Chapter 5
Deep Learning Techniques in Big Data Analytics

Maryam M. Najafabadi, Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald and Edin Muharemagc

Introduction

The general focus of machine learning is the representation of the input data and generalization of the learnt patterns for use on future unseen data. The goodness of the data representation has a large impact on the performance of machine learners on the data: a poor data representation is likely to reduce the performance of even an advanced, complex machine learner, while a good data representation can lead to high performance for a relatively simpler machine learner. Thus, feature engineering, which focuses on constructing features and data representations from raw data [1], is an important element of machine learning. Feature engineering consumes a large portion of the effort in a machine learning task, and is typically quite domain specific and involves considerable human input. For example, the Histogram of Oriented Gradients (HOG) [2] and Scale Invariant Feature Transform (SIFT) [3] are popular feature engineering algorithms developed specifically for the computer vision domain. Performing feature engineering in a more automated and general fashion would be a major breakthrough in machine learning as this would allow practitioners to automatically extract such features without direct human input.

Deep Learning algorithms are one promising avenue of research into the automated extraction of complex data representations (features) at high levels of abstraction. Such algorithms develop a layered, hierarchical architecture of learning and representing data, where higher-level (more abstract) features are defined in terms of lower-level (less abstract) features. The hierarchical learning architecture of Deep Learning algorithms is motivated by artificial intelligence emulating the

This chapter has been adopted from the Journal of Big Data, Borko Furht and Taghi Khoshgoftar, Editors-in-Chief.

B. Furht and F. Villanustre, *Big Data Technologies and Applications*,
DOI 10.1007/978-3-319-44550-2_5

deep, layered learning process of the primary sensorial areas of the neocortex in the human brain, which automatically extracts features and abstractions from the underlying data [4–6]. Deep Learning algorithms are quite beneficial when dealing with learning from large amounts of unsupervised data, and typically learn data representations in a greedy layer-wise fashion [7, 8]. Empirical studies have demonstrated that data representations obtained from stacking up nonlinear feature extractors (as in Deep Learning) often yield better machine learning results, e.g., improved classification modeling [9], better quality of generated samples by generative probabilistic models [10], and the invariant property of data representations [11]. Deep Learning solutions have yielded outstanding results in different machine learning applications, including speech recognition [12–16], computer vision [7, 8, 17], and natural language processing [18–20]. A more detailed overview of Deep Learning is presented in "Deep Learning in Data Mining and Machine Learning" section.

Big Data represents the general realm of problems and techniques used for application domains that collect and maintain massive volumes of raw data for domain-specific data analysis. Modern data-intensive technologies as well as increased computational and data storage resources have contributed heavily to the development of Big Data science [21]. Technology based companies such as Google, Yahoo, Microsoft, and Amazon have collected and maintained data that is measured in exabyte proportions or larger. Moreover, social media organizations such as Facebook, YouTube, and Twitter have billions of users that constantly generate a very large quantity of data. Various organizations have invested in developing products using Big Data Analytics to addressing their monitoring, experimentation, data analysis, simulations, and other knowledge and business needs [22], making it a central topic in data science research.

Mining and extracting meaningful patterns from massive input data for decision making, prediction, and other inferencing is at the core of Big Data Analytics. In addition to analyzing massive volumes of data, Big Data Analytics poses other unique challenges for machine learning and data analysis, including format variation of the raw data, fast moving streaming data, trustworthiness of the data analysis, highly distributed input sources, noisy and poor quality data, high dimensionality, scalability of algorithms, imbalanced input data, unsupervised and un-categorized data, limited supervised/labeled data, etc. Adequate data storage, data indexing/tagging, and fast information retrieval are other key problems in Big Data Analytics. Consequently, innovative data analysis and data management solutions are warranted when working with Big Data. For example, in a recent work we examined the high-dimensionality of bioinformatics domain data and investigated feature selection techniques to address the problem [23]. A more detailed overview of Big Data Analytics is presented in "Big Data Analytics" section.

The knowledge learnt from (and made available by) Deep Learning algorithms has been largely untapped in the context of Big Data Analytics. Certain Big Data domains, such as computer vision [17] and speech recognition [13], have seen the application of Deep Learning largely to improve classification modeling results. The ability of Deep Learning to extract high-level, complex abstractions and data

representations from large volumes of data, especially unsupervised data, makes it attractive as a valuable tool for Big Data Analytics. More specifically, Big Data problems such as semantic indexing, data tagging, fast information retrieval, and discriminative modeling can be better addressed with the aid of Deep Learning. More traditional machine learning and feature engineering algorithms are not efficient enough to extract the complex and non-linear patterns generally observed in Big Data. By extracting such features, Deep Learning enables the use of relatively simpler linear models for Big Data analysis tasks, such as classification and prediction, which is important when developing models to deal with the scale of Big Data. The novelty of this study is that it explores the application of Deep Learning algorithms for key problems in Big Data Analytics, motivating further targeted research by experts in these two fields.

The paper focuses on two key topics: (1) how Deep Learning can assist with specific problems in Big Data Analytics, and (2) how specific areas of Deep Learning can be improved to reflect certain challenges associated with Big Data Analytics. With respect to the first topic, we explore the application of Deep Learning for specific Big Data Analytics, including learning from massive volumes of data, semantic indexing, discriminative tasks, and data tagging. Our investigation regarding the second topic focuses on specific challenges Deep Learning faces due to existing problems in Big Data Analytics, including learning from streaming data, dealing with high dimensionality of data, scalability of models, and distributed and parallel computing. We conclude by identifying important future areas needing innovation in Deep Learning for Big Data Analytics, including data sampling for generating useful high-level abstractions, domain (data distribution) adaption, defining criteria for extracting good data representations for discriminative and indexing tasks, semi-supervised learning, and active learning.

The remainder of the paper is structured as follows: "Deep Learning in Data Mining and Machine Learning" section presents an overview of Deep Learning for data analysis in data mining and machine learning; "Big Data Analytics" section presents an overview of Big Data Analytics, including key characteristics of Big Data and identifying specific data analysis problems faced in Big Data Analytics; "Applications of Deep Learning in Big Data Analytics" section presents a targeted survey of works investigating Deep Learning based solutions for data analysis, and discusses how Deep Learning can be applied for Big Data Analytics problems; "Deep Learning Challenges in Big Data Analytics" section discusses some challenges faced by Deep Learning experts due to specific data analysis needs of Big Data; "Future Work on Deep Learning in Big Data Analytics" section presents our insights into further works that are necessary for extending the application of Deep Learning in Big Data, and poses important questions to domain experts; and in "Conclusion" section we reiterate the focus of the paper and summarize the work presented.

Deep Learning in Data Mining and Machine Learning

The main concept in deep leaning algorithms is automating the extraction of representations (abstractions) from the data [5, 24, 25]. Deep learning algorithms use a huge amount of unsupervised data to automatically extract complex representation. These algorithms are largely motivated by the field of artificial intelligence, which has the general goal of emulating the human brain's ability to observe, analyze, learn, and make decisions, especially for extremely complex problems. Work pertaining to these complex challenges has been a key motivation behind Deep Learning algorithms which strive to emulate the hierarchical learning approach of the human brain. Models based on shallow learning architectures such as decision trees, support vector machines, and case-based reasoning may fall short when attempting to extract useful information from complex structures and relationships in the input corpus. In contrast, Deep Learning architectures have the capability to generalize in non-local and global ways, generating learning patterns and relationships beyond immediate neighbors in the data [4]. Deep learning is in fact an important step toward artificial intelligence. It not only provides complex representations of data which are suitable for AI tasks but also makes the machines independent of human knowledge which is the ultimate goal of AI. It extracts representations directly from unsupervised data without human interference.

A key concept underlying Deep Learning methods is distributed representations of the data, in which a large number of possible configurations of the abstract features of the input data are feasible, allowing for a compact representation of each sample and leading to a richer generalization. The number of possible configurations is exponentially related to the number of extracted abstract features. Noting that the observed data was generated through interactions of several known/unknown factors, and thus when a data pattern is obtained through some configurations of learnt factors, additional (unseen) data patterns can likely be described through new configurations of the learnt factors and patterns [5, 24]. Compared to learning based on local generalizations, the number of patterns that can be obtained using a distributed representation scales quickly with the number of learnt factors.

Deep learning algorithms lead to abstract representations because more abstract representations are often constructed based on less abstract ones. An important advantage of more abstract representations is that they can be invariant to the local changes in the input data. Learning such invariant features is an ongoing major goal in pattern recognition (for example learning features that are invariant to the face orientation in a face recognition task). Beyond being invariant such representations can also disentangle the factors of variation in data. The real data used in AI-related tasks mostly arise from complicated interactions of many sources. For example an image is composed of different sources of variations such a light, object shapes, and object materials. The abstract representations provided by deep learning algorithms can separate the different sources of variations in data.

Deep learning algorithms are actually Deep architectures of consecutive layers. Each layer applies a nonlinear transformation on its input and provides a representation in its output. The objective is to learn a complicated and abstract representation of the data in a hierarchical manner by passing the data through multiple transformation layers. The sensory data (for example pixels in an image) is fed to the first layer. Consequently the output of each layer is provided as input to its next layer.

Stacking up the nonlinear transformation layers is the basic idea in deep learning algorithms. The more layers the data goes through in the deep architecture, the more complicated the nonlinear transformations which are constructed. These transformations represent the data, so Deep Learning can be considered as special case of representation learning algorithms which learn representations of the data in a Deep Architecture with multiple levels of representations. The achieved final representation is a highly non-linear function of the input data.

It is important to note that the transformations in the layers of deep architecture are non-linear transformations which try to extract underlying explanatory factors in the data. One cannot use a linear transformation like PCA as the transformation algorithms in the layers of the deep structure because the compositions of linear transformations yield another linear transformation. Therefore, there would be no point in having a deep architecture. For example by providing some face images to the Deep Learning algorithm, at the first layer it can learn the edges in different orientations; in the second layer it composes these edges to learn more complex features like different parts of a face such as lips, noses and eyes. In the third layer it composes these features to learn even more complex feature like face shapes of different persons. These final representations can be used as feature in applications of face recognition. This example is provided to simply explain in an understandable way how a deep learning algorithm finds more abstract and complicated representations of data by composing representations acquired in a hierarchical architecture. However, it must be considered that deep learning algorithms do not necessarily attempt to construct a pre-defined sequence of representations at each layer (such as edges, eyes, faces), but instead more generally perform non-linear transformations in different layers. These transformations tend to disentangle factors of variations in data. Translating this concept to appropriate training criteria is still one of the main open questions in deep learning algorithms [5].

The final representation of data constructed by the deep learning algorithm (output of the final layer) provides useful information from the data which can be used as features in building classifiers, or even can be used for data indexing and other applications which are more efficient when using abstract representations of data rather than high dimensional sensory data.

Learning the parameters in a deep architecture is a difficult optimization task, such as learning the parameters in neural networks with many hidden layers. In 2006 Hinton proposed learning deep architectures in an unsupervised greedy layer-wise learning manner [7]. At the beginning the sensory data is fed as learning data to the first layer. The first layer is then trained based on this data, and the output of the first layer (the first level of learnt representations) is provided as

learning data to the second layer. Such iteration is done until the desired number of layers is obtained. At this point the deep network is trained. The representations learnt on the last layer can be used for different tasks. If the task is a classification task usually another supervised layer is put on top of the last layer and its parameters are learnt (either randomly or by using supervised data and keeping the rest of the network fixed). At the end the whole network is fine-tuned by providing supervised data to it.

Here we explain two fundamental building blocks, unsupervised single layer learning algorithms which are used to construct deeper models: Autoencoders and Restricted Boltzmann Machines (RBMs). These are often employed in tandem to construct stacked Autoencoders [8, 26] and Deep belief networks [7], which are constructed by stacking up Autoencoders and Restricted Boltzmann Machines respectively. Autoencoders, also called autoassociators [27], are networks constructed of 3 layers: input, hidden and output. Autoencoders try to learn some representations of the input in the hidden layer in a way that makes it possible to reconstruct the input in the output layer based on these intermediate representations. Thus, the target output is the input itself. A basic Autoencoder learns its parameters by minimizing the reconstruction error. This minimization is usually done by stochastic gradient descent (much like what is done in Multilayer Perceptron). If the hidden layer is linear and the mean squared error is used as the reconstruction criteria, then the Autoencoder will learn the first k principle components of the data. Alternative strategies are proposed to make Autoencoders nonlinear which are appropriate to build deep networks as well as to extract meaningful representations of data rather than performing just as a dimensionality reduction method. Bengio et al. [5] have called these methods "regularized Autoencoders", and we refer an interested reader to that paper for more details on algorithms.

Another unsupervised single layer learning algorithm which is used as a building block in constructing Deep Belief Networks is the Restricted Boltzmann machine (RBM). RBMs are most likely the most popular version of Boltzmann machine [28]. They contains one visible layer and one hidden layer. The restriction is that there is no interaction between the units of the same layer and the connections are solely between units from different layers. The Contrastive Divergence algorithm [29] has mostly been used to train the Boltzmann machine.

Big Data Analytics

Big Data generally refers to data that exceeds the typical storage, processing, and computing capacity of conventional databases and data analysis techniques. As a resource, Big Data requires tools and methods that can be applied to analyze and extract patterns from large-scale data. The rise of Big Data has been caused by increased data storage capabilities, increased computational processing power, and availability of increased volumes of data, which give organization more data than they have computing resources and technologies to process. In addition to the

obvious great volumes of data, Big Data is also associated with other specific complexities, often referred to as the four Vs: Volume, Variety, Velocity, and Veracity [22, 30, 31]. We note that the aim of this section is not to extensively cover Big Data, but present a brief overview of its key concepts and challenges while keeping in mind that the use of Deep Learning in Big Data Analytics is the focus of this paper.

The unmanageable large Volume of data poses an immediate challenge to conventional computing environments and requires scalable storage and a distributed strategy to data querying and analysis. However, this large Volume of data is also a major positive feature of Big Data. Many companies, such as Facebook, Yahoo, Google, already have large amounts of data and have recently begun tapping into its benefits [21]. A general theme in Big Data systems is that the raw data is increasingly diverse and complex, consisting of largely un-categorized/unsupervised data along with perhaps a small quantity of categorized/supervised data. Working with the Variety among different data representations in a given repository poses unique challenges with Big Data, which requires Big Data preprocessing of unstructured data in order to extract structured/ordered representations of the data for human and/or downstream consumption. In today's data-intensive technology era, data Velocity—the increasing rate at which data is collected and obtained—is just as important as the Volume and Variety characteristics of Big Data. While the possibility of data loss exists with streaming data if it is generally not immediately processed and analyzed, there is the option to save fast-moving data into bulk storage for batch processing at a later time. However, the practical importance of dealing with Velocity associated with Big Data is the quickness of the feedback loop, that is, process of translating data input into useable information. This is especially important in the case of time-sensitive information processing. Some companies such as Twitter, Yahoo, and IBM have developed products that address the analysis of streaming data [22]. Veracity in Big Data deals with the trustworthiness or usefulness of results obtained from data analysis, and brings to light the old adage "Garbage-In-Garbage-Out" for decision making based on Big Data Analytics. As the number of data sources and types increases, sustaining trust in Big Data Analytics presents a practical challenge.

Big Data Analytics faces a number of challenges beyond those implied by the four Vs. While not meant to be an exhaustive list, some key problem areas include: data quality and validation, data cleansing, feature engineering, high-dimensionality and data reduction, data representations and distributed data sources, data sampling, scalability of algorithms, data visualization, parallel and distributed data processing, real-time analysis and decision making, crowdsourcing and semantic input for improved data analysis, tracing and analyzing data provenance, data discovery and integration, parallel and distributed computing, exploratory data analysis and interpretation, integrating heterogenous data, and developing new models for massive data computation.

Applications of Deep Learning in Big Data Analytics

As stated previously, Deep Learning algorithms extract meaningful abstract representations of the raw data through the use of an hierarchical multi-level learning approach, where in a higher-level more abstract and complex representations are learnt based on the less abstract concepts and representations in the lower level(s) of the learning hierarchy. While Deep Learning can be applied to learn from labeled data if it is available in sufficiently large amounts, it is primarily attractive for learning from large amounts of unlabeled/unsupervised data [4, 5, 25], making it attractive for extracting meaningful representations and patterns from Big Data.

Once the hierarchical data abstractions are learnt from unsupervised data with Deep Learning, more conventional discriminative models can be trained with the aid of relatively fewer supervised/labeled data points, where the labeled data is typically obtained through human/expert input. Deep Learning algorithms are shown to perform better at extracting non-local and global relationships and patterns in the data, compared to relatively shallow learning architectures [4]. Other useful characteristics of the learnt abstract representations by Deep Learning include: (1) relatively simple linear models can work effectively with the knowledge obtained from the more complex and more abstract data representations, (2) increased automation of data representation extraction from unsupervised data enables its broad application to different data types, such as image, textural, audio, etc., and (3) relational and semantic knowledge can be obtained at the higher levels of abstraction and representation of the raw data. While there are other useful aspects of Deep Learning based representations of data, the specific characteristics mentioned above are particularly important for Big Data Analytics.

Considering each of the four Vs of Big Data characteristics, i.e., Volume, Variety, Velocity, and Veracity, Deep Learning algorithms and architectures are more aptly suited to address issues related to Volume and Variety of Big Data Analytics. Deep Learning inherently exploits the availability of massive amounts of data, i.e., Volume in Big Data, where algorithms with shallow learning hierarchies fail to explore and understand the higher complexities of data patterns. Moreover, since Deep Learning deals with data abstraction and representations, it is quite likely suited for analyzing raw data presented in different formats and/or from different sources, i.e., Variety in Big Data, and may minimize need for input from human experts to extract features from every new data type observed in Big Data. While presenting different challenges for more conventional data analysis approaches, Big Data Analytics presents an important opportunity for developing novel algorithms and models to address specific issues related to Big Data. Deep Learning concepts provide one such solution venue for data analytics experts and practitioners. For example, the extracted representations by Deep Learning can be considered as a practical source of knowledge for decision-making, semantic indexing, information retrieval, and for other purposes in Big Data Analytics, and in addition, simple linear modeling techniques can be considered for Big Data Analytics when complex data is represented in higher forms of abstraction.

In the remainder of this section, we summarize some important works that have been performed in the field of Deep Learning algorithms and architectures, including semantic indexing, discriminative tasks, and data tagging. Our focus is that by presenting these works in Deep Learning, experts can observe the novel applicability of Deep Learning techniques in Big Data Analytics, particularly since some of the application domains in the works presented involve large scale data. Deep Learning algorithms are applicable to different kinds of input data; however, in this section we focus on its application on image, textual, and audio data.

Semantic Indexing

A key task associated with Big Data Analytics is information retrieval [21]. Efficient storage and retrieval of information is a growing problem in Big Data, particularly since very large-scale quantities of data such as text, image, video, and audio are being collected and made available across various domains, e.g., social networks, security systems, shopping and marketing systems, defense systems, fraud detection, and cyber traffic monitoring. Previous strategies and solutions for information storage and retrieval are challenged by the massive volumes of data and different data representations, both associated with Big Data. In these systems, massive amounts of data are available that needs semantic indexing rather than being stored as data bit strings. Semantic indexing presents the data in a more efficient manner and makes it useful as a source for knowledge discovery and comprehension, for example by making search engines work more quickly and efficiently.

Instead of using raw input for data indexing, Deep Learning can be used to generate high-level abstract data representations which will be used for semantic indexing. These representations can reveal complex associations and factors (especially when the raw input was Big Data), leading to semantic knowledge and understanding. Data representations play an important role in the indexing of data, for example by allowing data points/instances with relatively similar representations to be stored closer to one another in memory, aiding in efficient information retrieval. It should be noted, however, that the high-level abstract data representations need to be meaningful and demonstrate relational and semantic association in order to actually confer a good semantic understanding and comprehension of the input.

While Deep Learning aids in providing a semantic and relational understanding of the data, a vector representation (corresponding to the extracted representations) of data instances would provide faster searching and information retrieval. More specifically, since the learnt complex data representations contain semantic and relational information instead of just raw bit data, they can directly be used for semantic indexing when each data point (for example a given text document) is presented by a vector representation, allowing for a vector-based comparison which is more efficient than comparing instances based directly on raw data. The data

instances that have similar vector representations are likely to have similar semantic meaning. Thus, using vector representations of complex high-level data abstractions for indexing the data makes semantic indexing feasible. In the remainder of this section, we focus on document indexing based on knowledge gained from Deep Learning. However, the general idea of indexing based on data representations obtained from Deep Learning can be extended to other forms of data.

Document (or textual) representation is a key aspect in information retrieval for many domains. The goal of document representation is to create a representation that condenses specific and unique aspects of the document, e.g., document topic. Document retrieval and classification systems are largely based on word counts, representing the number of times each word occurs in the document. Various document retrieval schemas use such a strategy, e.g., TF-IDF [32] and BM25 [33]. Such document representation schemas consider individual words to be dimensions, with different dimensions being independent. In practice, it is often observed that the occurrence of words are highly correlated. Using Deep Learning techniques to extract meaningful data representations makes it possible to obtain semantic features from such high-dimensional textual data, which in turn also leads to the reduction of the dimensions of the document data representations.

Hinton and Salakhutdinov [34] describe a Deep Learning generative model to learn the binary codes for documents. The lowest layer of the Deep Learning network represents the word count vector of the document which accounts as high-dimensional data, while the highest layer represents the learnt binary code of the document. Using 128-bit codes, the authors demonstrate that the binary codes of the documents that are semantically similar lay relatively closer in the Hamming space. The binary code of the documents can then be used for information retrieval. For each query document, its Hamming distance compared to all other documents in the data is computed and the top D similar documents are retrieved. Binary codes require relatively little storage space, and in addition they allow relatively quicker searches by using algorithms such as fast-bit counting to compute the Hamming distance between two binary codes. The authors conclude that using these binary codes for document retrieval is more accurate and faster than semantic-based analysis.

Deep Learning generative models can also be used to produce shorter binary codes by forcing the highest layer in the learning hierarchy to use a relatively small number of variables. These shorter binary codes can then simply be used as memory addresses. One word of memory is used to describe each document in such a way that a small Hammingball around that memory address contains semantically similar documents—such a technique is referred as "semantic hashing" [35]. Using such a strategy, one can perform information retrieval on a very large document set with the retrieval time being independent of the document set size. Techniques such as semantic hashing are quite attractive for information retrieval, because documents that are similar to the query document can be retrieved by finding all the memory addresses that differ from the memory address of the query document by a few bits. The authors demonstrate that "memory hashing" is much faster than locality-sensitive hashing, which is one of the fastest methods among existing

algorithms. In addition, it is shown that by providing a document's binary codes to algorithms such as TF-IDF instead of providing the entire document, a higher level of accuracy can be achieved. While Deep Learning generative models can have a relatively slow learning/training time for producing binary codes for document retrieval, the resulting knowledge yields fast inferences which is one major goal of Big Data Analytics. More specifically, producing the binary code for a new document requires just a few vector matrix computations performing a feed-forward pass through the encoder component of the Deep Learning network architecture.

To learn better representations and abstractions, one can use some supervised data in training the Deep Learning model. Ranzato and Szummer [36] present a study in which parameters of the Deep Learning model are learnt based on both supervised and unsupervised data. The advantages of such a strategy are that there is no need to completely label a large collection of data (as some unlabeled data is expected) and that the model has some prior knowledge (via the supervised data) to capture relevant class/label information in the data. In other words, the model is required to learn data representations that produce good reconstructions of the input in addition to providing good predictions of document class labels. The authors show that for learning compact representations, Deep Learning models are better than shallow learning models. The compact representations are efficient because they require fewer computations when used in indexing, and in addition, also need less storage capacity.

Google's "word2vec" tool is another technique for automated extraction of semantic representations from Big Data. This tool takes a large-scale text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words, upon which the word vector file can be used as features in many Natural Language Processing (NLP) and machine learning applications. Miklov et al. [37] introduce techniques to learn high-quality word vectors from huge datasets with hundreds of millions of words (including some datasets containing 1.6 billion words), and with millions of distinct words in the vocabulary. They focus on artificial neural networks to learn the distributed representation of words. To train the network on such a massive dataset, the models are implemented on top of the large-scale distributed framework "DistBelief" [38]. The authors find that word vectors which are trained on massive amounts of data show subtle semantic relationships between words, such as a city and the country it belongs to—for example, Paris belongs to France and Berlin belongs to Germany. Word vectors with such semantic relationships could be used to improve many existing NLP applications, such as machine translation, information retrieval, and question response systems. For example, in a related work, Miklov et al. [39] demonstrate how word2vec can be applied for natural language translation.

Deep Learning algorithms make it possible to learn complex nonlinear representations between word occurrences, which allow the capture of high-level semantic aspects of the document (which could not normally be learned with linear models). Capturing these complex representations requires massive amounts of data for the input corpus, and producing labeled data from this massive input is a

difficult task. With Deep Learning one can leverage unlabeled documents (unsupervised data) to have access to a much larger amount of input data, using a smaller amount of supervised data to improve the data representations and make them more related to the specific learning and inference tasks. The extracted data representations have been shown to be effective for retrieving documents, making them very useful for search engines.

Similar to textual data, Deep Learning can be used on other kinds of data to extract semantic representations from the input corpus, allowing for semantic indexing of that data. Given the relatively recent emergence of Deep Learning, additional work needs to be done on using its hierarchical learning strategy as a method for semantic indexing of Big Data. A remaining open question is what criteria is used to define "similar" when trying to extract data representations for indexing purposes (recall, data points that are semantically similar will have similar data representations in a specific distance space).

Discriminative Tasks and Semantic Tagging

In performing discriminative tasks in Big Data Analytics one can use Deep Learning algorithms to extract complicated nonlinear features from the raw data, and then use simple linear models to perform discriminative tasks using the extracted features as input. This approach has two advantages: (1) extracting features with Deep Learning adds nonlinearity to the data analysis, associating the discriminative tasks closely to Artificial Intelligence, and (2) applying relatively simple linear analytical models on the extracted features is more computationally efficient, which is important for Big Data Analytics. The problem of developing efficient linear models for Big Data Analytics has been extensively investigated in the literature [21]. Hence, developing nonlinear features from massive amounts of input data allows the data analysts to benefit from the knowledge available through the massive amounts of data, by applying the learnt knowledge to simpler linear models for further analysis. This is an important benefit of using Deep Learning in Big Data Analytics, allowing practitioners to accomplish complicated tasks related to Artificial Intelligence, such as image comprehension, object recognition in images, etc., by using simpler models. Thus discriminative tasks are made relatively easier in Big Data Analytics with the aid of Deep Learning algorithms.

Discriminative analysis in Big Data Analytics can be the primary purpose of the data analysis, or it can be performed to conduct tagging (such as semantic tagging) on the data for the purpose of searching. For example, Li et al. [40] explore the Microsoft Research Audio Video Indexing System (MAVIS) that uses Deep Learning (with Artificial Neural Networks) based speech recognition technology to enable searching of audio and video files with speech. To converting digital audio and video signals into words, MAVIS automatically generates closed captions and keywords that can increase accessibility and discovery of audio and video files with speech content.

Considering the development of the Internet and the explosion of online users in recent years, there has been a very rapid increase in the size of digital image collections. These come from sources such as social networks, global positioning satellites, image sharing systems, medical imaging systems, military surveillance, and security systems. Google has explored and developed systems that provide image searches (e.g., the Google Images search service), including search systems that are only based on the image file name and document contents and do not consider/relate to the image content itself [41, 42]. Towards achieving artificial intelligence in providing improved image searches, practitioners should move beyond just the textual relationships of images, especially since textual representations of images are not always available in massive image collection repositories. Experts should strive towards collecting and organizing these massive image data collections, such that they can be browsed, searched, and retrieved more efficiently. To deal with large scale image data collections, one approach to consider is to automate the process of tagging images and extracting semantic information from the images. Deep Learning presents new frontiers towards constructing complicated representations for image and video data as relatively high levels of abstractions, which can then be used for image annotation and tagging that is useful for image indexing and retrieval. In the context of Big Data Analytics, here Deep Learning would aid in the discriminative task of semantic tagging of data.

Data tagging is another way to semantically index the input data corpus. However, it should not be confused with semantic indexing as discussed in the prior section. In semantic indexing, the focus is on using the Deep Learning abstract representations directly for data indexing purposes. Here the abstract data representations are considered as features for performing the discriminative task of data tagging. This tagging on data can also be used for data indexing as well, but the primary idea here is that Deep Leaning makes it possible to tag massive amounts of data by applying simple linear modeling methods on complicated features that were extracted by Deep Learning algorithms. The remainder of this section focuses largely on some results from using Deep Leaning for discriminative tasks that involve data tagging.

At the ImageNet Computer Vision Competition, Krizhevsky et al. [17] demonstrated an approach using Deep Learning and Convolutional Neural Networks which outperformed other existing approaches for image object recognition. Using the ImageNet dataset, one of the largest for image object recognition, Hinton's team showed the importance of Deep Learning for improving image searching. Dean et al. [38] demonstrated further success on ImageNet by using a similar Deep Learning modeling approach with a large-scale software infrastructure for training an artificial neural network.

Some other approaches have been tried for learning and extracting features from unlabeled image data, include Restricted Boltzmann Machines (RBMs) [7], autoencoders [26], and sparse coding [43]. However, these were only able to extract low-level features, such as edge and blob detection. Deep Learning can also be used to build very high-level features for image detection. For example, Google and Stanford formulated a very large deep neural network that was able to learn very high-level features, such as face detection or cat detection from scratch (without any

priors) by just using unlabeled data [44]. Their work was a large scale investigation on the feasibility of building high-level features with Deep Learning using only unlabeled (unsupervised) data, and clearly demonstrated the benefits of using Deep Learning with unsupervised data. In Google's experimentation, they trained a 9-layered locally connected sparse autoencoder on 10 million 200 × 200 images downloaded randomly from the Internet. The model had 1 billion connections and the training time lasted for 3 days. A computational cluster of 1000 machines and 16,000 cores was used to train the network with model parallelism and asynchronous SGD (Stochastic Gradient Descent). In their experiments they obtained neurons that function like face detectors, cat detectors, and human body detectors, and based on these features their approach also outperformed the state-of-the-art and recognized 22,000 object categories from the ImageNet dataset. This demonstrates the generalization ability of abstract representations extracted by Deep Learning algorithms on new/unseen data, i.e., using features extracted from a given dataset to successfully perform a discriminative task on another dataset. While Google's work involved the question of whether it is possible to build a face feature detector by just using unlabeled data, typically in computer vision labeled images are used to learn useful features [45]. For example, a large collection of face images with a bounding box around the faces can be used to learn a face detector feature. However, traditionally it would require a very large amount of labeled data to find the best features. The scarcity of labeled data in image data collections poses a challenging problem.

There are other Deep Learning works that have explored image tagging. Socher et al. [46] introduce recursive neural networks for predicting a tree structure for images in multiple modalities, and is the first Deep Learning method that achieves very good results on segmentation and annotation of complex image scenes. The recursive neural network architecture is able to predict hierarchical tree structures for scene images, and outperforms other methods based on conditional random fields or a combination of other methods, as well as outperforming other existing methods in segmentation, annotation and scene classification. Socher et al. [46] also show that their algorithm is a natural tool for predicting tree structures by using it to parse natural language sentences. This demonstrates the advantage of Deep Learning as an effective approach for extracting data representations from different varieties of data types. Kumar et al. [47] suggest that recurrent neural networks can be used to construct a meaningful search space via Deep Learning, where the search space can then be used for a designed-based search.

Le et al. [48] demonstrate that Deep Learning can be used for action scene recognition as well as video data tagging, by using an independent variant analysis to learn invariant spatio-temporal features from video data. Their approach outperforms other existing methods when combined with Deep Learning techniques such as stacking and convolution to learn hierarchical representations. Previous works used to adapt hand designed feature for images like SIFT and HOG to the video domain. The Le et al. [48] study shows that extracting features directly from video data is a very important research direction, which can be also generalized to many domains.

Deep Learning has achieved remarkable results in extracting useful features (i.e., representations) for performing discriminative tasks on image and video data, as well as extracting representations from other kinds of data. These discriminative results with Deep Learning are useful for data tagging and information retrieval and can be used in search engines. Thus, the high-level complex data representations obtained by Deep Learning are useful for the application of computationally feasible and relatively simpler linear models for Big Data Analytics. However, there is considerable work that remains for further exploration, including determining appropriate objectives in learning good representations for performing discriminative tasks in Big Data Analytics [5, 25].

Deep Learning Challenges in Big Data Analytics

The prior section focused on emphasizing the applicability and benefits of Deep Learning algorithms for Big Data Analytics. However, certain characteristics associated with Big Data pose challenges for modifying and adapting Deep Learning to address those issues. This section presents some areas of Big Data where Deep Learning needs further exploration, specifically, learning with streaming data, dealing with high-dimensional data, scalability of models, and distributed computing.

Incremental Learning for Non-stationary Data

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 [49], denoising autoencoders [50], and deep belief networks [51].

Zhou et al. [49] describe how a Deep Learning algorithm can be used for incremental feature learning on very large datasets, employing denoising autoencoders [50]. 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. Deep Learning algorithms in general use hidden layers to contribute towards the extraction of features or data representations. In a denoising autoencoder, there is one hidden layer which extracts features, with the number of nodes in this hidden layer initially being the same as the number of features that would be extracted. Incrementally, the samples that do not conform to the given objective function (for example, their classification error is more than a threshold, or their reconstruction error is high) are collected and are used for adding

new nodes to the hidden layer, with these new nodes being initialized based on those samples. Subsequently, incoming new data samples are used to jointly retrain all the features. This incremental feature learning and mapping can improve the discriminative or generative objective function; however, monotonically adding features can lead to having a lot of redundant features and overfitting of data. Consequently, similar features are merged to produce a more compact set of features. Zhou et al. [49] 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. Incremental feature learning and extraction can be generalized for other Deep Learning algorithms, such as RBM [7], and makes it possible to adapt to new incoming stream of an online large-scale data. Moreover, it avoids expensive cross-validation analysis in selecting the number of features in large-scale datasets.

Calandra et al. [51] introduce adaptive deep belief networks which demonstrates 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.

The targeted works presented in this section provide empirical support to further explore and develop novel Deep Learning algorithms and architectures for analyzing large-scale, fast moving streaming data, as is encountered in some Big Data application domains such as social media feeds, marketing and financial data feeds, web click stream data, operational logs, and metering data. For example, Amazon Kinesis is a managed service designed to handle real-time streaming of Big Data—though it is not based on the Deep Learning approach.

High-Dimensional Data

Some Deep Learning algorithms can become prohibitively computationally-expensive when dealing with high-dimensional data, such as images, likely due to the often slow learning process associated with a deep layered hierarchy of learning data abstractions and representations from a lower-level layer to a higher-level layer. That is to say, these Deep Learning algorithms can be stymied when working with Big Data that exhibits large Volume, one of the four Vs associated with Big Data Analytics. A high-dimensional data source contributes heavily to the volume of the raw data, in addition to complicating learning from the data.

Chen et al. [52] 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 [17, 26]. 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.

The application of Deep Learning algorithms for Big Data Analytics involving high dimensional data remains largely unexplored, and warrants development of Deep Learning based solutions that either adapt approaches similar to the ones presented above or develop novel solutions for addressing the high-dimensionality found in some Big Data domains.

Large-Scale Models

From a computation and analytics point of view, how do we scale the recent successes of Deep Learning to much larger-scale models and massive datasets? Empirical results have demonstrated the effectiveness of large-scale models [53–55], with particular focus on models with a very large number of model parameters which are able to extract more complicated features and representations [38, 56].

Dean et al. [38] consider the problem of training a Deep Learning neural network with billions of parameters using tens of thousands of CPU cores, in the context of speech recognition and computer vision. A software framework, DistBelief, is developed that can utilize computing clusters with thousands of machines to train large-scale models. The framework supports model parallelism both within a machine (via multithreading) and across machines (via message passing), with the details of parallelism, synchronization, and communication managed by DistBelief. In addition, the framework also supports data parallelism, where multiple replicas of a model are used to optimize a single objective. In order to make large-scale distributed training possible an asynchronous SGD as well as a distributed batch optimization procedure is developed that includes a distributed implementation of L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno, a quasi-Newton method for unconstrained optimization). The primary idea is to train multiple

versions of the model in parallel, each running on a different node in the network and analyzing different subsets of data. The authors report that in addition to accelerating the training of conventional sized models, their framework can also train models that are larger than could be contemplated otherwise. Moreover, while the framework focuses on training large-scale neural networks, the underlying algorithms are applicable to other gradient-based learning techniques. It should be noted, however, that the extensive computational resources utilized by DistBelief are generally unavailable to a larger audience. Coates et al.

Coates et al. [56] leverage the relatively inexpensive computing power of a cluster of GPU servers. More specifically, they develop their own system (using neural networks) based on Commodity Off-The-Shelf High Performance Computing (COTS HPC) technology and introduce a high-speed communication infrastructure to coordinate distributed computations. The system is able to train 1 billion parameter networks on just 3 machines in a couple of days, and it can scale to networks with over 11 billion parameters using just 16 machines and where the scalability is comparable to that of DistBelief. In comparison to the computational resources used by DistBelief, the distributed system network based on COTS HPC is more generally available to a larger audience, making it a reasonable alternative for other Deep Learning experts exploring large-scale models.

Large-scale Deep Learning models are quite suited to handle massive volumes of input associated with Big Data, and as demonstrated in the above works they are also better at learning complex data patterns from large volumes of data. Determining the optimal number of model parameters in such large-scale models and improving their computational practicality pose challenges in Deep Learning for Big Data Analytics. In addition to the problem of handling massive volumes of data, large-scale Deep Learning models for Big Data Analytics also have to contend with other Big Data problems, such as domain adaptation (see next section) and streaming data. This lends to the need for further innovations in large-scale models for Deep Learning algorithms and architectures.

Future Work on Deep Learning in Big Data Analytics

In the prior sections, we discussed some recent applications of Deep Learning algorithms for Big Data Analytics, as well as identified some areas where Deep Learning research needs further exploration to address specific data analysis problems observed in Big Data. Considering the low-maturity of Deep Learning, we note that considerable work remains to be done. In this section, we discuss our insights on some remaining questions in Deep Learning research, especially on work needed for improving machine learning and the formulation of the high-level abstractions and data representations for Big Data.

An important problem is whether to utilize the entire Big Data input corpus available when analyzing data with Deep Learning algorithms. The general focus is to apply Deep Learning algorithms to train the high-level data representation

patterns based on a portion of the available input corpus, and then utilize the remaining input corpus with the learnt patterns for extracting the data abstractions and representations. In the context of this problem, a question to explore is what volume of input data is generally necessary to train useful (good) data representations by Deep Learning algorithms which can then be generalized for new data in the specific Big Data application domain.

Upon further exploring the above problem, we recall the Variety characteristic of Big Data Analytics, which focuses on the variation of the input data types and domains in Big Data. Here, by considering the shift between the input data source (for training the representations) and the target data source (for generalizing the representations), the problem becomes one of domain adaptation for Deep Learning in Big Data Analytics. Domain adaptation during learning is an important focus of study in Deep Learning [57, 58], where the distribution of the training data (from which the representations are learnt) is different from the distribution of the test data (on which the learnt representations are deployed).

Glorot et al. [57] demonstrate that Deep Learning is able to discover intermediate data representations in a hierarchical learning manner, and that these representations are meaningful to, and can be shared among, different domains. In their work, a stacked denoising autoencoder is initially used to learn features and patterns from unlabeled data obtained from different source domains. Subsequently, a support vector machine (SVM) algorithm utilizes the learnt features and patterns for application on labeled data from a given source domain, resulting in a linear classification model that outperforms other methods. This domain adaptation study is successfully applied on a large industrial strength dataset consisting of 22 source domains. However, it should be noted that their study does not explicitly encode the distribution shift of the data between the source domain and the target domains. Chopra et al. [58] propose a Deep Learning model (based on neural networks) for domain adaptation which strives to learn a useful (for prediction purposes) representation of the unsupervised data by taking into consideration information available from the distribution shift between the training and test data. The focus is to hierarchically learn multiple intermediate representations along an interpolating path between the training and testing domains. In the context of object recognition, their study demonstrates an improvement over other methods. The two studies presented above raise the question about how to increase the generalization capacity of Deep Learning data representations and patterns, noting that the ability to generalize learnt patterns is an important requirement in Big Data Analytics where often there is a distribution shift between the input domain and the target domain.

Another key area of interest would be to explore the question of what criteria is necessary and should be defined for allowing the extracted data representations to provide useful semantic meaning to the Big Data. Earlier, we discussed some studies that utilize the data representations extracted through Deep Learning for semantic indexing. Bengio et al. [5] present some characteristics of what constitutes good data representations for performing discriminative tasks, and point to the open question regarding the definition of the criteria for learning good data representations in Deep Learning. Compared to more conventional learning algorithms where

misclassification error is generally used as an important criterion for model training and learning patterns, defining a corresponding criteria for training Deep Learning algorithms with Big Data is unsuitable since most Big Data Analytics involve learning from largely unsupervised data. While availability of supervised data in some Big Data domains can be helpful, the question of defining the criteria for obtaining good data abstractions and representations still remains largely unexplored in Big Data Analytics. Moreover, the question of defining the criteria required for extracting good data representations leads to the question of what would constitute a good data representation that is effective for semantic indexing and/or data tagging.

In some Big Data domains, the input corpus consists of a mix of both labeled and unlabeled data, e.g., cyber security [59], fraud detection [60], and computer vision [45]. In such cases, Deep Learning algorithms can incorporate semi-supervised training methods towards the goal of defining criteria for good data representation learning. For example, following learning representations and patterns from the unlabeled/unsupervised data, the available labeled/supervised data can be exploited to further tune and improve the learnt representations and patterns for a specific analytics task, including semantic indexing or discriminative modeling. A variation of semi-supervised learning in data mining, active learning methods could also be applicable towards obtaining improved data representations where input from crowd sourcing or human experts can be used to obtain labels for some data samples which can then be used to better tune and improve the learnt data representations.

Conclusion

In contrast to more conventional machine learning and feature engineering algorithms, Deep Learning has an advantage of potentially providing a solution to address the data analysis and learning problems found in massive volumes of input data. More specifically, it aids in automatically extracting complex data representations from large volumes of unsupervised data. This makes it a valuable tool for Big Data Analytics, which involves data analysis from very large collections of raw data that is generally unsupervised and un-categorized. The hierarchical learning and extraction of different levels of complex, data abstractions in Deep Learning provides a certain degree of simplification for Big Data Analytics tasks, especially for analyzing massive volumes of data, semantic indexing, data tagging, information retrieval, and discriminative tasks such a classification and prediction.

In the context of discussing key works in the literature and providing our insights on those specific topics, this study focused on two important areas related to Deep Learning and Big Data: (1) the application of Deep Learning algorithms and architectures for Big Data Analytics, and (2) how certain characteristics and issues of Big Data Analytics pose unique challenges towards adapting Deep Learning algorithms for those problems. A targeted survey of important literature in Deep

Learning research and application to different domains is presented in the paper as a means to identify how Deep Learning can be used for different purposes in Big Data Analytics.

The low-maturity of the Deep Learning field warrants extensive further research. In particular, more work is necessary on how we can adapt Deep Learning algorithms for problems associated with Big Data, including high dimensionality, streaming data analysis, scalability of Deep Learning models, improved formulation of data abstractions, distributed computing, semantic indexing, data tagging, information retrieval, criteria for extracting good data representations, and domain adaptation. Future works should focus on addressing one or more of these problems often seen in Big Data, thus contributing to the Deep Learning and Big Data Analytics research corpus.

Competing Interests The authors declare that they have no competing interests.

Authors' Contributions MMN performed the primary literature review and analysis for this work, and also drafted the manuscript. RW and NS worked with MMN to develop the article's framework and focus. TMK, FV and EM introduced this topic to MMN and TMK coordinated with the other authors to complete and finalize this work. All authors read and approved the final manuscript.

References

1. Domingos P. A few useful things to know about machine learning. Commun ACM. 2012;55 (10):78–87.
2. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005. IEEE, vol. 1. 2005;886–93.
3. Lowe DG. Object recognition from local scale-invariant features. In: Computer Vision, 1999. The Proceedings of the seventh IEEE international conference on IEEE computer society, vol. 2. 1999. p. 1150–7.
4. Bengio Y, LeCun Y. Scaling learning algorithms towards, AI. In: Bottou L, Chapelle O, DeCoste D, Weston J, editors. Large scale kernel machines, vol. 34. Cambridge: MIT Press; 2007. p. 321–60. http://www.iro.umontreal.ca/∼lisa/pointeurs/bengio+lecun_chapter2007.pdf.
5. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35(8):1798–828. doi:10.1109/TPAMI.2013.50.
6. Arel I, Rose DC, Karnowski TP. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. IEEE Comput Intell. 2010;5:13–8.
7. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18(7):1527–54.
8. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. 2007;19.
9. Larochelle H, Bengio Y, Louradour J, Lamblin P. Exploring strategies for training deep neural networks. J Mach Learn Res. 2009;10:1–40.
10. Salakhutdinov R, Hinton GE. Deep boltzmann machines. In: International conference on artificial intelligence and statistics. JMLR.org. 2009. p. 448–55.

11. Goodfellow I, Lee H, Le QV, Saxe A, Ng AY. Measuring invariances in deep networks. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2009. p. 646–54.
12. Dahl G, Ranzato M, Mohamed A-R, Hinton GE. Phone recognition with the mean-covariance restricted boltzmann machine. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2010. p. 469–77.
13. Hinton G, Deng L, Yu D, Mohamed A-R, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath T, Dahl G, Kingsbury B. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. Signal Process Mag IEEE. 2012;29 (6):82–97.
14. Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks. In: INTERSPEECH. ISCA. 2011. p. 437–40.
15. Mohamed A-R, Dahl GE, Hinton G. Acoustic modeling using deep belief networks. IEEE Trans Audio Speech Lang Process. 2012;20(1):14–22.
16. Dahl GE, Yu D, Deng L, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans Audio Speech Lang Process. 2012;20(1):30–42.
17. Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, vol. 25. Red Hook: Curran Associates, Inc.; 2012. p. 1106–14.
18. Mikolov T, Deoras A, Kombrink S, Burget L, Cernocky J. Empirical evaluation and combination of advanced language modeling techniques. In: INTERSPEECH. ISCA. 2011. p. 605–8.
19. Socher R, Huang EH, Pennin J, Manning CD, Ng A. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2011. p. 801–9.
20. Bordes A, Glorot X, Weston J, Bengio Y. Joint learning of words and meaning representations for open-text semantic parsing. In: International conference on artificial intelligence and statistics. JMLR.org. 2012. p. 127–35.
21. National Research Council. Frontiers in Massive Data Analysis. Washington, DC: The National Academies Press. 2013. http://www.nap.edu/openbook.php?record_id=18374.
22. Dumbill E. What is Big Data? An introduction to the big data landscape. In: Strata 2012: making data work. O'Reilly, Santa Clara, CA O'Reilly. 2012.
23. Khoshgoftaar TM. Overcoming big data challenges. In: Proceedings of the 25th international conference on software engineering and knowledge engineering. Boston. ICSE. Invited Keynote Speaker. 2013.
24. Bengio Y. Learning deep architectures for AI. Hanover: Now Publishers Inc.; 2009.
25. Bengio Y. Deep learning of representations: looking forward. Proceedings of the 1st international conference on statistical language and speech processing. SLSP'13. Tarragona: Springer; 2013. p. 1–37. doi:10.1007/978-3-642-39593-2_1.
26. Hinton GE, Salakhutdinov RR (Science) Reducing the dimensionality of data with neural networks 313(5786):504–7.
27. Hinton GE, Zemel RS. Autoencoders, minimum description length, and helmholtz free energy. Adv Neural Inf Process Syst. 1994;6:3–10.
28. Smolensky P. Information processing in dynamical systems: foundations of harmony theory. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. Cambridge: MIT Press; 1986. p. 194–281.
29. Hinton GE. Training products of experts by minimizing contrastive divergence. Neural Comput. 2002;14(8):1771–800.
30. Garshol LM. Introduction to big data/machine learning. Online slide show. 2013. http://www.slideshare.net/larsga/introduction-to-big-datamachine-learning.
31. Grobelnik M. Big Data tutorial. European Data Forum. 2013. http://www.slideshare.net/EUDataForum/edf2013-bigdatatutorialmarkogrobelnik?related=1.

32. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manag. 1988;24(5):513–23.
33. Robertson SE, Walker S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval. New York: Springer; 1994. p. 232–41.
34. Hinton G, Salakhutdinov R. Discovering binary codes for documents by learning deep generative models. Topics Cogn Sci. 2011;3(1):74–91.
35. Salakhutdinov R, Hinton G. Semantic hashing. Int J Approx Reason. 2009;50(7):969–78.
36. Ranzato M, Szummer M. Semi-supervised learning of compact document representations with deep networks. In: Proceedings of the 25th international conference on machine learning. ACM. 2008. p. 792–9.
37. Mikolov T, Chen K, Dean J. Efficient estimation of word representations in vector space. CoRR: Comput Res Repos. 2013;1–12. abs/1301.3781.
38. Dean J, Corrado G, Monga R, Chen K, Devin M, Le Q, Mao M, Ranzato M, Senior A, Tucker P, Yang K, Ng A. Large scale distributed deep networks. In: Bartlett P, Pereira FCN, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in neural information processing systems, vol. 25. 2012. p. 1232–40. http://books.nips.cc/papers/files/nips25/NIPS2012_0598.pdf.
39. Mikolov T, Le QV, Sutskever I. Exploiting similarities among languages for machine translation. CoRR: Comput Res Repos. 2013;1–10. abs/1309.4168.
40. Li G, Zhu H, Cheng G, Thambiratnam K, Chitsaz B, Yu D, Seide F. Context-dependent deep neural networks for audio indexing of real-life data. In: Spoken language technology workshop (SLT), 2012 IEEE. IEEE. 2012. p. 143–8.
41. Zipern A. A quick way to search for images on the web. The New York Times. News Watch Article. 2001. http://www.nytimes.com/2001/07/12/technology/news-watch-a-quick-way-to-search-for-images-on-the-web.html.
42. Cusumano MA. Google: what it is and what it is not. Commun ACM Med Image Model. 2005;48(2):15–7. doi:10.1145/1042091.1042107.
43. Lee H, Battle A, Raina R, Ng A. Efficient sparse coding algorithms. Advances in neural information processing systems. Cambridge: MIT Press; 2006. p. 801–8.
44. Le Q, Ranzato M, Monga R, Devin M, Chen K, Corrado G, Dean J, Ng A. Building high-level features using large scale unsupervised learning. In: Proceeding of the 29th international conference in machine learning. Edingburgh. 2012.
45. Freytag A, Rodner E, Bodesheim P, Denzler J. Labeling examples that matter: relevance-based active learning with gaussian processes. In: 35th German conference on pattern recognition (GCPR). Germany: Saarland University and Max-Planck-Institute for Informatics; 2013. p. 282–91.
46. Socher R, Lin CC, Ng A, Manning C. Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th international conference on machine learning. Madison: Omnipress; 2011. p. 129–36.
47. Kumar R, Talton JO, Ahmad S, Klemmer SR. Data-driven web design. In: Proceedings of the 29th international conference on machine learning..2012. icml.cc/Omnipress.
48. Le QV, Zou WY, Yeung SY, Ng AY. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: IEEE conference on computer vision and pattern recognition (CVPR) 2011 IEEE. 2011. p. 3361–8.
49. Zhou G, Sohn K, Lee H. Online incremental feature learning with denoising autoencoders. In: International conference on artificial intelligence and statistics. JMLR.org. 2012. p. 1453–61.
50. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning. ACM. 2008. p. 1096–103.
51. Calandra R, Raiko T, Deisenroth MP, Pouzols FM. Learning deep belief networks from non-stationary streams. Artificial neural networks and machine learning–ICANN 2012. Berlin: Springer; 2012. p. 379–86.

52. Chen M, Xu ZE, Weinberger KQ, Sha F. Marginalized denoising autoencoders for domain adaptation. In: Proceeding of the 29th international conference in machine learning. Edingburgh; 2012.
53. Coates A, Ng A. The importance of encoding versus training with sparse coding and vector quantization. In: Proceedings of the 28th international conference on machine learning. Madison: Omnipress; 2011. p. 921–8.
54. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R. Improving neural networks by preventing co-adaptation of feature detectors. CoRR: Comput Res Repos. 2012;1–18. abs/1207.0580.
55. Goodfellow IJ, Warde-Farley D, Mirza M, Courville A, Bengio Y. Maxout networks. In: Proceeding of the 30th international conference in machine learning. Atlanta. 2013.
56. Coates A, Huval B, Wang T, Wu D, Catanzaro B, Andrew N. Deep learning with Cots HPC systems. In: Proceedings of the 30th international conference on machine learning; 2013. p. 1337–45.
57. Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). 2011. p. 513–20.
58. Chopra S, Balakrishnan S, Gopalan R. Dlid: deep learning for domain adaptation by interpolating between domains. In: Workshop on challenges in representation learning, proceedings of the 30th international conference on machine learning. Atlanta. 2013.
59. Suthaharan S. Big data classification: problems and challenges in network intrusion prediction with machine learning. ACM sigmetrics: Big Data analytics workshop. Pittsburgh: ACM; 2013.
60. Wang W, Lu D, Zhou X, Zhang B, Mu J. Statistical wavelet-based anomaly detection in big data with compressive sensing. EURASIP J Wireless Commun Netw. 2013:269. http://www.bibsonomy.org/bibtex/25e432dc7230087ab1cdc65925be6d4cb/dblp.