Homework 3
Ma Xiaoqi
308932

Mod 3

Exercise 1:

Answer:

$$P(b|A,X) = \prod_{i=1}^{n} P(b_i|a_i,X)$$

$$\therefore -\log P(b|A,X) = -\log \prod_{i=1}^{n} P(b_i|a_i,X)$$

$$= \sum_{i=1}^{n} (-\log P(b_i|a_i,X))$$

$$\because \log P(b_i=1|a_i,X) = a_i^T x_1 - \log S$$

$$\vdots$$

$$\log P(b_i=c|a_i,X) = a_i^T x_c - \log S$$

$$\therefore \sum_{i=1}^{n} (-\log P(b_i|a_i,X))$$

$$= \sum_{i=1}^{n} (\log S - a_i x_{b_i}) \quad b_i \in \{1,2,\dots C\}$$

$$= \sum_{i=1}^{n} (\log \sum_{k=1}^{C} e^{a_i^T x_k} - a_i^T x_{b_i}) \quad b_i \in \{1,2,\dots C\}$$

$$\hat{x}_{ML} = \arg\min_x \left\{ f(x) \mid f: R^{d\times c} \to R,\ f(x) = \sum_{i=1}^{n} \left(\log \sum_{k=1}^{C} e^{a_i^T x_k} - a_i^T x_{b_i} \right),\ b_i \in \{1,2,\dots C\} \right\}$$

Exercise 2:

Answer:

$$f(x) = \sum_{i=1}^{n} (\log \sum_{k=1}^{C} e^{a_i^T x_k} - a_i^T x_{b_i})$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{C} e^{a_i^T x_k} - \sum_{i=1}^{n} a_i^T x_{b_i}$$

For the part $\sum_{i=1}^{n} \log \sum_{k=1}^{C} e^{a_i^T x_k} = G(X)$

$$\therefore \nabla (\log \sum_{k=1}^{C} e^{a_i^T x_k}) = \frac{1}{\sum_{k=1}^{C} e^{a_i^T x_k}}$$

$$\therefore \nabla G(X) = \left(\sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{C} e^{a_i^T x_k}} \right) \left[e^{\sum_{i=1}^{n} a_i^T \sum_{k=1}^{C} x_k} \right]' \quad \text{(For each } e^{a_i^T \sum_{k=1}^{C} x_k}\text{, they need to multiply } \frac{1}{S}\text{)}$$

$$= Z \left[e^{(Ax)} \right]'$$

$$= A^T Z e^{Ax}$$

For the part $\sum_{i=1}^{n} a_i^T x_{b_i} = H(X)$

$$H(X) = \underbrace{\begin{bmatrix} \leftarrow a_1^T \rightarrow \\ \leftarrow a_2^T \rightarrow \\ \vdots \\ \leftarrow a_n^T \rightarrow \end{bmatrix}_{n\times d}}_{A} \cdot \underbrace{\begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ x_{b_1} & x_{b_2} & \cdots & x_{b_n} \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{d\times n}}_{H}$$

$$\therefore X = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_c \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1c} \\ x_{21} & x_{22} & & \vdots \\ \vdots & \vdots & \cdots & \\ x_{d1} & x_{d2} & \cdots & x_{dc} \end{bmatrix}_{d\times c}$$

$$\therefore H = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1c} \\ x_{21} & x_{22} & & \vdots \\ \vdots & \vdots & & \\ x_{d1} & x_{d2} & & x_{dc} \end{bmatrix} \cdot \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ y_1 & y_2 & \cdots & y_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{c\times n}$$

y_i is one-hot-encoding which extract the x_{b_i} according to b_i

$$\therefore H(X) = \cancel{AY^TX}\ AXY^T$$

$$\frac{\partial H'(X)}{\partial X} = \frac{\partial (AXY^T)}{\partial X} = A^TY$$

$$\therefore \nabla f(X) = A^T(\Sigma \exp(AX) - Y)$$

Exercise 4

(a) $\because$ ~~For D_2 E~~

$$a_i a_i^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} [a_1\ a_2 \cdots a_n] = \begin{bmatrix} a_1^2 & a_1a_2 & & a_1a_n \\ & a_2^2 & & a_2a_n \\ & & \ddots & \\ a_na_1 & \cdots & & a_n^2 \end{bmatrix}$$

$\therefore Trace(a_i a_i^T) = a_1^2 + a_2^2 + \cdots a_n^2 = \|a_i\|_2^2$

$\because$ We have known $a_i a_i^T$ is a rank-1 matrix,

That means matrix $a_i a_i^T$ have the same $n-1$ engivalues: 0

$\therefore$ ~~λmax~~ $\therefore \lambda_n = Trace(a_i a_i^T) - (n-1)\cdot 0 = \|a_i\|_2^2$

$\because \|a_i\|_2^2 \geq 0$

$\therefore \lambda max(a_i a_i^T) = \|a_i\|_2^2$

(b) $\because$ We have known $\nabla^2 f(X) = \sum_{i=1}^{n} \Sigma_i \otimes a_i a_i^T$

$\because$ For $\Sigma_i \geq 0$, $a_i a_i^T \geq 0$, we have ~~$\lambda max(D \otimes E) =$~~

$\lambda max(\Sigma_i \otimes a_i a_i^T) = \lambda max(\Sigma_i) \cdot \lambda max(a_i a_i^T)$

$$\Sigma_i = \begin{bmatrix} \sigma_{i1}(1-\sigma_{i1}) & -\sigma_{i1}\sigma_{i2} & \cdots & -\sigma_{i1}\sigma_{ic} \\ -\sigma_{i2}\sigma_{i1} & \sigma_{i2}(1-\sigma_{i2}) & & \vdots \\ \vdots & & \ddots & \\ -\sigma_{ic}\sigma_{i1} & & & \sigma_{ic}(1-\sigma_{ic}) \end{bmatrix}$$

$\because \sigma_{ij} = \frac{e^{a_i^T x_j}}{\sum_{k=1}^{c} e^{a_i^T x_k}}$ $\therefore 0 \le \sigma_{ij} \le 1$

$\therefore \sum_j |(\Sigma_i)_{kj}|$

$= |\sigma_{ik}(1-\sigma_{ik})| + \sum_{j \neq k} |-\sigma_{ik}\sigma_{ij}|$

$= 2\sigma_{ik}(1-\sigma_{ik}) + \sigma_{ik}(1-\sigma_{ik})$

$= 2\sigma_{ik}(1-\sigma_{ik})$

$\because 0 \le \sigma_{ik} \le 1$ $\therefore \sigma_{ik}(1-\sigma_{ik}) \le \frac{1}{4}$

$\therefore \max_k \sum_j |(\Sigma_i)_{kj}| \le 2 \times \frac{1}{4} = \frac{1}{2}$ $\therefore \lambda_{max}(\Sigma_i) \le \frac{1}{2}$

(F means Frobenius norm)

$\therefore \lambda_{max}(\Sigma_i \otimes a_i a_i^T)$

$\le \frac{1}{2} \lambda_{max}(a_i a_i^T) = \frac{1}{2} \|a_i\|_2^2$

$\therefore \lambda_{max}(\nabla^2 f(x))$

$\le \sum_{i=1}^{n} \frac{\|\vec{a}_i\|_2^2}{2}$

$= \frac{\|A\|_F^2}{2}$

$\begin{cases} \because \|A\|_F^2 = \left(\sum_{i=1}^{n} \sum_{j=1}^{d} |a_{ij}|^2 \right) \\ \|a_i\|_2^2 = \sum_{j=1}^{d} |a_{ij}|^2 \\ \therefore \|A\|_F^2 = \sum_{i=1}^{n} \|a_i\|_2^2 \end{cases}$

$\therefore$ we can choose $L = \frac{\|A\|_F^2}{2}$

Exercise 5 :

Given $g: R^d \to R$ $g(x) := \|x\|_1$

and the proximal operator of g is:

$$Prox_g(z) := \arg\min_{y \in R^d} \{ g(y) + \frac{1}{2}\|y-z\|_2^2 \}$$

$\therefore$ the proximal operator of λg is

$$Prox_{\lambda g}(z) = \arg\min_{y \in R^d} \{ \lambda \|y\|_1 + \frac{1}{2}\|y-z\|_2^2 \}$$

Then considering the gradient of y

$\nabla y = \lambda \text{sign}(y) + (y - z) \quad (y_i \neq 0)$

$y \in R^d \quad z \in R^d$

Considering each entry i $i \in [1, d]$ and make $\nabla y = 0$

If $y_i > 0$: $y_i \Rightarrow z_i + \lambda = 0 \quad y_i = z_i - \lambda \quad z_i > \lambda$

If $y_i < 0$: $y_i - z_i - \lambda = 0 \quad y_i = z_i + \lambda \quad z_i < -\lambda$

$\therefore$ ~~$z_i > |y_i|$ and~~ It satisfys all the entry i

$\therefore$ ~~[illegible]~~ So if $y_i = 0$: We have $\underset{y}{\text{argmin}} = \begin{cases} z_i - \lambda & z_i > \lambda \\ z + \lambda & z < -\lambda \end{cases}$

If $y_i = 0$, $|y_i|$ doesn't have gradient at $y_i = 0$, so we calculate the subgradient. For $y_i \in [-1, 1]$:

$\lambda y_i + (0 - z) = 0 \quad \therefore -\lambda \leq z \leq \lambda \quad \therefore |z| \leq \lambda \quad |z| - \lambda \leq 0$

$$\therefore \text{argmin}\left\{g(y) + \frac{1}{2}\|y - z\|_2^2\right\} = \begin{cases} z + \lambda & z < -\lambda \\ 0 & |z| \leq \lambda \\ z - \lambda & z > \lambda \end{cases}$$

In conclusion:

$$\text{Prox}_{\lambda g}(z) = \max(|z| - \lambda, 0) \circ \overset{\text{sign}}{\cancel{\text{sing}}}(z)$$

Exercise 6:

$$\text{Prox}_{\lambda g}(z) = \underset{y \in R^d}{\text{argmin}} \left\{\frac{1}{2}\lambda\|y\|_2^2 + \frac{1}{2}\|y - z\|_2^2\right\}$$

$y \in R^d \quad z \in R^d$

Calculate the gradient ∇y and make it to equal to zero.

For each entry i: we have

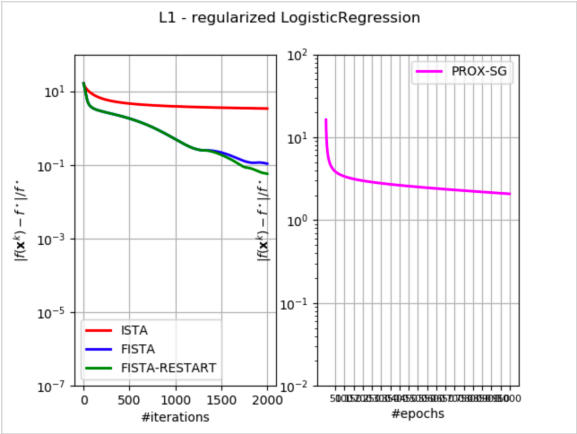
$$\lambda y_i + (y_i - z_i) = 0 \quad \therefore y_i = \frac{z_i}{1 + \lambda}$$

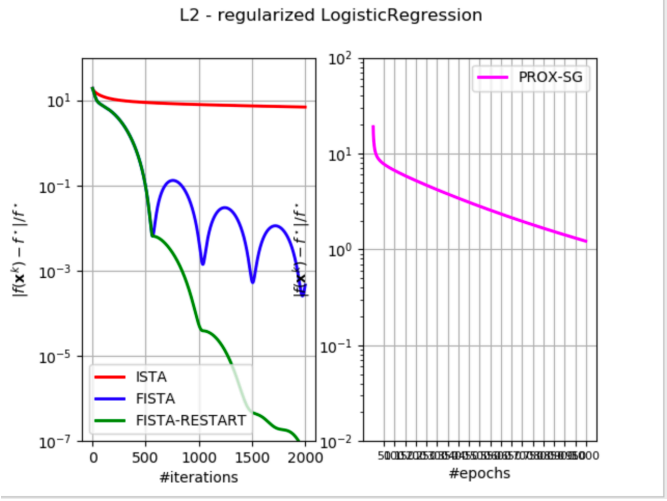
That holds for each entry of vector y and vector z

$$\therefore \text{Prox}_{\lambda g}(z) = \frac{z}{1 + \lambda}$$

Exercise 1, PART 2：

Exercise1.2.2:
the convergence plot as following:



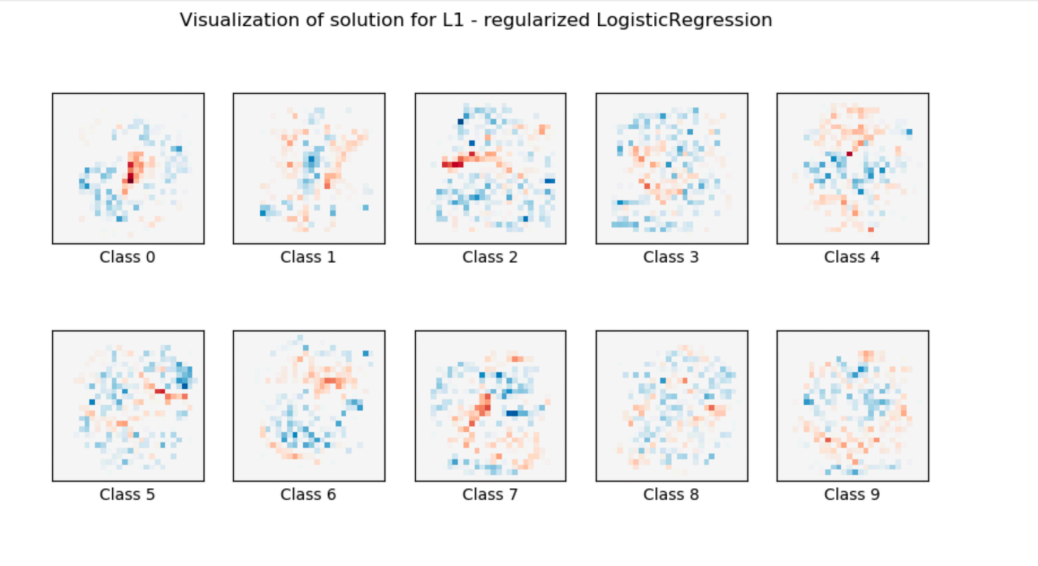


After 2000 iterations, the convergence rate of ISTA, FISTA,FISTA with restart(in l1 regularized) is converges slowly, and the convergence rate of L2-regularized algorithms performs as expected. Compared the Theoretical bound of these methods, their performance can be accepted.

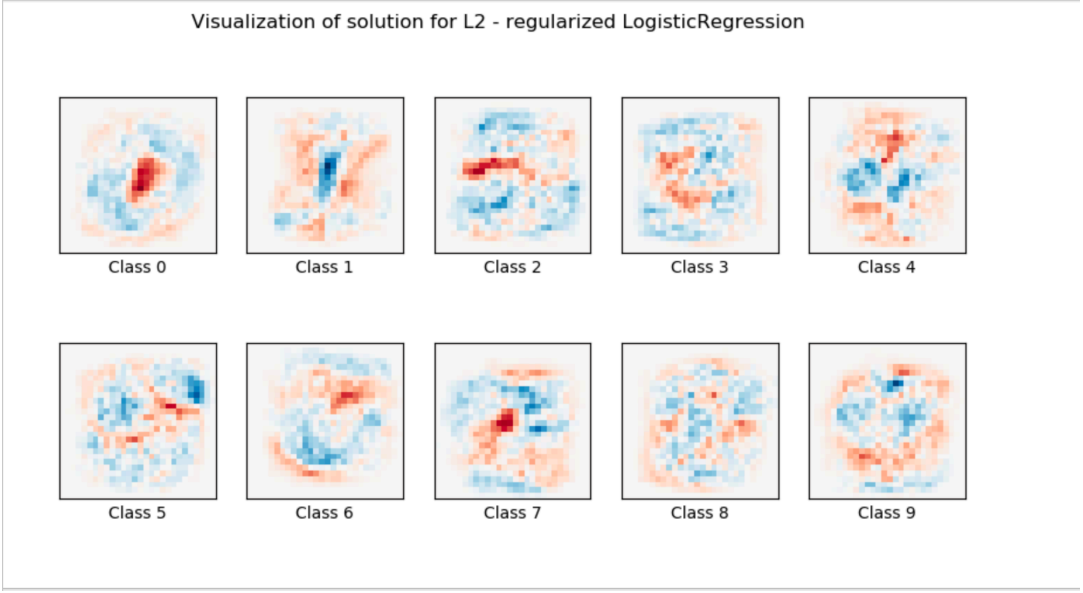
Compare the ISTA,FISTA, FISTA-RESTART, overall, the ISTA converges slower than the FISTA methods, In L1-regularized Logistic Regression, FISTA performs similarly with FISTA-RESTART, but FISTA-RESTART converges faster than FISTA at last. In L2-regularized Logistic Regression, FISTA-RESTART performs better than FISTA, FISTA oscillates after 500 iterations, but with restart method, FISTA-RESTART converges faster, and more stable than FISTA.

For the Prox-SG method, through the graph, the performance of Prox-SG is better than ISTA, but worse than FISTA, FISTA-RESTART, Prox-SG converges slower than FISTA methods, and the FISTA methods are more accurate than Prox-SG.

The visualization of solution for L1-regularized logisticRegression:



The visualization of solution for L2-regularized logisticRegression:



For the MNIST, after the classification of logistic regression , We classify the 10 numbers, I can observe the blue area is similar as the shape of representative number of its class.

Exercise1.2.2:
For the neural net solution, after 17 epoch, the test accuracy is:

```
Epoch  16/ 17
    Train loss: 0.000, accuracy: 100.0%
    Test loss:  0.292, accuracy: 94.7%
    Time: 1.30s
```

For the FISTA with l1 regularized problems, the accuracy is:

```
---------- Training over - PROX-SG. Took 829 seconds.

FISTA-RESTART-l1 accuracy = 89.210000%.
```

For the FISTA with l2 regularized problems, the accuracy is:

```
---------- Training over - PROX-SG. Took 823 seconds.

FISTA-RESTART-l2 accuracy = 89.890000%.
```

Through comparing the accuracies of these two methods, the neural net solution performs better than logistic regression, the neural network is non-linear method, which represented by activation function, NN has better expression power in high dimension space.

Exercise 2.1

(a) the gradient of $f(\alpha)$ in (11):

$\nabla f_{TV}(\alpha) = -W P_n^T (b - P_n W^T \alpha)$

the gradient of $f(x)$ in (12):

$\nabla f_{TV}(x) = -P_n^T (b - P_n x)$

(b) the Lipschitz constant of $\nabla_\alpha f(\alpha)$ in (11):

For α, β in function (11)

$$\|\nabla f_{TV}(\alpha) - \nabla f_{TV}(\beta)\| = \|W P_n^T P_n W^T (\alpha - \beta)\|$$
$$\leq \|W P_n^T P_n W^T\| \|\alpha - \beta\|$$

$\because P_n$ is an operator that select only few $n << p$. from $x \in R^p$

$P_n^T P_n \in R^{p \times p}$

$W W^T = I$

$\therefore \|W P_n^T P_n W^T\| \leq 1$

$\therefore$ the Lipschitz constant of $\nabla_\alpha f(\alpha)$ is 1

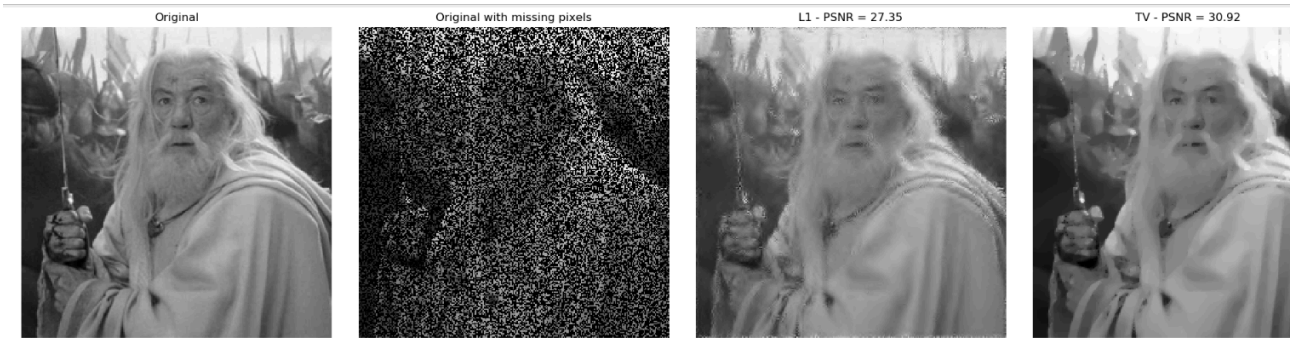
For x, y in function (12)

$$\|\nabla f_{TV}(x) - \nabla f_{TV}(y)\| = \|P_n^T P_n (x - y)\|$$
$$\leq \|P_n^T P_n\| \|x - y\|$$
$$\leq \|x - y\|$$

$\therefore$ the Lipschitz constant of $\nabla_x f_{TV}(x)$ is 1

Exercise 2

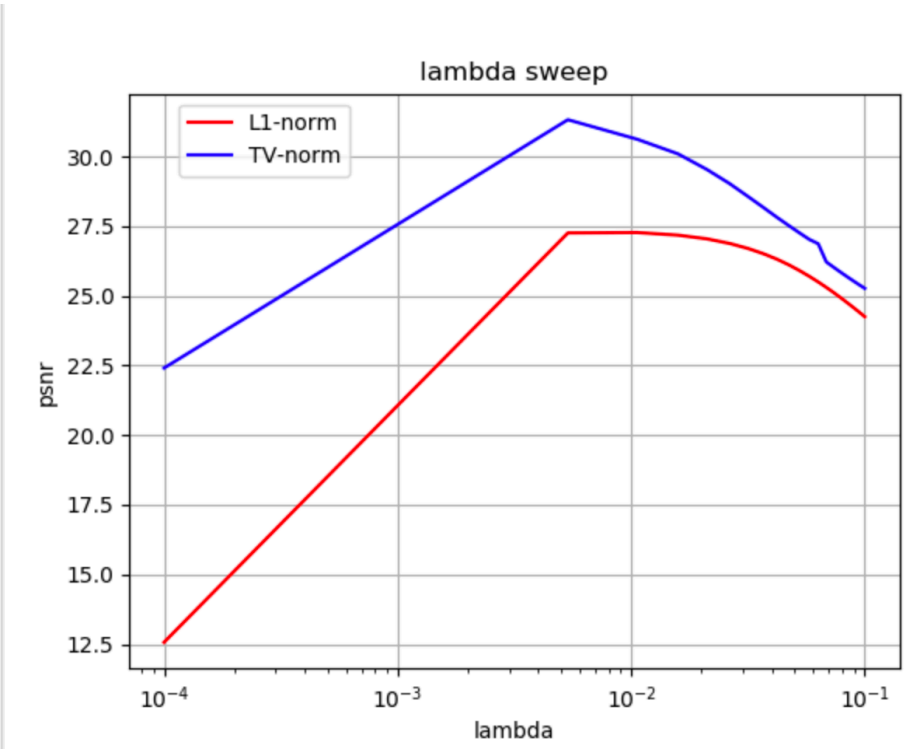
Exercise 2.2
In this exercise, I use the FISTA to reconstruct the pictures, Firstly, I give the outcomes about original, missing pixels, and reconstructed pictures with L1norm, TV norm
Example: the FISTA_RESTART, lambda=0.01



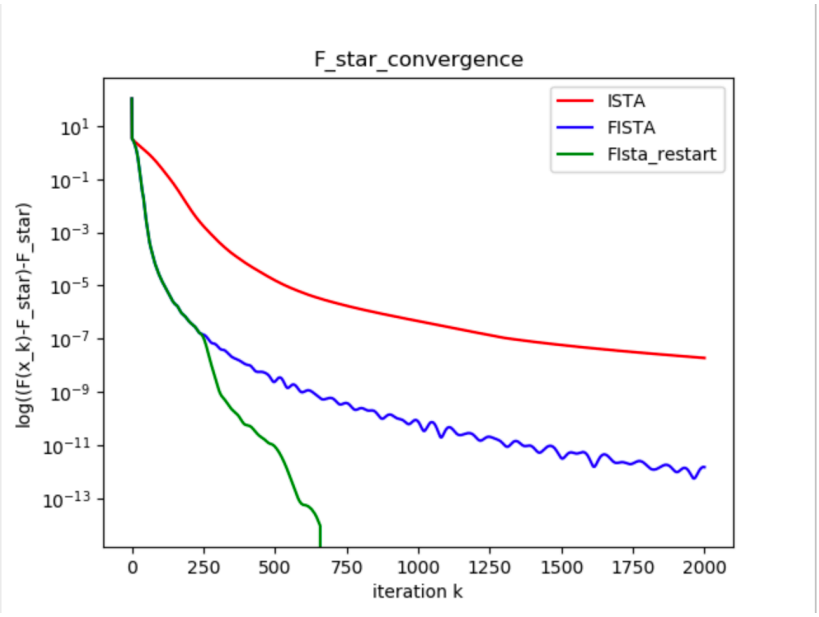
Example:the FISTA, lambda=0.01

The difference between FISTA_RESTART and FISTA is not big.

Then, I choose the parameter lambda values ‘line space’(0.0001,0.1,20) to observe the change of PSNR value, the algorithm is FISTA_RESTART
After 200 iterations:

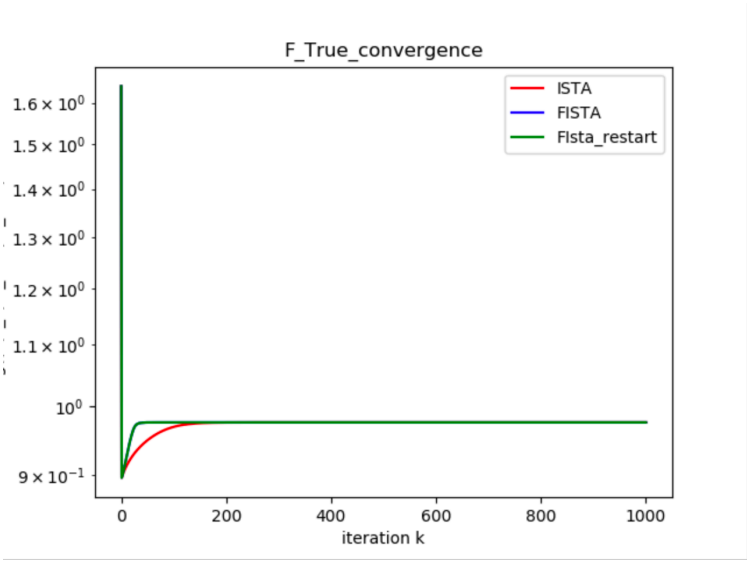


Exercise 2.3
In order to learn the convergence of these algorithms, Firstly, use the optimal solution F_star from the FISTA with restart, and then, plot the convergence graph about ISTA,FISTA,FISTA with restart. (code in inpainting_F_star.py)



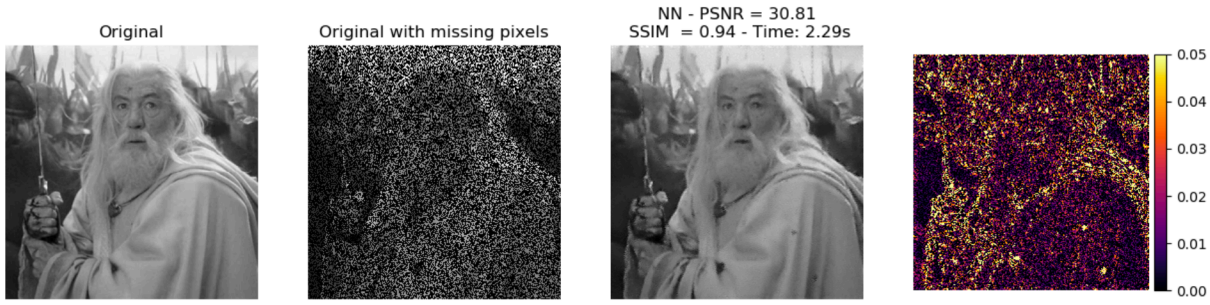
From the graph, I can observe the FISTA with restart convergences faster than other algorithms, and the ISTA convergences slowly, within 2000 iterations, the FISTA methods perform better than the ISTA

Then use the image_truth to learn the convergence of these 3 methods, after 1000 iterations: (code in inpainting_F_true.py)



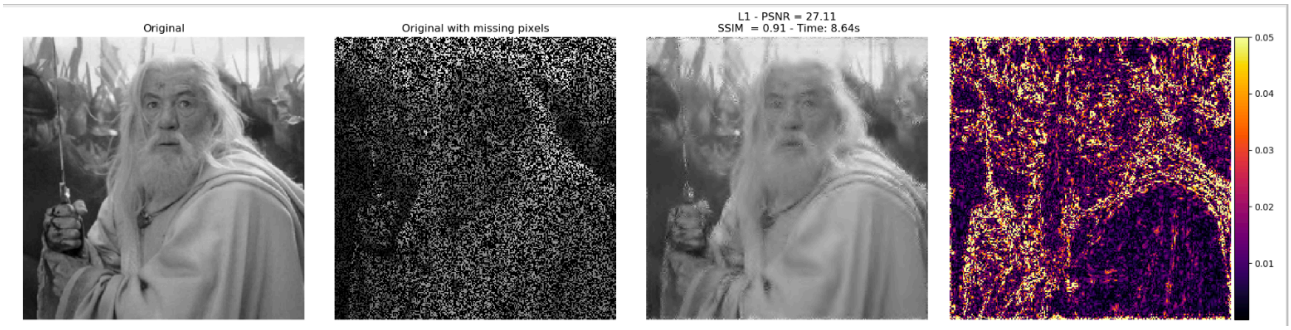
From this graph, I observe the ISTA converges slower than the other 2 methods, and for the convergence of ground truth, the final result is similar.

Exercise 2.4
The result of unrolled method:



The result of FISTA with gradient restart scheme:TV

The result of FISTA with gradient restart scheme:L1



After comparing these three methods, I can observe that the TV minimization is better than the L1minimization, and the NN performs similarly with TV minimization, As for the value SSIM, it represents the similarity of construction of two images, so, the TV and NN method performs better than the L1 method.From the error map, I can observe the error of NN is less than the error of FISTA_RESTART with L1 and TV.

Considering the convergence speed, unrolled NN performs best, and it converges faster than other two methods, TV method converges slowly, it needs more time to get optimal solution.For

me, NN is the best picture, because it uses the less time and performs best in these three methods.