Deep Learning in the era of Big Data

Milind Bhusari, Nagasushma Devarapalli, Chetan Hibare, Pratyusha Joginipally, Saikrishna Nadimpalli, Kaustubh Pimparkar, Sarvesh Sharma, Deeksha Sharma, Mahitha Thokala
Department of Computer Engineering
Santa Clara University
Santa Clara, CA 95053

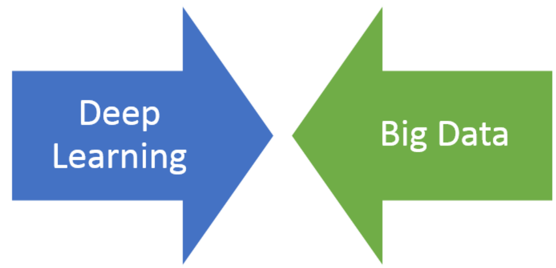
***Abstract*—Deep Learning, a family of iterative algorithms has been gaining ground in the recent days in several applications such as speech recognition, image recognition etc. It's main edge over other approaches of machine learning and artificial intelligence is to learn the representations from unlabeled data. While current machine learning methodologies in Big data rely highly on labeled data for better feature extraction, deep learning comes with its own set of advantages of working on unsupervised and semi-supervised feature learning taking it's roots from neural networks. This makes Deep Learning more attractive and has higher scope for break through. In this paper, we examine the different architectures of deep learning, their features, applications in several fields. Then we examine the machine learning methodologies that are currently dominant in the field of Big data. We also suggest future work that can improve Deep learning in the era of Big data which can be applied in several regular and mission critical applications.**

Keywords – Deep Learning, Big Data, Neural Networks, Machine Learning, Speech Recognition, Image Recognition

I. Introduction

Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. It consists of set of algorithms in machine learning that attempt to model high-level abstractions in data by using model architectures composed of multiple non-linear transformations. Deep learning is part of a broader family of machine learning methods based on learning representations of data. Deep Neural Networks (DNNs) have become popular in the machine learning community over the last few years, showing significant improvements in machine learning methods used in both speech and image processing. The development of pre-training algorithms and better forms of random initialization, as well as the availability of improved hardware, has made it possible to train deeper networks than before, and in practice these deep networks have achieved excellent performance in speech recognition applications.

However, one drawback of DNNs is that training remains very slow, particularly in applications involving a lot of data. Given the increased amount of data currently available, particularly from multimedia sources, the need for algorithms to run on tasks involving huge data is critical. Slow data training can be attributed to a variety of causes. First, models for big-data tasks are often trained with millions to billions of training examples, as using increased amounts of training data often improves DNN performance. Second, large data sets often use



DNNs with a large number of parameters (i.e., can be 10-50 million DNN parameters in many speech tasks). Third, to date the most popular methodology to train DNNs is a first order optimization technique, including techniques like stochastic gradient descent (SGD)–a serial algorithm executed on a multi-core CPU. While second order optimization techniques, which are much easier to parallelize across machines, have been explored for DNN training, these methods are not always faster than training DNNs via SGD.

Recently, increasing attention on massive data and large scale network architectures has driven parallel implementation of deep learning techniques. Locally connected neural networks and convolutional-alike neural networks were successfully paralleled on computer clusters.

Various deep learning architectures such as deep neural networks, convolutional deep neural networks, and deep belief networks have been applied to fields like computer vision, 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. Big data provides great perspective for revolutionizing many aspects of our society. As the data keeps getting bigger, for big data analytics and prediction analysis deep learning plays a major role. However, huge amounts of data presents a challenge for deep learning. New ways of deep learning algorithms, framework needs to come into place for addressing Big Data challenges. Rest of the paper is organized as follows. In section 2, architectural background of Neural Networks and different architectures are discussed briefly; in section 3, Big Data and its System Architecture is discussed; in section 4, Different Learning Approaches are explained; in section 5 two applications where Deep learning is widely used is mentioned in detail; and section 6 is about Challenges and section 6 summarizes the paper.

II. Neural Network Architecture

A. What is a Neural Network

Neural Networks is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The most crucial element is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural Networks just like people, learn by example. A Neural Network is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

B. Why to Choose Neural Network

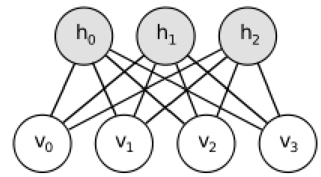
Neural networks, derive meaning from complicated or imprecise data and can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.
Other advantages include:

- *Learning Adaptively*: An ability to learn how to do tasks based on the data given for training or initial experience.
- *Self–Organization*: A Neural Net can create its own organization or representation of the information it receives during learning time.
- *Real Time Operation*: Computations by Neural Network may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- *Fault Tolerance via Redundant Information Coding*: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

The following sub sections present brief overview of well–known deep architectures.

C. Boltzmann Machine

A Boltzmann machine is a network of symmetrically connected, neuron-like units that make stochastic decisions about whether to be on or off. Boltzmann machines have a simple learning algorithm [1] that allows them to discover interesting features that represent complex regularities in the training data. A BM can be used to learn important aspects of an unknown probability distribution based on samples from this distribution.
In networks with many layers of feature detectors the learning is quite slow, but it is fast in Restricted Boltzmann Machines



(RBMs) where there is only single layer of feature detectors.

- These RBMs can be used to learn many hidden layers efficiently by using the feature activation of one of them as the training data for the next machine.
- They are most commonly used to solve two different kind of problems. Firstly the search problem, where in the weights on the connections are fixed and 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. Second, is the learning algorithm.
  The Boltzmann machine is shown as a set of binary data vectors and it must learn to generate these vectors with high probability. It does this by finding weights in the connections so that, data vectors have low values of the cost function relative to other possible binary vectors. In searching problem, Boltzmann machines make small updates to their weights, which requires them to solve many different search problems.
- Restricted Boltzmann Machines: Restricted Boltzmann Machines are a variant of Boltzmann machines wherein their neurons must form a bipartite graph. The two sets of nodes referred commonly as visible and hidden units, have symmetric connections between them but no visible–visible and hidden–hidden connections [2]. The Contrastive Divergence algorithm has mostly been used to train the Boltzmann machine.
- Learning Deep Networks from Restricted Boltzmann Machines: Once the learning on the hidden layer is done, we are left with the activity vectors of the hidden units. Since these hidden layers are being driven by the real data, this activity vectors can be used as data for training another Boltzmann machine. This can be done to learn as many layers as possible. After learning layers in this manner we can view it as a single multilayer generative model. Every new hidden layer added improves the minimum probability that this model will generate on the training data [3].

D. Deep Belief Networks

Deep Belief Networks (DBN) is a deep architecture in which a single RBM is stacked on top of each other taking the output of previous RBM as the input to the next layer. Each layer learns more complex feature than the layer before it. Hinton et al. introduced the deep belief network [4]. Fig. 1

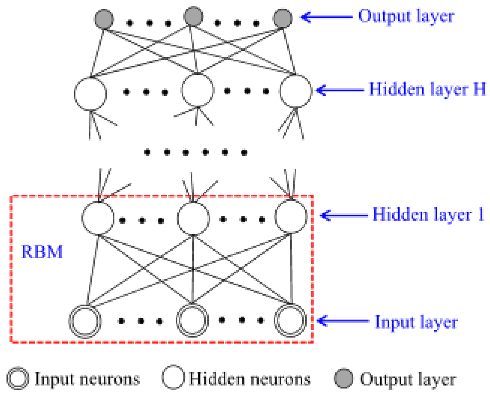


Fig. 1. An illustration of an RBM network

gives an illustration of a DBN with stacked RBMs.

When using unsupervised training, DBN can be used as a feature extraction method for dimensionality reduction. On the other hand, DBN is used for classification when class labels are associated with feature vectors.
There are two general types of DBN classifier architectures:

- Back–Propogation DBN (BP-DBN): In this architecture, a final layer of variables is added which represent the desired outputs (k outputs). A discriminative fine-tuning phase is performed using back-propagation. The use of back-propagation to fine-tune feature detectors that are learned as a generative model works better than when used with random initial weights in the traditional neural network.
- Associate Memory DBN (AMDBN): Associate Memory DBN [5] includes training of the top level of RBN which is obtained by concatenating the high–level representation produced by unsupervised learning with a binary label vector that contains a 1 in the location representing the correct class. The labels are provided as input.

E. Convolutional Neural Networks

The motivation for Convolutional Neural Networks (CNNs) proposal as a deep learning framework is minimal data preprocessing requirements. Convolutional neural networks is a modification of traditional neural network. In CNN, fully connected hidden layers are not used, in place of them CNN has special network structure [6], which has alternating convolution and pooling layers. It often has two types of altering layers namely convolutional and subsampling layers: Convolutional layer implements convolutional operations with various filter maps of same size, whereas subsampling layer diminishes the sizes of proceeding layers by averaging pixels within a limited region.
CNN comes in the category of multi–layer neural networks specially designed for use on two–dimensional data, such as images and videos. CNNs are affected by previous work in time–delay neural networks (TDNN) [7], which minimizes learning computation requirements by sharing weights in a temporal dimension and are meant for speech and time-series processing. With many layers of hierarchy CNNs are the first successfully trained approach in a strong manner.
Convolutional networks are multi–layer architectures where consecutive layers are designed in a such way to learn continuously higher-level features, till the last layer which represents categories. In order to minimize overall loss function, all layers are trained simultaneously. There is no unique feature extractor and classifier in CNN unlike many other models of pattern recognition and classification. All the layers are trained from data in a unified way and are identical in nature [8]. The advantage of using CNN architecture is that, it uses spatial relationships to reduce the number of parameters which must be learned and hence improves upon general feed-forwarding back propagation training.
To summarize, CNN utilizes different strategies like local receptive fields [9], shared weights and subsampling to learn a hierarchical feature representation [10]. Each of the filter banks can be trained with either supervised or unsupervised methods.

Advantages

- A CNN is capable of learning good feature hierarchies automatically and providing some degree of translational and distortional invariances.
- CNNs are an attempt to solve the dilemma between small networks that cannot learn the training set, and large networks that seem over-parameterized.
- CNNs are particularly well suited for recognizing or rejecting shapes with widely varying size, position, and orientation, such as the ones typically produced by heuristic segmenters in real-world string recognition systems.
- One major advantage of convolutional networks is the use of shared weight in convolutional layers, which means that the same filter (weights bank) is used for each pixel in the layer, this both reduces required memory size and improves performance.
- Compared to other image classification algorithms, convolutional neural networks use relatively little pre–processing. This means that the network is responsible for learning the filters that in traditional algorithms were hand–engineered. The lack of a dependence on prior–knowledge and the existence of difficult to design hand-engineered features is a major advantage for CNNs.

III. Big Data: Definition and System Architecture

Big data is an evolving term that describes any voluminous amount of structured, semi–structured and unstructured data that has the potential to be mined for information. Although Big data doesn't refer to any specific quantity, the term is often used when speaking about petabytes and exabytes of data. Let us understand the architecture of Big Data.

A. System Architecure

A Big data system is complex, providing functions to deal with different phases in the digital data life cycle, ranging from its birth to its destruction. At the same time, the system usually involves multiple distinct phases for different applications. Generally, the description for Big data comprises of four stages: generation, acquisition, storage and processing. Layered structure for Big Data system is given below. Decomposed layered architecture of big data system includes infrastructure layer, computing layer, and application layer. This layered view only provides a conceptual hierarchy to underscore the complexity of a big data system. The function of each layer is as follows.

- The infrastructure layer consists of a pool of ICT resources, which can be organized by cloud computing infrastructure and enabled by virtualization technology. New hardware technologies can replace older ones for faster and efficient functioning. These resources will be exposed to upper–layer systems in a fine–grained manner with a specific service–level agreement (SLA). Within this model, resources must be allocated to meet the big data demand while achieving resource efficiency by maximizing system utilization, energy awareness, operational simplification, etc.
- The computing layer encapsulates various data tools into a middleware layer that runs over raw ICT resources. In the context of big data, typical tools include data integration, data management, and the programming model. Data integration means acquiring data from disparate sources and integrating the dataset into a unified form with the necessary data pre–processing operations. Data management refers to mechanisms and tools that provide persistent data storage and highly efficient management, such as distributed file systems and SQL or NoSQL data stores. The programming model implements abstraction application logic and facilitates the data analysis applications.
- The application layer exploits the interface provided by the programming models to implement various data analysis functions, including querying, statistical analyses, clustering, and classification. Then, it combines basic analytical methods to develop various filed related applications.

IV. Learning Approaches

There are basically three types of deep learning approaches based on type of training data provided, order and method by which that data is received and selection of test data on which learning algorithm is implemented.

- Supervised learning: The learner is provided with labeled training data and is allowed to perform machine learning tasks, make predictions for unseen data provided for test. User need to decide kind of data used as training set. Training data is typically a pair of input object, typically a vector and desired output value, a supervisory signal. Structure and order of input data is then decided and given to the learned function. Learned function needs to be evaluated before its use. Performance of learned function should be measured on test data for classification.
- Unsupervised learning: The learner is provided with unlabeled training data and makes prediction for unseen data provided for test. As training data is unlabeled it is difficult to evaluate the quality of data. Dimensionality reduction and clustering are typical examples in unsupervised learning tasks.
- Semi–supervised Learning: The learner is provided with both labeled and unlabeled data and learner makes predictions for all unseen data provided for test. Semi–supervised learning is common in situations where unlabeled data is easily accessible but labels are expensive to obtain. Various types of problems arising in applications, including classification, regression, or ranking tasks, can be framed as instances of semi–supervised learning. Performance of semi–supervised depends on the distribution of unlabeled data compared to that in Supervised learning [11].

A. *Deep Learning for Massive amounts of data*

While deep learning applications are showing impressive results in efficient processing and computation on training data, it is challenging task to perform deep learning on massive amount of data because of iterative computation that are inherent in deep learning algorithms. Considering unprecedented growth in academic and commercial data, it is important to find scalable parallel processing algorithms to train deep models [10] Speeding up the training process with the usage of computing power has shown significant potential in Big Data deep learning. For example, one way to scale up DBNs is to use multiple CPU cores, with each core dealing with a subset of training data (data-parallel schemes).

1) Large–Scale Convolutional Networks: Convolution neural network is a type of locally connected deep learning method. Large–scale CNN is implemented using Graphical processing units comprised of hundreds of parallel processing CPU cores. CNN training involves forward and backward propagation. For forward propagation, more than one blocks are assigned for each feature map depending upon the size of feature map. Single thread is assigned to work on single neuron assigned to feature map. computation of neuron involves convolution of shared weights with neuron in performed in thread level of parallelism. Then computation outputs are stored in global memory. Back–propagation of errors is performed for weight updates to be done in previous layers and error in local units in current layer depends on error of previous layers. Back–propagation is done by pushing and pulling signals though previous layer [12]. Pulling is not straightforward process because of convolutions and subsampling involved in previous layers.
For illustration, consider below Fig. 2 which has single convolution and subsampling and many units having many number of connections with convolutions. Here, our task is to list neurons contributing to error signals of neurons from previous layer. Units have same number of incoming connections, so it's easy to push error signals from layer to layer in back-propagation. For convolution operation, only small portion

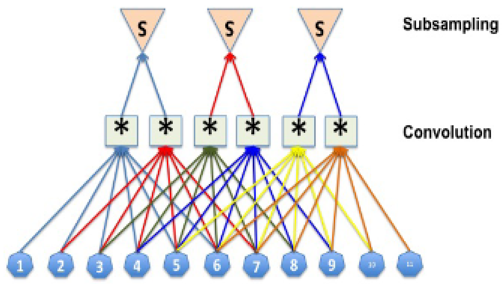


Fig. 2. Illustration of operations performed in 1-D convolution and subsampling

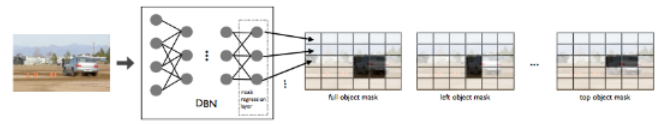


Fig. 3. A schematic view of object detection as DNN–based regression

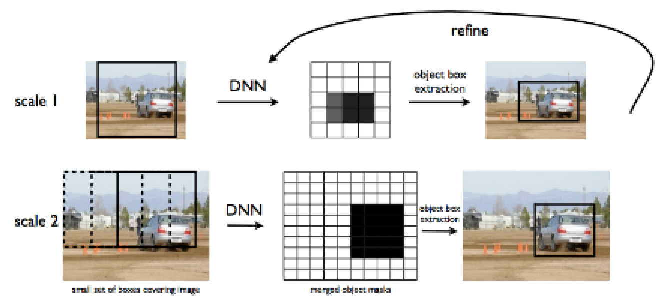


Fig. 4. After regressing to object masks across several scales and large image boxes, we perform object box extraction. The obtained boxes are refined by repeating the same procedure on the sub images, cropped via the current object boxes. For brevity, here only the full object mask is displayed. However, we use all five object masks.

of feature map is store in small amount of shared memory. So, for convolution operation, Scherer et. al. suggested use of shared buffer memory as a circular shared buffer memory. To further improve GPU memory utilization, author implemented modified architecture with both convolution and subsampling operations combines in one step. This allows storage of activities and error values with less usage of shared memory while running back-propagation.

V. Applications of Deep Learning in Big Data

This section focusses on two widely used applications of Deep Learning in Big Data.

A. *Deep Neural Networks in Image Recognition*

During the last few years, video cameras have been installed in a large number of public and private spaces. May be in the future, the camera software will not just identify particular human via facial recognition, but also identify certain types of behavior, perhaps even automatically alerting authorities. On an experiment that was run at Stanford using Neural Networks, where software program learned to observe patterns in the pictures and description, the researchers turned them on previously unseen images. The programs were able to identify objects and actions with roughly double the accuracy of earlier efforts, although still nowhere near human perception capabilities.

Also, Deep Neural Networks (DNNs) has shown fairly decent results on problems of image classification. But the problems in using DNNs are not only limited to classifying the object to a specific class but also accurately localizing these objects. A simple and powerful formulation of object detection is presented as a regression problem to object bounding box masks [13]. As explained in the paper a multi-scale inference procedure is defined that is able to produce high–resolution object detections at a low cost by a few network applications. To have complete image understanding, more precise and detailed object recognition becomes crucial. In this context, in addition to classifying images, one would also care about precisely estimating the class and location of objects within the images, a problem known as object detection.

The properties of Deep Neural Networks are best suited to solve the problems of object detection since they are different from traditional algorithms for classification. They are deep architectures which have the capacity to learn more complex models than shallow ones. This expressivity and robust training algorithms allow for learning powerful object representations without the need to hand design features. This has been empirically demonstrated on the challenging ImageNet classification task across thousands of classes. Deep Neural Networks is used for the problem of object detection in an image, where we not only classify but also try to precisely localize objects.

The approach is based on DNN–based regression model towards an object mask, as shown in Fig. 3 where masks can be generated for the full object as well as portions of the object. A single DNN regression can give masks of multiple objects in an image. To further increase the precision of the localization, the DNN localizer is applied on a small set of large subwindows. The full flow is presented in Fig. 4.

1) Deep Neural Networks Regression: It consists of total 7 layers, the first 5 of which being convolutional and the last 2 fully connected [13]. Each layer uses a rectified linear unit as a non-linear transformation. Three of the convolutional layers have an additional max pooling. Instead of using a softmax classifier as a last layer, we use a regression layer which generates an object binary mask DNN. Since the output of the network has a fixed dimension, a mask of a fixed size N = d x d is predicted. After being resized to the image size, the resulting binary mask represents one or several objects: it should have value 1 at particular pixel if this pixel lies within the bounding box of an object of a given class and 0 otherwise.

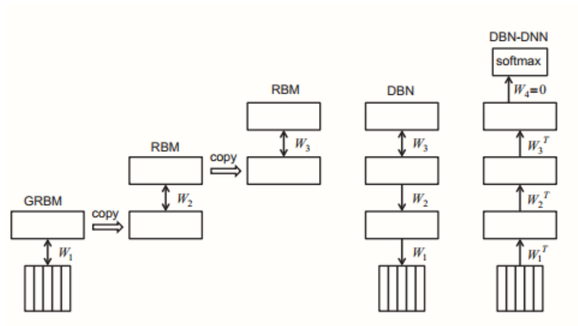
2) Object Localization via Deep Neural Network generated Masks: Although the presented approach is capable of

generating high-quality masks, there are several additional challenges. First, a single object mask might not be sufficient to remove uncertainty of the closely placed objects. Second, due to the limits in the output size, generated masks are much smaller than the original image. For example, for an image of size 400 400 and d = 24, each output would correspond to a cell of size 16 16 which would be insufficient to precisely localize an object, especially if it is a small one. Finally, since we use as an input the full image, small objects will affect very few input neurons and thus will be hard to recognize.
However these issues can be addressed by the following techniques:

- Multiple Masks for Robust Localization: To solve the the problem of objects placed very closely with each other, several masks are generated. Each mask represent either the entire object or partial object.To get the bounding box, one network is used to predict the object box mask and four additional networks to predict another four halves of the box.
  These five predictions help reduce uncertainty and deal with mistakes in some of the masks. Further, if two objects of the same type are placed next to each other, then at least two of the produced five masks would not have the objects merged which would allow to differentiate them. This would make the detection of multiple objects easier.
- Object Localization from DNN Output: To complete the detection process, an estimate of a set of bounding boxes for each image is required. Although the output resolution is smaller than the input image, the binary masks are rescaled to the resolution as the input image.
- Multi-scale Refinement of DNN Localizer: To improve the localization, a second stage of DNN regression called refinement is required. The Deep Neural Network localizer is applied on the windows defined by the initial detection stage — each of the 15 bounding boxes is enlarged by a factor of 1.2 and is applied to the network. Applying the localizer at higher resolution increases the precision of the detections significantly.

B. Deep Neural Network in Speech Recognition

Speech Recognition means the mechanism adopted by the computer to perform identification of Human Speech and accordingly generate appropriate response for it. This system transforms the vocal words into text at the system level. It is widely implemented in latest systems and mentioned with different names like "Automatic Speech Recognition", "Computer Speech Recognition" and "Speech to Text". The system is in place for many applications, however the study in the field clearly shows that the future scope for this system has a long way to go. It can be used in many different applications like security systems, automatic phone call system, user complaints system, ATM machines, etc. However, this system is not completely full proof and could fail in some scenarios. At times there's lots of noise and disturbance in the speech that completely changes the incoming speech signal and the basis on which the system is build fails.
The very ancient and the first model on which the speech recognition was built is Hidden Markov Models (HMM). The input to HMM model is given in form of acoustic sound. HMM, within the system makes use of sound notes and Gaussian Mixture Models in order to identify and understand the way every state of every HMM fits into short window frames that are nothing but the coefficients i.e. the acoustic input. The performance of GMM–HMM [14] model is further improved by handling the error rate for the words recorded on phone and in its sentences. A Discriminative Objective function that is trained to maximize the probability of producing the data that is being observed and it is followed by discriminatively fine tuning it. GMMs have reached great success heights in getting the acoustic modeling accuracyto standards such that it would be a rare case to outperform any new models.
An alternative way to evaluate the fit is to use a feed forward neural network that takes several frames of coefficients as input and produces posterior probabilities over HMM states as output. As mentioned earlier, Deep Neural Networks have an architecture with many hidden layers. The frames are trained with new techniques and the results clearly show that the performance of Deep Neural Networks is much better than Gaussian Mixture Models [15]. The comparison is tested and concluded on the basis of various types of speeches and at times the accuracy margin is too big with DNN clearly dominating the comparison.
Deep Neural Networks is constructed of multiple non–linear hidden units and a huge concluding output layer. The reason for having a large output layer is to show up all the HMM states. These high HMM states [16] are due to the modeling of each phone by various "triphone" HMMs which constitute to build up thousands of tied states. Using the new learning methods, several different research groups have shown that DNNs can outperform GMMs at acoustic modeling for speech recognition on a variety of datasets including large datasets with large vocabularies.
After training an RBM on the data, the inferred states of the hidden units can be used as data for training another RBM that learns to model the significant dependencies between the hidden units of the first RBM. This can be repeated as many times as desired to produce many layers of non–linear feature detectors that represent progressively more complex statistical structure in the data. The RBMs in a stack can be combined in a surprising way to produce a single, multi–layer generative model called a deep belief net (DBN). Even though each RBM is an undirected model, the DBN formed by the whole stack is a hybrid generative model whose top two layers are undirected (they are the final RBM in the stack) but whose lower layers have top–down, directed connections.
The sequence of operations used to create a DBN with three hidden layers and to convert it to a pre-trained DBN–DNN. First a Gaussian–Bernoulli restricted Boltzmann machine



(GRBM) [17] is trained to model a window of frames of real–valued acoustic coefficients. Then the states of the binary hidden units of the GRBM are used as data for training an RBM. This is repeated to create as many hidden layers as desired. Then the stack of RBMs is converted to a single generative model, a DBN, by replacing the undirected connections of the lower level RBMs by top–down, directed connections. Finally, a pre–trained DBN–DNN is created by adding a "softmax" output layer that contains one unit for each possible state of each HMM. The DBN–DNN is then discriminatively trained to predict the HMM state corresponding to the central frame of the input window in a forced alignment.

VI. Challenges

In spite of all the recent achievement in large–scale deep learning with neural networks, this field still is still in budding stage. Much more needs to be done to address many significant challenges posted by Big Data, often characterized by the three V's model: volume, variety, and velocity which refers to large scale of data, different types of data, and the speed of streaming data, respectively.

- Deep learning from high volumes of data
- Deep learning from high variety of data
- Deep learning from high velocity of data

Big Data presents significant challenges to deep learning with neural networks, including large scale, heterogeneity, noisy labels, and non–stationary distribution, among many others. In order to realize the full potential of Big Data, we need to address these technical challenges with new ways of thinking and transformative solutions for neural networks. We believe that these research challenges posed by Big Data are not only timely, but will also bring ample opportunities for deep learning. Together, they will provide major advances in science, medicine, and business.

Main criticism about deep learning is there is lack of theory to understand many of the methods. Theory surrounding some algorithms like contrastive divergence is less clear. Deep learning methods are often looked as a black box because most confirmations are done empirically rather that theoretically.

Main goal of deep learning is of building intelligent machines based on Artificial Intelligence, but mostly it is used as an all–encompassing solution. Despite the power of deep learning methods, they are still lack much of the functionality needed for realizing this goal entirely.

Deep learning presents a unique set of challenges in the audio processing,Music audio signals are time series where events are organized in musical time, rather than in real time, which changes as a function of rhythm and expression. The measured signals typically combine multiple voices that are synchronized in time and overlapping in frequency, mixing both short–term and long–term temporal dependencies. The influencing factors include musical tradition, style, composer and interpretation. The high complexity and variety give rise to the signal representation problems well–suited to the high levels of abstraction afforded by the perceptually and biologically motivated processing techniques of deep learning [18].

VII. Suggested Future work and Conclusion

This paper discussed about Deep learning and its architecture and two prominent applications of it. This paper also elaborates about how deep learning can be used for huge amounts of data and can be applied to Big Data and different challenges that need to be addressed for efficient use of Deep learning in Big Data. Deep learning is advantageous in extracting complex data representations from huge amounts of unsupervised data, which is of great importance in Big Data. Deep learning and its implementation in Big data is still in its infancy, much more research work needs to be done in various areas. For example, using Corpus input to predict user's next input word based on previous visited documents [19] which would dramatically reduce the burden of text input tasks.

References

[1] M. Welling and Y. W. Teh, "Approximate inference in boltzmann machines," *Artificial Intelligence*, vol. 143, no. 1, pp. 19–50, 2003.

[2] D. E. Rumelhart, J. L. McClelland, P. R. Group *et al.*, "Parallel distributed processing: Explorations in the microstructures of cognition. volume 1: Foundations," *MIT Press, Cambridge, MA*, vol. 2, pp. 560–567, 1986.

[3] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[4] G. 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.

[5] M. A. Salama, A. Fahmy *et al.*, "Deep belief network for clustering and classification of a continuous data," in *Signal Processing and Information Technology (ISSPIT), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 473–477.

[6] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 10, pp. 1533–1545, 2014.

[7] 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.

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

[9] Y. Bengio, Y. LeCun *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, 2007.

[10] X. Chen and X. Lin, “Big data deep learning: Challenges and perspectives,” 2014.

[11] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.

[12] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, “Flexible, high performance convolutional neural networks for image classification,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1237.

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

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “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.

[15] M. A. Hossan, S. Memon, and M. A. Gregory, “A novel approach for mfcc feature extraction,” in *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*. IEEE, 2010, pp. 1–5.

[16] I. Patel and Y. S. Rao, “Speech recognition using hidden markov model with mfcc-subband technique,” in *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*. IEEE, 2010, pp. 168–172.

[17] J. Huang and B. Kingsbury, “Audio-visual deep learning for noise robust speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7596–7599.

[18] L. Deng and Y. Dong, “Foundations and trends® in signal processing,” *Signal Processing*, vol. 7, pp. 3–4, 2014.

[19] H. Komatsu, S. Takabayashi, and T. Masui, “Corpus-based predictive text input,” in *Active Media Technology, 2005.(AMT 2005). Proceedings of the 2005 International Conference on*. IEEE, 2005, pp. 75–80.