**Prediction performance on random permuted 16-digit matrix**

There are four types of 16-digit matrix datasets used in this experiment:

(1) Ordered digit matrix – see Fig. 1;

(2) Permuted digit matrix (fixed random pixel positions of whole matrix) – see Fig. 2;

(3) Permuted digit matrix (varied random image positions of whole matrix), – see Fig. 4;

Note that dataset 2 is generated by randomly permuting pixels of whole matrix in dataset 1, and the random pixel positions are fixed for all samples. Dataset 3 is generated by randomly permuting images of whole matrix in dataset 1, the positions of digit images are varied for each sample.

Four predictive models are used in this experiment to investigate the effect of prediction performance on different permutations.

1. Linear SVM
2. Group Learning
3. Neural network (NN)
4. Convolution neural network (CNN)

The parameters/structures of NN and CNN are used in the TensorFlow tutorials (<https://www.tensorflow.org/tutorials>) for classifying MNIST data. The NN has one hidden layer (regular densely-connected, 128 neurons, activation function = ‘ReLU’), and 20% neurons are randomly dropped-out. Two neurons in the output layer represent the predictions of two classes. The CNN is form (in sequence from first to last) by one convolution layer (with filters = 32, kernel size = 3x3, activation function = ‘ReLU’), one hidden layer (regular densely-connected, 128 neurons, activation function = ‘ReLU’), and one output layer (2 neurons).

The detailed experimental settings are as follows:

- *positive class:* 16-digit matrix composed of digits ‘7’, ‘2’, ‘8’, ‘6’ (4 images for each digit, see Fig. 1);

- *negative class:* 16-digit matrix composed of digits ‘1’, ‘2’, ‘3’, ‘4’ (4 images for each digit, see Fig. 1);

- *feature vector (for group learning)*: real-valued vector of size 784 (representing a single image (28\*28 pixel) in the digit matrix)

- number of training inputs/matrices: 20, 40, 80, 160, 320 (equally distributed in two classes)

- number of validation matrices: 20, 40, 80, 160, 320 (equally distributed in two classes)

- number of test matrices: 1000 (500 per class)

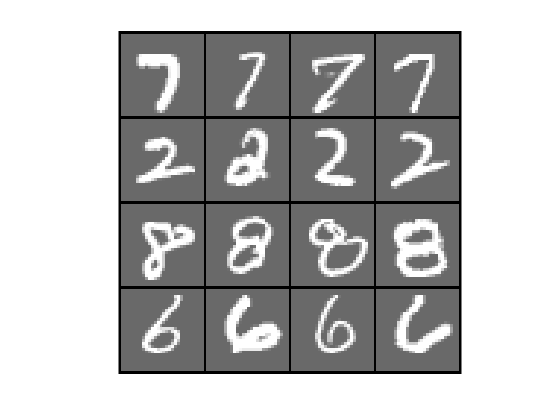
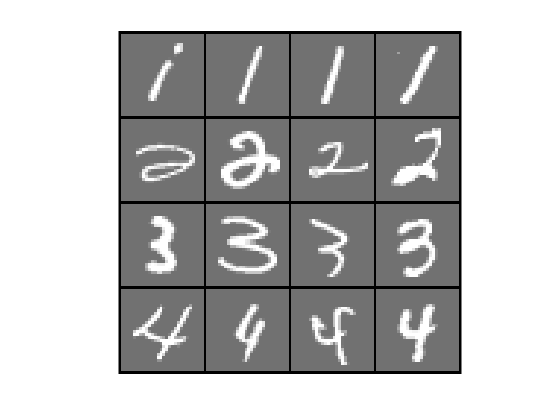
 

Fig. 1. Ordered digit matrix, examples in positive (left) and negative (right)

一張含有 建築物, 螢幕, 窗戶 的圖片

自動產生的描述 一張含有 螢幕, 建築物, 窗戶 的圖片

自動產生的描述

Fig. 2. Permuted digit matrix (fixed random pixel positions), examples in positive (left) and negative (right)

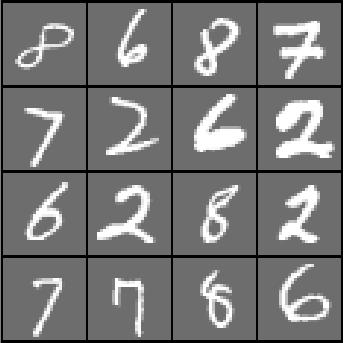
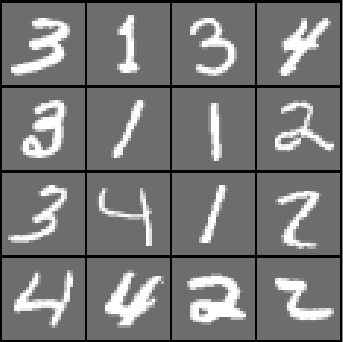
 

Fig. 3. Permuted digit matrix (varied random digit image positions), examples in positive (left) and negative (right)

**Results:**

The prediction results of four models on datasets 1, 2, and 3 are shown in Tables 1, 2, and 3 correspondingly.

For dataset 1, SVM and NN show the best prediction performance, both of them achieve SS=1.00, SP=1.00 for all training sample sizes (see Table 1). GL and CNN also show good prediction performance on dataset 1. However, when the training samples size is small, they make small number of errors. This result may suggest that when the classification task is simple, and the training sample size is small, complex models may overfit the problem.

For dataset 2, the results are similar to the results of dataset 1 (see Table 2). All models achieve good prediction performance. Especially for SVM, it still generates near perfect classification. Randomizing the pixel positions (with fixed order for all samples) does not increase the level of challenge as it looks like.

For dataset 3, the challenge level of classification is increased, which can be observed from the degradation of prediction performance (see Table 3). Most models can not present good prediction performance when the training sample size is small. Especially for NN and CNN, the degradation of prediction performance is very significant. However, GL still shows good prediction performance even for the small training sample size.

**Table 1.** Prediction performance of four models on dataset 1 (models are trained using five different samples sizes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # training samples | SVM | | GL | | NN | | CNN | |
| SS | SP | SS | SP | SS | SP | SS | SP |
| 20 | 1.00 | 1.00 | 1.00 | 0.94 | 1.00 | 1.00 | 0.97 | 1.00 |
| 40 | 1.00 | 1.00 | 1.00 | 0.94 | 1.00 | 1.00 | 1.00 | 0.99 |
| 80 | 1.00 | 1.00 | 1.00 | 0.98 | 1.00 | 1.00 | 1.00 | 1.00 |
| 160 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 1.00 |
| 320 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

**Table 2.** Prediction performance of four models on dataset 2 (models are trained using five different samples sizes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # training samples | SVM | | GL | | NN | | CNN | |
| SS | SP | SS | SP | SS | SP | SS | SP |
| 20 | 1.00 | 1.00 | 1.00 | 0.93 | 0.98 | 0.99 | 0.99 | 0.98 |
| 40 | 1.00 | 1.00 | 1.00 | 0.95 | 1.00 | 0.99 | 1.00 | 0.99 |
| 80 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 1.00 |
| 160 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 1.00 |
| 320 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

**Table 3.** Prediction performance of four models on dataset 3 (models are trained using five different samples sizes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # training samples | SVM | | GL | | NN | | CNN | |
| SS | SP | SS | SP | SS | SP | SS | SP |
| 20 | 0.91 | 0.82 | 1.00 | 0.96 | 0.87 | 0.62 | 0.72 | 0.61 |
| 40 | 0.96 | 0.89 | 1.00 | 0.96 | 0.77 | 0.80 | 0.99 | 0.55 |
| 80 | 0.97 | 0.97 | 1.00 | 0.98 | 0.96 | 0.94 | 0.97 | 0.95 |
| 160 | 0.99 | 0.99 | 1.00 | 0.98 | 0.97 | 0.98 | 0.98 | 0.97 |
| 320 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 0.97 | 0.99 | 0.99 |

**Fig. 4** Relationship between training sample size and performance indices, SS(left) and SP(right), of four models on dataset 3.