Optimization Of Deep Learning Techniques For Big Data

Lavish Thomas
MSc in Big Data Analytics and Artificial Intelligence, Letterkenny Institute of Technology
lavishthomas@gmail.com

Abstract—This paper provides benchmarking for various optimization techniques available for the deep neural networks which can be applied to the big data systems. Initially, the big data ecosystem is segmented into various aspects for identifying the challenges in each one of them. Then, various deep neural networks are presented after which the optimization techniques are listed to present the benefits of each method. Furthermore, the benchmarking criteria are presented for based on big data requirements, deep learning features and system performances. Subsequently, the optimization techniques are compared based on the benchmarks defined. In addition, the application arenas are explained for identifying the requirements for optimizations in each field. Afterwards, a summarizing report of applications effectiveness of each optimization technique is presented in a table format. Finally, the challenges in the field of big data analytics with deep learning is identified and an approximate future roadmap is presented.

Index Terms—Applications, Big Data, Deep Learning, Neural Networks, Optimization Techniques

I. Introduction

THIS research is carried out in order to bridge the gap of collective knowledge about a particular area of deep learning; a comparative study on optimization techniques which can be applied to deep learning systems implemented in the big data environment. Deep learning is gaining more popularity than ever before as a big data tool. Deep learning is a subset of the area called Artificial Intelligence, where tasks can be solved without explicitly programming the logic into the machine. This has created a broad line of innovative products and services which can solve complex tasks which were not coming under the capabilities of machines such as face detection, fraud analysis etc.

Initially in this paper, surveys conducted in the field of big data and deep learning is evaluated and distinguishing factors of this project are highlighted. Afterwards, a brief description of big data and deep learning is provided. Further, benchmarking criteria are presented based on deep learning, big data and system requirements. Subsequently, the optimization techniques will be compared to the various factors presented in Section VI (Benchmarks). Furthermore, the challenges faced in the integration of deep learning into big data systems is presented. Finally, the future roadmap of this area of study is discussed which will give insights to where are bottlenecks in the Big Data-Deep Learning pipeline.

II. Related works and Novelty

The concept of deep learning in the context of big data is relatively new which is gaining traction in the last few years after the immense growth of cloud technologies. Therefore, the literature work is not vast in this field. There have been two major surveys which have been done related to the deep learning in big data. In a survey done by Gheisari et al [1], a study of machine learning techniques into the big data has been presented along with a brief discussion of deep neural networks. Another survey of deep learning in big data was presented by Rani et al [2] which discusses the applications of deep learning and big data. A comparative study has been carried out by Jan et al [3] about how the deep neural network is affected by the large volume of data in their paper 'Deep Learning in Big Data Analytics: A comparative study'.

This project presents a specific topic of 'Deep Learning', that is 'Optimization Techniques' and the methods compared based on big data, deep learning, system and application requirements. Along with that, unlike the surveys mentioned, this survey will not take other Machine learning techniques into consideration other than deep neural networks.

III. Introduction to Big data

Big data is the keyword, which is driving the technology industry for the last decade. Definition of "Big Data" is ambiguous, which differs from simply a huge amount of data to entropy of the data. Most widely accepted properties for a technology to be considered as big data technology is the ability to deal with these three parameters [2] of the data: 1) Volume 2) Velocity 3) Variety. That is, the technologies which were developed to deal with the increasing Volume, Velocity and Variety of data in the current IoT era is called big data.

Under the supervision of Dr. Shagufta Henna, Letterkenny Institute of Technology, Letterkenny, CO. Donegal

A. Segmentation of tools and techniques for big data Ecosystem

In a big data environment, we have 3 basic segments which have its own challenges and solutions. That is, 1) Collection points 2) Storage techniques 3) Processing. As illustrated in Figure 1, the big data environment is a solution designed for the sustainability of a much wider ecosystem including IoT, Edge and Cloud Infrastructure [4].

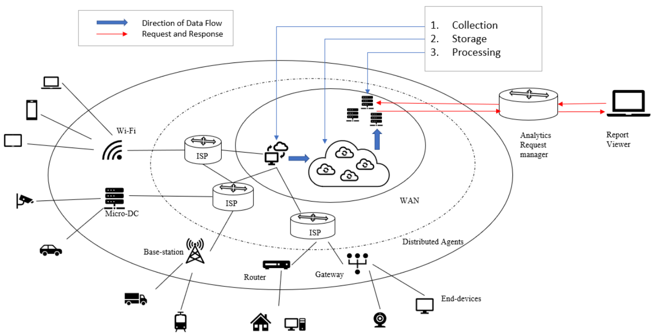

Fig. 1: Big data ecosystem [5]

All segments have to deal with all the aspects of big data: That is, 1) Volume 2) Velocity and 3) Variety. Based on these the expectations and challenges of each segment with respect to each aspect are described in Table I. These challenges will be later used in section VII and IX to address how each area could benefit from the optimization techniques under study.

IV. Deep learning

Deep learning is a subset of machine learning techniques which is a further subset of Artificial Intelligence. The reason this technique is gaining popularity is that it has removed the human intervention to a greater extent than the conventional machine learning techniques. deep neural networks are based on human neural cells and their schema of connections inside the brain. The DNN leverages the concept of an artificial neural network. neural networks are essentially an interconnected network of artificial perceptron which is modelled after the human neuron cells and their connecting behaviour. Neuron cells are connected to each other through weighted connections with multiple inputs and a single output.

A. Types of neural networks

Most implementations of the neural network consist of multiple layers of neurons and interconnections through which the data is propagated to infer insights. The DL model usually follows a 3-layer approach [6]. The layers work on weighted connections, each connection is given a random weight at first and then adjusted based on the deviation from the expected output in the training stage. The input layer processes the input data and generates the outputs required based on the assigned weights in each connection. There can be more than one hidden layer (middle layers) based on the complexity of the applications. Advantage of the DNN over the other Machine learning Models is that it can process high-level features because of the abstract layers. The various neural networks are:

1) Multilayer perceptron:: It is one of the simplest DNN models. The architecture is simple, where neurons in each layer are fully connected to the neurons in the adjacent layer. Due to this, the model is prone to overfitting of the data and redundancies in the data representation leading to inefficient memory usage [6].

2) Convolution neural network:: Using convolution operations, Convolutional neural networks can extract simple features. It is best suited for computer vision programs where identification of the features can be location independent. It is also very helpful in deriving the hierarchal relations between the data. It is very useful in classification and detection programs [6].

3) Recurrent neural network:: For sequential data feeding, this type of DNNs is used in order to resolve the issue of time-series. Internal memory is used in each neuron cells in order to retain the previous state of the cell. These are used in natural language processing and language translation services where contextual meaning is critical. The stock market is also a benefactor of the RNN networks [6].

4) Deep belief networks:: By combining multiple Restricted Boltzmann Machine, a model which can predict the probabilities are created. This is a powerful tool which can be trained based on maximum probability where hidden units act as intermediate units between multiple networks [7].

These types of neural networks will be discussed in various applications of big data systems with deep learning.

B. How deep learning is useful in big data

The deep learning methods are very data-centric in nature and the big data systems are designed to handle the same. This creates a perfect harmony to deploy the deep neural network into the big data applications. Various applications like medical diagnosis, stock market analysis, weather predictions have gained a substantial increase in the accuracy by implementing deep learning techniques into the respective big data applications. The increase in IoT based data has created a demand in techniques which do not have correlative factors between all types of data. The deep learning techniques have the capabilities to cater to this requirement, it can create feature space which is not fathomable for a human feature extraction process.

V. Deep learning optimization techniques

The deep learning optimization techniques are the core of this survey where it will introduce varies deep learning optimization techniques

A. Gradient descent

In most cases, the optimization of the parameters for the deep neural networks is done using the method of

Aspect	Collection of data	Storage of Data	Processing of Data
Volume	Devices and Protocols which can collect a huge amount of data parallelly. For example, weather sensors	Storage systems which can scale up infinitely (at least in theory) is required without compromise on reading and writing speed.	CPU's with high processing powers and memory are used in parallel in order process high volume of data.
Velocity	Devices and Protocols which can collect data at a high rate. For example, video data	With high volume, there should not be a bottleneck at writing as the conventional ACID based transactional data storage. That is, parallel writing should be enabled in the system.	For real-time big data applications, different optimization technologies like in-memory computing like SPARK is used.
Variety	Agents and probes which collect data from various sources like web spiders for search engines. The auto-protocol convertor is getting build for IoT systems which can make use of advantages of various physical devices.	Pre-defined data types are not the only data expected, so a universal system which can store almost all kinds of digital data should be deployed.	Software programs which can read data without predefined types have to be used. Loosely typed languages like JavaScript, Scala is getting more traction. JSON, XML formats are becoming the key in handling unstructured data.
Tools	Kafka, APACHE STROM, SQL Stream [3]	HDFS, CASSANDRA, HBASE, MongoDB [3]	Map-Reduce, Spark, KNIME [3]

TABLE I: Segmentation of big data ecosystem based on aspects

gradient descent. Hence, the learning speed and accuracy of the deep learning system can be optimized by exploring various gradient descent algorithms. Gradient descent [8] is an iterative algorithm, which finds the lowest point of the cost function by selecting an arbitrary point in the function first, then it calculates the gradient using the derivative at that point. In multi-feature problems, the partial derivative is used of each parameter is used in order to calculate the gradient. Afterwards, based on the learning rate defined as hyperparameter and current gradient, the step size is calculated. Finally, the new parameters are calculated using the step function described. This process is continued in each epoch until the gradient converges to zero. This method is not at all suitable for big data therefore it will not be compared with other methods. The methods are explained in order to present how other variations of gradient descent improves upon the normal gradient descent.

B. Stochastic gradient descent

The normal gradient descent is very inefficient in terms of the computing complexity because of the iterative process which scavenge through each data point [8] The stochastic gradient descent makes uses of the high probability properties for the selection of random points. In the training phase, as described in the section V-A the normal gradient descent selects the datapoints sequentially whereas the stochastic gradient descent selects random data points from the various segmented datasets [9]. This method has proved to increase the convergence of gradient to zero at a much faster rate than the normal gradient descent. Variations of the stochastic gradient descents have been introduced which have further increased the performance of the deep learning systems.

C. Downpour SGD

As per Dean et al [10], the stochastic gradient descent is inefficient to be applied to large datasets, therefore they have introduced a new method called Downpour SCD, which is variant of the asynchronous stochastic gradient descent optimization techniques. The overall approach is analogous to a distributed computing model, where the data is split into several chunks and parallelly trained in various models. The trained models communicate with each other using a central parameter storage server. The central model server will be having the parameter values corresponding to the various feature sections of the model, as a replica of the various nodes processing the respective training data sets. The model is claimed to be asynchronous at two levels [10]: one is that, the model replicas are trained independently on various computing nodes, second is that the model parameters are being segregated the in the parameter served are independently updated of each other. Downpour SCD brings other various advantages over the synchronous gradient descent algorithms such as fault-tolerance, redundancy, and improved randomness. The fault-tolerance is achieved because of the distributed and asynchronous nature of the method. Because of the segmentation of the dataset and independency of the randomness introduced, the convergence speed is improved which leads to the training time optimization.

D. Sandblaster L-BFGS

The distributed deep neural network increases the communication overhead especially with the large datasets. In other words, the speed of training will degrade as the number parameter increases. The Sandblaster algorithm is designed in order to address this issue. Contradictory to the Downpour SCD strategy the central server is replaced with a coordinator node as the primary manager of the deep learning system [10]. Unlike the central server, the coordinator will not have direct access to the parameters of the independent model servers, instead the operations performed are extracted from each individual node which will be shared across. This will avoid the overhead of passing the large parameter set across the network which enables the system to train larger deep neural networks. The L-BFGS method work in analogous to the MapReduce method of 'back up task' where when some nodes perform slower, the faster nodes are given the additional task and the coordinator accepts the result of the nodes whichever finishes first. This Sandblaster has improved upon the Downpour [10] in terms of network efficiency by sending the updated gradients instead of the entire parameters.

E. Stochastic diagonal approximate greatest descent

In the paper, 'Vanishing Gradient Mitigation with deep learning neural network optimization' by Tan et al [11] the repercussions of using the normal gradient descent is mitigated by employing long-term optimal trajectories. It is a numerical optimization technique where a search region is created which hovers over the error surfaces at different phases of the neural network training. The weight update rule for SDAGD algorithm is written as

$$W_{k+1} = W_k + [_{\mu k} J + H(W_k)]^{-1}\ _g(W_k)$$

where µk is the relative step length, J is the all-ones matrix and $H(W_k)$ is the truncated Hessian matrix. Additionally, the stochastic diagonal approximate greatest descent has two variants of Hessian approximations, i.e. (a) drop off diagonal terms of Hessian with respect to weights and (b) apply truncated Hessian approximation by ignoring higher-order differential terms. Both saturated and unsaturated activations functions were evaluated using the experiment research [11] and both have shown the mitigation of the issue of vanishing gradient. The misclassification issues also have reduced compared to the normal stochastic gradient descent.

F. Inconsistent stochastic gradient descent (ISGD)

Normal stochastic gradient descent is derived using the randomization of batch and by computing a noisy gradient. The weights are updated in an even manner for every batch. SGD method ignores the variance in training dynamics which is induced by intrinsic image difference and sampling bias. These shortcomings are being addressed in the inconsistent stochastic gradient descent [12]. The resources required for the training are dynamically modulated based on the loss function value in each epoch. The mean of the loss function for a batch is reduced in a stochastic way. A dynamic upper control limit is used to find the value of the loss function in real-time. It uses a penalty system to regularize the drastic parameter changes. The training is accelerated by encapsulating the ISGD with an in-process batch and adding the improvement using the additional gradient updates [12].

G. Averaged stochastic gradient descent (ASGD)

In the normal stochastic gradient descent, the cross-entropy criterion is used to train the deep neural network. This requires multiple passes of the data, that is, epochs to have good accuracy for the model. This makes the stochastic gradient descent an unviable option for the large networks, therefore, averaged stochastic gradient descent is proposed which lowers the computational requirement for the same. The primary aim is to reach the asymptotic region of the loss function by the single pass of the training data. The currently used version of the averaged stochastic gradient descent uses the following averaging procedure [13].

$$\theta_{t+1} = \theta_t - \gamma_t \nabla L(\theta_t)$$
$$\theta_{t+1} = (1 - \eta_t)\theta_t + \eta_t \theta_{t+1}$$

where η_t is the rate of averaging. $\eta_t = 0.01$ for all the experiments described in this paper. The implementation of ASGD has decreased the number of epochs required in the training time of the neural network. It is also observed in the experiment that the learning rate scheduling algorithm plays a vital role and therefore it has to be carefully chosen which will result in a faster convergence [13]. The size of the mini-batch also has an important role where it decreases the overall training time required.

H. Guided stochastic gradient descent (GSGD)

As described in section V-B, the normal stochastic gradient descent algorithm is not computationally feasible. The stochastic nature of the algorithm is considered as data inconsistency by Sharma et al [14]. Therefore, a variation of the SGD is proposed by the Sharma et al [14] which overcome this dependency on the randomness by using a greedy strategy for selecting consistent datasets for the gradient descent. The method is named as Guided stochastic gradient descent which has been tested in various datasets of medical data sets. The Guided Stochastic SGD works like an add-on which could be plugged in to the other SGD optimization techniques. The major improvement brought in by this guided mechanism is that, where normal stochastic gradient has an irregular path to the convergence point, the guided SGD smoothens the path to the convergence point. Logistic regression problems have benefitted most as per the empirical tests performed on the same. The popular optimizers were mutated with this optimization and variations were published such as GAdadelta, GAdagrad, GAdam, GMomentum etc .The tests results were compared against the original methods of Adam, Adagrad and momentum variations of the gradient descent and test results were seen as improved [14] .

I. Distributed optimization using gossip exchange(GoSGD)

'Distributed optimization for deep learning with gossip exchange' by Blot et al [15] gives a detailed method on how to distribute the training tasks of a neural network which allows the system to increase performance. This is achieved by using local variables optimization for the gradient descent algorithm in each of the nodes in the distributed environment. GoSGD have shown significant improvements in the communication efficiency as high as 0.01 message/update which makes the network usage almost insignificant [15]. This achieved by having precise control over the update strategy by optimizing the update requirements in a distributed deep neural network. Further, gossip exchange mechanism is used in order to synchronise the changes between the various threads. The method is named as GoSGD which claims to be decentralized and fully asynchronous.

J. Boltzmann methods optimization

The size of the deep neural networks is one of the key challenges faced in the computing field of deep learning when implemented for big data applications. The number of nodes increases tremendously as the features to be learned increases. When it comes to the big data applications, the features are numerous which creates a requirement for a very wide network. When it comes to image and video analysis, the number of weights to be trained increases exponentially. In an attempt to optimize this constraint, Omidyar et al [16] have formulated a combination of conjugate gradient and Boltzmann method to reduce the number of weights to a great extent while preserving the feature extraction capabilities. In the experimental analysis of the method, it was found that with 90.3% reduction in the number of weights with a temperature of 0.55 was able to have identical accuracy as the original network for the character recognition deep learning dataset. Vapnic-Chervonekis dimensionality [16] was employed for the compression of the information into fewer bytes than original using the sampling principles. This reduced information content to be trained along with the reduction of weight while training using Boltzmann criteria.

K. Pigeon inspired optimization (PIO)

With the arrival of the personalization devices such as Alexa, Siri, etc. the processing of the audio data has emerged as a big data problem to be solved. The number of voice clips which has to be processed is getting accumulated every second. This has created a requirement of systems which can process the speech-based commands in real-time by automatic speech-recognition. Deep neural networks have improved the performance of such systems by a great margin over the GMM-HMM systems. A specific optimization of deep neural network for ASR have been prosed by Waris et al [17] using the pigeon inspired optimization(PIO). This method makes use of the heuristic knowledge available about the context of speech and applications to optimize the matrix weight of the neural networks. The method has been proved to reduce the training time and the accuracy of the automatic speech recognition on the TIMIT database.

L. Rate-Accuracy optimization

The deep neural networks are getting wider and deeper with the big data applications as the limits for accuracy is being pushed closer to perfection day by day. But this creates an issue for resource management, a bigger network will have a number of parameters to be stored and a number of iterations to train. Filini et al [18] have proposed a framework for storing and transmission of the deep neural networks. A compression method is implemented in order to reduce the storage of the network weights. The method is almost exclusive of the classification problems using deep neural networks. Based on the depth and type of layers various quantization techniques are employed as several statistical models. The method is independent of the application area of deep learning or the implementation platform. The bitrate has been reduced by 92% [18] with a very insignificant accuracy loss for the image classification in the empirical tests performed .

M. Hyperparameters optimization

Finding the right hyperparameters is one of the greatest challenges in deep learning methods. In a normal offline deep learning system, the data can be split into training, validation and test sets in order to train, hyperparameter tuning and testing the model respectively. In big data systems, hyperparameters are very crucial as a shift in these values could cause longer training times or lower accuracy. A method for the same is suggested by Yoo [19] in using univariate dynamic encoding algorithm for searches. Another method is described in the paper 'A new hyperparameters optimization method for convolutional neural networks' where Multilevel and Multiscale evolutionary optimization with GPEI is used to optimize hyperparameter for deep learning [20]. In this method, the error function f [21] is evaluated for the hyperparameters considering the function as a black box with other hyperparameters as constant (other hyperparameters are considered to be optimized already).

N. Deep reinforcement learning

Deep reinforcement learning is a derivative of the Reinforcement learning which makes use of deep Q network to evaluate and find the optimal level of value-action function [22]. These are mainly used in the Wireless network systems to optimize the traffic load. The big data systems deployed by the telecom operator to access the traffic based out of VoIP and cellular networks can benefit from these types of deep learning optimization Techniques. Nallaperuma et al [23] have also demonstrated how the normal deep learning systems cannot cater to the requirement of highly volatile environments. They have proposed a method which uses deep reinforcement learning which optimizes the overall accuracy and enables a real-time responsive in the area of traffic control. The deep reinforcement learning is the most intriguing topics which have gained popularity by blending deep neural networks and big data. Especially in the field social media dynamics analysis and connected systems like autonomous bot's coordination, smart-driver applications where the data is enormous and dynamic in nature. Alpha Go is the one of most successful in terms of the implementation victory of the deep reinforcement learning [24].

O. Particle swarm optimization for deep learning (ConvNet-PSO)

For optimizing the CNN networks, a method using Particle Swarm optimization is suggested by Khalifa et al [25]. The original PSO was designed to improve the target solution in an iterative manner, which was a prominently a computational optimization of the deep neural network. In this technique, the optimization is achieved by the integration of intelligent heuristic algorithms into deep neural networks. In an abstract level, the particles are considered flying pigeons for which, each one is a plausible solution, then these particles are made to hover above the solution space. A dynamic updating of the equation is done for changing the position and velocity of each particle independently but optimizing the directions of every particle in order to find the global minimum [26]. The core parameters of hidden layers are tuned using the Particle swarm optimization Algorithm. The method is a combination of two previously established optimization algorithms where the first six layers of the ConvNet is optimized using the normal stochastic gradient descent and the final layer using the particle swarm optimization. As per the test results described in the paper [25], the accuracy has increased by 4% even from the first epoch of the training cycle.

P. Quantization parameter optimization

This optimization is specifically designed for the Convolutional neural network where the quantization problem is persistent. When the input data type is converted from floating-point to integer-points, the precision values are trimmed down due to the limitations of the memory and time complexity. The process of quantization mitigates the precision loss which is created by the resource optimization via sampling of the dataset for the deep learning [27]. Therefore, the quantization parameters are of utmost importance while the performance of the deep learning is boosted using quantization for the bigger datasets. As shown in the empirical tests in the study by Kwasniewska [27], the performance of the deep learning system was boosted significantly while preserving the accuracy to a greater extent.

VI. Benchmark criteria for optimization techniques

A. Big data benchmarks

In a big data system, optimization techniques have to be evaluated based on how they improve the capability of the system in the following dimensions [1]:

1) Volume: The volume represents the ability of the systems to remain robust when the system is scaled out to accommodate the increasing amount of data. This is the most important dimension which any big data system has to adhere to [28]. The concept has gained popularity with the introduction of Hadoop systems where the storage was handled in a distributed way in HDFS and the processing was designed using map-reduce programming paradigm [1]. There have been further improvements on the same using the further libraries like Apache Hive(SQL based map-reduce), Apache Spark(In-Memory map-reduce), etc. Introduction of deep learning into the big data is a very delicate matter; hence any implementation or optimization should not hamper the ability of the system to handle a large volume of data.

2) Velocity: When the concept of big data was conceived, the programming paradigm was only designed for batch processing. But the requirements with the big data systems were not able to stay in its limits, for which the systems evolved to handle the data in real-time also with the technologies like Kafka and Spark Streaming libraries [28]. This has created a game-changing effect on the ecosystem where new use cases have emerged with this added capability [1]. The deep learning methods also had the same limitation of batch processing, which had to be mitigated when integrated with the big data systems. Therefore, the optimization techniques should be chosen which can comply with this requirement of the big data system, where even with the high velocity of the data, the accuracy and the latency is not compromised.

3) Variety: With the introduction IoT enabled systems, the portfolio of the data collection points have diverged tremendously. This has led to the development of systems which can process unstructured and semi-structured data [28]. Furthermore, a cycle of upscaling has happened in the areas where the ability to handle heterogeneous data has created new use cases possible and this has led to advancement in the system to handle more complex type and vice-versa. The transformation functions are getting more complex as new data types have to be fed into a neural network on the normalized scale. Therefore, the optimization techniques should be chosen carefully by diligently evaluating how this multitype dataset which is normalized can remain at the desired level.

B. Deep learning benchmarks

The deep learning systems basically has two stages: 1) Training 2) Inferencing. Both these stages can have the two basic computing trade-off criteria, which every system has to go through Speed vs Accuracy. Hence the following benchmarks can be used in order to evaluate how each optimization techniques improves a deep learning-based big data system.

1) Training Speed: In a normal deep learning project itself, the training time is resource-intensive. When deep learning is applied to the big data systems, the training time will be increased analogous to the data volume. Hence an optimization method which can improve the training speed is one of crucial technique which can improve the performance of the overall system [29]. In real-time systems, training in a Q-learning paradigm could be a good option in order to incorporate knowledge from recent data.

2) Convergence Accuracy: As described in section IV deep neural network, the neural network is trained by correcting the RMSE between the expected and actual value. But due to several reasons, the accuracy does not increase after an arbitrary number of iterations. Some of the reasons are: the neural network is not deep enough, the hypermeters like learning rate are too high or low, features are not normalized, etc. But if any optimization techniques can be applied which could reduce this error rate, the accuracy of the deep learning will be higher. Care should be taken while attempting such optimization as it could lead to an overfitting state of the neural network [29].

3) Inferencing speed: Once the training is completed for the deep neural network, the new data is input into the same to do a prediction. Based on the size of the neural network and data set which is fed, the inferencing speed will vary. In a real-time big data system, with the high velocity of the data, it is not feasible to have a high inferencing latency. Hence, the optimization techniques which can improve the inferencing speed is also a key factor influencing the overall efficiency of the system [29].

4) Prediction Accuracy: This is the most important factor which the end-user can directly evaluate, therefore, any implementation of the deep neural network or its optimization should be done with great diligence so that accuracy of the prediction is as high as possible. As a general rule, any optimization technique which can hamper the accuracy should be discarded and any optimization technique which can increase should be treated with the highest regard [29].

C. System Benchmarks

For the implementation purposes, the technique should be evaluated based on how it acts as a system. Hence how much the optimization techniques are affecting the overall system has to be assessed to give the feasibility aspect of the optimization techniques. The systems parameters which will be compared will be:

1) Processing speed: Processing power is the most crucial element in any computing systems. This is applicable to the deep learning in big data systems also. The deep neural network has a hefty computing requirement because of the high order matrix operations. Hence, it is crucial that optimization techniques contribute to the improvement of computing constraints.

2) Memory: As the modern deep neural network have profound depth and contains a substantial number of nodes, the amount of the memory required is tremendous. Therefore, the memory optimization is another crucial feature which is required for the deep learning implementation in the big data systems.

3) Parallelization: With the introduction of the commercial GPU processors both in retail and cloud systems has created a new opportunity to take advantage of parallel processing. Unlike the normal computing programs, deep neural networks have an inherent ability to have parallel computing tasks because the deep learning heavily relies upon matrix operations. Therefore, optimization which increases the parallelization of tasks such as backpropagation will bring in enormous advantages.

4) Network bandwidth: The big data systems are notorious for the huge amount of data traffic which exhaust the network bandwidth in the data centres. In case of image and video-based application which use CNN will have to be optimized for reducing the number of iterations and epoch which in turn will decrease the number of reading operations from remote storages.

5) Latency: As the big data systems are gaining traction in social media; real-time applications are becoming more adaptive to the deep learning systems. In real-time applications, the time for new data to be processed should be reduced in order to make it a viable choice. Therefore, optimization techniques which can reduce the inferencing time will be of huge consolation to the resource-intensive deep learning systems.

VII. Comparison of optimization techniques

Based on the benchmarks defined in section VI the optimization techniques will be compared in this section using the Tables II & III. The tables depict which benchmarks are improved by each optimization technique. The results are categorised in 4 grades.

1) High: Significant improvement in the performance
2) Medium: It gives a mediocre improvement by applying this technique
3) Low: Implementation of the technique will give very minimal improvement
4) NA: cannot be applied or no improvement

A. Evaluating optimization techniques based on big data benchmarks

The big data benchmarks are selected in order to compare how the optimization techniques par with the scalability requirements. Hyperparameters optimization and Boltzmann methods are found to be most effective in terms of dealing with a high volume of data. GoSGD along with Hyperparameter optimization are the methods which give an edge while dealing with a high velocity of data streams. Deep Reinforcement Learning with its ability to learn adaptively in a dynamic changing environment makes it most apt to deal with diverse data streams.

B. Evaluating optimization techniques based on deep learning benchmarks

As described in the section of the Table II, the benchmarks are training speed, convergence accuracy, inferencing speed and prediction accuracy. For the training speed as the primary benchmark, the Averaged SCD and Boltzmann methods are the optimal selections. If the focus of the optimization requirements is to reach the training saturation point faster, then the averaged SCD and Guided SGD are the best options as per the table. When the inferencing speed is considered as the principal goal then the Pigeon inspired optimization is the best one to select from discussed optimization techniques. Quantization parameter method for the optimization is having the highest performance in terms of ability to improve the prediction accuracy of the system.

C. Evaluating optimization techniques based on system benchmarks

The Table III gives a comparative study on how the system benchmarks can be improved by implementing the optimization techniques. For improving the processing speed of the big data deep learning system, the Averaged SGD and Pigeon-Inspired optimization have the highest capabilities. For optimizing the memory complexity of the system, Averaged SGD and Boltzmann methods have shown good results in the empirical tests. In terms of the parallelization requirements, the Downpour SGD and Sandblaster L-BFGS are the toppers as they are designed for distributed deep neural network training. Along with sandblaster L-BFGS, GoSGD and PIO have good capabilities in reducing the network traffic while running in a distributed environment. For the latency related issues, the GoSCD and PIO would be the best options to implement for optimizing the deep neural networks.

VIII. Identification of optimization requirements

The opportunities created by the application of deep learning in big data system is close to omniscient where it can make use of prediction capabilities on medical-related data to find previously unexplored dataset properties in the market responses in the commercial field. This section will discuss the various applications in brief in order to explain later how the various optimization technologies can improve the performance in each area. The application areas will be addressed in this research will be:

A. IoT

Internet of Things is changing the IT ecosystem where the amount of data generated has increased exponentially with the introduction of smart devices [30]. This has created a requirement for systems which can collect the data at a high rate and volume [31]. The IoT is a new environment in itself with applications varying from personal devices to the public transport systems [32]. Xhafa has [30] pointed out that the biggest challenge in the IoT applications to be used in big data is the heterogeneity of the communication protocols which leads to the original issue of big data: Veracity. This heterogeneous nature of the data will have a direct impact on the optimization technique which can be used for deep learning.

The IoT systems play a key role in the fields of identity and access management, communication tracking, network management and service industry for customer experience [33]. Furthermore, as the AI systems implemented using deep neural networks brings in several advantages, it is necessary to adapt and address the issues while integrating them into big data systems build for IoT [30]. But as the IoT systems call for real-time processing and feedback, the direct adaption of the deep learning is not an apt option, therefore, various optimizations have to be evaluated in order to implement the same in the IoT arena.

B. Cybersecurity

In any computing systems, one of the major concerns is cybersecurity. The integration of the deep leaning into the big data has expanded the territory to be secured tremendously. But, as Masabo et al [34] have pointed out, the same concept can be used in order to detect the malware which is propagating through the networks. The number of devices which are connected to the internet has exponentially increased; therefore, an extremely large number of new malwares has been devised by hackers and will keep evolving. This creates a system which can deal with a large variety of device protocol and traffic data and the big data provides the same. The nature of malware varies a lot in terms of languages, platforms and operating systems which are human derivable features. But there are a lot of analogous behaviours of malware which could be derived using the deep neural networks. Hence, combining the deep learning capabilities with big data platforms creates a perfect tool for finding the malware getting injected into the networks. But the implementation of such a system needs to have optimized techniques from respective the fields which can result in some tangible outcomes.

Jallad et al. [35] have proposed a new method in which the distributed deep recurrent neural networks can be used to prevent contextual and collective attacks along with the improvement in the false positives and detection rate of the intrusion detection systems. This method has taken advantage of multiple areas of computing such as natural language processing, deep neural networks and big data. The method also integrates the techniques used for anomaly detection and contextual breakdown. The paper has given details on how deep learning techniques can be improved in order to have real-time protection against hacking attempts. Therefore, the optimization techniques which do not compromise the real-time capabilities have to be used for this type of applications.

C. Measurement systems

The measurements systems were one of the first adopters of analytics systems. With the introduction of the IoT, the measurements systems become divergent than ever before [33]. This has led to the growth of the analytics systems into the big data analytics system. In a paper by Seo et al [36], a system which can utilize

Benchmarks	Big data benchmarks			Deep learning benchmarks			
Optimization Techniques	Volume	Velocity	Veracity	Training Speed	Convergence Accuracy	Inferencing Speed	Prediction Accuracy
Stochastic gradient descent	Low	Low	Low	Medium	High	Low	NA
Downpour SGD	Medium	Low	Low	Medium	Low	Low	NA
Sandblaster L-BFGS	Medium	NA	NA	Medium	Low	Low	NA
Stochastic diagonal approximate greatest descent	NA	Medium	Low	Medium	Low	Low	Medium
Inconsistent SGD	NA	NA	NA	Low	Medium	Medium	Low
Averaged SGD	Low	NA	NA	High	High	Medium	Low
Guided SGD	NA	NA	Medium	NA	High	NA	Medium
GoSGD	Medium	High	Medium	Low	Medium	Low	Low
Boltzmann methods optimization	High	Medium	Low	High	Medium	Low	Medium
Pigeon inspired optimization	Low	Low	Low	Medium	Medium	High	Medium
Rate-Accuracy optimization	Low	Medium	Low	Medium	NA	Low	NA
Hyperparameters optimization	High	High	Medium	Medium	Low	NA	NA
Deep reinforcement learning	Medium	Medium	High	NA	NA	Low	Medium
ConvNet-PSO	Low	NA	NA	Medium	Medium	Low	Low
Quantization parameters	Medium	NA	NA	Medium	High	NA	High

TABLE II: Comparison based on Characteristics of big data and deep learning benchmarks

the deep learning capabilities in the big data systems for power measurements is introduced. The paper presents how power consumption can be optimized using the deep neural networks in a big data system. The technique consists of collecting data to the central FEP, NMS and SMS server of a power grid from the distributed meters across the distribution system. The major use case of the method is to provide efficient services in the area of analytics which can be further used to predict the future load energy requirements. A recurrent neural network with Long-Short Term Memory cells [36] is used in order to compare the change which further can derive the rate of change. This has two benefits in the overall use case:
1) To predict a failure due to peak load
2) The load balancing according to demand variation.
Another benefit described includes the compatibility to brownfield applications on the existing infrastructure. In conclusion, the accuracy of the systems is of utmost importance in this type of applications, hence the optimization techniques should be chosen which do not degrade the quality of these predictions.

D. Medical diagnosis

The medical field is one of the prominent beneficiaries of the advancements in deep learning. The nature of the medical field data is one of the key factors which makes the predictions of diseases very difficult [37]. The feature space itself is multi-tiered, where the combination of symptoms alone is not enough to conclude on the condition. The factors like patient's medical history(RNN techniques), family histories(DBN systems) also have an influence of the further diagnosis and treatments.

One of the most prominent type of deep learning networks used is in the field of scan image analysis. These include X-rays, CT scan, MRI scan, etc. Jakhar et al [37] have presented a very promising framework for detecting pneumonia using deep convolutional neural networks.

A comparative study was also presented which have clearly shown a substantial improvement of the method over the other machine learning techniques like SVM, random forest, etc. Since this type of use cases highly rely upon images and convolutional neural networks, optimization techniques which can improve the convolutional operations will be the most suitable technique for the medical field. In the field of medicine, there is a high amount of scrutiny because of the risk related to this. Because of which, the optimization techniques should be more concentrated in increasing the accuracy rather than speed.

Optimization Techniques	Processing speed	Memory	Parallelization	Network bandwidth	Latency
Stochastic gradient descent	Low	Low	NA	NA	NA
Downpour SGD	Low	Low	High	Medium	Low
Sandblaster L-BFGS	Low	Low	High	High	High
Stochastic diagonal approximate greatest descent	Medium	Low	NA	NA	Low
Inconsistent SGD	Medium	Low	NA	Low	Medium
Averaged SGD	High	High	NA	Medium	Medium
Guided SGD	Low	Low	Medium	NA	NA
GoSGD	Medium	Medium	High	High	High
Boltzmann methods optimization	Medium	High	Medium	NA	NA
Pigeon inspired optimization	High	Medium	Low	Medium	High
Rate-Accuracy optimization	Medium	High	Medium	High	Medium
Hyperparameters optimization	Low	NA	Medium	Low	NA
Deep reinforcement learning	NA	Low	Low	Medium	NA
ConvNet-PSO	Low	Medium	NA	Medium	Low
Quantization parameters	Medium	Low	Medium	Medium	NA

TABLE III: Comparison of deep learning systems based on the system benchmarks

E. Stock market

Stock markets are an area where is there is a high amount of volatility combined with the complexity of multi-level relations between the data. The text-based news and the tone of the same greatly drives the prices of the stocks; this includes new government policies, statements made by vital board members of a company. The ability of the Recurrent neural networks to handle the time-series data comes in handy in the stock market analysis [38] because the price of a stock at a point is a derivative of a previous price and will be contributing to the price of the stock in the future. The stock market has a variety of data streams which has to be analysed in order to have a decent prediction. For example, the stocks of parent and sibling company, the overall trend of the market, the reviews of the new products coming out etc. Sismanoglu et al [38] have proposed a method for estimation of high-volume financial time series based on big data and deep learning. The technique claims to reduce to Root Mean Square Error by the use of Long-Short Term Memory cells in the RNN networks. So, this type of applications will be benefitted with optimization techniques which can improve the ability of the DL to deal the variety of data streams and those which can improve the time-series feature of the RNN's.

The applications of finance have a direct impact on the customer's profile and the data is time-sensitive, therefore the deep learning optimization should be giving more edge on the latency related issues.

F. Mobile networks

The mobile networks are the backbone of the modern world. In this era of a digitally connected world, almost everything is connected to the internet; which is made possible by the advancement in wireless technologies [39]. Every year there is a wide range new product launched with better speed and power efficiency which adds to the ever-growing number of active devices. This expansion of the field has increased the network complexity which in turn increased cost of handling the outages in the cellular networks [40]. This enormous number of devices and its data pave the way to the application of big data systems into the cellular network management applications. According to Hussain et al, one of the major challenges is to handle the sleeping cell anomaly where the Quality of Service is degraded even though from the operator network, it is perfectly working. Various methods have been proposed to find this anomaly, but deep learning has provided a method to increase accuracy and to reduce the false alarms in the same. Such a system is proposed by Hussain et al, where an L-layer deep feedforward neural network is used which have achieved 94.6% accuracy with 1.7% false-positive rate (FPR). As described, this type of network-based use cases, the deep learning techniques which can act in real-time is desired.

G. Transportation

The invention of wheels is considered as one of the milestones which changed the course of human history. It has become so crucial in daily life that, a day without transportation facilities the world will come to a still. Due to the same reason, a lot of interest has been gained on how big data and deep learning can be utilized in the field of transportation. The cost of daily products is highly depended on the cost of transportation which is conventionally optimized by the logistics techniques of containerization, piggy bagging extra. The advancement of IoT and cellular networks along with the reduced cost for the wireless connectivity devices, the vehicular systems have also adapted the data-centric approach in order to increase the efficiency of the transportation systems [41]. Autonomous vehicles are one of the most debated topics which raised questions from an efficiency point of view to the ethical and moral behavioural of the systems. As a trade-off, assisted driver technologies have gained high popularity in the recent times which have increased the performance as per Jachmczyl et al [42]. This technology has varieties of other fields merged into the same from computer vision to the psychological analysis of the drivers. Accessing the driver's mood and behavioural patterns give a doubled folded benefit, that is, entertainment systems to improve the driver's mood will decrease a chance of rash driving [43] and second is the evaluate the drivers which could further be used for improvements in the driving style of other drivers. These methods highly rely upon the computer vision, natural language processing and GPS systems. This exhibits the variety aspect of the big data and real-time requirement in the deep learning system. This shows that the optimization techniques which can improve that real-time behaviour or the ability to deal with complex data structure will be most desirable.

IX. Impact of optimization techniques

Based on the optimization requirements section VIII, the table IV will indicate how much each optimization technique will be effective. The results are categorised in 4 grades.

1) High: Highly effective in this type of application
2) Medium: It gives a mediocre improvement by applying this technique
3) Low: Implementation of the technique will give very minimal improvement
4) NA: cannot be applied or no improvement

From the comparative tables, we can derive that the IoT applications are benefitted most from the techniques: Sandblaster L-BFGS, GoSGD, Rate-Accuracy optimization and Deep reinforcement learning. In the case of cybersecurity applications, the Guided SGD and GoSGD are viable options. Measurements systems can be benefitted from most of the SCD based optimization techniques. Averaged SGD, Guided SGD and Quantization parameters are very good options for medical diagnosis as they provide high accuracy improvements in the deep learning. Stock Market applications will be benefitted by Pigeon based optimization and Boltzmann methods as they provide good scalability and increases real-time responsiveness measures. Downpour SGD and Sandblaster L-BFGS which improves the distributed computing capabilities of the deep learning will be the most effective in mobile network-based applications. In the cases of transportation-based applications, the sandblaster L-BFGS with its capability to distribute workload without any additional network requirements will have the best performance in optimizing the deep neural network.

The results tables are based on the exploration of the literature and published research works in the respective fields. Further experimental research needs to be conducted to evaluate the projections made in the comparison table.

X. Challenges

The integration of deep learning into big data systems opens up a lot of new applications areas. But this comes with a very huge set of integration and performance issues. Starting with the size of the deep neural networks, where the number of the neurons is increased exponentially in order to accommodate the complexity of a large dataset, the problems extend to the latency of the results which have to inferred through the deep neural networks. Even though solutions like Edge Computing and Data sampling are existing, these reduce the overall utilization of the capabilities of the deep neural networks.

A. Ethics

The area of Artificial Intelligence has opened up a controversial debate wherever it has set its foot on. In the field of big data systems, where areas of Medical diagnosis to Real-Time Defence Systems are integrated, the concerns raised about the ethical and moral standards are numerous. As described in section VI-B.4, the integration of deep learning into the big data infrastructure presents itself with a lot of complexity [44]. Great care is already taken in this process which has to be preserved while applying any kind of optimization techniques. None of the optimizations should compromise ethical standards built into the deep learning big data systems.

B. Security

There is a high risk in the field of security when it comes to any IT system. Firewalls and malware detection systems are designed around the infrastructure with great diligence. In the case of big data systems, the collection points (Figure 1) are the most vulnerable points. In the case of real-time solutions, the time to scan the entire data is almost equal to zero. Therefore, the deep learning systems can be corrupted with fake data which could mislead the results [4]. This could have catastrophic effects on the outputs leading to the wrong decisions could cost life to a patient in the case of medical diagnosis or

Optimization Techniques	IoT	Cyber Security	Measurement Systems	Medical Diagnosis	Stock Market	Mobile networks	Transportation
Stochastic gradient descent	Low	NA	Low	NA	Low	Low	Low
Downpour SGD	Medium	NA	Medium	Medium	High	High	NA
Sandblaster L-BFGS	High	Low	Medium	NA	Medium	High	High
Stochastic diagonal approximate greatest descent	Low	NA	Medium	NA	Medium	NA	Medium
Inconsistent SGD	Medium	Low	Medium	Medium	Medium	Low	NA
Averaged SGD	Medium	Low	Medium	High	High	Low	Low
Guided SGD	NA	Medium	Low	High	Low	Medium	Low
GoSGD	High	Medium	Low	NA	Low	High	Medium
Boltzmann methods optimization	Medium	NA	Low	High	High	NA	Medium
Pigeon inspired optimization	Medium	Low	NA	High	High	NA	Medium
Rate-Accuracy optimization	High	Low	Low	Low	Low	Medium	Medium
Hyperparameters optimization	Medium	NA	NA	High	Medium	NA	NA
Deep reinforcement learning	High	NA	Low	Medium	Medium	NA	Medium
ConvNet-PSO	Low	Medium	NA	Low	Low	NA	Low
Quantization parameters	NA	Medium	NA	High	Medium	NA	Medium

TABLE IV: Impact of the optimization techniques in various field

a pedestrian in the case of autonomous vehicle control systems.

C. Non-stationary Data

The conventional deep learning systems are trained in batches where the data is collected and feed into the neural network as a whole. But in the big data systems, the data is not static [2]. The data flows in streams which requires a system which can tune itself based on new evidence and this is not a simple engineering task. Such a system will require tools which have to filter the data in real-time. Another toolchain which can do the transformations for prepping the data for feeding into the neural network. This has challenges in two aspects: 1) most of the transformation of the data has to be done through tailor-made logic which is time-consuming 2) such a complex pipeline of transformation will be resource-intensive, which has to be distributed around the collection points itself in order to achieve real-time feeding.

D. High dimensional data

High Dimensions of Data demands a wider neural network to handle the feature space of the data. This has an impact on the size of the neural network which in turn affects the performance because of the resource constraints. On another scenario, this also requires various custom functions for the transformation and normalization of the data. Combined with the heterogeneous nature of the data, the high dimensional data creates a complex pipeline design for collection, training and inferencing [2]. When the implementation of the optimization techniques is designed, the system will have a higher chance of being corrupted and failure of the pipeline seems unavoidable unless a high amount of validation cases is exerted in the testing phase.

E. Large Scale Models

The deep neural network is a complex data structure with a large number of parameters which need to be trained, tuned and stored. This property presents the computing community with a lot of challenges starting with high order mathematical calculations to the requirement for parallel computing. Even though some challenges are addressed with the introduction of Graphical Processors, the maturity of the frameworks is still at rudimentary state [2]. Adding to the existing issues, the optimization techniques especially the ones in gradient descent category adds complexity to the mathematical computations. These factors combinedly create a demanding ecosystem where multiple aspects have to be dealt with the diligence of resource allocation.

XI. Areas for future advancements

A. Deep reinforcement learning

Reinforcement learning is very apt for real-time systems, where continuous learning on the streaming data is enabled. This is one of the most promising optimization techniques in the area of deep learning, but it comes with its own challenges. Currently, there is no frameworks are available which explains why there are no much production quality projects of this type. Secondly, the supporting technologies like high-speed network bandwidth, swift processing power and a tremendous amount of random-access memory need to be improved if we need to exploit the capabilities of deep reinforcement learning [23].

B. Multimodal data handling

The big data systems are complex in nature where the data sets have heterogeneous composition. Even though there are systems are upgraded to accommodate the new data types, there are always new applications which contributes to a continuously evolving nature of data sets. This has a direct impact on the deep learning [2] systems, where there is a requirement for data to be normalized for training and inferencing purposes in the neural network. For example, text data are represented in a vectorised form which creates a sparse matrix. The limit of the deep neural network to correlate non-linear relations for low-level features is also an important issue to be solved [45]. Therefore, new optimization techniques are required which can deal with a greater number of data modal.

C. Storage options

As per Hodak [46], the storage techniques to be used in deep learning-based big data systems is of great importance. In their study, the comparison of various storage options was explored and evaluated, after which they have suggested the improvements to be incorporated which could improve the big data systems. Those are:

1) The bandwidth between the computing and storage cluster should be of at least 1Gb/s to achieve a faster learning process.
2) The storage or file system is not of that much importance as long as the network speed is achieved.
3) RAM of the computing systems has a direct impact which can contribute to the faster training by caching the data in the local memory units.
4) Local storage systems enable with disk caching will boost the performance which can be complemented with SSD storage.

These suggestions are a very good place to invest in order to create a better and faster deep learning solution.

D. Generalisation of frameworks

Currently, most of the deep learning systems are custom developed from base layer for each big data system, hence the optimization techniques used also will be of highly customized behaviour which is undesirable [30]. A more generic framework for implementing deep learning in big data analytics framework should be built. Distributed Keras Framework [47] developed under the reusable project under CERN is one the best example for deep learning framework on a big data infrastructure. But the maturity level of the library is quite low and the options available are few. Araki et al [48] argue that the current Hadoop and Spark systems are inefficient for implementing deep learning solutions for big data applications, which creates a requirement for a complete set of new big data tool-sets in the big data arena. Therefore, more big data deep learning frameworks are required to investigate more on how we can optimize neural networks for big data systems.

XII. Conclusion

The deep learning techniques have proven to be very effective in implementing the artificial intelligence use cases, along with the advancements of big data technologies a huge window of opportunity has been opened up in this field. But the consolidation of two huge technologies comes with its own new set of complexities for which the new optimization techniques have to be implemented. There have been various works done in order to decrease this complexity and constraints. This survey has provided insights to the various techniques which will give an approximate idea on which techniques is beneficial for various use-case requirements. The techniques have compared and contrasted in four dimensions such as big data, deep learning, systems and application area. Furthermore, the challenges faced in the big data-deep learning systems have been presented. Finally, the future work areas identified by various authors have been briefly discussed.

References

[1] M. Gheisari, G. Wang, and M. Z. A. Bhuiyan, "A survey on deep learning in big data," in 22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), pp. 173–180, IEEE.

[2] K. S. Rani, M. Kumari, V. B. Singh, and M. Sharma, "Deep learning with big data: An emerging trend," in 2019 19th International Conference on Computational Science and Its Applications (ICCSA), pp. 93–101, IEEE.

[3] B. Jan, H. Farman, M. Khan, M. Imran, I. U. Islam, A. Ahmad, S. Ali, and G. Jeon, "Deep learning in big data analytics: A comparative study," vol. 75, pp. 275–287.

[4] S. Serebryakov, D. Milojicic, N. Vassilieva, S. Fleischman, and R. D. Clark, "Deep learning cookbook: Recipes for your AI infrastructure and applications," in 2019 IEEE International Conference on Rebooting Computing (ICRC), pp. 1–9, IEEE.

[5] L. Thomas, "EDGE INTELLIGENCE: A BLEND OF AI WITH EDGE COMPUTING," p. 10.

[6] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing,"

[7] J.-R. Su, J.-G. Wang, Z.-T. Xie, Y. Yao, and J. Liu, "A method for EEG contributory channel selection based on deep belief network," pp. 1247–1252.

[8] A. V. Srinivasan, "Stochastic gradient descent — clearly explained !!."

[9] S. Clémençon, A. Bellet, O. Jelassi, and G. Papa, "Scalability of stochastic gradient descent based on "smart" sampling techniques," vol. 53, pp. 308–315.

[10] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, "Large scale distributed deep networks," p. 9.

[11] H. H. Tan and K. H. Lim, "Vanishing gradient mitigation with deep learning neural network optimization," in 2019 7th International Conference on Smart Computing & Communications (ICSCC), pp. 1–4, IEEE.

[12] L. Wang, Y. Yang, R. Min, and S. Chakradhar, "Accelerating deep neural network training with inconsistent stochastic gradient descent," vol. 93, pp. 219–229.

[13] Z. You, X. Wang, and B. Xu, "Exploring one pass learning for deep neural network training with averaged stochastic gradient descent," in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6854–6858, IEEE.

[14] A. Sharma, "Guided stochastic gradient descent algorithm for inconsistent datasets," vol. 73, pp. 1068–1080.

[15] M. Blot, D. Picard, N. Thome, and M. Cord, "Distributed optimization for deep learning with gossip exchange," vol. 330, pp. 287–296.

[16] O. Omidvar and C. Wilson, "Optimization of neural network topology and information content using boltzmann methods," in [Proceedings 1992] IJCNN International Joint Conference on Neural Networks, vol. 4, pp. 594–599 vol.4. ISSN: null.

[17] A. Waris and R. Aggarwal, "Optimization of deep neural network for automatic speech recognition," in 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 524–527, IEEE.

[18] A. Filini, J. Ascenso, and R. Leonardi, "Rate-accuracy optimization of deep convolutional neural network models," in 2017 IEEE International Symposium on Multimedia (ISM), pp. 91–98, IEEE.

[19] Y. Yoo, "Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches," vol. 178, pp. 74–83.

[20] H. Cui and J. Bai, "A new hyperparameters optimization method for convolutional neural networks," vol. 125, pp. 828–834.

[21] I. Ilievski, T. Akhtar, J. Feng, and C. A. Shoemaker, "Efficient hyperparameter optimization of deep learning algorithms using deterministic RBF surrogates," p. 8.

[22] Y. He, C. Liang, F. R. Yu, N. Zhao, and H. Yin, "Optimization of cache-enabled opportunistic interference alignment wireless networks: A big data deep reinforcement learning approach," in 2017 IEEE International Conference on Communications (ICC), pp. 1–6, IEEE.

[23] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," vol. 20, no. 12, pp. 4679–4690.

[24] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," vol. 529, no. 7587, pp. 484–489.

[25] M. H. Khalifa, M. Ammar, W. Ouarda, and A. M. Alimi, "Particle swarm optimization for deep learning of convolution neural network," in 2017 Sudan Conference on Computer Science and Information Technology (SCCSIT), pp. 1–5, IEEE.

[26] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in Proceedings of the 2007 IEEE Swarm Intelligence Symposium, pp. 120–127, IEEE Computer Society.

[27] A. Kwasniewska, M. Szankin, M. Ozga, J. Wolfe, A. Das, A. Zajac, J. Ruminski, and P. Rad, "Deep learning optimization for edge devices: Analysis of training quantization parameters," in IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, pp. 96–101, IEEE.

[28] P. Yu and H. Yan, "Study on feature selection and feature deep learning model for big data," in 2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE), pp. 792–795, IEEE.

[29] S. Mittal, "A survey on optimized implementation of deep learning models on the NVIDIA jetson platform," p. 15.

[30] M. Strohbach, H. Ziekow, V. Gazis, and N. Akiva, "Towards a big data analytics framework for IoT and smart city applications," in Modeling and Processing for Next-Generation Big-Data Technologies (F. Xhafa, L. Barolli, A. Barolli, and P. Papajorgji, eds.), vol. 4, pp. 257–282, Springer International Publishing.

[31] T. T. Chhowa, M. A. Rahman, A. K. Paul, and R. Ahmmed, "A narrative analysis on deep learning in IoT based medical big data analysis with future perspectives," in 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–6, IEEE.

[32] "Disruptive technologies: Advances that will transform life, business, and the global economy | McKinsey."

[33] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," vol. 10, no. 4, pp. 2233–2243.

[34] E. Masabo, K. S. Kaawaase, and J. Sansa-Otim, "Big data: deep learning for detecting malware," in Proceedings of the 2018 International Conference on Software Engineering in Africa - SEiA '18, pp. 20–26, ACM Press.

[35] K. Al Jallad, M. Aljnidi, and M. S. Desouki, "Big data analysis and distributed deep learning for next-generation intrusion detection system optimization," vol. 6, no. 1, p. 88.

[36] C.-K. Seo, J.-H. Kim, and S.-Y. Kwon, "A study on modeling using big data and deep learning method for failure diagnosis of system," in 2018 IEEE International Conference on Big Data (Big Data), pp. 4747–4751, IEEE.

[37] K. Jakhar and N. Hooda, "Big data deep learning framework using keras: A case study of pneumonia prediction," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1–5, IEEE.

[38] G. Sismanoglu, M. A. Onde, F. Kocer, and O. K. Sahingoz, "Deep learning based forecasting in stock market with big data analytics," in 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), pp. 1–4, IEEE.

[39] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5g: how to empower SON with big data for enabling 5g," vol. 28, no. 6, pp. 27–33.

[40] B. Hussain, Q. Du, and P. Ren, "Deep learning-based big data-assisted anomaly detection in cellular networks," in 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, IEEE.

[41] D. Jeong, M. Kim, K. Kim, T. Kim, J. Jin, C. Lee, and S. Lim, "Real-time driver identification using vehicular big data and deep learning," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 123–130, IEEE.

[42] B. Jachimczyk, D. Dziak, J. Czapla, P. Damps, and W. J. Kulesza, "IoT on-board system for driving style assessment," vol. 18, no. 4, p. 1233.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," vol. 60, no. 6, pp. 84–90.

[44] S. Okul, D. Aksu, and M. A. Aydin, "Applications of deep learning and big data technologies," in 2019 16th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 550–553, IEEE.

[45] Xue-Wen Chen and Xiaotong Lin, "Big data deep learning: Challenges and perspectives," vol. 2, pp. 514–525.

[46] M. Hodak, D. Ellison, P. Seidel, and A. Dholakia, "Performance implications of big data in scalable deep learning: On the importance of bandwidth and caching," in 2018 IEEE International Conference on Big Data (Big Data), pp. 1945–1950, IEEE.

[47] "cerndb/dist-keras." original-date: 2016-07-25T09:47:37Z.

[48] T. Araki and Y. Nakamura, "Future trend of deep learning frameworks — from the perspective of big data analytics and HPC," in 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 696–703, IEEE.