NeuraNet

P. Baillehache

October 30, 2018

Contents

Introduction

NeuraNet is a C library providing structures and functions to implement a neural network.

The neural network implemented in NeuraNet consists of a layer of input values, a layer of output values, a layer of hidden values, a set of generic base functions and a set of links. Each base function has 3 parameters (detailed below) and each links has 3 parameters: the base function index and the indices of input and output values. A NeuraNet is defined by the parameters' values of its generic base functions and links, and the number of input, output and hidden values.

The evaluation of the NeuraNet consists of taking each link, ordered on index of values, and apply the generic base function on the first value and store the result in the second value. If several links has the same second value index, the sum value of all these links is used. However if several links have same input and output values, the outputs of these links are multiplied instead of added (before being eventually added to other links having same output value but different input value).

The generic base functions is a linear function. However by using several links with same input and output values it is possible to simulate any polynomial function. Also, there is no concept of layer inside hidden values, but the input value index is constrained to be lower than the output one. So, the links can be arranged to form layers of subset of hidden values, while still allowing any other type of arrangement inside hidden values. Also, a link can be inactivated by setting its base function index to -1. Finally, the parameters of the base function and the hidden values are constrained to [-1.0,1.0].

NeuraNet provides functions to easily use the library GenAlg to search the values of base functions and links' parameters. An example is given in the unit tests (see below). It also provides functions to save and load the neural network (in JSON format).

NeuraNet has been validated on the Iris data set, the Abalone data set, the Arrhythmia data set, the Wisconsin Diagnostic Breast Cancer data set, the MNIST data set, the ORHD data set.

A utility tool allows to generate a file to be used as input of the CloudGraph tool to visualize the network of the NeuraNet.

It uses the `PBErr` library.

1 Definitions

The generic base function is defined as follow:

$$B(x) = [tan(1.57079 * b_0)(x + b_1) + b_2] \tag{1}$$

where $\{b_0, b_1, b_2\} \in [-1.0, 1.0]^3$ are the parameters of the base function and $x \in \mathbb{R}$ and $B(x) \in \mathbb{R}$.

2 Interface

3 Code

3.1 pbmath.c

3.2 pbmath-inline.c

4 Makefile

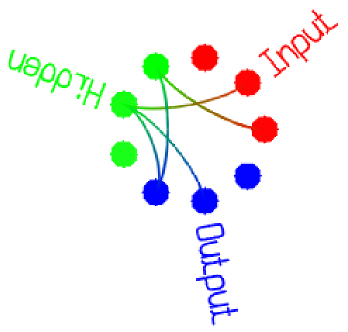
5 Unit tests

6 Unit tests output

neuranet.txt:

bestnn.txt:

cloud.txt:



7 Validation

7.1 Iris data set

Source: https://archive.ics.uci.edu/ml/datasets/iris

main.c:

learn.txt:

checktest.txt:

checkall.txt:

7.2 Abalone data set

Source: http://www.cs.toronto.edu/ delve/data/abalone/desc.html

main.c:

learn.txt:

checktest.txt:

checkall.txt:

7.3 Arrhythmia data set

Source: https://archive.ics.uci.edu/ml/datasets/arrhythmia

main.c:

learn.txt:

checktest.txt:

checkall.txt:

7.4 Wisconsin Diagnostic Breast Cancer

Source: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+Diagnostic

main.c:

learn.txt:

checktest.txt:

checkall.txt:

7.5 MNIST

Source: http://yann.lecun.com/exdb/mnist/

main.c:

learn.txt:

checktest.txt:

checkall.txt:

7.6 ORHD

Source: https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

main.c:

learn.txt:

checktest.txt:

checkall.txt:

8 nn2cloud