

Dynamic Resource Allocation and Memory Management using Deep Convolutional Neural Network



Dipak Raghunath Patil, Rajesh Purohit

Abstract: Memory management is very essential task for large-scale storage systems; in mobile platform generate storage errors due to insufficient memory as well as additional task overhead. Many existing systems have illustrated different solution for such issues, like load balancing and load rebalancing. Different unusable applications which are already installed in mobile platform user never access frequently but it allocates some memory space on hard device storage. In the proposed research work we describe dynamic resource allocation for mobile platforms using deep learning approach. In Real world mobile systems users may install different kind of applications which required ad-hoc basis. Such applications may be affect to execution performance of system as well space complexity, sometime they also affect another runnable applications performance. To eliminate of such issues, we carried out an approach to allocate runtime resources for data storage for mobile platform. When system connected with cloud data server it store complete file system on remote Virtual Machine (VM) and whenever a single application required which immediately install beginning as remote server to local device. For developed of proposed system we implemented deep learning base Convolutional Neural Network (CNN), algorithm has used with tensorflow environment which reduces the time complexity for data storage as well as extraction respectively.

Keywords — Deep Learning, transfer Learning.

I. INTRODUCTION

Deep learning affords new prospects for mobile platform to attain superior performance than earlier. Somewhat, the deep learning completion on mobile device today is mostly demanding on exclusive resource overheads, inflict a momentous burden on the battery existence and limited memory acquisition. Existing approaches moreover utilize cloud computing as well as edge infrastructure that necessitate to upload user data, conversely, resulting in a risk of confidentiality leakage and huge data transmit; or espouse compressed subterranean models, nonetheless, downgrading the algorithmic rule accuracy. the present deep learning reasoning is applied directly while not accuracy loss and no user-related information uploading is required. A series of light-weight ways are adopted to agitate potential operational

problems like mechanically collection garbage. The main contributions of present research work is that the combination of dynamic network conditions and application necessities into Computation-to-Communication quantitative relation, that considers the native interval for the employment, the number of information transfer, and network information measure.

This work basically proposed dynamic resource allocation as well as resource selection for storage devices to mobile platforms. This work also carried out proposed deep convolutional neural network (DCNN) with tensorflow libraries, which provides highly resource utilization and execute the entire task with minimum time complexity. This work also illustrates the real time application requirement to eliminate the low memory errors. This work uses multiple data storage devices in network environment, and sinks the local devices file system into the network storage. Basically the file system is correlated with respect to install application in mobile platform. Once the file system has uploaded on network instance it's having ability restore automatically whenever request generate from mobile users. This work achieves execution in minimum time complexity and reduces the network overhead during the data transmission. DCNN provides feature extraction and feature selection strategy which generates the data sinking event and front events required the event for a file system on local device.

II. LITERATURE SURVEY

According to J.M. Pierre et. al. [1] spatio-temporal deep knowledge for robotic illustration motor manage, this advance combine:

- (1) Continuous deep knowledge for conclude activity organize outputs from diagrams participation,
- (2) To proceed the multiple jobs in parallel manner with various outcomes using DNN and
- (3) And identifying the different features like a special temporal feature selection from different video frames. To extract the features from various images which is generated from input video file and define possibility according to the statement.

S. Leroux et. al. [2] the surge neural network: construction the internet of elegant things, suggest a new building called a Cascading system that is competent of distributing a deep neural network connecting a local machine and the cloud while performance the required announcement network traffic to a smallest. The network begins dispensation on the constrained machine and only relies on the inaccessible part when the restricted part does not afford an precise sufficient result S. Lian et. al.

Revised Manuscript Received on December 30, 2019.

* Correspondence Author

Mr. Dipak Raghunath Patil, Department of Computer Science & Engineering, School of Engineering & Technology, Suresh Gyan Vihar University, Jagatpura (Jaipur) India. dipak.patil@avcoe.org

Dr. Rajesh Purohit, Department of Computer Science & Engineering, School of Engineering & Technology, Suresh Gyan Vihar University, Jagatpura (Jaipur) India. gvset@mygyanvihar.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)



[3] Dadu: boosting the Inverse kinematics for High-DOF rRobots, proposed a new equivalent approach for automatic decision making system. System proposed new parallel algorithm called as Quick-IK, according to Jacobian transpose technique. Quick-IK reduce the time complexity as well as number of iterations around 95% then other evolutionary algorithms. It provides boosting to energy with the help of acceleration of hardware as well as resources. The proposed Quick-IK algorithm can be able to solve the problem within 12th milliseconds with degree 100 manipulator. The implementation has done with this system on High Performance Computing (HPC) as well as GPU's respectively. Quick-IK achieve higher energy implementation on HPC on environment.

J. Baek et. al. [4] landscape understanding networks for independent driving based on approximately examination monitoring system, this approach also proposed a new end-to-end resolution for delimiting the safe passable area for every edge by means of recognizing the neighboring obstacle in each way from the dynamic vehicle, we use this technique to estimate the detachment to the nearest obstacles and we integrate it into a integrated end to-end architecture proficient of joint object recognition, curb detection and safe passable area detection.

P. E. Sarlin et. al. [5] Leveraging deep visual descriptors for hierarchical efficient localization, system recommend leveraging new progress in deep learning to carried out an proficient hierarchical localization. We initial restrict at the map intensity using learned image-wide global information as well as metadata, and consequently estimation a accurate pose from 2D-3D contest computed in the contender places only. This confine the local investigate and thus allows to proficiently exploiting considerable non-binary descriptors typically dismissed on resource-constrained campaign.

J. Kaster et. Al. [6] Convolutional Neural Networks (CNN) on small un- manned aerial systems, review together hardware and convolutional neural network for small unmanaged arial system. Basically it proposed a data and coding approach during the feature extraction and how it selects best feature with the help of various convolutions. System also described it contains some feature selection as well as extraction strategies with classification approaches. Adom [5,9,11] is the inbuilt Optimization algorithm which already provide in deep learning libraries has used in this

entire execution. In conclusion of the research it can applicable for small system where we need automation in real time environment..

L. Cavigelli et. Al. [7] CBinfer: This work basically propose runtime video data capturing from video streaming. This technique has illustrated in [4] and [6]. System creates the multiple video files from video surveillance system and detects the objects in respective frames. System also focus on object movement and detect the anomaly from streaming data..

P. Viola et al. [8] present a approach for helping drivers This approach basically proposed face detection using various feature selection, it is an real time application which captures 15 frames in a second and proceed using deep learning without storing a single object. The time series data has collected with 2-3 seconds and extract heterogeneous features from individual frames. According to supervised learning system already built the train model for respective registered users and classify detection strategy in test module. The system provides accuracy around 90% for a training as well as pre-trained module .

P.S. Paolucci Power et. al. [9] power and momentum of embedded and attendant multi-cores applied to distributed replication of spiking neural networks (NN): arm in n-vidia tegra vs intel xeon quad-cores. This diminutive note regards a association of immediate power, total energy expenditure, effecting time and vigorous cost per synaptic event of a spiking neural network simulator dispersed on MPI processes when perform either on an embedded proposal or a server stage.

L. Cai et. Al. et. al. [10] TEA-DNN: the mission for Time as well as Energy-Accuracy with the help of co-optimized Deep Convolutional Neural Networks (DCNN). This approach provides multi objective data classification using various feature selection technique, this system can be able to work on various kind of data set like structured, semi-structured and unstructured respectively. This work also carried out the strategy of reduces the time, energy and cost during the job execution, which provides efficient time management and Hardware resource utilization in distributed environment. Basically this work also propose a model to utilization maximum bandwidth and Hardware resources using deep learning in entire execution.

Technique	Author	Year	Advantages	Disadvantages
YOLO [11]	Redmon J et. al.	2016	It is too much faster than other NN algorithm even RCNN	System not able to detect multiple objects on same grid, it unable to identify small size objects/
SSD [12]	Liu W et. al.	2016	Faster than faster RCNN Works well for bigger objects, and ability to large data processing	The algorithm cant able to extract heterogeneous or multiple feature from given input objects.
RFCN [13]	Dai J et. al.	2016	Reduce time complexity than existing algorithms. Highest accuracy for all kind of unstructured dataset.	Need more computation resources
FPN [14]	Lin T. et. al.	2017	Rich semantics in all levels	Eliminate the top-down connection and reduce accuracy

DeNet [15]	Tychsen Smith L et. al.	2017	Very faster than RCNN Predefined Background Knowledge not required during classification	More time required to generate the features and for evaluating base network
------------	-------------------------	------	--	---

III. SYSTEM ARCHITECTURE

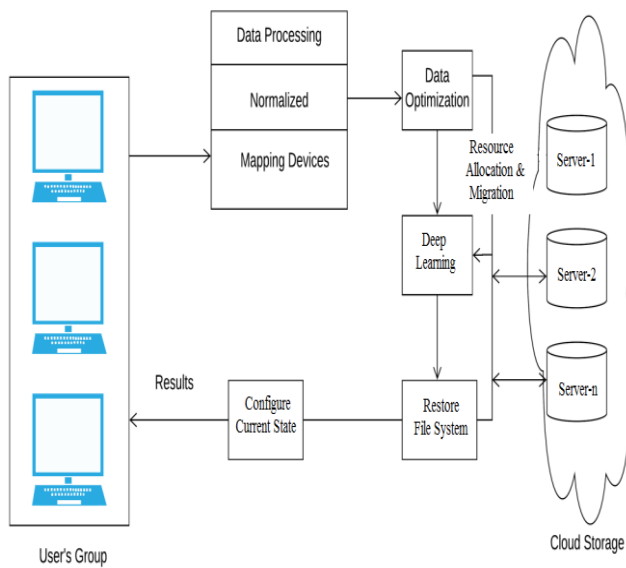


Figure 1: Proposed Block Diagram

The above figure 1 shows proposed system architecture which works for dynamic resource allocation and memory management for mobile platform. System deals with deep learning approach to achieve highest accuracy during the execution. System continuously communicates data centers whenever it need to upgrade the local storage with network storage. The system has train based on available storage of mobile devices, the training phase as done with the combination of NLP and Deep Learning algorithms. This is give assurance which eliminates the storage overhead of local device which improves the performance as well as accuracy respectively. We define machine learning based scheduling technique for mobile offloading schedulers. After investigating the outcome scheduler investigation of available storage system uses tensorflow library for processing the data which reduce the entire time.

We used convolutional neural network algorithm for online scheduling of proposed system. It provides advantages then classical fare schedulers and how system improves the job scheduling time complexity. In current scenario we have multiple remote data service to use as a storage and specify the computing capacity using algorithm strategy. The proposal scheduler not only take a data upload on data service as well as offloading data scheduling but also local execution respectively. The basic advantage of proposed system which provides maximum utilization of storage devices in mobile computing, in Big Data platform many systems has facing storage issue. Using propose approach system can eliminate such gaps with the help of online as well as offline data scheduling. Different inbuilt libraries has used in deep learning framework like tensorflow, caffe, keras etc.

IV. ALGORITHMS

4. Deep Convolutional Neural Network (DCNN)

Input : Training Rules $Tr[]$, Test Instances $Ts[]$, Threshold T .

Output : Weight $w=0.0$

Step 1 : Read each test instance from ($TsInstnace$ from Ts)

Step 2 : $TsIns = \sum_{k=0}^n \{Ak \dots An\}$

Step 3 : Read each train instance from ($TrInstnace$ from Tr)

Step 4 : $TrIns = \sum_{j=0}^n \{Aj \dots Am\}$

Step 5 : $w = WeightCalc(TsIns, TrIns)$

Step 6 : if ($w \geq T$)

Step 7 : Forward feed layer to input layer for encoding $layer[] \leftarrow \{Tsf, w\}$

Step 8 : optimized next hidden layer weight, $Cweight \leftarrow encodingLayer[0]$

Step 9 : Return $Cweight$

In the proposed system we illustrates deep learning base CNN, mainly the system includes two different sections like training and testing respectively. The training phase consists in the initial process has done with Natural Language Processing (NLP) which is used to extract the best feature The entire processing known as data preprocessing and data normalization. The system having ability to work like a supervised learning as well as unsupervised learning, in a first section of execution we introduced feature extraction based on various data attributes and trained the module based on current features. Similarly system deals with testing phase which is basically used for training rules to generate the final outcome. The entire storage platform has used as a input data set for propose system. According to the background knowledge of system we define some strategies to upload current storage load into the network data storage. The input parameters and classify by Deep learning algorithms and generate the possibility based on scheduler to store current application data into the web server.

In the second phase we define auto renewal application according to users need. The application is or entire file system which is already uploaded on web server, system automatically create local instance on respective device. This statement will prove, end user can use all the application based on ad-hoc basis whenever it required. Once entire processing has completed system create dynamically confusion matrix evaluation for different experimental analyses.

V. RESULTS ANALYSS

The system implementation has done in open source base Python platform, python 3.7 version has used on android as well as windows environment. TensorFlow Core r2.0 has used for propose implementation.

The mobile platform is considered for evaluate the performance of system. The DCNN algorithms has used for to achieve the best performance of system.

Below figure and figure 3 illustrates the performance of proposed system viz time for data processing and upload as well as data restore from web server.

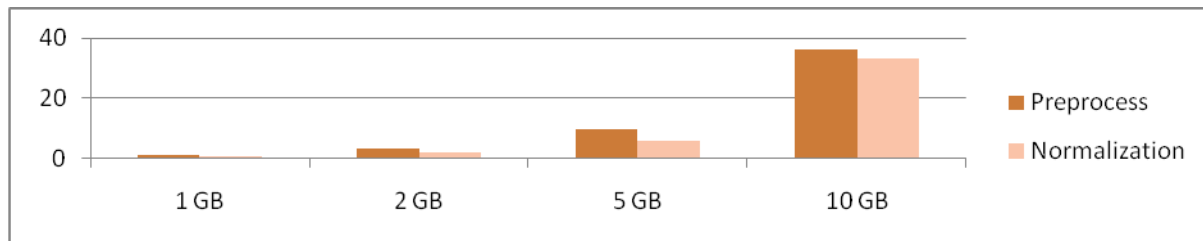


Figure 2 : Time required for data preprocessing and normalization using tensorflow with proposed machine learning algorithm

The above figure 2 shows time required to proposed local file systems with different storage scenario. Time has defined in seconds for process data like 1 GB to 10 GB respectively. This experiment shows whenever data size should be increased it will take more time for given process.

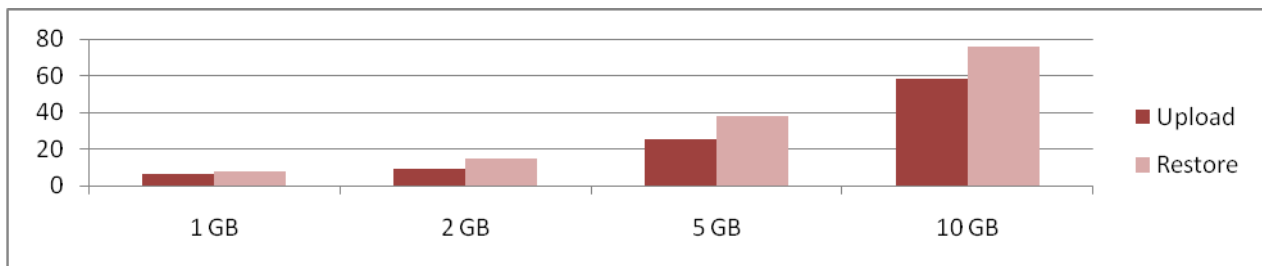


Figure 3 : Time required for data upload and data restoring using tensorflow with proposed machine learning algorithm

The above figure 3 carried out time required for upload data on web server as well as restore in local device respectively. Time has defined in seconds for process data like 1 GB to 10 GB respectively. This experiment shows sometime data upload and download time complexity may be affect according to available resources and internet speed.

VI.CONCLUSION

Basically system illustrates huge deep learning models which is basically provides high accuracy to system, but sometime generate large data overload problem due to hardware resource availability. The proposed CNN based tensorflow provides fast execution over the large data. The feature extraction phase eliminate redundant data processing problem which basically overcome on high time complexity. It also useful for real time applications to achieve the resource matchmaking. System also provides automatic resource management using resource matchmaking approach like right job to right VM, which eliminate allocation time as well as data leakage issues. The system proposed provides dynamic data storage destruction for local devices. The data offloading from current file system upload it into the virtual web servers. The proposed statement also reflects the application can be automatically install on local device whenever it requires to end user. The entire file system has been extracted from web server and automatically install into the device. Some time system can be generate high time complexity issue due to network availability.

REFERENCES

1. J.M. Pierre , Spatio-temporal deep learning for robotic visuomotor control, in: 2018 4th International Conference on Control, Automation and Robotics (ICCAR), IEEE, 2018, pp. 94–103 .
2. S. Leroux , S. Bohez , E. De Coninck , T. Verbelen , B. Vankeirsbilck , P. Simoens , B. Dhoedt , The cascading neural network: building the internet of smart things, *Knowl. Inf. Syst.* 52 (3) (2017) 791–814 .
3. S. Lian , Y. Han , Y. Wang , Y. Bao , H. Xiao , X. Li , N. Sun , Dadu: accelerating Inverse kinematics for High-DOF rRobots, in: Design Automation Conference (DAC), 2017, pp. 1–6 .

4. J. Baek, I.V. Chelu, L. Iordache, V. Paunescu, H. Ryu, A. Ghiuta, A. Petreanu, Y. Soh, A. Leica, B. Jeon, Scene understanding networks for autonomous driving based on around view monitoring system, 2018 arXiv: 1805.07029 .
5. P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, C. Cadena, Leveraging deep visual descriptors for hierarchical efficient localization, 2018 arXiv: 1809.01019 .
6. J. Kaster , J. Patrick , H.S. Clouse , Convolutional neural networks on small un- manned aerial systems, in: Aerospace and Electronics Conference (NAECON), 2017 IEEE National, IEEE, 2017, pp. 149–154 .
7. L. Cavigelli, P. Degen , L. Benini , CBInfer: change-based inference for convolutional neural networks on video data, in: International Conference on Distributed Smart Cameras, 2017, pp. 1–8 .
8. P. Viola , M. Jones , Rapid object detection using a boosted cascade of simple features, in: Conference on Computer Vision and Pattern Recognition, 2001 .
9. P.S. Paolucci, R. Ammendola, A. Biagioni, O. Frezza, F.L. Cicero, A. Lonardo, M. Martinelli, E. Pastorelli, F. Simula, P. Vicini, Power, energy and speed of em- bedded and server multi-cores applied to distributed simulation of spiking neural networks: arm in nvidia tegra vs intel xeon quad-cores, 2015 arXiv: 1505.03015 .
10. L. Cai, A.-M. Barneche, A. Herbout, C.S. Foo, J. Lin, V.R. Chandrasekhar, M.M. Sabry, TEA-DNN: the quest for Time-Energy-Accuracy co-optimized deep neu-ral networks, 2018 arXiv: 1811.12065 .
11. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
13. Dai, J., Li, Y., He, K. and Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems (pp. 379-387). [14] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2117-2125). [15] Tychsen-Smith, L. and Petersson, L., 2017. Denet: Scalable realtime object detection with directed sparse sampling. In Proceedings of the IEEE International Conference on Computer Vision (pp. 428-436).

AUTHORS PROFILE



Mr. Dipak Raghunath Patil did B.E in Computer engineering from North Maharashtra University in 2008 and M.Tech. in Software System from T.I.T.,Bhopal in 2012. He is Currently pursuing Ph.D in Computer Science & Engineering from Suresh Gyan Vihar University Jaipur (Rajasthan). His research field area is Machine Learning, Deep Learning.



Dr. Rajesh Purohit did Electrical Engineering with specialization in Control of Electrical Machines from BITS Pilani in 1993 and M.S from BITS Pilani in 2009. He had done Ph.D. **Birla Institute of Technology & Science, Pilani Campus** in 2014. He is presently working as Principal of School of Engineering & Technology at Suresh Gyan Vihar University Jaipur (Rajasthan).