EAS 595: Fundamentals of Artificial Intelligence (Spring 2020)

Dr. David Doerman, Mihir Chauhan
University at Buffalo, The State University of New York
Buffalo, New York 14260
Contact: mihirhem@buffalo.edu

March 23, 2020

1 Task

You are the co-founder of an AI startup who wants to build a deep learning model to detect breast cancer. Before training your model, the first step would be to acquire a dataset. One approach could be to work with a hospital and ask them to send you a copy of this dataset. However because of the sensitivity of the patients' data, the hospital might be exposed to liability risks. That's where federated learning comes into the picture. Instead of bringing training data to the model (a central server), you bring the model to the training data (wherever it may live). In this case, it would be the hospital.

The idea is that this allows owner of the data to have the only permanent copy, and thus maintain control over who ever has access to it.

The task of this project is to perform federated classification using privacy preserving AI library Pysyft. The features used for classification are pre-computed from images of a fine needle aspirate (FNA) of a breast mass. Your task is to classify suspected FNA cells to Benign (class 0) or Malignant (class 1) using federated logistic regression model with Pytorch and pysyft. You are required to document the results.

The dataset in use is the Wisconsin Diagnostic Breast Cancer (wdbc.dataset). Deadline to submit the code and the report on timberlake server is 11.59 PM April 15, 2020.

2 Dataset

Wisconsin Diagnostic Breast Cancer (WDBC) dataset will be used for training, validation and testing. The dataset contains 569 instances with 32 attributes (ID, diagnosis (B/M), 30 real-valued input features). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Computed features describes the following characteristics of the cell nuclei present in the image:

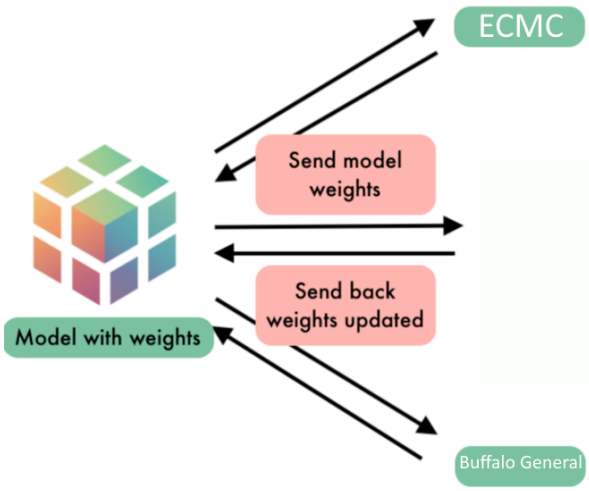

Figure 1: Federated Learning

1	radius (mean of distances from center to points on the perimeter)
2	texture (standard deviation of gray-scale values)
3	perimeter
4	area
5	smoothness (local variation in radius lengths)
6	compactness ($perimeter^2/area - 1.0$)
7	concavity (severity of concave portions of the contour)
8	concave points (number of concave portions of the contour)
9	symmetry
10	fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

3 Plan of Work

1. **Extract features values and Image Ids from the data:** Process the original CSV data files into a Numpy matrix or Pandas Dataframe.

2. **Apply feature scaling technique:** To make sure that all the features are in the same level of magnitude.

3. **Data Partitioning:** Partition your data into training and testing data. Randomly choose 80% of the data for training and the rest for testing.

4. **Create Logisitic Regression Architecture using Pytorch library:** Define the network model with Pytorch library to train logistic regression classifier on the dataset.

5. **Connect to the workers of the hospitals for training:** Create a torch hook with pysyft and create two virutal workers (ECMC and BuffaloGeneral)

6. **Send the data to the workers of the hospitals for training:** Send data to the ECMC and BuffaloGeneral virtual workers

7. **Traing and test the federated logistic regression model:** Create a torch hook with pysyft and create twp virutal workers (ECMC and BuffaloGeneral)

8. **Print results:** Your code should print Accuracy, Precision, Recall and Confusion matrix resulting from federated logistic regression model. Your report should describe the results.

4 Evaluation

1. **Task 1:** 40 points for writing code to use Pytorch library for performing logistic regression on dataset.

2. **Task 2:** 50 points for using implementing federated learning using Pysyft library for logistic regression using Python.

3. **Task 3:** 10 points for describing Accuracy, Precision, Recall and confusion matrix resulting from sklearn classifier in your report.

5 Definitions

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

6 Deliverable

You only need to submit the code. After finishing the project, you may be asked to demonstrate it to the graders, particularly if your results and reasoning in your code are not clear enough.

1. Code The code should be in Python only. Please provide necessary comments in the code. The name of the python notebook should be main.ipynb. Also we request all of you to kindly convert the .ipynb file to .py and include main.py file as well.

   Submit the Python code and report on UBlearns as well as Autolab as a zip file named proj1.zip

   Autolab link: https://autograder.cse.buffalo.edu/

7 Python Notebook Resource: Colaboratory

https://colab.research.google.com/drive/1_U2G7A9FF42B8ZLTBdBeQshch7f08-15