1

00:00:00,520 --> 00:00:02,170

[Auto-generated transcript. Edits may have been applied for clarity.]

Let's talk about model complexity.

2

00:00:02,530 --> 00:00:08,980

Model complexity refers to how sophisticated a machine learning model is in its ability to capture patterns in data.

3

00:00:09,550 --> 00:00:14,710

A more complex model has more parameters and can learn more intricate relationships,

4

00:00:15,070 --> 00:00:18,850

like a neural network with many layers and billions of parameters.

5

00:00:19,360 --> 00:00:23,140

A simpler model such as linear regression has fewer parameters.

6

00:00:23,650 --> 00:00:29,170

Let's talk about the trade off in complexity. Finding the right level of complexity is crucial.

7

00:00:29,560 --> 00:00:33,670

A model that's too simple might fail to capture important patterns in the data.

8

00:00:34,120 --> 00:00:40,840

A model that's too complex might fit the training data perfectly, including the noise, and fail to generalize.

9

00:00:41,680 --> 00:00:45,280

For example, a linear regression model might miss a curve in the data.

10

00:00:45,550 --> 00:00:49,390

A polynomial regression of appropriate degree can fit the trend better.

11

00:00:49,690 --> 00:00:56,830

However, a high order polynomial can perfectly match all training points, which often means it's overfitting.

12

00:00:58,130 --> 00:01:04,190

Although such a complex model fits the training data exactly, it performs poorly on new,

13

00:01:04,190 --> 00:01:08,690

unseen data because it has learned the noise rather than the actual pattern.

14

00:01:09,410 --> 00:01:14,390

What about noise in the data? Suppose you measure a sinusoidal pattern with sensors.

15

00:01:14,660 --> 00:01:23,330

The true underlying data follows a sine wave, but due to measurement error and sensor noise, the observed data has small deviations.

16

00:01:23,630 --> 00:01:29,240

A well fit model should capture the underlying sine trend, not the scattered deviations.

17

00:01:29,690 --> 00:01:36,950

A very complex model might trace the scattered points precisely, resulting in a curve that is no longer sinusoidal.

18

00:01:37,400 --> 00:01:45,500

This is overfitting. When a new input comes in, the overfit model will fail to generalize and predict an incorrect value.

19

00:01:46,130 --> 00:01:49,730

Let's explore underfitting and overfitting in more detail.

20

00:01:50,120 --> 00:01:53,780

Underfitting happens when the model is too simple to capture the trend.

21

00:01:54,290 --> 00:01:58,010

Overfitting happens when the model is too complex and fits the noise.

22

00:01:58,280 --> 00:02:04,790

In both cases, the model performs poorly on test data, even if it performs well on training data.

23

00:02:05,330 --> 00:02:07,100

Let's move on to generalization.

24

00:02:07,730 --> 00:02:17,000

What we want is a model that is complex enough to capture the true structure of the data, but not so complex that it memorizes noise.

25

00:02:17,450 --> 00:02:21,980

A well generalized model performs well on unseen data.

26

00:02:22,400 --> 00:02:26,270

Overfitting is like a student who memorizes instead of understanding.

27

00:02:26,480 --> 00:02:31,760

They perform well on homework training data, but poorly on the exam test data.

28

00:02:32,180 --> 00:02:37,430

The moment the test loss begins to increase is when generalization power is lost.

29

00:02:37,790 --> 00:02:43,430

Stopping training at this point is ideal, which is why techniques like early stopping are used.

30

00:02:43,970 --> 00:02:50,570

What about visualizing loss and overfitting? When training a model, you want to minimize the loss function.

31

00:02:50,570 --> 00:02:57,260

For example the mean squared error in regression. During training, the loss on the training data decreases.

32

00:02:57,470 --> 00:03:03,380

But if the model is too complex, the loss becomes extremely small because it's fitting the noise.

33

00:03:04,250 --> 00:03:09,050

Now consider the test data. A portion of the data set held out to evaluate the model.

34

00:03:09,440 --> 00:03:13,370

After training, you want the loss on the test data to be small as well.

35

00:03:13,790 --> 00:03:18,860

But if the model overfits, the loss on the test data starts to increase.

36

00:03:19,220 --> 00:03:24,380

This means the model is learning too much from the training data and losing its ability to generalize.

37

00:03:24,980 --> 00:03:32,990

It's also important to prevent overfitting. One mechanism to avoid overfitting is to split the data set into three parts.

38

00:03:33,560 --> 00:03:41,120

One training set used to train the model two validation set, used to monitor performance and guide training,

39

00:03:41,360 --> 00:03:46,640

and three test set completely unseen data used to evaluate the final model.

40

00:03:47,390 --> 00:03:51,530

During training, you assess how the loss is minimizing on the validation set.

41

00:03:51,830 --> 00:03:57,380

Although the validation set is not used to train the model directly, it helps guide the training process.

42

00:03:57,680 --> 00:04:01,370

It influences when to stop training and which parameters to finalize.

43

00:04:02,090 --> 00:04:06,500

Remember, the test set must remain completely unseen until the very end.

44

00:04:06,860 --> 00:04:12,709

It's used to evaluate how well the model generalizes the validation set, while not used for training,

45

00:04:12,710 --> 00:04:18,050

helps determine the best version of the model by guiding parameter selection and stopping criteria.