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

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

(1) Ordered matrix – see Fig. 1;

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

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

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

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 (representing a 16-digit matrix)*

a. SVM: a real-valued vector of size 12544 (i.e., 16\*28\*28)

b. GL: 16 real-valued vectors of size 784 (i.e., 28\*28)

c. NN & CNN: a real\*valued matrix (i.e., 112x112)

- 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)

**Model selection:**

The parameter C of SVM is selected via cross-validation, which generates the lowest validation error. GL is based on SVM, and the parameter C of SVM is selected via cross-validation as well. The real-valued outputs of SVM classifiers (corresponding to short group features) are combined to make a single prediction of GL. The threshold for making prediction of GL is estimated using validation data.

The parameters/structures of NN and CNN are used in the TensorFlow tutorials (<https://www.tensorflow.org/tutorials>) for classifying digit images in 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).

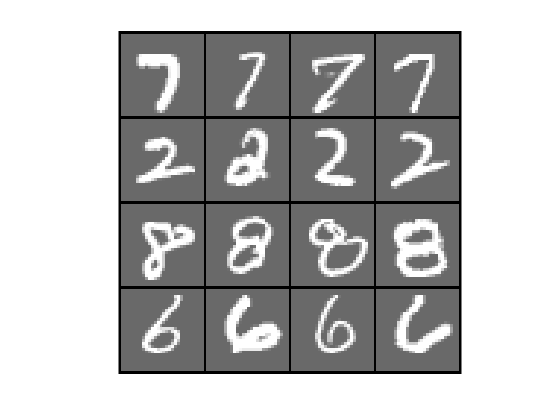
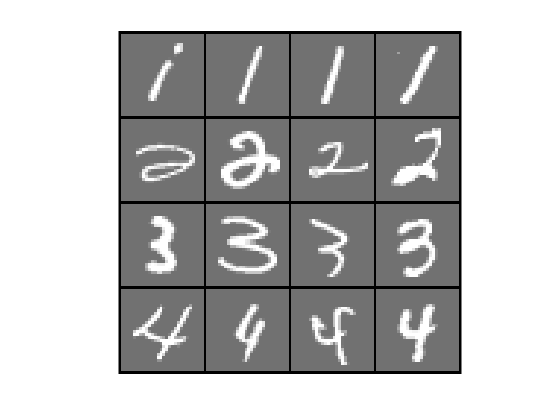
 

Fig. 1. Ordered matrix, examples in positive (left) and negative (right)

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-pixel matrix (fixed random pixel positions), examples in positive (left) and negative (right)

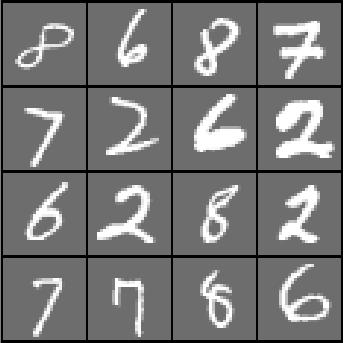
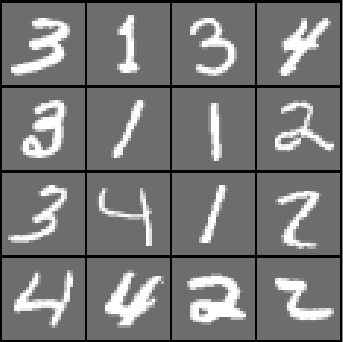
 

Fig. 3. Permuted-matrix matrix (varied random digit image positions), examples in positive (left) and negative (right)

**Results:**

The prediction (test) results of four models on datasets 1, 2, and 3 are shown in Tables 1, 2, and 3 correspondingly. Note that the training errors of all experiments (for all models applying to all datasets) are very small ~ 0.

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 sample size is small, they make a small number of errors. This result may suggest that when the classification task is simple, and the training sample size is small, complex models (e.g., GL, CNN) may overfit the problem (because of small training error ~ 0).

For dataset 2, the results are similar to the results of dataset 1 (see Tables 1 and 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 difficulty of classification is increased, which can be observed from the degradation of prediction performance (see Table 3). Most models can not show good prediction performance when the training sample size is small (< 40). 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 (< 40). Fig. 4 improves the understanding of the relationship between training sample size and performance indices (SS and SP). The prediction performances of all models are improved by increasing the training sample size. Significant improvements can be found on NN and CNN. Their SS and SP are poor and unstable when the training sample size is less than or equal to 40. After increasing the training size to 160, they can achieve performance comparable to SVM and GL (but still inferior to them).

In summary, when the classification problem is simple, SVM shows the best and stable prediction performance. Complex models (e.g., GL and CNN) may overfit the problem. On the contrary, when the difficulty of classification increases, GL shows the best prediction performance, especially, when the training sample size is small.

**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±0.00 | 1.00±0.00 | 1.00±0.00 | 0.94±0.04 | 1.00±0.00 | 1.00±0.00 | 0.97±0.06 | 1.00±0.00 |
| 40 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.94±0.04 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.99±0.01 |
| 80 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.98±0.02 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 |
| 160 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.99±0.01 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 |
| 320 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 |

**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±0.00 | 1.00±0.00 | 1.00±0.00 | 0.93±0.03 | 0.98±0.04 | 0.99±0.02 | 0.99±0.01 | 0.98±0.03 |
| 40 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.95±0.02 | 1.00±0.00 | 0.99±0.01 | 1.00±0.00 | 0.99±0.01 |
| 80 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.99±0.01 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 |
| 160 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.99±0.01 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 |
| 320 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.01 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 |

**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±0.02 | 0.82±0.05 | 1.00±0.00 | 0.96±0.03 | 0.87±0.14 | 0.62±0.25 | 0.72±0.29 | 0.61±0.37 |
| 40 | 0.96±0.02 | 0.89±0.04 | 1.00±0.00 | 0.96±0.03 | 0.77±0.25 | 0.80±0.18 | 0.99±0.01 | 0.55±0.18 |
| 80 | 0.97±0.01 | 0.97±0.01 | 1.00±0.00 | 0.98±0.03 | 0.96±0.02 | 0.94±0.02 | 0.97±0.03 | 0.95±0.02 |
| 160 | 0.99±0.01 | 0.99±0.00 | 1.00±0.00 | 0.98±0.01 | 0.97±0.02 | 0.98±0.02 | 0.98±0.01 | 0.97±0.01 |
| 320 | 1.00±0.00 | 0.99±0.01 | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.97±0.02 | 0.99±0.00 | 0.99±0.00 |

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