11/10/2023

CS 767 Draft Assignment 4

Jake Giguere <giguere@bu.edu>

Hand in a draft of Assignment 4, using the Word template supplied, with (only) the following sections drafted:

Two modifications in Part 1.1

## 1.1 Description of what you did and reasons this *could reasonably be* an improvement (up to two paragraphs, excluding figures and tables)

The T4 GPUs on google collab allow us to train these resource heavy models at a faster rate than a CPU. Increasing the number of epochs will definitely improve the model due to longer training.

Add early stopping to prevent overfitting/overtraining. Early stopping will stop the training episodes when there is no improvement in training. We compute the average epoch loss, and it is less than the best loss, then it becomes the best loss, else, we check to see the patience limit. If we reach the limit, then we save the best weights and stop training.

Another approach is using the Wasserstein loss function <https://en.wikipedia.org/wiki/Wasserstein_GAN#Gradient_penalty> for better gradient control.

Part 2.1

## 2.1 Give 2-4 requirements for a unique, highly functional application that you will implement with a GAN

## These describe *what* unique functionality your application will provide for the user, including the nature of inputs and outputs. This section should not include *how* you will design or code the application.

*ChatGPT prompt:* Describe requirements for a scientific visualization application implemented with a GAN. Provided examples of their inputs and outputs

*ChatGPT responses:*

**Realistic Scientific Visualizations**

* The application will generate high-quality visual representations of scientific phenomena (e.g., molecular structures, climate patterns, or astronomical events) based on raw scientific data inputs such as numerical datasets, time-series data, or 3D spatial data.
* Output: Visualizations in image or video format, such as molecular 3D renderings or weather heatmaps.

**Interactive Data Exploration**

* Users will be able to modify input parameters (e.g., time intervals, zoom levels, or environmental variables) to explore and dynamically generate new visualizations that reflect the changes in real time.
* Input: User-specified parameters like scale, density, or domain attributes.
* Output: Adapted visual outputs reflecting the changes, aiding in better understanding of the scientific data.

These two were the most fitting and achievable given the extensive list provided. Some others included **Synthetic Data Augmentation for Training AI Models** and **Cross-Domain Data Interpretation**.

Data links:  
**Protein Data Bank:**

<https://www.rcsb.org/>

**ChEMBL:**

<https://www.ebi.ac.uk/chembl/>