GHENT UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

E016350 - Artificial Intelligence

Lecture 1

Introduction

Part 2: Fundamental Machine Learning Concepts

Aleksandra Pizurica

Ghent University

Spring 2025

What is Machine Learning (ML)?

"Learning is any process by which a system improves performance from experience."
– Herbert Simon

"Machine learning is the science of getting computers to act without being explicitly programmed."
– Andrew Ng

Definition by Tom Mitchell (1998):
Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E.

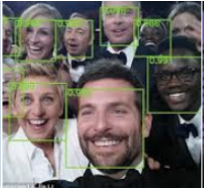
A well-defined learning task is given by $< P, T, E >$

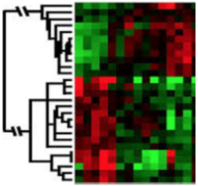
Why learning?

We need machine learning when

- We cannot anticipate all possible situations that the agent might face
- We cannot anticipate all changes over time
- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise – programmers don't know how to program a solution themselves (e.g., speech recognition, face recognition)
- Models must be customized (e.g. personalised medicine); huge amounts of data (e.g., genomics) ...







Relations to human learning

- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Can be quick but can also take years
- Machine learning doesn't have to look like human learning
  - It may borrow ideas from biological systems, e.g., neural networks
  - Doesn't need to "duplicate" our brain and our reasoning process
    - ⋆ Think of aerodynamics: we don't make air planes to fly exactly as pigeons!

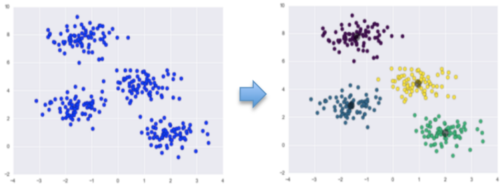
- It may perform better or worse than humans.

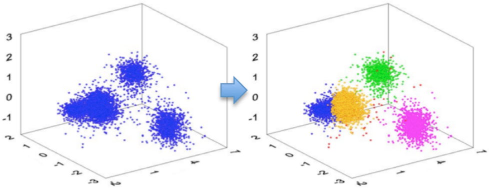
Slide based on: R. Grosse C. Maddison J. and Bae S. Pitis: Introduction to Machine Learning, University Toronto

Feedback in learning

- Unsupervised learning
  - Learning patterns without explicit feedback supplied
  - Example: clustering – identify potentially useful clusters of data samples that can correspond to different "classes"
- Supervised learning = learning from examples
  - Learning a function that maps input to output based on available (observed) input-output pairs (Correct answers for each instance)
- Semi-supervised learning
  - A few labeled samples available and a large collection of unlabeled ones
  - Learn from geometry of unlabeled samples and use the labeled ones to improve the learning
- Reinforcement learning
  - Learning from a series of reinforcements – rewards and punishments

Unsupervised learning – clustering





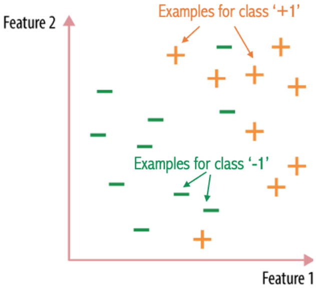
Example: motion segmentation:

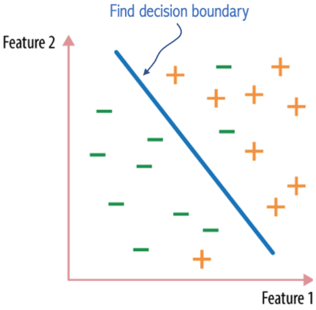
E. Elhamifar and R. Vidal: Sparse Subspace Clustering: Algorithm, Theory, and Applications, IEEE Trans. Pattern Anal. Mach. Intell. 2013

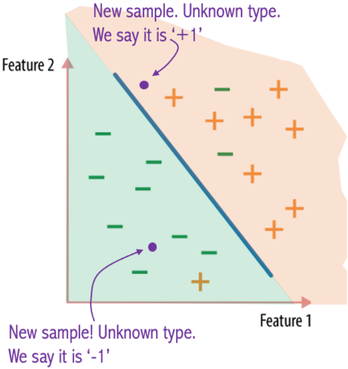
Supervised learning

Consider the case of classifying objects into two linearly separable classes, based on two attributes (two input features).

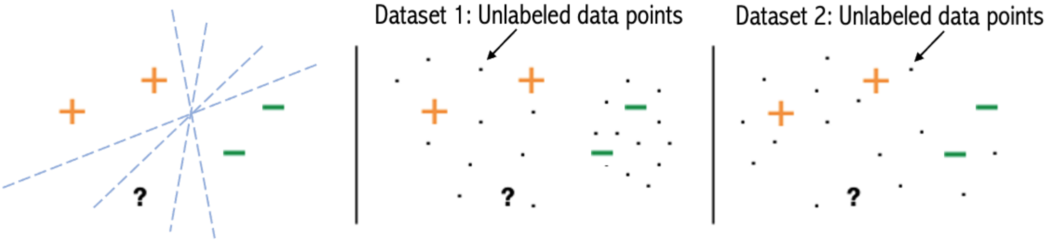
- the optimal decision boundary is determined based on the available labeled samples (examples of input-output pairs)
- input = feature vector; output = class '+1' or '-1'





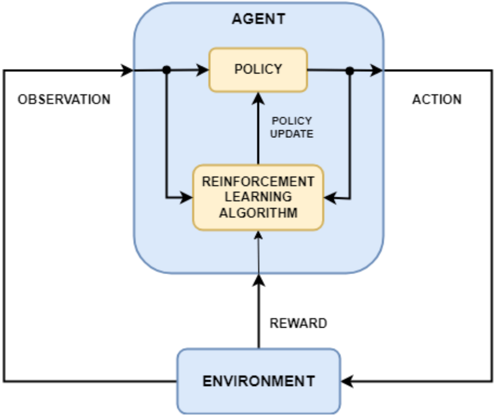


Semi-supervised learning



- Labelled data are often scarce. Unlabelled data are typically abundant
- With few labelled samples it is difficult to determine reliably the decision boundary.
- Making use of the unlabelled data, the ambiguity is reduced and more reliable classification reached.

Reinforcement learning



https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

- Beyond this course (included in the 6-credit version)

Supervised learning (a.k.a.inductive learning)

Simplest form: learn a function from examples

f is the target function

An example is a pair x, $f(x)$, e.g., $\begin{array}{c|c|c} O & O & X \\ \hline & X & \\ \hline X & & \end{array}$, $+1$

Problem: find a hypothesis h

such that $h \approx f$

given a training set of examples

Supervised learning

Given a training set of N example input-output pairs

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(N)}, y^{(N)})$$

where each $y^{(j)}$ was generated by an unknown function $y = f(\mathbf{x})$, discover a function h that approximates the true function f.

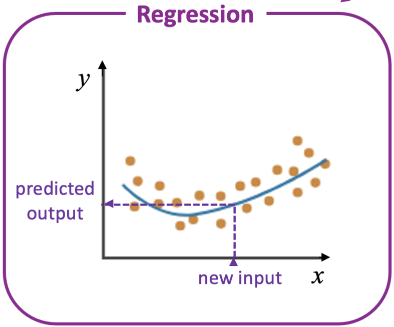
- To measure the accuracy of the hypothesis h, we use a test set, which is different than the training set
- The hypothesis is good and is said to generalize well if it correctly predicts the value of y for new examples
- Intelligence is ability to **predict** (e.g. the next sample) and **generalize to unseen scenarios** (T. Poggio & S. Smale, 2003).

Supervised learning

Given a training set of N example input-output pairs

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(N)}, y^{(N)})$$

where each $y^{(j)}$ was generated by an unknown function $y = f(\mathbf{x})$, discover a function h that approximates the true function f.

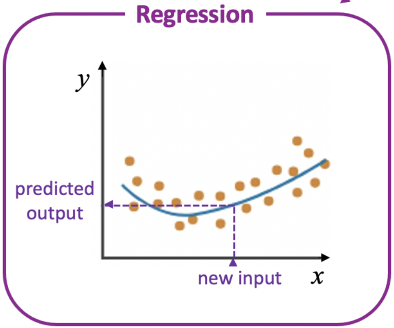


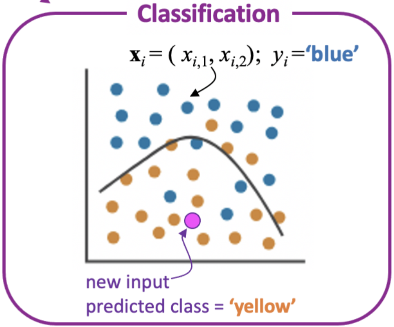
Supervised learning

Given a training set of N example input-output pairs

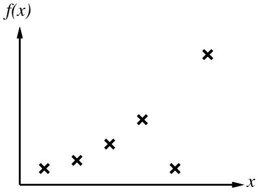
$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(N)}, y^{(N)})$$

where each $y^{(j)}$ was generated by an unknown function $y = f(\mathbf{x})$, discover a function h that approximates the true function f.

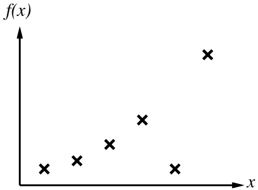


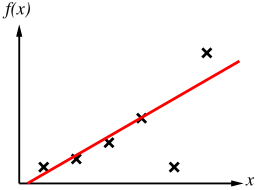


Finding hypotheses that fit data

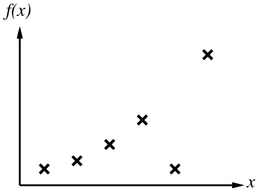


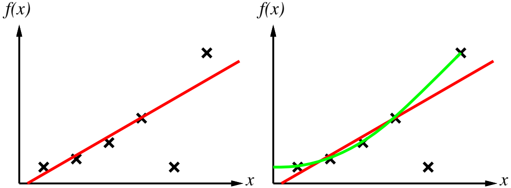
Finding hypotheses that fit data



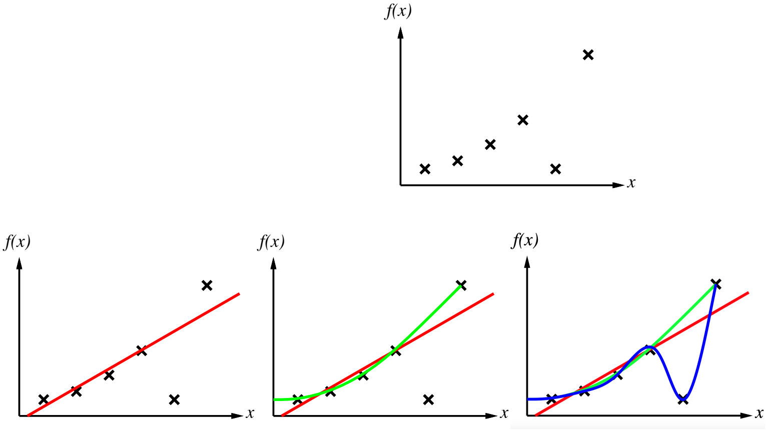


Finding hypotheses that fit data

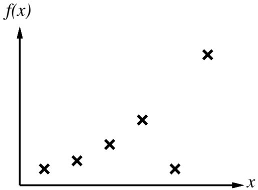


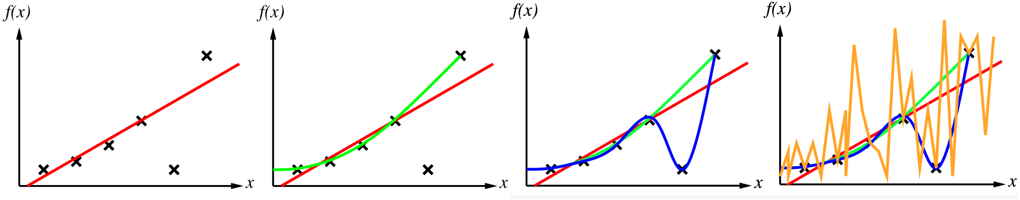


Finding hypotheses that fit data

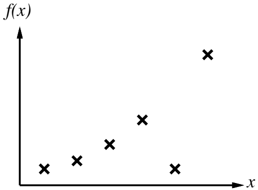


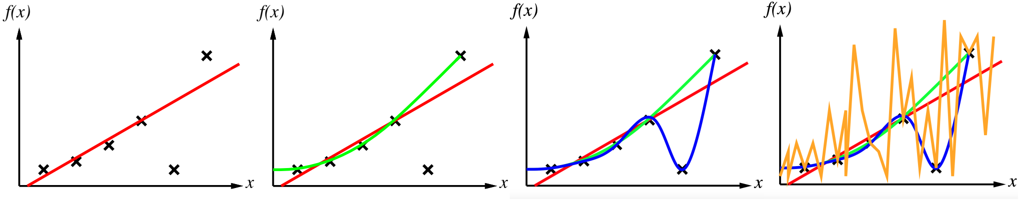
Finding hypotheses that fit data





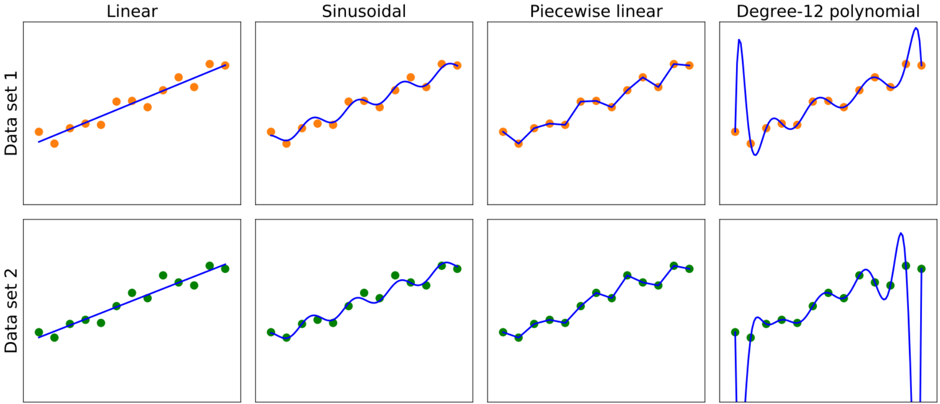
Finding hypotheses that fit data





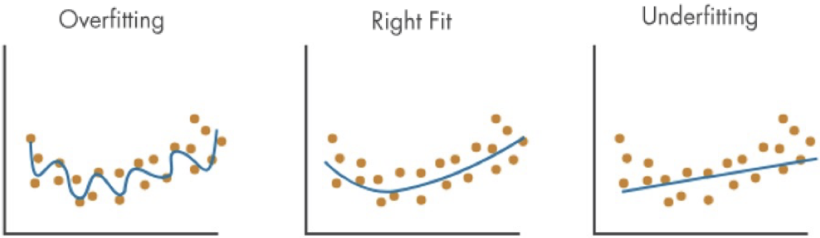
Ockham's razor - The best models are simple models that fit the data well. Simpler explanations are, other things being equal, generally better than more complex ones.

Finding hypotheses that fit data



- **Bias-Variance** trade-off
- Large bias typically means that h fails to find pattern in the data (it's underfitting)
- Large variance – h pays too much attention pattern to particular data (overfitting)

Overfitting and Underfitting

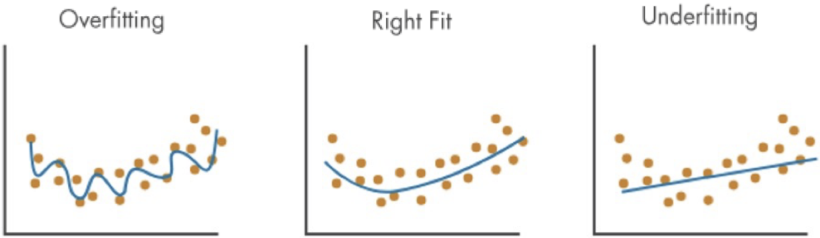


https://www.mathworks.com/discovery/overfitting.html

Overfitting occurs when the model is so closely aligned to the training data that it does not generalize well. Overfitting can happen because:

- The ML model is too complex; it memorizes irrelevant patterns in the training data (including noise).
- The training data size is too small for the model complexity and/or contains large amounts of irrelevant information.

Overfitting and Underfitting

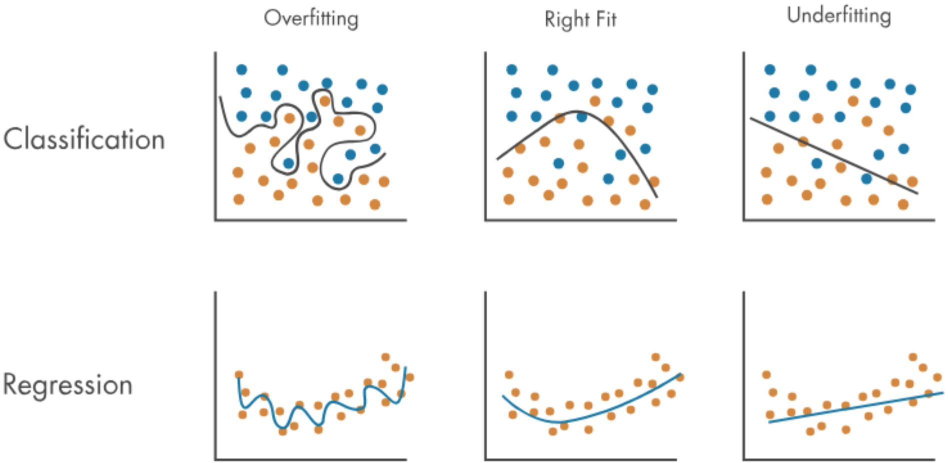


https://www.mathworks.com/discovery/overfitting.html

Underfitting is the opposite concept of overfitting:

- The model is too simple and doesn't learn the relevant patterns in the training data. It is unable to generalize well on the new data
- An underfit model has poor performance on the training data and will result in unreliable predictions.

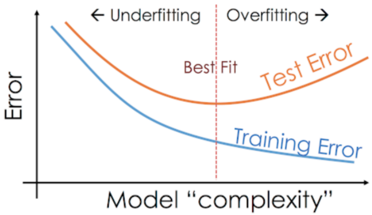
Overfitting and Underfitting



https://www.mathworks.com/discovery/overfitting.html

Overfitting and underfitting can be present in both classification and regression models.

Overfitting and Underfitting depending on the Model Complexity

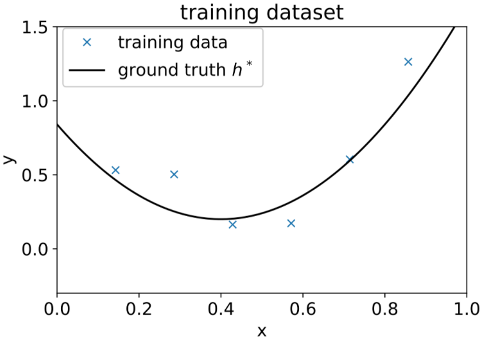


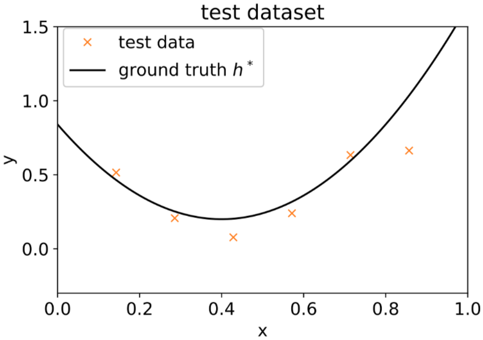
https://setscholars.net/understanding-overfitting-in-machine-learning-a-comprehensive-guide/

- In modern deep learning models, we are most often concerned with overfitting because the models are huge (e.g., hundreds of millions of parameters) and there is often not enough data to match the model size well.
- More complicated for very deep learning models (practice differs from theory)
- We will return to these issues when we address regularization of ML models and best practices in ML.

Bias-Variance tradeoff

Consider the following training and test datasets:

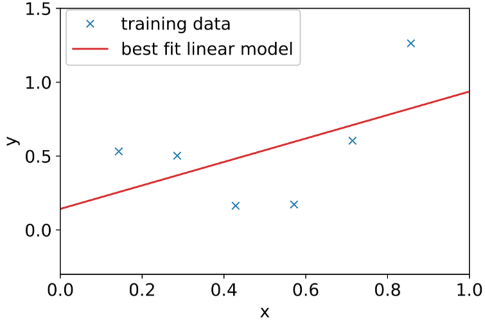


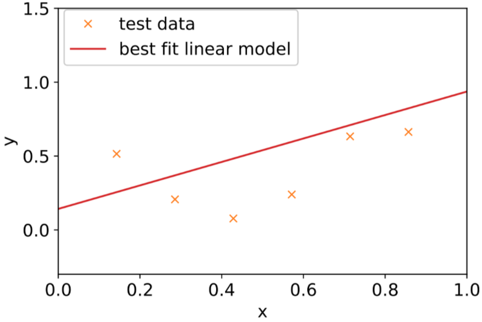


$$y^{(i)} = h^*(x^{(i)}) + \xi^{(i)} \quad ; \quad \xi^{(i)} \sim N(0, \sigma^2)$$

Example from A. Ng and T. Ma: Lecture Notes Machine Learning, Stanford University, 2023.

Bias-Variance tradeoff

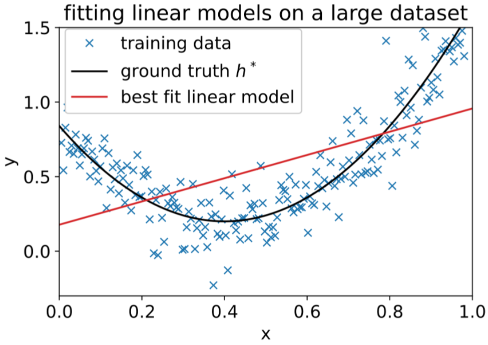




The best fitted linear model cannot predict y from x accurately even on the training dataset, let alone on the test dataset

Example from A. Ng and T. Ma: Lecture Notes Machine Learning, Stanford University, 2023.

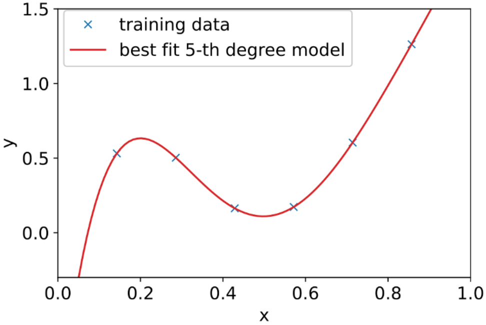
Bias-Variance tradeoff

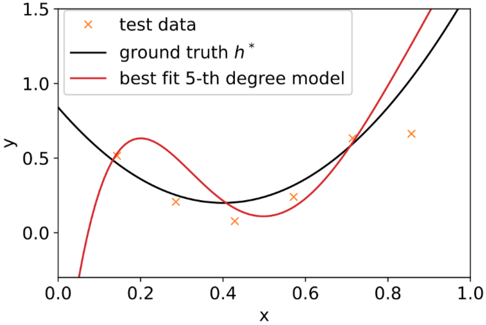


- Underfitting of this overly simple model cannot be mitigated with more training examples
- Informally, the **bias** of a model is the test error that remains even if we fit the model to an infinitely large dataset

Example from A. Ng and T. Ma: Lecture Notes Machine Learning, Stanford University, 2023.

Bias-Variance tradeoff

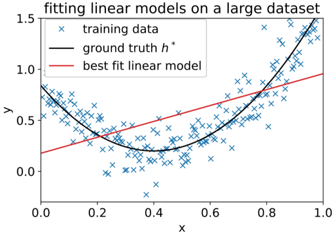


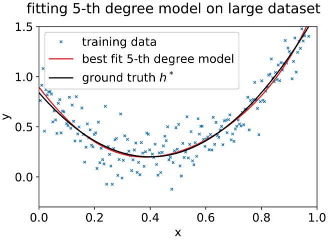


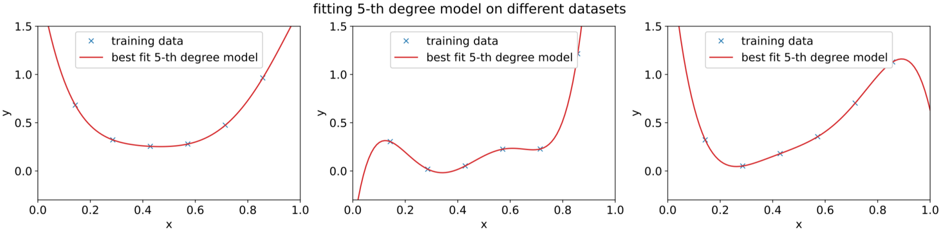
- A very complex model is likely to capture patterns specific to the (small) training set rather than reflecting the true relationship between x and y
- **Variance** measures the variation across models trained on different datasets (drawn from the same distribution)

Example from A. Ng and T. Ma: Lecture Notes Machine Learning, Stanford University, 2023.

Bias-Variance tradeoff

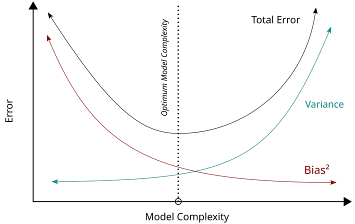




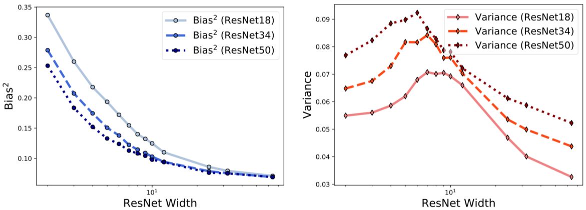


Example from A. Ng and T. Ma: Lecture Notes Machine Learning, Stanford University, 2023.

Bias-Variance depending on the Model Complexity



Classical theory bias-variance plot
`https://scott.fortmann-roe.com/docs/BiasVariance.html`

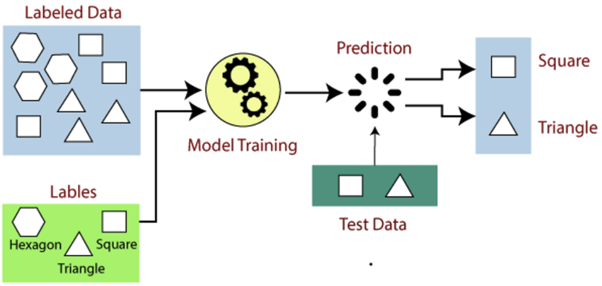


Practical behaviour observed in very deep neural networks [Yang et al, ICML 2020]

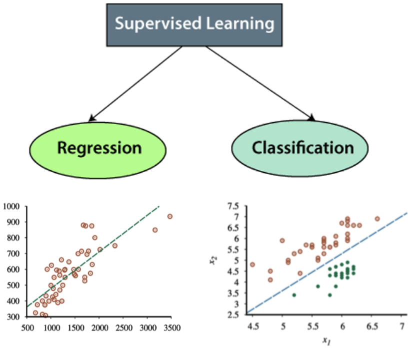
- Caution: recent work calls the classical theory into question for very deep neural networks and other over-parameterized models, for which it is often observed that larger models generalize better
  - The bias remains monotonically decreasing, but the variance is unimodal or bell-shaped: it increases then decreases with the width of the network

    Z. Yang, Y. Yu, C. You, J. Steinhardt and Y. Ma. Rethinking Bias-Variance Trade-off for Generalization of Neural Networks. International Conference on Machine Learning (ICML) 2020.

Some basic ML approaches: Logistic regression (next lesson)



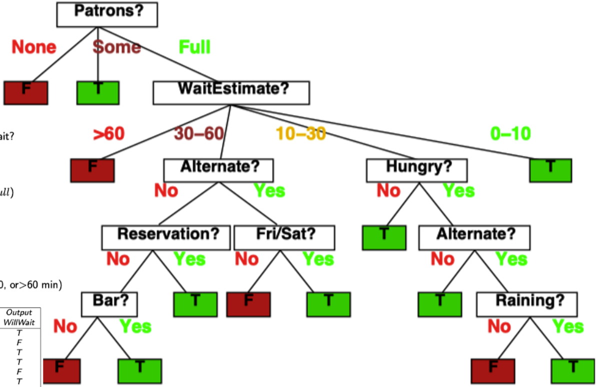
https://www.javatpoint.com/supervised-machine-learning



Given a training set $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(N)}, y^{(N)})$ where each $y^{(j)}$ was generated by an unknown function $y = f(\mathbf{x})$, discover a function h that approximates the true f.

Some basic ML approaches: Decision trees

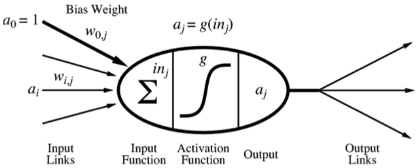
Example: decide whether to wait for a table in a restaurant

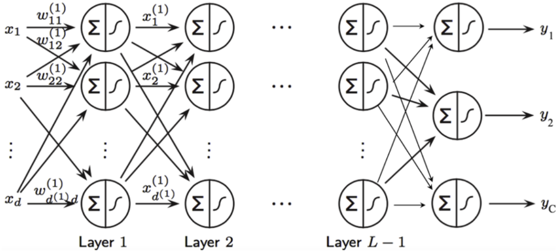


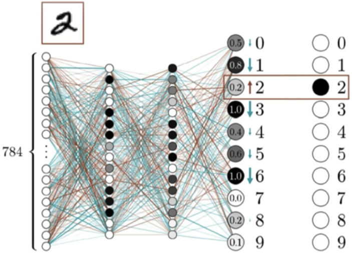
Alternate (Alt): Is there a suitable alternative restaurant nearby?
Bar (Bar): Is there a comfortable bar area in the restaurant, where I can wait?
Fri/Sat (Fri): True on Fridays/Saturdays
Hungry (Hun): Are we hungry?
Patrons (Pat): How many people are in the restaurant ($None$, $Some$ or $Full$)
Price ($Price$): the restaurant's price range (\$, \$\$, \$\$\$)
Raining ($Rain$): Is it raining outside?
Reservation (Res): Did we make a reservation?
Type ($Type$): the kind of restaurant (French, Italian, Thai or burger)
WaitEstimate (Est): the wait time estimated by the host (0-10, 10-30, 30-60, or>60 min)

Example	Input Attributes										Output
	Alt	Bar	Fri	Hun	Pat	$Price$	$Rain$	Res	$Type$	Est	WillWait
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

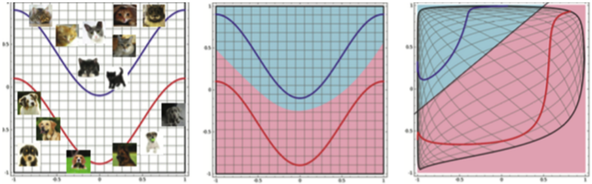
Some basic ML approaches: Neural networks



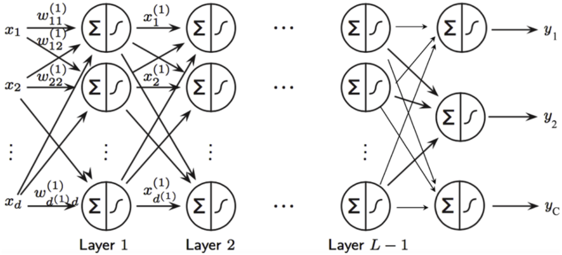


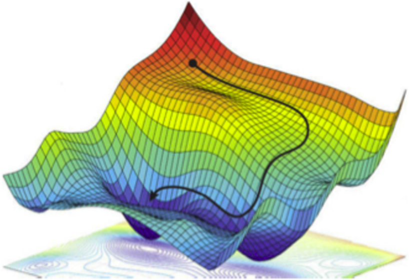


3Blue1Brown math channel https://www.youtube.com/watch?v=Ilg3gGewQ5U



Some basic ML approaches: Training deep neural networks





$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \sum_{j \in J} \nabla \log P_{\mathbf{w}}(\mathbf{y}_j | \mathbf{x}_j)$$

Minimizing cross-entropy; Maximizing log-likelihood; Mini-batch gradient descent

Some basic ML approaches: Bayesian learning

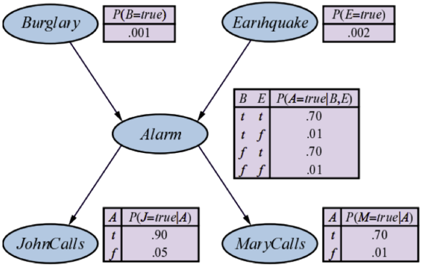
Why uncertainty?

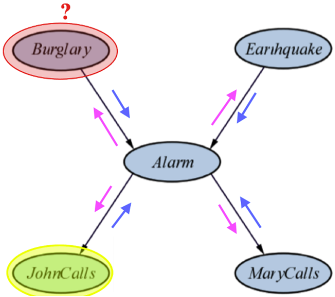
- Too complex, non-deterministic, partially observable environment

Probabilistic reasoning

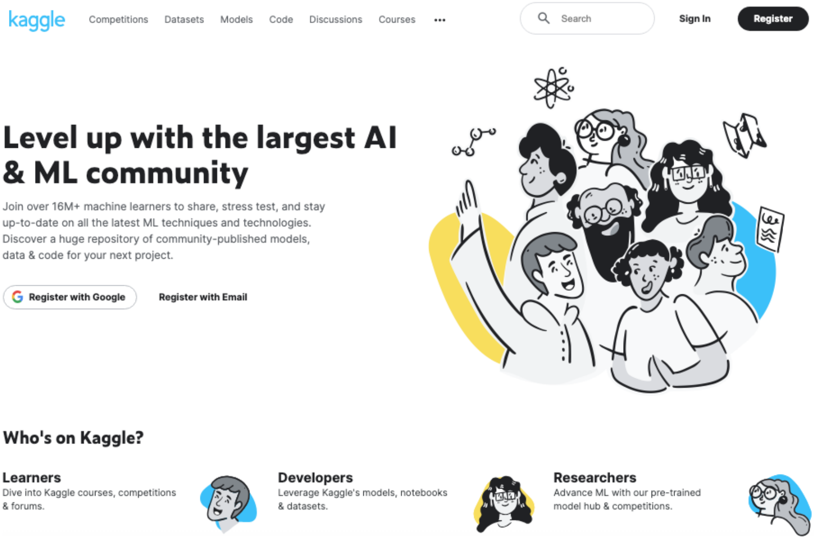
- Knowledge representation taking into account uncertainty

Bayesian networks



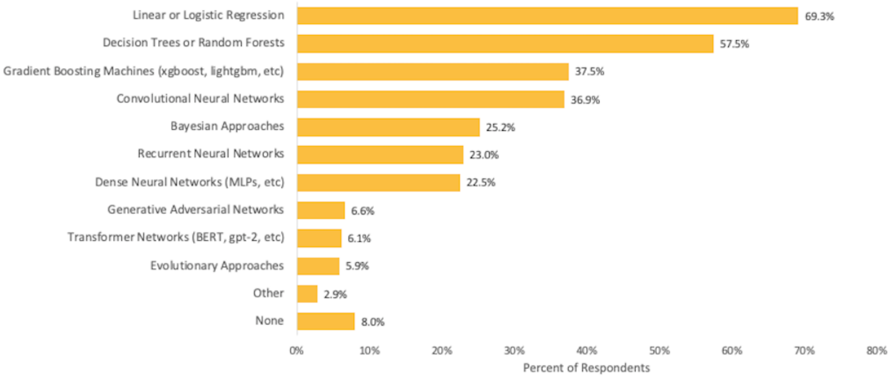


Hands on ML: Kaggle competitions



ML in practice: Which methods are commonly used?

2019 Kaggle survey of data science and ML practitioners: Which ML algorithms do you use on a regular basis? (Select all that apply)



Note: Data are from the 2019 Kaggle ML and Data Science Survey. You can learn more about the study here: https://www.kaggle.com/c/kaggle-survey-2019/data.
A total of 19717 respondents completed the survey; the percentages in the graph are based on a total of 14762 respondents who provided an answer to this question.

Why learning the ML fundamentals?

Why not jump straight to courses focused on deep learning?

- The principles you learn in this course will be essential to understand and apply deep neural networks.
- The techniques we will cover are still the most important ones to try for a new machine learning problem.
  - E.g., try logistic regression before building a deep neural net!
  - There's a whole world of Bayesian learning techniques.
- Probabilistic reasoning is the basis for building autonomous, rational agents (e.g., lies at the core of reinforcement learning)
- Probabilistic graphical models are at the core of deep generative models, including deep diffusion models in DALL-E2.
- Understanding well the ML fundamentals including probabilistic reasoning is necessary for building robust and **trustful AI systems of the future**!