Daily Notes

2020 LaTeX doc

Author: Unique Divine

Institute: ElegantLaTeX Program

Date: April 12, 2020

Bio: Information

ElegantLaTeX Program

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

Contents

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 Raghu and Schmidt (2020)

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. I first removed common words like "the" and "a". This also included removing non-standard english words, such as emojis and words in other languages. The next step was performing stemming, which turns words like "running" to "run". From there, I tried many methods of tokenizing the corpus, including bag of words, term frequency–inverse document frequency, and using bigrams and unigrams. For this particular dataset, a bag of words model using unigrams worked the best.
>
> Using our data from the bag of words model, I merged each day with whether or not the Dow Jones Industrial Average (DJIA) index will move up or down for the next day. I thought the movement of the DJIA would 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 Software Project Portfolio

Key factors

- Completeness
- Functionality
- Readability
- Documentation/information

2.6 Hugo

Start Sep 16 (Mike Dane Tutorial Series)

2.6.1 Intro to Hugo -(video)

- Hugo is a static site generator.
- Static website generators allow you to compromise between writing a bunch of static html pages and using a heavy, and potentially expensive, content management system.
- Why Hugo? It's extremely fast.
- 2 kinds of websites, dynamic and static. Dyanmic ex. Facebook. Facebook pages are dynamically generated for each user. For static websites, what you see is what you get.
- Static websites are notoriously harder to maintain b/c you lack some fo the flexibility of things on a dynamic site. Usually you can't use much conditional logic, functions, or variables.
- However, static pages are extremely fast.
- Hugo is great for a blog, portfolio website, etc.
- Hugo doesn't explicity require you to write a single line of HTML code.
- Flexibility | With that said, if you want to go in and change every little detail of the layout of the site, you cna do that. You can write as much of the HTML and have as much control as you'd like.
- Hugo is 100% free and open-source.

2.6.2 Intalling Hugo on Windows - (video)

- Mine's already installed. I'll skip this for now.

- a
- a

2.6.3 Creating a new site - (video)

- skip, a bit too easy

2.6.4 Installing & Using Themes - (video)

- my themes are already installed

2.6.5 Creating & Organizing Content - (video)

- Hugo has 2 types of content: single pages and list pages
- List content lists other content on the site. You can call this a list page.
- Individual blog posts are single pages.
- Your posts should not just be in the content directory. They should be in directories inside the content directory.
- A list page is automatically created for directories inside the content folder. Hugo automatically does this. Note, this only occurs for directories at the "root" level of the content directory. For example: `content/post/` would generate a list page at `site.com/post/`, but `content/post/dir0` would not.
- If you want a list page to be generated for a dir that is not at the root level of the content dir, you have to create an **index filed**, `_index.md`. For a convenient and efficient way to do this from the cmd, use `hugo new post/dir0/_index.md` (above above example), then there will be a list page for dir0. Content can also be added to`_index.md` and it should show up on the page.
- Additionally, for list pages that are aautomatically generate by hugo, you can edit the content by adding an index.md to those as well. Ex. `content/post/_index.md`.

2.6.6 Front Matter- (video)

- Front matter in Hugo is what is commonly called meta data.
- Front matter is data about our content files.
- The metadata automatically generated by Hugo at the top of md files when using `hugo new` is front matter
- Front matter is stored in key-value pairs
- Front matter can be written in 3 different languages: YAML, TOML, and JSON
- The defualt lang for front matter in Hugo is YAML
- YAML - indicated by "–",
- TOML - indicated by "+++" and uses "=" instead of ":",

- JSON - indicated
- You can create your own custom front matter variables.
- Front matter is super powerful in its utility.

2.6.7 Archetypes - (video)

- How does the default front matter from using `hugo new   .md` get selected? Short answer: archetypes
- An archetype is basically the default front matter template for when you create a new content file.
- Archetypes are modified under `static/themes/archetypes/default.md`
- Suppose your content dir has a subdirectory, `content/dir0`. If you wanted to create an archetype for the files in dir0, you'd simply create `dir0.md` inside the archetypes dir.

2.6.8 Shortcodes - (video)

- Shortcodes are predefined chunks of HTML that you can insert into your markdown files.
- Let's say you have a md file that you want to spice up by adding in some custom HTML. For instance, maybe you'd like to embed a YouTube video. Normally this would require lots of HTML that you'd have to paste it. Shortcodes can allow you to sidestep this. Hugo comes with a YouTube video shortcode predefined.
- General shortcode syntax `< shortcode_name param0 >`
- Youtube shortcode | For a YouTube video with url, "youtube.com/watch?v=random-text", the shortcode we'd use to embed would be `< youtube random-text >` because "random-text" is the id of the youtube video and the only parameter for that shortcode.

2.6.9 Taxonomies - (video)

- Taxonomies in hugo are basically ways that you can logically group different pieces of content together in order to organize it in a more efficient way.
- Hugo provids 2 defualt taxonomies: tags & categories
- All taxonomy information is declared in front matter. In YAML, tags has the syntax `tags: ["tag0", "tag1", ...]`

2.6.10 Templates - (video)

- Templates here mostly refers to HTML templates. If you're not comfortable writing HTML, CSS, and coding for the web, templates might be a little bit above your head.
- A hugo theme is actually made up of hugo templates.
- Any template that you use in Hugo is going to be inside `themes/theme-name/layouts`. This is where all the templates live.

- `/layouts/default` usually contains a default style for list and single pages by use of `list.html` and `single.html`.

2.6.11 List Templates - (video)

- List templates give default HTML layout to list content files.
-

Resources

- A clear and concise beginner hugo tutorial
- CSS Crash Course for Absolute Beginners

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 necessry 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 Project Proposals

3.2.1 PyTorch | Deep Learning

(August 25)

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 Adverserial 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

Meta Project:

There's a list of tasks below. You can mix up the steps and 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.

- Read through some of the deep learning book.
- Learn from sentdex's tutorial
- Complete a small project with PyTorch. Write it as a tutorial and share it with a few people.
- 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

Libraries | PyTorch ignite, fastai, skorch

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) Discard a task if it is worthless, (2) incubate it in the Someday/Maybe list, or (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.

  paragraph*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 eb doing at any time.

Your priorities are set by a hierachy 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 GTD | 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 aything 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?

3.4 How to Read

3.4.1 How to read a scientific paper

2017 scientific paper by Hubbard and Dunbar, about reading scientific papers.

Hubbard, K. E., & Dunbar, S. D. (2017). Perceptions of scientific research literature and strategies for reading papers depend on academic career stage. PloS one, 12(12), e0189753.

image

1. Read the intro and background first, then the abstract (fast pass).
2. Read discussion (fast pass)
3. Attempt to interpret figures and define variables (highlight)
4. Re-read the other sections(highlight, slow pass)
5. Summarize the intro and background
6. Synthesize notes and ask questions. Record this somewhere (like a commonplace book)
7. Read methods, experiments, results, conclusion (fast pass). If necessary, go through a slow pass, highlighting and archiving the most salient points
8. **Don't forget** to take note of key references.
9. Attempt to solve all of your mysteries.

Chapter 4 Deep Learning & AI

4.1 Important Papers

Goodfellow et al. (2016)
Singh et al. (2015)

4.2 Generative Adversarial Networks

4.2.1 GANs Goodfellow et al. (2014)

Abstract & Introduction

Abstract: Goodfellow et al. propose a new framework for estimating generative models via an adversarial process, in which two models are simultaneously trained:

- G: a generative model that captures the data distribution, and
- D: a discriminative model that estimates the probability that a sample came from the training data rather than G

The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples.

Introduction: In the proposed adversarial nets framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution.

- **Counterfeiters Analogy:** The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

In this article, we explore the special case when the generative model generates samples by passing random noise through a multilayer perceptron, and the discriminative model is also a multilayer perceptron. We refer to this special case as adversarial nets. In this case, we can **train both models using only the highly successful backpropagation and dropout algorithms** [17] and sample from the generative model using only forward propagation.

Adversarial Net Algorithm

The adversarial modeling framework is most straightforward to apply when the models are both multilayer perceptrons. To learn the generator's distribution p_g over data $\boldsymbol{x}$, $p_g(\boldsymbol{x})$, we define a prior on input noise variables, $p_{\boldsymbol{z}}(\boldsymbol{z})$, then represent a mapping to data space as $G(\boldsymbol{z}|\theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. We also define a second multilayer perceptron $D(\boldsymbol{x}|\theta_d)$ that outputs a single scalar. $D(\boldsymbol{x})$ represents the probability that $\boldsymbol{x}$ came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1 - D(G(\boldsymbol{z})))$. In other words, D and G play the following two-player minimax game with value function, $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} \left[\log D(\boldsymbol{x}) \right] + \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} \left[\log \left(1 - D(G(\boldsymbol{z})) \right) \right].$$

Goodfellow et al. present a theoretical analysis of adversarial nets, essentially showing that the training criterion allows one to recover the data generating distribution as G and D are given enough capacity, i.e., in the non-parametric limit.

Figure 1 explains this approach: GANs are trained by simultaneously updating the discriminative distribution, D, so that it discriminates between samples from the data generating distribution, $p_{\boldsymbol{x}}$, and those of the generative distribution p_g (G).

After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate b/w

the two distributions, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$.

Algorithm 1: Minibatch SGD training of GAN. The number of steps to apply the discriminator, k, is a hyperparameter. $k = 1$ is the least expensive option.

---

for *number of training iterations* **do**

  for *k steps* **do**

  # Apply discriminator

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior, $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution, $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log \left(D(\boldsymbol{x}^{(i)}) \right) + \log \left(1 - D(G(\boldsymbol{z}^{(i)})) \right) \right].$$

  end

  # Update generator

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior, $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(G(\boldsymbol{z}^{(i)})) \right).$$

end

The gradient-based updates can use any standard gradient-based learning rule.

Note:

- $k = 1$ was used in the experiments.
- Momentum was the gradient-based learning rule used in the experiments.

Related Work

The adversarial nets framework does not require a Markov chain for sampling. Because adversarial nets do not require feedback loops during generation, they are better able to leverage piece-wise linear units [19, 9, 10], which improve the performance of backpropagation but have problems with unbounded activation when used in a feedback loop.

4.3 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 Raghu and Schmidt (2020)

(Raghu and Schmidt, 2020, hello)

4.4 Deep Learning for Genomic Risk Scores

" A central aim of computational genomics is to identify variants (SNPs) in the genome which increase risks for diseases. Current analyses apply linear regression to identify SNPs with large associations, which are collected into a function called a Polygenic Risk Score (PRS) to predict disease for newly genotyped individuals. This project is broadly interested in whether we can improve performance of genomic risk scores using modern machine learning techniques.

A recent study assessed the disease prediction performance of neural networks in comparison to conventional PRSs, but did not find evidence of improvement. This project will explore whether neural networks can improve performance by incorporating gene expression data to the training process. Gene expression is often integrated with SNP data in Transcriptome-Wide Association Studies (TWAS), which bear some resemblance to neural network architectures with SNPs as input nodes, genes as intermediate nodes, and disease status as the output node. Modeling this process as a neural network however will require defining a more unconventional architecture in which a small subset of hidden nodes is anchored to observed values.

This project is designed for students with experience in machine learning topics and preferably with deep learning tools such as tensorflow or pytorch. Students should also be interested in applying machine learning and statistics to genomics applications." - Jie Yuan

Terms to know: Computational genomics, variants, single-nucleotide polymorphism (SNP), genome, Polygenic Risk Score (PRS), Transcriptome-Wide Association Studies (TWAS),

gene(s), genomics, genotype, intermediate node (NN), hidden node (NN)

Q: Explain at a high level how PRSs tells us disease risk.

Q: "In what task were neural networks outperformed by conventional PRSs?"

Q: What is meant by "improve performance of genomic risk scores?"

genotype : An individual's collection of genes. Also can refer to the two alleles inherited for a particular gene.

node (NN) : An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another[1].

hidden node (NN) : A node in a hidden layer.

hidden layer (NN) :

4.4.1 Papers

Polygenic Risk Scores (paper) Wray and Goddard (2010)

Abstract (mining)

recurrence risks : In genetics, the likelihood that a hereditary trait or disorder present in one family member will occur again in other family members[2].

"Evidence for genetic contribution to complex diseases is described by recurrence risks to relatives of diseased individuals."

This is distinguished from recurrence risk for cancer, which is the chance that a cancer that has been treated will recur.

gene : a sequence of DNA that codes for a specific peptide or RNA molecule; the physical and functional unit of heredity.

locus : the position of a gene on a chromosomes

somatic cell : any cell of the body except sperm and egg cells. A non-germline cell. any biological cell forming the body of an organism (except gametes).

sôma (Ancient Greek): body

genome : An organism's complete set of DNA, including all of its genes. Each genome contains all of the information needed to build and maintain that organism. In humans, a copy of the entire genome—more than 3 billion DNA base pairs—is contained in all cells that have a nucleus [3].

"genome-wide association"

allosome : (1) A sex chromosome such as the X and Y human sex chromosomes. (2) An atypical

---

[1] https://en.wikipedia.org/wiki/Artificial_neural_network

[2] https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/recurrence-risk

[3] https://ghr.nlm.nih.gov/primer/hgp/genome

chromosome [4].

allo- (Greek): other, differnt

autosome : Any chromosome that is not a sex chromosome. The numbered chromosomes.

auto (Greek): self, one's own, by oneself, of oneself

-some, soma (Greek): body

allele : (genetics) One of a number of alternative forms of the same gene occupying a given position, or locus, on a chromosome.

Borrowed from German Allel, shortened from English allelomorph. Ultimately from the Ancient Greek prefix allēl- from állos ("other").

"their effects and allele frequencies"

allelomorph: another term for allele.

risk loci :

"genome-wide association studies allow a description of the genetics of the same diseases in terms of risk loci..."

haploid : the quality of a cell or organism having a single set of chromosomes.

diploid : the quality of having two sets of chromosomes.

"Sexually reproducing organisms are diploid" (having two sets of chromosomes, one from each parent)

eukaryotes : Organisms whose cells have a nucleus enclosed within a nuclear envelope.

gamete : A mature sexual reproductive cell, as a sperm or egg, that unites with another cell to form a new organism. A haploid cell that fuses with another haploid cell during fertilization in organisms that sexually reproduce. A mature haploid male or female germ cell which is able to unite with another of the opposite sex in sexual reproduction to form a zygote.

gamete (Ancient Greek): to marry

zygote : A eukaryotic cell formed by a fertilization event between two gametes.

zygōtos (Greek): joined. yoked.

monozygotic : Monozygotic (MZ) or identical twins occur when a single egg is fertilized to form one zygote (hence, "monozygotic") which then divides into two separate embryos.

"monozygotic twins"

empirical :

"generate results more consistent with empirical estimates"

genetic variants :

Q: A human cell containing 22 autosomes and a Y chromosome is a sperm.

Neural Networks for Genomic Prediction (paper) Pinto et al. (2019)

---

[4] https://www.merriam-webster.com/medical/allosome

Transcriptome Wide Association

4.4.2 Meetings

Jie Yuan meeting #2 (Sep 10)

- There's a dot product and its output is passed through some activation function like sigmoid or ReLU. It's still linear in the sense that there's some sort of function that takes in a dot product (which is the definition of a linear operation). You can think of [the betas] as the weights in the neural network because the betas are the coefficients of a linear model.
- Liability is discussed in the multi-locus models paper.
- In the genomics context, this is called the **liability threshold model**. If you google that, you'll find some papers. In a broader machine learning sense, this is called the **probit regression model**. WIkipedia probably has a sufficient article on it.
- The gist of it is that it's a model to map continuous sums (the dot products basically) into binary labels: cases and controls. This can be any sort of logistic regression type thing where you have 0s and 1s. Basically, the **probit model is an alternative to the logistic regression model**.
- The liability is related to what's called the "link function". The link function in the probit model is the normal CDF function. There's a term that goes into the link function, and that term is what we're calling liability.
- The liability is basically the product of the vector of genotypes and the vector of betas. It's the linear part of the generalized linear model.
- Once you have the liability and plug it into the link function. Here, the link function is the probit regression. In a neural network, this would be the linear activation function (sigmoid, ReLU, etc.).
- You know how a sigmoid has a domain that's $(-\infty, \infty)$? It's range is (0,1), so what that function does is map a real value into being a probability. It gives you the probability of being a case. If you look at the normal CDF function, you find that it looks almost the same.

Why is the liability threshold model better than or different from the logistic reg model?

- One reason people use log reg more often is that the betas from log reg are interpretable. The have a meaning in terms of odds ratios. In a log reg model, the effect sizes are the natural log of the odds ratios.
- One disadvantage of log reg compared to liability-threshold is that log reg doesn't have a concept of the underlying distribution of the liability. In log reg, you get the linear $X\beta$ term, but there's no sort of distribution around it. In the liability-thresh model, we say that the liability, itself, is standard normally distributed. By taking the normal CDF, what you're doing is mapping that $X\beta$ value, the liability value, onto this distribution and

asking, "what's the probability that it's larger than some threshold?"

- larger than threhsold $\implies$ label as 1, smaller $\implies$ 0.
- Because liability-threshold has that normal distribution, it gives you more concepts to play with such as variance explained, which the logistic reg model doesn't have.
- liability threshold model in simple terms

  from wikipedia: Liability threshold model

schizophrenia (SZ) example:

- Whether or not someone has SZ comes from a combination of factors in both their genes and environment.
- There are all sorts of variables. Imagine hypothetically that you could collect all of them and produce a score from that (that determines whether or not you have SZ). That score would be the liability.
- The liability is a $\mathcal{N}(0,1)$ distribution that, if higher than the threshold, predicts/indicates the patient has SZ.
- The problem with that is that we don't actually observe everything. We don't truly observe every factor that goes into whether you have SZ b/c (1) we can't measure one's environment and (2) we can't know all of the genomic risk factors from GWAS. We can only identify the largest risk factors.

  This means we are identifying only a small fraction of what goes into genetic risk for SZ.
- Essentially, what we have is a small subset of the factors that actually determine your SZ risk (which we're reasonably confident indicate this relationship). In this example, some subset of the risk factors like the genomic variance which increase your risk for SZ.
- What that means is that inside this standard normal liability distribution/function, what we actually know is a small subset of the factors that go into that. If we score people on that subset of factors, what we get is a smaller distribution, a distribution that has a variance that's a small fraction of the total liability variance, which we can't observe. The variance of that small fraction that we observe, that's the **variance explained**.
-

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 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

"We cannot solve our problems with the same thinking we used when we created them." - Albert Einstein

"I can calculate the motion of heavenly bodies, but not the madness of people." - Isaac Newton

Chapter 6 Math (Probability, Statistics, Linear Algebra, etc.)

Bibliography

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

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.

Pinto, C., Gill, M., Heron, E. A., et al. (2019). Can artificial neural networks supplant the polygene risk score for risk prediction of complex disorders given very large sample sizes? *arXiv preprint arXiv:1911.08996*.

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.

Singh, K., Sandhu, R. K., and Kumar, D. (2015). Comment volume prediction using neural networks and decision trees. In *IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015)*, Cambridge, United Kingdom.

Wray, N. R. and Goddard, M. E. (2010). Multi-locus models of genetic risk of disease. *Genome Medicine*, 2(2):10.