Predicting sampling advantage of stochastic Ising Machines for Quantum Simulations

This repository contains the code for the paper "Predicting sampling advantage of stochastic Ising Machines for Quantum Simulations".

The code is organized as follows:

- Code for obtaining the exchange matrix and bias vector of the Ising model from the weights of a Restricted Boltzmann Machine (RBM) trained using the ULTRAFAST code.
- Example of the chromatic Gibbs sampling code, which is used to sample from the Ising model. The code can be found in the `ultrafast_full_boltzmann_machine/julia/gibbs/gibbs` folder.

The rest of this document describes how to install/setup the environment and code, and how to run the code.

Install

Python

The first step is to create a virtual environment. You can do this with `venv` or `conda`. For example, using `venv`:

```
python3 -m venv .venv
source .venv/bin/activate
```

Then, install the requirements:

```
pip install -r requirements.txt
```

This code should be compatible with at least Python version 3.11.5.

For further information, https://www.python.org

Julia

In order to install the proper Julia environment, please run the following command in the root of the project:

```
julia --project=.
```

This code should be compatible with at least Julia version 1.11.3.

For further information, https://julialang.org

Available commands through Make

Run `make` in this directory.

- **requirements**: Install Python Dependencies
- **clean**: Delete all compiled Python files

The ULTRAFAST code is used to train the models representing the ground state of the 2D Heisenberg model. The ULTRAFAST code only stores the weights of the RBM, which are independent due to translation invariance. The code in this repository converts these independent weights to the full weights (**convert_weights**), which are then used to compute the exchange matrix and bias vector of the Ising model (**isingweights**).

- **convert_weights**: Convert the independent weights to full weights, writes them to the `data` folder under the names `W_RBM_{nspins}_{alpha}_ti_W.csv` and `W_RBM_{nspins}_{alpha}_ti_b.csv`.
- **isingweights**: Promote weights to exchange matrix and bias vector of Ising model, writes them to the `data` folder under the names `Ising_{nspins}_{alpha}_ti_J.csv`, `Ising_{nspins}_{alpha}_ti_h.csv`, `W_ising_{nspins}_{alpha}_ti_W.csv`, `W_ising_{nspins}_{alpha}_ti_b.csv`,

Note that the files starting with `Ising` uses the following convention:

$$H_{\text{Ising}}(m) = -\sum_{i=1}^{an+n} h_i m_i - \sum_{j=1}^{an+n}\sum_{i=1}^{j-1} J_{ij} m_i m_j$$

whereas the files starting with `W_ising` uses the following convention:

$$H_{\text{Ising}}(m) = -\sum_{i=1}^{an+n} b_i m_i - \sum_{j=1}^{an+n}\sum_{i=1}^{an+n} W_{ij} m_i m_j$$

The files starting with `W_RBM` represent the non-zero part of the exchange matrix and bias vector of the Ising model, which are given by:

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & (\tilde{\mathbf{W}}^T)_{1\ldots an \times 1\ldots n} \\ \tilde{\mathbf{W}}_{1\ldots n \times 1\ldots an} & \mathbf{0} \end{bmatrix}$$

and

$$\mathbf{h} = [\tilde{\mathbf{b}}_{1\ldots an}\, \mathbf{0}_{1\ldots n}].$$

Usage of Chromatic Gibbs Sampling code

The chromatic Gibbs sampling code is provided in the `ultrafast_full_boltzmann_machine/julia/gibbs/gibbs/convenience_chromatic_gibbs.jl` folder where the function `chromatic_rbm_sampler` is defined. The settings and arguments are described below. Note that to obtain the required W and b, `make convert_weights` should be executed first.

```
chromatic_rbm_sampler(nspins, W, b, steps, precision, sampling_settings)
```

This function performs chromatic Gibbs sampling:

Arguments

- `nspins`: The number of visible spins in the system.
- `W`: Part of weight matrix for the Ising model representing RBM, corresponds to files `W_RBM_{nspins}_{alpha}_ti_W`
- `b`: Part of bias vector for the Ising model representing RBM, corresponds to files `W_RBM_{nspins}_{alpha}_ti_b`
- `steps`: The number of steps to perform in each sweep.
- `precision`: The precision of the calculations. Default is Float32.
- `sampling_settings`: A named tuple containing the settings for the sampling.
  - `all_up`: If true, the initial state is all spins up. Default is false.
  - `mag0`: If true, the hidden spins are not sampled. Default is false.
  - `thermalization`: The number of thermalization steps. Default is 0.
  - `sweeps`: The number of sweeps to perform. Default is 1.
  - `save_hidden`: If true, the hidden spins are saved. Default is false.

Returns

- `elapsed_time_sampling`: The elapsed time for the Gibbs sampling.
- `states`: The sampled states. A named tuple containing the sampled visible spins, (optionally, only mag0) the hidden spins, (optionally, only mag0) the actual number of thermalization and sampling steps, elapsed_time_sampling

Project Organization

```
.
├── Makefile                                <- Makefile with commands
├── Manifest.toml
├── Project.toml
├── README.md
├── data                                    <- Data directory in which the exchange matrix and bias vector of the Isin
├── env.yml                                 <- Environment file for reproducing the analysis environment
├── models
|   ├── all_weights_converted.tsv           <- List of all weights that should be converted to Ising model
|   ├── computed_configurations
|   |   └── converted
|   ├── modified_RBM_high_model_id=0        <- All the pre-trained models of the modified RBM, obtained with ULTRAFAST
|   ├── modified_RBM_high_model_id=1        <- divided into 5 models and two classes of hyperparameters (high and low)
|   ├── modified_RBM_high_model_id=2        <- ""
|   ├── modified_RBM_high_model_id=3        <- ""
|   ├── modified_RBM_high_model_id=4        <- ""
|   ├── modified_RBM_low_model_id=0         <- ""
|   ├── modified_RBM_low_model_id=1         <- ""
|   ├── modified_RBM_low_model_id=2         <- ""
|   ├── modified_RBM_low_model_id=3         <- ""
|   ├── modified_RBM_low_model_id=4         <- ""
|   ├── standard_RBM_high_model_id=0        <- All the pre-trained models of the RBM, obtained with ULTRAFAST
|   └── standard_RBM_low_model_id=0
├── pyproject.toml
├── requirements-apr24-Python-3.10.12.txt   <- Requirements file for reproducing the analysis environment
├── requirements.txt                        <- Requirements file for reproducing the analysis environment
├── setup.cfg
└── ultrafast_full_boltzmann_machine
    ├── __init__.py
    ├── isingweights.py                     <- Converts the weights of the RBM to the exchange matrix and bias vector
    ├── config.py
    ├── julia
    |   ├── gibbs_sampling.jl               <- Contains the chromatic Gibbs sampling code
    |   ├── translation_invariance
    |   └── weightconverter.jl              <- Converts independent weights (due to translation invariance) to full.
    ├── models
    |   └── models.py                       <- Defines all the models and their parameters
    └── weight_conversion
        └── weight_promotor.py
```