International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 25433-25448

http://www.ripublication.com

Deep Learning – An Overview

Remya Ramachandran[1], Devi C Rajeev[1], Swathi G Krishnan[1], P Subathra[2]
[1]*BTech Student, Department of Computer Science and Engineering,*
Amrita School of Engineering,Amrita Vishwavidyapeetham (University), Coimbatore, India.
[2]*Assistant Professor,Department of Computer Science and Engineering,*
Amrita School of Engineering, Amrita Vishwavidyapeetham (University), Coimbatore, India.
Email: p_subathra@cb.amrita.edu

Abstract

Deep Learning is a new and emerging field in Machine Learning, developed to model higher level abstraction in data. The goal of Deep Learning is to move towards Artificial Intelligence.It provides semi-supervised or unsupervised feature learning algorithms and hierarchical feature extraction, in place of the traditional handcrafted features. This survey paper is intended to provide an overall understanding of the basic concepts of Deep Learning, by providing answers to the following questions:

- What is Deep Learning?
- What is the importance of Deep Learning?
- How can Deep Learning improve Machine Learning?
- What are the different types of Deep Learning Architectures used?
- What tools are used for its implementation?
- What are its applications?
- How is it suitable for Big Data Analysis?
- What are the challenges it faces?

What Is Deep Learning?

Deep Structured Learning is a part of Machine Learning that takes it a lot closer to Artificial Intelligence, by creating better models and representations to learn from large-scale unlabeled data. It uses architectures that contain layers of neurons whose inputs go through multiple non-linear transformations, unlike those in ANNs, thereby avoiding the diminishing gradient problem.

Deep Learning [1][4] is based on learning representations of data, and is largely inspired by the advances in neuroscience. It attempts to replicate the way a human brain would comprehend data; in a hierarchical fashion with layers of data abstraction. Each layer is trained independently of the higher layers using greedy algorithm [7] and the features extracted from the data by that layer is used as input by the next layer and so on.

To explain the feature abstraction, let us take the example of identifying a house in a picture. This task can be divided into subtasks to identify a door, a few windows and a roof. The identification of a door can be further subdivided into the identification of a door lock and a handle, which can eventually be broken down into the intensity of a pixel in the picture. Now each of these tasks can be done by different neurons at different levels in the network working independently to get the final result. Such is the functioning of deep neural networks.

The following figure shows how deep learning improved over the years and what new deep learning methods came into picture over time.

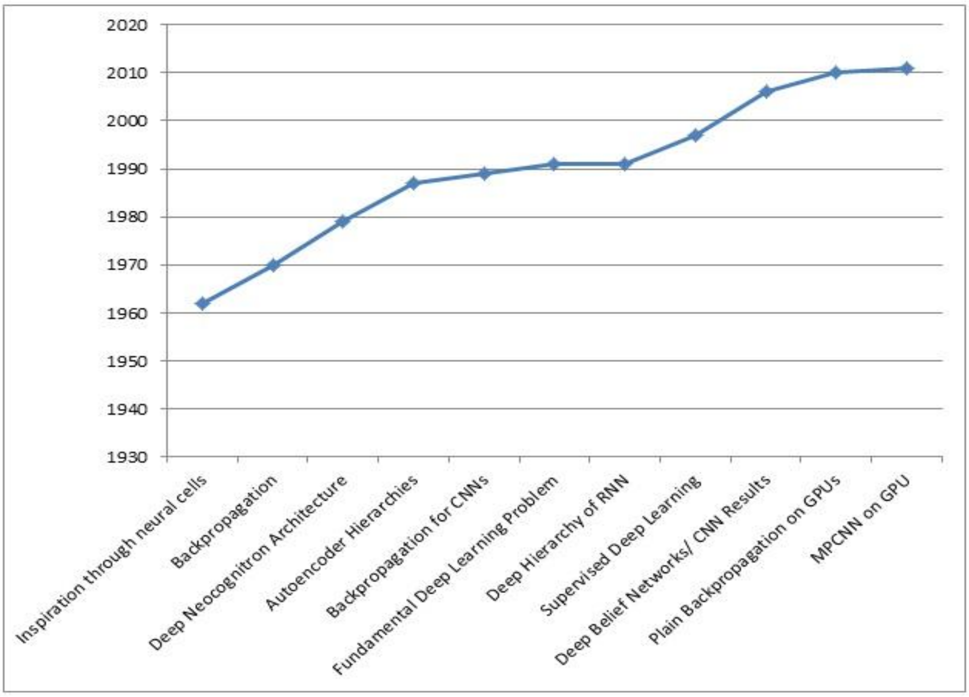

Figure 1.1: History of Deep Learning

What Is The Importance of Deep Learning?

There has been a rapid growth in the chip processing abilities over the years. There also was a notable decrease in the cost of computing hardware. These factors played an important role in bringing deep learning technology to its present hype. The recent advancement in machine learning and information processing research also contributed to this cause.

Shallow architecture can be used to solve simple problems. But when it comes to dealing with real-world applications, such as human speech, shallow architecture is found to be inadequate.

How deep learning can improve Machine Learning?

Machine Learning (ML) [6] uses supervised learning techniques, which is not a true measure of human intelligence. A human brain comprehends everything in a hierarchical fashion, and by performing a series of abstraction in each hierarchical layer. This is not achievable in Machine Learning without using Deep Learning Techniques.

As opposed to supervised learning in ML, where features are identified before the training begins, in Deep Learning the features are engineered during the process. It helps build large-scale Neural Networks where computers learn and think by itself without any kind of human intervention. It also allows the machine to generalize with more ease by including abstract representation of concepts. This leads to increasing gains in accuracy, and thereby enhances the traditional Machine Learning Techniques.

What are the different types of Deep Learning Architectures used?

There are basically three main categories of deep learning architectures: generative, discriminative and hybrid.

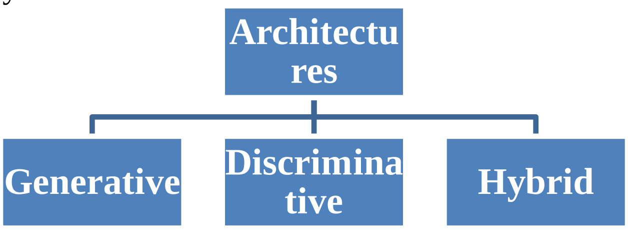

Figure 4.1: Types of Deep Learning Architecture

In a generative architecture, the actual distribution of the classification classes is modelled. That is, it models how data is generated, for the purpose of classification. Let X be the input data and Y be the classes into which X is classified, then generative architecture learns the joint probability P(x,y). This can be used to generate new likely (x,y) pairs from the P(x,y). P(x,y) can also be used for classification by applying simple conditioning on x using Bayes Rule to obtain P(y|x).

In contrast to the Generative model, in a Discriminative Architecture, the decision boundary between the classification classes is modelled. It directly learns the conditional probability P(y|x), which makes it more efficient for classification than generative models. However generative models can work with both labelled and unlabeled data.

The difference between Generative and Discriminative models can be understood more clearly through an example. Suppose pictures of different animals is the input data provided, and they have to be categorized into their corresponding species labels (output). A Generative model would learn the characteristics and features that make up each of those species (labels to be categorized into), and then use that knowledge to categorize the input. Whereas in the case of a Discriminative model, it would learn what makes a species different from another and then categorizes the input accordingly, and this proves to be more efficient.

In hybrid architecture, the advantages provided by both the above mentioned models are utilized. It uses an unsupervised generative model for its pre-training or learning phase, and then fine-tunes the system using discriminative model.

Table 4.1: Comparison between generative and discriminative models

Generative Model	**Discriminative Model**
Actual distribution is modelled	Decision boundary is modelled
Learns joint probability P(x,y)	Learns conditional probability P(y|x)
Works with labelled data	Works with labelled and unlabeled data
Learns characteristics and features of each label	Learns difference between labels

Before we dive into the details of these three architecture models, let us take a look at one of the main concepts of Deep Learning Architecture, called “Pre-training”.

Pre-Training:

Deep Neural Networks consist of more than one hidden layer where non-linear transformations are used. Linear transformations are not used because the output would only be a linear function of the input, which is equivalent to having only one hidden layer and hence defeats the whole purpose. Earlier in Deep Networks, the weights of the nodes were initialized randomly, and then the network would be trained using labelled datasets through supervised learning techniques. This method had a few disadvantages.

This method of training the network required problem specific labelled data to be available in abundance, to fit the parameters in a complex model. However, labelled data is very scarce, and would result in over-fitting of results if training was to be carried out on insufficient data.

While training shallow networks with supervised learning generally results in the parameters converging to good values, deep networks usually have non-convex optimization problems. Performing supervised training on them would result in bad local optima.

Moreover, non-convex objective functions in deep neural networks usually contain more than one local optima, which hinders learning. The local gradient descent algorithm usually begins by initializing random points and often leads to getting trapped in poor local optima. The gradient descent does not work well with randomly generated weights because of the diminishing gradient problem. As the depth of the neural network increases, the gradients that get propagated backward (especially in the Back Propagation Technique) get drastically diminished. This results in poor learning, and a slow change in the weights of the earlier layers in the network.

To overcome these disadvantages, a greedy layer-wise pre-training method [3][7][11] was introduced, where each layer of the network is trained one at a time. First a network of one layer is trained, and then two layers, and so on. The weights obtained for each of these layers after training is then used to initialize the weights of the overall network and fine tune it using a labelled data set. This pre-training uses a huge amount of unlabeled data to identify the structures and regularities in the features. There are three main ways of pre-training a deep network; using Deep Belief Networks (DBN), Auto-encoders or Sparse Encoding Symmetric Machine (SESM).

A DBN is a kind of deep generative model, consisting of a stack of Restricted Boltzmann Machine (RBM) which utilizes the above mentioned greedy algorithm to get optimized weights for all the nodes. Its time complexity is proportional to the depth and size of the network. After pre-training using DBN, BPT is used to fine tune the network. This method is also called DBN-DNN. Along with providing good weights to initialize the nodes with, this also takes care of the over-fitting and under-fitting problems often encountered in deep network training.

Another method to pre-train the network is to use SESM in DBNs, which have an architecture similar to RBMs.

Alternatively, each pair of layers can be presumed to be a de-noising auto-encoder, where a subset of inputs is set to zero to regularize it. By tweaking the gradient of hidden unit's functions with respect to the inputs, auto-encoders can also be used for pre-training.

Generative Architecture:

This architecture is easily interpreted and domain knowledge is easily embedded in it. It is composed easily and it handles uncertainty efficiently.

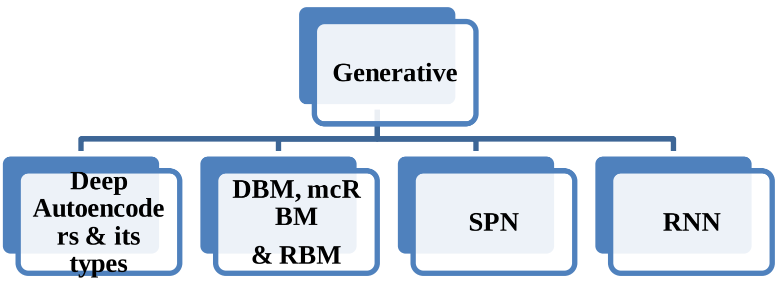

Figure 4.2: Types of Generative Models

Some frequently used Generative models are:

1. Deep auto encoders
2. DBM, mcRBM and RBM
3. SPN
4. RNN

Encoders:

This is the most common type of generative deep architecture [5]. This subtype starts off by corrupting a portion of the initial input. Then a network is designed so that the corrupted input, when passed through the hidden encoding nodes, will trace back the uncorrupted original input data. So encoders are used to undo any corruptions that occurred in the input data. For each input x, a feed-forward pass is performed to compute activations at all hidden layers. An output x' is obtained at the output layer.

Any deviation of x' from x is measured and the error is back-propagated through the neural network designed and weights are updated.

Deep Boltzmann Machine (DBM) :

DBM is a special case of Boltzmann machine (BM). A Boltzmann machine is a network consisting of stochastic binary units that are coupled symmetrically. It contains a set of visible and hidden units. The activities of hidden features in the layer below may have higher-order correlations between them. These correlations are captured in each layer in DBM. In a DBM, the variables within the same layer have no connections between them. DBMs have many advantages. DBMs have the potential of solving object and speech recognition problems by learning complex internal representations. It can utilize a large supply of unlabeled sensory inputs to build high-level representations. Then for a specific task, the model is fine-tuned using very limited labelled data. Deep Boltzmann machines propagate uncertainty about ambiguous inputs better. This is because the top-down feedback is incorporated by the initial bottom-up pass and the approximate inference procedure.

(i) Restricted Boltzmann Machine (RBM):

RBM [2] is a special case of Deep Boltzmann Machine. A DBM becomes an RBM when then is only one hidden layer present. RBM restricts BM to those with no two hidden and no two visible nodes connected to each other. Composition of many RBMs enables the usage of feature activations of one RBM as training data for the next. As a result, efficient learning of many hidden layers is possible.

(ii) mean-covariance RBM (mcRBM) :

The mcRBM is an extended version of the standard DBN with factored higher-order Boltzmann machine in its bottom layer. This is generally used for phone recognition. The mcRBM expands the data description by generating the covariance structure. Training mcRBM is not an easy task. When we go deeper into the layers of the architecture, it becomes increasingly difficult to work with mcRBM. Also, the results published are not easy to reproduce. The fine-tuning of the mcRBM parameters using the discriminative information is not an easy task.

Sum Product Network (SPN) :

An SPN is a directed acyclic graph consisting of data which are represented as leaves, sum and product which are internal nodes, and weighted edges. The sum nodes produce the mixture models, whereas the product nodes provide the field hierarchy. Unlike other deep architectures, SPNs can be trained with either a generative or discriminative object. The EM algorithm along with back propagation is used during the learning of SPN.

An SPN structure is then formed where connections are removed by zero weights. The vanishing gradient problem is one of the main difficulties in learning.

Recurrent neural networks (RNN):

This type of deep learning architecture has connections between units forming a circle. Sequential data are modelled and produced using this architecture. An arbitrary sequence of inputs is processed using RNNs. This is the reason for its impressive results in fields like handwriting recognition. They help network exhibit dynamic and temporal behavior. The upper bound to an RNN's depth can be equal to the input data sequence's length. As a result of the vanishing gradient problem, RNNs are difficult to train and therefore had not been used widely until now.

Discriminative Architecture

These models [8] are more flexible in their construction than that of Generative models. They are less constrained and more efficient for training and testing.

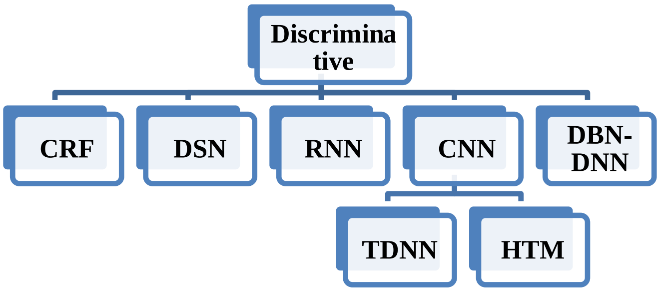

Figure 2.3: Types of Discriminative Architecture

Some commonly used Discriminative models are –

Conditional Random Fields (CRF) :

Intrinsically, the conditional probability on output labels and input data define this shallow architecture. However, by stacking the original input and the output of each layer to its higher layers, deep-structured CRFs can be developed. These deep-structured CRFs can then be applied effectively to natural language processing, phone recognition, spoken language identification, and so on.

Deep Stacking Network (DSN) :

DSN is a special type of deep model, which uses both parallel and scalable learning. DSN doesn't use stochastic gradient descent, thereby enabling it to learn network parameters parallely. The use of mean square error (MSE) between the network predictions and the target values in each module of the model is the reason behind the DSN's ability to do scalable learning.

In this model, simple classifiers or modules of functions are first composed, which are stacked on top of one another to learn complex classifiers and functions. Each of the modules in the stack is a shallow multilayer perceptron which uses convex optimization so as to learn perceptron weights. The hidden units of each module are sigmoidal and non-linear while the output units are linear.

Recurrent Neural Networks (RNN) :

RNNs can be used as both Generative and Discriminative models. When the output is considered to be the input data that is predicted in the future, the RNN becomes Generative (discussed in an earlier section). When the output is a labelled sequence associated with the input data sequence, it becomes Discriminative in nature. Pre-segmentation of training data and post-processing of output to transform it into labelled sequences is required for training RNNs for discrimination. HMMs are generally used to carry out the above mentioned requirements. However, there exists a method for RNNs to carry out sequence classification by avoiding costly pre-segmentation of the training dataset and post-processing of the outputs. The basic principle behind that method is as follows: given all input sequences, the outputs of the RNN have to be interpreted as conditional distributions of the labeled sequences. This is to obtain a differentiable objective function which will help optimize the conditional distributions of the appropriate labeled sequences.

Convolutional Neural Network (CNN) :

A CNN [10] generally consists of modules stacked one on top of each other, sometimes even with a DNN on the top, forming a deep model. Each of these modules consists of two kinds of layers; a convolutional layer and a pooling layer. Weight sharing takes place in the convolutional layer. The pooling layer reduces the data rate from layers below by sub-sampling the output of the convolutional layer. CNNs can have some invariance properties like translation invariance, owing to weight sharing done in the convolutional layers, and pooling schemes chosen. For complex pattern recognition tasks, a wider range of invariance needs to be handled and hence such equi-variance or limited invariance is not sufficient. Even so, CNNs are widely used for speech recognition, and in computer vision and image recognition.

Time Delay Neural Network (TDNN) :

This model is a special case of CNN, where weight sharing is limited to the time dimension, which is one of the two dimensions. However, it has been discovered that the right dimension to impose invariance on is the frequency dimension. It is more effective for pooling outputs and sharing weights.

Hierarchical Temporal Memory (HTM) :

HTM is another variant of the CNN, where information fusing and decision making are done using Bayesian Probabilistic formalism. As opposed to the use of only bottom-up information flow in CNNs, HTMs use both top-down and bottom-up information flows. It also uses the temporal or time dimension as the supervision information for discrimination.

DBN-DNN:

In this model, Deep Neural Networks (DNN) or MLPs with lots of hidden layers use unsupervised DBN pre-training for the network to learn, and then use back-propagation to fine-tune the network. In short, DBNs are used to initialize the training here.

Hybrid Architecture:
In hybrid architecture, both generative and discriminative modelling properties are utilized. Generative model can provide excellent initialization points and control the overall complexity of the model. This is the reason why it is used for pre-training the discriminative model.

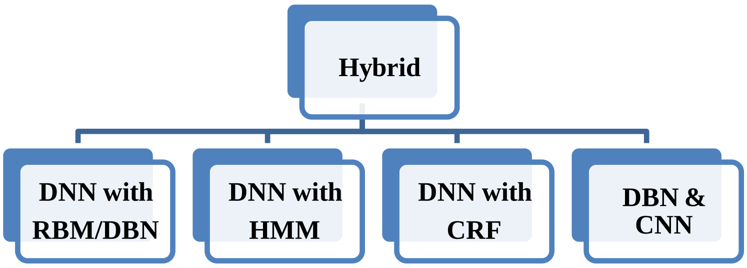

Figure 4.4: Types of Hybrid Architecture

DNN-RBM/DBN:
The generative DBN is fine-tuned by discriminative models to produce an equivalent DNN. This DNN is pre-trained by stacked RBMs or DBMs.

DNN-HMM:
The Hidden Markov Model is a finite set of states, and each state has a probability distribution. Transition probabilities control the transitions between the states. According to the associated probability distribution of a particular state an outcome is generated. The external observer can view only the outcome. The states are not visible to an external observer and hence are said to be hidden to the outside. The full-sequence maximum mutual information (MMI) between the entire label sequence and the input vector sequence helps learn the parameters in the case of DNN-HMM.

DNN-CRF:
Here, DNN weights are initialized by the generative DBN and it is fine-tuned by sequential discriminative information. DNN-CRF is a combination of DNN, which is static, and CRF, which has a shallow discriminative architecture. In DNN-CRF, the learning of the architecture is done using the probability of labelled sequences given in the input sequence (conditional probability).

DNN-CNN:
Deep convolutional neural networks (CNN) are pre-trained by the generative model of DBN. The discrimination of the deep is improved by the DBN pre-training.

What Tools Are Used For Deep Learning?

Deeplearning4j is an open source deep learning library written for Java and the Java Virtual Machine and a computing framework with wide support for deep learning algorithms. Deeplearning4j includes implementations of the restricted Boltzmann machine, deep belief net, deep auto-encoder, stacked de-noising auto-encoder and recursive neural tensor network, as well as word2vec, doc2vec and Glove. These algorithms all include distributed parallel versions that integrate with Hadoop and Spark

Training with Deeplearning4j takes place in the cluster, which means it can process massive amounts of data. Neural nets are trained in parallel via iterative reduce, which works on Hadoop/YARN and on Spark. Deeplearning4j also integrates with Cuda kernels to conduct pure GPU operations, and works with distributed GPUs. It works under a terminology called as neural networks. Hence deep learning neural networks are known for its accuracy and it is compatible with Java, Scala and Clojure. When a deep learning neural network is trained,there will be a lot of parameters that can be adjusted and the neural network also involves a variety of layers. Thus, it helps us to easily combine with auto-encoders, convolutional networks, restricted Boltzmann machine and so on.

It is more widely used by non-researchers than being used as a research tool. There are many other applications and use cases which might be dealt within the application part of the paper.

What Are The Different Applications of Deep Learning?

The applications of Deep Learning are numerous and multifarious, as it can be applied to almost all areas of research. In this paper, we shall take a look at four different applications of Deep Learning.

Natural Language Processing:

The deep learning concept was first applied in the area of Language Modelling (LM) which deals with words and symbols. The parameters in language modelling were based on N-gram counts. N-gram counts were later replaced by neural networks and other deep learning concepts. LM captures the qualities of the difference in the sequence of words in natural language. When neural networks were used in language modelling, it came to be known as the Neural Network Language Modelling (NNLM). The NNLM consisted of an input layer, hidden layer and an output layer. It used a concept called as "word embedding" for the distributed representation of words. In this way a series of text can thus be converted into feature vectors by mapping the inputs to the outputs. The computational complexity of the NNLM’s was high compared to the N-grams. The concept "embedding" refers to a symbolic information in a text at word-level or sentence-level. Sparse vectors are used for the symbolic representations. Current work suggests new methods of learning text embedding that can assess the meaning of words by combining all relevant document contexts using recursive neural networks. Hence this provides versatility when the deep learning concepts are applied to the unlabeled training data set. Later on the Continuous Bag-Of-Words model is introduced to handle huge amounts of data.

The "Skip-gram model" was introduced by performing inverse predictions to the word-embedding system. A new technique using the concept of data mining is developed which deals with the linear mapping of the source and target languages. Dictionaries and phrase tables are also developed that convert one language into another. Then comes the use of neural tensor networks that is used for knowledge classification by using knowledge graphs. Sentiment analysis task is done for input text information based on the positive or negative opinion by the neural tensor network. Hence the classification becomes simpler and easier by the use of these networks.

Information retrieval:

The process of obtaining a set of relevant documents by a query entered by the user from a collection of large amounts of documents is known as information retrieval [9]. Queries are entered in the form of strings in the search engine. A document or an object is stored in the form of a metadata in a database. There are various ways of retrieving a document (1) Boolean retrieval: Depending on how well the document matches the query, it is retrieved from the database accordingly. (2) Algebraic retrieval: Models such as the vector space model, topic-based vector space model, extended Boolean model, and latent semantic analysis, which represents documents and queries as vectors, matrices etc. (3) Probabilistic retrieval: Probabilities and probabilistic values are computed. Methods include binary independence model, probabilistic relevance model with the BM25 relevance function, methods of inference with uncertainty etc. (4) Feature-based retrieval:. Methods such as “learning to rank” are devised to combine features into a single relevance score. The main applications of information retrieval in deep learning are semantic hashing, deep-structured semantic modelling and use of deep stacking networks. In semantic hashing it uses a compact code so that the documents are mapped to the memory addresses for quick retrieval and gives faster results compared to the traditional TF-IDF approach. In deep structured semantic/similarity modelling, different terms are grouped into a similar context that falls into the same semantic cluster. This is done using the objective function and based on linear projection. The concept of stacking is done by using DSN architecture and it is used for solving ad-related problems.

Twitter analysis:

Deep Learning on Twitter analysis has created applications such as Photo Time and the like based on the principle that the deep learning systems perform better when it is fed with more data. The application which is developed by deep learning has a variety of features like it can tag smart phone photos including place, date, object and scene, search for photos and can also recognize faces from the social networking sites or from the contacts. Photo time can also index the album on the user's phone and it can be done without connecting to the internet.

Applications on Computer vision and Object Recognition:
Object recognition in computer vision had been depending on SIFT (scale-invariant feature transform) and HOG (histogram of oriented gradients) for many years. Since SIFT and HOG captured only the low-level edge information they were classified into two different features (1) supervised and (2) unsupervised learning. In unsupervised learning the deep auto-encoders were developed in the during the early stages to learn useful visual hierarchies. It also uses the principal of sparse connectivities and third-order Boltzmann machine. Deep learning methods here can also be applied in hierarchical and generative methods and can be used in representing the facial expression datasets. It mostly uses a stochastic method which is represented graphically and starts from an input node and ends in an output node. The most successful work done in this field of unsupervised learning is a nine-layer locally connected sparse auto-encoder with a local contrast normalization. The supervised feature learning is based on the Convolutional Neural Networks (CNN). When large amounts of training data are available, then it results in high -performance. The CNN involves layers of Convolution, sub-sampling, full and Gaussian connections. The CNN also has a receptive field and filter weights just like a two-dimensional filter which is mostly used in image-processing. Hence this model is called as the deep CNN. The deep CNN had an error rate of about 15.3%-16.4% depending on the training data.

Comparison between the different applications discussed in this paper:

Table 6.1: Comparison between different applications of deep learning

Natural Language Processing	**Information Retrieval**	**Computer Vision**
Deals with words and were based on N-gram counts.	Retrieving information by entering a query.	Based on the concepts of supervised and unsupervised learning.
The main advantage over the traditional counting-based N-gram is that history is seen as the projection to the lower dimension space.	Using the compact code produced by deep auto-encoders, documents are mapped to memory addresses in such a way that semantically similar text documents are located at nearby addresses to facilitate rapid document retrieval.	When labeled data are relatively scarce, unsupervised learning algorithms have been shown to learn useful visual feature hierarchies.
Relies on the learning algorithm to discover meaningful, continuous-valued features.	Latent semantic models are able to map a query to its relevant documents at the semantic level where lexical-matching often fails.	Deep learning methods have a rich family, including hierarchical probabilistic and generative models.
Compared with the class-based N-gram LMs, the NLP's project all words into the same low dimensional space, in which there can be many degrees of similarity between words.	By using the convolution structure, local contextual information at the N-gram level is modeled first and then the salient local features in a word sequence are combined to form a global feature vector.	While the deep CNN has demonstrated remarkable classification performance on object recognition tasks, there has been no clear understanding of why they perform so well until recently.

How is Deep Learning suitable for Big Data Analysis?

Deep learning is rapidly emerging in the field of Data Science and Machine Learning. It has provided good results in solving large, complex, and high-dimensional machine learning problems, including dimensionality reduction, feature selection, feature extraction, classification and clustering. Deep learning research has been carried out in many fields, but the main common ground is Big Data or Data mining. It brings out unique solutions for multiple disciplines in big data. Deep learning algorithms use a huge amount of unsupervised data to automatically extract complex representations. Big Data generally refers to data that exceeds the typical storage (i.e. The computing capacity of conventional databases and data analysis techniques). Deep Learning algorithms extract meaningful representations of the raw data through the use of a hierarchical multi-level learning approach. Deep Learning can be applied to labelled data as well as unlabeled/unsupervised data. Labelled data are available in sufficiently large amounts and it is usually available as user inputs. Unlabeled data on the other hand, can be used to produce meaningful or useful inputs.

With the advent of Deep Learning, it has become more efficient to analyze Big Data, especially because it eliminates the need for labelled data to a great extent. Data is abundant in the world, however, labelled data is not; and to generate labelled data is no simple task. Even if one sets about the task of labelling a huge dataset to provide as the training set to the network, it would probably introduce a lot of discrepancies due to human error. This would lead to ineffective pre-training and jeopardize the whole network. The remedy to this problem is unsupervised pre-training which is one of the major features in Deep Learning, and hence is more suitable for analyzing Big Data than the traditional Machine Learning Techniques.

Other useful characteristics of the learnt abstract representations of Deep Learning include the following: (1) Simpler models can be obtained from the complex representations (2) It can be applied to different kinds of data types such as audio, video or image processing, etc., which comes under unsupervised data (3) Deep learning concepts can be used for decision-making, semantic indexing, information retrieval and can be used in other big-data analytics for solving complex problems.

What are the challenges faced by Deep Learning Architectures?

As an upcoming field of Data Science, Deep Learning has a few challenges to be tackled. One of the most important issues is deciding whether the entire Big Data input corpus should be utilized by the Deep Learning Algorithms while analyzing data or not. Ideally, these algorithms should take only a portion of the available input corpus to train the data representation patterns, and then use those learnt patterns along with the remaining input corpus to extract data representations and abstractions. However, what percentage of the input data are required for better training of the network is a question that still remains unclear.

Domain Adaptation is another issue, related to the one described above. It is associated with the varying characteristic of Big Data. The variation in input data types and domains while shifting between the input data (which is used for training the representations) and the target data (which is used for generalizing the representations) poses a problem for Deep Learning Algorithms. The distribution of training data is much different from the distribution of test data, and a lot of research is still going on to find solutions to increase the capacity of data representations and patterns to generalize.

The criteria that are necessary for useful semantic meaning to be provided by the extracted data representations for the Big Data is still an unsolved issue. Since Big Data Analytics involves learning from largely unsupervised data, it is quite unsuitable to define a criterion for training Deep Learning Algorithms like the conventional learning algorithms that use misclassification error as criterion for training and learning patterns.

This leads to yet another challenge faced by Deep Learning. What would be a good data representation for effective semantic indexing and data tagging? There is a lot of ongoing research to incorporate semi-supervised training methods (since input corpus can contain both labelled and unlabelled data) in Deep Learning Algorithms so as to obtain good defining criteria for learning good data representation.

Conclusion

With its hierarchical learning and high level data abstraction techniques, Deep Learning has proven to be more efficient than conventional neural networks. Deep Learning provides both unsupervised and semi-supervised techniques to learn representations of data. The various types of architecture that exist in deep learning allow one to choose one among them after analyzing the pros and cons of each type and understanding which model suits the scenario the best. Deep Learning is also particularly suited for Big Data Analysis since it does not need labelled data for training the network..Domain adaptation, percentage of input data required for efficient training, criterion for training Deep Networks, good data representation for effective tagging and indexing are issues that still remain unaddressed.Even given all its short comings, this technique (Deep Learning) has taken technology a lot closer to Artificial Intelligence than ever before and is still subject to a lot of research.

References

[1] Arel, I, Rose, C, and Karnowski T, "Deep Machine Learning - A New Frontier in Artificial Intelligence," *IEEE Computational Intelligence Mag*., Nov., 2010.

[2] AsjaFischer and Christian Igel, "An Introduction to Restricted Boltzmann Machines", *Institutf¨urNeuroinformatik, Ruhr-Universit¨at Bochum, Germany*

[3] DumitruErhan, YoshuaBengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, "Why Does Unsupervised Pre-training Help Deep Learning?",*Journal of Machine Learning Research,* Published 2/10

[4] "Deep Learning Tutorial", *LISA lab, University of Montreal,* Release 0.1, March 18, 2015

[5] Pierre Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures", *Workshop on Unsupervised and Transfer Learning,* 2012

[6] YoshuaBengio. Learning deep architectures for AI.*Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[7] Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks, Vol. 19

[8] Collobert R. "Deep learning for efficient discriminative parsing", Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2010.

[9] Deng, L. “An Overview of Deep-Structured Learning for Information Processing, in Proceedings of Asian-Pacific Signal & Information Processing Annual Summit and Conference (APSIPA-ASC), October 2011.

[10] Deng, L., Abdel-Hamid, O., and Yu, D. “A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion,” Proc. ICASSP, 2013.

[11] Erhan, D., Bengio, Y., Courvelle, A., Manzagol, P., Vencent, P., and Bengio, S. “Why does unsupervised pre-training help deep learning?” J. Machine Learning Research, 2010, pp. 201-208.