HW 6

Jacob Thoma

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 uses an entire dataset to find the rate of steepest descent/ascent in a given loss function with a given model. The goal is to change sets of model parameters until a critical point (minimum or maximum) is reached for the loss function. After parameters are initialized, the gradient is calculated for the loss function at those initialized parameters. Then, the algorithm sets parameters in the direction of steepest descent/ascent based on the gradient and the optimization goals of the user. The rate at which this algorithm moves in a desired direction is controlled by the learning rate, alpha. As such, the update step looks like this: Theta i+1 = Theta I – alpha gradient(theta I, parameters of entire data) until a critical point is reach.

Stochastic gradient descent follows the exact same procedure as gradient descent, but instead of using the entire dataset, sampling is used. This keeps the idea the same, but changes the update step to: Theta i+1 = Theta I – alpha gradient(theta I, parameters of randomly selected data). The end product of Stochasitic gradient descent is a noisier but less computationally expensive optimization of the loss function.

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.*)

First step - Placing ω_{t+1}^k into second formulation equation 2: $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} (\omega_t - \eta \nabla F_k(\omega_t))$

Second step- Splitting sums: $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \omega_t - \sum_{k=1}^{k} \frac{n_k}{n} \eta \nabla F_k(\omega_t)$

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

Proof complete.

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 because its two equations allow a distinction to be made between local updates and global updates. The first equation depicts updating of individual local model weights, while the second equation derives a weighted average of local model weights together to inform the global model weights.

Splitting up the formula in this way also gives a more intuitive path to understanding stochastic gradient descent, where the local models draw similarity to the sampling of an entire dataset in stochastic gradient descent.

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 limiting an individual moral agent's ability to do anything that harms another moral agent. A saying that simply puts, "My right to swing my fist ends at your face" covers the idea of it. The harm principle can be applied uniformly to every moral agent without much debate. It is much more tricky (and subjective) for ML models that have debatable status as moral agents.

The most defendable position when considering degree of moral agency is to consider the sentience of the object in question. My first example will be comparing a human to an ant. A human feels much in a lifetime, with complex emotions and lots of nerve endings. An ant feels less pain in a much shorter life, and the ants feelings of pleasure and pain are based off of basic pheromones, not complex thought. In this case, a human demands a higher level of moral agency because they feel more than the ant. When GPT is compared to the same ant, the ant demands a higher level of moral agency for the same reasons in the previous example - it can think and feel at a higher level than GPT. Because GPT cannot inherently feel anything (in terms of pleasure or pain) or even think, it has no moral agency. Therefore, the harm principle is not applicable to GPT as a moral agent.