HW 6

Student Name

1/21/2024

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 is moving in the steepest decrease of a given functions, however this can have issues with getting stuck in local minima. Stochastic gradient descent uses a random subset of data to calculate the steepest descent which introduces noise that can aid in not getting stuck in local minima.

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.*)

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

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, in the first part shows that for each partition of the data, k, the loss formula is calculated to fine the steepest descent, and then in the second part all k loss formulas are averaged to find the true updated 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 suggests individual agency should only be constrained to prevent harm from others. In regards to machine learning, we have discussed how machine learning can cause harms due to biases, discrimination, and privacy concerns. This cause of harm can be significant and should be considered when discussing the agency of ML models. Although, I believe, we have not reached a point where we can consider specific ML models or LLMs to have individual agency, the creators of such models can and should be scrutinized under the lens of the harm principle and such creators have the the capcity to limit the autonomy of user's of their algorithms by biasing the model to guide and influence the outcome in a way that does not align with the intent of users, thereby limiting their agency.