CS231N Project Design Tips

Slides by Andrew Kondrich and Chris Waites

Adapted from Pedro Pablo Garzon

First pick a project

Main considerations

1. Data
    a. Depending on the task, it can be **very** hard to get your own data
    b. If you are collecting your own data, make sure you have a solid interface for collection and data processing: https://gym.openai.com/
2. Code base and framework
    a. Tensorflow, PyTorch, Keras, etc
3. Architecture
4. ML Objective

Start with focusing most of your effort right now to data

Do a little bit of Googling each day

Get some inspiration

1. Look up highly publicized material: OpenAI, Google Brain, Facebook FAIR, etc
2. CS230 section notes: https://cs230.stanford.edu/section/1/
3. Try to find cool web demos like this: https://worldmodels.github.io/
4. Look at list of accepted papers to recent conferences.
    a. https://openreview.net/group?id=ICLR.cc/2020/Conference
    b. http://openaccess.thecvf.com/ECCV2018.py
    c. https://sites.google.com/robot-learning.org/corl2019
    d. To see which conferences there are in your interest field: aideadlin.es
5. Plenty of awesome Medium posts detailing how-tos
6. Look at previous years projects!
    a. Neural Network University: CS231N, CS230, CS234, CS224N
    b. See what works and what doesn't!

TAs also like

1. Papers with code: https://paperswithcode.com/sota
2. IEEE 2019 summary report on GANs: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8667290&tag=1
3. Review of CV for 3D: https://www.researchgate.net/profile/Niall_O_Mahony/publication/327980521_Computer_Vision_for_3D_Perception_A_review/links/5bb1abd8a6fdccd3cb80a379/Computer-Vision-for-3D-Perception-A-review.pdf

Project flavors (not exhaustive)

1. Experiment with improving an architecture on a predefined task
2. The case study: Apply an architecture to a dataset in the real world
3. The challenge: compete in a predefined competition (Kaggle)
4. The researcher: join a Stanford/company research project
5. Stress test or comparison study of already known architectures
6. Design your own unit (complex layer, objective function, optimizer, etc)
7. Mix and match domains! (e.x use a CV GAN in RL game)
8. Don’t do video (unless you got $$$ and tons of time)

Design think it

1. Have each member of your team flesh out 10-20 quick ideas down on paper before meeting. Don’t be afraid to get creative
2. Filter out list by doing quick Google searches on data
   a. Anything below GB scale of data...good luck. Vision = big datasets
   b. If you have an idea, Google it first! **Don’t want to “just” reproduce the same result.** There’s probably a Github with your project already
3. Pay attention to how long and much data the models you see are trained on
4. Find pattern in data+architecture combos
5. Ask are there little tweaks or other experiments that haven’t been done yet?
6. Can you extend the idea in one paper with another?
7. Which idea gives you more things to experiment with?
8. How can you get pretty images / figures?

Paper reading process

1. Don’t read all of it
2. Look at the figures and captions before anything
3. First pass reading order
    a. Abstract
    b. Methods
    c. Results
    d. Conclusion
4. Plenty of blogs, Github repos, websites that summarize or explain papers even better!
5. Example: Yolo Paper https://arxiv.org/pdf/1506.02640.pdf

Try to avoid this scenario

1. Nothing special in data pipeline. Uses prepackaged source
2. Team starts late. Just instance and draft of code up by milestone
3. Explore 3 architectures with code that already exists
    a. One RES-net, then a VGG, and then some slightly different thing
4. Only ran models until they got ~65% accuracy
5. Didn’t hyperparameter search much
6. A few standard graphs: loss curves, accuracy chart, simple architecture graphic
7. Conclusion doesn’t have much to say about the task besides that it didn’t work

Aim for this

1. Workflow set-up configured ASAP
2. Have running code and have baseline model running and fully-trained
3. Creative hypothesis is being tested
4. Mixing knowledge from different aspects in DL
5. Have a meaningful graphic (pretty or info rich)
6. Conclusion and Results teach me something
7. ++interactive demo
8. ++novel / impressive engineering feat
9. ++good results

Milestone goals

1. We want to see you have code up and running
2. Data source explained correctly
   a. Give the true train/test/val split
   b. Number training examples
   c. Where you got the data
3. What Github repo, or other code you're basing off of
4. Ran baseline model have results
   a. Points off for no model running, no results
5. Data pipeline should be in place
6. Brief discussion of initial, preliminary results
7. Reasonable literature review (3+ sources)
8. 1-2 page progress report. Not super formal