Application of Linear Algebra Methods in MNIST Handwritten Digit Classification

CSCI 8363 Course Project

Yuchen Luo

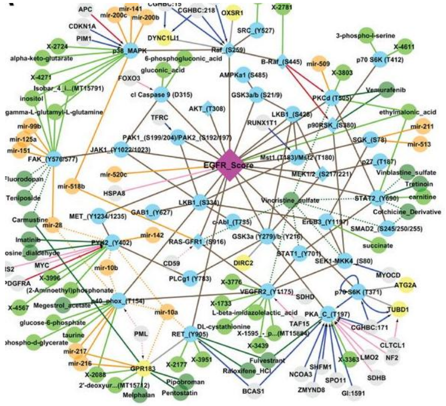
12/13/2017

Outline

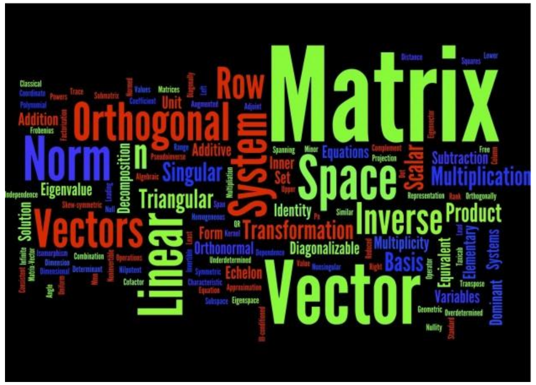
1. Introduction
2. Motivation
3. Method (SVD, NMF)
4. Numerical Experiment (Dataset and Problem)
5. Results (Accuracy Study, Robust Study)
6. Conclusion
7. Outlook

Introduction

Engineering Analysis

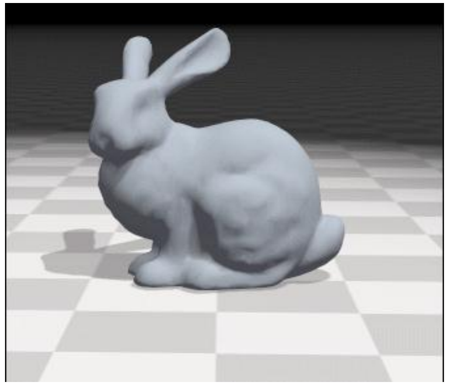


Chemical/Genetic mutation Pathways



Linear Algebra Method

High Performance Computing

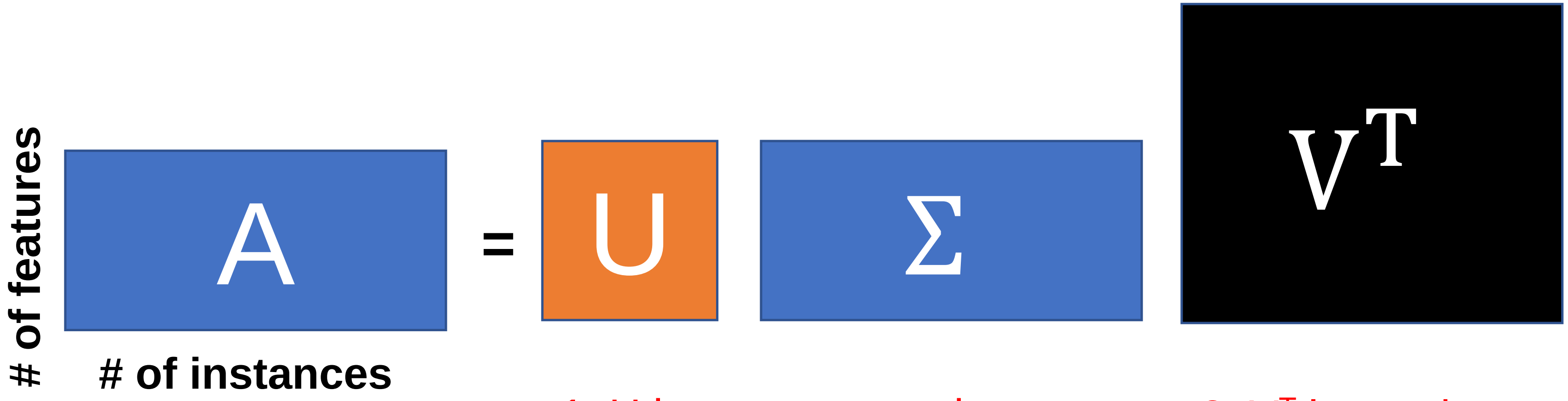


Computational Animation

Motivation

SVD

Singular Value Decompostion



# of features

A

# of instances

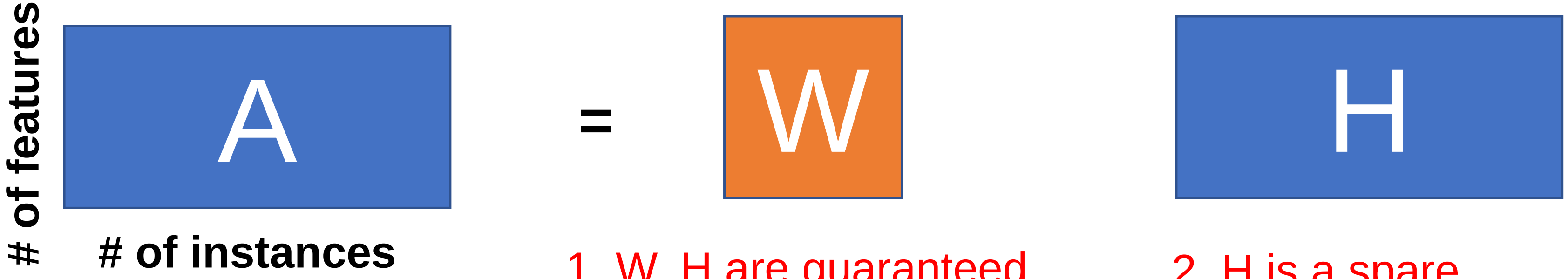
= U Σ V^T

1. U is not nonnegative guaranteed

2. V^T large size, storage challenge

NMF

Nonnegative matrix factorization



# of features

A

# of instances

= W H

1. W, H are guaranteed as nonnegative guaranteed

2. H is a spare matrix easy to store

Method (1) – SVD

Theorem*

For any matrix A ∈ R^{mxn} there exist unitary matrices U ∈ R^{mxm} and V ∈ R^{nxn} such that $A = U\Sigma V^T$ where Σ is a diagonal matrix with entries $\sigma_{ii} \geq 0$

*M. Mazack. “Algorithms for Handwritten Digit Recognition.” Master’s colloquium, Mathematics Department, Western Washington University, 2009.

Method (2) – NMF

Theorem*

Let $A \in R^{mxn}$ be matrix such that $a_{ij} \geq 0$ for all $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,n\}$ (henceforth, $A \geq 0$). Then for $k \leq \min\{m,n\}$, there exist $W \in R^{mxk}$ and $H \in R^{kxn}$ such that $A \approx WH$.

1. NMF gives an approximation for the matrix A and the number $k \leq \min\{m,n\}$ determines the rank of the approximation. Also, in the case of $k = \min\{m,n\}$, it is possible to achieve the equality $A = WH$.

2. Columns of W form a k-dimensional approximation for the column space of A.

3. NMF is not unique.

*M. Mazack. "Non-negative Matrix Factorization with Applications to Handwritten Digit Recognition." Department of Scientific Computation, University of Minnesota, 2009.

Method (2) – Dense NMF Algorithms

$$\text{Minimize} \quad f(W,H) = \frac{1}{2}\|A - WH\|_F^2 \quad \text{subject to:}$$
$$W \in \mathbb{R}^{m\times k} \geq 0,$$
$$H \in \mathbb{R}^{k\times n} \geq 0.$$

Multiplicative Updates

1.Random start on W and H

2.Iterate given number of times

$$H_{ij} \leftarrow H_{ij}\frac{(W^TA)_{ij}}{(W^TWH)_{ij}}$$

$$W_{ij} \leftarrow W_{ij}\frac{(AH^T)_{ij}}{(WHH^T)_{ij}}.$$

Drawbacks

1. Divide by zeros proble
2. Local Minimum

Alternating Least Square

Drawbacks:

1. Inefficient

1.Random start on W and W

2.Iterate given number of times

$$\min_H \|WH - A\|_F^2 \quad \text{such that} \quad H \geq 0,$$

$$\min_W \|H^TW^T - A^T\|_F^2 \quad \text{such that} \quad W \geq 0.$$

Accelerated version of ALS is achieved by projected gradient implementation.

Method (2) – Sparse NMF Algorithms

$$\text{Minimize} \quad f(W,H) = \frac{1}{2}\|A - WH\|_F^2 \quad \text{subject to:}$$

$$W \in \mathbb{R}^{m\times k} \geq 0,$$

$$H \in \mathbb{R}^{k\times n} \geq 0.$$

SNMF/R

$$f(W,H) = \frac{1}{2}\|A - WH\|_F^2 + \eta\|W\|_F^2 + \beta\sum_{j=1}^{n}\|H(:,j)\|_1^2.$$

$$\min_{H} \left\| \begin{pmatrix} W \\ \sqrt{\beta}e_{1\times k} \end{pmatrix} H - \begin{pmatrix} A \\ 0_{1\times n} \end{pmatrix} \right\|_F^2 \quad , \quad \text{such that} \quad H \geq 0.$$

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\eta}I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{k\times m} \end{pmatrix} \right\|_F^2 \quad , \quad \text{such that} \quad W \geq 0.$$

SNMF/L

$$f(W,H) = \frac{1}{2}\|A - WH\|_F^2 + \eta\|H\|_F^2 + \beta\sum_{i=1}^{m}\|W(i,:)\|_1^2.$$

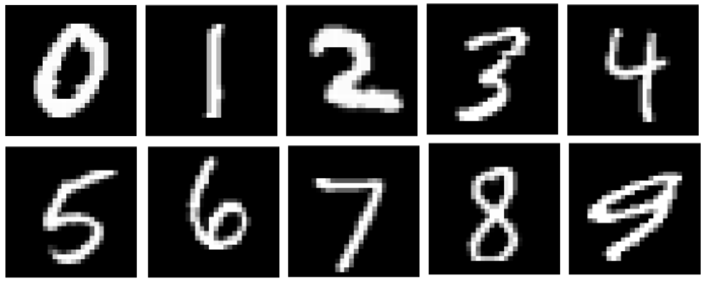
$$\min_{H} \left\| \begin{pmatrix} W \\ \sqrt{\eta}I_k \end{pmatrix} H - \begin{pmatrix} A \\ 0_{k\times n} \end{pmatrix} \right\|_F^2 \quad , \quad \text{such that} \quad H \geq 0$$

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\beta}e_{1\times k} \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{1\times m} \end{pmatrix} \right\|_F^2 \quad , \quad \text{such that} \quad W \geq 0.$$

Dataset

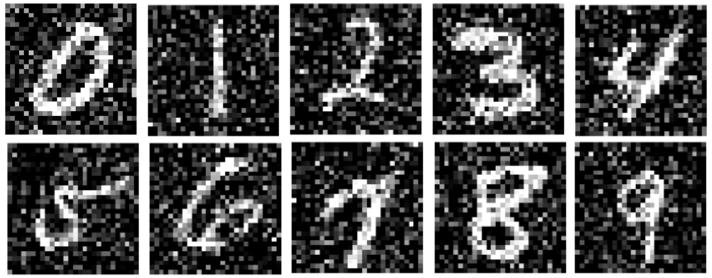
There are two dataset used in this study, one is conventional MNIST dataset, and the other one is the MINIST + Gaussian noise.

LeCun et al., 1998a



60,000 Training Set
10,000 Test Set
Image resolution 28 by 28

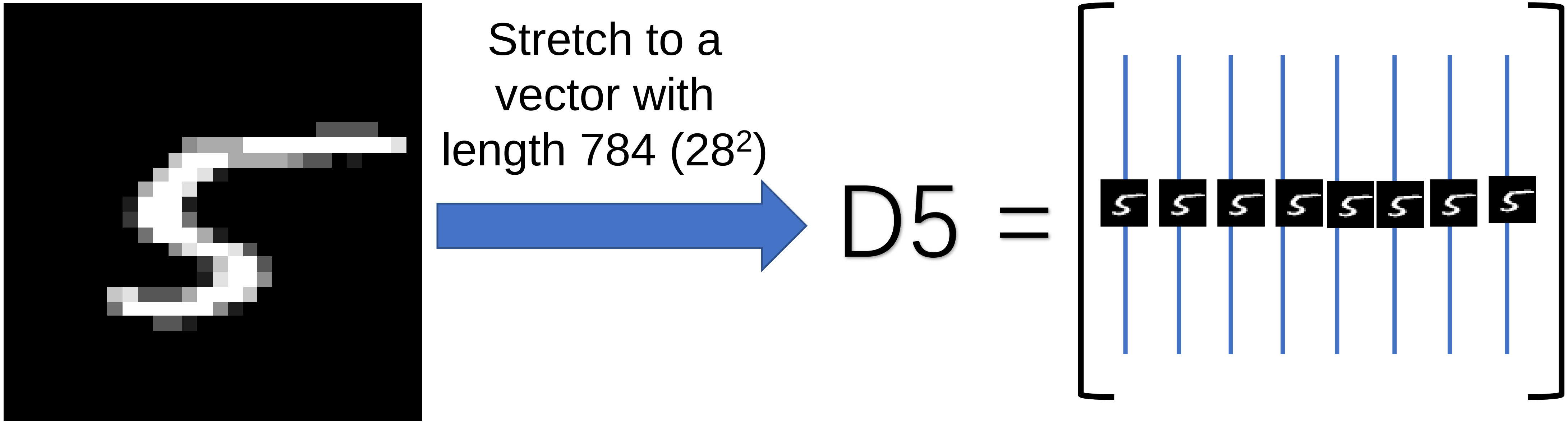
Basu et al., 2015



60,000 Training Set
10,000 Test Set
Image resolution 28 by 28

The AWGN dataset is created using an Additive White Gaussian Noise with signal-to-noise ratio of 9.5

Classification Problem



28 X 28

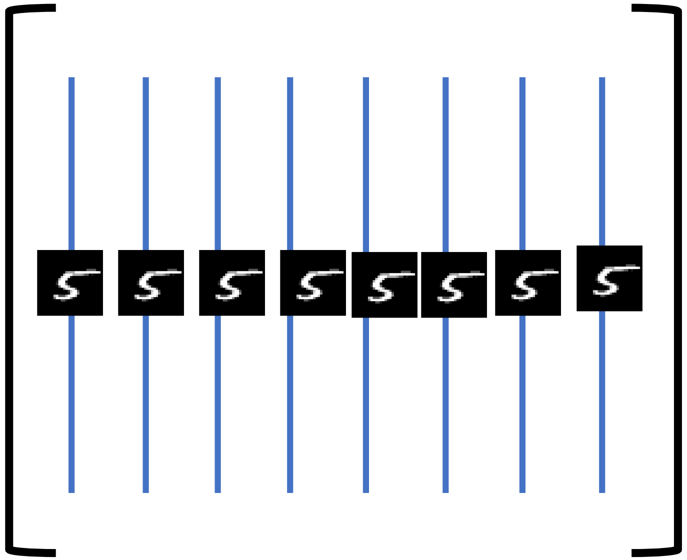
Stretch to a vector with length 784 (28^2)

D5 =

- What is this value? 3 ? , 5 ? or 6 ?
- Automatic recognition of the handwritten digit and classify it to ‘0’ to ‘9’ digits is the classification problem we are dealing.

Classification Problem (Cont.)

D5 =



If a test image is also digit 5, then it is most likely in the space spanned by the D5’s column space. In another word, We want to solve a **least square problem**, which is listed below as

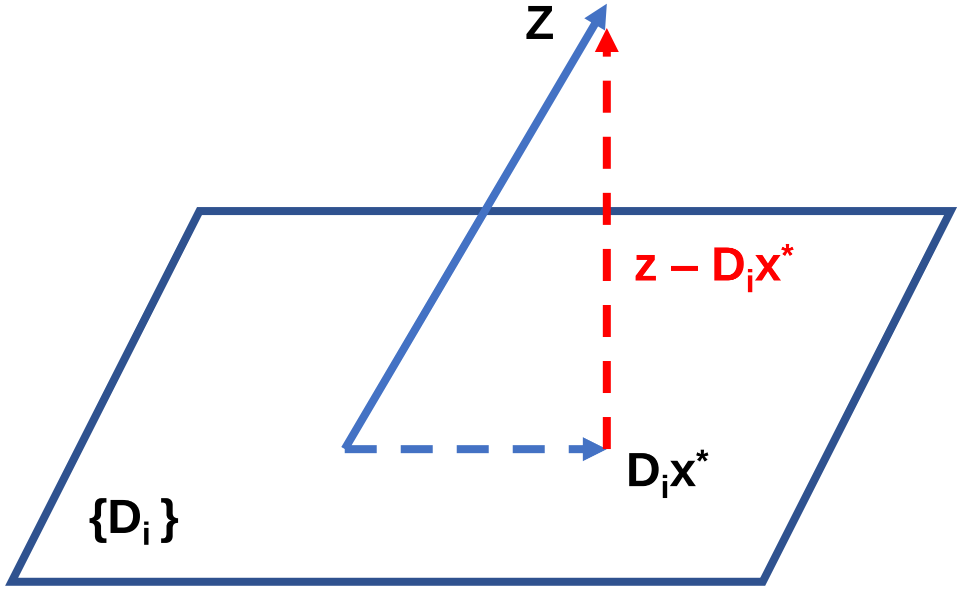
$$\rho = \ min_x||D_5x - z||$$

Z is the vector form of test image, D5 is the digit 5 image formed from the training set,

The digit matrix Di yields the smallest ρ is the class the test image belongs to

$$\boxed{\rho = min_x||D_ix - z||}$$

Solve a Least Square Problem



$$\boxed{\rho = min_x \|D_i x - z\|}$$

$$D_i^T (D_i x^* - z) = 0$$

b – D$_i$x* is what we are solving for each digit

Solve a Least Square Problem (SVD)

$$\rho = min_x \|Dx - z\|$$

$$\Downarrow$$

Leading r column of U spans the column space of D, given $D = U\Sigma V^T$ decomposition

$$\rho = min_y \|U_r y - z\|$$

$$\Downarrow$$

$$U_r^T U_r y = U_r^T z \Rightarrow \boxed{\rho = \|U_r U_r^T z - z\|}$$

Solve a Least Square Problem (NMF)

$$\begin{tikzcd}
\rho = min_x ||Dx - z|| \arrow[d, "\begin{matrix} \mathrm{D} = \mathrm{WH} \\ \mathrm{y} = \mathrm{Hx} \end{matrix}"] \\
\rho = min_y ||Wy - z|| \arrow[d] \\
\boxed{\rho = min_y ||W_r y - z||}
\end{tikzcd}$$

The first r column of W spans the column space of D

Result & Discussion

- **Accuracy**

- **Stability**

Accuracy- SVD

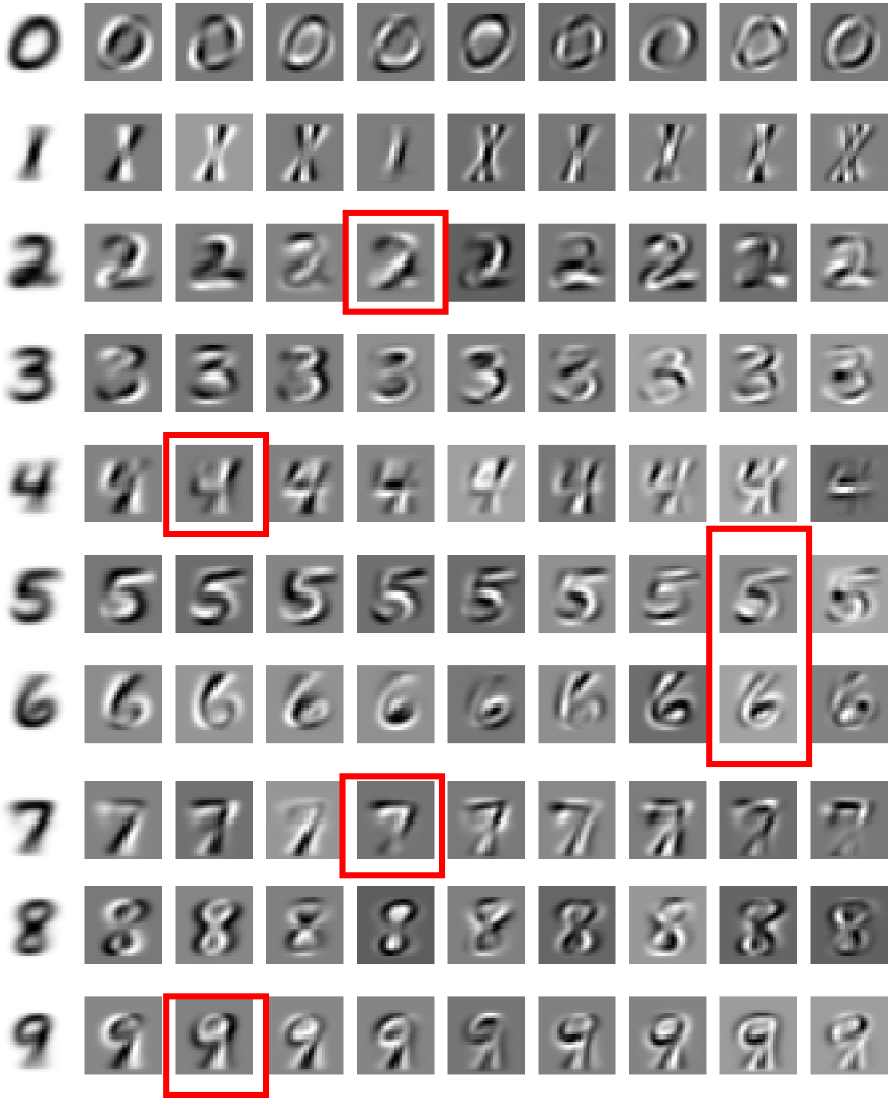
SVD setting is as following: 60,000 training images, 10,000 test images, truncated dimension is set to be k = 10 according to literature

True Value

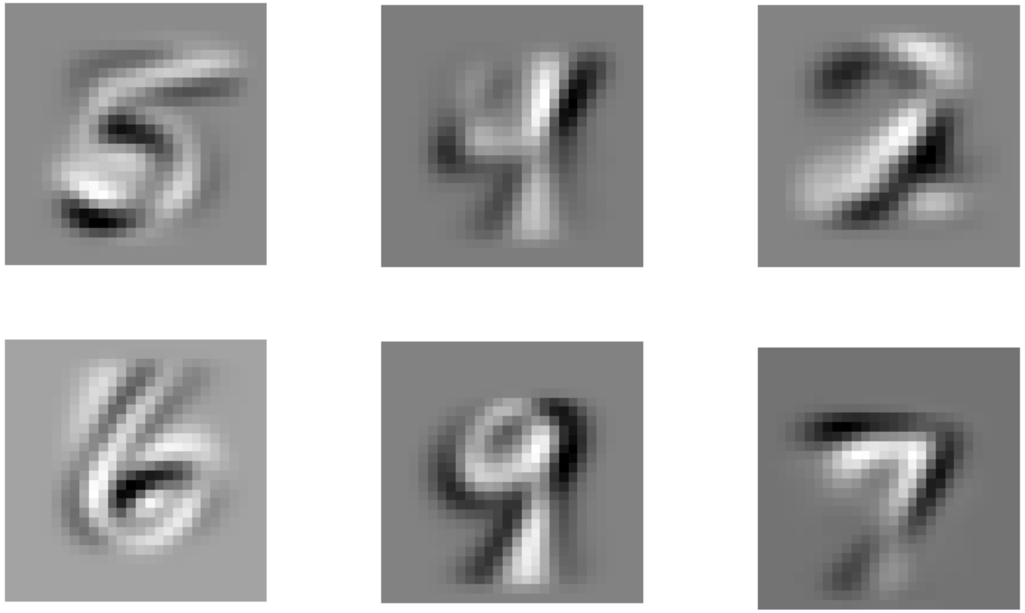
Predicted Value

	0	1	2	3	4	5	6	7	8	9	Accuracy %
0	968	0	15	2	3	4	13	0	7	6	95.088
1	0	1129	4	0	10	1	5	16	6	9	95.678
2	1	3	957	5	4	2	0	14	8	4	95.892
3	0	1	9	952	0	31	0	1	17	5	93.701
4	0	1	6	0	939	1	5	8	4	21	95.427
5	1	1	0	15	0	820	8	0	16	6	94.579
6	7	0	5	0	5	9	925	0	5	1	96.555
7	1	0	12	11	1	2	0	960	5	9	95.904
8	2	0	21	21	2	13	2	3	894	7	92.643
9	0	0	3	4	18	9	0	26	12	941	92.892
Successful rate %	98.775	99.47	92.73	94.26	95.62	91.93	96.56	93.39	91.79	93.26	

Leading 10 Singular Vector of Digits (SVD)



1. The first singular vector shows the feature of the digit
2. The second to tenth vector still shows some features but not as much as the first one
3. High order singular vectors of different digits share some similarities.



Accuracy- NMF

NMF setting is as following: 60,000 training images, 10,000 test images, truncated dimension is set to be k = 10 according to literature

True Value

Predicted Value

	0	1	2	3	4	5	6	7	8	9	Accuracy %
0	967	0	18	3	4	9	17	2	13	7	92.981
1	2	1129	17	1	9	5	4	16	13	5	94.005
2	2	4	944	3	3	2	0	16	6	4	95.935
3	0	0	9	942	0	34	0	2	9	6	94.012
4	0	1	5	0	919	1	0	12	5	25	94.938
5	0	0	0	18	1	792	8	0	13	5	94.624
6	6	1	3	0	6	11	925	0	14	0	95.756
7	1	0	12	10	2	3	0	947	9	19	94.417
8	2	0	22	26	5	28	3	2	881	9	90.082
9	0	0	2	7	33	7	1	31	11	929	90.989
Successful rate %	98.674	99.471	91.473	93.267	93.585	88.789	96.555	92.121	90.451	92.071	

NMF algorithms run-time/ accuracy

	Run Time (s)	Accuracy (%)
Multiplicative Update	115	93.71
ALS + Projected Gradient	144	93.47
SNMF/L	5400	93.17

MU: easy to implement, monotonicity is not guaranteed, convergence rate is not the fastest

ALS : nonnegativity constrain needs to implement, converge might be faster than MU

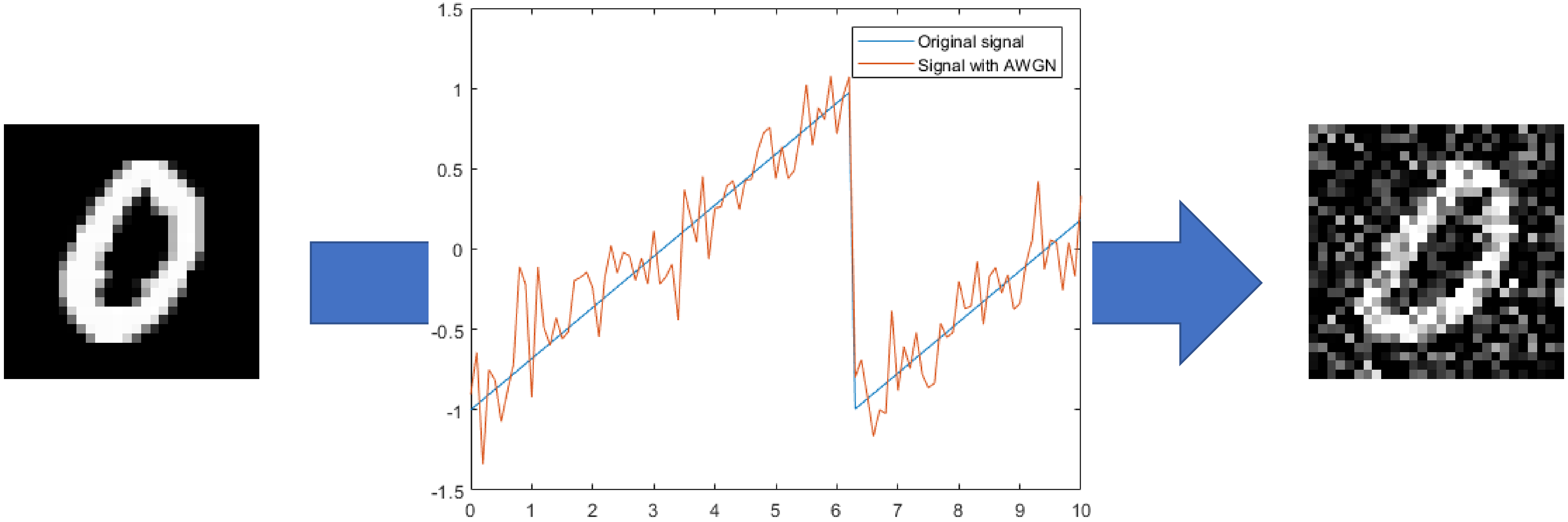
SNMF/L : control sparsity of the produced matrix by the factorization

Run on personal computer, i5-3210 M cpu @ 2.50 GHz, 8MHz ram, iteration times for algorithms are set to be 100 times

Stability SVD/NMF

After the SVD/NMF algorithms are trained as a classifier, how well will the classifier perform when the noise is added to the test data?

Let us use a simple gaussian noise to start with.



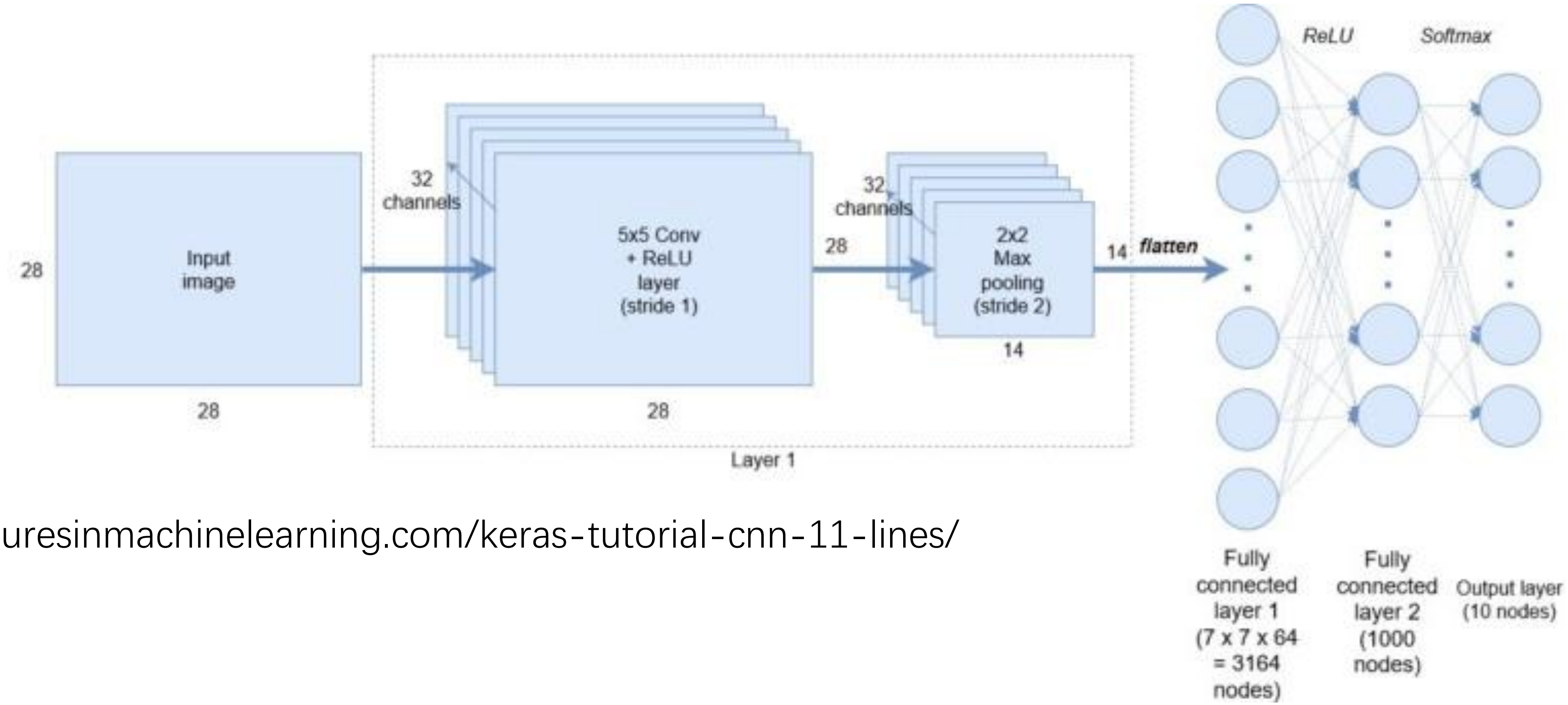
https://www.mathworks.com/help/comm/ref/awgn.html

Stability SVD/NMF

	Noise Free Test Set	Noise Added Test Set
SVD	**94.85%**	**93.45%**
NMF-ALS	**93.47%**	**92.94%**

Even with the perturbation of the noise, SVD and NMF can still work as a reliable classifier. The accuracy is not significantly degraded by the noise.

A little bit CNN, accuracy and stability


28
Input image
28
32 channels
5x5 Conv + ReLU layer (stride 1)
28
28
32 channels
2x2 Max pooling (stride 2)
14
14
flatten
Layer 1
ReLU
Softmax
Fully connected layer 1 (7 x 7 x 64 = 3164 nodes)
Fully connected layer 2 (1000 nodes)
Output layer (10 nodes)

http://adventuresinmachinelearning.com/keras-tutorial-cnn-11-lines/

1. Noisy Free data test: ~99%
2. Noise data test: ~90%
3. CNN prediction accuracy degraded by perturbation.

Conclusion

1. SVD can achieve ~ 95% accuracy in MNIST data classifier. The error rate can be potentially reduced by training in pair strategy and introducing “tie-breaker” algorithm in the training pair.

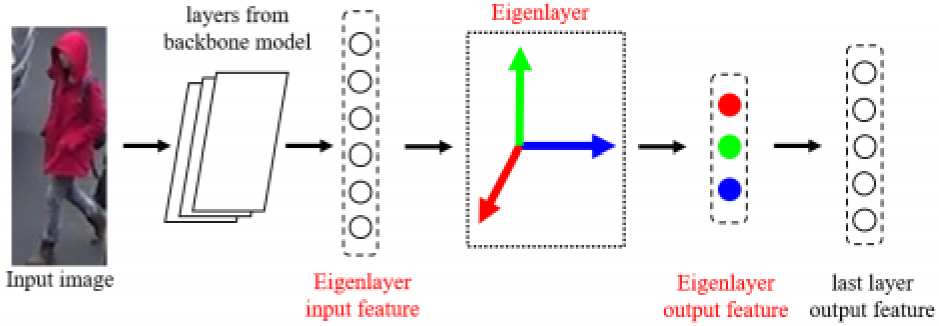
2. NMF can yield ~ 93% accuracy in MNIST data classifier. And NMF will produce a nonnegative base for the least square problem.

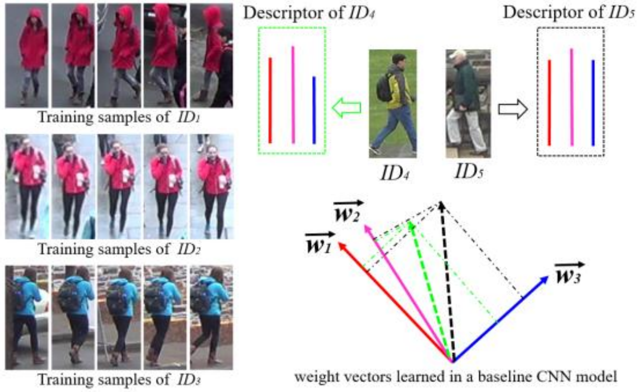
3. Both SVD and NMF can resist the perturbation of the noise. But CNN’s prediction accuracy is jeopardized by the noise and can be potentially restored by using new parameters or SVD/NMF as the first prescreening layer.

Outlook

1. SVD/NMF can be combined with CNN in a closer way. MINIST raw image can be written as a coefficient of the base vectors produced by SVD or NMF. And a 1d CNN can be used for the classification.

2. SVD can be added as a layer in the CNN (before the FC layer)


layers from backbone model
Eigenlayer
Input image
Eigenlayer input feature
Eigenlayer output feature
last layer output feature



arXiv:1703.05693 [cs.CV]