How to efficiently read a research paper?

(By Shreya Sharma)

Disclaimer: my advice is based on my experience with machine learning and computer vision papers. These tips may not be applicable in other domains.

Dual benefits: collection of ideas + improvement in your own tech writing

- 1. 3 pass approach (funnel structure)
 - a. First pass ~ 20 mins: abstract, introduction and conclusion [select]
 - b. Second pass ~ 2 hrs : Method and experiments [understand]
 - c. Third pass ~2-3 days : actually breakdown the maths and reimplement the paper in your mind and then through code [implement]

Note:

- If the paper does not pass the first pass, don't read it further. It does not interest you.
- If the paper implementation is not required immediately, do not conduct the third pass.
- After second pass, draw a block diagram of the proposed method by hand. This
 ensures that you have completely understood the flow of the method and
 functions of each block.
- 2. 6Q's to break down the structure
 - a. Each paper is trying to sell an idea to solve a problem. You need to clearly identify both. How important/tough is the problem and how much effective is the solution?
 - b. If a paper tries to sell two ideas, that's a warning sign of a bad paper.
 - c. Ask these 5 questions to extract from paper [usually should be clear from introduction]
 - i. What is the main research field and why it is important
 - ii. What are the common problems
 - iii. How the previous work has address these problems
 - iv. What problem is yet unsolved by the previous work
 - v. How is this paper tackling that problem?
 - vi. What is the main contribution/novelty of the paper?

From Andrej karpathy's blog:

"A good flow of ideas is then along the lines of 1) X (+define X if not obvious) is an important problem 2) The core challenges are this and that. 2) Previous work on X has addressed these with Y, but the problems with this are Z. 3) In this work we do W (?). 4) This has the following appealing properties and our experiments show this and that. You can play with this structure a bit but these core points should be clearly made."

- 3. Active and critical reading
 - a. Talk to your paper and question the arguments as you read
 - b. Literally, mark the parts which does not fit your common sense

4. Circle the unknown terms

- a. While reading, there may be many terms or concepts that you are unfamiliar with. Circle all those. Google them after first reading. These are bridge to fill your knowledge gaps.
- Another recommendation is to circle the words which eased your understanding and made your reading smooth. This helps you to learn how to select words for concise, cohesive writing.
- c. If there are too many circles in the paper, it's a warning sign that you are jumping into this paper too early. You must read some more basic paper first, and then shift to the advanced one

5. Mark all the important references

- a. You can refer them to find new papers for your literature survey. Literature survey is another skill which I will discuss about in another blog.
- b. Helps you to create a chain of thoughts and innovations.

6. Build upon it

- a. Your goal to read a paper is to get new ideas and further stimulate new ideas in your mind for your own problem.
- b. So after reading the paper, you can do the following things:
 - i. If you do not have a problem at hand, find the areas of improvement in the paper. Which assumptions/limitations can you minimize? Which application you can make with this technique? What problems are yet to be solved?
 - ii. If you have the problem at hand from a different domain, find how this technique can be used in your domain?
 - iii. What's the author perspective on future work?

7. Formats to save your knowledge

After reading the paper you must create a system to save your knowledge because in this era of information overload, our memory is highly volatile. I suggest three tools:

- a. Spreadsheet create columns
- b. Powerpoint same points as in spreadsheet but you can save figure too
- c. Research journal on the go paper reading