Lecture 12: Exam Preparation

Week 7 · revision

Eoin O’Brien

This week

Dates and venues

	When	Where
Final exam	Tuesday 20 October, 09:00–11:00	MSc Teaching Space
Lab submission	Wednesday 21 October, 12:00	Brightspace and email
Project submission	Thursday 22 October, 12:00	Brightspace and email
Project interviews	Friday 23 October, your group’s slot	MSc Teaching Space

- Times are Irish time
- All four are on the landing page, each with a button that adds it to your calendar
- Each submission has its own email address and subject line, given on its slide

Lab submission

- Due Wednesday at 12:00
- **What**: the six graded notebooks, one from each of weeks 1 to 6
  - the ones marked with an asterisk in the landing page's Labs section
- **Where**: Brightspace and email, the same zip to both
  - to `eoin+advanced-ai-labs@eoin.ai`
  - subject `Advanced AI labs: Firstname Lastname, StudentID`
- **Name**: `Labs_Firstname_Lastname_StudentID.zip`
  - for example `Labs_Aoife_Murphy_24123456.zip`
  - a surname with a space in it takes a hyphen instead: `Mac-Giolla`
- Each notebook keeps the name Colab downloads it with
  - `7_2_Backpropagation.ipynb`, not `Copy of 7_2_Backpropagation.ipynb`
- Run every cell before you download it: a notebook is marked on what it printed

Lab submission: checking the zip

- Download `check_labs.py` from the landing page's Labs section
- Run it on your zip before you upload it:

```
python3 check_labs.py Labs_Aoife_Murphy_24123456.zip
```

- It checks three things and changes nothing
  - the zip's name
  - that it holds the six notebooks under their exact names and nothing else
  - that each one was saved after it was run
- Fix whatever it reports, then run it again

Project submission

- Due Thursday at 12:00, one submission per group
- **Where**: Brightspace and email
  - to `eoin+advanced-ai-project@eoin.ai`, copying every member of the group
  - subject `Advanced AI project: Team Name`
- **What**: the four parts the project specification lists
  - code and reproducibility material
  - written analysis
  - presentation material
  - contribution statement
- The project specification on the landing page has what each part must contain

Project interviews

- Friday, in the MSc Teaching Space, one slot per group
- **Presentation**: up to 8 minutes, as a group
- **Questions**: brief ones on the group's experiment, then about 6 minutes for each of you individually
- You may look at your submitted code, results and presentation
- During your individual questions: no web search, no AI assistants, no messaging

Booking your interview slot

- One slot per group, booked in my calendar:

https://calendar.app.google/4LFrFn15krFEMRSW8

- Pick a time every member of the group can make

The exam

Shape

Section	Questions	Time guide	Marks
Multiple choice	30, four options each	50 minutes	30
Written	3, in three parts each	60 minutes	30

Two hours in total, so about ten minutes are left for reading and checking.

Conditions

- Tuesday at 09:00, for two hours, in the MSc Teaching Space
- Typed, on Brightspace
- Closed book
- No calculators
  - no question needs more arithmetic than you can do on paper
- A sheet of rough-work paper is provided
- The integrity rules from week 1 apply as they did to every quiz

What is examinable

- Every session from weeks 1 to 6
- The assigned reading
- Each written question draws on more than one session
  - so revising by topic in isolation is the weaker strategy

How it was designed

What it tests

- Understanding, not recall
- Three kinds of question, in roughly equal measure
  - **Understand**: say what something is, or why it happens
  - **Apply**: use an idea in a case you haven't seen
  - **Analyse**: diagnose from evidence, compare two designs, evaluate a claim
- The multiple-choice half covers breadth
- The written half is where you justify

What it does not test

- ***No question hinges on remembering a formula.***
  - if a question needs a formula or a loss function, the question gives it to you
  - your job is to reason about it
- No question turns on a number, a name or a date from a slide
- No trick wording
  - no “all of the above”, no “which of these is NOT”
- No question about Python or PyTorch syntax

Why it is built this way

- **Four options, not five**: a fifth option is rarely a real choice and costs you reading time
- **No filler options**: every wrong option is a real misunderstanding, so guessing from the wording won't help
- **Options are short**: the thinking is in the question, not in parsing the answers
- **Three written questions**: more of the major is sampled than one essay would, and one bad topic costs less
- **Weight on the last part**: reasoning is what a typed answer can show and a multiple-choice option can't

What that means for revision

- For each idea, practise saying *why*, out loud or on paper
- Know what every symbol in a formula means and what happens when one changes
- Don’t memorise derivations
- Ask of each session: what problem did this solve, and what did it cost?
- The weekly quizzes show the level of the easier multiple-choice questions

Multiple-choice questions

Two kinds of stem

- **One line**: a direct question about what something is or why it happens
- **A short case**: a situation or a given formula, then a question about it
- Read the question and answer it in your head *before* you read the options
  - every question is written so that this works
- Then pick the option closest to your answer
- Exactly one option is the best answer

Sample: one line

Why does averaging the predictions of several separately trained models usually give lower error than a typical one of them?

A. The combined model has more capacity, so it fits the training data more closely.

B. The models’ errors partly differ, so some of them cancel in the average.

C. Each model corrects mistakes that the model trained before it had made.

D. The label noise each model had fitted is removed from the average.

Sample: one line, answer

- **B**: the models fail in partly different ways, so the average is steadier than its members
- **A**: treats the ensemble as one bigger model
  - the gain is in variance, not in a closer fit
- **C**: describes models trained in sequence on each other’s errors
  - these were trained separately
- **D**: noise in the targets can’t be averaged away by the predictor

Another one-line sample

With momentum, what is each gradient-descent step computed from?

A. The current gradient, scaled by a learning rate that grows over time.

B. The current gradient, divided by its recent typical magnitude.

C. The current gradient, corrected by the curvature of the loss there.

D. A weighted average of the current gradient and earlier ones.

Another one-line sample: answer

- **D**: directions that keep reversing cancel, and consistent ones build up
- **A**: the build-up comes from accumulated gradients, not from a growing step size
- **B**: that is the normalisation in an adaptive method such as Adam
- **C**: momentum helps in a narrow valley without computing any curvature

Sample: a given formula

For one example with true label $y \in \{0, 1\}$ and predicted probability $\lambda = \Pr(y = 1 \mid x)$, binary cross-entropy is

$$\ell = -y \ln \lambda - (1 - y) \ln(1 - \lambda).$$

Two models each get an example with true label 1 wrong: model A gives $\lambda = 0.4$ and model B gives $\lambda = 0.01$. Use $\ln 0.4 \approx -0.92$, $\ln 0.6 \approx -0.51$, $\ln 0.01 \approx -4.61$ and $\ln 0.99 \approx -0.01$.

Which statement correctly compares their losses on this example?

A. B’s loss is about 5 times A’s.

B. B’s loss is about 1.65 times A’s.

C. The two losses are about equal.

D. A’s loss is about 50 times B’s.

Sample: a given formula, answer

- The label is 1, so the second term vanishes and $\ell = -\ln \lambda$
- **A**: 4.61 against 0.92, a ratio of about 5
- **B**: assumes the loss grows in proportion to the miss
  - the logarithm makes a confident miss cost far more than its distance suggests
- **C**: reads the loss as the error rate, where both models are simply wrong
- **D**: keeps the wrong term, $-\ln(1 - \lambda)$

Sample: a short case

A residual block computes $h_k = h_{k-1} + f_k[h_{k-1}]$; a plain block computes $h_k = f_k[h_{k-1}]$. Backpropagation multiplies the gradient by each block's Jacobian $\partial h_k / \partial h_{k-1}$ in turn.

Early in training the branches f_k have last layers initialised close to zero, so each branch's own Jacobian J_k is close to zero.

Compared with the gradient at the output, what is the gradient with respect to h_1 like in a deep residual stack, and in a plain stack built from the same branches?

A. Residual: far smaller. Plain: far smaller.

B. Residual: roughly unchanged. Plain: far smaller.

C. Residual: roughly unchanged. Plain: roughly unchanged.

D. Residual: far larger. Plain: roughly unchanged.

Sample: a short case, answer

- A residual block's Jacobian is $I + J_k$, which is close to I here
- A plain block's Jacobian is J_k, which is close to zero
- **B**: a product of near-identities stays near the identity; a product of near-zeros vanishes
- **A**: leaves the skip out of the backward pass
- **C**: treats the gradient as independent of the architecture
- **D**: reads the chain rule as adding, when it multiplies

Written questions

Structure

- Each question opens with a **Given**: the formulas and facts you'll need
- Then three parts, climbing
  - **(a)**, 2 marks: explain something
  - **(b)**, 3 marks: apply it to a case
  - **(c)**, 5 marks: evaluate a claim or justify a choice
- The parts are independent: a slip in (a) can't cost you (b)
- No part needs a calculation
- About twenty minutes each; a few sentences per part is enough

How they are marked

- One mark for one claim, stated plainly
  - the question tells you how many marks, so it tells you how many claims
- Prose is fine: you never need to type an equation to earn a mark
- A claim you contradict later in the same part earns no mark
- If a part asks for one change and you give three, only the first is marked
- You need nothing beyond what the question asks for

Sample written question

Given: at initialisation, the variance of a ReLU layer's pre-activations is the previous layer's variance multiplied by a gain that depends on the scale of that layer's weights. Gradients are scaled the same way on the backward pass. The variances of uncorrelated quantities add. A residual block computes $h_k = h_{k-1} + f_k[h_{k-1}]$.

(a) [2 marks] Explain why a deep network is far more sensitive than a shallow one to the scale of its initial weights.

(b) [3 marks] A 30-layer plain ReLU network is initialised with weights that are too small. Describe what you would expect to see in the activations of its last layers, in the gradients reaching its first layers, and in the training loss over the first few epochs.

(c) [5 marks] A colleague chooses the weight scale so that each layer's gain is exactly one, then wraps every layer in a residual connection, keeping that scale. They claim the signal is still stable through depth "because each branch preserves variance".

Assume a block's input and its branch output are uncorrelated. Evaluate the claim (3 marks), then propose one change that restores stability and explain why it works (2 marks).

Model answer: (a) and (b)

- **(a)**
  - each layer multiplies the variance by its gain
  - the gains compound through depth, so a gain slightly off one is harmless over a few layers and extreme over many
- **(b)**
  - the last layers' activations shrink towards zero and look alike for every input
  - the gradients reaching the first layers are vanishingly small
  - the training loss stays near its starting value and barely falls
- Five marks, five claims, each a single sentence

Model answer: (c)

- Each branch does preserve variance, but a block outputs its input *plus* its branch
- The two variances add, so every block roughly doubles the variance
- That growth compounds through depth, so the signal is not stable and the claim is wrong
- One change: scale each block's output by $1/\sqrt{2}$ after the addition
- It works because halving the doubled variance returns it to the input's level

The marks for (c)

- **1**: a block's output is the sum of its input and its branch
- **1**: the variances add, so the output has more variance than the input
- **1**: the growth compounds through depth, so the claim is wrong
- **1**: one change that restores stability
  - rescaling after the addition, normalising within each block, and starting each branch near zero all earn it
- **1**: an explanation that matches the change you chose
- "The claim is wrong" with no reason earns nothing
- "Lower the learning rate" earns nothing: the problem exists before any step is taken

Writing an answer

- Lead with the claim, then give the reason
- One claim per sentence makes your marks easy to find
- Use the vocabulary of the Given; you don’t need to restate it
- If a part asks you to evaluate, say whether the claim holds *and* why
- Move on at twenty minutes: the first marks of the next question are cheaper than the last mark of this one

Questions?