Transcript:

Differential privacy is a framework for measuring the privacy guarantees provided by an algorithm. We can design differentially private machine learning algorithms to train on sensitive data. It provides provable guarantees of privacy (point to first image), reducing the risk of exposing sensitive training data through the learned classifier. Intuitively, for any two adjacent training set that are only differed by one data record, the learned classifiers should not be too different. In the context of deep learning, differential private stochastic gradient descent, i.e. DPSGD, is the state-of-art algorithm to train such a privacy-preserving neural network.

Nowadays, the DPSGD algorithm is in urgent need of combining with big compute and big data technology. On the one hand, due to the features of the DPSGD algorithm, such as limiting the gradient size in each step of parameter update, its convergence time will be 10 to 100 times longer than that of the original SGD algorithm. Without the use of computational processing, the training process of DPSGD will be extremely time-consuming. On the other hand, the datasets processed by DPSGD will be up to thousands of petabyte. For example, in some high-tech companies, they need to process billions of private user data every day, and it is impossible to process data without big data processing. Therefore, we aim to improve the performance of DPSGD algorithm by using big compute and big data technology, so that DPSGD can be applied to more complex application scenario and work with huge amount of sensitive data.

As an optimization method, differential private SGD is developed from original SGD. In the process of parameter updating, there are two more steps for DPSGD in each iteration: gradient clipping and noise addition. These two steps reduce the effect of one single anomaly so that the results of two similar datasets will not be too different. We can deploy parallel computing methods in this parameter updating part.
Our data comes from American Community Survey Public Use Microdata Sample (PUMS) files. It includes useful but somehow sensitive census information such as Sex, Married, College degree. Our objective is to train a deep learning model to predict the unemployment rate based on other demographic information using DPSGD and HPC and HTC tools so that we can both protect privacy and obtain a satisfiable runtime of the algorithm.

Proposed infrastructure diagram:
First, to load and preprocess the data, we can employ **MapReduce on AWS cluster** to first preprocess the large amount of data. The mapper processes the input from s3 bucket line by line, and the reducer combines the sorted intermediate output. **HDF5 file format** could be used to load the data without blowing up the memory.

After processing the data, we propose two potential ways to train the network at scale:
One option is we could use an AWS GPU cluster to distribute the workload across multiple GPUs by using **large minibatch technique** to speed up the RNN training on Pytorch and using its **MPI interface with NCCL backend** for multi-GPU communication between nodes, as shown in the diagram.
Another training option is we could use apache spark with amazon sagamaker, which could be used for training and deploying a model using custom PyTorch code. With our own customized ML algorithm built into SageMaker compatible Docker containers, we can use SageMaker Spark to train and infer on dataframes at Spark scale.