Machine Learning.
Project brief

September 2, 2019

Abstract

As a project you have to code and Neural Network to classify the imaged.
Submission:

1. Software manual (50% of mark)
2. Code of Neural Net (50% of mark)

Submission is due 26 of September. Total mark is 10% of course mark

1 Your training data sets

We provide training and working datasets for your Neural Network. A dataset is a set of low-resolution 8x8 black-and-white images.
Each image can be one out of 8 possible types.

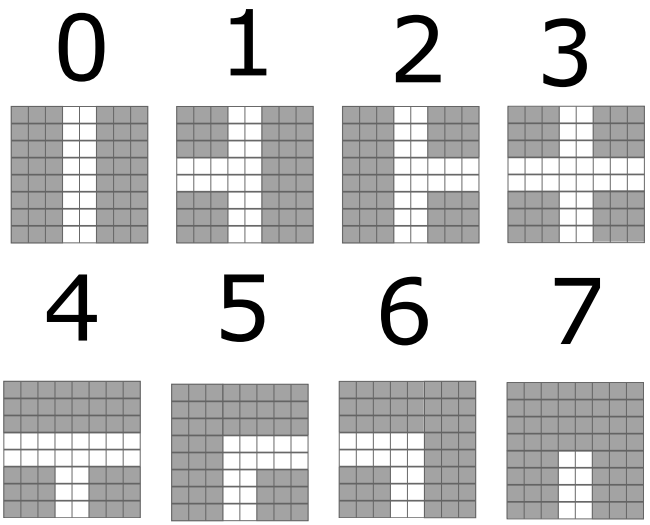

Figure 1: Images and their classifications

The dataset is a text file. All entries are of type **double**. Each row (string) is one entry of the dataset.
Each string is comprised of 64 pixel values (-10.0 for black and +10.0 for white, you can scale the values if your program requires it) which are inputs for the Neural Network and 8 **target** values. Only one of 8 target values contains 1.0 and the rest are 0.0.

The working dataset contains images without targets.

2 Coding Neural Network

We provide some code as a starting point for your Neural Network code.
Feel free to modify this code any way you see fit.
You are welcome to write code from the scratch. There are some functions already written and some function you have to write.
Code is written in C++.

Variable names are shown at Fig.2

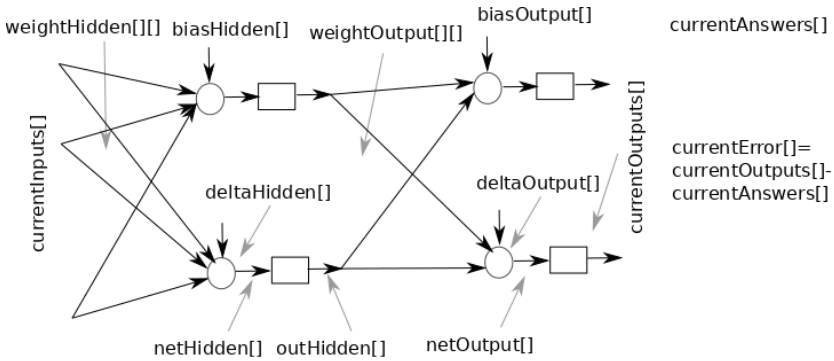

Figure 2: Variable names

Function **int NN::LoadTrainingSet(string file,int nInp,int nHid , int nOut)** loads training dataset from file.
When loading the file function splits it into two 2D arrays:inputsTraining[][] (inputs for whole dataset) and outputsTraining[][] (ditto for outputs) in accordance with nInp and nOut specified.

Function GetTrainingEntry(int iTrainRow) copies one row of inputsTraining[][] and outputsTraining[][] into currentInputs[] and urrentAnswers[] arrays. NN does caculalations based on currentInputs[] and urrentAnswers[].
Function to display currentInputs[] as an ASCII art is DisplayDigit().

2D arrays are stored as 1D arrays by rows: It can be beneficial in some situations.

As a minimum you should implement following functions:

void DirectGradientEstimation();
void StepByGradient();
void Train1();

Tune parameters of your Network until you achieve convergence. You can try to change:

1. Number of neurons in hidden layer
2. Deviation of biases and weights initial values
3. Learning rate

Add function LoadWorkingSet(), following LoadTrainingSet(). Working set does not contain answers.

After training is completed program should load working set and wait for user input of which working set row(entry) to use.

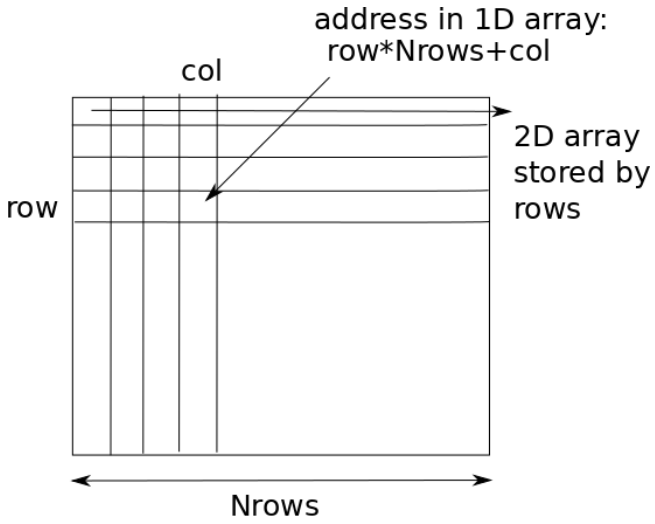

Figure 3: Storing 2D array as 1D

To make trianing process faster you should implement:
void BackProp();

As your program runs it should display totalDatasetError to estimate searc convergence,

3 Marks

3.1 Marks for software (5% of course mark of total marks for the project):

- Search converges - 0.65/1 (Tutor will give your software 5 tries)
- Working dataset is loaded. One user-selected entry of working dataset is loaded and outputs calculated. 0.8/1
- Back-propagation is implemented 1/1

3.2 Description of your software (5% of course marks). Result - software manual, no longer than 3 pages

You are shipping your program to the customer. The customer knows nothing about Neural Networks but heard that they can do amazing things. You should write a manual for how the customer can use your program.
Following should be covered:

1. What a Neural Network is and what it can do

2. What your particular program can do
3. Which input data your program can take
4. What should be changed in your code if input data are different from what you used (number of inputs, number of outputs changed)
5. What the customer should do to achieve good results from the Neural network (all the things you changed when trying to achieve convergence)
6. You can make program interface more user-friendly

Your report can be informal but must be courteous to the paying customer. You can start with "Thanks for buying our software...".

4 Submission

Software manual should be pdf file. Also submit zipped file of your program.