METL TP4

Tientso Ning

Packaging and Development

1. I used `pip3 install tf-word2vec` command to my terminal, following the instructions given on the GitHub source-code page. We could additionally add the `-e` flag to the `pip3 install` command to ensure the package can be updated dynamically upon modification, as this command flag is used for development mode. Regarding avoiding conflicts, we could use a virtual environment.
2. The `.yml` extension refers to the file type YAML, standing for "YAML ain't Markdown Language". It is a type of data-serialization language. Essentially, it is used for configuration files and is easier to read for humans. This specific file attempts to detect the correct python version in order to install the correct version, and run some tests to make sure that the intended version was installed.

Running

1. The minimal command to train is `w2v train --data ./path_to_txt --outputdir ./output_path`
2. The training parameters are as follows:

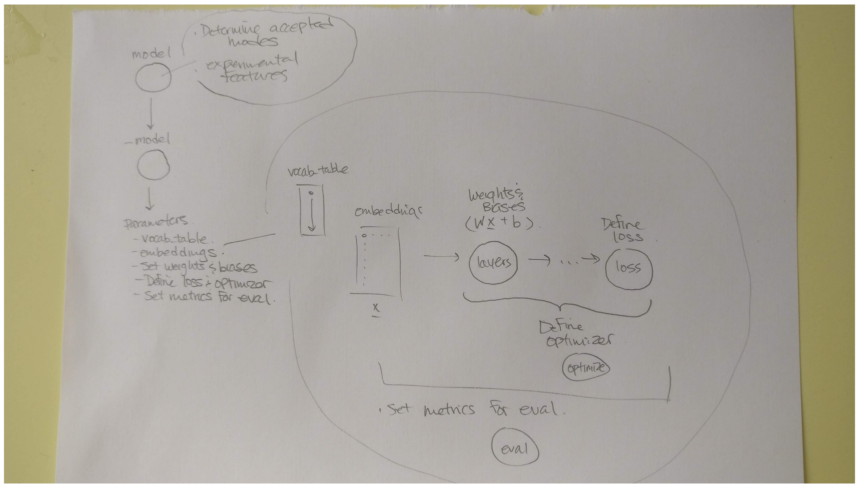
- data: this specifies the data file that we need as input to train on (in our case, the wikipedia as txt file)
- outputdir: this specifies the output location where the trained model is stored
- alpha: the initial learning rate (we tune this to have it learn faster or slower)
- neg: the number of negative samples
- window: the window size
- epochs: the number of epochs (for training)
- size: the dimension of the vector
- min-count: the minimum frequency count
- sample: the subsampling rate (for training)
- train-mode: the type of training setup (i.e: we can do skipgram as shown in the default, or we can use cbow)
- t-num-threads: number of threads used for training (CPU stuff)
- p-num-threads: number of threads used for pre-processing (CPU stuff)
- keep-checkpoint-max: when training, there are certain checkpoints. This determines how many is kept during training (meaning if 0 or None, then there is no max and all are kept).

- batch: the batch size (for training)
- shuffling-buffer-size: the size of the buffer used for shuffling training data (for training)
- save-summary-steps: the number of steps after which summaries are saved.
- save-checkpoints-steps: the number of steps after which checkpoints are saved.
- log-step-count-steps: the number of global steps after which loss will be logged.

3. The default for each of the values are as follows:

- alpha: 0.025
- neg: 5
- window: 2
- epochs: 5
- size: 300
- min-count: 50
- sample: 1e-5
- train-mode: skipgram
- t-num-threads: 1
- p-num-threads: 1
- keep-checkpoint-max: 3
- batch: 1
- shuffling-buffer-size: 10000
- save-summary-steps: 100000
- save-checkpoints-steps: 1000000
- log-step-count-steps: 100000 It could be recommended to alter the number of threads used if your device can handle multiple threads (making training faster). Additionally, altering the batch size could also assist in efficiency.

Architecture

1. The architecture as follows: 

2. The benefits of this architecture compared to the original tensorflow based Word2Vec implementation is that this implementation utilizes the TF estimator APIs. This high-level model abstraction allows you to run models locally using CPU/GPU without rewriting the models, and gives you a training loop that allows you to control when to load data, handle errors, create checkpoints, and have summaries easier.

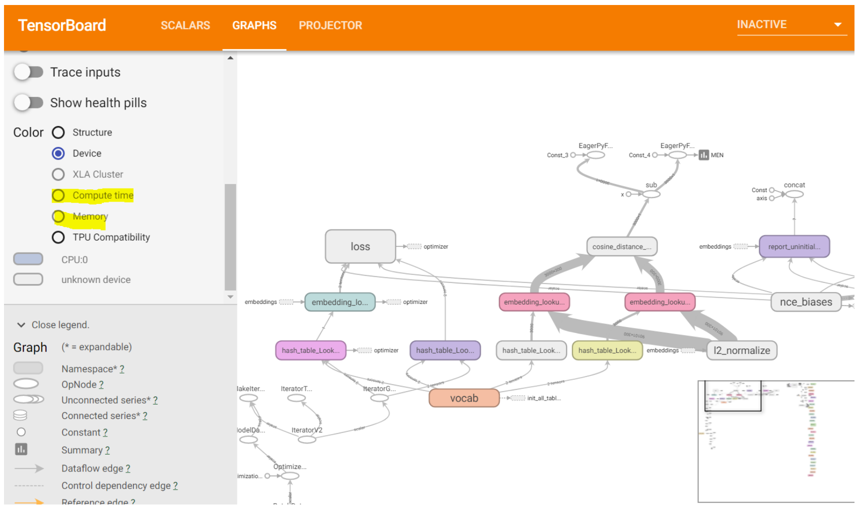
Monitoring

1. The three categories under SCALAR tab is:

- MEN: The dataset
- global_step: The number of global steps done per second.
- loss_1: This informs us of the loss of the model.

2. The regular performance drops under the global_step tab is most likely due to . . .

3. The computation that takes the most CPU and Memory time is unclear due to the options not being available on my device (Please see attached image).



Trace inputs
Show health pills
Color ( ) Structure [✓ Device] ( ) XLA Cluster ( ) Compute time ( ) Memory ( ) TPU Compatibility
CPU:0
unknown device

Theoretically, however, the bottleneck should be application of the softmax on the last layer to get the output since the vocabulary sizes should be big (as explored in TP2).

4. If we used a parallel architecture, we would want to visualize the GPU and CPU thread usage. The parameters affected would be the GPU and CPU thread count respectively.

Implementation

1. The step missing compared to the pre-processing done in the original word2vec is the replacement of rare words with the UNK token.
2. Negative sampling size is passed as a parameter to the loss function with the call to `nce_loss`. This differs from the original implementation, which uses subsampling on the same loss function call. A fix could be ...
3. The purpose of the higher order function is ...

Testing

Debugging

1. The command to run word2vec in debug mode is adding the flag `--debug` and setting that value to `store_true` the default port should be `2333`.
2. The tensorboard debugger allows us to examine the values during the computation of the session, and assists us in determining where issues might be in the code or the training of the model by visualizing as well as providing step by step values for variables during computation.

3.