Toward Controlled Generation of Text

1 Idea

The authors aim to disentangle representations of style and content in the latent code of an Adversarial Variational AutoEncoder. The style is called the **structured code** and is learned by discriminators for each attribute that needs to be disentangled from the latent space.

2 Method

- This method does not use adversarial training.
- The basic approach can be described as below:
  - x is the source corpus
  - The encoder is parameterized to generate a latent code z, which is a variational latent space that resembles a Gaussian prior. (This is enforced by a KL-divergence loss)
  - The structured code c is a known label of the text (discrete or continouous)
  - The decoder generator produces the output corpus $\hat{x}$ conditioned on (z, c). It uses greedy decoding.
  - A classifier/regressor discriminator predicts the structured code of the output corpus $\hat{x}$ to ensure that it is the same as the one the generator was conditioned on i.e. $G(z, c)$. The discriminator is pretrained.
  - Each decoder step in $\hat{x}$ is predicted using a softmax function scaled by a temperature τ. Higher temperatures flatten the softmax

distribution for each word prediction and increase word diversity. Conversely, setting $\tau = 0$ will resembled a hardmax. For their experiments the authors gradually anneal $\tau \to 0$

- The authors describe 3 separate losses to train their model.
  - A reconstruction loss that ensures that the generated sentence $\hat{x}$ is the same as the original sentence x. This is equivalent to minimizing the negative log-likelihood of generating $\hat{x}$.
  - A discriminator validates if the predicted class/value for $\hat{x}$ is the same as the corresponding class/value for x. This is a cross-entropy loss over the probability distribution of the labels. This discriminator loss can be further subdivided into 2 terms.
    - Maximize the expected log likelihood of predicting the correct distribution of the structured code c given the labelled examples X_L. This happens before the generator model training.
    - Maximize the expected log likelihood of predicting the correct distribution of the structured code c given the generated sentences $\hat{x}$. Also minimize the empirically observed Shannon entropy of the observed discriminator prediction $q_D(c\prime|\hat{x})$, which reduces uncertainty and increases confidence of the structured code prediction.
  - The encoder from loss 1, is used to regenerate the latent distribution z devoid of the structured code from the output distribution $\hat{x}$. The authors call this an **independence constraint**, in that regardless of the structured code c that is currently present in either x or $\hat{x}$, processing either through the generator should produce a consistent z. This allows the encoder to encode only latent factors that are independent of the structured code.
- A wake-sleep algorithm [1] is used to alternatively train the generator and discriminator.
- The model was applied only to short sentences with length < 15 words.
- The encoder/decoder setup is implemented using single layer LSTMs and the discriminator is implemented using conv-net. The KL term is annealed from 0 to 1 during training.

3 Architecture

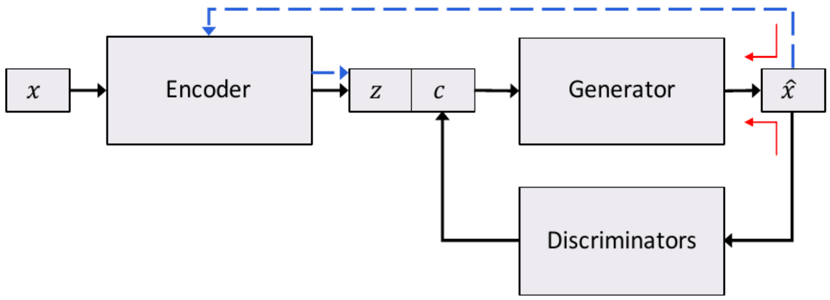

Figure 1. The generative model, where $\boldsymbol{z}$ is unstructured latent code and $\boldsymbol{c}$ is structured code targeting sentence attributes to control. Blue dashed arrows denote the proposed independency constraint (section 3.2 for details), and red arrows denote gradient propagation enabled by the differentiable approximation.

4 Learning Process

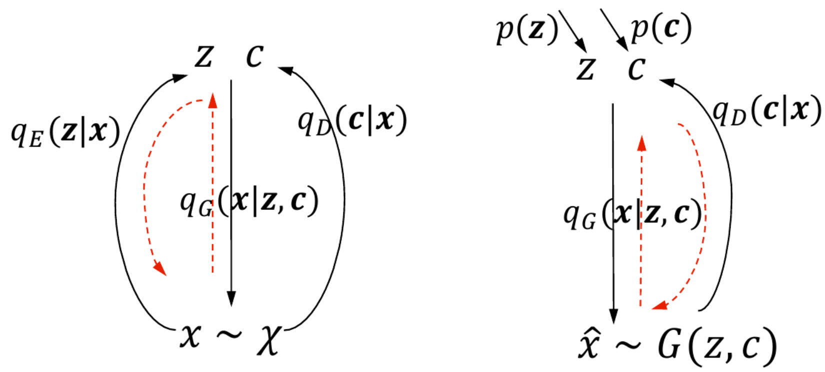

Figure 2. **Left:** The VAE and wake procedure, corresponding to Eq.(4). **Right:** The sleep procedure, corresponding to Eqs.(6)-(7) and (10). Black arrows denote inference and generation; red dashed arrows denote gradient propagation. The two steps in the sleep procedure, i.e., optimizing the discriminator and the generator, respectively, are performed in an alternating manner.

5 Observations

- The model performs better than the S-VAE [2] implementation in terms of sentiment accuracy of generated sentences.
- From the reported results, it seems that adding the independence constraint helps the generated sentences retain the content successfully.

References

[1] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[2] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.