Sequence and Document representation

Yuting Gui
Center of Data Science
60 5th Ave, New York, NY 10011
yg1281@nyu.edu

Abstract

Generative models aim to learn the hidden representation of the input data. It define an explicit stochastic model of data, so that ensure the sample drawn from the hidden representation are likely come from the original data. This paper summarizes popular sentence and document level generative models proposed in recent years, including variational inference, sequence to sequence learning, and some other related methods.

1 Introduction

Unsupervised learning aims to learn the inherent structure in data so as to facilitated future work such as generation, and prediction on image and text process. Generative model is an approach in unsupervised learning, which define an explicit stochastic model of data, so that ensure the samples drawn from model are likely come from the original data.

Auto encoder and decoder framework has been widely used on learning generative model (Bengio et al., 2007) (MarcAurelio Ranzato and LeCun, 2007). The encoder is trained to encoder input $\boldsymbol{X}$ to hidden representation $\boldsymbol{z}$, at the same time, the decoder is trained to reconstruct input $\boldsymbol{X}$ from $\boldsymbol{z}$ (Boureau et al., 2008). So that the target output of auto-encoder is the input $\boldsymbol{X}$. The autoencoder model trained to maximize the likelihood of sample $\boldsymbol{X}$ conditional on $\boldsymbol{z}$, which means the autoencoder attempt to sample values of $\boldsymbol{z}$ that are likely to have produced $\boldsymbol{X}$ (Doersch, 2016).

Recent researches in autoencoder model have lead to some amazing works in natural language process. For tasks such as document modeling and sequence learning, the variational autoencoder has provided a new way to derive an effective approximation for the intractable distributions over latent variables, and are able to generates samples from the conditional distributions. In particular, a major task in modeling text data is to learn the document level representations. Using variational inference method in the autoencoder and decoder model has performed state-of-the-art results. In addition, the autoencoder model also has been successfully used in sentence representation. The encoder that encode the sentence to hidden representation served as a generic feature extractor for different tasks, such as sentiment analysis, text classification, language modeling, and machine translation.

2 Related works

Stochastic Gradient Variational Bayes (SGVB) (Kingma and Welling, 2013) is an estimator that used for approximate posterior inference, which provides a very effective way to train models where the data is assumed to be generated from some continuous latent variables. Based on that, Variational Recurrent Autoencoder(VRAE) (Fabius and van Amersfoort, 2014) combines the strength of RNN and SGVB, enable us to do unsupervised learning on time series data. Moreover, a generic variational framework for generative and conditional models of text has been introduced to solve tasks of text modeling and question answering (Miao et al., 2015). Using this framework, a document level variational model (NVDM) has been exploited to reconstruct document from bags of words input. Kingma et al. also introduced a semi-supervised variational inference framework especially for partially labeled data (Kingma et al., 2014). A semi-supervised sequential variational autoencoder (SSVAE) has been recently proposed to incorporate sequence to sequence learning with semi-supervised variational inference framework, and achieves state-of-the-art results on text classification.

In sentence level unsupervised learning, Skip-

Thought model reconstruct adjacent sentences from an encoded passage (Kiros et al., 2015). A sentence autoencoder model that use LSTM RNN (Hochreiter and Schmidhuber, 1997) to read sentence into a single vector (Dai and Le, 2015), and use that to reconstruct the input sentence, which can be used as a pretrain algorithm for supervised learning tasks. A variational autoencoder language model aims to learn global latent representations of sentence (Bowman et al., 2015), and the results shown that it performs very well in imputing missing words task.

In the following part, we will discuss models that have successfully used unsupervised learning concepts in detail, and analyze the strength and weakness of the unsupervised learning realization in NLP related problems.

3 Variational autoencoder in Language

3.1 Variational autoencoder

Autoencoder aims to map every datapoint $\boldsymbol{X}$ to a hidden representation $\boldsymbol{z}$, so that datapoint $\boldsymbol{X}$ can be easily reconstruct conditioned on the hidden representation. Variational autoencoder (VAE) (Kingma and Welling, 2013) assumes that samples of the hidden representation $\boldsymbol{z}$ can be drawn from a simple Gaussian distribution $N(0, I)$, which is the prior $p(\boldsymbol{z})$. A VAE aims to minimize the KL divergence between the posterior distribution $q_\phi(z|X)$ and the prior distribution $p(\boldsymbol{z})$, which turns out can be used to extract a valid lower bound on the true log likelihood of the datapoint $\boldsymbol{X}$, $(\log p(\boldsymbol{X}))$.

$$\begin{aligned}\mathcal{L}(\boldsymbol{X};\boldsymbol{\phi},\boldsymbol{\theta}) = &-D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{X})||p(\boldsymbol{z})] \\ &+\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{X})}[\log p_\theta(\boldsymbol{X}|\boldsymbol{z})] \\ &\le \log p(\boldsymbol{X})\end{aligned}$$

This can be interpret as an autoencoder, which consists a probabilistic encoder $q_\phi(\boldsymbol{z}|\boldsymbol{X})$ and a decoder $p_\theta(\boldsymbol{X}|\boldsymbol{z})$.

3.2 Neural Variational Document Model

The Neural Variational Document Model (NVDM) (see Figure 1) is an unsupervised text generative model which learns the continuous semantic hidden representation for each document. The model exploits the idea of variational autoencoder: It take the document bag-of-words representation $\boldsymbol{X} \in \mathbb{R}^{|V|}$ as an input, where V is the vocabulary size. The MLP encoder (inference network) $q_\phi(\boldsymbol{z}|\boldsymbol{X})$ aims to learn the continuous hidden representation $\boldsymbol{z} \in \mathbb{R}^K$ of a document, where K is the hidden vector's size. The softmax decoder (generative model) $p_\theta(\boldsymbol{X}|\boldsymbol{z}) = \prod_{i=1}^{N} p(\boldsymbol{x_i}|\boldsymbol{z})$ aims to reconstruct the document by independently generating the words $(\boldsymbol{z} \rightarrow \{\boldsymbol{x_i}\})$, where $\boldsymbol{x_i}$ represents each word in the document, one document has N words in total, and N is variant over all documents. In order to maximize the log likelihood $\log \sum_z p(\boldsymbol{X}|\boldsymbol{z})p(\boldsymbol{z})$ of documents, the lower bound has been derived as:

$$\begin{aligned}\mathcal{L}(\boldsymbol{X};\phi,\theta) = &-D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{X})||p(\boldsymbol{z})] \\ &+\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\sum_{i=1}^{N}\log(p_\theta(\boldsymbol{x_i}|\boldsymbol{z}))] \\ &\le \log \sum_z p(\boldsymbol{X}|\boldsymbol{z})p(\boldsymbol{z})\end{aligned}$$

where $p(\boldsymbol{z})$ is the Gaussian prior of $\boldsymbol{z}$.

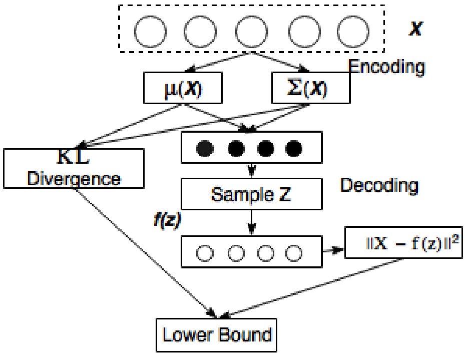

Figure 1: NVDM for document modelling

As a generative model, NVDM generates each word directly from the continuous hidden representation. NVDM achieved state-of-the-art perplexities on the *20NewsGroups* and *RCV1-v2*.

3.3 Generating Sentences with Variational Autoencoder

A variational autoencoder language model (Bowman et al., 2015) is used to capture global features of sentences in a continuous hidden space. The model depicted in Figure 2.

This model exploits LSTM language model (Hochreiter and Schmidhuber, 1997) for both encoder and decoder. The encoder maps the input sequence to a hidden vector $\boldsymbol{z}$, and the decoder language model reconstruct the input sentence word

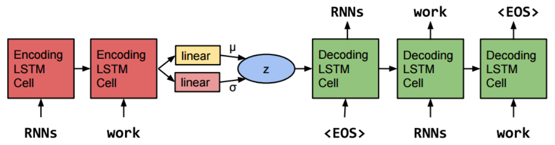

Figure 2: Variational autoencoder language model

by word conditioned on z. In the degenerate setting, the hidden representation does not incorporate useful information thus this model perform effectively equivalent with regular language model in language modeling task. However, it turns out the global features extracted by VAE are useful in imputing missing words task. The model is trained with Books Corpus (Kiros et al., 2015). First, the words in the dataset has been randomly dropped with fixed drop rate. Then train the model to fill in the missing words. In order to evaluate the model, an *adversarial evaluation* method inspired by generative adversarial networks (Goodfellow et al., 2014) has been proposed. The performance of imputation measured by examining imputing sentence's distinguishablility from the true sentence. The results show VAE language model is substantially better than regular LSTM language model, which suggested the VAE language model produce more diverse sample than LSTM language model.

4 Sentence representation

Sentence and document representation is an important intermediate task for many NLP applications, such as, document retrieval, web search, spam filtering, and text classification. Traditional classification and clustering algorithms require fixed dimension vector as input. However, not all of the sentences or documents in training set have same length, which prevent us from directly use text as input. The most common solution are bag-of-words, and bag-of-n-grams text representation (Harris, 1954), which provide a vocabulary size vector for each text. Although both of two methods represent a text with a fixed size vector, they suffer from different extent of word order lost and data sparsity problems. From semantic perspective, bag-of-words and bag-of-n-grams almost embed no semantic information of words, they consider every word equally. In the following part, we introduce several effective model for generating text representation.

4.1 Paragraph Vector

Le and Mikolov introduce a continuous distributed vector representations for text called Paragraph Vector (PV) (Le and Mikolov, 2014), the text could be sentence, paragraph and document. They used word *paragraph* to emphasis that model can apply to variant-length text representation tasks. PV is an unsupervised framework that generate a hidden representation of paragraph, to help predicts word in this text. The paragraph vectors are different among paragraphs, but the word vectors, initialized with trained word embedding, are shared across all paragraphs. PV encompasses two different models: a distributed memory model and distributed bag of words model.

A distributed memory model's structure displayed in Figure 3. Every paragraph associated with an unique id, and each id is mapped to a unique p dimensional vector D, and every word is also mapped to a unique q dimensional vector W. The goal is to use paragraph vector and word vector as hidden layer to predict next word in a context. The hidden layer can be constructed with paragraph vector and concatenated or averaged word vectors. This model is similar to the Continuous Skip-gram Model (Mikolov et al., 2013a).

The author used paragraph vector and concatenated or averaged word vector as hidden representation, which treat word vectors equally. It might not make sense because a paragraph's meaning could depends more on several important words. One possible solution is to train an attention model between paragraph vector and word vectors, such that increase important word's weight and decrease others' on making the prediction. This method is more useful when context window size is large.

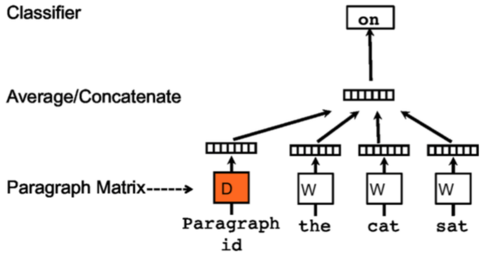

Figure 3: Paragraph Vector: A distributed memory model

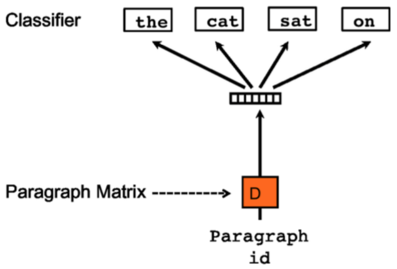

Figure 4: Paragraph Vector without word ordering: Distributed bag of words

Distributed bag of words model (Figure 4) ignores context words in the input. It only takes the paragraph representation as input, and make the model to predict words that randomly selected from paragraph. This model requires to store less data, thus is faster to train. This idea is inspired from the skip-gram model in word vectors (Mikolov et al., 2013b).

These two models have following main advantages over bag-of-words: Both of them embed semantic information of word by use word embedding; The distributed memory model contains word order information in a small context; The hidden representation of paragraph has same size with word vector, therefore it does not suffer high dimensionality problem. The paragraph Vector models achieve state-of-the-art results on Sentiment Analysis and Information Retrieval tasks.

4.2 Skip-Thought

Skip-Thought is an unsupervised learning method, which learns a sentence level representation. The model's structure inherits from the Skip-gram (Mikolov et al., 2013b). Instead of use a word to predict its surrounding words, this model use a sentence to predict surrounding sentences. Intuitively, a sentence's meaning can be infer from its neighbor sentences, which enable us to learn the representation of a sentence by its neighbors. Skip-Thought (See Figure 5) can be framed as an autodecoder model. For each (s_{i-1}, s_i, s_{i+1}) tuple sentences, the encoder and decoder do the following jobs:

The encoder is a neural language model which generates a representation of s_i. In Skip-Thought, a GRU encoder scans words in s_i one by one. At each time step, the encoder generates a hidden state h_i^t, which can be regarded as a hidden meaning representation of all words before the t^{th} word. So that the last hidden state h_i of sentence s_i is a representation of this whole sentence, which wraps all words information.

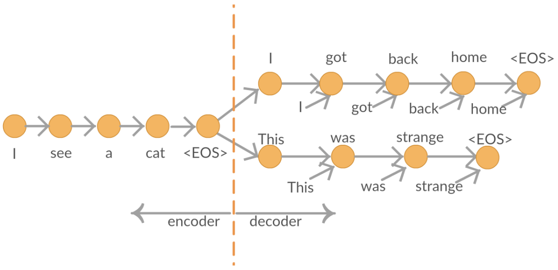

Figure 5: Skip-Thought encoder-decoder model

The decoder is also a neural language model, but conditions on the hidden representation of s_i. Two GRU decoders scan words in s_{i-1} and s_{i+1} separately. The decoder GRU has similar structure with the encoder GRU, except the decoder GRU add bias to the update gate, reset gate, and hidden state, which computes by hidden representation h_i. The decoders trained as an language model, and the objective function is the sum of the log probabilities for s_{i-1} and s_{i+1} conditioned on the encoder representation.

4.3 Sequence to Sequence learning and Sequence Autoencoder

Sequence to sequence learning (seq2seq) is a supervised model (Sutskever et al., 2014) that map variable-dimension input sequences to variable-dimension output sequences. Many NLP tasks are related with this mapping. For example, machine translation, speech recognition and question answering. This model can be frame as an encoder-decoder model (See Figure 6). The encoder is an LSTM which read the input sentence and warp it to a fixed length vector. The decoder LSTM is a recurrent neural network language model, initialized with the last state of encoder LSTM.

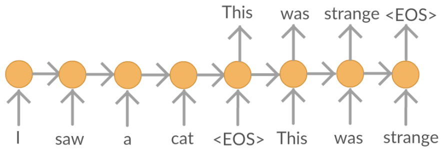

Figure 6: Sequence to sequence model

Sentence autoencoder (Dai and Le, 2015) is an unsupervised learning method that learn sentence representation (see Figure 7). The idea is inspired

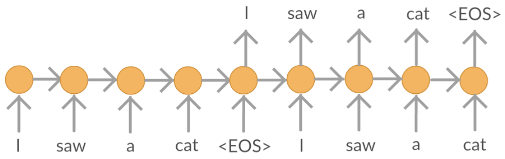

Figure 7: Sentence autoencoder model

by aforementioned sequence to sequence learning model. Sentence autoencoder can be regarded as a variation of sequence to sequence learning. It also used a LSTM as a recurrent neural network encoder to read the input sentence into hidden state. The decoder LSTM is a language model aims to reconstruct the input sentence, and the two LSTM share same weights. For each training step t, a decoder LSTM unit generates y(t), which represents probability distribution of next word given previous word, hidden state and context vector. Softmax ensures the output probability distribution is valid. The error is calculated by cross-entropy criterion and weight is updated by back propagation.

Sentence autoencoder has similar structure with the Skip-Thought. The difference is the sentence autoencoder aims to reconstruct the input sentence, whereas Skip-Thought's goal is to predict adjacent sentences. Since sentence autoencoder is an unsupervised model, thus can be trained with large amount of unlabeled data to improve it quality. The weights obtained from sentence autoencoder can be used as pretrain model to initialized other supervised network, thus improve the classification performance.

4.4 Multi-task Sequence to Sequence

Multi-task learning or lifelong learning framework has been widely studied by Thrun (1996), and Caruana (1998), whose goal is to improve generalization performance of a task by re-use the knowledge gathered in the other related tasks. Recently seq2seq approach has achieve state-of-the-art performance in many tasks, such as machine translation (Luong and Manning, 2015), image captioning (Vinyals et al., 2015b), and constituency parsing (Vinyals et al., 2015a). Multi-task seq2seq learning (Luong et al., 2015) aims to exploited the power of seq2seq model across many tasks, thus boost the performance of seq2seq model in a specific task. With this purpose, Luong et al. proposed three multi-task learning approaches with seq2seq. (1) the one-to-many approach, which has a common encoder and different decoders. (2) the many-to-one approach, which has a decoder in common, but used multiple encoders (3) many-to-many approach, which share multiple encoders and decoders.

One-to-many approach (see Figure 8) involves one common encoder and multiple decoders for different tasks. The encoder map an English sentence to a hidden vector, multiple separate decoders map the hidden vector to a translation sentence in German, and to a sequence of parsing tags, and its surrounding sentences (skip-thought) or itself (seq2seq autoencoder). This approach only suitable for tasks that need to encode an sentence, image captioning is not a suitable subtask, because it requires encode a picture. All decoders are separate RNN language model.

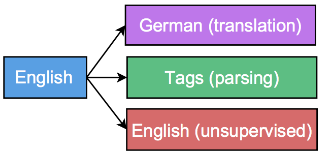

Figure 8: One-to-many multi-task model

Many-to-one approach (see Figure 9) involves multiple different encoders and shared one decoder. This approach is useful when only decoder can be shared. For example, constituency parsing task is not a suitable task, because a parsing tag sequence can not map to an English sentence.

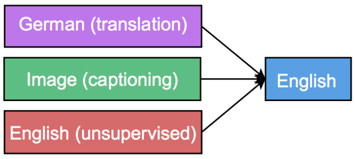

Figure 9: Many-to-one multi-task model

Many-to-many approach (see Figure 10) extend the aforementioned approaches, it involves multiple encoders and multiple decoders. The author design this approach especially for machine translation, in order to utilize the large monolingual corpora in both the source and the target languages.

One thing is worth to mention here is the training strategy. Inspired by Dong et al. (2015), Multi-task seq2seq learning allocate different times of parameter updates for each task, the up-

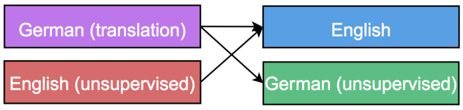

Figure 10: Many-to-many multi-task model

dating frequency is associated with the mixing ratio α_i for each task i. Each parameter update consists on one task training data. Task is switch to another with probability $\frac{\alpha_i}{\sum_j \alpha_j}$ for task i. The convention is that the first task is the *reference* task with $\alpha_1 = 1$, and the number of training parameter updates for the first task is pre-specified to be N. Therefore, every typical task i will then be trained for $\frac{\alpha_i}{\alpha_1}N$ times parameter updates.

The results show that syntactic parsing and image caption generation improves the translation quality between English and German. For unsupervised learning objectives, seq2seq autoencoder help less in terms of perplexities, but more on BLEU scores compared to Skip-Thought.

4.5 Variational Autoencoder for Semi-supervised Text Classification

Recently a Semi-supervised Sequential Variational Autoencoder (SSVAE) is proposed for semi-supervised sequential text classification task. The SSVAE adapt the seq2seq model to the semi-supervised variational inference framework (Kingma et al., 2014), with a modification of the decoder, which feed the label $\boldsymbol{y}$ at each time step.

In order to fully understand the SSVAE, we briefly introduce the semi-supervised variational inference model. This model consists of two objective functions for labeled data $(\boldsymbol{X}, \boldsymbol{y})$ and unlabeled data $(\boldsymbol{X})$. For labeled data $(\boldsymbol{X}, \boldsymbol{y})$, the variational lower bound with corresponding latent variable $\boldsymbol{z}$ is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}; \phi, \theta) &= -D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{y})||p(\boldsymbol{z})] \\ &+ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{y})}[\log p_\theta(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{y})] + \log p_\theta(\boldsymbol{y}) \\ &\leq \log p_\theta(\boldsymbol{X}, \boldsymbol{y})\end{aligned}$$

where the first term is the KL divergence between the prior distribution $p(\boldsymbol{z})$ and the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{y})$, and the second term is the expectation of the conditional log likelihood on $\boldsymbol{z}$.

For labeled data $(\boldsymbol{X})$, the unobserved label $\boldsymbol{y}$ is treated as a latent variable, predicted by the inferred posterior distribution $q_\phi(\boldsymbol{y}|\boldsymbol{X})$, the variational lower bound is:

$$\begin{aligned}\mathcal{U}(\boldsymbol{X}; \phi, \theta) = &\sum_y q_\phi(\boldsymbol{y}|\boldsymbol{X})(-\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}; \phi, \theta)) \\ &+ \log q_\phi(\boldsymbol{y}|\boldsymbol{X}) \\ &\leq \log p_\theta(\boldsymbol{X})\end{aligned}$$

Thus, the objective for the entire dataset is:

$$\begin{aligned}J = &\sum_{(X,y)\in S_l} \mathcal{L}(\boldsymbol{X}, \boldsymbol{y}) + \sum_{X\in S_u} \mathcal{U}(\boldsymbol{X}) \\ &+ \alpha \mathbb{E}_{(X,y)\in S_l}[-\log q_\phi(\boldsymbol{y}|\boldsymbol{X})]\end{aligned}$$

where S_l and S_u represent labeled and unlabeled data respectively, α is a weight for classification loss of labeled data. The semi-supervised variational inference model consists of three components: an encoder network $q_\phi(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{y})$, a decoder network $p_\theta(\boldsymbol{X}|\boldsymbol{y}, \boldsymbol{z})$, and a classifier $q_\phi(\boldsymbol{y}|\boldsymbol{X})$.

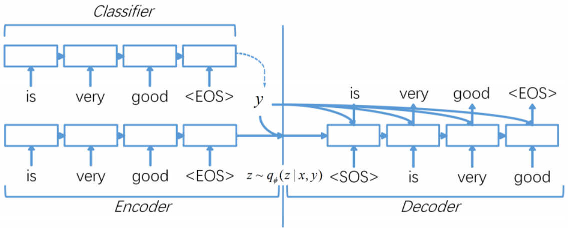

Figure 11: Semi-supervised Sequential AutoEncoder

SSVAE (see Figure 11) incorporates seq2seq model with semi-supervised variational inference model. The encoder and the classifier are LSTM network, which encode the input sequence to hidden representation $\boldsymbol{z}$ and hidden label $\boldsymbol{y}$ if $\boldsymbol{y}$ is unobserved. The decoder is a little bit different in terms of seq2seq decoder (RMMLM). The decoder is a LSTM language model conditioned on concatenation of $\boldsymbol{z}$ and $\boldsymbol{y}$, and the true or estimated label $\boldsymbol{y}$ feeds into every step of the language model when it estimate the probablity $p_\theta(\boldsymbol{X}|\boldsymbol{y}, \boldsymbol{z})$.

The author studied with two methods to incorporate $\boldsymbol{y}$ in decoder. The first on concatenates the word embedding and label vector at each time step, which has been widely used in Ghosh et al. (2016) and Sarban et al. (2016). The second method made a modification of the cell gate in LSTM, which is inspired by Wen et al. (2015).

The model is applied with IMDB and AG's news dataset, and achieves the state-of-the-art performance in the classification task by combining the pretraining method (Dai and Le, 2015).

5 Conclusion

This paper summarizes several document and sentence level generative models based on variational inference method and sequence to sequence learning model. Variational inference method has been shown its power in image generation and classification, we expect to see more application with variational inference in text related tasks. Sequence to sequence model efficiently solve the mapping problem form the variant length input to variant length output. It achieves state-of-the-art performances in many language tasks, we also expect more effective and efficient models inherits the seq2seq ideas.

6 Acknowledgements

This paper is for the final paper of. We would like to thank our professor and adviser Sam Bowman for inspiring suggestions and insightful lectures.

References

Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. 2007. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153.

Y-lan Boureau, Yann L Cun, et al. 2008. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, pages 95–133. Springer.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087.

Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing*, pages 1723–1732.

Otto Fabius and Joost R van Amersfoort. 2014. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Y MarcAurelio Ranzato and Lan Boureau Sumit Chopra Yann LeCun. 2007. A unified energy-based framework for unsupervised learning. In *Proc. Conference on AI and Statistics (AI-Stats)*, volume 24.

Yishu Miao, Lei Yu, and Phil Blunsom. 2015. Neural variational inference for text processing. *arXiv preprint arXiv:1511.06038*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Sebastian Thrun. 1996. Is learning the n-th thing any easier than learning the first? *Advances in neural information processing systems*, pages 640–646.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015a. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.