HW 6

IZ Raad

4/10/2024

1

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 methods have an update step of the form $\theta_{i+1} = \theta_i - \alpha\nabla f(\theta_i, x, y)$. They iteratively compute the gradient of their loss functions, utilizing the full data. They then "slide" along this gradient in search of a local minimum, updating parameters with each iteration. In the case of a non-convex loss function, this idea of "sliding" along a gradient may leave us at a local minimum rather than a global minimum.

In the case of stochastic gradient descent, our update step is of the form $\theta_{i+1} = \theta_i - \alpha\nabla f(\theta_i, x_i, y_i)$. Note that our x and y parameters are now indexed, indicating that they differ with each step update. Indeed, the stochastic gradient descent method calls upon a random subset of our data with each iteration, rather than using the entire dataset to compute gradients. This lends this method to being less likely to settle at local minima. Moreover, a smaller dataset used with each iteration makes this method less computationally expensive.

2

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.*)

$$\begin{aligned}
\omega_{t+1} &= \omega_t - \eta\textstyle\sum_{k=1}^{K}\frac{n_k}{n}\nabla F_k(\omega_t) \\
&= \omega_t - \textstyle\sum_{k=1}^{K}\frac{n_k}{n}\eta\nabla F_k(\omega_t) \\
&= \omega_t - \textstyle\sum_{k=1}^{K}\frac{n_k}{n}(\omega_t - \omega_{t+1}^k) \text{ by substituting in our definition of } \omega_{t+1}^k. \\
&= \omega_t - \textstyle\sum_{k=1}^{K}\frac{n_k}{n}\omega_t - \frac{n_k}{n}\omega_{t+1}^k \\
&= \omega_t - \textstyle\sum_{k=1}^{K}\frac{n_k}{n}\omega_t + \sum_{k=1}^{K}\frac{n_k}{n}\omega_{t+1}^k \\
&= \omega_t - \omega_t\textstyle\sum_{k=1}^{K}\frac{n_k}{n} + \sum_{k=1}^{K}\frac{n_k}{n}\omega_{t+1}^k \\
&= \omega_t - \omega_t + \textstyle\sum_{k=1}^{K}\frac{n_k}{n}\omega_{t+1}^k \text{ because the sum of all partitions is equal to the whole.} \\
&= \textstyle\sum_{k=1}^{K}\frac{n_k}{n}\omega_{t+1}^k
\end{aligned}$$

3

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.

In our second formulation, we say that our data is split up into K partitions. Then, in each of these partitions, we locally progress one step of gradient descent. We then take the weighted average of each of these local steps of gradient descent (adjusted by partition size). This weighted average becomes our global step update.

4

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.*)

In accordance with the harm principle, one may act freely (or, posses personal autonomy) until these actions have the potential to cause harm unto oneself of others. This is obviously a constraint on personal autonomy, as our domain of permissible actions is restricted to those that do not cause harm. Considering ML, the harm principle may not be applied ... yet. Of course, I am no judge of the future and cannot speak on the agency of future algorithms. As it stands, ML serves as a tool to carry out tasks specified by its user. They do *not* have the agency to assess the harm principle and limit user actions. Instead, it is the duty of the user to act in accordance with the harm principle. In the same way that you cannot hold a shovel accountable for the head that it is swung at, we cannot hold ML accountable for the people that it harms. It is up to the user to be informed about the potential harm a tool may cause before utilizing it. In the case of COMPAS, the judge is making the choice to weigh COMPAS's prediction when considering parole. Therefore, inequities caused by the use of COMPAS are a result of the judge's choice.