

# A Review about Building Hidden Layer Methods of Deep Learning

Shuo Hu and Yaqing Zuo

School of Electrical Engineering, Yanshan University, Qinhuangdao, China

Email: hus@ysu.edu.cn, thboma@163.com

Lizhe Wang and Peng Liu

Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, P. R. China

Email: {lzwang, pengliu}@radi.ac.cn

**Abstract**—Deep learning has shown its great potential and function in algorithm research and practical application (such as speech recognition, natural language processing, computer vision). Deep learning is a kind of new multilayer neural network learning algorithm, which alleviates the optimization difficulty of traditional deep models and arouses wide attention in the field of machine learning. Firstly, the origin of deep learning is discussed and the concept of deep learning is also introduced. Secondly, according to the architectural characteristics, deep learning algorithms are classified into three classes, this paper emphatically introduces deep networks for unsupervised and supervised learning model and elaborates typical deep learning models and the corresponding extension models. This paper also analyzes both advantage and disadvantage of each model and points out each extension method's inheritance relationship with the corresponding typical model. Finally, applications of deep learning algorithms is illustrated, the remaining issues and the future orientation are concluded as well.

**Index Terms**—deep learning, auto-encoder, restricted boltzmann machine, convolutional neural network, deep neural network

## I. INTRODUCTION

According to the related studies, it is necessary to introduce the deep learning in order to study higher-order abstract concept of complex functions and solve the artificial intelligence related tasks. Kunihiko Fukusima's introduction of the Neocognitron in 1980 helped facilitate modern deep learning architectures. Before that, Alexey Grigorevich Ivakhnenko published the first general, working learning algorithms for deep networks in 1965. Since 2006, the deep structured learning has emerged as a new area of machine learning research [1], [2], [3]. The motivation of deep learning is by establishing and simulating human brain to analysis and learn neural network. Deep learning copies the human brain mechanism to explain the data, such as image, sound and text. The concept of deep learning is the result of the

artificial neural network research. A good example of deep learning model is MLP containing many hidden layers. Deep learning combines low-level features to form a more abstract high-level representation (category or feature) in order to find out distributed characteristic presentation of data [4]. The definition of the deep learning has several versions. In this paper, one of the versions which is easy to be understood is introduced: a class of machine learning techniques that discover many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification [5]. Deep learning architecture is composed of multilayer nonlinear units, in which each lower output as an input of the higher level can learn effective features from a large number of the input data, and higher level of learning includes a great deal of structural information contained in the input data. It is a good method to extract representation of the data. This method can be utilized for specific problems like classification [6], [7], regression [8] and information retrieval [9], dimensionality reduction [10].

In view of the deep learning of theoretical significance and practical application value, domestic study of deep structure is still in its infancy. Compared to other countries, the published literatures are relatively few. This paper summarizes the latest progress of the deep learning system, which lays a certain foundation for the further study of deep learning theory and expands its application fields.

## II. THE BASIC METHOD OF DEEP LEARNING

Deep learning involves quite a wide range of machine learning techniques and structures. A three-way categorization is performed depending on how the architectures and techniques are intended for use. a) Deep networks for unsupervised or generative learning, this structure describes the high order correlation characteristics of the data or characterize joint probability distribution of the observed data and the corresponding categories. Examples of commonly used: Denoising Auto-Encoder (DAE), Restricted Boltzmann Machine

(RBM), Deep Belief Networks (DBN), and Deep Boltzmann Machine (DBM). b) Deep networks for supervised learning, which aims at providing discriminative power for pattern classification and describes the posterior distribution of the data, such as Convolutional Neural Network (CNN), Deep Neural Network (DNN). c) Hybrid deep networks, the goal is discrimination. Since it usually takes advantage of the structure of the generative learning output, the optimization will be easier, like DBN-DNN, deep CNNs. Next, deep networks for unsupervised and supervised learning are emphatically introduced. Typical deep learning models (Auto-Encoder (AE), Restricted Boltzmann Machine (RBM), Convolutional Neural Network (CNN) and so on) and the extensions of each model are introduced. Fig. 1 shows inheritance relationships of typical deep learning models.

#### A. Deep Networks for Unsupervised or Generative Learning

Next, the AE, the RBM and the corresponding extension model are introduced.

##### Auto-Encoder (AE)

Auto-Encoder (AE) is an unsupervised machine learning technique, using neural network to produce low dimension to represent the high dimension input. An auto-encoder [11], [12] takes an input  $x \in R^d$ , then maps it to a hidden representation  $h \in R^{d'}$ , using a deterministic function of the type  $h = f_{\theta}(x) = \sigma_f(W, b) \cdot \theta = \{W, b\}$  represents the weight and bias.  $\sigma(t) = 1/(1 + \exp(-t))$  is the sigmoidal function, then it is used to reconstruct the input  $y = g_{\theta'}(h) = \sigma_g(W'h + b')$  with parameters  $\theta' = \{W', b'\}$ . The two parameter sets are usually constrained to be of the form  $W' = W^T$ . Parameters  $\theta$  and  $\theta'$  are trained to minimize the average reconstruction error over the training set. The purpose is to have  $y$  as close as possible to the input  $x$ . The parameters of this model (namely  $\theta = \{W, b_h, b_y\}$ ) are optimized such that the average reconstruction error is minimized. This corresponds to the minimum of the objective function. Compared with the linearity of PCA which limits the extraction of feature dimension, AE uses the inherent nonlinear neural network to overcome this limitation.

##### Regularized auto-encoder (AE+wd)

The simplest form of regularization is weight-decay [11]. It favors small weights by optimizing the regularized objective, where the hyper-parameter controls the strength of the regularization. Note that rather than having a prior on what the weights should be, it is possible to have a prior on what the hidden unit activations should be. From this viewpoint, several techniques have been developed to encourage the sparsity of representation.

##### Sparse Auto-Encoder (SAE)

The auto-encoder (AE) is able to capture the most important factor of the input data, so it can as much as

possible to emersion the input. The constraint condition is joined on the basis of the auto encoder, which demands most of the nodes are zero and only a few are not zero, this is the sparse auto-encoder (SAE) [12]. The aim is to aid the expression code as sparse as possible. The sparse expression is more effective than other expressions. Just like the brain, an input simply stimulates certain neurons and most of neurons are suppressed. The strengths are that SAE can not only reduce the data dimension, but also extract more helpful characteristics of data. But it also has weaknesses. When the network layer is not the same, characteristics of the model are different. If the number of layer is too low, learning efforts may be not enough, which leads to characteristics cannot reach the best effect; if the number of layer is too high, a fitting phenomenon may occur.

##### Denoising Auto-Encoder (DAE)

The disadvantage of AE is that once testing and training samples are not in the same distribution, the effect will be not good. So DAE is demanded to make up this defect and improve the robustness of the system.

The technique of denoising auto-encoder is a successful alternative form of regularization. The denoising auto-encoder [13] is trying to reconstruct noisy inputs. It first corrupts the initial input  $x$  into  $\tilde{x}$ , then maps it to a hidden representation, which is identical with the basic auto-encoder. The input  $x$  is stochastically corrupted to  $\tilde{x}$  by means of a stochastic mapping. The key difference from the basic auto-encoder is that  $y$  is now a deterministic function of  $\tilde{x}$  rather than  $x$ . Noise is added when to train the initial input, so the encoder must learn to remove the noise and obtain the real input with no noise pollution. As a result, it will force the encoder to study a more robust expression of the input signal, this is the reason why its generalization ability is superior to general encoder.

##### Stacked Denoising Auto-Encoder (SDAE)

Stacking denoising auto encoders [14] initializing a deep network and stacking RBMs in DBNs work in almost the same way. In the beginning, a first level of DAE is trained in which its learnt encoding function  $f_{\theta}$  is used on clean input, the aim is to get the resulting representation which is used for training a second level of DAE and learns a second level encoding function  $f_{\theta}^{(2)}$ , then the procedure is repeated. Once a stacking of encoders has been built, the result of top level representation is utilized as input for the supervised learning algorithm. In the end, stochastic gradient descent is used to simultaneously fine-tune the parameters of all layers.

The difference between DAE and SDAE is that: the corruption of DAE could be done in the training phase, and it doesn't need to do with the forward feedback. But the corruption and the denoising of SDAE need to be done in the training of each layer [14].

##### Discriminative Recurrent Sparse Auto-Encoder (DrSAE)

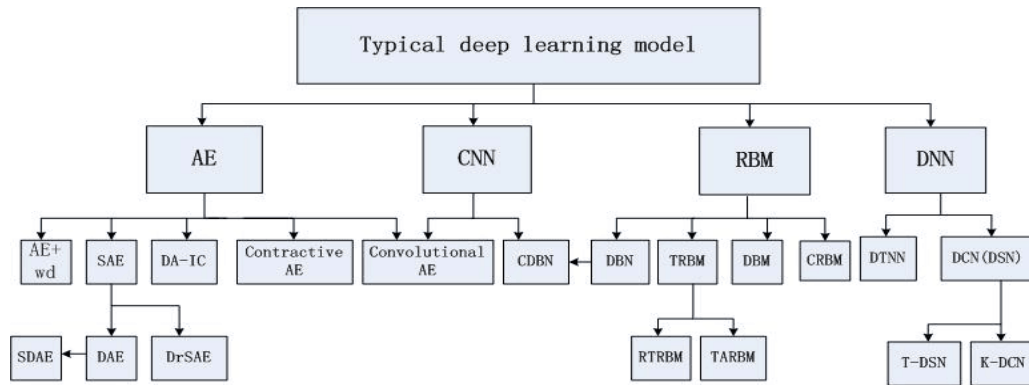


Figure 1. Inheritance relationships of typical deep learning models.

The discriminative recurrent sparse auto-encoder (DrSAE) [15] which comprises a recurrent encoder of rectified linear units [16], [17], unrolls for a fixed number of iterations and has a connection with two linear decoders for the purpose of reconstructing the input and predicting the classification. The aim of training is via back propagation-through-time [18] to minimize an unsupervised sparse reconstruction loss function, and then the loss function is added on the supervised classification by using a discriminative term. That is to say stochastic gradient descent is used to pre-train the unsupervised loss function and get parameters, it is also used to perform discriminative fine-tune on the unsupervised sparse reconstruction loss function and the supervised classification loss function.

DrSAEs are comparable to the recurrent neural networks [19], expect that the nonlinearity of DrSAEs is different and the loss function is heavily regularized. DrSAEs also resemble the recurrent networks [20], other than recurrent connections exist between the hidden units, not between the hidden units and the input units.

#### Contractive Auto-Encoder (CAE)

The contractive auto-encoder (CAE) is put forward by Bengio etc. as a new auto-encoder [11] which adds the new penalty term on the traditional auto-encoder reconstruction error. The new penalty term equals to the squared Frobenius norm of the Jacobian of the encoder activations function of the input. CAE can produce localized space contraction, therefore its characteristic is more robust.

Two components of the loss function are proposed as two optimization objectives of CAE: the first part (auto-encoder reconstruction) makes CAE will try best to capture a lot of information about the input image. The second part (the Jacobi matrix of Frobenius Norm) can be seen that the encoder throws away all information, hence the CAE is just capture the variance in the training data and insensitive to other variances.

The relationship with other auto-encoder variant: firstly, the relationship with AE + weight decay: The squared Frobenius norm of the Jacobian is equivalent to a linear encoder with an L2 weight decay. Secondly, the relationship with sparse auto-encoder: The purpose of sparse auto-encoder is to make the most feature of each sample become zero. For the sigmoid function of CAE, it means the derivative is small and the corresponding part

of the value of the Jacobi matrix is small, so they are similar. Thirdly, the relationship with denoising auto-encoders: CAEs encourage robustness of representation, but DAEs encourage robustness of reconstruction. Because for classification, we only need the encoder to extract features among them, the robustness of extracted features becomes more important than robustness of the reconstruction, thus this property makes CAEs easier than DAEs to learn robust features.

What's more, good features represent roughly in two metrics: one can well reconstruct the input data; the other is that under a certain extent disturbance the input data has noise formation. Ordinary auto-encoder and SAE mainly conform to the first standard, while DAE or CAE is mainly embodied in the second. If as a classification task, the second standard is more important, this is the superiority of the CAE. In general, CAE mainly inhibit the training sample disturbance in all directions.

#### Denoising Auto-encoder with Interdependent Codes (DA-IC)

The DA-IC [21] is a variant of DAE. The main idea is to capture the interaction between hidden layer nodes, such as inhibitory and excitatory interactions. That is the activation of hidden layer nodes, which is not only related to the input, but also will influence each other. The DA-IC's thought is to treat the inhibitory and excitatory lateral connections between the hidden layer units as adding an extra non-linear processing layer on the basis of regular encoding. The DA-IC using a simpler way to say is adding a hidden layer in the encoding function, in that the computational complexity will decrease. The DA-IC merely considers asymmetric lateral connections between the hidden layers when encoding and does not change when decoding. In encoding and decoding the same weight is shared. Compared with a recursive update equation [22], the DA-IC could copy with two defects. One is when we meet large layers or the number of iterations, the computation of the encoding becomes expensive. The other is that it is costly and hard to optimize the encoding through gradient descent.

#### Convolutional Auto-Encoder (CAE)

Fully connected AEs and DAEs both ignore the 2D image structure. When dealing with real image data, this is not the only problem. Another problem is that it involves redundancy in structure parameters, which makes each learned feature become global. But in the

field of machine vision and target recognition adopted by the most successful models, it can be discovered that localized features are implied in the whole data set. CAE [23] can better resolve the above problem. It is different from conventional AEs, as their weights are shared between all locations in the whole data set, such processing can preserve spatial locality. At this point the reconstruction becomes using a linear combination of basic image patches based on the implicit code.

The CAE which differs from the AE is to learn localized features of the image and add the convolution and pooling operation.

#### *Restricted Boltzmann Machine (RBM)*

The RBM is a stochastic neural network, it only has two layers of neurons. The visible layer, which composes of visible units, is used for input training data. The hidden layer, consisting of hidden units, is used as feature detectors. Connections only exist between the visible units of the input layer and the hidden units of the hidden layer, there are no visible-visible or hidden-hidden connections. Connections between neurons are bidirectional and symmetric, this means that during the training, information flows in both directions; during the usage of the network, weights are the same in both directions. In an RBM, the hidden units are conditionally independent given the visible states, so we can quickly get an unbiased sample from the posterior distribution when given the observed data. This is a big benefit over directed belief nets. It is the same way to the visible units. In training a single RBM, weight updates are performed with gradient ascent [24].

The RBM network works in the following way: First the network is trained by using some data sets and the neurons on visible layer are settled to match data points in data sets. After the network is trained, it can be put to use to classify other data. The RBM [25] which is used to initialize the feed forward neural network is a valid method of feature extraction and can obviously improve the generalization ability. The main characteristic of boltzmann machine is the activation layer features of the inputs the training data of the next layer, so the study is very quick.

#### *Temporal Restricted Boltzmann Machine (TRBM)*

For the extension of RBM, the Temporal Restricted Boltzmann Machine (TRBM) is put forward [26], [27]. The TRBM is a directed graphical model. It consists of a sequence of RBMs and the RBM is undirected at each time step. In TRBM, the bias of the RBM in next time step depends on the state of the previous RBMs. The advantage of TRBM is that it is able to successfully model several very high dimensional sequences, like motion capture data or the pixels of lower solution videos of balls bouncing in a box. The disadvantage of the TRBM is that it is very hard for exact inference, since computing a Gibbs update for a single variable of the posterior is exponentially expensive. In order to settle the difficult, a heuristic inference procedure is appear, which is related to the RTRBM [26].

Although the Recurrent TRBM (RTRBM) is similar to the TRBM, its performance is better than the TRBM. It

learns to use the hidden-to-hidden connections to store information, so exact inference is very easy and computing the gradient of the loglikelihood becomes feasible. However, due to it is a recurrent neural network, the disadvantage of the RTRBM is that it is difficult to learn the full potential of its hidden units.

#### *Conditional Restricted Boltzmann Machine (CRBM)*

Conditional Restricted Boltzmann Machine (CRBM) contains connections from the visible layer at previous time steps to the current hidden and visible layers. The energy function just has a small change of the RBM, it can be achieved by contrastive divergence. The shortage is that it is fail to clearly model the evolution of the hidden features without resorting to a deep network architecture.

#### *Temporal Auto encoding Restricted Boltzmann Machine (TARBM)*

The TARBM [27] is an extension of the TRBM, it simply has hidden-to-hidden temporal connections. When to pre train the temporal weights it uses a denoising auto-encoder approach, thus it has an advantage over contrastive divergence. The motivation is to gain deeper insight into the typical evolution of learned hidden layer features. Stacking the RBMs side by side through time and training the temporal connections between hidden layers use a similar way to training the AE, the difference is through time.

#### *Deep Belief Networks (DBN)*

Hinton shows that stacking and training RBMs in a greedy manner to form Deep Belief Networks (DBN) [10], [28], [29]. DBNs are graphical models. The DBN learns to extract a deep hierarchical representation of the training data. The structure of the DBN shows that the DBN has a visible layer, an output layer and multiple hidden layers, where the visible layer is also the input layer. The DBN which is composed of a stacking of RBMs can extract characteristics of more abstract and can be efficiently trained in an unsupervised and layer-by-layer manner.

The learning process is as follows: Random samples are selected as training samples, and then they are put into the network directly. The first RBM is trained, so that the hidden layer neurons can capture important features of the input data. This hidden layer is put as DBN's first hidden layer. The features are obtained by training, where the features are served as the input datas to train the second RBM. Once the first RBM is trained, another RBM will be "stacked" at the top of it to create a multilayer model. In fact, the above training process can be seen as features of the learning process and it can last until the specified layers of the DBN hidden layers are all trained. The training method of RBMs uses Contrastive Divergence (CD)[30]. The framework bypasses training the overall DBN directly, and it transforms the training of DBN into the training of multiple RBMs so as to simplify problems.

In general, the whole process is equivalent to first training RBM step by step. The model parameters are initialized to the optimal value, afterwards a small amount of traditional learning algorithm further training is processed. In this way, it not only can solve the model problem of slow training speed, but also can obtain good

effect. A large number of experiments show that this process can produce very good values of initial parameters and greatly improve modeling ability of the model.

DBN can overcome problems of traditional BP algorithm when training multilayer neural network: 1) it needs a lot of labeled training samples; 2) the convergence speed is slow; 3) because of the inappropriate parameter selection, it is easy to fall into local optimization.

#### *Convolutional Deep Belief Network (CDBN)*

By introducing a convolution operation, Lee, etc. firstly extend the processing object of the deep model from small scale image (32 pixel x 32 pixel) to large scale image pixel(200 pixel x 200 pixel) and put forward the convolutional DBN(CDBN). Through visual learning to the characteristics of each floor, it illustrates that the high-level abstraction process is constantly generated by the low-level features [31].

To introduce CDBN, it is necessary to realize CRBM at first. Convolutional RBM [32] is an extension of the RBM model and is a lot like the RBM. The CRBM corresponds to a simple structure simply with two layers: a visible layer and a hidden layer. The model uses visible matrix to represent the image, so sub windows of it represent image patches. Beside hidden units are divided into feature maps. The feature map is a binary matrix, which represents a feature at different location of image. Features are extracted from neighboring patches complement each other and they are cooperated to reconstruct the input. All nodes in the hidden and the visible layer share the same weight. Up the two layers, at an attempt to reduce the computation burden and put up with small translational misalignment, the pooling layer is imported which allows higher-layer representations to be invariant to small translations of the input.

One disadvantage of CRBM is the over completeness of features, although it uses CD learning, it is fail to deal with highly over completeness. Another disadvantage is that sampled images become highly close to the original ones after parameters are updated, hence the learning signal will disappear. We often increase Gibbs sampling steps, but it is time consuming.

Refer to the convolutional deep belief network (CDBN) [31], this architecture consists of several max-pooling-CRBMs stacked on top of one another which is similar to DBNs. The defect of RBMs and DBNs is they both ignore the 2D structure of images. But for the CDBN, it is able to exploit the 2D structure of images combined with the advantage gained by pre-training in DBN.

#### *Deep Boltzmann Machine (DBM)*

DBM [33] is a type of Markov random field, in which all connections between layers are undirected. DBM has the potential of learning internal representations that become increasingly complex at higher layers, so this is a promising way to resolve object and speech recognition issues. The approximate inference procedure, other than a bottom-up pass, can incorporate top-down feedback, which allows DBM can better propagate uncertainty about ambiguous inputs. We train the whole model online,

and process one example at a time. High-level representations are built from unlabeled inputs and labeled datas are used to slightly fine-tune the model.

However, the DBM [34] has more than one hidden layer, which increases its uncertainty and makes the learning process get quite slow, particularly when the hidden units form layers which become more and more distant from the visible units. In view of the above situation, a fast way to initialize model parameters to sensible values is described in the following. When considering initializing the model parameters of DBM, we compose the lower-level RBM and the top-level RBM to form a single system. For the lower-level RBM, we double the input and tie the visible-to-hidden weights; For the top-level RBM, we double the number of hidden units. When the two modules are composed, it can be seen that the conditional probability distributions defined by the composed model and the DBM are exactly the same. The above is greedily pre-training the two modified RBM to form a DBM. Moreover, when greedily training the RBM is more than two, it only needs the modification for the first and the last RBM in the stacking. For all the intermediate RBM's, the weights are simply halved in both directions when composing them to form a DBM.

Taking a three-layer Deep Boltzmann Machine as an example, DBM with within-layer connections is different from DBN [28] (a three-layer as example), where the top two layers form a restricted boltzmann machine which is an undirected graphical model, but the lower layers form a directed generative model.

#### *B. Deep Networks for Supervised Learning*

Next, the CNN, the DNN and the corresponding extension model are introduced.

##### *Convolutional Neural Network (CNN)*

The convolutional neural network (CNN) which first proposed by Le Cun in 1989 is a network structure [35]. CNNs belongs to the feed forward network, but it combines three architectural ideas to ensure some degree of shift and distortion invariance, they are local receptive field, shared weights, and sub-sampling. The CNNs is comprised of a sequence of convolution process and sub-sampling process. The network architecture is composed of three basic building blocks: the convolutional layer, the max-pooling layer and the classification layer. The input is converted into a convolution layer via the convolution process, and then the output feature is treated as the input data, converted as a set of smaller-dimension feature maps via the sub-sampling process [36].

CNNs are influenced by the earlier work in time-delay neural networks (TDNN) [37]. The goal of the TDNN is by sharing weights in a temporal dimension to reduce the need of learning computation, which is employed for speech and time-series processing [38]. Compared with the general neural network, the CNN has the following strengths in image processing: A) the input image and the network topology structure can be a very good match; B) feature extraction and pattern classification process simultaneously and also produce at the same time in training; C) shared weights can reduce the training of the

network parameters, which lets the neural network structure get simpler and more flexible. The disadvantages include that the implementation is more complex and the time of training is longer.

#### *Deep Neural Network (DNN)*

The deep network structure obtained by the deep learning fits the characteristics of the neural network, this is the deep neural network. The deep network structure of deep learning contains a large number of single neurons and each neuron is connected to a large number of other neurons. Connection strength between neurons changes during the learning process and determines the function of the network. Two common issues of DNN are over fitting and computation time. Because the added layers of abstraction lead them to model rare dependencies in the training data, thereby DNN tends to over fitting. In order to combat over fitting, regularization methods are used, such as weight decay, sparsity or dropout. What's more, mini-batching is used to speed up computation, due to it computes the gradient on several training examples at once rather than individual examples. In addition, researchers have been more careful to distinguish the DNNs and DBNs [39], [40].

#### *Deep Tensor Neural Network (DTNN)*

In the paper, Dong Yu extends the DNN to a novel deep tensor neural network (DTNN) [41], where one or more layers are double-projection (DP) and tensor layers. Why the author consider DTNN comes from our realization about some factors, like noisy speech, interact with each other to predict an output and so on. For the purpose of showing interactions, the author divides the input into two nonlinear subspaces through the DP layer and models the interactions between these two nonlinear subspaces and neurons of the output by a tensor with three way connections.

Generally speaking, the DTNN has two types of hidden layers: the conventional sigmoid layer and the DP layer. Each of the two types can be flexibly placed in hidden layers. The softmax layer which connects the final hidden layer to labels in the DTNN is the same with the DNN. A DTNN can be seen as the DNN augmented with DP layers. We use a unified way to train DNN and DTNN and map the input features of each layer to a vector and the tensor to a matrix.

#### *Deep Convex Network (DCN)*

To overcome the learning scalability problem, a new algorithm of deep learning-Deep Convex Network (DCN) [42] is proposed. A DCN consists of a variable number of layered modules and each module is a specialized neural network including one hidden layer and two trainable weights. In the DCN, the module consists of a first linear layer with a set of linear input units whose number equals to the dimensionality of input, a hidden layer with a series of non-linear parameter tunable units, a second linear layer with a set of linear output units. That is to say the input units of the second module include the output units of the lowest module and the raw training data and the output of a top module represents the target classification classes. The DCN blocks, each consisting of a simple and easy-to-learn module, are stacked to form the whole deep

network. When training, block-wise is used without the need of back-propagation for the entire blocks.

The DCN is called Deep Stacking Networks (DSN) later by Deng [43]. He considers for Deep Convex Network, it accentuates the role of convex optimization, but for Deep Stacking Network, it emphasizes the key operation of stacking.

#### *Tensor Deep Stacking Network (T-DSN)*

As developed in [42] and [44], each DSN block forms the basis of the T-DSN. The stacking operation of the T-DSN is exactly the same as that for the DSN described in [45]. Unlike the DSN, however, each block of the T-DSN has two sets of lower layer weight matrices  $W_{(1)}$  and  $W_{(2)}$ . They connect the input layer with two parallel branches of sigmoidal hidden layers  $H_{(1)}$  and  $H_{(2)}$ . Each T-DSN block also contains a three-way connection, the upper layer weight tensor  $U$  that connects the two branches of the hidden layer with the output layer. It changes from a matrix in DSN to a tensor in the T-DSN. This is difficult for the DSN through stacking by concatenating hidden layers with the input, in that its hidden layer is too large for practical purposes. The DSN owns the computational advantage in parallelism and scalability when learning all parameters, as a result the T-DSN reserves this superiority. The T-DSN also has an advantage in incorporating speaker or environmental factor, when training one of the hidden representations to encode speaker or environmental factor, we can effectively gate the other hidden-to-output mapping.

#### *Kernel Deep Convex Network (K-DCN)*

Deng then put forward Kernel Deep Convex Network (K-DCN) [46], in which kernel trick is used. It first bases on the DCN, and then is extended to the kernel version (resulting in K-DCN). K-DCN constructs infinite-dimensional hidden representations in each of the DCN modules using the kernel trick and gets infinite-sized hidden layers without infinite-sized parameters.

In this article, the author mentions comparing with DCN, the K-DCN vastly increases the size of hidden units avoiding subjecting to the difficulty of computation and over fitting. Parameters to tune in K-DCN are much fewer than in DCN, T-DSN, and DNN. What's more, regularization is more important in K-DCN than in DCN and T-DSN. K-DCN also can handle mixed binary and continuous-valued inputs without data and output calibration is more easily than other methods. In DNN or DCN, data normalization is often essential, but in K-DCN, it doesn't need. As a summary, the K-DCN has a lot of advantages. But it also has fault. Once the training and testing samples become very large, the scalability is a problem. We tackle it by using random Fourier features, which makes possible by stacking kernel modules to form a deep architecture.

### *C. Hybrid Deep Networks*

Hybrid model refers to the deep architecture which contains or uses the generative and discriminative model components at the same time. In the existing hybrid architectures, the main use of generative model is to help discrimination. The ultimate goal of hybrid model is

distinction, and generative model can help discriminative model. It can be accomplished by better optimization.

The existing typical generative model is usually used as discriminative task at the end. When the generative model is applied to the classification task, the pre-training can combine with other typical discriminative learning algorithms to optimize all weights. This discriminative optimization process is often attached a top-level variable to represent the expected output or label which are provided by the training set. The BP algorithm can be used to optimize the DBN weight, and this initial weight is obtained by the pre-training of RBM and DBN rather than random, so the performance of this network is often superior than just through the BP algorithm training the network. It can be seen that for the DBN training, the BP only completes local parameter search space, and it accelerates the training and convergence time, compared with the feed forward neural network.

Recently, the research based on DBNs includes applying stacking auto-encoder to replace RBMs of the traditional DBNs. This method uses the same training standard of DBNs, but the difference is that the auto-encoder uses the discriminative model. The generalization performance of the denoising auto-encoder, which brings in random changes in training process can match with the traditional DBNs. As for the training of a single denoising auto-encoder, it has no difference with the generative model of RBMs.

A hybrid deep model- DBN-DNN [47] is an example. The DBN, for unsupervised learning can be converted as the initial model of the DNN. Then for supervised learning, further discriminatively training or fine-tuning uses the target labels, which helps to make the discriminative model effectively.

To pre-train deep CNNs, the generative models of DBNs is used in which pre-training can help to improve the performance of deep CNNs based on random initialization, just like the fully connected DNN. This is also an example of hybrid deep networks [31], [48]. What's more, a similar example of hybrid deep networks is using a set of regularized deep auto encoders (DAEs, contractive AEs, and SAEs) to pre-train DNNs or CNNs.

### III. DEEP LEARNING APPLICATIONS

This article introduces the AE, the DAE, the SDAE, the CAE, the DA-IC, the CNN, the RBM, the DBN, the CDBN, the DBM, the DTNN, the DCN, the DSN, and the T-DSN and so on. Some of these models' architectures are analyzed in details. The above models are chosen as they seem to be popular and promising approaches based on the authors' personal research experiences. As for applications of these models, they have been successfully used to solve problem of different machine learning [49].

Speech is one of the earliest applications of neural network. Although the study of neural networks has been interrupted for a while, the neural network has made a breakthrough in the field of speech recognition. Around the year 2010, the voice group of Microsoft and Google both recruited the professor Hinton's students to learn, they abandoned traditional characteristics of MFCC/PLP

and used deep learning to study characteristics in speech signal. What's more, the deep learning technology was also used to contribute the acoustic model. Finally, it had a good effect in standard data sets on TIMIT.

In China, many enterprises have joined in the research of deep learning. For example, in the annual meeting of Baidu, Founder and CEO Robin Lee announced to plan to set up the institute of Baidu in January, 2013. One of the most important directions was deep learning and the Institute of Deep Learning (IDL) was also established for this research. This was the first time to establish research institute for Baidu, which had been founded more than 10 years. In April, 2013, the MIT Technology Review listed deep learning as the first of ten big breakthrough technologies of 2013. Meanwhile, Henry Markman, the neuroscientist of South Africa cooperating with other scientists hoped to simulate human brain through thousands of tests on a computer.

Since 2006, the application of deep learning in the field of target recognition has mainly focused on the question of MNIST handwritten image. It broke the hegemony position of SVM in this data set and refreshed the error rate from 1.4% to 0.27%.

In recent years, the vision of deep learning has moved from digital identification to the target identification of natural images. For example, the Google research institute also put into the research of deep learning. The theory study of deep learning is in its infancy, but it has revealed a huge energy in the field of application. Since 2011, Microsoft research and Google's speech recognition researchers have successively adopted the DNN technology to decrease the speech recognition error rate by 20% ~ 30%, which is the biggest breakthrough for more than ten years in speech recognition field.

The New York Times reported the Google Brain project in June, 2012. The guiding ideology of the Google Brain project combined the computer science and the neuroscience, which was never implemented in the field of artificial intelligence. The result of this project was that the cat was identified by the machine independent study and the Image Net evaluation error rate was reduced from 26% to 15%, which achieved astonishing results in the field of image recognition. Although its accuracy and flexibility were far less than the human brain, we believe that one day the result will achieve our desired effect.

Nair etc. put forward the modified DBN using third-order boltzman machine on the top floor. This DBN is applied to the NORB database of 3 d object recognition task, and the result closes to the best historical recognition error. In particular, he points out that the DBN is substantially better than shallow models like SVM.

Taking an example of RBMs, they have found applications in classification, dimensionality reduction, collaborative filtering, feature learning and topic modelling. The deep neural network such as convolutional DBN and stack auto-encoder network have been used for voice and audio data processing, like music artists genre classification, speaker recognition, the speaker gender classification and classification of voice, etc. Like Deep Neural Networks (DNN), compared with



the lowest error rate, the recognition error rate of this model on the Switchboard standard data sets was reduced by 33%. It is reported that the Microsoft demonstrated a fully automatic simultaneous interpretation system in Tianjin on November, 2012. The key to support it is also DNN. As for DBN, the DBN is utilized for data dimension reduction [50], information retrieval [51], human behavior analysis [52], natural language understanding [30] and other tasks. They all have obtained very good learning results.

The following is roughly a summary about applications: in addition to the most popular application: the MNIST handwriting challenge [53], there are also face detection [54], speech [55], audio and music, natural language processing [56], [57], [58], spoken language understanding [59], voice, image, modeling textures [60], modeling motion [61], language-type recognition, information retrieval [62], feature extraction, general object recognition [63], computer vision [64], and multi-modal and multi-task learning [65], dimensionality reduction [10], object segmentation [66], collaborative filtering [25] and robotics [67]. There are many other applications which are not listed in this paper. What's more, several private organizations, like Numenta [68] and Binatix [69] have paid attention to commercializing deep learning technologies with applications to a wide range of areas. Moreover, the America Defense Advanced Research Projects Agency (DARPA) declares a research project focused specially on deep learning. It can be seen that deep machine learning not only can be used in academic research, but also in organizations.

#### IV. CONCLUSION AND FINAL THOUGHTS

Deep learning as a research field of machine learning has caused more and more attention in recent years, many scholars have extensively studied in deep learning. This paper gives a summary on typical deep learning models and describes the extension of each model. The paper emphatically introduces deep networks for unsupervised and supervised learning model.

The advantage of deep learning: due to the strong model expression capability, it can handle very complex problems (such as target and behavior recognition) and learn more complex function relations. Because this method has a certain biological basis, more structure units or deep learning algorithms will be discovered in the future in order to better solve problems. Of course, deep learning also has some shortcomings: the time of training model is long; it needs constantly iteration for model optimization; it can't guarantee to get the global optimal solution and so on, which are needed to overcome in the future. Besides, deep learning theory also needs to solve the following problems: 1. Where is the deep learning theory limit, whether there is a fixed number of layers, once when we meet it, the computer can realize the artificial intelligence. 2. Whether the value of each layer unit is regular, and if there are rules to follow, when we adjust the parameters. 3. How to weigh the number of pretraining epochs, the training speed and the training accuracy. On the premise of guaranteeing the training

precision, how to improve the training speed is still need to study. 4. We can consider to merge with other methods (the method can involve deep learning algorithm or outside the deep learning), the single deep learning method often not bring the best result, the fusion of other methods may be bring higher accuracy. Therefore, deep learning method merging with other methods has a certain practical significance and research value.

In conclusion, despite a large number of researchers study the theory and experiment research of artificial neural network in recent decades, the study has made a certain progress in the field of deep learning, and the experimental results also have shown the good learning performance, but in the field of the current deep learning research there are still some problem should to be solved further. The future study of deep learning includes theoretical analysis, data representation and model, training and optimization solution, research development. Predictably, with the depth of the deep study theory and method research, deep learning will be widely used in more domains.

#### REFERENCES

- [1] G. E. Hinton and S. Osindero, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, May 2006.
- [2] Y. Bengio, "Learning deep architectures for AI," *Foundations & Trends in Machine Learning*, vol. 2, Jan. 2009.
- [3] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, "A quantitative theory of immediate visual recognition," *Progress in Brain Research*, vol. 165, no. 6, pp. 33–56, 2007.
- [4] Y. Bengio and O. Delalleau, "On the expressive power of deep architectures," *Alt*, pp. 18–36, Oct. 2011.
- [5] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, June 2014.
- [6] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," *Lecture Notes in Computer Science*, pp. 69–82, 2008.
- [7] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. 24th International Conference on Machine Learning*, New York, NY, USA: ACM, 2007, pp. 473–480.
- [8] R. Salakhutdinov and G. E. Hinton, "Using deep belief nets to learn covariance kernels for gaussian processes," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1249–1256, 2008.
- [9] M. A. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *Proc. International Conference on Machine Learning*, New York, NY, USA: ACM, 2008, pp. 792–799.
- [10] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.
- [11] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th International Conference on Machine Learning*, 2011, pp. 833–840.
- [12] Y. Luo and Y. Wan, "A novel efficient method for training sparse auto-encoders," in *Proc. 6th International Congress on Image and Signal Processing (CISP)*, 2013, pp. 1019–1023.
- [13] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising auto encoders," in *Proc. 25th International Conference on Machine Learning*, New York, USA: ACM, 2008, pp. 1096–1103.
- [14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising auto encoders: Learning useful representations in a deep network with a local denoising criterion,"



- Journal of Machine Learning Research*, vol. 11, no. 6, pp. 3371–3408, 2010.
- [15] J. T. Rolfe and Y. Lecun, “Discriminative recurrent sparse auto-encoders,” *Eprint Arxiv*, Mar. 2013.
  - [16] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proc. 14th International Conference on Artificial Intelligence and Statistics*, vol. 15, 2010, pp. 315–323.
  - [17] G. E. Hinton, “Rectified linear units improve restricted boltzmann machines vinod nair,” in *Proc. 27th International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.
  - [18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” *Readings in Cognitive Science*, vol. 1, pp. 399–421, Mar.–Sep. 1986.
  - [19] Y. Bengio and F. Gingras, “Recurrent neural networks for missing or asynchronous data,” *Advances in Neural Information Processing Systems*, pp. 395–401, 1996.
  - [20] P. Simard, Y. Lecun, J. S. Denker, and B. Victorri, “Transformation invariance in pattern recognition-tangent distance and tangent propagation,” *Neural Networks: Tricks of the Trade*, pp. 239–274, 1998.
  - [21] H. Larochelle, D. Erhan, and P. Vincent, “Deep learning using robust interdependent codes,” in *Proc. 12th International Conference on Artificial Intelligence and Statistics*, 2009, pp. 312–319.
  - [22] O. Shriki, H. Sompolinsky, and D. D. Lee, “An information maximization approach to over complete and recurrent representations,” *Advances in Neural Information Processing Systems*, pp. 612–618, 2002.
  - [23] J. Masci, U. Meier, D. C., and J. Schmidhuber, “Stackedconvolutional auto-encoders for hierarchical feature extraction,” in *Proc. International Conference on Artificial Neural Networks*, June 14–17, 2011, pp. 52–59.
  - [24] A. Fischer and C. Igel, “Training restricted boltzmann machines: An introduction,” *Pattern Recognition*, vol. 47, no. 1, pp. 25–39, Jan. 2014.
  - [25] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted boltzmann machines for collaborative filtering,” in *Proc. 24th International Conference on Machine Learning*, New York, NY, USA: ACM, 2007, pp. 791–798.
  - [26] I. Sutskever, G. Hinton, and G. Taylor, “The recurrent temporal restricted boltzmann machine,” *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2008.
  - [27] C. H. ¨ausler and A. Susemihl, “Temporal auto encoding restricted boltzmann machine,” *Eprint Arxiv*, Oct. 2012.
  - [28] M. Ranzato, F. J. Huang, Y. L. Boureau, and Y. Lecun, “Unsupervised learning of invariant feature hierarchies with applications to object recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.
  - [29] Y. Bengio, P. Lamblin, P. Dan, H. Larochelle, U. D. Montral, and M. Qubec, “Greedy layer-wise training of deep networks,” *Advances in Neural Information Processing Systems (NIPS)*, 2007.
  - [30] R. Sarikaya, G. E. Hinton, and A. Deoras, “Application of deep belief networks for natural language understanding,” in *Proc. IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 4, Apr. 2014, pp. 778–784.
  - [31] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proc. 26th International Conference on Machine Learning*, New York, NY, USA: ACM, 2009, pp. 609–616.
  - [32] M. Norouzi, M. Ranjbar, and G. Mori, “Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning,” *Computer Vision and Pattern Recognition (CVPR)*, pp. 2735–2742, June 2009.
  - [33] N. Srivastava, R. R. Salakhutdinov, and G. E. Hinton, “Modeling documents with deep boltzmann machines,” in *Proc. 29th Conference on Uncertainty in Artificial Intelligence*, Sep. 2013.
  - [34] R. Salakhutdinov and G. E. Hinton, “Deep Boltzmann machines,” in *Proc. 12th Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.
  - [35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proc. the IEEE*, vol. 86, no. 11, Nov. 1998, pp. 2278–2324.
  - [36] A. K. I. S. G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems (NIPS)*, 2012.
  - [37] M. Sugiyama, H. Sawai, and A. H. Waibel, “Review of TDNN (time delay neural network) architectures for speech recognition,” in *Proc. IEEE International Symposium on Circuits and Systems*, June 1991, pp. 582–585.
  - [38] R. Sitte and J. Sitte, “Neural networks approach to the random walk dilemma of financial time series,” *Applied Intelligence*, vol. 16, no. 3, pp. 163–171(9), May. 2002.
  - [39] G. E. Dahl, S. Member, Y. Dong, S. Member, D. Li, and A. Acero, “Context-dependent pre-trained deep neural networks for large vocabulary speech recognition,” *IEEE Transactions on Audio Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
  - [40] A. R. Mohamed, D. Yu, and L. Deng, “Investigation of full-sequence training of deep belief networks for speech recognition,” *Interspeech*, 2010.
  - [41] D. Yu, L. Deng, and F. Seide, “Large vocabulary speech recognition using deep tensor neural networks,” *Proc Interspeech*, 2012.
  - [42] L. Deng and D. Yu, “Deep convex net: A scalable architecture for speech pattern classification,” in *Proc. 12th Annual Conference of the International Speech Communication Association*, 2011.
  - [43] D. Li, B. Hutchinson, and Y. Dong, “Parallel training of deep stacking networks,” *Interspeech*, 2012.
  - [44] L. Deng and D. Yu, “Deep convex networks for image and speech classification,” in *Proc. Icml Workshop on Learning Architectures*, 2011.
  - [45] B. Hutchinson, L. Deng, and D. Yu, “Tensor deep stacking networks,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2013.
  - [46] P. S. Huang, L. Deng, M. Hasegawa-Johnson, and X. He, “Random features for kernel deep convex network,” in *Proc. the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3143–3147.
  - [47] G. Hinton, D. Li, Y. Dong, G. Dahl, A. R. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
  - [48] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Unsupervised learning of hierarchical representations with convolutional deep belief networks,” *Communications of the Acm*, vol. 54, no. 10, pp. 95–103, Oct. 2011.
  - [49] I. Arel, D. C. Rose, and T. P. Karnowski, “Deep machine learning—a new frontier in artificial intelligence research [research frontier],” *IEEE on Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, Nov. 2010.
  - [50] R. Salakhutdinov, “Learning deep generative models,” *Topics in Cognitive Science*, vol. 3, no. 1, pp. 74–91, 2009.
  - [51] A. Torralba, R. Fergus, and Y. Weiss, “Small codes and large image databases for recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.
  - [52] G. W. Taylor, G. E. Hinton, and S. Roweis, “Modeling human motion using binary latent variables,” *Advances in Neural Information Processing Systems*, 2006.
  - [53] Y. Lecun and C. Cortes, “The MNIST database of handwritten digits,” in *Proc. International Conference on Auditory Display*, 1998.
  - [54] B. Kwolek, “Face detection using convolutional neural networks and gabor filters,” in *Proc. International Conference on Artificial Neural Networks*, Poland, Sep. 2005, pp. 551–556.
  - [55] G. Wang and K. C. Sim, “Regression-based context-dependent modeling of deep neural networks for speech recognition,” *IEEE/Association for Computing Machinery (ACM) Transactions on Audio Speech and Language Processing*, vol. 22, no. 11, pp. 1660–1669, Nov. 2014.
  - [56] A. Mnih and G. E. Hinton, “A scalable hierarchical distributed language model,” *Advances in Neural Information Processing Systems*, 2008.
  - [57] H. Lee, P. T. Pham, L. Yan, and A. Y. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” *Advances in Neural Information Processing Systems*, pp. 1096–1104, 2009.
  - [58] J. Weston, F. Ratle, and R. Collobert, “Deep learning via semi-supervised embedding,” in *Proc. 25th International Conference on Machine Learning*, New York, NY, USA: ACM, 2008, pp. 1168–1175.

- [59] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, and D. Yu, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 23, no. 3, pp. 530–539, Mar. 2015.
- [60] S. O. G. Hinton, "Modeling image patches with a directed hierarchy of markov random fields," *Advances in Neural Information Processing Systems*, 2008.
- [61] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," in *Proc. 26th Annual International Conference on Machine Learning*, New York, NY, USA: ACM, 2009, pp. 1025–1032.
- [62] R. Salakhutdinov and G. Hinton, "Semantic hashing," in *Proc. Special Interest Group on Information Retrieval (SIGIR) Workshop on Information Retrieval and Applications of Graphical Models*, vol. 50, no. 7, pp. 969–978, 2009.
- [63] F. J. Huang and Y. Lecun, "Large-scale learning with svm and convolutional for generic object categorization," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 2006, pp. 284–291.
- [64] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high level features using large scale unsupervised learning," in *Proc. the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8595–8598.
- [65] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov, "Devise: A deep visual-semantic embedding model," *Advances in Neural Information Processing Systems*, 2013.
- [66] I. Levner, "Data driven object segmentation," PhD thesis, Department of Computer Science, University of Alberta, 2006.
- [67] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller, and Y. Lecun, "Deep belief net learning in a long-range vision system for autonomous off-road driving," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 2008, pp. 628–633.
- [68] Numenta. (2014). [Online]. Available: <http://www.numenta.com>
- [69] Binatix. (2014). [Online]. Available: <http://www.binatix.com>



**Shuo Hu.** The author received a B.Sc. in Electronics and Information System and M.Sc. in Circuits and Systems from the Northeast Normal University in 2000 and 2003, respectively. He received a Ph.D. in Optical Engineering from Institute of Optics Fine Mechanics and Physics, Chinese Academy of Sciences, in 2006. He was an assistant research fellow in State Key Laboratory of Multimedia Computing,

Hisense Groups. Currently, he is an associate professor at Institute of Electrical Engineering, Yanshan University. His research interests include activity recognition, video surveillance and time series analysis.



**Yaqing Zuo** received the bachelor degree in automation from Yanshan University, China, in 2013. She is currently a master student at the Institute of Electrical Engineering, Yanshan University. Her research interest covers active deep learning of remote sensing image classification.



**Peng Liu** received the M.S. degree in 2004 and the Ph.D. degree in 2009, both in signal processing, from Chinese Academic of Science. From 2009–now, he is an assistant professor at the Center for Earth Observation and Digital Earth, Chinese Academy of Sciences. From May 2012 to May 2013, he is with Department of Electrical and Computer Engineering, George Washington University as a Visiting Scholar. His research is focused on sparse representation, big data, compressive sensing, image processing and its applications to remote sensing.



**Lizhe Wang** is a "ChuTian" Chair Professor at School of Computer Science, China Univ. of Geosciences (CUG), and a Professor at Inst. of Remote Sensing & Digital Earth, Chinese Academy of Sciences (CAS). Prof. Wang received B.E. & M.E from Tsinghua Univ. and Doctor of Eng. from Univ. Karlsruhe (Magna Cum Laude), Germany. Prof. Wang is a Fellow of IET, Fellow of British Computer Society. Prof. Wang serves as an Associate Editor of IEEE Tran. Computers and IEEE Tran. on Cloud Computing. His main research interests include high performance computing, e-Science, and spatial data processing