ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

An Evaluation of Deep Learning Miniature Concerning in Soft Computing

Dr. Yusuf Perwej

M.Tech, Ph.D (Computer Science & Engg.) Assistant Professor, Department of Computer Science & Engineering,

Al Baha University, Al Baha, Kingdom of Saudi Arabia (KSA)

Abstract: In recent years, Deep Learning at the latest developed field belonging to soft computing. The Deep learning has been a hot topic in the communities of artificial intelligence, artificial neural networks and machine learning. It tries to mimic the human brain, which is capable of processing the intricate input data, learning various knowledge's intellectually and intense as well as solving sundry kinds of sophisticated tasks well. The deep learning paradigm tackles problems in which shallow architectures (e.g. SVM) are impressed with the curse of dimensionality. Deep architectures are composed of multiple levels of non-linear operations, such as in neural nets with many hidden layers or in complicated propositional formulae re-using many sub-formulae. The transformation these characteristics of the human brain to a learning model, we wish the model can deal with the high-dimensional data, support an intense and intellectual learning algorithm and perform well in the inextricable artificial intelligence, real tasks, such as pattern recognition speech recognition, image classification, computer vision and natural language processing. In this paper, we are discussing the history of deep learning, Deep Learning Architectures, abridge the components of Deep Boltzmann Machines (DBM), Deep Stacking Networks (DSN), Compound Hierarchical Deep Models(CHDM), Deep Convolutional Neural Network (DCNN) and Deep Belief Network (DBN) their learning algorithms.

Keywords: Soft Computing, Support Vector Machine (SVM), Artificial Intelligence (AI), Deep Boltzmann Machines (DBM), Compound Hierarchical Deep Models (CHDM), Deep Learning.

I. INTRODUCTION

The Deep learning is a topic in the field of Soft Computing and is a relatively new research area, although based on the popular artificial intelligence, artificial neural networks and other (supposedly mirroring brain function). Deep learning is an emerging technology [1]. Since 2006, deep learning algorithms which rely on deep architectures with several layers of increasingly complex representations have been able to outperform state-of-the-art methods in various settings. Deep architectures can be very efficient in terms of the number of parameters required to represent complex operations, which makes them very appealing to achieve good generalization with small amounts of data. Artificial intelligence is the simulation [2] of human intelligence processes by machines, especially computer systems [3]. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction [4]. The later neural network with multiple hidden layers can learn more complicated functions, but it lacks a good learning algorithm. The appearance of SVM considers people within a short time since it facilitates the learning procedures and performs well in many practical problems, but SVM also encounters its obstruction due to its shallow architectures. Deep architectures are designed by opposition with shallow architectures which have only one layer of hidden variables. In the supervised setting, this hidden layer is typically followed by a linear layer to produce the output [5]. Many shallow architectures such as Gaussian mixtures, RBMs and neural networks with one non-linear hidden layer are universal approximators, thus they can represent any function in theory. In this paper, we are discussing the deep learning in the nowadays. Finally, deep learning researchers have been advised by neuroscientists to seriously consider a broader set of issues and learning architectures so as to [6] gain insight into biologically plausible representations in the brain that may be useful for practical applications Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text.

II. THE HISTORY OF DEEP LEARNING

The brain also appears to process information through multiple stages of transformation and representation. This is particularly clear in the primate visual system (Serre et al., 2007), with its sequence of processing stages, detection of edges, primitive shapes, and moving up to gradually more complex visual shapes. The deep learning has a long history, and its basic concept is originated from artificial neural network research [2]. The feed-forward neural networks with many hidden layers are indeed a good example of the models with a deep architecture. Back-propagation, popularized in 1980's, has been a well-known algorithm [5] for learning the weights of these networks. Inspired by the architectural depth of the brain, neural network researchers had wanted for decades to train deep, multi-layer neural networks (Utgoff & Stracuzzi, 2002; Bengio & LeCun, 2007), but no successful attempts were reported before 2006 researchers [7] reported positive experimental results with typically two or three levels (i.e. one or two hidden layers), but training deeper networks consistently yielded fortuneless results. Something that can be considered a breakthrough happened in 2006. The Hinton and collaborators at introducing Deep Belief Networks or DBNs for short

DOI 10.17148/IJARCCE.2015.4203

(Hinton, Osindero, & Teh, 2006), with a learning algorithm that greedily trains, one layer at a time, exploiting an [8] unsupervised learning algorithm for each layer, a Restricted Boltzmann Machine (RBM) (Freund & Haussler, 1994). Since 2006, deep networks have been applied with success not only on classification tasks, but also in regression, dimensionality reduction, modeling, textures, modeling motion, object segmentation, information retrieval, robotics, natural language processing, and collaborative filtering.

III. THE DEEP LEARNING ACHITECTURES

Today scenario several deep learning architectures such as deep neural networks, convolutional deep neural networks, and deep belief networks have been applied to fields like computer vision, pattern recognition, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks [9].

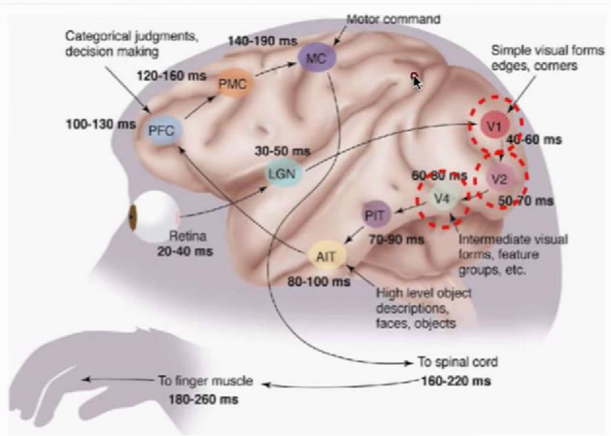

Figure 1. The Typical Human Deep Learning Architectures

The question arises, Why go deep Learning? In figure 1 because the Deep architectures are representationally efficient, fewer computational units for the same function. Allow for showing a hierarchy and non-local generalization finally easier to monitor what is being learned and guide the machine. The deep architectures are designed by antagonism with shallow architectures which have only one layer of hidden variables. In the supervised [10] setting, this hidden layer is typically followed by a linear layer to produce the output. Numerous shallow architectures such as Gaussian mixtures, RBM and neural networks with one non-linear hidden layer are universal approximators, thus they can represent any function in theory. However, there is an important restriction, namely they can only approximate any function given a sufficient number of hidden variables.

The assumption becomes unrealistic with highly differing functions as the number of [11] parameters needed can scale exponentially in terms of the input dimension, a typical example of the curse of dimensionality. The deep architectures which allow for more layers, can lead to much more accomplished representations while still being universal approximators. Because they may require less parameter [12], deep architectures have the potential to both improve generalization and reduce computational costs. In deep architectures seem to learn very interesting representations which can be invariant to several transformations of the input such as translations, and rotations. These representations are [13] almost always distributed (which allows for non-local generalization), and infrequent.

IV. THE DEEP BOLTZMANN MACHINES (DBM)

A Boltzmann machine is a network of equationally connected, neuron-like units that make stochastic decisions about whether to be on or off. Boltzmann machines have a simple learning algorithm (Hinton & Sejnowski, 1983) that allows them to discover interesting features that represent complex regularities in the training data. They were one of the first examples of a neural network capable of learning internal representations, [14] and are able to represent and (given sufficient time) solve cumbersome combinatoric problems. Boltzmann machines are used to solve two quite different computational problems. For a search problem, the weights on the connections are fixed and are used to represent a cost function. The stochastic dynamics of a Boltzmann machine, then allow it to sample binary state vectors that have low values of the cost function.

For a learning problem, the Boltzmann machine is a set of binary data vectors and it must learn to generate these vectors with exalted probability [15]. To do this, it must find weights on the connections so that, relative to other possible binary vectors, the data vectors have low values of the cost function. To solve a learning problem, Boltzmann machines make many small updates to their weights, and each update requires them to solve many various search problems. A Deep Boltzmann Machine (DBM) is a type of binary pair wise Markov random field (undirected probabilistic graphical models) with multiple layers of hidden random variables [16]. It is a network of symmetrically coupled stochastic binary units. It comprises a set of visible units $v \in \{0, 1\}^{D}$ and a series of layers of hidden units $h^{(1)} \in \{0, 1\}^{F_1}$, $h^{(2)} \in \{0, 1\}^{F_2}$, $h^{(L)} \in \{0, 1\}^{F_L}$. In figure 2, there is no connection between the units of the same layer. For the DBM of the figure 2, we can write the probability which is assigned to vector v as:

$$p(v) = \frac{1}{Z}\sum_{h} e^{\sum_{ij} W_{ij}^{(1)} v_i h_j^{1} + \sum_{jl} W_{jl}^{(2)} h_j^{(1)} h_l^{(2)} + \sum_{lm} W_{lm}^{(3)} h_l^{(2)} h_m^{(3)}}$$

where $h = \{ h^{(1)}, h^{(2)}, h^{(3)} \}$ are the set of hidden units, and $\theta = \{ w^{(1)}, w^{(2)}, w^{(3)} \}$ are the model parameters, representing visible-hidden and hidden-hidden symmetric interaction, since they are undirected links terms.

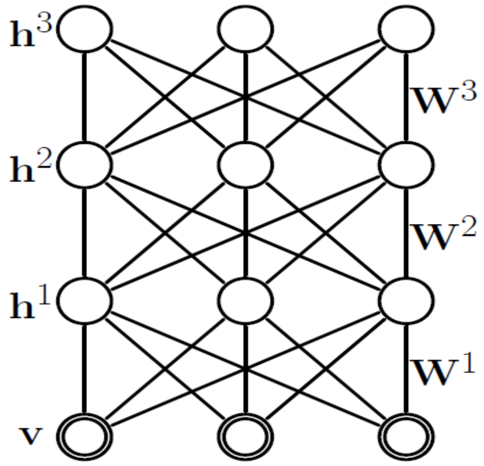

Figure 2. A Deep Boltzmann Machine

There many reasons which motivate us to take advantage of deep Boltzmann machine architectures. The deep convolutional neural networks, they adopt the inference and training procedure in both directions, bottom-up and top-down pass, which enable the deep Boltzmann machine to better unveil the representations of the ambiguous and complex input structures. Another important advantage of deep Boltzmann machine is the joint optimization of all layers using the approximate gradient of a variational lower bound on the likelihood function which influence greatly on the more proper learning of generative models. The deep Boltzmann machine which is an undirected graphical model, but the lower layers form a directed generative model. A greedy layer-wise [17] unsupervised learning algorithm was introduced in [Hinton et al., 2006].

Apart from all the advantages of DBMs discussed so far, now the issue of learning algorithm is very slow in networks with many layers of feature detectors, but it is fast in "restricted Boltzmann machines" that have a single layer of feature detectors. Many hidden layers can be learned efficiently by composing restricted Boltzmann machines, using the feature activations of one as the training data for the next.

V. THE DEEP STACKING NETWORKS (DSN)

The Deep Stacking Network (DSN) is a special type of deep architecture developed to enable and benefit from parallel learning of its model parameters on large CPU clusters. The central idea of the 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 [18]. The deep stacking network architecture was originally presented in (Deng and Yu, 2011) and was referred as a deep convex network or DCN to emphasize the convex nature of a major portion of the algorithm used for learning the network. The deep stacking network 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 unit's permits highly efficient, parallelizable, and closed-form estimation for the output network weights given the hidden units commotion [19].

A deep stacking network, as shown in figure 3 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. In figure 3, only four such modules are illustrated, where each module is shown with a separate color. The lowest module in the deep stacking network comprises a linear layer with a set of linear input units, a hidden non-linear layer with a set of non-linear units, and a second linear layer with a set of linear output units. The lower-layer weight matrix, which we denote by *W*, connects the linear input layer and the hidden nonlinear layer. The upper-layer weight matrix, which we denote by *U*, connects the nonlinear hidden layer with the linear output layer. The weight matrix *U* can be determined through a closed-form solution given the weight matrix *W* when the mean square error training criterion is used.

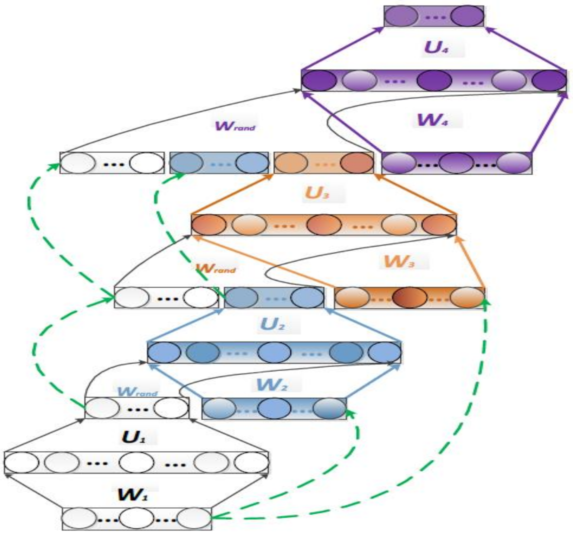

Figure 3. A Deep Stacking Network Architecture

In this paper, we are discussing some technical description on how the use of linear output units in the DSN facilitates the learning of the DSN weights. A single module is used to illustrate the advantage for [20] simplicity reasons. First, it is clear that the upper layer weight matrix *U* can be efficiently learned once the activity matrix *H* over all training samples in the hidden layer is known. Let's denote the training vectors by $\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_i,\cdots,\boldsymbol{x}_N]$, in which each vector is denoted by $\boldsymbol{x}_i=[x_{1i},\cdots,x_{ji},\cdots,x_{Di}]^T$ where D is the dimension of the input vector, which is a function of the block, and N is the total number of training samples. Denote by L the number of hidden units and by C the dimension of the output vector. Then the output of a DSN block is $\boldsymbol{y}_i=\boldsymbol{U}^T\boldsymbol{h}_i$, where $\boldsymbol{h}_i=\sigma(\boldsymbol{W}^T\boldsymbol{x}_i)$ is the hidden-layer vector for sample *i*, $\boldsymbol{U}$ is an $L\times C$ weight matrix at the upper layer of a block. $\boldsymbol{W}$ is a $D\times L$ weight matrix at the lower layer of a block, and σ (·) is a sigmoid function. Bias terms are implicitly represented in the above formulation if $\boldsymbol{x}_i$ and $\boldsymbol{h}_i$ are augmented with ones. Given target vectors in the full training set with a total of N samples, $\boldsymbol{T}=[\boldsymbol{t}_1,\cdots,\boldsymbol{t}_i,\cdots,\boldsymbol{t}_N]$, where each vector is $\boldsymbol{t}_i=[t_{1i},\cdots,t_{ji},\cdots,t_{Ci}]^T$, the parameters $\boldsymbol{U}$ and $\boldsymbol{W}$ are learned so as to minimize the average of the total square error below.

$$\mathrm{E} = \frac{1}{2}\sum_i ||\boldsymbol{y}_i - \boldsymbol{t}_i||^2 = \frac{1}{2}\mathrm{Tr}[(\mathbf{Y}-\mathbf{T})(\mathbf{Y}-\mathbf{T})^T]$$

Where the output of the network is

$$y_i = U^T h_i = U^T\sigma(W^T x_i) = G_i(U, W)$$

This depends on both weight matrices, as in the standard neural net. Assuming $\boldsymbol{H}=[\boldsymbol{h}_1,\cdots,\boldsymbol{h}_i,\cdots,\boldsymbol{h}_N]$ is known, or equivalently, $\boldsymbol{W}$ is known. Then, setting the error derivative with respective to U to zero gives

$$U = (HH^T)^{-1}HT^T = \mathrm{F}(W), \text{where } h_i = \sigma(W^T x_i)$$

This provides an explicit constraint between $\boldsymbol{U}$ and $\boldsymbol{W}$which were treated independently in the conventional backpropagation algorithm. Now, given the equality constraint $\boldsymbol{U}=$ F $(\boldsymbol{W})$, let's use Lagrangian multiplier method to solve the optimization problem in learning $\boldsymbol{W}$. Optimizing the Lagrangian:

$$E = \frac{1}{2}\sum_i ||G_i(U, W) - t_i||^2 + \lambda\,||\mathrm{U} - \mathrm{F}(W)||$$

We can derive batch-mode gradient descent learning algorithm where the gradient takes the following form

$$\frac{\partial E}{\partial W} = 2X\left[H^T \circ (1-H)^T \circ [H^\dagger(HT^T)(TH^\dagger) - T^T(TH^\dagger)]\right]$$

Where $\boldsymbol{H}\dagger=\boldsymbol{H}T(\boldsymbol{H}\boldsymbol{H}T)-\mathbf{1}$ is pseudo-inverse of $\boldsymbol{H}$ and symbol ∘ denotes element-wise multiplication.

VI. THE COMPOUND HIERARCHICAL DEEP MODELS (CHDM)

For a human brain in comparison with the current state-of-the-art artificial systems, fewer numbers of examples is needed to categorize and even extend the already existing categories for the novel instances (generalization). This is the main motivation of this subsection, by means of learning the abstract knowledge of the data and uses them for novel cases in the future. We call our architectures compound Hierarchical-Deep models, [21] because they are derived by composing hierarchical nonparametric Bayesian models with deep networks.

In this paper, we rediscover compound hierarchical-deep architectures that integrate these deep models with structured hierarchical Bayesian models. We show how we can learn a hierarchical Dirichlet process (HDP) prior over the activities of the top-level features in a Deep Boltzmann Machine (DBM), coming to represent both a layered hierarchy of increasingly abstract features, and a tree-structured hierarchy of classes [22]. Our model depends minimally on domain specific representations and achieves state-of-the-art one-shot learning performance by the unsupervised discovery of three component firstly low-level features that abstract from the raw high-dimensional sensory input (e.g. pixels, 3D joint angles) secondly high-level part-like features that express the distinctive perceptual structure of a specific class, in terms of class-specific correlations over low-level features and thirdly a hierarchy of super-classes for sharing abstract knowledge among related classes.

It is a full generative model, generalized from abstract concepts owing through the layers of the model, which is able to synthesize new examples in novel classes that look reasonably natural. Note that all the [23] levels are learned jointly by maximizing a joint log-probability score. Consider a DBM with three hidden layers, the probability of a visible input v is

$$P(v;\psi) = \frac{1}{Z}\sum_h exp\left(\sum_{ij} W_{ij}^{(1)} v_i h_j^1 + \sum_{jl} W_{jl}^{(2)} h_j^1 h_l^2 + \sum_{lm} W_{lm}^{(2)} h_l^2 h_m^3\right)$$

Where h = {h^1, h^2, h^3} are the set of hidden units, and ψ = {$W^{(1)}$, $W^{(2)}$, $W^{(3)}$} are the model parameters, representing visible-hidden and hidden-hidden symmetric interaction terms. After a DBM model has been learned, we have an undirected model that defines the joint distribution P (v, h^1, h^2, h^3). One way to express what has been learned is the conditional model P (v, h1, h2|h^3) and a prior term P (h^3). We can therefore rewrite the variational bound as.

$$logP(v) \geq \sum_{h^1,h^2,h^3} Q(h|v;\mu)logP(v,h^1,h^2|h^3) + \mathcal{H}(Q) + \sum_{h^3} Q(h^3|v;\mu)logP(h^3)$$

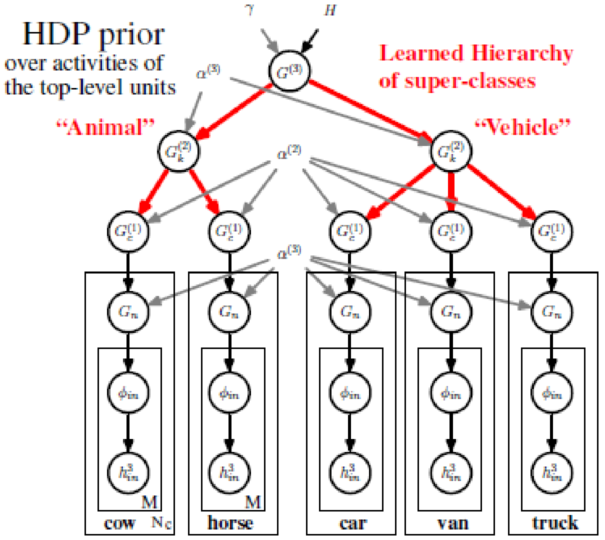

Figure 4. The Hierarchical Dirichlet Process Prior Over the States of h^3

DOI 10.17148/IJARCCE.2015.4203

This distinctive decomposition lies at the core of the greedy recursive pre-training algorithm. We keep the learned conditional model P (v, h1, h2|h^3), [24] but maximize the variational lower-bound with respect to the last term.

Instead of adding an additional undirected layer, (e.g. a restricted Boltzmann machine), to model P (h^3), we can place a hierarchical Dirichlet process prior over h^3, that will allow us to learn category hierarchies, and more importantly, useful representations of classes that contain few training examples.

The part we keep, P(v,h^1,h^2|h^3), represents a conditional DBM model, which can be viewed as a two-layer DBM but with bias terms given by the states of h^3.

$$P(v,h^1,h^2|h^3)=\frac{1}{Z(\psi,h^3)}exp(\sum_{ij}W_{ij}^{(1)}v_ih_j^1\sum_{jl}W_{jl}^{(2)}h_j^1h_l^2\sum_{lm}W_{lm}^{(3)}h_l^2h_m^3)$$

VII. THE DEEP CONVOLUTIONAL NEURAL NETWORK (DCNN)

The Deep convolutional neural networks are a class of models that forms an influential tool to [25] help solve visual classification problems. Convolutional networks are deep neural networks, notably adapted to vision tasks, which improve generalization and decrease computational costs by reducing the number of parameters [26]. The number of parameters is reduced by constraining many weights to [27] share the same values. The model has a succession C$^{(1)}$,P$^{(1)}$,C$^{(2)}$ P$^{(2)}$, ... ,C$^{(M)}$ P$^{(M)}$ of convolutional (C) layers and pooling (P) layers, inspired by the simple/complex cell organization of the visual cortex (Fukushima, 1980) [28].

The connections are feedforward such that all the inputs of a convolutional layer are in the preceding pooling layer, and all the inputs of a pooling layer are in the preceding convolutional layer. Each convolutional layer C$^{(m)}$ is composed of several feature planes c$_1^{(m)}$, c$_2^{(m)}$,…………………,c$_k^{(m)}$ which compute convolutions with the input x and a set of small features

f$_1^{(m)}$, f$_2^{(m)}$,…………………,f$_k^{(m)}$ i.e. computing c$_k^{(m)}$ = x* f$_k^{(m)}$. See figure 5 for a visual representation of convolutional layers. Every pooling layer P$^{(m)}$ corresponds to an averaging/sub-sampling operation [28] of the convolutional layer C$^{(m)}$ where all weights are shared, event within a single neuron.

A quintessential pooling operation reduces an image size by a factor of 2 along all directions by taking the averages of regions of 4*4 pixels which do not overlap. Every pooling plane can then be used as input for a new convolutional layer. See figure 5 for a visual representation of pooling layers.

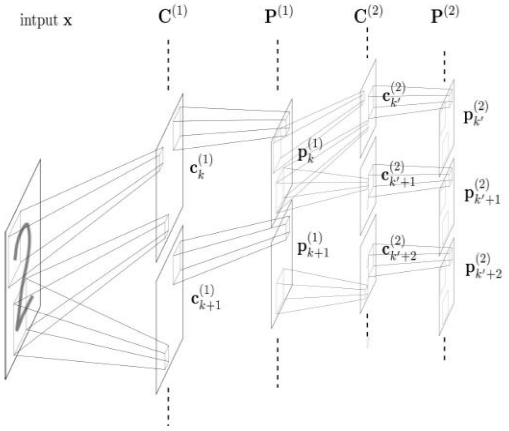

Figure 5. The Convolutional and Pooling Layers of a Convolutional Network

Convolutional networks can be applied in any setting where the input is expected to be invariant by translation. For instance, convolutional networks have been very victorious on the classification of [29] handwritten digits, object recognition, pedestrian detection, speech and time series, and Natural Language Processing. In addition, due to the reduced number of parameters, the optimization takes place in a search space of smaller dimension which is expected to speed up the learning procedure.

VIII. THE DEEP BELIEF NETWORK (DBN)

Deep Belief Networks are probabilistic models that are usually trained in an unsupervised [30], greedy manner. DBNs have proven to be powerful and flexible models. Deep Belief Networks (DBNs) are probably amongst the most famous and basic kind of deep neural network architectures. This is a generative probabilistic model with one visible layer at the bottom and many hidden layers up to the output. Each hidden layer unit learns the statistical representation via the links to the lower layers [31]. Deep Belief Networks are a hybrid model consisting of two parts. As figure 6 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) [32]. 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 or a generative model.

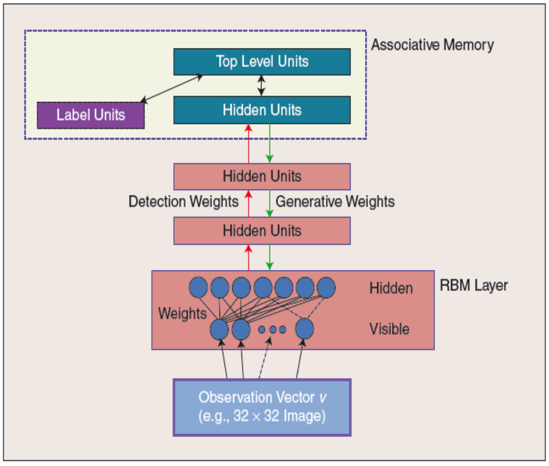

Figure 6. The Illustration of the Deep Belief Network framework

DOI 10.17148/IJARCCE.2015.4203

In figure 6, the discriminating model is a “bottoms-up" procedure (red arrow), trying to [33] 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 greatest advantage of DBNs is its 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 [34] and the higher-level features are believed to be more tangled and better reflects the information contained in the input data structures.

IX. CONCLUSION

This paper is aimed to provide an overview of general deep learning methodology. Deep structured learning, or more commonly called deep learning or hierarchical learning, have emerged as a new area of Soft Computing research. During the past several years, the techniques developed from deep learning, research have already been impacted a wide range of signal and information processing work within the traditional and the new, widened scopes including key aspects of artificial neural networks, machine learning and artificial intelligence. The deep learning algorithms which rely on deep architectures with various layers of increasingly complex representations have been able to outperform state-of-the-art methods in several settings. Deep architectures can be very efficient in terms of the number of parameters required to represent complex operations, which makes them very appealing to achieve good generalization with small amounts of data. Albeit training deep architectures has traditionally been considered a difficult problem, a successful approach has been to employ an unsupervised layer-wise pre-training step to initialize deep supervised models. In this paper, we are discussing the Deep Boltzmann Machines (DBM),

Deep Stacking Networks (DSN), Compound Hierarchical Deep Models(CHDM), Deep Convolutional Neural Network (DCNN) and Deep Belief Network (DBN) their learning algorithms. Finally, the Deep learning methods achieve very glorious accuracy, often the best one, for tasks where a large set of data is available, even if only a small number of instances are labeled.

REFERENCES

[1] Yoshua Bengio and Xavier Glorot. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of AISTATS 2010, volume 9, pages 249–256, 2010.

[2] Jonathan Laserson. From Neural Networks to Deep Learning. XRDS: Crossroads, The ACM Magazine for Students, 18(1):29, September 2011. ISSN 15284972. doi: 10.1145/2000775.2000787.

[3] Dr. Yusuf Perwej, “The Bidirectional Long-Short-Term Memory Neural Network based Word Retrieval for Arabic Documents” Transactions on Machine Learning and Artificial Intelligence (TMLAI), Society for Science and Education, Manchester, United Kingdom (UK), Vol. 03, No.01, Pages 16 – 27, February 2015, ISSN 2054 - 7390, DOI : 10.14738/tmlai.31.863

[4] Yann Lecun and Marc'Aurelio Ranzato. Deep Learning Tutorial, ICML, Atlanta, 2013..

[5] Li Deng. Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey. research. microsoft.com, 2013.

[6] David Marr. Vision: A computational investigation into the human representation and processing of visual information, Henry Holt & Co. Inc., New York, NY, 1982.

[7] Yann Lecun and Marc'Aurelio Ranzato. Deep Learning Tutorial, ICML, Atlanta, 2013.

[8] G. E. Hinton, S. Osindero and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, nr. 7, pp. 1527-1554, 2006.

[9] Yoshua Bengio. Learning deep architectures for AI. Foundations and trends in Machine Learning, 2(1):1{127, 2009. ISSN 1935-8237.

[10] Li Deng, Dong Yu, and John Platt. Scalable stacking and learning for building deep architectures. 2012 IEEE International Conference on Acoustics, Speech and Sig- nal Processing (ICASSP), pages 2133{2136, March 2012.

[11] Collobert, R. and Weston J. “A unified architecture for natural language processing: Deep neural networks with multitask learning,” Proc. ICML, 2008.

[12] Deng, L., Yu, D., and Platt, J. “Scalable stacking and learning for building deep architectures,” Proc. ICASSP, 2012a.

[13] Lena, P., Nagata, K., and Baldi, P. “Deep spatiotemporal architectures and learning for protein structure prediction,” Proc. NIPS, 2012.

[14] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. Cognitive Science, 9(1):147–169, 1985.

[15] Paul Smolensky. Information processing in dynamical systems: foundations of harmony theory. In D. Rumelhart and J. McClelland, editors, Parallel Distributed Processing, volume 1, chapter 6, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

[16] Nicolas Le Roux and Yoshua Bengio” Representational power of restricted Boltzmann machines and deep belief networks” Neural Computation, 20:1631–1649, June 2008.

[17] R. R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In Proceedings of the International Conference on Artificial Intelligence and Statistics, volume 12, 2009.

[18] Wolpert, D. “Stacked generalization,” Neural Networks, vol. 5, no. 2, pp. 241-259, 1992.

[19] Deng, L. and Yu, D. “Deep Convex Network: A scalable architecture for speech pattern classification,” Proc. Interspeech, 2011.

[20] Brian Hutchinson, Li Deng, and Dong Yu. Tensor deep stacking networks. IEEE trans- actions on pattern analysis and machine intelligence, pages 1{15, December 2012a. ISSN 1939-3539.

[21] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 28(4):594{611, April 2006.

[22] Ruslan Salakhutdinov, Joshua B Tenenbaum, and Antonio Torralba. Learning with Hierarchical-Deep Models. IEEE transactions on pattern analysis and machine intelligence, pages 1-14, Dec 2012. ISSN 1939-3539.

[23] Charles Kemp, Amy Perfors, and Joshua B Tenenbaum. Learning overhypotheses with hierarchical Bayesian models. Developmental science, 10(3):307{21, May 2007. ISSN 1363-755X.

[24] G. E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527–1554, 2006.

[25] Itamar Arel, Derek Rose, and Robert Coop. Destin: A scalable deep learning architecture with application to high-dimensional robust pattern recognition. Proc. AAAI Workshop on Biologically Inspired, 2009a

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11): 2278–2324, November 1998a.

[27] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In ICML ’08: Proceedings of the 25th international conference on Machine learning, 2008.

[28] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4):193–202, 1980.

[29] Kevin Jarrett, Koray Kavukcuoglu, Marc’Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In Proc. International Conference on Computer Vision (ICCV’09). IEEE, 2009.

[30] Honglak Lee, Peter Pham, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. Pages 1-9, 2009a.

[31] Philippe Hamel and Douglas Eck. Learning Features from Music Audio with Deep Belief Networks. ISMIR, (Ismir): pp 339-344, 2010.

[32] Yoshua Bengio and Y LeCun. Scaling learning algorithms towards AI. Large-Scale Kernel Machines, (1):1{41, 2007.

[33] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm [.

[34] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Proceedings of the 26th Annual International Conference on Machine Learning – ICML '09, pages 1-8, 2009.

BIOGRAPHY

Dr. Yusuf Perwej Assistant Professor in the Department of Computer Science & Engineering Al Baha University, Al Baha , Kingdom of Saudi Arabia (KSA). He has authored a number of different journal and paper. His research interests include Soft Computing, Artificial Neural Network, Machine Learning, Pattern Matching, Pattern Recognition, Artificial Intelligence, Image Processing, Fuzzy Logic, Genetic Algorithm, Robotics, Bluetooth and Network etc. He is a member of IEEE.

DOI 10.17148/IJARCCE.2015.4203