**TensorFlow Federated – evaluation**

* **General information**

TensorFlow Federated is built on TensorFlow, following a simple analogy: TF is dedicated to machine learning solutions, while TFF is suitable for federated learning that builds upon classic deep learning. It was developed by Google and can be combined with other tools created with TensorFlow as a base. It can be used not only for machine learning but also for data analysis. The best known TFF application in real-world is Google’s keyboard predictions. They trained the model supporting keyboard suggestions for each user without sharing their data. In other cases, this package is mostly used in academia for experimentation.

TensorFlow Federated provides decorators to determine what kind of function we are creating. It can separate our general code from the process designed for the edge device specifically. The concepts of federated values and federated computations are introduced. We also need to define server and client updates and synchronize their cooperation. A separate function calling next step is necessary. There is a lot of flexibility within this framework; changes to server and/or client can be done at any time.

Given the above description, it can be noticed that this approach slightly resembles solutions applied in other packages. The workflow matches the classic federated learning dynamics, there are decorators used to dignify different functions of the code, and separate calls are needed to push the flow forward.

* **How is privacy preserved?**

By default, federated learning is secure – no data is shared, only the model updates. So as long as the designed code aligns with those assumptions, your data is safe. TensorFlow has also its own package for Differential Privacy, and it can be applied on top of federated learning application. Moreover, other popular FL solutions such as secure aggregation can be implemented to the whole training loop. However, in practice, it is often used with encryption protocols and other cryptographic methods.

* **Ease of use**

Overall, once the package is installed and the environment is set in a stable way, TensorFlow Federated is relatively used to use, given that you understand machine learning principles. Adding layers, connecting them, defining functions can be done is a few lines of code. The same goes for evaluation and metrics definition. It is similar to PyTorch but can be more compact at times in terms of syntax.

In general, the package is well developed, however, has many caveats related to strict version requirements. It has wide support by Google and active community, but it does not always come with the quality of the experience of programming with it.

* **What works well**
  + Big community for TensorFlow, available on multiple platforms. Only some parts of it consider TFF as well; however, it seems like it is relatively easy to receive help from the community (for TF).
  + There are datasets and models available (for TF but can be used with TFF): it handles downloading and preparing the data deterministically and constructing a ready Dataset. Models are available from Kaggle.
  + TensorFlow is compatible with many other tools offered by Google. Therefore, it is possible to start using it from quick prototype to fully developed product ready for real-world use cases.
  + Visualization tool available: TensorBoard. It allows for clear visualization of all machine learning experimentations.
  + LiteRT: a tool allowing for quick deployment on edge devices (important for FL); it is especially handy since applications can be written in different programming languages.
  + TFF has many commands that can speed up the whole process of setting up experiments.
  + Simple way of building machine learning models.
  + Many tutorials on how to approach different topics and showing possibilities within TFF.
  + Flexible way of adjusting models.
* **What could be improved** 
  + Installation, it is absolutely horrible and takes way too much time. Even though there are notebooks, some of them do not take into account versions mismatches. User has to figure out everything on their own since notebooks do not mention when version of a particular package was changed.
  + Even on the official website, it is mentioned that some of the versions are not available on specific software – how can it be widely used without any issues?
  + Deployment tool – there is not that much attention to that on the official website.
* **How to make a model in TensorFlow (Federated)?**

1. Define a model; here we can do it by assigning it to a variable, like with any other variable assignment. The simplest one is Sequential, where we can define layers and what happens with the input at each step. You should not return probabilities at this step.
2. You can normalize your data; however, you can do it only on the train data. It can be done by Normalize() function that can be applied on the inputs via adapt() method. In that case, you also need to add that layer to your model at the beginning.
3. Define function converting the output to probabilities.
4. Define a loss function.
5. Compile your model by adding optimizer and loss function and choose your metrics.
6. Run the model with fit() function.

Note: even if you want to use trained model to predict a result for a single input, you have to add it to a list: this is what keras expects.

Note 2: Keras can preprocessing layer to the model!

If you want to advance this approach to federated learning, you have to:

1. Initialize or import weights.
2. Add evaluation function – to match TFF pipeline where model is evaluated before the training to define a baseline result.
3. Define a training loop collecting all defined elements. Include also updating weights.
4. Return the results.

* **System requirements**

While working with this method I encountered a compatibility issue. Officially, it is confirmed that TensorFlow 2.11 is compatible with TensorFlow-Federated 0.48. Any other installations might cause conflicts. However, this is slightly problematic scenario. TensorFlow is supported mainly on CPU processors. If you do not have one, there are some packages available on the official TensorFlow website that allow for the installation of a desired version of TensorFlow. Unfortunately, the oldest version available is 2.16 (the newest is 2.19), making it impossible to find a compatible version with TensorFlow Federated. Therefore, I tried running it on Google Colab. Unfortunately, it runs on the 3.11 version of Python. As it turns out, the oldest available version of TensorFlow compatible with this version of Python is 2.12 which creates conflict with TensorFlow-Federated. Consequently, primary I was not able to run TensorFlow Federated on my local machine.

As it is stated on the website “*Currently there is no official GPU support for running TensorFlow on MacOS. The following instructions are for running on CPU.”* There are some plugins that work around this issue.

* **Sources**

Installation of TensorFlow on different machines (files to be downloaded): <https://www.tensorflow.org/install/pip#macos_1>

Tutorials with notebooks ready to run: <https://www.tensorflow.org/tutorials/text>

Official tutorial and walkthrough about TensorFlow-Federated: <https://www.youtube.com/watch?v=JBNas6Yd30A>

Official Git repository: <https://github.com/google-parfait/tensorflow-federated/tree/main>

Official Git tutorials and tips: <https://github.com/google-parfait/tensorflow-federated/blob/main/docs/federated_learning.md>

Official website: <https://www.tensorflow.org/federated>

Datasets and how to use them: <https://www.tensorflow.org/datasets/overview>

Quick deployment on edge devices: <https://ai.google.dev/edge/litert>

Visualization tool: <https://www.tensorflow.org/tensorboard>

TensorFlow documentation (knowledge of how to build networks required for TFF): <https://www.tensorflow.org/api_docs/python/tf>

How to build your own algorithm: <https://github.com/google-parfait/tensorflow-federated/blob/v0.88.0/docs/tutorials/building_your_own_federated_learning_algorithm.ipynb>

* **Notebooks from TensorFlow**

Beginner’s introduction: <https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb#scrollTo=h3IKyzTCDNGo>

General tutorial page: <https://www.tensorflow.org/tutorials>