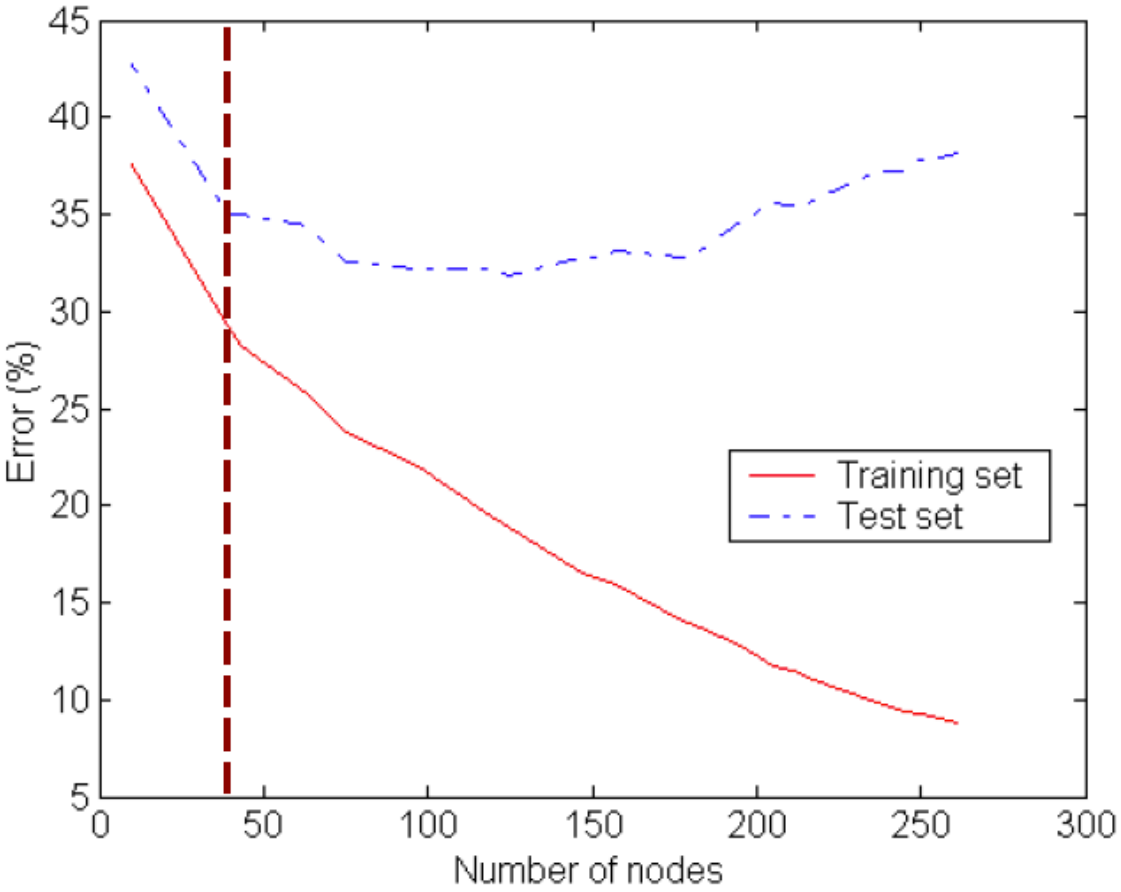
Hyperparameter Tuning

Exercise 6

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Overfitting

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics



Overfitting
Task

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

You are optimizing a deep neural network with a huge number of hyperparameters (layers, hidden size, ..). You have a fairly large dataset, but nearly infinite computing resources. So you decide to split your dataset into a training and a test set. After an exhaustive search, you choose the combination of hyperparameters that performs best on your test set.

As soon as you apply your model in practice, you notice that the performance is considerably worse than the performance on your test set. What could be reasons for this observation?

Overfitting
Solution

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Most obvious reason:

Not using a validation set can lead to an (indirect) overfitting of the model to the test set via hyperparameters

But there might also be other reasons like:

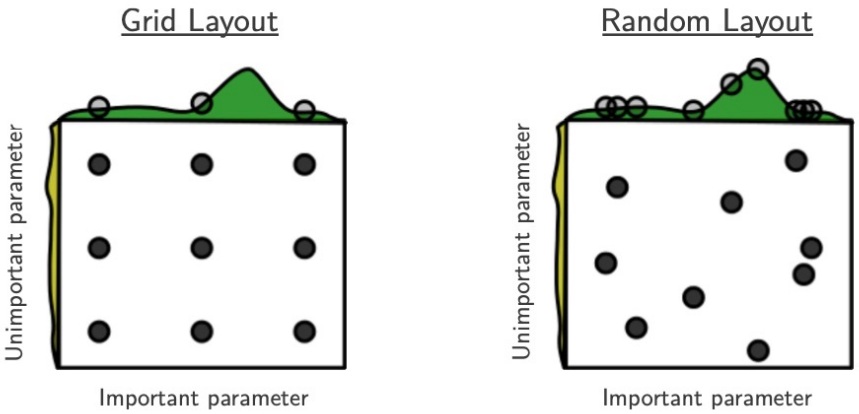
- Concept Drift (i.e., the data is outdated or does not fit)
- Incorrect application of the model (e.g. due to incorrect preprocessing of the real-world data)

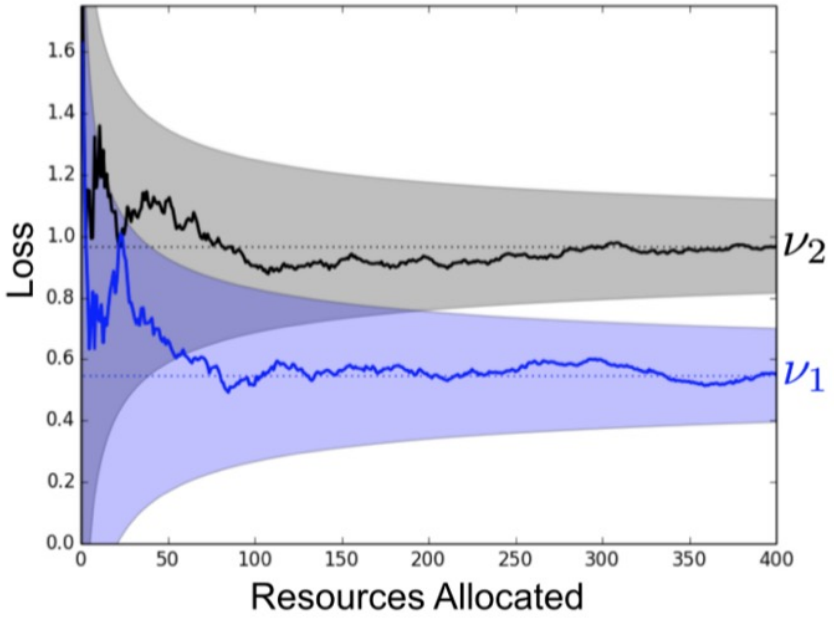
Hyperparameter Tuning
Search Strategies

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

- Brute Force Search
  - try out **all** hyperparameter combinations
  - computationally impossible; “blind” evaluation of parameters
- Grid Search
  - manually restrict search space to certain parameter combinations
  - quality of solution strongly dependent on grid definition
- Hill-climbing Heuristics
  - find (near-)optimal solution through evaluation of neighborhood
- Genetic Algorithms
  - find (near-)optimal solution through cross-over and mutation

Hyperparameter Tuning
Successive Halving





Successive Halving

1. Sample set of hyperparameter configurations
2. Evaluate performances of current configurations
3. Sort by performance and throw out bottom half
4. Go back to 2. and repeat until one config remains

Figure 1: http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf
Figure 2: https://arxiv.org/abs/1603.06560

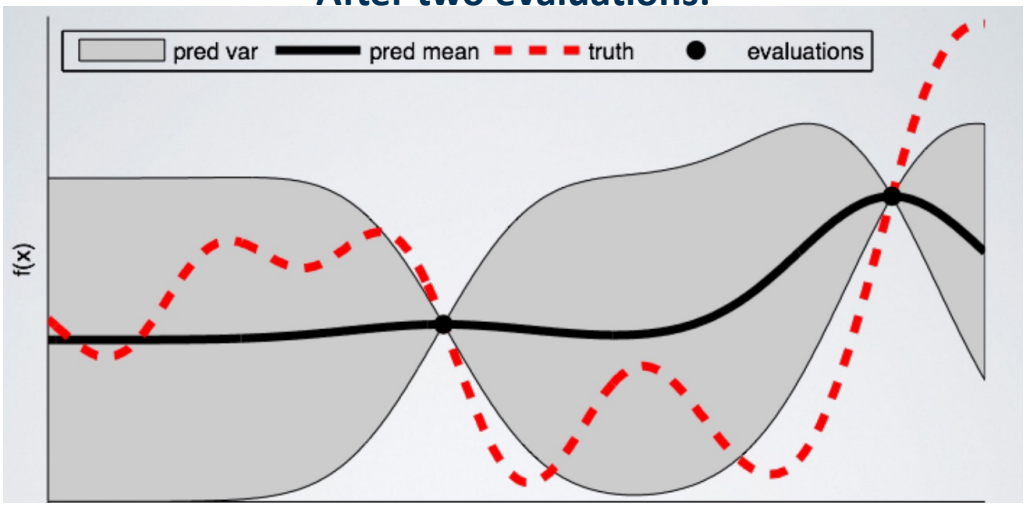
Hyperparameter Tuning
Bayesian Optimization

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

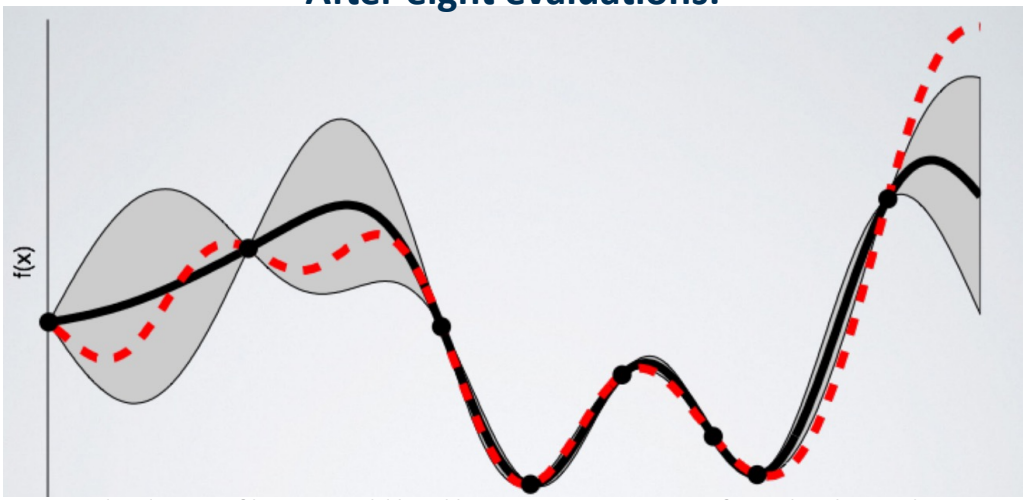
Bayesian Hyperparameter Optimization

1. Build a surrogate probability model of the objective function
2. Find the hyperparameters that perform best on the surrogate
3. Apply these hyperparameters to the true objective function
4. Update the surrogate model incorporating the new results
5. Repeat steps 2-4 until max. iterations or time is reached

After two evaluations:



After eight evaluations:



https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f

Hyperparameter Tuning
Task

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

You are training a model with 8 binary hyperparameters.

1) To find the optimal solution, how many runs over the complete dataset would you need with Grid Search?
   How many if we assume the hyperparameters to be independent?
2) You have a budget of 8 runs over the complete dataset.
   Describe how this would be realized with:
   - Random Search
   - Successive Halving (with the dataset as resource)
   - Bayesian Search

Hyperparameter Tuning
Solution

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

1) **Grid search**: We would need to evaluate $2^8 = 256$ configs.
   With independence assumption: 2 * 8 = 16 configs.
2) **Random Search**: Randomly sample 8 configs and take the best one of them.

   Successive Halving: First run with all 256 configs using $\frac{1}{256}$ of the dataset to train. Second run with 128 best configs and $\frac{1}{128}$ of the training data, etc..

   Bayesian Search: Refine the surrogate model with 8 runs and take the best config based on the surrogate.

Genetic Algorithms
Task

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

You are training a model with 8 binary hyperparameters. You are currently optimizing the hyperparameters using SGA with a crossover probability of *0.5* and a mutation probability of *0*. The optimal hyperparameter configuration is *[1,0,0,0,1,1,0,1]*.

Given the following pairs of parents, how likely is it that we generate the optimal configuration as a child?

1) [1,1,0,0,1,0,0,1] and [1,0,0,0,1,1,1,1]
2) [0,0,0,0,1,1,1,1] and [1,0,0,0,1,0,0,1]
3) [1,0,1,0,1,0,1,0] and [0,0,0,0,1,1,0,1]

Genetic Algorithms
Solution

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Optimal Config: [1,0,0,0,1,1,0,1]

1) **[1,1,0,0,1,0,0,1] and [1,0,0,0,1,1,1,1]**
   Only a split after pos. 6 would lead to the optimal solution.
   P(Split=Yes,Pos=6) = $\frac{1}{2} * \frac{1}{7} = \frac{1}{14}$

2) **[0,0,0,0,1,1,1,1] and [1,0,0,0,1,0,0,1]**
   There is no split that could lead to the optimal solution.

3) **[1,0,1,0,1,0,1,0] and [0,0,0,0,1,1,0,1]**
   Split after pos. 1 or 2 would lead to the optimal solution.
   P(Split=Yes,Pos=1) + P(Split=Yes,Pos=2) = $\frac{1}{14} + \frac{1}{14} = \frac{1}{7}$