sensity

Interview Homework

In this task, we aim to build a command line application that, given an image of clothing in the format of the FashionMNIST dataset, can tell whether the image is real or it has been synthesised by a GAN. The code shall be delivered in a Github private repository, including instructions for installing dependencies and how to run it.

Tech stack required: python, pytorch. Other libraries can be used.

Estimated effort: 2 days.
Deadline: 7 days after the homework is sent

Delivery: create a private github repository and invite giorgio@sensity.ai and kieran@sensity.ai as collaborators

Your solution will be judged as:

- ⅖ correctness of the code solution and answers to the questions
- ⅕ Idiomatic, clean, consistent coding style
- ⅕ documentation, in particular installation and run instructions
- ⅕ use of git commits

Instructions

1. Train a Generative Adversarial Network, with architectures of your choice, for generating FashionMNIST data. The model must be conditional: at the end of training, you must be able to input the index of one of the FashionMNIST's classes that you wish to generate.
2. Question: Write down the loss function for the Generator and the Discriminator as a mathematical formula
3. You don't need to train it to full convergence, if it takes too long. Plot the loss curves for Generator and Discriminator, and one example of generated images for each class, paired with a corresponding example of the same class from FashionMNIST.
4. Generate 1k synthetic examples from the Generator and sample 1k real examples from FashionMNIST uniformly at random. Measure the accuracy of the Discriminator on this new dataset.
5. Question: how easy/hard is this task for the classifier? Why?
6. Finally, build a command line interface to the Discriminator, with this JSON format as output:

   {"label": {predicted_label}, "confidence": { model_probability }}

The application should return either ‘real’ or ‘generated’ in the “label” key, depending on the label output of the Discriminator on the provided image, as well as the softmax probability of the result under the “confidence” key. Let the application fail gracefully if any errors are encountered, e.g. in the case of unexpected input formats. Make sure the application is properly documented and provide an example on how to run it.