An In-Depth Look at Feature Transformation of CNN: A Case Study on MNIST Dataset

Beijing Fang
Department of Civil and Environmental Engineering
hippo@connect.ust.hk

Huangshi Tian
Department of Computer Science and Engineering
htianaa@connect.ust.hk

Yunfei Yang
Department of Mathematics
yunfei.yang@connect.ust.hk

1 Introduction

In this project, we take a close inspection of different techniques of feature extraction on the MNIST [6] dataset. More specifically, we first use two types of convolutional neural network (CNN), scattering networks [8] and residual network [4], to extract features (Section 2). Then we visualize the features to demonstrate the similarity and difference between their transformation ability (Section 3). Furthermore, we apply traditional machine learning algorithms to the features and compare their accuracy (Section 4). Finally, we analyze and discuss our results in Section 5

2 Feature Extraction

The MNIST database is a data set of handwritten digits which has a training set of 60,000 examples, and a test set of 10,000 examples. Each example in the database is a 28×28 image in gray scale. Thus, each example can be treated as raw features of size 784. We use scattering network [8] (ScatNet) and residual network [4] (ResNet) to extract features from the raw data.

2.1 Scattering Network

The ScatNet we use to extract features is set with the following parameters.

$$\text{the maximum scattering order } M = 2,$$
$$\text{the number of scale } J = 3,$$
$$\text{the number of orientations } L = 6.$$

Thus, the scattering transform of an image $x(u)$ is

$$S_J x(u) := [x * \phi_J(u), |x * \psi_{j,q}| * \phi_J(u), ||x * \psi_{j,q}| * \psi_{j',q'}| * \phi_J(u)]_{1 \le j \le j' \le J, 1 \le q, q' \le L},$$

where convolutions with ϕ_J are followed by a subsampling of 2^J, $\phi_J(u) = 2^{-2J}\phi(2^{-J}u)$ is a Gaussian low-pass filter, and $\psi_{j,q}$ is a Morlet wavelet ψ scaled by 2^j and rotated along angle $q\pi/L$. These filters are showed in detail in Figure 1.

In order to further reduce the dimension, we take spatial averages of scattering coefficients:

$$\text{features of } x(u) = \sum_u S_J x(u).$$

The summation destroys the spatial information contained in scattering coefficients, and provides only a crude approximation of their distribution. Nonetheless, our experiments show that it is sufficient for classification purposes.

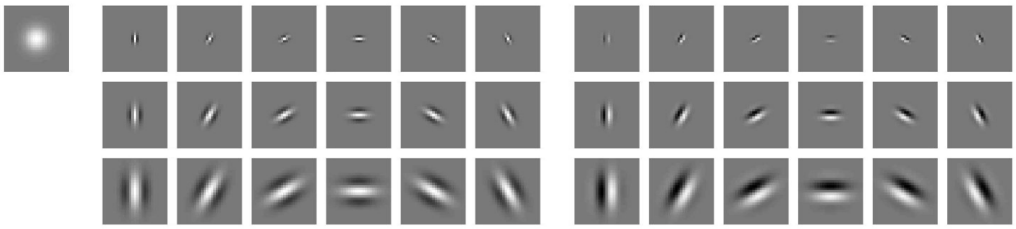


Figure 1: The top left image corresponds to ϕ. The first left half corresponds to the real parts of $\psi_{j,q}$, arranged according scales (rows) and orientations (columns). The right half image corresponds to the imaginary part.

In summary, for each image $x(u)$, the features extracted by the scattering transform is a vector of dimension $1 + JL + L^2J(J-1)/2 = 127$.

2.2 Residual Network

We use as the feature extractor an 18-layer residual network, with its final fully connected layer removed. Since it is pre-trained with ImageNet dataset, it only allows input images of shape at least $3 \times 224 \times 224$. Given that the MNIST images are of shape 28×28, we rescale them to 224×224 with bi-linear interpolation and then triple it to get the desired shape. The features generated by the ResNet is a vector of dimension 512.

3 Feature Visualization

In order to examine the extracted features with different techniques, we utilize visualization to inspect three aspects of the features: (1) distance, (2) structure and (3) principal components.

3.1 Methodology

For distance metrics, we resort to three types of distance: (1) global distance, (2) local distance of near neighbors and (3) geodesic distance. Technically, we use Multidimensional Scaling [2] (MDS), Locally Linear Embedding [9] (LLE) and Isomap [10] respectively. Such a variety of distances helps us better understand how transformation affects the output features, especially in their mutual distances.

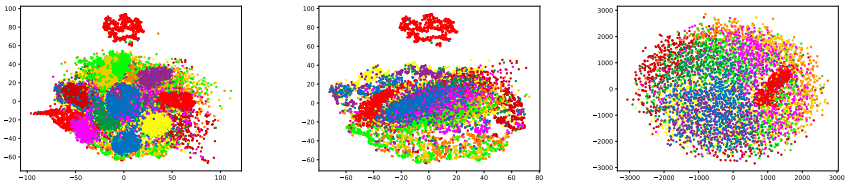
Then we dig into the structure information of those features. On one hand, we employ Spectral Embedding [1] (SE) to compute its graph Laplacian and decompose it to examine how connected components vary. On the other hand, we make use of t-distributed Stochastic Neighbor Embedding [7] (t-SNE) to investigate local affinities among data samples.

Finally, we put to use a traditional technique, Principal Component Analysis [11] (PCA), to explore the principal components of the extracted features.

For each visualization technique, we apply it to (1) the raw image, (2) the features extracted with scattering network and (3) ResNet. We plot with multiple colors to differentiate the features of different digits. Our comparison is centered around how CNN transforms the feature and how scattering network differs with traditional CNN.

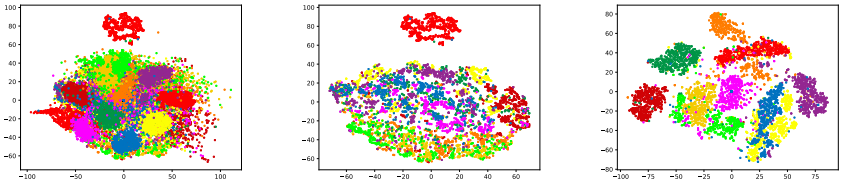
3.2 Visualization

Figure 2 presents the visualization of features generated with MDS. As clearly shown, compared with raw images, CNN could significantly reduce the distance between same-class data samples. This can be easily explained by the translation and rotation invariance of CNN, which tends to produce similar features for similar digits. Furthermore, ResNet outperforms ScatNet in that its "clustering" effect is more obvious.



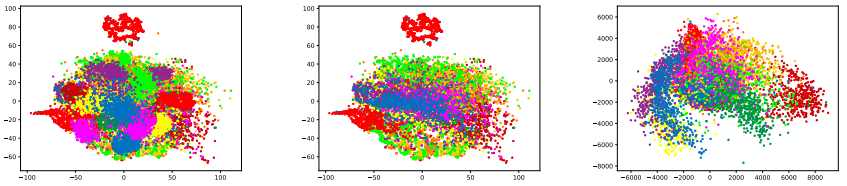
(a) ResNet (b) ScatNet (c) Raw

Figure 2: Visualization of features generated with Multidimensional Scaling (MDS).



(a) ResNet (b) ScatNet (c) Raw

Figure 3: Visualization of features generated with Locally Linear Embedding (LLE).



(a) ResNet (b) ScatNet (c) Raw

Figure 4: Visualization of features generated with Isomap.

Figure 3 shows the result of LLE method. Although the features transformed by CNN show certain degree of cliques, the raw features display better grouping of intra-class data samples in terms of local distances among close neighbors.

The visualization with Isomap is plotted in Figure 4. Slightly similar to that of MDS, another global distance-oriented method, the result of Isomap reveals that CNN is better at contracting the distance among inter-class samples. Both results are also similar in that ResNet performs better than ScatNet, which confirms the superiority of ResNet in distance contraction.

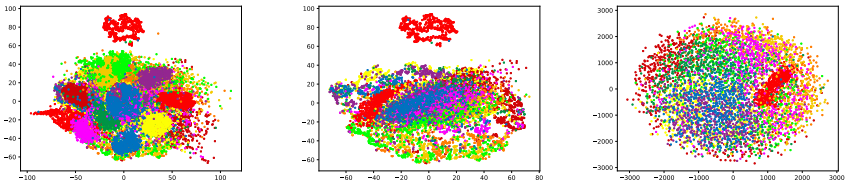
Figure 5 portrays the result of SE. By treating the data samples as a graph and visualizing their connectivity, SE shows that, compared with raw features, CNN groups similar digits and boost their connectivity. Still, ResNet transcends ScatNet because the former displays stronger connectivity.

We apply t-SNE to those features and obtain the result presented in Figure 6. Raw features demonstrate better separability in terms of local affinity among near neighbors. For CNN, it also shows certain grouping effect among intra-class data samples, but it mixes those samples with different digits and blur their boundaries. The result also resonates with that of LLE, another local metric among near neighbors. An empirical explanation could be that CNN transforms features at a global level for all data samples but appears to lose a certain extent of local information as the expense.

Figure 7 exhibits the visualization generated with PCA. Compared with raw features, CNN reduces the variance of data samples and RestNet shows better ability of reduction than ScatNet.

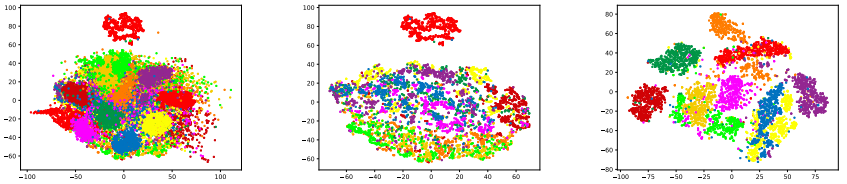
4 Image Classification

We utilize Support Vector Machine [3] (SVM) and Random Forest [5] (RF) for classification in our case. Multiple models are trained with varying sample sizes (300, 1000, 2000, 5000, 10000,



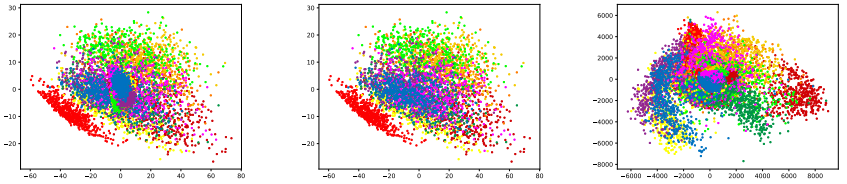
(a) ResNet (b) ScatNet (c) Raw

Figure 5: Visualization of features generated with Spectral Embedding (SE).



(a) ResNet (b) ScatNet (c) Raw

Figure 6: Visualization of features generated with t-distributed Stochastic Neighbor Embedding (t-SNE).



(a) ResNet (b) ScatNet (c) Raw

Figure 7: Visualization of features generated with Principal Component Analysis (PCA).

20000, 40000, 60000) and features (raw, Scatnet, and Resnet features). In addition, all features are normalized by their standard deviation and we drops those features with zero deviation, thus the size of raw features 717. Such dropping is for the sake of computational efficiency. For SVM, we apply five-fold cross validation and grid search to find out the optimal parameters. For RF, we try several choices of estimator numbers and finally settle with 1000. The details of classification methods could be found in our code (`mnist_svm.py, mnist_rfs.py`). Table 4 lists the accuracy results of all models. We further calculate the average accuracy within the same algorithm, the same sample size, and the same feature type.

5 Discussion and Conclusion

Which feature is better and why? The results show that raw features perform the best (92.25%) *on average*, while ScatNet and ResNet features give slightly poorer performance. This may be caused by the information loss during the feature extraction, as the number of features is reduced from 717 to 127 (ScatNet) and 512 (ResNet). Specifically, for ScatNet, the average processing may smooth out some useful information and, meanwhile, introduce biases. For ResNet, since the model used for feature extraction is pre-trained with ImageNet rather MNIST, its may have failed to capture the structural difference among handwritten digits.

How accuracy changes with the training size? Generally, the accuracy increases with growing training size, that is, the more training samples, the better classifier performance. In particular, the performance improves more quickly when the training size is below 5000. On the contrary, when it exceeds 20000, the performance improves relatively slower. In addition, ResNet features

Table 1: Accuracy results of models trained with different sample size and features.

Training Size	**Raw Features** (717)		**ScatNet Features** (127)		**ResNet Features** (512)		**Average (Models)**	
	SVM	RF	SVM	RF	SVM	RF	SVM	RF
300	80.22%	78.26%	82.15%	71.53%	85.06%	78.60%	82.48%	76.13%
1000	87.95%	89.09%	91.05%	80.95%	92.71%	87.69%	90.57%	85.91%
2000	90.05%	92.08%	92.67%	84.22%	93.52%	89.57%	92.08%	88.62%
5000	93.21%	94.35%	95.12%	87.31%	95.25%	90.65%	94.53%	90.77%
10000	94.91%	95.37%	96.09%	88.96%	96.17%	91.32%	95.72%	91.88%
20000	96.06%	96.24%	97.01%	90.73%	96.88%	92.17%	96.65%	93.05%
40000	96.99%	96.85%	97.48%	91.85%	97.49%	92.86%	97.32%	93.85%
60000	97.17%	97.22%	97.72%	92.57%	97.82%	93.00%	97.57%	94.26%
Average (training size)	92.07%	92.43%	93.66%	86.02%	94.36%	89.48%	**93.36%**	**89.31%**
Average (features)	**92.25%**		**89.84%**		**91.92%**			

mostly outperforms ScatNet ones with scarce training samples, which underscores the benefit of non-parametric models. As expected, when we increase the training size, their difference diminishes.

Which classification algorithm is better and why? In our experimentation, SVM (93.36%) mildly outshines RF (89.31%) on average. Using the features extracted with ScatNet and ResNet, RF always performs worse than SVM no matter how we tune the parameters. Since RF randomly chooses data samples during training, how representative the selected data are will determine the model performance. Therefore, the poor performance of RF may be ascribed to the information loss during feature extraction, which also explains why, with raw features, two algorithms yield similar results.

How visualization explains the classification result? Throughout all types of visualization, CNN shows better grouping effect in global metrics, e.g., global distance, geodesic distance, graph connectivity. However, in terms of local metrics within near neighbors, raw features always demonstrate reasonable separability, e.g., local distance, point affinity. Such difference is further confirmed by our classification results, where raw features result in the best accuracy. The cause may be attributed to the deformation stability of CNN. For instance, as digit '3' is similar to '8', CNN tends to generate similar feature so that they appear closer in the visualization. The stability is nonetheless at the expense of certain loss of local information among nearest neighbor because the boundaries are blurred by the transformation.

Furthermore, we could explain the reason why SVM outperforms RF with transformed features but the opposite holds true with raw data. During classification, SVM first applies Radial Basis Function (RBF) as kernel function to all data samples, which involves calculating the distances of each data sample to the support vectors. The visualization of MDS shows that CNN would make the distances of the same-class data samples closer to each other. Hence SVM could classify digits more accurately. However, RF has a nature of decision tree, which focuses on how each data sample is separated from its neighbors. The LLE method reveals that raw images already have reasonable separability among data samples, but CNN transformation will disturb their local structure. Therefore, with raw features, RF outdoes SVM because of better-reserved local structure. But transformed features endow SVM with better performance because the transformed features of same digits will result in similar distances.

To summarize, we have come to following conclusions in our case study.

- On MNIST dataset, raw images could generally serve as better features than those transformed with CNN.
- When training samples are scarce, ScatNet could better capture structural difference than ResNet.
- CNN could contract global distance (metrics) among data samples, but may perturb the local structure at meantime.

- When data samples are ample, ResNet could better cluster data samples by generating closer features.

Contribution

In this project, we have jointly discussed the overall plan. Individually, Yunfei Yang extracts features from the dataset, Huangshi Tian conducts feature visualization and Beijing Fang applies machine learning algorithms to test their performance. After that, we collaboratively write the report.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[2] I. Borg and P. Groenen. Modern multidimensional scaling: theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] T. K. Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[6] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[7] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[8] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65 (10):1331–1398, 2012.

[9] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[10] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[11] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.