Mini-Project 1.
Feature Extraction and Transfer Learning

WU Huimin
Dept of CSE
HKUST
Clear Water Bay
Student ID: 20669876
hwubl@connect.ust.hk

HE Changxiang
Dept of MAE
HKUST
Clear Water Bay
Student ID: 20675461
cheag@connect.ust.hk

Abstract

In this project, we implement feature extraction by scattering net with known invariants and resnet18 based on the MNIST dataset, with visualizing these features using classical unsupervised learning methods, including PCA and t-SNE, and classification of traditional supervised learning methods including LDA, logistic regression, SVM, and random forest. The statistics of the features are calculated and analyzed as well.

Remark of contribution

WU Huimin: Programming

HE Changxiang: Analysis the results and write the report

1 Dataset

We have chosen MNIST dataset[1] to work, using 48,000 training data, 12,000 validation data and 10,000 testing data.

2 Feature extraction

2.1 Scattering net

In this project, we applied the scattering net to extract features of MNIST images. Scattering Net can be regarded as a feature extractor with a structure similar to a convolutional network. It is composed of three convolutional layers, and every output of the convolution operation would be collected as output of the network. We can see that it behaves like a convolutional network without fully connected layers. However, the main difference between them is that the filter in Scattering Net is fixed and carefully designed by wavelet transform (or other methods), rather than learning from data using gradient descent like what deep convolutional networks do. Therefore, there are some properties that can be guaranteed such as its invariant under some transformation. Scattering Net captures translation invariant features which is stable to deformations[2]. The filter in Scattering Net is carefully designed and fixed rather than learned from data.

There is a mature python package called Kymatio[8] which offers wavelet scattering transform in Python and can use GPUs. The implementation process is quite easy. The image size of MNIST is $p = 28^2$. Based on these features, we used traditional supervised learning methods. We use scale parameter of 2 and obtain 81 coefficients for each output position. The output feature map is of size 7 by 7. After flattening an image is represented by a $81 \times 7 \times 7 = 3969$ long feature vector.

2.2 Resnet18

The network architectures for resnet are shown as Table 1[3]. Resnet considers "shortcut connections", which are the connections skipping one or more layers.

Table 1 The network architectures for resnet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

A pre-trained ResNet-18 is applied to Mnist and its 1000-d fc output is extracted as features used for visualization and classification in the following sections.

3 Feature visualization

These features are visualized using classical unsupervised learning methods, including PCA and t-SNE. After embedding the features we have extracted into two-dimension space, it is much easier to make a visualization that can be directly perceived by people. In PCA, using only the first two PCs can hardly distinguish the different labels from each other, owing to the low percentage of variance explained by the first two PCs. t-SNE performs much better in this case since it can better capture the non-linear structure than PCA which is essentially a linear projection.

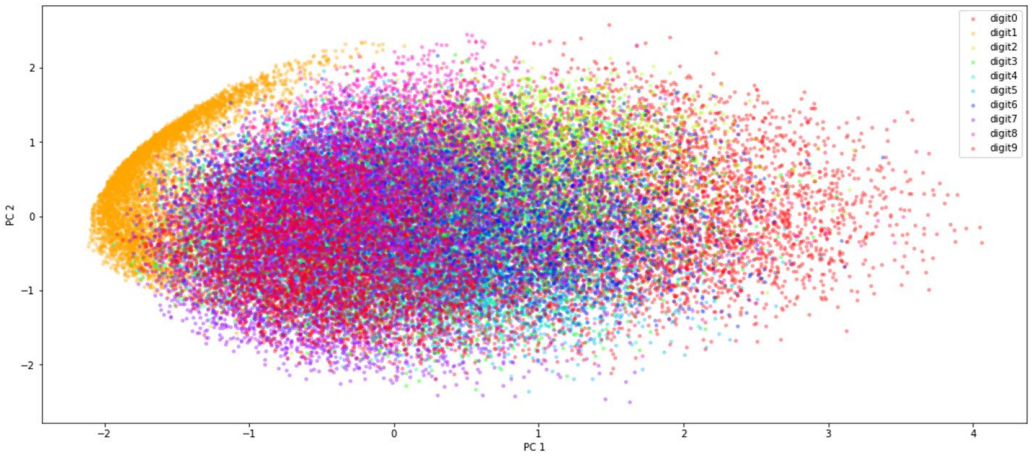

Figure 1: PCA decomposition of features extracted from scattering net

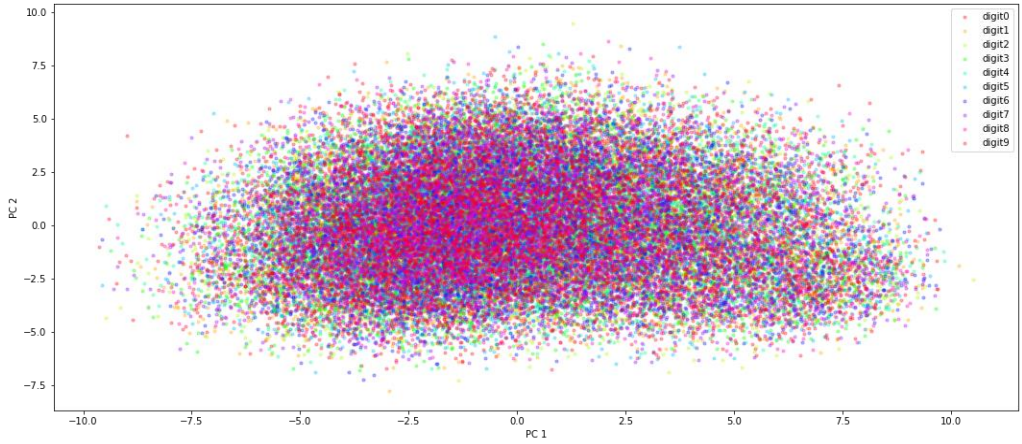

Figure 2: PCA decomposition of features extracted from resnet18

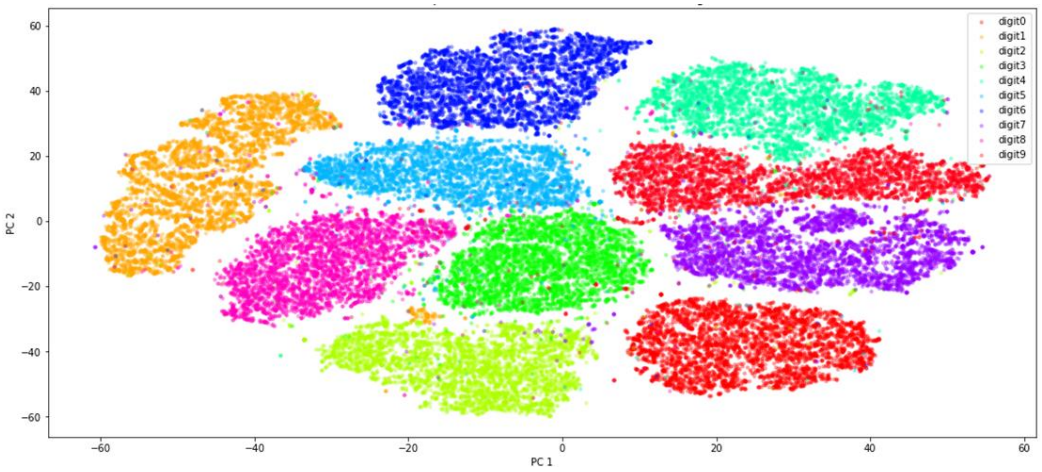

Figure 3: t-SNE decomposition of features extracted from scattering net

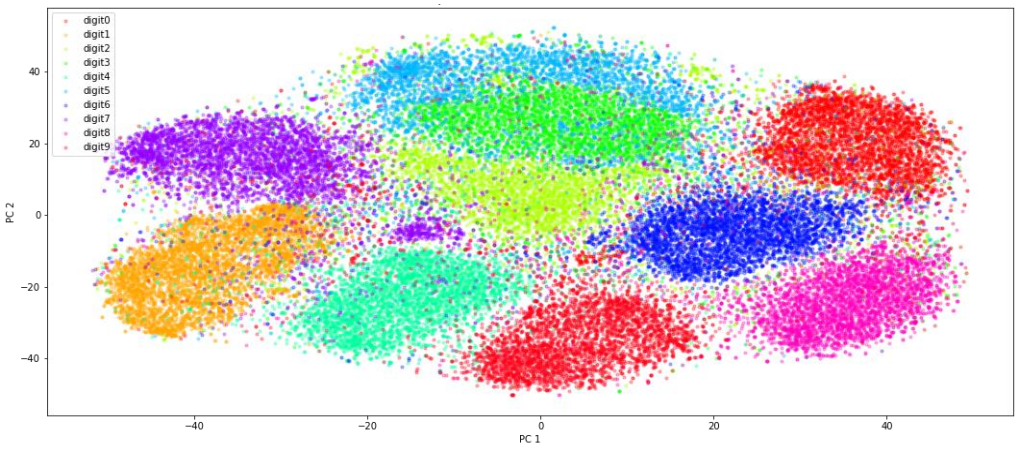

Figure 4: t-SNE decomposition of features extracted from resnet18

4 Compute the global mean of features

We compute the global mean of features over all samples

$$\boldsymbol{\mu}_G \triangleq \underset{i,c}{\text{Ave}} \{\boldsymbol{\Phi}_{i,c}\}$$

class-means

$$\boldsymbol{\mu}_c \triangleq \underset{i}{\text{Ave}} \{\boldsymbol{\Phi}_{i,c}\}, \quad c = 1, \ldots, C$$

total covariance matrix

$$\boldsymbol{\Sigma}_T \triangleq \underset{i,c}{\text{Ave}} \left\{ (\boldsymbol{\Phi}_{i,c} - \boldsymbol{\mu}_G) (\boldsymbol{\Phi}_{i,c} - \boldsymbol{\mu}_G)^\top \right\}$$

between class covariance

$$\boldsymbol{\Sigma}_B \triangleq \underset{c}{\text{Ave}} \left\{ (\boldsymbol{\mu}_c - \boldsymbol{\mu}_G) (\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)^\top \right\}$$

and within class covariance

$$\boldsymbol{\Sigma}_W \triangleq \underset{i,c}{\text{Ave}} \left\{ (\boldsymbol{\Phi}_{i,c} - \boldsymbol{\mu}_c) (\boldsymbol{\Phi}_{i,c} - \boldsymbol{\mu}_c)^\top \right\}$$

such that $\boldsymbol{\Sigma}_T = \boldsymbol{\Sigma}_B + \boldsymbol{\Sigma}_W$. Verify the contraction of within class variation (NC1),

$$\text{Tr} \left\{ \boldsymbol{\Sigma}_W \boldsymbol{\Sigma}_B^\dagger \right\} / C;$$

closeness to equal-norms of class-means

$$\text{Std}_c \left(\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2 \right) / \text{Avg}_c \left(\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2 \right),$$

equal-angularity,

$$\text{Std}_c \left(\cos_\mu \left(c, c' \right) \right) = \text{Std}_c [\langle \boldsymbol{\mu}_c - \boldsymbol{\mu}_G, \boldsymbol{\mu}_{c'} - \boldsymbol{\mu}_G \rangle / \left(\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2 \|\boldsymbol{\mu}_{c'} - \boldsymbol{\mu}_G\|_2 \right)]$$

closeness to maximal-angle equiangularity,

$$\text{Avg}_{c,c'} \left| \cos_\mu \left(c, c' \right) + 1/(C-1) \right|$$

Table 2 shows the NC1, equal-norms of class-means, equal-angularity, and closeness to maximal-angle equiangularity with features extracted from scattering net and resnet18, respectively. Other detailed statistics including the global mean of the features, class-means, total covariance matrix, etc. can be found in the Jupyter notebook.

Table 2 statistics of the features

statistics	Scattering net	Resnet18
NC1	0.0501	0.0047
equal-norms of class-means	0.1134	0.0321
equal-angularity	0.4153	0.3446
closeness to maximal-angle equiangularity	0.1177	0.1114

5 Image classifications

Image classifications with traditional supervised learning methods based on the features extracted may give the identification accuracy straightforward. As Sebastian Mika, etc[4] indicated, LDA is a linear classifier with the main purpose of selecting the best projection direction. After determining the considerable direction, thresholds of different classes can be dissolved. It is easy and straightforward. However, it may cause the problem of overfitting sometimes and the performance may be poor when dealing with the high-dimensional data. According to David W Hosmer and Stanley Lemesbow[7], Logistic regression is a typical method with the regression method to predict the results. It is easy to understand but can be under-fitting sometimes. SVM is a nonlinear method aiming to find a maximum margin in even higher dimensions. It can solve nonlinear and high-dimensional problems but the cost is running time[6]. Random forest is a decision tree algorithm. Some samples are selected from the whole database to generate a cart tree and then be put back in the initial database. After some random sampling process, it derives the final result from the cart trees. Random forest is easy to understand and can be fast in the learning process although it can be overfitted sometimes[5]. Different learning methods including LDA, logistic regression, SVM, and random forest are used to make classification in our case, respectively. The test set data is prepared by scattering net and resnet18. The classification accuracy is shown in Table 3.

Table 3: Classification Accuracy

Learning methods	Scattering net	Resnet18
LDA	0.9909	0.9196
Logistic regression	0.9875	0.9219
SVM	0.9914	0.9175
Random forest	0.9676	0.6795

6 Analysis and Explanation

According to feature visualization figures, the features from scattering net is more clearly clustered by resnet18. As figure 4 shows, the boundaries between different classes are blurred. For example, it's hard for resnet18 to distinguish digit3 and digit5. The classes of them are mixed.

Based on the feature extracted by using scattering net and resnet18 network, we use different learning methods including LDA, logistic regression, SVM, and random forest to further classify the MNIST data. According to Table 3, the classification accuracy from scattering net is always higher than resnet18 under different learning methods. The highest classification accuracy is reached by SVM from scattering net, which is 0.9914. The accuracy of random forest methods from resnet18 is only 0.6795.

references

[1] Christopher J.C. Burges Yann LeCun, Corinna Cortes. The mnist database of handwritten digits.

[2] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence, 35(8):1872–1886, 2013.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468), pages 41–48. Ieee 1999.

[5] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. IEEE transactions on pattern analysis and machine intelligence, 28(10):1619-1630, 2006.

[6] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[7] David W Hosmer and Stanley Lemesbow. Goodness of fit tests for the multiple logistic regression model. Communications in statistics-Theory and Methods, 9(10):1043–1069, 1980.

[8] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim andén, Eugene Belilovsky, Joan Bruna, Vincent Lostanlen, Matthew J. Hirn, Edouard Oyallon, Sixin Zhang, Carmine Cella, and Michael Eickenberg. Kymatio: Scattering transforms in python, 2018.