Deep Belief Nets for Topic Modeling

Maaløe, Lars; Arngren, Morten; Winther, Ole

Published in:
Proceedings of the 31st International Conference on Machine Learning (ICML 2014)

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Deep Belief Nets for Topic Modeling
Workshop on Knowledge-Powered Deep Learning for Text Mining (KPDLTM-2014)

Lars Maaloe LARSMA@DTU.DK
DTU Compute, Technical University of Denmark (DTU) B322, DK-2800 Lyngby

Morten Arngren MOA@ISSUU.COM
Issuu, Gasværksvej 16, 3., DK-1656 Copenhagen

Ole Winther OWI@IMM.DTU.DK
Cognitive Systems, DTU Informatics, Technical University of Denmark (DTU) B321, DK-2800 Lyngby

Abstract

Applying traditional collaborative filtering to digital publishing is challenging because user data is very sparse due to the high volume of documents relative to the number of users. Content based approaches, on the other hand, is attractive because textual content is often very informative. In this paper we describe large-scale content based collaborative filtering for digital publishing. To solve the digital publishing recommender problem we compare two approaches: latent Dirichlet allocation (LDA) and deep belief nets (DBN) that both find low-dimensional latent representations for documents. Efficient retrieval can be carried out in the latent representation. We work both on public benchmarks and digital media content provided by Issuu, an online publishing platform. This article also comes with a newly developed deep belief nets toolbox for topic modeling tailored towards performance evaluation of the DBN model and comparisons to the LDA model.

1. Introduction

This article concerns the comparison of deep belief nets (DBN) and latent Dirichlet allocation (LDA) for finding a low-dimensional latent representation of documents. DBN and LDA are both generative bag-of-words models and represent conceptual meanings of documents. Similar documents to a query document are retrieved from the low-dimensional output space through a distance measurement. A deep belief net toolbox (DBNT)[1] has been developed to implement the DBN and evaluate comparisons. The advantage of the DBN is that it has the ability of a highly non-linear dimensionality reduction, due to its *deep* architecture (Hinton & Salakhutdinov, 2006). A very low-dimensional representation in output space results in a fast retrieval of similar documents to a query document. The LDA model is a mixture model seeking to find the posterior distribution between its visible and hidden variables (Blei et al., 2003). The number of topics K must be given for the LDA model defining the dimensionality of the Dirichlet-distributed output space. The latent representation of a document is the probability for the document to be in each topic, comprising of a vector of size K. To run simulations on the LDA model, we have used the Gensim package for Python[2] (Řehůřek & Sojka, 2010). The article is conducted in collaboration with Issuu[3], a digital publishing platform delivering reading experiences of magazines, books, catalogs and newspapers.

2. Deep Belief Nets

The DBN is a direct acyclic graph except from the top two layers that form an undirected bipartite graph. The top two layers is what gives the DBN the ability to unroll into a deep autoencoder (DA) and perform reconstructions of the input data (Bengio, 2009). The DBN consist of a visible layer, output layer and a number of hidden layers. The training process of the DBN is defined by two

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

[1] Refer to Github.com *Deep Belief Nets for Topic Modeling*.
[2] http://radimrehurek.com/gensim/models/ldamodel.html.
[3] http://issuu.com

steps: *pre-training* and *fine-tuning*. In pre-training the layers of the DBN are separated pairwise to form *restricted Boltzmann machines* (RBM). Each RBM is trained independently, such that the output of the lower RBM is provided as input to the next higher-level RBM and so forth. This way the layers of the DBN are trained as partly independent systems. The goal of the pre-training process is to achieve approximations of the model parameters. A document is modeled by its word count vector. To model the word-count vectors the bottom RBM is a *replicated softmax model* (RSM) (Salakhutdinov & Hinton, 2010). The hidden layers of the RBMs consist of *stochastic binary units*. Training are executed through *Gibbs sampling* using contrastive divergence as the approximation to the gradient (Hinton, 2002). The RBMs applies to *batch learning* and the model only performs a single Gibbs step before updating the weights (Hinton, 2012). Given a visible input vector $\hat{v} = [v_1, ..., v_D]$ the probability of a hidden unit j is given by

$$p(h_j = 1|\hat{v}) = \sigma(a_j + \sum_{i=1}^{D} v_i W_{ij}), \tag{1}$$

where σ denotes the logistic sigmoid function, a_j the bias for the hidden unit j, v_i the state of visible unit i, W_{ij} the weight between visible unit i and hidden unit j and D denotes the number of visible units. Except for the RSM, the visible units are binary, where the probability is given by

$$p(v_i = 1|\hat{h}) = \sigma(b_i + \sum_{j=1}^{M} h_j W_{ij}), \tag{2}$$

where b_i denotes the bias of visible unit i and M the number of hidden units. The RSM assumes a multinomial distribution where the units of the visible layer are softmax units. Having a number of softmax units with identical weights is equivalent to having one multinomial unit sampled the same number of times (Salakhutdinov & Hinton, 2010). The probability of v_i taking on value n is

$$p(v_i = n|\hat{h}) = \frac{e^{b_i + \sum_{j=1}^{M} h_j W_{ij}}}{\sum_{q=1}^{D} e^{b_q + \sum_{j=1}^{M} h_j W_{qj}}}. \tag{3}$$

The RSM consider the number of words in each document by scaling the bias terms of the hidden units with the length of each document. The weights and biases of the RBM are updated by

$$\Delta W = \epsilon(\mathbb{E}_{p_{data}}[\hat{v}\hat{h}^T] - \mathbb{E}_{p_{recon}}[\hat{v}\hat{h}^T]), \tag{4}$$

$$\Delta \hat{b} = \epsilon(\mathbb{E}_{p_{data}}[\hat{h}] - \mathbb{E}_{p_{recon}}[\hat{h}]), \tag{5}$$

$$\Delta \hat{a} = \epsilon(\mathbb{E}_{p_{data}}[\hat{v}] - \mathbb{E}_{p_{recon}}[\hat{v}]), \tag{6}$$

where ϵ is the learning rate and the distribution denoted p_{recon} defines the reconstruction of the input data p_{data} and is the result of a Gibbs chain running a single Gibbs step. $\mathbb{E}_{p_{data}}[\cdot]$ is the expectation with respect to the joint distribution of the real data $p_{data}(\hat{h}, \hat{v}) = p_{data}(\hat{h}|\hat{v})p_{data}(\hat{v})$. $\mathbb{E}_{p_{recon}}[\cdot]$ denotes the expectation with respect to the reconstructions. To optimize the training we add weight decay and momentum to the parameter update (Hinton & Salakhutdinov, 2010). The model parameters from pre-training is passed on to the fine-tuning. The network is transformed into a DA, by replicating and mirroring the input and hidden layers and attaching them to the output of the DBN. Backpropagation on unlabeled data can be performed on the DA, by computing a probability of the input data $p(\hat{x})$ instead of computing the probability of a label $\hat{t}$ provided the input data $p(\hat{t}|\hat{x})$. This way it is possible to generate an error estimation by comparing the normalized input data to the output probability. The stochastic binary units of the pre-training is replaced by sigmoid units with deterministic, real-valued probabilities. Since the input data is under a multinomial distribution, *cross-entropy* is applied as the error function. The *conjugate gradient* optimization framework is used to produce new values of the model parameters that will ensure convergence. The DBN can output binary and real output values (Salakhutdinov & Hinton, 2009). The binary output values are computed by adding deterministic Gaussian noise to the input of the output layer during fine-tuning. This way the output of the logistic sigmoid function at the output units will be close to 0 or 1 (Salakhutdinov & Hinton, 2009). The output values of the trained DBN are compared to a threshold[4] in order to decide the binary value. Distance metrics when using binary output vectors are much faster (Hinton & Salakhutdinov, 2010).

3. Simulations

We have performed model evaluations on the 20 Newsgroups dataset[5]. A dataset based on the *Wikipedia Corpus* is used to compare the DBN to the LDA model, since it contains labeled data. The *Issuu Corpus* has no labeled test set, so we compare the DBN to labels defined by a human perception of the topic distributions of Issuu's LDA model. The models are evaluated by retrieving a number of similar documents to a query document in the test set and average over all possible queries. This provides a fraction of the number of documents in the test set having similar documents in their proximity in the output space[6]. The number of neighbors evaluated are $1, 3, 7, 15, 31$, and 63. The evaluation is denoted the *accuracy measurement*.

[4] A threshold of 0.1 is applied due to a high number of output values lying closer to 0 than 1 (Hinton & Salakhutdinov, 2010).

[5] Refer to (Hinton & Salakhutdinov, 2010) for details.

[6] Euclidean distance and hamming distance is applied as distance metric on real valued and binary output vectors respectively.

The learning rate is set to $\epsilon = 0.01$, momentum $m = 0.9$ and a weight decay $\lambda = 0.0002$. The weights are initialized from a 0-mean normal distribution with variance 0.01. The biases are initialized to 0 and the number of epochs are set to 50. The pre-training procedure applies to *batch*-learning, where each batch represents 100 documents. For fine-tuning, larger batches of 1000 documents are generated. We perform three line searches for the conjugate gradient algorithm and the number of epochs is set to 50. The Gaussian noise for the binary output DBN, is defined as deterministic noise with mean 0 and variance 16 (Hinton & Salakhutdinov, 2010).

3.1. Model Evaluation

From Fig. 1 the DBNT performs in comparison to the model by Hinton and Salakhutdinov in (Hinton & Salakhutdinov, 2010). When comparing the real valued output DBN with a binary output DBN, we have observed that the accuracy measurements are very similar for a higher dimensional output vector (cf. Fig. 2). For the following simulations we have only considered real valued output vectors though. Fig. 3 shows that the DBN manages to find an internal representation of the documents that are better than the high dimensional input vectors.

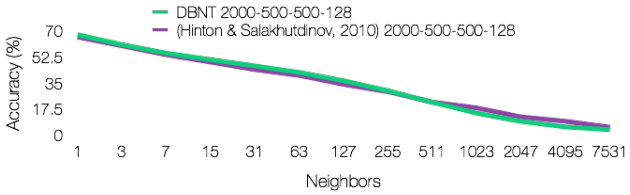

Figure 1. Accuracy measurements of the 2000-500-500-128-DBN with binary output units from (Hinton & Salakhutdinov, 2010) and a 2000-500-500-128-DBN with binary output units from the DBNT. The models are trained on the 20 Newsgroups dataset. **NB:** The results from (Hinton & Salakhutdinov, 2010) are read directly of the graph.

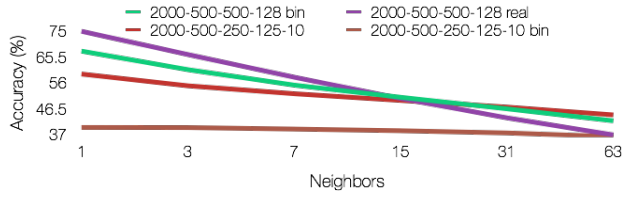

Figure 2. Accuracy measurements of two 2000-500-500-128-DBNs with binary output units and real valued output units trained on the 20 Newsgroups dataset.

3.2. Wikipedia Corpus

We have generated a dataset based on the Wikipedia Corpus. The dataset is denoted *Wikipedia Business* and contain articles from 12 subcategories from the *Business* category. We will use categories with a large pool of articles and a strong connectivity to the remaining categories of the Wikipedia Corpus. The categories are *administration*, *commerce*, *companies*, *finance*, *globalization*, *industry*, *labor*, *management*, *marketing*, *occupations*, *sales* and *sports business*. The Wikipedia Business dataset consists of 32843 documents split into 22987 (70%) training set documents and 9856 (30%) test set documents. Wikipedia Business provide an indication on how well the DBN and LDA model captures the granularity of the data within subcategories of the Wikipedia Corpus. In order to compare the DBN model to the LDA model, we have computed accuracy measurements on a 2000-500-250-125-10-DBN with real numbered linear output units and accuracy measurements on two LDA models, one with $K = 12$ topics and another with $K = 150$ topics. The accuracy measurement of the 2000-500-250-125-10-DBN is outperforming the two LDA models (cf. Fig. 4). The LDA model with $K = 12$ topics perform much worse than the DBN. The LDA model with a $K = 150$ topics perform well when evaluating 1 neighbor, but deteriorates quickly throughout the evaluation points. The DBN is the superior model for dimensionality reduction on the Wikipedia Business dataset. Its accuracy measurements are higher and the output is 10-dimensional compared to the 150-dimensional topic distribution of the LDA model with the lowest error.

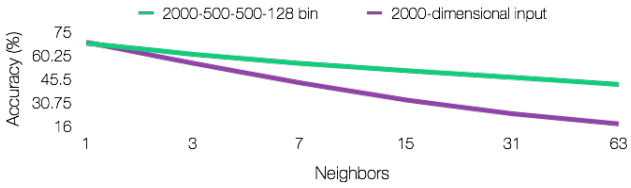

Figure 3. Accuracy measurements of the 2000-500-500-128-DBN with binary output units and the 2000-dimensional input vectors.

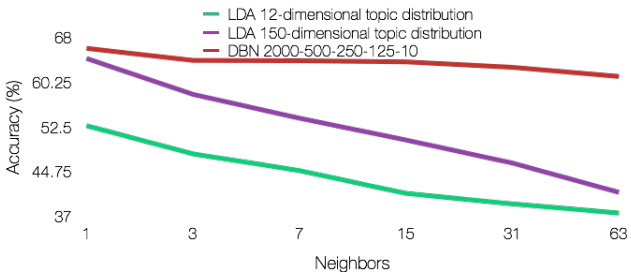

Figure 4. Accuracy measurements of two LDA models and a 2000-500-250-125-10 DBN.

We have computed accuracy measurements for: 2000-500-250-125-2-DBN, 2000-500-250-125-10-DBN, 2000-500-250-125-50-DBN and 2000-500-250-125-100-DBN (cf.

Fig. 5). It is evident that the DBN with a 2-dimensional output scores a much lower accuracy measurement, due to its inability to differentiate between the documents. When increasing the number of output units by modeling the 2000-500-250-125-50-DBN and the 2000-500-250-125-100-DBN, we see that they outperform the original 2000-500-250-125-10-DBN. Even though one DBN has an output vector twice the size of the other, the two evaluations are almost identical, which indicates saturation. Fig.

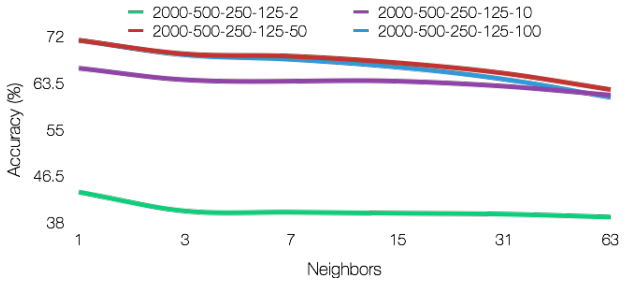

Figure 5. Accuracy measurements on DBNs with different number of output units.

6 shows how the DBN spreads the data in output space. Since PCA has its limitations it is not possible to visualize more categories unless an approach such as t-SNE is applied (van der Maaten & Hinton, 2008).

Figure 6. PCA on the 1st and 2nd principal components on the test dataset input vectors and output vectors from a 2000-500-250-125-10-DBN. **Left:** PCA on the 2000-dimensional input. **Right:** PCA on the 10-dimensional output.

3.3. Issuu Corpus

To test the DBN on the Issuu dataset we have extracted a dataset across 5 categories defined from Issuu's LDA model. The documents in the dataset belong to the categories *Business*, *Cars*, *Food & Cooking*, *Individual & Team Sports* and *Travel*. The training set contains 13650 documents and the test set contains 5850 documents. As mentioned, Issuu has applied labels to the dataset from the results of their LDA model with a 150-dimensional latent representation. In order to compare the models, we have performed accuracy measurements for the 2000-500-250-125-10-DBN on these labels (cf. Fig. 7). From the accuracy measurements it is evident how similar the results of the two models are. The big difference is that the DBN generates a 10-dimensional latent representation as opposed to the 150-dimensional latent representation of the LDA model.

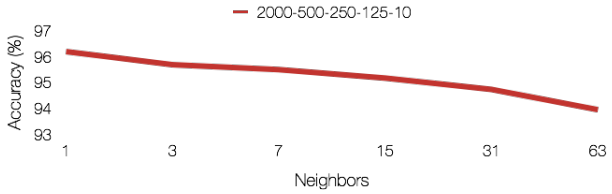

Figure 7. Accuracy measurements of a 2000-500-250-125-10-DBN on the labels defined on the basis of Issuu's LDA model.

When plotting the test dataset output vectors of the 2000-500-250-125-10-DBN for the 1st and 2nd principal component, it is evident how the input data is cluttered and how the DBN manages to spread the documents in output space according to their labels (cf. Fig. 8). By performing an analysis of the output space, categories such as *Business* and *Cars* are in close proximity to each other and far from a category like *Food & Cooking*.

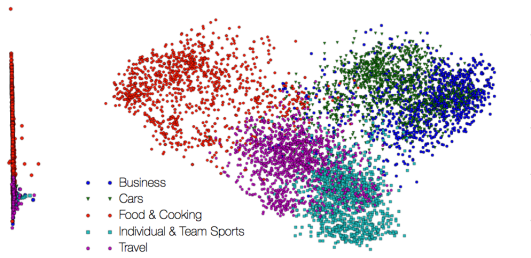

Figure 8. PCA on the 1st and 2nd principal components on the test dataset input vectors and output vectors from a 2000-500-250-125-10-DBN. **Left:** PCA on the 2000-dimensional input. **Right:** PCA on the 10-dimensional output.

Exploratory data analysis on the Issuu Corpus show how the 2000-500-250-125-10-DBN maps documents into output space[7]. We have chosen random query documents from different categories and retrieved 10 documents within the nearest proximity. When we query a car publication about an *SUV*, the 10 documents retrieved from output space are about cars. They are all publications promoting a new car, published by the car manufacturer. 7 out of the 10 related publications concern the same type of car. When comparing a query in output space with the same query in the high-dimensional input space, we see that the similar documents are more accurate in output space from a human perception.

[7]Due to copyright issues and the terms of services/privacy policy at Issuu the results are not visualized in this article.

4. Conclusion

On the Wikipedia and Issuu corporas we have shown how the DBN is superior compared to the proposed LDA models. The DBN manages to find a better internal representation of the documents in an output space of lower dimensionality. The low dimensionality of the output space results in fast retrieval of similar documents. A binary output vector of a larger dimensionality performs almost as good as a real valued output vector of equivalent dimensionality. Finding similar documents from binary latent representations is even faster.

References

Bengio, Yoshua. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

Blei, D.M., Ng, A.Y., and Jordan, M.I. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, March 2003.

Hinton, G.E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8): 1771–1800, August 2002.

Hinton, G.E. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade (2nd ed.)*, volume 7700 of *Lecture Notes in Computer Science*, pp. 599–619. Springer, 2012.

Hinton, G.E. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, Jul 2006.

Hinton, G.E. and Salakhutdinov, R. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3:74–91, 2010.

Řehůřek, Radim and Sojka, Petr. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

Salakhutdinov, R. and Hinton, G.E. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7): 969–978, July 2009.

Salakhutdinov, R. and Hinton, G.E. Replicated softmax: an undirected topic model. In *NIPS*, volume 22, pp. 1607–1614, 2010.

van der Maaten, Laurens and Hinton, Geoffrey. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, November 2008.