



Daily Notes

2020 L^AT_EX doc

Author: Unique Divine

Institute: ElegantL^AT_EX Program

Date: April 12, 2020

Bio: Information



Victory won't come to us unless we go to it.

Contents

1	Introduction	1
1.1	A Commonplace (Book)	1
2	Daily Scratch	2
2.1	Aug 9-15	2
2.2	Medical Research Meetings	2
2.3	Algo Trading	2
2.4	Daily Schedule (when free)	3
2.5	Resources	4
3	Metalearning & Doing	5
3.1	Ultralearning	5
3.2	Ultralearning Project Proposals	6
3.3	Getting Things Done (GTD)	9
4	Deep Learning & AI	12
4.1	12
4.2	ML Finance Project	12
5	Quotes	13
6	Math (Probability, Statistics, & Linear Algebra)	14
6.1	Statistics	14

Chapter 1 Introduction

1.1 A Commonplace (Book)¹

A commonplace book is a system for writing down and sorting all manner of tidbits: quotes, anecdotes, observations, and information gleaned from books, conversations, movies, song lyrics, social posts, podcasts, life experiences, or anything else that you might want to return to later.

It's called a commonplace it's an aggregate, an all-encompassing system for storing all of the the most important points you learn in one common place—a central resource that makes it easy to find, re-read, and utilize each piece of wisdom you have obtained.

[2](#)

Why? | The Purpose of this Commonplace Book

1.1.1 We are at war.

Constantly seeking out new experiences and opportunities to grow. We employ active recall methods, watch videos, lectures, and seminars. We listen to mentors and podcasts, captivated by the new insights and perspectives that enter our psychology.

As voracious learners, we are in a forever war

1.1.2 Save Hours on Research

1.1.3 Find Unexpected Connections

1.1.4 Focus future Learning

¹[Wikipedia : Commonplace Book](#)

²[“What Is a Commonplace Book?” \(Article\)](#)

Chapter 2 Daily Scratch

2.1 Aug 9-15

Who is François Chollet?

- works on deep learning at Google in Mountain View, CA.
- creator of the Keras deep-learning library
- contributor to the TensorFlow machine-learning framework.
- wrote a book on deep learning that is very applied rather than theoretical.

“We know the field is fast moving. If the reader looking for more recent free reading resources, there are some good introductory/tutorial/survey papers on Arxiv; I happen to be compiling a list of them”. - Auto Snipe

One of said review papers [\[2\]](#)

2.2 Medical Research Meetings

Aug 18 | Marwah

Aug 24 | Itsif

2.3 Algo Trading

2.3.1 ML Finance Project

Terms to Learn: NLP, bag of words, stop words, stemming, tokenizing, term-frequency inverse document frequency, bigrams, unigrams, Dow Jones Industrial Average (DJIA) index

“ The preprocessing for the text data involved three parts: removing stop words, stemming, and tokenizing. We first removed common words like "the" and "a". This also includes removing non-standard english words, such as emojis and words in other languages. The next step is to performing stemming, which turns words like "running" to "run". From there, we tried many methods of tokenizing our corpus, including bag of words, term frequency–inverse document frequency, and using bigrams and unigrams. We found that for this particular dataset, a bag of words model using unigrams worked the best.

Using our data from the bag of words model, we merged each day with whether or not the Dow Jones Industrial Average (DJIA) index will move up or down for the next day. We thought the movement of the DJIA is able to capture the health of the

overall economy, while not giving too much information into how tech stocks are specifically doing...

The top words associated with DJIA moving up are "gadafi", "Iran" and "sanctions". "

2.3.2 Time Series in Pytorch | Introductory Example

Q: (cloze) Neural networks can be constructed using the `torch.nn` package.

Q: Import the package for constructing neural networks in PyTorch.

```
import torch.nn as nn
```

Q: (cloze) Seaborn comes with built-in datasets.

Q: Load seaborn's flights dataset.

```
flight_data = sns.load_dataset("flights")
```

Q: Why must time series data be scaled for sequence predictions?

When a network is fit on unscaled data, it is possible for large inputs to slow down the learning and convergence of your network and in some cases prevent the network from effectively learning your problem.

Q: sklearn import for scaling data?

```
from sklearn.preprocessing import MinMaxScaler
```

2.4 Daily Schedule (when free)

Lex Fridman posted a video today (2020 Aug 27), detailing [his daily](#) schedule outside of work commitments. The stages are as follows:

- (1) Morning (2) Mantra (3) Deep work I (4) Medium depth (5) Exercise
 - (6) Shower (7) Deep work II (8) Eat (9) Shallow work (10) Paper reading
- I'll go into a bit more depth on the sections below and try to implement some of these

practices in my daily schedule to see what I can learn from Lex.

1. Morning
2. Mantra / Daily Review
 - Rules & constraints
 - Gratitude
 - Long-term goals
 - Short-term goals
 - Core principles
3. Deep Work I (4 hr)
4. Social Media and Music Practice
5. Exercise

Exercise every day, no matter what. Even if injured, find a body part and exercise it.
6. Shower

7. Deep Work II (4 hr)
8. Eat
2k calories, all your nutrients vegetables, protein, and fat
9. Shallow Work (1 hr)
10. Paper Reading (1 hr)

2.5 Resources

- [A clear and concise beginner hugo tutorial](#)

Chapter 3 Metalearning & Doing

3.1 Ultralearning

Ultralearning: It's about learning quickly and effectively. - It's a strategy - self-directed - intense - rapid speed with great efficiency

"Despite their idiosyncrasies, the ultralearners had a lot of shared traits. They usually worked alone, often toiling for months and years without much more than a blog entry to announce their efforts. Their interests tended toward obsession. They were aggressive about optimizing their strategies, fiercely debating the merits of esoteric concepts such as interleaving practice, leech thresholds, or keyword mnemonics. Above all, they cared about learning. Their motivation to learn pushed them to tackle intense projects, even if it often came at the sacrifice of credentials or conformity."

How to become an ultralearner

1. **Metalearning:** First Draw a Map
2. **Focus:** Sharpen Your Knife
3. **Directness:** Go Straight Ahead
4. **Drill**
5. **Retrieval:** Test to Learn
6. **Feedback:** Don't Dodge the Punches
7. **Retention:** Don't Fill a Leaky Bucket
8. **Intuition: Dig Deep Before Building Up** Develop your intuition through play and exploration of concepts and skills.
9. **Experimentation**

3.1.1 Core Message (YouTube)

A lot of what's taught in school is theory useful for PhD students.

An ultralearning project is about building skills, not just knowledge.

To Start Any Ultralearning Project

1. Make and follow a Metalearning map
If I want to do X ,
 - what concepts do I need to understand?
 - what facts do I need to memorize?

- what procedures do I need to practice?

Think hard about what you want to be able to do by the end of your ultralearning project. Then identify what skills will be critical to your success.

2. Design and use practice drills

- Based on the info from metalearning, design drills.
- Practice drills should have quick feedback loops that target key areas, just like a bodybuilder would use dips to target the triceps.

3. Overlearn

Go beyond the requirements of what is necessary to learn to further reinforce what IS necessary.

To embrace overlearning, ask:

- what's my target performance, and
- what's the next level?

Commit to an even harder performance than the one you're training for.

To recap: 1. Metalearn 2. Drill 3. Overlearn

3.1.2 How to Start Your Own Ultralearning Project (Part One)

Designing your own ultralearning project has three parts:

1. Figuring out what you want to learn deeply, intensely and quickly.
2. Choosing which format you want for your project.
3. Preparing to start learning.

Step 1: What Do You Want to Ultralearn?

Step 2: Choose the Project Format

Step 3: Preparing to Learn

3.2 Ultralearning Project Proposals

3.2.1 PyTorch | Deep Learning

Structured Resources:

- [Deep Learning Book](#)
- [sentdex's PyTorch | Tutorial](#)
- [freeCodeCamp | Tutorial](#)
 - (0:00:00) Introduction
 - (0:03:25) PyTorch Basics & Linear Regression
 - (1:32:15) Image Classification with Logistic Regression
 - (3:06:59) Training Deep Neural Networks on a GPU with PyTorch

- (4:44:51) Image Classification using Convolutional Neural Networks
- (6:35:11) Residual Networks, Data Augmentation and Regularization
- (8:12:08) Training Generative Adversarial Networks (GANs)
- [Intro to PyTorch | Tutorial](#)
- [MIT Deep Learning | Course](#)
- [MIT Deep Learning and AI | Lectures](#)

Keywords:

Backpropagation, Hyperparameter, Feed-forward neural nets, Convolutional neural nets, Regularization

Mini Project:

- Read through some of the deep learning book.
- Learn from sentdex's tutorial
- Complete a small project with PyTorch. Maybe dogs/cats or something
- freeCodeCamp tutorial
- Gain high-level understanding of the keywords, enough that you could explain these concepts off the top of your head. Understand how they work and their significance.
- Intro to PyTorch Tutorial

Note You can mix up the steps. If you want to bounce between tutorials or start working on the project before you've finished everything else, that's completely fine. The same goes for the reading. As long as everything gets done, consider this mini project a success.

3.2.2 Blogging

3.2.3 Python DS Handbook

It's Tuesday, August 18th and there are about two weeks until the semester starts.

I could definitely do with improving my Python data analysis fundamentals before the semester starts. Both the Python DS Handbook and Wes Mickinney's book are awesome resources that I could use to up my skills. So here, I'm proposing a 6 day project to quickly and efficiently milk these resources.

Motivation:

I've been trailing off on too many tangents. My recent project involving LSTMs sent me on an RNN binge that turned into a neural network binge that turned into a deep learning and AI

binge. This isn't necessarily a bad thing. Some of that playful curiosity and interest was reignited in delving in this exploration. Heck, I even started reading academic papers for fun again.

Perhaps I'd lost sight of my true mission because I got too engrossed in the daily grind and tasks in front of me. Stopping to consider my interests didn't even cross my mind because lately I keep having to put out new fires. So, I took a step back. I wrote, I thought, and I've read a lot in the past few days. I burned through dozens of books and articles, particularly on finance, investing, deep learning, metalearning, and productivity. And, I'm ready to go back and apply some more of what I've learned.

Read,

Proposal:

You (I) have 6 full days to accomplish the follow, starting now:

- Read (and highlight), note-take from Python DS Handbook.
 - This is a pace of roughly 40 pages per day. Try to get through 100 a day.
 - 1. **Skim.** Search for interesting content you want to learn and relevant sections. Mark or record these sections for reading.
 - 2. **Read and Highlight** the sections you want to learn in more detail.
- Create content from the notes in the DS Handbook. This includes but is not limited to
 - creating anki cards
 - adding to your cookbook
- Depending on how early you finish, you can decide whether you'd gain more from going through Wes's book or some other resource.
- **Also**, study 1-2 hours of Japanese to catch up on reps each day and exercise 30 minutes minimum.

Reflection

At a high level, tell me what happened: Surprise surprise. I didn't finish the reading goal, however this does not mean that the project was a failure. I believe I discovered pretty quickly in working on this reading project that I wasn't making the most time-efficient gains from taking a bottom-up approach to NumPy.

What to change for future projects: I saw specific examples of NumPy concepts that would've helped me so much in my recent ML for Finance course. And the crazy thing was, I didn't need to know a lot of this info cold. I would've benefit just as much from having the familiarity of recognizing what utilities were even possible. Even just reading through the sections on broadcasting, masking, and universal functions would've helped me greatly.

If I could go back in time 3 months, I would read and highlight through this book cover to cover. I would take notes on some of the concepts and utilities available and potentially add

them to a cookbook instead of making Anki card active recall questions of everything.

The time saved using the text as a reference instead of a tutorial would've helped me spend more time working on end-to-end projects, where my learning environment was stimulating, motivating, and had clear transferable value to potential employers and others around me.

Key Takeaways:

1. Read first.
2. Archive.

3.3 Getting Things Done (GTD)

“If your mind is empty, it is always ready for anything; it is open to everything”-
Shunryu Suzuki

Basic workflow 1. Capture 2. Clarify 3. Organize
 4. Reflect 5. Engage

3.3.1 GTD Overview

Projects

A project is any outcome for which you need more than one action step to achieve it. You often do not *do* projects, you do actions which may be related to the completion of a project.

Natural Planning of Projects is a way to think of projects to create maximum value with minimal effort and time expenditure.

1. Why | Define the purpose, principles, and values of the project
2. What | Visualize the desired outcome
3. How | Brainstorm and organize ideas on how to accomplish your goal(s)
4. When / What now? | Identify your next immediate actions.

Capture

You must capture all the things you consider incomplete in your universe; personal or professional, big or small, urgent or not.

Clarify

Clarifying is the act of defining and clarifying, one by one, all the stuff in your inbox, until it is empty.

(1) trash it, if it is worthless, (2) incubate it in the Someday/Maybe list (3) keep it as Reference Material, if it is potentially useful information.

If an action takes less than two minutes, do it. If now is not the right time (be careful not to procrastinate), defer it to a specific time in the future.

Organize

To organize things that are actionable, you need a Projects List

The calendar is sacred territory. If you put something there, it is because it has to be done on the date indicated.

Reflect

The system cannot be static. If you want to be able to correctly choose your actions, you have to keep the system current. You need to review the whole picture of your life and your work at regular intervals and at the appropriate levels.

Every day, you will need to review your Calendar and Next Actions list organized by contexts.

Every week, you must review the remaining lists to keep your system clean, clear, complete and updated. The purpose of the Weekly Review is that your mind gets empty again.

Every so often you should review the big picture; clarifying your long-term goals and visions and principles that ultimately determine how you make your decisions.

Engage

The purpose of all this management and workflow is to facilitate the decision of what you should be doing at any time.

Your priorities are set by a hierarchy of levels or perspective, from top down (1) your life purpose, (2) your vision of yourself in the future, (3) your medium-term goals, (4) your areas of responsibility, (5) your current projects and (6) the actions you need to do every day.

3.3.2 Further Reading

[The Science Behind GTD](#)

[How to find your life purpose](#)

[10 reasons why GTD will fail](#)

[The Weekly Review, in detail](#)

3.3.3 Review, in detail

Review is a key component of GTD, to the point that **if you do not do it effectively, you are probably not following the methodology the right way**. It is a time we spend to ensure that we are always working on what is actually important, a way to be proactive and take control.

Get clear

The first phase of review is to do clean-up, collecting everything and processing it. The result should be that everything is in the GTD system and the Inbox is empty. Clarify and assign dates to tasks.

Get current

The second stage is to keep your system up to date. This is **absolutely necessary for you to fully trust your system**. Review the:

- **Inbox, past, and upcoming events** | Is there anything pending? Confirm that everything is done or update deadlines if necessary.
- **Projects** | Evaluate the status of each project. Review the support material and add new actions if necessary. Make sure there is a next action for each active project.

Get creative

You have everything under control. Now it is time to dig up old longings, to imagine, to dream, and to invent.

- **Someday/Maybe list** | Is it time to activate a project that was put off or abandoned? Are there things that no longer make sense?
- **Imagine** | What are your big picture ambitions? How can you improve? What new ideas should you add to the system?

Chapter 4 Deep Learning & AI

4.1

Generative Adversarial Networks

Read [1]

4.2 ML Finance Project

example w/ multivariate time series in PyTorch

Q: (cloze) Neural networks can be constructed using the `torch.nn` package.

Q: Import the package for constructing neural networks in PyTorch.

```
import torch.nn as nn
```

Q: (cloze) Seaborn comes with built-in datasets.

Q: Load seaborn's flights dataset.

```
flight_data = sns.load_dataset("flights")
```

Q: Why must time series data be scaled for sequence predictions?

When a network is fit on unscaled data, it is possible for large inputs to slow down the learning and convergence of your network and in some cases prevent the network from effectively learning your problem.

Q: sklearn import for scaling data?

```
from sklearn.preprocessing import MinMaxScaler
```

We know the field is fast moving. If the reader looking for more recent free reading resources, there are some good introductory/tutorial/survey papers on Arxiv; I happen to be compiling a list of them.

One of said review papers [2]

[2, hello]

Chapter 5 Quotes

“your mind is for having ideas, not holding them.” - David Allen

“Some very specific but seemingly mundane behaviors, when applied, produce the capacity to exist in a kind of sophisticated spontaneity, which, in my experience, is a key element to a successful life.” - David Allen

“We should hunt out the helpful pieces of teaching and the spirited and noble-minded sayings which are capable of immediate practical application—not far fetched or archaic expressions or extravagant metaphors and figures of speech—and learn them so well that words become works.” - Seneca

“I have a friend who’s an artist and has sometimes taken a view which I don’t agree with very well. He’ll hold up a flower and say “look how beautiful it is,” and I’ll agree. Then he says “I as an artist can see how beautiful this is but you as a scientist take this all apart and it becomes a dull thing,” and I think that he’s kind of nutty. First of all, the beauty that he sees is available to other people and to me too, I believe. . .

I can appreciate the beauty of a flower. At the same time, I see much more about the flower than he sees. I could imagine the cells in there, the complicated actions inside, which also have a beauty. I mean it’s not just beauty at this dimension, at one centimeter; there’s also beauty at smaller dimensions, the inner structure, also the processes. The fact that the colors in the flower evolved in order to attract insects to pollinate it is interesting; it means that insects can see the color. It adds a question: does this aesthetic sense also exist in the lower forms? Why is it aesthetic? All kinds of interesting questions which the science knowledge only adds to the excitement, the mystery and the awe of a flower. It only adds. I don’t understand how it subtracts.”
- Richard Feynman on the beauty of a flower

“AI began with an ancient wish to forge the gods.” - Pamela McCorduck, *Machines Who Think*, 1979

Chapter 6 Math (Probability, Statistics, & Linear Algebra)

6.1 Statistics

Bibliography

- [1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- [2] Raghu, M. and Schmidt, E. (2020). A survey of deep learning for scientific discovery. *arXiv preprint arXiv:2003.11755*. <https://arxiv.org/pdf/2003.11755.pdf>.