



Australian
National
University

Honours Logbook

Semester 1 2024 - Semester 1 2025

N.B.: This logbook contains notes and updates on my progress and findings throughout my Honours year.

Maintainer:

Jeffrey Liang (u7013004), ANU School of Computing

Contributors:

Jeffrey Liang (u7013004), ANU School of Computing

Contents

1 Overview	3
2 Progress Logbook	4
2.1 Progress	4
3 Topic A to focus on	7
4 Topic B to focus on	8
4.1 Algorithms	8
4.2 Code	8
4.3 Diagrams	9
5 Topic C to focus on	10
5.1 Figures	10
5.2 Tables	10
A References	11
B Resources	12
C TODOS	13

1 Overview

Welcome to my progress logbook!

Hello. Topic. Challenges. Research interest.

2 Progress Logbook

2.1 Progress

Sun 12/05/2024:

Reading MeshSDF paper.

Sat 18/05/2024:

Reading MeshSDF, the Learning Implicit Fields papers, and Structure from Motion. Need to look up what the following mean:

- differentiable rendering, surface reconstruction

Sun 19/05/2024:

Reading more from Structure-from-Motion Revisited. Didn't really understand much of SfM last night.

Thu 27/06/2024:

Attempting 3D version of the ellipse problem.

Question: Why do we invert the u_i s? Answer: you bring the 1 over and multiply by some expression on both sides.

Fri 5/07/2024:

Having been making headway on the non-axis-aligned ellipsoid. I've been able to fit an ellipse to one but I haven't been able to do the bi-level optimisation. I'm going to try a couple approaches:

- reparametrise everything to radians and a, b, c rather than squared inverses of semi-axes
- try original objective function rather than focusing on optimising rotation
- last resort is to figure out what the SqrtBackward error is

Sun 7/07/2024:

Was able to switch implementation to radians and a, b, c rather than squared inverses $1/a^2, \dots, 1/c^2$. It's still temperamental though and will hopefully be finetuned when meeting with Chamin who hopefully has a few tricks. Going to read some papers today about DRWR and the DDN and take notes to try and fully understand derivations etc.

Sat 20/07/2024:

Reading additional papers and waiting for Steve's meeting. Reading "Small Steps and Level Sets" by Chamin and will document.

Mon 29/07/2024:

Trying to do experiments to ensure I know the cause of my errors and that it's not some implementation error. Will do:

- small volume initialisations
- close to ground truth initialisations
- tightening the constraint slowly
- PCA?

Ideal semi-axes lengths is: 0.28209 Trials:

1. initialisation: 0.2, 0.4, 0.3, 30°, 25°, -120°. final: 0.271, 0.296, 0.28, 29.98°, 24.99°, 60.0°
2. initialisation: 0.1, 0.2, 0.3, 30°, 25°, -120°. final: 0.479, 0.29, 0.0913, 83.83°, 128.2°, 47.02° (no improvement).
the issue here is that the fitted ellipse is too far away from a sphere for it to approach a sphere?
3. initialisation: close to ideal. result: optimised fine and as expected.

Perhaps farthest point sampling would prevent the ellipse from over concentrating at the poles.

Issue! For Trial 2, a close reinitialisation of 0.1, 0.3, 0.5 and angles 30°, 25°, -120° worked well. In fact, an exact reinitialisation worked?! But then when reverting back so that the fitted ellipse had those params, the gradients calculated were zero.

Steps to reproduce error:

1. Initialise with 0.1, 0.2, 0.3 and 30°, 25°, -120°. Run the first cell and then second cell, and we should see no deformation of the fitted ellipsoid to a sphere - the parameters remain as 0.479, 0.29, 0.0913, 83.83°, 128.2°, 47.02°. The gradients are very close to 0.
2. Initialise with the parameters 0.479, 0.29, 0.0913 and 83.83°, 128.2°, 47.02°. We see that the gradients are non-zero and we optimise to a sphere.

a, b, c = 0.4786546028, 0.2901328092, 0.09132074619
yaw, pitch, roll = 83.82856741, 128.3492915, 47.0228292

Thu 15/08/2024:

Steve told me that I needed to update the DDN package and it worked!
I spent the past week a bit confused on how to do projection but Steve gave me some clear instructions. I've switched the volume and surface area constraints, but the optimisation is being worse.

Sun 18/08/2024:

Going to work on optimising the volume to be a certain target now and read a paper. Then, I'll write some things in the template. Efficiency!

Wed 4/09/2024:

For the past few days, I've been trying to mathematically formulate the projection. Just tested it in code today and it seems that I am minimising the area of the cross-section with the x - y plane instead. So, just need to take Schur complement instead of projection with matrix P .

One strange observation is that the gradients seem to increase in magnitude as time goes on; not decreasing.

3 Topic A to focus on

We can use sections organize the text into different parts such that specific topics are addressed in their own section.

TODOs

You will often have pending tasks that you need to track. This project logbook allows you to include both high and low priority todos that will be summarised in a list at the end of the file. Use the commands `\hightodo{<date_added>}{<author_key>}` and `\lowtodo`, including a date is recommended for tracking purposes.

Note: define your `userId` in the preamble

2016-05-23, : Try the code in the newest version of numpy compiled with optimized BLAS.

2016-05-27, u7013004: Perform convergence analysis with the latest version of the code.

4 Topic B to focus on

4.1 Algorithms

Sometimes the most straightforward way to explain a procedure is just to give it in an algorithmic format, it takes a little time but it will force you to go through the steps and you will most likely be able to reuse it on your paper. For the full documentation see here <https://texdoc.org/serve/algorithmicx/0>.

Algorithm 1 Euclid's algorithm

<pre> 1: procedure EUCLID(a, b) 2: $r \leftarrow a \bmod b$ 3: while $r \neq 0$ do 4: $a \leftarrow b$ 5: $b \leftarrow r$ 6: $r \leftarrow a \bmod b$ 7: return b </pre>	<p>▷ The g.c.d. of a and b</p> <p>▷ We have the answer if r is 0</p> <p>▷ The gcd is b</p>
--	--

4.2 Code

If the algorithm is too vague and you feel like you need the source code you can also insert it. You can put LaTeX code inside by using `<@ @>` delimiters, highlight small pieces of the code with `<| |>` delimiters, and highlight full lines (see the source code). For now only Python colored syntax highlighting is available, but most languages are supported and the color scheme can be added. More information available here https://www.overleaf.com/learn/latex/Code_listing.

Note that these pieces of code in the L^AT_EX document must start with no indent.

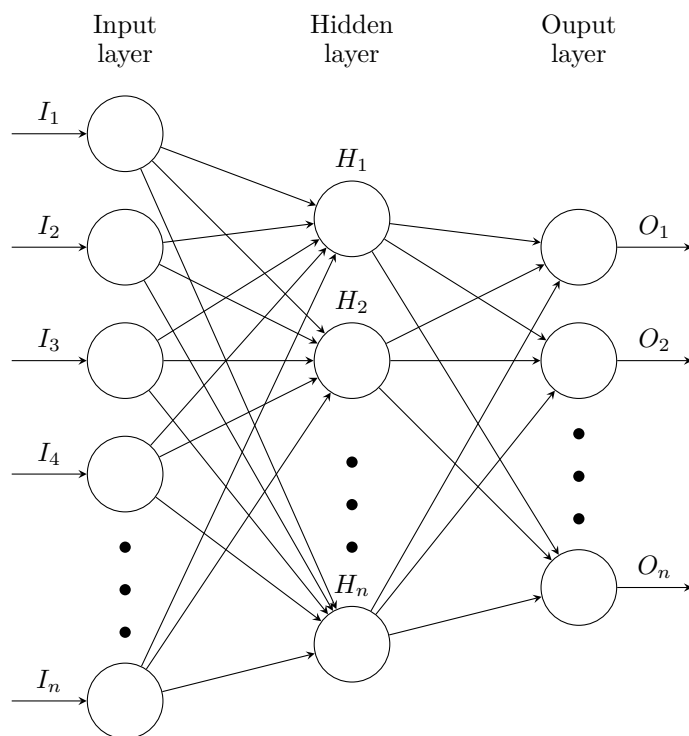
```

1 def DTW_distance(s1, s2):
2     """
3     Function to compute the Dynamic Time Warping in Python between two
4     signals
5     """
6     DTW={}
7     for i in range(len(s1)):
8         DTW[(i, -1)] = float('inf') # By default ∞
9     for i in range(len(s2)):
10        DTW[(-1, i)] = float('inf') # By default ∞
11    DTW[(-1, -1)] = 0
12
13    for i in range(len(s1)):
14        for j in range(len(s2)):
15            dist= (s1[i]-s2[j])**2
16            DTW[(i,j)] = dist + min(DTW[(i-1, j)],DTW[(i, j-1)], \
17                                   DTW[(i-1, j-1)])
18
19    return sqrt(DTW[len(s1)-1, len(s2)-1])

```


4.3 Diagrams

For simple diagrams I highly recommend learning TikZ, you will be drawing the diagrams in pure \LaTeX which has a steep learning curve but once you get used to it, it can be quite easy to display and do `for` loops to draw multiples line at once.



However, sometimes you will need more complicated diagrams (or maybe you do not like TikZ, in that case I recommend a vector drawing tool such as Inkscape which allows \LaTeX embedding)

5 Topic C to focus on

5.1 Figures

In general the best way to visualize your results will be some figures, I recommend Python's matplotlib for generating them or any other tool you are familiar with.

5.2 Tables

\LaTeX booktab environments are really good to showcase and track your results, however they can get fairly messy. My suggestion is to generate them via Python automatically and store the results in either a plain text file or a spreadsheet (there are packages to read spreadsheets with Python)

$\sigma \setminus \tau$	0	1	2	3	4	5	6	7	8
0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.6	98.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.8	84.7	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1.0	28.1	98.3	99.9	100.0	100.0	100.0	100.0	100.0	100.0
1.2	1.3	88.7	99.4	99.9	99.8	99.9	100.0	99.9	100.0
1.4	0.0	57.1	96.2	99.3	99.0	99.3	99.4	99.8	99.7
1.6	0.0	18.6	81.2	93.0	93.7	94.8	95.6	92.3	93.3
1.8	0.0	2.4	42.8	67.0	70.1	72.1	69.0	69.1	68.6
2.0	0.0	0.1	9.0	23.1	24.5	26.9	28.2	27.3	27.3
$t(\text{ms})$	27.92	40.23	77.30	157.27	252.05	342.18	381.46	399.85	413.72

Table 5.1: Performance of the algorithm for 128-bit key and with multiple readings per key

A References

Do not forget to cite the papers that you are using in your research, this way your work will be infinitely easier to write down and to review when the time comes.

B Resources

It is a good idea to record sources that explain concepts or provide tools so the research is both better documented and if someone has to continue it there is enough supporting documentation.

- Quick read in DTW and Keogh Lower Bounding.
<http://alexminnaar.com/time-series-classification-and-clustering-with-python.html>
<http://nbviewer.jupyter.org/github/alexminnaar/time-series-classification-and-clustering/blob/master/Time%20Series%20Classification%20and%20Clustering.ipynb>
- Parallelizing DTW – Good article on making a parallel version of DTW. Uses Keogh lower bound not as a linear approximation but as a pruning device.
<https://www.andrew.cmu.edu/user/mmohta/15418Project/finalreport.html>
- Deep Learning
 - Intro to LSTM
<https://colah.github.io/posts/2015-08-Understanding-LSTMs>
 - Intro to CNN
<https://colah.github.io/posts/2014-07-Conv-Nets-Modular/>
 - Why are LSTMs are so useful, impressive result in character pattern and syntax learning
<https://karpathy.github.io/2015/05/21/rnn-effectiveness/>

C TODOS

Here you will have all your TODOs grouped with anchor links to the parts of the document where they are. Really handy if you do not know where to continue with your project.

Todo list

- **2016-05-23**, : Try the code in the newest version of numpy compiled with optimized BLAS. 7
- **2016-05-27, u7013004**: Perform convergence analysis with the latest version of the code. 7