HW 6

Serena Zhang

4/7/2024

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 aims to minimize empirical risk associated with estimating parameters by describing the direction of steepest descent. The update rule is given by $\theta_i + 1 = \theta_i - \alpha \nabla f(\theta_i, X, Y)$, where θ is the current parameters and (X,Y) are all data.

Stochastic gradient descent computes the gradient for a random subset of data, rather than all data. The update rule is given by $\theta_i + 1 = \theta_i - \alpha \nabla f(\theta_i, X_i, Y_i)$, where (Xi, Yi) is a random tuple of data.

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.*)

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

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 since it first describes the local update step. Next, all of the local steps are averaged to get the global step.

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 states that autonomy (the capacity of an agent to act in accordance with their own free will) should extend exactly up until its exercise results in the harm of another moral agent. Put simply, individual actions should only be limited if they hurt others. Agency is the capacity of an entity to act in accordance with their moral principles. Machine learning models have not necessarily achieved agency in that they are designed by human beings, and while they can "act" and make choices without explicit instruction in a sense, they cannot distinguish between a moral "right" and "wrong". Therefore, only humans have the discretion to determine whether use of a machine learning model is morally right or wrong, and whether it will cause harm. For example, if people have observed that the COMPAS algorithm causes harm, then under the harm principle lawmakers should limit the autonomy of the users of the COMPAS algorithm.