THE PERFORMANCE ANALYSIS OF DEEP LEARNING NEURAL NETWORKS ON MOBILE DATA

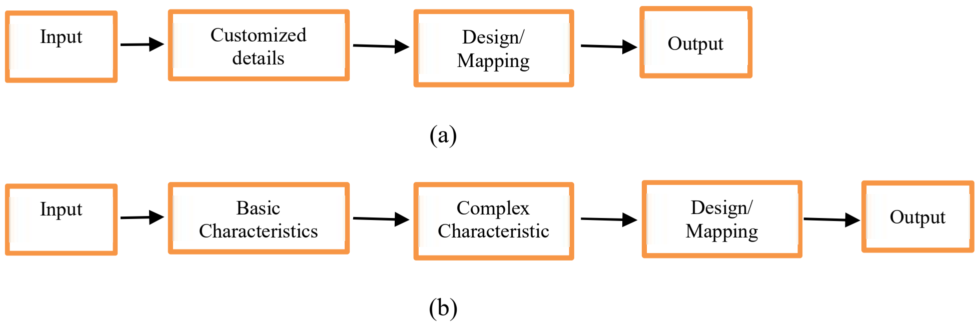
[1]Mr. Rajesh Kumar Singh, [2]Dr Kalpana Sharma
[1]Research Scholar, CS Bhagwant University Ajmer
[2]Associate professor, Department of CSE, Bhagwant University.
rksmps@gmail.com

Abstract: In recent years, the genuine feasibility of training deep learning models on mobile data has been facilitated by the advancements in processing power of mobile technology and the advantages it offers in terms of enabling enhanced user experiences. The predominant focus of contemporary research in mobile machine learning pertains to the inference phase within the realm of deep learning model creation. The exploration of performance characterization for training deep learning models on mobile devices is mostly unexplored, despite its vital importance in the development and deployment of such models. Deep learning approaches have shown superior performance compared to earlier state-of-the-art machine learning methods, particularly in the domain of computer vision. In addition to using deep learning methodologies for tasks like as classification and segmentation, it is also used for the purpose of training neural networks via the utilization of photographic data. As the accuracy of contemporary deep learning models continues to advance, there has been a corresponding increase in their size and depth, which enables them to effectively address certain tasks.

Keywords: Deep Learning, Machine Learning, Neuronal Network

I. Introduction

The proliferation of deep learning models on mobile devices, such as smartphones and smart home hubs, has facilitated the development of many mobile applications, including but not limited to machine translation, speech recognition, cognitive assistance, and street navigation [1]-[3]. The goal of deploying models is to do model inference, which is distinct from model training. Previous research has previously been conducted to examine the efficiency and power requirements of deep learning workloads on mobile devices for model inference [4-10]. The research endeavors play a pivotal role in augmenting the efficacy of deep learning models on mobile platforms.



(a)

(b)

Figure 1. (a) Standard Model for Machine Learning (b)Deep (End-to-End) Machine Learning Model

Due to improvements in mobile hardware's computational capacity and the benefits of allowing great user

DOI: https://doi-ds.org/doilink/10.2023-33621986

experiences, training deep learning models on mobile devices has just been practical. In particular, there is a novel way to put computers to use in the training of deep learning models. To take advantage of these distributed computational resources, training deep learning models is shifting from the cloud to mobile devices. This is especially true with the emergence of artificial intelligence (AI) chipsets and powerful mobile GPU [11]-[14]. In addition, unlike the conventional technique, which requires sending user data to the cloud, building deep learning models on mobile devices may circumvent this problem. Users' privacy may be compromised by the conventional approach (even when utilizing an anonymous dataset and combining it with additional data). By restricting the attack surface to only the device, privacy and security concerns may be drastically reduced during mobile device training for applications where the training aim is determined based on data accessible on each mobile device. Most of the current research into machine learning on mobile devices is concerned with inferencing the results of deep learning models (especially those based on the convolutional neural network (CNN) and the recur- rent neural network (RNN)). Despite the importance of knowing the performance characterization for creating and deploying deep learning models on mobile devices, nothing has been done to far to characterize the performance of training these models on mobile devices.

Recent research [15] investigates how well deep learning models may be trained on remote servers. However, there are key differences between mobile and server-based rainfall that must be investigated independently. First, the user's usual mobile device activities should not be interrupted by the training of deep learning networks; Interference may take the form of longer than anticipated response times for user actions or a decrease in battery life due to the high-power requirements of training deep learning networks. Second, it's probable that the training data will be gathered and utilized on a daily basis. A daily set of photographs gathered by the system might be used to train a deep learning network for image categorization. Third, some big deep learning models (such as ResNet201 and VGG19) have tens of GB memory footprints and are thus not suited to be trained on mobile devices due to the limited memory available.

Feed-on networks, which are sparsely linked to the neural networks, are used in Deep Convolutional Neural Networks (CNNs) [16]. CNNs typically have a linear structure composed of layers. Each layer takes in a dataset called a Function Map (FM) and outputs a new dataset with improved semantics [17]. CNNs have gone through two stages of implementation, similar to those of traditional machine learning methods. The first step of training uses a predetermined set of annotated topics. The restrictions of a fresh training are often set using the weights of a previously trained network during model fine-tuning. These weights are then adjusted to accommodate the new constraint, whether it a smaller training set or lower precision. In the second phase, called inference, the learned model is applied to fresh data samples (i.e., inputs that the model has never seen before). In most setups, convolutional neural networks (CNNs) are trained and fine-tuned once on massive GPU clusters. However, every time a new data sample has to be sorted, an assumption must be made. Therefore, the focus of the study is on finding ways to speed up the inference procedure. In this section, we'll go through the main techniques utilized to speed up the inference [18]. In addition, this work is focused on image classification applications since that is how most CNN accelerators evaluate their effectiveness. This review focuses on using CNNs for specific tasks like object identification and picture segmentation, however the techniques shown here may also be utilized to improve CNN performance in other contexts. [19].

In order for machine learning to function, a network of very basic computer nodes must be constructed to conduct the "output = F (weight*input + bias)" computation and, given a set of known data, determine the

DOI: https://doi-ds.org/doilink/10.2023-33621986

appropriate weights and biases. The model is taught using labeled datasets. Unlabeled (or inferential) incoming data sets may be predicted using the trained model. Tissues of nodes [20] reflect the computational processes on multi-dimensional data arrays [21] in deep neural networks (DNNs). Training and Inference are the two phases of deep learning, a kind of machine learning that draws cues from the way the human brain works. There is no need for a human to define these qualities or to manually program the factors that determine whether a product is excellent or terrible. Self-training occurs inside the neuronal network. After being shown new pictures, the trained neural network then makes an inference about the object's quality and the degree of certainty with which it makes that evaluation [22]. This article introduces the use of artificial neural networks (ANNs) in a common computational intelligence numerical approach for solving second-order nonlinear singular functional differential equations (FDEs) [23].

II. Related Work

In [24], the authors discuss three distinct deep learning approaches for computer vision. Stacked Denoising Auto-encoders (SdAs), Boltzmann Machines, and Convolutional Neural Networks are the three main types of deep learning used in computer vision. Each of these categories has been used to great effect in order to obtain high production rates in various forms of visual comprehension. CNNs have a unique capacity to learn the features, which means they can understand the features automatically based on data. Many computer vision applications also benefit from their stability throughout the shift. But they rely too much on the availability of labeled data, unlike Deep Boltzmann Machines (DBMs) and Deep Belief Networks (DBNs; both DBMs and DBNs belong under Boltzmann family) and SdAs, which can function in an unsupervised fashion. SdAs may be trained under the many real-time scenarios, but DBNs/DBMs and CNNs present significant computing challenges during training. Using Amazon Web Services (AWS) P3, NVIDIA DGX-2, IBM Power System Accelerated Computation Server AC922, and TensorEX GPU, a consumer-grade server,

The authors of [25] compared and contrasted the performance of these and other cutting-edge deep learning systems. Workloads in deep learning are studied, with a particular emphasis on NLP and CV applications. The evaluation of performance is conducted in conjunction with a number of critical factors. Machine learning models with high throughput and good communication efficiency are taken into account. Many potential future use ideas for the different systems are also explored, both independently and in the cloud. The research also includes the effects of several optimization techniques for machine learning and machine architectures.

The authors of [26] analyzed the settings for deep learning object identification and discussed the results. This review begins out with a brief introduction to the history of deep learning and the main technique it employs, CNN. After that, we spoke about the global standard frameworks for object recognition and how to tweak them and add some useful extras to make them even more effective. Salient item identification, face detection, and other uncommon tasks have all been studied because they provide light on these differences. In order to distinguish between the various methods and to derive conclusive conclusions, real-time test analyses are also done. Finally, a variety of promising actions for directing future research in object identification and training systems of neural networks are described. To investigate the efficacy of several deep learning models.

The authors of [27] conducted a significant number of demos on mobile devices (NVIDIA TX2). We provide several commonly used traditional techniques for investigating the performance of machine learning models on mobile devices with respect to resource consumption. In order to understand the performance variance and fine-

DOI: https://doi-ds.org/doilink/10.2023-33621986

grained difficulties, several methods may compare the behavior with compact operations in machine learning models. Understanding the behavior and characteristics is essential for developing and deploying machine learning models for mobile devices.

Text, audio, and video processing; social network survey; and natural language processing are all covered in detail, along with their historical contexts, by the authors of [28]. processing, and a comprehensive look at how machine learning is being used in cutting-edge ways throughout the world. Problems with online education and unsupervised learning were investigated, and it was shown how these first sparks of interest might lead to fruitful new lines of inquiry. Recent developments in the use of computer vision, picture recognition, and deep neural networks to the investigation and analysis of particle collision events at the Large Hadron Collider (LHC) are summarized in [29]. Jet-image theory lies at the heart of the connection between LHC data processing and computer vision methods. Compared to conventional approaches, it has been shown that contemporary picture classification strategies based on neural network frameworks of deep learning significantly improve detection of highly excited electroweak particles. Incorporating a new capacity to understand physics and developing more effective LHC classification methods, novel techniques are implemented for visualizing and understanding the high-level features that deep neural networks have learned to differentiate beyond physically derived variables. With no need for human intervention,

The authors of [30] found a way to tackle the challenge of agricultural product classification. In order to classify photos using Machine Learning, a comparison of popular approaches is performed. The findings in the observational field showed that the suggested approach produced good outcomes. From the perspective of the illustrative model of CNN named AlexNet.

The authors in [31] analysed the GPU behavioral properties of five major machine learning frameworks: Caffe, Theano, TensorFlow, CNTK, and Torch. Some optimization strategies have been proposed to enhance the CNN model created by the enabling setup on the basis of the gathered features. Different convolution methods, such as General Matrix Multiply (GEMM), Fast Fourier Transform (FFT), and direct convolution, have had their GPU output characteristics shown. Overhead and scalability of CNN models in the context of multi-GPU machine learning setups have been analysed. According to the findings, changing the default options provided by the setups may boost the AlexNet model's training speed by a factor of two.

In [32], the authors provided a unified visual analytics framework that can be used to evaluate the efficacy of various machine learning models applied to picture classification tasks in a number of ways. The suggested methodology intends to solve these issues by visualizing misclassification cases and providing a multi-level performance analysis. The user may engage with the technique to compare and analyze several models by switching between views such as ranking, projection, matrix, and instance list views. To prove the efficacy of the suggested strategy, it has been applied to the Modified National Institute of Standards and Technology (MNIST) dataset, where several instances of using different machine learning models have been found. Because of the ease with which data can be collected from sources like the internet and sensors.

The authors of [33] argue that computer systems will undergo a radical shift from traditional data processing to deep learning. According to the results, there are three types of deep learning approaches used in computer vision: supervised, unsupervised, and semi-supervised. Algorithms using k-means clustering and neural networks are particularly popular. Most recently, deep learning has been put to use in computer vision for tasks such as object detection and information extraction from visual data such as photographs and video footage.

DOI: https://doi-ds.org/doilink/10.2023-33621986

The authors of [34] prompted discussion on whether or not it is possible to maintain knowledge of traditional computer vision methods. They also spoke about ways to bring together diverse branches of computer vision. New methods based on hybridization of existing approaches have been tried and evaluated. All of these have shown promise as ways to improve computer vision performance and address problems that machine learning has so far been unable to address. For instance, it has become normal practice to combine traditional computer vision methods with machine learning in newly emerging fields like 3D vision, where the machine learning models have not yet been thoroughly optimised and verified.

Deep examination of the objective detection strategy using Convolutional Neural Networks is offered in [35], which is relevant to all three methods of object detection (salient, objectivity, and category-specific). Validation of standard deep learning settings and guiding principles for object identification tasks is also performed.

III. Types of Networks for Training and Deployment

The training and evaluation of Convolutional Neural Networks often require substantial computational resources due to the unique computational patterns involved, even when utilizing specialized hardware such as graphic processing units (GPUs), digital signal processors (DSPs), or other energy-efficient silicon frameworks. In order to achieve optimal performance in neural network operations, contemporary processors such as Cadence's Tensilica Vision P5 Digital Signal Processor has an extensive array of processing and memory capabilities.

In various image recognition studies, it has been demonstrated that multi-level algorithms, characterized by the execution of diverse filter operations at each level and the utilization of outcomes from previous levels in subsequent deeper levels, exhibit greater efficacy compared to their single-level counterparts. The efficiency of deep algorithms may be significantly improved by using filters that are specifically customized for each level. Multi-resolution filters are a common example of filters that reveal the features of a picture at various resolutions. Upon doing a comparative analysis of satellite imagery depicting metropolitan areas, it becomes evident that the commercial sector exhibits a greater prevalence of skyscrapers and wider thoroughfares in comparison to the residential regions. Various networks with distinct characteristics have arisen, each showcasing their efficacy in semantic image interpretation. Presented below is an enumeration of some prevalent network architectures used in the field of satellite image processing.

Deep Neural Networks (DNNs): According to [36], the layers of these networks have an input layer, at least one hidden layer, and an output layer. Each layer is responsible for its own pixel processing. Deep learning might then be used to describe the training process that follows.

Recursive Neural Networks (RNNs): They must not get confused by the repeating neural networks. These networks, which are also utilized to handle speech and understanding, may be employed successfully after the incoming data has been arranged. Additionally, these networks may be used for real-world scenarios, such as pictures with a recursive architecture [5]. RNNs may thus be used to partition and annotate semantic situations.

Convolutional Neural Networks (CNNs): They were designed for the more accurate categorization of photos with several categories. According to [3], it is possible to assign more than a thousand distinct categories to over a million photos. Five convolutional layers, three fully linked layers, and an infinite number of internal parameters work together to accomplish this. The system conducts regularization to reduce overfitting by ignoring the problematic variables.

DOI: https://doi-ds.org/doilink/10.2023-33621986

Generative Adversarial Networks (GANs): These networks enable the adversarial training of two multilayer Perceptron-based models, G and D, where G controls the distribution of data and D determines the probability that a sample will be acquired from the training data. In addition, D represents multi-dimensional input data for semantic category labelling.

IV. Methods of Neural Network Development and Implementation

As per the proposed approach, subsequent to the analysis of photographs, arbitrary sections of size 256x256 pixels are chosen and subjected to various transformations such as noise addition, distortion, flipping, or rotation. By manipulating the intervals between each step, it is possible to do many rounds of convolution and pooling. Pooling is performed by first applying a mask to each individual pixel, followed by the selection of a singular value from the mask, often the maximum value. During the process of training the model weights, the output of a gradient optimizer is inputted into a SoftMax function. Through the incorporation of non-linearities throughout the learning process using a Rectified Linear Unit (ReLU) and the introduction of random dropouts, the model is capable of autonomously acquiring the requisite decision boundaries for classifying the images into either of the two categories. The network's size undergoes variation as the data passes through it, enabling the training and identification of equivalent properties with different scaling. Convolutional neural networks (CNNs) are known to exhibit robustness against feature transformations such as rotation and translation. This is mostly due to the fact that CNNs apply convolution filters to the whole image throughout the training process. Convolutional Neural Networks (CNNs) have enhanced resilience to feature distortions due to the use of pooling layers.

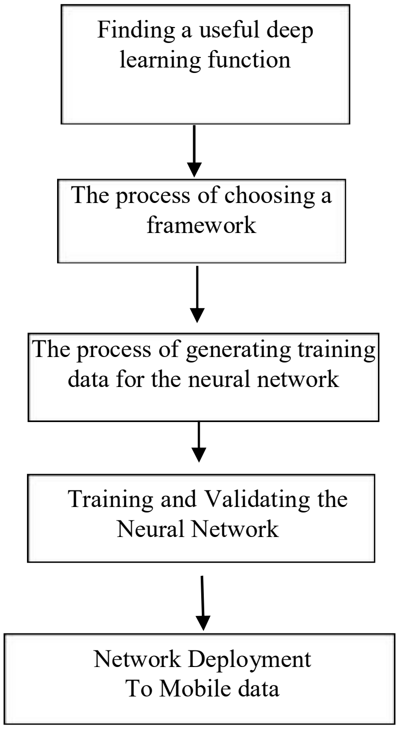

Finding a useful deep learning function
The process of choosing a framework
The process of generating training data for the neural network
Training and Validating the Neural Network
Network Deployment To Mobile data

Figure 2. A Neural Network's Training and Deployment Process

Figure 2 outlines the many processes involved in a neural network's development and deployment.

Step 1: Find the right deep learning operation

The goal of an activation function is to provide the function with a non-linear characteristic. Without the activation functions, the neural network could only carry out the linear mappings between inputs x and output y. When training a neural network for classification tasks, Softmax is only used in the last layer to forecast probability scores. The probabilities are represented by the values of the output neurons, which are constrained to fall between zero and one by the Softmax activation function. It's also important to remember that the numerous categories we use to categorize the input attributes are mutually exclusive. This indicates that there is a unique class for each feature vector x. Classification, identification and localization, segmentation, and anomaly detection are the four most significant deep learning tasks. To classify photographs, you must first arrange them into their respective categories. Images are often classified as either "Pass" or "Fail" depending on their many qualities [37].

Step 2: Select a framework

A neural network's foundational structure, or tool set, generally consists of training and testing utilities. Free, simple frameworks like PyTorch, TensorFlow, and Caffe2 offer excellent documentation and examples, allowing even unskilled users to successfully train and deploy neural networks. It allows for versatile development and deployment, and it makes mobile deployment easier. It has a steeper learning curve than PyTorch, however. Caffe2, one of the first frameworks, is optimized for use on mobile devices thanks to OpenCV's library support for convolutional neural networks and computer vision applications. How complicated and how quickly an inference must be made determines the best structure for a given activity. The inference speed of a neural network decreases linearly with the number of layers [38].

Step 3: Collecting and organizing neural network training data

How many pictures are required for training depends on the neural network and the data it can process. In most cases, the neural network can't be put through its paces without first being trained on a large set of test photos. The more examples of each category are shown to the neural network, the more specific its classifications will become. Synthetic datasets labelled and optimized for neural network training are created by companies like Cvedia. When alternatives are unavailable, people often resort to creating their own pictures and labelling them. Time may be saved by converting a single picture into many by rotating, resizing, extending, and adjusting the brightness and contrast. Several researchers and programmers in the deep learning field have released their image labelling tools as open source and made them available at no cost. Labelling is a graphical image annotation technique that helps label the artifacts in the delimited boxes inside photographs; it is particularly helpful for unmarked datasets. in the other hand, other parties may be keen in getting in on the labelling action. Some deep learning systems only support a restricted range of hardware [39], thus it is very crucial to prepare the training data in light of any specific hardware restrictions or preferences.

Step 4: Train and verify the neural network for accuracy.

At this point, the device is set up and the scripts are run until the training process provides an adequate level of accuracy for the intended application. By keeping them apart, you can ensure that the neural network isn't accidentally taught on the assessment data. This may be sped up by using techniques like transfer learning or reusing a trained network for a different task. In order for a neural network that has been trained for feature extraction to recognize a new feature, all it needs is a fresh set of photos. Open, pre-trained networks may be

DOI: https://doi-ds.org/doilink/10.2023-33621986

found in frameworks like Caffe2 and TensorFlow [40]. In contrast to traditional methods, neural networks can automatically extract relevant characteristics from training data. Matrox Imaging Library (MIL) X is one example of a graphical user interface-based program that may be used to train and deploy neural networks; it is compatible with a number of popular frameworks.

Step 5: Use inference on Mobile data after deploying the neural network.

Finally, a certified neural network is deployed on the chosen hardware, and the output and field data are evaluated and collected. Initial inference stages may be utilized to train subsequent iterations if they are deployed in the field to collect more test data. By using the cloud, you may save a lot of money on hardware and have the flexibility to simply expand, install, and roll out updates to several locations. However, severe faults may be caused by intermittent internet connections, and cloud deployment has higher latency than edge deployment. There are two primary tasks that must be completed before a trained model may be properly deployed.

1)The first task is to feed our model with data it is expecting.

2)second responsibility is to provide actionable results to the customer.

Both the network's structure and the way it was trained influence the kinds of data it will accept as input. To deploy, we must first write code to transform our data into a format that the network can use. Our network's output is a function of its design and its acquired knowledge. During deployment, we must additionally write code to transform the produced output into the form our end user anticipates.

V. Performance Analysis

To train and evaluate a model, data from a dataset must be collected. Datasets are collections of information that are gathered with the purpose of being used in some other program. Training and testing supervised models is best done using labeled datasets. On the other hand, unsupervised models are trained using unlabeled datasets.

Despite the need for specialist knowledge in this interdisciplinary area of performance analysis, deep learning brings new hurdles. In terms of problem spaces, underlying models, and data sources, deep learning models are quite varied. The state of the art in this subject changes rapidly, with new models emerging every few months. Performance criteria, such as accuracy, training time, and inference latency, for deep learning systems are varied and continually changing. Picking the right metrics is crucial. Careful performance analysis is required to eliminate bias and provide meaningful conclusions. A suitable performance study technique should contain a complete collection of benchmarks that reflect both established and new workloads and datasets. The researchers need a thorough familiarity with the cross-stack workloads' resource needs and bottlenecks. Meanwhile, scientists need to be conscious of their increased workloads.

DOI: https://doi-ds.org/doilink/10.2023-33621986

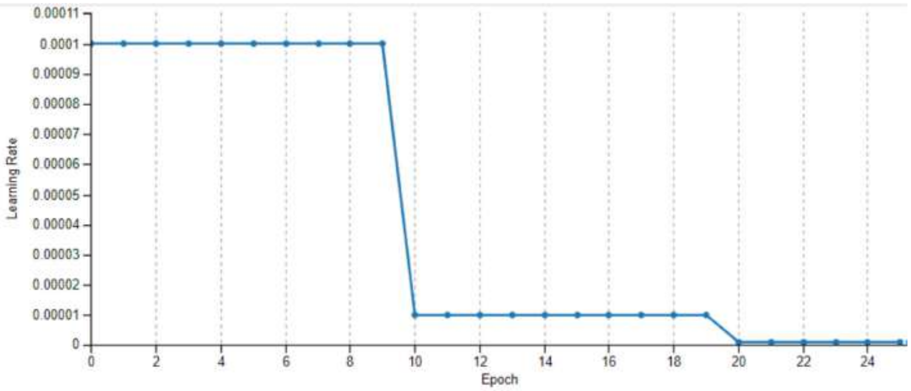


Figure3: Performance Analysis of Learning Ratedur ing Training

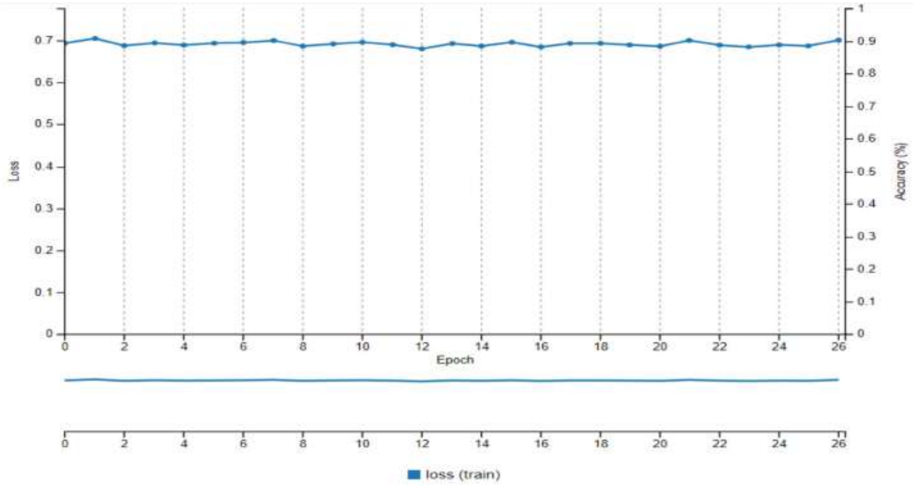


Figure 4. Performance Analysis of Loss rate duringTraining

Figure 3 and Figure 4 show the two key ideas, Learning Rate and Loss Rate, respectively, when it is training. The training rate of a model is defined by the learning rate of its weights. Each weight is changing direction in a way that minimizes the loss by a factor equal to the learning rate. As the training progresses, the network's learning rate falls, indicating that it is coming closer to the optimal answer. At the outset of the training process, the network has no prior knowledge of the possible inputs. As the network improves after many iterations, it becomes less sensitive to individual images. From this already-trained model, a training task is created with an initial learning rate of 0.0001, as seen in Figure 3. As can be seen in Figure 4, training loss is determined at the end of each period.

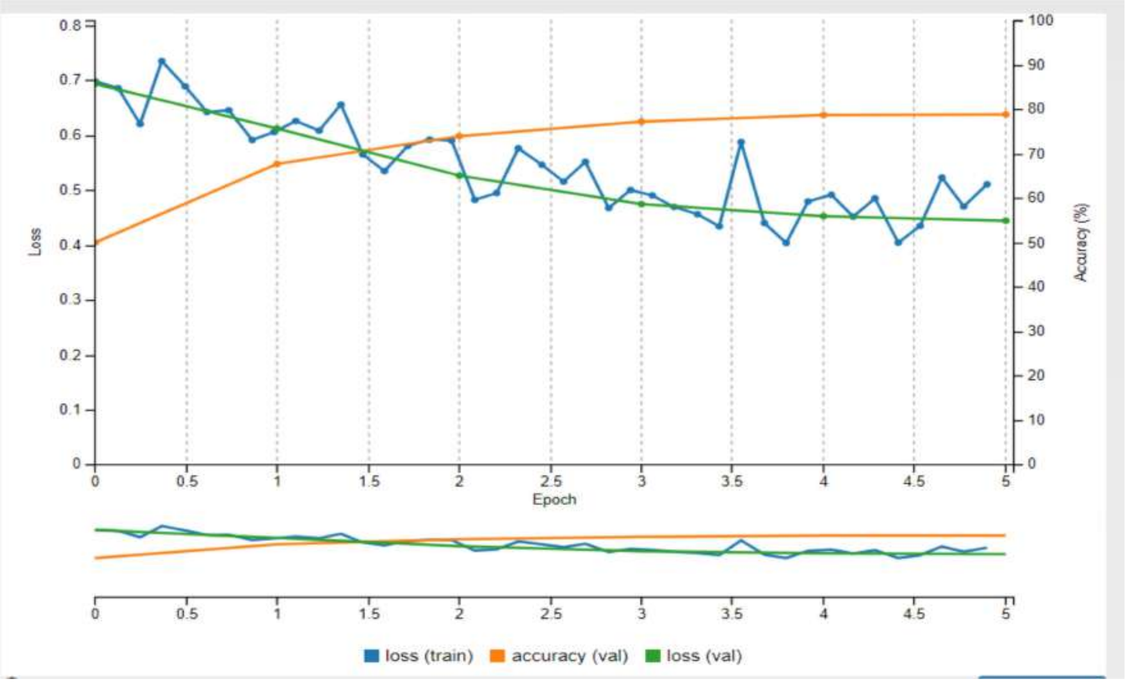


Figure 5. Performance Analysis of Loss, Validation and Accuracy

A good indicator of whether or not the model is learning correctly is the shape of the learning curve it generates from the training dataset. Validation Learning Curve refers to the learning curve that is based on the hold-out validation dataset and gives a notion of how effectively the model generalizes. Optimization Learning Curve [41] is the Learning Curve assessed on the metric from which the different model variables are being optimized, for instance, loss. Performance Learning Curve refers to the curve of improvement in the metric by which the model is evaluated and selected. Figure 10 shows a loss plot, which may be used to infer if the proposed model exhibits similar behavior on training and validation datasets. If these parallel plots start to flourish together, it's time to wrap up training. The accuracy graph reveals room for improvement in the model, since the trend toward correctness has been rising across both datasets throughout the recent past. It is also clear from the training dataset that the model is equally proficient on both datasets, suggesting that it has not learned anything new so far.

VI. Conclusion

The development of deep learning networks has been driven by the enhanced processing capabilities of mobile devices and the benefits associated with delivering a superior user experience. Recognizing the significance of performance characterization in the development and execution of deep-learning models is crucial, even when the behavior and attributes of training devices are mostly non-transversal. The primary objective of this research is to assess the efficacy of contemporary deep convolutional networks. This paper presents a collection of benchmarks designed for the purpose of evaluating the development and implementation of Deep Neural Networks. The increasing popularity of training deep learning networks on mobile devices may be attributed to the advantages associated with offering satisfactory user experiences and the expanding processing capabilities of mobile hardware. Despite the significance of training and deploying deep learning models on mobile devices,

DOI: https://doi-ds.org/doilink/10.2023-33621986

there exists a dearth of knowledge on the performance attributes associated with training such models on these devices. This study is the first attempt to thoroughly examine the effectiveness of training deep learning networks on a mobile device. The foundation of our study is based on a comprehensive set of mobile device profiling technologies, which are used in conjunction with a diverse range of generic deep learning models. Upon conducting our analysis, we have identified numerous domains that need further exploration. The objective of our study is to stimulate further investigation into enhancing the efficacy of training deep learning networks on mobile data.

References:

[1] Sindhwani, N.; Verma, S.; Bajaj, T.; and Anand, R.; “Comparative Analysis of Intelligent Driving and Safety Assistance Systems Using YOLO and SSD Model of Deep Learning.” *International Journal of Information SystemModeling and Design* 12, no. 1 (2021): 131-146.

[2] Kamalraj, R.; Neelakandan, S.; Kumar, M. R.; Rao, V. C. S.; Anand, R.; and Singh, H.; “Interpretable filter based convolutional neural network (IF-CNN) for glucose prediction and classification using PD-SS algorithm.” *Measurement* 183: 109804.

[3] Ren, A.; Li, Z.; Ding, C.; Qiu, Q.; Wang, Y.; Li, J.; ... and Yuan, B.; “Sc-dcnn: Highly-scalable deep convolutionalneural network using stochastic computing.” *ACM SIGPLAN Notices* 52, no. 4 (2017): 405-418.

[4] Gupta, A.; Yadav, V.; “Hybrid Intelligence Model on the Second Generation Neural Network”, *International Journal of Advanced Intelligence Paradigms*, Vol. 18, No. 3, pp. 398-416, February 2021.

[5] Rhu, M.; Gimelshein, N.; Clemons, J.; Zulfiqar, A. and Keckler, S. W.; “vDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design.” In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),* pp. 1-13. IEEE, 2016.

[6] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; .... and Fei-Fei, L.; “Imagenet large scale visual recognition challenge.” *International journal of computer vision* 115, no. 3 (2015): 211-252.

[7] N. D. Lane, P. Georgiev, and L. Qendro, “Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learn- ing,” in Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 283–294, ACM, 2015.

[8] A. Mathur, N. D. Lane, S. Bhattacharya, A. Boran, C. Forlivesi, and F. Kawsar, “Deepeye: Resource efficient local execution of multiple deep vision models using wearable commodity hardware,” in Proceedings of the 15th Annual International Conference on Mobile Systems, Applica- tions, and Services, pp. 68–81, ACM, 2017.

[9] M. Xu, M. Zhu, Y. Liu, F. X. Lin, and X. Liu, “Deepcache: Principled cache for mobile deep vision,” in Proceedings of the 24th Annual Inter- national Conference on Mobile Computing and Networking, pp. 129–144, ACM, 2018.

[10] Z. Lu, S. Rallapalli, K. Chan, and T. La Porta, “Modeling the resource requirements of convolutional neural networks on mobile devices,” in Proceedings of the 25th ACM international conference on Multimedia, pp. 1663–1671, ACM, 2017.

[11] Anand, R.; Singh, B.; and Sindhwani, N.; “Speech Perception & Analysis of Fluent Digits’ Strings using Level- By-Level Time Alignment.” International Journal of Information Technology and Knowledge

DOI: https://doi-ds.org/doilink/10.2023-33621986

Management 2, no. 1 (2009): 65-68.

[12] Juneja, S.; and Anand, R.; "Contrast Enhancement of an Image by DWT-SVD and DCT-SVD." In Data Engineering and Intelligent Computing, pp. 595-603. Springer, Singapore, 2018.

[13] Anand, R.; Shrivastava, G.; Gupta, S.; Peng, S. L.; and Sindhwani, N.; "Audio Watermarking With Reduced Number of Random Samples." In Handbook of Research on Network Forensics and Analysis Techniques, pp. 372-394. IGI Global, 2018.

[14] Umar, M.; Sabir, Z.; Raja, M. A. Z.; Baskonus, H. M.; Yao, S. W.; and Ilhan, E.; "A novel study of Morlet neural networks to solve the nonlinear HIV infection system of latently infected cells." Results in Physics 25 (2021): 104235..

[15] Parveen, H,; Yadav, V; "A Particle Swarm Optimization Strategy using QSAR modeling on the Second generation Neural Network", IEEE International Conference on Advances in Computing, Communication Control & Networking (ICACCCN-2018), pp. 1188-1199, 2018.

[16] Umar, M.; Sabir, Z.; and Raja, M. A. Z.; "Intelligent computing for numerical treatment of nonlinear prey– predator models." Applied Soft Computing 80 (2019): 506- 524.

[17] Sabir, Z.; Raja, M. A. Z.; Umar, M.; and Shoaib, M.; "Neuro-swarm intelligent computing to solve the second- order singular functional differential model." The European Physical Journal Plus 135 no. 6 (2020): 1-19.

[18] H. Zhu and Y. Jin, "Multi-objective evolutionary federated learning," arXiv preprint arXiv:1812.07478, 2018.

[19] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," arXiv preprint arXiv:1806.00582, 2018.

[20] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated dis- tillation and augmentation under non-iid private data," arXiv preprint arXiv:1811.11479, 2018.

[21] X. Qi and C. Liu, "Enabling deep learning on iot edge: Approaches and evaluation," in 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp. 367–372, IEEE, 2018.

[22] W. Du, X. Zeng, M. Yan, and M. Zhang, "Efficient federated learning via variational dropout," 2018.

[23] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous net- works," arXiv preprint arXiv:1812.06127, 2018.

[24] Voulodimos, A.; Doulamis, N.; Doulamis, A.; and Protopapadakis, E.; "Deep learning for computer vision: A brief review." Computational intelligence and neuroscience, 2018. https://doi.org/10.1155/2018/7068349

[25] Ren, Y.; Yoo, S.; and Hoisie, A.; "Performance analysis of deep learning workloads on leading-edge systems." In 2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), pp. 103-113. IEEE, 2019.

[26] Zhao, Z. Q.; Zheng, P.; Xu, S. T.; and Wu, X.; "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30, no. 11 (2019): 3212-3232.

[27] Liu, J.; Liu, J.; Du, W.; and Li, D.; "Performance analysis and characterization of training deep learning models on nvidia tx2." arXiv preprint arXiv 1906 (2019): 04278.

[28] Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M. P.; ... and Iyengar, S. S.; "A survey on

DOI: https://doi-ds.org/doilink/10.2023-33621986

deep learning: Algorithms, techniques, and applications." ACM Computing Surveys (CSUR) 51, no. 5 (2018): 1-36.

[29] Schwartzman, A.; Kagan, M.; Mackey, L.; Nachman, B.; and De Oliveira, L.; "Image Processing, Computer Vision, and Deep Learning: new approaches to the analysis and physics interpretation of LHC events." Journal of Physics: Conference Series 762, no. 1 (2016): 012035.

[30] Mouhssine,R.; Otman, A.; and Haimoudi, E.K; "Performance Analysis of Machine Learning Techniques for Smart Agriculture: Comparison of Supervised Classification Approaches." International Journal of Advanced Computer Science and Applications, 11, no. 3 (2020): 610-619.

[31] Kim, H.; Nam, H.; Jung, W.; and Lee, J.; "Performance analysis of CNN frameworks for GPUs." In 2017 IEEE

[32] Rahul, M,; Yadav, V,; "Zernike Moments based Facial Expression Recognition using Two staged Hidden Markov Model", Advances in Computer Communication & Computational Sciences, Vol. 924, pp. 661-670, 2019.

[33] Khan, A. I.; and Al-Habsi, S.; "Machine learning in computer vision." Procedia Computer Science 167 (2020): 1444-1451.

[34] O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G. V.; Krpalkova, L.; ... and Walsh, J.; "Deep learning vs. traditional computer vision." In Science and Information Conference, pp. 128-144. Springer, Cham, 2019.

[35] Suhail, A.; Jayabalan, M.; and Thiruchelvam, V.; "Convolutional neural network based object detection: A review" Journal of Critical Reviews 7, no. 11 (2020): 786- 792.

[36] Sze, V.; Chen, Y. H.; Yang, T. J.; and Emer, J. S.; "Efficient processing of deep neural networks: A tutorial and survey." Proceedings of the IEEE 105, no. 12 (2017): 2295-2329.

[37] Mr. Pathan Ahmed Khan, Dr. M.A Bari,: Impact Of Emergence With Robotics At Educational Institution And Emerging Challenges", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 6, Issue 12, December 2021,Page 43-46

[38] Dr. Abdul Wasay Mudasser, Dr. Pathan Ahmed Khan, "Artificial Intelligence Usage in Wireless Sensor Network: An Overview", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 7, Issue 10, October 2022,Page 9-14.

[39] Sharma, D.; and Kumar, N.; "A review on machine learning algorithms, tasks and applications." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 6, no. 10 (2017)): 2278-1323.

[40] LeCun, Y.; Bengio, Y.; and Hinton, G.; "Deep learning." Nature 521 (2015): 436-444.

[41] Ibrahim, Zahid, Sumair Khan, faizan,, "Smart Apartment Building Managed By Artificial Intelligence, Including Additional", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 7, Issue 11, November 2022,Page 1-8.

[42] Abu, M. A.; Indra, N. H.; Abd Rahman, A. H.; Sapiee, N. A.; and Ahmad, I.; "A study on Image Classification based on Deep Learning and Tensorflow." International Journal of Engineering Research and Technology 12, no. 4 (2019): 563-569.

[43] Zhu, S.C.; and Mumford, D.; "A stochastic grammar of images. Foundations and Trends in Computer Graphics and Vision 2, no.4 (2006): 259-362.

DOI: https://doi-ds.org/doilink/10.2023-33621986

[44] Sindhwani, N.; "Performance Analysis of Optimal Scheduling Based Firefly algorithm in MIMO system." Optimization 2, no. 12 (2017): 19-26.

DOI: https://doi-ds.org/doilink/10.2023-33621986