Image Classification

STAT5630 Final Project
Yiqin Huang yh4bf

1. Introduction

Recent years, deep learning has obtained remarkable achievements in many fields, including computer vision, speech recognition and natural language processing. As for image classification, convolutional neural network (CNN) has outperformed traditional machine learning models and even produced results in some cases superior to human recognition. There are a lot of news and blogs telling about them and I was told that deep neural networks are designed to extract features automatically and offer an end to end architecture, directly connect raw data and goal. In this report, I make a series experiments to confirm the statement by comparing traditional machine learning models with features extracted from raw data by expanding pixels, principal component analysis and histogram of oriented Gradient with convolutional neural network with raw data.

2. Dataset

Fashion-MNIST[1] is a dataset of Zalando’ s article images, totally 10 classes, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image and some are shown in Figure 1. A summary on data is as Table 1.

Table 1. Summary of classes of Fashion-MNIST

Label	0	1	2	3	4	5	6	7	8	9
Class	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
Train-size	5543	5444	5496	5499	5512	5507	5507	5488	5510	5494
Test-size	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

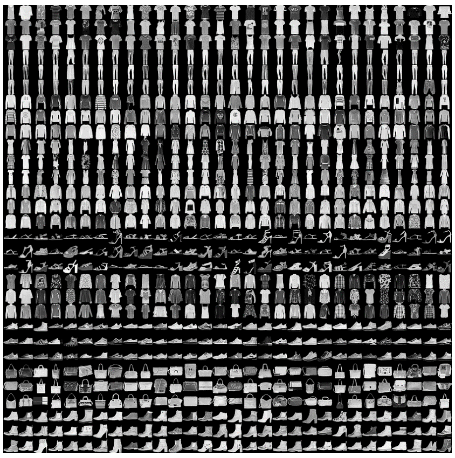

Figure 1. Sample images from Fashion-MNIST, each class 3 rows.

The dataset is intended to be a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. MNIST is easy and most of machine learning models obtain great performance over MNIST. For a classification task, we may see little difference among different models. Comparing to MNIST, Fashion-MNIST can be more challenge and we are going to figure out some methods to improve our models.

3. Experiment Design

In this part, I am going to make a brief introduction on experiment design. The following experiments are aim to make a comparison in four application scenarios, including traditional machine learning with raw data as baseline, separated feature learning and model learning, learning with hand design features and deep learning with raw data, so as to prove the ability of deep neural networks to extract useful features. I have selected some representative features and models.

Support vector machine and random forest models are classical machine learning and known to have great performance in classification. Convolutional neural network have obtained great honor in image classification recent years. Here I'm going to make some comparison.

Principal component analysis(PCA) extracts features from data also by learning. However, this part is independent of model learning, which means what PCA define 'good' features may be useless for classification. Comparing to PCA, convolutional neural network combines feature learning and model learning.

Histogram of oriented Gradient(HOG)[4] is a feature descriptor used in computer vision and image processing for the purpose of object detection. It's regarded as a representative of hand design features in this task. Hand design features are based on human experience or knowledge and can help to improve model performance for some specific datasets or tasks. HOG is proved useful in computer vision but not sure if it is useful for Fashion-MNIST classification.

4. Experiment

In this experiment, I use python to implement models. Use the function to load MNIST in tensorflow module to load Fashion-MNIST for their same store structure, use skimage[2] module to extract HOG feature from dataset, use sklearn module to train support vector machine and random forest and use keras[3] module to train convolutional neural network. The detail of software and modules are as follows.

Table 2. Detail of Software and Module

Software/Module	Version
python	3.6
skimage	0.13.0
sklearn	0.20.0
tensorflow	1.8.0
keras	2.2.0

Step I

In this part, I extract different features from dataset and construct several new datasets, including:

1. Pixel expanding. Expand all the pixels of image by row to a vector so I obtain 784-d dataset. This one will used as baseline.
2. PCA. Limited to computing resource and time, I haven’ t test with setting parameter explained variance ratio range from 0 to 1 and give out the best features. Set the explained variance ratio as 0.8 and 0.9, the number of components are respectively 23 and 83. I train the PCA using training set and transform both training and testing set to new feature. Then I obtain 23-d and 83-d new datasets. The explained variance ratio plot is as figure 2.

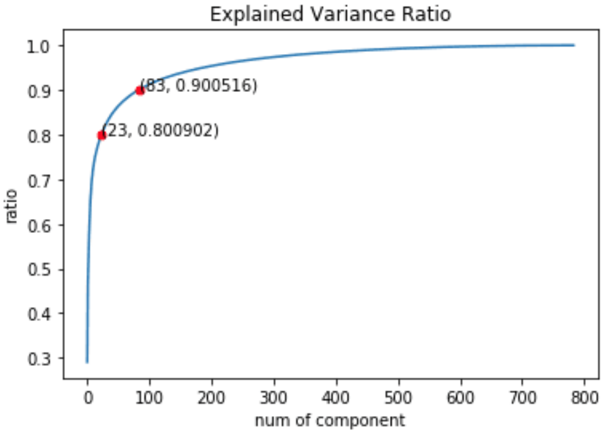

Figure 2. Explained Variance Ratio of PCA

3. HOG. Knowing little about histogram of oriented histogram, I search for information on the Internet. I’ m not going to introduce the theory of HOG but treat it as a method to extract features from images. Using the function in python-skimage, I directly extract

feature from both training and testing set and obtain 81-d new dataset.

In summary, all the datasets are as Table 3.

Table 3. Summary of Datasets

Dataset	Dimension
Original	28x28
Pixel-Expanding	784
PCA(0.8)	23
PCA(0.9)	83
HOG	81

Step II

Using datasets in step I, I train support vector machine, random forest on pixel-expanding, PCA(0.8), PCA(0.9) and HOG, train convolutional neural network on original dataset and test them on corresponding testing set. For not familiar with CNN, I search for some sample code in Github. I split the training set as training and validation set. The curve of loss and accuracy are as Figure 3, which confirms model convergence. And the results of experiments are as Table 4.

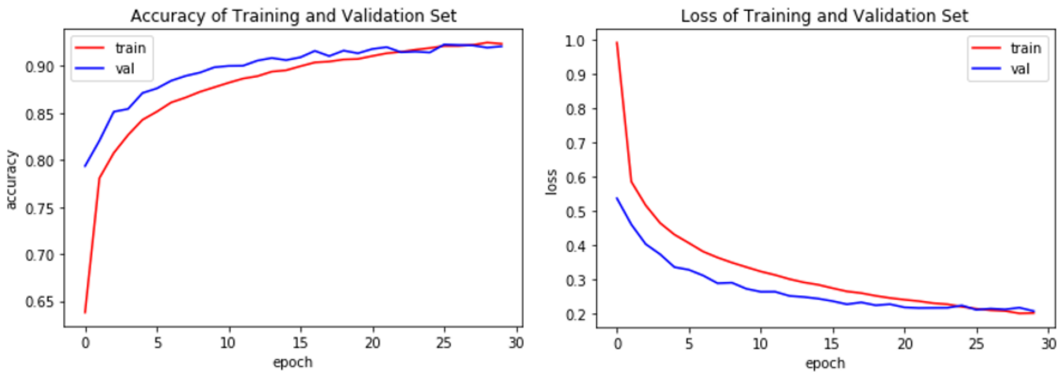

Figure 3. Loss and Accuracy in Training Process

Table 4. Classification Accuracy of Different Models on Testing Set

	Original	Pixel-expanding	PCA(0.8)	PCA(0.9)	HOG
SVM	-	88.43%	**88.92%**	88.85%	85.12%
RF	-	**85.05%**	82.45%	83.79%	79.29%
CNN	**91.70%**	-	-	-	-

From the result,

1. SVM obtains best performance 88.92% with PCA(0.8), slightly improve comparing to pixel-expanding and worst with HOG; Random Forest obtains best performance 85.05% with pixel-expanding and all the extracted features reduce its performance, HOG worst, only

79.29%. CNN obtains the best performance over three models, 91.70%.

2. Considering traditional models with PCA features, the SVM model obtains a better performance with 0.8 explained variance ratio but random forest better with 0.9. We can see, the better PCA fits the data doesn' t mean the higher the classification accuracy is. The features extracting part and classification part are separated. The objective function of PCA focus on explained variance, which may be useless for classification.
3. As for HOG, the feature seems not suitable for classification with Fashion-MNIST. Both SVM and random forest obtain worst performance with HOG. Most time I try to improve the performance with hand design feature by trial and error, which is not efficient both for time and performance.

Step III

In this part, I make a simple experiment to show that feature extracted by CNN is well suitable for classification. In detail, extract the values in hidden nodes of CNN in step II and create a 128-d dataset, retrain support vector machine and random forest model on new dataset and compare the classification accuracy on testing set. The result is as follows.

Table 5. Classification Accuracy of Different Models on Testing Set

	Pixel-expanding	PCA(0.8)	PCA(0.9)	HOG	CNN
SVM	88.43%	88.92%	88.85%	85.12%	**91.78%**
RF	85.05%	82.45%	83.79%	79.29%	**91.29%**
CNN	-	-	-	-	**91.70%**

From the Table, using feature extracted by CNN, both SVM and Random Forest model obtain a great improve on classification accuracy and gain similar performance as CNN. So the CNN extracts useful features for classification automatically.

5. Conclusion

In conclusion,

1. CNN provides an end to end structure to train a classification model directly from raw images. It combines the feature learning and classier learning and involve the feature learning into objective function so that it can extract useful features and obtain better performance than separated structure and hand designed features.

2. In this view, we can try to combine traditional machine learning model with feature extract method, like PCA and give out a objective function consisting of two parts to improve performance.

6. Reference

[1]. https://github.com/zalandoresearch/fashion-mnist
[2]. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
[3]. https://scikit-image.org/
[4]. https://keras.io

7. Code

sol.py