**Dimensionality Reduction in Big Data using Deep Learning: an Autoencoder Based Approach**

***Abstract-*** There are various Machine Learning challenges associated with BigData. In this paper we try to address the challenges caused due massive volume associated with Big Data specifically high dimensionality and try to investigate the dimensionality reduction ability of the autoencoder to identify if it has any property that can help in improving the efficiency of autoencoder. This paper first starts by describing the problem faced due to high dimensionality of big data and then continues to discuss autoencoder and dimensionality reduction ability of autoencoder along with a brief introduction of other popular dimensionality reduction techniques. Finally an approach is proposed which primarily aims at identifying the influence of number of hidden layer nodes on the dimensionality reduction ability of the autoencoder along with comparing the effectiveness of an autoencoder with respect to other dimensionality reduction techniques.

**1. Introduction**

The amount of data is increasing at a very high rate because of developments in Web technologies, mobile and sensing devices. Twitter processes approximately 80M tweets in a day this generates over 8TB of data on a daily basis [1]. Research by ABI gives an estimate at the given rate there will be more that thirty billion connected devices [2]. These Data possess remarkable possibility in terms of corporate fields such as health care, transportation, marketing and financial services [3],[4]. But conventional paradigms and approaches face a lot of challenges in dealing with such a large amount of data

ML is one of the main drivers of the Big Data revolution [5]. The reason for this is its ability to learn from data and provide data driven insights, decisions, and predictions [6]. An assumption is that algorithms can learn better with more data providing more accurate results [7]. However, massive datasets cause a lot of problems because conventional algorithms are not designed to meet up such requirements. For instance, quite a lot of ML algorithms were designed for smaller datasets, hoping that the entire data can fit in memory. Big Data break these stereotypes, making conventional algorithms inoperative or greatly impeding their performance.

A large number of approaches have been designed to make machine learning algorithms able to work with large datasets for examples new processing paradigms like MapReduce [8] and frameworks like Hadoop [9]. Branches of Machine Learning such as deep learning and online learning are extensively used to tackle challenges associated with big data

This paper focuses on handling the challenge associated with the first and most important characteristic of Big Data i.e Volume using Deep learning: Autoencoders, to handle this challenge. This paper first compiles and summarises the importance of Big Data and need for development of new ways to handle Machine learning challenges in Big Data.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents machine learning challenges associated with the Volume characteristic of Big Data and then Section IV Identifies the ability of an Autoencoder to reduce the dimensionality by understanding the difference between auto encoder and state-of-the-art dimensionality reduction methods. Section V aggregates the findings and lay foundations for future works concluding the paper

**2. Existing Work**

One of the challenging aspects in Big Data Analytics is dealing with streaming and fast-moving input data. Such data analysis is useful in monitoring tasks, such as fraud detection. It is important to adapt Deep Learning to handle streaming data, as there is a need for algorithms that can deal with large amounts of continuous input data. In this section, we discuss some works associated with Deep Learning and streaming data, including incremental feature learning and extraction [10], denoising autoencoders [11], and deep belief networks [12].

Zhou et al. [10] describe how a Deep Learning algorithm can be used for incremental feature learning on very large datasets, employing denoising autoencoders [11]. Denoising autoencoders are a variant of autoencoders which extract features from corrupted input, where the extracted features are robust to noisy data and good for classification purposes. Zhou et al. [10] demonstrate that the incremental feature learning method quickly converges to the optimal number of features in a large-scale online setting. This kind of incremental feature extraction is useful in applications where the distribution of data changes with respect to time in massive online data streams

Calandra et al. [12] introduce adaptive deep belief networks which demonstrate how Deep Learning can be generalized to learn from online non-stationary and streaming data. Their study exploits the generative property of deep belief networks to mimic the samples from the original data, where these samples and the new observed samples are used to learn the new deep belief network which has adapted to the newly observed data. However, a downside of an adaptive deep belief network is the requirement for constant memory consumption.

Chen et al. [13] introduce marginalized stacked denoising autoencoders (mSDAs) which scale effectively for high-dimensional data and is computationally faster than regular stacked denoising autoencoders (SDAs). Their approach marginalizes noise in SDA training and thus does not require stochastic gradient descent or other optimization algorithms to learn parameters. The marginalized denoising autoencoder layers to have hidden nodes, thus allowing a closed-form solution with substantial speed-ups. Moreover, marginalized SDA only has two free meta-parameters, controlling the amount of noise as well as the number of layers to be stacked, which greatly simplifies the model selection process. The fast training time, the capability to scale to large-scale and high-dimensional data, and implementation simplicity make mSDA a promising method with appeal to a large audience in data mining and machine learning.

Convolutional neural networks are another method which scales up effectively on high-dimensional data. Researchers have taken advantages of convolutional neural networks on ImageNet dataset with 256 ×256 RGB images to achieve state of the art results [14],[15]. In convolutional neural networks, the neurons in the hidden layers units do not need to be connected to all of the nodes in the previous layer, but just to the neurons that are in the same spatial area. Moreover, the resolution of the image data is also reduced when moving toward higher layers in the network.

Empirical results have demonstrated the effectiveness of large-scale models [16],[17], with particular focus on models with a very large number of model parameters which are able to extract more complicated features and representations [18],[19].

Although achieving comparable performance and widely applied, deep learning is kind of like a “black-box” actually and there is not very sufficient and strict theoretical system to support. So a problem is: we have impressive performance using deep learning but we do not know why theoretically. In deep learning, a number of researchers tend to make progress by employing increasingly deep models and complex unsupervised learning algorithms.

This paper comes from the idea that whether auto-encoder has some kind of good property which might accumulate when being stacked and thus contribute to the success of deep learning. We start from a “building block “of deep learning—auto-encoder and focus on its dimensionality reduction ability. When restricting the number of hidden layer nodes less than the number of original input nodes in an auto-encoder, the desired dimensionality reduction effect can be achieved.

In this paper we start by introducing Autoencoder and focus on its ability to reduce the dimensionality, trying to understand the difference between auto-encoder and state-of-the-art dimensionality reduction methods

In the next part of this paper we will design the experiment to test the hypotheses of the proposed methodology.

On the whole, we expect that analyzing the fundamental methods in deep learning, e.g. auto-encoder, could help us understand deep learning better.

**3. Volume Related Machine Learning Challenges in BigData**

The first and the most talked about characteristic of Big Data is volume: it is the amount, size, and scale of the data. In the machine learning context, size can be defined either vertically by the number of records or samples in a dataset or horizontally by the number of features or attributes it contains. Furthermore, volume is relative to the type of data: a smaller number of very complex data points may be considered equivalent to a larger quantity of simple data [20]. This is perhaps the easiest dimension of Big Data to define, but at the same time, it is the cause of numerous challenges. The following sub-section discusses the challenges faced because of High Dimensionality associated with BigData.

3.1 Curse of Dimensionality

Curse of Dimensionality refers to difficulties encountered when working in high dimensional space. Specifically, the dimensionality describes the number of features or attributes present in the dataset. The Hughes effect [38] states that for a training set of static size, the predictive ability and effectiveness of an algorithm decreases as the dimensionality increases. Therefore, as the number of features increases, the performance and accuracy of machine learning algorithms degrades. This can be explained by the breakdown of the similarity-based reasoning upon which many machine learning algorithms rely [37]. Unfortunately, the greater the amount of data available to describe a phenomenon, the greater becomes the potential for high dimensionality because there are more prospective features. Consequently, as the volume of Big Data increases, so does the likelihood of high dimensionality.

In addition, dimensionality affects processing performance: the time and space complexity of ML algorithms is closely related to data dimensionality [21]. The time complexity of many ML algorithms is polynomial in the number of dimensions. As already mentioned, the time complexity of the principal component analysis is O(mn2+n3) and that of logistic regression O(mn2+n3) , where m is the number of samples and n is the number of dimensions.

3.2 Feature Engineering

High dimensionality is closely related to another volume challenge: feature engineering. This is the process of creating features, typically using domain knowledge, to make machine learning perform better. Indeed, the selection of the most appropriate features is one of the most time consuming pre-processing tasks in machine learning [22]. As the dataset grows, both vertically and horizontally, it becomes more difficult to create new, highly relevant features. Consequently, in a manner similar to dimensionality, as the size of the dataset increases, so do the difficulties associated with feature engineering.

Feature engineering is related to feature selection: whereas feature engineering creates new features in an effort to improve learning outcomes, feature selection (dimensionality reduction) aims to select the most relevant features. Although feature selection reduces dimensionality and hence has the potential to reduce ML time, in high dimensions it is challenging due to spurious correlations and incidental endogeneity (correlation of an explanatory variable with the error term) [23].

**4. Methodology**

In this section we briefly discuss Autoencoders, four representative dimensionality reduction methods and concept of dimensionality reduction and intrinsic dimensionality reduction techniques.

Dimensionality reduction primarily addresses the curse of dimensionality and the processing performance challenges.

Dimensionality reduction aims to map high dimensionality space onto lower-dimensionality one without significant loss of information. A variety of methods exists to reduce dimensions in the Big Data context. In this paper we are primarily concerned with the Dimensionality reduction using Autoencoder.

4.1 Autoencoder

Autoencoder: Architecture and Objective Function

*Input (Layer L1) Layer L2 Output (Layer L3)*

Figure 1. Autoencoder Architecture

The input is compressed and is then reconstructed

Decoder

Encoder

Input Code O Output

Figure 2. The Visualization Description of an auto-encoder

Suppose the original input ***x*** belongs to n-dimensional space and the new representation ***y*** belongs to m-dimensional space, an auto-encoder is a special and tricky three-layered neural network in which we set the output equal to the Input . J is the reconstruction error. It uses back propagation for training.

We try to minimize reconstruction error **J** by adjusting the parameters in encoder and decoder to get the code.

When restricting the number of original hidden layer nodes *p* less than the number of original input nodes *q,* we get the compressed representation of input which helps us in achieving desired dimensionality reduction.

Autoencoders have also been used to reduce dimensions; they learn an encoding of a dataset. They are similar to a multi-layer perceptron (MLP): one input, one output, and one or more hidden layers. The difference from the MLP is that an autoencoder always has the same number of input and output nodes. Whereas the MLP learns the mapping between the input and target variables, an autoencoder learns to reconstruct its inputs. The hidden layers are responsible for encoding, they map the input feature space X to a lower-dimension space F, creating a compressed representation of X. The output layer on the other hand, serves as the decoder and reconstructs the input X from the compressed representation.

Following is a brief introduction of other dimensionality reduction methods [19]

*PCA*: Principal component analysis is a very popular linear technique for dimensionality reduction. Given a dataset on R^n, PCA aims to and a linear subspace of dimension d lower than n which attempts to maintain most of the variability of the data.

*LDA*: Linear discriminate analysis is another popular linear dimensionality reduction method. The basic idea is to ensure the samples after projection to have maximum-between-cluster-distance and minimum-in-cluster-distance in the new subspace.

*LLE*: Locally linear embedding is a nonlinear approach to reduce dimensionality by computing low-dimensional, neighbourhood preserving embedding of high-dimensional data. A dataset of dimensionality n, which is assumed to lie on or near a smooth nonlinear manifold of dimensionality don, is mapped into a sin-gle global coordinate system of lower dimensionality d. The global nonlinear structure is recovered by locally linear ﬁts.

*Isomap*: Isomap is a nonlinear generalization of classical multidimensional scaling. The main idea is to perform multi-dimensional scaling, not in the input space, but in the geodesic space of the nonlinear data manifold. The geodesic distance represents the shortest paths along the curved surface of the manifold measured as if the surface were ﬂat. This can be approximated by a sequence of short steps between neighbouring sample point.

4.2 Dimensionalty Reduction Ability of Autoencoder

As a kind of common rule, superﬁcially high-dimensional and complex phenomena can actually be dominated by a small amount of simple variables in most situations. Dimensionality reduction is an old and young, dynamic research topic [11,12]. It is looking for a projection method that maps the data from high feature space to low feature space. Dimensionality reduction methods in general can be divided into two categories, linear and nonlinear. This paper describes auto-encoder's dimensionality reduction ability by comparing auto-encoder with several linear and non-linear dimensionality reduction methods in both a number of cases from two-dimensional and three-dimensional spaces for more intuitive results and real datasets

The next part of paper will be dedicated to the Experimental setup and Analysing the results of the experiment to Identify the dimensionality reduction ability of an autoencoder and compare it with other popular methods mentioned above.

Evaluation of Autoencoder’s dimensionality reduction ability is divided in two parts. The first part deals with synthesised data and real datasets. The second part deals with real datasets and identifies influence of number of nodes in hidden layer on the performance of autoencoder.

**V. Conclusion and Future work**

This paper starts by discussing the importance of Big Data and discusses how these challenges are dealt with in the context of machine learning, we than list the two important challenges associated with the Volume characteristic of Big Data namely, the curse of dimensionality and feature extraction. Then we discuss about autoencoders and provide a brief description about other Dimensionality reduction techniques. Next we propose an approach to analyse the Dimensionality reduction ability of Autoencoder in Comparison to other popular dimensionality reduction techniques. The next part of this paper will focus designing the experimental setup to test the hypotheses proposed in the Methodology mentioned in this paper. This might help in identifying a possible relationship between number of hidden layer nodes and intrinsic dimensionality of the dataset.

.