› Course Shortcuts

› Student Lounge

› Q&A

› Create Decision Trees

⌄ Choose the Right Model

Module Introduction: Choose the Right Model

Tune Hyperparameters of a CART Tree

Hyperparameters

Underfitting and Overfitting

Find the Best Hyperparameter Setting

Pruning

Set Multiple Hyperparameters

Regulate the Complexity of a Classifier

Find the Best Depth for Your Regression Tree

Hyperparameters

Module Wrap-up: Choose the Right Model

› Data Science in the Wild

› Course Resources

Hyperparameters

What is a hyperparameter in a machine learning model?

A model hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data. Hyperparameters are often:

- used to adapt a model to a particular setting.
- specified by the practitioner.
- set using heuristics.
- tuned for a given predictive modeling problem.

☆ **Key Points**

If you have to specify a model parameter manually, then it is probably a model hyperparameter.

Hyperparameters are the settings of an algorithm that can be adjusted to optimize performance. Same idea as turning the knobs of a radio to get a clear signal.

You cannot know the best value of a model hyperparameter for a given problem and data set. You may use common approaches, copy values used on other models, or search for the best value by trial and error. When a machine learning algorithm is tuned for a specific problem, essentially you are tuning the hyperparameters of the model to discover the model that result in the most accurate predictions.

Model hyperparameters are sometimes also referred to as model parameters, which can make things confusing. A good rule of thumb to overcome this confusion is as follows: “If you have to specify a model parameter manually, then it is probably a model hyperparameter. ”

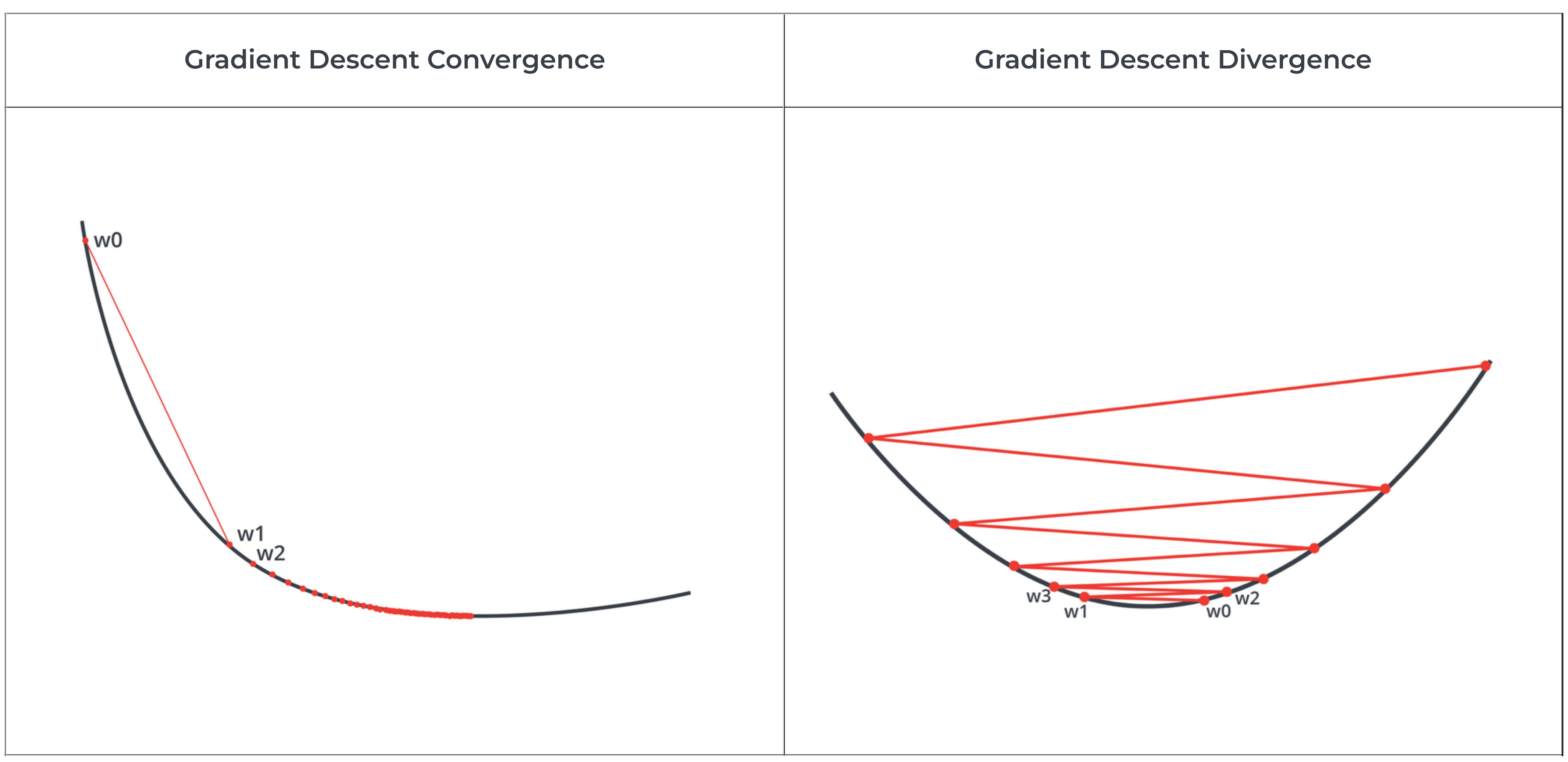
Examples of Model Hyperparameters

Model hyperparameters are the properties that govern the entire training process. Below are some hyperparameters we have encountered throughout the class:

- **Number of nearest neighbors (k)** in k-NN.
- **Depth of tree** in decision trees.
- **Learning rate (alpha)** in gradient descent.

Let's take a closer look using the example of Gradient Descent:

Setting the **learning rate** in gradient descent can sometimes be more of an art than a science. Only if it is sufficiently small will gradient descent *converge* (see the first figure below). If it is too large the algorithm can easily *diverge* out of control (see the second figure below). A safe but often slow choice is to set $a = \frac{t_0}{t}$, where $t_0 > 0$ is the starting learning rate and t is the number of steps. This guarantees that the learning rate "decays with time", thus eventually becoming small enough to converge.



Why Hyperparameters Matter

The best way to think about hyperparameters is like the settings of an algorithm that can be adjusted to optimize performance, just as you might turn the knobs of a radio to get a clear signal. When creating a machine learning model, you'll be presented with design choices as to how to define your model architecture. Often, you won't immediately know what the optimal model architecture should be for a given model, and thus you'd like to be able to explore a range of possibilities. In true machine learning fashion, you'll ideally ask the machine to perform this exploration and select the optimal model parameters automatically.

◂ Previous

Next ▸