Luigi Faticoso

10 May 2020

Report Trabajo

Introduction

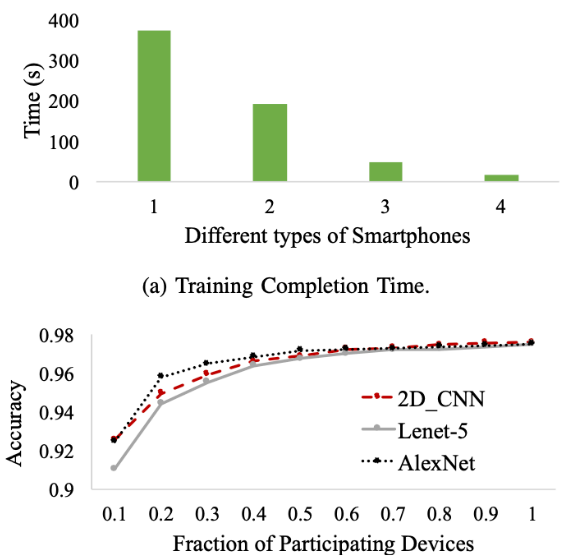
The objective of this work is to study and analyze the paper SmartPC: Hierarchical Pace Control in Real-Time Federated Learning System. The paper proposes an improvement to the current approach of decentralized learning called Federated Learning.

In the classical approach of learning, the data from devices is sent to a shared server that process the data and build a model to give back to the devices. Subsequently, Federated Learning proposes to develop learning of AI models through the collaboration of the devices by doing training on-device that results in preserving privacy as the data is processed internally and not send to a shared server.

The paper proposes a new framework for Federated Learning that balances training time and model accuracy in an energy-efficient manner.

Observation

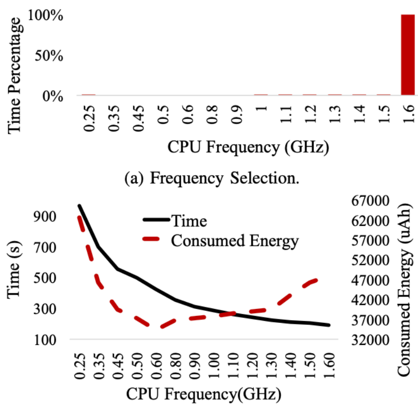
The starting point of the work is based on the fact that in FL the server waits for all the devices to send their model to the server and every device works independently from the others and so they use the maximum of their resources to train the model by leading to a massive reduction of energy. The observation they made were based on different devices with different hardware specifications. They found out that some devices can be 12 times faster than others with the same size of data set. The central server has to wait for all participants to send their updates before entering the next round so it can be bottle-necked by the slowest devices. The first step to optimize the FL technique is to not necessarily wait for all the devices to complete the training round. With some experiment they made they found out that above a certain



threshold the accuracy improvement is very low which means that not all the devices are required in the training process but only a certain percentage.

The first graph here show the different computational time of 4 devices with different hardware specifics based on the same size of data set. The second graph shows the model accuracy that, as we said, it increases very slowly after 75-80% of the devices have submitted their models. By this observation we can understand that only a certain percentage of devices is required.

For the second step, we can observe that the most powerful devices will use the maximum of their resource for a short time while the other devices will take longer time to train the model.



The following first graph shows the CPU Frequency of the devices involved in the training process and they use the maximum amount of CPU frequency and the second graph shows the energy consumption that has an higher value by increasing the CPU frequency and a high value at low frequency as it increases the training process time, so a good pick for the frequency will be in the middle of the available frequencies.

Framework Design

The paper proposes a global and a local pace controller. The global pace controller firstly checks the status of every connected devices (connectivity, availability and energy/resource remained). After that each device sends to the server its hardware specifics. Upon reception of informations, the global pace controller estimates a virtual deadline within that the devices have to complete the training process, then broadcasts it to all participants. The local pace controller at this point determines the optimal hardware configuration in order to meet the virtual deadline. When the deadline arrives the global controller checks for all the received models and, if it has received enough weight updates to guarantee the model accuracy, the systems enters the next stage of training, in case it has not enough weight updates then it compute synchronization deadline and broadcasts it to the remaining devices in order to meet the minimum amount of weights update. Upon receiving the new deadline, the local pace controller again configure the hardware in order to meet the new deadline and try to complete their remaining job. If, at the end of the synchronization deadline, the server has enough weight updates it will perform a model averaging and start the next round. Otherwise the current trading round restarts without waiting indefinitely.

Experiment

My experiment consist on simulating a federated learning environment with a distribution of devices with different hardware specifics (# of CPUs and Interval of CPU Frequency available - Max and Mix Frequency) All were based on a variable called quality. First step was of the global pace controller to compute a virtual deadline using the formula on the paper:

```
completition_time_model = lambda nCPU,nData,frequency : (nCPU*nData)/frequency
global_deadline = deadline_determination(devices,nData,required_devices,start-
ing_deadline)
```

And send the deadline to all devices. Upon receiving the deadline, the devices will adjust the cpu to be used by using the setOptimalCpu function that is based on the deadline, the size of the training data and interval of cpu frequencies of the device. It saves all the CPU frequency alongside the energy consumption for that specific frequency, for example:

```
Device ID: 190 - quality [15]: Frequency MAX: 1.65 OPTIMAL CPU: 0.0 POWER
TRAIN: 0.0 POWER IDLE: 0.0
{0.9: 3862.327, 1.0: 4847.4, 1.1: 5962.337, 1.2: 7210.283, 1.3: 8596.354, 1.4:
10126.277, 1.5: 11804.335, 1.6: 13633.921}
```

After this is checked how many devices can meet the virtual deadline, if the number is below the required_devices variable then it compute the synchronization deadline until it reaches the number of required devices.

At the end all the optimal CPU and Energy consumption is stored and compared to the classical approach which is waiting for all the devices to send the weights update e using the maximum amount of energy and this results in:

```
SMART PC AVG ENERGY CONSUMPTION:3830.0
CLASSIC APPROACH AVG ENERGY CONSUMPTION:18215.78
SMARTPC performs 21.03% better than classical approach
################################################################
SMART PC DEADLINE:6680.0
CLASSIC APPROACH DEADLINE:20045.0
SMARTPC performs 33.33% better than classical approach
```

Evaluation

This framework proposed can achieve great improvement over the classical base approach by reaching up to 32.8% less energy consumption and accelerate the overall training progress up to 2.27 time. In my experiment I have used pseudo random numbers based on real life values and managed to achieve a great improvement over the base model.