Project: Using Cascade Models to Reduce Inference Times and the Resource Usage in Edge Computing Scenarios

Context:
The exponential advances in Machine Learning (ML) are leading to the deployment of Machine Learning models in constrained and embedded devices, to solve complex inference tasks. At the moment, to serve these tasks, there exist two main solutions: run the model on the end device, or send the request to a remote server. However, these solutions may not suit all the possible scenarios in terms of accuracy or inference time, requiring alternative solutions.
Cascade inference is an important technique for performing real-time and accurate inference given limited computing resources such as MEC servers. It combines more than two models to perform inference: a highly-accurate but expensive model with a low-accuracy but fast model, and determines whether the expensive model should make a prediction or not based on the confidence score of the fast model. A large pool of works exploited this solution. The first ones to propose a sequential combination of models were [1] for face detection tasks, then, in the context of deep learning, cascades have been applied in numerous tasks.

Goal:
The project is to use cascade models in the context of Edge Computing to improve the delay and reduce the resource usage of ML inference tasks at the edge.
Of crucial importance for cascade models is the confidence of the fast model. Indeed, if the prediction of the first model is used but wrong, it may lead to a low accuracy of the cascade model, even if the accuracy of the best model is very high. Similarly, if the first model confidence is set too low, it will never be used, and the computations will be higher than using only the second model by itself, additionally, we will use unnecessary network resources and have higher deals than necessary.

In this project, we will consider the **regression task** of *predicting an individual's income* using the *dataset Income* (Ding et al. 2024). This dataset contains census information from 50 U.S. states and Puerto Rico, spanning from 2014 to 2018. It includes 15 features related to demographic information such as age, occupation, and education level. We will use methods coming from probabilistic inference to assess the confidence of the model when providing an answer.

1- Read the paper:

[1] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). Ieee.

Explain how cascade models can be used to improve Edge Computing. Give precise scenarios.

2- Explain how to assess the confidence of the result of a regression task.
- Using the method explained in the slide with the Gaussian Negative Log-Likelihood loss.
- Show the method on a simple data set with a single feature with a data set with some non uniform noise.
- Going further: Get some information about **quantile regression.** Explain how it could be used.

3- Train both a small and a large model on the dataset, e.g.
- Small model: linear regression + confidence
- Multilayer Linear Perceptron with confidence

Present the error obtained by both models on a test set.

4- Try different model sizes for the datasets, show their error and inference time as a function of their size. Select a small and a large model for your cascade system.

5- Present the error obtained by the cascade system using both models. Test different confidence thresholds and find the best one. Discuss the gain of your system in terms of inference time and resource usage (bandwidth, computation, memory, energy).

6- Test the cascade system with different loss functions for the small model. Adapt the following losses to our context, i.e. to have a confidence score:

- Mean Squared Error (MSE)
- Mean squared logarithmic error (MSLE)
- Root mean squared error (RMSE)
- Mean absolute error (MAE) or L1-Loss
- Huber loss also called smooth L1 loss

Compare the results and conclude.

Going further:

7- Present results for other regression tasks and other datasets.

8- Use a different training procedure.

- Train first the fast model on the dataset.
- Then, train the large model on images/individuals not well recognised/modeled by the small model. Does this improve the cascade system?
- Hard: implement the quantile regression.