LEARNING DEEP REPRESENTATIONS USING CONVOLUTIONAL AUTO-ENCODERS WITH SYMMETRIC SKIP CONNECTIONS

*Jian-Feng Dong, Yuan-Zhu Gan, Xiao-Jiao Mao, Yu-Bin Yang**

Nanjing University
163 Xianlin Avenue, Nanjing, China

Chunhua Shen

The University of Adelaide
Adelaide, SA 5005, Australia

ABSTRACT

Convolutional neural networks (CNNs) have shown their power on many computer vision tasks. However, there are still some limitations, including their sensitivity to weight initialization and dependency to large scale labeled data. In this paper, we try to address these two problems by proposing a simple yet powerful CNN based denoising auto-encoder network which can be trained end-to-end in an unsupervised manner. The network architecture we use is a fully convolutional auto-encoder with symmetric encoder-decoder connections. The proposed method can not only reconstruct clean images from corrupted ones, but also learn abstract image representation through the reconstruction training. The encoder part of the network can be a good all-convolution network for classification. With the help of unsupervised pre-training, it achieves very competitive results even without extra unlabeled data. Further more, we show experimentally that our network also performs well in semi-supervised learning tasks.

***Index Terms*—** Unsupervised Pre-training, Semi-supervised Learning, Denoising Auto-encoders, Deep Learning

1. INTRODUCTION

Unsupervised pre-training using denosing auto-encoders [1] was a critical technique to train deep neural networks ten years ago. As non-saturate activations [2], properer initialization [3, 2] and sufficiently large labeled data [4] have been successfully used, it is now possible to train very deep convolutional neural networks (CNNs) from scratch.

Yet there are still two problems: 1) Most commonly used initialization methods randomly initialize parameters, tuning it for a specific dataset needs lots of work. 2) In many computer vision fields, there are limited labeled images but plentty of unlebeled ones, which makes it is very hard to train a deep CNN in a purely supervised way.

*This work is supported by the Natural Science Foundation of China (Grant No. 61673204), State Grid Corporation of Science and Technology Projects (Funded No. SGLNXT00DKJS1700166), the Program for Distinguished Talents of Jiangsu Province, China (Grant No. 2013-XXRJ-018), and the Fundamental Research Funds for the Central Universities (Grant No. 020214380026).

A obvious solution for these problems is to simply transfer the idea of unsupervised pre-training using auto-encoders to CNN. Unfortunately, As we will show in experiments, simply building auto-encoder network with a stack of convolutional layers can hardly learn abstract features from unlabeled data and cannot provide good data-driven initialization. Hence in this paper, inspired by shortcut strategies used in [5, 6] and other works to learn abstract features from image reconstruction [7, 8, 9], we propose a novel method to learn abstract representations from unlabeled data based on a network composing of a stack of convolutional auto-encoders with symmetric shortcut connections. Starting from the auto-encoder network, we develop a concise all-convolutional network for classification which achieves fairly competitive accuracy compared to state-of-the-art methods like ResNet [5, 10] but very simple to be built and trained. We show experimentally, when no extra unlabeled data are available, the proposed method can perform unsupervised pre-training to search a proper network initialization and achieve competitive classification results after fine-tuning. Furthermore, when there are extra unlabeled data, the proposed method can improve semi-supervised learning to achieve better results, and scale well to large-scale unlabeled datasets as well.

The contributions of this work are:

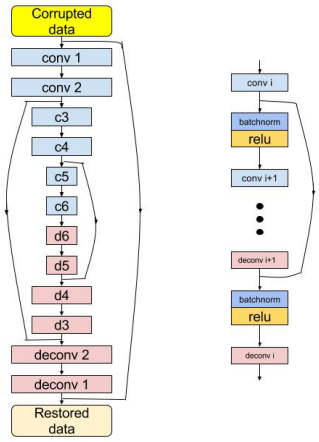
- A convolutional auto-encoder network with symmetric skip connections to learn abstract representations and find good initialization from training data.
- A new method to learn from limited labeled data and extra unlabeled data.
- A concise all-convolution classification network which can achieve very competitive results and is easy to train.

2. SYMMETRICALLY CONNECTED CONVOLUTIONAL AUTO-ENCODERS

2.1. Architecture

The basic architecture of our network is a fully convolutional auto-encoder. The encoder part is a chain of convolutional layers, and the decoder part is a chain of decovolutional layers symmetric to the convolutional ones. The corresponding

encoder and decoder layers are connected by shortcut connections. Each convolutional/deconvolutional layer is followed by a Batch Normalization [11] layer and a ReLU non-linearity layer. An illustration is shown in Figure 1.



(a) Network architecture (b) Shortcut connection

Fig. 1: Network architecture and the topology of a single shortcut connection.

Encoder The encoder acts as feature extractor. We use 3×3 convolutional layers following VGGNet [12]. As in [13], down-sampling is conducted by convolution with stride 2 instead of pooling. With a proper fully-connected layers added, encoder part also serves as our classification network.

Decoder The decoder takes features learned by the encoder and reconstructs the "clean" images. We use deconvolution as our decoder unit, which is often referred as learnable up-sampling operation in tasks such as semantic segmentation [14] and image generation [15]. Since we use convolution with strides as learnable down-sampling operation in encoder, it is important to also make the up-sampling learnable. The layers in the decoder are organized symmetrically to the ones in the encoder. For a specific deconvolutinal layer, the size of its input/output is equal to the output/input of its corresponding convolutional layer to achieve pixel-wise correspondence for shortcut connections.

Shortcut Connections In our network, shortcut connections are used to pass feature maps forwardly. The feature maps from a shortcut connection and the connected deconvolutional layer are then added element-wise. For a single shortcut connection, the connecting strategy follows the pre-activation version of deep residual network [16] as shown in Figure 1b. But at the network scale, instead of block-by-block as in ResNet, our shortcut connections are linked symmetrically as shown in Figure 1a.

2.2. Training Pipeline

We follow the unsupervised pre-training and fine-tuning pipeline to learn representations from unlabeled data, and then transfer them to supervised tasks. In contrast to the vanilla stacked denoising auto-encoders [1], our pre-training is conducted end-to-end instead of greedy layer-wise. During pre-training, for a clean image x, the input of the network is its corrupted version $\tilde{x}$. The output is the reconstruction represented as $f(\tilde{x})$. We train the network end-to-end by minimizing the mean square error between x and $f(\tilde{x})$. Although for masking noise, most previous works [1, 7] also use a masked loss to emphasize the dropped pixels, we find that it is unnecessary when we use input-output shortcut connection to make the learned function as a residual one. After training the pre-training network, we make use of the learned representations by fine-tuning the encoder part on new tasks.

2.3. Corruptions

We investigate two types of corruptions for pre-training .

Pixel-level Gaussian noise. It is a common type of corruption used in image denoising tasks. Given an image x, we add random Gaussian noise with 0 mean and standard deviation σ to each pixel uniformly. σ is also called noise level.

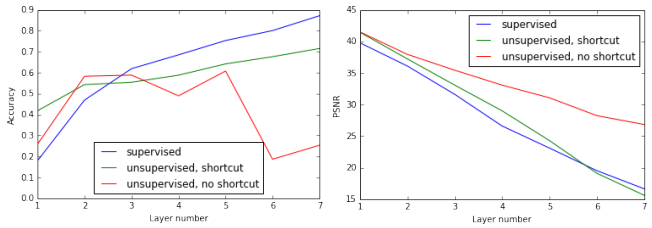
Block masking noise. Another type of corruption we use is to set some pixels of a image to 0. It is also used in vanilla denoising auto-encoder [1]. We use a special case of the masking noise, in which the pixels dropped out are adjacent. Images with this type of corruption are used by Pathak *et al.*[7] for their Context Encoder.

We find that using pixel-level Gaussian noise works better when the images are small and labeled data are sufficient. In other cases, masking noise with multiple small blocks works better.

2.4. Discussion for Shortcut Connection

Since the skip connections are essential parts in our auto-encoder network, we firstly conduct a ablation study to show their importance. We train 3 different models on 4000 CIFAR-10 images: 1) plain classification network. 2) auto-encoder network with shortcuts and 3) auto-encoder network without shortcut. The first one is trained on annotated labels, and the other two are purely unsupervised. Then we fix the learned parameters and train a separate classification/reconstruction probe on top of each layer of each network. The results are shown in Figure 2. For classification we train a linear classifier for each layer's feature maps, and for reconstruction we train a stack of up-sampling layers to reconstruct raw images from these features.

We can draw following conclusions from the results: 1) Our auto-encoder network with shortcuts learns more abstract representations which are useful for high-level tasks like classification. 2) As the network goes deeper, auto-encoder net-



(a) Classification (b) Reconstruction

Fig. 2: Comparisons: The classification and reconstruction performance of features extracted from each layers in different networks. Pre-training is conducted on Cifar-10.

work with shortcuts gradually drops low-level image details like the supervised trained network does.

2.5. Implementation Details

Our method is implemented with Keras [17]. ADAM [18] is used for optimization. The learning rate is set as 0.0001 at first, and divided by 10 gradually. If not specified, the weights are initialized by random Gaussian numbers with 0 mean and standard deviation 0.01.

Shortcut connections are linked every 2 convolutional layers to their corresponding deconvolutional layers, as well as one connection from the input to the output. During training, we use simple data augmentation following [5] to randomly crop the image to proper size, which is 29×29 for CIFAR-10 and CIFAR-100, 89×89 for STL-10 and 225×225 for PASCAL VOC. Then we randomly flip it horizontally (No flipping for CIFAR-10 with 4000 labels experiment).All pixels are zero-centered and normalized. At testing time, central crops are taken for CIFAR-10, CIFAR-100 and STL-10. For PASCAL VOC, we follow [7] to average the results of 10 random crops.

3. EXPERIMENTS

3.1. Classification with no Extra Unlabeled Data

In this experiment, we show that in the scenario in which there are no extra unlabeled data, the proposed classification method based on our simple all-convolutional network can obtain competitive classification results with the help of unsupervised pre-training using auto-encoder network.

We conduct experiments on both CIFAR-10 and CIFAR-100 dataset. The architecture is a 15-layer all-convolution network.

Table 1 shows the overall classification accuracy of our network and other state-of-the-art results on CIFAR-10 and CIFAR-100.

The classification results achieved by our simple stack all-convolution network are comparable with those achieved by the state-of-the-art method based on deep residual network (ResNet) [16, 10].

Method	CIFAR-10	CIFAR-100
ELU [19]	93.45	75.72
ResNet 164 [5]	93.39	74.84
ResNet V2 1001 [16]	95.08	77.29
Wide ResNet [10]	**96.11**	**81.15**
Random Gaussian	93.95	74.11
Pre-training no shortcut	92.07	70.21
Ours	95.15	75.41

Table 1: Comparisons with state-of-the-arts: CIFAR-10 and CIFAR-100 classification classification accuracy (%).

3.2. Classification with Limited Labels

In this experiment , we show that in the semi-supervised setting when there are extra unlabeled data but limited labeled data, our method can achieve more competitive results even compared to jointly semi-supervised learning methods.

We conduct experiments on two commonly used tasks for semi-supervised learning: CIFAR-10 with 4,000 labeled images, and STL-10 Classification. The first experiment uses 4,000 labeled images from CIFAR-10 and treats other 46,000 images as unlabeled data. The second experiment uses 100,000 unlabeled images and 5,000 labeled images with 1,000 in each fold for training. We report the results in Table 2 and Table 3.

For a fair comparison, we use no pooling version of the 9-layer network in [13] for the first experiment and the network in [8] for the second one .

Method	Accuracy (%)
Supervised state-of-the-art [20]	76.67±0.61
Ladder Network [20]	79.60±0.47
CatGAN [21]	80.42±0.58
Improved GAN [22]	**81.37±2.32**
No pre-training	70.89±0.30
Pre-training without shortcut	74.07±0.43
Ours	80.22±0.37

Table 2: Test accurasy on CIFAR-10 with 4000 labels

Method	Accuracy (%)
Exemplar-CNN [23]	**75.4±0.3**
SWWAE [8]	74.3
No pre-training	64.6±0.7
Pre-training without shortcut	70.2±0.6
Ours	**75.8±0.5**

Table 3: STL-10 classification accuracy.

Some of the compared methods are jointly semi-supervised learning framework while we use unsupervised pre-training, yet the results of our method are still very competitive.

3.3. Learning from Large-scale Unlabeled Data

In this section, we show that our method based on auto-encoder network scale well to large dataset and different supervised tasks.

Specifically, we train our convolutional auto-encoders on ImageNet 2012 training data without using any labels. Then, we fine-tune it on the PASCAL VOC 2007 classification and PASCAL VOC 2012 segmentation task respectively.

We use a fully convolutional VGG-16 network [12] for both experiments. For classification, we compare our method with Context Encoder [7], which used the same network architecture. Results are reported in 4.

For semantic segmentation, we did not intend to achieve state-of-the-art result since these methods often use supervised pre-training while our pre-training is unsupervised. Instead, we want to show that both encoder and decoder part of our network are transferable. We use our network to perform segmentation by simply replacing the last deconvolutional layer with a convolution layer of proper number of channels for segmentation. 3 segmentation networks are trained with different initialization strategies: (1) initializing all layers with small random Gaussian numbers, (2) initializing the encoder by unsupervised pre-training and initializing the decoder randomly, and (3) initializing both the encoder and decoder by pre-training.

The results are reported in Table 4 and Figure 3.

Method	mAP(%)
Random Gaussian	67.11
Context Encoder	70.24
Ours without shortcut	69.38
Ours shortcut	**71.25**

Table 4: PASCAL VOC classification results.

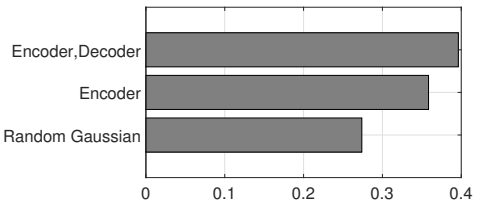


Fig. 3: Mean IUs of PASCAL VOC 2012 Segmentation task on validation data with different fine-tuning strategies.

We can observe: 1) Unsupervised pre-training can improve final result. 2) The learned parameters of the decoder of our auto-encoder network can also be transferred to high-level task.

3.4. Analysis and Visualization

Although our method perform well on different tasks, it is hard to justify whether the pre-training learns abstract features or it just eases the optimization. In this experiment, we intend to show our auto-encoder network with symmetric shortcuts can learn high-level representations from unlabeled images.

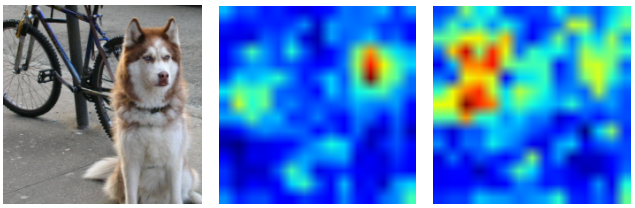
To show this, we pre-train our network on 2 subset of Imagenet 2012 training images without label. One belong to super class "conveyance, transport", and the other belong to "mammal" or "bird". We select 93 thousand images from each subset for balance.

Then, we fine-tune the network on PASCAL VOC 2007 Classification dataset and report the results in Table 5 for Animal and Vehicle hyper class as well as mean accuracy for all 20 classes.

Pre-trained on	Animal	Vechicle	Mean All
No pre-training	67.02	76.87	67.11
Animal 93K	**73.04**	79.04	70.77
Vehicle 93K	72.17	**80.00**	**70.93**

Table 5: Compare pre-training with different data distributions: Mean average precision (%) of PASCAL VOC 2007 classification on different coarse classes.

We also visualize some of the learned representations in Figure 4.



(a) Original Images (b) Animal Trained (c) Vehicle Trained

Fig. 4: Visualization of feature maps trained on unlabeled images with different distributions.

The results and visualizations clearly show that our method can indeed learn abstract representations from unlabeled data during unsupervised pre-training, but not merely ease the optimization by providing a better parameter initialization.

4. CONCLUSIONS

We propose a novel method for learning representations in unlabeled and partially labeled images. The architecture is a fully convolutional encoder-decoder network with symmetric shortcut connections. We show experimentally that our mothod can help to find good network initialization and perform well in semi-supervised learning.

5. REFERENCES

[1] 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," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015, pp. 1026–1034.

[3] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artificial Intell. & Stat.*, 2010, pp. 249–256.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, pp. 770–778.

[6] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," in *Proc. Advances in Neural Inf. Process. Syst.*, 2016.

[7] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, pp. 2536–2544.

[8] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[9] Harri Valpola, "From neural PCA to deep unsupervised learning," *Advances in Independent Component Analysis and Learning Machines*, pp. 143–171, 2015.

[10] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[12] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[13] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. Int. Conf. Learn. Representations*, 2015.

[14] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015, pp. 1520–1528.

[15] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comp. Vis.*, 2016, pp. 630–645.

[17] François Chollet, "Keras," 2015.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[19] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Proc. Int. Conf. Learn. Representations*, 2016.

[20] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko, "Semi-supervised learning with ladder networks," in *Proc. Advances in Neural Inf. Process. Syst.*, 2015, pp. 3546–3554.

[21] Jost Tobias Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[22] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training gans," in *Proc. Advances in Neural Inf. Process. Syst.*, 2016, pp. 2226–2234.

[23] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proc. Advances in Neural Inf. Process. Syst.*, 2014, pp. 766–774.