

Introduction

2 minutes

The way we train models is by no means a perfectly automated process. Training's blind reliance on data can lead it to learn things that aren't helpful in the end, or to not learn things effectively that are actually useful. The following learning material walks through some simple reasons why under-fitting and over-fitting take place, and what can be done about it.

Scenario: Training avalanche rescue dogs

Throughout this module, we'll be using the following example scenario to explain underfitting and overfitting. This scenario is designed to provide an example for how you might meet these concepts while programming yourself. Keep in mind that these principles generally apply to almost all types of models, not just those we work with here.

It's time for your charity to train a new generation of dogs in how to find hikers swept up by avalanches. There's debate in the office as to which dogs are best – is a large dog better than a smaller dog? Should the dogs be trained when they're young or when they're more mature? Thankfully, you have statistics on rescues performed over the last few years that you can look to. Training dogs is expensive though – you need to be sure that your dog-picking criteria are sound.

Prerequisites

- Familiarity with machine learning models

Learning objectives

In this module, you will:

- Define feature normalization
- Create and work with test datasets
- Articulate how testing models can both improve and harm training

Next unit: Normalization and standardization

Continue >
