METL – TP4
Neural Networks and Implementation
Word2Vec & Tensorflow

Alexandre Kabbach
alexandre.kabbach@unige.ch

Paola Merlo
paola.merlo@unige.ch

02.04.2020

Evaluation: You are allowed an unlimited number of submissions in order to receive feedback. When you are satisfied with your work, you can ask for it to be graded. You can also ask for it to be graded upon a single submission, without receiving feedback. All your TPs must have been graded and must have received an average grade of at least 4/6 for you to register for the METL exam. Indicative deadline: April 15 2020 (this TP should take you two weeks).

1 Introduction

The purpose of this TP is to practice using and understanding someone else's machine learning code. We will work on a re-implementation of Word2Vec based on Tensorflow and the Estimators API available here: https://github.com/akb89/word2vec

When answering the below questions, try to be as exhaustive as possible. Use diagrams, screenshots and code samples whenever necessary.

Here are some potentially useful references:

1. on Word2Vec itself:
   (a) https://arxiv.org/abs/1301.3781
   (b) https://arxiv.org/abs/1310.4546
   (c) https://arxiv.org/abs/1402.3722
   (d) http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
   (e) http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/
2. on Tensorflow Estimators and Datasets APIs:
   (a) https://www.tensorflow.org/guide/estimators
   (b) https://www.tensorflow.org/guide/datasets
   (c) https://www.tensorflow.org/guide/datasets_for_estimators
   (d) https://www.tensorflow.org/guide/performance/datasets

Note: if you discover a bug, please file an issue on Github here: https://github.com/akb89/word2vec/issues

2 Packaging and development

1. How would you install the `word2vec` package in development mode to ensure no inteference between `word2vec` and your other python package dependencies? Which command would ensure the package to be updated dynamically upon modifications?
2. What is the purpose of the `.travis.yml` file? Explain and detail its content: what does it do and why?

3 Running

1. Specify the minimal command to train `word2vec` on the given sample of the English Wikipedia
2. Detail all the train method parameters: what do they correspond to?
3. Specify the default values of those parameters? Would you recommand changing any of those default values and if so why?

4 Architecture

1. Detail the overall architecture of the program. Use diagrams
2. What are the benefits of this architecture compared to the original Tensorflow-based Word2Vec implementation available here: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/word2vec/word2vec_basic.py

5 Monitoring

Launch Tensorboard locally using the following command:

1. Detail each category appearing under the `SCALARS` menu upon training Word2Vec (you should see 3 of them)
2. Explain the regular performance drops visible under the `global_step` tab
3. Using the `GRAPHS` menu, determine which computation step(s) take most of the compute and memory time. Justify using screenshots
4. Under a parallel architecture using both CPUs and GPUs, what kind of information would be interesting to have to identify performance bottlenecks of the application. Which parameters would this impact?

6 Implementation

1. Which step is missing in preprocessing compared to the original `word2vec` code?

2. How is negative sampling implemented in the code? How does it differ from the original `word2vec` implementation (see the original paper). Propose a fix to best align with the original paper
3. What is the purpose of having a higher order function (`def` of `def`) in the `concat_mean_to_avg_tensor` function of the `models/word2vec.py` file? Explain the function: how does it relate to the `word2vec` model

7 Testing

1. How would you measure test coverage on the `word2vec` package? Propose a method and output the current code coverage. Does it seem satisfacory to you? Justify
2. Why the need for the `_CBOW_!MASK` values under the `test_avg_ctx_features_embeddings` of the `tests/models_w2v_test.py` file

8 Debugging

1. Specify the command to launch `word2vec` in debug mode
2. What is the use of the Tensorboard debugger?
3. Run the debugger for a single step of loss computation. Explain the columns in Tensor Value Overview. Use screenshots if necessary