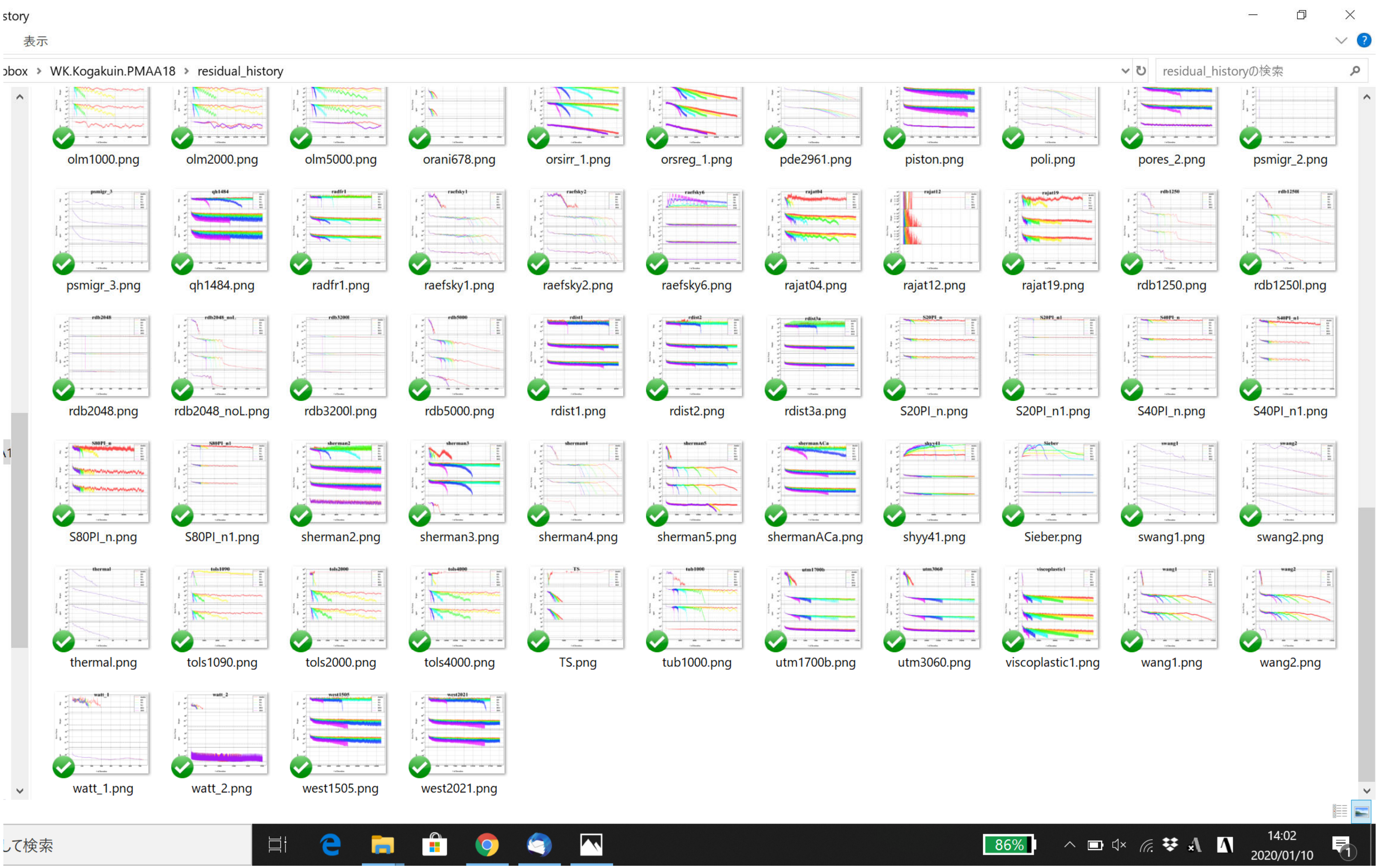
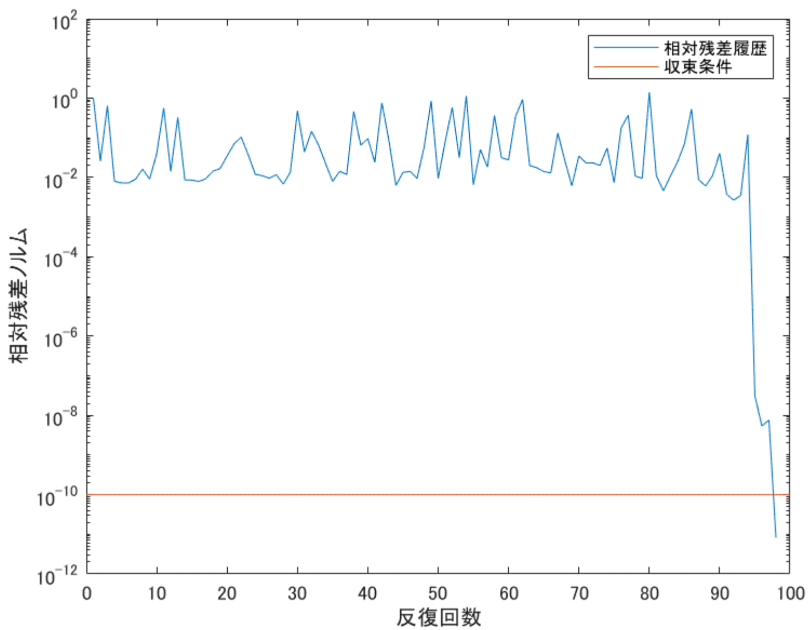
Predicting the convergence of BiCG method from grayscale matrix images

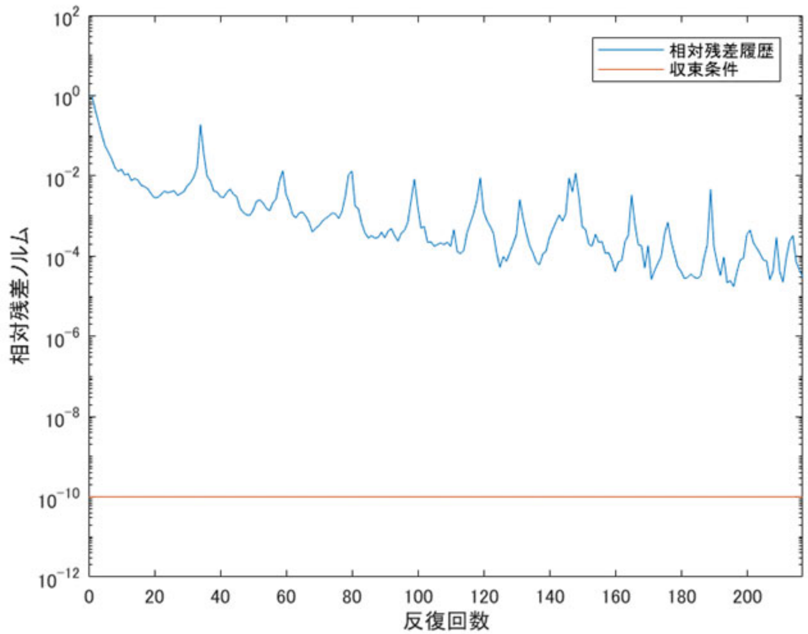
Ryo Ota and Hidehiko Hasegawa (U. of Tsukuba)

Image-net.org


story
表示
›box › WK.Kogakuin.PMAA18 › residual_history
residual_historyの検索
olm1000.png
olm2000.png
olm5000.png
orani678.png
orsirr_1.png
orsreg_1.png
pde2961.png
piston.png
poli.png
pores_2.png
psmigr_2.png
psmigr_3.png
qh1484.png
radfr1.png
raefsky1.png
raefsky2.png
raefsky6.png
rajat04.png
rajat12.png
rajat19.png
rdb1250.png
rdb1250l.png
rdb2048.png
rdb2048_noL.png
rdb3200l.png
rdb5000.png
rdist1.png
rdist2.png
rdist3a.png
S20PI_n.png
S20PI_n1.png
S40PI_n.png
S40PI_n1.png
S80PI_n.png
S80PI_n1.png
sherman2.png
sherman3.png
sherman4.png
sherman5.png
shermanACa.png
shyy41.png
Sieber.png
swang1.png
swang2.png
thermal.png
tols1090.png
tols2000.png
tols4000.png
TS.png
tub1000.png
utm1700b.png
utm3060.png
viscoplastic1.png
wang1.png
wang2.png
watt_1.png
watt_2.png
west1505.png
west2021.png
して検索
86%
14:02
2020/01/10

Converge vs Not-converge





name	Dim.	Itr.	Res.
08blocks	300	98	8.23E-12

name	Dim.	Itr.	Res.
ex1	217	217	3.04E-05

Image recognition

- Get attributes (Species, type, name) from Images
- Predict Converge/Not-converge from sparse-matrix images
- Indirect : Convergence history is not used

Sparse matrix and BiCG

- Many zero elements in sparse matrix
- Infinite combination of values
  - Location of nonzero elements and values
- Unknown property of matrix
- Converge/Not-converge is clear
- Convergence of BiCG and CG is NOT-monotone (irregular)

Grayscale images

- MNIST (Handwritten digits) http://yann.lecun.com/exdb/mnist/
- Gray scale
  - 0 black
  - 255 white
- Image size
  - 28*28 pixels
  - 56*56 pixels
  - 112*112 pixels
  - 224*224 Pixels
- JPEG

Conversion

- Based on SuiteSparse Matrix Collection
- Absolute value (NO sign)
- Pick a maximum element of each block
- Gray scale
  - 0 (black) for zero
  - 1 : log10(abs(val)) =< median-sigma
  - 128 : median of log10(abs(val))
  - 255 (white) : log10(abs(val)) >= median+sigma

How to make grayscale image

1. Take **absolute value**

2. s = Ceil[$\frac{\text{Dimension of Matrix}}{Image\ Size}$]

3. Divide Matrix A into blocks whose size is s × s

4. Set **maximum** value in each block to an element of small_A

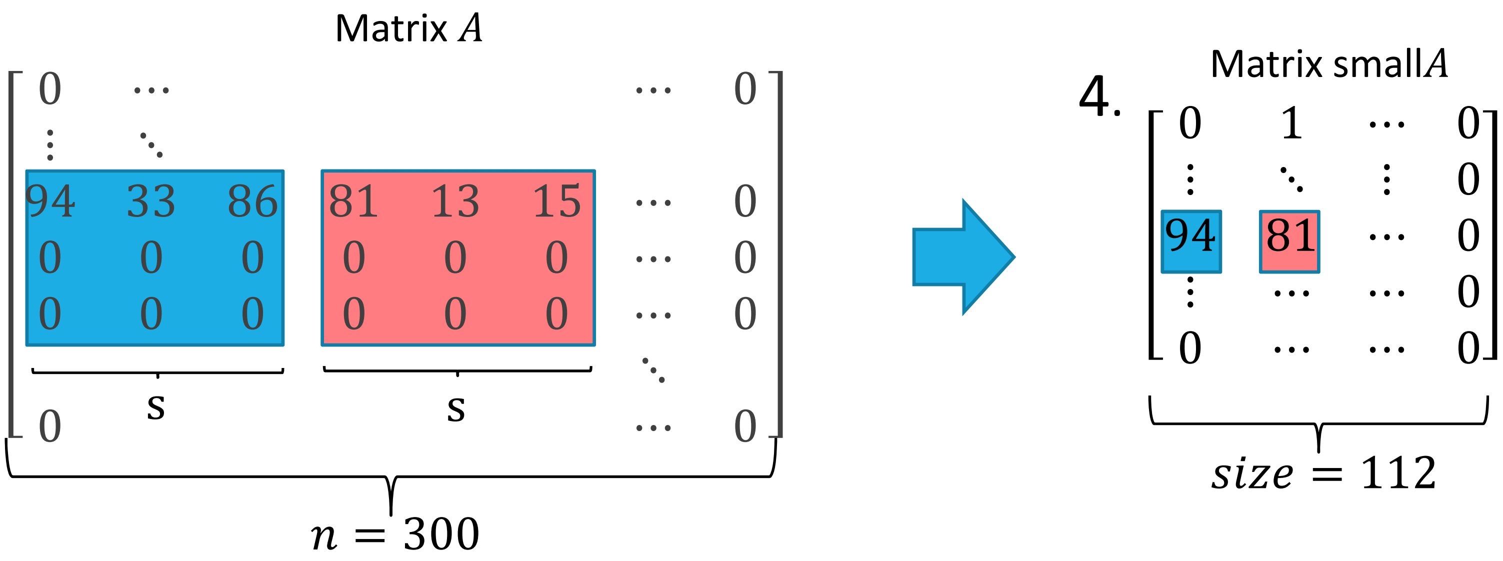
5. Compute median and std of log10(value) for nonzero elements of $\text{small\_}A$ ($\boldsymbol{x}$)

6. Set grayscale values (128 to the median of $\boldsymbol{x}$)

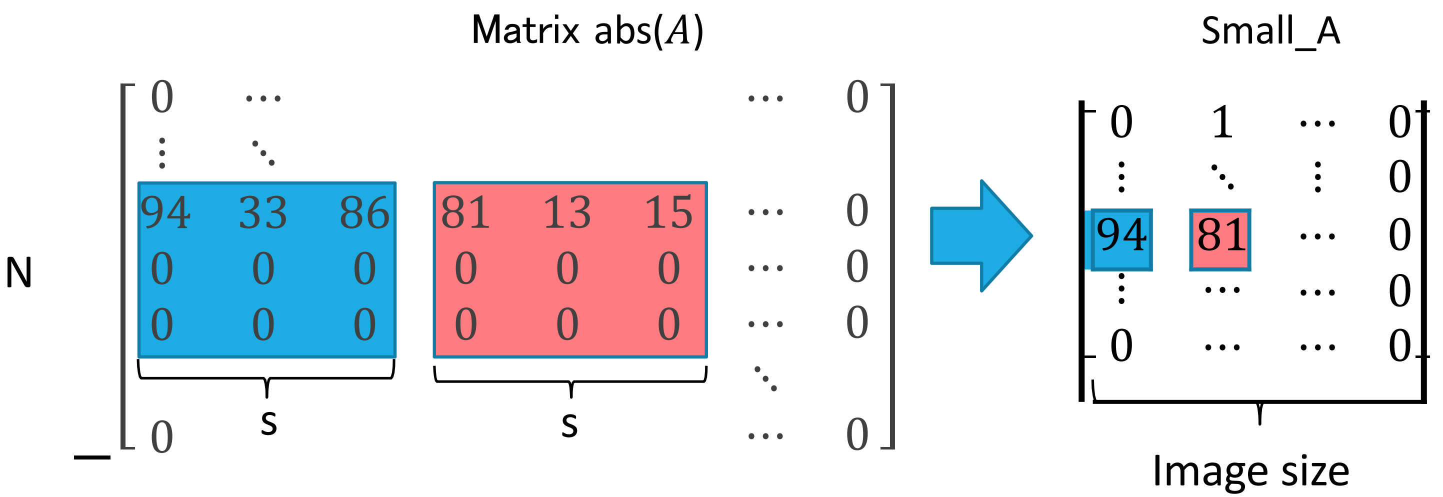
7. Resize image

Example: 08blocks (n=300 to 112^{2}pixels)

1. $A = \text{fabs}(A)$
2. Block size $s = ceil\left(\frac{N}{size}\right) = ceil\left(\frac{300}{112}\right) = 3$
3.
4.



Conversion Process



$\boldsymbol{x} = \log_{10}(nonzeros(\text{Small } A))$

$$1 + \text{ceil}\left(\frac{(256-2)\left(x_i - \left(Me(\boldsymbol{x}) - \sigma(\boldsymbol{x})\right)\right)}{2\sigma(\boldsymbol{x})}\right)$$

Set grayscale value 128 to median of $\boldsymbol{x}$

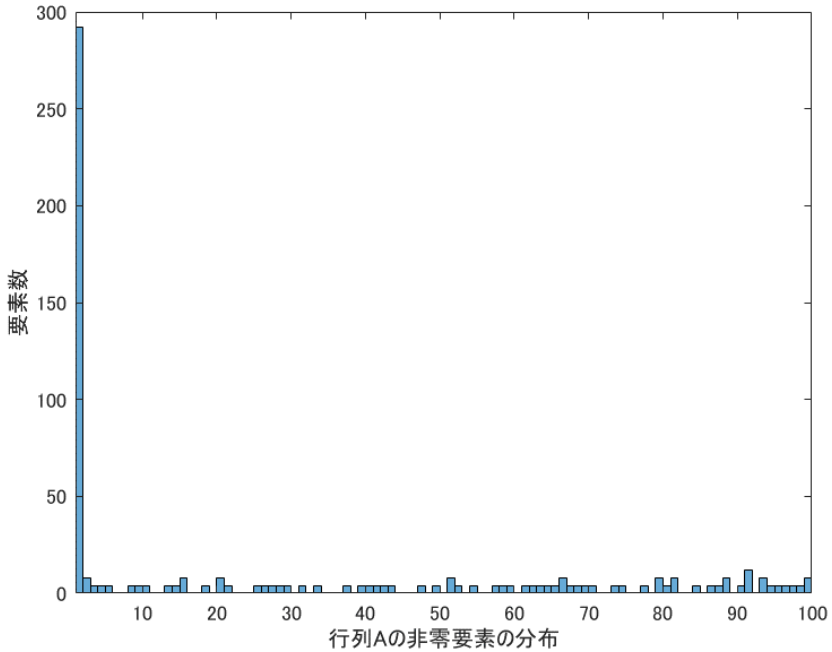
n: dimension
s : ceil(N/image size)

$\sigma(\boldsymbol{x})$: standard deviation of $\boldsymbol{x}$
$Me(\boldsymbol{x})$: median of $\boldsymbol{x}$

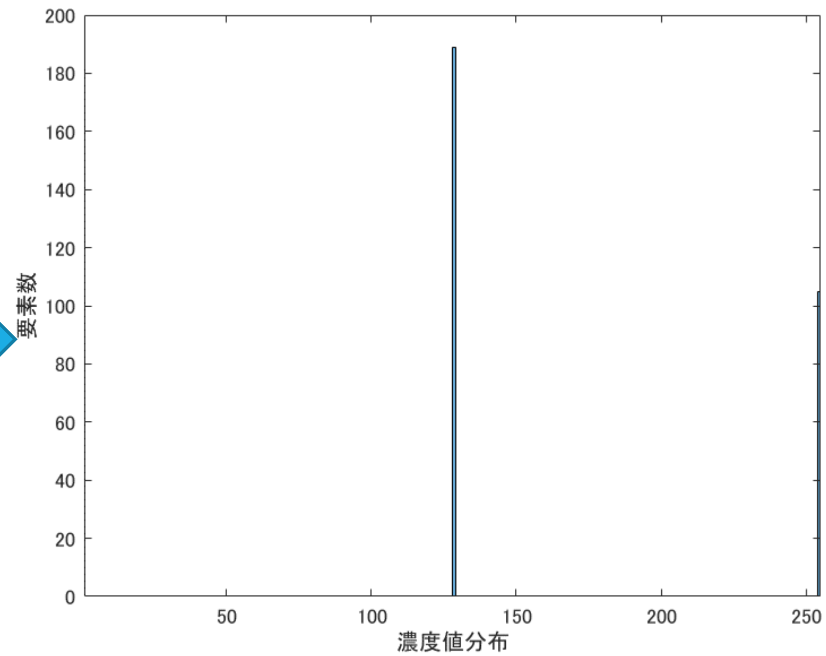
Target matrices

- SuiteSparse Matrix Collection
- Real, square, non-symmetric
- 875 matrices of the 982 matrices
- Dimension : 5 – 445,315
- Sparsity (%) : $6*10^{-6}$ – 76

Conversion to grayscale value 08blocks



Distribution of values



Distribution of grayscale values

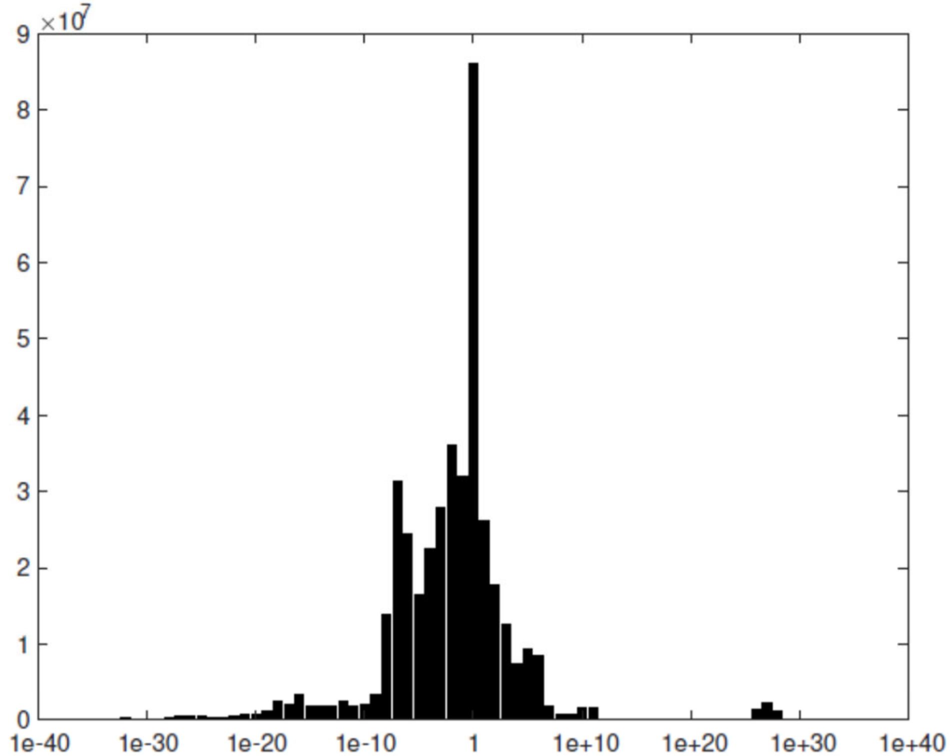


Fig. 2. Distribution of absolute value of nonzero elements

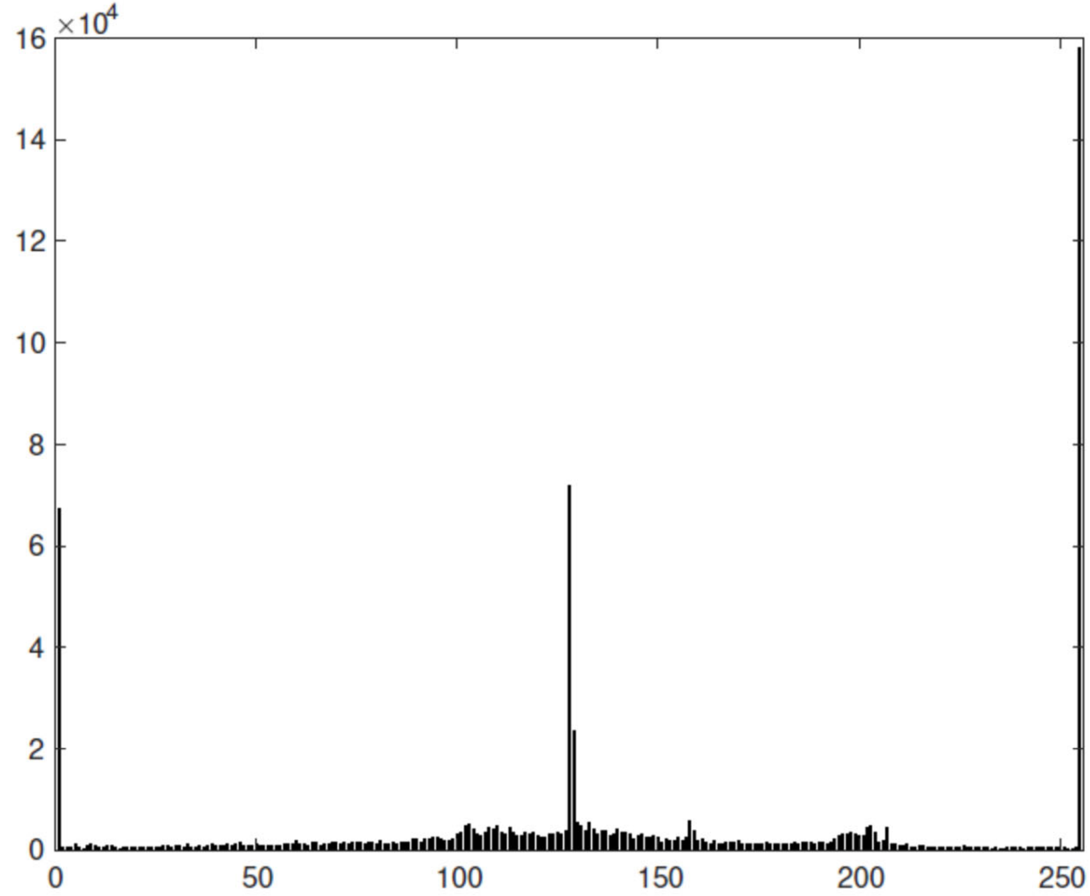


Fig. 3. Distribution of grayscale values for nonzero elements

BiCG : BiConjugate Gradient method

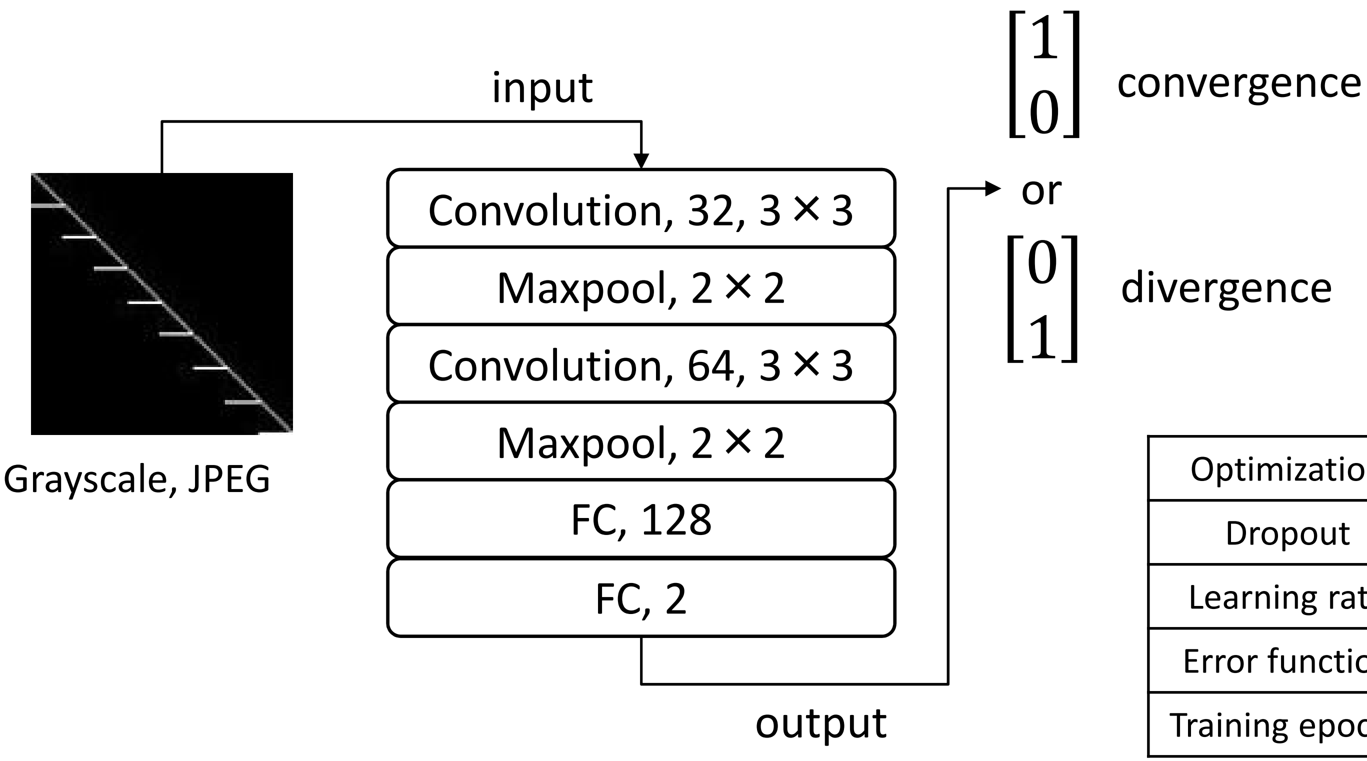
$\boldsymbol{x_0}$ is initial guess ,
Compute $\boldsymbol{r_0} = \boldsymbol{b} - A\boldsymbol{x_0}$,
Choose $\boldsymbol{r_0^*}$ such that $(\boldsymbol{r_0^*}, \boldsymbol{r_0}) \neq 0$,
Set $\boldsymbol{p_0} = \boldsymbol{r_0}$ and $\boldsymbol{p_0^*} = \boldsymbol{r_0^*}$,
For $k = 0,1,2 \ldots$, **until** $\|\boldsymbol{r}_k\|_2 \leq \varepsilon_{TOL}\|\boldsymbol{b}\|_2$ $\boldsymbol{do}$:

$$\boldsymbol{q_k} = A\boldsymbol{p_k}, \quad \boldsymbol{q_k^*} = A^T\boldsymbol{p_k^*},$$
$$\alpha_k = \frac{(\boldsymbol{r_k^*}, \boldsymbol{r_k})}{(\boldsymbol{p_k^*}, \boldsymbol{q_k})},$$
$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k \boldsymbol{p_k},$$
$$\boldsymbol{r_{k+1}} = \boldsymbol{r_k} - \alpha_k \boldsymbol{q_k}, \quad \boldsymbol{r_{k+1}^*} = \boldsymbol{r_k^*} - \alpha_k \boldsymbol{q_k^*},$$
$$\beta_k = \frac{(\boldsymbol{r_{k+1}^*}, \boldsymbol{r_{k+1}})}{(\boldsymbol{r_k^*}, \boldsymbol{r_k})},$$
$$\boldsymbol{p_{k+1}} = \boldsymbol{r_{k+1}} + \beta_k \boldsymbol{p_k}, \quad \boldsymbol{p_{k+1}^*} = \boldsymbol{r_{k+1}^*} + \beta_k \boldsymbol{p_k^*},$$

End For

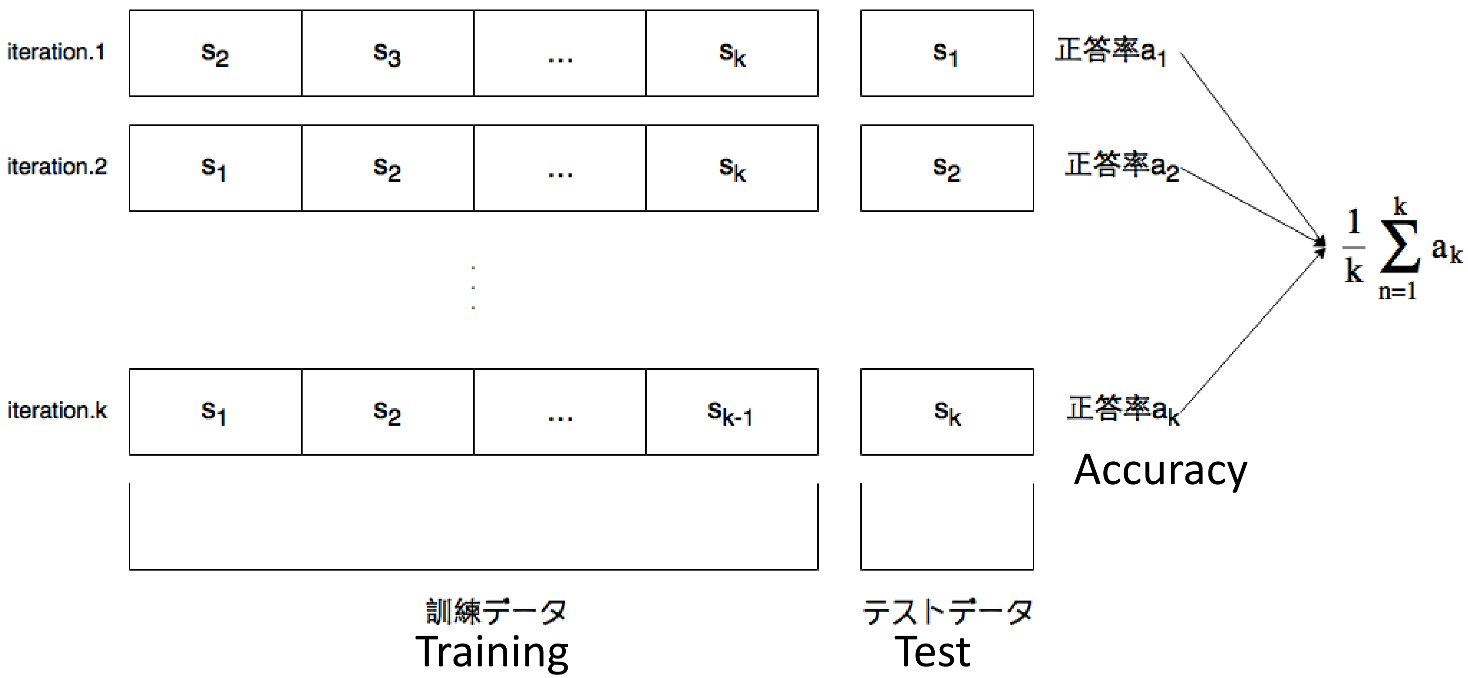
	condition
Rel. res. Norm: ε_{TOL}	$\varepsilon_{TOL} = 1.0 \times 10^{-6}$
initial $\boldsymbol{x_0}$	$\boldsymbol{x_0} = \boldsymbol{0}$
Sol. $\boldsymbol{x}$	$\boldsymbol{x} = [1, -1, \ldots, (-1)^{n-1}]^T$
r.h.s. $\boldsymbol{b}$	$\boldsymbol{b} = A[1, -1, \ldots, (-1)^{n-1}]^T$
$\boldsymbol{r^*}_0$	$\boldsymbol{r^*}_0 = \boldsymbol{r_0}$
limit	600 seconds or n
precision	double

Convolutional Neural Network



Optimization	Adam method
Dropout	25%
Learning rate	1.0×10^{-3}
Error function	Sigmoid
Training epochs	30

5-fold cross-validation (k=5)



Accuracy := (TF+TP)/ All

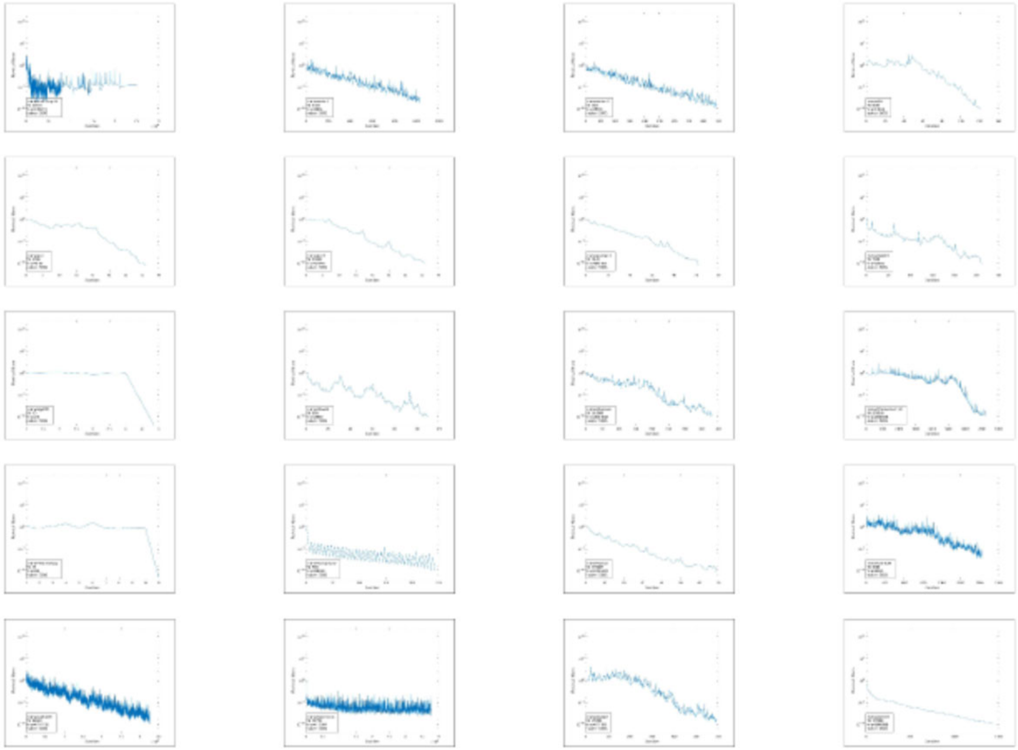
	Prediction		
Real		Non-convergence	Convergence
	Non-convergence	TN	FN
	Convergence	FP	TP

T: True vs F: False;

N: Negative vs P: Positive

TN: True – Negative (Not-converge, Correct)

FN: False – Negative (Not-converge, Incorrect : Will converge)



FP: False – Positive (Converge, Incorrect : Will not converge)

TP: True – Positive (Converge, Correct)

Test datasets

- F (equal) 470 matrices
  - Converge 235
  - Not-converge 235
  - Each group has 47 Converge and 47 Not-converge
- H (full) 875 matrices
  - Converge 235
  - Not-converge 640
  - Each group has 47 Converge and 128 Not-converge
- F_i is a subset of H_i (i=1, 2, 3, 4, 5)

Table 3. Accuracy (%) of dataset F and H

Size (pixels)	Data set	1	2	3	4	5	Average
28×28	F	65.9	81.9	76.5	74.4	75.5	74.9
	H	78.9	82.9	84.6	85.7	85.1	83.4
56×56	F	68.0	83.0	83.0	80.9	79.8	78.9
	H	81.7	82.9	86.9	84.0	85.1	84.1
112×112	F	67.0	85.1	76.5	77.6	72.3	75.7
	H	82.3	84.0	80.6	85.7	80.0	82.5
224×224	F	69.1	84.0	71.2	79.7	73.0	75.5
	H	79.4	82.3	83.4	85.1	82.8	82.6

Table 4. Classification result (correct answer rate) of dataset F, 56 × 56 pixels

	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	Total (%)
TN	25 (53.1)	36 (76.5)	38 (80.8)	39 (82.9)	37 (78.7)	175 (74.4)
FN	22	11	9	8	10	60
FP	8	5	7	10	9	39
TP	39 (82.9)	42 (89.3)	40 (85.1)	37 (78.7)	38 (80.8)	196 (83.4)
Total	94	94	94	94	94	470

Converge 47 vs Not-converge 47 in each group

Table 5. Classification result (correct answer rate) of dataset H, 56 × 56 pixels

	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	Total (%)
TN	111 (86.7)	114 (89.0)	119 (92.9)	113 (88.2)	122 (95.3)	579 (90.4)
FN	17	14	9	15	6	61
FP	15	16	14	13	20	78
TP	32 (68.0)	31 (65.9)	33 (70.2)	34 (72.3)	27 (57.4)	157 (66.8)
Total	175	175	175	175	175	875

Converge 47 vs Not-converge 128 in each group

Computation time (Seconds)

Dataset F, Epochs=30

	28×28	56×56	112×112	224×224
Training	9.77	15.0	30.2	87.9
Test	2.55	2.68	2.72	2.74

OS	ubuntu16.04LTS
CPU	Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50GHz×4
GPU	Tesla K40m
memory	64GB

Results

- Dataset H (all data) is good accuracy.
- **The correct answer rate depends on the number of samples.**
- Smaller image has good accuracy.
- Average accuracy exceeds 80%.
- Computation time is not a problem.

Future work

- ~~Changing Condition of Convergence~~
- Other mapping from value to grayscale value
- Modifying distribution of Grayscale values
- Other iterative methods
- Multiple values (Converge/Nearly-converge/Not-converge)
- Standardization of matrix

Accuracy (%) of dataset I (10^{-10})

size	1	2	3	4	5	Average
28×28	84.0	86.2	86.2	86.8	81.7	85.0
56×56	87.4	86.8	86.2	86.8	81.7	85.7
112×112	86.2	86.8	88.0	88.0	81.7	86.1
224×224	86.8	82.8	86.8	86.8	81.7	85.0

Converge : 176 (20%) vs Not-converge : 699 (80%)

Classification result (correct answer rate) of Dataset I (10^{-10}), 28*28 pixels

	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	Total (%)
TN	129 (92.1)	129 (92.1)	135 (96.4)	127 (90.7)	130 (92.8)	650 (93.5)
FN	17	13	19	10	23	82
FP	11	11	5	13	9	49
TP	18 (51.4)	22 (62.8)	16 (45.7)	25 (71.4)	13 (36.1)	94 (53.4)
Total	175	175	175	175	175	875

Converge 35-36 vs Not-converge 139-140 in each group

Summary

- **The number of samples is Key!**
- A simple method is used
  - Small grayscale images 28*28, 56*56, 112*112, and 224*224 pixels
  - Absolute value (sign is omitted)
  - Maximum element in each block
  - Less than 1,000 samples (Converge is 27% for 10^{-6})
- Good average accuracy
  - Over 80%
- Small computation time
- Easiness of prediction depends on origin-field-type of matrix (in progress)

However

I can not believe…