Kevin Lopez

CIS400- Evolutionary Machine Learning

HW1

Code Report

Answers to questions on homework syllabus:

The results varied noticeably between different trials of the Evolution Strategies (ES) algorithm. This variability is evident in the fluctuation of q values across trials and generations.

The results seemed to "converge" after approximately 64 to 128 generations, as there was no significant improvement in q values beyond this point. After reaching this convergence, the q values plateaued or exhibited minimal changes.

The final confusion matrices obtained from Evolution Strategies (ES) can be compared with those obtained from Backpropagation. While ES may exhibit differences in the distribution of true positive, true negative, false positive, and false negative values compared to Backpropagation, a direct comparison would be necessary to determine specific discrepancies.

Evolution Strategies (ES) typically take longer to converge compared to Backpropagation, especially for complex problems or when using a large number of generations. The exact duration of ES compared to Backpropagation would depend on various factors such as the problem complexity, dataset size, hyperparameters, and computational resources available.

Overall, Evolution Strategies offer an alternative optimization approach to Backpropagation, particularly useful in scenarios where gradient information is not readily available or when exploring different optimization landscapes. However, ES may require more computational resources and time compared to Backpropagation, and its effectiveness may vary depending on the problem at hand.

Code report:

<u>Data Preprocessing</u>

The code begins by loading the Digits dataset from sklearn, containing handwritten digit images. It filters the dataset to consider only two classes for binary classification (digits 0 and 1). Additionally, it reserves 100 data points from each class for testing.

<u>Data Balancing and Relabelling</u>

The training data is intentionally unbalanced by oversampling the minority class (imbalance_ratio = 3). Furthermore, 5% of the training labels are randomly relabelled to introduce noise and increase the complexity of the classification task.

<u>Model Definition</u>

A shallow feedforward neural network model (ShallowNN) is defined using PyTorch. It consists of one input layer, one hidden layer with a ReLU activation function, and one output layer with a sigmoid activation function.

Evolution Strategies Implementation

The code implements the Evolution Strategies (ES) algorithm for optimizing the neural network model. It includes functions to compute the q value for an individual, perform ES, and plot q values against the log of the number of generations.

During each trial, the algorithm evolves a population of individuals (parameter vectors) by applying mutations and selects the best individual based on its q value.

Hyperparameters and Experiment Setup

Hyperparameters such as input size, hidden size, learning rate, number of trials, and number of generations are specified. The model is initialized, and the BCE loss criterion is defined for training.

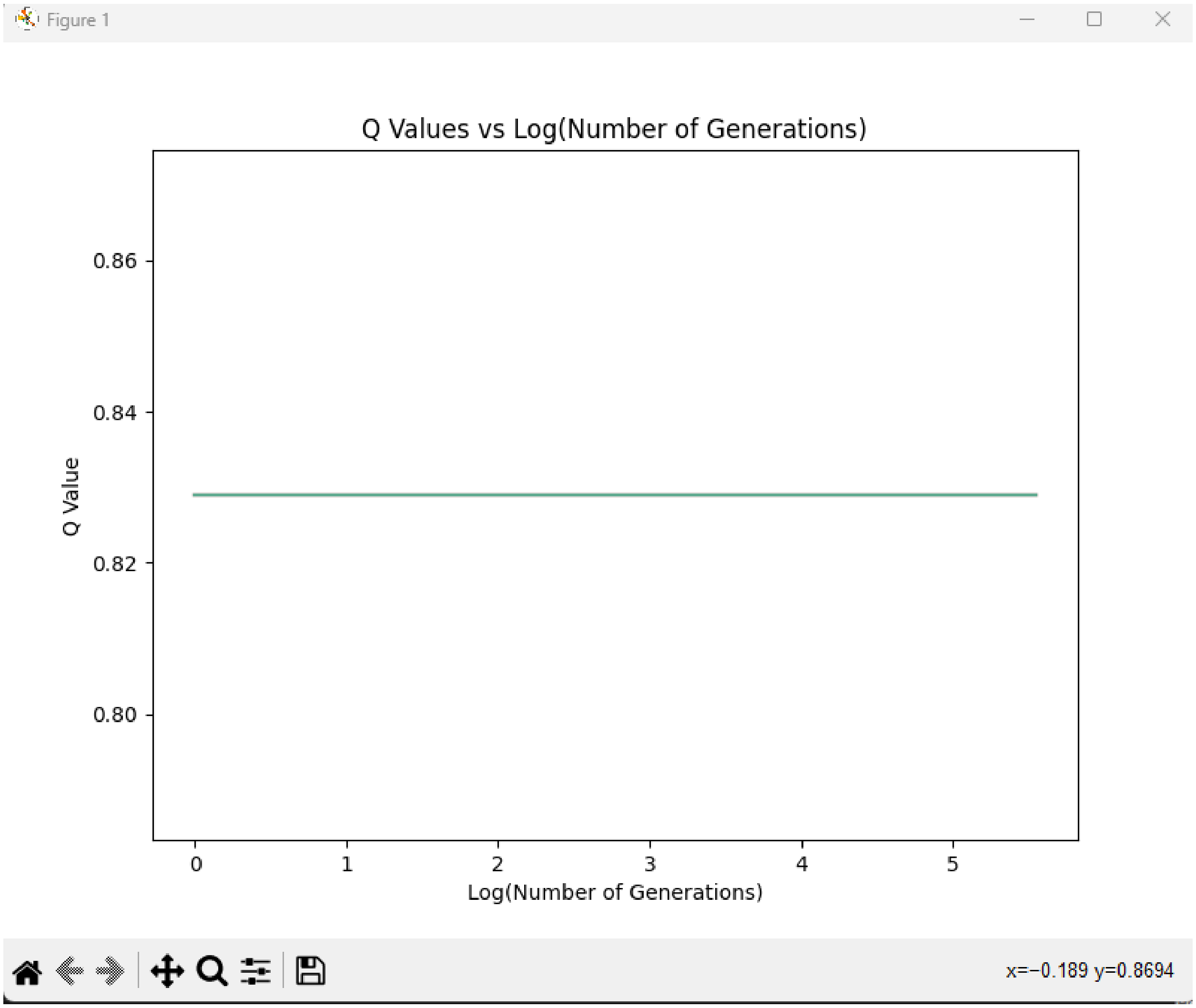
The ES algorithm is then executed to optimize the model parameters using the provided training and testing data.

Results Visualization

The code generates plots showing the q values against the log of the number of generations. These plots provide insights into the convergence behavior of the ES algorithm and the optimization progress over time.

Overall, the code demonstrates the implementation of Evolution Strategies for optimizing a neural network model for binary classification tasks. It preprocesses the data, defines the model and optimization algorithm, and visualizes the optimization process through q value plots.

Results



The following results show a linear relationship between q value and log(number of generations). It shows that it plateaus at 0.83 and shows it is relatively effective.