MATH4267: Deep Learning and Artificial Intelligence
Assignment

2024-25

Contents

Overview

This assignment has two parts. In the first part, you will read and summarise a fairly recent paper in deep learning, from a pre-specified list. In the second part, you will fit a supervised machine learning model to a dataset, submit a set of predicted values for some unknown data.
This assignment is worth a total of **60 Marks**, with the following breakdown:

- 30 marks for the paper summary
- 15 marks for your description of what you have done in the prediction task; and
- 15 marks for how well you have managed to predict the unknown outcome

In total, you will submit:

- A report (in PDF format) of $\leq$ 7 pages covering the response to parts 1 (about 4 pages) and 2 (about 2 pages)
- An R data frame (in .RData format) with your predictions, as specified in part 2.

Please read this assignment through carefully (particularly the 'Hints and Notes' sections) before beginning.
This assignment is due at 5PM on 9 May 2025. Please submit reports and .RData objects to Ultra. **If you have any trouble with uploads, please just e-mail me your report and .RData object before the due date, and I will count it as on time.**

1 Part 1

Please choose **one** of the following papers on which to do the first part of the assignment. All these papers are from the 2023 International Conference on Machine Learning. **Do not worry if you do not fully understand the abstracts**.

i. Chen L, Bruna J. Beyond the edge of stability via two-step gradient updates. International Conference on Machine Learning 2023 Jul 3. PDF

> Gradient Descent (GD) is a powerful workhorse of modern machine learning thanks to its scalability and efficiency in high-dimensional spaces. Its ability to find local minimisers is only guaranteed for losses with Lipschitz gradients, where it can be seen as a 'bona-fide' discretisation of an underlying gradient flow. Yet, many ML setups involving overparametrised models do not fall into this problem class, which has motivated research beyond the so-called "Edge of Stability" (EoS), where the step-size crosses the admissibility threshold inversely proportional to the Lipschitz constant above. Perhaps surprisingly, GD has been empirically observed to still converge regardless of local instability and oscillatory behavior. The incipient theoretical analysis of this phenomena has mainly focused in the overparametrised regime, where the effect of choosing a large learning rate may be associated to a 'Sharpness-Minimisation' implicit regularisation within the manifold of minimisers, under appropriate asymptotic limits. In contrast, in this work we directly examine the conditions for such unstable convergence, focusing on simple, yet representative, learning problems, via analysis of two-step gradient updates. Specifically, we characterize a local condition involving third-order derivatives that guarantees existence and convergence to fixed points of the two-step updates, and leverage such property in a teacher-student setting, under population loss. Finally, starting from Matrix Factorization, we provide observations of period-2 orbit of GD in high-dimensional settings with intuition of its dynamics, along with exploration into more general settings.

ii. Zhang S, Lu J, Zhao H. On Enhancing Expressive Power via Compositions of Single Fixed-Size ReLU Network. International Conference on Machine Learning 2023 Jul 3. PDF

> This paper explores the expressive power of deep neural networks through the framework of function compositions. We demonstrate that the repeated compositions of a single fixed-size ReLU network exhibit surprising expressive power, despite the limited expressive capabilities of the individual network itself. Specifically, we prove by construction that $\mathscr{L}_2 \circ \mathbf{g}^{\circ r} \circ \mathscr{L}_1$ can approximate 1-Lipschitz continuous functions on $[0,1]^d$ with an error $O(r^{-1/d})$, where g is realized by a fixedsize ReLU network, $\mathscr{L}_1$ and $\mathscr{L}_1$ are two affine linear maps matching the dimensions, and $\mathbf{g}^{\circ r}$ denotes the r-times composition of g. Furthermore, we extend such a result to generic continuous functions on $[0,1]^d$ with the approximation error characterized by the modulus of continuity. Our results reveal that a continuous-depth network generated via a dynamical system has immense approximation power even if its dynamics function is time-independent and realized by a fixed-size ReLU network.

iii. Draxler F, Kühmichel L, Rousselot A, Müller J, Schnörr C, Köthe U. On the Convergence Rate of Gaussianization with Random Rotations. International Conference on Machine Learning 2023 Jul 3. PDF

> Gaussianization (Chen & Gopinath, 2000) is a simple generative model that can be trained without backpropagation. It has shown compelling performance on low dimensional data.

As the dimension increases, however, it has been observed that the convergence speed slows down. We show analytically that the number of required layers scales linearly with the dimension for Gaussian input. We argue that this is because the model is unable to capture dependencies between dimensions. Empirically, we find the same linear increase in cost for arbitrary input $p(x)$, but observe favorable scaling for some distributions. We explore potential speed-ups and formulate challenges for further research.

The following similar paper **will be used for example answers** to parts 4 and 3 below:

iv. Wang R, Manchester I. Direct parameterization of lipschitz-bounded deep networks. International Conference on Machine Learning 2023 Jul 3. PDF

This paper introduces a new parameterization of deep neural networks (both fully-connected and convolutional) with guaranteed ℓ^2 Lipschitz bounds, i.e. limited sensitivity to input perturbations. The Lipschitz guarantees are equivalent to the tightest-known bounds based on certification via a semidefinite program (SDP). We provide a "direct" parameterization, i.e., a smooth mapping from R N onto the set of weights satisfying the SDP-based bound. Moreover, our parameterization is complete, i.e. a neural network satisfies the SDP bound if and only if it can be represented via our parameterization. This enables training using standard gradient methods, without any inner approximation or computationally intensive tasks (e.g. projections or barrier terms) for the SDP constraint. The new parameterization can equivalently be thought of as either a new layer type (the sandwich layer), or a novel parameterization of standard feedforward networks with parameter sharing between neighbouring layers. A comprehensive set of experiments on image classification shows that sandwich layers outperform previous approaches on both empirical and certified robust accuracy. Code is available at `https://github.com/acfr/LBDN`.

1.1 Tasks

Once you have picked a manuscript, do the following tasks:

1. Summarise the *background* to the paper, in your own words, at a level interpretable to your classmates. Describe the setting to the paper, and what the paper contributes, in a general setting. Suggest avenues for further research following the conclusions of the paper. (**5 marks**)

2. Suggest a potential new real-world application of the method proposed in the paper. In particular, specify mathematically a quantify that could be increased or decreased by the method, and why this is a good thing. Describe in detail how the method proposed in the paper applies to the new application. (**5 marks**)

3. For the following theorem in the paper that you picked:

Paper i: Theorem 1

Paper ii: Theorem 1.1

Paper iii. Theorem 1

iv. **example only** Theorem 3.1

Find a use case. Specify objects of the type used in the theorem, specify specificially what the assumptions you mean in the context of the specific object, and indicate what is 'surprising' or 'new' about the theorem. You may include plots or use code if you like, but this is not mandatory. (**10 marks**)

4. For each of the following parts of the paper that you picked, specify *why* the given theorem/ proposition/ figure is of interest. You do not need to be as specific as in part 3. (**10 marks**)

Paper i. Proposition 2, Observation 1, Figure 3

Paper ii. Theorem 1.3, Figure 1, Figure 3

Paper iii. Equation 20, Theorem 2, Figure 4,

iv. (**example only**) Equation (9), Theorem 3.2, Figure 1, Table 2

In particular, try and say:

(a) What is being said?

(b) What is the relevance to the overall point of the paper?

(c) Why is what is being said useful to someone who wants to use or apply the overall method?

1.2 Hints and Notes for part 1

Please bear in mind

- This assignment is intended to get you moderately comfortable reading recent papers in machine learning.
- You will **probably come across mathematical ideas you haven't seen before**. Try and look things up online and teach yourself how they work. If you don't understand the ideas completely, please do your best with what you do understand.
- Many (even most) published papers have mistakes or omissions. Sometimes, key theorems do not actually hold, or key assumptions are missing. I do not think there are major mistakes in any of the main parts you have been asked to summarise, but there might be. Let me know if you find any.
- These papers are all recent and complicated. You will probably struggle to understand them entirely. Fear not! **You should not have to completely understand them to do a good job on this assignment**.
- Focus on the overall message of the paper, which will be summarised in the abstract. Also, look carefully at definitions, and work out specifically what the relevant theorems/figures/etc mean. It is often helpful to think about *why* definitions use the notation they use: why does the definition focus on a particular parameter or variable?
- All papers include long supplements, usually with extended proofs. **You do not need to read these to do this assignment** (unless you are particularly interested).
- Example answers can be found in the accompanying document.
- If you are struggling to find how to draw a particular symbol in LaTeX, then the 'detexify' website can help (link)
- These papers do not necessarily use the same notation as we have been using in class. Read the notation sections very carefully.

2 Part 2

In this part of the assignment, you will firstly perform a supervised learning task to predict the figure represented in a series of images, and secondly a generative task to produce similar images.
Please download the following datasets from Ultra:

- `xy_train.csv`. This is a `.csv` file, containing rows with comma-separated fields. Download it to a file (for instance, `/Downloads/xy_train.csv`). To open it in R, run the following commands:

  ```
  > PATH="/Downloads/xy_train.csv"
  > xy_train=read.csv(PATH,header=FALSE)
  ```

  replacing `/Downloads/xy_train.csv` with wherever you have saved your `.csv` file.

  The file `xy_train.csv` contains 500 rows and 785 columns. Rows 2-785 are intensity values (numbers between 0 and 255) representing a 28 x 28 pixel 2D image, and row 1 gives the label of the image: 'A', 'B', 'C' etc. The dataset does not have a header row.

- `x_text.csv`. This is also a `.csv` file. Download it and open it in R in the same way as for `xy_train.csv`:

  ```
  > PATH="/Downloads/x_test.csv"
  > x_test=read.csv(PATH,header=FALSE)
  ```

  The object `x_test.csv` is a table containing 10,000 rows and 785 columns. As for `xy_test.csv`, columns 2-785 define a 28 x 28 pixel image. The first column is all NA: this is what you are trying to predict.

The images depicted in the dataset are one of several (nonsense) characters:

$$(A): \otimes \qquad (B): \otimes \qquad (C): \mp \qquad (D): \mp \qquad (E): \odot \qquad (F): \odot$$

As an example, the first image in the training set can be viewed as follows:

2.1 Tasks

2.1.1 Task A

Your first goal is to formulate a decision rule which predicts the character depicted in the image. Part of your mark will be determined by the proportion of correctly-classified images in `x_text.csv`. Note that the labels for the images are hidden.
You will submit a completed data frame similar to `x_test.csv`, but with your predicted labels (A, B, or C) filled in in the first column. You may do this essentially however you like, but it should involve a neural network. I suggest you read the hints at the end of the assignment!

2.1.2 Task B

Your second goal is to generate 3000 new images, split evenly between the image types. The images should resemble those in `xy_test`. **This must be done with a neural network: either a generative adversarial network, a variational autoencoder, or a diffusion model**. You may also use the dataset `x_test` for training this generative model.

2.1.3 Task C

Summarise your approach to task A and B in a two-page report. This can be in the same document as your report for part 1, but the reports themselves should be separate.
Your report should cover the following:

- Your general strategy for these problems
- Any preliminary analysis
- Description and justification of any methods used
- Any protocols to prevent overfitting
- Any comparison of methods

In your report, focus on *why* you made particular decisions, and why a particular idea *could* have helped, even if it ultimately did not.

2.1.4 To submit:

Please submit the following

1. A CSV file called `predictions.csv` identical to `x_test.csv` but with the first column replaced with your predicted label (A, B, or C).
2. A CSV file called `new.csv` which has 3000 rows and 785 columns which contains your generated images. The first column is the label for the generated image in columns 2-785. The first column should contain 1000 A's, 1000 B's and 1000 C's.
3. A two page report, as above (which can be submitted either together or separately to your answer to part 1.

2.2 Hints and Notes for part 2

Please bear in mind the following:

- Most of the marks for this assignment will come from a sensible line of approach: you will get credit for attempting and describing sensible methods which do not turn out to 'work', and in general you will get more points for moderate predictions but attempting good ideas to improve performance than you will for very good prediction without attempting to improve performance. Technically you could do this assignment by manually inspecting and labelling all images, and by sketching 3000 new ones, but this will not earn many points!
- This dataset is simulated, and does not correspond to any real application. The figures drawn are deliberate nonsense, so you will not find a verbatim task online.
- Note that for the classification task the dataset is quite small. You may wish to make use of some of the following methods:

  1. Training and testing sets or cross-validation
  2. Dropout or other regularisation
  3. Convolutional neural networks
  4. Hyperparameter tuning
  5. Data augmentation
  6. Transfer learning

  There are a range of ways you could approach these tasks - there is no one best method.
- There is no need to submit code; it will not be marked. You may, however, include code snippets in your report if you like.
- You may use any programming language or tool you like for this task.
- The GitHub codespaces we use in tutorials have ample computational power for this task (and there will be little to be gained from applying computational power beyond this).
- Refer to the practical code and worksheets for analysis of the MNIST dataset in Keras, and the code for the optional section of assignment 3 code for an example of a VAE (which is adapted from here). Look at the tensorflow and keras manuals for further examples.
- To upload large files to GitHub codespaces, follow the instructions in the practical 3 worksheet.