Reproduce results

Paul Lodder

January 28, 2021

1 Introduction

This file serves the purpose of documenting our code and providing the commands that produce the results we report in our paper.

2 Getting started

First of all, make sure to have followed the steps documented in README.org. Make sure that you have activated the virtualenvironment that you initialized to run any of our scripts.

3 General overview

To train and evaluate a model for any given dataset, we run

```
python scripts/train.py --dataset <dataset name>
```

where `<dataset name>` is one of `adult`, `german`, `yaleb`, `cifar-10`, `cifar-100`. The correct model for the given dataset is loaded by taking the (hyper)parameters specified in src/defaults.py. E.g. if we are working with `adult` data, the following variable is used to determine all parameters:

```
DEFAULTS_ADULT = {
    "z_dim": 2,
    "max_epochs": 2,
    "batch_size": 64,
    "lambda_od": 0.037,
    "gamma_od": 0.8,
    "encoder_hidden_dims": [64],
    "lambda_entropy": 0.55,
    "gamma_entropy": 1.66,
    "input_dim": 108,
    "target_output_dim": 1,
    "sens_output_dim": 1,
    "target_disc_hidden_dims": [64, 64],
    "sens_disc_hidden_dims": [64, 64],
    "target_disc_batch_norm": False,
    "predictor_epochs": 10,
    "encoder_lr": 1e-3,
    "encoder_weight_decay": 5e-4,
    "discs_lr": 1e-3,
    "discs_weight_decay": 5e-4,
```

```
    "loss_components": "entropy,kl,orth",
    "step_size": 30,
}
```

Most of these parameters can also be changed by explicitly providing a value as a command line argument:

```
python scripts/train.py --dataset <dataset name> --step_size 100
```

The loaded configuration is passed to functions defined in src/iniitializers.py, like `get_fodvae(config)` which returns the correct model by looking at the given config and associated hyperparameters.

4 Reproducing our results

Now that you a main idea of our code, let us talk you through reproducing the specific results that we present in our paper.

4.1 Table 2

This table presents the target and sensitive accuracy achieved on the Cifar-10 and Cifar-100 dataset. We produced these results by running:

```
python scripts/run_cifar.py
```

This script trains the correct models for Cifar-10 and Cifar-100 for 3 different random seeds, and pickles the results stored in a pickled `pd.DataFrame` in the `RESULTS_DIR` defined in your `.env`.

4.2 Figure 1

Figure 1 shows multiple bar plots that compare Sarhan et al's (2020) sensitive and target accuracies achieved on the raw data, embeddings from a regular VAE, and the target representations produced by the proposed model. They performed this experiment only on the Adult, German, and YaleB dataset. To produce all plots for this figure, we ran:

```
bash scripts/train_and_plot_target_and_sens_accs.sh
```

This script runs the following script for each of the three datasets:

```
python scripts/make_fig2.py --dataset <dataset-name>
```

It saves two figures per dataset in `FIGURES_DIR` (as specified in your `.env`): `<dataset-name>_sens.png` and `<dataset-name>_target.png` which show the sensitive and target accuracies, respectively.