Proceedings of the 2017 International Conference on Wavelet Analysis and Pattern Recognition, Ningbo, China, 9-12 July, 2017

EXPERIMENTAL STUDY ON STACKED AUTOENCODER ON INSUFFICIENT TRAINING SAMPLES

JIN XUE, PATRICK P. K. CHAN, XIAN HU

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
E-MAIL: csxuejin@mail.scut.edu.cn, patrickchan@mail.scut.edu.cn, huxian_scut@163.com

Abstract:

Many studies have shown that deep learning outperforms traditional machine learning methods in many applications. To prevent overfitting, a huge number of training samples is usually required in training process of deep learning. However, collecting such a large dataset is time-consumed and costly. Recently, several methods have been proposed to effectively learn the models with a limited number of training samples. In this paper, we experimentally evaluate the data augmentation and the transfer learning methods on the MNIST dataset with stacked autoencoders. The experiment results suggest that the transfer learning method can improve the performance of the SAE when training samples are insufficient.

Keywords:

Deep learning; Small dataset; Stacked autoencoder

1 Introduction

Deep learning outperforms traditional machine learning methods on challenging tasks, for example, speech recognition[11] and object detection[12]. Unlike traditional machine learning methods, Deep Neural Network (DNN) consists of a hierarchy of layers to extract high-level representations from the raw input. The output of each hidden layer can be regarded as feature extraction of the input data[13]. The successive layers construct high-level representation from low-level representation of original data. This process is called feature learning. After good feature representation is obtained, the classification with satisfying performance can be achieved. However, DNNs generally have a large number of parameters due to its deep architecture. The large freedom provides DNNs powerful ability to classify a very complicated task, but also causes overfitting easily. Many studies show that the performance of DNNs is worse than other classifiers when training samples are insufficient.

Collecting samples is usually costly and time-consumed. Even worse, the number of samples are limited in some applications. For example, medical related applications usually contain limited number of available patient samples. Due to these reasons, improving the performance of deep learning on a small dataset is important. Data augmentation is one solution for the problem of insufficient training samples. It is one of widely used techniques in deep learning to prevent overfitting [3, 5, 6]. The basic idea of data augmentation is to use new samples to enlarge the small dataset. New samples are generated from the original samples by applying transformations. Besides data augmentation, transfer learning which transfers knowledge from other similar datasets for the specific problem to the original one is also used to provide additional information to DNNs [7].

In this paper, we will analyze whether data augmentation and transfer learning methods provides additional information to DNNs when the training samples are insufficient. Noise is added for data augmentation method, while the samples of a related problem is added to the training set in the transfer learning method. Stacked autoencoder, which is one of the most popular DNNs, is chosen as the classifier. The methods are evaluated experimentally using MNIST dataset .

The rest of the paper is organized as follows. Section 2 introduces the concepts related in this study. The data augmentation and the transfer learning methods used in the study are described in Section 3, while Section 4 presents and discusses the experimental results. Finally, the conclusion is discussed in Section 5.

978-1-5386-0412-0/17/$31.00 ©2017 IEEE

2 Related Work

2.1. Stacked Autoencoder

There are several types of deep learning models. For example, convolution neural networks [3], recurrent neural networks [4] and deep belief networks [14]. In this paper, we will focus on the stacked autoencoder (SAE) [15, 16], which is one of the most commonly used architectures in deep learning. SAE [15, 16] is a neural network which consists of multiple autoencoders. The training process of SAE can be divided into two phases: feature learning and fine-tuning. In feature learning, an unsupervised learning algorithm is used to pre-train the network layer by layer[17, 18]. The target output value of each autoencoder is set as its input. An autoencoder approximates an identity function of the input, i.e. $h_{W,b}(x) \approx x$, in order to learn the reconstruction ability. The training process of an autoencoder minimizes the objective function:

$$\frac{1}{m}\sum_{i=1}^{m} L(h_{W,b}(x^{(i)}), x^{(i)}) \tag{1}$$

where $x^{(i)}$ is the ith training sample and $L(h_{W,b}(x^{(i)}), x^{(i)})$ is the loss function between the network output and the input data.

After the pre-training phase, the network parameters are then fine-tuned by supervised learning methods. A softmax classification layer is added after the last hidden layer, taking the output of the last hidden layer as its input. Then SAE is trained by backpropagating minimizing the classification error.

2.2. Data Augmentation

In computer vision, data augmentation is widely used to improve the generalization ability of a model [3]. Overfitting is most likely to occur when the number of training samples is too small in comparison with the excessive number of network parameters. One simple solution of overfitting problem is to increase the size of a dataset. Data augmentation generates new samples from original samples by transformations, for example, translation, rotation and flipping. Data augmentation is also useful in feature learning. By applying transformations like translation and rotation, the network learns different variation of features, *i.e.,* the trained network is invariant to transformations. Consequently, the prediction ability of the trained network increases. In this paper, we will investigate the existing data augmentation methods and see whether they can be used to improve the performance of the model.

2.3. Transfer Learning

Transfer learning focuses on transferring the knowledge from the model in other similar tasks [7, 18]. In practice, a training set with sufficient samples for a target task is sometimes expensive in terms of collection time and cost. The knowledge of a model learned from some source tasks similar to the target task increases more generalization ability on the target task. There are several approaches to transfer the knowledge. Instance transfer learns the samples of source task with different values of weights [8]; feature representation transfer aims to find a feature representation that can reduce the difference between the distribution of data of the source and the target task[9], and model transfer discovers parameters that can be shared in both the source and the target domain [10].

3 Deep Learning with Insufficient Training Samples

Insufficient training samples may cause overfitting easily for SAE due to its complexity. Two methods are focused in this study, which are data augmentation and transfer learning. This section introduces the methods we evaluate in this paper in detail.

A training set is enlarged by inserting transformed samples while the label information is preserved. The transformation includes translation, scaling and rotation. The type of transformations is carefully chosen in order to avoid the distributions of classes are not changed significantly. Since our study focuses on the handwritten digit dataset, rotation should not be considered, otherwise, the concepts of classes may be confused. For example, the number 6 may become the number 9. Translation and scaling are not suitable too since the pre-processed handwritten digit images are located in the middle and also their size are similar. As a result, a general transformation method is chosen, *i.e.,* random noise is added to samples. The parameter of the method is the degree of noise.

On the other hand, the simple transfer learning technique is used. The samples of the datasets related to the target task are directly added to the training set. The importance of each training sample is equal in training, *i.e.,* no weight is considered. In our study, several datasets with different similairty degree to the target task are used in the transfer learning.

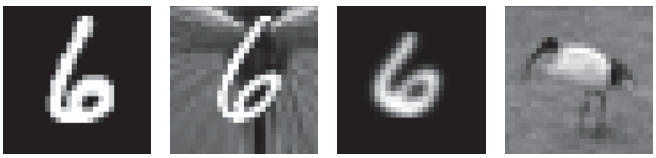
4 Experiment

In this section, we will show the experiment results of various data augmentation and transfer learning methods. We firstly introduce the datasets and experimental setting. The

performance of SAE trained with a different sized datasets is evaluated to determine when the training set is insufficient for training a classifier with good generalization ability. The small dataset is then enlarged by data augmentation and transfer learning methods described in Section 5. The enlarged datasets are used to train new SAE models. We evaluate the improvement of the data augmentation and the transfer learning methods to SAE with insufficient training samples finally.

4.1 Dataset Description

Several image datasets are considered in the experiments. They are MNIST, MNIST with background, United States Postal Service (USPS), and Caltech 101 datasets. MNIST has been used in many deep learning related studies. We will use MNIST as the major dataset in our experiment. MNIST with background contains samples made by adding random background images to the original MNIST images. USPS consists of black and white handwritten digit images obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service. USPS is similar to MNIST except that the images in the USPS dataset has a size of 16×16. In the experiment, black padding is added to the USPS images to get them into the same shape as the MNIST images. The last one is Caltech 101. It contains objects instead of handwritten digits. There are 101 categories with 40 to 800 images per category. The sizes of the images are roughly equal to 300×200. They are resized to match the size of the MNIST images. 10 categories are selected from the Caltech 101 dataset. Each selected category will be assigned a new label from 0 to 9 as the MNIST dataset does. Caltech 101 is very different from the MNIST dataset. Sample images of these datasets are shown in Fig. 1.



(a) (b) (c) (d)

FIGURE 1. Sample images from (a) the MNIST dataset (b) the MNIST with background dataset (c) the USPS dataset (d) the Caltech 101 dataset

4.2 Experimental Setting

SAEs with six different architectures are used. The number of hidden layers in the SAE varies from 1 to 3, where each layer has 100 or 500 neurons. SAEs are trained using the algorithm in section 2.1. 60000 and 10000 images are randomly selected from MNIST as training and test set. The experiments are repeated 5 times. In each experiment, a subset is then drawn from the training set randomly to generate an insufficient training set. The trained model is evaluated by test the set. In the transfer learning experiment, the images of the USPS dataset are padded to the same size of MNIST images. The images of the Caltech 101 dataset are scaled to the size of MNIST images.

4.3 Results and Discussion

4.3.1 SAE on small datasets

The results of SAE trained only using small datasets without other technique to improve the performance are shown in Figs. 2.

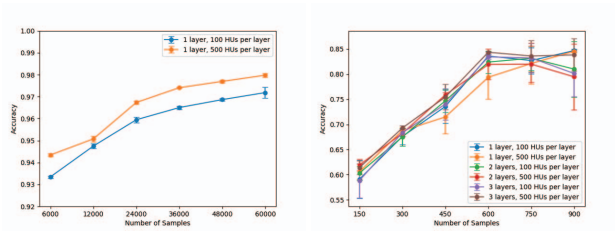


FIGURE 2. Baseline accuracy of SAEs on a large dataset without data augmentation

The x-axis represents the number of training samples, while the average accuracy of the model on the test set in 5 independent runs is shown in the y-axis. Each line is SAE with different architectures. From the left figure in 2, when at least 6000 (10% of the full set) training images are given, all SAEs are able to achieve high accuracy, *i.e.,* higher than 93%. However, when the size of the training set drops to $[150, 600]$, the accuracy of SAEs decreases to lower than 85% significantly, shown in the right figure. As a results, in the rest of the experiment, the size of training set is set to 150. Without any other information, the accuracy of SAEs is lower than 64%.

4.3.2 SAE with Data Augmentation by Adding Noise on small datasets

We firstly evaluate the data augmentation method of adding noise on MNIST. The binary masking noise is used. A portion of pixels in the training images are set to zero to generate new images. Three degrees of noise (30%, 50% and 70%) are chosen. The training dataset is combined by 150 original images and a number of generated images. This dataset is used in both

pre-training and fine-tuning phases of the model. The accuracy of the models on the MNIST test set is given in in Fig. 3.

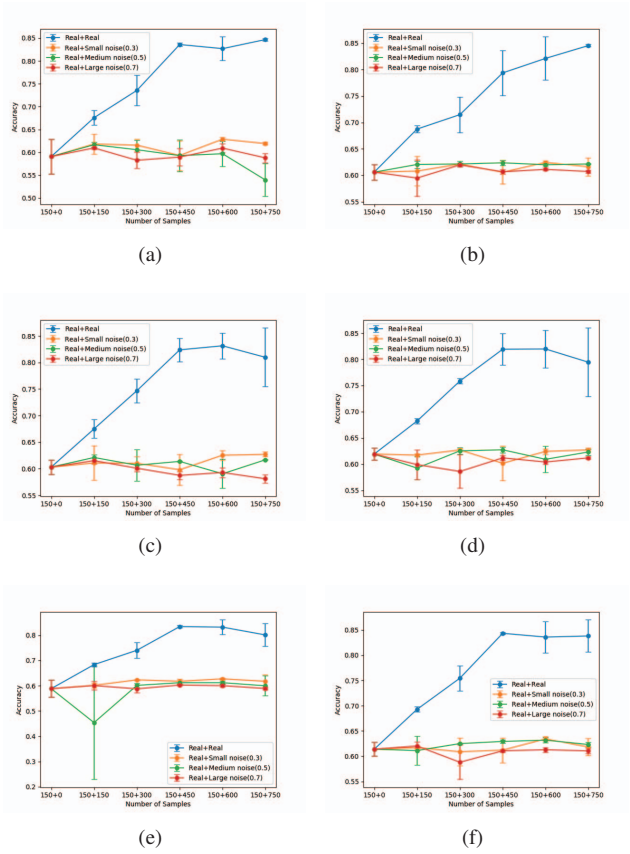


FIGURE 3. Accuracy of SAEs trained on 150 MNIST images plus samples generated by adding noise. The configurations of SAEs are (a) 1 layer with 100 hidden units per layer (b) 1 layer with 500 hidden units per layer (c) 2 layer with 100 hidden units per layer (d) 2 layer with 500 hidden units per layer (e) 3 layer with 100 hidden units per layer (f) 3 layer with 500 hidden units per layer

The x-axis represents the number of real samples plus the number of added samples. "Real" indicates the samples from the same distribution of the target task, *i.e.,* MNIST. "Real+Small/Medium/Large" indicates the strength of noise in data augmentation method. "Real+Real" is the benchmark for comparison. All samples in this method is chosen from MNIST dataset. There are no significant accuracy gains after adding noise to generate new images. The results show that the model performance is not improved by simply adding noise to make a larger training set. Adding noise to training set may improve the performance in some applications. However, it is a kind of application dependent, but not a general method.

4.4 SAE with Transfer Learning on small datasets

Images from the datasets except MNIST are added to the training set containing with 150 original MNIST images to make a larger dataset. The large dataset is then fed to the SAEs in both pre-training and fine-tuning phases. The accuracy of the SAE is shown in Fig. 4.

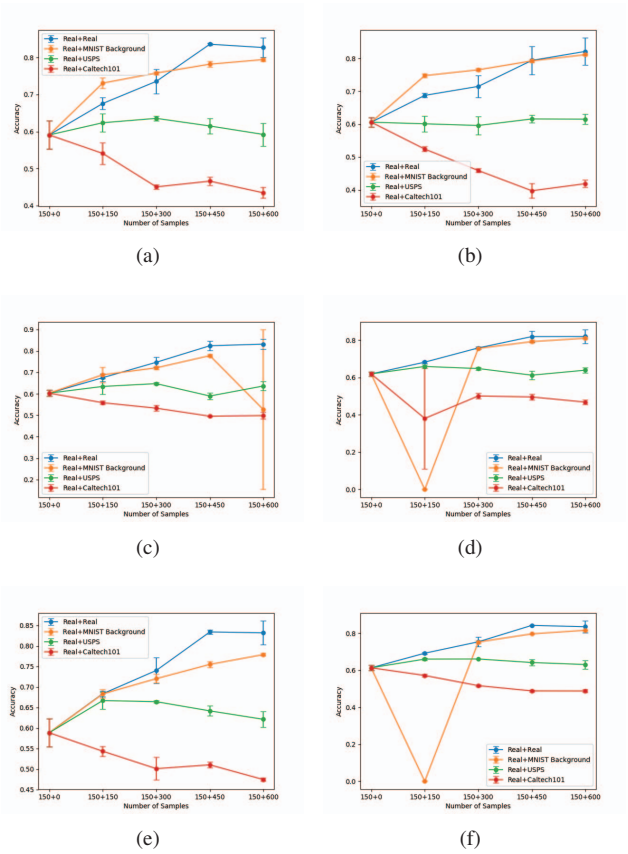


FIGURE 4. Accuracy of SAEs trained on 150 MNIST images plus samples from other datasets. The configurations of SAEs are (a) 1 layer with 100 hidden units per layer (b) 1 layer with 500 hidden units per layer (c) 2 layer with 100 hidden units per layer (d) 2 layer with 500 hidden units per layer (e) 3 layer with 100 hidden units per layer (f) 3 layer with 500 hidden units per layer

Similar to previous experiment, "Real+Real" is the ideal case, in which all samples are from MNIST, for comparison. "Real+MNIST Background/USPS/Caltech101" represents the transfer learning using other datasets. The results show that the performance of SAE can be improved if the knowledge is transfer from the samples similar to MNIST ones. For example, the accuracy of Real+MNIST Background is close to Real+Real.

Moreover, the samples which are more similar to MNIST increase accuracy more. The accuracy of the ones using MNIST background is higher than the ones of using USPS and Caltech 101, and Real+Caltech 101 achieves the worst performance.

4.5 SAE with Transfer Learning in pre-training or fine-tuning on small datasets

The training process of SAE is divided into pre-training and fine-tuning phases. This experiment applies the transfer learning into only one phase. The experimental results of SAE with transfer learning in pre-training and fine-tuning is shown in Fig. 5 and Fig. 6 respectively.

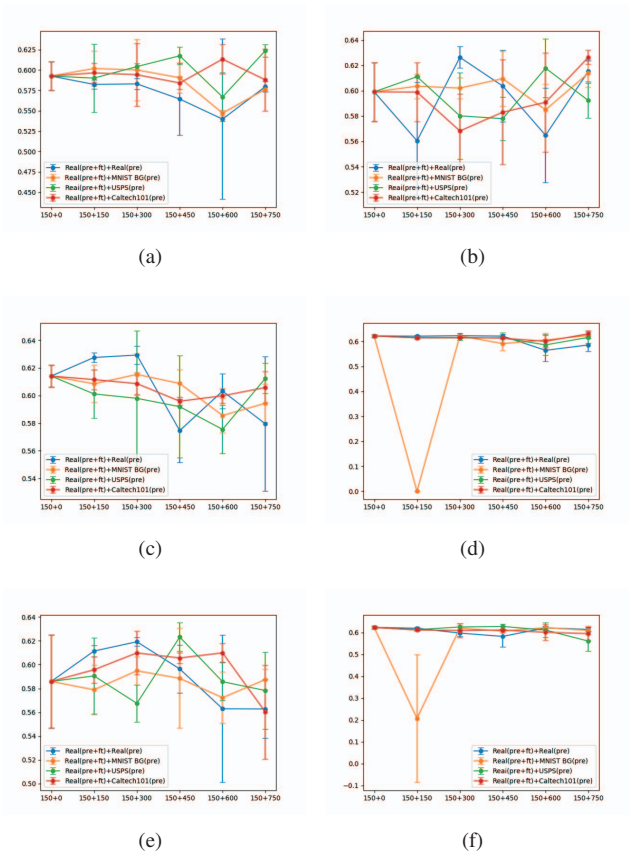


FIGURE 5. Accuracy of SAEs trained on 150 MNIST images plus samples from other datasets used only in pre-training phase. The configurations of SAEs are (a) 1 layer with 100 hidden units per layer (b) 1 layer with 500 hidden units per layer (c) 2 layer with 100 hidden units per layer (d) 2 layer with 500 hidden units per layer (e) 3 layer with 100 hidden units per layer (f) 3 layer with 500 hidden units per layer

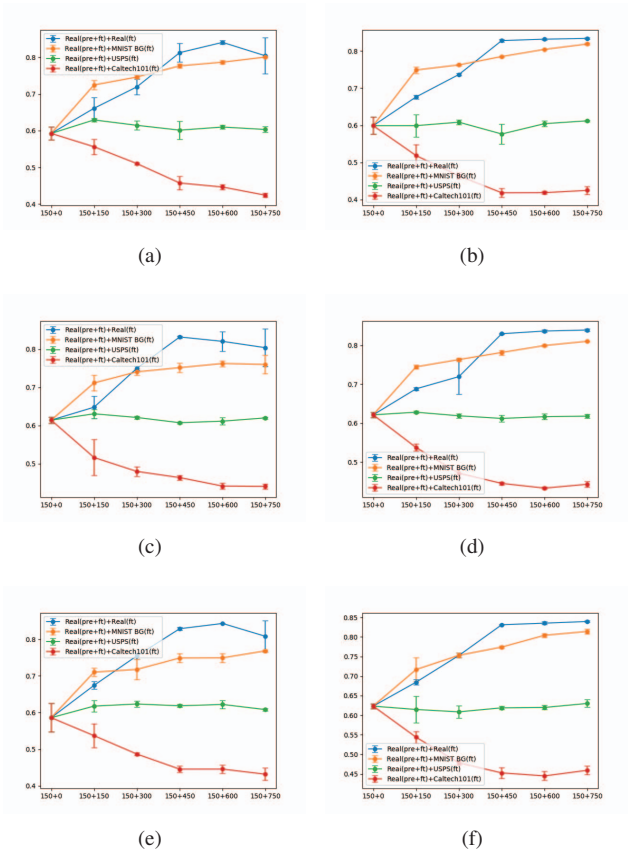


FIGURE 6. Accuracy of SAEs trained on 150 MNIST images plus samples from other datasets used only in fine-tuning phase. The configurations of SAEs are (a) 1 layer with 100 hidden units per layer (b) 1 layer with 500 hidden units per layer (c) 2 layer with 100 hidden units per layer (d) 2 layer with 500 hidden units per layer (e) 3 layer with 100 hidden units per layer (f) 3 layer with 500 hidden units per layer

The results suggest that applying transfer learning to the pre-training phase does not improve the performance in SAE. Even MNIST samples are added in the pre-training stage, the accuracy of SAE does not increase. The models still achieve the same performance as the one only using 150 training samples. On the other hand, the performance of SAE is more closely related to the number of samples used in the fine-tuning phase, shown in Fig. 6. Transferring samples from USPS and Caltech 101 cause the accuracy of SAE drop. However, when samples of MNIST with background are used, performance of SAE is improved. These results show that providing more information in pre-training phase is not useful in improving the ability of SAE. Fine-tuning phase is more important in training process.

5 Conclusion

In this study, several methods improved the learning of SAE on insufficient samples are evaluated experimentally. The experiment shows that when the number of samples is insufficient, the performance of SAE drops greatly. Data augmentation by adding noise does not efficiently improve the performance of SAE with insufficient training samples. However, transferring knowledge from the tasks which are more similar to the target task provides useful knowledge. This study shows that negative impact may be obtained when unsuitable samples are used in transfer learning. Transferring knowledge from suitable samples is an important problem. In future, the selection on suitable dataset for transfer learning should be further studied.

6 Acknowledgement

This work is supported by a National Training Program of Innovation and Entrepreneurship for Undergraduates of China(No. 201710561129).

References

[1] Y. Bengio, "Learning deep architectures for AI", in *Foundations and Trends in Machine Learning*, 2013.

[2] Bengio, Yoshua, and Olivier Delalleau. "On the expressive power of deep architectures." *Algorithmic Learning Theory*. Springer Berlin/Heidelberg, 2011.

[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in *International Conference on Neural Information Processing Systems*, 2012.

[4] A. Graves, A. Mohamed, G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks". in *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 38. . 10.1109/ICASSP.2013.6638947.

[5] X. Cui, V. Goel and B. Kingsbury, "Data Augmentation for deep neural network acoustic modeling", in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, 2014, pp. 5582-5586.*

[6] S. C. Wong, A. Gatt, V. Stamatescu, M. D. McDonnell, "Understanding data augmentation for classification: when to warp?", *arXiv:1609.08764*, 2016.

[7] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[8] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning", in *Proceedings of the 24th International Conference on Machine Learning, Corvalis, Oregon, USA, June 2007, pp. 193200.*

[9] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data", in *Proceedings of the 24th International Conference on Machine Learning, Corvalis, Oregon, USA, June 2007, pp. 759766.*

[10] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine", in *Proceedings of the 21st International Conference on Machine Learning. Banff, Alberta, Canada: ACM, July 2004.*

[11] G. E. Hinton, L. Deng, D. Yu, et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". *IEEE Signal Processing Magazine*, 2012, 29(6):82-97.

[12] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *arXiv:1506.01497*, 2015.

[13] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, Aug. 2013.

[14] Hinton G E, Osindero S, Teh Y W. "A fast learning algorithm for deep belief nets". *Neural computation*, 2006, 18(7): 1527-1554.

[15] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of Machine Learning Research* 11.Dec (2010): 3371-3408.

[16] Cho, KyungHyun. "Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images." *International Conference on Machine Learning*, 2013.

[17] Bengio Y, Lamblin P, Popovici D, et al. "Greedy layer-wise training of deep networks". *International Conference on Neural Information Processing Systems*. MIT Press, 2006:153-160.

[18] Bengio, Yoshua. "Deep learning of representations for unsupervised and transfer learning." *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. 2012.

[19] Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[20] Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015