Assignment 2

CSE 447 and 517: Natural Language Processing – University of Washington

Winter 2021 – Due: January 27, 2021, 11:59 pm

Submit: You will submit your writeup (a pdf, one-inch margins, 11-point Times font) and your code (do not include data) via Gradescope. Instructions can be found here. Note that you will make two submissions:

Code for problem 4 You will submit your code together with a neatly written README file to instruct how to run your code with different settings. We assume that you always follow good practice of coding (commenting, structuring), and these factors are not central to your grade. You may implement the language models in the programming language of your choice. However, please provide well commented code if you want partial credit. If you have multiple files, please provide a short description in the preamble of each file.

Writeup Part of the training we aim to give you in this class includes practice with technical writing. Organize your report as neatly as possible, and articulate your thoughts as clearly as possible. We prefer quality over quantity. Do not flood the report with tangential information such as low-level documentation of your code that belongs in code comments or the README. Similarly, when discussing the experimental results, do not copy and paste the entire system output directly to the report. Instead, create tables and figures to organize the experimental results.

1 CSE 447 and CSE 517 Students: Based on Eisenstein 6.4 (p. 135)

Consider a simple language in which each token is drawn from the vocabulary $\mathcal{V}$ with probability $\frac{1}{|\mathcal{V}|}$, independent of all other tokens. Given a corpus of size M, what is the expectation of the fraction of all possible bigrams that have zero count? You may assume $\mathcal{V}$ is large enough that $\frac{1}{|\mathcal{V}|} \approx \frac{1}{|\mathcal{V}|-1}$.

2 CSE 447 and CSE 517 Students: Based on Eisenstein 3.4 (p. 65)

Design a feedforward network to compute this function, which is closely related to XOR:

$$f(x_1, x_2) = \begin{cases} -1 & \text{if} \quad x_1 = 1 \wedge x_2 = 1 \\ 1 & \text{if} \quad x_1 = 1 \wedge x_2 = 0 \\ 1 & \text{if} \quad x_1 = 0 \wedge x_2 = 1 \\ -1 & \text{if} \quad x_1 = 0 \wedge x_2 = 0 \end{cases}$$

Your network should have a single output node that uses the “sign” activation function,

$$\text{sign}(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ -1 & \text{if} \quad x \leq 0 \end{cases}$$

Use a single hidden layer, with ReLU activation functions. Describe all weights and offsets.

3 CSE 447 and CSE 517 Students: Based on Eisenstein 3.5 (p. 65)

Consider the same network as in problem 2 (with ReLU activations for the hidden layer), with an arbitrary differentiable loss function $\ell(y^{(i)}, \tilde{y})$, where $\tilde{y}$ is the activation of the output node. Suppose all weights and offsets are initialized to zero. Show that gradient descent will not learn the desired function from this initialization.

4 CSE 447 and CSE 517 Students: Implementation

Your final writeup for this problem should be no more than three pages long.

4.1 Dataset

This tarball provides you with three data files (a subset of the One Billion Word Language Modeling Benchmark [1]). Each line in each file contains a whitespace-tokenized sentence.

- `1b_benchmark.train.tokens`: data for training your language models.
- `1b_benchmark.dev.tokens`: data for debugging and choosing the best hyperparameters.
- `1b_benchmark.test.tokens`: data for evaluating your language models.

A word of caution: You will primarily use the development/validation dataset as the previously unseen data while (i) developing and testing your code, (ii) trying out different model and training design decisions, (iii) tuning the hyperparameters, and (iv) performing error analysis (not applicable in this assignment, but a key portion of future ones). For scientific integrity, it is extremely important that you use the test data only once, just before you report all the final results. Otherwise, you will start overfitting on the test set indirectly. Please don't be tempted to run the same experiment more than once on the test data.

4.2 n-gram Language Modeling

You will build and evaluate unigram, bigram, and trigram language models. To handle out-of-vocabulary (OOV) words, convert tokens that occur **less than three times in the training data** into a special UNK token during training. If you did this correctly, your language model's vocabulary (including the UNK token and STOP, but excluding START) should have 26,602 unique tokens (types).
You will turn in your source code along with the writeup. Your submission will not be evaluated for efficiency, but we recommend keeping such issues in mind to better streamline the experiments.

Deliverables In the writeup, be sure to fully describe your models and experimental procedure. Provide graphs, tables, charts or other summary evidence to support any claims you make.

1. Report the perplexity scores of the unigram, bigram, and trigram language models for your training, development, and test sets. Briefly discuss the experimental results.

4.3 Smoothing

To make your language model work better, you will implement linear interpolation smoothing between unigram, bigram, and trigram models:

$$\theta'_{x_j|x_{j-2},x_{j-1}} = \lambda_1\theta_{x_j} + \lambda_2\theta_{x_j|x_{j-1}} + \lambda_3\theta_{x_j|x_{j-2},x_{j-1}}$$

where θ' represents the smoothed parameters, and the hyperparameters $\lambda_1, \lambda_2, \lambda_3$ are weights on the unigram, bigram, and trigram language models, respectively. $\lambda_1 + \lambda_2 + \lambda_3 = 1$.
You should use the development data to choose the best values for the hyperparameters. Hyperparameter optimization is an active area of research; for this homework, you can simply try a few combinations to find reasonable values.

Deliverables In the writeup, be sure to fully describe your models and experimental procedure. Provide graphs, tables, charts or other summary evidence to support any claims you make.

1. Report perplexity scores on training and development sets for various values of $\lambda_1, \lambda_2, \lambda_3$. Report no more than 5 different sets of λ's. In addition to this, report the training and development perplexity for the values $\lambda_1 = 0.1, \lambda_2 = 0.3, \lambda_3 = 0.6$.

2. Putting it all together, report perplexity on the test set, using the hyperparameters that you chose from the development set. Specify those hyperparameters.

3. If you use half of the training data, would it increase or decrease the perplexity on previously unseen data? Why? Provide empirical experimental evidence if necessary.

4. If you convert all tokens that appeared less than 5 times to `<unk>` (a special symbol for out-of-vocabulary tokens), would it increase or decrease the perplexity on the previously unseen data compared to an approach that converts only a fraction of words that appeared just once to `<unk>`? Why? Provide empirical experimental evidence if necessary.

4.4 Extra Credit: Neural Language Modeling

Train a neural language model; we recommend you use a package such as AllenNLP [2], which has considerable documentation. Training neural models is computationally expensive—start early if you can!

Deliverables

1. Fully describe your model and experimental procedure.

2. Report your neural language model's training, development, and test set perplexity. Discuss your experimental results, especially in comparison to your n-gram models. Provide graphs, tables, charts or other summary evidence to support any claims you make.

5 CSE 517 Students: Based on Eisenstein 3.6 (p. 65)

The simplest solution to problem 3 relies on the use of the ReLU activation function at the hidden layer. Now consider a network with arbitrary monotonic activations on the hidden layer. Show that if the initial weights are any uniform constant, then gradient descent will not learn the desired function from this initialization.

6 CSE 517 Students: Based on Eisenstein 6.5 (p. 135)

Continuing problem 1, determine the value of M such that the fraction of bigrams with zero count is at most $\epsilon \in (0, 1)$. As a hint, you may use the approximation $\ln(1 + \alpha) \approx \alpha$ for $\alpha \approx 0$.

7 CSE 517 Students: Based on Eisenstein 6.6 (p. 136)

In real languages, words' probabilities are neither uniform nor independent. Assume that word probabilities are independent but not uniform, so that in general $p(w) \neq \frac{1}{|\mathcal{V}|}$. Prove that the expected fraction of unseen bigrams will be higher than in the independent and uniform case.

References

[1] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. One billion word benchmark for measuring progress in statistical language modeling. In *Proc. of INTERSPEECH*, 2014.

[2] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proc. of NLP-OSS*, 2018.