HW 6

Nimalan Subramanian

4/10/2024

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 (GD) and Stochastic Gradient Descent (SGD) are both optimization algorithms that can be used to find the minimum of a function in the context of minimizing the cost function for machine learning. GD finds the gradient of the cost function with respect to the parameters of the whole training set in order to determine the direction in which the parameters need to be adjusted to minimize cost. The update step for this is $\theta = \theta - \eta * \nabla_\theta J(\theta)$. θ represents the parameters, η is the learning rate, and $\nabla_\theta J(\theta)$ is the gradient of cost function J with respect to parameters θ found over the whole training set. On the other hand, SGD updates the parameters using the gradient of the cost function from a single randomly selected example from the training set. As such, SGD requires less computation compared to GD. The update of SGD is $\theta = \theta - \eta * \nabla_\theta J(\theta; x^i, y^i)$. In this update step, θ and η are the same as the update step in GD, but $\nabla_\theta J(\theta; x^i, y^i)$ is the gradient of the cost function with respect to parameters θ found for a single data point (x^i, y^i).

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.*)

If we substitute the the first equation into the second equation, we get $\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} (\omega_t - \eta \nabla F_k(\omega_t))$. From this, we can distribute $\frac{n_k}{n}$ inside the parenthesis, then factor out ω_t. $\sum_{k=1}^{K} \frac{n_k}{n}$ can be simplified to 1, due to it being the sum of all fractions in the data set, resulting in the simplified $\omega_{t+1} = \omega_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla F_k(\omega_t)$, which is the compact form of the FedAve algorithm.

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 update step is more intuitive as it breaks down the algorithm into two steps that models the natural flow of federated learning. In the first step, each k client updates the

local model parameters ω_t independently. After the local models have been updated by all clients, the central server aggregates the updates to make a new global model.

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 places a constraint on personal autonomy by stating that the freedom to act ends when it begins to harm others. The principle is mainly applied to agents that can make decisions and understand the consequences of their actions. While agents are found in intent, understanding, and the ability to distinguish from right and wrong, ML models do not possess this level of consciousness and cannot be considered moral agents in the same way as humans. Additionally, they can cause harm (as seen in the case of biased algorithms that can perpetuate inequalities). In such cases, the harm principle would advocate for constraints on the development of models to prevent harm to others. As such, the principle is applicable to the human actions that relate to the creation and use of ML models, but not the models themselves. The models are just extensions of human intent, so the autonomy of those developing and using ML models may be limited to prevent harm to others, which aligns with the harm principle.