MATH 4432 Project 1 Report

Kaijun HOU (20413837), Qiurui MA (20413112)

March 15, 2018

1 Abstract

We implement LDA, QDA, Logistic Regression and Convolutional Neural Network on MNIST dataset[1]. We do mathematical analysis based on our experiment result. In the following sections, we will "reconstruct" our progress step by step with corresponding explanations.

2 Model design

Many knowledge are from [1]

2.1 Linear Discriminant Analysis

We would like to know the class posteriors $P(Y|X)$. If we denote $f_k(x)$ be the class-conditional probability of X in class $Y = k$ and let $\pi_k = \frac{C_k(X)}{C(X)}$ be the prior probability of class k, where $C_k(X)$ is the count of number of data points belonging to class k. Then, $P(Y = k|X = x) = \frac{f_k(x)\pi_x}{\sum_{l=1}^{K} f_l(x)\pi_l}$. Suppose data in each classes follow Gaussian distribution, then

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)} \tag{1}$$

Linear Discriminant Analysis(LDA) algorithm assumes the within-class variances of all classes are the same ($\Sigma_k = \Sigma \forall k$). Then the log-odds is:

$$log(\frac{P(Y = k|X = x)}{P_{l\neq k}(Y = l|X = x)}) = log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T\Sigma^{-1}(\mu_k - \mu_l) + x^T\Sigma^{-1}(\mu_k - \mu_l) \tag{2}$$

We can solve for the linear discriminant function by setting $P(Y = k|X = x) = P_{l\neq k}(Y = l|X = x)$. We have:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log\pi_k \tag{3}$$

This function is linear in x, which is the reason why it's called "linear" discriminant analysis. The LDA rule classifies a data point to class 2 (suppose binary classification problem) if

$$x^T\Sigma^{-1}(\mu_2 - \mu_1) > \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + log\frac{C_1(x)}{C(x)} - log\frac{C_2(x)}{C(x)} \tag{4}$$

and class 1 otherwise.

2.2 Quadratic Discriminant Analysis

The idea of Quadratic Discriminant Analysis(QDA) is simple. If we relax the assumption of "the within-class variances of all classes are the same" in LDA, then the discriminant funciton would be:

$$\delta_k(x) = -\frac{1}{2}log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T\Sigma_k^{-1}(x - \mu_k) + log\pi_k \tag{5}$$

This function is quadratic form in x.

[1] http://yann.lecun.com/exdb/mnist/

Figure 1: MNIST dataset overview

2.3 Logistic Regression

Logistic Regression assumes the log-odds is linear to x:

$$log\frac{P(Y=k|X=x)}{P_{l\neq k}(Y=l|X=x)}=\beta^T x \tag{6}$$

2.4 Convolutional Neural Network

For the completeness of this project (state-of-art algorithm), we also and one simple Convolutional Neural Network(CNN) model. This model was first proposed by Yann le cunn, and later on gained branchs of improments and extensions (e.g. VGG, ResNets). But since most of the improvements in deep learning are not explainable, we only mention the very basic CNN here.

Model Structure is from [3]: Conv(512,9,relu)MaxPool(2,2)Conv(256,5,relu)MaxPool(2,2)Dense(1024,relu)Dense(10,softmax)

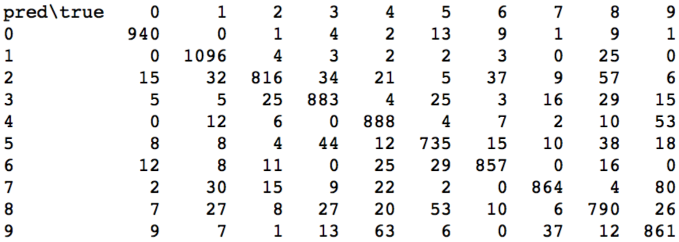
3 Dataset

The dataset we use is MNIST hand-written figure dataset. It has 10 classes (figure 0 to figure 9), 60000 training figures and 10000 testing figures. The digits are size-normalized and centered in size 28×28. *Fig. 1*

4 Experiment[2]

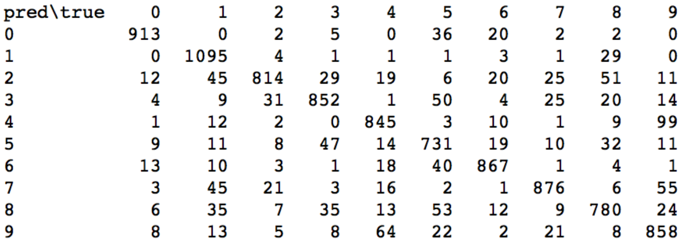
We run our model on AWS EC2 p2.xlarge instance. CPU: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz 4 cores. GPU: NVIDIA Tesla K80. Memory: 60.0G.

[2] code of 4.1 and 4.2 are available in main.ipynb, 4.3 in cnn.ipynb



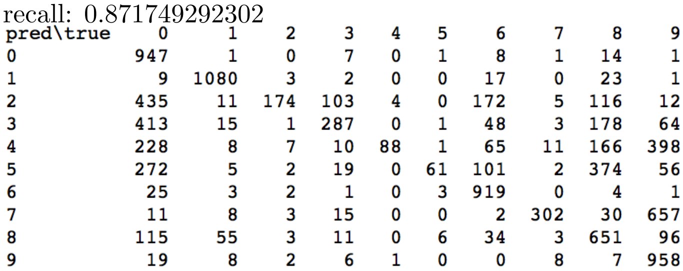
```
pred\true     0     1    2    3    4    5    6    7    8    9
0           940     0    1    4    2   13    9    1    9    1
1             0  1096    4    3    2    2    3    0   25    0
2            15    32  816   34   21    5   37    9   57    6
3             5     5   25  883    4   25    3   16   29   15
4             0    12    6    0  888    4    7    2   10   53
5             8     8    4   44   12  735   15   10   38   18
6            12     8   11    0   25   29  857    0   16    0
7             2    30   15    9   22    2    0  864    4   80
8             7    27    8   27   20   53   10    6  790   26
9             9     7    1   13   63    6    0   37   12  861
```

Figure 2: LDA
precision: 0.873539341852
recall: 0.871749292302



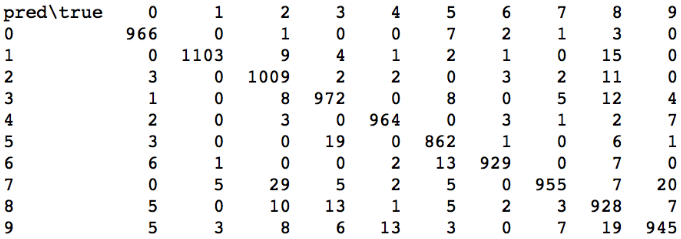
```
pred\true     0     1    2    3    4    5    6    7    8    9
0           913     0    2    5    0   36   20    2    2    0
1             0  1095    4    1    1    1    3    1   29    0
2            12    45  814   29   19    6   20   25   51   11
3             4     9   31  852    1   50    4   25   20   14
4             1    12    2    0  845    3   10    1    9   99
5             9    11    8   47   14  731   19   10   32   11
6            13    10    3    1   18   40  867    1    4    1
7             3    45   21    3   16    2    1  876    6   55
8             6    35    7   35   13   53   12    9  780   24
9             8    13    5    8   64   22    2   21    8  858
```

Figure 3: PCA-LDA
precision: 0.863974712687
recall: 0.861702868307



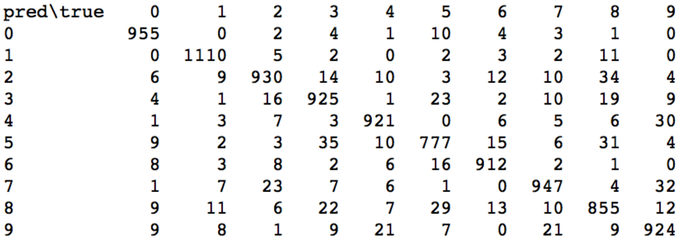
```
pred\true     0     1    2    3    4    5    6    7    8    9
0           947     1    0    7    0    1    8    1   14    1
1             9  1080    3    2    0    0   17    0   23    1
2           435    11  174  103    4    0  172    5  116   12
3           413    15    1  287    0    1   48    3  178   64
4           228     8    7   10   88    1   65   11  166  398
5           272     5    2   19    0   61  101    2  374   56
6            25     3    2    1    0    3  919    0    4    1
7            11     8    3   15    0    0    2  302   30  657
8           115    55    3   11    0    6   34    3  651   96
9            19     8    2    6    1    0    0    8    7  958
```

Figure 4: QDA
precision: 0.699264759575
recall: 0.53995273689



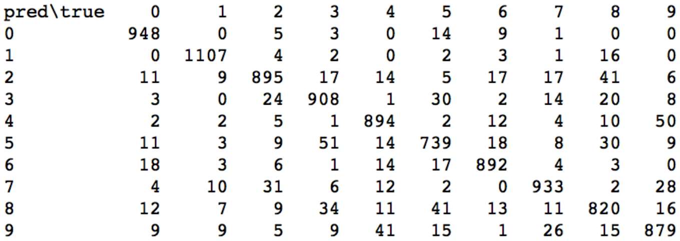
```
pred\true     0     1     2    3    4    5    6    7    8    9
0           966     0     1    0    0    7    2    1    3    0
1             0  1103     9    4    1    2    1    0   15    0
2             3     0  1009    2    2    0    3    2   11    0
3             1     0     8  972    0    8    0    5   12    4
4             2     0     3    0  964    0    3    1    2    7
5             3     0     0   19    0  862    1    0    6    1
6             6     1     0    0    2   13  929    0    7    0
7             0     5    29    5    2    5    0  955    7   20
8             5     0    10   13    1    5    2    3  928    7
9             5     3     8    6   13    3    0    7   19  945
```

Figure 5: PCA-QDA
precision: 0.963363346743
recall: 0.96337075074



```
pred\true     0     1    2    3    4    5    6    7    8    9
0           955     0    2    4    1   10    4    3    1    0
1             0  1110    5    2    0    2    3    2   11    0
2             6     9  930   14   10    3   12   10   34    4
3             4     1   16  925    1   23    2   10   19    9
4             1     3    7    3  921    0    6    5    6   30
5             9     2    3   35   10  777   15    6   31    4
6             8     3    8    2    6   16  912    2    1    0
7             1     7   23    7    6    1    0  947    4   32
8             9    11    6   22    7   29   13   10  855   12
9             9     8    1    9   21    7    0   21    9  924
```

Figure 6: LogReg
precision: 0.924641891104
recall: 0.924519695522



```
pred\true     0     1    2    3    4    5    6    7    8    9
0           948     0    5    3    0   14    9    1    0    0
1             0  1107    4    2    0    2    3    1   16    0
2            11     9  895   17   14    5   17   17   41    6
3             3     0   24  908    1   30    2   14   20    8
4             2     2    5    1  894    2   12    4   10   50
5            11     3    9   51   14  739   18    8   30    9
6            18     3    6    1   14   17  892    4    3    0
7             4    10   31    6   12    2    0  933    2   28
8            12     7    9   34   11   41   13   11  820   16
9             9     9    5    9   41   15    1   26   15  879
```

Figure 7: PCA-LogReg
precision: 0.899789040659
recall: 0.899954030138

4.1 LDA, QDA and Logistic Regression

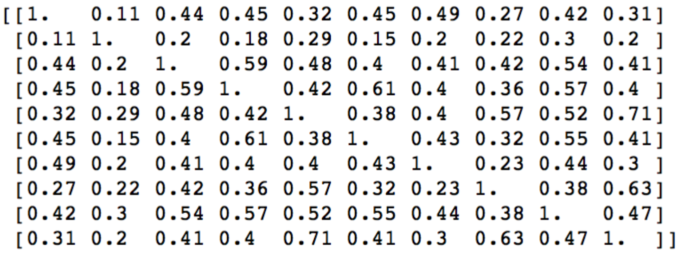
Initally, we wanted to explore the assumptions made by LDA and QDA on classification, namely the underlying distribution of each class and whether each class has a unique covariance matrix.

On the way of exploring the basic assumtpions made by LDA and QDA, we ran into the following problem: the LDA and Logistic Regression yields accuracy as expected (87%, 92% respecitevely) , but the training error as well as test error for QDA is astonishingly high, reaching 30% and even more. This is contradictory to the prior knowledge that QDA should have lower training error at least than LDA. This intriguing fact prompts us into experiments described in section 4.2

The original discussions are carried out after pca dimensionality reduction is done in 4.2. The new dimensionality of the data is 33. Results are illustrated below.

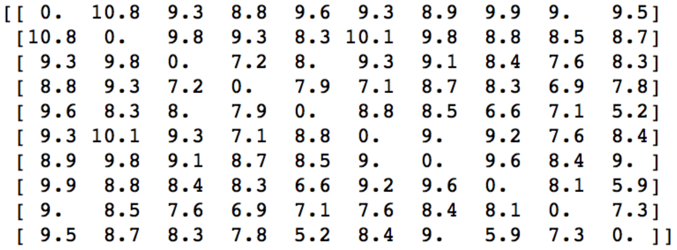
We first carried out the normality test to study whether predictors follow gaussian distribution in each class. With a majority voting mechanism, only one out of 33 predictors is not gaussian in more than 5 classes. The data follow the assumption quite well.

We then turned to the assumption that each class possess its own covariance matrix. By calculating cosine-distance and euclidean-distance of the covariance matrix between ten classes, figure 10 is obtained. This illustrates that covariance indeed differs between classes and QDA achieves more accurate results in this dataset. The figure 8 and figure 9, 2,4 demonstates this perfectly.



```
[[1.   0.11 0.44 0.45 0.32 0.45 0.49 0.27 0.42 0.31]
 [0.11 1.   0.2  0.18 0.29 0.15 0.2  0.22 0.3  0.2 ]
 [0.44 0.2  1.   0.59 0.48 0.4  0.41 0.42 0.54 0.41]
 [0.45 0.18 0.59 1.   0.42 0.61 0.4  0.36 0.57 0.4 ]
 [0.32 0.29 0.48 0.42 1.   0.38 0.4  0.57 0.52 0.71]
 [0.45 0.15 0.4  0.61 0.38 1.   0.43 0.32 0.55 0.41]
 [0.49 0.2  0.41 0.4  0.4  0.43 1.   0.23 0.44 0.3 ]
 [0.27 0.22 0.42 0.36 0.57 0.32 0.23 1.   0.38 0.63]
 [0.42 0.3  0.54 0.57 0.52 0.55 0.44 0.38 1.   0.47]
 [0.31 0.2  0.41 0.4  0.71 0.41 0.3  0.63 0.47 1.  ]]
```

Figure 8: Cosine similarity of within-class covariances



```
[[ 0.  10.8  9.3  8.8  9.6  9.3  8.9  9.9  9.   9.5]
 [10.8  0.   9.8  9.3  8.3 10.1  9.8  8.8  8.5  8.7]
 [ 9.3  9.8  0.   7.2  8.   9.3  9.1  8.4  7.6  8.3]
 [ 8.8  9.3  7.2  0.   7.9  7.1  8.7  8.3  6.9  7.8]
 [ 9.6  8.3  8.   7.9  0.   8.8  8.5  6.6  7.1  5.2]
 [ 9.3 10.1  9.3  7.1  8.8  0.   9.   9.2  7.6  8.4]
 [ 8.9  9.8  9.1  8.7  8.5  9.   0.   9.6  8.4  9. ]
 [ 9.9  8.8  8.4  8.3  6.6  9.2  9.6  0.   8.1  5.9]
 [ 9.   8.5  7.6  6.9  7.1  7.6  8.4  8.1  0.   7.3]
 [ 9.5  8.7  8.3  7.8  5.2  8.4  9.   5.9  7.3  0. ]]
```

Figure 9: Euclidean Differences of within-class covariances

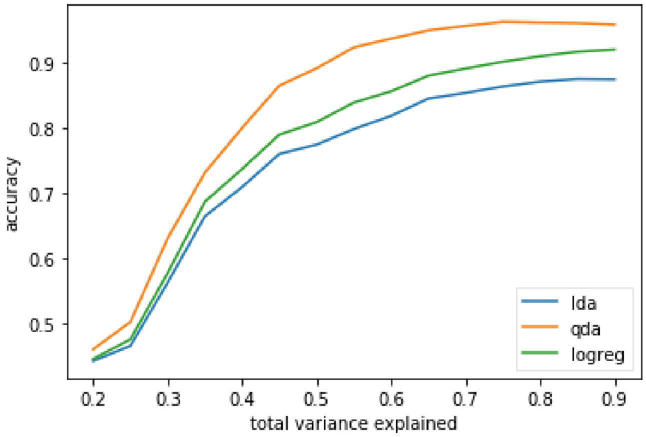


Figure 10: classification accuracy on PCA dimensionality reduced data

4.2 PCA-LDA, PCA-QDA, PCA-Logreg

To resolve the prominent collinearity problem on the raw dataset, which has 784 dimensions, PCA is applied to decrease the dimensions. We do grid search on hyperparameter "total variance explained" to modify the PCA part. Result is shown in *Fig. 10.* The higher the total-variance-explained, the more principle components are kept. According to our observations, accuracy increases monotonically as the total-variance-explained increases and the accuracy of QDA is strictly larger than that of logistic regression than LDA. This follow from the fact that each class has a distinct covariance matrix for the predictors. Compared to data without PCA dimensionality reduction, the performance of LDA is similar, QDA's performance is greatly boosted while logistics regression suffers somehow.

Result of LDA is similar to before. Result of QDA is much better. Result of logistic regression is slightly lower than before. We will explain these observations one by one.

Let's first explain the reason why QDA is significantly better. As described in ..., let's rewrite equation 5 as:

$$\delta_k(x) = -\frac{1}{2}log|\sum_{i=1}^{p} v_{ik}\lambda_{ik}v_{ik}^T| - \frac{1}{2}\sum_{i=1}^{p}\frac{t_{ik}^2}{\lambda_{ik}} + log\pi_k \tag{7}$$

where v_{ik} is the i's eigenvector of Σ_k. λ_{ik} is the corresponding eigenvalue. t_{ik} is the coordinate of x along eigenvector i in class k. When there is a high collinearity among the predictors, λ_{ik} tends to be small. Notice that small λ_{ik}'s tend to explode $\sum_{i=1}^{p}\frac{t_{ik}^2}{\lambda_{ik}}$, this will make $\sigma_k(x)$ super large, which leads to the unstability of QDA [2]. The problem get exacerbated when predictors with small prediction powers has a small λ, When the value of discriminant functions of different class gets compared, small λ with low prediction power shall lead to high bias. This is the reason why QDA has a much higher training error compared with LDA before PCA dimension reduction. However, if we run a PCA to remove the low λ_{ik}'s, the issue is nicely solved.

The reason why the result of LDA and PCA-LDA does not differ a lot is then trivial. Since LDA assumes the same

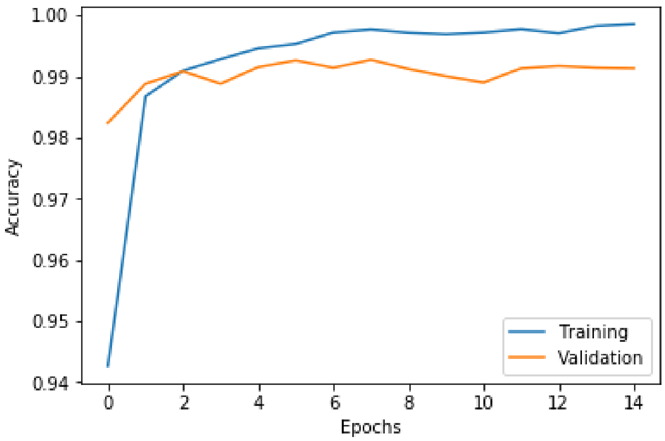


Figure 11: CNN accuracy ˜ epochprecision: 0.991247019038recall: 0.991163365613

within-class variance, when we use equation 4 to do classification, $log\frac{C_1(x)}{C(x)} - log\frac{C_2(x)}{C(x)}$. We now have

$$x^T\Sigma^{-1}(\mu_2 - \mu_1) > \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 \tag{8}$$

The λ term cancels out since λ represents the variance.

We then explain why the result of PCA-Logreg is slightly worse than logreg. Since the discriminant condition is equation 6, the intermediate condition is when the equation equals 0. When the dimension of x is larger, though some of them might coorelated, the degree of freedom is still large to give more description to the equation. When we gradually decrease the dimension of x, both independent dimensions and dependent dimension are descreasing. This leads to a decrease in degree of freedom and a further decrease in the ability to describe the figure. Thus, we can see the decrease in precision and recall. Also, we observe that $precision > recall$ in most cases. Recall that

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

However, we cannot come up with a conclusion. We want to leave this for a open discussion.

4.3 CNN

This result works well on MNIST than other algorithms since the Convolution layers can extract more features[3] as in *Fig. 12*. The training process is illustrated in *Fig. 11*

A future exploration could be related to the dataset itself. As mentioned in section 3, the MNIST dataset is size-normalized and centered. This will make it work well on basic machine learning models since the spatial feature does not change a lot. In this case, MNIST is similar to traditional digital figure in *Fig. 13*. This well designed dataset does not always exist in the real world. Another dataset is created after MNIST to impress this problem. affNIST[4] applies a lot linear transformation, rotation to the original dataset as shown in *Fig. 14*. As proven in [4](IV.A.) Convolution in CNN has translation invariance, which means that no matter how the figure moves within the range, the feature extraction wouldn't be affected. Result of CNN is in *Fig. 15*. We also wrote the code for LDA, QDA and Logistic Regression. However, our machine is not strong enough to finish this task (not enough memory).

[3] if you are interested, feel free to visit `https://yashk2810.github.io/Applying-Convolutional-Neural-Network-on-the-MNIST-dataset/` and `http://cs231n.github.io/understanding-cnn/` for more information

[4] `https://www.cs.toronto.edu/~tijmen/affNIST/`

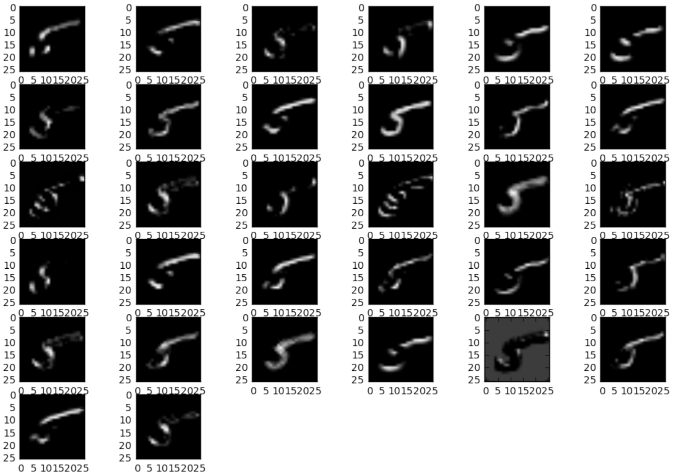


Figure 12: CNN feature layer visualization

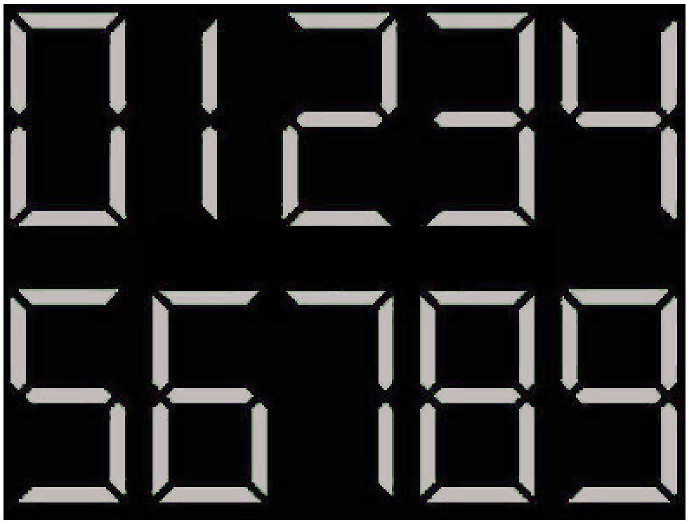


Figure 13: Digital figure

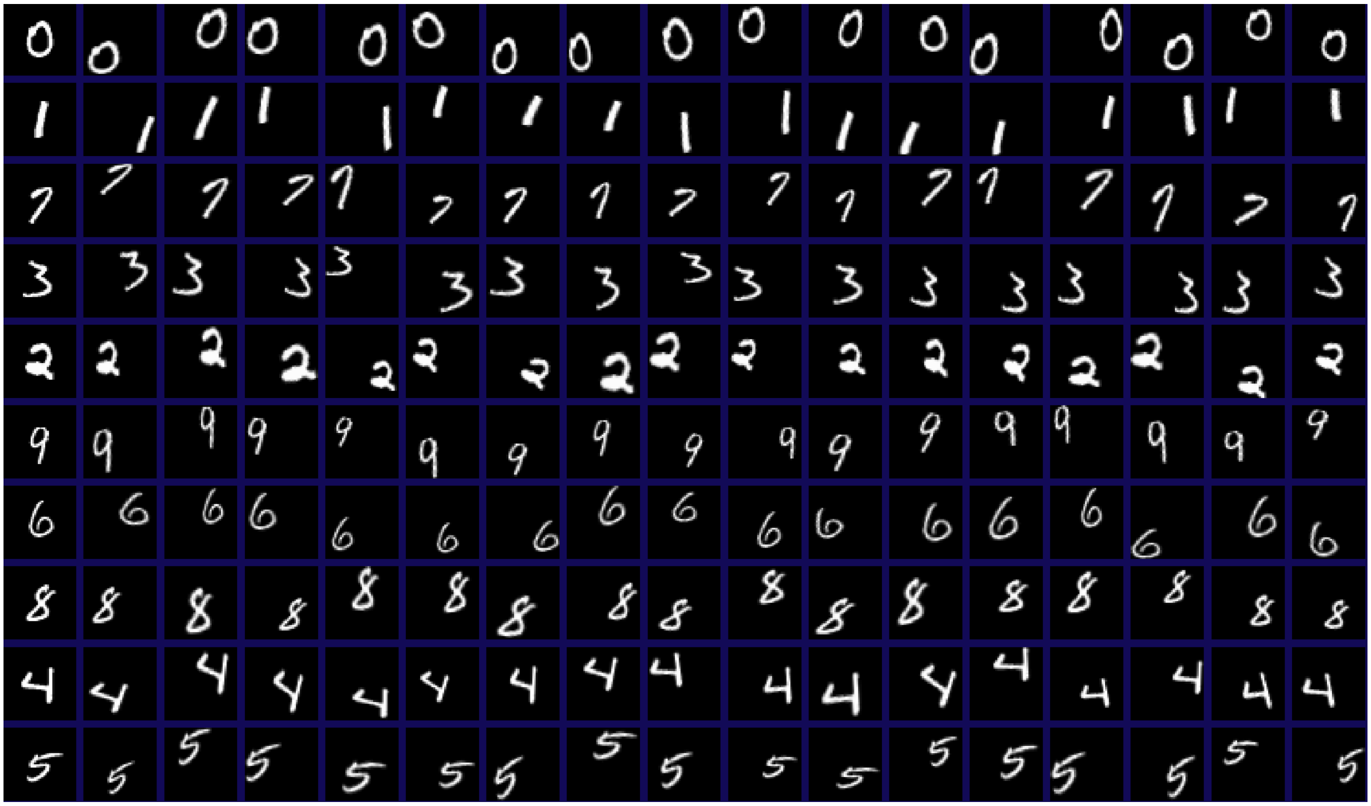


Figure 14: affNIST example

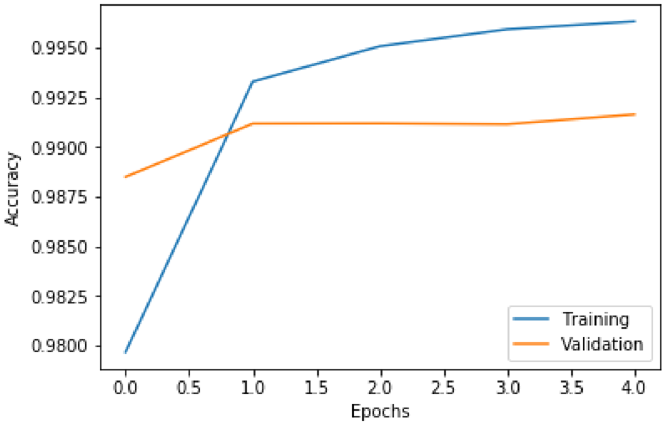


Figure 15: CNN affNIST accuracy ˜ epoch

5 Contribution

5.1 Kaijun HOU

I finished LDA, QDA, logistic regression, CNN on MNIST and extension to affNIST. I explain most of the experiment result and write the most of the report.

5.2 Qiurui MA

1. Explore the gaussian assumption and covariance assumption in discriminant analysis
2. Encounter and research on collinearity issue that QDA faces
3. PCA reduction of mnist and grid search on the best variant-explained ratio.
4. Feature extraction from CNN and classify based on the features. (The QDA suffered greatly in this case, so it does not appear in the paper)
5. Great teamwork with Jefferey

References

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York, 2001.

[2] Tormod Næs and Bjørn-Helge Mevik. "Understanding the collinearity problem in regression and discriminant analysis". In: *Journal of Chemometrics* 15.4 (2001), pp. 413–426.

[3] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules". In: *CoRR* abs/1710.09829 (2017). arXiv: 1710.09829. URL: http://arxiv.org/abs/1710.09829.

[4] Thomas Wiatowski and Helmut Bölcskei. "A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction". In: *CoRR* abs/1512.06293 (2015). arXiv: 1512.06293. URL: http://arxiv.org/abs/1512.06293.