Image Recognition Method Based on Deep Learning

Xin Jia[1]

1. Tianjin University of Technology, Tianjin 300384

E-mail: 1026311742@qq.com

Abstract: Deep learning algorithms are a subset of the machine learning algorithms, which aim at discovering multiple levels of distributed representations. Recently, numerous deep learning algorithms have been proposed to solve traditional artificial intelligence problems.This work aims to review the state-of-the-art in deep learning algorithms in computer vision by highlighting the contributions and challenges from recent research papers. It first gives an overview of various deep learning approaches and their recent developments, and then briefly describes their applications in diverse vision tasks. Finally,the paper summarizes the future trends and challenges in designing and training deep neural networks.

Keywords: deep learning, computer vision, developments, applications, trends, challenges

1. INTRODUCTION

Deep learning is a subfield of machine learning which attempts to learn high-level abstractions in data by utilizing hierarchical architectures. It is an emerging approach and has been widely applied in traditional artificial intelligence domains, such as semantic parsing [1], transfer learning [2,3], natural language processing[4], computer vision [5,6] and many more. There are mainly three important reasons for the booming of deep learning today: the dramatically increased chip processing abilities (e.g. GPU units), the significantly lowered cost of computing hardware, and the considerable advances in the machine learning algorithms[9].

Various deep learning approaches have been extensively reviewed and discussed in recent years [8-12].Among those Schmidhuber et al. emphasized the important inspirations and technical contributions in a historical timeline format, while Bengio examined the challenges of deep learning research and proposed a few forward-looking research directions. Deep networks have been shown to be successful for computer vision tasks because they can extract appropriate features while jointly performing discrimination[9,13].In recent ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competitions[11], deep learning methods have been widely adopted by different researchers and achieved top accuracy scores [7].

This survey is intended to be useful to general neural computing, computer vision and multimedia researchers who are interested in the state-of-the-art in deep learning in computer vision. It provides an overview of various deep learning algorithms and their applications, especially those that can be applied in the computer vision domain. In Section2, we divide the deep learning algorithms into four categories:Convolutional Neural Networks,Restricted Boltzmann Machines, Autoencoder and Sparse Coding. Some well-known models inthese categories as well as their developments are listed. We also describe the contributions and limitations for these models in this section. In Section 3, we describe the achievements of deep learning schemes in various computer vision applications, i.e. image classification, object detection, image retrieval, semantic segmentation and human pose estimation. The results on these applications are shown and compared in the pipeline of their commonly used datasets. In Section 4, along with the success deep learning methods have achieved, we also face several challenges when designing and training the deep networks.In this section, we summarize some major challenges for deep learning, together with the inherent trends that might be developed in the future. In Section 5, we conclude the paper.

2. RECENT DEVELOPMENTS

In recent years, deep learning has been extensively studied in the field of computer vision and as a consequence, a large number of related approaches have emerged. Generally, these methods can be divided into four categories according to the basic method they are derived from: Convolutional Neural Networks (CNNs), Restricted Boltzmann Machines (RBMs), Autoencoder and Sparse Coding. The categorization of deep learning

978-1-5090-4657-7/17/$31.00 ©2017 IEEE

methods along with some representative works is shown in Fig. 1.

In the next four parts, we will briefly review each of these deep learning methods and their most recent development.

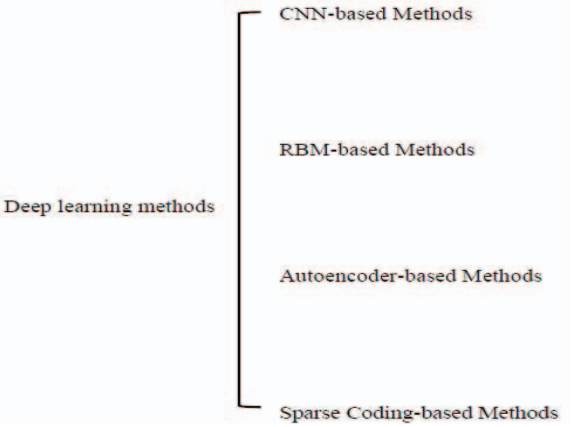


Fig. 1. A categorization of the deep learning methods and their representative works.

2.1 Convolutional Neural Networks (CNNs)

The Convolutional Neural Networks (CNN) is one of the most notable deep learning approaches where multiple layers are trained in a robust manner [17]. It has been found highly effective and is also the most commonly used in diverse computer vision applications.The pipeline of the general CNN architecture is shown in Fig.2.

Generally, a CNN consists of three main neural layers, which are convolutional layers, pooling layers,and fully connected layers. Different kinds of layers play different roles. In Fig. 2, a general CNN architecture for image classification[6] is shown layer-by-layer. There are two stages for training the network: a forward stage and a backward stage. First, the main goal of the forward stage is to represent the input image with the current parameters (weights and bias) in each layer. Then the prediction output is used to compute the loss cost with the ground truth labels. Second, based on the loss cost, the backward stage computes the gradients of each parameter with chain rules. All the parameters are updated based on the gradients, and are prepared for the next forward computation. After sufficient iterations of the forward and backward stages,the network learning can be stopped.

Next, we will first introduce the functions along with the recent developments of each layer, and then summarize the commonly used training strategies of the networks. Finally, we present several well-known CNN models, derived models, and conclude with the current tendency for using these models in real applications.

Generally, a CNN is a hierarchical neural network whose convolutional layers alternate with pooling layers, followed by some fully connected layers (see Fig.2). In this section, we will present the function of the three layers and briefly review the recent advances that have appeared in research on those layers.

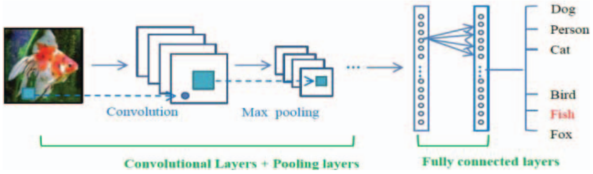


Fig 2. The pipeline of the general CNN architecture.

Convolutional layers

In the convolutional layers, a CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps, as shown in Fig. 3.There are three main advantages of the convolution operation [19]: 1) the weight sharing mechanism in the same feature map reduces the number of parameters 2) local connectivity learns correlations among neighboring pixels 3) invariance to the location of the object.Due to the benefits introduced by the convolution operation, some well-known research papers use it as a replacement for the fully connected layers to accelerate the learning process[20].One interesting approach of handling the convolutional layers is the Network in Network (NIN)[21] method, where themain idea is to substitute the conventional convolutional layer with a small multilayer perceptron consisting of multiple fully connected layers with nonlinear activation functions, thereby replacing the linear filters with nonlinear neural networks. This method achieves good results in image classification.

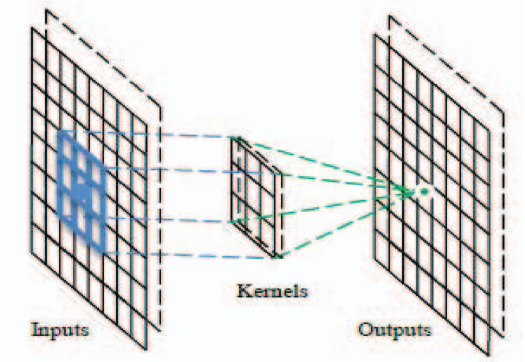


Fig 3. The operation of the convolutional layer.

Pooling layers

Generally, a pooling layer follows a convolutional layer, and can be used to reduce the dimensions of feature maps and network parameters. Similar to convolutional layers, pooling layers are also translation invariant, because their computations take neighboring pixels into account. Average pooling and max pooling are the most commonly used strategies. Fig. 4 gives an example for a max pooling process. For 8x8 feature maps, the output maps reduce to 4x4 dimensions, with a max pooling operator which has size 2x2 and stride 2.

For max pooling and average pooling, Boureau et al. provided a detailed theoretical analysis of their performances. Scherer et al.further conducted a comparison between the two pooling operations and found that max-pooling can lead to faster convergence, select superior invariant features and improve generalization.In recent years,various fast GPU implementations of CNN variants were presented, most of them utilize max-pooling strategy [6].

Stochastic Pooling: A drawback of max pooling is that it is sensitive to overfit the training set, making it hard to generalize well to test samples[19]. Aiming to solve this problem, Zeiler et al.proposed a stochastic pooling approach which replaces the conventional deterministic pooling operations with a stochastic procedure, by randomly picking the activation within each pooling region according to a multinomial distribution. It is equivalent to standard max pooling but with many copies of the input image, each having small local deformations.This stochastic nature is helpful to prevent the overfitting problem.

Spatial Pyramid Pooling (SPP):Normally, the CNN-based methods require a fixed-size input image. This restriction may reduce the recognition accuracy for images of an arbitrary size. To eliminate this limitation, He et al. utilized the general CNN architecture but replaced the last pooling layer with a spatial pyramid pooling layer. The spatial pyramid pooling can extract fixed-length representations from arbitrary images, generating a flexible solution for handling different scales, sizes, aspect ratios, and can be applied in any CNN structure to boost the performance of this structure.

Def-Pooling:Handling deformation is a fundamental challenge in computer vision, especially for the object recognition task. Max pooling and average pooling are useful in handling deformation but they are not able to learn the deformation constraint and geometric model of object parts. To deal with deformation more efficiently, Ouyang et al.introduced a new deformation constrained pooling layer, called def-pooling layer,to enrich the deep model by learning the deformation of visual patterns. It can substitute the traditional max-pooling layer at any information abstraction level.

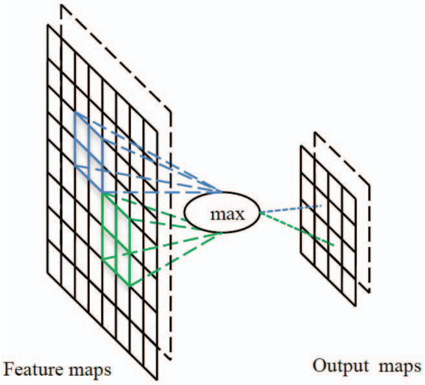


Fig 4. The operation of the max pooling layer.

Fully-connected Layers

Following the last pooling layer in the network as seen in Fig. 2, there are several fully-connected layers converting the 2D feature maps into a 1D feature vector, for further feature representation, as seen in Fig. 5.

Fully-connected layers perform like a traditional neural network and contain about 90% of the parameters in a CNN. It enables us to feed forward the neural network into a vector with a pre-defined length. We could either feed forward the vector into certain number categories for image classification[6] or take it as a feature vector for follow-up processing.

Changing the structure of the fully-connected layer is uncommon, however an example came in the transferred learning approach, which preserved the parameters learned by ImageNet[6], but replaced the last fully-connected layer with two new fully-connected layers to adapt to the new visual recognition tasks.

The drawback of these layers is that they contain many parameters, which results in a large computational effort for training them. Therefore, a promising and commonly applied direction is to remove these layers or decrease the connections with a certain method. For example, GoogLeNet[20] designed a deep and wide network while keeping the computational budget constant, by switching from fully connected to sparsely connected architectures.

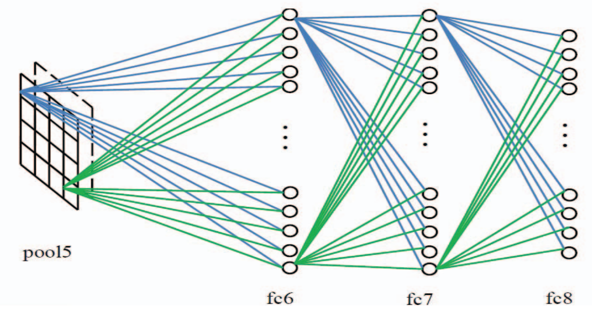


Fig 5. The operation of the fully-connected layer.

Training Strategy

Compared to shallow learning, the advantage of deep learning is that it can build deep architectures to learn more abstract information. However, the large amount of parameters introduced may also lead to another problem: overfitting. Recently, numerous regularization methods have emerged in defense of overfitting, including the stochastic pooling mentioned above. In this section, we will introduce several other regularization techniques that may influence the training performance.

Dropout and DropConnect:Dropout was proposed by Hinton et al. and explained in-depth by Baldi et al. During each training case, the algorithm will randomly omit half of the feature detectors in order to prevent complex coadaptations on the training data and enhance the generalization ability. This method was further improved in[21]. Specifically, research by Warde-Farley et al. analyzed the efficacy of dropouts and suggested that dropout is an extremely effective ensemble learning method.

One well-known generalization derived from Dropout is called DropConnect, which randomlydrops weights rather than the activations. Experiments showed that it can achieve competitive or even better results on a variety of standard benchmarks, although slightly slower.

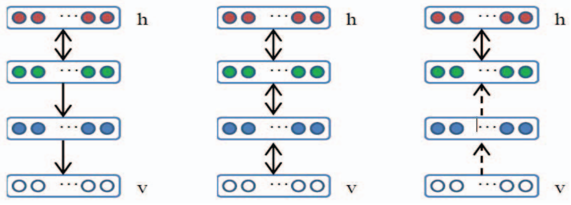
Data Augmentation:When a CNN is applied to visual object recognition, data augmentation is often utilized to generate additional data without introducing extra labeling costs. The well-known AlexNet [6] employed two distinct forms of data augmentation: the first form of data augmentation consists of generating image translations and horizontal reflections, and the second form consists of altering the intensities of the RGB channels in training images. Howard et al. took AlexNet as the base model and added additional transformations that improved the translation invariance and color invariance by extending image crops with extra pixels and adding additional color manipulations. This data augmentation method was widely utilized by some of the more recent studies[20]. Dosovitskiy et al. proposed an unsupervised feature learning approach based on data augmentation: it first randomly sampled a set of image patches and declares each of them as a surrogate class, then extended these classes by applying transformations corresponding to translation, scale, color and contrast. Finally, it trained a CNN to discriminate between these surrogate classes. The features learnt by the network showed good results on a variety of classification tasks. Aside from the classical methods such as scaling, rotating and cropping, Wu et al.further adopted color casting, vignetting and lens distortion techniques, which produced more training examples with broad coverage.

Pre-training and fine-tuning:Pre-training means to initialize the networks with pre-trained parameters rather than randomly set parameters.It is quite popular in models based on CNNs, due to the advantages that it can accelerate the learning process as well as improve the generalization ability. Erhan et al.[16] has conducted extensivesimulations on the existing algorithms to find why pre-trained networks work better than networks trained in a traditional way. As AlexNet[6] achieved excellent performance and is released to the public, numerous approaches choose AlexNet trained on ImageNet2012 as their baseline deep model, and use fine-tuning of the parameters according to their specific tasks. Nevertheless, there are approaches that deliver better performance by training on other models, e.g. Clarifai,GoogLeNet[20],and VGG.Another interesting thing to note is that these regularization techniques for training are not mutually exclusive and they can be combined to boost the performance.

2.2 Restricted Boltzmann Machines (RBMs)

A Restricted Boltzmann Machine (RBM) is a generative stochastic neural network, and was proposed by Hinton et al. in 1986 [23]. An RBM is a variant of the Boltzmann Machine, with the restriction that the visible units and hidden units must form a bipartite graph. This restriction allows for more efficient training algorithms, in particular the gradient-based contrastive divergence algorithm.

Utilizing RBMs as learning modules, we can compose the following deep models: Deep Belief Networks(DBNs), Deep Boltzmann Machines (DBMs) and Deep Energy Models (DEMs). The comparison between the three models is shown in Fig.6.



(a)DBN (b)DBM (c)DEM

Fig 6. The comparison of the three models.

Deep Belief Networks (DBNs)

The Deep Boltzmann Machine (DBM), proposed by Salakhutdinov el.al, is another deep learning algorithm where the units are again arranged in layers. Compared to DBNs, whose top two layers form an undirected graphical model and whose lower layers form a directed generative model, the DBM has undirected connections across its structure. There are also many other approaches that aim to improve the effectiveness of DBMs. The improvements can either take place at the pre-training stage or at the training stage. For example, Montavon et al. introduced the centering trick to improve the stability of a DBM and made it to be more discriminative and generative. The multi-prediction training scheme was utilized to jointly train the DBM which outperforms the previous methods in image classification proposed in .

Deep Energy Models (DEMs)

The Deep Energy Model (DEM), introduced by Ngiam et al. , is a more recent approach to train deep architectures. Unlike DBNs and DBMs which share the property of having multiple stochastic hidden layers, the DEM just has a single layer of stochastic hidden units for efficient training and inference.

Although RBMs are not as suitable as CNNs for vision applications, there are also some good examples adopting RBMs to vision tasks. The Shape Boltzmann Machine was proposed by Eslami et al. to handle the task of modeling binary shape images, which learns high quality probability distributions over object shapes, for both realism of samples from the distribution and generalization to new examples of the same shape class. Kae et al. combined the CRF and the RBM to model both local and global structure in face segmentation, which has consistently reduced the error in face labeling. A new deep architecture has been presented for phone recognition that combines a Mean-Covariance RBM feature extraction module with a standard DBN. This approach attacks both the representational inefficiency issues of GMMs and an important limitation of previous work applying DBNs to phone recognition.

2.3 Autoencoder

The autoencoder is a special type of artificial neural network used for learning efficient encodings.Instead of training the network to predict some target value Y given inputs X, an autoencoder is trained to reconstruct its own inputs X, therefore, the output vectors have the same dimensionality as the input vector.

The deep autoencoder was first proposed by Hinton et al, and is still extensively studied in recent papers [19]. A deep autoencoder is often trained with a variant of back-propagation, e.g. the conjugate gradient method. Though often reasonably effective, this model could become quite ineffective if errors are present in the first few layers. This may cause the network to learn to reconstruct the average of the training data. A proper approach to remove this problem is to pre-train the network with initial weights that approximate the final solution[23]. There are also variants of autoencoder proposed to make the representation as“constant” as possible with respect to the changes in input.

2.4 Sparse Coding

The purpose of sparse coding is to learn an over-complete set of basic functions to describe the input data[24]. There are numerous advantages of sparse coding[15-18]: (1) It can reconstruct the descriptor better by using multiple bases and capturing the correlations between similar descriptors which share bases;(2)the sparsity allows the representation to capture salient properties of images; (3) it is in line with the biological visual system, which argues that sparse features of signals are useful for learning; (4)image statistics study shows that image patches are sparse signals; (5)patterns with sparse features are more linearly separable.

As we have briefly stated how to generate the sparse representation given the objective function, in this subsection, we will introduce some well-known algorithms related to sparse coding, in particular those that are used in computer vision tasks. The well-known sparse coding algorithms and relations, along with their contributions and drawbacks are shown in Fig 7.

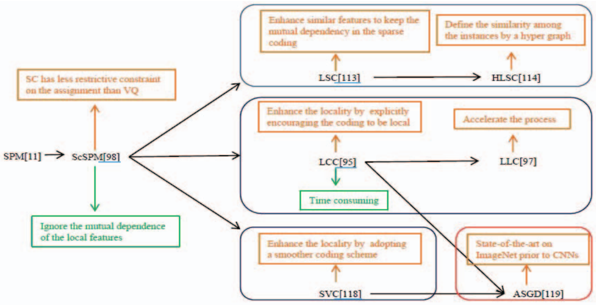

Enhance similar features to keep the mutual dependency in the sparse coding
Define the similarity among the instances by a hyper graph
SC has less restrictive constraint on the assignment than VQ
LSC[113]
HLSC[114]
Enhance the locality by explicitly encouraging the coding to be local
Accelerate the process
SPM[11]
ScSPM[98]
LCC[95]
LLC[97]
Time consuming
Ignore the mutual dependence of the local features
Enhance the locality by adopting a smoother coding scheme
State-of-the-art on ImageNet prior to CNNs
SVC[118]
ASGD[119]

Fig 7.The well-known sparse coding algorithms, relations, contributions and drawbacks.

3. APPLICATIONS AND RESULTS

Deep learning has been widely adopted in various directions of computer vision, such as image classification, object detection, image retrieval and semantic segmentation, and human pose estimation, which are key tasks for image understanding. In this part, we will briefly summarize the developments of deep learning (all of the results are referred from the original papers), especially the CNN based algorithms, in these five areas.

3.1 Image Classification

The image classification task consists of labeling input images with a probability of the presence of a particular visual object class [19], as is shown in Fig 8.

Prior to deep learning, perhaps the most commonly used methods in image classification were methods based on bags of visual words (BoW) [13], which first describes the image as a histogram of quantized visual words, and then feeds the histogram into a classifier (typically an SVM). This pipeline was based on the orderless statistics, to incorporate spatial geometry into the BoW descriptors. Lazebnik et al.[11] integrated a spatial pyramid approach into the pipeline, which counts the number of visual words inside a set of image sub-regions instead of the whole region. Thereafter, this pipeline was further improved by importing sparse coding optimization problems to the building of codebooks [12], which receives the best performance on the ImageNet 1000-class classification in 2010. Sparse coding is one of the basic algorithms in deep learning, and it is more discriminative than the original hand-designed ones, i.e. HOG and LBP [14].

Despite the potential capacity possessed by larger models, they also suffered from overfitting and underfitting problems when there is little training data or little training time. To avoid this shortcoming, Wu et al. developed new strategies, i.e. DeepImage, for data augmentation and usage of multi-scale images. They also built a large supercomputer for deep neural networks and developed a highly optimized parallel algorithm, and the classification result achieved a relative 20% improvement over the previous one with a top-5 error rate of 5.33%. More Recently, He et al.proposed the Parametric Rectified Linear Unit to generate the traditional rectified activation units and derived a robust initialization method. This scheme led to 4.94% top-5 test error and surpassed human-level performance (5.1%) for the first time. Similar results were achieved by Ioffe et al. whose method reached a 4.8% test error by utilizing an ensemble of batch-normalized networks.

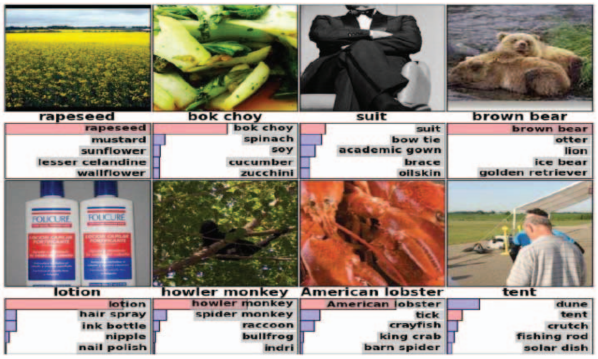


Fig 8.Image classification examples from AlexNet.

4. TRENDS AND CHALLENGES

Theoretical Understanding:Despite the progress achieved in the theory of deep learning, there is significant room for better understanding in evolving and optimizing the CNN architectures toward improving desirable properties such as invariance and class discrimination.

Training with limited data:Larger models demonstrate more potential capacity and have become the tendency of recent developments.However, the shortage of training data may limit the size and learning ability of such models, especially when it is expensive to obtain fully labeled data.

How to overcome the need for enormous amounts of training data and how to train large networks effectively remains to be addressed.

More Powerful Models:As deep learning related algorithms have moved forward the-state-of-the-art results of various computer vision tasks by a large margin, it becomes more challenging to make progress on top of that. There might be several directions for more powerful models:The first direction is to increase the generalization ability by increasing the size of the networks [20].A second direction is to combine the information from multiple sources.A third direction towards more powerful models is to design more specific deep networks.

5. CONCLUSION

This paper presents a comprehensive review of deep learning and develops a categorization scheme to analyze the existing deep learning literature. It divides the deep learning algorithms into four categories according to the basic model they derived from: Convolutional Neural Networks, Restricted Boltzmann Machines, Autoencoder and Sparse Coding. The state-of-the-art approaches of the four classes are discussed and analyzed in detail. For the applications in the computer vision domain, the paper mainly reports the advancements of CNN based schemes, as it is the most extensively utilized and most suitable for images. Most notably, some recent articles have reported inspiring advances showing that some CNN-based algorithms have already exceeded the accuracy of human raters.Despite the promising results reported so far, there is significant room for further advances. For example, the underlying theoretical foundation does not yet explain under what conditions they will perform well or outperform other approaches, and how to determine the optimal structure for a certain task. This paper describes these challenges and summarizes the new trends in designing and training deep neural networks, along with several directions that may be further explored in the future.

REFERENCES

[1] Bordes A, Glorot X, Weston J, et al. Joint learning of words and meaning representations for open-text semantic parsing,in: AISTATS, 2012.

[2] Ciresan D C, Meier U, Schmidhuber J. Transfer learning for Latin and Chinese characters with deep neural networks, in:IJCNN, 2012.

[3] Ren J S J, Xu L. On Vectorization of Deep Convolutional Neural Networks for Vision Tasks, in AAAI, 2015.

[4] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality, in:NIPS, 2013.

[5] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting, The Journal of MachineLearning Research,15(1): 1929-1958,2014.

[6] LeCun Y. Learning invariant feature hierarchies, in: ECCV workshop, 2012.

[7] Bengio Y. Deep learning of representations: Looking forward, Statistical Language and Speech Processing. Springer Berlin Heidelberg,1-37,2013.

[8] Goroshin R, LeCun Y. Saturating auto-encoders, in: ICLR, 2013.

[9] Li H, Zhao R, Wang X. Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification,in: technical report, 2014.

[10] Erhan D, Bengio Y, Courville A, et al. Why does unsupervised pre-training help deep learning? The Journal of Machine Learning Research,11: 625-660,2010.

[11] Weston J, Ratle F, Mobahi H, et al. Deep learning via semi-supervised embedding, Neural Networks: Tricks of the Trade.Springer Berlin Heidelberg, 639-655, 2012.

[12] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors,Technique report, 2012.

[13] Baldi P, Sadowski P J. Understanding dropout, in: NIPS, 2013.

[14] Warde-Farley D, Goodfellow I J, Courville A, et al. An empirical analysis of dropout in piecewise linear networks, in:ICLR, 2014.

[15] Liou C Y, Cheng W C, Liou J W, et al. Autoencoder for words,Neurocomputing, 139: 84-96,2014.

[16] Jiang X, Zhang Y, Zhang W, et al. A novel sparse auto-encoder for deep unsupervised learning, in: ICACI, 2013.

[17] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, The Journal of Machine Learning Research,11: 3371-3408,2010.

[18] Rifai S, Vincent P, Muller X, et al. Contractive auto-encoders: Explicit invariance during feature extraction, in: ICML,2011.

[19] Alain G, Bengio Y. What regularized auto-encoders learn from the data generating distribution, in: ICLR, 2013.

[20] Masci J, Meier U, Cireşan D, et al. Stacked convolutional auto-encoders for hierarchical feature extraction, in: ICANN,2011.

[21] Baccouche M, Mamalet F, Wolf C, et al. Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification,in: BMVC, 2012.

[22] Mairal J, Bach F, Ponce J, et al. Online dictionary learning for sparse coding, in: ICML, 2009.

[23] Mairal J, Bach F, Ponce J, et al. Online learning for matrix factorization and sparse coding. The Journal of Machine Learning Research, 11: 19-60,2010.

[24] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders, in:ICML, 2008.

[25] Lin Y, Lv F, Zhu S, et al. Large-scale image classification: fast feature extraction and svm training, in: CVPR, 2011.