Volume 7, Issue 1, January 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Deep Learning Architectures, Algorithms for Speech Recognition: An Overview

Banumathi .A .C
Department of Computer Science
Mother Teresa University,
Tamil Nadu, India

Dr. E. Chandra
Department of Computer Science
Bharathiar University,
Tamil Nadu, India

DOI: 10.23956/ijarcsse/V7I1/0107

Abstract*— Speech is the most natural form of human communication and speech processing has been one of the most inspiring expansions of signal processing. The Speech Recognition could be of great interest in the future. Speech is an easy mode of communication for the people to interact with the computer, rather than using keyboard and mouse. The goal of Deep Learning is to invent a machine which can sense, remember, learn, and recognize like a real human being. In this paper, we explore the different Deep Learning architectures and the algorithms applied to train the architectures. Our paper brings a study of the different classifiers of Neural networks like Recurrent Neural Network (RNN), Deep Recurrent Neural Network(RNN), and Deep Belief Network(DBN) and the algorithms used to train them.

Keywords*— Convolutional Neural Network(CNN), Recurrent Neural Network (RNN), Restricted Boltzmann Machines(RBM), Deep Belief Network(DBN), Deep Convex Nets(DCN), Deep Neural Networks(DNN), Deep AutoEncoder, Deep Stacking Network(DSN).

I. INTRODUCTION

Speech recognition means enabling the uttered words of the user by the machines. Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them into an understandable form. Neural networks have a long history in speech recognition, usually in combination with hidden Markov models [1, 2]. They have gained attention in recent years with the dramatic improvements in acoustic modeling yielded by deep feed forward networks. Deep Learning is a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for unsupervised feature learning and for pattern analysis/classification. The essence of deep learning is to compute hierarchical features or representations of the observational data, where the higher -level features or factors are defined from lower -level ones. Capability of deep architectures of representing acoustic features and training on large amount of data has motivated many researchers towards use of deep learning in speech processing. It is machine learning methodology. Deep learning has turned out to be successful in tackling with many AI problems including speech information processing. In the field of Artificial Intelligence there is need for a model which is capable of processing the complex input data and solving different kinds of task. Deep architectures has been proved to be such kind of model. It is believed that deep architectures have good learning algorithms and excellent performance. There are several deep architectures, but CNNs and DBNs are the two milestones in the field of speech processing.

II. THREE BROAD CLASSES OF DEEP ARCHITECTURE

"Deep architectures are compositions of many layers of adaptive non-linear components, in other words, they are cascades of parameterized non-linear modules that contain trainable parameters at all levels". The greatest difference between the shallow architectures and the deep architectures are the number of layers. Why the researches are looking for the deep architecture? The reason is, we need an architecture which is capable of learning the features from the data given to it, as well as dealing with the unlabeled data. An efficient and general learning algorithm must be able to apply to this architecture. Additionally, this architecture should be able to be used for solving different kinds of AI problems. With the purpose of finding an architecture that meets the requirements above, some researchers started to look back to the multi-layer neural network, trying to exploit its advantages related to "deep" architecture and overcome the limitations.

Deep Learning refers to the machine learning Techniques and Architectures based on how the architectures and techniques are intended for use in, synthesis/generation or recognition/classification. we can broadly categorize most of the work in this area into three main classes:

1. **Deep networks for unsupervised or generative learning. (**Generative deep architectures.)
2. **Deep networks for supervised learning**.(Discriminative deep architectures)
3. **Hybrid deep architectures**.

Generative deep architectures:, It is also known as "unsupervised feature learning". This architecture is intended to characterize the high-order correlation properties of the observed or visible data for pattern analysis or synthesis purposes, and/or characterize the joint statistical distributions of the visible data and their associated classes. In this type of architecture, learning the lower levels of networks is difficult, especially when training data are limited. Therefore, it is desirable to learn each lower layer without relying on all the layers above and to learn all layers in a greedy, layer-by-layer manner from bottom up. Another prominent type of generative model is deep Boltzmann machine or DBM (Salakhutdinov and Hinton, 2009, 2012; Srivastava and Salakhutdinov 2012).The use of Bayes rule can turn this type of architecture into a discriminative one.

Discriminative deep architectures: This type of architecture is applied for shallow architectures HMM. (e.g., Juang et al., 1997; Chengalvarayan and Deng,1998;). These are intended to directly provide discriminative power for pattern classification, often by characterizing the posterior distributions of classes conditioned on the visible data. These networks mainly depend on the traditional Neural Network or MLP(Multi Level Perceptron). It argues for the importance of the increased width and Depth of the network. In the recent work a new type of Deep Learning Architecture called the Deep Stacking Network is formed (DSN) which focuses on the on discrimination with scalable, parallelization learning relying on more generative component. Another type of discriminative deep architecture is Convolutional Neural Network(CNN). It has been highly effective and been commonly used in the computer vision and image recognition and in the recent research it is found to be very effective for Speech Recognition making some small changes in the image recognition.

Hybrid deep architectures:

The Hybrid refers to the architecture that comprises of both generative and discriminative model components. The generative component is exploited to help with discrimination, which is the final goal of the hybrid architecture. These have the goal of discrimination but is assisted (often in a significant way) with the outcomes of generative architectures via better optimization or/and regularization, or when discriminative criteria are used to learn the parameters in any of the deep generative models in category.

III. DIFFERENT TYPES OF DEEP ARCHITECTURES

The different types of deep architectures are given below:

A) Convolutional Neural Network(CNN)
B) Recurrent Neural Network (RNN)
C) Restricted Boltzmann Machines(RBM)
D) Deep Belief Network(DBN)
E) Deep Convex Nets(DCN)
F) Deep Neural Networks(DNN)
G). Deep AutoEncoder
H). Deep Stacking Network(DSN)

The Above deep architectures fall in category of generative, discriminative and hybrid deep architectures [12].

3.a. Convolution Neural Network(CNN):

CNNs belong to the feed forward network, but it combines three architectural ideas to ensure some degree of shift and distortion invariance: local receptive field, shared weights, and sometimes spatial or temporal sub sampling. The local receptive field is a small portion of the data, it has another name called the shift window; the shared weights makes all the feature maps at the same level to be formed by the same parameterization; spatial and temporal sub sampling (usually the spatial subsuming) can shrink the feature maps by dimensions in the previous level, forming a new set of feature maps to which the convolution process can apply again. It belongs to the feed-forward network, but it combines three architectural ideas to ensure some degree of shift and distortion invariance:

1. ***local receptive field:*** It has another name called the shift window. It contains a small portion of the data
2. ***shared weights:*** It maintains all the feature maps at the same level formed by the same parametrization.
3. ***Spatial and temporal subsampling***:(usually the spatial subsampling) It forms a new set of feature maps by shrinking the dimension of the feature map in the previous level.

The CNNs is comprised of a sequence of convolution process and sub sampling process. The convolution process convolves an input with a trainable filter f_x and adds a trainable bias b_x to produce the convolution layer C_x, and the sub sampling process sums a neighborhood (four pixels), weights by scalar w_{x+1}, adds the trainable bias b_{x+1}, and passes through a sigmoid function to produce a smaller feature map S_{x+1}. Then, C_x is treated as the input data, converted to a set of smaller-dimension feature maps S_{x+1} via the sub sampling process. The input is converted into a convolution layer C_x via the convolution process. Then, C_x is treated as the input data, converted to a set of smaller-dimension feature maps S_{x+1} via the sub sampling process. We can see that after each layer, the dimensions of the feature maps are decreased, which makes CNNs a useful model to reduce the number of parameters that must be learned and thus improves upon general feed-forward back-propagation training".

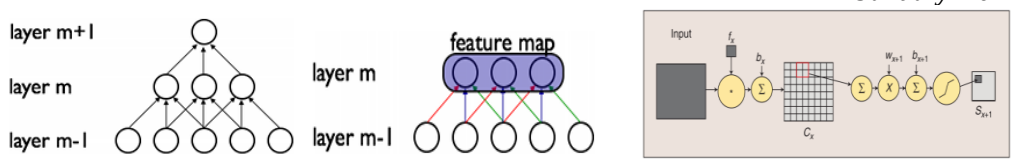


Fig1. **a.** Sparse connectivity of CNNs **b**. Shared weights of CNNs **c.** The convolution process and subsampling process

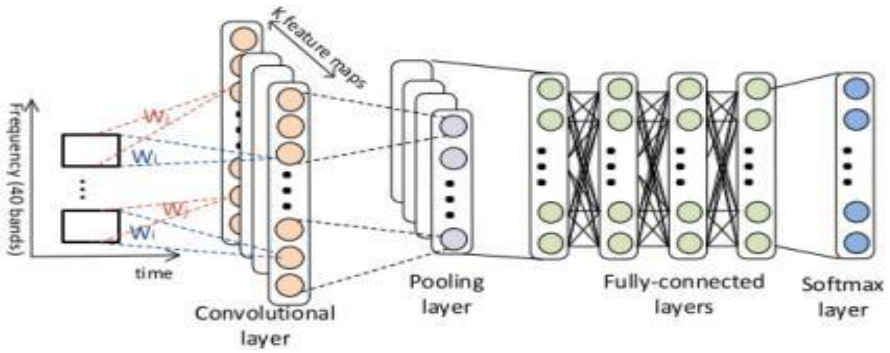
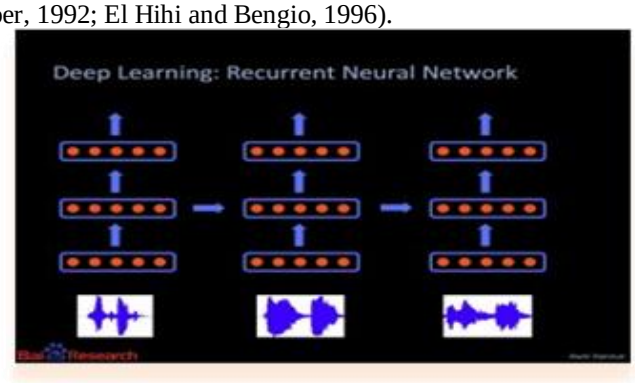


Fig .2. Convolution Neural Network. A complete procedure of CNNs. The input passes through a sequence of covolution process and subsampling process.

3.b. Recurrent Neural Networks(RNN)

Recurrent Neural Networks (RNNs) are popular models that have shown great promise in many NLP(Natural Language Processing) tasks. Recurrent neural networks (RNNs) are a powerful model for sequential data. End-to-end training methods such as Connectionist Temporal Classification make it possible to train RNNs for sequence labeling problems where the input-output alignment is unknown. The fundamental feature of a Recurrent Neural Network (RNN) is that the network contains at least one feed-back connection, so the activations can flow round in a loop. The concept of depth in an RNN is not as clear as it is in feedforward neural networks. By carefully analyzing and understanding the architecture of an RNN, however, we find three points of an RNN which may be made deeper; (1) input-to-hidden function, (2) hidden-to-hidden transition and (3) hidden-to-output function. Based on this observation, we propose two novel architectures of a deep RNN which are orthogonal to an earlier attempt of stacking multiple recurrent layers to build a deep RNN (Schmidhuber, 1992; El Hihi and Bengio, 1996).



Given an input sequence x = (x_1; : : : ; x_T), a standard recurrent neural network (RNN) computes the hidden vector sequence h = (h_1; : : : ; h_T) and output vector sequence y = (y_1; : : : ; y_T) by iterating the following equations from t = 1 to T:

$h_t = H(W_{xh}x_t+W_{hh}h_{t-1}+ b_h)$…… (1)

$y_t= W_{hy} h_t+b_y$ …… .(2)

Recurrent neural networks (RNNs) contain cyclic connections that make them a more powerful tool to model such sequence data than feed forward neural networks. A recurrent neural network (RNN) is a neural network that simulates a discrete-time dynamical system that has an input X_t, an output Y_t and a hidden state h_t.

3.c. Restricted Boltzmann Machines(RBM)

A special BM(Boltzman Machine) consisting of a layer of visible units and a layer of hidden units with no visible-visible or hidden - hidden connections is RBM. Deep Boltzmann machine (DBM) is a special BM where the hidden units are organized in a deep layered manner, only adjacent layers are connected, and there are no visible -visible or hidden -hidden connections within the same layer. An RBM is a neural network that contains two layers. It has a single layer of hidden units that are not connected with each other. Additionally, the hidden units have undirected, symmetrical connections to a layer of visible units. Each unit, including both hidden units and visible units, in the network has a bias. The value of visible units and hidden units are often binary or stochastic units (assume 0 or 1 based on probability)(Feng Liu 1, Bingquan Liu Entropy 2015),

An RBM is a special type of Markov random field that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible or observable units. RBMs can be represented as bipartite graphs, where all visible units are connected to all hidden units, and there are no visible-visible or hidden-hidden connections. In an RBM, the joint distribution p(**v**, **h**; θ) over the visible units **v** and hidden units **h**, given the model parameters θ, is defined in terms of an energy function E(**v**, **h**; θ) of

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{exp(-E(\mathbf{v}, \mathbf{h}; \theta))}{Z}$$

where $Z = \Sigma\mathbf{v}\Sigma\mathbf{h}\ exp(-E(\mathbf{v}, \mathbf{h}; \theta))$ is a normalization factor or partition function, and the marginal probability that the model assigns to a visible vector **v** is

$$p(\mathbf{v}; \theta) = \frac{\Sigma \boldsymbol{h}\ exp(-E(\mathbf{v}, \mathbf{h}; \theta))}{Z}$$

Careful training of RBMs is essential to the success of applying RBM and related deep learning techniques to solve practical problems. See the Technical Report (Hinton 2010) for a very useful practical guide for training RBMs.

The RBM discussed above is a generative model, which characterizes the input data distribution using hidden variables and there is no label information involved. However, when the label information is available, it can be used together with the data to form the joint "data" set. Then the same CD learning can be applied to optimize the approximate "generative" objective function related to data likelihood. Further, and more interestingly, a "discriminative" objective function can be defined in terms of conditional likelihood of labels. This discriminative RBM can be used to "fine tune" RBM for classification tasks (Larochelle and Bengio, 2008).

3.d. Deep belief network (DBN):

A Deep belief network is not the same as a Deep Neural Network. A deep belief network has undirected connections between some layers. This means that the topology of the DNN and DBN is different by definition. The undirected layers in the DBN are called Restricted Boltzmann Machines. These layers can be trained using an unsupervised learning algorithm (Contrastive Divergence) that is very fast. In 2006 Hinton discovered that much better results could be achieved in deeper architectures when each layer (RBM) is pre-trained with an unsupervised learning algorithm (Contrastive Divergence). Then the Network can be trained in a supervised way using backpropagation in order to "fine-tune" the weights.

DBNs is a hybrid model consisting of two parts. As shown in figure 2 shows, the top two levels are undirected graph model which form the associative memory, the remaining layers are directed graph model which is actually a stacked Restricted Boltzmann Machines(RBM). Different from the CNNs, DBNs is a stochastically learning architecture whose object function depends on the learning purpose. Generally speaking, DBNs can be trained as a discriminative model(Inference problem, classification problem,etc...) or a generative model(generate training data etc.).

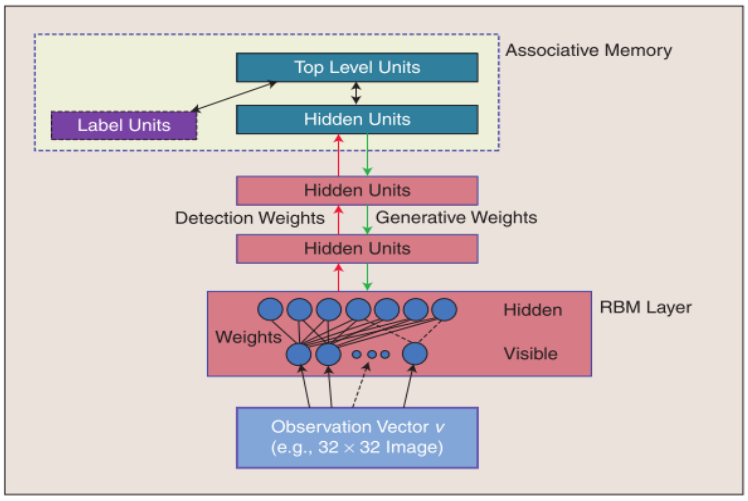


Fig.2. Illustration on the Deep Belief Networks(DBNs)

In figure2, the discriminative model is a "bottoms-up" procedure(red arrow), trying to learn the Detection weights via optimizing a posterior probability P(Label/Obversion); the generative model is a \top-down" procedure(green arrow), trying to learn the Generative weights via optimizing a joint probability P(*Label*,*Observation*). The details of the learning strategies will be discussed in section 3.

The greatest advantage of DBNs is it capability of "learning features", which is achieved by a "layer-by-layer" learning strategies where the higher-level features are learned from the previous layers and the higher-level features are believed to be more complicated and better reacts the information contained in the input data's structures. In the perspective of math, we can prove that after we add another layer of features, we improve a variational lower bound on the log probability of the training data [1, 7].

The probabilistic generative models composed of multiple layers of stochastic, hidden variables. The top two layers have undirected, symmetric connections between them. The lower layers receive top -down, directed connections from the layer above. The two most significant properties of deep belief nets are:

- There is an efficient, layer-by-layer procedure for learning the top-down, generative weights that determine how the variables in one layer depend on the variables in the layer above.
- After learning, the values of the latent variables in every layer can be inferred by a single, bottom-up pass that starts with an observed data vector in the bottom layer and uses the generative weights in the reverse direction.
- The main problem with the deep neural network architectures was the learning process, since the ordinary gradient descent algorithm does not work well and sometimes it makes the training quite impossible for a DBN. To circumvent this problem, a greedy layer wise unsupervised pre-training can be used. After the pre-training it is possible to do a successful supervised learning, done by a procedure called fine tuning, using the renowned gradient descent. The pre-training phase impacts on the choice of initial weights values for the actual supervised training stage. In practice it works magnificently better than the conventional random initialization of the weights, and causes to avoid local minima while using gradient descent in back propagation.

DBNs are constructed by stacking many layers of restricted Boltzmann machines. An RBM comprises two layers, one is the visible and the other is hidden. Once we stack two RBMs on top of each other, the hidden layer of lower becomes visible to the top. The goal here is using the multiple layers of RBMs to model (represent) as close as possible to the reality, the distribution of the input. We are achieving this goal by multiple layers of non-linearity, which results in extraction the more accurate probabilistic representation for the input

3.e. Deep Convex Nets(DCN)

A DCN includes a variable number of layered modules, wherein each module is a specialized neural network consisting of a single hidden layer and two trainable sets of weights. More particularly, the lowest module in the DCN comprises a first linear layer with a set of linear input units, a non-linear layer with a set of non-linear hidden units, and a second linear layer with a set of linear output units. For instance, if the DCN is utilized in connection with recognizing an image, the input units can correspond to a number of pixels (or extracted features) in the image, and can be assigned values based at least in part upon intensity values, RGB values, or the like corresponding to the respective pixels.

If the DCN is utilized in connection with speech recognition, the set of input units may correspond to samples of speech waveform, or the extracted features from speech waveforms, such as power spectra or cepstral coefficients. Note the use of speech waveform as the raw features to a speech recognizer is not a crazy idea. An early study for an HMM-like system (i.e., the hidden filter) that models speech waveform directly as the observation can be found in [9]. And many years later the use of more powerful Restricted Boltzmann Machine (RBM) overcomes some difficulty encountered earlier [10].

The hidden layer of the lowest module of a DCN comprises a set of non-linear units that are mapped to the input units by way of a first, lower-layer weight matrix, which we denote by W. For instance, the weight matrix may comprise a plurality of randomly generated values between zero and one, or the weights of an RBM trained separately. The non-linear units may be sigmoidal units that are configured to perform non-linear operations on weighted outputs from the input units (weighted in accordance with the first weight matrix W).

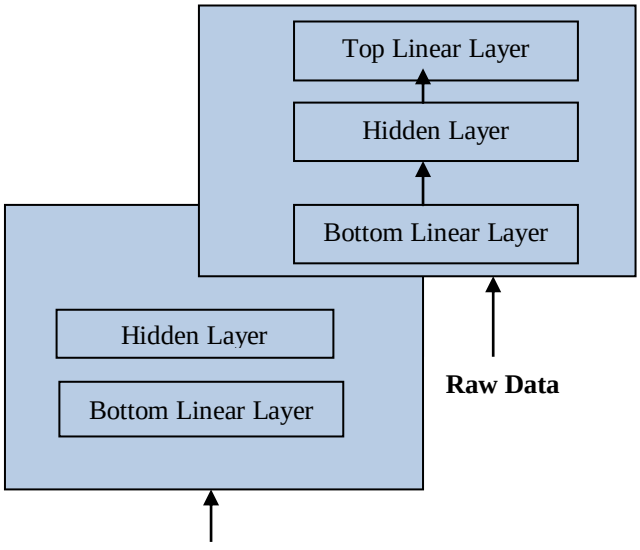


Fig 3. Deep Convex Nets(DCN)

The second, linear layer in any module of a DCN includes a set of output units that are representative of the targets of classification. For instance, if the DCN is configured to perform digit recognition (e.g., the digits 1-10), then the plurality of output units may be representative of the values 1, 2, 3, and so forth up to 10 with a 0-1 coding scheme. If the DCN is configured to perform ASR, then the output units may be representative of phones, HMM states of phones, or context-dependent HMM states of phones in a way that is similar to [5][6]. The non-linear units in each module of the DCN may be mapped to a set of the linear output units by way of a second, upper-layer weight matrix, which we denote by U.

This second weight matrix can be learned by way of a batch learning process, such that learning can be undertaken in parallel. Convex optimization can be employed in connection with learning U. For instance, U can be learned based at least in part upon the first weight matrix W, values of the coded classification targets, and values of the input units.

3.f. Deep Neural Networks(DNN)

A multilayer network with many hidden layers, whose weights are fully connected and are often initialized (pre -trained) using stacked RBMs or DBN. (In the literature, DBN is sometimes used to mean DNN). Deep auto –encoder is a DNN whose output target is the data input itself, often pre -trained with DBN or using distorted training data to regularize the learning.

Indeed, most industrial speech recognition systems rely on Deep Neural Networks as a component, usually combined with other algorithms. Many researchers have long believed that Deep Neural Networks (DNNs) could provide even better accuracy for speech recognition if they were used for the entire system, rather than just as the acoustic modeling component. However, it has proven difficult to find an end-to-end speech recognition system based on Deep Learning that improves on the state of the art before passing it on to the next, until finally the last layer provides the output. The main difference between the RNN and the DRNN is that, firstly the RNN undergoes only one layer before it comes out in the output layer and secondly processing of time scales at several times. Common RNNs do not explicitly support multiple time scales, and any temporal hierarchy that is present in the input signal needs to be embedded implicitly in the network dynamics.

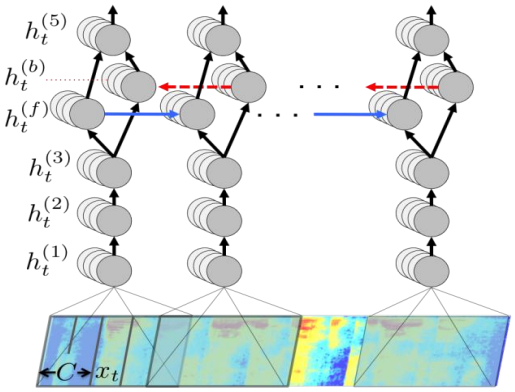


Figure 4. The structure of the Deep Neural Network showing the layers.

Optimization techniques used in Machine Learning play an important role in the training of the Neural Network in regression and classification tasks. The standard training algorithm for deep neural networks (DNNs) is stochastic gradient descent (SGD). Typically, the stochastic gradient is computed on mini-batches. On large and redundant datasets, SGD quickly reaches a region with acceptable performance. On large datasets, stochastic gradient descent improves quickly during the beginning of the optimization. But since it does not make use of second order information, its asymptotic convergence behavior is slow and almost stagnate and thereby falsely indicate convergence. Another drawback of stochastic gradient descent is that it can only be parallelized within mini batches. The Hessian-free algorithm is a second order batch optimization algorithm that does not suffer from these problems. In a recent work, Hessian-free optimization has been applied to a training of deep neural networks according to a sequence criterion. In contrast to SGD, batch algorithms can be parallelized straight-forwardly. The Hessian-free (HF) algorithm [7] is a second-order batch algorithm which has been specifically designed for the training of deep neural networks. The advantage of HF over LBFGS is that it directly uses a full second-order model. In a recent work [8], Kingsbury et al. applied the HF algorithm to a training of a DNN acoustic model according to a sequence criterion. In comparison to SGD, Kingsbury et al. reported a speedup by a factor of 5.5 by parallelizing HF. In addition, they achieved a relative word error rate (WER) reduction by roughly four percent.

Deep Architectures can be divided into three groups, generative, discriminative, or hybrid models [11]. Hybrid architectures have been successfully used in speech recognition and natural language processing [10, 22, 23]. In hybrid architectures, the goal is discrimination, which is assisted with the outcomes of generative architectures via better optimization or/and regularization, combining the advantages of generative and discriminative models. The learning of the hybrid model parameters could be done in three ways: staged [24]; iterative [25]; joint [26]. The staged learning allows scalable learning of the model parameters.

3.g. Deep Autoencoder

Deep auto-encoder is a special type of DNN whose output is the data input itself, and is used for learning efficient encoding or dimensionality reduction for a set of data. More specifically, it is a nonlinear feature extraction method involving no class labels; hence generative. An auto-encoder uses three or more layers in the neural network:

- An input layer of data to be efficiently coded (e.g., pixels in image or spectra in speech); One or more considerably smaller hidden layers, which will form the encoding.
- An output layer, where each neuron has the same meaning as in the input layer.

When the number of hidden layers is greater than one, the auto-encoder is considered to be deep. An auto-encoder is often trained using one of the many back-propagation variants (e.g., conjugate gradient method, steepest descent, etc.) Though often reasonably effective, there are fundamental problems with using back-propagation to train networks with many hidden layers. Once the errors get back-propagated to the first few layers, they become minuscule, and quite ineffective. This causes the network to almost always learn to reconstruct the average of all the training data. Though more advanced back-propagation methods (e.g., the conjugate gradient method) help with this to some degree, it still results in very slow learning and poor solutions. This problem is remedied by using initial weights that approximate the final solution. The process to find these initial weights is often called pre-training. A successful prê-training technique developed in (Hinton et al., 2006) for training deep auto-encoders involves treating each neighboring set of two layers like an RBM for pre-training to approximate a good solution and then using a back-propagation technique to fine-tune so as the minimize the "coding" error. This training technique is applied to construct a deep auto-encoder to map images to short binary code for fast, content-based image retrieval. It is also applied to coding documents (called semantic hashing), and to coding spectrogram-like speech features which we review below.

3.h. Deep Stacking Network

The Deep Stacking Network (DSN) is a scalable deep architecture amenable to parallel weight learning [1]. It is trained in a supervised, block-wise fashion, without the need for back-propagation over all blocks as is common in other popular deep architectures [13]. The DSN blocks, each consisting of a simple, easy-to-learn module, are stacked to form the overall deep network. Deep Stacking Network (DSN), which attacks the learning scalability problem. The central idea of DSN design relates to the concept of stacking, as proposed originally in (Wolpert, 1992), where simple modules of functions or classifiers are composed first and then they are "stacked" on top of each other in order to learn complex functions or classifiers.

Various ways of implementing stacking operations have been developed in the past, typically making use of supervised information in the simple modules. The new features for the stacked classifier at a higher level of the stacking architecture often come from concatenation of the classifier output of a lower module and the raw input features. In (Cohen and de Carvalho, 2005), the simple module used for stacking was a conditional random field (CRF). This type of deep architecture was further developed with hidden states added for successful natural language and speech recognition applications where segmentation information in unknown in the training data (Yu et al., 2010a). The DSN discussed in this section makes use of supervision information for stacking each of the basic modules, which takes the simplified form of multilayer perceptron. In the basic module, the output units are linear and the hidden units are sigmoidal nonlinear.
The linearity in the output units permits highly efficient, parallelizable, and closed-form estimation (a result of convex optimization) for the output network weights given the hidden units' activities. It also points to the importance of the closed-form constraints, derived from the convexity, between the input and output weights. Such constraints make the learning the remaining network parameters (i.e., the input network weights) much easier than otherwise, enabling batch-mode learning of DSN that can be distributed over CPU clusters. And in more recent publications, DSN was used when the key operation of stacking is emphasized.

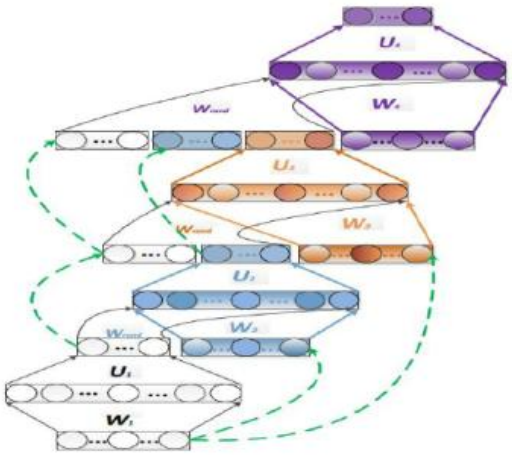


Fig 5. Deep Stacking Network.

IV. CONCLUSION

In this proposed work a brief study of different Deep Learning Algorithm has been learnt. In Modeling Technique and feature representation Deep Learning plays an important role. Speech Signals normally vary in Nature and so there are many challenges faced by the speech processing task. In identifying the phoneme recognition the Deep Learning found to be more efficient and effective. DBN's and CNN's are appreciable and found to be prevalent in solving the mixed noise problem. Based on the intended use of these architectures it can be classified as generative, discriminative and hybrid classes. In our survey the DBN seem to be more flexible as it have many layers according to the input need and the top layers are not given any direction. This makes it to be efficient in online speech Recognition. The main challenges faced by the deep architecture is scalability, complexity of network structure and dealing with more number of features

REFERENCES

[1] Li Deng and Dong Yu, " Deep Learning Methods and Applications"."Signal Processing Magazine, IEEE, 28.1(2011):145-154.
[2] L.R.Rabiner and B.Juang. Fundamentals of Speech Recogntion", Prentice Hall, New Jersy, 1993.
[3] Razvan Pascanu1, Caglar Gulcehre1, Kyunghyun Cho2, and Yoshua Bengio1, " arXiv:1312.6026v5 [cs.NE] 24 Apr 2014
[4] Renu Karule, M. A. Potey, "Deep Architectures for Speech Processing: Survey", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 12, December 2015
[5] Feng Liu 1, Bingquan Liu 1;* , Chengjie Sun 1, Ming Liu 1 and Xiaolong Wang 1;2, "Deep Belief Network-Based Approaches for Link Prediction in Signed Social Networks ", Entropy 2015, 17, 2140-2169; doi:10.3390/e17042140
[6] Geofrey Hinton. To Recognize Shapes , First Learn to Generate Images. 2006.
[7] Geofrey E Hinton. A Fast Learning Algorithm for Deep Belief Nets. 1554:1527{1554, 2006.
[8] Yoshua Bengio and Yann Lecun. Scaling Learning Algorithms towards AI To appear in Large-Scale Kernel Machines ,. (1):1{41, 2007.
[9] Yoshua Bengio. Learning Deep Architectures for AI, volume 2. 2009.
[10] Fabien Lauer. Incorporating Prior Knowledge in Support Vector Machines for Classi- fication : a Review. 9(April 2007):1578{1594, 2008.
[11] Y LeCun and Y Bengio. Convolutional networks for images, speech, and time series. handbook of brain theory and neural networks, 1995.
[12] Itamar Arel, Derek C Rose, and Thomas P Karnowski. Deep Machine Learning A New Frontier in Arti_cial Intelligence Research. (November):13{18, 2010.
[13] Geo_rey Hinton. To Recognize Shapes , First Learn to Generate Images. 2006.
[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D Convolutional Neural Networks for Human Action Recognition. IEEE transactions on pattern analysis and machine intelligence, March 2012.
[15] Quoc V Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Koh, and An-drew Y Ng. Tiled convolutional neural networks. pages 1.
[16] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.
[17] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *NeuralComputation*, vol. 18, pp. 1527–1554, 2006

