Modeling documents with Generative Adversarial Networks

John Glover
Aylien Ltd.
Dublin, Ireland
john@aylien.com

Abstract

This paper describes a method for using Generative Adversarial Networks to learn distributed representations of natural language documents. We propose a model that is based on the recently proposed Energy-Based GAN, but instead uses a Denoising Autoencoder as the discriminator network. Document representations are extracted from the hidden layer of the discriminator and evaluated both quantitatively and qualitatively.

1 Introduction

The ability to learn robust, reusable feature representations from unlabelled data has potential applications in a wide variety of machine learning tasks, such as data retrieval and classification. One way to create such representations is to train deep generative models that can learn to capture the complex distributions of real-world data.

In recent years, two effective approaches to training deep generative models have emerged. The first is based on the Variational Autoencoder (VAE) [1; 2], where observed data x is assumed to be generated from a set of stochastic latent variables z. The VAE introduces an inference network (implemented using a deep neural network) to approximate the intractable distributions over z, and then maximizes a lower bound on the log-likelihood of $p(x)$. The second approach is to use Generative Adversarial Networks (GANs) [3]. In the original GAN formulation, a *generator* deep neural network learns to map samples from an arbitrary distribution to the observed data distribution. A second deep neural network called the *discriminator* is trained to distinguish between samples from the empirical distribution and samples that are produced by the generator. The generator is trained to create samples that will fool the discriminator, and so an adversarial game is played between the two networks, converging on a saddle point that is a local minimum for the discriminator and a local maximum for the generator.

Both VAE and GAN approaches have shown impressive results in producing generative models of images [4; 5], but relatively little work has been done on evaluating the performance of these methods for learning representations of natural language. Recently however, VAEs have been used successfully to create language models [6], to model documents, and to perform question answering [7]. This paper investigates whether GANs can also be used to learn representations of natural language in an unsupervised setting. We describe a neural network architecture based on a variation of the recently proposed Energy-Based GAN that is suitable for this task, provide a qualitative evaluation of the learned representations, and quantitatively compare the performance of the model against a strong baseline on a standard document retrieval task.

Workshop on Adversarial Training, NIPS 2016, Barcelona, Spain.

1.1 Related work

Representations for documents are often created by using generative topic models such as Latent Dirichlet Allocation (LDA) [8]. In LDA, documents consist of a mixture of topics, with each topic defining a probability distribution over the words in the vocabulary. Each document is therefore represented by a vector of mixture weights over its associated topics.

More recently, an undirected topic model based on the restricted Boltzmann machine (RBM) [9] called the Replicated Softmax [10] was proposed. Instead of viewing documents as distributions over topics, it forms a binary distributed representation of each document, and was shown to outperform LDA both as a generative document model and as a means of representing documents for a retrieval task. One problem with the Replicated Softmax model is that it becomes too computationally expensive to use it with larger vocabulary sizes, as the complexity scales linearly with the vocabulary size. This was one of the factors that led to the development of an autoregressive neural topic model called DocNADE [11], which is based on the NADE model [12]. The DocNADE model outperformed the Replicated Softmax when evaluated using the same document modelling and retrieval benchmark, and had an added advantage in that the probability of an observation could be computed exactly and efficiently. The most recent state-of-the-art for this task is a deeper version of DocNADE [13].

2 Learning representations with Generative Adversarial Networks

The original GAN formulation [3] consists of a min-max adversarial game between a generative model G and a discriminative model D. $G(z)$ is a neural network, that is trained to map samples z from a prior noise distribution $p(z)$ to the data space. $D(x)$ is another neural network that takes a data sample x as input and outputs a single scalar value representing the probability that x came from the data distribution instead of $G(z)$. D is trained to maximise the probability of assigning the correct label to the input x, while G is trained to maximally confuse D, using the gradient of $D(x)$ with respect to x to update its parameters. The training procedure is described by Equation 1.

$$\min_G \max_D E_{x\sim p(data)}[\log D(x)] + E_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{1}$$

One shortcoming with this model is that there is no explicit means for inference, and so it is unclear how GANs should be used to do unsupervised representation learning. In [3] two possible solutions are suggested, and have been explored by the research community in subsequent works. The first approach is to train another network to do inference, learning a mapping from x back to z [14; 15; 16], with a variation on this method being to instead use the adversarial training process to regularize an Autoencoder's representation layer [17]. The second idea is to use internal components of the discriminator network as a representation [4]. We investigated both approaches, but in initial experiments we found it difficult to find an architecture that resulted in stable training across a range of datasets and model hyperparameters when using a probabilistic discriminator network. Performance improved significantly however when we switched to using the Energy-Based GAN architecture, where the discriminator is an Autoencoder [18]. Document representations can then be formed from the encoded representation of the discriminator. We describe this model concretely in Section 3.

3 An adversarial document model

Let $\mathbf{x} \in \{0,1\}^V$ be the binary bag-of-words representation of a document, where V is the vocabulary size and $\mathbf{x}_i$ is a binary value indicating whether the i^{th} word is present in the document or not. We define a feedforward generator network $G(\mathbf{z})$ that takes a vector $\mathbf{z} \in \mathbb{R}^{h_g}$ as input and produces a vector $\hat{\mathbf{x}} \in \mathbb{R}^V$, with h_g being the number of dimensions in the input noise vector (sampled from $\mathcal{N}(0, I)$). We also define a discriminator network $D(\mathbf{x})$, seen as an energy function, that takes vectors $\mathbf{x} \in \mathbb{R}^V$ and produces an energy estimate $E \in \mathbb{R}$.

One difference to [18] is that we use a Denoising Autoencoder (DAE) as our energy function, as the DAE has been found to produce superior representations to the standard Autoencoder [19]. In this work we use single encoding and decoding layers, so the encoding process is

$$\mathbf{h} = f(\mathbf{W}^e\mathbf{x}^c + \mathbf{b}_e) \tag{2}$$

where $\mathbf{W}^e$ is a set of learned parameters, $\mathbf{b}_e$ is a learned bias term, f is a nonlinearity, $\mathbf{x}^c$ is a corrupted version of $\mathbf{x}$, and $\mathbf{h} \in \mathbb{R}^{h_d}$ is the hidden representation of size h_d. The decoding process is given by

$$\mathbf{y} = \mathbf{W}^d \mathbf{h} + \mathbf{b}_d \tag{3}$$

where $\mathbf{W}^d$ and $\mathbf{b}_d$ are another learned set of weights and bias terms. The final energy value is the mean squared reconstruction error:

$$\frac{1}{V} \sum_{i=1}^{V} (\mathbf{x}_i - \mathbf{y}_i)^2 \tag{4}$$

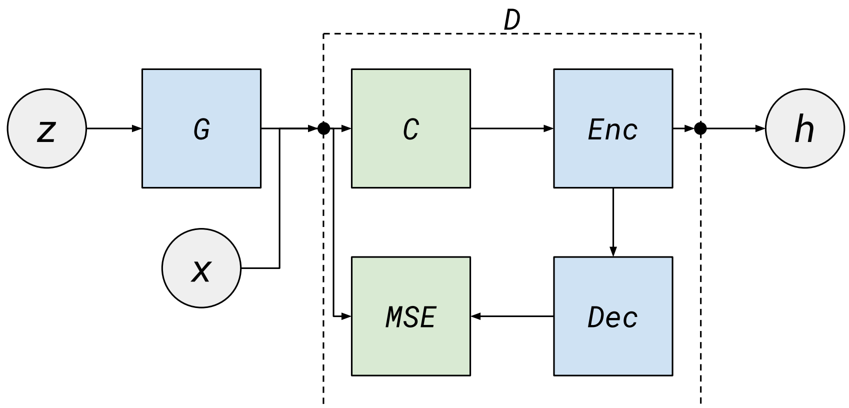

Figure 1: Using an Energy-Based GAN to learn document representations. G is the generator, Enc and Dec are DAE encoder and decoder networks, C is a corruption process (bypassed at test time) and D is the discriminator.

The model is depicted in Figure 1. The energy function is trained to push down on the energy of real samples $\mathbf{x}$, and to push up on the energy of generated samples $\hat{\mathbf{x}}$ [18]. This is given by Equation 5, where f_D is the value to be minimised at each iteration and m is a margin between positive and negative energies.

$$f_D(\mathbf{x}, \mathbf{z}) = D(\mathbf{x}) + \max(0, m - D(G(\mathbf{z}))) \tag{5}$$

At each iteration, the generator G is trained adversarially against D to minimize f_G:

$$f_G(\mathbf{z}) = D(G(\mathbf{z})) \tag{6}$$

4 Experiments

In this section we present experimental results based on the 20 Newsgroups[1] corpus, comparing the adversarial document model to DocNADE. 20 Newsgroups consists of 18,786 documents (postings) partitioned into 20 different newsgroups, where each document is assigned to a single topic. The data is split into 11,314 training and 7,532 test documents. We apply the standard preprocessing as in [10] and set the vocabulary size to 2000.

4.1 Training details

In order to make a direct comparison with [11] we set our representation size h_d (the size of the DAE hidden state) to 50. The generator input noise vector h_g is also set to be the same size. The generator is a 3-layer feedforward network, with ReLU activations in the first 2 layers and a sigmoid nonlinearity in the output layer. Layers 1 and 2 are both of size 300, with the final layer being the same size as the vocabulary. Layers 1 and 2 use batch normalization [20]. The discriminator encoder consists of a single linear layer followed by a leaky ReLU nonlinearity (with a leak of 0.02). The decoder is a linear transformation back to the vocabularly size. We optimize both G and D using Adam [21] with an initial learning rate of 0.0001. Our DAE corruption process is to randomly zero 40% of the input values, and we use a margin size m of 5% of the vocabulary size. We follow the same validation procedure as [11], setting aside a validation set of 1000 documents and using

[1] `http://qwone.com/~jason/20Newsgroups`

the average precision at 0.02% retrieved documents as a performance measure for model selection. DocNADE was trained using the publicly available code[2], with a learning rate of 0.01 and using the tanh activation function.

4.2 Document retrieval evaluation

We performed the same document retrieval evaluation as in [10; 11; 13]. All of the held-out test documents are used as queries and compared to a fraction of the closest documents in the training set, where similarity is calculated using the cosine similarity between the vector representations. The average number of returned documents that have the same label as the query document (the precision) are recorded. The results are shown in Figure 2. In its current formulation, the adversarial document model still falls short of DocNADE's performance, particularly for recall values between 0.002 and 0.05. It does approach DocNADE performance for lower recall values however. As the adversarial document model uses a DAE as a discriminator, for reference we also include results obtained by just training a single layer DAE with the same corruption process, nonlinearity and mean-squared error loss, and also to a version of the adversarial document model that uses a standard Autoencoder instead of a DAE as a discriminator. Both of these models perform worse than the adversarial document model and DocNADE.

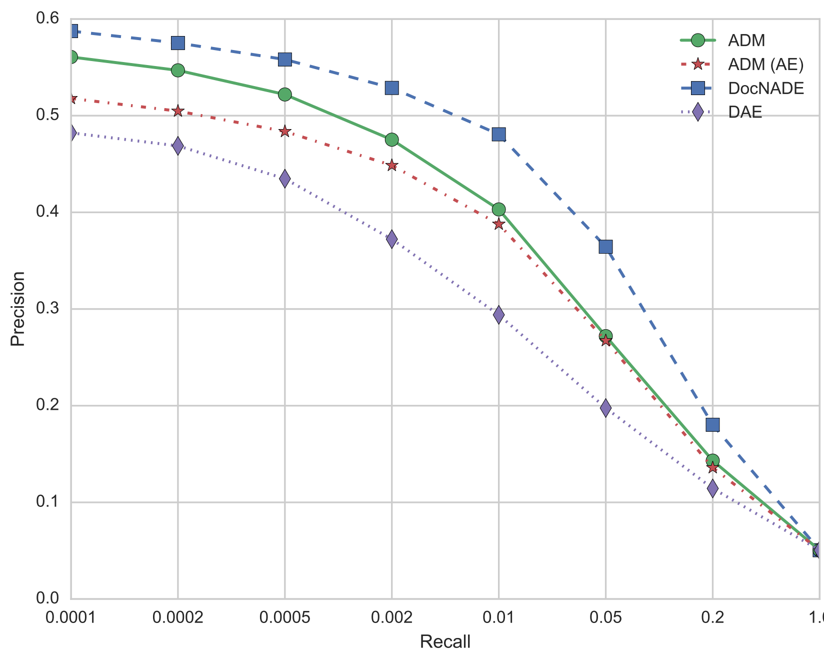

Figure 2: Precision-recall curves for the document retrieval task on the 20 Newsgroups dataset. ADM is the adversarial document model, ADM (AE) is the adversarial document model with a standard Autoencoder as the discriminator (and so it similar to the Energy-Based GAN [18]), and DAE is a Denoising Autoencoder [19].

4.3 Qualitative evaluation

In this section we explore the semantic properties of the representations that are learned by the adversarial document model. For each hidden unit in the discriminator DAE, the weight values associated with each word in the vocabulary can be viewed as the relative importance of a word to that particular hidden unit, and so the hidden units can be interpreted as topics. Table 1 shows the words with the strongest absolute weight connections to selected hidden units in the discriminator encoder layer, i.e. the words w that have the largest absolute values of $\mathbf{W}^e_{:,i}$ for a selected hidden unit i. We can deduce that these collections of words represent the topics of computing, sports and religion,

[2] `http://www.dmi.usherb.ca/~larocheh/code/docnade.zip`

Table 1: 20 Newsgroups hidden unit topics

Computing	Sports	Religion
windows	hockey	christians
pc	season	windows
modem	players	atheists
scsi	baseball	waco
quadra	rangers	batf
floppy	braves	christ
xlib	leafs	heart
vga	sale	arguments
xterm	handgun	bike
shipping	bike	rangers

showing that the model is able to learn locally interpretable structure despite having no interpretability constraints imposed on the representation layer. However, in general we notice that the words that are strongly associated with each hidden unit do not necessarily belong to a single coherent topic, as is evident by the inclusion of the words *bike* and *rangers* in a collection of words related to religion. Figure 3 shows a visualization of the learned representations created using t-SNE [22].

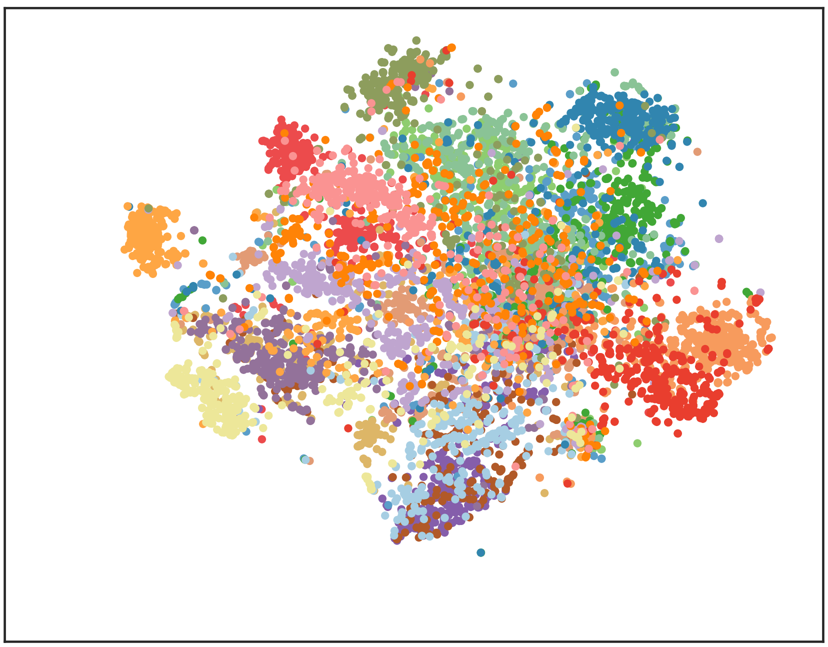

Figure 3: t-SNE visualizations of the document representations learned by the adversarial document model on the held-out test dataset of 20 Newsgroups. The documents belong to 20 different topics, which correspond to different coloured points in the figure.

5 Conclusion

This paper shows that a variation on the recently proposed Energy-Based GAN can be used to learn document representations in an unsupervised setting. It also suggests some interesting areas for future research, including understanding why the DAE in the GAN discriminator appears to produce significantly better representations than a standalone DAE, and exploring the impact of applying additional constraints to the representation layer.

References

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *Proceedings of the International Conference on Learning Representations*, 2014.

[2] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *Proceedings of The 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[4] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *Proceedings of the International Conference on Learning Representations*, 2016.

[5] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A Recurrent Neural Network For Image Generation," *Proceedings of The 32nd International Conference on Machine Learning*, 2015.

[6] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating Sentences from a Continuous Space," *Proceedings of the International Conference on Learning Representations*, 2016.

[7] Y. Miao, L. Yu, and P. Blunsom, "Neural Variational Inference for Text Processing," *Proceedings of the International Conference on Learning Representations*, 2016.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2012.

[9] G. E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[10] R. Salakhutdinov and G. Hinton, "Replicated softmax: An undirected topic model," *Advances in Neural Information Processing Systems*, pp. 1607–1614, 2009.

[11] H. Larochelle and S. Lauly, "A neural autoregressive topic model," *Advances in Neural Information Processing Systems*, pp. 2708–2716, 2012.

[12] H. Larochelle and I. Murray, "The Neural Autoregressive Distribution Estimator," *International Conference on Machine Learning*, vol. 15, pp. 29–37, 2011.

[13] S. Lauly, Y. Zheng, A. Allauzen, and H. Larochelle, "Document Neural Autoregressive Distribution Estimation," *Journal of Machine Learning Research*, vol. 1, pp. 1–19, 2016.

[14] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arxiv:1512.09300*, 2015.

[15] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially Learned Inference," *arXiv preprint arxiv:1606.00704*, 2016.

[16] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial Feature Learning," *arXiv preprint arxiv:1605.09782*, 2016.

[17] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial autoencoders," *International Conference on Learning Representations Workshop*, 2016.

[18] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based Generative Adversarial Network," *arXiv preprint arxiv:1609.03126*, 2016.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, no. 3, pp. 3371–3408, 2010.

[20] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint arxiv:1502.03167*, 2015.

[21] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[22] L. Van Der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.