Introduction to Machine Learning

CS4731
Dr. Mihail
Fall 2019
Slide content based on books by Bishop and Barber.
https://www.microsoft.com/en-us/research/people/cmbishop/
http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.HomePage

August 27, 2019

Machine Learning (ML)

What is ML?

Machine learning is the study of data-driven methods capable of mimicking, understanding and aiding human and biological information processing tasks. Machine learning can also be thought of as the science of getting machines to solve problems without being explicitly programmed how to.

Related issues:

- how to find patterns in data
- how to compress data
- how to interpret data
- how to process data

Patterns

Finding patterns is fundamental

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories.

Hard problems for ML

Credit card fraud detection

Hard problems for ML

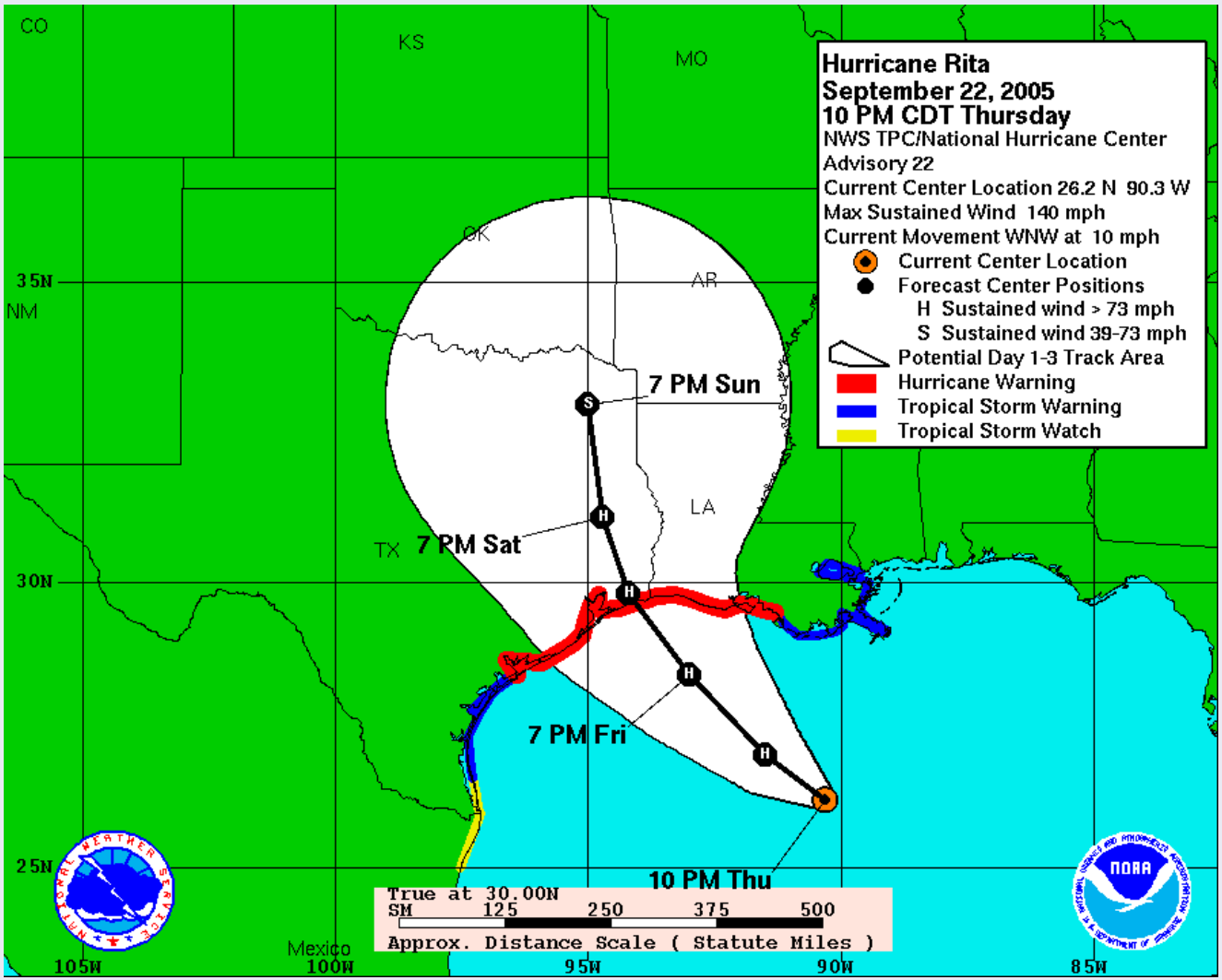
Face detection and recognition

Hard problems for ML

Self driving cars

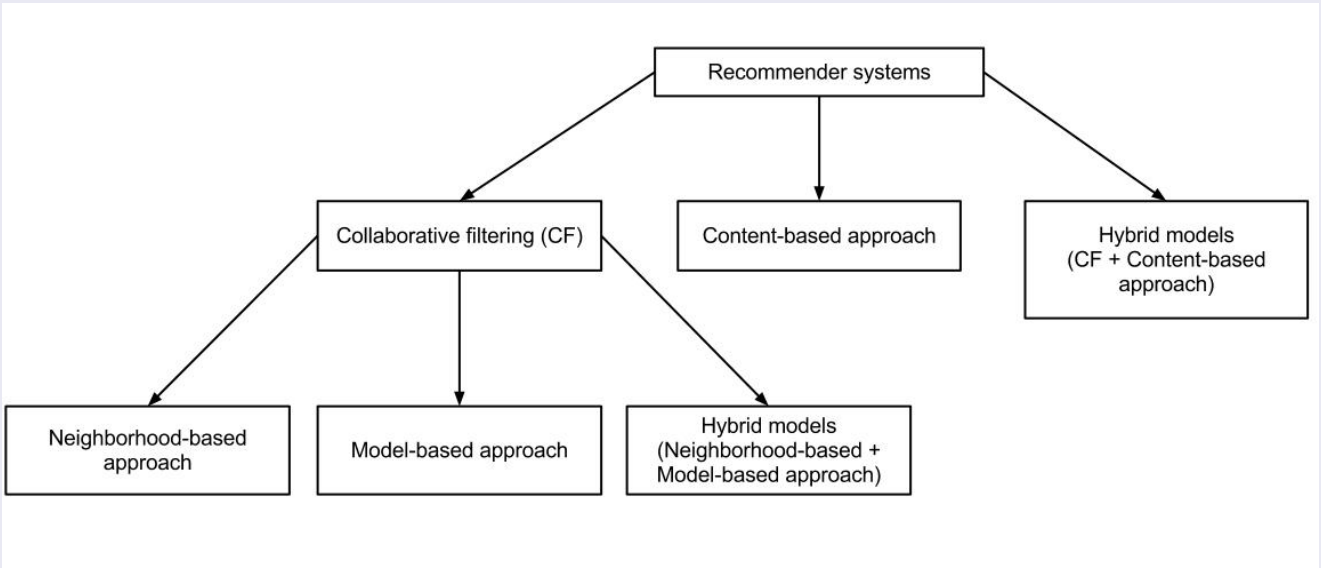
Hard problems for ML

Weather prediction



Hard problems for ML

Recommender systems

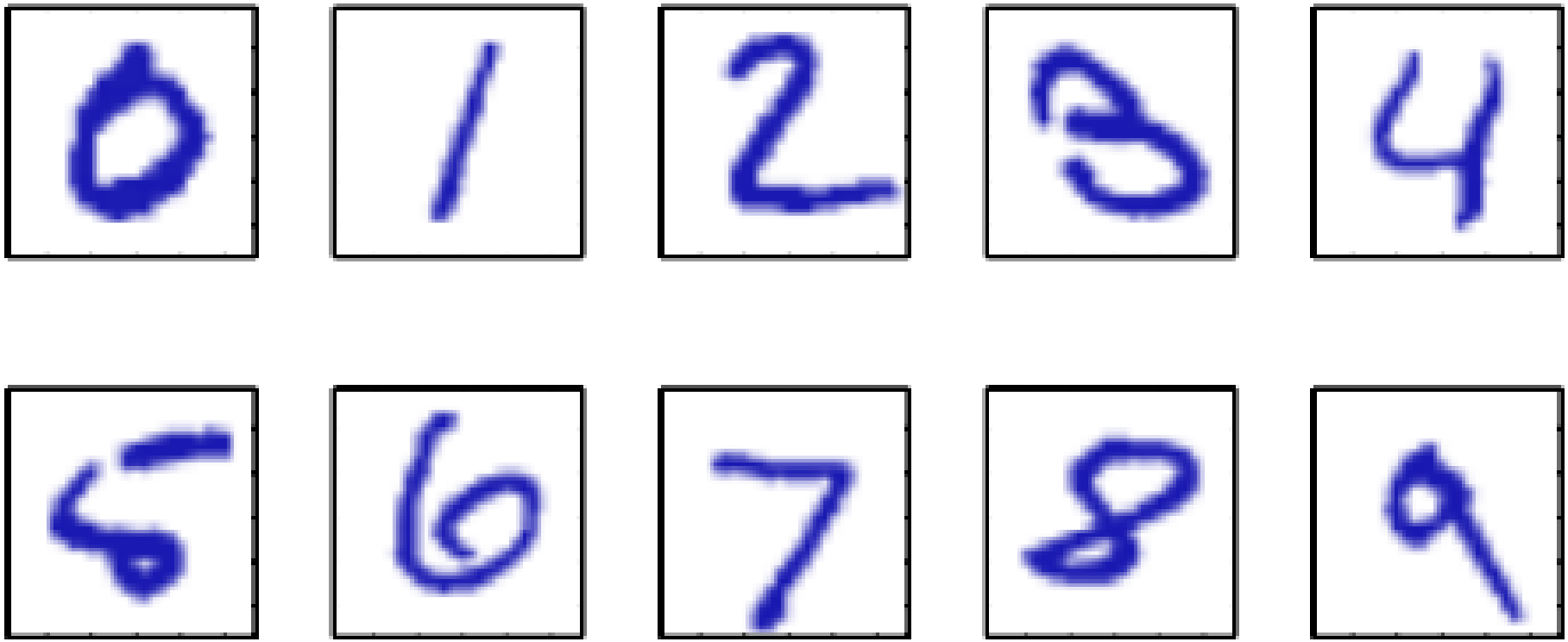


Hard problems for ML

Cyber security

Handwritten Digit Recognition

Commercial systems currently in use by USPS



How would you solve this problem?

Handwritten Digit Recognition

Problem

Each digit corresponds to a 28 x 28 pixel image and so can be represented by a vector x comprising 784 real numbers.
The goal is to build a machine that will take such a vector x as input and that will produce the identity of the digit 0,..., 9 as the output. This is a nontrivial problem due to the wide variability of handwriting. It could be solved in several ways:

- Craft rules or heuristics, such as shape of strokes. What are some shortcomings of this method?

Handwritten Digit Recognition

Problem

Each digit corresponds to a 28 x 28 pixel image and so can be represented by a vector x comprising 784 real numbers.
The goal is to build a machine that will take such a vector x as input and that will produce the identity of the digit 0,..., 9 as the output. This is a nontrivial problem due to the wide variability of handwriting. It could be solved in several ways:

- Craft rules or heuristics, such as shape of strokes. What are some shortcomings of this method? Uncontrolled growth of rules, invariably gives poor results.

Handwritten Digit Recognition

Problem

Each digit corresponds to a 28 x 28 pixel image and so can be represented by a vector x comprising 784 real numbers.
The goal is to build a machine that will take such a vector x as input and that will produce the identity of the digit 0,..., 9 as the output. This is a nontrivial problem due to the wide variability of handwriting. It could be solved in several ways:

- Craft rules or heuristics, such as shape of strokes. What are some shortcomings of this method? Uncontrolled growth of rules, invariably gives poor results.
- Adopt a ML based approach, where:
  - A large set of N digits $\{x_1, x_2, ..., x_n\}$ called a *training set* is used to tune the parameters of an *adaptive* method
  - The categories are known ahead of time, typically but not always, by hand-labeling them, expressed by a target vector t, which represents the identiy of the digit

ML Algorithm

- The result of running the machine learning algorithm can be expressed as a function $y(x)$ which takes a new digit image x as input and that generates an output vector y, encoded in the same way as the target vectors

ML Algorithm

- The result of running the machine learning algorithm can be expressed as a function $y(x)$ which takes a new digit image x as input and that generates an output vector y, encoded in the same way as the target vectors
- The precise form of the function y(x) is determined during the training phase, also known as the learning phase, on the basis of the training data

ML Algorithm

- The result of running the machine learning algorithm can be expressed as a function $y(x)$ which takes a new digit image x as input and that generates an output vector y, encoded in the same way as the target vectors
- The precise form of the function y(x) is determined during the training phase, also known as the learning phase, on the basis of the training data
- Once the model is trained it can then determine the identity of new digit images, which are said to comprise a test set

ML Algorithm

- The result of running the machine learning algorithm can be expressed as a function $y(x)$ which takes a new digit image x as input and that generates an output vector y, encoded in the same way as the target vectors
- The precise form of the function y(x) is determined during the training phase, also known as the learning phase, on the basis of the training data
- Once the model is trained it can then determine the identity of new digit images, which are said to comprise a test set
- The ability to categorize correctly new examples that differ from those used for training is known as generalization

ML Algorithm

- The result of running the machine learning algorithm can be expressed as a function $y(x)$ which takes a new digit image x as input and that generates an output vector y, encoded in the same way as the target vectors
- The precise form of the function y(x) is determined during the training phase, also known as the learning phase, on the basis of the training data
- Once the model is trained it can then determine the identity of new digit images, which are said to comprise a test set
- The ability to categorize correctly new examples that differ from those used for training is known as generalization
- In practical applications, the variability of the input vectors will be such that the training data can comprise only a tiny fraction of all possible input vectors, and so generalization is a central goal in pattern recognition

Input data

- For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve

Input data

- For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve
- For instance, in the digit recognition problem, the images of the digits are typically translated and scaled so that each digit is contained within a box of a fixed size

Input data

- For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve
- For instance, in the digit recognition problem, the images of the digits are typically translated and scaled so that each digit is contained within a box of a fixed size
- This pre-processing stage is sometimes also called feature extraction

Types of ML

- Supervised
  - Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as supervised learning problems
    - Regression: if the desired output consists of one or more continuous variables
    - Classification: mapping input vector to one of a finite number of discrete categories
- Unsupervised
  - In other pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization

Types of ML

- Reinforcement Learning
  - finding suitable actions to take in a given situation in order to maximize a reward
  - learning algorithm is not given examples of optimal outputs, in contrast to supervised learning, but must instead discover them by a process of trial and error

Example problem: polynomial curve fitting

Regression Problem

- suppose we observe a real-valued input variable x and we wish to use this observation to predict the value of a real-valued target variable t
- data for this example is generated from the function $sin(2\pi x)$ with random noise included in the target values
- generating data in this way, we are capturing a property of many real data sets, namely that they possess an underlying regularity, which we wish to learn, but that individual observations are corrupted by random noise
- noise might arise from intrinsically stochastic (i.e., random) processes such as radioactive decay but more typically is due to there being sources of variability that are themselves unobserved

Example problem: polynomial curve fitting

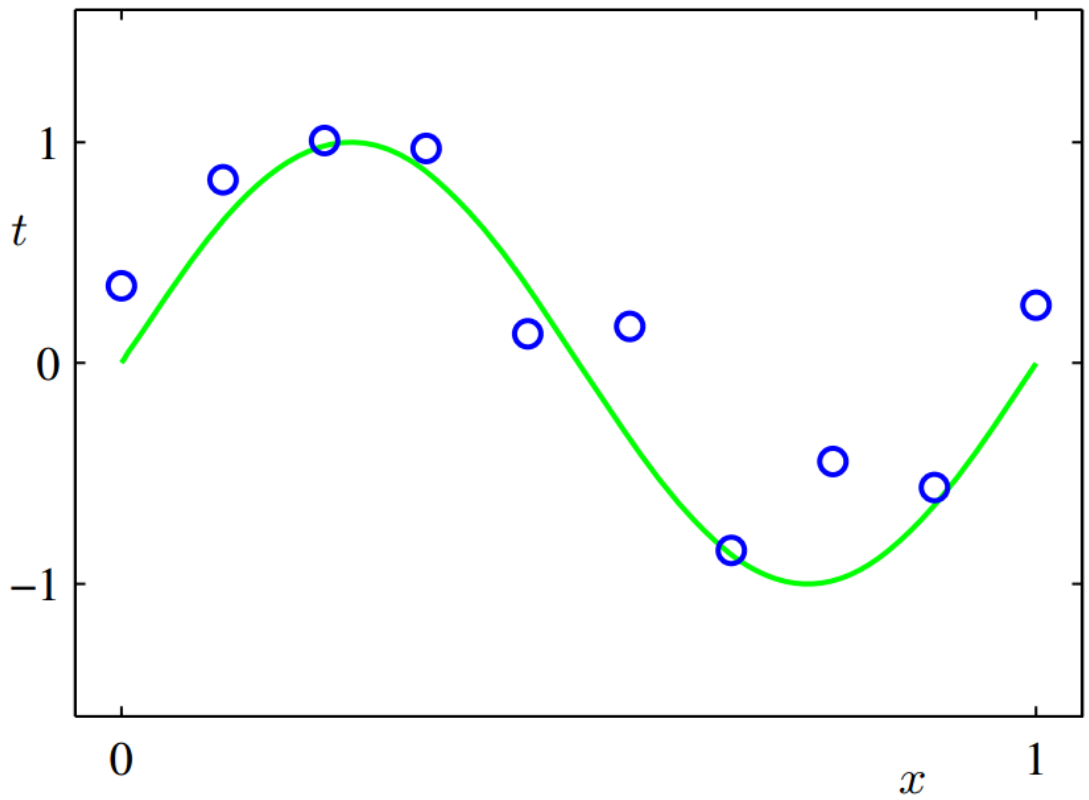
Regression Problem

- We are given a training set comprising of N observations of x, written as $x \equiv (x_1, x_2, ..., x_N)^T$ and
- observations of t, denoted $t \equiv (t_1, ..., t_N)^T$

Polynomial Curve Fitting

Sample data

Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t. The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the green curve.



Polynomial Curve Fitting

Goal

- Make prediction of the value $\hat{t}$, given an unobserved $\hat{x}$

Polynomial Curve Fitting

Goal

- Make prediction of the value $\hat{t}$, given an unobserved $\hat{x}$
- This involves discovering the underlying generating function $sin(2\pi x)$, with added noise

Polynomial Curve Fitting

Goal

- Make prediction of the value $\hat{t}$, given an unobserved $\hat{x}$
- This involves discovering the underlying generating function $sin(2\pi x)$, with added noise

Attempt

- Assume underlying function is a polynomial
- $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j$
- M is the order of the polynomial. The coefficients are $\mathbf{w}$
- Functions like the polynomial are **linear in the unknown parameters** and all fall under linear models

Polynomial Curve Fitting

How?

- The values of $\mathbf{w}$ will be determined by fitting the polynomial to the training data

Polynomial Curve Fitting

How?

- The values of **w** will be determined by fitting the polynomial to the training data
- Done by minimizing an **error function**

Polynomial Curve Fitting

How?

- The values of $\mathbf{w}$ will be determined by fitting the polynomial to the training data
- Done by minimizing an **error function**
- The error function measures the distance between the function $y(x, \mathbf{w})$, for any given value of w and the training dataset points

Polynomial Curve Fitting

How?

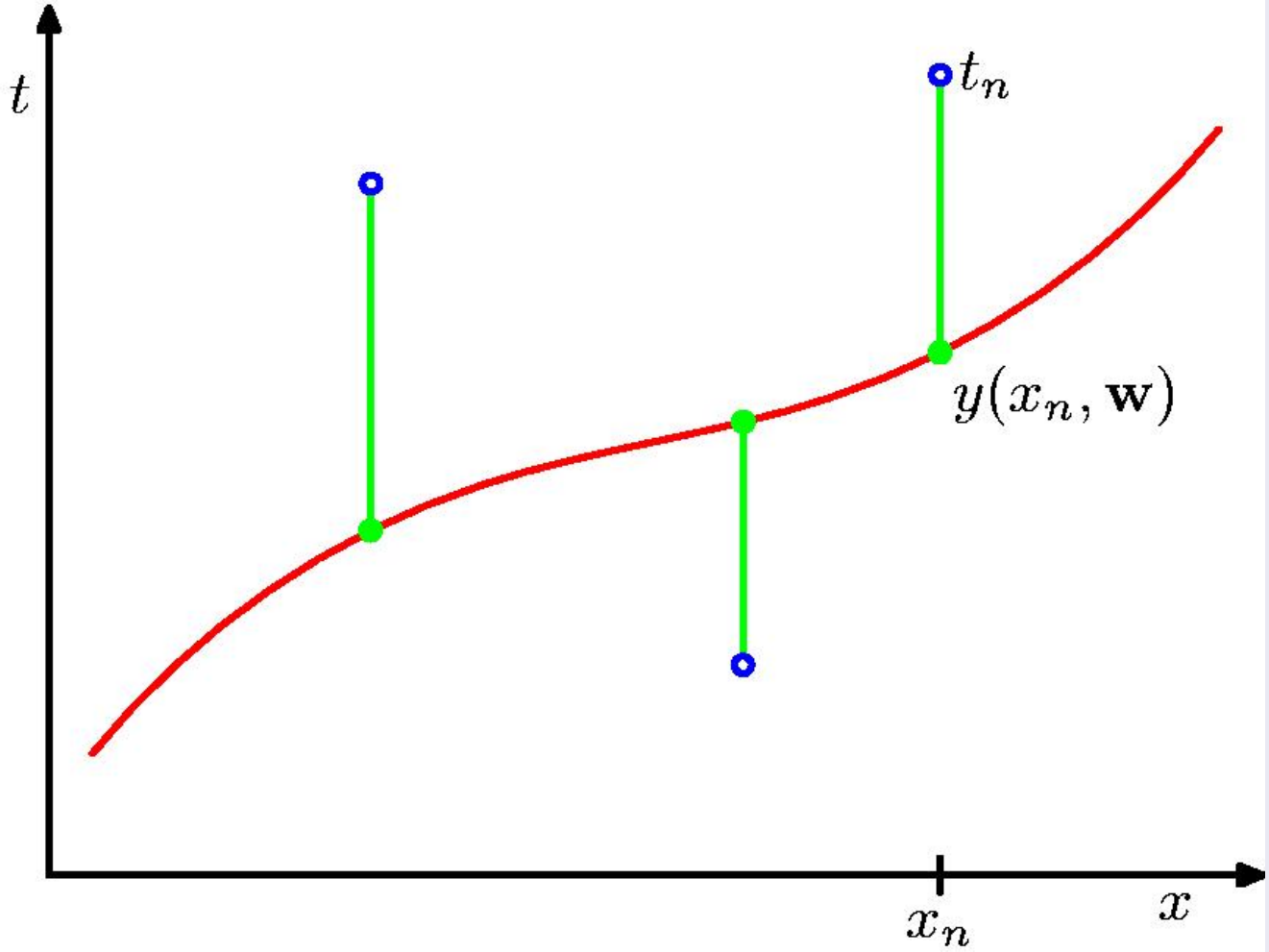
- The values of **w** will be determined by fitting the polynomial to the training data
- Done by minimizing an **error function**
- The error function measures the distance between the function $y(x, \mathbf{w})$, for any given value of w and the training dataset points
- Many choices for error function. One popular choice: sum of squares of errors between predictions and each known target value

Sum of squares

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

Polynomial Curve Fitting

Errors



Minimizing the error

Derivatives

- Choose $\mathbf{w}$ that makes the value of E as small as possible
- The resulting polynomial is $y(x, w^*)$