ECE 469/ECE 568 Machine Learning

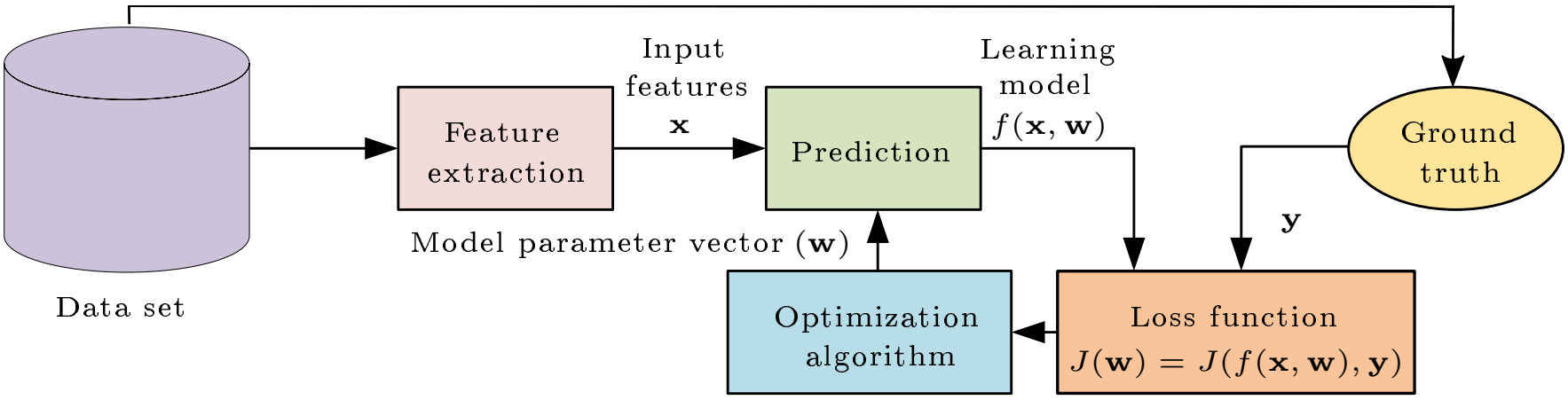
Textbook:
Machine Learning: a Probabilistic Perspective by Kevin Patrick Murphy

Southern Illinois University

September 9, 2024

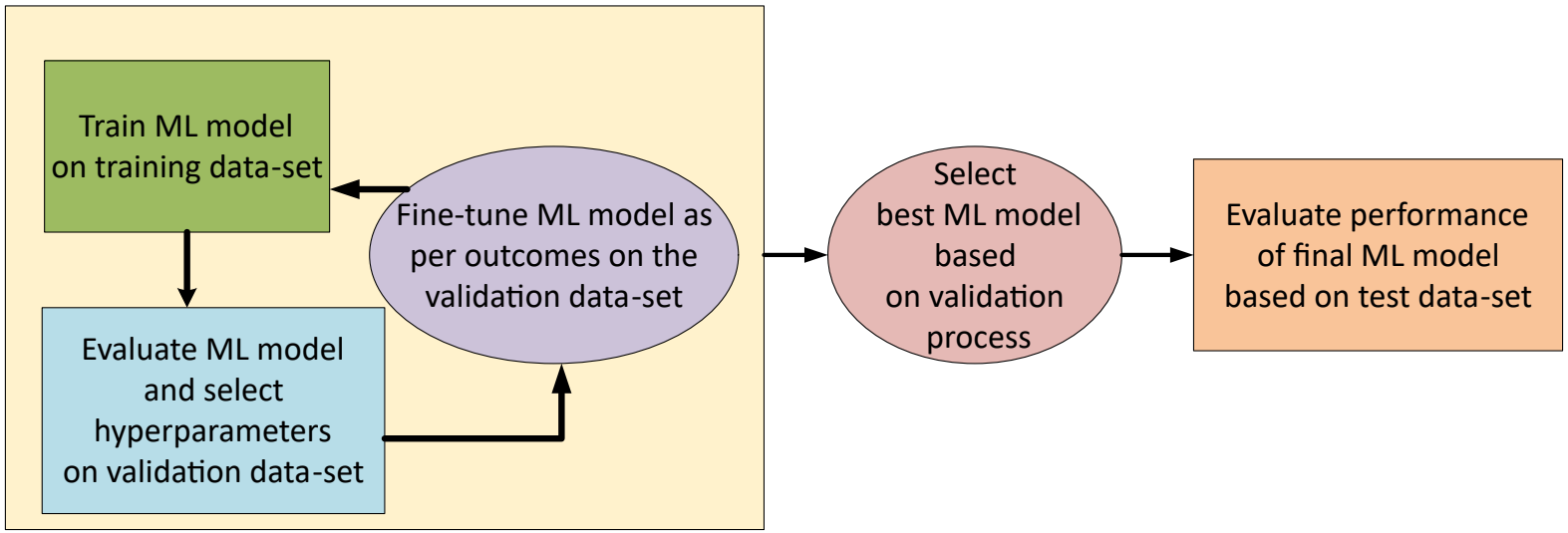
A recap for the last lecture

- The main processes in supervised machine learning is depicted below.



- In the last lecture, we discussed optimizers used in machine learning:
  - Batch gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent
  - Gradient descent with adaptive learning rates:
    - Momentum method
    - AdaGrad
    - RSMProp
    - Adam
- We also discussed implementing the underlying algorithms in popular ML frameworks: **Scikit – Learn**, **Keras** and **PyTorch**.

Training, validating, and testing of ML models



- In today's lecture, we are going to discuss:
  - Training, validation, and testing of machine learning models
  - Model comparison and selection
  - Underfitting vs. overfitting
  - Generalization and capacity
  - Regularization

Training, validation, and testing

- Typically, we split a date-set in machine learning into three portions:
  - Training set
  - Validation set
  - Test set

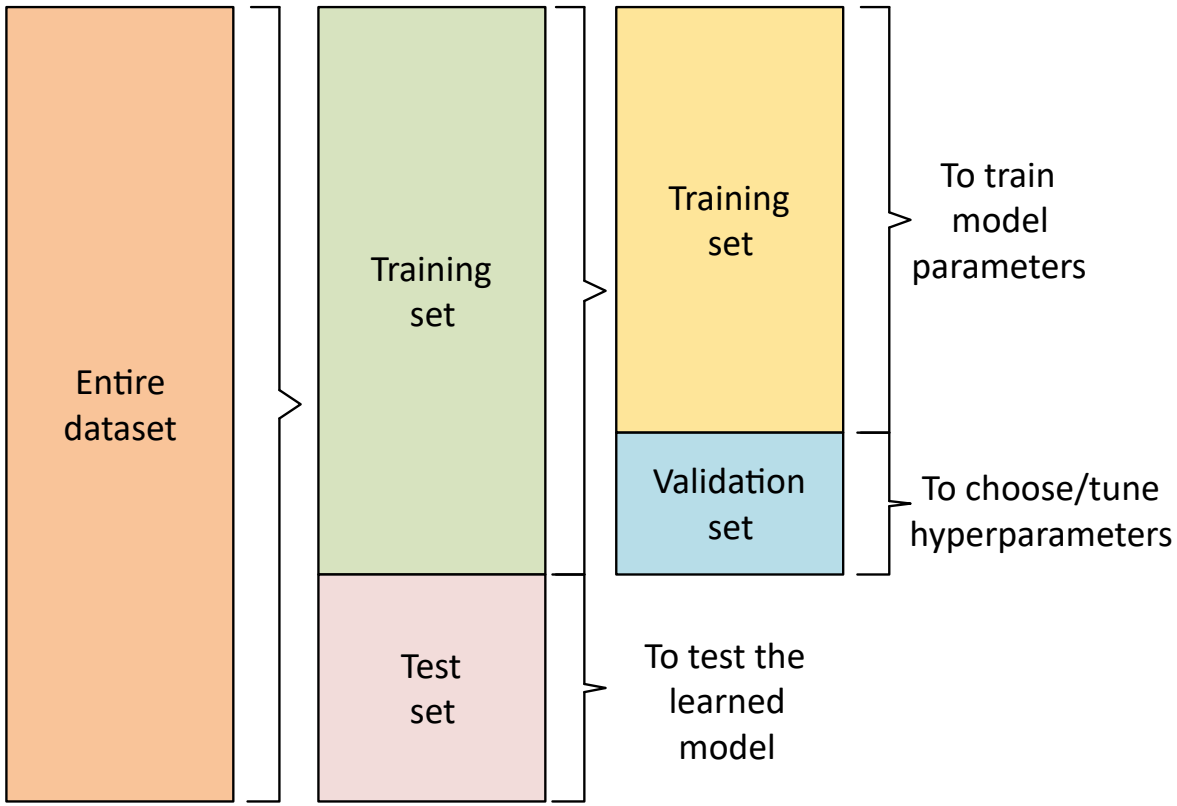


Figure: Splitting a data-set into training, validating, and testing sets

Training, validation, and testing

- Training set: This is a data set of examples/instances/samples used during the learning process to learn model parameters.
  - Example: Used to learn model parameters such as weights $(w_0, \cdots, w_m)$ in a linear regression model.

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{i=0}^{m} w_i x_i.$$

Training vs. validation sets

- We may split the training data into two disjoint subsets: –> One subset is used to learn the model parameters: –> Training set.
- The other subset is used to estimate the loss/error function during training, allowing for the hyperparameters to be selected: –> Validation set.
- The validation set can either be dedicated (hold-out validation) or shared across training data (cross-validation).

Training, validation, and testing

- Validation set: This is a set of data examples used to tune/select the hyperparameters of model architectures.
  - Example: Used to select the degree (M) of a non-linear polynomial model for a regression-based machine learning model

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

Validation set

- The subset of data used to guide the selection of hyperparameters is called the validation set.

  - Validation set: Again note that this set can be either dedicated (hold-out validation) or shared across training data (as in cross-validation).
  - Algorithms and their implementation for hold-out and cross validations will be discussed in Lecture 10 upon introducing regularization.

Training, validation, and testing

- Test data-set: This is a set of dedicated data reserved for testing the learned ML models.
  - Independent from training/validating sets
  - Follows the same probability distribution as the training data-set
  - Used to provide an unbiased evaluation of the performance of a final ML model learned on the training data-set and fine-tuned upon validation

Test set

- A test set is comprised of examples coming from the same distribution as the training set.
- A test set can be used to estimate the test/generalization error of a learner, after the learning process has completed.
- Test/generalization error is used to evaluate the performance of ML models.
- Test set must be dedicated and it is never shared across training/validation processes.

Training, validation and testing

Validation vs. testing

- Machine learning literature sometimes reverses the meanings of validation and test sets.
- This is a blatant example of the terminological confusion in modern machine learning.
- Remember that validation is used to select/fine-tune hyperparameters, while test-set is used to estimate the final test/generalization error of an optimized machine learning model.

ML model comparison and model selection

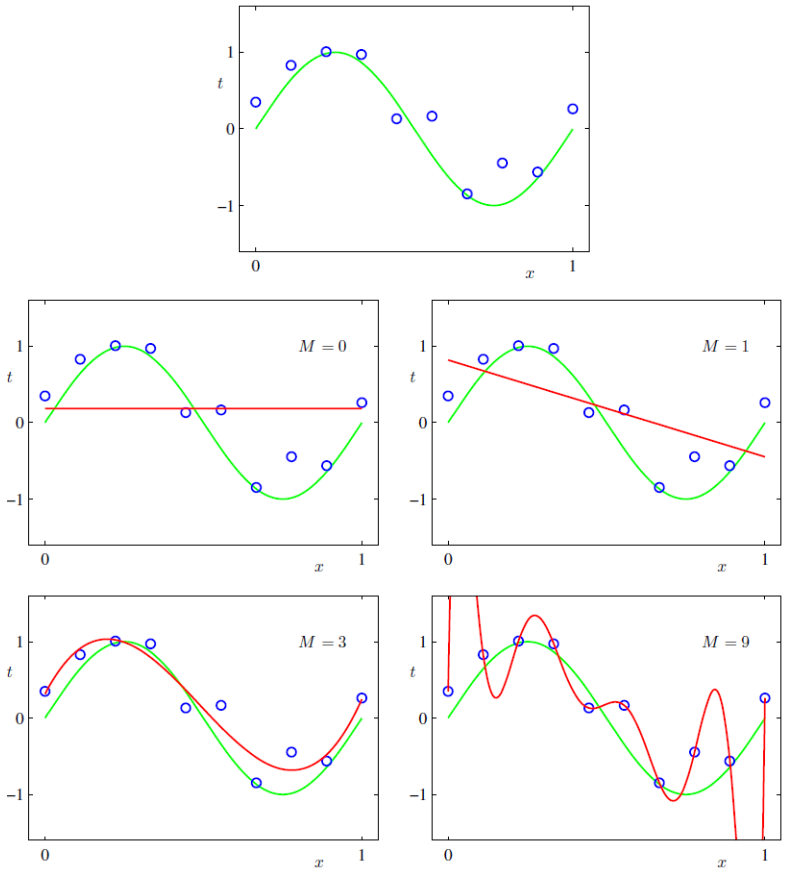
- For a given hypothesis class, there are multiple choices for a ML model.
- For example, consider a polynomial class of degree M.

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

- There remains the problem of choosing the degree M of the polynomial.

ML model comparison and model selection

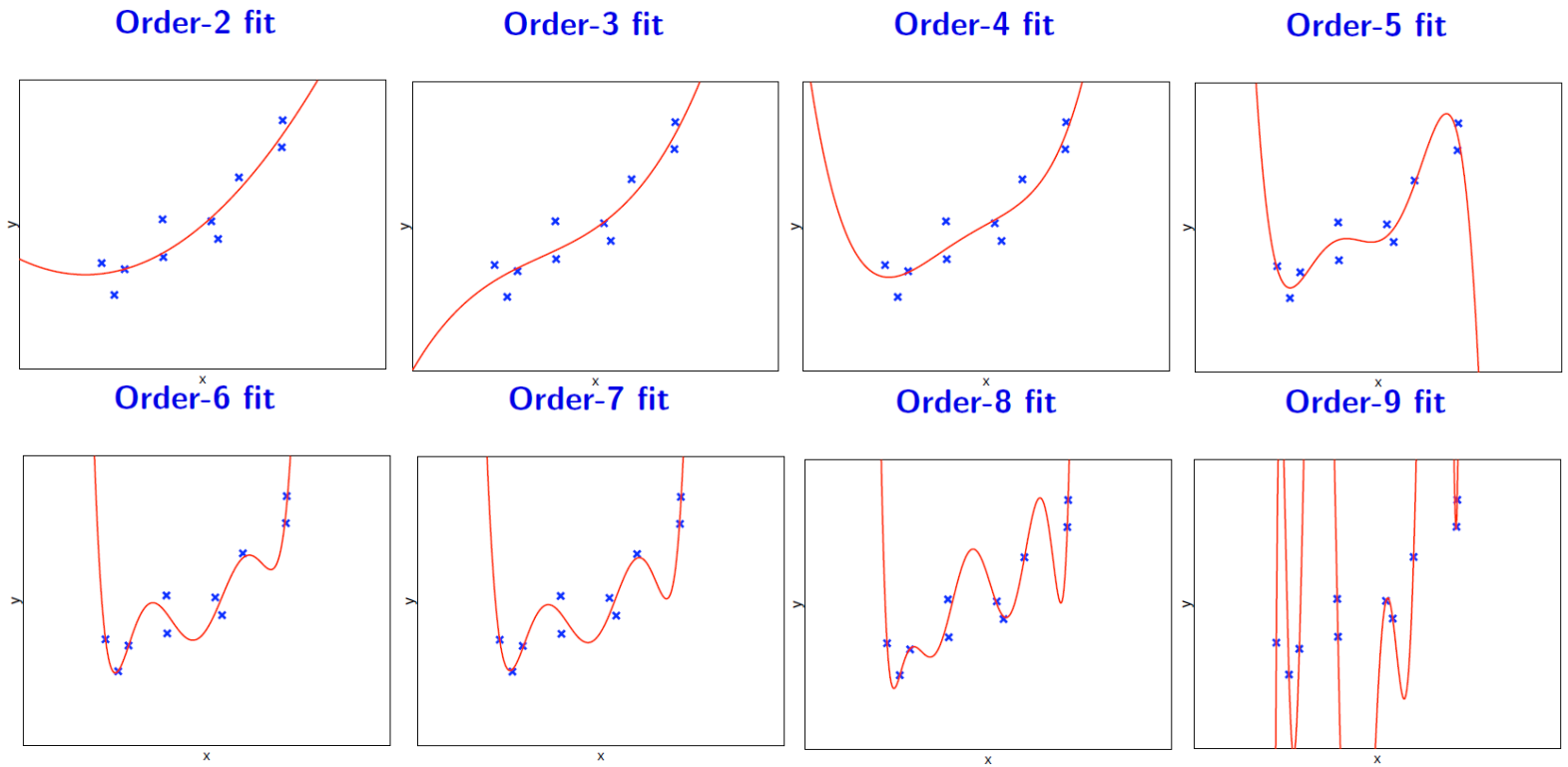
- Let us consider the following example.



- The blue dots represent the data points, green curve represents the best fit, and the red curves represent a set of probable models in the hypothesis class.

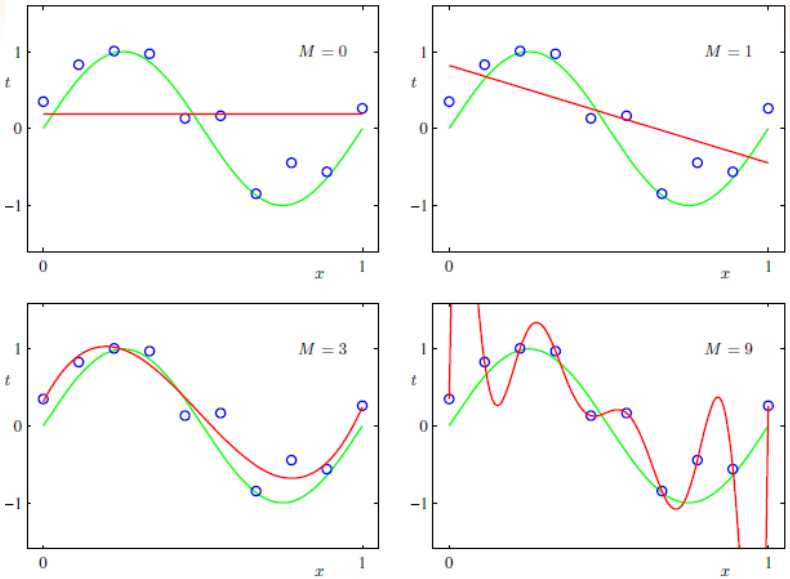
ML model comparison or model selection

- A systematic approach is needed to compare and select ML models.



Which order is a better fit?

ML model comparison or model selection



- Notice that the zero-order/constant ($M = 0$) and first order ($M = 1$) polynomials give poor fits to the training data. –> This behavior is known as underfitting.
- The third order ($M = 3$) polynomial seems to give the best fit.
- we obtain an excellent fit to the training data when we go to a much higher order polynomial ($M = 9$).
- But, the fitted curve with $M = 9$ oscillates wildly and gives a very poor representation of the actual function. –> This behavior is known as overfitting.

ML model comparison or model selection

- Recall our mean squared loss function:

$$J(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left[y_n - f(\mathbf{x}_n, \mathbf{w}) \right]^2 ,$$

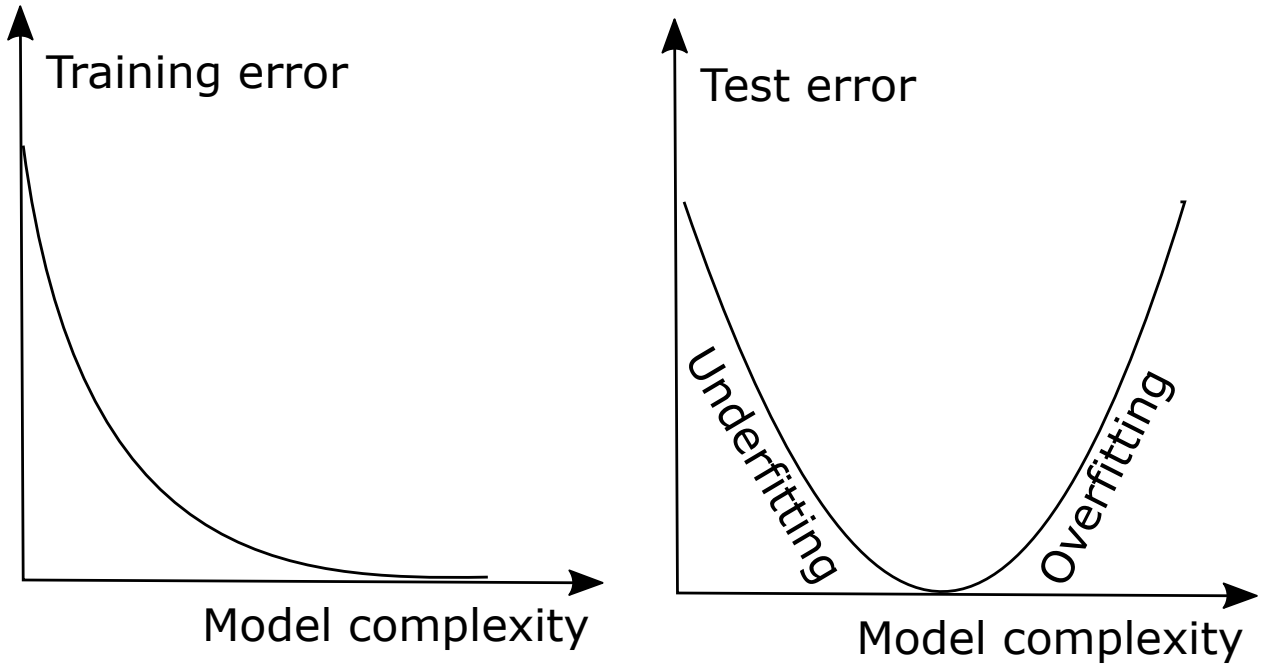
  where $f(\mathbf{x}_n, \mathbf{w})$ is the ML model, and $\mathbf{w}$ is the weight vector that needs to be learned.

- Here, $\mathbf{x}_n$ is the nth row of the input feature set, and y_n is its corresponding output. Here, N is the data-set size.
- When this loss function is evaluated on the training set, we term it as the training error.
- When this loss function is evaluated on the validation set, we term it as the validation error.
- When this loss function is evaluated on the test set, we term it as the test error.
- We would like to minimize both training and test error types in ML algorithms by means of efficient training and validation.

ML model comparison or model selection

- The training error always decreases with increased model complexity, for example, the degree of the polynomial (M).
- When the model is very shallow in complexity, the learned model will neither fit the existing data nor predict the output for new data accurately –> leads to underfitting problem in ML.
- When the model is very complicated, the learned model tends to memorize only the existing data and will not predict the output accurately for new/unseen input data –> leads to overfitting problem in ML.
- Hence, the test error first decreases with increasing model complexity.
- But the test error increases beyond a certain model complexity (when the model is overly complicated) and the learned model losses its generalization capacity.

ML model comparison or model selection



- Both underfitting and overfitting are problematic in ML algorithms.
- There is a fundamental trade-off between the underfitting and overfitting.

ML model comparison or model selection

- Let us first focus on overfitting.
- We can find a model for a given hypothesis class that predicts perfectly the training data but does not generalize well to new data.
- If we have an overly complicated model with a lot of parameters (ex., a higher order polynomial), the hypothesis "memorizes" training data points, but is wild everywhere else.
- Higher the degree of polynomial (M), the more degrees-of-freedom to fit all data points in the training set.
- Typical overfitting scenario exhibits the error on training data is very low (very low training error), but error on new instances is high (high test error).

Overfitting versus underfitting

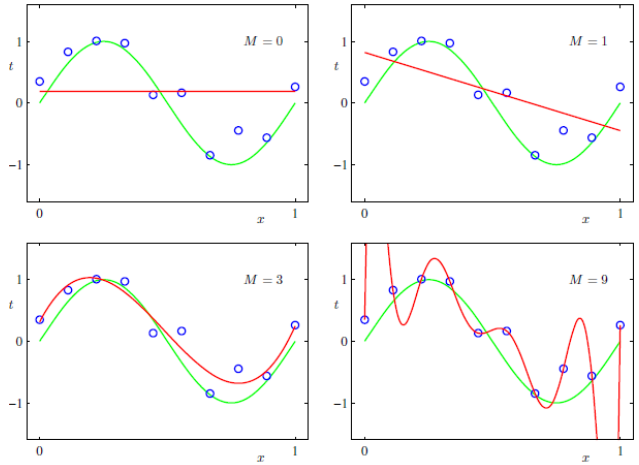
- Shallow ML models tend to underfit training data and will not predict accurately for unseen input features.
- Typical underfitting means that error on the training data is very high (or a few degrees-of-freedom available in the model).

Overfitting versus underfitting

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

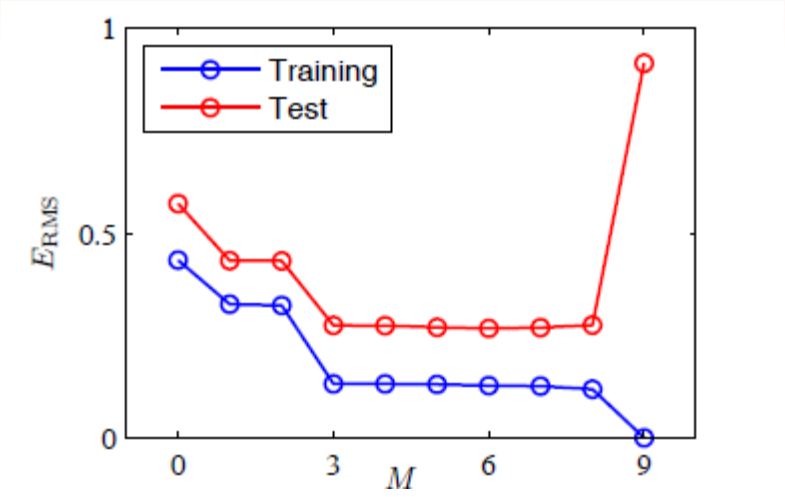
Overfitting vs. underfitting

Example: underfitting versus overfitting



- For each choice of degree M, we can evaluate the mean square error (MSE) denoted by $J(\mathbf{w})$ for the training data: –> This provides a measure for the "training error".
- Then we evaluate the same for test data set: –> This is "test error".
- Sometimes, it is more convenient to use the root-mean-square $E_{\mathrm{RMS}} = \sqrt{J(\mathbf{w})}$.
  an equal footing.
- The square-root also ensures that RMS error is measured on the same scale (i.e., in the same units) as the target/output variable $\mathbf{y}$.

Training error versus test error



- The training error decreases with the degree of the polynomial M (i.e., the complexity of the ML model).
- The testing error, measured on independent data, decreases at first, then starts increasing.
- A validation would help us: –> We can find a good model inside a given hypothesis class (i.e., M in this example) by using a "validation set".
- Then we can report unbiased results by using a test set, which is in theory untouched during either parameter training or validation.

Generalization and Capacity

- A key challenge in machine learning is that a ML model must perform well on new, previously unseen inputs, not just those on which our model was trained.

Generalization

The ability to perform well on previously unobserved inputs is called generalization.

- Generalization is measured via test error.
- Hence, test error is also known as the generalization error.