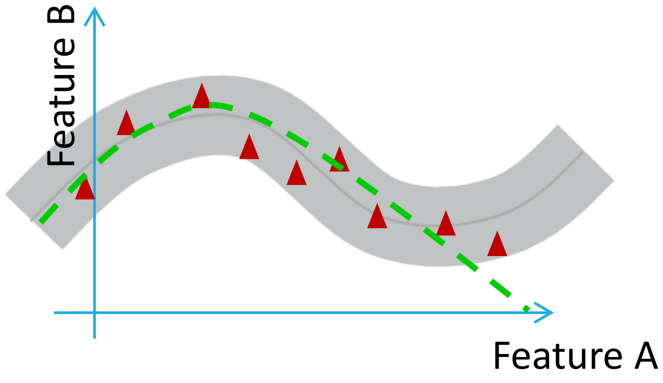


Training Data Distribution vs Actual Distribution

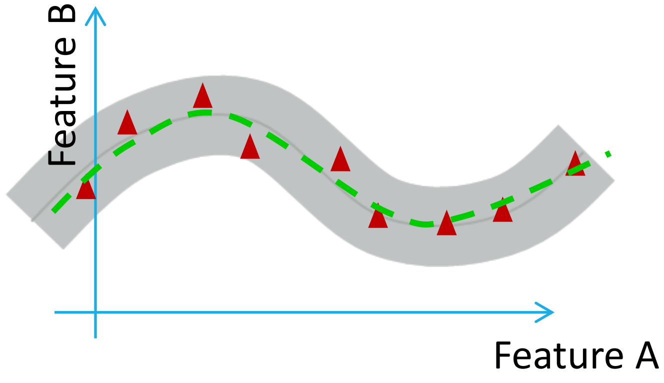
Example: Regression



Actual Distribution



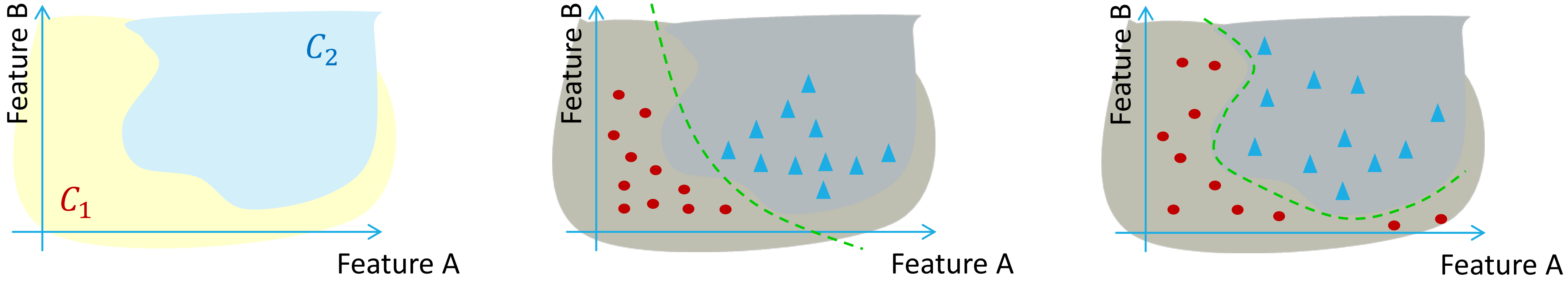
Less Proper Training Samples



More proper Training Samples

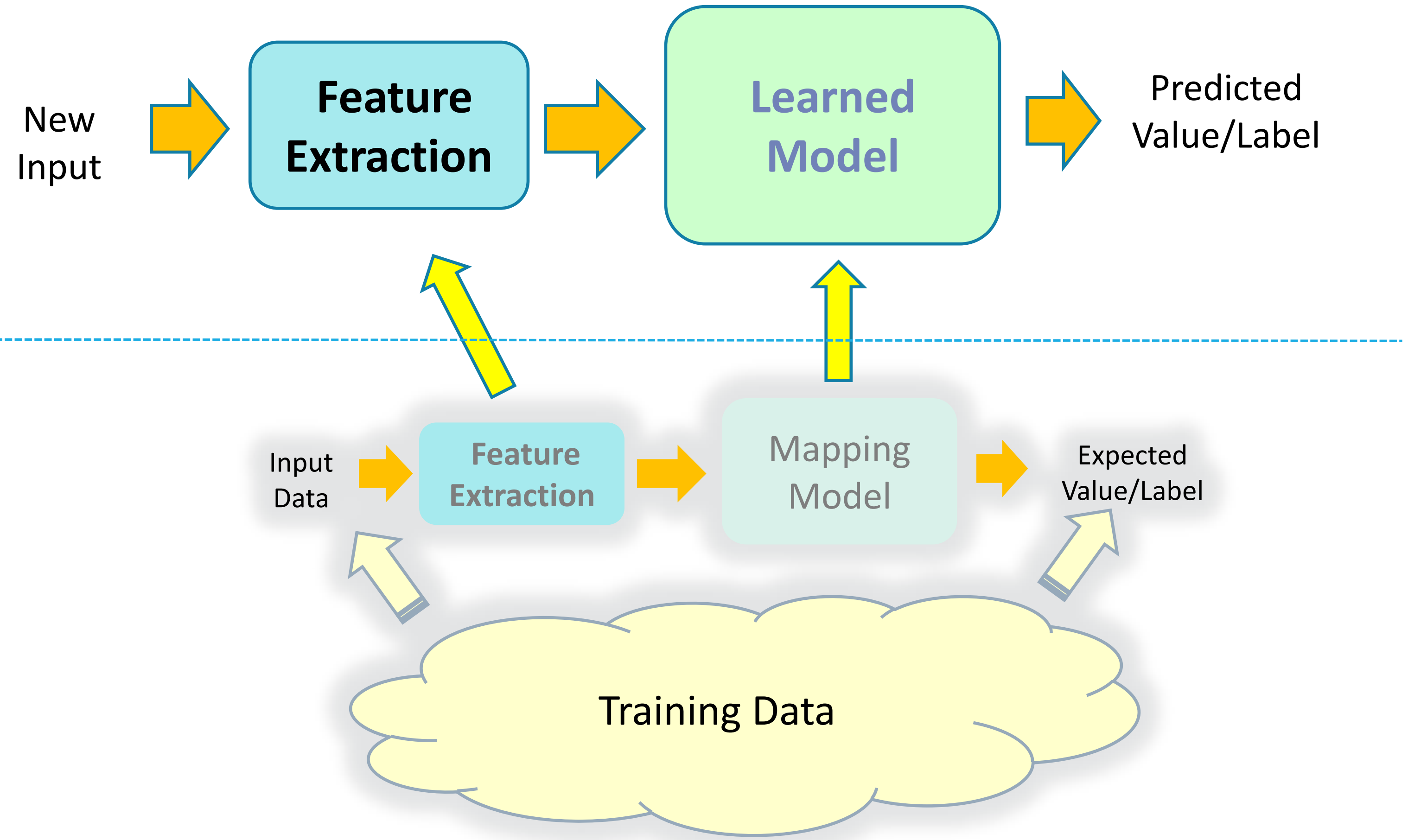
Training Data Distribution vs Actual Distribution

Example: Binary Classification



Actual Distribution **Less Proper Training Samples** **More proper Training Samples**

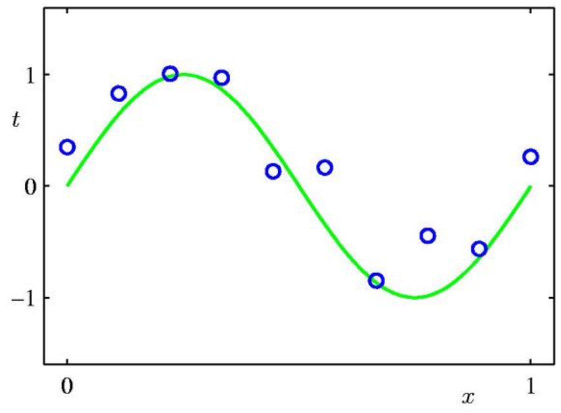
Generalization (1/8)



Generalization (2/8)

Example: Polynomial Curve Fitting

Training data:



Goal: to exploit this training set in order to make predictions of the value $\hat{t}$ of the target variable for some new value $\hat{x}$ of the input variable.

Generalization (3/8)

Fit the data using a polynomial function of the form:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

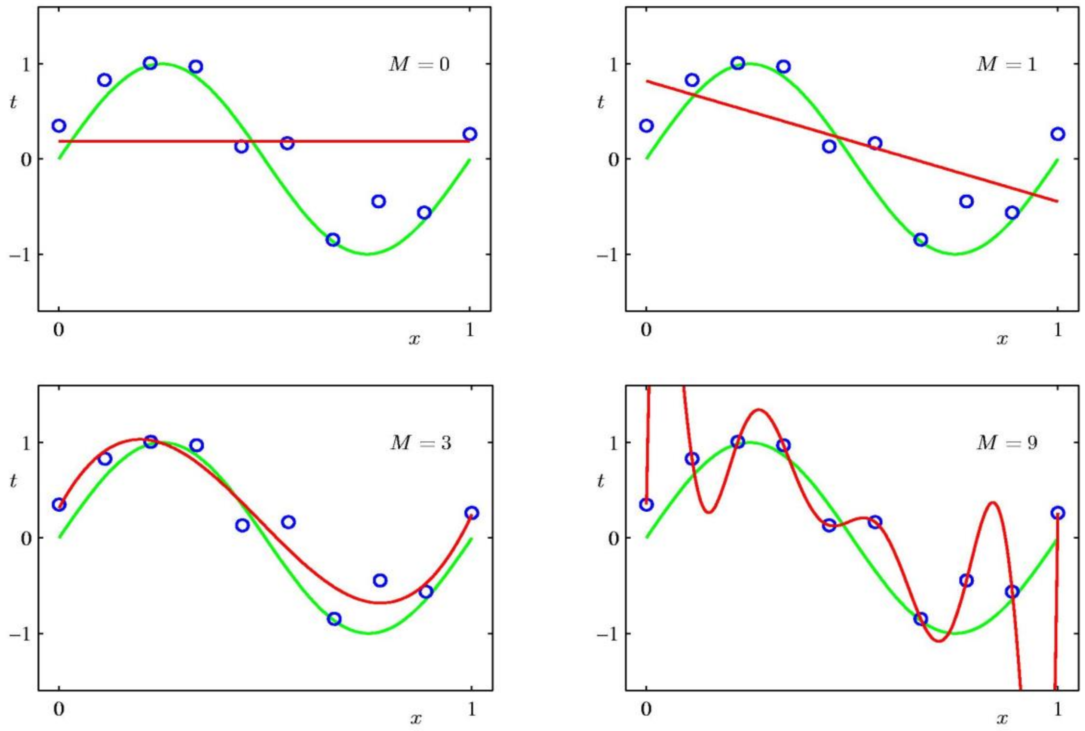
Linear Model

Here we minimize the error function

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

Generalization (4/8)

Model Selection (***Model Comparison***):



Over-fitting!

Generalization (5/8)

Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

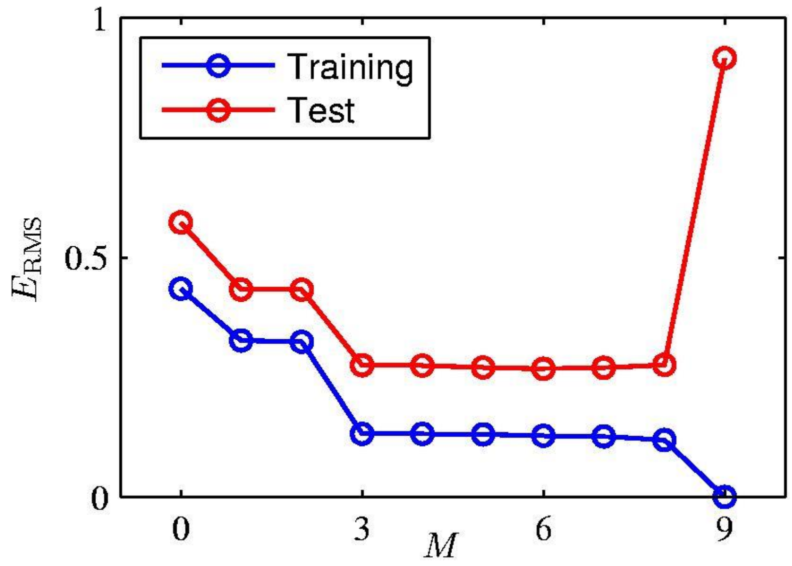
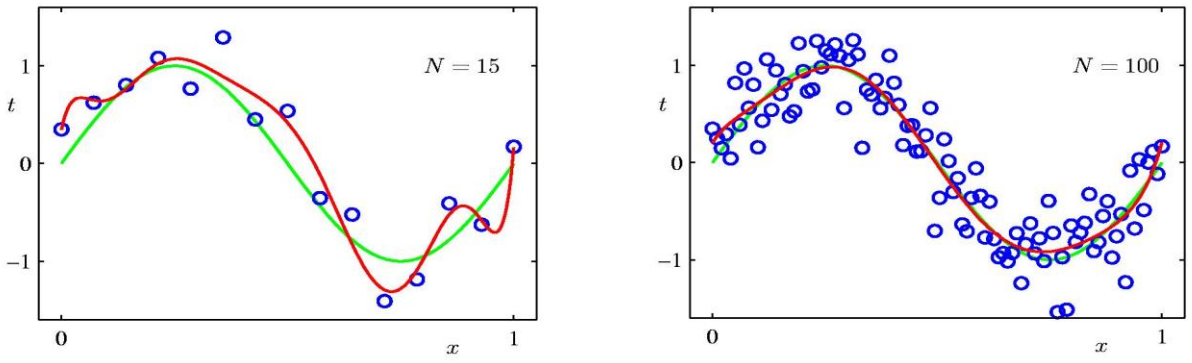


Table of the coefficient w^*

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Generalization (6/8)

M = 9



- ✓ The over-fitting problem becomes less severe as the size of the data set increases.
- ✓ In general, the number of data points should be no less than some multiple (say 5 or 10) of the number of adaptive parameters in the model.
- ✓ Regularization is often used to control the over-fitting phenomenon.
- ✓ In a Bayesian model, the effective number of parameters adapts automatically to the size of the data set.

Generalization (7/8)

Regularization

Add a penalty term to the error function to discourage the coefficients from reaching large values.

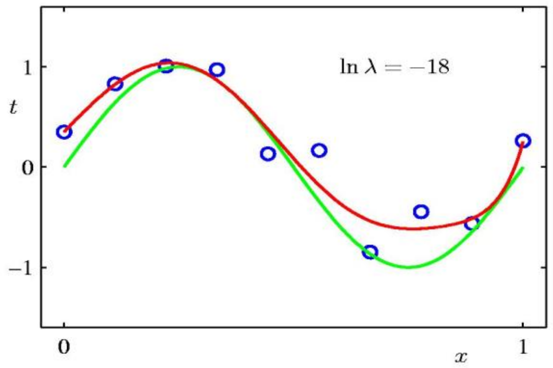
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

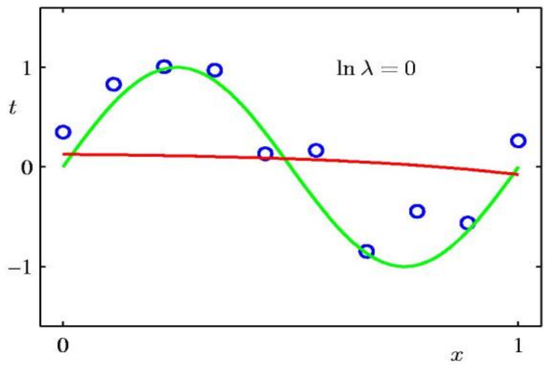
where $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = \omega_0^2 + \omega_1^2 + \cdots + \omega_M^2$

- ✓ The coefficient ω_0 is usually omitted.
- ✓ This kind of techniques is called *shrinkage* methods in the statistics literature. The particular case of a quadratic regularizer is called *ridge regression*.
- ✓ In neural networks, this approach is known as *weight decay*.

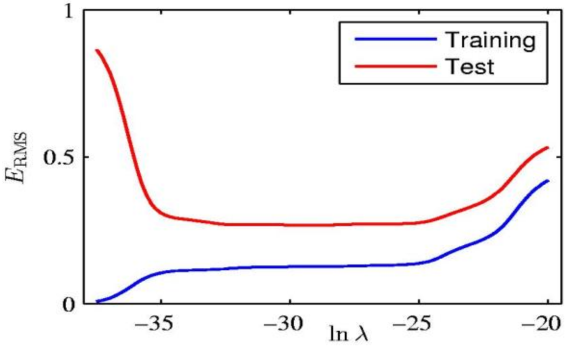
Generalization (8/8)

M = 9





	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

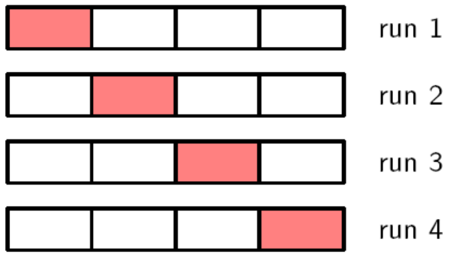


Model Selection (1/2)

- In model selection, we may split the data set into a **training** set, a **validation** set, and/or a **test** set.
- *S*-fold cross-validation: use (S-1)/S of the available data for training.

  (*Leave-one-out* technique: S = N)

The technique of S-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into S groups (in the simplest case these are of equal size). Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure is then repeated for all S possible choices for the held-out group, indicated here by the red blocks, and the performance scores from the S runs are then averaged.



- Drawbacks of cross-validation:
  - The number of training runs increases by a factor of S.
  - The number of parameter combinations increases exponentially.

Model Selection (2/2)

"Information criteria" have been proposed that adds a penalty term to compensate for the over-fitting of more complex models.

e.g. *Akaike Information Criterion* (AIK)

$$\ln p(\mathcal{D}|\mathbf{w}_{\mathrm{ML}}) - M$$

Bayesian Information Criterion (BIC)

$$\ln p(D \mid \mathbf{w}_{\mathbf{MAP}}) - \frac{1}{2} M \ln N$$

Three Ways to Build the Mapping Model

- ***Discriminant function***: find a function that maps each input **x** directly onto the target value/label.
- ***Generative models:***
  - ✓ Model $p(\mathbf{x}|t)$ and $p(t)$, or model $p(\mathbf{x},t)$.

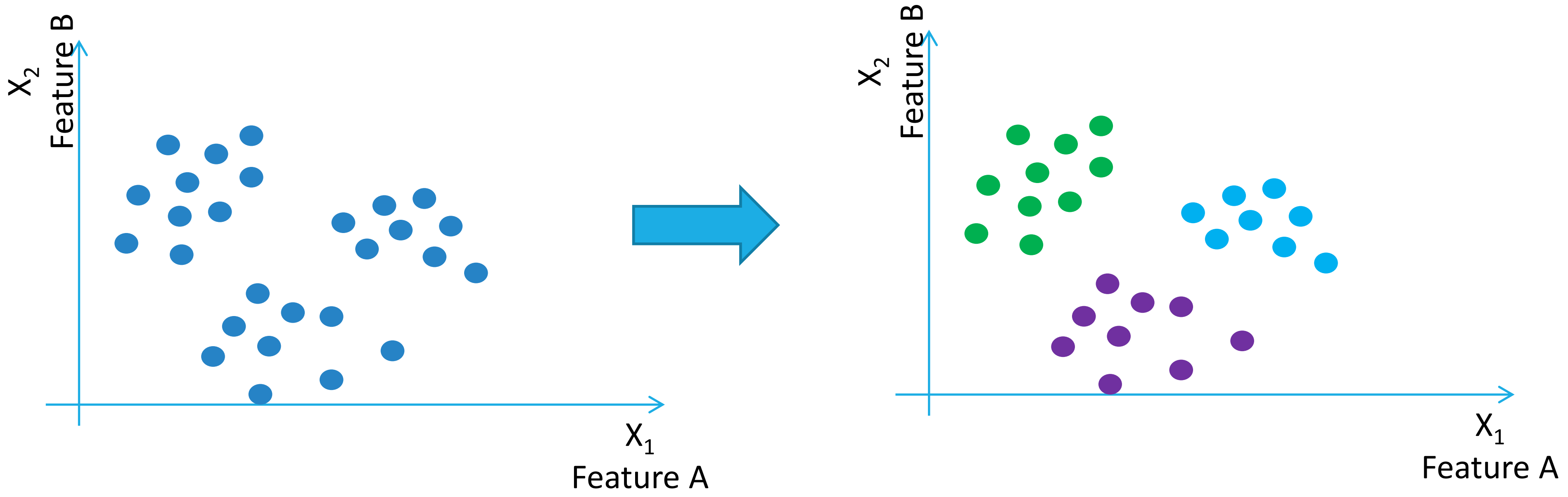
    $\Rightarrow$ Find $p(t|\mathbf{x})$.
  - ✓Use decision theory to determine class membership for each new input x.
  - ✓Allow p(x) to be determined. $\Rightarrow$ can detect outliers.
- ***Discriminative models:***
  - ✓Model p(t|x) directly.
  - ✓Use decision theory to determine class membership for each new input x.

Unsupervised Learning (1/8)

- ***Unsupervised learning***: the training data consists of a set of input vectors x without any corresponding target values.
  - ✓ ***Clustering***: to discover groups of similar examples within the data.
  - ✓ ***Density Estimation***: to determine the distribution of data within the input space.
  - ✓ ***Dimension Reduction:*** to project the data from a high-dimensional space down to a low-dimensional space.
  - ✓ ***Generative Model***: to learn a model that can generate data like the training data.
  - ✓ ***Self-supervised Learning***: use the data itself to generate supervisory signals.

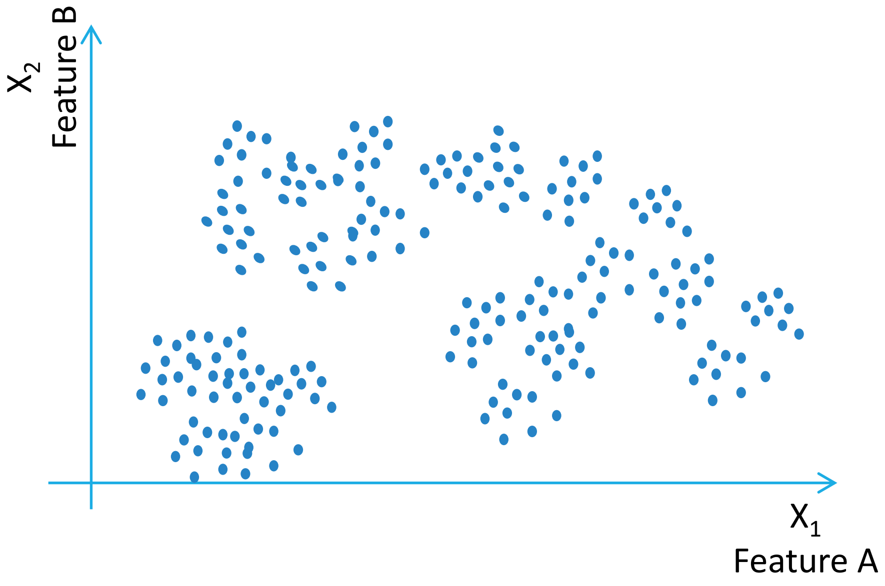
Unsupervised Learning (2/8)

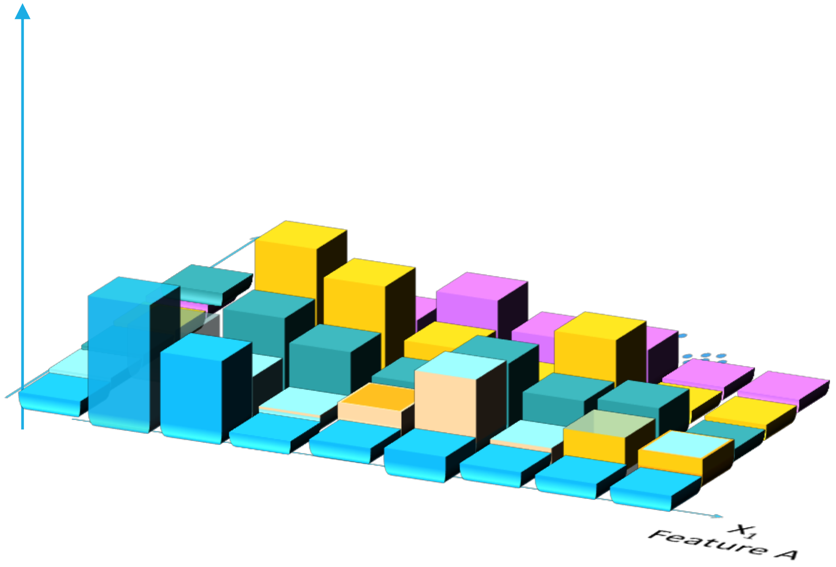
Clustering



Unsupervised Learning (3/8)

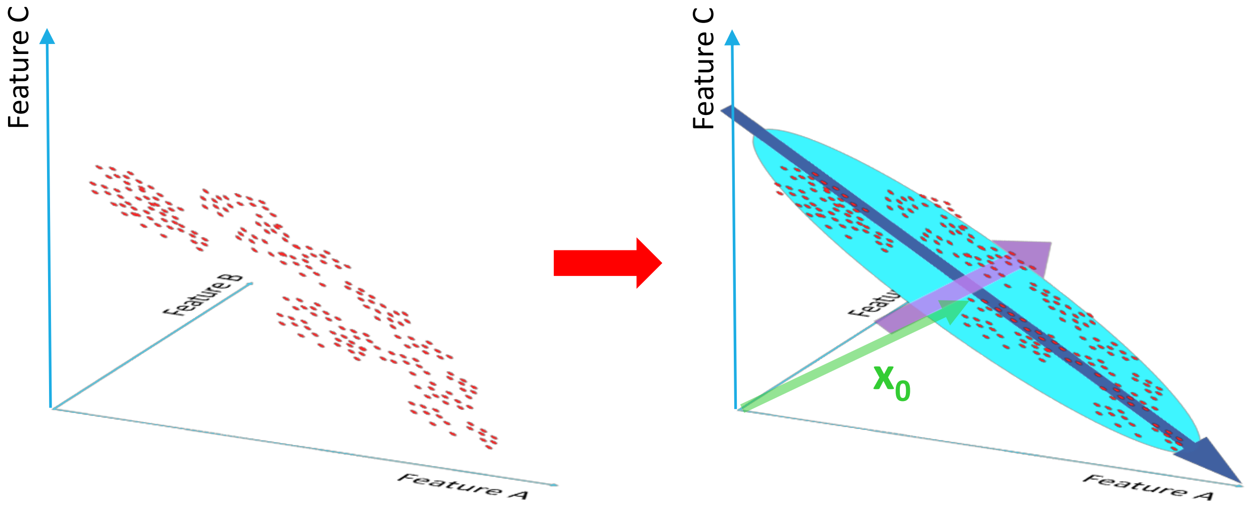
Density Estimation





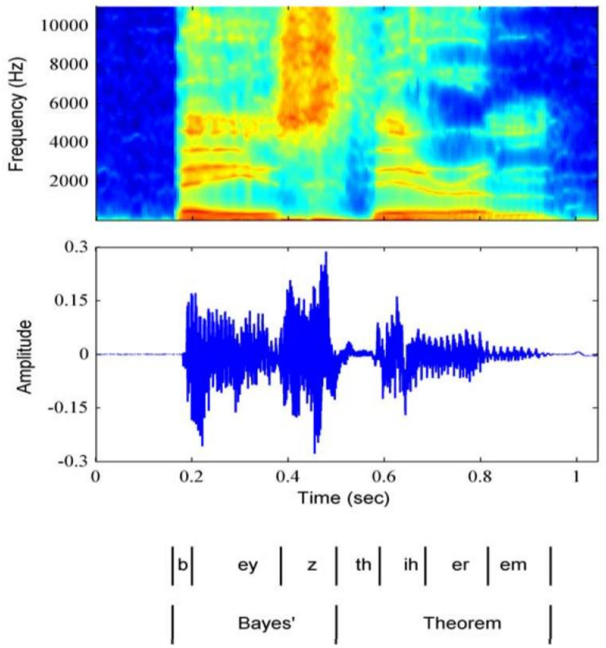
Unsupervised Learning (4/8)

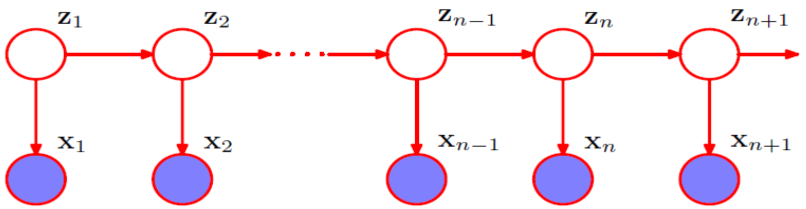
Dimension Reduction



Unsupervised Learning (5/8)

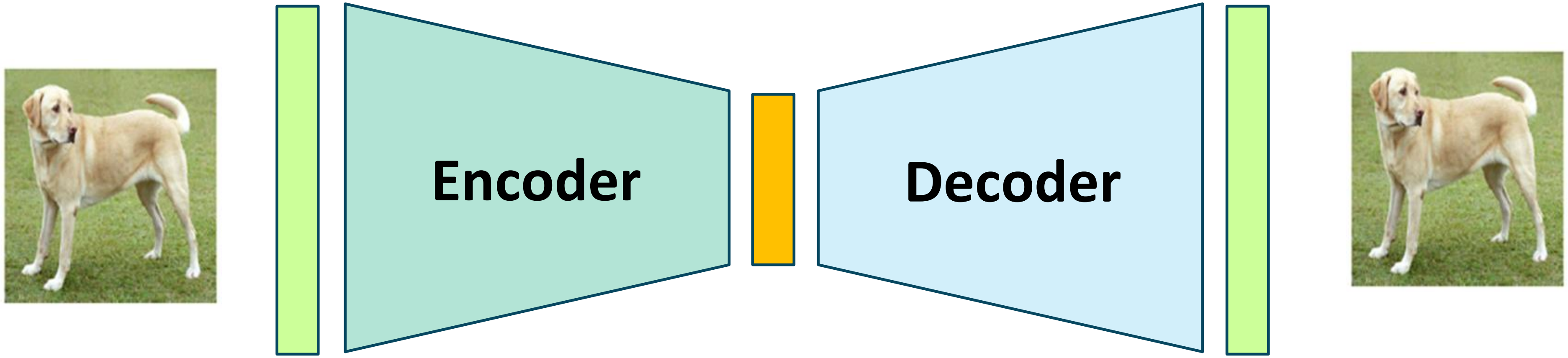
Hidden Markov Model (HMM)





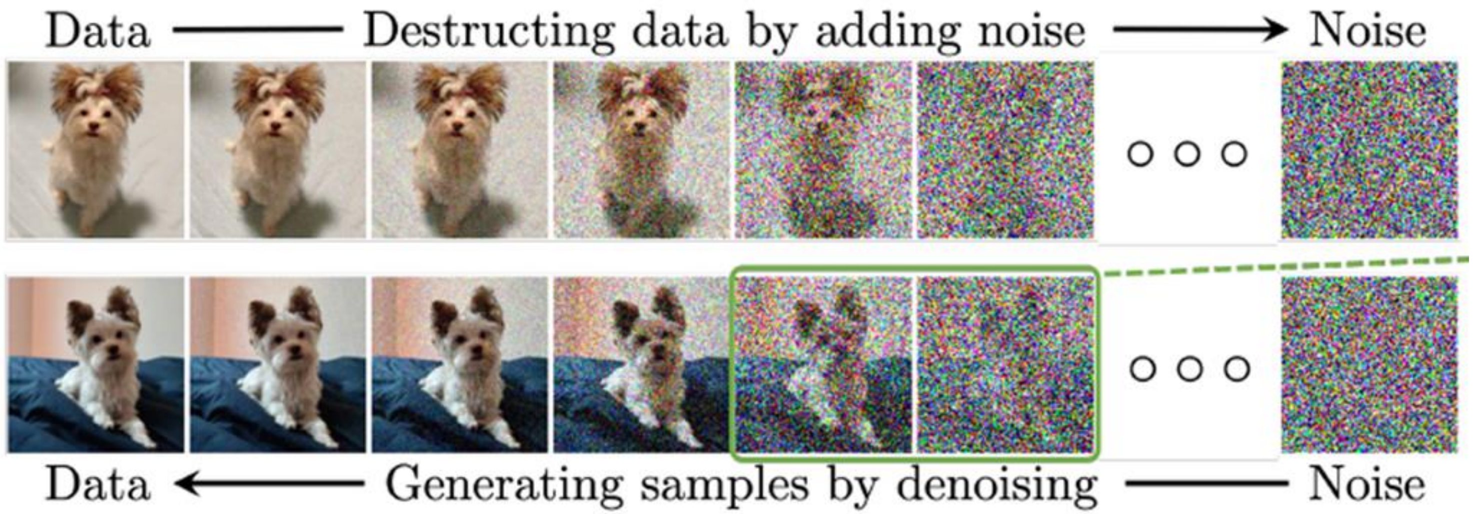
Unsupervised Learning (6/8)

Auto-encoder



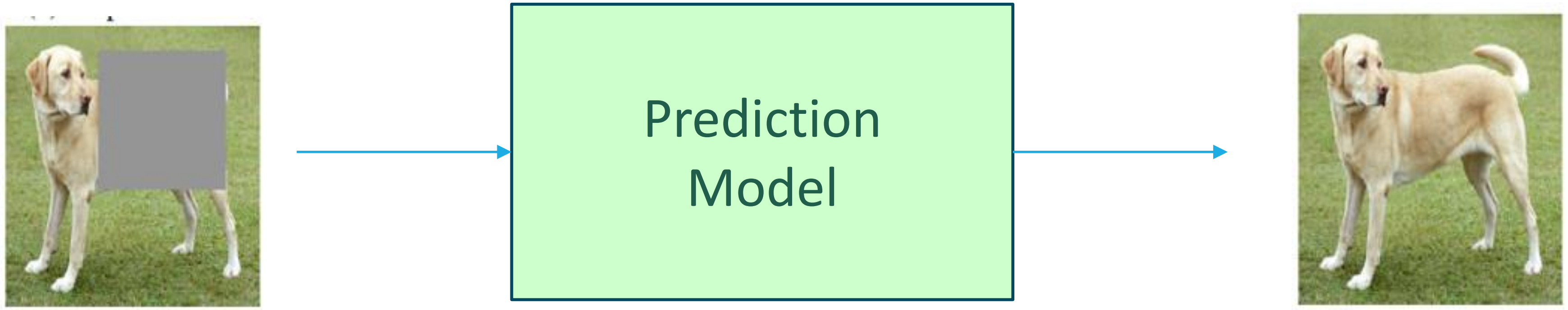
Unsupervised Learning (7/8)

Diffusion Model



Unsupervised Learning (8/8)

Self-supervised Learning

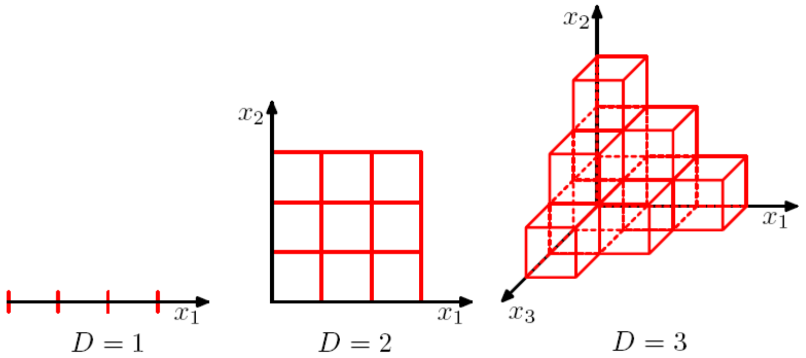


Manipulated Data

Original Data

Curse of Dimensionality (1/3)

Example: Exponentially grow of the number of regions in a regular grid



Example: Exponentially grow of polynomial coefficients

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

M = 3

Curse of Dimensionality (2/3)

Our geometrical intuitions can fail badly in a space of higher dimensionality.

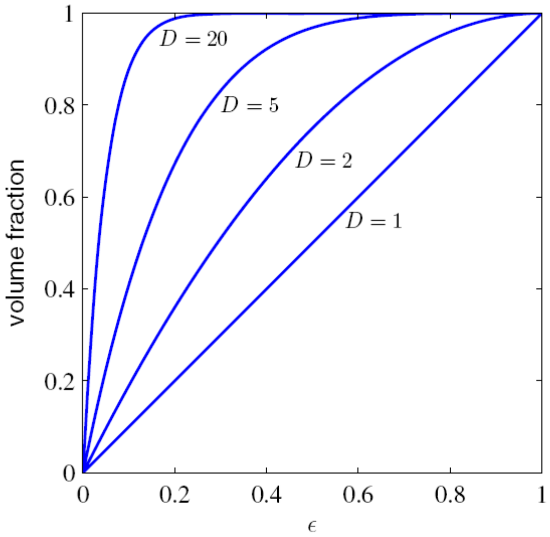
Not all intuitions developed in spaces of low dimensionality will generalize to spaces of high dimensions!

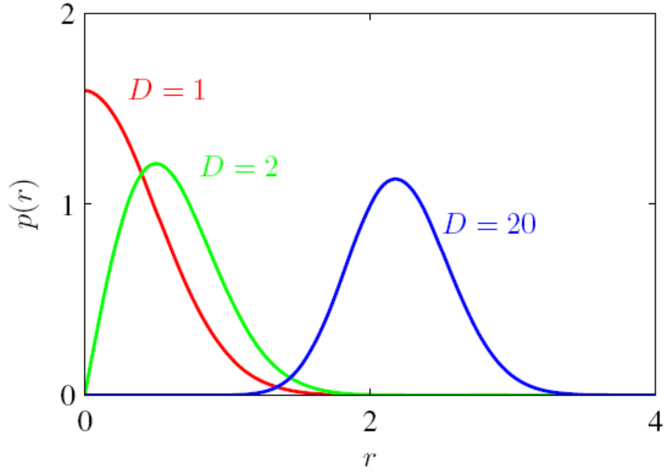
e.g., volume fraction of a sphere

e.g., Gaussian Densities in Higher Dimensions

$$V_D(r) = K_D r^D$$

$$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = 1 - (1-\epsilon)^D$$





Curse of Dimensionality (3/3)

Good News:

- ✓ Real data can often be confined to a subspace with lower effective dimensionality
- ✓ Real data typically exhibit some smoothness properties (at least locally) so we can exploit local interpolation-like techniques for the prediction of the target variables.

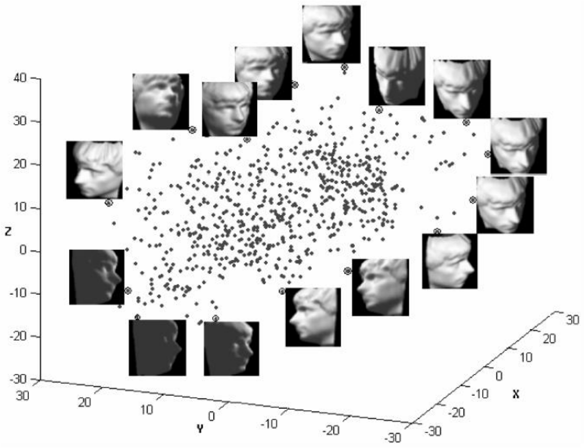


Fig. 19. Three-dimensional embedding of ISOMAP face data using RML.

Ref: T. Lin & H. Zha, “Reimannian Manifold Learning”, PAMI, May 2008

Reinforcement Learning (1/7)

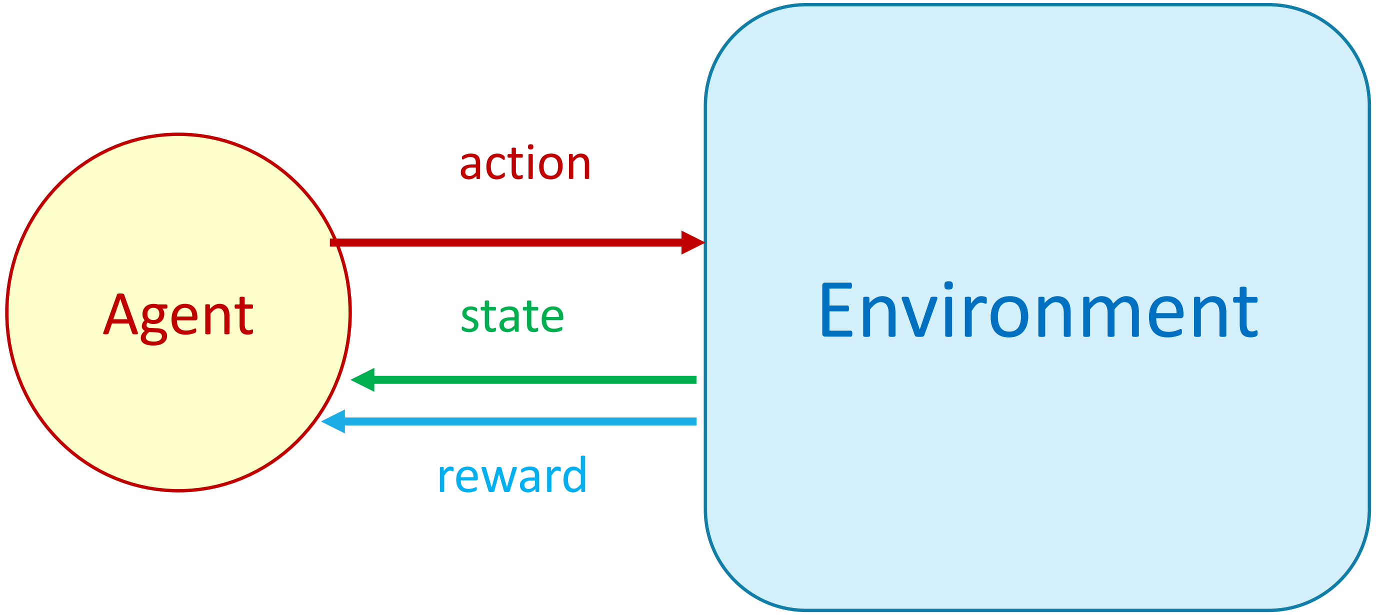
History of Reinforcement Learning

- Trial and error learning
- Optimal control
- Temporal-difference methods

Reinforcement Learning (2/7)

- Learn to map **situations** to **actions**
  - ✓ Discover which action yields the most reward
- Major characteristics
  - ✓ Closed-Loop Problems
  - ✓ Do not have direction instructions about what actions to take
  - ✓ Actions may not only affect the immediate reward but also the next situation and all subsequent rewards
- Three major aspects – Sensation, Action, and Goal

Reinforcement Learning (3/7)



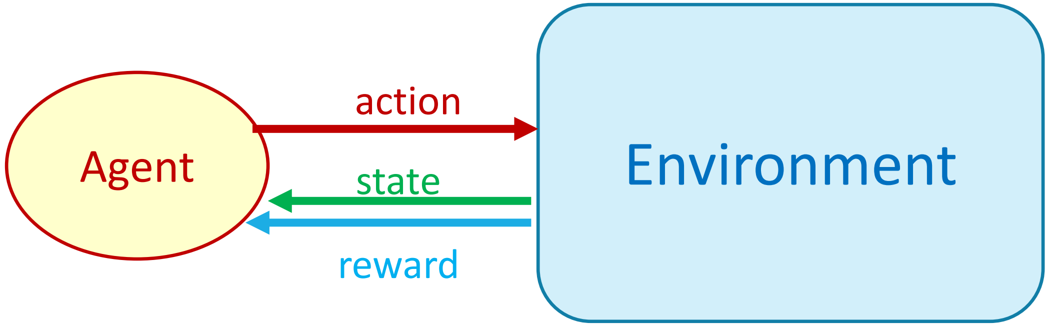
Reinforcement Learning (4/7)

RL v.s. Supervised/Unsupervised Learning

- There is no supervisor. All we have is the reward signal.
- Do not predict the correct action simply based on the current situation.
- Deal with sequential data but not i.i.d. data.
- Try to maximize a reward signal instead of finding the hidden structure
- There is balance between exploitation and exploration
- Involve the interaction between an agent with an environment

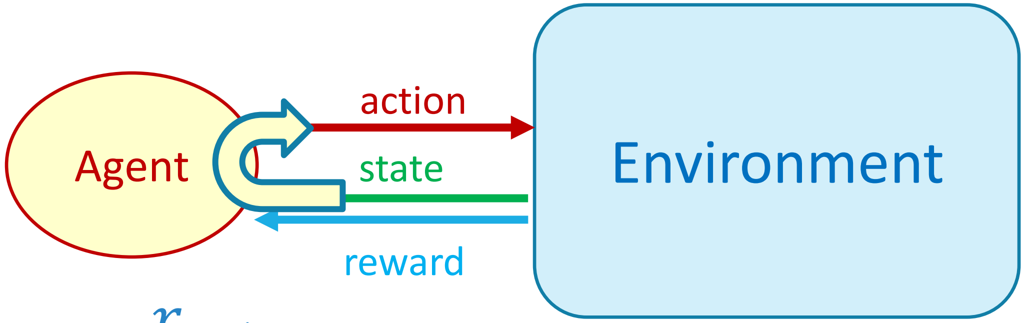
Reinforcement Learning (5/7)

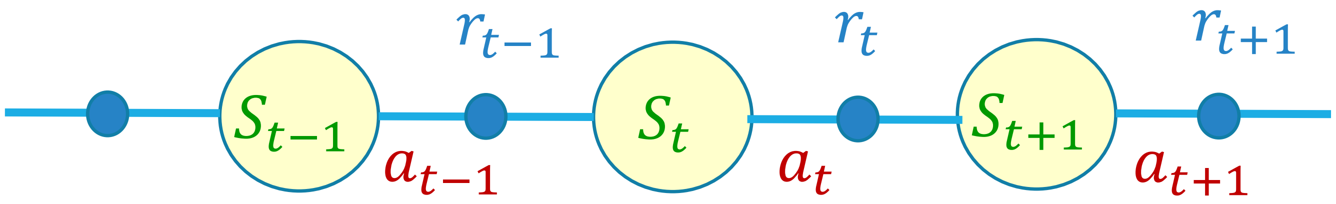
- Learn how to act or behave when given occasional reward or punishment signals.
- Close to the way human learns to interact with the environment
- Basic reinforcement learning is modeled as a Markov decision process (MDP)
- Every action impacts the environment, and the environment provides the reward to guide the learning process ⇒ learn how to act in order to maximize the reward.



Reinforcement Learning (6/7)

- **Reward**: indicate which action is preferred in an immediate sense
- **Value**: indicate which action is preferred in the long run.
- **Policy**: a mapping from states to actions





Reinforcement Learning (7/7)

Methods for Reinforcement Learning

- **Dynamic programming**
- **Monte Carlo Methods**
- **Temporal-difference Learning**