

Course Overview

ACTL3143 & ACTL5111 Deep Learning for Actuaries
Patrick Laub



Your lecturer



Dr Patrick Laub (LIC)

- Bachelor in Software Engineering / Mathematics (Uni. Queensland)
- PhD in Applied Probability (Denmark & Uni. Queensland)
- Post-doc in Lyon, France
- Post-doc in Melbourne, Australia
- Lecturer at UNSW since Jan. 2022

Course objectives

Artificial intelligence and *deep learning* for actuaries (in that order).

You will:

- understand common neural network architectures,
- create deep learning models (in Keras) to solve actuarial data science problems,
- gain experience with practical computational tools (e.g. Python).



Lecture plans

1. Artificial Intelligence & Python
2. Deep Learning with Tabular Data
3. Computer Vision
4. Natural Language Processing
5. Recurrent Neural Networks
6. *Away for flexibility week*
7. Distributional Regression
8. Interpretability
9. Generative Networks
10. Next Steps



Moodle & Ed Forum

The Moodle page contains:

- assessment (upload StoryWall, project & exam here),
- lecture recordings,
- link to lecture materials (<https://laub.au/ai>),
- link to Ed forum.

Ed forum will be used for announcements and for questions about the course.

If it is something confidential, then email me.



Lecture Outline

- **Learning activities**
- Assessment
- Past student feedback



Learning activities

The learning activities of this course involve the following (besides additional self-revision):

1. Self-study:

- Performing reading of relevant textbook chapters
- Doing lab questions (conceptual and applied)

2. Lectures:

- Preparing for & engaging in each week's lecture

3. Labs:

- Actively engaging in the lab sessions



Contact hours

The lectures are 2 hours each week.

The tutorials are a mix of practical coding and theoretical questions.

Make sure to use this time to ask your tutor for guidance on your project.

In later weeks, the tutorials will focus on project help and exam preparation.

Consultation hours will be online and scheduled weekly (see Moodle for Zoom link).



Exercises

On the website, I have added longer exercises for you to try.

Try to finish them around the week they are released (previously they were StoryWall questions).

These will be useful practice for the final exam.



Lecture Outline

- Learning activities
- **Assessment**
- Past student feedback



Course Grade Breakdown

1. StoryWall (30%)
2. Project (40%)
3. Exam (30%)



StoryWall

There are 7 StoryWall tasks, each worth 5% each.

The best 6 of 7 being counted, adding up to 30%.

These are formative assessments, so are marked pass/fail.

They are due on Friday at noon in Weeks 2, 3, 4, 5, 7, 9, 10.

I'll release them at least 10 days before the due date.



A complete deep learning project

Individual project over the term. You will:

- specify a *supervised learning* problem,
- collect and clean the data,
- perform an exploratory data analysis (EDA),
- create a simple (non-deep learning) benchmark model,
- fit two different deep learning architectures,
- perform hyperparameter tuning,
- write a discussion of the results.



Project components

The deliverables for the project will include:

1. Report Part 1 due at noon on Friday in Week 5 (10%),
2. Recorded presentation due at noon on Friday in Week 8 (15%),
3. Report Part 2 due at noon on Monday of Week 10 (15%).



Due dates

All due dates are at noon of the following weeks (“SW” = StoryWall):

- | | |
|--|--|
| 1. None | 6. None |
| 2. SW ₁ (Fri) | 7. SW ₅ (Fri) |
| 3. SW ₂ (Fri) | 8. Presentation (Fri) |
| 4. SW ₃ (Fri) | 9. SW ₆ (Fri) |
| 5. SW ₄ (Fri) and Report I (Fri) | 10. Report II (Mon) and SW ₇ (Fri) |



Late policy

If submitting late, you must apply for special considerations through UNSW central system. If you ask me for an extension, I will refer you to the special considerations system.

Without special consideration, **late StoryWalls will not be marked**. I have noticed that special considerations will not be granted for StoryWall tasks if you can still get full marks without that task.

For the project, the general policy is:

Late submission will incur a penalty of 5% per day or part thereof (including weekends) from the due date and time. An assessment will not be accepted after 5 days (120 hours) of the original deadline unless special consideration has been approved.



Example: Late policy for Report Part 2

Report Part 2 (worth 15% course grade) is due Week 10 Monday noon.

If you submit without special consideration on:

- Week 10 Monday 11:59 am, you have no late penalty.
- Week 10 Monday 12:01 pm, you have a **5% penalty**.
- Week 10 Tuesday 12:01 pm, you have a **10% penalty**.
- Week 10 Wednesday 12:01 pm, you have a **15% penalty**.
- Week 10 Thursday 12:01 pm, you have a **20% penalty**.
- Week 10 Friday 12:01 pm, you have a **25% penalty**.
- Week 10 Saturday 12:01 pm, you will **get 0 marks**.

E.g. a submission on Tuesday 12:01 pm (10% penalty) which was graded as 80/100, would be recorded as 72/100, and hence an overall course grade of 10.8% out of the maximum 15%.



Special case: Late policy for Report Part 1

However, as a special case **just for Project Report Part 1**, I will not apply the 5% per day penalty for the first 72 hours after the deadline.

Report Part 1 is due Week 5 Friday noon.

If you submit without special consideration on:

- Week 6 Monday 11:59 am, you have no late penalty.
- Week 6 Monday 12:01 pm, you have a **20% penalty**.
- Week 6 Tuesday 12:01 pm, you have a **25% penalty**.
- Week 6 Wednesday 12:01 pm, you will **get 0 marks**.



Final exam

The exam will test the concepts presented in the lectures.

The exam is a take-home format, and thus will be open book and open notes.

You'll be given a neural network task (similar to the exercises, shorter than the project), and will work **individually** to complete it.



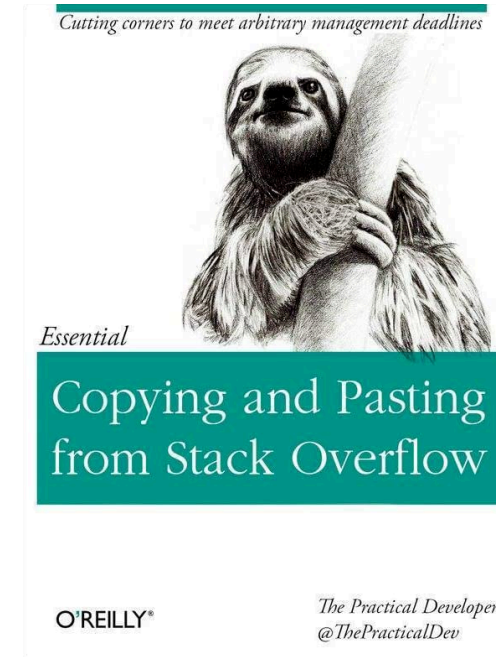
Copying code...

If you copy, tag it:

```
1 # Suppress endless warnings from Keras.
2 # Source: https://stackoverflow.com/a/38645250
3 import tensorflow as tf
4
5 tf.get_logger().setLevel("INFO")
```

Even if you then edit it a little:

```
1 # Create a basic Convolutional Network.
2 # Adapted from: https://www.tensorflow.org/tutorials/images/c
3 model = models.Sequential()
4 model.add(layers.Input((32, 32, 3)))
5 model.add(layers.Conv2D(32, (3, 3), activation="relu"))
6 model.add(layers.MaxPooling2D((2, 2)))
7 model.add(layers.Flatten())
8 model.add(layers.Dense(64, activation="relu"))
9 model.add(layers.Dense(10))
```



Recommended reading.



Source: Anonymous (2016), [Essential Copying and Pasting from Stack Overflow](#).

Plagiarism and ChatGPT

Plagiarism

Do not send or show your work to another student. You will be penalised along with them!

ChatGPT

You will add a “Generative AI usage” appendix to your reports, detailing how you used AI (what outputs, what prompts).

If you do not use AI, then you will still need the appendix that says that.



Sharing code is definitely cheating.

Source: DALL-E generated image.



Lecture Outline

- Learning activities
- Assessment
- **Past student feedback**



Best parts of the course? Project

“Having an open ended project allowed me to do something I had interest in.”

“The assignment was really good as it gave you ownership of what problem to choose and investigate. The course itself is super interesting and insightful, and really practical since time isn’t wasted on coding but rather the concepts and how to apply these models.”

“The course content was very interesting and provided a refreshing break from the endless calculations in other actuarial courses. It’s also great to learn how to build models in Python as an alternative to R.”



The style

“I think the hands-on approach to learning how to code up neural networks was quite useful. Not going too deeply into the inner workings of the neural network models (some ways this might happen would be deriving certian formulas and proofs of certian theorems) allowed us to devote more attention to learning how neural networks worked in an intuitive way. This perhaps is more helpful for a first timer to the subject area. I think that being too entrenched in the theorem-and-proof style of learning, though instilling rigor to the course as well as being entertaining to the geniuses and masochists, can make content too heavy and turgid for most. Having a lighter approach is more suitable to the needs of us actuarial students and allows us to gain a sufficient grasp of more types of neural networks.”



Comments put in the wrong form

Not about the course but about me:

“crazy amount of interest in field demonstrated by lecturer.
Amazing personality.”

“He carried some of that young energy with him.”



Improvements

What could be improved?

“More time for the project and removal of the final exam. I believe this project can reflect the real world even more if we had more time.”

There were others related to deadlines and number of StoryWall tasks, which were already implemented.

The next slides are answering the question: *Is there anything you wish you could have told your former self before starting to help them be prepared to learn these topics?*



Brushing up on ACTL3142

“In hindsight, a piece of advice I would give my past self before starting this course would be to have a solid grasp of introductory statistical learning. I believe reinstating such prerequisites would greatly enhance the ability for students to fully thrive in this course.”

We are adding this back again for next year.

“Something I wish I could have told my former self before taking this course, was to brush up on general Python skills, as well as, my understanding of non-deep learning models, such as those covered in ACTL3142.”



Being careful about project selection I

“If I had the chance to redo this course, I would add an additional project milestone of selecting the initial dataset and determining the problem statement. I believe the selection of the dataset is crucial since it broadly determines what models you can feasibly build, and how complex your model can be. I selected a almost purely numerical tabular dataset, which meant that when later concepts such as RNN and CNN were introduced, it was not feasible to implement those models since my dataset did not represent time-series or pictures. Only later in the modelling phase did I fully capture the importance of selecting a good dataset, but it was a lesson learned through the course.”



Being careful about project selection II

“Something I would have done differently for this course would be to choose an alternative supervised learning problem for my assignment, in particular, the dataset. Specifically, I would have preferred to explore something beyond the conventional 2D longitudinal dataset, such as working with images or unstructured text. This would have added a new dimension to my learning experience and allowed me to broaden my skill set even further.”



Time management

“If I could give advice to my former self it would be to start the project early and try to read ahead. Looking forward, I hope to continue to be able to learn more about deep learning and am glad to have been exposed to resources like Keras.”

“Something that I would have told my former self before starting is to not fall behind in the early stages of the term as the first couple weeks really builds the foundation for the remaining weeks of the term. Because I fell behind earlier, it was quite difficult for me to catch up during the rest of the term.”

“Moreover, I would still ask myself to start assignments earlier instead of leaving until the last minute.”



Planning

“If I could tell my former self something before starting, it would be to really focus on understanding the concepts first before jumping into coding the deep learning techniques. I think I could have saved a lot of time in the story walls and the assignment if I spent more time understanding the concepts in more depth instead of rushing into the coding aspect of things.”

“In terms of the assignment and coding I would have told my former self to do things properly the first time rather than doing a rough job and having to go back and fix it. Going back and altering early code can lead to really frustrating problems later so I would have told myself to not be lazy in the initial stages because it makes life a lot easier towards the end of the project.”



Patience

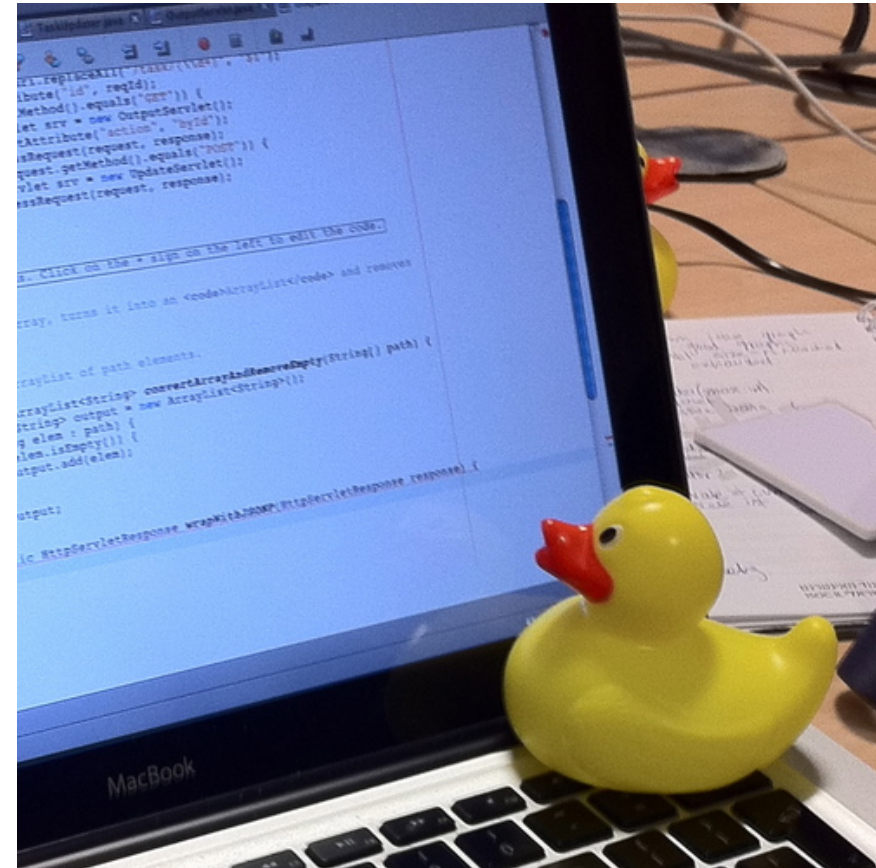
“If I could go back and advise my former self before starting the course, I would emphasise the importance of patience. Deep learning can be quite challenging, and it’s easy to get discouraged when facing complex algorithms and debugging errors. I would advise myself to just clear my mind then come back when this happens, rather than rack my brain for hours to no avail.”

“If I was give some advice to my former self, I would say that coding takes a long time to master and be prepared to put in the time and effort. There were some storywalls which I really struggled to understand the concept, resulting in many errors (some of which were just caused by Python behaviour I was not aware of) and many hours of debugging.”



Don't panic

“If I could go back to tell my former self (when I was struggling to work with the first ever StoryWall), I would tell myself that do not panic and read the lecture slides carefully. The slides are actually extremely helpful in tackling the StoryWall (and even assignment) questions.”



Get a rubber duck