**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 pixel positions of whole matrix), – see Fig. 3;

(4) 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 pixels of whole matrix in dataset 1, and the random pixel positions are varied for each sample. Dataset 4 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 (10 per class);

- number of validation matrices: 20 (10 per class);

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

**Results:**

The prediction results on test data of four datasets are shown in Table 1. All models show good prediction on datasets 1 and 2 (i.e., SS > 0.90, SP > 0.90). Especially, traditional SVM shows perfect prediction on both datasets. It shows that permuting pixels with **fixed random positions** does not affect the prediction performance of all models. Prediction task for dataset 3 is especially challenging; all models show poor prediction on it. The result indicates permuting pixels with **varied random positions** increases the difficulty of classification. To the dataset 4, Group Learning shows the best prediction performance (SS = 1.00, SP = 0.90). The prediction performance of traditional SVM is inferior to Group Learning (SS = 0.93, SP = 0.77). Two neural network models (NN and CNN) show the worst prediction performance (NN: SS = 0.76 and SP = 0.44, CNN: SS = 0.61 and SP = 0.72). This result indicates that permuting digit images with **varied random positions** increases the difficulty of prediction task. However, Group Learning still can predict successfully, which may be because the local information of digit images is well preserved. Two neural network models (NN, CNN) show the worst prediction performance because of the difficulty of prediction task, which requires more data for training. Their prediction performance can be improved by increasing the number of training data. Table 2 shows the prediction performance of NN and CNN on the same four datasets; both models are trained using twice amount of training data (i.e., 40 training samples for NN and CNN). The NN shows higher SP = 0.66. The CNN shows significant improvement to SS and SP (SS = 0.90, SP = 0.84), which are comparable to SVM’s prediction performance (but still worse than Group Learning).

**Table 1.** Prediction performance of four models on test data of four datasets (average of five repeats)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | SVM | | Group Learning | | NN | | CNN | |
| SS | SP | SS | SP | SS | SP | SS | SP |
| 1 | 1 | 1 | 1 | 0.94 | 1 | 1 | 0.95 | 0.98 |
| 2 | 1 | 1 | 1 | 0.90 | 0.98 | 1 | 0.98 | 0.94 |
| 3 | 0.74 | 0.28 | 0.87 | 0.27 | 0.80 | 0.20 | 0.41 | 0.59 |
| 4 | 0.93 | 0.77 | 1.00 | 0.90 | 0.76 | 0.44 | 0.61 | 0.72 |

**Table 2.** Prediction performance of DNN and CNN (trained using double amount of training data) on test data of four datasets (average of five repeats)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | NN | | CNN | |
| SS | SP | SS | SP |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 0.99 |
| 3 | 0.61 | 0.41 | 0.56 | 0.43 |
| 4 | 0.77 | 0.66 | 0.90 | 0.84 |

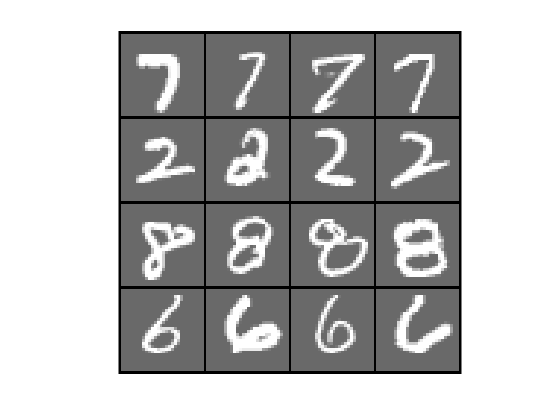
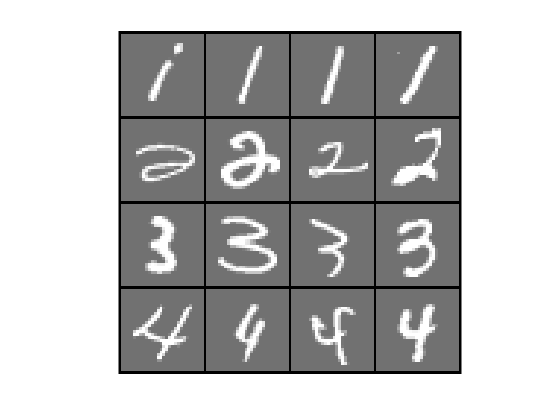
 

Fig. 1. Ordered digit matrix, examples in positive (left) and negative (right)

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)

一張含有 螢幕, 建築物, 窗戶 的圖片

自動產生的描述 一張含有 建築物, 螢幕, 光, 窗戶 的圖片

自動產生的描述

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

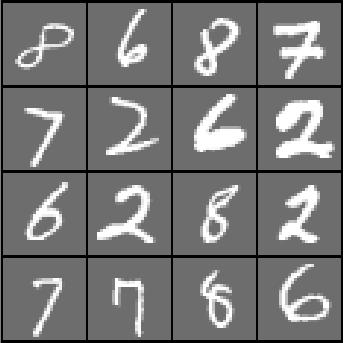
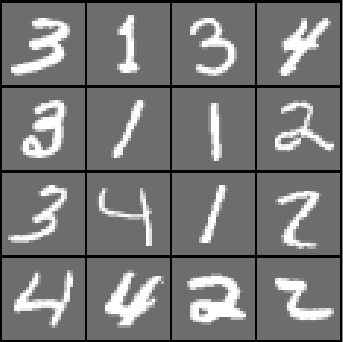
 

Fig. 4. Permuted digit matrix (varied random digit image positions), examples in positive (left) and negative (right)