How to do a research project in deep learning?

I will focus on algorithms and theory oriented projects since these are my domain expertise.

Step 1 Pick a problem: There are many ways to pick a problem.

(i) Read recent papers (NeurIPS / ICML / ICLR / COLT etc)

(ii) Attend research seminars (many online seminars are publicly available)

Example. (a) supervised learning

(b) Get data: images / texts etc

(c) Design model

(d) Train model

(e) Test model

(f) Analyze the model and its performance

iterate.

Suppose we are interested in studying data aug for solving text classification tasks. Then we want to try out different text transformation methods on various prediction tasks.

What properties make for a good candidate PhD project?

(*) Interests

(*) Data availability

(*) Domain knowledge: one thing I would encourage you is to work on sth where you have domain knowledge. For example, if you are a mechanical engineer, and have unique knowledge in some aspects of mechanical engineering where you

want to apply machine learning, that will let you do very interesting project that is actually difficult for others to do.

(4) Utility

(4) Feasibility; Having an accurate judgment for whether a project is feasible or not is actually very difficult. Especially in the area of algorithms / theory, it takes lots of experience and expertise to build up a sense for whether a project is feasible or not.

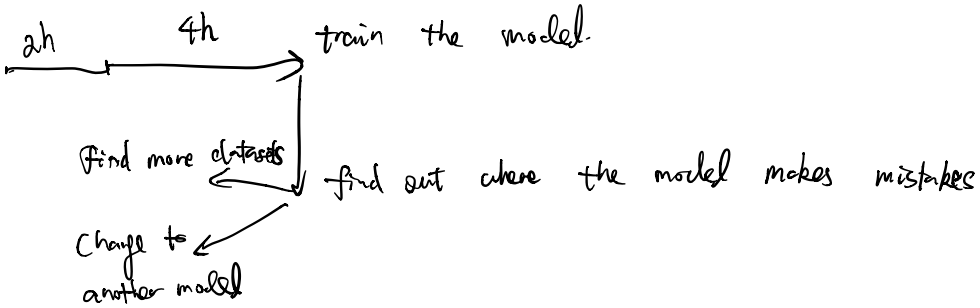
Step 2 Get data

Option 1. A sentiment analysis task that has about 5K samples,

Option 2. A Yelp review dataset that has > 1M samples.

Which one would you start with? Let's suppose that both datasets require some cleaning preprocessing before you can use it in your code? How many hours would be spend? 1, 2, 3, 5, 8, ...?

If you have not worked on this problem before, it turns out that it's very hard to say what's difficult about the problem.



Often times, you make multiple iterations of this process.

Tip : keep clear notes on experiments run

keep a spreadsheet of what parameters you have tried

Tip : strong inference. [John R. Platt, Science '64]

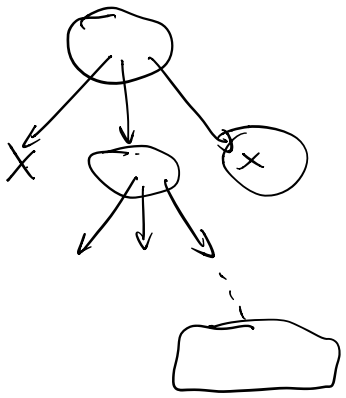
— In ML, especially in DL, often you find that there are so many papers online. Each of them say that their model has a certain advantage over others.

— The idea of strong inference is just the simple and old-fashioned method of inductive inference that goes back to

* Devise alternative hypotheses
* Devise a crucial experiment (or several of them) with alternative possible outcomes, each of which will, as nearly as possible, exclude one or more of the hypotheses
* Carry out the experiment so as to get a clean result

↳ Recycle the procedure, make subhypotheses or sequential hypotheses to refine the possibilities that remain; and so on.

At the end of the day, the experience is kind of like climbing a tree.



But then once you start to write the paper, you want to reverse from the leaf to the root :)

Ex.

Deployment. One last part of the cycle that I did not mention is deployment. Let's say that you are developing a voice activation detection system. Which one would you use?

Non-ML based: See if volume $\geq \varepsilon$

Train small NN on human speech vs. silence.

If the data changes from US to UK speakers, which system will be more robust?

Non-ML

Learned small NN

This is a general problem in deployment, which is often also called domain adaptation.

- Non-ML methods are often more robust
- Learning algorithms esp. NN are better at "automatic" feature extraction