Minimal Mastery: Efficient Discriminative and Generative Modelling

Deep Learning Summative Assignment

Michaelmas term, 2024

Information

Module code and title	COMP3547 Deep Learning
Academic year	2024-25
Coursework title	Minimal Mastery: Efficient Discriminative and Generative Modelling
Coursework credits	10 credits
% of module's final mark	100%
Lecturer	Amir Atapour-Abarghouei
Submission date*	Tuesday, 28 January 2025 at 14:00
Estimated hours of work	20 hours
Submission method	Gradescope
Files to submit	Report (PDF) and code (IPYNB)

* This is the deadline for all submissions except where an approved extension is in place.

Late submissions received within 5 working days of the deadline will be capped at 40%. Late submissions received later than 5 days after the deadline will receive a mark of 0. It is your responsibility to check that your submission has uploaded successfully and obtain a submission receipt.

Your work must be done by yourself (or your group, if there is an assigned groupwork component) and comply with the university rules about plagiarism and collusion. Students suspected of plagiarism, either of published or unpublished sources, including the work of other students, or of collusion will be dealt with according to university guidelines.

Introduction

This assignment is to design and train a classifier and a generative model on two different datasets. You will use the AddNIST dataset to train the classifier and the CIFAR-100 dataset to train the generative model. While this might seem straightforward, the challenge is that you are only allowed to train small models (with a limited number of total parameters) for a short amount of training time (gradient update steps).

You are also to write a short scientific report detailing both methods, experimental results, and limitations in a provided LaTeX template that closely follows parts of the ICLR conference style guidelines. These files must be zipped together and submitted to Gradescope like this, replacing the username with your CIS username:

```
username.zip
   username-paper.pdf
   username-generative-model.ipynb (or .py)
   username-classifer.ipynb (or .py)
```

You may **not** submit any images or model weights. You may **not** submit additional source code files.

To assist in this, the following template reports and starter code are provided to build on:

- [Deep Learning Paper Template] - *login with Durham email, on 'overleaf.com' click 'make a copy' to edit*
- [Google Colab Discriminative Model Starter Code]
- [Google Colab Generative Model Starter Code]

The deep discriminative model

Design and train a classifier on AddNIST. This dataset was created as part of a challenge to demonstrate the need for optimised neural network architectures. To learn more about the dataset, refer to:

- [the Paper]
- [the GitHub Repository]

When you have trained your classifier, you must report both the training and testing accuracy as in the above starter code. However, your model must have fewer than 100,000 (one hundred thousand) parameters (this is a low number of parameters for such a task). To display the total number of parameters for a deep neural network `N`, you can use something similar to the following code:

```
print(f'> Number of parameters {len(torch.nn.utils.parameters_to_vector(N.parameters()))}')
```

Furthermore, you may not train the neural network for more than 10,000 (ten thousand) optimisation steps. You must clearly state the total number of parameters and optimisation steps in both the code and report.

Exceeding either of these constraints will result in a 5 mark penalty for every 10% exceeded. For example, a model of 110,000 parameters incurs -5 marks whereas 150,000 parameters incurs -25 marks. A model trained for 11,000 steps incurs -5 marks and training for 15,000 steps incurs -25 marks. You must state in your report what penalty your implementation has incurred.

You should design the architecture yourself based on content covered in the lectures, practicals, and supplementary reading. If you reference existing code, this must be cited clearly in both the submitted python code and in the .pdf report.

As in the guidance paper template, your report should include: (i) a plot of the training and test accuracy over the length of your training, (ii) the total number of parameters in your network, (iii) the final values for training loss, training accuracy and test accuracy (means and standard deviations) as in the format provided to you in the discriminative model starter code.

You will be assessed primarily based on the quality of the report and the model accuracy. There is no specific target for the model accuracy and you should try to reach as high an accuracy as you can while remaining within the constraints of the problem. Further details of how this is graded are given in the marking scheme.

The deep generative model

Using the CIFAR-100 dataset, train a deep generative model to synthesise unique images that will be judged on their realism, diversity and uniqueness from the original training data.

Limitations in terms of model size and training length also apply to the generative modelling task. Your model must have fewer than 1,000,000 (one million) parameters. Note that if your approach consists of multiple networks (e.g., like a GAN), the parameter limit applies to the whole model with all the networks combined (e.g., the parameter count of the generator and the discriminator added together should be less that 1,000,000).

Furthermore, you may not train the neural network for more than 50,000 (fifty thousand) optimisation steps. Each optimisation step is counted when your entire model is trained by calculating the gradients for the whole model once. Again, if your approach consists of multiple networks (e.g., like a GAN), the optimisation step limit applies to the whole model with all the networks combined (e.g., if each step is training the generator and the discriminator separately, the steps training each of the generator and discriminator should be fewer that 50,000).

You must clearly state the total number of parameters and optimisation steps for the generative model of your choice in both the code and report.

Exceeding either of these constraints will result in a 5 mark penalty for every 10% exceeded, for example, a model of 1,100,000 parameters incurs -5 marks whereas 1,500,000 parameters incurs -25 marks. A generative model trained for 55,000 steps incurs -5 marks and training for 75,000 steps incurs -25 marks. You must state in your report what penalty your implementation has incurred.

In the report, you must display (i) a unique batch of 64 non cherry-picked model samples, (ii) interpolations between 8 pairs of your samples, and (iii) you must provide FID scores between 10k model samples and the 10k images in the CIFAR-100 test dataset. You are not permitted to train on the test data.

There is no specific target for the FID scores and you should try to reach the best scores you can while remaining within the constraints of the problem

You can use CIFAR-100 class labels to aid your training and sampling without penalty. You are permitted to train on a subset of the CIFAR-100 dataset, but you will not score as highly in diversity by doing so. Your model should generate samples based on a noise vector $\mathbf{z}$, drawn from a prior distribution, rather than being conditioned on $\mathbf{x}$. In other words, the samples should not be derived from a function of $\mathbf{x}$ during inference.

Paper submission

You must write a short paper up to a maximum of 4 pages using the provided LaTeX template, writing up the methodology, results, and limitations of your approaches alongside a short abstract. References do not count as part of the 4-page limit. The approach and the results of both the discriminative and generative models should be included in the same report. You will only submit one report.

The report should be written like an academic paper, with formal mathematical notation that should try to follow the ICLR guidelines (see the template for more information). Therefore, your discussions should be short, clear, and concise—*less is more*. Where appropriate, it is recommended to include a high-level architectural diagram in the paper to help explain your approach.

You can use any discriminative and generative model architecture that you like, and you can use any sampling strategy for your generative task. Your models, however, must use deep learning.

If you have deliberately incurred a penalty, please state at the end of the paper the total penalty you are expecting to receive. Your penalties should be calculated as summarised in Table 1.

Marking scheme

Essentially, you will be marked on the quality of the work, comprehension of the research field, presentation of the underpinning theory, quantitative results, the degree of rigour and the thoroughness of the scientific analysis—as if it were submitted and reviewed as a real conference paper.

The paper and submitted code will be marked as follows:

- **[40 marks]** Scientific quality and rigour of the paper and solution
  - Strategy and presentation of the underpinning theory for both the generative and discriminative tasks
  - Architectural design, function optimisation, sophistication, appropriateness and novelty
  - Clarity, simplicity and frugality of both the scientific writing and the implementation of both the generative and discriminative tasks
- **[30 marks]** Accuracy and generalisability of the discriminative model
  - How well does the model generalise to the test data? What is the test accuracy and gap between train and test accuracies? Does it have high variance?
  - How many optimisation steps are needed? Would the generalisation performance scale with more parameters and longer training times?
  - What other techniques did you use to enhance your method and boost your performance (e.g., regularisation, hyperparameter optimisation, inspecting the results via more advanced analysis, scientific approach, etc.)

- **[20 marks]** Realism of the samples from the generative model
  - Is the sampled batch of images blurry? What are the FID scores?
  - Do the image objects have realistic shapes and textures? Do they look real?
  - Do the interpolations look like linear alpha blendings or are all midpoints realistic?
- **[10 marks]** Diversity and uniqueness of the sampled batch of the generative model outputs
  - How different are the images from their nearest neighbours in the CIFAR-100 dataset?
  - How diverse are the samples within the batch of 64 provided?
  - Do the samples model the complete dataset, or only a subset of the dataset?
  - Do all sampled images look similar? Is there any mode collapse?

Exceeding parameter count limit (100,000) for the classifier - for each 10%	-5 marks
Exceeding optimisation step limit (10,000) for the classifier - for each 10%	-5 marks
Exceeding parameter count limit (1,000,000) for the generative model - for each 10%	-5 marks
Exceeding optimisation step limit (50,000) for the generative model - for each 10%	-5 marks
Manually edit (paint) any images or outputs	-100 marks
Use or modify someones code without referencing it	-100 marks
Use pre-trained weights from another model	-100 marks
Training either of your models on the test data	-100 marks
Every page over 4 pages in the paper (excluding references)	-10 marks

Table 1: Penalties stack, and affect the final mark. Therefore if you train your classifier for 200,000 steps, you will receive a penalty of -50 as you have exceeded the limit 10 times by 10%, so $10 * -5 = -50$.

NCC, PyTorch and JAX training

This assignment can be completed entirely using NCC as outlined in the first practical. We do not recommend Colab. Given the limitations in training time and model size, the computational resources needed for this assignment is minimal. If using NCC, please carefully read the documentation and respect other users on the job queuing system. Accessing NCC from outside of the university network requires using the VPN; you will need to fill in a VPN request from available from sharepoint. It is recommended to first get a simple model working—before exploring more advanced methods from state-of-the-art papers. You may use PyTorch or JAX, but no other DL libraries are permitted (such as Tensorflow, Keras, Flux, etc).

Feedback opportunities

There will be opportunities for feedback on your progress, writing, and solution during the practicals.

No redistribution policy and referencing code

Solutions for this assignment cannot later be publicised without request. If you would like to publish your solutions after marking, requests will be considered on a case-by-case basis. There is a lot of freedom with this assignment; where the licence permits, you can reference and extend existing code from GitHub (there must be citation in both report and code), if the imposed size and step limitations are followed.

Closing comment

I hope that you enjoy this coursework. If you are struggling, please ask questions where we can discuss any issues.