Best practices for [academic] research in Machine Learning

Makarand Tapaswi

Last edited: December 21st, 2022

Note: Some of these may be applicable even outside ML and academia. But these are almost certainly true for academic, grad-school settings.

Table of Contents

[Research attitude 1](#_Toc804827836)

[Research is different from course work 1](#_Toc2035721620)

[Creating new information is not easy 1](#_Toc1078912353)

[Celebrate the little wins! 2](#_Toc326942759)

[Adopt the scientific method 2](#_Toc666803165)

[Planning for a paper at a top conference? Get ready 2](#_Toc1380137782)

[Never forget ML 101, even after you are experienced 3](#_Toc143021417)

[When working with a new dataset 3](#_Toc321499480)

[When working with a new model or a new variation 3](#_Toc40709683)

[Loss function curves 4](#_Toc364360672)

[Testing performance 4](#_Toc1414362138)

[After a round of interesting results 4](#_Toc1373275653)

[Tips on experiment tracking 4](#_Toc1726084793)

[Weekly meetings 5](#_Toc748008153)

[What to present? 5](#_Toc1035718807)

[How to present? 6](#_Toc926962271)

[Improving written communication 7](#_Toc46729448)

[Searching for related work 7](#_Toc2102835640)

[Writing the related work section in a publication 8](#_Toc2144636005)

[Making publication-ready figures 8](#_Toc1552201670)

# Research attitude

### Research is different from course work

* You are creating new information which the world does not know about! You need to approach it as such.
* Credits, number of hours per week, etc. do not matter. Remember that your research thesis (for MS) carries equal weight as all courses put together!
* You own your ideas and research! Do it because YOU care. If others also care, then that’s even better!
* That said, keep a balanced life. Don’t have super hectic periods (especially away from deadlines) as this could lead to burn out. Also don’t completely abandon thinking about the research topic during the other periods like exams.
* Overall, enjoy your time doing research, contributing new knowledge to the world.
* As you get more senior (e.g., PhD), you can think about what this new knowledge means in the larger context of the research community, public at large, etc. but at the start have fun learning and dabbling in creating something new.

### Creating new information is not easy

* Perseverance is key. Often, a lot of experiments will not yield the desired results. But if you believe in the approach, don’t give up easily. Some level of stubbornness is desirable. Simultaneously, don’t be too stubborn in the face of overwhelming evidence if a particular idea is not working.
* In the current pace of arXiv, every other week you’ll find some shiny new topic or tool or idea. While it’s good to be aware of them, don’t chase after everything that glitters.
* In relation to the point above, keep a lookout for new publications / directions that are related to the topic at hand. It’s generally good to stay updated with related work, so that we are not blindsided.

### Celebrate the little wins!

* There’s a high chance that a lot will NOT go according to your expectations. Most experiments tend to fail. Try and learn something from them and move on.
* It’s important to feel good about the little things that happen as expected, or go your way. For example,
  + When your experiment produces some meaningful result.
  + When you get improvement (even if small) over a previous experiment.
  + When you have a nice new idea that you feel excited to try.
  + When a meeting or presentation goes well and you feel pumped.

### Adopt the scientific method

* Think and design experiments carefully, don’t run them blindly.
* When writing up a paper, consider the thought process that led to the model that is finally adopted. Present those as ablation studies.
* We do not typically use significance testing in ML, but do try and get a feeling for what is a “significant” improvement.
  + This can be done by changing the random seed and analyzing variance in performance for example.
  + Report performance accordingly, it may not be important to consider an improvement in the 3rd decimal place!
* An experiment is not done once it finishes running. It is only complete when we try and understand why that number was obtained, how does it compare against previous experiments or models, what do we learn from the result, and finally, what does the result unlock or what is a good next step.
  + Admittedly, understanding *why* with deep learning is hard! But try your best. Use probes if applicable (nearest neighbors, attention scores, intermediate predictions, etc.).

### Planning for a paper at a top conference? Get ready

* Contribution: new problem, new way to solve existing problems, pointing out something of relevance to the community.
* Clear writing and illustrative, intuitive figures.
  + A good figure pushes reviewers and readers to be wowed. Everyone likes to look at pretty and informative things!
  + Writing should be simple and clear. Don’t convolute things. Bad writing kills the best of ideas.
  + Get feedback on your paper from colleagues, see if they understand it.
* Experiments to validate the claimed contributions:
  + Get ready for it, ML and especially DL is highly empirical!
  + On the order of several hundred (think 500+) to a thousand!
    - Think of how you can extract the gist of 500 experiments, assuming you already had done them. This is writing a paper. This requires good documentation, suggested weekly meeting notes can be very helpful.
  + Many papers from top vision and ML conferences spend 3-4 pages describing experiments. Approximately, 50 results are reported in the paper. Behind the scene though, there are usually 10x more experiments that the authors performed before they settled on the proposed model.
  + Ablation studies showing the method you propose is indeed better than other relatively similar architectures.
* Theoretical papers can be different, but I don’t tend to work on them much.
* It’s a lot of hard work and perseverance. It’s also super exciting to have good reviews and have the paper accepted. Reviewers can also be harsh, don’t take it personally. Do your best, don’t anticipate the result.

# Never forget ML 101, even after you are experienced

### When working with a new dataset

* Look at the dataset, do *human learning* before setting out to teach your machine.
  + Identify key properties, statistics, pitfalls, erroneous or noisy labels, etc.
  + Do the necessary “exploratory data analysis”.
* Check that the dataset and dataloader is set up properly.
* Check that the augmentations make sense for the task and are not messing up the data in a way that creates issues down the line.
  + Visualize the data after applying augmentations to see that reasonable things are being done.
* Check the model inputs and labels are indeed as expected before you send the batch of data to your model.

### When working with a new model or a new variation

* Compute the theoretical loss for a randomly initialized model. Ensure that your validation loss (before any training) is indeed around your theoretical value.
* Check that the model overfits successfully on a small sample of data.
  + Overfitting does not guarantee correctness, but it is a good check nevertheless.
* Think of hyperparameters: batch sizes that allow to fit in GPU memory, appropriate learning rates for optimizers, etc.
  + If fine-tuning, consider when the backbone should be frozen or trained.
* If the model is not learning:
  + Increase the learning rate
  + Play with other hyperparameters
  + Simplify the model, do the overfitting test
  + Check that the gradients are not 0 for some reason
* When things don’t work, or even when things work:
  + Strip the model to an architecture that we know should work.
  + Make small single additions to it.
  + Do not jump the gun and add 5 new ideas at once, however brilliant they are.
    - Even if they work, we will not know why it worked.
    - Usually, adding multiple changes at once doesn’t work.
* Shortcut learning: Deep learning is very prone to “shortcut learning” (<https://arxiv.org/abs/2004.07780> is a great read). Think about how the model could cheat on your data. Try to minimize opportunities for such an occurrence. Typically done by bombarding the model with a ton of data (and compute) or smartly crafting “inductive” biases in the model architecture.

### Loss function curves

* Look at the loss and metric curves (use tools like Weights & Biases <https://wandb.ai/>).
* Don’t try to draw numbers in your head by looking at a column of numbers, that is impossibly hard to do and make sense.
* Look at train and validation curves simultaneously (ideally in the same plot).

### Testing performance

* Do NOT look at test performance until the final models are chosen.
* Ideally, we’ll work with benchmarks that require uploading a set of predictions that returns a black-box number.

### After a round of interesting results

* Go back to the data, see it simultaneously with predictions from your new model.
* See what works, try to create groups of scenarios where the model works.
* See what doesn’t work, try to create groups of scenarios where the model fails.
* Think of new ideas to prevent the model from failing.
* ML research is basically a loop of this process!

# Tips on experiment tracking

ML is a highly empirical field. Most of the research we will do will rely on running experiments with appropriate hypothesis and validating them. A major part of maintaining sanity in ML is tracking experiments properly, making only one change at a time and evaluating things very rigorously. Pay full attention to detail. Every little choice like learning rate, scheduler, whether you have one or two linear layers, including batch norm before or after dropout, before/after ReLU, etc. all can have consequences on results.

* Use version control (preferably GitHub) regularly on your codebase.
  + This is not optional.
  + If you are not familiar with git, please spend a few hours reading and trying <https://git-scm.com/book/en/v2>, it will be totally worth in the long run.
  + Reading the first 2-3 chapters will be enough for most common usage.
  + Make small commits, ensure each experiment has a unique commit associated with it.
* ML has evolved to use config files more than just command line arguments (see the <https://github.com/facebookresearch/detectron2/> repository for example).
  + This is another way to track and log experiments.
  + Everything you run should have a unique config file.
  + Hydra (<https://hydra.cc/docs/intro/>) is a great config manager.
* Use tools like W&B for tracking experiments. They also have some nice academic discounts which we can avail if necessary, or we can always have one id for each project. There is some overhead in setting the logging up, but it’s worth the time investment.
* Summarize results comparing multiple experiments through spreadsheets, W&B reports, etc. Don’t just talk about them, show.
* Whenever you present experiments in your notes or slides, it is a good practice to link the experiment results to the corresponding W&B run, so that we can always dig into details during the meetings or even later, and don’t waste time searching for the right experiment.
* These practices are an integral part of learning, and irrespective of whether you go for a PhD or work in an ML-related industry.

# Weekly meetings

Read these wonderful tweet threads by Jia-Bin Huang for inspiration on meetings:

<https://twitter.com/jbhuang0604/status/1453378296608137229><https://twitter.com/jbhuang0604/status/1418407079077842944>

### What to present?

* There are a lot of memes on how professors don’t remember the projects. I have been at both ends and can vouch that they are true! It’s hard to remember every single detail when working with 10+ projects simultaneously.
  + Provide a two-to-three-minute recap to set the stage for everyone concerned.
  + Briefly state the agenda, the objectives, before diving into model or experiment details.
  + When talking about new tasks or methods, clarify the inputs and outputs.
* Provide frequent and meaningful updates.
  + Use offline messaging tools like Slack / Teams / email / etc. Remember that they are offline tools but try to be reasonably prompt in answering during working hours. Beyond working hours and far away from deadlines, don’t feel pressured to answer – everyone deserves a break.
  + If stuck, please ask the questions, don’t wait till the scheduled meeting.
    - Also, get help from your peer group, often they will be in a good position to clarify things.
    - Asking for help on a messaging platform (instead of talking) forces you to also communicate the question clearly in the first message.
  + Asking for help or a question is not a sign of weakness! It just means you are eager to learn about something. That said, do spend some time trying to find the answer yourself first, we are doing research after all.
* Stick with the plan once agreed during the weekly meeting.
  + If the plan needs to change, do let your advisor know that XYZ is changing because of ABC reasons.
* Please don’t only say “it doesn’t work”, indicate efforts you took to try and debug.

### How to present?

* Be ready!
  + Don’t be late. Sometimes, there are emergencies, try and inform as early as possible. This is just standard courtesy and value for each other’s time.
  + Don’t come to or join the meeting and then start opening relevant tabs.
  + The advisor and the student have one hour per week. Make best use of it.
  + Spending time opening tabs, finding slides, finding results, opening papers on arXiv which are bread and butter for the project, waiting for connection to the VPN, are all a significant waste of time.
  + Five wasted minutes per meeting translates to 4 lost hours per year.
* Use one single slide deck for each project. Present updates in reverse chronological order.
  + Having it all in one place really helps go back and refer to previous experiments, ideas, etc.
  + This presentation can be a living doc – you can edit it during the meeting. Add weekly TODOs, discussions, clarifications, during the meeting. It doubles as a note taking system.
  + Several ideas will be discussed during the meeting as next steps. Don’t try to remember them in your head, note them down on the slides so that it will be available for everyone to look at next week.
  + Also, ideally, create *one* shared drive folder. Put all documents related to the project in that folder. Be kind to your advisor who is probably working on 10 projects and don’t make them juggle across hundreds of loose files.
  + A few months, or a year down the line, it’s fun to look back at the progress.
* Present at the right level of abstraction.
  + Think of how to make the most effective use of time when we get to talk for 1 hour, while you work for the remainder 30-40 hours!
  + It’s useful to spend an hour or two collecting results, creating good graphics.
  + Keep all the necessary information handy: loss plots, final evaluation metrics, etc. to have efficient meetings.
* Do not skip meetings, especially if you are feeling guilty about not being productive.
  + Chat for at least a few minutes, at least on Slack, explain the situation. Talking can provide that much needed shot of motivation.
  + Remember, we have gone through such phases in our student life as well and you are not alone – don’t try and figure everything out by yourself.

# Improving written communication

Jia-Bin Huang has some wonderful tweets on this. I will just refer to them for now.

* Writing clearly: <https://twitter.com/jbhuang0604/status/1437931004451250176>
* Writing a paper: <https://twitter.com/jbhuang0604/status/1437443017510621185>
* Math is prose, number your equations, help the reader:

<http://www.ai.mit.edu/courses/6.899/papers/mermin.pdf>

# Searching for related work

* Find at least one paper in the field which catches your attention – let’s call it paper P.
  + Ideally, find a survey paper, so that most of the below work is done for you!
  + But also learn enough about your subject that you can write one such survey paper.
* Read through paper P’s introduction and related work sections, identify relevant previous material, and start digging.
* Use Google Scholar to find other papers that may be citing the work in P.
  + You can use tools like Semantic scholar to further identify influential citations.
* Starting from a single paper, you can build out a complex web of how each publication relates to each other.
* Get a historical perspective on how the field has evolved.
  + Identify salient points which caused some of these changes.
* Try and cluster the publications into groups that are closely related to each other.
  + This will also help in writing the related work section.

### Writing the related work section in a publication

* Following the above steps will gather enough content to write a related work section. Try to describe each cluster above, indicate examples as each publication.
* Writing a related work section as:

“There are several directions in this area: question-answering [1, 2, 3], captioning [4, 5], fill-in-the-blanks [6, 7], …”

reads much better than a laundry list of papers that basically correspond to the title

“[1] used memory networks for QA, [2] used attention, [3] created a dataset, [4] had a CNN-RNN framework, …”

* + Summarize and abstract away the details for your audience.
* Present some details for works that are closely related to your topic. Also indicate briefly (1-2 sentences) how the proposed work is different.

# Making publication-ready figures

Please refer to my other notes for a detailed exploration of creating scientific figures.

<https://tinyurl.com/IIITH-MT-CreatingFigures>