**Thesis title:** Distributed device-embedded classification and prediction in near-to-real time

**Orange – Université Côte d'Azur**

**Subject:** In this thesis we study the problem of efficient classification and prediction of multivariate time-series captured by embedded devices by using joint data-model distributed algorithms for applications that preserve private data.

**Introduction**

Novel generations of devices, for example smartphones or CPEs (Customer-premises equipment), have the large capacity of on-chip processing. They are capable of capturing, storing and processing a large volume of data in well-defined setups. Today, models used for task-resolving are trained in the cloud by using all the available data received by devices. The classification or prediction model is further sent to each device (offline or Transfer learning [1]) and deployed locally for inference to obtain the final result. This process may be viewed as the centralized model training. We are in contrary interested to develop and explore the limits of fully distributed Machine Learning algorithms, where the learning process is performed on each device, and model parameters are exchanged locally with neighboring devices. Each time that neighbors’ model parameters reach the device, the full model is updated.

Our recent study [2] demonstrates that today’s generation of smartphones may be already used for accurate device-embedded classification (prediction) based on a simple three-layer Neural Network with several hours of training and a real-time testing time. For example, we trained MNIST [3] dataset on a smartphone with a Kirin 980 chip (phone ranked as 59th by the IA benchmarking criteria, see [4], [5]). The training that takes 2h (memory and CPU usage of approximately 24% and 32%, respectively), results in real-time classification accuracy of 94%. We note that during all the experiments the smartphone fully keeps its primary purpose, since the maximal number of cores used at the time is four (out of eight) and calculations exploit maximally 24% of the memory. The fact that current smartphones can perform challenging classification tasks accurately opens new perspectives for creating innovative applications that minimize energy consumption and latency (datasets are not transmitted to the cloud), and assure user data protection (GDPR compliance [6]).

The main contribution to the efficiency of new-age embedded devices comes from a jointly-tailored algorithm and system design. The algorithm design assures that properties of Machine Learning (ML) algorithms well exploit the system architecture. In the last decade multiple works on distributed ML algorithms have demonstrated a great potential. In general there exist three different distribution types: (i) data distribution [7-11] (common model trained by local device model computations and exchanges), (ii) model distribution [9-13] (portions of models trained by different devices) and (iii) combined data-model distribution [14]. We want to study the latter model, which is the most complex amongst the tree approaches.

**State-of-the-art**

Current state-of-the-art for distributed approaches can be classified into the three categories, as detailed below.
The most known data-distributed methods are Bosen [9] and Federated Learning [7, 8], the approaches that use iterative-convergent Machine Learning (ML) algorithms for training. They can be applied generically to any ML method if data samples are independent and identically distributed (i.i.d.). The Bosen platform provides a distributed version for a number of well-known ML algorithms (for example, Deep Learning, Sparse Coding, K-means clustering, Random forests or Multi-class Logistic Regression), while Federated Learning is designed to be efficient in setups with a large number of users and unreliable or slow connections. Final classification or prediction models represent a weight matrix that is stored across a large number of clients. Local weight matrix is calculated in the initial step and refined over the rounds, where updates are based on the exchange of parameters with local neighbors or a single master node.

Model-distributed approaches such as Strads platform [9] require ML-specialized systems that perform a partition of ML algorithms into a set of parallel tasks, in general scheduled by master node(s) and executed by a set of workers. Schedulers’ task is to separate the problem into a non-overlapping set of subproblems, divide a workload and synchronize the updates amongst the workers. This setup admits non-conflicting model updates that lead to convergence. Numerous algorithms can be deployed in this framework, such as Latent Dirichlet Allocation, Matrix Factorization, Support Vector Machine or Deep Learning algorithm based on Caffe, called Poseidon, to name a few.

Our goal is to develop model- and data-distributed algorithms for classification and prediction problems. In the literature, there exist only a few works. A hybrid distributed platform known as Angel [14] appropriately combines data partitioning, scheduling and parameter synchronization tasks and demonstrates accuracy improvement in comparison with a Petuum-based data or model distribution. There exist a number of calculus-parallelization methods, such as FlexFlow [15]. It is a hybrid data and model parallel (non-distributed) approach worth of exploring in a distributed setup, because it performs automated search of parallelization strategies that incorporates data, attribute, parameter and operator parallelization for DNN algorithms.

The above approaches mainly represent simulated environments with unlimited battery, memory and calculus capacity. They are based on data assumptions, such as data i.i.d. [9] or particular properties of the update matrix [16, 17] (for example, existence of factor vectors sufficient for accurate reconstruction of the update matrix).

In our setup, a large number of computationaly performant devices with certain constraints (battery or memory constraints) capture and process sensitive datasets. In particular, we will study the influence of the latency and quantization on convergence of the existing data-model distributed algorithms, as well as propose and develop new ones. We will focus on applications that preserve user privacy, for example for improving chatbot speech-to-text capabilities based on local usage, personal preference datasets captured by smartphones or network usage log datasets captured by Livebox devices (CPEs).
During the thesis, the candidate will:

- Gain the knowledge in domain of distributed Machine Learning with limited memory, computational and battery resources by literature search of state-of-the-art models
- Elaborate on pros and cons of Deep Learning and Machine Learning technics depending of data characterization and use cases
- Develop and deploy the high-dimensional classification and prediction algorithms that guarantee the convergence of the classification model under the device constraints in several setups,
- Write scientific papers in international journals and conferences and participate in the scientific exchanges (participation in conferences and other scientific exchanges)
- Vividly participate in the scientific life, as well as in laboratory meetings of both hosting laboratories

Contact: Frederic Precioso (frederic.precioso@univ-cotedazur.fr), Michel Riveill (michel.riveill@univ-cotedazur.fr), Thierry Nagellen (thierry.nagellen@orange.com) and Tamara Tosić (tamara.tosic@orange.com)

Required competences

You like to learn and search for the answers and are highly motivated to do the thesis in the emerging field of distributed algorithms for embedded devices. You have competences in Machine Learning, Optimization and Statistics (indispensable) as well as good programming skills and knowledge in Signal Processing (desirable). Interest in the field of embedded devices is a plus.

References


