Federated learning on edge: age&gender recognition on FPGA

Tania Di Mascio[1], Paolo Giammatteo[1], Luigi Laura[2], Valerio Rughetti[2], Giacomo Valente[1]
[1]UnivAQ Università dell'Aquila, [2]Università Telematica Internazionale UNINETTUNO
Corresponding author: valerio.rughetti@uninettunouniversity.net

Abstract

There are a set of key reasons why industry executives are transitioning from a traditional cloud-based model to edge computing platforms. The major technological factors are low latency and high bandwidth but greater security reasons play a key role in the developing of the edge computing. Nowadays this scenario, with AI purpose, is tipically used exclusively to make inference on data read. Using Federated Learning , a machine learning scheme in which a shared prediction model can be collaboratively learned by a number of distributed nodes, it can provide better data privacy because training data are not transmitted to a central server. Federated learning is well suited for edge computing applications and can leverage the computation power of edge servers and the data collected on widely dispersed edge devices. Our work aims to generate a CNN model capable of recognizing the age and gender of people whose face is framed via cameras installed on the edge device. The model will be trained directly on the edge devices themselves, using Federated learning. The tested edge devices will be Nvidia Jetson Nano (GPU acceleration) and FPGAs (DPU acceleration) equipped with appropriate cameras.

1 Introduction

The Adoption of Internet of Things (IoT) paradigm is nowadays rapidly expanding, causing an influx of data needing to be processed in centralised cloud computing and storage solutions. However, due to latency, bandwidth and security concerns, the viability of using the cloud is unclear and organisations are looking instead to edge computing solutions. Edge computing provides compute closer to the IoT device, for data collection and analytics [Zhou *et al.*, 2019]. In doing so, data is more secure, network latency is reduced as the round trip to the data centre and back is shorter and remain on site [Wang *et al.*, 2020]. Edge computing can optimise IoT applications, in particular ones that require real-time actions. So we are attending a speed up in the interest of applications on the edge [Giammatteo *et al.*, 2019] [Ndikumana *et al.*, 2021].

Artificial Intelligence (AI) applications are the most prominent in this context, in particular machine learning applications on edge computing [Murshed *et al.*, 2019]. Deep learning, which is a sub-branch of machine learning, plays a key role in these applications and accounts for almost all machine learning applications on the edge [Wang *et al.*, 2020] [Voghoei *et al.*, 2019].

The usage of deep learning methodologies and algorithms, such as Convolutional Neural Networks (CNN), is nowadays widespread and several examples are considered in edge regime [Xu *et al.*, 2020] [Nagnath *et al.*, 2020]. The workflow involves training of the algorithm on the cloud, or generally on computers with computational resources capable to face the training of a CNN, and inference of a specific occurrence on the edge [Xu *et al.*, 2017]. A particular case of deep learning with CNN is represented by the Age&Gender recognition, starting from a simple face image of a person. Many works in the scientific literature focused on this issue [Di Mascio *et al.*, 2021], consisting in training a CNN algorithm on a powerful accelerating device, such as the a GPU, and then performing the inference on edge device, such as GPU or FPGA [Xu *et al.*, 2017] [Greco *et al.*, 2020] [Chen *et al.*, 2016].

Recent developments and applications have led the interest of the scientific literature in moving training from the cloud to the edge itself. This technique is known as Federated Learning [Aledhari *et al.*, 2020], which is a machine learning technique that trains an algorithm across multiple decentralised edge devices [Wang *et al.*, 2019] or servers holding local data samples, without exchanging data themselves [Xia *et al.*, 2021]. This approach is in contrast to traditional centralised machine learning techniques in which all local data sets are uploaded to a server, which is what happens in the cloud.

Our interest is focused in considering federated learning approach on the specific case of Age&Gender recognition application, moving the training process on the edge. The motivations for moving the training from cloud to the edge are to be sought in: an increase in data privacy, as data would not leave the place where they are produced; a decrease in latency times inside the edge devices network; a reduction in energy consumption and therefore in network maintenance costs; and finally, an increase in system reliability even when the network connecting the edge devices malfunctions. The issues that such a system can produce are certainly: method

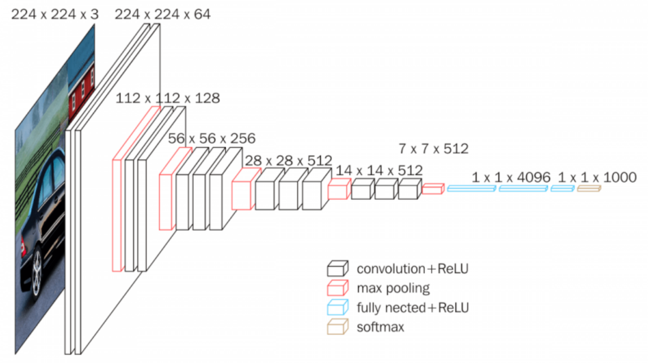

Figura 1: VGG16 architecture

of updating the weights of a CNN, ethic issues,

The contribution we intend to point out in this paper is to study the behaviour of a federated learning system for Age&Gender recognition. All of this by exploiting the use of an edge device, or platform in this case, used for both training and inference. The aim is to evaluate the performance in terms of time, in view of the application of the Age&Gender prediction starting from the image of a person.

2 Age and Gender Recognition: a VGG16 CNN

The topic of this paper is the creation of an automatic system capable of recognizing the gender and age of people who approach the information totems that may be inside shopping centers. Once the age and gender of the people framed are recognized, the totem will immediately present a graphic interface and customized content on the inferred data. The context of use therefore fits perfectly within an edge computing scenario, in which the edge devices (totems) are distributed within a limited environment (shopping center) where they can be suitably arranged and communicate with an edge gateway. The use of Federated Learning therefore appears perfectly usable in this context to allow continuous training on each individual device and the presence of the edge gateway will allow you to average the weights of the models generated by each device. The reasons for choosing this framework will be discussed in the next two subsections. In order to obtain comparable results with the classic cloud-based approach, we decided to create a convolutional neural network of type VGG16 customized [Xu *et al.*, 2020] [Di Mascio *et al.*, 2021]. The VGG16 architecture is depicted in figure 1. The output of the network is passed in another 4 dense layers where sigmoid and ReLu activation functions are applied to differentiate respectively gender and age results. In figure 2 is shown the mean percentage error per age and in figure 3 the accuracy and the dataset composition of gender classification by race. The dataset used for the training is the Morph2 academyc version ((link)). This network was trained on a single node using an Nvidia Titan XP GPU, the model generated with the related metrics will be used as a basis for comparison of the results obtained on the edge devices that will be trained in the next steps of the project.

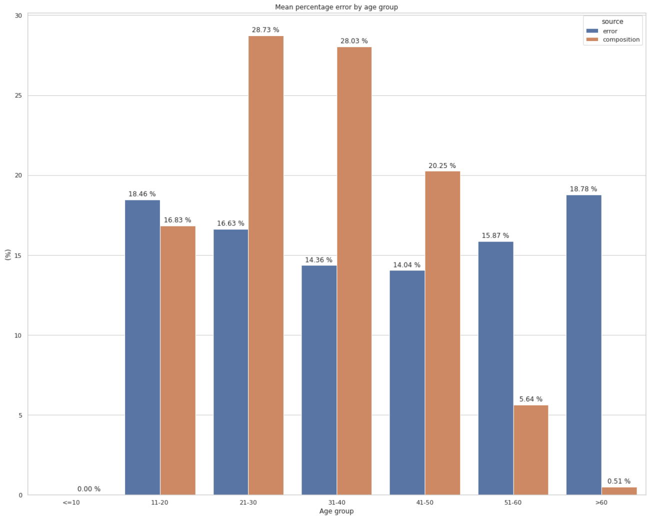

Figura 2: Accuracy and the dataset composition of gender classification by race

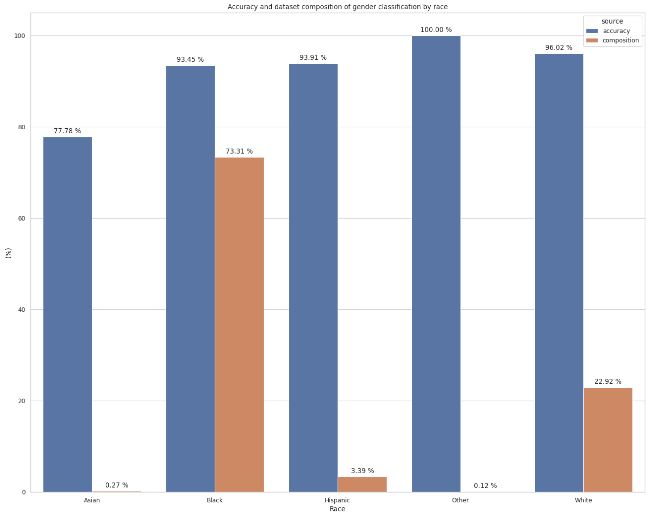

Figura 3: Mean Percentage error by age group

2.1 Edge Computing

As written in the previous paragraph there are a set of key reasons to render attractive edge computing platforms respect to a traditional cloud-based model. Technical reasons (low latency and high bandwidth), security (privacy or attacks like DDoS, almost harmless in an edge computing environment), reliability reasons (they do not have a single point of failure and the failure of a device not has effect on the network as a whole). Also, as well explained in [Xia *et al.*, 2021], in order to deliver these benefits to end-users, engineers have relied on a common set of key operating principles when building edge computing systems:

- Mobility: For applications like self-driving cars, the edge devices have to accommodate a constantly moving end-user without sacrificing latency or bandwidth. Some approaches solve this problem by positioning edge devices on the roadside.
- Proximity: In order to deliver low latency guarantees, the edge devices must be positioned as close as possible to the end users. This could mean performing computation directly at the edge device or investing in a local edge computing data center that is close to the end-user.
- Coverage: For edge computing to become ubiquitous, network coverage must be far-reaching. Thus, the exact distribution of nodes in an edge computing framework is imperative to achieving an optimal user experience. Of course, a dense distribution is preferred, but this must be balanced with cost constraints.

2.2 Federated Learning

Federated learning is a method for training neural networks across many devices. We focus on gradient-descent based federated learning algorithms, which have general applicability to a wide range of machine learning models. The learning process includes local update steps where each edge node performs gradient descent to adjust the (local) model parameter to minimize the loss function defined on its own dataset. The data used to train the neural network is stored locally across multiple nodes and are usually heterogeneous. It also includes global aggregation steps where model parameters obtained at different edge nodes are sent to an aggregator (edge gateway), which is a logical component that can run on the remote cloud, a network element, or an edge node. The aggregator aggregates these parameters (e.g., by taking a weighted average) and sends an updated parameter back to the edge nodes for the next round of iteration. The frequency of global aggregation is configurable; one can aggregate at an interval of one or multiple local updates. Each local update consumes computation resource of the edge node, and each global aggregation consumes communication resource of the network. The amount of consumed resources may vary over time, and there is a complex relationship among the frequency of global aggregation, the model training accuracy, and resource consumption [Wang *et al.*, 2019]. Because of the heterogeneity of federated learning, we do not require all nodes to participate in one synchronization. Only some of the nodes will be randomly selected to perform the computation.

3 System environments in case study

The project described above involves the implementation of a system that replicates the behavior of the totems. With this in mind, we will proceed with the implementation of two kinds of networks of edge devices. One consisting of boards with GPU acceleration and the other make by FPGA with DPU acceleration. On each of these devices a camera will be installed. On both systems we will proceed to the data inference phase on the model already generated and subsequently to the training of a new model and the subsequent inference phase. For the first system a good board could be The Nvidia Jetson Nano. This is a developer board with a Tegra SOC and a fully software compatibility. It would flawlessly deliver 472 GFLOPS of computing power combined with a quad-core 64-bit ARM CPU and 128-core integrated NVIDIA GPU. It also includes 4 Gigabytes of LPDDR4 memory with a low-powered 5 and 10W power nodes along with 5v of DC input. The board offers a Linux environment based on Ubuntu OS version 18.04 offers accelerated graphics with NVIDIA CUDA Toolkit 10.0, and libraries including cuDNN 7.3, and TensorRT 5. The 4.2 SDK provides the option of installing the popular Machine Learning frameworks. These frameworks include TensorFlow, PyTorch, Keras, Caffe, and MXNeta. Other frameworks for computer vision and robotics development include the OpenCV and ROS. The full compatibility of the NVIDIAs latest Jetson Nano AI platform makes it easier to deploy AI-based inference workloads to Jetson. The Nano brings real-time computer vision and inference across a variety of the complex Deep Neural Network (DNN) models. It enables the multi-sensor autonomous robots, IoT (Internet of Things) devices with intelligent edge analytics, and advanced Artificial Intelligence systems. The Jetson makes transfer learning easy for re-training networks locally onboard the Jetson Nano with ML frameworks. For the second scenario the Zynq UltraScale+ could represent an optimal choise. In fact, In the context of CNN execution, systems that rely on embedded heterogeneous SoCs built around ARM Cortex processors and FPGAs have been proposed, such as the Xilinx Zynq (link), Xilinx Ultrascale+ (link), Xilinx Versal (link), and Altera Arria10 (link). These architectures allow to integrate powerful and efficient accelerators on the reconfigurable logic, exploiting spatial computation typical of application specific integrated circuits, rather than thread-level parallelism typical of GPGPUs. Several dedicated accelerators have been proposed in the embedded domain both from companies, such as Xilinx (DPU spec) and Movidius [Venieris e Bouganis, 2017], and from the research community [Cavigelli e Benini, 2017][Chen *et al.*, 2016][Du *et al.*, 2015], outperforming programmable solutions in both performance and energy efficiency. This work makes usage of Xilinx DPU Deep Learning Unit (DPU) accelerator for CNN inference. The DPU programmable engine dedicated for convolutional neural network. The unit contains register configure module, data controller module, and convolution computing module. There is a specialized instruction set for DPU, which enables DPU to work efficiently for many convolutional neural networks.

4 Conclusion

The work is still in the initial state but with the convolutional neural network ready the performance of the inference on the edge will be verified in the short term and quickly. The training phase could instead present problems of a technical nature, however we are confident that, with the support of the scientific literature, it will be possible to implement the framework in its entirety and evaluate the differences in performance compared to development in the classic Cloud environment.

Riferimenti bibliografici

[Aledhari *et al.*, 2020] M. Aledhari, R. Razzak, R. M. Parizi, e F. Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[Cavigelli e Benini, 2017] Lukas Cavigelli e Luca Benini. Origami: A 803-gop/s/w convolutional network accelerator. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2461–2475, 2017.

[Chen *et al.*, 2016] A. T.-Y. Chen, M. Biglari-Abhari, K. I-K. Wang, A. Bouzerdoum, e F. H. C. Tivive. Hardware/software co-design for a gender recognition embedded system. In *Trends in Applied Knowledge-Based Systems and Data Science*, pages 541–552, Cham, 2016. Springer International Publishing.

[Di Mascio *et al.*, 2021] T. Di Mascio, P. Fantozzi, L. Laura, e V. Rughetti. Age and gender (face) recognition: A brief survey. In *Methodologies and Intelligent Systems for Technology Enhanced Learning, 11th International Conference, MIS4TEL 2021, Salamanca, Spain, 6-8 October, 2021*, volume 326 of *Lecture Notes in Networks and Systems*, pages 105–113. Springer, 2021.

[Du *et al.*, 2015] Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Tao Luo, Xiaobing Feng, Yunji Chen, e Olivier Temam. Shidiannao: Shifting vision processing closer to the sensor. *SIGARCH Comput. Archit. News*, 43(3S):92–104, jun 2015.

[Giammatteo *et al.*, 2019] P. Giammatteo, G. Valente, e A. D'Ortenzio. An intelligent informative totem application based on deep cnn in edge regime. In *ApplePies*, 2019.

[Greco *et al.*, 2020] A. Greco, A. Saggese, M. Vento, e V. Vigilante. A convolutional neural network for gender recognition optimizing the accuracy/speed tradeoff. *IEEE Access*, 8:130771–130781, 2020.

[Murshed *et al.*, 2019] M. G. Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, e Faraz Hussain. Machine learning at the network edge: A survey. *CoRR*, abs/1908.00080, 2019.

[Nagnath *et al.*, 2020] Y. S. Nagnath, C.-C. Kao, W.-C. Sun, C.-H. Lin, e C.-W. Hsieh. Realtime customer merchandise engagement detection and customer attribute estimation with edge device. In *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, pages 1–2, 2020.

[Ndikumana *et al.*, 2021] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, e C. S. Hong. Deep learning based caching for self-driving cars in multi-access edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2862–2877, 2021.

[Venieris e Bouganis, 2017] Stylianos I. Venieris e Christos-Savvas Bouganis. Latency-driven design for fpga-based convolutional neural networks. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–8, 2017.

[Voghoei *et al.*, 2019] S. Voghoei, N. T. Tonekaboni, J. G. Wallace, e H. R. Arabnia. Deep learning at the edge. 2019.

[Wang *et al.*, 2019] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, e K. Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[Wang *et al.*, 2020] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, e X. Chen. Convergence of edge computing and deeplearning: A comprehensive survey. *ieee communications surveys & tutorials*, 22(2):869–904, 2020.

[Xia *et al.*, 2021] Q. Xia, W. Ye, Z. Tao, J. Wu, e Q. Li. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1(1):100008, 2021.

[Xu *et al.*, 2017] J. Xu, B. Wang, J. Li, C. Hu, e J. Pan. Deep learning application based on embedded gpu. In *2017 First International Conference on Electronics Instrumentation Information Systems (EIIS)*, pages 1–4, 2017.

[Xu *et al.*, 2020] G. Xu, H. Yin, e J. Yang. Facial expression recognition based on convolutional neural networks and edge computing. In *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, pages 226–232, 2020.

[Zhou *et al.*, 2019] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, e J. Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.