# HW 6

Alyson Longworth

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.*)

As discussed in class, a gradient represents the direction of steepest ascent of a function f by displaying a vector of partial derivatives of f in respect to each variable. If we are trying to minimize f, we would use the opposite of this gradient. Similarly, gradient descent finds the gradient of the loss function in order to minimize $\theta$ and improve the accuracy of the model using the formula $\theta_{i+1} = \theta_i - \alpha * \nabla F(\theta_i, x, y)$. Here, $\theta_i$ represents the current parameters, x,y represent the whole data set, and $\alpha * \nabla F$ represents the step size of the parameter $\theta$. Consequently, if you set $\alpha$ to be too large, you risk overshooting the minimum and receiving an inaccurate result. Stochastic gradient descent, on the other hand, finds the gradient of a randomly selected data point or group of data points rather than the gradient of the entire data set, like gradient descent does. Here, we would use the formula $\theta_{i+1} = \theta_i - \alpha * \nabla F(\theta_i, x_i, y_i)$ where $(x_i, y_i)$ represents the randomly selected data points rather than (x,y) used in gradient descent to represent the entire data set.

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 $\omega_{t+1}^k$ 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.*)

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

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.

First formula: $\omega_{t+1} = \omega_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla F_k(\omega_t)$ Second formula: $\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} * (\omega_t - \eta \nabla F_k(\omega_t))$. $\omega_{t+1}^k$ is found by the gradient of the loss function at each k scaled by $\eta$, which tells us the direction of steepest ascent of the loss function, or where we should focus on minimizing the loss. $\omega_{t+1}$ is then found by the sum of the

average of $\omega_{t+1}^k$ and weighted by the proportion of data samples at each k. Including this weighted average results in a more accurate result because it allows the model to more accurately include the contributions of each parameter. The second formulation is therefore more intuitive because it allows the model to weigh the parameters in a way that more accurately aligns with their occurrence.

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 that people should be able to act in the way they please as long as their actions do not cause harm to others. This principle places a constraint on personal autonomy by limiting a person's actions to only actions that do not cause harm to others. The harm principle is currently applicable to machine learning models, especially regarding informed consent and data privacy. It can be argued, using the harm principle, that as long as a model does not directly harm anyone or is created for a person's well-being, it could use as much openly available data or surveillance as possible, with or without consent. It can, on the other hand, also be argued that the use of this data or surveillance could potentially harm someone in the long run from either misuse or the potential for a data leak and therefore requires informed consent when using this sort of data.