Homework 5 Written Assignment

10-605/10-805: Machine Learning with Large Datasets

Due Thursday, November November 19th at 1:30:00 PM Eastern Time

Instructions: Submit your solutions via Gradescope, *with your solution to each subproblem on a separate page*, i.e., following the template below. Note that Homework 5 consists of two parts: this written assignment, and a programming assignment. The written part is worth **50%** of your total HW5 grade. The programming part makes up the remaining 50%.

Submitting via Gradescope: When submitting on Gradescope, you must assign pages to each question correctly (it prompts you to do this after submitting your work). This significantly streamlines the grading process for the course staff. Failure to do this may result in a score of 0 for any questions that you didn't correctly assign pages to. It is also your responsibility to make sure that your scan/submission is legible so that we can grade it.

Collaboration: While you may talk with others about the homework, you must each turn in your own solution and it should be your own work. Please list the names of any collaborators below.

Collaborators:

As we consider more hyperparameters, the space we search over to find good hyperparameter configurations grows very quickly. In the first two questions, we will explore just how quickly this space grows.

1 Curse of Dimensionality (12 pts)

We define the ϵ-cover of a search space S as a subset $E \subset S$:

$$E = \{x \in \mathbb{R}^n \mid \forall y \in S, \exists x \; s.t. \; \|y - x\|_\infty \leq \epsilon\}$$

In words, every coordinate of every point in our search space S is within ϵ of a point in our set E. The ϵ-covering number is the size of the smallest such set that provides an ϵ-cover of S.

$$N(\epsilon, S) = \min_E \{|E| : E \text{ is an } \epsilon\text{-cover of } S\}$$

Let's work through a concrete example to make the concept clearer: let's say we want to tune the learning rate for gradient descent on a logistic regression model by considering learning rates in the range of $[1, 2]$. We want to have $\epsilon = .05$-coverage of our search space $S = [1, 2]$. Then $E = \{1.05, 1.15, 1.25, 1.35, 1.45, 1.55, 1.65, 1.75, 1.85, 1.95\}$. We see that any point in $[1, 2]$ is within 0.05 of a point in E. So E provides ϵ-coverage of S. The ϵ-covering number is 10. The questions below will ask you to generalize this idea.

(a) *[4 points]* Find the size of a set needed to provide ϵ-coverage (ϵ-covering number) of $S = [0, 1] \times [0, 2]$, the Cartesian product of two intervals. Note that we want this to hold for a generic ϵ.

(b) *[8 points]* Find the ϵ-covering number of $S = [a_1, b_1] \times \ldots \times [a_d, b_d]$, the Cartesian product of d arbitrary closed intervals. On what order does this grow with respect to the dimension d and interval lengths (i.e., with the volume of this space)?

2 Grid versus Random Search (12 pts)

There are two basic strategies commonly employed in hyperparameter search: grid search and random search. Grid search is defined as choosing an independent set of values to try for each hyperparameter and the configurations are the Cartesian product of these sets. Random search chooses a random value for each hyperparameter at each configuration. Imagine we are in the (extreme) case where we have d hyperparameters all in the range $[0, 1]$, but only *one* (h_1) of them has any impact on the model, while the rest have *no* effect on model performance.

(a) *[4 points]* Imagine that we consider q hyperparameter configurations, which are enough to provide ϵ-coverage for grid search. How many unique models will we train? What is this as a percent of the total number of configurations?

(b) *[4 points]* Alternatively, if we consider q random configurations as part of random search, then what percent of the configurations will cause a change to the model? Why?

(c) *[4 points]* Now we change our point of view; instead of desiring ϵ-coverage, we have a fixed budget of $B \ll |S|$ configurations to try. Which search strategy should we employ? Why?

3 Uniform Allocation versus Early Stopping (26 pts)

Consider two configurations, and imagine we plot validation error as a function of number of iterations (or resources). This validation error is noisy since it's based on partially trained models. However, let's assume we know that these models eventually converge to some fixed validation errors ν_1 and ν_2, which implies that there exists some envelope of uncertainty bounding this noise. For simplicity, imagine we have the same envelope of uncertainty for both configurations, γ_k, which is a function of the k, the number of iterations run. Figure 1 below will aid you in visualizing what we are looking for.

$$\lim_{k\to\infty} \ell_{i,k} = \nu_i \qquad |\ell_{i,k} - \nu_i| \leq \gamma_k \quad \forall i \in [n], k \geq 1$$

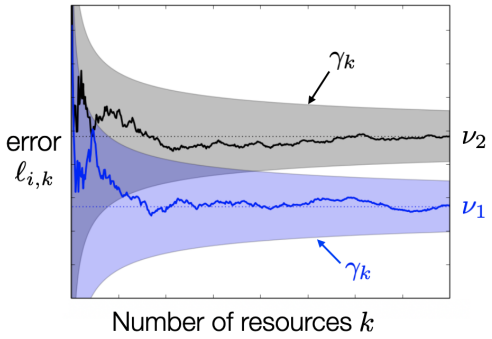

Figure 1: A visualization of the envelopes of uncertainty around the true convergent validation errors at each training iteration.

(a) *[5 points]* What are the possible values for configuration 1 validation error after $k = 1$ iteration, expressed in terms of ν_1 and γ_1?

(b) *[5 points]* Now imagine we were given access to this envelope of uncertainty. At what point could we, with certainty, know that configuration 1 is better than configuration 2? Can you express this mathematically as a function of the given γ_k, ν_2 and ν_1? Show your work.

Next, imagine the we have N configurations we're considering (with their quality ordered by configuration number, i.e., configuration 1 is the best, configuration 2 is second best, etc.), all of which are governed by the same envelope of uncertainty. Our goal is to find an efficient allocation of resources across these N configurations, i.e., to choose how many iterations to run each configuration for such that we identity the best configuration with the fewest number of total iterations.

(c) *[8 points]* Consider a uniform allocation strategy, as is done with vanilla random search, in which we are required to allocate the same number of iterations to all configurations. What is our required total budget in order to, with certainty, be able to identify that configuration 1 is the best? Please characterize this total budget as a function of ν_i and γ.

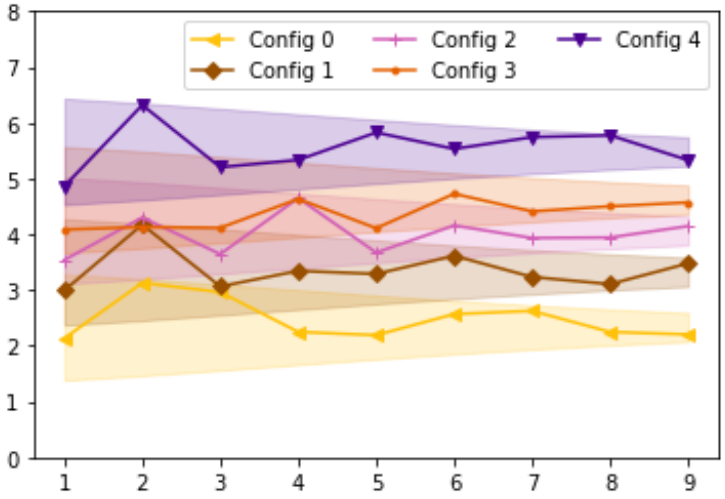

Figure 2: Validation loss as a function of iteration number for five configurations. Note that the shaded regions are the associated regions of uncertainty.

(d) *[8 points]* Now imagine we are in the situation of Figure 2. We run iterations one at a time for each configuration we haven't ruled out as being best. At each iteration, we adaptively choose whether to continue running that configuration or not. How many configuration-iterations will we have run? This will be lower-bounded by 5 (running them all for one iteration and stopping) and 45 (running all configurations for all 9 iterations of the experiment). How does this adaptive allocation approach compare to the uniform allocation strategy previously discussed? Please fill out the table below.

Note: The simple answer is '0', we already know $\nu_1, \ldots, \nu_5$. We are *not* looking for this answer. Instead, we are considering the artificial exercise of only using ν_i to center our γ_k, and to make decisions based on these centered γ_k.

Note: We will accept some minor variations to our expected final answers given that slight visual ambiguity of the shaded regions of uncertainty in Figure 2.

Configuration	**Iterations with Uniform Allocation**	**Iterations with Adaptive Allocation**
0		
1		
2		
3		
4		
sum		