**Data Engineering/Design Patterns:**

* Designing Data-Intensive Applications
* Machine Learning Design Patterns

**Stats/Probability:**

Data Science from Scratch: Joel

Jason Brownlee

3Blue1Brown

**SQL:**

SQL for Data Analysis : Cathy

**Daniel Bourke:**

Sayak paul:

I tried to utilize my off-work hours in a way that helped me improve my ML-specific knowledge as well as strengthen my candidature.

Books: Grokking Machine Learning

If you’re an absolute beginner then I recommend picking up a book ([an example](https://www.manning.com/books/grokking-machine-learning)) or a course ([an example](https://developers.google.com/machine-learning/crash-course)) to get started

Learning is not enough. You need to be able to develop evidence that shows you can apply what you’ve learned successfully. I highly recommend reading [this interview with Emil Wallner](https://blog.floydhub.com/emils-story-as-a-self-taught-ai-researcher/) who’s an “internet-taught” ML Researcher working as a resident at Google.

**Kaggle**

Kaggle is arguably one of the best platforms to develop skills for data preprocessing and applying ML in creative ways to solve unique problems. So, pick an interesting dataset or a competition *just for learning purposes*. Putting the competitive mindset aside, during my initial years it really helped me to develop a mindset of always learning to facilitate self-improvement. If you commit to it hard enough, you will have developed a bunch of useful skills. Over time, you’ll definitely get better.

### Papers / Concepts

Reading research papers is a common undertaking in ML. It can be rewarding to summarize, implement, and blog about a paper that is impactful and tackles interesting problems. Extending

You can summarize a paper in your own words and publish it on platforms like [Medium](https://medium.com/) or even on [your own blog](https://github.com/fastai/fastpages)

If you’re looking for an example, definitely check out [Aakash Kumar Nain’s paper summaries](https://medium.com/@nainaakash012).

Picking a paper could be a non-trivial work especially when there’s always a flood of papers on [arXiv](https://arxiv.org/). I usually follow the blogs of research labs at [Google](https://ai.googleblog.com/), [Meta](https://ai.facebook.com/blog), [AI2](https://blog.allenai.org/), [BAIR](https://bair.berkeley.edu/), etc., to keep myself up-to-date about the work I care about. There’s a good chance you’ll find your niche there. Following the works of the most accomplished researchers from my favorite domains is another practice I incorporate.

In this regard, I highly recommend the following two books that actively cite examples of relevant research papers and also implement them in ways that are practically beneficial: [Deep Learning for Coders with fastai and PyTorch](https://www.oreilly.com/library/view/deep-learning-for/9781492045519/) by Jeremy Howard and Sylvain Gugger, [Natural Language Processing with Transformers](https://www.oreilly.com/library/view/natural-language-processing/9781098103231/) by Lewis Tunstall, Leandro von Werra, and Thomas Wolf. For developing a general understanding of different areas in ML, I recommend the articles on [Distill Pub](https://distill.pub/).

it’s a good exercise to try to implement the novel bits of a paper. The [timm libary](https://github.com/rwightman/pytorch-image-models) is a great example of how paper reimplementations should be structured.

Many ML stalwarts keep pressing on why you should blog and here is one such example: [Why you (yes, you) should blog](https://medium.com/@racheltho/why-you-yes-you-should-blog-7d2544ac1045) by Rachel Thomas.

making videos on papers, concepts, and so on. If you haven’t already, then definitely get to know [Yannic Kilcher](https://www.youtube.com/channel/UCZHmQk67mSJgfCCTn7xBfew) who has revolutionized the way forward in this theme

### Open-source Contributions

From my personal experience, I can confirm that making open-source contributions is one of the most useful ways to stay involved in the domain

I have a [separate presentation](https://www.youtube.com/watch?v=VsBEFUoESR4&t=3123s) on this topic but here, I provide my perspectives for context:

When you’re contributing to a well-maintained open-source library for the first time there’s a high chance that you’ll learn a few things other than just ML. These include writing unit tests, setting up the local development environment, library building tools, etc. This way, you get first-hand exposure to how software engineering is approached in the ML domain in general.

So, not only do you get to contribute to your favorite open-source library (which is an inexplicable feeling anyway), but you also get to learn skills that are practically quite demanding. Beyond these, you get a chance to interact with experts and get their feedback to improve your work. Additionally, you get to collect objective evidence of your skills - coding, thorough understanding of a critical component and the library, building a library, etc. - all of which are noteworthy.

Note that you’re not alone if you’re feeling lost when you’re just starting to contribute to an open-source library. It happens to most. But when you put your mind toward making your contribution anyway, you get to get better in the process.

* If you feel you’re not ready yet to make contributions, working on your own open-source projects is another promising avenue to pursue. Take Andrej Karpathy’s [miniGPT](https://github.com/karpathy/minGPT) project as an example. Besides being an amazing educational resource for learning about the [GPT model](https://en.wikipedia.org/wiki/GPT-3), it serves as a great reference for implementing many of the foundational blocks of [Transformer](https://arxiv.org/abs/1706.03762)-based architectures.

If you’re looking for open-source project ideas then [my presentation](https://youtu.be/dllfKQKlzvg) on this topic might be helpful.

## References from the Community

[Matt](https://twitter.com/carrigmat) (ML Engineer at Hugging Face) says -

[…] I did a few small projects to get familiar with Keras and then tried reimplementing papers or building examples to contribute to places like [keras-contrib](https://github.com/keras-team/keras-contrib) or [keras.io](https://keras.io/).

[Chansung](https://twitter.com/algo_diver) (ML-GDE and MLOps Engineer) says -

[...] Anyways, I actually didn’t plan what to do for the next few years. I just have followed my interests and the joy to participate as a community member. And whenever I make any moves, I found other exciting events are waiting for me. These days, I am really enjoying creating open-source projects and applied ML products, and collaborative projects with you as well.

Damien Benveniste:

Bojan Tungus:

Abhishek Thakur:

SRK: