COL341 : Machine Learning
Assignment 1.2

Gaurav Jain (2019CS10349)

September 26, 2021

1 Hyperparameter Tuning

Hyperparameter tuning is one of the most essential components in the developing a good ML model. Hyperparameters of a multinomial logistic regression are tuned in this section.

1.1 Batch Gradient Descent v/s Mini-batch Gradient Descent

The selection of a right algorithm for gradient descent is the first step towards hyperparameter tuning.
We compare the two gradient descent algorithms- batch gradient descent and mini-batch gradient descent. Same value is used for common hyperparameters like learning rate and number of iterations(epochs in mini-batch case). The following cross-entropy loss v/s runtime graph is observed:

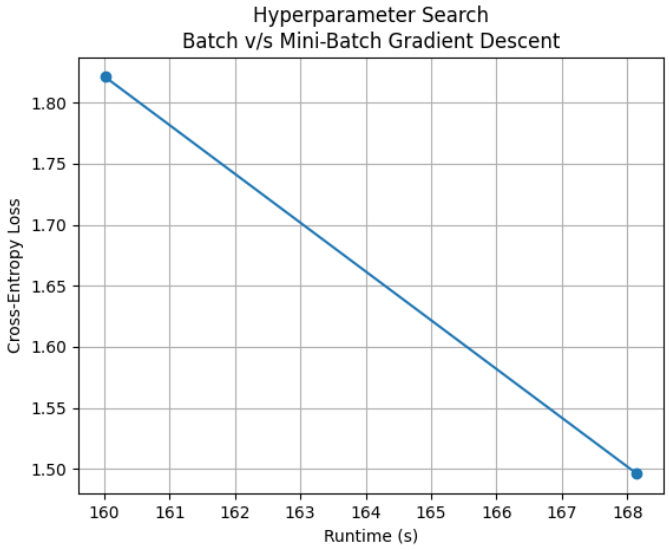

Figure 1: Batch v/s Mini-batch Gradient Descent

As it is clearly visible in the graph, mini-batch gradient descent gives less value of loss compared to batch gradient descent with small increase in runtime. Thus, mini-batch gradient descent algorithm is selected.

1.2 Learning Rate Algorithm

There are three algorithms for learning rate- fixed learning rate, adaptive learning rate and $\alpha - \beta$ backtracking search. The three algorithms are compared with same parameters like batch size, initial learning rate and number of epochs. The following cross-entropy loss v/s runtime graph is observed:

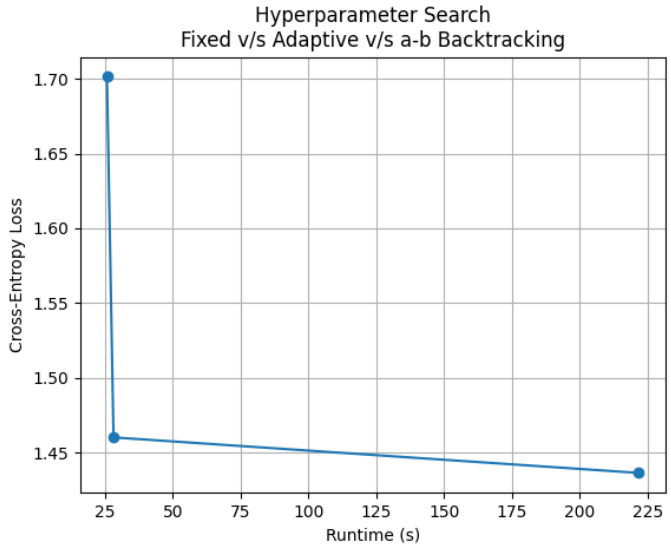

Figure 2: Fixed v/s Adaptive v/s $\alpha - \beta$ backtracking search

Fixed learning rate algorithm gives very high cross-entropy loss and therefore, it is rejected. There is a very small difference between loss values for adaptive learning rate and $\alpha - \beta$ backtracking search but there is a large increase in runtime for $\alpha - \beta$ backtracking search. Thus, adaptive learning rate method is selected.

1.3 Initial Learning Rate

In adaptive learning rate method, initial learning rate is a tunable hyperparameter and different values for the initial learning rate are compared. The following cross-entropy loss v/s runtime graph is observed:

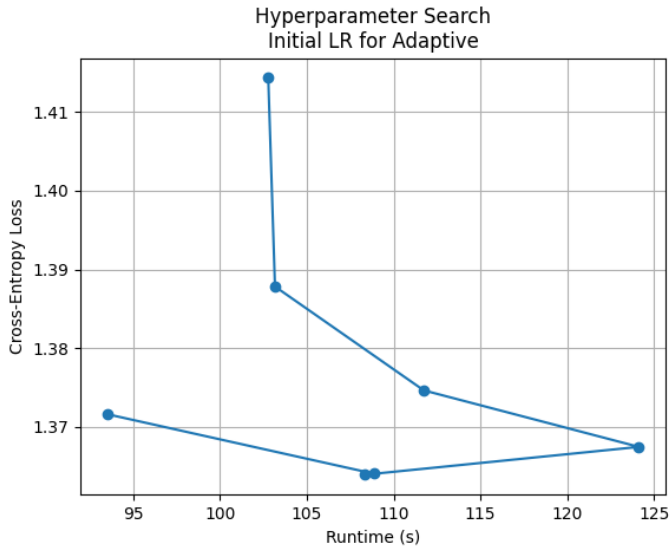

Figure 3: Initial Learning Rate

There is a very large decrease in the cross-entropy loss initially and runtime also increases but the method adapts to higher intial learning rate quickly and thus runtime decreases. The minimum value of cross-entropy loss is observed at $lr = 10$ and $lr = 12$. Thus, 10 is selected as the intial learning.

1.4 Batch Size

Different values of batch size are tried and the following cross-entropy loss v/s runtime is observed:

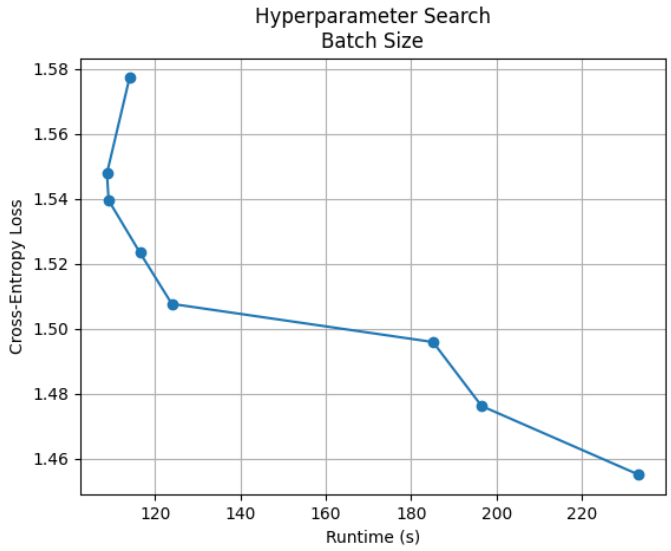

Figure 4: Batch Size

For selecting a good value for batch size, we have to trade-off between loss and runtime. For two values, there is a large decrease in runtime. These values are 100 and 300. For size=300, the loss is quite high and thus, size=100 is selected for the model.

1.5 Number of Iterations(Epochs)

Theoretically, loss should decrease as we increase number of iterations but the loss will ultimately tend to some value. Thus, to select a good number of iterations, we check for the point where the graph stabilizes. The following cross-entropy loss v/s runtime is observed:

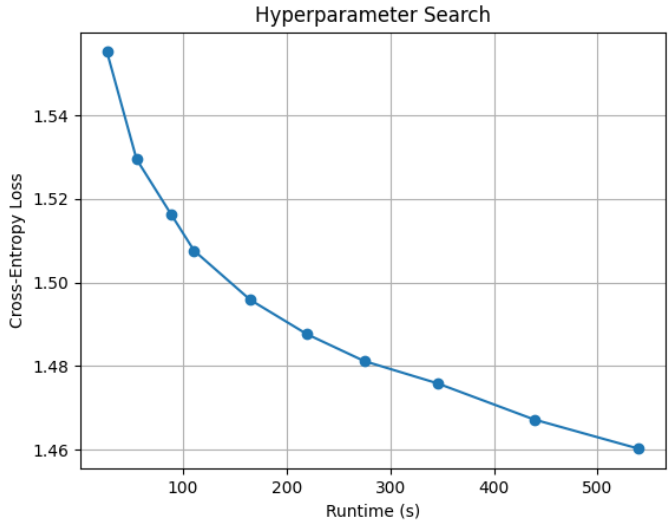

Figure 5: Number of Iterations

If we select 400 or 600 epochs with adaptive learning rate method, then the loss function almost stabilizes. Thus, to run the model within a fixed time, 600 epochs are performed in the model.

2 Feature Creation and Selection

Feature creation and selection are important part of Machine Learning. The accuracy of prediction on unseen data can be improved by adding non-linear features to the logistic regression model. In this report, the first section contains the features created using various techniques and the second section contains how these created features are selected.

2.1 Feature Creation

The following techniques are used for adding new features:

2.1.1 Dropping Non-Essential Features

The features present in the original don't have the same relation to the total cost. Thus to improve the accuracy one should drop non-essential features. Features like birth-weight are removed.

2.1.2 Adding Combinations of Features

After removing non-essential features, the accuracy of logistic regression on unseen data can be improved by adding combinations of essential features with very high co-variance to the length of stay.
APR DRG Code,*Facility Name* and *CSS Procedure Code*, *Facility Name* are two of the most important feature cobinations. So to capture their importance, these two feature combinations are encoded together in a single column in the input matrix. There are other combinations as well which can improve the accuracy. However there is a heavy computation involved in this technique so other combinations are not added.

2.1.3 Target Encoding

Target Encoding is extremely helpful in features where there are only qualitative classes and no precise relation can be found. Target Encoding is applied using the mean of total cost for a particular class of a feature.
Target Encoding is applied on the combination of the features added previously. This gives a good variance to the length of stay.

2.1.4 One Hot Encoding

One hot encoding is necessary for logistic regression models where the features represent qualitative values. Thus, OHE is done on all features except total cost.

2.2 Feature Selection

After applying the above described techniques of creating features, a subset of features are selected from the set of created features.
Selection of features is done using *SelectKBest* function of *sklearn.featureselection*. To test created features, Cross-validation accuracy and loss are used to select such features.
We don't select features from the set of features because the we have very small set of features and every single feature is almost independent of another. So, all features are selected for the model.