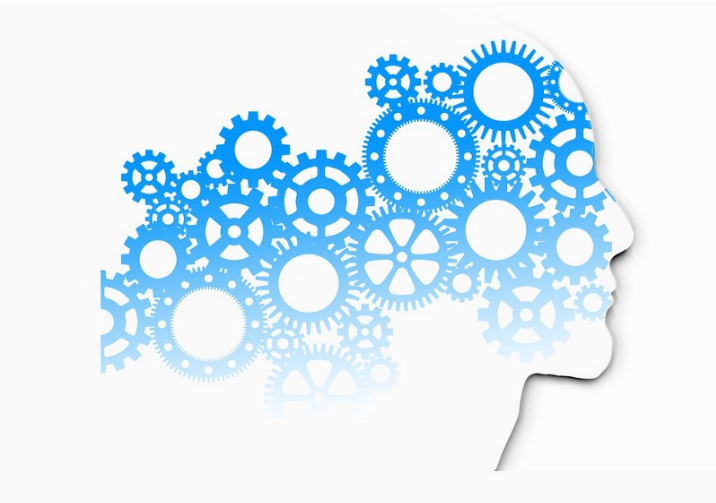

Figure 1: https://pixabay.com/en/thoughts-think-psyche-psychology-551263/

Understanding Hyperparameters and its Optimisation techniques

Prabhu Follow
Jul 3, 2018 · 5 min read

What are Hyperparameters?
In statistics, hyperparameter is a parameter from a prior distribution; it captures the prior belief before data is observed.

In any machine learning algorithm, these parameters need to be initialized before training a model.

Model parameters vs Hyperparameters

Model parameters are the properties of training data that will learn on its own during training by the classifier or other ML model. For example,

- Weights and Biases
- Split points in Decision Tree

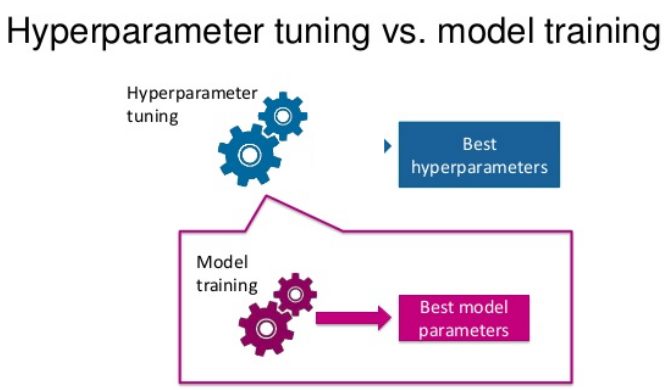

Figure 2: Hyperparameters vs model parameters → Source

Model Hyperparameters are the properties that govern the entire training process. The below are the variables usually configure before training a model.

- Learning Rate
- Number of Epochs
- Hidden Layers
- Hidden Units
- Activations Functions

Why are Hyperparameters essential?

> *Hyperparameters are important because they directly control the behaviour of the training algorithm and have a significant impact on the performance of the model is being trained.*

> *"A good choice of hyperparameters can really make an algorithm shine".*

Choosing appropriate hyperparameters plays a crucial role in the success of our neural network architecture. Since it makes a huge impact on the learned model. For example, if the learning rate is too low, the model will miss the important patterns in the data. If it is high, it may have collisions.

Choosing good hyperparameters gives two benefits:

- Efficiently search the space of possible hyperparameters
- Easy to manage a large set of experiments for hyperparameter tuning.

Hyperparameters Optimisation Techniques

> *The process of finding most optimal hyperparameters in machine learning is called hyperparameter optimisation.*

Common algorithms include:

- Grid Search
- Random Search
- Bayesian Optimisation

Grid Search

Grid search is a very traditional technique for implementing hyperparameters. It brute force all combinations. Grid search requires to create two set of hyperparameters.

1. Learning Rate
2. Number of Layers

Grid search trains the algorithm for all combinations by using the two set of hyperparameters (learning rate and number of layers) and measures the performance using "Cross Validation" technique. This validation technique gives assurance that our trained model got most of the patterns from the dataset. One of the best methods to do validation by using "*K-Fold Cross Validation*" which helps to provide ample data for training the model and ample data for validations.

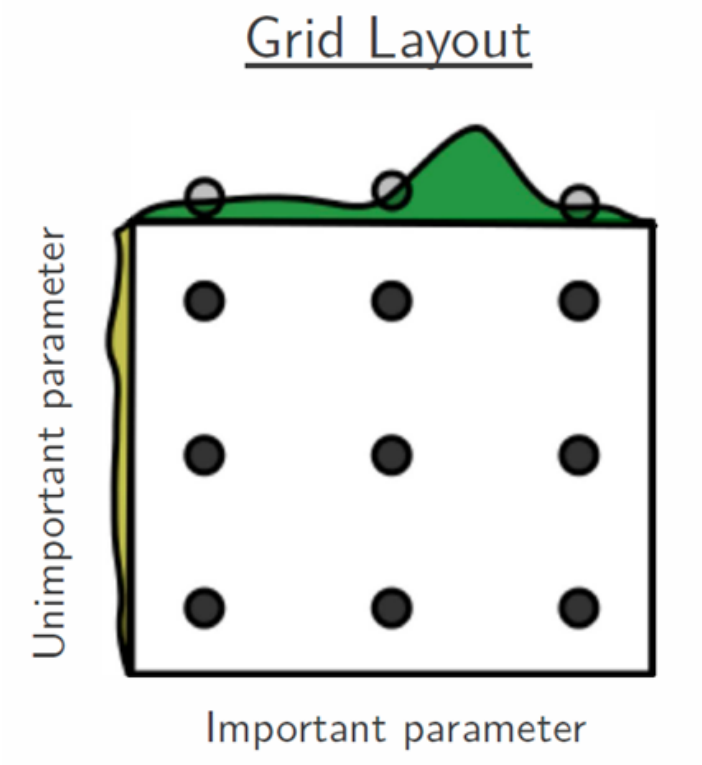

Figure 3: Grid Search → Source

The Grid search method is a simpler algorithm to use but it suffers if data have high dimensional space called the *curse of dimensionality*.

Random Search

Randomly samples the search space and evaluates sets from a specified probability distribution. For example, Instead of trying to check all 100,000 samples, we can check 1000 random parameters.

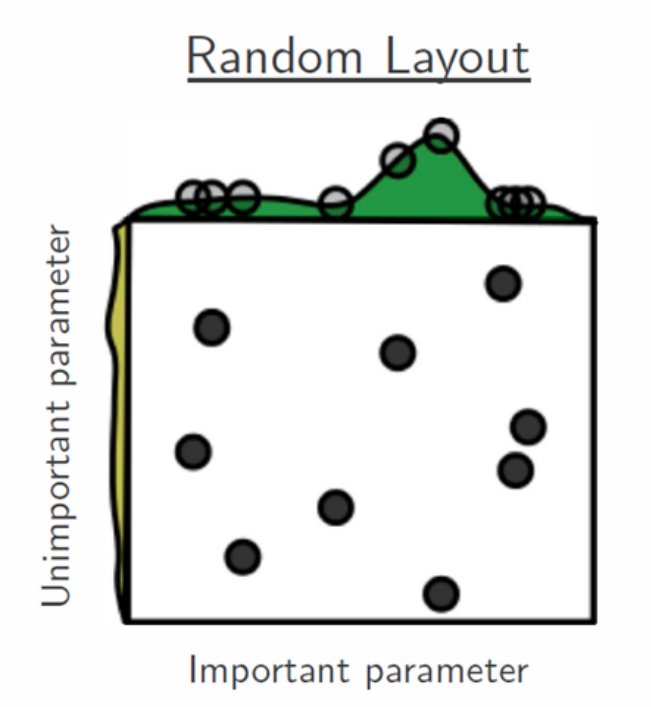

Figure 4: Random Search → Source

The drawback of using the random search algorithm, however, it doesn't use information from prior experiments to select the next set and also it is very difficult to predict the next of experiments.

Bayesian Optimisation

Hyperparameter setting maximizes the performance of the model on a validation set. Machine learning algorithms frequently require to fine-tuning of model hyperparameters. Unfortunately, that tuning is often called as '*black function*' because it cannot be written into a formula since the derivates of the function are unknown.

Much more appealing way to optimize and fine-tune hyperparameters are ***enabling automated model tuning approach*** **by using Bayesian optimization algorithm**. The model used for approximating the objective function is called surrogate model. *A popular surrogate model for Bayesian optimization is Gaussian process (GP)*. Bayesian optimization typically works by assuming the unknown function was sampled from a Gaussian Process (GP) and maintains a posterior distribution for this function as observations are made.

There are two major choices must be made when performing Bayesian optimization.

1. Select prior over functions that will express assumptions about the function being optimized. For this, we choose ***Gaussian Process*** prior
2. Next, we must choose an ***acquisition function*** which is used to construct a utility function from the model posterior, allowing us to determine the next point to evaluate.

Gaussian Process

A Gaussian process defines the prior distribution over functions which can be converted into a posterior over functions once we have seen some data. The Gaussian process uses Covariance matrix to ensure that values that are close together. The covariance matrix along with a mean μ function to output the expected value $f(x)$ defines a Gaussian process.

> *1. Gaussian process will be used as a* **prior** *for Bayesian inference*
> *2. To computing the* **posterior** *is that it can be used to make predictions for unseen test cases.*

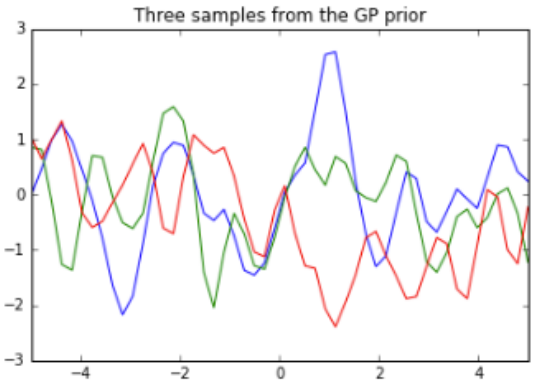

Figure 5: Gaussian Process **prior** distribution → Source

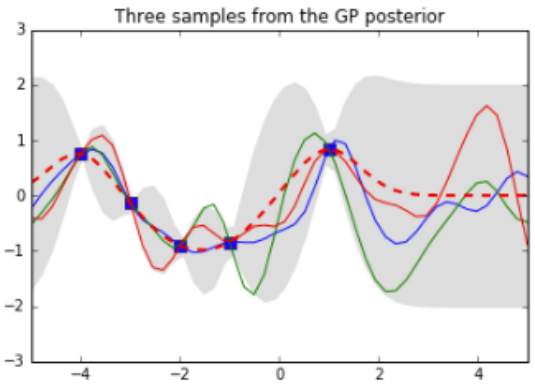

Figure 6: Gaussian Process **posterior** distribution by applying covariance matrix → Source

Acquisition Function

Introducing sampling data into the search space is done by acquisition functions. It helps to maximize the acquisition function to determine the next sampling point. Popular acquisition functions are

- Maximum Probability of Improvement (MPI)
- Expected Improvement (EI)
- Upper Confidence Bound (UCB)

The **Expected Improvement (EI)** function seems to be a popular one. It is defined as

$$EI(x) = \mathbb{E}[\max\{0, f(x) - f(\hat{x})\}]$$

where $f(\hat{x})$ is the current optimal set of hyperparameters. Maximising the hyperparameters will improve upon f.

1. EI is high when the posterior expected value of the loss $\mu(x)$ is higher than the current best value $f(\hat{x})$
2. EI is high when the uncertainty $\sigma(\mathbf{x})\sigma(x)$ around the point **x**x is high.

Summary:

- Hyperparameter tuning is an art as we often call as “black function”. Choosing appropriate hyperparameters will make the algorithm shine and produce maximum accuracy
- Hyperparameter optimization techniques mostly use any one of optimization algorithms

1. Grid Search
2. Random Search
3. Bayesian Optimization

- Bayesian Optimization uses Gaussian Process (GP) function to get posterior functions to make predictions based on prior function
- Acquisition function helps to maximize and determine the next sampling point.