Project 2: Sentence VAE

May 3, 2019

This project will help you learn and implement a deep generative language model. In summary, your task is to:

- Pre-process the training, validation, and test data;
- Implement the models as described below;
- Evaluate your model on the test data;
- Perform a qualitative analysis;
- Optionally do the bonus assignment;
- Write a report on the entire process.

1 Deterministic Language Model

Implement a deterministic recurrent neural language model [4]. At each step, an RNNLM parameterises a categorical distribution over the vocabulary of English words V_x, i.e.

$$X_i|x_{<i} \sim \text{Cat}\left(f\left(x_{<i};\theta\right)\right) \tag{1}$$

where

$$\mathbf{h_0} = \mathbf{0} \tag{2a}$$
$$x_0 = \text{SOS} \tag{2b}$$
$$\mathbf{e}_i = \text{emb}\left(x_i;\theta_{\text{emb}}\right) \tag{2c}$$
$$\mathbf{h}_i = \text{RNN}\left(\mathbf{h}_{i-1},\mathbf{e}_{i-1};\theta_{\text{RNN}}\right) \tag{2d}$$
$$\mathbf{s}_i = \text{affine}\left(\mathbf{h}_i;\theta_{\text{out}}\right) \tag{2e}$$
$$f\left(x_{<i};\theta\right) = \text{softmax}\left(\mathbf{s}_i\right) \tag{2f}$$

Here, SOS is a start-of-sentence symbol. $\mathbf{0}$ is a zero vector. The emb(.) function is a lookup in an embedding matrix. RNN(.) indicates one or multiple layers of GRU or LSTM cells,

and the affine transformation maps from the hidden state size to the vocabulary size.

This will be our baseline model.

2 Sentence VAE

Implement a deep generative language model [1]. The probabilistic model now includes an additional continuous latent variable z:

$$X_i|x_{<i}, z \sim \text{Cat}\left(f(x_{<i}, z; \theta)\right) \tag{3}$$

with a prior distribution:

$$Z \sim \mathcal{N}(0, I) \tag{4}$$

We still do generation of one word at a time without Markov assumptions, but now $f(\cdot)$ additionally conditions on z. The architecture of the generative model remains largely the same, but we replace the initialization of the RNN (2a) with:

$$\mathbf{h}_0 = \tanh\left(\text{affine}\left(z; \theta_{\text{ init }}\right)\right) \tag{5}$$

We want to estimate the parameters θ for maximum likelihood. However, computing gradients for the marginal log-likelihood $P(x) = \int P(x, z)dz$ is intractable precluding gradient-based optimization. Therefore, we resort to variational inference, introducing an approximate posterior distribution $q(z|x)$, and optimizing instead a lower-bound (the ELBO) on the log-likelihood. The inference network should be a diagonal Gaussian so that you can compute the KL-divergence term of the ELBO in closed form and obtain reparameterizable samples for gradient estimation [2]. An example architecture of an inference network architecture is:

$$\mathbf{e}_i = \text{emb}\left(x_i; \phi_{\text{emb}}\right) \tag{6a}$$
$$\mathbf{f}_i = \text{RNN}\left(\mathbf{f}_{i-1}, \mathbf{e}_i; \phi_{\text{fwd}}\right) \tag{6b}$$
$$\mathbf{b}_i = \text{RNN}\left(\mathbf{b}_{i+1}, \mathbf{e}_i; \phi_{\text{bwd}}\right) \tag{6c}$$
$$h = \text{dense}([f_n; b_1]; \phi_{hid}) \tag{6d}$$
$$\mu = \text{dense}(h; \phi_{loc}) \tag{6e}$$
$$\sigma = \text{softplus}(\text{dense}(h; \phi_{scale})) \tag{6f}$$
$$z = \mu + \sigma \odot \epsilon \tag{6g}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and the softplus function ensures a positive standard deviation.

3 Metrics

Language models are typically evaluated using the perplexity metric. Perplexity is the exponentiated negative log-likelihood averaged over the number of predictions:

$$\text{ppl} = \exp\left(\frac{\sum_{i=n}^{N} -\log(P(x_n))}{\sum_{i=n}^{N} |x_n|}\right) \tag{7}$$

where N is the size of the dataset, x_n is a sentence in the dataset and $|x_n|$ denotes the length of x_n (including the end of sentence token but excluding the start of sentence token). For latent-variable models, the negative log-likelihood is not always available in closed-form (neither is it in the case of our sentence VAE). Instead, we can approximate it using importance sampling using the approximate posterior as our importance sampling distribution:

$$-\log p(x) \approx -\frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{1}{S}\sum_{k}^{S}\left[\frac{p\left(z_{nk}, x_n\right)}{q\left(z_{nk}|x_n\right)}\right]\right) \quad z_{nk} \sim q\left(z|x_n\right) \tag{8}$$

Report the following quantities when evaluating your models:

- Negative log-likelihood (exact in the RNNLM case, approximated using importance sampling in the sentence VAE case).
- Per-word perplexity, derived from the negative log-likelihood.
- A multi-sample estimate of the evidence lower-bound (ELBO) for the sentence VAE.
- Word prediction accuracy, i.e. the fraction of correctly predicted words, using greedy decoding. When decoding with the sentence VAE, a commonly used heuristic is to use the mean of the approximate posterior instead of a sample.

4 Qualitative Analysis

Qualitatively analyse your models doing the following:

- Sample 10 sentences from the RNNLM.
- Do a greedy decoding from the RNNLM.
- Sample 10 sentences from the Sentence VAE by greedy decoding from a sample z from the prior distribution.
- Test the reconstruction capability of your model by reconstructing a sentence from the test dataset using the approximate posterior mean and 10 samples.

- Show a homotopy between the latent codes of two sentences, i.e. interpolate between the latent code of two sentences and reconstruct a sentence from the intermediate latent codes. Decode from $z_\alpha = \alpha * z_1 + (1 - \alpha) * z_2$, where z_1 is the encoding of the first sentence and z_2 is the encoding of the second sentence and α varies from 0 to 1.

5 Bonus

As a bonus assignment explain the problem of posterior collapse and explain and compare methods to tackle it. Implement KL annealing and word dropout [1], and implement KL free-bits [3, Appendix C.8]. Explain how these methods attempt to tackle posterior collapse. Compare using none of these methods with using KL annealing and word dropout, and with KL free-bits by reporting KL at the end of training on the validation and test sets.

6 Data

All relevant data (including details about file formats) are available from https://uva-slpl.github.io/nlp2/projects.html.

In this project, you will work with the Penn Treebank (PTB) dataset, that consists of English sentences. The training data is *02-21.10way.clean*, the validation data is *22.auto.clean*, and the test data is *23.auto.clean*. Note that the sentences in the PTB dataset are annotated with syntactic trees.

We release the *training* data (which you can use to perform parameter estimation), the *validation* data (which you can use to debug your implementation as well as to perform model selection), and finally in due time the *test* data (which you will use to conduct your final empirical comparison).

7 Report

You should use LaTeX for your report, and you should use the ACL template available from http://acl2017.org/downloads/acl17-latex.zip.
We expect reports (5 pages plus references). The typical submission is organised as follows:

- **Abstract**: conveys scope and contributions;
- **Introduction**: present the problem and relevant background;
- **Model**: technical description of models;

- **Experiments**: details about the data, experimental setup, findings, and qualitative analysis;
- **Conclusion**: a critical take on contributions and limitations.

8 Submission

You should submit a `.tgz` file containing a folder (folder name `lastname1.lastname2.lastname3`) with the report as a single `pdf` file. Your report should contain a link to an open-source repository (such as github), but please do not attach code or additional data to your submission. We will not grade based on the quality of the code, but we will use it to verify that you have implemented the models. You can complete your project submission on Canvas.

9 Assessment

You will be assessed according to the following evaluation criteria:

Implementation
- RNNLM
- Sentence VAE
- Importance sampled NLL
- **Bonus**: KL annealing, word dropout and KL free-bits

Report
- **Scope** (max 2 points): Is the problem well presented? Do students understand the challenges/contributions?
- **Theoretical description** (max 3 points): Are the models presented clearly and correctly?
- **Empirical evaluation** (max 4 points): Is the experimental setup sound/convincing? Are experimental findings presented in an organised and effective manner?
- **Writing style** (max 1 points): use of LaTeX, structure of report, use of tables/figures/plots, command of English.

References

[1] Samuel R. Bowman et al. "Generating Sentences from a Continuous Space". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 10–21. DOI: 10.18653/v1/K16-1002. URL: https://www.aclweb.org/anthology/K16-1002.

[2] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *ICLR*. 2014.

[3] Durk P Kingma et al. “Improved variational inference with inverse autoregressive flow”. In: *Advances in neural information processing systems*. 2016, pp. 4743–4751.

[4] Tomáš Mikolov et al. “Recurrent neural network based language model”. In: *Eleventh annual conference of the international speech communication association*. 2010.