



Deploying Machine Learning Application on Edge Devices, a Study of Challenges and Solutions

Students: Osmany Pujol Cancio, Manuel Muniz, Ethan Schnellinger, Manuel Briseno, Jonathan Diaz-Arencibia, Matthew Duyvelaar, Gabriela Ramirez, Damaso Alberto, Jose Espinoza

Mentor: Luis Robaina, MITRE

Instructor/Faculty: Seyedmasoud Sadjadi, PhD, Florida International University

PROBLEM

Even though there has been a lot of innovation in the world of Artificial Intelligence, most of the current research efforts are focused on developing better algorithms and architectures while not much effort into understanding the engineering challenges of bringing AI systems to edge devices. This project is aimed at understanding the theoretical and practical challenges of building AI solutions, best practices and rationale for developing, securing and deploying AI into edge devices.

CURRENT SYSTEM

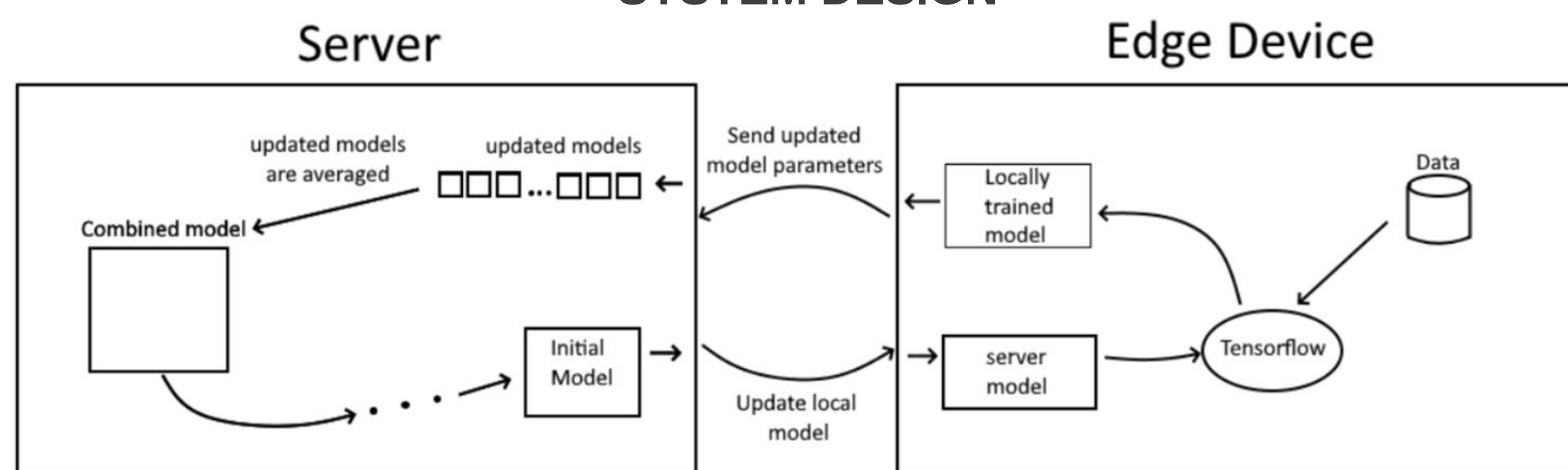
There are techniques like Federated Learning for training the model on capable devices and update the model parameters to the cloud, and Edge2Train which is a framework that allows for the training of AI models in resource scarce edge devices locally and offline. These techniques have a lot of potential but they still not always applicable to every edge device.

REQUIREMENTS

Create a research paper on machine learning with edge devices by:

- Researching
 - The development of AI/ ML models for edge devices.
 - The deployment of AI/ ML applications to edge devices.
 - How to secure the AI/ ML applications once deployed on edge devices.
- Learning how to use LaTeX in VSCode to write all research.
- Uploading all research to GitHub.

SYSTEM DESIGN



Federated Learning

This ML architecture model is a hybrid between fully centralized cloud computing and fully local edge training.

SCREENSHOTS

2.4 Running the Model: Cloud vs Local

A common approach to ML at the edge is to process and train data at data centers or cloud servers. Cloud computing has allowed for devices that are otherwise constrained in power to gain access to higher level capabilities, granted that the device has internet access[6]. In addition to internet requirement, cloud computing imposes additional constraints. One must factor in the latency (as measured with respect to a cloud), the risk of having cloud servers running and the edge device connected, and the risks of having potentially private data transferred over the network. It is a world where network capacity can often be constrained and speed can vary from location to location, cloud computing is not a feasible method of granting access to models for every device. Distance inherently impacts the latency and connection between cloud and device, with the optimal solution often being the deployment of additional data centers or networks near the device when far from the host machine[2]. In situations where internet access is consistent or latency isn't a worry, cloud computing is the best way to ensure accuracy and model strength. However, bandwidth and network have been addressed in order to allow edge devices to collect and train data locally while still maintaining performance similar to that of models running on the cloud. There are two methods in particular that offer extreme potential for ML at the edge, federated learning and the Edge2Train framework.

FIU: ALL RIGHTS RESERVED FOR INTERNAL USE ONLY

2

2.5 Federated Learning

Federated learning can be considered a hybrid between fully centralized cloud computing and fully local edge training. Federated learning addresses some of the aforementioned issues that arise with cloud computing, while allowing access to some new benefits. As a baseline, federated learning is primarily dedicated to using edge devices as a resource in order to simply train and parameterize data across locations after being given access to the global ML model from a central server. It is then possible for said devices to push their model change to the cloud, allowing the global model to be updated[5]. The decentralized method of data capturing and training allow for the more specialized and developed models to form. In fact, federated learning often allows for higher levels of accuracy than found in models run on solely the cloud.

Figure 1: Diagram of federated learning.

In an experiment run by the Nguyen Institute of Technology[2], simulations were done in order to compare centralized ML and ML with federated learning. Using the MNIST and CIFAR-10 datasets, the researchers found that centralized ML achieved 92.7% accuracy with MNIST and 74.9% accuracy with CIFAR-10, whereas federated learning models achieved accuracy levels of 93.9% and 86.9% on each respective model. At worst, federated learning's accuracy was 2% higher than centralized ML, and at best an increase of 2% was seen. These results show that the advantage of federated learning are much greater than that of centralized ML, while also allowing for offline functionality.

2.6 Edge2Train

Not every edge device comes with the resources necessary to take advantage of federated learning, yet would likely still prefer an alternative to training on the cloud. For such devices, the Edge2Train framework is an alternative with a large amount of potential. Edge2Train is a novel framework designed in order to allow small resource-constrained edge devices (MOTUs) which are often limited in storage and computational power, to gain the ability to train models locally and offline. The framework is able to learn from the edge device that data data commands that can interact with the edge device's application, allowing for activation. The Edge2Train framework then pushes the data to a gate in the application, runs, training and updating the model [3]. Edge2Train uses C++ in order to optimize training methods to function on non resource-capable applications. Edge2Train's training method optimizes data so as to more compatible with those lower-powered edge devices, ensuring low resource while maintaining reliability. The MOTUs used in the experiment were highly limited in terms of accuracy, with the range of 50-60% from a system of 200 to a system of 500. Despite this limited ML, all models were comparable of being trained on the ML, the general edge devices are subject to such data as baseline in order to learn correctly. Researchers also observed that accuracy is relatively maintained despite the low in resources when compared to models trained on computers. The first dataset in the experiment showed that accuracy on the computer and 80.9% on the Edge2Train device. The second dataset actually indicated higher accuracy on the edge device than on the computer, with Edge2Train reaching 82.9% accuracy, whereas the computer had 80.4%. These results indicate that Edge2Train's framework is capable of producing classification at the same level as those on models trained on powerful computers, without the requirement of sending a cloud connection.

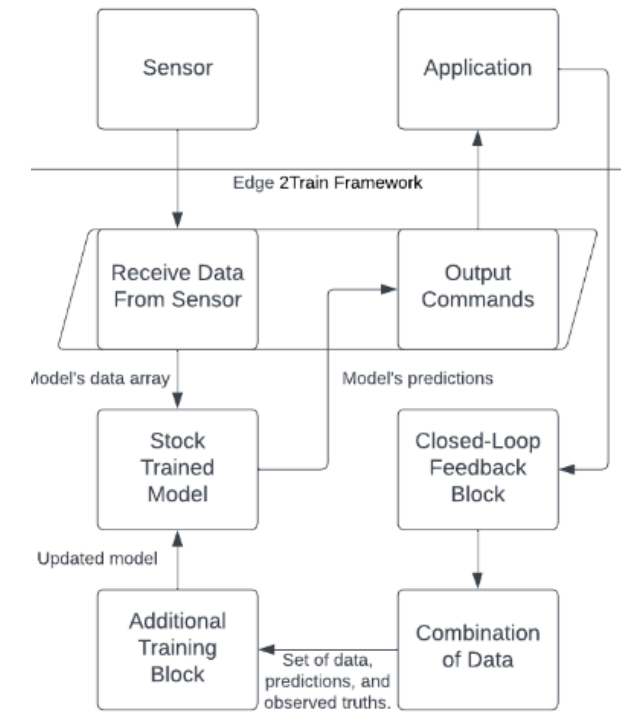
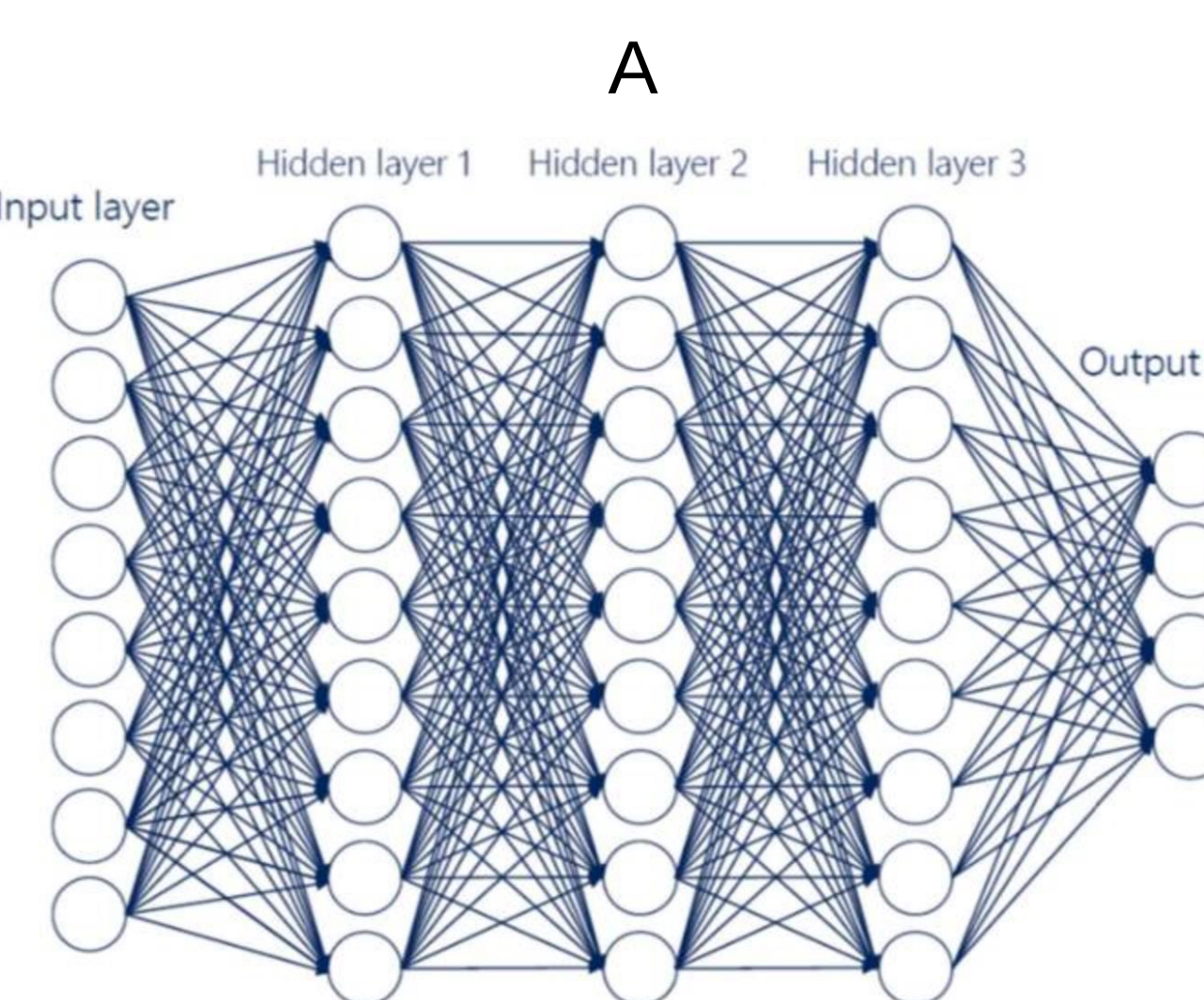


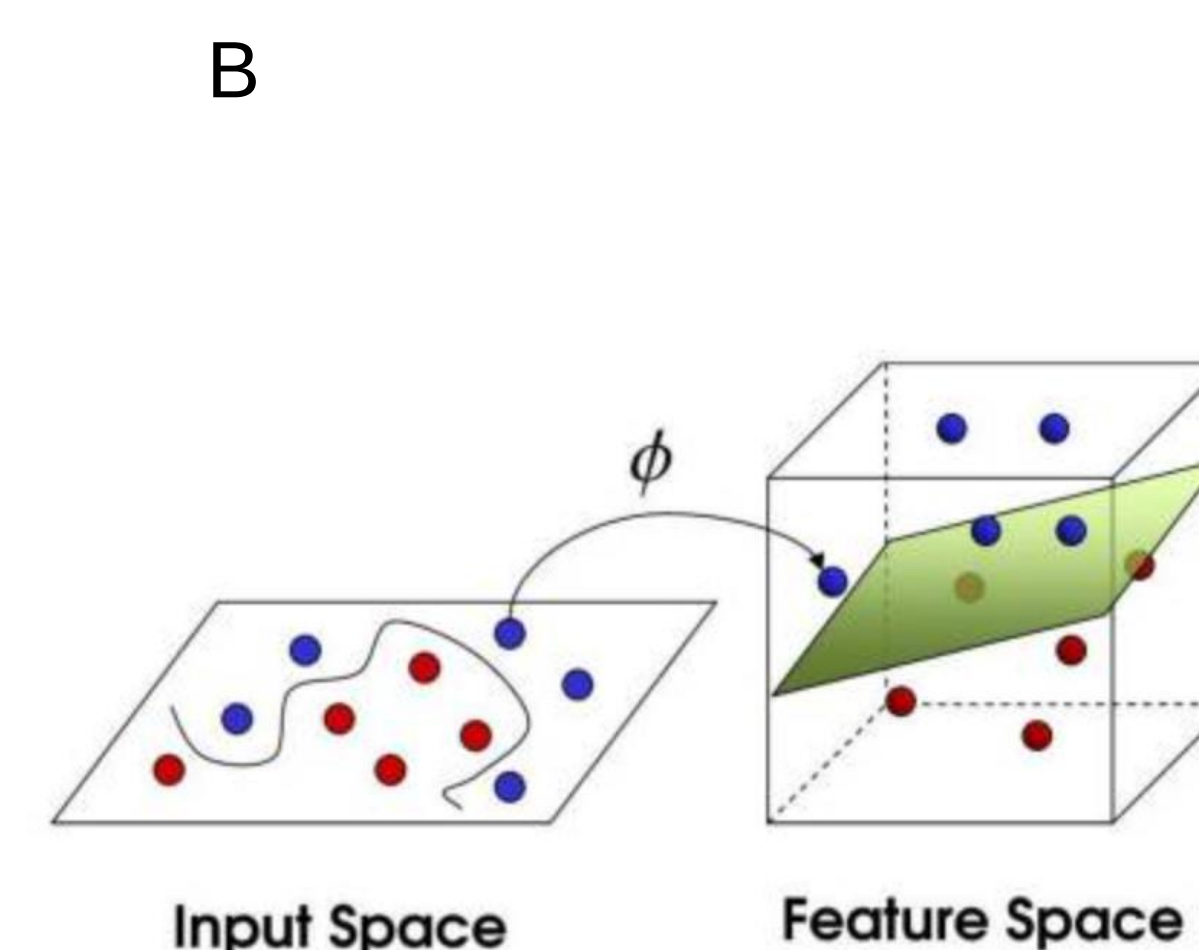
Figure 2: Diagram of Edge2Train

OBJECT DESIGN



(A) Deep Neural Network (DNN)

DNNs involves the usage of the matrix multiplications with convolutional filter operations, which is common in DNNs that are designed for image and video analysis



(B) Support Vector Machine (SVM)

SVM is a supervised learning algorithm that can be used for both classification and regression problem

SUMMARY

In summary, we covered in this research paper the challenges of bringing machine learning to edge devices. We covered the different areas such as development, deployment, and security of ML on edge devices. We touched upon the difference of running the model on cloud vs locally and the different types of frameworks. In terms of security we looked over the types of threats that can occur on edge devices and the defense for those security threats. Finally, we looked at monitoring after deployment of the model and wrote on the future work of this project.

Verification

The verification process starts by having our CAP 1 students research and review academic articles relating to our areas of research, then providing the resources to our CAP 2 students where they will rereview articles and take out important information that will help further our research.

REFERENCES

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7273223/>

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Luis Robaina with MITRE. We thank Luis Robaina and MITRE for their assistance and mentorship that we received throughout the senior design project.