[ICLR 2020] Tips for prospective and early-stage PhD students

- how to get started in ML/DL related research -

Contributors (sorted by first name): Akari Asai, Carlos Miranda, Chiara Mugnai, Claas Voelcker, Divyansh Kaushik, Fairoza Amira Binti Hamzah, Jade Abbott, Jaydeep Borkar, Kalpesh Krishna, Karmanya Aggarwal, Makbule Gulcin Ozsoy, Marija Stanojevic, Martha White, Michael McCabe, Moritz Schneider, Rajarshi Das, Sabrina J. Mielke, Sagar Devkate, Tornike Tsereteli

Note: Document is US-biased, but many answers are internationally applicable.

How to get started in ML/DL?

Useful tips/materials for learning ML related programming and basic deep learning knowledge:

- Books: "Hands On Machine Learning with Scikit Learn Keras and TensorFlow", "Deep Learning" and "Mathematics for Machine Learning"
- Watching fast.ai courses (learn fast.ai library and pytorch).
- To learn about optimization, Stephen Boyd's course on Convex Optimization.
- Reading published code helps to see what the actual contribution of the paper was and you can map that back to the mathematical explanation in the paper. Then, the next time you see that mathematical explanation, you know exactly what it means computationally.
- For programming skills in DL, you can follow François Chollet (@fchollet) on twitter (he usually posts some of the best tutorials on TF+Keras) or deep dive in github repos (e.g. TF object detection API, HuggingFace□ transformers...), you will find lots of best practices! Just study them, understand why they are done as they are and practice.
- Reimplement papers on your own.
- Attend ML schools, such as JSALT or those mentioned here.
- More tips in this twitter thread.
- It would take time to go through those materials, but don't give up. If possible set aside two-three weeks to go through a course / book in detail without distractions. Otherwise, set aside a specific time of the day when you'll do this.
- Auditing or watching youtube videos of ML/DL courses from some universities is also great way to learn (e.g., Deep Learning for NLP, Deep learning for Computer Vision)

How to keep track of novel research?

- Once a month, check arxiv-sanity.com for the last month, "top recent", "top hype" and "recommended" tabs. Scan papers titles and abstracts and based on that download papers for further reading (~15).
- Read fully just the most interesting of those (~5) and get the most important ideas of the others.
- Listen to the talks from ~ 5 high quality workshops/conferences per year, within the area of interest to gain new ideas. You can search for those on youtube (Institute for

Advanced Studies channel, for example), slideslive.com, videolectures.net. If possible, try to attend some of them live.

- Attend highly relevant/quality talks in your area to get to know relevant people.
- Subscribe to relevant blogs' weekly/monthly updates. Pick blogs based on their quality and my interests. Many companies or big ML research groups have blogs. Examples of blogs/newsletters to follow: on ML theory, also ML theory (no new posts), probabilistic, broad and beginner friendly, on ML theory, on NLP, on deep learning, broad machine learning with industry-related updates, and BAIR research. If there is no subscription to newsletters, you can subscribe to RSS feeds or follow the authors on twitter.
- Talk to colleagues and ask them what they currently think is important.
- Following some twitter bot posting the new Arxiv papers daily might be helpful -- you don't need to read all of them, but if the titles look relevant to your current project or sounds interesting, just click the link to the Arxiv page and skim the abstract.
- Keep a detailed multi-doc journal of ideas and research you've read and use a paper organizing software. Zotero/Mendeley/roam-research/Google docs/Microsoft OneNote Folders/Notion/Trello are some of the software you can look into. Notion has a bit of a learning curve which makes it non ideal but it's the best thing I've found that syncs to multiple machines. Trello allows you to set up a board with a list of categories (e.g. research topics, resources), under which cards represent subcategories (e.g. tools, datasets, tutorials). You can also create a GitHub repo with a readme with different subsections of the research topics I'm interested in (can put papers, blog-posts, libraries and tools related to the research, other resources, etc under the topics).

How to apply to PhD?

- Check out blogs: Machine Learning PhD applications, How to Pick Your Grad School and Student Perspectives on Applying to NLP PhD programs
- Find/meet/contact a potential advisor before the application period. Read papers in areas you are interested in and look into websites of those researchers. If they don't have open positions announced, contact them by email (except if it's otherwise mentioned on their website). Conferences are a good way to meet potential advisors.
- Try contacting advisors as early as you've made up your mind about grad school / research direction.
- Good letters of recommendations from someone already in the field is the best way to be noticed during applications. Recommendations coming from academia are very important, while those coming from industry are rarely taken into account, unless the recommender has a good reputation in the research community and has Ph.D. in the field. Recommendations from senior PhD students or even postdocs are not very valuable.
- You should consider applying for a PhD if you think you have a good research profile (e.g., having worked on research projects during undergrad, did research internships / attended research intensive summer programs like JHU's JSALT). If you don't have research experience and want to gain some before applying, applying to master's might be more appropriate. It may happen that professors expect more from people who finished masters.

- In recent years PhD admission in CS/ML got more competitive in general, so people with previous research experience (and published papers) seem to have better chances.
- Other ways of getting research experience are: applying for a semester of undergraduate research assistant, fellowship or course. Working as a technician / software engineer within a research lab.
- Residency programs offered by company research (e.g., Facebook AI, Google AI, UBER AI, IBM, VinAI, Shell AI) are also great opportunities, although the application process is known to be competitive.
- **How do you think Covid-19 will affect next year's application rates and acceptances?** It's still unknown, and depends on the institution, money flow, and federal imigration regulations.

How to find and pick a PhD advisor

- **How to find a PhD advisor?** Find a topic that you are interested in researching. Look into papers in that area and who authored them (which groups are active in that field). Out of those, select groups within universities/locations in which you can imagine living for the next five years. Look into websites of those groups, their departments or twitter accounts of the professors as they might have publicly available calls for PhD students. If not, try to contact professors or students from those groups to ask if there are available positions (attach your resume).
- Try to meet or get to know potential advisors as much as possible before applying to Phd. Contact his/her students / postdocs with questions you have.
- It's important that your advisor's working style fits you well.
- **Questions to ask prospective advisor:** How much time he/she spends with students? What's his/her working style? What kind of and how much work has her/him published in the last few years? How stable would your financing be? How many other faculty are in the department who work on your area of interest? What is their advising style? What traits do they prioritise? What do they expect from their PhD student? How much non-research work would you need to do (lab maintenance, meetings,...)?
- If you are interested in doing internships, ask: Are his/her current students doing an internship and at what places?

Going to graduate studies (getting into research) after working in industry

- What will change: you'll get less money, but more time and freedom.
- Many of the current Ph.D. students are satisfied with the current payments.
- You'll probably need to move. Think if that is feasible in your situation.
- **Should you be moving from industry to academia if they are interested in a specific research topic?** Depends on the topic and your background. If you want to transfer from physics into ML research and you know how to code, there is a way to do it without going to academia, but you need to choose an appropriate employer. In many other cases that's not possible. Try applying for a role that you aspire to. If you get invited to interviews and can do pretty well in them, you are good. Otherwise, you'll see what's missing and how much more work you need (if you need to go back to school).

- **When to give up industry for academia?** If you see yourself working on a research topic for the next five+ years and you extremely value time and freedom to pursue your research aspirations, academia is probably a good choice. If you want to get into ML because it's popular and you heard it's cool, you should start by listening to a few online courses in your free time and start building some small stuff. Then, you can re-evaluate your aspirations.

Things to keep in mind in order to make the most of your PhD

- **Are you solely responsible for generating new research ideas? Or do advisors play a significant role in it?** That's going to depend a lot on your advisor, the size of your research group, and how well your research interests line up with your advisor's goals. It's almost a given that your advisor will have some project in mind that they think you'd be a good fit for when they accept you. But if that doesn't line up with your career goals, then you might find yourself needing to be significantly more independent (though your advisor should still be a good resource to help refine your ideas). Talk about this with your potential advisor during the application process - that's really the only way you'll know what their expectations are. I also think it depends on the advisors' advising style. Talking to his/her students to ask the advising style would help.
- **What is one advice you wish you had gotten before starting your PhD?** Don't forget to take care of yourself, including enough sleep, basic exercise, good quality food and asking for mental health counselling, if needed. Expect to have good and bad periods. Procrastination is part of most PhDs. Find ways that help you shorten the procrastination periods (exercise, short trips, gathering with friends, more relaxed/tight schedule...)
- **What is the role that the groups you are in (excluding the prof) have played in your learning?** If people work on a related topic, you can get help from them or actively discuss with them to brainstorm. If people are working on different topics but are open, you can still brainstorm with them and get some inspiration from different fields.
- **What is the balance between doing everything yourself vs taking help and support from your group?** It's good to be in a collaborative group, where people help each other (especially before the deadline). Too competitive environment is not really good for mental health. On the other hand, doing everything by yourself is good to learn to be more independent.
- **Is it common to change a PhD program after spending a year or two?** It is not common, but it happens for different reasons. Give your best to pick the suitable university, program, advisor and research topic, so you don't need to change.
- **What is the procedure to transfer to another program?** Many schools don't have this option, so you might need to start a PhD program from the beginning. If you want to transfer, try doing it as soon as possible. Some schools will allow credit transfer for courses you've taken. When it comes to applying to a new school, the process is the same as for the freshmen PhD. Therefore, try looking first into transfer options within the same university.
- What are tips to organize, optimize and make one's research project faster? How can I shave off time by better preparing? I learnt the hard way that I didn't need to write a lot of code since it was available online as a functionality in a well established library like scikit-learn etc or someone had a good github repo with the same. So checking online can save

some time. Also writing modular code that can be reused, as a library for yourself can be very helpful. A lot of non-CS students do not use Git as often, I think they are missing out on a lot. Documenting your experimental setup, results and thoughts about future experiments/ideas can be very helpful, specially when working in a team.

- **What's the best part about conferences? How do you make the most of them? Tips and tricks to share?** Don't go to talks, but chat with everyone at posters. The most important track is the hallway track. You will miss out on some things and that's okay. Sneak out and enjoy the city. but go to posters and ask questions. Hang out with cool people (like us) and talk to them.
- **How useful/important is it to have Academic Twitter?** As many good researchers are active on twitter, having it would help you in many ways. You will get many novel ideas/events/papers information earlier. Also, you will grow your network and promote your work easier. That being said, try not to become addicted to it. Also, they often advertise their work soon after they post it on Arxiv, so you will not overlook it if you follow them on Twitter.
- Put attention into writing and presentation skills as it would influence your research path. A useful writing course is available on coursera. Also, look into this article and this talk.
- **How to manage time and what to prioritize?** Look into Devi Parikh's blog for advice.

Finding industry position after ML finishing master/PhD

- If you want to go to industry, internships are very important, so try doing at least one.
- Leetcode helps a lot, but your GitHub portfolio helps more than that in some cases. Industry doesn't look at your publications a lot (except for research scientist positions). Instead, they will rather ask how to turn your publications into productions.
- In Silicon Valey and similar ecumens, more often than not, the initial rounds of interviews are organized by software engineers, so leetcode problems play a major role.
- Open source contributions are valuable.

See also Grad Resources blog with many other useful details.