Donald Dunagan and Andrew Stout

18 October 2018

Evolutionary Computation Project Proposal

While there exist many machine learning algorithms capable of performing regression analysis, that is, the calculation of output based on input, such as linear regression, support vector machines (SVM), and regression trees (RT), this project is interested in regression analysis with artificial neural networks (ANN). Specifically, multilayer perceptrons (MLP), (feed forward artificial neural networks with one hidden layer). The data set to be used is The Parkinson's Telemonitoring Dataset, a dataset from the University of California Irvine Machine Learning Repository. The dataset was derived from speech samples taken from patients with Parkinson's Disease (PD) with each observation containing 16 measurements of disordered speech, 2 measurements of PD severity, and information about the patient. The goal is to predict PD severity based on measures of disordered speech found in voice samples.

The calculation of target output with a neural network takes the form of a series of matrix and vector operations involving feature values, node weights, bias nodes, and activation functions. The architecture of the MLPs for this project will have 17 nodes in the input layer (one for each feature in the dataset plus one bias node), 8 nodes in the hidden layer (a relatively arbitrary number less than the number of input nodes but greater than the number of output nodes), and one output node. The bias node will have a value of 1, the activation function applied at each of the hidden layer nodes will be a rectifying linear unit (ReLU), and the activation function at the output node will be linear. With this in place, all that is left for the neural network regression calculation is weight values.

A common method for calculating the weights of a neural network is the use of backpropogation. This project, however, will optimize weights through various dialects of evolutionary computing: genetic algorithms, evolution strategies, and evolutionary programming. With 16 feature values and 8 hidden nodes, the network will require 136 weights. Thus, for the purposes of evolutionary computation, a network will be represented by a real valued string of length 136. The evolutionary algorithms will be implemented in Python with the help of the DEAP (Distributed Evolutionary Algorithms in Python) package. The data will be shuffled and split into training and testing sets. The fitness function for the evolutionary algorithms will be the mean absolute error (MAE) on the training set and it is to be minimized. The best solution for each of the dialects will then be tested on the testing set and their MAEs will be compared to other algorithms that have been applied to this dataset in the literature such as RT, SVM, and other ANNs.