Preserving Privacy for Medical Data

Hemant Koti
University at Buffalo
hemantko@buffalo.edu

Abstract

Deep Learning has emerged as a promising approach for building accurate and robust statistical models from medical data, which is collected in huge volumes by modern healthcare systems. Existing medical data is not fully exploited by Deep Learning techniques primarily because privacy concerns restrict access to this data. However, without access to sufficient data, Deep Learning will be prevented from reaching its full potential from making the transition from research to clinical practice [5].

This research attempts to address these issues by using SyferText (a python library for privacy-preserving Natural Language Processing) that leverages PySyft (a python library for secure and private Deep Learning [6]) to perform Federated Learning and Encrypted Computations (Multi-Party Computation (MPC)) on text data. Additionally, we will also compare the private classifier accuracy with the classifier trained on data without any encryption or privacy.

Index Terms - Federated Learning, Secure Multiparty Computation, Differential Privacy, Natural Language Processing.

1. Problem Statement

Deep Learning in healthcare can be used to streamline several administrative tasks as well as enabling physicians to make smart decisions based on the data. However, healthcare data is highly regulated and private making it inaccessible to perform deep learning tasks. In a medical data scenario, the Deep Learning model should be trained on the data without looking at it.

In this paper, we explore ways (using PySyft [6] and SyferText libraries [1]) to enable secure and private Deep Learning to decouple sensitive data from the process of model training. We implement a classifier that correctly classifies the medical specialty based on the transcription text without looking at the dataset itself.

[1] https://github.com/OpenMined/SyferText

Training	Validation	Recall	Precision	F1
0.87	0.89	0.88	0.98	0.93

Table 1. Benchmark results on MTSamples dataset [1]

We perform the following tasks on the medical dataset.

- Using the PySyft library to create a bigger dataset out of all the client's smaller datasets.
- Using SyferText library to prepare and preprocess the text data on the client's machines without revealing it, and without moving any datasets to your machine.

Existing work [1] on this dataset uses GloVe embeddings [4] and 1D CNN to classify the medical specialty. The benchmark results on this dataset include an accuracy of 89% on the validation dataset as described in Table 1.

In the end of this paper we compare the accuracy achieved for the classifier with and without encryption. We create a classifier that achieves an accuracy score close to the original model trained on the dataset without privacy.

2. Dataset

The original dataset is taken from https://www.mtsamples.com which contains 40 different medical specialties which in turn contain around 5000 transcribed medical reports under all specialties. The raw data has been pre-processed and converted into a CSV format for research purposes on Kaggle [2].
The data provided are only sample reports that are provided by various transcriptionists and users for reference purposes. In a real-world scenario, medical data is extremely hard to find due to privacy regulations.

2.1. Exploratory Data Analysis

In an ideal scenario, we would not get access to the full dataset. Instead, we would be given only the files contain-

[2] https://www.kaggle.com/tboyle10/medicaltranscriptions

Figure 1. Top medical specialities in the dataset after feature reduction

ing the features needed to train our model. However, for the convenience of this paper, we derive some of the important statistics about the dataset.

The two essential features in the dataset needed for training are as follows.

- transcription: The medical transcription text (corpus).
- medical_specialty: The medical specialty tagged to the corpus.

We would also need the stop words file and a vocabulary file which will be added to the NLP pipeline provided by SyferText. In our project, we use the files provided by clinical concepts repository [3] designed for large datasets [2]. We generated the vocab words based on the classes in Systematized Nomenclature of Medicine (SNMI) data [3].

2.2. Dimensionality Reduction

The original dataset consists of around 20 classes with the data distribution being highly skewed towards the surgery class. We, therefore, reduced the number of features thereby making the distinction of whether the transcription text is of the surgery specialty or not. A multi-class classifier is trained using the first four specialties with the most frequencies as shown in Figure 1.

3. Method

In our project, we simulate an environment where each client owns a part of the full dataset and prepare each worker to perform encrypted training on these datasets.

3.1. PySyft

The PySyft library (leveraged by SyferText) can create virtual workers which is a simulation of hospital environments with datasets stored locally at each place without any need for sharing the data to a central server. In our use case, we create a virtual environment (Alice and Bob as virtual workers) using PySyft to train our classifier securely.

The other essential components of the PySyft library include.

- Federated Learning: A type of remote execution wherein models are sent to remote data-holding machines for local training. This eliminates the need to store sensitive training data on a central server.
- Multi-party computation: When a model has multiple owners, multi-party computation allows for individuals to share control of a model without seeing its contents such that no sole owner can use or train it.

3.2. Virtual Environment

A work environment is simulated with three main actors - a company and two clients owning two private datasets (Bob and Alice) but also a crypto provider that will provide the primitives for Secure Multi-Party Computation (SMPC). We simulate two private datasets owned by two clients (Bob and Alice) and distribute the respective datasets privately using a special *send()* function by the SyferText library.

3.3. SyferText NLP Pipeline

The SyferText NLP pipeline consists of three blocks - a tokenizer, a stop words tagger, and a vocabulary tagger. Additionally, a language object provided by the SyferText has to be loaded which contains all the natural language processing objects and functions.
[4].

The object created for NLP tasks by SyferText will remove the stop words tokens added to the pipeline and the rest of the tokens will be filtered out if not marked as words from the vocabulary file. This pipeline will allow to process the text more efficiently and also assigns weights to tokens that have a high correlation with the output classes.

3.4. Encrypted Classifier and Hyper-parameters

The hyper-parameters used for training and validation are as follows.

- Embedding Dimension: The dimension of the embedding vector for the training dataset.
- Batch Size: 128
- Learning Rate: 0.001
- Output classes: 4

We create a classifier with the following configuration as shown below.

[3]https://github.com/kavgan/clinical-concepts

[4]https://blog.openmined.org/sentiment-analysis-syfertext/

Training Accuracy	Validation Accuracy	Average Loss
78%	75%	27

Table 2. Training and Validation results on encrypted classifier

Classifier (
(fc1) Linear(in_features=300, out_features=128, bias=True)
(fc2) Linear(in_features=128, out_features=64, bias=True)
(fc3) Linear(in_features=64, out_features=32, bias=True)
(fc4) Linear(in_features=32, out_features=16, bias=True)
(fc5) Linear(in_features=16, out_features=2, bias=True)
)

The network is a multi-class classifier that outputs any of the labels - 'Surgery', Medical Records', 'Internal Medicine', 'Other' depending on the transcription text.

3.5. Encrypted Deep Learning

We create a hook for PyTorch to link it with PySyft to extend the functionalities of PyTorch so that we can use it for PySyft methods. We load the data, define our network structure, and share it across the virtual workers using a simple *share()* function in PySyft.
Sending the tensors to virtual workers is as simple as calling the *send(worker)* method on the tensor. Any kind of remote operations can be performed on these tensors, in our case, we perform model training using forward and backward pass on these tensors. After the operation is performed we can call a simple *get()* function to return the tensor securely.

4. Results

We observed that the model achieved around 76% (Figure 2) validation accuracy while the loss (Figure 3) was reduced. This can be due to the choice of our optimizer (SGD and MSE) since better optimizers are not yet available for this framework.
The results (Table 2) did not improve either by increasing epoch quantity or by reducing the learning rate or batch size hyper-parameters. We can assume that a different, deeper network architecture (RNN or LSTM) could potentially increase the model accuracy.

5. Conclusion

Our goal to achieve an accuracy closer to the benchmark results is partially achieved. Ideally, there is always a trade-off between privacy and accuracy, especially when it comes to sensitive information the data and model privacy must be ensured at all costs.

Using SyferText and PySyft we demonstrate a secure use case of NLP text classification on the medical dataset.

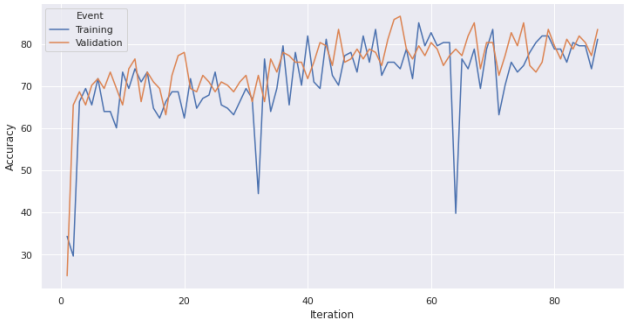

Figure 2. Accuracy graph on training and validation dataset with private classifier using PySyft and SyferText

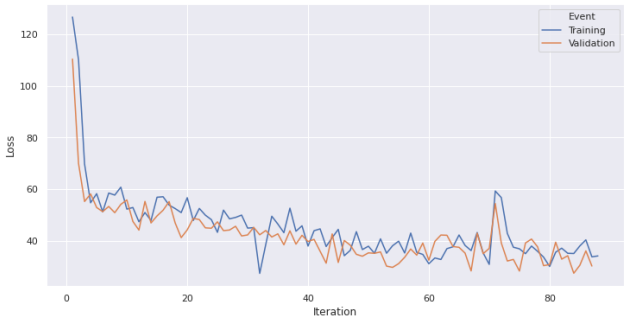

Figure 3. Loss graph on training and validation dataset with private classifier using PySyft and SyferText

These libraries are still in development, therefore, cannot be used to encrypt data in production scenarios yet.

References

[1] Marchawala Alizar, Patel Preetkumar, Paresh Thaker, Gunjal Hardik, Nagrecha Abhishek, and Mohammed Sabah. Text summarization and classification of clinical discharge summaries using deep learning, 2020. 1

[2] Kavita Ganesan, Shane Lloyd, and Vikren Sarkar. Discovering related clinical concepts using large amounts of clinical notes. *Biomed Eng Comput Biol*, 7(Suppl 2):27–33, Sep 2016. becb-suppl.2-2016-027[PII]. 2

[3] Musen MA, Noy NF, Shah NH, Whetzel PL, Chute CG, Story MA, Smith B, and NCBO team. The national center for biomedical ontology. j am med inform assoc., 2012. 2

[4] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation, 2014. 1

[5] Rieke, N. Hancox, J. Li, W., and et al. The future of digital health with federated learning. *PAMIDigit. Med.*, 3(119), 2020. 1

[6] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning, 2018. 1