HW 6

Maria Rubio Navarro

11/18/2024

What is the difference between gradient descent and *stochastic* gradient descent as discussed in class? (*You need not give full details of each algorithm. Instead you can describe what each does and provide the update step for each. Make sure that in providing the update step for each algorithm you emphasize what is different and why.*)

Gradient Descent computes the gradient of the loss function using the entire dataset at each step, which ensures stable and accurate updates but can be slow when working with large datasets. In contrast, Stochastic Gradient Descent (SGD) updates the parameters using the gradient calculated from a single data point at a time, making it much faster but less stable since it doesn't consider the full dataset. The key difference lies in the data used for each update: Gradient Descent uses all data points, providing stability and precision, while SGD uses one data point per update, prioritizing speed but introducing noise in the process.

Consider the `FedAve` algorithm. In its most compact form we said the update step is $\omega_{t+1} = \omega_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla F_k(\omega_t)$. However, we also emphasized a more intuitive, yet equivalent, formulation given by $\omega_{t+1}^k = \omega_t - \eta \nabla F_k(\omega_t); w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$.

Prove that these two formulations are equivalent.
(*Hint: show that if you place ω_{t+1}^k from the first equation (of the second formulation) into the second equation (of the second formulation), this second formulation will reduce to exactly the first formulation.*) —

1. $\omega_{t+1}^k = \omega_t - \eta \cdot \nabla F_k(\omega_t)$
2. $\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \cdot \omega_{t+1}^k$

Substitute ω_{t+1}^k from the first equation into the second equation:

$$\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \cdot (\omega_t - \eta \cdot \nabla F_k(\omega_t))$$

Expand the sum by distributing $\frac{n_k}{n}$:

$$\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \cdot \omega_t - \eta \cdot \sum_{k=1}^{K} \frac{n_k}{n} \cdot \nabla F_k(\omega_t)$$

Since ω_t is constant in the sum, it can be factored out:

$$\omega_{t+1} = \omega_t \cdot \sum_{k=1}^{K} \frac{n_k}{n} - \eta \cdot \sum_{k=1}^{K} \frac{n_k}{n} \cdot \nabla F_k(\omega_t)$$

Since $\sum_{k=1}^{K} \frac{n_k}{n} = 1$ (because the proportions sum to 1):

$$\omega_{t+1} = \omega_t - \eta \cdot \sum_{k=1}^{K} \frac{n_k}{n} \cdot \nabla F_k(\omega_t)$$

This result is exactly the same as the first formulation:

$$\omega_{t+1} = \omega_t - \eta \cdot \sum_{k=1}^{K} \frac{n_k}{n} \cdot \nabla F_k(\omega_t)$$

Both formulations are equivalent. The second simply divides the calculation into two steps: first, it computes local updates (ω_{t+1}^k) and then combines these updates into a global one (ω_{t+1}).

Now give a brief explanation as to why the second formulation is more intuitive. That is, you should be able to explain broadly what this update is doing.

The second formulation is more intuitive because it separates the process into local and global steps. Each client computes its own local update ($\omega_{t+1}^k = \omega_t - \eta \nabla F_k(\omega_t)$) based on its data, ensuring privacy. Then, the server aggregates these updates using a weighted average ($\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k$), reflecting each client's contribution proportionally. This mirrors how federated learning operates in practice.

Prove that randomized-response differential privacy is ϵ-differentially private.

Differential random response privacy complies with ϵ privacy because it adds some randomness when the data is collected. This makes it so that the probability of getting a result does not change much, at most by a factor of e^{ϵ}, if a person's data is added or removed. Because of this randomness, it is harder to tell what each individual contributes, which helps protect their privacy.

Define the harm principle. Then, discuss whether the harm principle is *currently* applicable to machine learning models. (*Hint: recall our discussions in the moral philosophy primer as to what grounds agency. You should in effect be arguing whether ML models have achieved agency enough to limit the autonomy of the users of said algorithms.*)

John Stuart Mill's harm principle states that individual freedom should only be limited to prevent harm to others. In the case of machine learning, this applies if models cause direct harm, such as discrimination or privacy violations. Although models do not have agency, they can influence autonomy by impacting important decisions, such as hiring or lending. The harm principle is therefore relevant, but the responsibility lies with the humans who use and regulate these systems.