Video 25

Monday, November 9, 2020 9:57 PM

Training loss :

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(y(x^{(i)}), t^{(i)}\right)$$

θ : parameters of the model

$y(x^{(i)})$: model prediction

$x^{(i)}$: input

$t^{(i)}$: target

Generalizes to new data ?

Introduce a split between :

- training data → used to find the parameters of our model
- validation data → used to find the hyperparameters
- test data → used to measure the generalization performance

generalization : ability of the model to classify / predict on inputs it hasn't seen during training.

For each of these 3 sets, we can compute a loss:

- training loss
- validation loss
- test loss

Why do we need a validation set ?

→ we can't tune hyperparameters on training data

↳ we want hyperparameters that generalize.

example : picked a large number of neurons when designing a neural network could memorize training data

→ we can't tune hyperparameters on test data

↳ a form of "cheating"

↳ we should only measure test performance once

↳ this would make it difficult to realistically evaluate/measure generalization.

Strategies for hyperparameter tuning

- Grid search : → for each parameter, choose candidate values
  → train one model for each possible combination of hyperparameter values.
  → choose the combination that achieves minimal validation loss / error

- Random search : → for each parameter, pick random values
  → train a lot of models with different combinations of random hyperparameter values.
  → choose the combination that achieves minimal validation loss.

Advantage of random search (over grid search) is that we have to try less combinations when the number of hyperparameters is large.

example : difficult to perform a grid search over 10 hyperparameters

but a random search could still be viable.

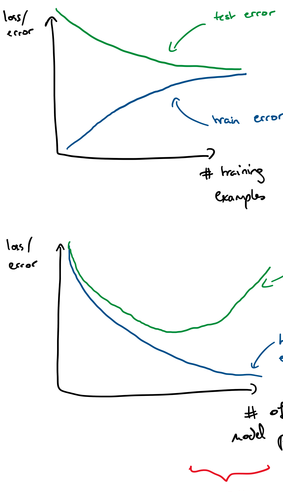
Reasoning about generalization

Avoid overfitting : avoid the model from recognizing data based on accidental regularities in the training set

Recall the example of a deep neural network which we covered in the video on the universal approximation theorem.

↳ Pb : model has too much capacity (i.e., has ability to memorize training data)

Visualize generalization in practice :



(similar figure for loss wrt # epochs)