

Second project review

TopoHyperDrive: Accelerating Meta-Search in Hyperparameter Optimization through Topological Analysis
Sergey Kastrulyin

An overview

The goal of the project is to develop a topology-based method for optimal metaparameters search to surpass baselines on CIFAR-100 classification task. The method is based on the intuition of importance of diversity of image embeddings and directly enforces it to maximize evaluation metrics.

Project strengths

The project proposes an interesting and potentially useful topology-based method. The method is proposed to address the pressing problem of difficulty of hyperparameter search for large multimodal models. The authors provide implementation of their method, which is well presented and documented.

Project weaknesses

Major:

- The method (S.3.2) states that large distinction of models' embeddings automatically results in "numerous topological features". It is a very generic and, at the same time, arguable statement, provided without any concrete confirmation. Please clarify the point since the whole proposed metric is based on this intuition;
- The paper title and main proposal are misleading, regarding the term hyperparameters. The term hyperparameter is typically referred to parameters related to the training process such as learning rate, batch size, the choice and parameters of optimizer [1]. This is in contrast to parameters which determine the model itself. Please stick to the common terminology or explicitly state what is meant by the term in later paper revisions.
- Authors use Figure 2 to justify experimental, i.e., to state the the proposed method surpasses the baselines according to classification accuracy on the CIFAR-100 dataset. However, the methodology of the comparison is rather questionable. For instance it is not clear:
 - Why best validation accuracy is selected over the mean or medial accuracy over a number of runs;
 - What is the variance of values across several runs of each method. Even minimal variance in end results would lead to the lack of statistical difference between different methods considering that the gap between the RandomSearch baseline and the best-performin version of the proposed metric is about 0.005 on validation and RandomSeach even surpasses the method on test data;
- The flow of the introduction section is not clear. For example, it is not obvious how generic embeddings produced by modern foundation models are related to the proposed method and the following contributions;

- Related work section relies on a relatively outdated prior work. For instance, the Black-Box Optimization subsection does not mention any papers newer than 6 years old;

Minor:

- Small typos e.g., the lack of space in "... *real-world applications. However, as these*" (Intro) mismatched grammar e.g., "*The contribution of this paper is as follows:*" leading to several contributions, which are themselves lack subject (from the grammatical point of view);
- It is general preferred to avoid passive voice in academic writing. I would recommend to re-formulate at least critical parts, for example paper contributions;

On code and completed experiments

Overall, the codebase is well written and documented. All modules that I encountered contained docstrings and inline comments that help to understand implementation details. The repository is well structured, naming and module organisation is rather conventional and intuitive.

I was able to install all required packages using Conda and pip as described in the Readme. However I had problems running the code on two independent servers.

I had the following problems trying to install RTD.

`pip install git+https://github.com/simonzhang00/ripser-plusplus.git`
results in the following:

```
make[1]: Leaving directory
'/home/s-kastryulin/dev/TopolyperDrive/pip-req-build-vtthcSog/build/temp.linux-x86_64-cpython-311/MakeFiles/CMaketmp'

Makefile:126: recipe for target 'cmTC_lab5a/fast' failed
make: *** [cmTC_lab5a/fast] Error 2

[20:45:11]

  OMakelists.txt:2 (project)

-- Configuring incomplete, errors occurred!
See also "/home/s-kastryulin/dev/TopolyperDrive/pip-req-build-vtthcSog/build/temp.linux-x86_64-cpython-311/OMakeFiles/OMakelists.txt".
See also "/home/s-kastryulin/dev/TopolyperDrive/pip-req-build-vtthcSog/build/temp.linux-x86_64-cpython-311/OMakeFiles/OMakefile".
error: command '"/usr/bin/cmake' failed with exit code 1
[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.
ERROR: Failed building wheel for ripserplusplus
Running setup.py clean for ripserplusplus
running 'clean'
running 'build'
running 'build_ext' for ripserplusplus, which is classified as an optional dependency.
[studs) s-kastryulin@zomb-research-cv-1:~/dev/TopolyperDrive$ pip install git+https://github.com/IlyaTrefimov/RTD.git
```

I was also not able to run the default configuration
`python src/train.py trainer=gpu`
results in the following:

```
/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/grammar_visitor.py", line 125, in visitInterpolation
    return self.visit(ctx.getChild())
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/antlr4/tree/Tree.py", line 34, in visit
    return tree.accept(self)
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/gen/OmegaConfGrammarParser.py", line 921, in doInterpolation
    return self.visitInterpolationNodeSet()
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/grammar_visitor.py", line 158, in visitInterpolationNo
    return self._node_interpolation_callback(inter_key, self.memo)
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/gen/OmegaConfGrammarParser.py", line 745, in node_interpolation_callback
    return self._resolve_node_interpolation(inter_key=inter_key, memo=memo)
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/base.py", line 671, in _resolve_node_interpolation
    raise InterpolationNotFoundMissingValueError()
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/base.py", line 664, in _resolve_node_interpolation
    parent, last_key, value = self._get_node_value(inter_key, self.memo)
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/base.py", line 514, in _select_impl
    value, _ = self._select_one()
           ^^^^^^^^^^
File "/home/s-kastryulin/miniconda3/envs/studs/lib/python3.11/site-packages/omegconf/omegconf.py", line 1158, in _select_one
    raise MissingMandatoryValue(
omegconf.errors.InterpolationNotFoundMissingValueError: MissingMandatoryValue while resolving interpolation: Missing mandatory value: task_name
for file: /tmp/run.dir
object_type_id:ct
(studs) s-kastryulin@zomb-research-cv-1:~/dev/TopolyperDrive$
```

Similar outcome is observed running the `random_search_optuna` version:
`python src/train.py -m trainer=gpu hparams_search=random_search_optuna`
results in the following:

References

- [1] [https://en.wikipedia.org/wiki/Hyperparameter_\(machine_learning\)](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning))