Machine learning strategy

Zoltan Miklos

University of Rennes 1

2022

Why do we need a machine learning strategy ?

(*Credits : The material is based on notes from Andrew Ng.*)
Your project involves machine learning component. You realize that the results are disappointing. What to do ?

- collect more training data ?
- use a larger / deeper network ? (or smaller ?)
- change the algorithm ?
- change the parameters of the algorithm ? (longer training, other learning rate ? etc.)
- add a regularization technique ?

Which solution to chose ?

Dev and test set

Your project involves machine learning component. You realize that the results are disappointing. What to do ?

- Split the available data to training/test : a random split 70% / 30% is a good start, but it is important to ensure that the test set contains all the challenging cases you expect to have for the unseen data
- Training/Dev/Test split : Training to run your algorithm, Dev : to set the parameters, Test to evaluate
- Test : it should the data where your algorithm should perform well

Dev and test sets should have the same distribution

Verify ! Run a small program that compares the distributions.
Otherwise a number of things can go wrong

- If dev is different from test, the efforts spent on improving on dev are wasted
- You can overfit the dev set

How large should be the dev/test sets ?

- If your classifier A results 90% accuracy on the dev set and the classifier B has 90.1% then a dev set of size of size 100 is small
  - a difference in accuracy (0.1) in this case is likely to come from numerical errors, it is not a real difference
  - 10 000 would be better
- How much data is classified differently by A and B ?
  - You could use statistical methods (significance tests) to understand whether your improvement is real or it is only a noise
  - Hypothesis : "A is better than B", and estimate the statistical significance of this hypothesis
- Test set : it is not the size that matters (reasonably large), but it should be diverse enough to correspond to the distribution of unseen data

Entropy of training set

- it is not (only) the size that matters
- The test and train set should be diverse enough (high entropy)
- the distribution of test set it should also correspond to the distribution of unseen data
- A useful trick : compute the entropy of training and test sets (compare the distributions)

Iterative projects

Machine learning projects should be iterative. It is extremely hard to tell a priori what is the "right approach" (a sophisticated algorithm, large collection of training examples, etc.)

- Start with an idea
- Implement
- Experiment

Having a fixed dev set and a clear metric will enable to measure the advancement.

Analyse errors in the dev (test) set

- Analyse individually wrongly classified examples from the dev set to see what is the reason for the problem
- If you propose a solution (to improve the learning accuracy), estimate the effect of your approach (for example, if it would eliminate 1 other case ? or 10 ? or 10000 ? 50% ? or 99 % of all problems)
- If there are too many errors, select a small fraction (Eyeball dev set) randomly in order to analyse

Bias and variance

Bias and variance are the two major sources of error in machine learning projects. Informally,

- Bias : the error rate on the training set
- Variance : how much worse the algorithm works on the dev set than the training set ?

Bias and variance

Bias and variance are the two major sources of error in machine learning projects.

- High bias (under-fitting) : we miss relevant relations between features of data and the output (prediction).
- High variance (over-fitting) : the algorithm also learns the (random) noise in the data, rather then only the intended output

Optimal error rate (Bayes rate)

- Sometimes even humans cannot achieve 0 error rate. For example, voice recognition with noisy audio files.
- Analyse the optimal error rate and understand the avoidable and the unavoidable bias.

How to address high bias or high variance ?

- high (avoidable) bias : increase the size of your model (more layers in a neural network, bigger decision trees, etc.) (however this comes with higher computational costs, longer training times, etc.)
- high variance : add more data to your training set (if it is available, or reasonable to construct - with human efforts)

Variance vs. bias trade-off

- increasing the model size can effectively reduce the bias, but might increase the risk of over-fitting (increase variance)
- Regularization : increases bias, but reduces variance

Early stopping

- Stop gradient descent before it reaches convergence
- Reduces variance, but increases bias
- Similar to regularization

Reducing variance

- More training data
- Add regularization
- Early stopping
- Decrease the number and type of input features
- Decrease model size