Statistics and Artificial Intelligence

Lecture 15: Fundamentals of Machine Learning III

Yixin Wang

Roadmap for Today

- We have discussed the core idea in “Fundamentals of ML”: overfitting
  - One common solution: validation
    - A more efficient variant / When data is limited: K-fold cross validation
    - We have seen Boston Housing Example
- Today: Tuning parameters; improving model fit; dropout

Logistics

Teaching Evaluation

- Midterm Teaching Evaluation
  - Everyone gets 0.5 points if the response rate exceeds 93%.
  - Deadline Friday (tomorrow)
  - Current response rate has gone up from 6% to 62%. Almost there!
- Last 30 mins of today's lecture
  - Dr. Bethany Morrison from CRLT will conduct an anonymous midterm student feedback session.
  - The goal is to help us improve your course experience.

Fundamentals of ML

- Fundamentals II: https://colab.research.google.com/drive/18LbrzZtrsjv_XZ99K7C372e8EmFx6VNB#scrollTo=hRhdqVnFI7vD

The materials in this course are adapted from materials created by Alexander Amini, Alfredo Canziani, Justin Johnson, Andrew Ng, Bhiksha Raj, Grant Sanderson and the 3blue1brown channel, Rita Singh, Ava Soleimany, and Ambuj Tewari.

Validation set - proxy for unseen data in the real world

K-fold cross validation - split data into fold

- use each fold as a validation set + training set
- Helpful for combatting overfitting when we have limited data

want model to have enough representational complexity but not overfit

Overfitting concerns the hyperparameters we pick

ie learning rate, neurons in each layer, etc

Before we worry about overfitting we need to ensure the model is properly trained

↳ see training loss go down

Does it beat a common sense baseline?

↳ logistic/linear regression, old ML techniques like random forest

want model to have enough capacity to be able to overfit, but avoid overfitting

↓ learning rate helps w/ optimization, but model takes longer to converge

- Typically start high + keep decreasing it until the loss goes down

↑ learning rate - can see loss increase

- want it to generally go down
- Failure indicates issues we aren't converging to a smaller loss

"large learning rate" is relative to any particular dataset

pick arbitrary α

→ If we observe training loss going up + down we know α is too large

can ↑ batch size if training gets stuck

Meaningful Generalization
- is our architecture better than a simple baseline

Failure
- Input data doesn't have sufficient info to predict targets
  → model trains fine, but validation accuracy is low
- Model your using isn't suited to the problem at hand

→ model needs to make the right assumptions about the sweater

Fit simple baseline whenever you create a neural net to measure impact of ↑ model capacity

Not able to overfit → model isn't flexible enough

- want to see validation loss go down + then up

Regularization

- impede models ability to fit so it doesn't overfit

- always paired w/ validation

can reduce network size once we have enough power to overfit we figure out right # of neurons + layers by trial + error

- → start w/ few layers + parameters
- → Add more until we see diminishing gains in validation loss

very ↑ capacity model will train very quickly but is very susceptible to overfitting