

Local and Global Sparsity for Deep Learning Networks

Long Zhang, Jieyu Zhao (✉), Xiangfu Shi, and Xulun Ye

Department of Computer Science, Ningbo University,
818 Fenghua Road, Ningbo 315211, China
{1401082013, zhao_jieyu}@nbu.edu.cn, xfu_shi@163.com,
1132548710@qq.com

Abstract. It has been proved that applying sparsity regularization in deep learning networks is an efficient approach. Researchers have developed several algorithms to control the sparseness of activation probability of hidden units. However, each of them has inherent limitations. In this paper, we firstly analyze weaknesses and strengths for popular sparsity algorithms, and categorize them into two groups: local and global sparsity. $L_{1/2}$ regularization is first time introduced as a global sparsity method for deep learning networks. Secondly, a combined solution is proposed to integrate local and global sparsity methods. Thirdly we customize proposed solution to fit in two deep learning networks: deep belief network (DBN) and generative adversarial network (GAN), and then test on benchmark datasets MNIST and CelebA. Experimental results show that our method outperforms existing sparsity algorithm on digits recognition, and achieves a better performance on human face generation. Additionally, proposed method could also stabilize GAN loss changes and eliminate noises.

Keywords: Sparsity · Regularization · Deep learning · GAN

1 Introduction

In the last decade, deep learning networks have an ambitious development. Its success has influenced not only many research fields but also our lives (self-piloting, language translation, etc.). From many experimental results, deep learning algorithms overcome most of traditional machine learning methods. Similar with human brain, deep learning networks contain multiple layers of neuron. Connections are built to link neurons between adjoining layers. Given an input, a deep learning network could abstract features through its deep architecture. It has been successfully applied to different fields like object recognition [1], human motion capture data [2, 3], information retrieval [4], speech recognition [5, 6], visual data analysis [7], and archives a wonderful performance.

There are several famous deep learning network structures. Deep belief network was firstly introduced by Hinton [4] in 2006, which started the age of deep learning. DBN brings researchers a new vision that a stack of generative models (like RBMs) is trainable by maximizing the likelihood of its training data. Unsupervised learning and generative models play key roles in this kind of network structure, and become more

and more important in the later deep learning network development. In 2012, Krizhevsky [8] applied a deep CNN on ImageNet dataset, and won the contest of ILSVRC-2012. GPU was implemented to accelerate training process, while “dropout” theory [9] was also implemented to solve “overfitting” problems. GAN (Generative Adversarial Nets) has drawn a lot of attentions from deep learning researchers. It was firstly introduced by Goodfellow [10] in 2014 and became a hot topic in recent two years [11]. Even Prof. LeCun said “Adversarial training is the coolest thing since sliced bread”.

Although deep learning networks have achieved a great success, without constraints on the hidden layers and units, it may produce redundant, continuous-valued codes and unstructured weight patterns [12]. Researchers have developed several useful constraints which improved networks performance greatly. Among them, adding sparsity regularization to networks has been proved as an efficient and effective approach.

This paper focus on the usage of sparsity regularization in deep learning networks. Section 2 presents related works. Section 3 categories different sparsity methods, lists their pros and cons. A novel sparsity regularization framework is introduced which could be customized to fit different networks structure. $L1/2$ regularization is first time applied with deep learning network. Section 4 presents two applications with our proposed method – Sparse DBN and Sparse GAN. Section 5 demonstrates experiments on two benchmarks, results support our proposal. Finally, this paper is conclude with a summary in Sect. 6.

2 Related Works

Bengio [13] said that if one is going to have fixed size representations, then sparse representations are more efficient in an information-theoretic sense, allowing for varying the effective number of bits per example. Traditional sparse coding learns low-level features for unlabeled data. However, deep learning networks provide a deep architecture with multiple layers. Network abstracts high-level features from lower ones. Applying sparse coding algorithm straightforwardly to build multiple levels of hierarchy is difficult. Firstly, building sparse coding on top of another sparse coding output may not satisfy the modeling assumption. Secondly, optimization is expensive [11, 14].

Luckily, there are several methods proposed to solve this problem. In 2008, Lee [15] developed a sparsity variant based on deep belief networks. A regularization term was added to loss function which penalized as the deviation of the expected activation of hidden units. Keyvanrad [14] applies a normal distribution on the deviation of the expected activation to control the degree of sparseness. The activation probability of hidden units get little penalty when they are close to zero or one. Similarly, Ji [12] implements L_1 -norm on the activation probability of hidden units together with rate distortion theory. According to Xu [16], the L_q regularization plays special important role on sparse modeling. However, it is a non-convex, non-smooth, and non-Lipschitz optimization problem which is difficult in general to have a thorough theoretical understanding and efficient algorithms for solutions. Somehow, studies in [16–18] have

resolved partially these problems. Krishnan and Fergus [17] demonstrated that $L_{1/2}$ and $L_{2/3}$ regularization are very efficient when applied to image deconvolution. Xu [18] ensured that $L_{1/2}$ plays a representative role among all L_q regularization with q in $(0, 1)$. Xu [16] also proved the superiority of $L_{1/2}$ over L_1 regularization.

Another approach for sparsity in deep learning networks is the choice of activation function. For a certain period, sigmoid and hyperbolic tangent functions were widely used in the literature. However in practice, training process has a slow convergence speed. Network may stuck at a poor local solution. Then, Nair [19] achieved a promising result by using rectifier linear unit (ReLU) in network. Compared with sigmoid or hyperbolic tangent functions, about 50% to 75% of hidden units are inactivate, and also with Leaky ReLU [20, 21] for higher resolution modeling.

3 Local and Global Sparsity

According to previous research results [12, 14, 15], applying sparsity terms to the activation probability of hidden units in deep learning networks could gain a much better performance. Some papers focus on individual hidden unit's probability, and others focus on the aggregation of them. We name the local sparsity for the prior ones, and the global sparsity for the after ones. However, deficiencies exist for each of them, it is inherent and difficult to solve. After a study of those methods, we found out that the weakness of local sparsity is just the strength of global sparsity, and vice versa. Therefore we propose a combined sparsity regularization, which could outcome each single ones.

3.1 Local Sparsity

The optimization problem of a sparse deep learning network is generally done by

$$\min f(x) + \lambda_1 L_{sparse} \quad (1)$$

where $f(x)$ is deep learning network's original loss function, λ_1 is a regularization constant, a tradeoff between "likelihood" and "sparsity" [14]. L_{sparse} is the sparsity regularization term.

Local sparsity methods in the deep belief network use a sparse variant or function to control average activation probability of hidden units. Different methods implement L_{sparse} in different way. In paper [15], the regularization term penalizes a deviation of the expected activation of hidden units from a fixed level p . Authors in [15] believe it could keep the "firing rate" of network neurons at a low value, so that network neurons are sparse. Given a training set $\{v^1, \dots, v^m\}$, regularization term is defined as

$$L_{sparse} = \sum_{j=1}^n \left| p - \frac{1}{m} \sum_{l=1}^m E[h_j^{(l)} | v^{(l)}] \right|^2 \quad (2)$$

where p is a constant which control the sparseness of hidden units, n is the number of hidden units, $E[\cdot]$ is the conditional expectation on hidden unit h_j . Since it is

implemented on RBM, therefore we can call it sparseRBM. This method achieved a great performance in 2008.

However, p is a fixed value. All hidden units share the same deviation level is logically inappropriate and crude. In paper [14], situation is improved by replacing with a normal function and a variance parameter to control the force degree of sparseness, so called normal sparse RBM (nsRBM). According to its authors, network parameters get little updates only when activation probability of hidden units are near to zero or one. That indicates hidden units with activation probability near one are important factors, therefor gradient penalizations are little. Given a training set $\{v^1, \dots, v^m\}$, regularization term is constructed as

$$L_{\text{sparse}} = - \sum_{j=1}^n f(k_j, p, \sigma^2) = - \sum_{j=1}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{k_j-p}{\sigma}\right)^2} \quad (3)$$

$$k_j = \frac{1}{m} \sum_{l=1}^m E[h_j^{(l)} | v^{(l)}] \quad (4)$$

where $f(\cdot)$ is a normal probability density function, k_j is the average of conditional expectation on hidden unit h_j , p is a constant, σ is the standard deviation. Same with sparseRBM, p controls the sparseness level, but changing σ can control the force degree of sparseness.

nsRBM can be seen as a “soft” version of sparseRBM. It softens the “hard” influence of the fixed p level, and achieves a better performance. However, fixed p level is still in use. Currently there is no good way to get the right level except try-and-error. Secondly, there is too many parameters. Finding a good combination is time-consuming. Thirdly, interactions cross hidden units are not considered.

3.2 Global Sparsity

Global sparsity in deep learning networks focus on the aggregation of activation probability of hidden units. It provides an overview on network’s sparseness. L_q -norm is generally applied as the regularization term. In deep learning networks, given a training set $\{v^{(1)}, \dots, v^{(m)}\}$, L_q regularization can be described as

$$L_{\text{sparse}} = L_q = \left(\sum_{j=1}^n |k_j|^q \right)^{\frac{1}{q}} \quad (5)$$

where q is a constant ($0 < q \leq 1$), k_j is the average of conditional expectation on hidden unit h_j , see Eq. (4). In [12], Ji has implemented L_1 regularization in deep belief networks, with its help the activation probability of hidden units has been greatly reduced near to zero. L_1 regularization is just an instance of L_q regularization. Recent studies [16, 18] shows $L_{1/2}$ regularization could generate more sparse solutions than it. Our experiments also prove that applying $L_{1/2}$ regularization in deep learning networks could achieve a better performance than L_1 .

Compare with local sparsity methods, global sparsity aggregates activation possibility of hidden units. It has no fixed p level ($p = 0$), and no additional parameters to adjust. Sparsity logic is easy to understand and simple to use. However, it has no control on the sparseness of each hidden unit. All hidden units have the same penalty mechanism. If a hidden unit with activation possibility near one indicates an “important” factor, global sparsity forces “less important” hidden units (between zero and one) become to zero. This is not a good behavior if we want to see more details from network results.

3.3 A Combined Solution

Problems of local and global sparsity are inherent, and difficult to resolve. Therefore we propose a combined solution, ideally local and global sparsity can complement with each other. The new sparsity regularization is constructed as

$$L_{sparse} = L_{local} + \lambda_2 L_{global} \quad (6)$$

where L_{local} indicates one of local sparsity methods, L_{global} indicates an instance of L_q regularization, λ_2 is a constant, a tradeoff between local and global sparsity. Experiments in Sect. 5 demonstrate this combined solution outperforms each single sparsity method mentioned above.

4 Sparse DBN and Sparse GAN

In this section, we implement proposed method in a deep belief network (DBN) and a generative adversarial network (GAN). The purpose for the sparse DBN is to compare with previous single sparsity methods [12, 14, 15]. Sparse GAN shows our proposed method could benefit data generations, stabilize loss changes and eliminate noises.

4.1 Sparse DBN

Deep belief network is consist of several RBMs. Therefore, sparse DBN means sparse RBM. RBM is a two layer, bipartite, undirected graphical model (see Fig. 1) with a set of visible units v , and a set of hidden units h . Visible units represent observable data, hidden units represent features captured from observable data. Visible layer and hidden layer are connected by a symmetrical weight matrix. There is no connection within the same layer.

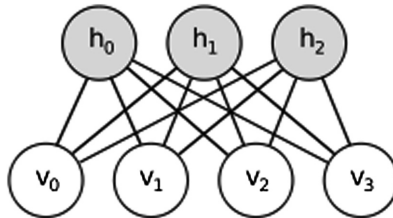


Fig. 1. Undirected graphical model of an RBM

If units are binary-valued, the energy function of RBM can be defined as

$$E(v, h) = - \sum_{i=1}^n \sum_{j=1}^m v_i w_{ij} h_j - \sum_{i=1}^n a_i v_i - \sum_{j=1}^m b_j h_j \quad (7)$$

where n and m are the total number of visible units and hidden units. v_i and h_j represent the value of visible neuron i and hidden neuron j . a_i and b_j are bias terms. w_{ij} represents the connection weight between i and j . The joint probability distribution for visible and hidden units can be defined as

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (8)$$

$$Z = \sum_v \sum_h e^{-E(v, h)} \quad (9)$$

where Z is the normalization factor. And the probability assigned to a visible vector v by the network, is obtained by marginalizing out hidden vector h

$$P(v) = \sum_h P(v, h) = \frac{1}{Z} \sum_h e^{-E(v, h)} \quad (10)$$

Parameters can be optimized by performing stochastic gradient descent on the log-likelihood of training data. Given a training set $\{v^1, \dots, v^m\}$, loss function of RBM can be defined as

$$\min \frac{1}{m} \sum_{l=1}^m \log(P(v^{(l)})) \quad (11)$$

Finally, by integrating Eqs. (1) and (11), we can get the loss function for proposed sparse RBM method as

$$\min \frac{1}{m} \sum_{l=1}^m \log(P(v^{(l)})) + \lambda_1 L_{local} + \lambda_2 L_{global} \quad (12)$$

Results in Sect. 5.1 shows our proposed method is efficient, and achieved the best recognition accuracy in MNIST dataset for all different number of training and testing samples. The performance overcomes each single local and global sparsity algorithms mentioned above.

4.2 Sparse GAN

GAN contains two models: generative and discriminative model. The objective of generative model is to synthesize data resembling real data, while the objective of

discriminative model is to distinguish real data from synthesized ones [22]. They both are multilayer perceptrons.

Given the training data set $\{x^1, \dots, x^n\}$, p_x is the data's distribution. Let z be a random vector sampled from p_z , generative model takes z as input and output synthesized data as $G(z)$. Input of discriminative model is a mix of training data and synthesized data, output is a single scalar as $D(x)$ or $D(G(z))$ depending on input's source, which demonstrates the probability of input data come from real training dataset. Ideally $D(x) = 1$ and $D(G(z)) = 0$. Network plays a two-player minimax game, and they can be trained by solving

$$\min_G \max_D E_{x \sim p_x} [\log D(x)] + E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (13)$$

If we denote the distribution of $G(z)$ as p_G , this minimax game has a global optimum for $p_G = p_x$ [1]. The training processes of generative and discriminative model are proceeded alternately. Parameters of generative model are fixed when updating discriminative model, vice versa. Be aware that discriminative model might learn faster than generative model. To keep in sync, we could train discriminative model k times, and then train generative model one time.

Due to the special mechanism of GAN, sparsity terms are added separately into the loss function of generative and discriminative models. The loss function for discriminative model is

$$\max E_{x \sim p_x} [\log D(x)] + E_{z \sim p_z} [\log 1 - D(G(z))] + \lambda_{d1} L_{local} + \lambda_{d2} L_{global} \quad (14)$$

Meanwhile the loss function for generative model is

$$\min E_{z \sim p_z} [\log 1 - D(G(z))] + \lambda_{g1} L_{local} + \lambda_{g2} L_{global} \quad (15)$$

Result in Sect. 5.2 shows, with the help of sparsity terms, quality of generated images are significantly improved. Moreover, the loss changes of generative and discriminative models are stabilized and noises in later iterations can be eliminated by our proposed method.

5 Experiments

5.1 MNIST

The MNIST digit dataset contains 60,000 training and 10,000 test images of 10 handwritten digits (0–9), each image with size 28×28 pixels [23]. The image pixel values are normalized between 0 and 1. In our experiment, we implement proposed sparsity method in a RBM network with 500 hidden units. Contrastive divergence and stochastic gradient descent are applied for sampling and parameter updating. Mini-batch size is set to 100. Sparsity term in nsRBM is selected as L_{local} regularization term,

while $L_{1/2}$ is selected as L_{global} regularization term. Original nsRBM contains two parameters: p level and standard deviation σ , in our experiment we use standard normal distribution ($p = 0, \sigma = 1$). We get the best performance when “tradeoff” constants are set to $\lambda_1 = 3, \lambda_2 = 0.005$.

Similar with [12, 14], we firstly train proposed method and several other sparsity regularization algorithms (RBM, sparseRBM [15], nsRBM [14], L_1 [12], $L_{1/2}$, and proposed method) on 20,000 images (2000 images per class), and then the learnt features are used as input for the same linear classifier. For the classification, we use 500, 1000 and 5000 images per class for training, 10,000 images for testing. We train 1000 epochs for all different methods. From Table 1, we can see our proposed method achieves the best recognition accuracy for all different number of training and testing samples. We demonstrate error rate changes for every 100 epochs in Fig. 2, and our method also achieves the best.

Table 1. Recognition error rate for training 100, 500, 1000 samples per class on MNIST dataset

Algorithms	100 Samples		500 Samples		1000 Samples	
	Training	Testing	Training	Testing	Training	Testing
RBM	10.80	11.47	5.84	6.38	4.54	5.06
sparseRBM	11.78	12.77	6.44	6.99	4.18	4.74
nsRBM	10.99	11.82	5.92	6.15	3.59	4.06
L_1	11.21	12.34	6.31	6.78	3.63	4.07
$L_{1/2}$	10.77	12.09	5.89	6.14	3.44	3.95
Ours	10.77	11.33	5.32	5.75	3.14	3.63

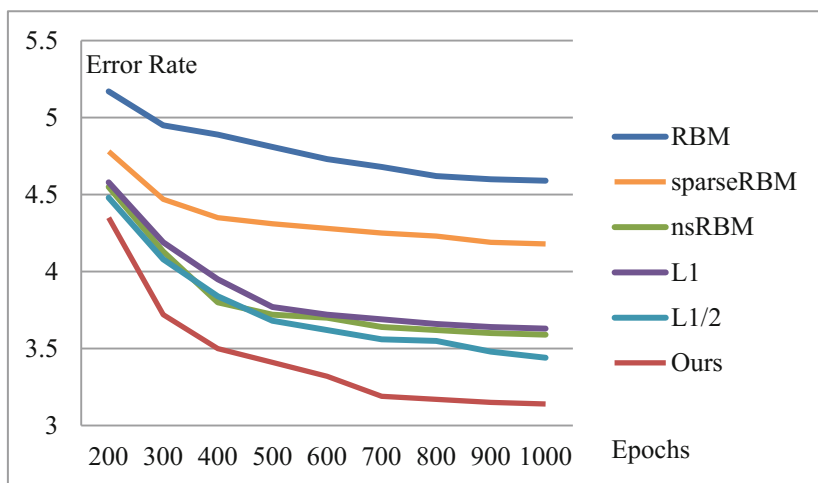


Fig. 2. Recognition error rate for every 100 epochs on MNIST dataset

5.2 CelebA

CelebA is a large-scale dataset with 202,599 number of face images [24]. Images cover large pose variations and background clutter. We implement our proposed sparsity algorithm in a deep convolutional generative adversarial network [11] with 5 convolutional layers. For the generative model, input is a 64-dimension vector which is randomly sampled from a uniform distribution. Filter numbers for each layers are 1024, 512, 256, 128, and 3, kernel size is 4×4 and stride is 2, output is a 64×64 synthesized human face. Structure of discriminative model is reverse expect output is a scalar. For training, mini-batch size is set to 64, and totally 3166 batches for one epoch.

Figure 3 shows some synthetic images generated by GAN (left side) and sparse GAN (right side). First row of images are sampled after epoch 1 iteration 500 of 3166, images generated by sparse GAN could describe face contours roughly. In the third row (sampled after epoch 1 complete), a human face could be easily recognized. Images generated at same steps by GAN could not achieve that. Last row of images are sampled after epoch 6, images on right side are obviously better.



Fig. 3. Synthetic images generated after different epochs by (a) GAN (b) sparse GAN.

Figures 4 and 5 demonstrate that sparsity terms could also stabilize loss changes while GAN playing minimax game. Moreover, noises in later iterations are surprisingly eliminated by sparse GAN. This is beyond our expectation.

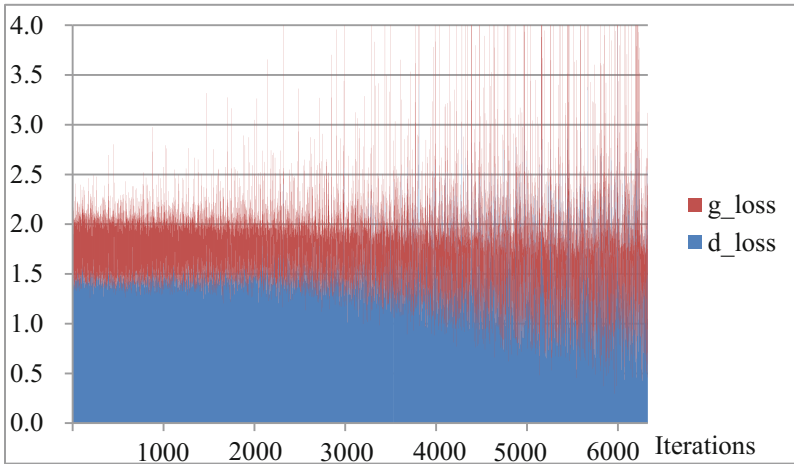


Fig. 4. Loss values of discriminator and generator models in GAN

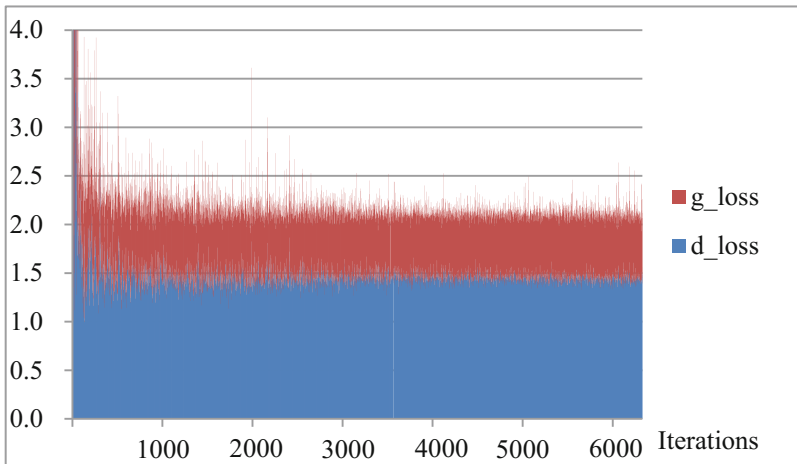


Fig. 5. Loss values of discriminator and generator models in sparse GAN

6 Conclusion

We studied popular sparsity algorithms and categorized according to their mechanism. After analyze their weaknesses and strengths, we presented a combined solution for local and global sparsity regularization. Two deep learning networks (DBN and GAN)

were implemented to verify proposed solution. Additionally, experiments on two benchmarks showed promising results of our method.

Acknowledgments. This work was supported by National Natural Science Foundation of China under Grant 61571247, the National Natural Science Foundation of Zhejiang Province under Grant LZ16F030001, and the International Cooperation Projects of Zhejiang Province under Grant No. 2013C24027.

References

1. Hinton, G.: To recognize shapes, first learn to generate images. *Prog. Brain Res.* **165**, 535–547 (2007)
2. Taylor, G., Hinton, G., Roweis, S.: Modeling human motion using binary latent variables. In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 1345–1352 (2007)
3. Taylor, G., Hinton, G.: Factored conditional restricted Boltzmann machines for modeling motion style. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1025–1032 (2009)
4. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
5. Mohamed, A., Dahl, G., Hinton, G.: Acoustic modeling using deep belief networks. *IEEE Trans. Audio Speech Lang. Process.* **20**(1), 14–22 (2012)
6. Hinton, G., Deng, L., Yu, D.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Sig. Process. Mag.* **29**(6), 82–97 (2012)
7. Liu, Y., Zhou, S., Chen, Q.: Discriminative deep belief networks for visual data classification. *Pattern Recogn.* **44**(10), 2287–2296 (2011)
8. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: *International Conference on Neural Information Processing Systems*, pp. 1097–1105 (2012)
9. Hinton, G., Srivastava, N., Krizhevsky, A.: Improving neural networks by preventing co-adaptation of feature detectors. *Comput. Sci.* **3**(4), 212–223 (2012)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M.: Generative adversarial nets. In: *International Conference on Neural Information Processing Systems*, pp. 2672–2680 MIT Press (2014)
11. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: *4th International Conference on Learning Representations* (2016)
12. Ji, N., Zhang, J.: A sparse-response deep belief network based on rate distortion theory. *Pattern Recogn.* **47**(9), 3179–3191 (2014)
13. Bengio, Y.: *Learning Deep Architectures for AI*. Foundations and Trends in Machine Learning (2009)
14. Keyvanrad, M., Homayounpour, M.: Normal sparse deep belief network. In: *International Joint Conference on Neural Networks*, pp. 1–7 (2015)
15. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area V2. In: *Advances in Neural Information Processing Systems*, pp. 873–880 (2008)
16. Xu, Z., Chang, X., Xu, F., Zhang, H.: L1/2 regularization: a thresholding representation theory and a fast solver. *IEEE Trans. Neural Networks Learn. Syst.* **23**(7), 1013–1027 (2012)

17. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-Laplacian priors. In: *Advances in Neural Information Processing Systems*, pp. 1033–1041. MIT Press, Cambridge (2009)
18. Xu, Z., Guo, H., Wang, Y., Zhang, H.: Representative of $L_{1/2}$ regularization among L_q ($0 < q \leq 1$) regularizations: an experimental study based on phase diagram. *Acta Automatica Sinica* **38**(7), 1225–1228 (2012)
19. Nair, V., Hinton, G.: Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814 (2010)
20. Maas, A., Hannun, A., Ng, A.: Rectifier nonlinearities improve neural network acoustic models. In: *Proceedings of the 30th International Conference on Machine Learning*, vol. 30, no. 1 (2013)
21. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint [arXiv:1505.00853](https://arxiv.org/abs/1505.00853) (2015)
22. Liu, M., Tuzel, O.: Coupled generative adversarial networks. In: *Advances in Neural Information Processing Systems* (2016)
23. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist>. Accessed 29 May 2017
24. Large-scale CelebA Dataset. <http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>. Accessed 29 May 2017