COMP90051 – Assignment 2

2014-09-25

Overview

For this project, you will need to implement feed-forward neural networks of varying depth and with each layer capable of varying number of nodes. To train this network, you will need to implement stochastic gradient descent with back propagation to calculate the necessary gradients. Finally, you will optimise the network architecture using genetic algorithms. Note that the second stage will require knowledge from the forthcoming evolutionary algorithm lectures (week starting Oct 6th), hence you will not be able to start stage two yet. However in preparation for state two, you can implement a backprop algorithm that can handle networks with arbitrary number of layers and number of nodes in each layers.

For this project, you can choose to either work individually or in a team of maximum 2 members.

Stage 1 – Implementing backprop

The first task is to implement feed forward neural networks and back propagation training algorithm. You are *not permitted* to use an existing neural network library, you must implement the training yourself. You may choose any language to work in, however we strongly suggest a language with linear algebra support.

For this first stage the architecture is fixed. You will have a single hidden layer of 100 nodes, and 10 output nodes (one for each class). Use the hyperbolic tangent as your activation function for all nodes. Note that the derivative of the hyperbolic tangent $\tanh(x)$ is the square of the hyperbolic secant $\operatorname{sech}(x) = \frac{2}{e^x + e^{-x}}$.

Submit predictions on the test set for this stage after you have trained your model on the test set.

Stage 2 - Learning the Optimal Network Structure via Genetic Algorithm (GA)

This second task is to learn the optimal network structure, i.e. with varying number of layers and nodes in each layers via *Genetic Algorithm* (GA). The maximum number of hidden nodes per layer is restricted to 100 nodes, while the maximum number of hidden layers is restricted to 10 (you can employ less layers and nodes if ever desired). The activation function is still the hyperbolic tangent. The nodes between two consecutive layers are assumed to be fully connected. One possible approach for this task is as follows:

- Make sure you have implemented a backprop algorithm that can handle networks with arbitrary number of layers and number of nodes in each layers.
- Divide the training data into two subsets: sub-training and sub-validation.
- For each network structure, train the network using backprop on the sub-training subset. Using the performance on the sub-validation subset as the fitness function for GA.
- Finally, with the identified best network structure, retrain the network on the whole training dataset via back propagation. Make predictions for the test set.

(Bear in mind that this is not necessarily the best approach!)

Submit a separate prediction file on the test set for this stage.

Data description

The data is a hand written digit recognition task. You may download it from LMS. There are two CSV files in the archive, *train.csv* and *test-nolabel.csv*, corresponding to training and test data sets. The first column of the CSV files are the labels and indicate which of the 10 different classes the sample belongs to. There are no labels for the testing dataset hence the first column is set to -1 in this file; this dataset is purely for submitting predictions and subsequent grading.

Final submission

The final submission will be in four parts:

1. Predictions resulting from stage 1 of the assignment. File named as: "your student number_your name_stage1.txt", e.g. "30403020_John Doe_stage1.txt". 1 prediction label per line;

2. Predictions resulting from stage 2 of the assignment. File named as: "your student number_your name_stage2.txt", e.g. "30403020_John Doe_stage2.txt". 1 prediction label per line; Make sure you name your submission files as required, or it might be missed by the automatic marking script!
3. Your source code used for both stages of the assignment. File named as: "your student number_your name_code1.zip" and "your student number_your name_code2.zip" in a zip archive, e.g. "30403020_John Doe_code1.zip" Important: The source code should be clearly documented!;
4. A maximum 3-page report, named as "your student number_your name_report.ext", e.g., "30403020_John Doe_report.pdf":
    - Outlining your approach to stage 1, what were the difficulties and lessons learned?
    - Outlining your approach to stage 2, what was the best architecture you were able to find?
    - You should substantiate points in your report by making correspondence between the report and source code (e.g. this idea is implemented in lines 10-18 of the code).

Note: all submissions are to be made via LMS. If you work in a team, each member still has to submit a separate, though identical, set of result files, code and report *under your own name and student number*. You must indicate both members of your team in your report.

Assessment

Assessment will be out of 20 marks allocated as follows:

1. 10 marks from your report and submitted code.
2. 10 marks from your performance for stages one and two (80% from accuracy and 20% from ranking).