CS60010: Deep Learning

Sudeshna Sarkar

Spring 2018

9 Jan 2018

ML BASICS

Capacity, Underfitting and Overfitting in Polynomial Estimation

- The central challenge in machine learning – generalization
- ML: we want the generalization error (test error) to be low.

- The train and test data are generated by a probability distribution over datasets called the data generating process.

- We typically make i.i.d. assumptions.

Underfitting and Overfitting

- We sample the training set, use it to choose the parameters to reduce training set error, then sample the test set.
  - the expected test error is greater than or equal to the expected value of training error.
- The factors determining how well a machine learning algorithm will perform are its ability to:
  1. Make the training error small
  2. Make the gap between training and test error small

1. Underfitting occurs when the model is not able to obtain a sufficiently low error value on the training set.
2. Overfitting occurs when the gap between the training error and test error is too large

Capacity

- We can control whether a model is more likely to overfit or underfit by altering its capacity.
- A model's capacity is its ability to fit a wide variety of functions.
  - Models with low capacity may struggle to fit the training set.
  - Models with high capacity can overfit by memorizing properties of the training set
- One way to control the capacity of a learning algorithm is by choosing its hypothesis space.
  - Generalizing linear regression to include polynomials in its hypothesis space increases the model's capacity.

- A polynomial of degree one gives us the linear regression model

$$\hat{y} = b + wx$$

- Quadratic model

$$\hat{y} = b + w_1 x + w_2 x^2$$

- The output is still a linear function of the parameters, so we can still use the normal equations to train the model in closed form
- Polynomial of degree 10

$$\hat{y} = b + \sum_{i=1}^{10} w_i x^i$$

- ML algorithms generally perform best when their capacity is appropriate for
  - the true complexity of the task and
  - the amount of training data

Underfitting and Overfitting in Polynomial Estimation

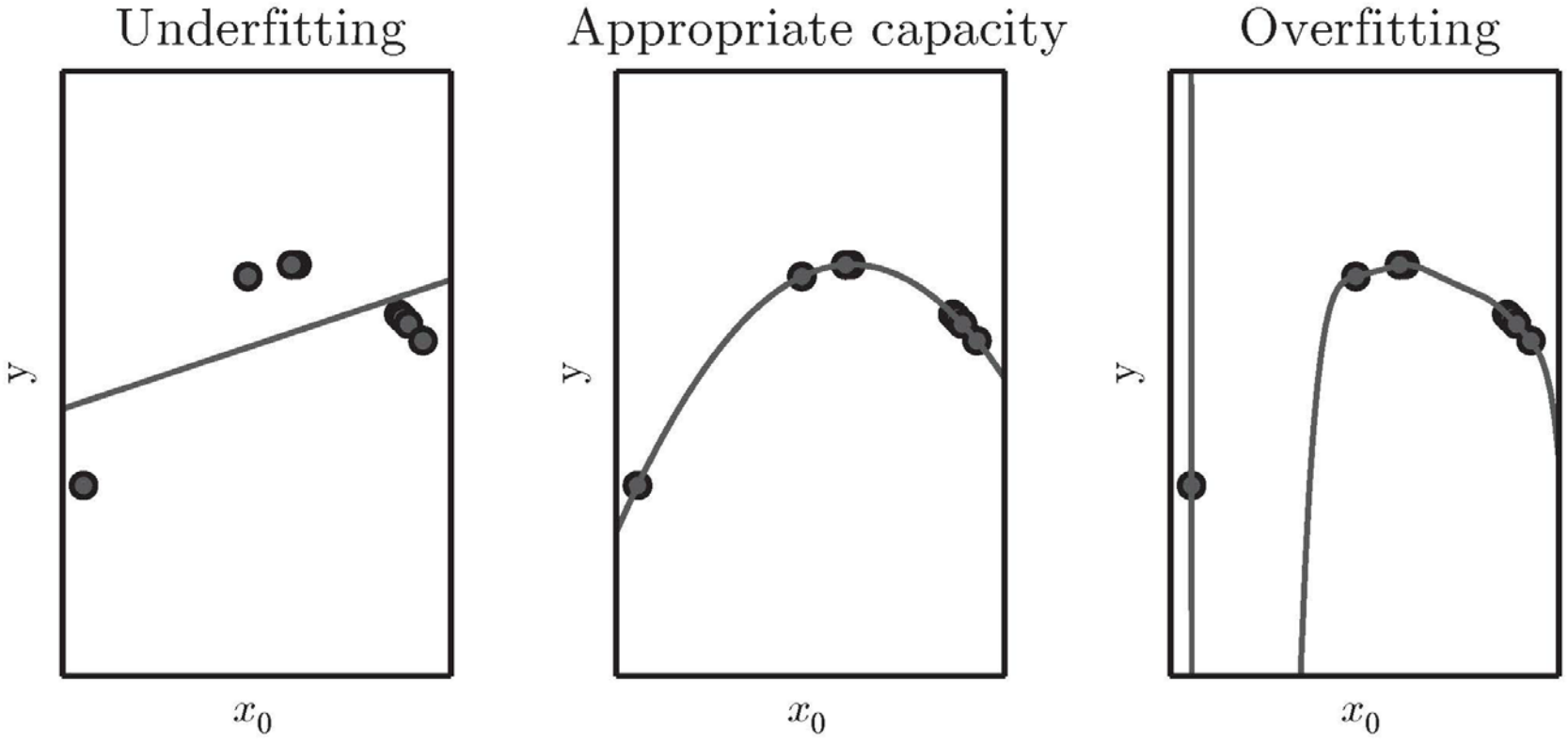

Figure 5.2

Capacity

- Many ways of changing a model's capacity.
  - changing the number of input features and adding corresponding parameters
- Representational capacity – the model specifies which family of functions the learning algorithm can choose from
  - Finding the best function within this family is an optimization problem
- Imperfection of the optimization
  - => the effective capacity may be less than the representational capacity of the model family

Occams Razor, VC Dimension

- Occam's razor (c. 1287-1347) among competing hypotheses that explain known observations equally well, one should choose the "simplest" one.
- Statistical learning theory provides various means of quantifying model capacity.
- The most well known is the Vapnik-Chervonenkis dimension, or VC dimension.
  - measures the capacity of a binary classifier.

- *The VC dimension is defined as being the largest possible value of m for which there exists a training set of m different x points that the classifier can label arbitrarily*

VC Dimension

- Quantifying the capacity of the model enables statistical learning theory to make quantitative predictions.

- Statistical learning theory shows that the discrepancy between training error and generalization error is bounded from above by a quantity that grows as the model capacity grows but shrinks as the number of training examples increases.

- The effective capacity is also limited by the capabilities of the optimization algorithm, and we have little theoretical understanding of the general nonconvex optimization problems involved in deep learning

Generalization and Capacity

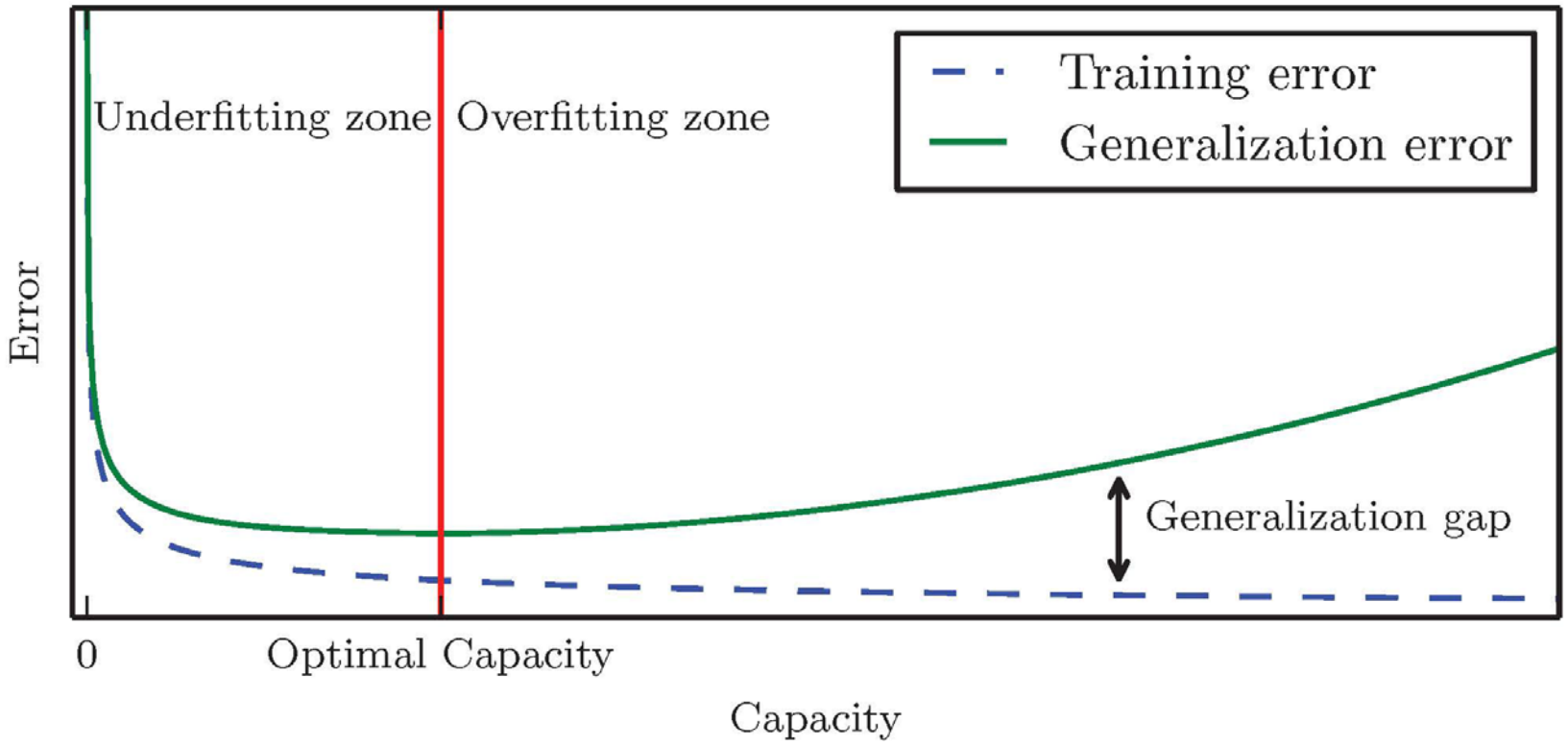

Figure 5.3

Non-parametric models

- Parametric models learn a function described by a parameter vector whose size is finite and fixed.
- Non-parametric models can have arbitrarily high capacity
  - Example: nearest neighbour regression

- We can also create a non-parametric learning algorithm by wrapping a parametric learning algorithm inside another algorithm that increases the number of parameters as needed.