Semi-supervised Variational Autoencoders for Text Classification

Weidi Xu, Haoze Sun, Chao Deng, Ying Tan
Key laboratory of Machine Perception(MOE), Peking University
Department of Machine Intelligence, School of Electronics
Engineering and Computer Science, Peking University, Beijing, 100871, China
wead_hsu@pku.edu.cn

Abstract

Although semi-supervised learning method based on variational autoencoder (Semi-VAE) works well in image classification tasks, it fails in text classification tasks if using vanilla LSTM as its conditional generative model. We find that the model with this setting is unable to utilize the positive feedback mechanism of Semi-VAE and hence fails to boost the performance. To tackle this problem, a conditional Long Short-Term Memory network (conditional LSTM) is presented, which receives the conditional information all the time-steps. In addition, auxiliary variable is also found to be useful in our method in terms of both training speed and prediction accuracy. Experimental results on Large Movie Review Dataset (IMDB) show that the proposed approach significantly improves the classification accuracy compared with pure-supervised classifier and achieves competitive performance against previous pre-training based methods. Additional improvement can be obtained by integrating pre-training based methods.

1 Introduction

Semi-supervised learning is a critical problem due to the fact that the data size nowadays is increasing much more faster than before, while only a limited subset of data samples has their corresponding labels. Hence lots of attention has been drawn from researchers over machine learning and deep learning communities, giving rise to many semi-supervised learning methods. In previous pre-training based semi-supervised methods [Hinton et al., 2006, Vincent et al., 2010, Bengio et al., 2007, Socher et al., 2013, Dai and Le, 2015], unsupervised learning is often considered to get better initial weight parameters by trying to extract the hidden features that are beneficial to data reconstruction. And then a supervised fine-tuning phase is followed. However some of these features may be irrelevant to determining the class label in certain classification tasks [Rasmus et al., 2015].

Variational autoencoder is recently proposed by Kingma and Welling [2014], Rezende et al. [2014] and it has been applied for semi-supervised learning [Kingma et al., 2014, Maaløe et al., 2016]. We refer to these kinds of models as SemiVAEs. In these models, the classifier is separated from generative model and hence it can learn to classify directly. They have shown strong performance on image classification tasks, however their application in sequential text classification problem has been out of sight for a long time. Text classification problem is much different from common image classification problem in several aspects: 1) the lengths of data samples are variable, 2) the data distributions with same category are more diverse and 3) the categorical information (e.g., sentiment) is more abstract.

Due to these difficulties we propose a novel semi-supervised learning method for text classification, combining both SemiVAE and its application in sequence generation [Bowman et al., 2016]. Two crucial components are found to apply SemiVAEs successfully in textual domain. 1) A conditional...
LSTM is proposed for conditional generative model, as the vanilla LSTM based generative model failed to distinguish between different categorical labels. By using conditional LSTM, the categorical feature is fed to generative model at each step, explicitly informing the generative model to be conditioned on given category. With its help, the resulting model becomes effective in text classification tasks. 2) Further we show that auxiliary variable [Maaløe et al., 2016] works well for our method. By using auxiliary variable the training time of our model can be shorted about 3 times while the accuracy still be improved.

In summary our contributions are:

- A novel semi-supervised deep generative model for text classification tasks is proposed. Assuming that textual context is roughly independent from sentimental expression, our model is trained to jointly optimize both text classifier and sequential variational autoencoder. The resulting model is able to generate diverse sentences conditioned on certain categorical labels.
- We show how our method can be applied in text domain effectively with the help of conditional LSTM, and verify the effectiveness of auxiliary variable for our method. These techniques are analyzed in details in this paper.
- We demonstrate the performance of our approach by providing competitive result on IMDB dataset, particularly with a few data samples. Our model can still achieve less 9% classification error using only one-tenth of labeled data.

The article is organized as follows. In the next section, we introduce several related works. And then our model is presented in section 3. In section 4, we obtain both quantitative results and qualitative analysis of our models. At last we conclude our paper with a detailed discussion.

2 Preliminaries

2.1 Semi-supervised learning using variational autoencoder

Based on variational auto-encoders [Kingma and Welling, 2014, Gregor et al., 2015, Burda et al., 2015], conditional varional autoencoders [Maaløe et al., 2016, Yan et al., 2015] are proposed to generate samples according to certain attributions of given labels. Kingma et al. [2014] firstly introduced a semi-supervised learning method (SemiVAE) to successfully separate the image style and content information. This method consists of two objectives for labeled and unlabeled data. Given labeled data $X,Y = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ the lower bound of SemiVAE is:

$$\log p_\theta(x, y) \geq E_{q_\phi(z|x,y)}[\log p_\theta(x, y|z)] - D_{KL}(q_\phi(z|x,y)||p(z)) = -L(x, y),$$

where $D_{KL}$ is the Kullback-Leibler divergence.

For the unlabeled data, the unobserved label $y$ is predicted from the inference model with a learnable classifier $q_\phi(y|x)$. The lower bound is hence:

$$\log p_\theta(x) \geq \sum_y q_\phi(y|x)(-L(x, y)) + H(q_\phi(y|x)) = -U(x).$$

This formula is the key for this semi-supervised learning method. Note that the first term is the expectation of labeled lower bound on $q_\phi(y|x)$ and intuitively we can explain the training procedure as follows: 1) once the model can roughly figure out correctly the likelihood $p_\theta(x, y)$ of a certain sample $x$ between ground-true label $y_t$ and negative label $y_n$. The lower bound $-L(x, y_t)$ is likely to be greater than $-L(x, y_n)$. Hence the gradient computed for $q_\phi(y_t|x)$ has a greater coefficient than for $q_\phi(y_n|x)$, and thus reinforce the classifier. 2) if the classifier can predict the right label for the most time, the generative network will be strengthened in a similar way. This procedure forms a positive feedback for optimizing both classifier and generative model. We will show that using conditional LSTM is necessary to apply this mechanism successfully in sequence classification tasks.

The objective for entire dataset is now:

$$J = \sum_{(x,y) \in S_l} L(x, y) + \sum_{x \in S_u} U(x) + \alpha E_{(x,y) \in S_l}[-\log q_\phi(y|x)],$$

where $\alpha$ is a hyper-parameter of additional classification loss of labeled data.
Figure 1: This is the sketch of our conditional variational autoencoder for sequence, which is the essential component of our method. **Left:** The sequence is encoded by a recurrent neural network and **Right:** followed by another recurrent neural network for generation. Conditional input $y$ is imported to both parts. A sample $z$ from prior $p_\theta(z|x, y)$ is passed to generative nework and it is used to recover the sequential data $x$.

### 2.2 Variational autoencoder for sentence generation

Recurrent neural network (RNN) is one of the most successful methods for sequence generation tasks such as machine translation and image captioning. However, one-by-one generation mechanism in RNNs can’t extract high level features like topic, style and sentiment properties. To solve this problem, variational recurrent autoencoders (VRAEs) have been employed in modeling global features for sequential data like sentences [Bowman et al., 2016] and music [Fabius and van Amersfoort, 2014]. Similar with the aforementioned models, VRAEs use encoder-decoder structure. In inference model, sequence $x$ is processed by a encoder RNN to extract the global information, while in generative model latent variable $z$ is used to initialize the hidden state for decoder RNNs.

### 3 Sequential deep generative model for semi-supervised learning

Motivated by the recent works on semi-supervised learning, we propose **Sequential Deep Generative Model** (SeqDGM) for text classification. This model consists of two components: a conditional variational autoencoder sketched in Figure 1 and a typical sequential classifier $q_\phi(y|x)$.

Specifically the inference model of our method is described as following equations:

$$
\hat{x} = f_{enc}(x), \\
\mu(x, y) = W([\hat{x}; y]), \\
\sigma^2(x, y) = \exp(W([\hat{x}; y])), \\
q_\phi(z|y,x) = N(z|\mu(x, y), \text{diag}(\sigma^2(x, y))),
$$

where sequence $x$ is encoded by $f_{enc}$ function (typically a LSTM network [Hochreiter and Schmidhuber, 1997]) and $y$ is represented as a one-hot vector. The encoding is concatenated with $y$ to set the diagonal Gaussian distribution, parameterized by $\mu(x, y)$ and $\sigma^2(x, y)$. Throughout the paper we use the notation $b = W(a)$ to denote a linear weight matrix with bias from vector $a$ to vector $b$ for clarity.

Another component in conditional variational autoencoder is the sequential conditional generative network, which decodes the sequence $x$ according to latent variable $z$ and class label $y$. Stochastic sequential data is generated from a pre-defined prior distribution $p(z) = N(0, 1)$:

$$
z \sim p(z), \\
\hat{x} = f_{dec}(y, z),
$$

where sequence $\hat{x}$ is recovered by a conditional generative model $f_{dec}$.

#### 3.1 Conditional LSTM as conditional generative model

To model the sequential generation $f_{dec}(y, z)$ conditioned on both latent variable $z$ and class label $y$, we originally simply concatenate $y$ and $z$ as the initial state for a LSTM network. The model with this setting do actually learn to capture label information with labeled data only. However in semi-supervised learning experiments where the unlabeled data is included, we found this simple implementation failed to improve the classification performance. The model figures out that ignoring
the class feature $y$ and minimizing the generation likelihood according to language model (i.e.,
predicting next word according to a small context windows) is the best strategy to optimize the
objective function. This is because the categorical information is passed to generative network
only at the first time-step but the conditional generation probability is maximized over all kinds of
categories (each multiplied with a coefficient produced by classifier $q_\phi(y|x)$ in equation\textsuperscript{2}). When the conditional generative model can not distinguish between different categorical labels, the model
will be incapable of taking advantage of positive feedback mechanism described in section \textsuperscript{2.1} and
hence fails to boost the performance using \textit{SemiVAEs}.

Considering this, conditional LSTM network is proposed as conditional generative model, which
explicitly forces the generation to be aware of given label at each time-step. Assuming the categorical
information will be possibly reflected anywhere in the sentences, the conditional LSTM is defined by
the following equations:

\begin{align}
  i_t &= \sigma(W_{wi}w_t + W_{hi}h_{t-1}), \\
  f_t &= \sigma(W_{wf}w_t + W_{hf}h_{t-1}), \\
  o_t &= \sigma(W_{wo}w_t + W_{ho}h_{t-1}), \\
  c_t &= \tanh(W_{wc}w_t + W_{hc}h_{t-1}), \\
  h_t &= o_t \odot c_t + i_t \odot \tilde{c}_t + \tanh(W_{yc}y), \\
  p(x|y) &= \frac{p(x,y,a,z)}{q_\phi(a|x)q_\phi(z|a,y,x)} \\
\end{align}

where the equations are the same as in standard LSTM networks except that equation\textsuperscript{14} has an extra
term about $y$. Therefore the model is informed of the existence of $y$ all the steps, and the conditional
generative model has to distinguish between the different labels during training.

### 3.2 Using auxiliary variable

Auxiliary variable [Maaløe et al., 2016] is brought into VAEs to improve the variational approximation
of latent space. Considering the complexity of text sequence distribution, we would expect that
auxiliary variable works well in the domain of text. Combining auxiliary variable, the objectives keep
similar but are equipped with an extra latent variable $a$. For labeled data, the objective changes to:

\begin{align}
  \log p_\theta(x,y) &\geq \mathbb{E}_{q_\phi(a,z|x,y)} \left[ \log \frac{p_\theta(x,y,a,z)}{q_\phi(a|x)q_\phi(z|a,y,x)} \right] = -\mathcal{L}(x,y). \quad (16) \\
\end{align}

As for unlabeled data:

\begin{align}
  \log p_\theta(x) &\geq \mathbb{E}_{q_\phi(a,z,y|x)} \left[ \log \frac{p_\theta(x,y,a,z)}{q_\phi(a|x)q_\phi(y|a,x)q_\phi(z|a,y,x)} \right] = -\mathcal{U}(x). \quad (17) \\
\end{align}

In our implementation, $p_\theta(x,y,a,z)$ is modelled by $p_\theta(y)p_\theta(z)p_\theta(a|y,z)p_\theta(x|a,y,z)$. In each
inference function, sequence $x$ will be firstly encoded into a vector and then concatenated with
other conditional factors for succeeding MLP layer. For factor $p_\theta(x|a,y,z)$, the data sample $x$ is
decoded conditioned on $y$ using a conditional LSTM, whose initial hidden state is presented by
auxiliary variable $a$ and latent variable $z$. We call this the \textbf{Auxiliary Sequential Deep Generative Model (SeqDGM-A)}.

### 4 Experimental results and analysis

This section will show experimental results on Large Movie Review Dataset [Maas et al., 2011],
sometimes known as IMDB dataset. The dataset consists of 25K labeled data samples for training,
25K labeled data samples for testing and 50K unlabeled samples used for semi-supervised learning.
For valid set, we randomly chose 5K samples from training set. In our implementation, the dataset
was preprocesssd and results in a vocabulary of 20K most frequent words.

The system was implemented using Theano [Bastien et al., 2012] [Bergstra et al., 2010] and
Lasagne [Dieleman et al., 2015]. Both SeqDGM and SeqDGM-A were trained end-to-end using the
ADAM [Kingma and Ba, 2015] optimizer with learning rate of 4e-3 and a mini-batch size of 56. The
reparameterization trick [Kingma and Welling, 2014] [Rezende et al., 2014] was used for stochastic
Table 1: Performance of methods on the IMDB sentiment classification task

| Model                                      | Test error rate |
|--------------------------------------------|-----------------|
| LSTM with tuning and dropout [Dai and Le, 2015] | 13.50%          |
| LSTM initialized with word2vec embeddings [Dai and Le, 2015] | 10.00%          |
| Full+Unlabeled+BoW [Maas et al., 2011]    | 11.11%          |
| WRRBM + BoW (bnc) [Maas et al., 2011]     | 10.77%          |
| NBSVM-bi (Naive Bayes SVM with bigrams)   | 8.78%           |
| seq2-bown-CNN (ConvNet with dynamic pooling) [Johnson and Zhang, 2015] | 7.67%           |
| LM-LSTM [Dai and Le, 2015]                | 7.64%           |
| SA-LSTM [Dai and Le, 2015]                | 7.24%           |
| SeqDGM                                     | 9.68%           |
| SeqDGM-A                                   | 8.73%           |
| SeqDGM-A (with pre-training)              | 7.55%           |

Table 2: Performance of methods with different amount of labeled data

| Method                                      | 2.5K  | 5K   | 10K  | 20K  |
|---------------------------------------------|-------|------|------|------|
| LSTM (our implementation)                  | 17.97%| 15.67% | 12.99% | 10.90% |
| SeqDGM                                      | 13.10%| 10.46% | 10.16% | 9.68%  |
| SeqDGM-A                                   | 10.28%| 9.50%  | 9.62%  | 8.73%  |
| LM-LSTM (our implementation)               | 9.41% | 8.90% | 8.44% | 7.65% |
| SeqDGM-A (with pre-training)               | 8.94% | 8.68% | 8.22% | 7.55% |

backpropagation through the Gaussian latent variables. In all the experiments, we used 512 units for memory cells, 300 units for the input embedding projection layer and 50 units for latent variable $z$. Hyper-parameter $\alpha$ was scaled from 1 to 2. Auxiliary variable was presented by a vector of size 20 in SeqDGM-A. We apply both dropout [Srivastava et al., 2014] and batch normalization [Ioffe and Szegedy, 2015] to the output of the word embedding projection layer and to the feature vectors that serve as the inputs and outputs to the MLP that precedes the final classifier.

The experiments were conducted on data with different size of labeled samples. To do this, we sampled data points uniformly from training set of each category and left the others unlabeled. The results shown reflect the training and test set performance of each model at the training step at which the model performs best on the development set.

Although various classifiers can be utilized for $q_{\phi}(y|x)$ in our method, standard LSTM-based classifier was used in both SeqDGM and SeqDGM-A. Averaged hidden state over the entireties of the sentence is fed to final softmax layer, since we found that averaging hidden states can ease the flow of gradient, particularly for earlier words. A LSTM-based classifier with averaged hidden state input was implemented and it served as a baseline for both SeqDGM and SeqDGM-A. From Table 2, our models are able to outperform this baseline by a large margin, but still a little worse than LM-LSTM. It is a pre-training based semi-supervised learning method [Dai and Le, 2015] which use the final hidden state as the input for softmax layer. One explanation would be that the LSTM classifier with averaged hidden state input has its performance limit.

Hence we also investigated the ability of our models without averaged hidden state input, in which case pre-trained weights would be a necessity. Note that our models are well suited with pre-training based methods, because their classifiers are independent from other components. We re-implemented LM-LSTM and SA-LSTM models according to [Dai and Le, 2015] (their codes have not been published yet) and reproduced the equivalent LM-LSTM performance reported in their paper. Unfortunately we are unable to reproduce their best results of SA-LSTM. Therefore LM-LSTM was used as a baseline for this comparison. LM-LSTM was initialized with a language model and the same initial LSTM weights were used by SeqDGM-A. By using SeqDGM-A, additional improvement can still be obtained, particularly with a small amount of data (cf. Table 2). With a few of labeled data samples, the classifier can not reach its best performance through supervised learning. However, the proposed method makes full use of all the unlabeled data and learns additional discriminative information to help improving the performance. A summary of previous results on IMDB dataset are listed in Table 1.
4.1 Importance of using conditional LSTM

Here we investigated the model using vanilla LSTM or conditional LSTM as its conditional generative model quantitatively. In vanilla LSTM setting, the categorical label $y$ is simply concatenated with latent variable $z$ and it is fed into the standard LSTM. The importance of using conditional LSTM as conditional sequential generative model, rather than vanilla LSTM, has been emphasized in Section 3. Recall the positive feedback process mentioned in Section 2.1, both classifier and conditional generative model need to capture the difference between categorical labels. We show that model using vanilla LSTM fails to be aware of the label $y$ and hence fails to boost the performance using SemiVAEs. At first we define the following index to evaluate the discrimination ability of conditional generative model:

$$D = \frac{\#(−L(x, y_p) > −L(x, y_n))}{N},$$

where $y_p$, $y_n$ denotes ground positive categorical label and negative categorical label, $N$ is the number of total labeled data and $−L(x, y)$ is the lower bound in equation $[1]$. $\#(−L(x, y_p) > −L(x, y_n))$ denotes the number of samples that conditional generative model can produce higher evidence lower bound of generative likelihood with correct labels. We draw the curves of models using these two conditional generative models together with classification accuracy $A$ in Figure 2.

By using conditional LSTM, the accuracy improves rapidly as well as $D$ index, which indicates the strong correlation between the accuracy of classifier and discrimination ability of conditional generative model. At the early phase of training, the accuracy of vanilla LSTM improves quickly as well, but diverges at epoch 13. Meanwhile the $D$ index improves very slowly, making it impossible to achieve well performance. This is believed that the reconstruction objective will mislead the training process, if not enough guiding information is provided by the conditional generative model.

In addition we have also investigated the model which receives the concatenation of word embedding and label vector as in [Ghosh et al., 2016], but it performed a little worse than conditional LSTM by about 2%.

4.2 Effectiveness of auxiliary variable

Auxiliary variable has shown the capacity in modeling complex variational distribution for image classification problems. Here we demonstrate that it is also very helpful in textual data in terms of both training speed and prediction accuracy. Table 2 shows that the performance is evidently improved by using auxiliary variable, and more gain can be obtained when the number of labeled data is less. Figure 3 demonstrates the learning process for both SeqDGM and SeqDGM-A with same learning rate. SeqDGM-A converges in 15 epochs, which is much faster than the model without the auxiliary variable that converges in more than 50 epochs.
Figure 3: IMDB classification accuracy with 5K labeled data samples using learning rate of 4e-3.

Table 3: Generated sentences conditioned on different categorical label $y$ and same latent state $z$.

| Negative                                                                 | Positive                                                                 |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------|
| this has to be one of the worst movies I’ve seen in a long time .       | this has to be one of the best movies I’ve seen in a long time .         |
| what a waste of time ! ! !                                              | what a great movie ! ! !                                               |
| all i can say is that this is one of the worst movies i have ever seen . | anyone who wants to see this movie is a must see ! ! !                 |
| <UNK> is one of the worst movies i’ve seen in a long time .              | <UNK> is one of my favorite movies of all time.                         |
| if you haven’t seen this film , don’t waste your time ! ! !             | if you haven’t seen this film , don’t miss it ! ! !                    |
| suffice to say that the movie is about a group of people who want to see this movie , but this is the only reason why this movie was made in united states . | suffice to say that this is one of those movies that will appeal to children and adults alike , but this is one of the best movies i have ever seen . |
| and most of the characters are so <UNK> that they have no idea what they were doing in the movie , and they were <UNK> in the movie . | and of course the characters are very well done , and they have to be the best of his life , and he has to be the best of his career . |

4.3 Conditional generation

A good explorative evaluation of the model’s ability to comprehend the data manifold is to evaluate the generative model. To demonstrate sentiment and context separation, we randomly sampled several $z$ and generate sentences using trained conditional generative model $p_{\theta}(x|y, z)$. Table 3 demonstrates several cases using the same latent variable $z$ but with opposite sentimental labels. Sentences generated by the same $z$ share a similar syntactic structure and words, but their sentimental implications are much different with each other. The model seems to be able to recognize the frequent sentimental phrases and remember them according to categorical label $y$. While faced with the difficulty for a model to understand real sentiment implication, it is interesting that some sentences can even express the sentimental information beyond the sentimental phrases, e.g., “but this is the only reason why this movie was made in united states”.

4.4 Implementation details

**Cost annealing**  Cost annealing is a training trick introduced by [Bowman et al., 2016], [Kaae Sønderby et al., 2016], which gradually increases the weight of KL cost from zero to one. Without this trick the model tends to ignore the input $x$ and most trained models is likely to consistently set $q_{\phi}(z|\cdot)$ approximate to $p(z)$. This technique is adopted in our implementation.

**Sentence sampling**  Truncated sequence is used for generative module for fast training. In the implementation we randomly draw sub-sequence uniformly from each data sample with length of
600. This trick alleviates the difficulty for training the conditional generative model and accelerate the training speed.

**Word dropout** To improve generalization ability, word dropout [Bowman et al., 2016] technique is utilized. In conditional sequence generative network, we randomly drop out some words and replace them with blanks. This method introduces noise into networks and help them to be more general. The dropout rate was scaled from 0.1 to 0.25 in our experiments.

5 Conclusion and future works

Based on deep generative model, a semi-supervised learning method has been proposed for text classification. To the best of our knowledge, this is the first time for *SemiVAEs* to be applied for sequential data. Two essential techniques to make our approach effective have been represented and explained, including the usage of conditional LSTM as conditional generative model and adoption of auxiliary variable for better performance. Our method can achieve competitive performance compared with previous state-of-the-art results. When using 2.5K labeled samples, less 9% classification error can still be obtained. With a conditional generative model, our method can generate diverse sentences conditioned on certain sentimental labels.

The conditional generative model plays an significant role in our approach, however current conditional LSTM may still be too straightforward to understand the high level abstract. In the future we expect to improve the conditional generative model by integrating more information, e.g., syntactic structure and grammar, which we believe is a necessary work for general language generation.

References

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. *Theano: new features and speed improvements*. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

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

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. *Theano: a CPU and GPU math expression compiler*. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. *Generating sentences from a continuous space*. In The International Conference on Learning Representations (ICLR), Caribe Hilton, San Juan, Puerto Rico, 2016.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance weighted autoencoders*. *arXiv preprint arXiv:1509.00519*, 2015.

Andrew M Dai and Quoc V Le. *Semi-supervised sequence learning*. In *Advances in Neural Information Processing Systems*, pages 3061–3069, 2015.

Sander Dieleman, Jan Schlüter, Colin Raffel, Colin Olson, SK Sønderby, D Nouri, D Maturana, M Thoma, E Battenberg, J Kelly, et al. *Lasagne: First release*. Zenodo: Geneva, Switzerland, 2015.

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

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

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. *DRAW: A recurrent neural network for image generation*. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1462–1471, 2015.
Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML 2015, Lille, France, 6-11 July 2015, pages 448–456, 2015.

Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *NAACL HLT 2015, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 103–112, 2015.

Casper Kaæe Sonderby, Tapani Raiko, Lars Maaløe, Søren Kaæe Sonderby, and Ole Winther. How to train deep variational autoencoders and probabilistic ladder networks. *arXiv preprint arXiv:1602.02282*, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *The International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

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.

Lars Maaløe, Casper Kaæe Sonderby, Søren Kaæe Sonderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *NAACL HLT 2011*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3532–3540, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

Nitin Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.