Secure Collaborative Training of Machine Learning Model using MPC

By Souhail Meftah, Ruomu Hou

October 22, 2019

1 Background and Motivation

1.1 Multi-party Computation

1.2 iDash Privacy and Security Workshop 2019: Secure genome analytics competition

2 Methodology and Technologies

3 Implementation

3.1 Requirement and Setup

Notice: It requires at least a setup with 12GB memory to run the code.

```
# prepare dependency
sudo apt install -y git python3 python3-pip jupyter
# install the python dependencies
pip3 install jupyter syft torch torchvision pandas
# due to syft compatibility issue, we need to downgrade torch
pip3 install --upgrade torch==1.1.0
# clone the repository
git clone https://github.com/Souhail-MEFTAH/i-dash2019.git
# launch the project
cd i-dash2019/
jupyter notebook
```

Listing 1: Dependency Setup Code

Our experiment runs on 1 machine on the Tembusu Cluster with Intel E5-2620V3, 256GB DDR3 RAM, and CentOS 7.x.x. The code has been tested on Ubuntu 18.04 as well.

3.2 Load the Data

```
import pandas as pd
import numpy as np

# read the data from the input files
def getSamples(filename):
    data = pd.read_csv(filename, sep='\t')
    return data.values[:, 1:].transpose()

data1 = getSamples("GSE2034-Normal-train.txt")
data2 = getSamples("GSE2034-Tumor-train.txt")

# code for formatting the data to numpy arrays

# partition the data into training data and test data
x_train = x[:n_train_items]
y_train = y[:n_train_items]

x_test = x[n_train_items:]
y_test = y[n_train_items:]
```

Listing 2: Loading the data

3.3 Model Creation

```
# The class defining our sub-network
class Res1d(nn.Module):
    def __init__(self, inSize, outSize, kernel=(3,), strides=1,):
        # code for defining the layers

    def forward(self, x):
        # code for defining how the layers are composed

# The class defining the overall network
class Net(nn.Module):
    def __init__(self):
        # code for defining the layers

    def forward(self, x):
        # code for defining how the layers are composed
```

Listing 3: Define the model