Project Overview

Grace Zhang

Overview

1. Logistics
2. Choosing a Project Topic
3. ML Advice

Logistics: Project Proposal (20/100 points)

- Teams of 3-4
- Written proposal due tonight Friday 3/8 11:59PM on Gradescope (does not need to be very detailed)
- Fill out this form with team members, project title, and a short description
- Each team will be matched with a TA mentor
- Initial check-in with TA mentor in the week of 3/18-3/22: Please prepare a 5-slide presentation version of your project proposal

Logistics: Midterm Report & Check-in

- Midterm Report and TA Check-in around week of 4/15: more details TBA
- What to accomplish by midterm report:
  - Gather or source dataset
  - Implement core methods
  - Reproduce results from prior work
- Good opportunity to check in with TA if you need to pivot project direction
- Final Report Due around 5/6, no final presentation

Choosing a Project Topic

- A good project idea should incorporate ideas discussed in class but also some exploration outside of course content.
- Choose a domain you're excited about!
- Make sure the data and task you choose has enough complexity for you to try some different things (e.g. for a Kaggle competition, prefer real data to synthetic data).
- Design a project that can realistically be finished in one semester, your TA check-in will help make sure the scope is right.
- Feel free to use the suggested ideas or use them as a template. Think about substituting a different dataset or application, using different methods, or designing your own analysis experiments.

ML Tips

1. Acquire & Pre-process Data
2. Create train/val/test splits
3. Define model inputs and outputs
4. Build model (simplest that works!)
5. Measurement
6. Iterate

ML Tips

1. Acquire & Pre-process Data
2. Create train/dev/test splits
3. Define model inputs and outputs
4. **Build model (simplest that works!)**
5. Measurement
6. Iterate

Choosing a Model

- What are the input features and output labels? Are you doing regression or classification?
- Generally start with the simplest model (unless your project topic is focusing on a particular model)
    - Easy to debug, iterate quickly, good baseline for future work
- From there on try: different feature spaces, regularization, more complex models etc.

ML Tips

1. Acquire & Pre-process Data
2. Create train/dev/test splits
3. Define model inputs and outputs
4. Build model (simplest that works!)
5. **Measurement**
6. Iterate

Measurement: Designing Experiments

- What question am I trying to answer?
- What experiments do I need to run?
  - Could be comparing methods or hyperparameters on your selected dataset
  - Could be adding regularization, varying dataset size or quality
- What figures should I report? Think about what kind of graph or table you'd like to end up in your final report.

ML Tips

1. Acquire & Pre-process Data
2. Create train/dev/test splits
3. Define model inputs and outputs
4. Build model (simplest that works!)
5. Measurement
6. **Iterate**

Iterating: When things don’t go as expected

- Good idea to have in mind what your expected result is before you run an experiment instead of trying to figure out what a result means after you see it
- Negative results are a normal part of the process, more important to understand why the results aren’t as expected. Was there an error in your assumptions? Are there any diagnostic experiments that you could do?
- Resources: TA office hours, Googling can help with common ML mistakes, best practices, what hyperparameters to prioritize tuning, etc.

Other Tips

- Use python, plenty of online resources and useful libraries (pandas, numpy, pytorch, scikit)
- You should implement your main method (we will ask you to submit code with final report), but you can use open source code for benchmarks, data processing, plotting, etc.
- If your project requires additional compute resources, we would recommend Kaggle, Google Colab, or Paperspace
- If your code is running slowly:
  - Vectorize code when possible (use arrays with libraries like numpy that have built in vectorized functions instead of for loops)
  - Try using GPU if using neural networks
- Log important metrics (training/test loss, training/test accuracy) during training, useful for debugging or plotting learning curves