CSCI 447 — Machine Learning

Project #4

Assigned: November 2, 2020
Project Due: November 25, 2020

This assignment requires you to apply three different evolutionary approaches to train a feedforward neural network and then compare the results of evolutionary training to your implementation of backpropagation. For this particular assignment, you will focus only on training the weights of a pre-defined neural network, thus the only difference between the various implementations for a given problem will be the training algorithm.

Starting with an untrained neural network, you will train the weights of the network using each of the following algorithms (separately):

- Backpropagation
- Genetic algorithm with real-valued chromosomes
- Differential evolution
- Particle swarm optimization

For each of these algorithms, you will conduct appropriate studies to determine the best parameter settings (e.g., learning rate, momentum, inertia, etc.) prior to running the head-to-head comparisons. Be sure to apply good experimental design in your process (e.g., tuning sets, cross-validation, statistical hypothesis testing).

For this assignment, you will use three classification datasets and three regression data sets that you will download from the UCI Machine Learning Repository, namely:

1. Breast Cancer [Classification]

   `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29`

   This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

2. Glass [Classification]

   `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`

   The study of classification of types of glass was motivated by criminological investigation.

3. Soybean (small) [Classification]

   `https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29`

   A small subset of the original soybean database.

4. Abalone [Regression]

   `https://archive.ics.uci.edu/ml/datasets/Abalone`

   Predicting the age of abalone from physical measurements.

5. Computer Hardware [Regression]

   `https://archive.ics.uci.edu/ml/datasets/Computer+Hardware`

   The estimated relative performance values were estimated by the authors using a linear regression method. The gives you a chance to see how well you can replicate the results with these two models.

6. Forest Fires [Regression]

   `https://archive.ics.uci.edu/ml/datasets/Forest+Fires`

   This is a difficult regression task, where the aim is to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data .

When using these data sets, be careful of some issues.

1. Some of the data sets have missing attribute values, which is usually indicated by "?". When this occurs in low numbers, you may simply edit the corresponding data items out of the data sets. For more occurrences, you should do some kind of "data imputation" where, basically, you generate a value of some kind. A naïve approach is to impute the missing value with a random number or the attribute's mean (or median). A better approach is to sample according to the conditional probability of the values occurring, given the underlying class for that example. The choice of strategy is yours, but be sure to document your choice.

2. For networks with multiple outputs, you should use what is called a "multi-net." This is where you train a single network with multiple outputs, one for each outcome you need to predict. This is distinct from training separate networks for each output, which is effectively what you have done with prior linear models.

3. The attributes should not require any special handling with either model. It is highly recommended that you normalize numerical attributes first to be in the range -1 to $+1$ or by using z-score normalization (i.e., $z = (x - \mu)/\sigma$) and apply the inputs directly.

4. With respect to the network architectures, you should use the "best" architectures from Project 3. This means you do not need to worry about tuning the number of hidden nodes per layer since you already did that.

Your assignment consists of the following steps:

1. Download the six (6) data sets from the UCI Machine Learning repository. You can use the links above or download them from Brightspace. You can also find this repository at `http://archive.ics.uci.edu/ml/`.

2. Pre-process the data to ensure you are working with complete examples (i.e., no missing attribute values).

3. Implement feedforward neural network training using a tunable genetic algorithm. Use your GA to train the 0, 1, and 2-hidden layer networks you produced from Project 3.

4. Implement feedforward neural network training using a tunable differential evolution procedure. Use DE to train the 0, 1, and 2-hidden layer networks you produced from Project 3.

5. Implement feedforward neural network training using a tunable particle swarm optimization algorithm. Use PSO to train the 0, 1, and 2-hidden layer networks you produced from Project 3.

6. Develop a hypothesis focusing on convergence rate and final performance of each of the chosen algorithms.

7. Compare the results of all three of your population-based algorithms and backpropagation (which should have been implemented previously), after tuning, on the data sets given above.

8. Write a very brief paper summarizing the results of your experiments. Your paper is required to be at least 5 pages and no more than 10 pages using the JMLR format You can find templates for this format at `http://www.jmlr.org/format/format.html`. The format is also available within Overleaf. Make sure you explain the experimental setup, the tuning process, and the final parameters used for each algorithm.

9. Your paper should contain the following elements:

   (a) Title and author name

   (b) Problem statement, including hypothesis

   (c) Description of your experimental approach

(d) Presentation of the results of your experiments

(e) A discussion of the behavior of your algorithms, combined with any conclusions you can draw

(f) Summary

(g) References (Only required if you use a resource other than the course content.)

10. Create a video that is no longer than 5 minutes long demonstrating the functioning of your code. For the video, the following constitute minimal requirements that must be satisfied:
    - The video is to be no longer than 5 minutes long.
    - The video should be provided in mp4 format. Alternatively, it can be uploaded to a streaming service such as YouTube with a link provided.
    - Fast forwarding is permitted through long computational cycles. Fast forwarding is *not permitted* whenever there is a voice-over or when results are being presented.
    - Be sure to provide verbal commentary or explanation on all of the elements you are demonstrating.
    - Provide sample outputs from one test set showing performance on your networks. Show results for the two hidden layer cases only but for each of the learning methods.
    - Demonstrate each of the main operations for the GA: selection, crossover, and mutation.
    - Demonstrate each of the main operations for the DE: selection, crossover, and mutation.
    - Demonstrate each of the main operations for the PSO: pbest calculation, gbest calculation, velocity update, and position update.
    - Show the average performance over the ten folds for one of the data sets for each of the networks trained with each of the algorithms.

11. Submit your fully documented code with the outputs from running your programs, your video, and your paper.