Overfitting and Generalization

Chapter 5

Cesar Acosta, PhD.

OVERVIEW

- Overfitting
- Hyperparameters
- Holdout cross-validation
- Generalization
- learning curves
- Ways to improve model performance
- Regularization methods

5.1 OVERFITTING

A machine learning model starts overfitting
when it begins to learn patterns that are specific to the train data
but are misleading or not relevant to new data

PARAMETERS AND HYPERPARAMETERS

Parameter

Unknown coefficient that can be estimated from the data

Example: Coefficients from linear regression model

PARAMETERS AND HYPERPARAMETERS

Hyperparameter

Unknown coefficient that cannot be estimated from the data

Examples: KNN (n. neighbors), RR (shrinkage parameter α)

Tree (depth), RF (n. trees), GB (learning rate)

Process of choosing best hyperparameter value is called tuning

MODELS WITH NO HYPERPARAMETERS

To avoid overfitting dataset should be split into subsets

If model has no hyperparameters, it should be split into 2 subsets

- Train set (dataset to build the model)
- Test set (dataset to test the model)

MODELS WITH HYPERPARAMETERS

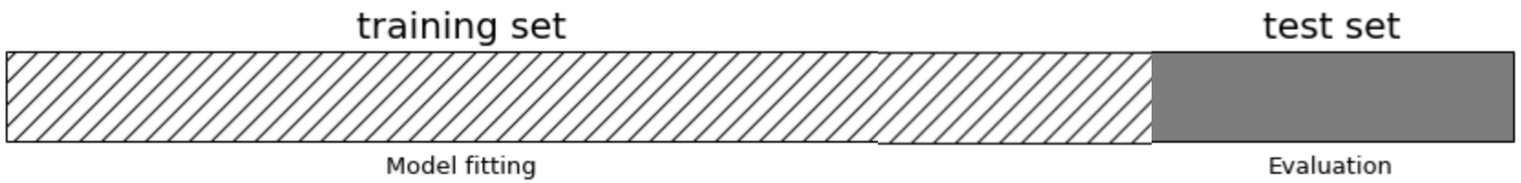
To avoid overfitting dataset should be split into subsets

If model has no hyperparameters, it should be split into 3 subsets

- Train set (dataset to build the model)
- Validation set (dataset for tuning hyperparameters)
- Test set (dataset to test the model)

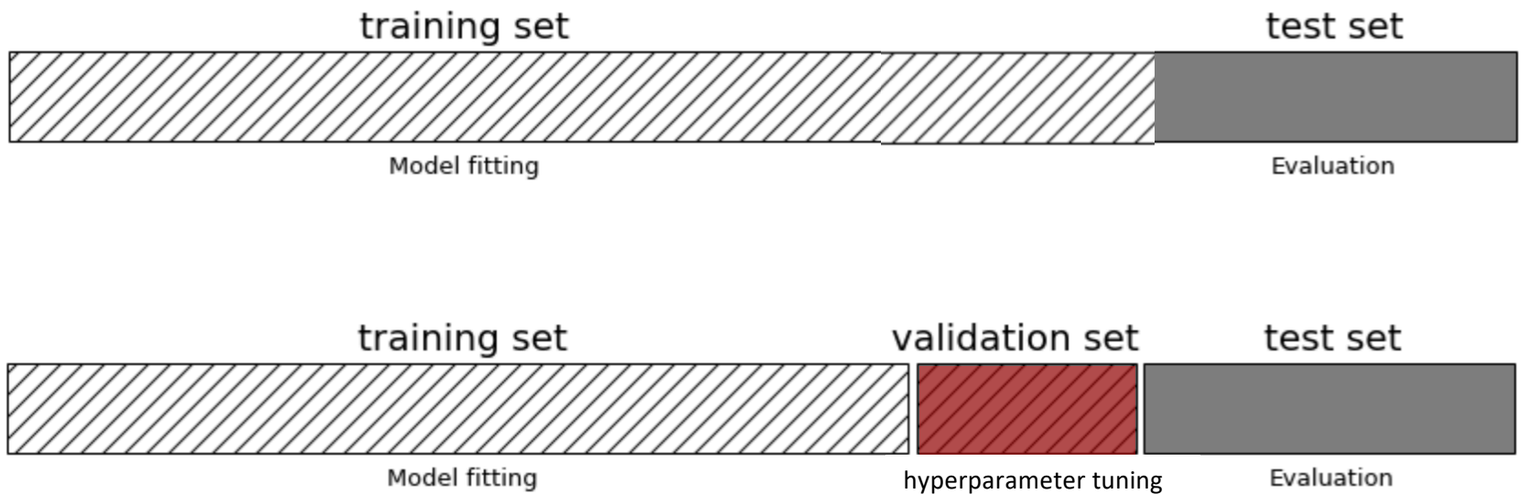
HOLDOUT CROSS VALIDATION – No hyperparameters

.



HOLDOUT CROSS VALIDATION – For hyperparameter tuning

.



MODEL PERFORMANCE MEASURES

REGRESSION

- Train MSE
- Validation MSE
- Test MSE

CLASSIFICATION

- Train accuracy rate
- Validation accuracy rate
- Test accuracy rate

MODEL PERFORMANCE MEASURES

REGRESSION	CLASSIFICATION	BOTH
• Train MSE	• Train accuracy rate	• Train Loss
• Validation MSE	• Validation accuracy rate	• Validation Loss
• Test MSE	• Test accuracy rate	• Test Loss

METRICS AND LOSS

3.6.3 The "compile" step: Configuring the learning process

Once the model architecture is defined, you still have to choose three more things:

- *Loss function (objective function)*—The quantity that will be minimized during training. It represents a measure of success for the task at hand.
- *Optimizer*—Determines how the network will be updated based on the loss function. It implements a specific variant of stochastic gradient descent (SGD).
- *Metrics*—The measures of success you want to monitor during training and validation, such as classification accuracy. Unlike the loss, training will not optimize directly for these metrics. As such, metrics don't need to be differentiable.

METRICS AND LOSS

3.6.3 The "compile" step: Configuring the learning process

Once the model architecture is defined, you still have to choose

- *Loss function (objective function)*—The quantity that will be minimized during training. It represents a measure of success for the task at hand.

5.1 OVERFITTING AND GENERALIZATION

- GENERALIZATION

  The purpose of a machine learning model is to generalize (to perform accurately on new data)

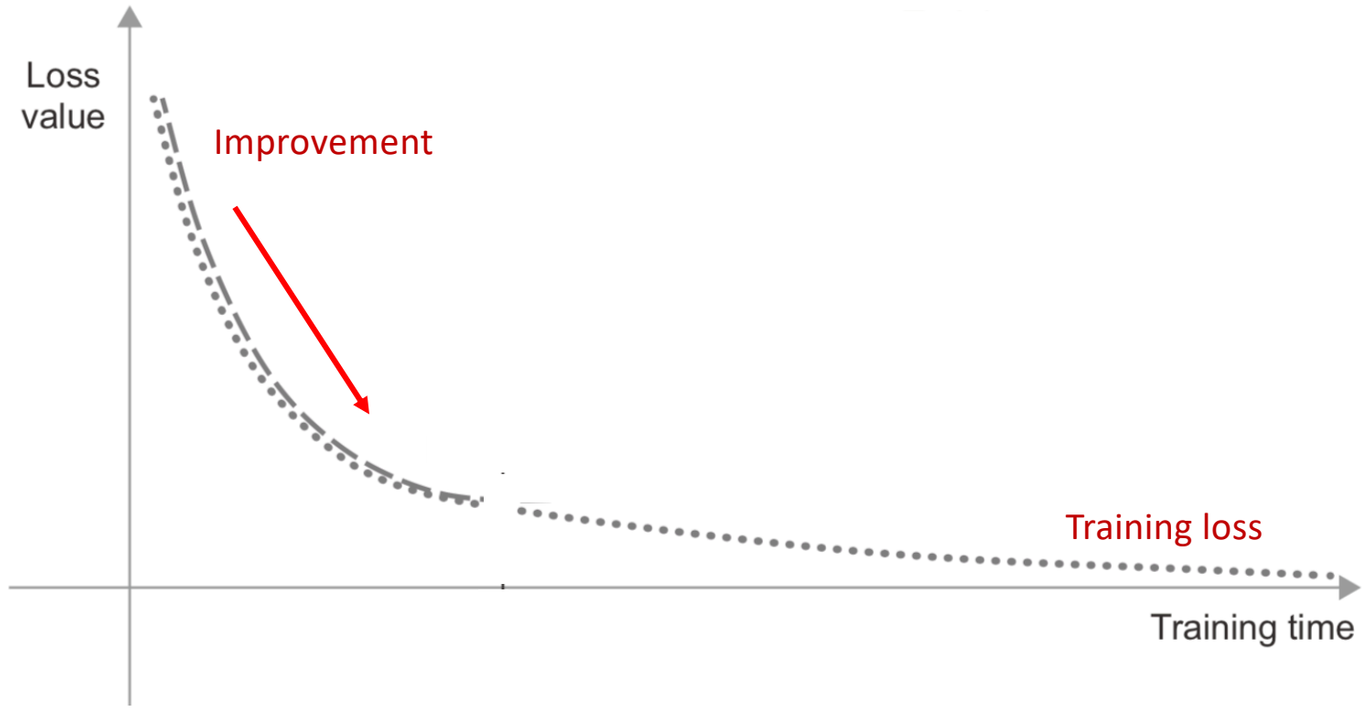
- OVERFITTING

  ✓ How to identify overfitting?

  ✓ How to prevent overfitting?

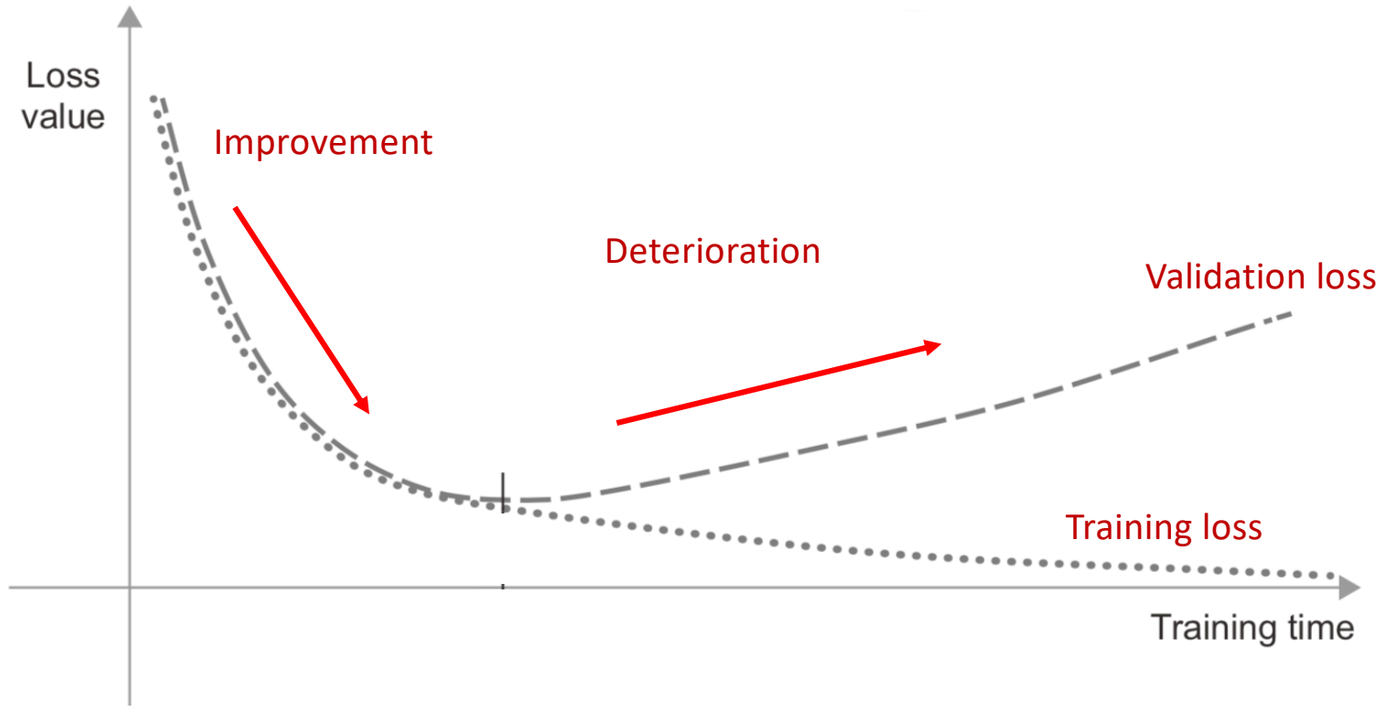
5.1 LEARNING CURVES (p122)

- *Model performance on the train set improves with the Training time, always*



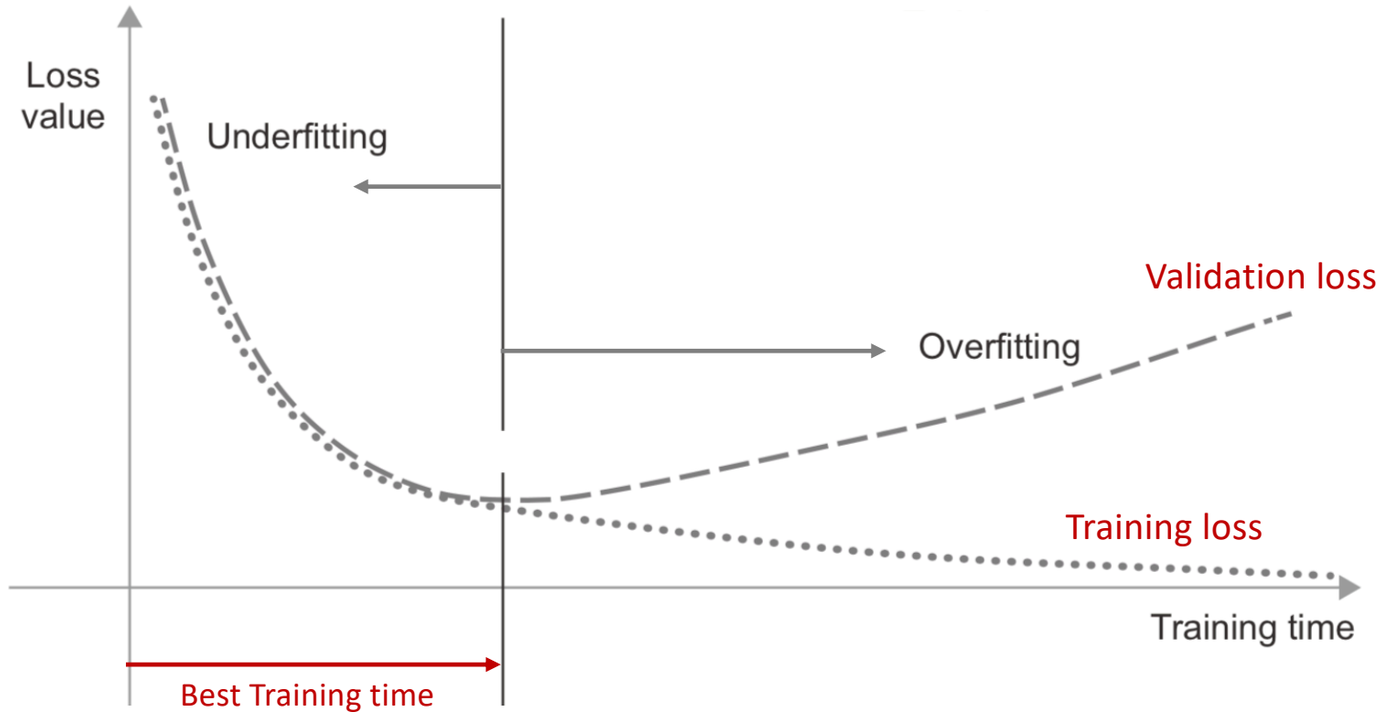
5.1 LEARNING CURVES (p122)

- *Model performance on the train set improves with the Training time, always*
- *Model performance on the validation set improves then deteriorates as the Training time increases*



5.1 LEARNING CURVES (p122)

- *Model performance on the train set improves with the Training time, always*
- *Model performance on the validation set improves then deteriorates as the Training time increases*
- *Use Learning Curves to identify the training time that prevents overfitting*



5.4 HOW TO PREVENT OVERFITTING

- Increase the size of the train dataset
- Adjust the model size
  - simpler models are likely to underfit
  - complex models are likely to overfit
- Weight regularization (L1, L2) useful for small DL models
- Dropout useful for large DL models