

Homework 2

Learning Goals

- Understand skip-gram with negative sampling in more detail
 - Compute various derivatives in order to get the gradient of the loss with respect to the parameters
- Learn how to translate math into code, for
 - The model forward pass
 - Gradient computations

Understanding Word2Vec

- Count parameters
- Understand sigmoid, and the role it plays in SGNS
 - Compute its derivative
- Compute the gradient of L_{CE} with respect to parameters
 - Done in stages
 - Uses:
 - Logarithm rules
 - Derivative of logarithm
 - Addition / product / chain rule for derivatives

Implementing Word2Vec

- SGNS will be implemented in raw numpy
- We provide the entire training loop, but various methods that are called need to be filled in
 - Data processing: generating positive and negative samples
 - Model computation: implement the $P(1 | w, c; \theta)$ computation
 - Gradient computation: compute ∇L_{CE} w/r/t each of the relevant parameters

Training Word Vectors

- Finally, you will train word vectors by iterating through the SST training set
- Plot the vectors of a list of words, using PCA for dimensionality reduction
 - We provide all of this code!
- Describe any trends you see in the embeddings

Testing Your Code

- From hw2 onwards, testing methods change. HW1/570-572: compare your output files to a gold standard provided
- Now: we provide *unit tests* (in test_all.py)
 - Small tests that check individual components of code in isolation
 - Very good practice to get in the habit of writing, always used in industry
 - NB: not `_exhaustive_`, i.e. passing unit tests is necessary but not sufficient for “correct” implementation
- How to run: simply run `pytest` from your hw2 folder, with the course’s conda environment activated. It will find all the tests in `test_all.py`.
 - Pytest docs: <https://docs.pytest.org>