A Survey of Deep Neural Network Architectures and Their Applications

Weibo Liu[a], Zidong Wang[a,*], Xiaohui Liu[a], Nianyin Zeng[b], Yurong Liu[c,d] and Fuad E. Alsaadi[d]

Abstract

Since the proposal of a fast learning algorithm for deep belief networks in 2006, the deep learning techniques have drawn ever-increasing research interests because of their inherent capability of overcoming the drawback of traditional algorithms dependent on hand-designed features. Deep learning approaches have also been found to be suitable for big data analysis with successful applications to computer vision, pattern recognition, speech recognition, natural language processing, and recommendation systems. In this paper, we discuss some widely-used deep learning architectures and their practical applications. An up-to-date overview is provided on four deep learning architectures, namely, autoencoder, convolutional neural network, deep belief network, and restricted Boltzmann machine. Different types of deep neural networks are surveyed and recent progresses are summarized. Applications of deep learning techniques on some selected areas (speech recognition, pattern recognition and computer vision) are highlighted. A list of future research topics are finally given with clear justifications.

Index Terms

Autoencoder, Convolutional Neural Network, Deep Learning, Deep Belief Network, Restricted Boltzmann Machine

I. Introduction

Machine learning techniques have been widely applied in a variety of areas such as pattern recognition, natural language processing and computational learning. With machine learning techniques, computers are endowed with the capability of acting without being explicitly programmed, constructing algorithms that can learn from data, and making data-driven decisions or predictions. During the past decades, machine learning has brought enormous influence on our daily life with examples including efficient web search, self-driving systems, computer vision, and optical character recognition. In addition, by adopting machine learning methods, the human-level artificial intelligence (AI) has been improved as well, see [101], [137], [165] for more discussions. Nevertheless, when it comes to the human information processing mechanisms (e.g. speech and vision), the performance of traditional machine learning techniques are far from satisfactory. Inspired by deep hierarchical structures of human speech perception and production systems, the concept of deep learning algorithms was introduced in the late 20th century. Breakthroughs on deep learning have been achieved since 2006 when Hinton proposed a novel deep structured learning architecture called deep belief network (DBN) [59]. The past decade has seen rapid developments of

This work was supported in part the Royal Society of the UK, the National Natural Science Foundation of China under Grants 61329301, 61374010, and 61403319, and the Alexander von Humboldt Foundation of Germany.

[a] Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. Email address: Zidong.Wang@brunel.ac.uk

[b] Department of Instrumental and Electrical Engineering, Xiamen University, Xiamen 361005, Fujian, China

[c] Department of Mathematics, Yangzhou University, Yangzhou 225002, China

[d] Communication Systems and Networks (CSN) Research Group, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia.

[*] Corresponding author

deep learning techniques with significant impacts on signal and information processing. Research on neuromorphic systems also supports the development of deep network models [75]. In contrast to traditional machine learning and artificial intelligence approaches, the deep learning technologies have recently been progressing massively with successful applications to speech recognition, natural language processing (NLP), information retrieval, compute vision, and image analysis [91], [125], [159].

The concept of deep learning originated from the study on artificial neural networks (ANNs) [60]. ANNs have become an active research area during the past few decades [63], [162], [166], [167], [175]. To construct a standard neural network (NN), it is essential to utilize neurons to produce real-valued activations and, by adjusting the weights, the NNs behave as expected. However, depending on the problems, the process of training a NN may take long causal chains of computational stages. Backpropagation is an efficient gradient descent algorithm which has played an important role in NNs since 1980. It trains the ANNs with a teacher-based supervised learning approach. Although the training accuracy is high, the performance of backpropagation when applied to the testing data might not be satisfactory. As backpropagation is based on local gradient information with a random initial point, the algorithm often gets trapped in local optima. Furthermore, if the size of the training data is not big enough, NNs will face the problem of overfitting. Consequently, other effective machine learning algorithms such as support vector machine (SVM), boosting and K-nearest neighbour (KNN) have been adopted to obtain global optimum with lower power consumption. In 2006, Hinton [59] proposed a new training method (called layer-wise-greedy-learning) which marked the birth of deep learning techniques. The basic idea of the layer-wise-greedy-learning is that unsupervised learning should be performed for network pre-training before the subsequent layer-by-layer training. By extracting features from the inputs, the data dimension is reduced and a compact representation is hence obtained. Then, exporting the features to the next layer, all of the samples will be labeled and the network will be fine-tuned with the labeled data. The reason for the popularity of deep learning is twofold: on one hand, the development of big data analysis techniques indicates that the overfitting problem in training data can be partially solved; on the other hand, the pre-training procedure before unsupervised learning will assign non-random initial values to the network. Therefore, a better local minimum can be reached after the training process and a faster converge rate can be achieved.

Up to now, the research on deep learning techniques has stirred a great deal of attention and a series of exciting results have been reported in the literatures. Since 2009, the ImageNet's competition has attracted a great many computer vision research groups throughout the world from both academia and industry. In 2012, the research group led by Hinton won the competition of ImageNet Image Classification by using deep learning approaches [86]. Hinton's group attended the competition for the first time and their results were 10% better than that in the second place. Both Google and Baidu have updated their image search engines based on Hinton's deep learning architecture with great improvements in searching accuracy. Baidu also set up the Institute of Deep Learning (IDL) in 2013 and invited Andrew Ng, the associate professor at Stanford University, as the Chief Scientist. In March 2016, a Go Game match was held in South Korea by Google's deep learning project (called DeepMind) between their AI player AlphaGo and one of the world's strongest players Lee Se-dol [140]. It turned out that AlphaGo, adopting deep learning techniques, showed surprising strength and beat Lee Se-dol by 4:1. In addition, deep learning algorithms have also shown outstanding performance in predicting the activity of potential drug molecules and the effects of mutations in non-coding DNA on gene expression.

With rapid development of computation techniques, a powerful framework has been provided by ANNs with deep architectures for supervised learning. Generally speaking, the deep learning algorithm consists of a hierarchical architecture with many layers each of which constitutes a non-linear information processing unit. In this paper, we only discuss deep architectures in NNs. Deep neural networks (DNNs), which employ deep architectures in NNs,

can represent functions with higher complexity if the numbers of layers and units in a single layer are increased. Given enough labeled training datasets and suitable models, deep learning approaches can help humans establish mapping functions for operation convenience. In this paper, four main deep architectures are recalled and other methods (e.g. sparse coding) are also briefly discussed. Additionally, some recent advances in the field of deep learning are described.

The purpose of this article is to provide a timely review and introduction on the deep learning technologies and their applications. It is aimed to provide the readers with a background on different deep learning architectures and also the latest development as well as achievements in this area. The rest of the paper is organized as follows. In Sections II-V, four main deep learning architectures, which are restricted Boltzmann machines (RBMs), deep belief networks (DBNs), autoencoder (AE), and convolutional neural networks (CNNs), are reviewed, respectively. Comparisons are made among these deep architectures and recent developments on these algorithms are discussed. The applications of those deep architectures are highlighted in Section VI. Conclusions and future topics of research are presented in Section VII.

II. Deep Learning Architectures: Restricted Boltzmann Machine

A. The motivation

In this part, a brief review of RBMs is given. RBMs are widely used in deep learning networks on account of their historical importance and relative simplicity. The RBM was first proposed as a concept by Smolensky, and has become prominent since Hinton published his work [59] in 2006. RBMs have been used to generate stochastic models of ANNs which can learn the probability distribution with respect to their inputs. RBMs consist of a variant of Boltzmann machines (BMs). BMs can be interpreted as NNs with stochastic processing units connected bidirectionally. Since it is difficult to learn aspects of an unknown probability distribution, RBMs have been proposed to simplify the topology of the network and to enhance the efficiency of the model. It is well recognized that an RBM is a special type of Markov random fields with stochastic visible units in one layer and stochastic observable units in the other layer.

B. The structure and the algorithm

As shown in Figure 1, the neurons are restricted to form a bipartite graph in an RBM. It can be seen that there is a full connection between the visible units and the hidden ones, while no connection exists between units from the same layer [165]. To train an RBM, the Gibbs sampler is adopted. Starting with a random state in one layer and performing Gibbs sampling, we can generate data from an RBM. Once the states of the units in one layer are given, all the units in the other layers will be updated. This update process will carry on until the equilibrium distribution is reached. Next, the weights within an RBM are obtained by maximizing the likelihood of this RBM. Specifically, taking the gradient of the log-probability of the training data, the weights can be updated according to:

$$\frac{\partial \log p(v^0)}{\partial \omega_{ij}} = \langle v_i^0 h_j^0 \rangle - \langle v_i^\infty h_j^\infty \rangle, \tag{1}$$

where ω_{ij} represents the weight between the visible unit i and the hidden unit j. $\langle v_i^0 h_j^0 \rangle$ and $\langle v_i^\infty h_j^\infty \rangle$ are the correlations when the visible and hidden units are in the lowest layer and the highest layer, respectively. The detailed proof can be found in [59]. It should be noted that the training process will be more efficient when using the gradient-based contrastive divergence (CD) algorithm. The CD algorithm for RBM training was developed by Hinton in 2002 [56]. The procedure of the k-step CD algorithm is given in Algorithm 1.

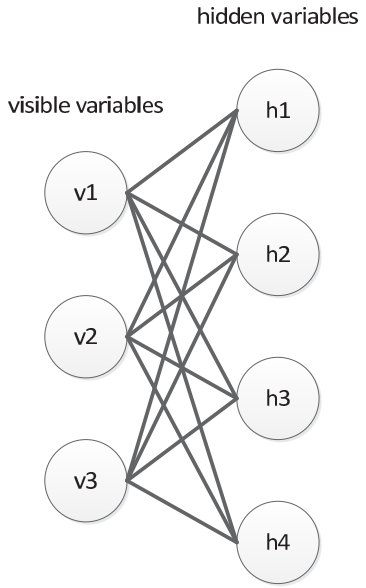


Fig. 1. Schematic diagram of RBMs

Algorithm 1 k-step contrastive divergence for RBMs

Input: RBM(V_1,...,V_m,H_1,...,H_n), training period T, learning rate ϵ

Output: The RBM weight matrix ω, gradient approximation $\Delta\omega_{ij}$, Δa_i and Δb_j for $i = 1, ..., m$, $j = 1, ..., n$

Initialize ω with random values distributed uniformly in $[0, 1]$, $\Delta\omega_{ij} = \Delta a_i = \Delta b_j = 0$ for $i = 1, ..., m$, $j = 1, ..., n$

For $\forall$ t = 1 : T do,

Sample $v_i^{(t)} \quad P(v_i|h^{(t)})$ when $i = 1, ..., m$

Sample $h_j^{(t+1)} \quad P(h_j|v^{(t)})$ when $j = 1, ..., n$

For $i = 1, ..., m$, $j = 1, ..., n$ do

$\Delta\omega_{ij} = \Delta\omega_{ij} + \epsilon \times [P(H_j = 1|v^{(0)}) \times v_i^{(0)} - P(H_j = 1|v^{(T)}) \times v_i^{(T)}]$

$\Delta a_i = \Delta a_i + \epsilon \times (v_i^{(0)} - v_i^{(T)})$

$\Delta b_j = \Delta b_j + \epsilon \times [P(H_j = 1|v^{(0)}) - P(H_j = 1|v^{(T)})]$

$\omega = \omega + \epsilon \times \Delta\omega_{ij}$

End For

End For

Assuming that the difference between the model and the target distribution is not large, we can use the samples generated by the Gibbs chain to approximate the negative gradient. Ideally, as the length of the chain increases, its contribution to the likelihood decreases and tends to zero [12]. However, in [147], we can find that the estimation of the gradient cannot represent the gradient itself. Moreover, most CD components and the corresponding log-likelihood gradient have equal signs [45]. Hence, a more practical algorithm called persistent contrastive divergence was proposed in [115]. In this approach, the authors suggested to trace the states of persistent chains rather than searching the initial value of the Gibbs Markov chain at a given data vector. The states of the hidden and visible units in the persistent chain are renewed following the update of each weight. In this way, even a small learning rate will not cause much difference between the updates and the persistent chain states while bringing more accurate

estimates.

C. Variations of RBMs

Nowadays, RBMs are playing an important role in various applications such as topic modeling, dimensionality reduction, collaborative filtering, classification and feature learning. For example, an RBM can be used to encode the data and then applied to unsupervised learning for regression or classification. Additionally, the RBMs can be used as a generative model. We can calculate the joint distribution of the visible and hidden units $P(v, h)$ with the Bayesian law. The conditional probability of a single unit $p(h|v)$ can also be calculated with RBMs. Therefore, an RBM can also be used as a discriminative model.

Generally, RBMs are used as feature extractors in the pre-training process for classification tasks. However, the features extracted by the RBMs in unsupervised learning may not be useful in the supervised learning process. In addition, the selection of parameters, which is critical to the performance of learning algorithms, will also bring difficulties. To handle these problems, discriminative restricted Boltzmann machines (DRBMs) was proposed by Larochelle and Bengio in 2008. Furthermore, for online learning with big datasets, the model of hybrid DBRMs (HDRBMs) performs well due to their combined advantages of both generative and discriminative learning. In multi-label classification tasks, however, the performance of RBMs is not satisfactory. Mnih et al. [115] proposed the so-called conditional restricted Boltzmann machines (CRBMs) for further performance improvement. Meanwhile, in high-dimensional time series, the CRBMs can be used as non-linear generative models. In [153], an undirected model is established with real-valued visible variables and binary latent ones. In this model, the visible variables at the last few time-steps can be directly influenced by the latent and visible variables at each time step. With this property, online inference can be carried out more efficiently by the CRBMs. In addition, learning from time series, the CRBMs are able to obtain rich distributed representations in order to guarantee the efficiency of accurate inference.

Recently, a self-contained DRBM (called FE-RBM) was developed by Elfwing based on a novel discriminative learning algorithm [41]. In the FE-RBM, the output for any input and class vectors is computed according to the negative free energy of an RBM. The learning objective is achieved through minimizing the mean-squared training error using a stochastic gradient descent method. Moreover, inspired by the previous research, the free energy is scaled by a constant based on the network size to improve the robustness of function approximation in the FE-RBMs.

When RBMs are applied to areas like image and speech recognition, their performance may be severely degraded by the noises in the data [55]. In 2012, Tang et al. [152] introduced a state-of-the-art model, the robust Boltzmann machine (RoBM), which can be used to deal with noises and occlusions in visual recognition. With the RoBM, a better generalization can be achieved by eliminating the influence of corrupted pixels. Trained with unlabeled data with noises using unsupervised leaning algorithms, the RoBM model can also learn the spatial structure of the occluders. Compared with traditional algorithms, the RoBMs have shown enhanced performance in various applications such as image inpainting and facial recognition.

As a key factor in the Boltzmann distribution, temperature is, for the first time, taken into consideration in the graphical model of DBNs by Li et al. [97]. The temperature based restricted Boltzmann machines (TRBMs) were proposed where the temperature acts as an independent parameter to be adjusted. Theoretical analysis reveals that the temperature is a key factor that controls the selectivity of the firing neurons in the hidden layers. It is proved that the performance of the proposed TRBMs can be enhanced by properly setting the sharpness parameter of the logistic function. Since an extra level of flexibility is introduced, the TRBMs can obtain more accurate results. Furthermore, the research also provides some insights into the RBMs from a physical point of view which indicates that there may exist some relationship between the temperature and some real-life NNs.

III. Deep Learning Architectures: Deep Belief Network

A. The motivation

As mentioned in the previous section, the hidden and visible variables are not mutually independent [165]. To explore the dependencies between these variables, in 2006, Hinton constructed the DBNs by stacking a bank of RBMs. Specifically, the DBNs are composed of multiple layers of stochastic and latent variables and can be regarded as a special form of the Bayesian probabilistic generative model. Compared with ANNs, DBNs are more effective, especially when applied to problems with unlabeled data.

B. The structure and the algorithm

The schematic diagram of the model is shown below in Figure 2.

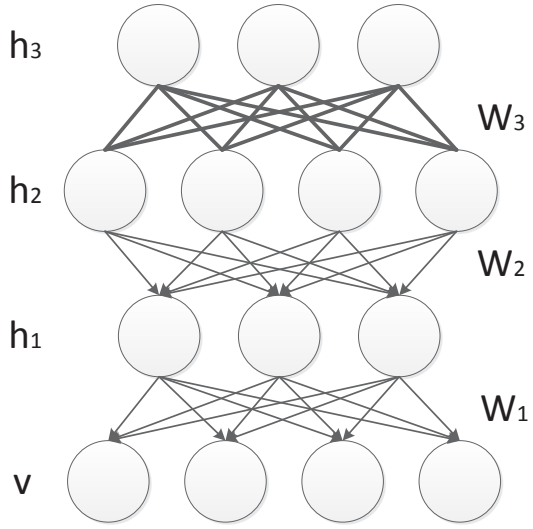


Fig. 2. Schematic Diagram of DBNs

It can be seen from Figure 2 that in a DBN, every two adjacent layers form an RBM. The visible layer of each RBM is connected to the hidden layer of the previous RBM and the top two layers are non-directional. The directed connection between the above layer and the lower layer is in a top-down manner. Different layers of RBMs in a DBN are trained sequentially: the lower RBMs are trained first, then the higher ones. After features are extracted by the top RBM, they will be propagated back to the lower layers [30]. Compared with a single RBM, the stacked model will increase the upper bound of the log-likelihood, which implies stronger learning abilities [5].

The training process of a DBN can be divided into two stages: the pre-training stage and the fine-tuning stage. In the pre-training stage, an unsupervised learning based training is carried out in the down-up direction for feature extraction; while in the fine-tuning stage, a supervised learning based up-down algorithm is performed for further adjustment of the network parameters. We note that the improved performance of the DBNs can be largely attributed to the pre-training stage in which the initial weights of the network are learned from the structure of the input data. Compared with the randomly initialized ones, these weights are closer to the global optima and can therefore bring better performance.

The CD algorithm introduced in the previous section can be used to pre-train a DBN. The performance, however, is usually unsatisfactory especially when the input data are clamped. To overcome this problem, a greedy layer-by-layer learning algorithm was introduced which optimizes the weights of a DBN at time complexity linear to

the size and depth of the network [59]. In the greedy layer-by-layer learning algorithm, the RBMs that constitute a DBN are trained sequentially. Specifically, the visible layer of the lowest RBM is trained first with $h^{(0)}$ as the input. The values in the visible layer are then imported to the hidden layers where the activation probabilities $P(h|v)$ of the hidden variables are calculated. The representation obtained in the previous RBM will be used as the training data for the next RBM and this training process continues until all the layers are traversed. Since in this algorithm, the approximation of the likelihood function is only required in one step, the training time has been significantly reduced. The underfitting problem that usually occurs in deep networks can also be overcome in the pre-training process. This pre-training algorithm is also called the greedy layer-by-layer unsupervised training algorithm. For clarity, we have provided its implementation procedure in Algorithm 2.

Algorithm 2 Greedy layer-by-layer algorithm for DBN

Input: Input visible vector v_{in}, training period T, learning rate ϵ, number of layers J.

Output: Weight matrix ω^i of layer $i, i = 1, 2, ..., J-2$.

Initialize ω with random values from 0 to 1, $h^0 = v_{in}$; where h^0 denotes the value of units in the input layer. h^i represents the units' value of the ith layer. layer = 1;

for $\forall$ t = 1 : T do,

for $layer < L$, do gibbs sampling h^{layer} using $P(h^{layer}|h^{layer-1})$

Computing the CD in Algorithm 1, $\omega(layer)$ is achieved using $\omega(t+1) = \omega(t) + \epsilon \times \Delta\omega$

End for

End for

In the fine-tuning stage, the DBNs are trained with labeled data by the up-down algorithm which is a contrastive version of the wake-sleep algorithm [57]. To find out the category boundaries of the network, a set of labels are set to the top layer for the recognition weights learning process. Also, the backpropagation algorithm is used to fine-tune the weights with labeled data [149]. Compared with the original wake-sleep algorithm, the up-down algorithm does not suffer from the problems of mode-averaging which may bring poor recognition weights.

To summarize, the training process of a DBN includes an unsupervised layer-by-layer pre-training procedure performed in a bottom-up manner and a supervised up-down fine-tune process. The pre-training process can be regarded as feature learning through which a better initial value for the weights can be obtained, and the up-down algorithm is then used to adjust the whole network. It's worthy to mention that with DBNs, the unlabeled data is processed effectively. Moreover, the overfitting and underfitting problems can also be avoided [30].

C. Variations of DBNs

In 2009, Nair and Hinton [121] introduced a top-level model for DBNs and evaluated it on a 3D object recognition task. A third-order Boltzmann machine is used as the top-level model and trained by a hybrid algorithm which combines both generative and discriminative gradients. Based on Indiveri and Liu's work [75], it is claimed that the brain-inspired processor architectures are support models of DNNs and cortical networks. Moreover, it was proved that the complementary priors can be used to overcome the inference difficulty in densely connected belief networks. In 2008, Salakhutdinov and Hinton [136] introduced a method to learn a good covariance kernel for a Gaussian process with unlabeled data and a DBN. Compared with a normal kernel based on the raw input, a Gaussian kernel performs better if the data sets are in high dimensions and highly structured.

Due to successful applications of the DBNs to the TIMIT Acoustic-Phonetic Continuous Speech Corpus benchmark, researchers are motivated to deal with a much more challenging task, the large vocabulary topic. It was proved that for such a task, training DBNs is computationally more difficult. Although the backpropagation of

stochastic gradient descent has shown its power in the fine-tune step, it is difficult to modify the learning process especially for a large-scale dataset. On the basis of an extreme powerful GPU machine, it is possible to train a deep architecture for dozens of speech recognizers using a large quantity of speech training data with remarkable results. However, it won't be able to obtain acceptable results with only one GPU machine since current architectures cannot guarantee the training efficiency. Hence, Deng and Yu [35] proposed a novel deep architecture, referred to as deep convex networks (DCNs), to overcome the shortcomings in learning scalability. The DCNs consist of a variety of layered modules. One module is formed with a single hidden layer as well as two sets of weights in a special neural network. More specifically, the lowest module is composed of two linear layers and a non-linear layer. One linear layer contains the input variables and the other one contains output variables. Besides, the non-linear layer contains nonlinear input variables. The learning method in DCNs is batch-mode based which leads to a parallel training. Additionally, the performance of DCNs can be improved by the structure-exploited fine-tuning process.

Compared with standard classification algorithms such as SVM and KNN, DBNs can also be employed in image classification because of their outstanding performance in feature learning. Based on the greedy layer-by-layer unsupervised training algorithm, Abdel et al. [1] proposed an automatic diagnosis system which includes a DBN for pre-training and a backpropagation NN for fine-tuning. Compared with the standard NN with only one supervised phase, the diagnosis system can achieve higher classification accuracy.

Recently, Liao et al. [100] proposed a novel image retrieval method which is based on DBNs and a Softmax Classifier. The standard Content-Based Image Retrieval (CBIR) algorithm that exploits automated feature extraction methods is employed to retrieve similar images from the database. However, the image feature representation is not as good as expected. It is shown that the DBN-Softmax model obtains higher precision and better recall than previous ones, such as the shape-based algorithm and the perceptual hash algorithm. Generally, based on simulations of the human visual system architecture, DBN-Softmax can provide a valid representation and extraction measurement more effectively than the standard algorithms in which a threshold is required to be set manually based on the hamming distance computation.

To increase the flexibility of DBNs, a novel model of convolutional deep belief networks (CDBNs) was introduced [3]. As the inputs should be vectorized as an image matrix, two-dimensional (2-D) structure information such as an input image cannot be imported as input directly in DBNs. However, in CDBNs, features of high dimensional images can be extracted. Although the greedy layer-by-layer algorithm plays an important role in training DBNs, many other deep learning techniques have also been investigated. In 2009, Bengio [10] claimed that we can regard each pair of layers of the DBN as a denoising autoencoder (DAE).

IV. Deep Learning Architectures: Autoencoder

A. The motivation

An autoencoder (AE), which is another type of ANNs, is also called an autoassociator. It is an unsupervised learning algorithm used to efficiently code the dataset for the purpose of dimensionality reduction [10], [60], [61], [137]. During the past few decades, the AEs have been at the cutting edge among researches on the ANN. In 1988, Bourlard and Kamp [15] found that a multilayer perceptron (MLP) in auto-association mode could achieve data compression and dimensionality reduction in the areas like information processing.

Recently, the AEs have been employed to learn generative models of data [30]. The input data is first converted into an abstract representation which is then converted back into the original format by the encoder function. More specifically, it is trained to encode the input into some representation so that the input can be reconstructed from that representation. Essentially, the AE tries to approximate the identity function in this process. One key advantage of the AE is that this model can extract useful features continuously during the propagation and filter the useless

information. Besides, since the input vector is transformed into a lower dimensional representation in the coding process, the efficiency of the learning process can be enhanced.

B. The structure and the algorithm

The AE is a one-hidden-layer feed-forward neural network similar to the MLP [13]. The difference between an MLP and an AE is that the aim of the AE is to reconstruct the input, while the purpose of the MLP is to predict the target values with certain inputs. The numbers of nodes in the input layer and the output layer are identical. In the coding process, the AE first converts the input vector x into a hidden representation h using a weight matrix ω; then in the decoding process, the AE maps h back to the original format to obtain $\tilde{x}$ with another weight matrix ω'. Theoretically, ω' should be the transpose of ω. Parameter optimization is adopted in order to minimize the average reconstruction error between x and $\tilde{x}$. Mean square errors (MSEs) are used to measure the reconstruction accuracy according to assumed distribution of the input features [101]. The schematic diagram of the model is shown below in Figure 3.

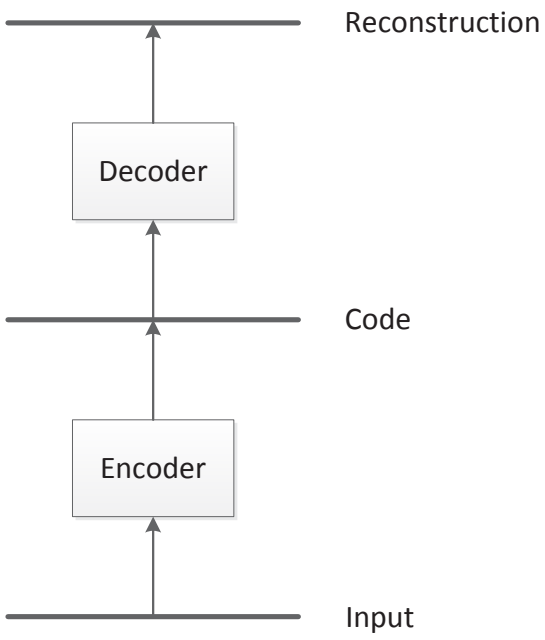


Fig. 3. Schematic Diagram of AEs

Similar to that for the DBNs, the training process for an AE can also be divided into two stages: the first stage is to learn features using unsupervised learning and the second is to fine-tune the network using supervised learning. To be specific, in the first stage, feed-forward propagation is first performed for each input to obtain the output value $\tilde{x}$. Then squared errors are used to measure the deviation of $\tilde{x}$ from the input value. Finally, the error will be backpropagated through the network to update the weights. In the fine-tuning stage, with the network having suitable features at each layer, we can adopt the standard supervised learning method and the gradient descent algorithm to adjust the parameters at each layer.

C. Variations of AEs

In 2008, Vincent et al. [154], [155] proposed the DAEs for denoising the traditional AEs. The DAE intentionally adds noises into the training data and trains the AEs with these corrupted data. Through the training process, the DAE can recover the noise-free version of the training data, which implies an enhanced robustness. Compared with RBMs, some standard optimization methods can be used in DAEs [101], [155]. It should be noted that by exploiting the statistical dependencies inherent in the inputs, the DAE will undo the adverse effects of the noisy inputs corrupted in a stochastic manner. The objective function for optimization in the DAE is shown in Equation 2:

$$J_{DAE} = \sum_{t} IE_{q(\tilde{x}|x^{(t)})}[L(x^{(t)}, g_\theta(f_\theta(\tilde{x})))], \tag{2}$$

where $IE_{q(\tilde{x}|x^{(t)})}[L(x^{(t)}, g_\theta(f_\theta(\tilde{x})))]$ represents the average value over corrupted data $\tilde{x}$ drawn from the corruption procedure $q(\tilde{x}|x^{(t)})$. In practice, stochastic gradient descent is employed to optimize the objective function. A novel architecture was developed in [156] based on stacked layers of DAEs. With the stacked model, the implementation of the DAE becomes easier since we only need to determine the type and level of the corrupting noise.

Recently, it has been observed that the performance of classification tasks will be improved when sparsity is encouraged to learn the representations. Sparse representations are used to produce a simple interpretation of the input data by extracting the hidden structure of the data. The learning algorithm for sparse representation was firstly proposed by Ranzato in 2006 [128]. To tune a code vector into a quasi-binary sparse one, a non-linear sparsity is added between a linear encoder and a linear decoder. We note that for binary inputs, large weights are required to minimize the reconstruction error. The overall cost function in a sparse AE is shown in Equation 3:

$$J_{sparse}(\omega, b) = J(\omega, b) + \beta \sum_{j=1}^{N} KL(\rho \| \rho'_j) \tag{3}$$

where ρ is a sparsity parameter, typically a small quantity close to zero, N is the number of neurons in the hidden layer, ρ'_j is the average activation of hidden unit j, and $J_{sparse}(\omega, b)$ is the previous cost function. β controls the weight of the sparsity penalty term.

Furthermore, Makhzani and Frey [110] proposed a k-sparse AE in 2013. The k-sparse AE consists of the basic architecture of a standard AE while keeping only the highest k activations in the hidden layers. The results obtained show that the k-sparse AEs perform better than the DAEs and RBMs. They claimed that the k-sparse AEs can be easily trained, and the advanced encoding process will contribute to achieving satisfactory performance for large-scale problems.

In 2011, Rifai et al. [133] proposed the contractive autoencoders (CAEs) where a well selected penalty term is added to the standard cost function in the reconstruction stage. This penalty term is employed to penalize the sensitivity of the features with respect to the inputs. In this way, the mapping from the input vector to the representation will converge with higher probability. Results obtained by CAEs are identical to or even better than those obtained by other regularized AEs such as DAEs. The training objective of the CAEs is shown in Figure 4:

$$J_{CAE} = \sum_{t} L(x^{(t)}, g_\theta(f_\theta(x^{(t)}))) + \lambda \|J(x^{(t)})\|_F^2, \tag{4}$$

where $L(\cdot)$ is the cost function, λ is the parameter to control the regularization strength, and $J(x)$ is the function which represents the Jacobian matrix of the encoder. Rifai et al. discovered that the penalty term would produce robust features on the activation layer. Moreover, the penalty can be used to address the trade-off between the

robustness and reconstruction accuracy. It is also shown that a DAE with slight corrupting noises can be regarded as a CAE in which the whole reconstruction function is penalized [11]. Furthermore, in 2016, Sun et al. [146] proposed a separable deep autoencoder (SDAE) which is used to deal with the unseen noise estimation. The total reconstruction error of the noisy speech spectrum can be minimized by adjusting the unknown parameters of the DAE and the estimation of the clean speech spectrum [19].

V. Deep Learning Architectures: Deep Convolutional Neural Networks

A. The motivation

CNNs are a subtype of the discriminative deep architecture [3] and have shown satisfactory performance in processing two-dimensional data with grid-like topology, such as images and videos. The architecture of CNNs is inspired by the animal visual cortex organization. In the 1960s, Hubel and Wisel [73] proposed a concept called receptive fields. They found that the complex arrangements of cells were contained in the animal visual cortex in charge of light detection in overlapping and small sub-regions of the visual field. Furthermore, the computational model Neocognitron was introduced in [46] with hierarchically organized image transformations. However, the Neocognitron differs from the CNNs in that it does not require a shared weight.

The concept of CNNs is inspired by time-delay neural networks (TDNN). In a TDNN, the weights are shared in a temporal dimension, which leads to reduction in computation. In CNNs, the convolution has replaced the general matrix multiplication in standard NNs. In this way, the number of weights is decreased, thereby reducing the complexity of the network. Furthermore, the images, as raw inputs, can be directly imported to the network, thus avoiding the feature extraction procedure in the standard learning algorithms. It should be noted that CNNs are the first truly successful deep learning architecture due to the successful training of the hierarchical layers. The CNN topology leverages spatial relationships so as to reduce the number of parameters in the network, and the performance is therefore improved using the standard backpropagation algorithms. Another advantage of the CNN model is that it requires minimal pre-processing.

With rapid development of computation techniques, the GPU-accelerated computing techniques have been exploited to train CNNs more efficiently. Nowadays, CNNs have already been successfully applied to handwriting recognition, face detection, behavior recognition, speech recognition, recommender systems, image classification, and NLP.

B. The structure and the algorithm

Three factors play a key role in the learning process of a CNN: sparse interaction, parameter sharing and equivariant representation [74]. Different from the traditional NNs where the relationship between the input and output units are derived by matrix multiplication, the CNNs reduce the computational burden with sparse interaction where the kernels are made smaller than the inputs and used for the whole image. The basic idea of parameter sharing is that, instead of learning a separate set of parameters at each location, we only need to learn one set of them, which implies a better performance of the CNN. Parameter sharing has also endowed the CNN with an attractive property called equivariance, meaning that whenever the input changes, the output changes in the same way. Consequently, fewer parameters are required for CNN as compared to other traditional NN algorithms, which leads to reduction in memory and improvement in efficiency. The components of a standard CNN layer are shown in Figure 4, and a conceptual schematic diagram of a standard CNN is shown in Figure 5.

As shown in Figure 5, a CNN is a multi-layer neural network that consists of two different types of layers, i.e., convolution layers (c-layers) and sub-sampling layers (s-layers) [30], [74], [86]. C-layers and s-layers are connected alternately and form the middle part of the network. As Figure 4 shows, the input image is convolved with trainable

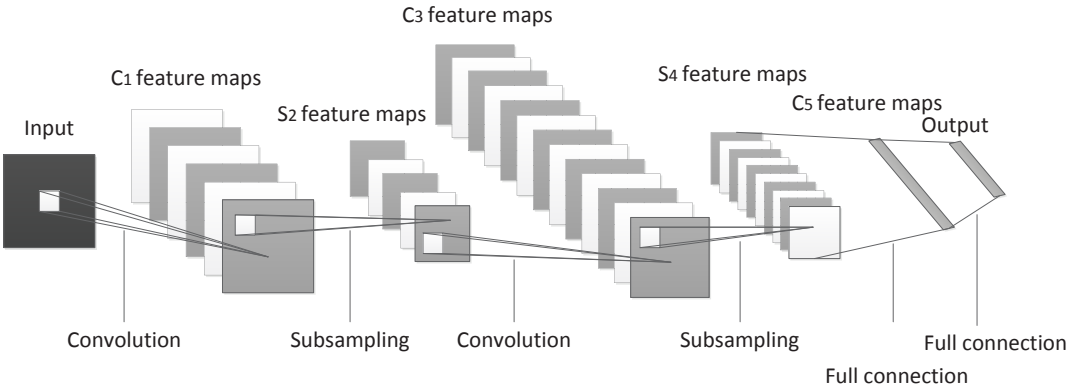


Fig. 4. Schematic structure of CNNs

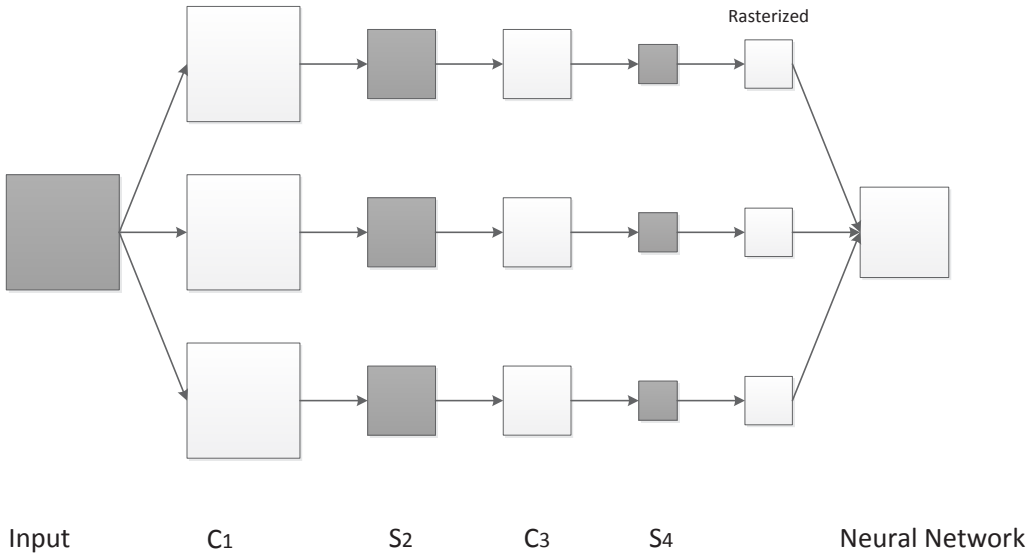


Fig. 5. Conceptual structure of CNNs

filters at all possible offsets in order to produce feature maps in the first c-layer. A layer of connection weights are included in each filter. Normally, four pixels in the feature map form a group. Passed through a sigmoid function, these pixels produce additional feature maps in the first s-layer. This procedure carries on and we can thus obtain the feature maps in the following c-layers and s-layers. Finally, the values of these pixels are rasterized and displayed in a single vector as the input of the network [3].

Generally, c-layers are used to extract features when the input of each neuron is linked to the local receptive field of the previous layer. Once all the local features are extracted, the position relationship between them can be figured out. An s-layer is essentially a layer for feature mapping. These feature mapping layers share the weights and form a plane. Additionally, to achieve scale invariance, the sigmoid function is selected as the activation function due to its slight influence on the function kernel. It should also be noted that, the filters in this model are used to connect a series of overlapping receptive fields and transform the 2-D image batch input into a single unit in the output.

However, when the dimensionality of the inputs equals to that of the filter output, it will be difficult to maintain

translation invariance with additional filters. Due to the high dimensionality, the application of a classifier may cause overfitting. To solve this problem, a pooling process, also called sub-sampling or down-sampling, is introduced to reduce the overall size of the signal. In fact, sub-sampling has already been successfully applied for data size reduction in audio compression. In the 2-D filter, sub-sampling has also been used to increase the position invariance.

The training procedure for a CNN is similar to that for a standard NN using backpropagation. More specifically, Lecun et al. [10] introduced error gradient to train the CNNs. In the first stage, information is propagated in the feed-forward direction through different layers. Salient features are obtained by applying digital filters at each layer. The values of the output are then computed. During the second stage, the error between the expected and actual values of the output is calculated. Backpropagating and minimizing this error, the weight matrix is further adjusted and network is thus fine-tuned. Unlike other standard algorithms in image classification, the pre-processing is not frequently performed in CNNs. Instead of setting parameters, as is the case with traditional NNs, we just need to train the filters in CNNs. Moreover, in feature extraction, CNNs are independent of prior knowledge and human interference.

In 1998, the max pooling method was proposed in LeNets for sub-sampling [92]. By summarizing the statistics of the nearby outputs, a pooling function is used to replace the output of the network at a certain position. Using the max-pooling method, we can obtain the maximum output in a rectangular neighborhood. The pooling procedure can also make the representation invariant to the translations of the input. Now, by adding a max pooling layer between the convolutional layers, spatial abstractness increases with the increase of feature abstractness.

As mentioned in [17], pooling is used to obtain invariance in image transformations. This process will lead to better robustness against noise. It is pointed out that the performance of various pooling methods depends on several factors, such as the resolution at which low-level features are extracted and the links between sample cardinalities. In 2011, Boureau [16] found that even if features are widely dissimilar, it is possible to pool them together as long as their locations are close. Furthermore, it is found that better performance can be delivered by performing clustering ahead of the pooling stage. In [78], it is shown that better pooling performance can be achieved by learning receptive fields more adaptively. Specifically, utilizing the concept of over-completeness, an efficient learning algorithm is proposed to accelerate the training process based on incremental feature selection.

More recently, Sermanet et al. [138] proposed a novel pooling method called L_p pooling and obtained high accuracy on the SVHN dataset. L_p pooling is a biological model inspired by complex cells. In 2013, Zeiler and Fergus [171] proposed a stochastic pooling method to regularize large CNNs which is equivalent to introduce a stochastic pooling procedure in each convolutional layer. According to a multinomial distribution, the activation is randomly selected in each pooling region. Moreover, since the selections in higher layers are independent of those in the lower ones, stochastic pooling is used to compute the deformations in a multi-layer model.

C. Variations of CNNs

CNN has become a popular research topic in the past few years. In 2013, Eigen et al. [40] introduced a novel model called recursive convolutional networks (RCNs). The architecture of RCNs can be viewed as a CNN with identical number of feature maps in all layers and tied filter weights across layers. It is shown that a larger number of layers imply an increased computational burden, which makes little sense to precisely specify the size of the feature maps dimensions.

CNN has also been used for feature extraction in areas like object recognition. In 2009, Jarrett et al. [77] proposed a novel model which combines convolution with an AE. Based on the AE architecture, predictive sparse decomposition unsupervised feature learning is employed with sparsity constraints on the feature vector. The feature extraction stage involves a filter bank, a non-linear transformation, and a feature pooling layer. More recently,

Masci et al. [112] developed an advanced stacked convolutional AE for unsupervised feature learning. During the training process, conventional gradient descent algorithm is used by each convolutional AE without adding additional regularization terms. It is proved that the stacked convolutional AE can achieve satisfactory CNN initializations by avoiding the local minima of highly non-convex objective functions.

Great success has been achieved when CNNs are applied to the research of computer vision. In 2008, Desjardins and Bengio [36] proposed a novel model to employ RBMs in a CNN, which constitutes the convolutional restricted Boltzmann machines (CRBMs). In the CRBMs, a convolution is computed with a normal RBM as the kernel. Although the number of parameters in RBMs depends on the dimension of the input image, the complexity of CRBMs only depends on the number of features to be extracted and the size of the receptive field. The CD algorithm can also be applied to train CRBMs. The visible layer is initialized with the image input. An upward pass is performed to compute the pixel states in the hidden layer. Compared with standard RBMs in vision applications, CRBMs can achieve a higher convergence rate with a smaller value of the negative likelihood function. Besides, the convolutional deep belief networks (CDBNs) have also been developed [87] and applied to scalable unsupervised learning for hierarchical representations, and unsupervised feature learning for audio classification [94], [95].

Recently, fast Fourier Transform (FFT) has been employed in original CNNs. In 2014, Mathieu and Henaff [113] introduced a fast training procedure for CNNs using FFT. Since large amounts of data are required for CNNs to learn complex functions, even with the modern GPUs, it will take long time, sometimes several weeks, to train the CNNs to produce promising results. When dealing with web-scale datasets, the cost of producing labels with a trained network is high. Towards this problem, a simple algorithm is developed in [113] to accelerate the training process with a significance factor. The method is realized by computing convolutions as products in the Fourier domain. The same transformed feature map is used many times. The challenge of training CNNs lies in the convolution of pairs of 2-D matrices. With the Fourier transformation, convolution of the matrices is converted into pairwise products, which can be carried out efficiently. Based on the computation requirement, a GPU processor can be used to implement the algorithm.

Sainath et al. [135] proposed an advanced CNN algorithm for speech recognition by introducing an extra filter bank layer to replace the mel-filter bank. The filter bank is learned jointly with other network parameters, through which the cross-entropy objective function is optimized. Moreover, a novel method is developed to normalize the filter-bank features while maintaining their positivity so that the logarithm non-linearity can be applied. Similar to the standard CNNs, the initial weights of the filter bank layer are not randomly selected but identical to those of the mel-filter bank.

VI. Applications of Deep Learning

In this section, we will review some practical applications of the deep learning architectures. In fact, due to its ability to handle large amounts of unlabeled data, deep learning techniques have provided powerful tools to deal with big data analysis [31], [122]. In recent years, massive amounts of data have been collected in various fields including cyber security, medical informatics [173], and social media. Deep learning algorithms are used to extract high-level features from these data in order to obtain hierarchical representations. Recently, deep learning has attracted the attention of many high-tech enterprises such as Google, Facebook and Microsoft.

The architecture of deep networks has been widely applied in speech recognition and acoustic modeling for audio classification [95]. Besides, deep learning approaches also play an important role in the area of image processing such as handwritten classification [84], high-resolution remote sensing scene classification [66], single image super-resolution (SR) [38], multi-category rapid serial visual presentation Brain Computer Interfaces (BCI) [111], redand domain adaptation for large-scale sentiment classification [47]. Moreover, deep architectures have also been employed in multi-task learning for NLP with an enhanced inference robustness [26], [88]. In the following,

we will make a general review on several selected applications of the deep networks: speech recognition, computer vision, and pattern recognition.

A. Speech Recognition

During the past few decades, machine learning algorithms have been widely used in areas such as automatic speech recognition (ASR) and acoustic modeling [76], [116], [118], [126]. The ASR can be regarded as a standard classification problem which identifies word sequences from feature sequences or speech waveforms. In some well-defined applications such as transcription and dictation, commercial speech recognizers have been widely used. Many issues have to be considered for the ASR to achieve satisfactory performance, for instance, noisy environment, multi-model recognition, and multilingual recognition. Normally, the data should be pre-processed using noise removal techniques before the speech recognition algorithms are applied. Singh et al. [141] reviewed some general approaches for noise removal and speech enhancement such as spectral subtraction, Wiener filtering, windowing, and spectral amplitude estimation. Traditional machine learning algorithms, such as the SVM, and NNs, have provided promising results in speech recognition [58]. For example, Gaussian mixture models (GMMs) have been used to develop speech recognition systems by representing the relationship between the acoustic input and the hidden states of the hidden Markov model (HMM) [7].

1) Standard Speech Recognition Architecture and Algorithms:

The standard architecture of an ASR system is given in Figure 6.

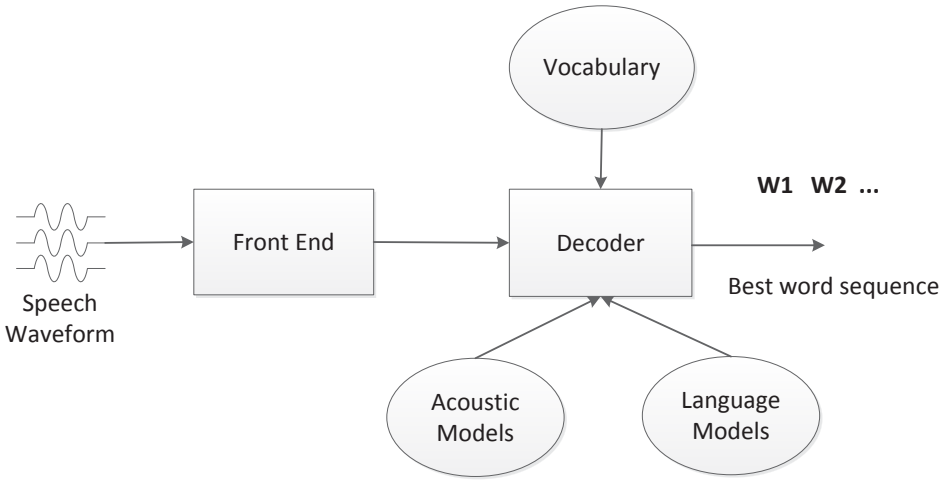


Fig. 6. Speech recognition system architecture

First, the speech waveform passes through the auditory front end where the signal is pre-processed and spectral-like features are produced. Then the features will be passed to a phone likelihood estimator in order to estimate the likelihood of each phone. After that, the decoder will decode the speech with phone likelihoods using n-gram language model (LM) and the HMM. Finally, the output will be sent to the parser, transformed to the best word sequence and converted to a readable format.

As mentioned previously, traditional machine learning schemes have achieved satisfactory results for ASR. Among them, HMMs and GMMs are widely used for acoustic modeling to generate low-level acoustic contents from high-level acoustic inputs [32], [34], [79], [101], [116], [120], [130]. Here we will make a brief introduction about these two models. Since a speech signal can be regarded as a short-time or a piecewise stationary signal, we can, for a

short time period, assume that the speech process is stationary. A Markov model can therefore be to describe the stochastic speech process. Additionally, the training process of HMMs is automatic and simple to implement. We can use a sequence of hidden states of the HMM to represent acoustic features with non-stationary distributions. The HMM will generate a sequence of vectors representing the likelihood of each state.

It should be noted that the performance of HMMs can be greatly affected by the mismatch between the training and testing conditions. In such a case, large amounts of data are required. In [126], GMMs were used to estimate the output density of the HMM states. Furthermore, GMMs play an important role in speech-generation tasks and are frequently used in frame-by-frame mapping, especially for speech enhancement, articulatory-to-acoustic mapping and voice conversion. The GMM-HMM systems have significantly improved the accuracy of classification and can also be applied for noise removal in noisy speech utterances. Admittedly, the GMM-HMM still has some limitations. It is difficult for the GMM-HMM to represent non-linear or more complex relationships between the acoustic features and the speech inputs. The modeling efficiency is usually very low for data near a non-linear manifold. Besides, the assumption of conditional independence is another well-known drawback of the GMMs. Furthermore, the loss of raw information can also degrade the performance of the GMM-HMM systems.

It is widely recognized that NNs lying on or near a non-linear manifold can deliver better performance than the GMM-HMM systems. Elegant results have been achieved two decades ago when researchers adopted ANNs with one layer of non-linear hidden units to predict the HMM states from windows of acoustic coefficients [58]. However, due to the limited computation resources, it was difficult to implement standard NNs with many hidden layers at that time. During the past few years, the computation techniques have developed rapidly, which leads to more efficient ways to train the DNNs. Since 2006, deep learning has emerged as a new research area of machine learning. As we just mentioned, deep learning algorithms can bring satisfactory results in feature extraction and transformation, and they have been successfully applied to pattern recognition. Compositional models are generated using DNNs models where features are obtained from lower layers. Through some well-known datasets including the large vocabulary datasets, researchers have shown that DNNs could achieve better performance than GMMs on acoustic modeling for speech recognition. Due to their outstanding performance in modeling data correlation, the deep learning architectures are now replacing the GMMs in speech recognition [33], [58].

Early applications of the deep learning techniques consist of large vocabulary continuous speech recognition (LVCSR) [28] and phone recognition [117]–[119]. In these applications, DBNs are used to train the unlabeled data discriminatively. Moreover, the DBN-HMM method, which combines HMMs with the deep learning models, has achieved a great success. The observation probability is estimated using DBNs, while the HMM is used to model the sequential information. As pointed out in [117], the advanced DBN-HMM method has adopted the conditional random fields (CRFs) to replace the HMM in modeling the sequential information. The maximum mutual information (MMI) is employed to train the DBN-CRF. In this case, the transition weights, the weights of the DBN, and the phone language model are jointly optimized using the sequential discriminative learning technique. Compared with the DBN-HMM with frame-discriminative training process, the DBN-CRF can achieve higher accuracy. In [176], a combination of the heterogeneous DNNs and the CRF has been proposed for Chinese dialogue act recognition.

Next we will review the recent progress in speech recognition during the past few years. In 2015, the DNNs have been employed in automatic language identification (LID). Experiments have been carried out on short test utterance [49] from two datasets: the Google 5 million utterances LID and the public NIST Language Recognition Evaluation dataset. More recently, a multi-task learning (MTL) method is proposed to improve the low-resource ASR using DNNs with no requirement on additional language resources [20]. Many other achievements have also been obtained in ASR, especially in distant talking situations [132], [161], [164], audio-visual speech recognition

(AVSR) systems [125], and data augmentation on the basis of label-preserving transformations [27]. Great progress has also been made using RBMs for enhanced sound wave representation [76]. Moreover, the DNNs have been employed for tracking dialog state [53], transferring cross-language knowledge [71], learning the filter banks [135], designing an automatic feature extraction systems from audio [51], and speaker adaptive training (SAT) for acoustic modeling [114].

2) Large Vocabulary Continuous Speech Recognition:

In 2010, the context-dependent DBN-HMM approach has been proposed for LVSCR to replace the context-independent one [28]. Experiments on Bing mobile voice search data showed that the context-dependent DBN-HMM achieved enhanced performance compared with the standard HMM approach. Five pre-trained layers with 2048 hidden units at each layer are trained to classify the central frame. The performance of the DBN-HMM system can be greatly improved by using triphone senones as the DNN training labels and tuning the transition probabilities properly.

It should be noted that the CNNs can be regarded as an alternative model for speech recognition [134], [139]. Compared with DNNs, CNNs have attracted researchers' attention for the following reasons: on one hand, the input of DNNs can be interpreted in any order without any influence on the network performance, whereas speech spectral representations are strongly correlated in frequency and time. CNNs with shared weights have distinct advantages in modeling such local correlations [93]; on the other hand, due to the influence of different speaking styles, the formants will be shifted and DNNs may lead to poor results for translational variance models [90]. Moreover, a large number of parameters and large networks are required for DNNs to capture translational invariance. However, by averaging the outputs of the hidden units in different frequencies, the CNNs can capture the translational invariance with fewer parameters. Experiment results in [134] showed that CNNs can achieve relatively better performance than DNNs for LVCSR tasks.

In 2015, Li et al. [99] compared different acoustic modeling approaches using DNNs and evaluated the performance of Chinese dialog model on the basis of LVSCR. It is difficult for traditional ASR systems to handle Chinese because it is a syllabic language with 1254 distinct syllables and 408 toneless base-syllables. Additionally, the syllable based architecture may lead to poor coverage and non-uniform distribution of the training data. Furthermore, large size of modeling units are required for data training, which excludes the possibility of using GMM based acoustic models. In [99], a multi-task learning strategy was introduced to combine different models in the DNN based speech recognition systems and achieved better performance than the GMMs based model. Moreover, by using DNNs, Aryal et al. [6] developed a novel method for real-time data-driven articulatory synthesis. A tapped-delay input line is adopted to capture context information in the articulatory trajectory, which means there is no need to post-processing the data. Additionally, deep learning techniques can also be used for head motion synthesis [37] and speech enhancement [81].

It is recognized that both audio information and visual component are key factors for human speech recognition. Synthetic talking avatar has been introduced in many human-computer interaction applications such as virtual newscaster, computer agent, email reader, and information kiosk. In 2015, Wu at el. [160] developed a real-time speech driven talking avatar system using DNNs. With the acoustic speech as its input, the three-dimensional avatar system can react with articulatory movements accordingly. The most important factor in this system is the acoustic-to-articulatory mapping. This mapping procedure is not trivial due to the non-linear relationship between the acoustic and articulatory features. The challenge is to compute the articulator movements according to both the current and the preceding phonemes. Four models that have been widely used are GMMs, general linear model (GLM), ANNs, and DNNs.

The simplest approach to determine the relationship between the acoustic input and the articulatory output is the linear mapping method such as the GLM. However, as mentioned before, the acoustic-to-articulatory mapping is

non-linear, which indicates that GLM cannot achieve ideal performance. In the training process, the HMM method requires phonetic information as constraints to tackle the mapping problem. The relationship between the acoustic and articulatory features is regarded as a linear mapping in each state of the HMM. In this case, the GMM is employed to model the joint distribution of the articulatory and acoustic features to address the unconstrained mapping problem. As shown in [126], the ANNs can be used to build the real-time speech driven talking avatar system due to their relatively short computation time. Compared with other models, the ANNs have delivered superior performance. However, it usually takes long time to train an ANN with multiple hidden layers and the training process tends to get trapped in poor local optima. Besides, the performance can be significantly affected by how the ANNs are initialized. Motivated by these facts, the DNNs are adopted for an effective treatment of a large quantity of unlabeled data. With the DNNs, a more sensible initialization is made and a more efficient pre-training is performed, which has also to some extent relieved the overfitting problem.

B. Computer Vision and Pattern Recognition

Computer vision aims to make computers accurately understand and efficiently process visual data like videos and images [8], [144]. The machine is required to perceive real-world high-dimensional data and produce symbolic or numerical information accordingly. The ultimate goal of computer vision is to endow computers with the perceptual capability of human. Conceptually, computer vision refers to the scientific discipline which investigates how to extract information from images in artificial systems. The following areas are included as sub-domains of computer vision: event detection, scene reconstruction, object detection and recognition, object posture estimation, image restoration, statistical learning, image editing and video enhancement.

Pattern recognition is a scientific discipline which aims to identify the pattern of a given input value [14]. It is a rather general concept which encompasses several sub-domains like classification, regression, sequence labeling, and speech tagging. Due to the rapid industrial development, there are ever increasing requirements on the capability of information retrieval and processing, which has brought new challenges to pattern recognition. Recently, the development in deep learning architectures has provided novel approaches to the problem of pattern recognition, which will be discussed in what follows.

1) Recognition:

During the past few years, deep learning techniques have achieved tremendous progress in the domains of computer vision and pattern recognition, especially in areas such as object recognition. We will discuss some classical problems in computer vision regarding recognition tasks. In classification applications, feature selection is an important issue. Normally, features are specified manually in traditional classification algorithms, which have limited generality. Some typical deep learning architecture, such as the CNNs, can select the features automatically and achieve outstanding performance based on GPU-accelerated computational resources. Note that human vision systems are different from computer vision systems, and it has been shown that DNNs can be easily fooled by unrecognizable images [124]. However, this does not mean that deep learning techniques are not suitable for classification tasks. Recent researches have shown that in classification tasks, deep learning techniques can obtain promising results [9], [89].

In object recognition, which is also called object classification, deep learning methods have achieved superior performance compared with conventional classification algorithms [151]. Here we review some recent progress on classification tasks. For German traffic sign recognition, the multi-column DNN has been proposed [24], [25]. To study neuropsychiatric conditions based on functional connectivity (FC) patterns, standard classifiers like the SVM have been widely used. Recently, the DNNs have been employed to classify the whole-brain resting-state FC patterns of schizophrenia (SZ) [83]. To improve the performance of classification, a novel maximum margin multimodal

deep neural network (3mDNN) was proposed to take advantage of the multiple local descriptors of an image [131]. Compared with standard algorithms, this method, considering the information of multiple descriptors, can achieve discriminative ability. DNNs can also be used for the wind speed patterns classification and the supervised multispectral land-use classification [70], [105].

In computer vision and pattern recognition, sometimes we need to build and process 3D models. Mesh understanding is one of the key factors in this field. Particularly, mesh labeling can be used to find out the inherent characteristics of the mesh. For image labeling tasks, Lerouge et al. [96] proposed an input output deep architecture (IODA) in 2015. Previously, the mesh labeling approaches focused on the mesh triangle which was characterized by heuristically designed geometry features [72], [80]. Although these standard methods could obtain satisfactory results, they suffered from a serious problem that the geometric features obtained could provide promising results only for few 3D mesh types. Therefore, it is of great importance to develop new approaches for feature generation and labeling using different types of meshes. Towards this target, a more effective representation of meshes was introduced in [50]. Combining human vision knowledge and deep learning models, CNNS are employed to learn mesh representations with much better performance. Moreover, since promising abilities were shown in learning multi-layered non-linear features, DBNs have been widely used in object classification tasks.

It is well recognized that when larger datasets are used for training, the problem of overfitting can be prevented effectively, which implies improved performance. Therefore, as a dataset that includes over 15 million labeled images, the ImageNet has attracted great attention [29]. As mentioned in previous sections, the performance of the CNNs can be controlled by adjusting the depth and breadth, and the weights of the CNNs are shared, which imply a shorter tuning process. Therefore, CNNs are employed in ImageNet and have achieved satisfactory performance [86]. In addition, the CNNs have also been used for high-resolution remote sensing (HRRS) scene classification [66] and handwritten Hangul recognition [84].

It should be noted that deep learning techniques can also be applied to hand posture recognition (HPR). Since the features generated by traditional algorithms are limited, and it is difficult to detect and track hands with normal cameras, the DNNs are employed to produce enhanced features [149]. Based on functional near infrared spectroscopy (FNIRS), deep learning techniques have achieved promising results in classifying brain activation patterns for BCI [54].

2) Detection:

Detection is one of the most widely known sub-domains in computer vision. It seeks to precisely locate and classify the target objects in an image. In the detection tasks, the image is scanned to find out certain special issues. For example, we can use image detection to find out the possible abnormal tissues or cells in medical images. The deformable part-based model (DPM) proposed by Felzenszwalb is one of the most popular methods [43]. As demonstrated in [148], due to their strong abilities to capture the geometric information such as object locations, DNNs have been widely used for detection and have shown outstanding performance.

As mentioned in [1], DBNs are employed in the computer-aided diagnosis (CAD) systems for early detection of breast cancer. In this case, the accuracy of the classifier is the most important factor for the CAD system. Compared with standard classification algorithms such as the C4.5 decision tree method, the supervised fuzzy clustering (SFC) technique, the Fuzzy-GA approach, the radial bases function neural network (RBFNN) method, and the particle swarm optimized wavelet neural network (PSOWNN), the DBNs can achieve better performance for the CAD system. The DBNs can also be used to reduce the non-linear dimensionality of the input features. Studies on the brain tumor detection and the segmentation task have received increasing attention during the past few years [127]. Due to its computational efficiency, the magnetic resonance imaging (MRI) method for clinical brain tumor detection was introduced using deep learning techniques. However, the MRI based brain tumor detection method suffers from the discrepancy between the predicted size and shape of the brain tumors. The CNNs are employed to

solve this problem for its strong learning capability. It should be noted that promising results have been achieved by CNNs in areas like human detection and Doppler radar based activity classification [85].

Similarly, deep learning methods can also be applied to annotate genetic variants to identify pathogenic variants. Normally, the combined annotation-dependent depletion (CADD) algorithm is most widely used to annotate the coding and non-coding variants [129]. In the CADD method, a linear kernel SVM is trained as the classifier. However, because of limitations of SVM, CADD method cannot capture the non-linear relationships among the features. Therefore, the DANNs are used instead of the SVM classifier. The DANNs are suitable for large amounts of samples and features. More specifically, to predict the protein order/disorder regions, the deep convolutional neural fields (DCNF) method was introduced in [157]. In recent years, the saliency detection models have attracted increasing research attention in predicting human eye-attended locations in the visual field. Traditional approaches are based on contrast inference mechanisms and hand-designed features. The deep learning techniques only require raw image data [52]. Besides, deep learning techniques have also been applied to Glaucoma detection [23], and human-robot interaction systems with promising results [82].

As another significant application of computer vision, image change detection plays an important role in not only civil but also military fields. The target of image change detection is to sort out the differences between two images taken at different time for the same scene. The image detection has been widely employed in remote sensing, medical diagnosis, disaster evaluation, and video surveillance. In particular, the synthetic aperture radar (SAR) image processing is a widely used application in change detection [48]. In the state-of-the-art methods, a difference image (DI) is produce between multi temporal SAR images for change detection. However, the DI may have an adverse influence on the change detection performance. To avoid this, the deep learning techniques have ignored the process of generating a DI. For traffic control and maritime security monitoring, ship detection on spaceborne images has been widely used. Due to their visualized contents and high resolution properties, spaceborne images are superior to other remote sensing images in object detection. However, compared with the infrared and synthetic aperture radar images, the spaceborne images are easily affected by the weather condition. In addition, the difficulty of image processing increases as larger database is treated for higher resolution. To overcome these two shortcomings, the DNNs are combined with the extreme learning machines (ELMs) in [150]. Compared with other state-of-the-art methods, this approach has achieved higher accuracy with less detection time.

3) Other Applications:

Face alignment plays an important role in various visual applications such as face recognition. However, for the extreme situations where the face images are taken, face alignment may lead to difficulties during the analyzing process. Therefore, different models for shape and appearance variation have been considered to solve this problem. Based on the model used, the standard approaches can be roughly divided into three groups: the active appearance model, the constrained local model and the regression. Compared with the normal regression based methods, the adaptive cascade deep convolutional neural networks (ACDCNN), proposed by Dong for facial point detection, have dramatically reduced the system complexity [39]. Dong et al. improved the basic DCNNs by exploiting an adaptive manner for training with different network architectures. Experiment results showed that their networks can achieve better performance than the DCNNs or other start-of-the-art methods.

It should be noted that the multi label image annotation is a hot topic in the field of computer vision [177]. Furthermore, deep learning techniques have recently been applied to the content-based image retrieval applications [42]. More specifically, to address the cross-modal retrieval tasks, a correspondence AE (Corr-AE) was introduced by correlating the hidden representations of two uni-modal AEs [44]. Compared with bimodal AEs and bimodal DBNs [123], [145], the Corr-AEs focus more on the correlation across data than the complementarity learned from different modalities. The correlation learning and representation learning are carried out at the same time so that the computation efficiency is improved. Additionally, on the basis of multimodal fusion and backpropagation deep

learning models, ideal results have been obtained for video-based human pose recovery [62].

Pose estimation is another important sub-domain in computer vision. The general target of pose estimation is to estimate the relative position of a specific object with respect to the camera. This technique plays an irreplaceable role in various tasks such as building a robot arm. Taking various moth poses and cluttered background into consideration, researchers found that the on-trap moths automated identification using traditional approaches was affected by misidentification and incomplete feature extraction. Therefore, a deep learning architecture was introduced for on-trap field moth sample images [158]. Particularly, Li et al. [98] proposed a heterogeneous multi-task learning framework where a DCNN was adopted for monocular image based human pose estimation.

In addition, deep learning approaches have been successfully employed in motion estimation especially for video tracking tasks [22]. Object tracking has attracted much research attention due to its theoretical value and application prospects in areas such as self-driving vehicles, robotics, and intelligence video surveillance. To design a robust object appearance model, classifier construction and feature representation are two major issues. As mentioned previously, typical classifiers for designing the robust object appearance model include the SVM, sparse coding, random forest, boosting, and so on. Conventional classifiers have achieved certain degree of success, their structure, however, has limited performance especially for highly non-linear or time-varying object appearance variations. Similarly, the traditional feature representation consists of various well-known features such as SFIT, HoG, covariance matrix, subspace-based features, and color histograms etc. These handcrafted and pre-defined features have achieved great success for low-level features. Nevertheless, most handcrafted features cannot reflect time-varying properties. Thus, the CNN tracker was introduced to solve the limitations of shallow classifier structures and handcrafted features in object tracking tasks. One restriction of traditional deep learning architectures comes from the usage of a single observation model. In this case, the trackers have to cope with contaminated features due to occlusion. Hence, Wu et al. [159] introduced a regional deep learning tracker containing multiple deep models, and each of which is in charge of tracking one sub-region.

In computer vision, denoising is an important issue because the digital images are corrupted by noise through acquisition and transmission. Although there are many denoising algorithms, most of them are designed for special cases and lack generality. The Wiener filter performs well for removing Gaussian noises [18]. However, it requires the knowledge on the autocorrelation functions of the input. With respect to suppressing noisy images with edges, median filtering deals with salt and pepper noises effectively [4]. Therefore, stacked sparse denoising autoencoders (SSDAEs) have been proposed with promising noise removal performance. Furthermore, the adaptive multi-column SSDAEs (AMC-SSDAEs) have been introduced to improve the robustness of the filter [2].

VII. Conclusion

In this paper, we have reviewed the latest developments of deep neural networks. Some widely-used deep learning architectures are investigated and selected applications to computer vision, pattern recognition and speech recognition are highlighted. More specifically, four classes of deep learning architectures, namely the restricted Boltzmann machine, the deep belief networks, the autoencoder, and the convolutional neural network, are discussed in detail. Since it is rarely possible to obtain labeled data in applications involving big data analysis, the supervised learning algorithms can hardly provide satisfactory performance in such cases. Based on these deep learning approaches, we can now use unsupervised learning algorithms to process the unlabeled data. Moreover, the trade-off between accuracy and computational complexity can be adjusted with flexibility in most deep learning algorithms. With the rapid development of hardware resources and computation technologies, we are confident that deep neural networks will receive wider attention and find broader applications in the future.

Based on the literature review, some related topics for future research are listed as follows.

- **Design of deep models to learn from fewer training data**: With the development of big data analysis, deep learning have been used for scenarios where massive amounts of unsupervised data are involved. As an efficient tool for big data analysis, the deep learning technique have achieved great success with huge amounts of unlabeled training data. However, when only a limited amount of training data is available, more powerful models are required to achieve an enhanced learning ability. It is therefore of great significance to consider how to design deep models to learn from fewer training data, especially for speech and visual recognition systems.
- **Ues of optimization algorithms to adjust the network parameters**: The method to adjust the parameters in machine learning algorithms is an emerging topic in computer science. In DNNs, a large number of parameters need to be adjusted. Moreover, with an increasing number of hidden nodes, the algorithm is more likely get trapped in the local optimum. Optimization techniques, such as the PSO [172], are therefore required to avoid this problem. The proposed training algorithm should be able to extract the features automatically and reduce the loss of information so as to mitigate both the curse of dimensionality and the local optimum.
- **Applications of unsupervised, semi-supervised and reinforcement-learning approaches to DNNs for complex systems**: As mentioned previously, deep learning techniques have not brought satisfactory results in NLP. With the development of deep unsupervised learning and deep reinforcement learning, we have more alternatives to train the DNNs for complex systems. The Alpha Go, which combines CNNs and reinforcement learning, has already achieved a great success. Compared with the supervised learning approaches, the unsupervised, semi-supervised and reinforcement-learning approaches, capable of overcoming the computational limitations, deserve further investigation.
- **Implementation of deep learning algorithms on mobile devices**: It should be noted that deep learning approaches, especially CNNs, usually require great computational burden. Recently, the idea of deep learning chips has emerged and attracted great research attention. A chip for neural networks implementation has already been presented by MIT researchers. This chip is 10 times as efficient as a mobile GPU, which means that we can run AI algorithms in mobile devices with lower power consumption. Additionally, Stanford has started the project aiming at optimizing the CPU for deep learning. This area can bring numerous benefits for both industries and academia.
- **Analysis of the stability of deep neural networks**: Dynamic neural networks have been widely used to solve optimization problems and applied to many engineering applications. Nowadays, the stability analysis of deep neural networks has become a hot research topic because of the numerous benefits for industries. It should be pointed out that, so far, there have been a multitude of research results on the stability analysis, stabilization and synchronization problems for various types of systems and networks in the literature, see [64], [67], [68], [102], [143], [163], [168], [174] for some recent publications. By utilizing these exploited techniques, we can further deal with the corresponding issues including stability analysis, synchronization and state estimation for deep neural networks.
- **Applications of deep neural networks in nonlinear networked control systems (NCSs)**: Neural networks have been extensively used in control engineering and signal processing to approximate the nonlinear systems. On the other hand, up to now, the NCSs have been widely investigated and considerable results have been reported in the literature, see [21], [65], [69], [103], [104], [106]–[109], [142], [169], [170], among which the networked control systems under consideration are either linear or nonlinear with relative simple forms. Thus, it is natural to apply the deep neural networks to approximate the nonlinear NCSs with complicated dynamics to obtain better control/filtering performances.

REFERENCES

[1] A. M. Abdel Zaher, and A. M. Eldeib, "Breast cancer classification using deep belief networks," *Expert Systems with Applications*, vol. 46, pp. 139–144, 2016.

[2] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," *Advances in Neural Information Processing Systems 26*, pp. 1493–1501, 2013.

[3] I. Arel, D. C. Rose, and T. P. Karnowski, " Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[4] E. Arias Castro, and D. L. Donoho, "Does median filtering truly preserve edges better than linear filtering?" *The Annals of Statistics*, vol, 37, no. 3, pp. 1172–1206, 2009.

[5] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam Moisy, "An introduction to deep learning," in *ESANN*, pp. 477–488, 2011.

[6] S. Aryal and R. Gutierrez Osuna, "Data driven articulatory synthesis with deep neural networks," *Computer Speech & Language*, vol. 36, pp. 260–273, 2016.

[7] J. M. Baker, L. Deng, J. Glass, S. Khudanpur, C. H. Lee, N. Morgan, and D. O. Shaughnessy, "Developments and directions in speech recognition and understanding, part 1 [dsp education]," *IEEE Transactions on Signal Processing Magazine*, vol. 26, no. 3, pp. 75–80, 2009.

[8] D. H. Ballard, and C. M. Brown, "computer vision," *Prenice-Hall, Englewood Cliffs*, 1982.

[9] S. Bell, and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 98, 2015.

[10] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[12] Y. Bengio, and O. Delalleau, "Justifying and generalizing contrastive divergence," *Neural computation*, vol. 21, no. 6, pp. 1601–1621, 2009.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153–160, 2007.

[14] C. M. Bishop, "Pattern recognition and Machine Learning," *Springer*, 2006.

[15] H. Bourlard, and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.

[16] Y. L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2651–2658, 2011.

[17] Y. L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 111–118, 2010.

[18] R. G. Brown, and P. Y. Hwang, "Introduction to random signals and applied kalman filtering: with matlab exercises and solutions," *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997*, vol. 1, 1997.

[19] B. Chandra and R. K. Sharma, "Fast learning in deep neural networks," *Neurocomputing*, vol. 171, pp. 1205–1215, 2016.

[20] D. Chen, and B. K. W. Mak, "Multitask learning of deep neural networks for low-resource speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 7, pp. 1172–1183, 2015.

[21] H. Chen, J. Liang, and Z. Wang, "Pinning controllability of autonomous Boolean control networks," *Science China Information Sciences*, Vol. 59, No. 7, pp. 1–14, 2016.

[22] Y. Chen, X. Yang, B. Zhong, S. Pan, D. Chen, and H. Zhang, "Cnntracker: Online discriminative object tracking via deep convolutional neural network," *Applied Soft Computing*, vol. 38, pp. 1088–1098, 2016.

[23] X. Chen, Y. Xu, D. W. K. Wong, T. Y. Wong, and J. Liu, "Glaucoma detection based on deep convolutional neural network," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 715–718, 2015.

[24] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[25] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3642–3649, 2012.

[26] R. Collobert, and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

[27] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 9, pp. 1469–1477, 2015.

[28] G. E. Dahl, D. Yu, L. Deng, 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, 2012.

[29] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, 2009.

[30] L. Deng, "Three classes of deep learning architectures and their applications: a tutorial survey," *APSIPA transactions on signal and information processing*, 2012.

[31] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, p. e2, 2014.

[32] L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelstein, "Phonemic hidden markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Transactions on Signal Processing*, vol. 39, no. 7, pp. 1677–1681, 1991.

[33] L. Deng, J. Li, J. T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at microsoft," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8604–8608, 2013.

[34] L. Deng, M. Lennig, F. Seitz, and P. Mermelstein, "Large vocabulary word recognition using context-dependent allophonic hidden markov models," *Computer Speech & Language*, vol. 4, no. 4, pp. 345–357, 1990.

[35] L. Deng, and D. Yu, "Deep convex net: A scalable architecture for speech pattern classification," in *Proceedings of the Interspeech*, 2011.

[36] G. Desjardins, and Y. Bengio, "Empirical evaluation of convolutional RBMs for vision," Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Tech. Rep. 1327, 2008.

[37] C. Ding, L. Xie, and P. Zhu, "Head motion synthesis from speech using deep neural networks," *Multimedia Tools and Applications*, vol. 74, no. 22, pp. 9871–9888, 2015.

[38] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[39] Y. Dong and Y. Wu, "Adaptive cascade deep convolutional neural networks for face alignment," *Computer Standards & Interfaces*, vol. 42, pp. 105–112, 2015.

[40] D. Eigen, J. Rolfe, R. Fergus, and Y. LeCun, "Understanding deep architectures using a recursive convolutional network," *arXiv preprint arXiv:1312.1847*, 2013.

[41] S. Elfwing, E. Uchibe, and K. Doya, "Expected energy-based restricted boltzmann machine for classification," *Neural Networks*, vol. 64, pp. 29–38, 2015.

[42] O. Emad, I. A. Yassine, and A. S. Fahmy, "Automatic localization of the left ventricle in cardiac mri images using deep learning," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 683–686, 2015.

[43] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[44] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proceedings of the ACM International Conference on Multimedia*, pp. 7–16, 2014.

[45] A. Fischer and C. Igel, "Training RBMs based on the signs of the CD approximation of the log-likelihood derivatives," in *ESANN*, 2011.

[46] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[47] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 513–520, 2011.

[48] M. Gong, J. Zhao, J. Liu, Q. Miao, and L. Jiao, "Change detection in synthetic aperture radar images based on deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 125–138, 2016.

[49] J. Gonzalez-Dominguez, I. Lopez-Moreno, P. J. Moreno, and J. Gonzalez-Rodriguez, "Frame-by-frame language identification in short utterances using deep neural networks," *Neural Networks*, vol. 64, pp. 49–58, 2015.

[50] K. Guo, D. Zou, and X. Chen, "3D mesh labeling via deep convolutional neural networks," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 1, p. 3, 2015.

[51] P. Hamel, and D. Eck, "Learning features from music audio with deep belief networks," in *ISMIR*, pp. 339–344, 2010.

[52] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via SADEs," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 487–498, 2015.

[53] M. Henderson, B. Thomson, and S. Young, "Deep neural network approach for the dialog state tracking challenge," in *Proceedings of the SIGDIAL 2013 Conference*, pp. 467–471, 2013.

[54] J. Hennrich, C. Herff, D. Heger, and T. Schultz, "Investigating deep learning for FNIRs based BCI," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 2844–2847, 2015.

[55] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, pp. 599–619, 2012.

[56] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[57] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The "wake-sleep" algorithm for unsupervised neural networks," *Science*, vol. 268, no. 5214, pp. 1158–1161, 1995.

[58] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath , "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[59] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[60] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[61] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length, and helmholtz free energy," *Advances in neural information processing systems*, pp. 3, 1994.

[62] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5659–5670, 2015.

[63] N. Hou, H. Dong, Z. Wang, W. Ren, and F. E. Alsaadi, "Non-fragile state estimation for discrete Markovian jumping neural networks," *Neurocomputing*, Vol. 179, pp. 238–245, 2016.

[64] J. Hu, D. Chen, and J. Du, "State estimation for a class of discrete nonlinear systems with randomly occurring uncertainties and distributed sensor delays," *International Journal of General Systems*, Vol. 43, No. 3-4, pp. 387–401, 2014.

[65] J. Hu, S. Liu, D. Ji, and S. Li, "On co-design of filter and fault estimator against randomly occurring nonlinearities and randomly occurring deception attacks," *International Journal of General Systems*, Vol. 45, No. 5, pp. 619–632, 2016.

[66] F. Hu, G. S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.

[67] J. Hu, Z. Wang, D. Chen, and F. E. Alsaadi, "Estimation, filtering and fusion for networked systems with network-induced phenomena: New progress and prospects," *Information Fusion*, Vol. 31, pp. 65–75, 2016.

[68] J. Hu, Z. Wang, S. Liu, and H. Gao, "A variance-constrained approach to recursive state estimation for time-varying complex networks with missing measurements," *Automatica*, Vol. 64, pp. 155–162, 2016.

[69] J. Hu, Z. Wang, B. Shen, and H. Gao, "Quantised recursive filtering for a class of nonlinear systems with multiplicative noises and missing measurements," *International Journal of Control*, Vol. 86, No. 4, pp. 650–663, 2013.

[70] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–95, 2016.

[71] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 7304–7308, 2013.

[72] Q. X. Huang, H. Su, and L. Guibas, "Fine-grained semi-supervised labeling of large shape collections," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 190, 2013.

[73] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[74] I. G. Y. Bengio, and A. Courville, "Deep learning," *book in preparation for MIT Press [Online]*, 2016

[75] G. Indiveri, and S. C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[76] N. Jaitly, and G. Hinton, "Learning a better representation of speech soundwaves using restricted Boltzmann machines," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5884–5887, 2011.

[77] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153, 2009.

[78] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3370–3377, 2012.

[79] B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.)," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 307–309, 1986.

[80] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D mesh segmentation and labeling," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 102, 2010.

[81] T. G. Kang, K. Kwon, J. W. Shin, and N. S. Kim, "NMF-based target source separation using deep neural network," *IEEE Signal Processing Letters*, vol. 22, no. 2, pp. 229–233, 2015.

[82] R. Kelley, L. Wigand, B. Hamilton, K. Browne, M. Nicolescu, and M. Nicolescu, "Deep networks for predicting human intent

with respect to objects," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pp. 171–172, 2012.

[83] J. Kim, V. D. Calhoun, E. Shim, and J. H. Lee, "Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia," *NeuroImage*, vol. 124, part. A, pp. 127–146, 2016.

[84] I. J. Kim, and X. Xie, "Handwritten hangul recognition using deep convolutional neural networks," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 1, pp. 1–13, 2015.

[85] Y. Kim, and T. Moon, "Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2015.

[86] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems 25*, pp. 1097–1105, 2012.

[87] A. Krizhevsky, and G. E. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, 2010.

[88] N. D. Lane, and P. Georgiev, "Can deep learning revolutionize mobile sensing?" in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 117–122, 2015.

[89] H. Larochelle, and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proceedings of the 25th international conference on Machine learning*, pp. 536–543, 2008.

[90] Y. LeCun, and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, 1995.

[91] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[92] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[93] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. 97–104, 2004.

[94] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609–616, 2009.

[95] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems 22*, pp. 1096–1104, 2009.

[96] J. Lerouge, R. Herault, C. Chatelain, F. Jardin, and R. Modzelewski, "IODA: an input/output deep architecture for image labeling," *Pattern Recognition*, vol. 48, no. 9, pp. 2847–2858, 2015.

[97] G. Li, L. Deng, Y. Xu, C. Wen, W. Wang, J. Pei, and L. Shi, "Temperature based restricted Boltzmann machines," *Scientific reports*, vol. 6, no. 19133, 2016.

[98] S. Li, Z. Q. Liu, and A. Chan, "Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 482–489, 2014.

[99] X. Li, Y. Yang, Z. Pang, and X. Wu, "A comparative study on selecting acoustic modeling units in deep neural networks based large vocabulary chinese speech recognition," *Neurocomputing*, vol. 170, pp. 251–256, 2015.

[100] B. Liao, J. Xu, J. Lv, and S. Zhou, "An image retrieval method for binary images based on DBN and Softmax classifier," *IETE Technical Review*, vol. 32, no. 4, pp. 294–303, 2015.

[101] Z. H. Ling, S. Y. Kang, H. Zen, A. Senior, M. Schuster, X. J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[102] Q. Liu, Z. Wang, X. He and D. H. Zhou, "Event-based H_∞ consensus control of multi-agent systems with relative output feedback: the finite-horizon case," *IEEE Transactions on Automatic Control*, Vol. 60, No. 9, pp. 2553–2558, 2015.

[103] Q. Liu, Z. Wang, X. He and D. H. Zhou, "Event-based recursive distributed filtering over wireless sensor networks," *IEEE Transactions on Automatic Control*, Vol. 60, No. 9, pp. 2470–2475, 2015.

[104] Q. Liu, Z. Wang, X. He, G. Ghinea and F. E. Alsaadi, "A resilient approach to distributed filter design for time-varying systems under stochastic nonlinearities and sensor degradation," *IEEE Transactions on Signal Processing*, accepted for publication.

[105] F. Luus, B. Salmon, F. Van Den Bergh, and B. Maharaj, "Multiview deep learning for land-use classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2448–2452, 2015.

[106] L. Ma, Z. Wang, and H. -K. Lam, "Event-triggered mean-square consensus control for time-varying stochastic multi-agent system with sensor saturations," *IEEE Transactions on Automatic Control*, 2016, DOI: 10.1109/TAC.2016.2614486.

[107] L. Ma, Z. Wang, H. -K. Lam, and N. Kyriakoulis, "Distributed event-based set-membership filtering for a class of nonlinear systems with sensor saturations over sensor networks," *IEEE Transactions on Cybernetics*, 2016, DOI: 10.1109/TCYB.2016.2582081.

[108] L. Ma, Z. Wang, H. -K. Lam, F. E. Alsaadi, and X.Liu, "Robust filtering for a class of nonlinear stochastic systems with probability constraints," *Automation and Remote Control*, Vol. 77, No. 1, pp. 37–54, 2016.

[109] L. Ma, Z. Wang, and H. -K. Lam, "Mean-square H_∞ consensus control for a class of nonlinear time-varying stochastic multiagent systems: the finite-horizon case," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 2016, DOI: 10.1109/TSMC.2016.2531657.

[110] A. Makhzani, and B. Frey, "k-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[111] R. Manor, and A. B. Geva, "Convolutional neural network for multi-category rapid serial visual presentation BCI," *Frontiers in computational neuroscience*, vol. 9, 2015.

[112] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 52–59, 2011.

[113] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through FFTs," *arXiv preprint arXiv:1312.5851*, 2013.

[114] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using I-vectors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1938–1949, 2015.

[115] V. Mnih, H. Larochelle, and G. E. Hinton, "Conditional restricted Boltzmann machines for structured output prediction," *arXiv preprint arXiv:1202.3748*, 2012.

[116] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[117] A. Mohamed, G. Dahl, and G. E. Hinton, "Deep belief networks for phone recognition," in Nips workshop on deep learning for speech recognition and related applications, vol. 1, no. 9, p. 39, 2009.

[118] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in *INTERSPEECH*, pp. 2846–2849, 2010.

[119] A. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5060–5063, 2011.

[120] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 7–13, 2012.

[121] V. Nair, and G. E. Hinton, "3D object recognition with deep belief nets," in *Advances in Neural Information Processing Systems*, pp. 1339–1347, 2009.

[122] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.

[123] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[124] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 427–436, 2015.

[125] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[126] J. Padmanabhan, and M. J. Johnson Premkumar, "Machine learning in automatic speech recognition: A survey," *IETE Technical Review*, vol. 32, no. 4, pp. 240–251, 2015.

[127] Y. Pan, W. Huang, Z. Lin, W. Zhu, J. Zhou, J. Wong, and Z. Ding, "Brain tumor grading based on neural networks and convolutional neural networks," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 699–702, 2015.

[128] C. Poultney, S. Chopra, and Y. L. Cun, "Efficient learning of sparse representations with an energy-based model," in *Advances in neural information processing systems*, pp. 1137–1144, 2006.

[129] D. Quang, Y. Chen, and X. Xie, "DANN: a deep learning approach for annotating the pathogenicity of genetic variants," *Bioinformatics*, vol 31, pp. 761–763, 2014.

[130] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[131] Z. Ren, Y. Deng, and Q. Dai, "Local visual feature fusion via maximum margin multimodal deep neural network," *Neurocomputing*, vol. 175, pp. 427–432, 2016.

[132] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[133] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 833–840, 2011.

[134] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. r. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015.

[135] T. N. Sainath, B. Kingsbury, A. r. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pp. 297–302, 2013.

[136] R. Salakhutdinov, and G. E. Hinton, "Using deep belief nets to learn covariance kernels for gaussian processes," in *Conference on Neural Information Processing Systems*, pp. 1249–1256, 2007.

[137] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[138] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 3288–3291, 2012.

[139] T. Shinozaki, and S. Watanabe, "Structure discovery of deep neural network based on evolutionary algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4979–4983, 2015.

[140] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[141] S. Singh, M. Tripathy, and R. Anand, "Subjective and objective analysis of speech enhancement algorithms for single channel speech patterns of indian and english languages," *IETE Technical Review*, vol. 31, no. 1, 2014.

[142] Y. Song, J. Hu, D. Chen, D. Ji, and F. Liu, "Recursive approach to networked fault estimation with packet dropouts and randomly occurring uncertainties," *Neurocomputing*, Vol. 214, pp. 340–349, 2016.

[143] J. Song, and Y. Niu, "Resilisent finite-time stabilization of fuzzy stochastic systems with randomly occurring uncertainities and randomly occurring gain fluctuations," *Neurocomputing*, Vol. 171, pp. 444–451, 2016.

[144] M. Sonka, V. Hlavac, and R. Boyle, "Image processing, analysis, and machine vision," *Cengage Learning*, 2014.

[145] N. Srivastava, and R. Salakhutdinov, "Learning representations for multimodal data with deep belief nets," in *International Conference on Machine Learning Workshop*, 2012.

[146] M. Sun, X. Zhang, and T. F. Zheng, "Unseen noise estimation using separable deep auto encoder for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 93–104, 2016.

[147] I. Sutskever, and T. Tieleman, "On the convergence properties of contrastive divergence," in *International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 789–795, 2010.

[148] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, pp. 2553–2561, 2013.

[149] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, "A real-time hand posture recognition system using deep neural networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 2, p. 21, 2015.

[150] J. Tang, C. Deng, G. B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1174–1185, 2015.

[151] Y. Tang, R. Salakhutdinov, and G. E. Hinton, "Deep lambertian networks," *arXiv preprint arXiv:1206.6445*, 2012.

[152] Y. Tang, R. Salakhutdinov, and G. E. Hinton, "Robust boltzmann machines for recognition and denoising," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2264–2271, 2012.

[153] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series," *The Journal of Machine Learning Research*, vol. 12, pp. 1025–1068, 2011.

[154] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[155] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[156] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[157] S. Wang, S. Weng, J. Ma, and Q. Tang, "DeepCNF-D: Predicting Protein Order/Disorder Regions by Weighted Deep Convolutional Neural Fields," *International journal of molecular sciences*, vol. 16, no. 8, pp. 17315–17330, 2015.

[158] C. Wen, D. Wu, H. Hu, and W. Pan, "Pose estimation-dependent identification method for field moth images using deep learning architecture," *Biosystems Engineering*, vol. 136, pp. 117–128, 2015.

[159] G. Wu, W. Lu, G. Gao, C. Zhao, and J. Liu, "Regional deep learning model for visual tracking," *Neurocomputing*, vol. 175, pp. 310–323, 2016.

[160] Z. Wu, K. Zhao, X. Wu, X. Lan, and H. Meng, "Acoustic to articulatory mapping with deep neural network," *Multimedia Tools and Applications*, vol. 74, no. 22, pp. 9889–9907, 2015.

[161] S. Xue, O. Abdel Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[162] F. Yang, H. Dong, Z. Wang, W. Ren, and F. E. Alsaadi, "A new approach to non-fragile state estimation for continuous neural networks with time-delays," *Neurocomputing*, Vol. 197, pp. 205–211, 2016.

[163] H. Yang, Z. Wang, H. Shu, F. E. Alsaadi, and T. Hayat, "Almost sure H_∞ sliding mode control for nonlinear stochastic systems with Markovian switching and time-delays," *Neurocomputing*, Vol. 175, Part A, pp. 392–400, 2016.

[164] T. Yoshioka, and M. J. Gales, "Environmentally robust asr front-end for deep neural network acoustic models," *Computer Speech & Language*, vol. 31, no. 1, pp. 65–86, 2015.

[165] D. Yu, and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[166] Y. Yu, H. Dong, Z. Wang, W. Ren, and F. E. Alsaadi, "Design of non-fragile state estimators for discrete time-delayed neural networks with parameter uncertainties," *Neurocomputing*, Vol. 182, pp. 18–24, 2016.

[167] Y. Yuan, and F. Sun, "Delay-dependent stability criteria for time-varying delay neural networks in the delta domain," *Neurocomputing*, Vol. 125, pp. 17-21, 2014.

[168] Y. Yuan, F. Sun, H. Liu, and H. Yang, "Finite frequency property-based robust control for singularly perturbed system," *IET Control Theory & Applications*, Vol. 9, No. 2, pp. 203–210, 2015.

[169] Y. Yuan, F. Sun, and Q. Zhu, "Resilient control in the presence of DoS Attack: switched system approach," *International Journal of Control, Automation, and Systems*, Vol. 13, No. 6, pp. 1425–1435, 2015.

[170] Y. Yuan, H. Yuan, L. Guo, H. Yang, and S. Sun, "Robust control of networked control system under DoS attacks: a unified game approach," *IEEE transactions on Industrial Informatics*, DOI 10.1109/TII.2016.2542208.

[171] M. D. Zeiler, and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[172] N. Zeng, Z. Wang, H. Zhang, and F. E. Alsaadi, "A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay," *Cognitive Computation*, Vol. 8, No. 2, pp. 143–152, 2016.

[173] N. Zeng, Z. Wang, H. Zhang, W. Liu, and F. E. Alsaadi, "Deep Belief Networks for Quantitative Analysis of a Gold Immunochromatographic Strip," *Cognitive Computation*, Vol. 8, No. 4, pp. 684–692, 2016.

[174] N. Zeng, Z. Wang, and H. Zhang, "Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter," *Science China Information Sciences*, Vol. 59, No. 11, pp. 112204, 2016.

[175] J. Zhang, L. Ma, and Y. Liu, "Passivity analysis for discrete-time neural networks with mixed time-delays and randomly occurring quantization effects," *Neurocomputing*, Vol. 216, pp. 657–665, 2016.

[176] Y. Zhou, Q. Hu, J. Liu, and Y. Jia, "Combining heterogeneous deep neural networks with conditional random fields for chinese dialogue act recognition," *Neurocomputing*, vol. 168, pp. 408–417, 2015.

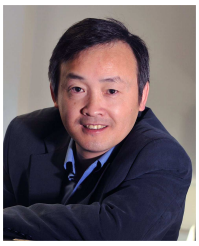
[177] S. Zhu, Z. Shi, C. Sun, and S. Shen, "Deep neural network based image annotation," *Pattern Recognition Letters*, vol. 65, pp. 103–108, 2015.

Weibo Liu received his B. S. degree in electrical engineering from the Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool, UK, in 2015. He is currently pursuing the Ph. D. degree in Computer Science at Brunel University London, London, UK. His research interests include big data analysis and deep learning techniques.

Zidong Wang was born in Jiangsu, China, in 1966. He received the B. Sc. degree in mathematics in 1986 from Suzhou University, Suzhou, China, and the M. Sc. degree in applied mathematics in 1990 and the Ph. D. degree in electrical engineering in 1994, both from Nanjing University of Science and Technology, Nanjing, China.

He is currently a Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany and the UK. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published more than 300 papers in refereed international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, William Mong Visiting Research Fellowship of Hong Kong.

Prof. Wang serves (or has served) as the Editor-in-Chief for Neurocomputing and an Associate Editor for 12 international journals, including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Signal Processing, and IEEE Transactions on Systems, Man, and Cybernetics - Part C. He is a Fellow of the IEEE, a Fellow of the Royal Statistical Society and a member of program committee for many international conferences.



Xiaohui Liu received the B.Eng. degree in computing from Hohai University, Nanjing, China, in 1982 and the Ph. D. degree in computer science from Heriot-Watt University, Edinburg, U.K., in 1988. He is currently a Professor of Computing at Brunel University. He leads the Intelligent Data Analysis (IDA) Group, performing interdisciplinary research involving artificial intelligence, dynamic systems, image and signal processing, and statistics, particularly for applications in biology, engineering and medicine. Professor Liu serves on editorial boards of four computing journals, founded the biennial international conference series on IDA in 1995, and has given numerous invited talks in bioinformatics, data mining and statistics conferences.

Nianyin Zeng was born in Fujian Province, China, in 1986. He received the B.Eng. degree in electrical engineering and automation in 2008 and the Ph. D. degree in electrical engineering in 2013, both from Fuzhou University. From October 2012 to March 2013, he was a RA in the Department of Electrical and Electronic Engineering, the University of Hong Kong.

Currently, he is an Assistant Professor with the Department of Instrumental & Electrical Engineering of Xiamen University. His current research interests include intelligent data analysis, computational intelligent, time-series modeling and applications. He is the author or co-author of several technical papers and also a very active reviewer for many international journals and conferences.

Dr. Zeng is currently serving as an Editorial board member for Biomedical Engineering Online (Springer), Journal of Advances in Biomedical Engineering and Technology, and Smart Healthcare.

Yurong Liu received the BSc degree in mathematics from Suzhou University, Suzhou, China, in 1986, the M. Sc. degree in applied mathematics from Nanjing University of Science and Technology, Nanjing, China, in 1989, and the PhD degree in applied mathematics from Suzhou University, Suzhou, China, in 2000.

Currently, he is a Professor at the Department of Mathematics, Yangzhou University, Yangzhou, China. His current interests include neural networks, nonlinear dynamics, time-delay systems, and chaotic dynamics.

Fuad E. Alsaadi received the B.S. and M.Sc. degrees in electronic and communication from King AbdulAziz University, Jeddah, Saudi Arabia, in 1996 and 2002. He then received the Ph.D. degree in Optical Wireless Communication Systems from the University of Leeds, Leeds, UK, in 2011. Between 1996 and 2005, he worked in Jeddah as a communication instructor in the College of Electronics and Communication. He is currently an associate professor of the Electrical and Computer Engineering Department within the Faculty of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. He published widely in the top IEEE communications conferences and journals and has received the Carter award, University of Leeds for the best PhD. He has research interests in optical systems and networks, signal processing, synchronization and systems design.