Hyperparameter tuning, Dataset splitting, and the Gradient Descent Method

Total points 20/25

Email *

shinoda.c.i@gmail.com

✗ Which description is wrong about machine learning hyperparameters? * 0/1

- (●) Hyperparameters are parameters that set values before the algorithm begins learning ✗
- ( ) Most machine learning algorithms have hyperparameters
- ( ) Hyperparameters cannot be modified
- ( ) The value of the hyperparameter is not learned by the algorithm itself

Correct answer

- (●) Hyperparameters cannot be modified

✓ Hyperparameters are not automatically adjusted during model training. Sometimes they need to be manually optimized. *1/1

- (●) True ✓
- ( ) False

✓ Which of the following is not a search strategy to tune hyperparameters? * 1/1

- ○ Grid search
- ◉ Random gradient descent ✓
- ○ Random search
- ○ Bayesian model-based optimization

✗ Hyperparameter tuning is the process of optimizing the parameters of a machine learning model to improve performance. *0/1

- ◉ True ✗
- ○ False

Correct answer

- ◉ False

✓ Grid search differs from Random search in that: * 1/1

- ○ Grid search requires less computational resources.
- ○ Grid search is more likely to find the optimal hyperparameters.
- ◉ Grid search explores the hyperparameter space in a systematic way. ✓

✓ We looked at a process of using a test set and a training set to drive iterations of model development. On each iteration, we'd train on the training data and evaluate on the test data, using the evaluation results on test data to guide choices of and changes to various model hyperparameters like learning rate and features. Is there anything wrong with this approach? (select only one answer) *1/1

- ○ This is computationally inefficient. We should just pick a default set of hyperparameters and live with them to save resources.
- ◉ Doing many rounds of this procedure might cause us to implicitly fit to the peculiarities of our specific test set. ✓
- ○ Totally fine, we're training on training data and evaluating on separate, held-out test data

Feedback

The more often we evaluate on a given test set, the more we are at risk for implicitly overfitting to that one test set. Correct answer.

✓ Finding the balance between Bias and Variance can be achieved by an iterative process, training the model multiple times with different combinations of features, hyperparameters and with different training datasets. *1/1

- ◉ True ✓
- ○ False

✓ Why is it not a good idea to use the test set to evaluate the models when performing hyperparameter search? *1/1

- ○ Because the test set will be discarded after the training
- ◉ Because the model can become biased towards the test set if it is used repeatedly for evaluation ✓
- ○ Because the test set can become too small to provide reliable estimates of model performance.

✗ Why is it important to use a validation set in machine learning? * 0/1

- ○ To prevent overfitting of the model on the training set.
- ○ To ensure that the model is generalizing well to new data
- ◉ To select the best model from a set of candidate models. ✗

Correct answer

- ◉ To ensure that the model is generalizing well to new data

✓ Which of the following best describes a solution to the negative effect of an iterative machine learning process on the test set? *1/1

- ○ Increasing the size of the test set.
- ○ Using a separate training set for hyperparameter tuning
- ◉ Using cross-validation instead of a fixed test set ✓

✓ K Folding cross-validation refers to dividing the test data set into K Sub-data sets *1/1

- ○ True
- ◉ False ✓

Feedback

General feedback: Refers to dividing the training data set into K sub-data sets

✓ Gradient descent can be used to find the optimal values of the parameters in a machine learning model *1/1

- ◉ True ✓
- ○ False

✓ For a linear regression model, start with random values for each coefficient. The sum of the squared errors is calculated for each pair of true labels and model output values. A learning rate is used as a scale factor and the model parameters are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible. This technique is called _____? *1/1

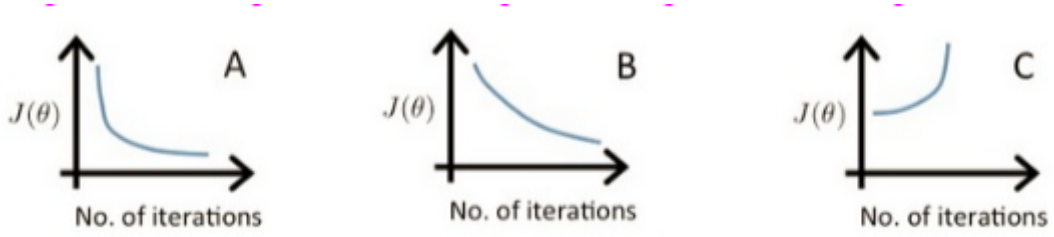
- ◉ Gradient Descent ✓
- ○ Ordinary Least Squares
- ○ Homoscedasticity
- ○ Regularization

✓ Indicate the true options: * 1/1

- [x] A cost function, also known as a loss function, is a mathematical function that measures the difference between the predicted output of a model and the actual output. ✓
- [x] The goal of a machine learning model is to minimize the cost function by adjusting the model parameters through optimization ✓
- [x] A quadratic cost function is a type of cost function used in machine learning and optimization problems. ✓

✗ The following graphs show how the training error evolves when a machine linear model is being trained with a gradient descent algorithm using different learning rates. *0/1

Suppose l1, l2 and l3 are the three learning rates for A, B, and C, respectively. Which of the following is true about l1,l2 and l3?



- l2 < l1 < l3
- l1 > l2 > l3 ✗
- l1 = l2 = l3
- None of these

Correct answer

- l2 < l1 < l3

Feedback

If the learning rate is too high, steps may be too large, causing the process to find new parameters that produce higher values for the cost function, i.e., the objective function value will increase. If the learning rate is too low, steps will be small and the objective function will decrease slowly.

✓ Which parameter determines the size of the improvement step to take on each iteration of Gradient Descent? *1/1

- (●) learning rate ✓
- ( ) epoch
- ( ) batch size
- ( ) regularization parameter

✓ What is the purpose of the gradient descent algorithm in machine learning? *1/1

- ( ) To maximize the cost function.
- (●) To minimize the cost function. ✓
- ( ) To calculate the derivative of the cost function.

✓ What does the gradient descent algorithm update at each iteration? * 1/1

- ( ) The learning rate.
- ( ) The cost function.
- (●) The model parameters. ✓

Feedback

c) the algorithm updates the model parameters in the opposite direction of the gradient to move towards the optimal solution. This process is repeated iteratively until the algorithm converges to a minimum point of the cost function

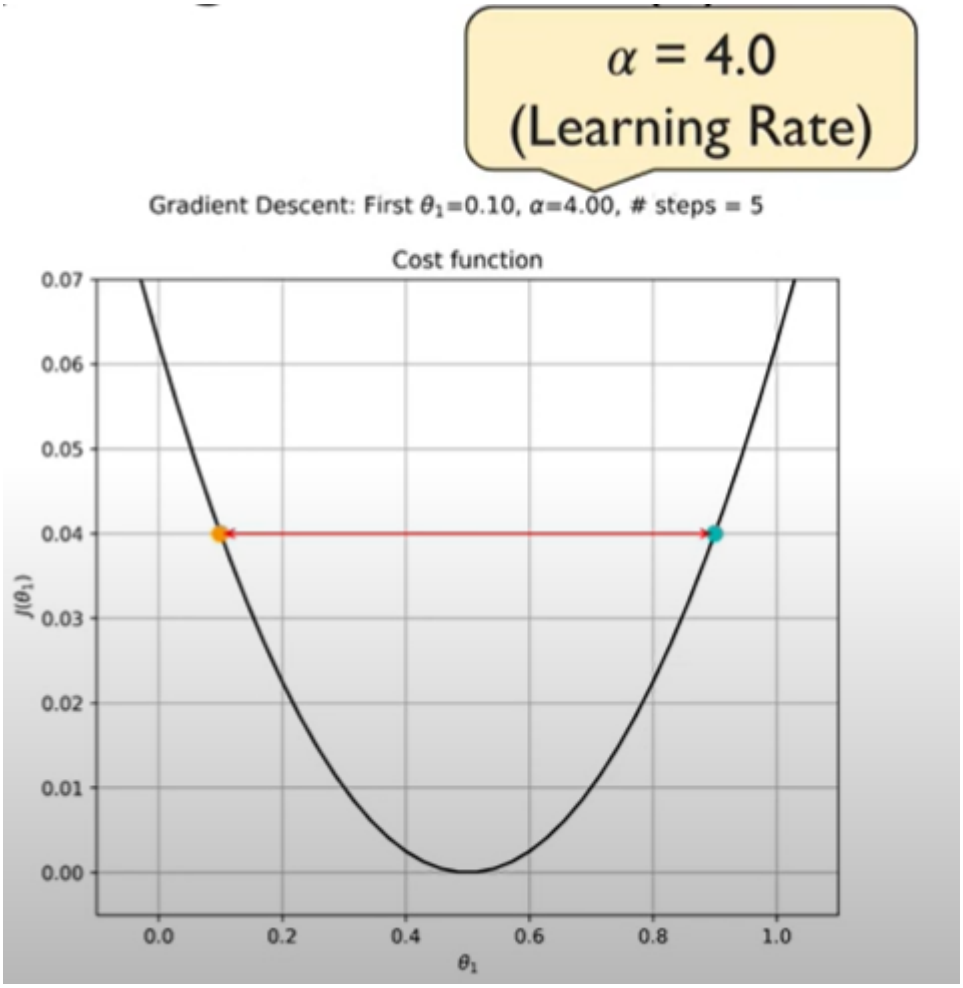
✓ Which of the following is a correct interpretation of the learning rate in gradient descent? *1/1

- ○ The learning rate controls the size of the cost function
- ○ The learning rate is updated at each iteration to optimize the cost function
- ◉ The learning rate controls the size of the updates to the parameters at each iteration ✓

Feedback

Indeed, the learning rate is a hyperparameter that determines the step size of the algorithm during each iteration.

✓ Based on the following image shown in class, which statement describes what might happen: *1/1



- [x] The gradient descent algorithm oscillating back and forth without improving the cost function ✓
- [x] It might never converge ✓
- [ ] It always converge to the minimum

✗ How does increasing the training set size affect the cost of computing the gradient in machine learning? *0/1

- ○ The cost increases linearly with the training set size.
- ◉ The cost increases exponentially with the training set size. ✗
- ○ The cost decreases linearly with the training set size.
- ○ The cost remains constant regardless of the training set size.

Correct answer

- ◉ The cost increases linearly with the training set size.

✓ When performing gradient descent on a large data set, which of the following batch sizes will likely be more efficient? *1/1

- ◉ A small batch or even a batch of one example (SGD). ✓
- ○ The full batch.

Feedback

Amazingly enough, performing gradient descent on a small batch or even a batch of one example is usually more efficient than the full batch. After all, finding the gradient of one example is far cheaper than finding the gradient of millions of examples. To ensure a good representative sample, the algorithm scoops up another random small batch (or batch of one) on every iteration.

✓ What is the most important difference between batch gradient descent, mini-batch gradient descent, and stochastic gradient descent? *1/1

- Gradient size
- Gradient direction
- Learning rate
- ◉ Number of samples used on each step ✓

✓ In the gradient descent algorithm, which of the following algorithms is the most confusing algorithm for the trajectory on the loss function surface? *1/1

- ◉ SGD ✓
- BGD
- MBGD

✓ Which of the following is an advantage of stochastic gradient descent (SGD) over batch gradient descent (BGD)? *1/1

- SGD is less sensitive to noisy data.
- SGD converges faster than BGD.
- ◉ SGD requires less computational resources than BGD. ✓

Feedback

Indeed, SGD only needs to process a single example at each iteration, whereas BGD needs to process the entire training set at each iteration.

Google Forms