IDIAP COMMUNICATION

THE MNIST DATABASE

Of handwritten upper-case letters

Haiyan Wang
Samy Bengio

IDIAP-COM-02-04

JULY 2002

Dalle Molle Institute
for Perceptual Artificial
Intelligence • P.O.Box 592 •
Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

The MNIST database of handwritten upper-case letters has a training set of 332,178 examples, and a test set of examples 92,977. It is a subset of a larger set available from NIST. The letters have been size-normalized and centered in a fixed-size image.

This database is for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. Generally the files were written in svm-torch format, it's convenient for the torch users, but there is also some software especially for MNIST of handwritten upper-case letters to transfer the data into other useful format.

The file format is described at the bottom of this page.

The original images information from NIST was size normalized while preserving their aspect ratio. The resulting images contain gray levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 32x32 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 32x32 field.

The MNIST of handwritten upper-case letters database was constructed from NIST's Special Database 19, which contains segmented characters images of handwritten digits and letters. Each character occupy a 128x128 pixel raster and are labeled by one of 62 classes corresponding to "0"- "9", "A"-"Z" and "a"- "z". In the MNIST of handwritten upper-case letters database, we use all characters from class "A"-"Z".

The result files are available on /com/databases/NIST/SD19-CAPITAL/41~5a:
Among these files

```
train0~25.svm :              training set images (335690 examples)
test0~25.svm:                test set images ( 92977 examples)
```

The MNIST of handwritten upper-case letters training set is altogether composed of 335,690 patterns from files train0~25.svm. The test set was composed of 92,977 patterns from test0~25.svm.

SVM- Torch has been used for training and testing set. The error on the test set is 1.743%.

Software

There are some software were developed for the usage of MNIST of handwritten upper-case letters database. On /com/databases/NIST/SD19-CAPITAL /tool/svm_pnm

svm_pnm – get viewable image file from a standard svm-torch file.

svm_pnm [input.svm]

input.svm is in format of svm_torch file.

The results are a series of PNM (Portable Anymap) format image.

Hereafter is an image file shown by standard image display program--*"xview"*.

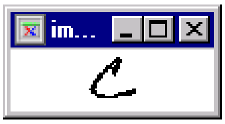

Figure 1 an example of Image file

Acknowledgements

I would be grateful to Dr. Samy Bengio for his explanation of SVM-Torch. Moreover, he is the first person who used this database.

References

Y. LeCun, AT&T Labs-Research "The MNIST Database of handwritten digits."

Patrick J. Grother NIST Information Access and User Interfaces Division 1995 "NIST Special Database 19 Handprinted Forms and Characters Database"

http://www.torch.ch

FILE FORMATS FOR THE MNIST DATABASE OF UPPER-CASE LETTERS

The data is stored in a very simple file format designed for storing vectors and multidimensional matrices.

All the integers in the files are stored in the MSB first (high endian) format used by most non-Intel processors. Users of Intel processors and other low-endian machines must flip the bytes of the header.

TRAINING SET svm-torch FILE (train0~25.svm):

[offset]	[type]	[value]	[description]
0000	32 bit integer	20000	number of images
0005	unsigned byte	0x20	space
0009	32 bit integer	1024	row*columns
0013	unsigned byte	0x0A	carriage return
0014	32 bit integer	112	number of foreground pixel
0018	unsigned byte	0x20	space

```
0019     32 bit integer  174           position of foreground pixel
0023     unsigned byte   0x20          space
0024     32 bit integer  255           pixel value of foreground(black)
0028     unsigned byte   0x20          space
........
xxxx     32 bit integer  1             class number [0~25]
                                    (1->A,2->B…25->Y,0->Z)
xxxx     unsigned byte   0x0A          carriage return
(the next image)
xxxx     32 bit integer  116           number of foreground pixel
xxxx     unsigned byte   0x20          space
```

Pixels are organized row-wise. Pixel values are 0 or 255. 0 means background (white), 255 means foreground (black).

TEST SET svm-torch FILE (test0~25.svm):

```
[offset] [type]          [value]       [description]
0000     32 bit integer  20000         number of images
0005     unsigned byte   0x20          space
0009     32 bit integer  1024          row*columns
0013     unsigned byte   0x0A          carriage return
0014     32 bit integer  112           number of foreground pixel
0018     unsigned byte   0x20          space
0019     32 bit integer  174           position of foreground pixel
0023     unsigned byte   0x20          space
0024     32 bit integer  255           pixel value of foreground(black)
0028     unsigned byte   0x20          space
........
xxxx     32 bit integer  1             class number
xxxx     unsigned byte   0x0A          carriage return
(the next image)
xxxx     32 bit integer  116           number of foreground pixel
xxxx     unsigned byte   0x20          space
```

On /home/speech/wang/svm/NIST/sd19/41~5a: You will see some files which named “hsf_*.svm” and “train_*.svm”. They are results file directly mapped with /com/database/NIST/SD19/data/by_class/21~5a “hsf_*.mis” and “train_*.mis”. The format is a bit different from the svm_torch format. But it’s very convenient to merge and transfer into svm_torch.

Example of a “hsf_*.svm” FILE (or train_*.svm):

```
[offset] [type]          [value]       [description]
0000     32 bit integer  112           number of foreground pixel
0004     unsigned byte   0x20          space
0005     32 bit integer  174           position of foreground
0009     unsigned byte   0x20          space
0010     32 bit integer  255           pixel value of foreground(black)
0014     unsigned byte   0x20          space
........
xxxx     32 bit integer  1             number of class
xxxx     unsigned byte   0x0A          carriage return
(the next image)
```

xxxx	32 bit integer	116	number of foreground pixel
xxxx	unsigned byte	0x20	space

Pixels are organized row-wise. Pixel values are 0 or 255. 0 means background (white), 255 means foreground (black).

TRAINING PROCESS BY USING MNIST DATABASE OF UPPER-CASE LETTERS (Provided by Samy Bengio)

SVM-TORCH training methodology used:

The model was trained using SVMTorchII, available on the web through http://www.idiap.ch/learning/SVMTorch.html.

A first series of experiment was performed in order to select the appropriate hyper-parameter "std" for the RBF kernel of the SVM. We tested various values of "std" in the range from 1000 to 4000 on a small set of training and test examples with the following commands:

```
For each i ( various_values_from_1000_to_4000 )
  SVMTorch -load 5000 -multi -t 2 -std $i -m 400 -sparse train.dat model
  SVMTest  -load 5000 -multi -sparse model valid.dat > tmp_res
  echo $i "  " `grep missclassified tmp_res | awk '{print $4}'`
end
```

We selected the best value of "std" according to this loop.
It was equal to 2770.

Then, we trained the model with the following command:

```
SVMTorch -multi -t 2 -std 2770 -m 300 -load 200000 -sparse shuffle_train.dat model
```

Where shuffle_train.dat is a version of train.dat with all the examples shuffled. We only used the first 200000 examples of this file to train the model and not all the training set, in order to speed up training.

Afterward, we tested the quality of the model on the test examples with the following command:

```
SVMTest -multi -sparse model test.dat
```

and obtained 1.7% misclassification rate.

An introduction to SVMs can be found in:

```
@book{vapnik:1995,
 author     = "V. Vapnik",
 title      = "The Nature of Statistical Learning Theory",
 publisher  = "Springer-Verlag",
 address    = "New York",
 year       = 1995
}
```

or

```
@article{burges:1998:dmkd,
  author    = "C. J. C. Burges",
  title     = "A Tutorial on Support Vector Machines for Pattern Recognition",
  journal   = "Data Mining and Knowledge Discovery",
  year      = 1998,
  volume    = 2,
  number    = 2,
  pages     = "1--47"
}
```