HW 6

Lalida Kungval

3/10/2024

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 update rule: $\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t)$.The update rule uses the gradient of the loss function computed over the entire dataset to update the parameters. This leads to a more accurate estimate of the true gradient direction but at the cost of higher computational expense.

Stochastic Gradient Descent update rule: $\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t; x_i, y_i)$.This updates the gradient of the loss function computed on a single randomly selected data point or a small batch of data points. The process is faster but introduces more noise into the gradient estimate, potentially leading to fluctuations in the learning 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. definition of ω_{t+1}^k in the more intuitive formulation:

$$\omega_{t+1}^k = \omega_t - \eta \nabla F_k(\omega_t)$$

2. plugged into the aggregation step of the same formulation:

$$\omega_{t+1} = \sum_{k=1}^{K} \left(\frac{n_k}{n} \omega_{t+1}^k \right)$$

substituting the expression for ω_{t+1}^k:

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

expanded:

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

3. separate the terms involving ω_t and $\nabla F_k(\omega_t)$:

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

4. simplified:

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

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 of the FedAve algorithm is considered more intuitive because it clearly separates the local update step from the global aggregation step, which mirrors the actual process in Federated Learning.

Explain how the harm principle places a constraint on personal autonomy. 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.*)

The harm principle asserts that individual freedom should be limited to prevent harm to others, thus constraining personal autonomy. Although machine learning models lack agency as they do not have intentions or moral understanding, their outputs can still cause harm, such as bias or privacy violations. Therefore, the harm principle applies to the use of ML models, suggesting that the autonomy of developers and users should be limited to prevent potential harm. This principle advocates for ethical guidelines and regulatory measures to ensure ML practices do not negatively impact individuals or society.