Deep Machine Learning In Neural Networks

Basheer A. Hassoon, Mushtaq A. Hasson

Abstract: A major challenge in neural network is computationally and memory intensive. To solve this difficult we explained deep neural network. In machine learning models, we explained and compared Deep Neural Networks (DNN's) and Deep learning methods. This paper mainly contains the Deep Compression in three stages of pipeline. Such us trained quantization, Huffman coding and pruning. In this method, compressed the neural networks are done without affecting accuracy. The main aim is to maximize the energy and storage, and its required to run interprets on such large networks. Both compression and learning algorithms are discussed. We estimated the large scale deep neural network applications using multiple GPU machines. Various datasets are compared in this survey.

Index Terms: Deep compression, Deep neural network, Heterogeneous Networks, Machine learning algorithm, Network Pruning, Server, and Trained Quantization.

1 Introduction

This chapter contains the details of the distributer deep neural network, deep compression and various machine learning algorithms. [1] In deep learning vision, its application improves the control of heterogeneous network traffic. In deep learning the characterizing is one of the difficult task. In this approach produce the characterizing of appropriate input and output in heterogeneous network traffic. It proposed deep neural network system and explained how it is varied from traditional neural networks. In results compared, proposed deep learning system with benchmark routing strategy based on signaling overhead, throughput, and delay. [2] Machine learning algorithm is used to solve the computer science problems and it contains many applications. Deep learning algorithm contains limited number of applications. And it is not applying in heterogeneous network control. The scale and complexity of machine learning (ML) algorithms becomes large when working with very deep layers.

Proposed Learning system contains three steps such us

- Initial phase
- Training phase
- Running phase.

In deep learning, the initial phase generates the data for training phase. In training phase, its trains the learning system collected from initial phase. Running phase executes the fixed time intervals using routing algorithm.

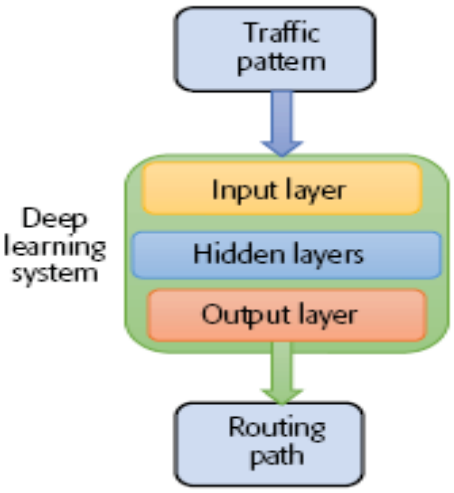

Figure 1: Layers in Deep learning Organization of Paper

The remaining division is specified as mentioned below: section 1 provides the explanation of deep learning in network servers; Section 2 provides the brief explanation of the existing methods of compression techniques and compared various frameworks. Section 3 defines the comparison of various machine learning algorithm Section 4 explains the conclusion of learning methods with neural networks.

2 Deep Learning with Severs

2.1 A primer on deep learning

To recover the classification problems and pattern recognition, the deep neural networks have the highest accuracy between machine learning models. But it contains most expensive for training. A DNN is defined as a set of neurons and it's characterized in layers, and successive layer of input is called as output. Deep neural networks training are achieved in several period; normally the weights w are determined using back propagation algorithm. With human knowledge of transfer to the dataset for neural network for learn the correlation between labels and data. This process is supervised learning [3].

2.2 DNN systems with parameter servers

The deep neural networks consists millions of features and these features are organized into layers and they were trained by repeat iteration with high accuracy. The training time is

• *Basheer A. Hassoon is currently pursuing university of Basra. E-mail: basheerahassoon8@gmail.com*
• *Mushtaq A. Hasson is currently pursuing university of Basra*

required to maintain the same accuracy and execution process on clusters changing from tens to thousands of machines[4] proposed reinforcement learning algorithm for scheduling. And this scheduling, using this algorithm when the problem occurs in distributed system. The machine learning algorithm obtains heterogeneity of the nodes, and it also determines the scheduling policy for better execution system. Deep learning and neural network methods are the greatest important concepts in Artificial Intelligence (AI), and it produces most efficient solutions than heuristic approaches. If any problem occurs in distributed systems, the reinforcement learning algorithms package will works with machine learning. This reinforcement algorithms schedule the tasks for cluster of computers. This paper mainly explored the scheduling problem using learning algorithms. The proposed machine learning algorithm used as a name in MLBox, and it offers the scheduling services. [5] Experiments shows the machine learning algorithm can attain greater performances in task scheduling. A classic reinforcement learning algorithm has been shown in temporal difference, that can able to solve the scheduling problem. Scheduling defines the process of allocating resources to the tasks. The paper contains the problem addressing and right scheduling mechanism choosing. [6] Many machine learning models depends a dynamic control flow for inference and training. The recurrent neural network models and reinforcement learning depends on the recurrence relations. The machine learning system must reinforce the dynamic control flow of heterogeneous environments and distributed systems. This paper contains programming model for distributed machine learning. And it also supports flow of dynamic control. To represent the machine learning models, the dataflow graphs are used in this approach. To run on a set of heterogeneous devices, the conditional branches and loop bodies are divided across many machines. After that the program was written to proposed model. The automatic differentiation and distributed gradient computations were necessary for training the machine learning models. This evaluation proved the performance and scalability and it is used for many real world applications. Advanced machine learning and their applications are focused conflicting design objectives for the underlying systems. This system can achieve scalable and also achieve usage of hardware resources in efficiently. For modern machine learning system, it's used Tensor Flow as representative system architecture. The core Tensor Flow runtime was developed in C++ for portability and performance. The memory demands are challenge in on specialized devices such as GPUs and it uses some of the algorithm to rectify this. [7] defined the complete distributed machine learning in heterogeneous environments. In homogeneous environments, the distributed systems have minimum performance. In heterogeneous environment, the stragglers are common because of synchronization protocols cannot fit a heterogeneous setting. This paper proposed heterogeneity algorithm and it is used to constant learning rate schedule and sophisticated learning rate schedule for global parameter, and it also allows to suppress stragglers. They proved the valid convergenceFor both learning rate schedule and implemented a prototype system. Using machine learning workloads, the performance of this prototype are validated. The proposed prototype has two times faster than other systems (TensorFlow, Spark and Petuum,). The mentioned algorithm extracts the limited iterations to converge. [8] TensorFlow differs from batch dataflow systems in two respects:

- In overlapping subgraphs of the overall graph, the model supports multiple concurrent executions.
- Individual vertices may have mutable state that can be shared between different executions of the graph. [9] introduced three stage pipeline of pipeline in deep compression, such us trained quantization, Huffman coding, pruning, and [10-12]. These stages are worked together to reduce the neural networks storage requirements by without affecting the accuracy. In this method the first, prunes process is done by learning in network. Second one is quantizing the weights to imposed weight sharing. The remaining networks are retained in final and the quantized centroids. Also the quantization reduces the number of bits. In ImageNet dataset, the storage is reduced from 240MB to 6.9MB using Alex Net. The proposed method had the features of better privacy, less network bandwidth. The deep neural networks have the disadvantage of energy consumption. For problem solving the genetic algorithm defines the following steps

1. Initialization,
2. Selection,
3. Crossover,
4. Mutation,
5. Proceeding fitness.

3 Deep Compression and Methods

[13] Deep compression defines, to reduce the storage requirements of neural networks. And this process is done by without affecting the accuracy of data. 1. Pruning the network by important connections learning; 2. For enforce weight sharing it quantizes the weights; 3. It uses Huffman coding. Network pruning contains the steps,

- Through normal network training, it learns the connectivity
- Pruning the connections of small-weight
- Retraining the network, for final weights learning

Also it have the disadvantage of lot of computational resources contains wastage. [14] Typically by using some backprop-based algorithms , the deep artificial neural networks (DNNs) are trained such as policy gradients and Q-learning on challenging of Deep Reinforcement Learning (DRL) algorithm.[15] described Ako, and decentralised dataflow-based DNN system without using parameter servers, and it is designed for cluster resources saturation. [16] proposed efficient framework ThiNet, to compress the CNN models in pair of training and inference stages, it has most generalization ability. To reduce the network complexity, they established filter level pruning, and it consists prune filter. The experimental result examines the effectiveness of this filtering strategy. They also showed the ThiNet performance in ILSVRC-12 benchmark. Proposed framework reduces more than half of the parameters. But it have the disadvantage of accuracy loss compared with AlexNet. The process was done through

- Filter selection,
- Pruning,
- Fine-tuning.

3.1 Network Pruning or Sharing

To compress the CNN models, the network pruning act as important source. It's used to reduce the complexity of network. [17] Pruning the minimum weight of connections is done by normal network training of connections. Below

threshold, all the connections have weights that are removed from the network. Pruning reduces the number of parameters by 9 times and 13 times with that of. Network pruning proved the performance to minimize the over-fitting and network complexity. [18] proposed the Biased Weight Decay method for pruning. [19] used Hessian of the loss function, and it reduce the number of connections.[20] for parameter sharing they explained Hashed Nets model, its used for small-cost hash function, the group of weights depends hash buckets.

3.2 Trained Quantization and Weight Sharing

The pruned network was compressed by network quantization and weight sharing. It's done by reducing the number of bits needs to represent each weight. The multiple connections are share the same weight and number of weights are stored. [21] showed that 8-bit of the quantization without affecting accuracy and gives the best result in significant speed-up. Based on neural networks [22] used 16-bit fixed-point representation training. The method used in [9], the link weights are quantized by sharing of weight, after the Huffman coding, and it is done for the further reduction of quantized weights. When working with large convolutional neural networks, the binary nets achieved minimum accuracy. These binary nets are depends on simple matrix approximations and break the binarization outcomes. These was found to be the drawback.

3.3 Initialization of Shared Weights

[13] The quality of clustering is based on centroid initialization. It reduces and affects the accuracy of network's prediction. It contains three initialization methods,

- Forgy method
- Density-based method
- Linear initialization

These methods have high compression errors and low compression ratios. [10-12] The deep neural networks (DNNs) have the demand on quality analysis. DNNs consits millions of parameters in an unparalleled representation and the deeper or larger DNNs can improve the analysis of data. Proposed DeepSZ, it involves four key steps to compress the neural network framework. The optimization, network pruning and error bound assessments are done by error bound configuration, for compression of model generation, and it have maximum compression ratio and minimum encoding time. For each layer the adaptive approach was developed to detect the reasonable error bounds. The optimization algorithm acts as efficient and it was developed to control the best-fit configuration of error bounds and finally achieves compression ratio maximization. This paper explains the model for estimation, that minimize reduction of accuracy assumption and it depends on the degradation of inference accuracy. The evaluation results proved the DeepSZ can improve the compression ratio compared with other techniques. [23] To compress the neural network, presented the new approach for improving the pruning method. The proposed novelty of the pruning technique, that allows best performance of pruning during the back propagation phase of the network training. The evaluation results proved the joint optimization with pair of thresholds and network weights are achieved the higher compression rate. Proposed approach reduces the pruned network weights in number of 14% to 33%. The features are extracted by a pruned network using transfer learning tasks. To achieve this they showed learned representations using the proposed pruning methodology. This process is done without effectiveness. [16] Proposed Thinet framework, in inside a deep neural network the overall framework worked as similar in pruning filters. Approached the pre-trained model, it was pruned layer by layer with a pretend compression rate.

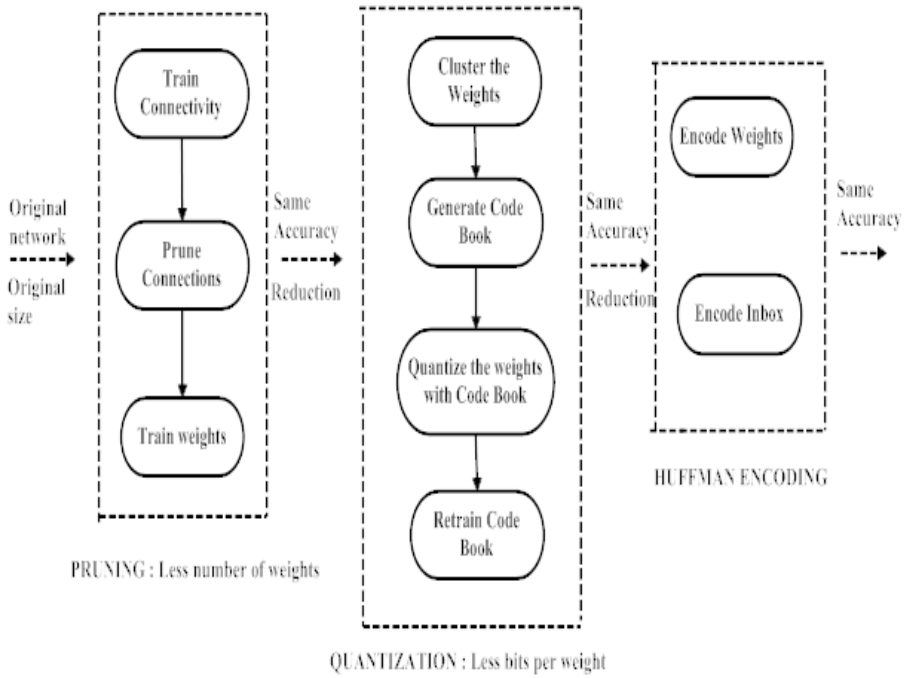

Figure 2 : *Three stage pipeline: pruning, weight sharing and quantization.*

3.4 Filter selection and pruning

The process of pruning of layer in filters was done. The pruned

network had same structure using few filters and channels. [24] The method “ThiNet” defines, the original wide network, and it becomes much thinner.

3.5 Fine-tuning

If any problem occurs in generalization ability, the fine tuning process works necessary by filter pruning. But it will extract the large number of datasets to reduce the complex models. For time saving they defined fine-tune one or two times later the pruning of one layer. Quantization was performed to minimize the number of bits, and it needs to estimate each shared weight value. [25] To convert floating point weight values, they have used fixed point quantization. For compression, they also explored ResNet50 on proposed powerful CNN architecture. Pruned prune ResNet-50, similar to VGG-16. After pruning, the fine tuned in one epoch with fixed learning rate ten to four. [26] For image and speech recognition, the training of Deep Neural Networks (DNNs) with large amounts of data improves the accuracy of machine learning models. This paper explained the decentralized dataflow-based DNN system without using parameters. The training of DNNs was achieved by scalable distributed systems in parallel on compute clusters. To achieve the fastest time-to-convergence, a DNN system have two performance aspects: High hardware efficiency: With more workers, it have multiple parallelism, so it have minimum iteration time; High statistical efficiency: To improve statistical efficiency, the DNN systems scale out through multiple parameter servers. [15] The fastest time-to-convergence workers and parameter servers should have optimal resource allocation. [27] Proposed structured quantization method and it represents in 2D convolution kernels. It performs clustering on kernels with centroids. So it used compressed model with a set of centroids, and with a corresponding cluster index per each kernel. Proposed compressed model follows many benefits. CNN compression was done by using k-means clustering and 2D kernels. The ResNet framework was used to compressed CNN. It achieves high accuracy on ILSVRC12 image.

Table 1 *Various machine learning approach for model compression*

Theme Name	*Description*	*Applications*	*More details*
Sharing and Pruning of parameters	*Redundant parameters are reduced which are not sensitive to the performance*	*Fully connected layer and convolutional layer*	*Achieves the good performance in various settings, and it can reinforce either train and pre-train from scratch.*
Low-rank factorization	*For estimating the informative parameters by using tensor and matrix decomposition*	*Fully connected layer and convolutional layer*	*Standardized pipeline was easily estimated, and it can reinforce both train and pre-train from scratch.*
Compact convolutional filters and transferred convolutional filters.	*The convolutional filters are saved by using special structural*	*Single convolutional layer*	*Algorithms are basically achieved good performance based on their applications, and it only reinforce training from scratch*
Knowledge distillation	*Training the solid neural network, its achieved by large model of distilled knowledge*	*Fully connected layer and convolutional layer*	*Reproduction performances are highly sensitive with their applications and network structure, reinforced train from scratch*

A CNN comprises three main types of neural layers, namely, (i) Fully connected layers (ii) Convolutional layers, (iii) Pooling layers. On the other side the network quantization had limited attempts of the precision of parameters. The main goal of the quantization is, it also quantized intermediate features, neural networks and gradients [28]To maintain the assignment of each kernel, the training method is used after clustering. Clustering kernels from various models they applied VGG-16 variant first, it defines 13 convolution layers, and 3 fully-connected layers. Normalization is conducted to all convolution layers. They also compressed more recent architectures, ResNets [29] and DenseNets [30]. Similar to VGG-16, the 3 × 3 kernels are also dominant in those models. [31] Proposed a Associative Compression Networks (ACNs),

and a new framework are introduced for variation auto encoding with neural networks. There are three datasets are compared in this paper. For the CIFAR-10 experiments the encoder after VGG-style classifier with 11 convolutional layers and 3x3 filters. ACN codes for [32] CIFAR-10 images were linearly classified with 55.3% accuracy versus 38.4% accuracy for pixels. Experiments the setup was the same as for CIFAR-10, except the decoder had 20 gated residual layers of 36, 5x5 filters. ACN ImageNet codes can be linearly classified with 18.5% top first accuracy and 40.5% top fifth accuracy, and compared to 3.0% and 9.0% respectively for pixels. The CelebA images to 32x32 resolutions and the same setup as for CIFAR-10. [33] proposed a novel dataflow-based joint quantization approach to increase the Deep Neural Networks (DNNs). It makes the quantization operations to reduce the information loss. Evaluation results proved the effectiveness of quantized model with Image Net and KITTI. The main aim of this paper is to reduce energy consumption. However, these quantization methods reduce the model complexity, it have some disadvantages: 1. additional hardware costs are required. 2. Noticeable performance drop exists. To evaluate the hardware cost they created RTL model for each method. Also it showed the Mean Squared Error (MSE) between activations of quantized and floating-point activations. [34] in computationally intensive and memory intensive the neural networks makes difficult in embedded systems. There are three stages worked together for reducing the requirements of storage, such us pruning, trained quantization and Huffman coding. This paper clearly explained about these three stages. Pruning, it reduces the number of connections, Quantization, it reduces the number of bits, and they used Image Net dataset with no loss of accuracy. [35] In deep learning technique the compression and efficiency acts as two hands. They have used Alex net framework, in this framework consider 91% of the computation and accounts for 4% only. In network pruning, the unnecessary connections are removed and larger network is used for smaller network training. This paper explained Bayesian perspective network pruning and reduce the bits with achieving high accuracy. Paper contains two approaches 1. To prune nodes they used hierarchical priors by replacing individual weights, 2. To detect the Optimal fixed point precision, used the posterior uncertainties. [36] introduced binary connect method for training a DNN, with binary weights. It contains forward and backward propagations, deep neural networks achieves wide range of tasks with large number of training set. [37] proposed the novel binary scheme for weights, in forward and backward propagations.

Table 2 *Outline of Baseline models with illustration works of Network Compression.*

Models	*Representation*
AlexNet [38]	*Structural matrix, minimum rank factorization*
Network in network	*Low-rank factorization*
Visual Geometry Group nets	*Transferred filters, minimum rank factorization*
Residual networks [39]	*Compact filters ,stochastic depth, parameter sharing*
All – Convolutional neural network nets	*Transferred filters*
LeNets[9]	*Parameter sharing and pruning*

4 Applications and Discussions

4.1 Combining Learning with Compression Algorithms

To improve the network compression ratios and efficiency with the attempts of learning algorithm applying,[40] in machine learning the genetic algorithm improves the quantization process to achieve the optimal DCT compression. [41] proposed reinforcement learning algorithm that is able to learn neural network dynamics functions. Deep neural networks (DNNs) [42] have revolutionized the accuracy of machine learning models in many areas, [43]compared DNN's architectures with various metrices and it have some advantages, such us

- High accuracy,
- Operations count,
- Power consumption
- Parameters,
- Memory footprint, and
- Inference time.

Machine learning combines the symbolic expression with tensor computation, it increase efficiency and flexibility. The modernized traditional machine learning techniques are outperformed in many fields. Some of the most significant deep learning schemes used in computer vision problems that are (CNN) Convolutional Neural Networks, (DBM) Deep Boltzmann Machines, (DBN) Deep Belief Networks and (SDA) Stacked Denoising Autoencoders.

Table 3 *Comparison of machine learning, based on compression and clustering mechanisms*

Mechanisms	*Machine learning algorithm(S)*	*Balancing energy consumption*	*Overhead*	*Topology aware*	*Complexity*	*Delay*
Clustering of Large scale network [44]	*Neural Networks*	*Present*	*Minimum*	*Present*	*Average*	*Maximum*
Cluster head election	*Directory Traversal*	*Present*	*Minimum*	*Present*	*Minimum*	*Minimum*
Gaussian process models for censored sensor readings	*Ground Penetrating Radar*	*Un present*	*Average*	*Un present*	*Average*	*Average*
Adaptive sampling		*Present*	*Minimum*	*Un present*		*Maximum*

[45]						
Clustering using SOM and sink distance	*Self Organizing Feature*	*Un present*	*Average*	*Present*	*Maximum*	*Maximum*
Online data compression [46]	*Learning Vector Quantization*	*Un present*	*Maximum*	*Present*	*Maximum*	*Maximum*
Data acquisition using compressive sensing	*PCA*	*Present*	*Maximum*	*Present*	*Average*	*Maximum*
Transmission reduction		*Un present*	*Maximum*	*Presented*	*Average*	*Maximum*
Consensus-based distributed PCA		*Presented*	*Maximum*	*Un present*	*Average*	*Maximum*
Lossy data compression		*Un present*	*Maximum*	*Present*	*Average*	*Average*
Collaborative signal processing	*k-means clustering algorithm*	*Present*	*Average*	*Un present*	*Low*	*Average*
Advanced surveillance systems		*Present*	*Minimum*	*Presented*	*Average*	*Minimum*
Role-free clustering [47]	*Q-learning clustering algorithm*	*Un present*	*Minimum*	*Un present*	*Minimum*	*Minimum*
Decentralized learning for data latency [48]	*Reinforcement Learning Algorithm*	*Present*	*Minimum*	*Un present*	*Average*	*Minimum*

5 Conclusion

The result of this paper showed that the evolutionary approach to optimizing the deep neural networks and presented the learning algorithm that is able to learn network dynamic function. This survey belongs to various distributed DNN's systems achieve the best performance. In discussion part we explained the neural networks achieve highest accuracy between machine learning models. Paper clearly explained about heterogeneous network and compression techniques with respect to learning algorithm.

References

[1] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, et al., "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," IEEE wireless communications, vol. 24, pp. 146-153, 2016.

[2] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," IEEE Wireless Communications, vol. 24, pp. 98-105, 2016.

[3] B. Zhang, L. Sun, H. Yuan, J. Lv, and Z. Ma, "An improved regularized extreme learning machine based on symbiotic organisms search," in 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), 2016, pp. 1645-1648.

[4] A. I. Orhean, F. Pop, and I. Raicu, "New scheduling approach using reinforcement learning for heterogeneous distributed systems," Journal of Parallel and Distributed Computing, vol. 117, pp. 292-302, 2018.

[5] E. Barbierato, M. Gribaudo, M. Iacono, and S. Marrone, "Performability modeling of exceptions-aware systems in multiformalism tools," in International Conference on Analytical and Stochastic Modeling Techniques and Applications, 2011, pp. 257-272.

[6] Y. Yu, M. Abadi, P. Barham, E. Brevdo, M. Burrows, A. Davis, et al., "Dynamic control flow in large-scale machine learning," in Proceedings of the Thirteenth EuroSys Conference, 2018, p. 18.

[7] J. Jiang, B. Cui, C. Zhang, and L. Yu, "Heterogeneity-aware distributed parameter servers," in Proceedings of the 2017 ACM International Conference on Management of Data, 2017, pp. 463-478.

[8] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

[9] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149, 2015.

[10] S. Wiedemann, H. Kirchhoffer, S. Matlage, P. Haase, A. Marban, T. Marinc, et al., "DeepCABAC: Context-adaptive binary arithmetic coding for deep neural network compression," arXiv preprint arXiv:1905.08318, 2019.
[11] J. Wu, H. Ren, Y. Kong, C. Yang, L. Senhadji, and H. Shu, "Compressing complex convolutional neural network based on an improved deep compression algorithm," arXiv preprint arXiv:1903.02358, 2019.
[12] S. Jin, S. Di, X. Liang, J. Tian, D. Tao, and F. Cappello, "DeepSZ: A Novel Framework to Compress Deep Neural Networks by Using Error-Bounded Lossy Compression," arXiv preprint arXiv:1901.09124, 2019.
[13] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," arXiv preprint arXiv:1802.03268, 2018.
[14] F. Petroski Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," arXiv preprint arXiv:1712.06567, 2017.
[15] P. Pietzuch, P. Watcharapichat, V. Lopez Morales, and R. Castro Fernandez, "Ako: Decentralised Deep Learning with Partial Gradient Exchange."
[16] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 5058-5066.
[17] F. Tung and G. Mori, "Deep Neural Network Compression by In-Parallel Pruning-Quantization," IEEE transactions on pattern analysis and machine intelligence, 2018.
[18] D. Lückehe, S. Veith, and G. von Voigt, "Evolutionary Structure Minimization of Deep Neural Networks for Motion Sensor Data," in Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), 2018, pp. 243-257.
[19] R. Ma and L. Niu, "A Survey of Sparse-Learning Methods for Deep Neural Networks," in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018, pp. 647-650.
[20] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in International Conference on Machine Learning, 2015, pp. 2285-2294.
[21] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," 2011.
[22] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in International Conference on Machine Learning, 2015, pp. 1737-1746.
[23] F. Manessi, A. Rozza, S. Bianco, P. Napoletano, and R. Schettini, "Automated pruning for deep neural network compression," in 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 657-664.
[24] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, "ThiNet: pruning CNN filters for a thinner net," IEEE transactions on pattern analysis and machine intelligence, 2018.
[25] A. Aguinaldo, P.-Y. Chiang, A. Gain, A. Patil, K. Pearson, and S. Feizi, "Compressing GANs using Knowledge Distillation," arXiv preprint arXiv:1902.00159, 2019.
[26] P. Watcharapichat, V. L. Morales, R. C. Fernandez, and P. Pietzuch, "Ako: Decentralised deep learning with partial gradient exchange," in Proceedings of the Seventh ACM Symposium on Cloud Computing, 2016, pp. 84-97.
[27] S. Son, S. Nah, and K. Mu Lee, "Clustering convolutional kernels to compress deep neural networks," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 216-232.
[28] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in Advances in Neural Information Processing Systems, 2016, pp. 901-909.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.
[31] A. Graves, J. Menick, and A. v. d. Oord, "Associative compression networks," arXiv preprint arXiv:1804.02476, 2018.
[32] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.
[33] X. Geng, J. Fu, B. Zhao, J. Lin, M. M. S. Aly, C. Pal, et al., "Dataflow-based Joint Quantization of Weights and Activations for Deep Neural Networks," arXiv preprint arXiv:1901.02064, 2019.
[34] V. S. R. ANNAPUREDDY, D. H. F. DIJKMAN, and D. J. Julian, "Model compression and fine-tuning," ed: Google Patents, 2019.
[35] C. Louizos, K. Ullrich, and M. Welling, "Bayesian compression for deep learning," in Advances in Neural Information Processing Systems, 2017, pp. 3288-3298.
[36] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in Advances in neural information processing systems, 2015, pp. 3123-3131.
[37] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," arXiv preprint arXiv:1602.02830, 2016.
[38] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4829-4837.
[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492-1500.
[40] A. Jamil, M. Majid, and S. M. Anwar, "An Optimal Codebook for Content-Based Image Retrieval in

JPEG Compressed Domain," Arabian Journal for Science and Engineering, pp. 1-13, 2019.
[41] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 7559-7566.
[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning. nature 521 (7553): 436," Google Scholar, 2015.
[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, p. 436, 2015.
[44] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, et al., "Large scale distributed deep networks," in Advances in neural information processing systems, 2012, pp. 1223-1231.
[45] M. Gastegger, J. Behler, and P. Marquetand, "Machine learning molecular dynamics for the simulation of infrared spectra," Chemical science, vol. 8, pp. 6924-6935, 2017.
[46] A. Asuncion and D. Newman, "UCI machine learning repository," ed, 2007.
[47] K. Shahina and V. Vaidehi, "Clustering and Data Aggregation in Wireless Sensor Networks Using Machine Learning Algorithms," in 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), 2018, pp. 109-115.
[48] G. Russo Russo, M. Nardelli, V. Cardellini, and F. Lo Presti, "Multi-Level Elasticity for Wide-Area Data Streaming Systems: A Reinforcement Learning Approach," Algorithms, vol. 11, p. 134, 2018.