Hyperparameter optimization

Grid-search, random-search and other methods for hyperparameter-search

see Ch. 05 in book “Python Machine Learning” by Raschka & Mirjalili

Overview

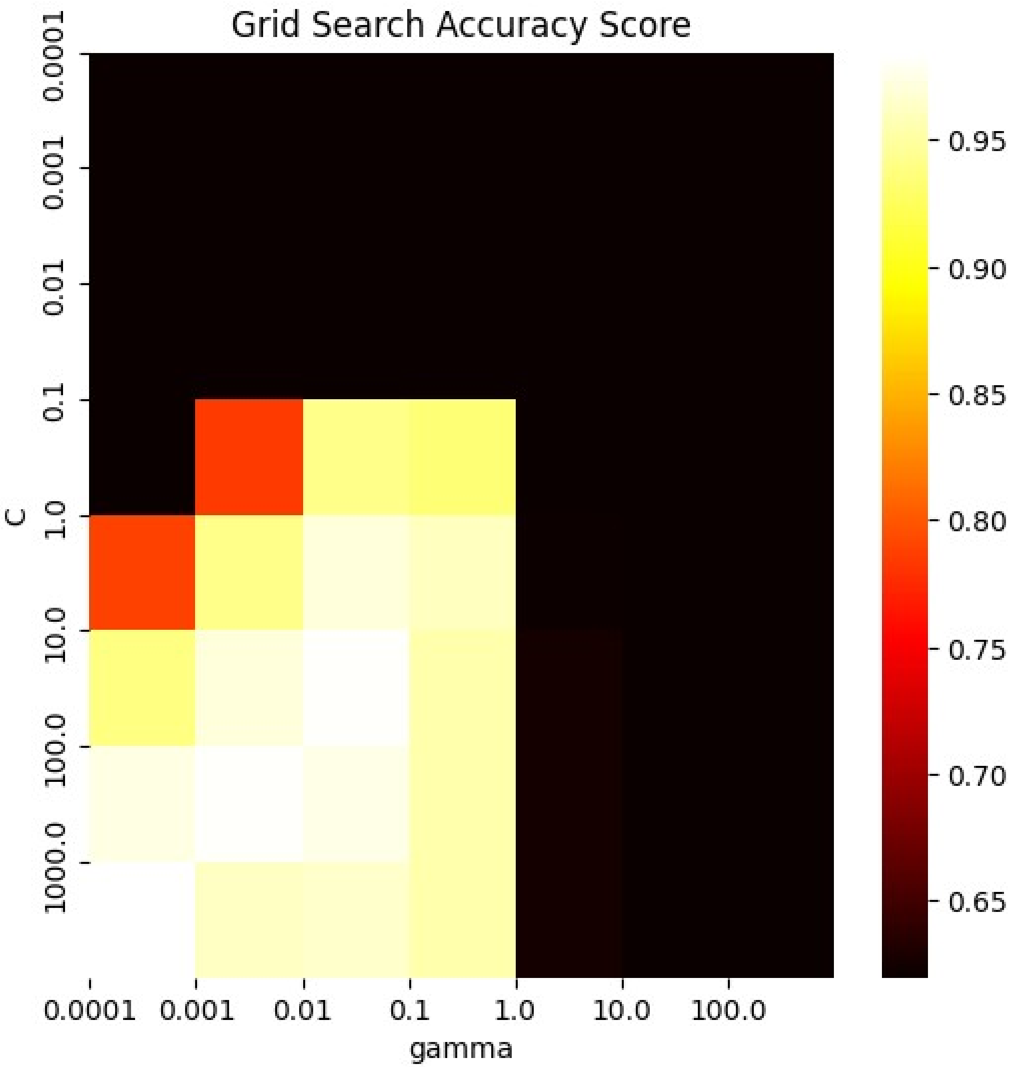
- GridSearchCV
- RandomizedSearchCV
- Nested cross-validation

Grid Search

- Brute-force exhaustive search through grid of possible hyperparameters
- Proper validation of each model-hyperparameter-version required to avoid overfitting
- Still need to make some choices:
  - Parameters to tune
  - What range of parameters to search through
  - What metric will they be scored after
  - What type of validation will be used
- Any suggestion as to why we do a grid-search and don’t optimize one parameter at a time?

GridSearchCV

- Setup with:
  - Pipeline
  - Parameters
  - Scoring
  - Cross-validation
- Returns all model scores for the specified parameter grid

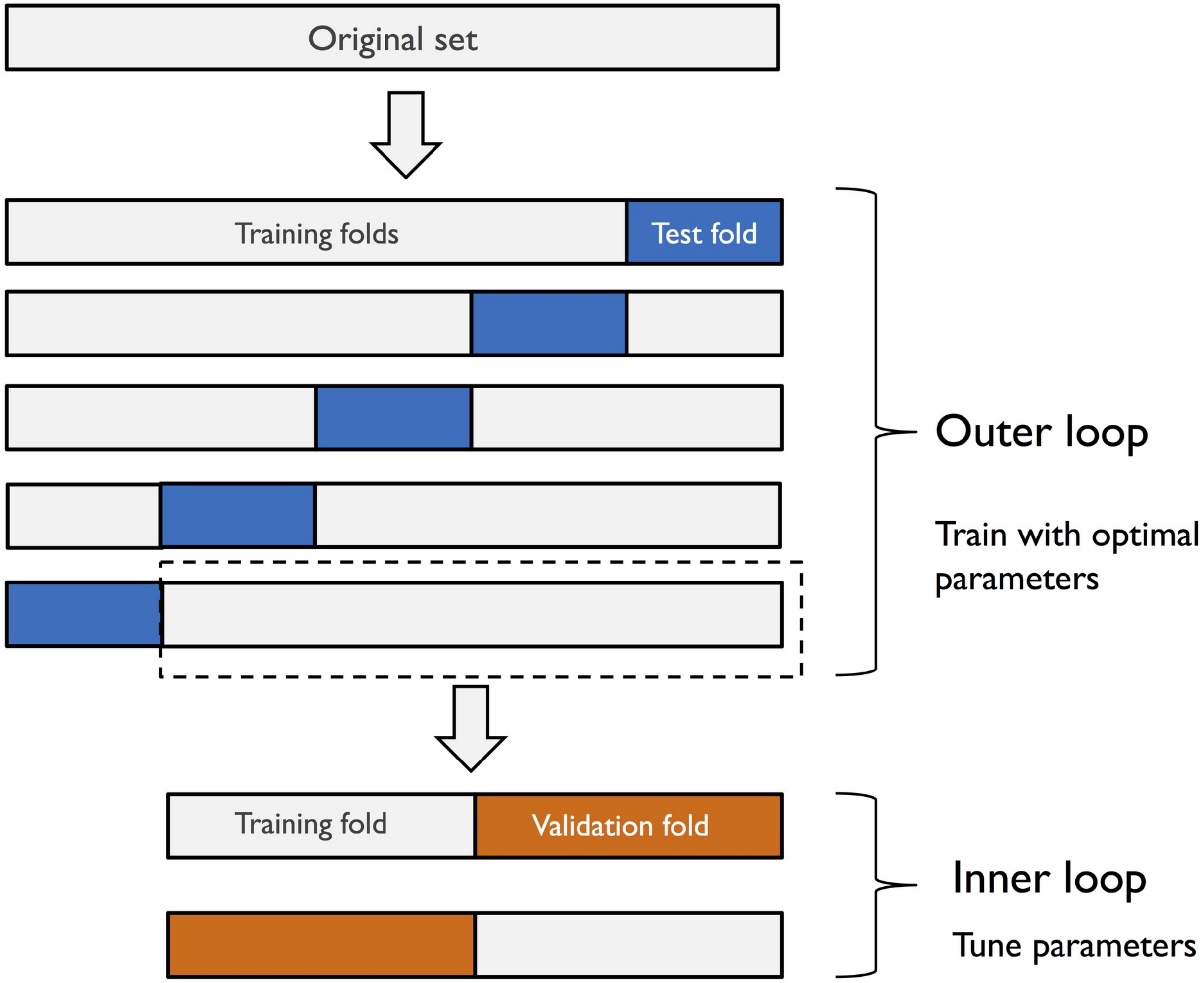


RandomizedSearchCV

- We don’t specify a grid of hyperparameter combinations to search exhaustively
- We specify a set of possible values for each parameter, often a continous range
- At each iteration parameter values are sampled randomly
- Each range of possible parameter-values also needs a corresponding **probability distribution**
  - If no distribution is specified, samples are drawn from uniform distribution
- We also specify a max set of iterations for the search
- Can anyone think of any advantages of random-search over exhaustive grid search?

Nested cross-validation

- Cross-validation loop within a cross-validation loop
- Addresses the fact that the initial split between the training/val set and test set is also sensitive to how the split is done
- Becomes very computationally expensive
- Is rarely done when you are working with datasets of over a certain size



Thank you for listening