Collaboration without sharing
- A solution for Vietnam Companies

Dr. Cao Tiến Dũng

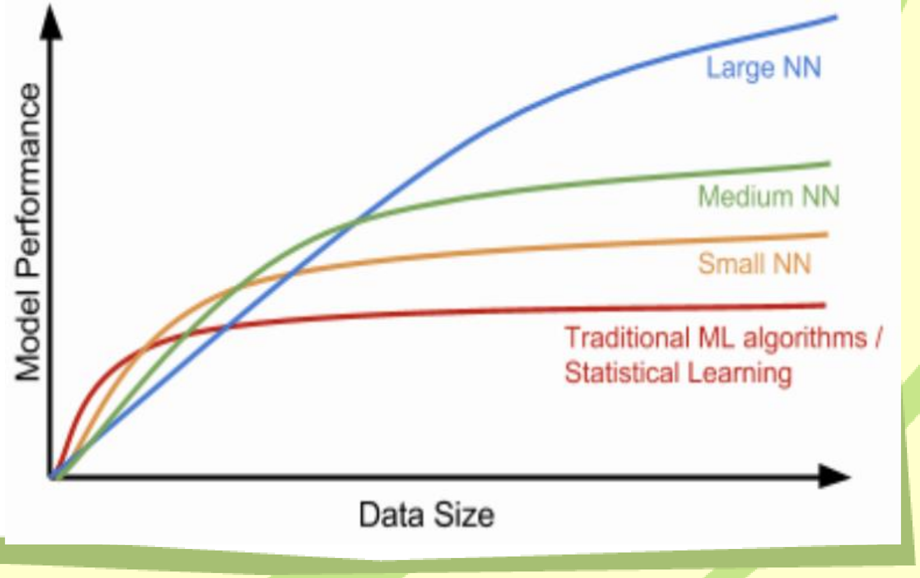
MOBILEDAY

About me

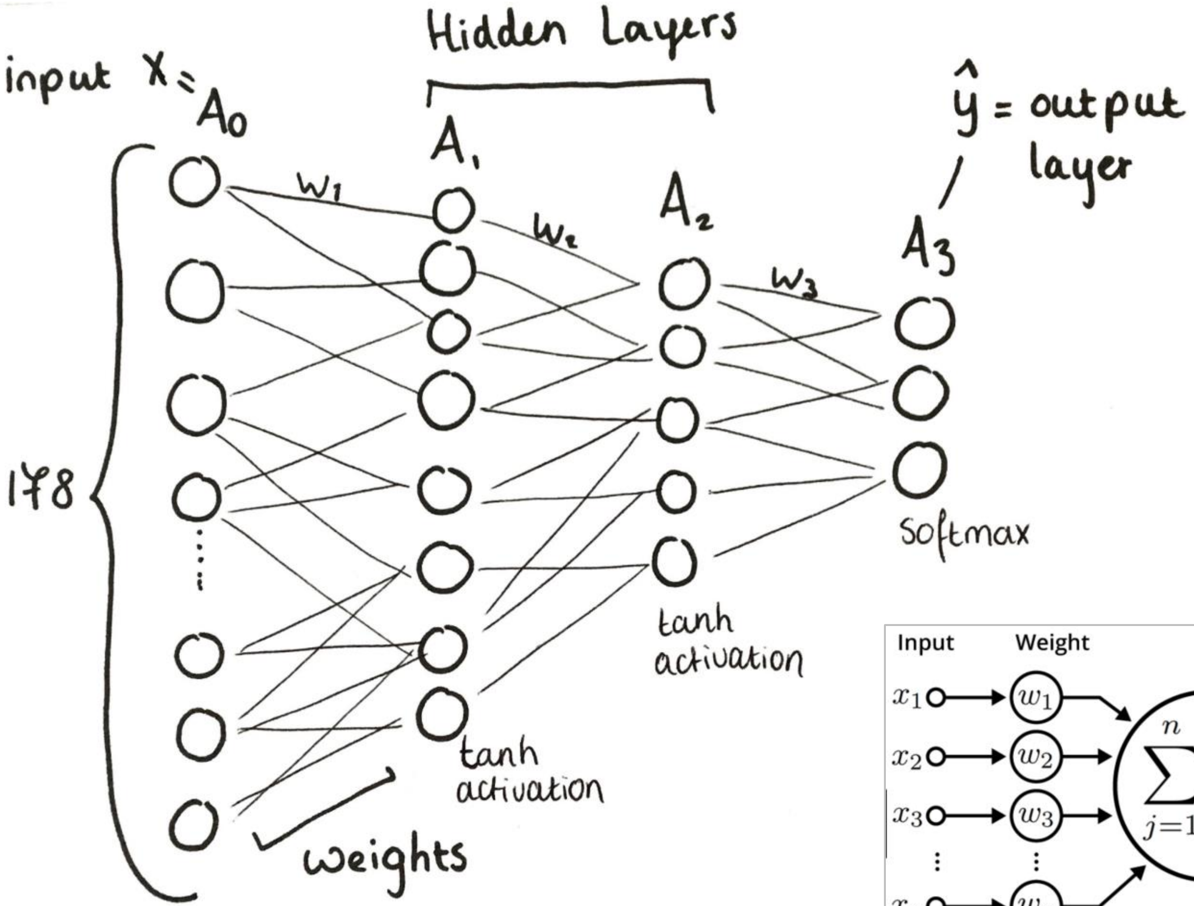
- PhD in Computer Science, University of Bordeaux (2010).
- Co-Founder, Vicohub
- Vice-Dean, School of Engineering, Tan Tao University
- Research interests:
  - Data analytics
  - Machine/Deep Learning
  - Cloud computing

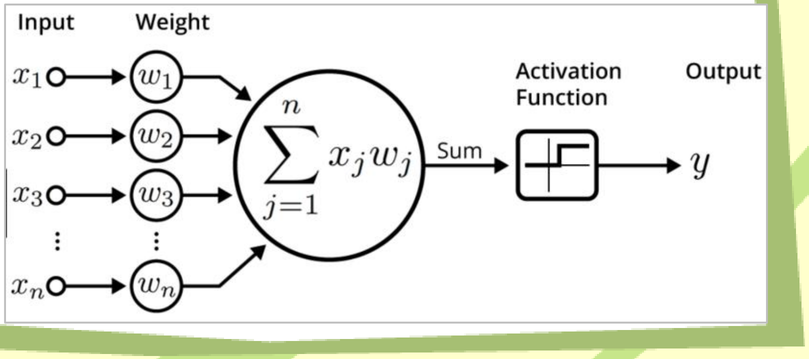
Deep Learning

- A machine learning methods based on Artificial Neural Networks.
  - A type of Supervised Learning (Output is known for learning)
- Why deep learning?



Neural Network





Learning = Optimization

- Finding the weight W of the neural network minimizing the function.

$$J(W) = \sum_{i=1}^{N} |\mathrm{NeuralNet}(W, x_i) - y_i|^2$$

  - x_i is data sample and y_i is its label

Question???

- How much (**clean**) data is enough for DL?
  - 1 GB, TB, PB, ZB…
- What if I have a small data.
  - Your **accuracy/performance** ?
- How current platforms (e.g., TensorFlow) train my model??
  - Need to centralize data

How to get more data??

- Do your partners/competitors have data like you?
  - May YES
- But cannot share to you because of
  - Data is money
  - Data is strategy
  - Data is everything
  - …

Collaboration or nothing

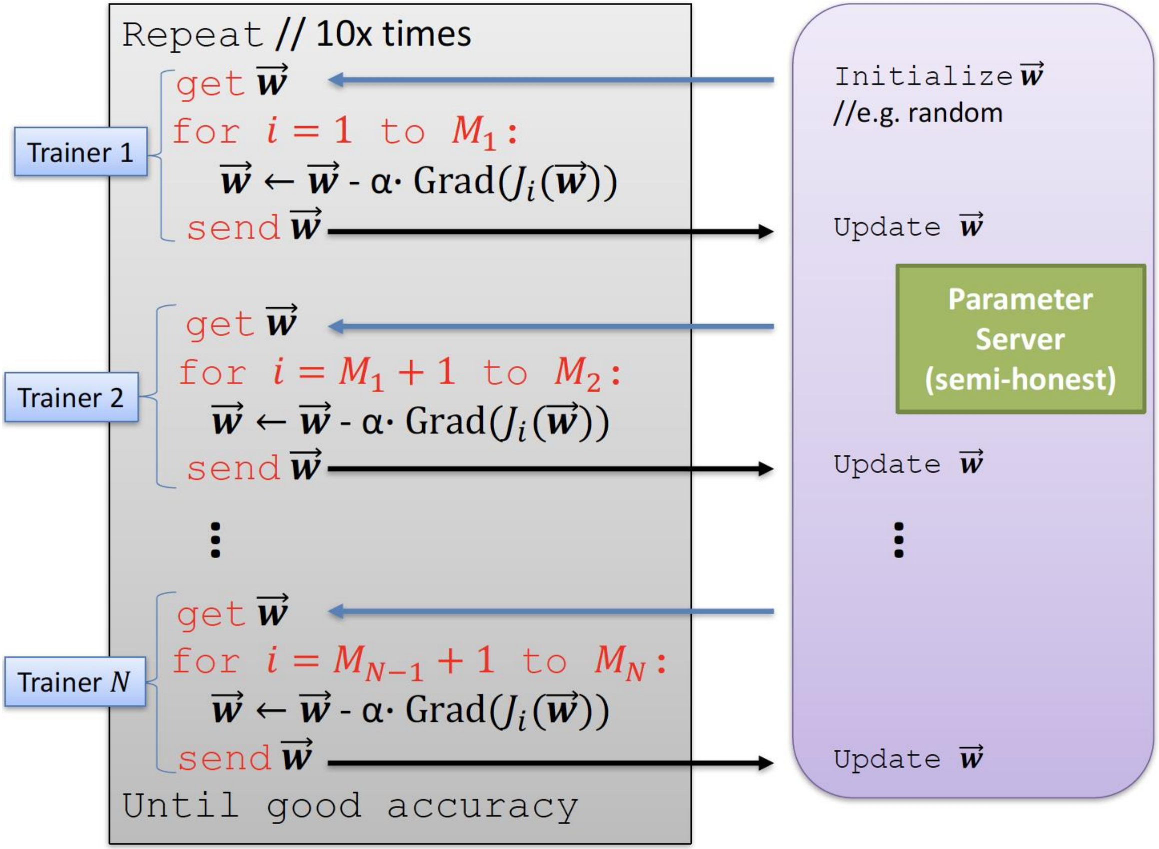
- N datasets D_1, D_2, … ,D_N of N **distributed/separated** owners.
- The owners would like to do machine learning over $D_1 \cup D_2 \cup \ldots \cup D_N$

to maximize the learning outcome.

- But worry about **security/privacy** of their data.

Solution

- Each owner setups a platform to train a network, but sharing parameter (weight or gradient) with others during training process to optimize outcome.

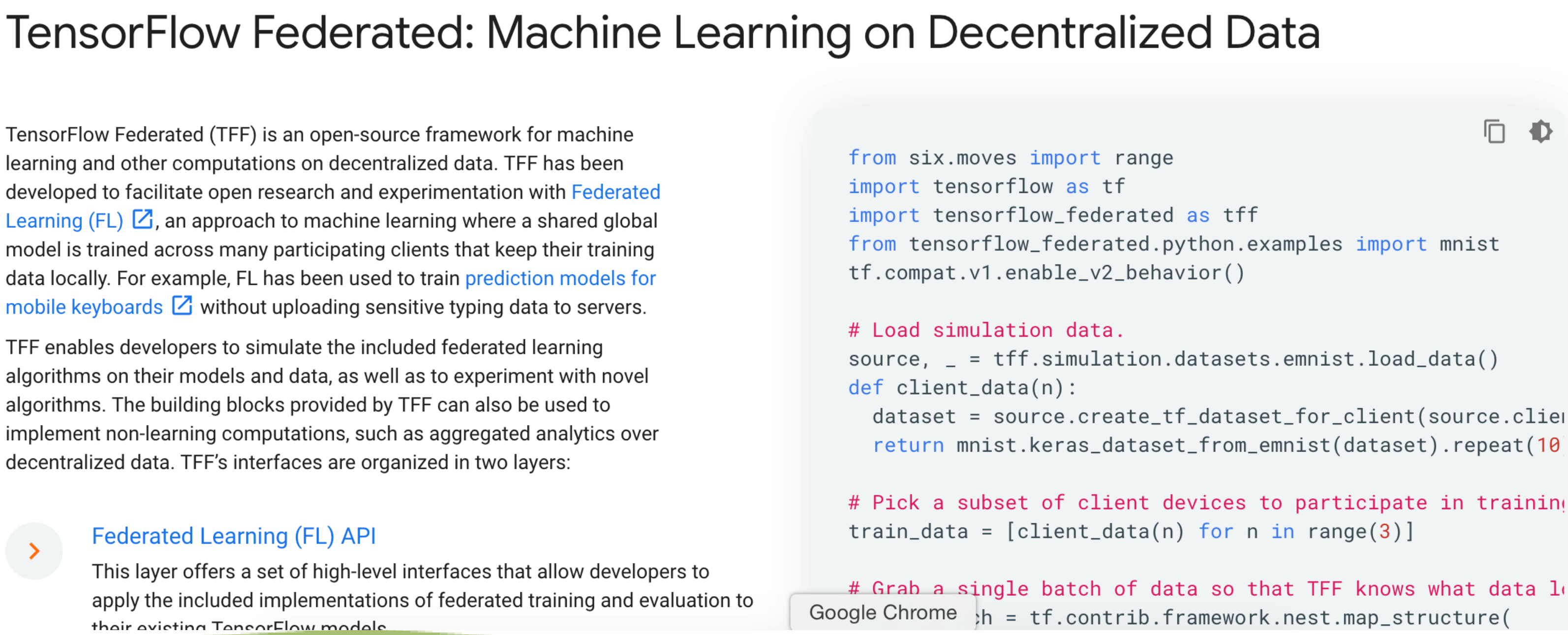


```
Repeat // 10x times
    get $\vec{w}$
    for $i = 1$ to $M_1$:
        $\vec{w} \leftarrow \vec{w}$ - α· Grad($J_i(\vec{w})$)
    send $\vec{w}$

    get $\vec{w}$
    for $i = M_1 + 1$ to $M_2$:
        $\vec{w} \leftarrow \vec{w}$ - α· Grad($J_i(\vec{w})$)
    send $\vec{w}$

    ⋮

    get $\vec{w}$
    for $i = M_{N-1} + 1$ to $M_N$:
        $\vec{w} \leftarrow \vec{w}$ - α· Grad($J_i(\vec{w})$)
    send $\vec{w}$
Until good accuracy
```

```
Initialize $\vec{w}$
//e.g. random

Update $\vec{w}$

Update $\vec{w}$

⋮

Update $\vec{w}$
```

Google Federated Learning

- https://www.tensorflow.org/federated
- Release March 2019.



TensorFlow Federated: Machine Learning on Decentralized Data

TensorFlow Federated (TFF) is an open-source framework for machine learning and other computations on decentralized data. TFF has been developed to facilitate open research and experimentation with Federated Learning (FL) ↗, an approach to machine learning where a shared global model is trained across many participating clients that keep their training data locally. For example, FL has been used to train prediction models for mobile keyboards ↗ without uploading sensitive typing data to servers.

TFF enables developers to simulate the included federated learning algorithms on their models and data, as well as to experiment with novel algorithms. The building blocks provided by TFF can also be used to implement non-learning computations, such as aggregated analytics over decentralized data. TFF's interfaces are organized in two layers:

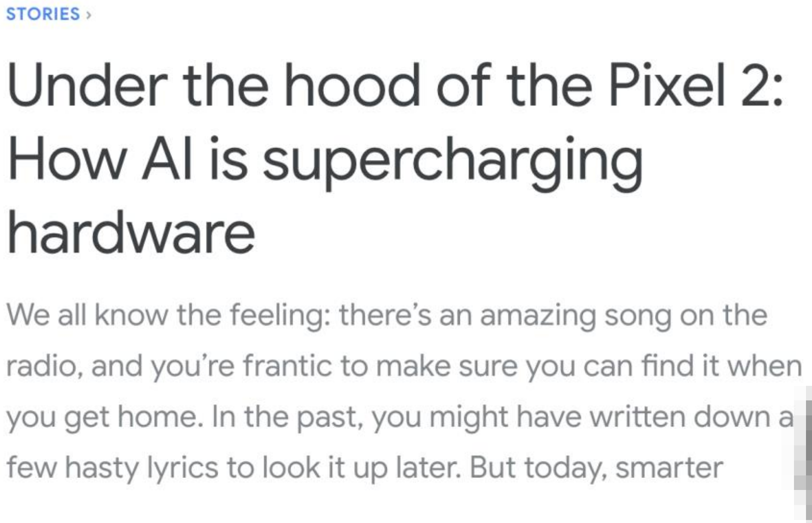
› Federated Learning (FL) API

This layer offers a set of high-level interfaces that allow developers to apply the included implementations of federated training and evaluation to their existing TensorFlow models.

```
from six.moves import range
import tensorflow as tf
import tensorflow_federated as tff
from tensorflow_federated.python.examples import mnist
tf.compat.v1.enable_v2_behavior()

# Load simulation data.
source, _ = tff.simulation.datasets.emnist.load_data()
def client_data(n):
  dataset = source.create_tf_dataset_for_client(source.clie
  return mnist.keras_dataset_from_emnist(dataset).repeat(10

# Pick a subset of client devices to participate in trainin
train_data = [client_data(n) for n in range(3)]

# Grab a single batch of data so that TFF knows what data l
      :h = tf.contrib.framework.nest.map_structure(
```

Federated Learning on Google product

- https://ai.google/stories/ai-in-hardware/



STORIES ›

Under the hood of the Pixel 2: How AI is supercharging hardware

We all know the feeling: there's an amazing song on the radio, and you're frantic to make sure you can find it when you get home. In the past, you might have written down a few hasty lyrics to look it up later. But today, smarter

“Let collaboration to understand more your customer…”

Thank you!