Embedded Machine Learning

Subjects: Computer Science, Artificial Intelligence | Engineering, Electrical & Electronic | Computer Science, Hardware & Architecture
Contributor: Ajani Taiwo

Embedded machine learning (EML) can be applied in the areas of accurate computer vision schemes, reliable speech recognition, innovative healthcare, robotics, and more. However, there exists a critical drawback in the efficient implementation of ML algorithms targeting embedded applications. Machine learning algorithms are generally computationally and memory intensive, making them unsuitable for resource-constrained environments such as embedded and mobile devices. In order to efficiently implement these compute and memory-intensive algorithms within the embedded and mobile computing space, innovative optimization techniques are required at the algorithm and hardware levels.

Keywords: embedded computing systems ; computer architecture ; mobile computing ; machine learning ; TinyML ; deep learning ; mobile devices ; optimization techniques

1. Introduction

Machine learning is a branch of artificial intelligence that describes techniques through which systems learn and make intelligent decisions from available data. Machine learning techniques can be classified under three major groups, which are supervised learning, unsupervised learning, and reinforcement learning as described in **Table 1**. In supervised learning, labeled data can be learned while in unsupervised learning, hidden patterns can be discovered from unlabeled data, and in reinforcement learning, a system may learn from its immediate environment through the trial and error method [1][2][3]. The process of learning is referred to as the *training phase* of the model and is often carried out using computer architectures with high computational resources such as multiple GPUs. After learning, the trained model is then used to make intelligent decisions on new data. This process is referred to as the *inference phase* of the implementation. The inference is often intended to be carried out within user devices with low computational resources such as IoT and mobile devices.

Table 1. Machine learning techniques.

Machine Learning Techniques			
Supervised Learning		**Unsupervised Learning**	**Reinforcement Learning**
Classification	**Regression**	**Clustering**	**Genetic Algorithms**
SVM	**SVR**	**HMM**	**Estimated Value Functions**
Naïve Bayes	**Linear Regression**	**GMM**	**Simulated Annealing**
***k*-NN**	**Decision Trees**	***k*-means**	
Logistic Regression	**ANN**	**DNN**	
Discriminant Analysis	**Ensemble Methods**		
DNN	**DNN**		

In recent times, machine learning techniques have been finding useful applications in various research areas and particularly in embedded computing systems. In this research, we surveyed recent works of literature concerning machine learning techniques implemented within resource-scarce environments such as mobile devices and other IoT devices between 2014 and 2020. We present the results of this survey in a tabular form given in **Table 2**. Our survey revealed that of all available machine learning techniques, SVMs, GMMs, DNNs, *k*-NNs, HMMs, decision trees, logistic regression, *k*-means, and naïve Bayes are common techniques adopted for embedded and mobile applications. Naïve Bayes and decision trees have low complexity in terms of computation and memory costs and thus do not require innovative optimizations as pointed out by Sayali and Channe [37]. Logistic regression algorithms are computationally cheaper than

naïve Bayes and decision trees, meaning they have even lower complexity [38]. HMMs, *k*-NNs, SVMs, GMMs, and DNNs are however computationally and memory intensive and hence, require novel optimization techniques to be carried out to be efficiently squeezed within resource-limited environments. We have thus limited our focus to these compute intensive ML models and discuss state-of-the-art optimization techniques through which these algorithms may be efficiently implemented within resource-constrained environments.

Table 2. Machine Learning Techniques in Resource-Constrained Environments.

Reference	ML Method	Embedded/Mobile Platform	Application	Year
[4]	SVM	ARMv7, IBM PPC440	Network Configuration	2015
[5]	DNN	FPGA Zedboard with 2 ARM Cortex Cores	Character Recognition	2015
[6]	DNN	Xilinx FPGA board	Image classification	2016
[7]	LSTM RNN	Zynq 7020 FPGA	Character Prediction	2016
[8]	CNN	VC707 Board with Xilinx FPGA chip	Image Classification	2015
[9]	GMM	Raspberry Pi	Integer processing	2014
[10]	*k*-NN, SVM	Mobile Device	Fingerprinting	2014
[11]	*k*-NN	Mobile Device	Fingerprinting	2014
[12]	*k*-NN, GMM	Mobile Device	Mobile Device Identification	2015
[13]	SVM	Xilinx Virtex 7 XC7VX980 FPGA	Histopathological image classification	2015
[14]	HMM	Nvidia Kepler	Speech Recognition	2015
[15]	Logistic Regression	Smart band	Stress Detection	2015
[16]	*k*-means	Smartphone	Indoor Localization	2015
[17]	Naïve Bayes	AVR ATmega-32	Home Automation	2015
[18]	*k*-NN	Smartphone	Image Recognition	2015
[19]	Decision Tree	Mobile Device	Health Monitoring	2015
[20]	GMM	FRDM-K64F equipped with ARM Cortex-M4F core	IoT sensor data analysis	2016
[21]	CNN	FPGA Xilinx Zynq ZC706 Board	Image Classification	2016
[22]	CNN	Mobile Device	Mobile Sensing	2016
[23]	SVM	Mobile Device	Fingerprinting	2016
[24]	*k*-NN, SVM	Mobile Device	Fingerprinting	2016
[25]	*k*-NN	Xilinx Virtex-6 FPGA	Image Classification	2016
[26]	HMM	Arduino UNO	Disease detection	2016
[27]	Logistic Regression	Wearable Sensor	Stress Detection	2016
[28]	Naïve Bayes	Smartphone	Health Monitoring	2016
[29]	Naïve Bayes	Mobile Devices	Emotion Recognition	2016
[30]	*k*-NN	Smartphone	Data Mining	2016
[31]	HMM	Smartphone Sensors	Activity Recognition	2017
[32]	DNN	Smartphone	Face detection, activity recognition	2017
[33]	CNN	Mobile Device	Image classification	2017
[34]	SVM	Mobile Device	Mobile Device Identification	2017
[35]	SVM	Jetson-TK1	Healthcare	2017

Reference	ML Method	Embedded/Mobile Platform	Application	Year
[36]	SVM, Logistic Regression	Arduino UNO	Stress Detection	2017
[37]	Naïve Bayes	Smartphone	Emotion Recognition	2017
[38]	*k*-means	Smartphones	Safe Driving	2017
[39]	HMM	Mobile Device	Health Monitoring	2017
[40]	*k*-NN	Arduino UNO	Image Classification	2017
[41]	SVM	Wearable Device (nRF51822 SoC+BLE)	Battery Life Management	2018
[42]	SVM	Zybo Board with Z-7010 FPSoC	Face Detection	2018
[43]	CNN	Raspberry Pi + Movidus Neural Compute Stick	Vehicular Edge Computing	2018
[44]	CNN	Jetson TX2	Image Classification	2018
[45]	HMM	Smartphone	Healthcare	2018
[46]	*k*-NN	Smartphone	Health Monitoring	2019
[47]	Decision Trees	Arduino UNO	Wound Monitoring	2019
[48]	RNN	ATmega640	Smart Sensors	2019
[49]	SVM, Logistic Regression, *k*-means, CNN	Raspberry Pi	Federated Learning	2019
[50]	DNN	Raspberry Pi	Transient Reduction	2020
[51]	MLP	Embedded SoC (ESP4ML)	Classification	2020
[52]	HMM	Smartphone	Indoor Localization	2020
[53]	*k*-NN	Smartphone	Energy Management	2020
[54]	ANN, Decision Trees	Raspberry Pi	Classification and Regression	2021

2. Challenges and Optimization Opportunities in Embedded Machine Learning

Embedded computing systems are generally limited in terms of available computational power and memory requirements. Furthermore, they are required to consume very low power and to meet real-time constraints. Thus, for these computationally intensive machine learning models to be executed efficiently in the embedded systems space, appropriate optimizations are required both at the hardware architecture and algorithm levels [55][56]. In this section, we survey optimization methods to tackle bottlenecks in terms of power consumption, memory footprint, latency concerns, and throughput and accuracy loss.

2.1. Power Consumption

The total energy consumed by an embedded computing application is the sum of the energy required to fetch data from the available memory storage and the energy required to perform the necessary computation in the processor. **Table 3** shows the energy required to perform different operations in an ASIC. It can be observed from **Table 3** that the amount of energy required to fetch data from the SRAM is much less, than when fetching data from the off-chip DRAM and very minimal if the computation is done at the register files. From this insight, we can conclude that computation should be done as close to the processor as possible to save energy. However, this is a bottleneck because the standard size of available on-chip memory in embedded architectures is very low compared to the size of deep learning models [57]. Algorithmic-based optimization techniques for model compression such as parameter pruning, sparsity, and quantization may be applied to address this challenge [58]. Also, hardware design-based optimizations such as Tiling and data reuse may be utilized [8]. The next section expatiates some of these optimization methods in further detail. Furthermore, most machine-learning models, especially deep learning models, require huge amounts of multiply and accumulate (MAC) operations for effective training and inference. **Figure 1** describes the power consumed by the MAC unit as a function of the bit precision adopted by the system. We may observe that the higher the number of bits, the higher the power

consumed. Thus, to reduce the power consumed during computation, reduced bit precision arithmetic and data quantization may be utilized [59].

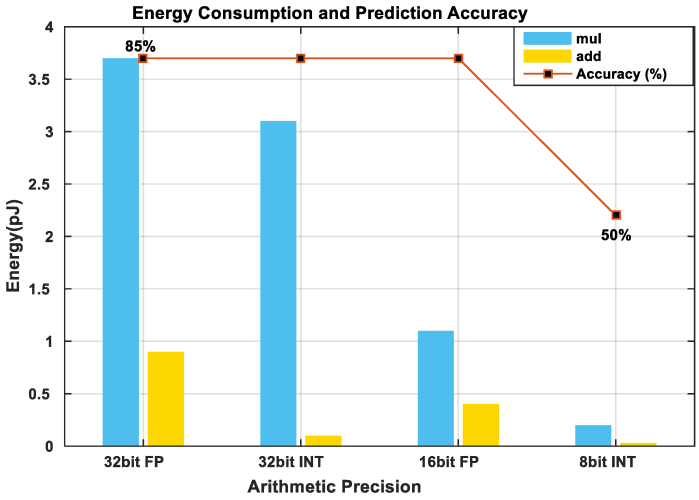

Figure 1. This graph describes the energy consumption and prediction accuracy of a DNN as a function of the Arithmetic Precision adopted for a single MAC unit in a 45 nm CMOS [57]. It may be deduced from the graph that lower number precisions consume less power than high precisions with no loss in prediction accuracy. However, we can observe that when precision is reduced below a particular threshold (16 bit fp), the accuracy of the model is greatly affected. Thus, quantization may be performed successfully to conserve energy but quantizing below 16-bit fp may require retraining and fine-tuning to restore the accuracy of the model.

Table 3. Energy Consumption in (pJ) of performing operations.

Operation	Energy (pJ)
8 bit int ADD	0.03
16 bit int ADD	0.05
32 bit int ADD	0.1
16 bit float ADD	0.4
32 bit float ADD	0.9
8 bit MULT	0.2
32 bit MULT	3.1
16 bit float MULT	1.1
32 bit float MULT	3.7
32 bit SRAM READ	5.0
32 bit DRAM READ	640

Source: Bill Dally, Cadence Embedded Neural Network Summit, 1 February 2017.

2.2. Memory Footprint

The available on-chip and off-chip memory in embedded systems are very limited compared to the size of ML parameters (synapses and activations) [60]. Thus, there is a bottleneck for storing model parameters and activations within this constrained memory. Network pruning (removing redundant parameters) [58] and data quantization [59] (reducing the number of bits used to represent model parameters) are the primary optimization techniques adopted to significantly compress the overall model size such that they can fit into the standard memory sizes of embedded computers.

2.3. Latency and Throughput Concerns

Embedded systems are required to meet real-time deadlines. Thus, latency and overall throughput can be a major concern as an inability to meet these tight constraints could sometimes result in devastating consequences. The parameters of deep learning models are very large and are often stored off-chip or in external SDCARDs, which

introduces latency concerns. Latency results from the time required to fetch model parameters from off-chip DRAM or external SDCARDs before appropriate computation can be performed on these parameters [58]. Thus, storing the parameters as close as possible to the computation unit using Tiling and data reuse, hardware-oriented direct memory access (DMA) optimization techniques would reduce the latency and thus, inform high computation speed [61]. In addition, because ML models require a high level of parallelism for efficient performance, throughput is a major issue. Memory throughput can be optimized by introducing pipelining [5].

2.4. Prediction Accuracy

Although deep learning models are tolerant of low bit precision [62], reducing the bit precision below a certain threshold could significantly affect the prediction accuracy of these models and introduce no little errors, which could be costly for the embedded application. To address the errors which model compression techniques such as reduced precision or quantization introduce, the compressed model can be retrained or fine-tuned to improve precision accuracy [57][58][63][64].

2.5. Some Hardware-Oriented and Algorithm-Based Optimization Techniques

Hardware acceleration units may be designed using custom FPGAs or ASICs to inform low latency and high throughput. These designs are such that they may optimize the data access from external memory and/or introduce an efficient pipeline structure using buffers to increase the throughput of the architecture. In sum, some hardware-based optimization techniques are presented in this section to guide computer architects in designing and developing highly efficient acceleration units to inform high performance

References

1. Frank, M.; Drikakis, D.; Charissis, V. Machine-learning methods for computational science and engineering. Computati on 2020, 8, 15.
2. Xiong, Z.; Zhang, Y.; Niyato, D.; Deng, R.; Wang, P.; Wang, L.C. Deep reinforcement learning for mobile 5G and beyon d: Fundamentals, applications, and challenges. IEEE Veh. Technol. Mag. 2019, 14, 44–52.
3. Carbonell, J.G. Machine learning research. ACM SIGART Bull. 1981, 18, 29.
4. Jadhav, S.D.; Channe, H.P. Comparative STUDY of K-NN, naive bayes and decision tree classification techniques. Int. J. Sci. Res. 2016, 5, 1842–1845.
5. Chapter 4 Logistic Regression as a Classifier. Available online: (accessed on 29 December 2020).
6. Haigh, K.Z.; Mackay, A.M.; Cook, M.R.; Lin, L.G. Machine Learning for Embedded Systems: A Case Study; BBN Techn ologies: Cambridge, MA, USA, 2015; Volume 8571, pp. 1–12.
7. Yu, Q.; Wang, C.; Ma, X.; Li, X.; Zhou, X. A deep learning prediction process accelerator based FPGA. In Proceedings of the 2015 IEEE/ACM 15th International Symposium Cluster Cloud, Grid Computer CCGrid 2015, Shenzhen, China, 4 –7 May 2015; pp. 1159–1162.
8. Wang, C.; Gong, L.; Yu, Q.; Li, X.; Xie, Y.; Zhou, X. DLAU: A scalable deep learning accelerator unit on FPGA. IEEE Tr ans. Comput. Des. Integr. Circuits Syst. 2016, 36, 513–517.
9. Chang, A.X.M.; Martini, B.; Culurciello, E. Recurrent Neural Networks Hardware Implementationon FPGA. Available onl ine: (accessed on 15 January 2021).
10. Zhang, C.; Li, P.; Sun, G.; Guan, Y.; Xiao, B.; Cong, J. Optimizing FPGA-based accelerator design for deep convolution al neural networks. In Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arr ays, Monterey, CA, USA, 22–24 February 2015; pp. 161–170.
11. Salvadori, C.; Petracca, M.; del Rincon, J.M.; Velastin, S.A.; Makris, D. An optimisation of Gaussian mixture models for integer processing units. J. Real Time Image Process. 2017, 13, 273–289.
12. Das, A.; Borisov, N.; Caesar, M. Do you hear what i hear? Fingerprinting smart devices through embedded acoustic co mponents. In Proceedings of the ACM Conference on Computer, Communication and Security, Scottsdale, AZ, USA, 3 –7 November 2014; pp. 441–452.
13. Bojinov, H.; Michalevsky, Y.; Nakibly, G.; Boneh, D. Mobile Device Identification via Sensor Fingerprinting. Available onli ne: (accessed on 12 January 2021).
14. Huynh, M.; Nguyen, P.; Gruteser, M.; Vu, T. Mobile device identification by leveraging built-in capacitive signature. In Pr oceedings of the ACM Conference on Compututer, Communication and Security, Denver, CO, USA, 12–16 October 20 15; pp. 1635–1637.

15. Dhar, S.; Sreeraj, K.P. FPGA implementation of feature extraction based on histopathalogical image and subsequent cl assification by support vector machine. IJISET Int. J. Innov. Sci. Eng. Technol. 2015, 2, 744–749.
16. Yu, L.; Ukidave, Y.; Kaeli, D. GPU-accelerated HMM for speech recognition. In Proceedings of the International Confer ence Parallel Processing Work, Minneapolis, MN, USA, 9–12 September 2014; pp. 395–402.
17. Zubair, M.; Yoon, C.; Kim, H.; Kim, J.; Kim, J. Smart wearable band for stress detection. In Proceedings of the 2015 5th International Conference IT Converg. Secur. ICITCS, Kuala Lumpur, Malaysia, 24–27 August 2015; pp. 1–4.
18. Razavi, A.; Valkama, M.; Lohan, E.S. K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization. In Proceedings of the 2015 IEEE Globecom Work. GC Wkshps 2015, San Diego, CA, USA, 6–10 Decemb er 2015.
19. Bhide, V.H.; Wagh, S. I-learning IoT: An intelligent self learning system for home automation using IoT. In Proceedings of the 2015 International Conference Communication Signalling Process. ICCSP 2015, Melmaruvathur, India, 2–4 April 2015; pp. 1763–1767.
20. Munisami, T.; Ramsurn, M.; Kishnah, S.; Pudaruth, S. Plant Leaf recognition using shape features and colour histogra m with K-nearest neighbour classifiers. Proc. Comput. Sci. 2015, 58, 740–747.
21. Sowjanya, K.; Singhal, A.; Choudhary, C. MobDBTest: A machine learning based system for predicting diabetes risk usi ng mobile devices. In Proceedings of the Souvenir 2015 IEEE Int. Adv. Comput. Conference IACC 2015, Banglore, Indi a, 12–13 June 2015; pp. 397–402.
22. Lee, J.; Stanley, M.; Spanias, A.; Tepedelenlioglu, C. Integrating machine learning in embedded sensor systems for Inte rnet-of-Things applications. In Proceedings of the 2016 IEEE International Symposium on Signal Processing and Infor mation Technology (ISSPIT), Limassol, Cyprus, 12–14 December 2016; pp. 290–294.
23. Qiu, J.; Wang, J.; Yao, S.; Guo, K.; Li, B.; Zhou, E.; Yu, J.; Tang, T.; Xu, N.; Song, S.; et al. Going deeper with embedde d FPGA platform for convolutional neural network. In Proceedings of the FPGA 2016ACM/SIGDA International Symposi um Field-Programmable Gate Arrays, Monterey, CA, USA, 21–23 February 2016; pp. 26–35.
24. Huynh, L.N.; Balan, R.K.; Lee, Y. DeepSense: A GPU-based deep convolutional neural network framework on commodi ty mobile devices. In Proceedings of the Workshop on Wearable Systems and Application Co-Located with MobiSys 20 16, Singapore, 30 June 2016; pp. 25–30.
25. Tuama, A.; Comby, F.; Chaumont, M. Camera model identification based machine learning approach with high order st atistics features. In Proceedings of the 24th European Signal Processing Conference (EUSIPCO), Budapest, Hungary, 29 August–2 September 2016; pp. 1183–1187.
26. Kurtz, A.; Gascon, H.; Becker, T.; Rieck, K.; Freiling, F. Fingerprinting Mobile Devices Using Personalized Configuration s. Proc. Priv. Enhanc. Technol. 2016, 1, 4–19.
27. Mohsin, M.A.; Perera, D.G. An FPGA-based hardware accelerator for k-nearest neighbor classification for machine lear ning on mobile devices. In Proceedings of the ACM International Conference Proceeding Series, HEART 2018, Toront o, ON, Canada, 20–22 June 2018; pp. 6–12.
28. Patil, S.S.; Thorat, S.A. Early detection of grapes diseases using machine learning and IoT. In Proceedings of the 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), Mysuru, India, 12–13 Au gust 2016.
29. Ollander, S.; Godin, C.; Campagne, A.; Charbonnier, S. A comparison of wearable and stationary sensors for stress det ection. In Proceedings of the IEEE International Conference System Man, and Cybernetic SMC 2016, Budapest, Hung ary, 9–12 October 2016; pp. 4362–4366.
30. Moreira, M.W.L.; Rodrigues, J.J.P.C.; Oliveira, A.M.B.; Saleem, K. Smart mobile system for pregnancy care using body sensors. In Proceedings of the International Conference Sel. Top. Mob. Wirel. Networking, MoWNeT 2016, Cairo Egyp t, 11–13 April 2016; pp. 1–4.
31. Shapsough, S.; Hesham, A.; Elkhorazaty, Y.; Zualkernan, I.A.; Aloul, F. Emotion recognition using mobile phones. In Pr oceedings of the 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthc om), Munich, Germany, 14–16 September 2016; pp. 276–281.
32. Hakim, A.; Huq, M.S.; Shanta, S.; Ibrahim, B.S.K.K. Smartphone based data mining for fall detection: Analysis and desi gn. Proc. Comput. Sci. 2016, 105, 46–51.
33. Ronao, C.A.; Cho, S.B. Recognizing human activities from smartphone sensors using hierarchical continuous hidden M arkov models. Int. J. Distrib. Sens. Netw. 2017, 13, 1–16.
34. Kodali, S.; Hansen, P.; Mulholland, N.; Whatmough, P.; Brooks, D.; Wei, G.Y. Applications of deep neural networks for u ltra low power IoT. In Proceedings of the 35th IEEE International Conference on Computer Design ICCD 2017, Boston, MA, USA, 5–8 November 2017; pp. 589–592.

35. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An extremely efficient convolution neural network for mobile devices. I n Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
36. Baldini, G.; Dimc, F.; Kamnik, R.; Steri, G.; Giuliani, R.; Gentile, C. Identification of mobile phones using the built-in mag netometers stimulated by motion patterns. Sensors 2017, 17, 783.
37. Azimi, I.; Anzanpour, A.; Rahmani, A.M.; Pahikkala, T.; Levorato, M.; Liljeberg, P.; Dutt, N. HiCH: Hierarchical fog-assist ed computing architecture for healthcare IoT. ACM Trans. Embed. Comput. Syst. 2017, 16, 1–20.
38. Pandey, P.S. Machine Learning and IoT for prediction and detection of stress. In Proceedings of the 17th International Conference on Computational Science and Its Applications ICCSA 2017, Trieste, Italy, 3–6 July 2017.
39. Sneha, H.R.; Rafi, M.; Kumar, M.V.M.; Thomas, L.; Annappa, B. Smartphone based emotion recognition and classificati on. In Proceedings of the 2nd IEEE International Conference on Electrical, Computer and Communication Technology I CECCT 2017, Coimbatore, India, 22–24 February 2017.
40. Al Mamun, M.A.; Puspo, J.A.; Das, A.K. An intelligent smartphone based approach using IoT for ensuring safe driving. I n Proceedings of the 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Pale mbang, Indonesia, 22–23 August 2017; pp. 217–223.
41. Neyja, M.; Mumtaz, S.; Huq, K.M.S.; Busari, S.A.; Rodriguez, J.; Zhou, Z. An IoT-based e-health monitoring system usi ng ECG signal. In Proceedings of the IEEE Global Communications Conference GLOBECOM 2017, Singapore, 4–8 D ecember 2017; pp. 1–6.
42. Gupta, C.; Suggala, A.S.; Goyal, A.; Simhadri, H.V.; Paranjape, B.; Kumar, A.; Goyal, S.; Udupa, R.; Varma, M.; Jain, P. ProtoNN: Compressed and accurate kNN for resource-scarce devices. In Proceedings of the 34th International Confere nce on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1331–1340.
43. Fafoutis, X.; Marchegiani, L.; Elsts, A.; Pope, J.; Piechocki, R.; Craddock, I. Extending the battery lifetime of wearable s ensors with embedded machine learning. In Proceedings of the IEEE World Forum on Internet Things, WF-IoT 2018, Si ngapore, 5–8 February 2018; pp. 269–274.
44. Damljanovic, A.; Lanza-Gutierrez, J.M. An embedded cascade SVM approach for face detection in the IoT edge layer. I n Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, D C, USA, 21–23 October 2018; pp. 2809–2814.
45. Hochstetler, J.; Padidela, R.; Chen, Q.; Yang, Q.; Fu, S. Embedded deep learning for vehicular edge computing. In Pro ceedings of the 3rd ACM/IEEE Symposium on Edge Computing SEC 2018, Seattle, WA, USA, 25–27 October 2018; p p. 341–343.
46. Taylor, B.; Marco, V.S.; Wolff, W.; Elkhatib, Y.; Wang, Z. Adaptive deep learning model selection on embedded systems. ACM SIGPLAN Not. 2018, 53, 31–43.
47. Strielkina, A.; Kharchenko, V.; Uzun, D. A markov model of healthcare internet of things system considering failures of c omponents. CEUR Workshop Proc. 2018, 2104, 530–543.
48. Vhaduri, S.; van Kessel, T.; Ko, B.; Wood, D.; Wang, S.; Brunschwiler, T. Nocturnal cough and snore detection in noisy environments using smartphone-microphones. In Proceedings of the IEEE International Conference on Healthcare Info rmatics, ICHI 2019, Xi'an, China, 10–13 June 2019.
49. Sattar, H.; Bajwa, I.S.; Amin, R.U.; Sarwar, N.; Jamil, N.; Malik, M.A.; Mahmood, A.; Shafi, U. An IoT-based intelligent w ound monitoring system. IEEE Access 2019, 7, 144500–144515.
50. Mengistu, D.; Frisk, F. Edge machine learning for energy efficiency of resource constrained IoT devices. In Proceedings of the Fifth International Conference on Smart Portable, Wearable, Implantable and Disabilityoriented Devices and Syst ems, SPWID 2019, Nice, France, 28 July–1 August 2019; pp. 9–14.
51. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive Federated Learning in Resource C onstrained Edge Computing Systems. IEEE J. Sel. Areas Commun. 2019, 37, 1205–1221.
52. Suresh, P.; Fernandez, S.G.; Vidyasagar, S.; Kalyanasundaram, V.; Vijayakumar, K.; Archana, V.; Chatterjee, S. Reduct ion of transients in switches using embedded machine learning. Int. J. Power Electron. Drive Syst. 2020, 11, 235–241.
53. Giri, D.; Chiu, K.L.; di Guglielmo, G.; Mantovani, P.; Carloni, L.P. ESP4ML: Platform-based design of systems-on-chip f or embedded machine learning. In Proceedings of the 2020 Design, Automation and Test in European Conference Exhi bition DATE 2020, Grenoble, France, 9–13 March 2020; pp. 1049–1054.
54. Tiku, S.; Pasricha, S.; Notaros, B.; Han, Q. A hidden markov model based smartphone heterogeneity resilient portable i ndoor localization framework. J. Syst. Archit. 2020, 108, 101806.

55. Mazlan, N.; Ramli, N.A.; Awalin, L.; Ismail, M.; Kassim, A.; Menon, A. A smart building energy management using intern et of things (IoT) and machine learning. Test. Eng. Manag. 2020, 83, 8083–8090.
56. Cornetta, G.; Touhafi, A. Design and evaluation of a new machine learning framework for iot and embedded devices. El ectronics 2021, 10, 600.
57. Capra, M.; Bussolino, B.; Marchisio, A.; Shafique, M.; Masera, G.; Martina, M. An Updated survey of efficient hardware architectures for accelerating deep convolutional neural networks. Future Internet 2020, 12, 113.
58. Sun, S.; Cao, Z.; Zhu, H.; Zhao, J. A survey of optimization methods from a machine learning perspective. IEEE Trans. Cybern. 2020, 50, 3668–3681.
59. Han, S.; Pool, J.; Tran, J.; Dally, W.J. Learning both weights and connections for efficient neural networks. In Proceedin gs of the NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems; ACM: New York, NY, USA, 2015; Volume 1, pp. 1135–1143.
60. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puert o Rico, 2–4 May 2016; pp. 1–14. Available online: (accessed on 17 January 2021).
61. Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Quantized neural networks: Training neural networks w ith low precision weights and activations. J. Mach. Learn. Res. 2018, 18, 1–30.
62. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level Accuracy With 50× Fewer Parameters and <0.5 mb Model Size. Available online: (accessed on 15 February 2021).
63. Tanaka, K.; Arikawa, Y.; Ito, T.; Morita, K.; Nemoto, N.; Miura, F.; Terada, K.; Teramoto, J.; Sakamoto, T. Communication -efficient distributed deep learning with GPU-FPGA heterogeneous computing. In Proceedings of the 2020 IEEE Symp osium on High-Performance Interconnects (HOTI), Piscataway, NJ, USA, 19–21 August 2020; pp. 43–46.
64. Lane, N.; Bhattacharya, S.; Georgiev, P.; Forlivesi, C. Squeezing deep learning into mobile and embedded devices. IEE E Pervasive Comput. 2017, 16, 82–88.
65. Gysel, P. Ristretto: Hardware-Oriented Approximation of Convolutional Neural Networks. Available online: (accessed on 20 February 2021).
66. Moons, B.; Goetschalckx, K.; van Berckelaer, N.; Verhelst, M. Minimum energy quantized neural networks. In Proceedi ngs of the 2017 51st Asilomar Conference on Signals, Systems, and Computers ACSSC 2017, Pacific Grove, CA, US A, 29 October–1 November 2017; pp. 1921–1925.