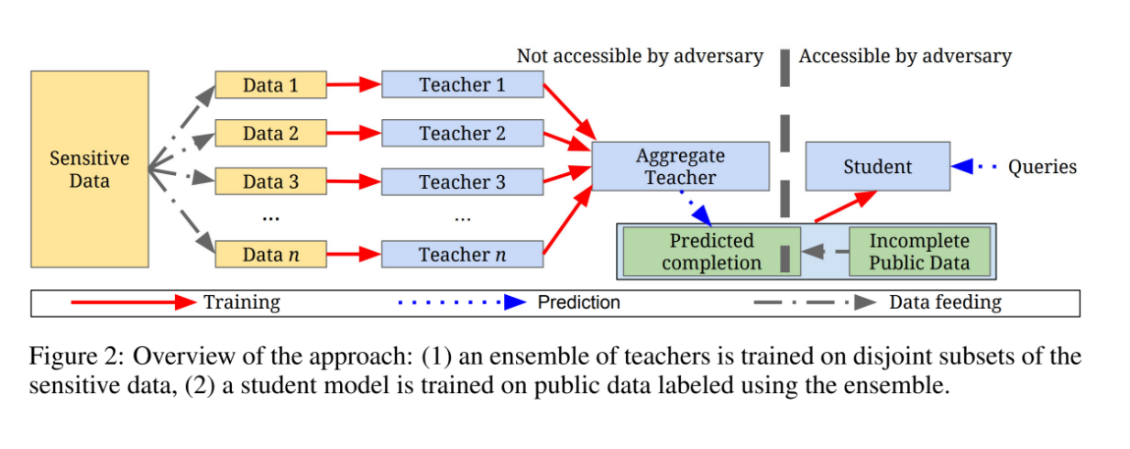
**Overview of DPSGD**

Stochastic Gradient Descent is one of the most prominent deep learning optimization algorithms. The idea is that you adjust the weights and biases of your model during the backpropogation step of training according to the gradient (or derivative) of the loss function. When the loss function has a steep gradient, there is a chance for the model to make more drastic changes to the weights and biases and have much better performance. When the loss function has a shallow gradient, weights and biases of the model will be fine-tuned and slightly adjusted in the direction of the gradient. Stepwise, SGD computes the loss between the predicted y and the true y, then uses that loss function to compute the gradient/derivative, then multiplies these gradients by the learning rate to update the parameters of the model.

In order to make this algorithm differentially private, small modifications need to be made. Differential privacy is the idea that no unauthorized user should be able to see or make changes to the training data. A model with differential privacy guarantees that it will not be affected by any single training sample and that it will not expose sensitive training data. SGD becomes differentially private (DPSGD) when you clip the gradient to ensure that a single datapoint cannot greatly influence the model parameters and when you compare the updates made on the parameters with the datapoint in question versus the updates made on the parameters without the datapoint’s inclusion in the training set. To do this, prior to updating the parameters in the backpropogation step, DPSGD clips the gradient and applies random noise, then uses these factors to update the parameters. Different gradient clips and noise factors will result in different privacy guarantees. Generally, more noise results in better privacy (often, but not necessarily, at the expense of lower utility). This is confirmed below with the section titled “DPSGD Experiments.”

**Overview of PATE**

The PATE framework achieves private learning by coordinating the activity of several ML models, where only the student model and the student model’s predictions are available to the end-user. Dr. Alqahtani used a very helpful image in class:



The idea here is that the Teacher models are trained on the complete sensitive dataset and then each model makes its own decision. The decisions are tallied and the one with the most choices gets to be the final label. Then, the Student machine is only fed the incomplete nonsensitive data and trained using the labels created by the Teacher Ensemble. The only model used is the Student model, and the Student model’s training data is not sensitive, so there is no reason to worry about its privacy. The end result is PATE ensuring privacy because the final decision does not reveal any information about any individual training sample.

**Discussion of DPSGD Experiment Results:**

**DPSGD Experiments:**

**(learning rate** .25 constant, **epochs** at 15 constant, **microbatches** at 256 constant, **batch size** at 256 constant, **delta\*** at 1e-5)

CHANGING THE NOISE MULTIPLIER

1. **Accuracy: .954**
   1. **Noise Multiplier:** 1.3
   2. **L2 Norm Clip:** 1.5
   3. the current epsilon$$$ is: 1.19
2. **Accuracy: .955**
   1. **Noise Multiplier:** 1.1
   2. **L2 Norm Clip:** 1.5
   3. the current epsilon$$$ is: 1.59
3. **Accuracy: .957**
   1. **Noise Multiplier:** 0.7
   2. **L2 Norm Clip:** 1.5
   3. the current epsilon$$$ is: 4.76
4. **Accuracy: .967**
   1. **Noise Multiplier:** .2
   2. **L2 Norm Clip:** 1.5
   3. The current epsilon$$$ is: 722.11

CHANGING THE L2 NORM CLIP

1. **Accuracy: .945**
   1. **Noise Multiplier:** 1.3
   2. **L2 Norm Clip:** .7
   3. the current epsilon$$$ is: 1.19
2. **Accuracy: .942**
   1. **Noise Multiplier:** 1.3
   2. **L2 Norm Clip:** 1.9
   3. the current epsilon$$$ is: 1.19
3. **Accuracy: .932**
   1. **Noise Multiplier:** 1.3
   2. **L2 Norm Clip:** 2.9
   3. the current epsilon$$$ is: 1.19

\*Delta bounds the probability of our privacy guarantee not holding. A rule of thumb is to set it to be less than the inverse of the training data size (i.e., the population size). Here, we set it to 10^-5 because MNIST has 60000 training points.

$$$Epsilon measures the strength of our privacy guarantee. In the case of differentially private machine learning, it gives a bound on how much the probability of a particular model output can vary by including (or removing) a single training example. We usually want it to be a small constant. However, this is only an upper bound, and a large value of epsilon could still mean good practical privacy.