Wesley Smith

CS7641 – Assignment 2 – Fall 2018

# Part 1 - Neural Network Training

## Dataset / Network Recap

I chose the Adult dataset for a comparative analysis. In Assignment 1, the neural network that performed best had two hidden layers consisting of 100 nodes each. The dataset, after one-hot encoding, had 104 features. Combining this input size along with a single output node results in a network with 20604 links, each having their own weight. This would come to present a difficult challenge for randomized optimization algorithms.

When trained using backpropagation on Assignment 1, this network achieved a train, test, and validation accuracy[[1]](#footnote-1) of 86%, 78%, and 77%, respectively.

## Randomized Search

For learning weight vectors with randomized search algorithms, 30% of the data was split into a holdout test set. The remainder was split into 80% training data and 20% validation data.

The ABAGAIL library was used for implementations of all randomized search algorithms.

Training iterations were 5000. The fitness function was mean square error on the training data.

## Randomized Hill Climbing

ABAGAIL implements RHC by initializing all weights to uniform random samples in the interval [-0.5, 0.5]. At each step, a neighbor is generated by picking a single weight at random and resampling along the same interval. If the neighbor produces a better fitness score, it is selected as the optimum. The algorithm proceeds along this path until training iterations have completed. As such, the fitness function curve for RHC increases monotonically, as shown in Fig. X.

The shortcomings of this approach for a search space this large are apparent. Over 5000 iterations, even if the optimizer was lucky enough to pick 5000 unique weights to modify, it would still have only chosen less than 25% of the available weights to modify. Given that these weights are continuous and randomly distributed, it is clear that the RHC algorithm could not possibly test a significant portion of the available search space. Therefore, the result is highly dependent on the initial random sample taken and will almost certainly only be able to find a local optimum.

Figure X. Learning curves for Randomized Hill Climbing

# Part 2 – Optimization Problems

## Continuous Peaks – GA

The Continuous Peaks (CP) problem[[2]](#footnote-2) is an extension on the classic Four Peaks problem[[3]](#footnote-3), in which the fitness function depends on the size of contiguous strings of 0s and 1s. For CP, an input vector X of N bits as well as an integer parameter T are supplied. The fitness function is calculated as the sum of the length of the largest string of consecutive ones and that of the largest string of consecutive zeros. An additional reward of N is added if both Max(0) and Max(1) are greater than T. This creates the presence of a global optimum that is reachable by algorithms such as GA but hard to find for algorithms such as RHC.

* Most interesting T: 15
* Easy to show random hill climbing gets stuck in local optima
* SA is often capable of finding the reward
* GA performs very well
* MIMIC does ok – seems to have good variance, but when it fails, it is pretty bad

## Traveling Salesman – GA?

Initial impressions:

* RHC and SA are comparably not great
* GA has high highs and low lows
* MIMIC does very poorly again
  + Very low lows, low highs, large variance

## FlipFlop – MIMIC

Initial impressions:

* MIMIC finds near optimum performance very quickly
* GA has a lot of variance, relatively low numbers
* SA actually finds high performance at max iterations
* RHC is RHC

1. Balanced accuracy was used because the dataset contained approx. 75% negative samples [↑](#footnote-ref-1)
2. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.8826&rep=rep1&type=pdf [↑](#footnote-ref-2)
3. https://pdfs.semanticscholar.org/cd4f/e89d8dd6060e2957041f90fc699a30058d01.pdf [↑](#footnote-ref-3)