# TinyML: The challenge of edge devices

## Summary of the paper TinyMLOps: Operational Challenges for Widespread Edge AI Adoption

Author: Morsinaldo de Azevedo Medeiros - Undergraduate student in Computer Engineering at the Federal University of Rio Grande do Norte - Brazil
E-mail: morsinaldo.medeiros.075@ufrn.edu.br
LinkedIn: https://www.linkedin.com/in/morsinaldo-medeiros-288053105/

### English Version

Tiny Machine Learning (TinyML for short) is a subject that has been increasingly researched on the internet and more and more requested by large companies. This area consists of implementing a machine learning model on an edge device with limited resources, i.e. low processing power, low power consumption, low memory, etc. As Leroux et al (2022) discusses in their article, there are several challenges in this area that is so commercially attractive to enterprises today. The first point to discuss is TinyMLOps which is a set of development best practices that enable the automation, monitoring, integration and testing of machine learning models. Since the deployment of the models will be done on the users' edge devices, instead of training a single model, we might need to support multiple models, each with their own computational cost and accuracy trade off. One factor that contributes to solving this problem is to reduce the precision of the operations from 32 to 8 bits, for example. The second challenge is observability, that is, the monitoring of user-generated data by the company to know if data drift is occurring or not. There are some platforms that monitor and warn if the data distribution is changing, but this requires users to agree to send their data to the cloud, an issue that involves security and privacy. The third point mentioned by the author is pay-per-query, a viable business model in this area, i.e. the user pays a certain amount for a certain number of inferences from the model. Google's Cloud vision API for example charges users $1.50 per 1,000 requests for tasks such as face detection. The fourth challenge mentioned by the author is retraining and customizing the models, because assuming users agree to send their data to the cloud (second challenge), the author suggests creating a Federated Learning, which is composed of the user's own information (thus creating a personalized model) along with the average of other users' models. But this creates another challenge which is the heterogeneity of the data that can be generated by people around the world, thus needing further research regarding the solution to this problem. Perhaps the creation of a Federated Learning by region? In addition to these challenges, the diversity of devices at the edge is very

large, each with different architectures and optimizations. This creates the need to create virtual environments and use containers for the correct functioning of the applications. This virtualization could also enable hybrid edge-cloud applications where, depending on the available resources, the model is evaluated on edge or cloud hardware. Once these difficulties are overcome, the need for intellectual protection arises, because the trained machine learning model can represent a great commercial value for a certain company, meaning that it is not feasible for other people/companies to use your model without limitations for their purposes. Furthermore, training a model requires a great deal of knowledge on the part of the programmer, not to mention the days or weeks that these models are left training and tuning on powerful infrastructures, especially if the model is optimized for edge devices. So it is not only necessary to protect the model in the cloud, it is necessary to protect the model (including its weights) on the edge device as well, because someone could hack and get access to the model or even put a monitor collecting the data and predictions to create a new model very similar with a sufficient amount of data. A possible solution to this is to use encrypted models, however this technique greatly increases the computational cost of decryption before the model is loaded into memory. A malicious user could change the predictions of the model, e.g. infect other devices and steal personal data, banking data, etc. Therefore, it is necessary to create solutions for these problems in order to spread TinyML more and more.

**Portuguese Version**

Tiny Machine Learning (TinyML) é um assunto que vem sendo cada vez mais pesquisado na internet e mais requisitado pelas grandes empresas. Essa área consiste na implementação de um modelo de aprendizagem de máquina em dispositivo da borda com recursos limitados, isto é, com baixa capacidade de processamento, baixo consumo de energia, baixa memória etc. Como Leroux et al (2022) discute em seu artigo, existem vários desafios nessa área que é tão atrativa comercialmente para as empresas nos dias de hoje. O primeiro ponto a ser discutido é o TinyMLOps que é um conjunto de boas práticas de desenvolvimento que permitem a automação, monitoramento, integração e teste de modelos de aprendizagem de máquina. Como o deploy dos modelos será feito nos dispositivos da borda dos usuários, ao invés de treinar apenas um modelo, nós precisamos oferecer suporte para vários modelos, cada um com uma relação entre custo computacional e acurácia.. Um fator que contribui para a solução deste problema é reduzir a precisão das operações de 32 para 8 bits, por exemplo. O segundo desafio é a observabilidade, isto é, o monitoramento dos dados gerados pelo usuário por parte da empresa para que ela saiba se está ocorrendo ou não o data drift. Existem algumas plataformas que fazem o monitoramento e avisam caso a distribuição dos dados esteja mudando, mas isso requer que os usuários concordem em enviar os seus dados para a cloud, questão essa que envolve

segurança e privacidade. O terceiro ponto mencionado pelo autor é o pay-per-query, um modelo de negócios viável nessa área, ou seja, o usuário paga um determinado valor para um determinado número de inferências do modelo. O API Cloud Vision da Google, por exemplo, cobra U$1.50 para cada 1.000 requisições para tarefas como detecção de faces. O quarto desafio mencionado pelo autor é o retreinamento e a personalização dos modelos, pois, supondo que os usuários concordem em enviar os seus dados para a cloud (segundo desafio), o autor sugere a criação de um modelo compartilhado, o que é composto pelas informações do próprio usuário (criando assim um modelo personalizado) juntamente com a média dos modelos dos outros usuários. Mas isso cria outro desafio que é a heterogeneidade dos dados que pode ser gerada por pessoas ao redor do mundo, necessitando assim de mais pesquisas em relação à solução deste problema. Quem sabe a criação de um modelo compartilhado por região? Além desses desafios, a diversidade de dispositivos na borda é muito grande, cada um com arquiteturas e otimizações diferentes. Isso cria a necessidade de criar ambientes virtuais e de utilizar containers para o funcionamento correto das aplicações. Esta virtualização pode também permitir aplicações híbridas em nuvem onde, dependendo dos recursos disponíveis, o modelo é avaliado em hardware de bordo ou de nuvem. Uma vez que essas dificuldades vão sendo superadas, vai surgindo a necessidade de proteção intelectual, pois o modelo de machine learning treinado pode representar um grande valor comercial para determinada empresa, significando assim que não é viável que outras pessoas/empresas utilizem o seu modelo sem limitações para os seus propósitos. Além disso, treinar um modelo requer um grande conhecimento por parte do programador, sem contar os dias ou semanas que esses modelos ficam treinando e tunando em infraestruturas potentes, especialmente se o modelo for otimizado para os dispositivos da borda. Assim, não se faz necessário proteger o modelo apenas na cloud, é necessário proteger o modelo (incluindo seus pesos) no dispositivo da borda também, pois alguém poderia hackear e conseguir o acesso ao modelo ou até mesmo colocar um monitorador coletando os dados e as previsões para criar um novo modelo muito parecido com uma quantidade suficiente de dados. Uma possível solução para isso é a utilização de modelos encriptados, porém essa técnica aumenta muito o custo computacional da desencriptação antes da utilização do modelo ser carregado na memória. Um usuário malicioso poderia mudar as previsões do modelo, por exemplo, infectar outros dispositivos e roubar dados pessoais, bancários etc. Por isso, faz-se necessário a criação de soluções para esses problemas visando a difusão de cada vez maior do TinyML.

**Reference:**

Leroux, Sam et al. TinyMLOps: Operational Challenges for Widespread Edge AI Adoption, 2022.